Part No. 070-7386-99
Product Group 47

11401 and 11402
Digitizing Oscilloscopes

Extended Service Manual
Volume 1

WARNING

The following servicing instructions are for use by qualified personnel only. To avoid personal injury, do not perform any servicing unless you are qualified to do so. Refer to the Safety Summary prior to performing any service.

Please check for CHANGE INFORMATION at the rear of this manual.

Serial Number_____

First Printing OCT 1988

Tektronix®
COMMITTED TO EXCELLENCE

Scans
By
Artek Media

Artek Media
1042 Plummer Cir. SW
Rochester, MN 55902

www.artekmedia.com

"High resolution scans of obsolete technical manuals"

If your looking for a quality scanned technical manual in PDF format please visit our WEB site at www.artekmedia.com or drop us an email at manuals@artekmedia.com and we will be happy to email you a current list of the manuals we have available.

If you don't see the manual you need on the list drop us a line anyway we may still be able to point you to other sources. If you have an existing manual you would like scanned please write for details, This can often be done very reasonably in consideration for adding your manual to our library.

Typically the scans in our manuals are done as follows;

1) Typed text pages are typically scanned in black and white at 300 dpi.
2) Photo pages are typically scanned in gray scale mode at 600 dpi
3) Schematic diagram pages are typically scanned in black and white at 600 dpi unless the original manual had colored high lighting (as is the case for some 70's vintage Tektronix manuals).

If you purchased this manual from us (typically through our Ebay name of ArtekMedia) thank you very much. If you received this from a well-meaning "friend" for free we would appreciate your treating this much like you would "share ware". By that we mean a donation of at least $5-10 per manual is appreciated in recognition of the time (a manual can take as much as 40 hours to reproduce, book, link etc.), energy and quality of effort that went into preserving this manual. Donations via PayPal go to: manuals@artekmedia.com or can be mailed to us the address above.

Thanks

Dave & Lynn Henderson
Artek Media

INSTRUMENT SERIAL NUMBERS

Each instrument manufactured by Tektronix has a serial number on a panel insert, tag, or stamped on the chassis. The first letter in the serial number designates the country of manufacture. The last five digits of the serial number are assigned sequentially and are unique to each instrument. Those manufactured in the United States have six unique digits. The country of manufacture is identified as follows:

B010000	Tektronix, Inc. Beaverton, Oregon, USA
G100000	Tektronix Guernsey, Ltd., Channel Islands
E200000	Tektronix United Kingdom, Ltd., London
J300000	Sony/Tektronix, Japan
H700000	Tektronix Holland, NV, Heerenveen, The Netherlands

Instruments manufactured for Tektronix by external vendors outside the United States are assigned a two digit alpha code to identify the country of manufacture (e.g., JP for Japan, HK for Hong Kong, etc.).

Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077

Printed in U.S.A.

Contents

List of Illustrations

List of Tables

Safety Summary

The general safety information in this part of the summary is for both operating and servicing personnel. Specific warnings and cautions will be found throughout the manual where they apply, but may not appear in this summary.

Terms

In This Manual

CAUTION statements identify conditions or practices that could result in damage to the equipment or other property.

WARNING statements identify conditions or practices that could result in personal injury or loss of life.

As Marked on Equipment

CAUTION indicates personal injury hazard not immediately accessible as one reads the marking, or a hazard to property including the equipment itself.

DANGER indicates a personal injury hazard immediately accessible as one reads the marking.

Symbols

In This Manual

Static-Sensitive Devices.

This symbol indicates where applicable cautionary or other information is to be found.

As Marked on Equipment

DANGER – High Voltage.

Protective ground (earth) terminal.

ATTENTION – refer to manual.

Warnings

Power Source

This product is intended to operate from a power source that will not apply more than 250 V rms between the supply conductors or between either supply conductor and ground. A protective ground connection, by way of the grounding conductor in the power cord, is essential for safe operation.

Grounding the Product

This product is grounded through the grounding conductor of the oscilloscope power cord. To avoid electric shock, plug the power cord into a properly wired receptacle before connecting to the product input or output terminal. A protective–ground connection, by way of the grounding conductor in the power cord, is essential for safe operation.

Danger Arising from Loss of Ground

Upon loss of the protective–ground connection, all accessible conductive parts (including knobs and controls that may appear to be insulating), can render an electric shock.

Do Not Operate in Explosive Atmospheres

To avoid explosion, do not operate this product in an atmosphere of explosive gasses.

Do Not Service Alone

Do not perform internal service or adjustment of this product unless another person capable of rendering first aid and resuscitation is present.

Use Care When Servicing with Power On

Dangerous voltages exist at several points in this product. To avoid personal injury, do not touch exposed connections and components while power is on.

Disconnect power before removing protective panels, soldering, or replacing components.

Crt Handling

Use care when handling a crt. Breakage of the crt causes a high–velocity scattering of glass fragments (implosion). Protective clothing and safety glasses should be worn. Avoid striking the crt on any object which might cause it to crack or implode When storing a crt, place it in a protective carton or set it face down in a protected location on a smooth surface with a soft mat under the faceplate.

Use the Proper Fuse

To avoid fire hazard, use only a fuse which is identical in type, voltage rating, and current rating to the fuse specified in the parts list for your product.

Contents

Section 1—General Information

Section 1
General Information

This section gives all the information needed to apply power to the 11401/11402 Digitizing Oscilloscope.

Information on operating voltage and power cord needs, as well as environmental conditions such as operating temperature and ventilation requirements, is included here.

Operating Power Information

This instrument can be operated from either a 115 V or 230 V nominal supply source, 48 to 440 Hz. The 6-ampere, 250 V line fuse is used for both 115 V and 230 V operation.

WARNING

AC POWER SOURCE AND CONNECTION. The 11401/11402 operates from a single-phase power source. It has a three-wire power cord and two-pole, three-terminal grounding type plug. The voltage to ground (earth) from either pole of the power source must not exceed the maximum rated operating voltage, 250 volts.

Before making connection to the power source, check that the 11401/11402 LINE VOLTAGE SELECTOR is set to match the voltage of the power source, and has a suitable two-pole, three-terminal grounding-type plug.

GROUNDING. This instrument is safety Class 1 equipment (IEC designation). All accessible conductive parts are directly connected through the grounding conductor of the power cord to the grounded (earthing) contact of the power plug.

The power input plug must be inserted only in a mating receptacle with a grounding contact where earth ground has been verified by a qualified service person. Do not defeat the grounding connection. Any interruption of the grounding connection can create an electric shock hazard.

For electric shock protection, the grounding connection must be made before making connection to the instrument's input or output terminals.

TABLE 1-1
Power-Cord Conductor Identification

Conductor	Color	Alternate Color
Ungrounded (Line)	Brown	Black
Grounded (Neutral)	Light Blue	White
Grounded (Earthing)	Green/Yellow	Green

Power Cord Information

A power cord with appropriate plug configuration is supplied with each 11401/11402. Table 1-1 gives the color-coding of the conductors in the power cord. If you require a power cord other than the one supplied, refer to Table 1-2, Power-Cord and Plug Identification.

Operating Voltage

The LINE VOLTAGE SELECTOR (located on the rear panel) allows you to select 115 V or 230 V nominal line-voltage operation. The same line fuse is used for both 115 V and 230 V operation.

Memory Backup Power

A self-contained power source within the 11401/11402 allows the retention of volatile memory upon loss of the ac power source. The self-contained power source provides memory backup power which stores the last selected front- and crt touch-panel settings of the mainframe and plug-in units. Waveforms stored in memory are not retained. The self-contained power source also supplies power to the IC that generates the Time/Date parameters, and records the hours of instrument on-time and the number of power-up sequences.

The self-contained power-source has a nominal shelf life of approximately five years. Partial or total loss of stored settings upon power-up may indicate that the power source needs to be replaced.

Operating Environment

The following environmental requirements are provided to ensure proper operation and long instrument life .

Operating Temperature

The 11401/11402 can be operated where the ambient air temperature is between 0° and +50°C and can be stored in ambient temperatures from –40° to +75° C. After storage at temperatures outside the operating limits, allow the chassis to reach the safe operating temperature before applying power.

Enhanced system accuracy is available after a 20-minute warmup period. After entry into Enhanced accuracy, the instrument will revert to Not-enhanced accuracy if the internal mainframe temperature changes more than ±5° C.

Ventilation Requirements

The 11401/11402 is cooled by air drawn in through the side panels of the instrument by the fan and blown out through the rear. To ensure proper cooling of the instrument, allow at least two inches clearance on both sides and the rear of the instrument. The top and bottom of the instrument does not require ventilation clearance.

CAUTION

If air flow is restricted, the instrument's power supply may temporarily shut down.

TABLE 1-2
Power-Cord and Plug Identification

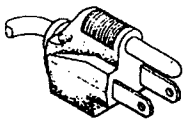
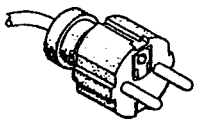
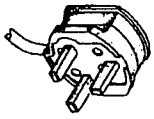
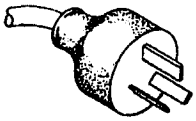
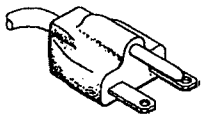
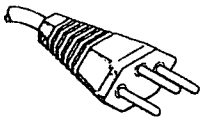
Plug Configuration	Usage (Max Rating)	Reference Standards & Certification	Option #
	North American 125V/6A	[1]ANSI C73.11 [2]NEMA 5-15-P [3]IEC 83 [10]UL [11]CSA	Standard
	European 220V/6A	[4]CEE (7), II,IV,VII [3]IEC 83 [8]VDE [9]SEMKO	A1
	United Kingdom 240V/6A	[5]BSI 1363 [3]IEC 83	A2

TABLE 1-2 (cont)
Power-Cord and Plug Identification

Plug Configuration	Usage (Max Rating)	Reference Standards & Certification	Option #
	Australian 240 V/6A	[6]AS C112 [12]ETSA	A3
	North American 250V/10A	[1]ANSI C73.20 [2]NEMA 6-15-P [3]IEC 83 [10]UL [11]CSA	A4
	Switzerland 240V/6A	[7]SEV	A5

[1]*ANSI—American National Standards Institute*
[2]*NEMA—National Electrical Manufacturer's Association*
[3]*IEC—International Electrotechnical Commission*
[4]*CEE—International Commission on Rules for the Approval of Electrical Equipment*
[5]*BSI—British Standards Institution*
[6]*AS—Standards Association of Australia*
[7]*SEV—Schweizevischer Elektrotechischer Verein*
[8]*VDE—Verband Deutscher Elektrotechniker*
[9]*SEMKO—Swedish Institute for Testing and Approval of Electrical Equipment*
[10]*UL—Underwriters Laboratories*
[11]*CSA—Canadian Standards Association*
[12]*ETSA—Electricity Trust of South Australia*

Contents

Section 2—Theory of Operaation

Section 2
Theory of Operation

This section provides general and specific information on 11401 and 11402 Digitizing Oscilloscope circuitry. The circuitry is grouped into functionally named modules that correspond to a specific circuit board (e.g., Main Processor or CRT Driver boards). In some cases, however, only the circuitry related to that board's function is discussed within that subsection.

The General System Description is presented first. This outlines the oscilloscope and describes its functions at a system level. Next is a Detailed System Description that divides each board into functional blocks. When reading these descriptions, refer to the system block diagram, and the appropriate schematics. The detailed descriptions also provide schematic diamond numbers referencing the appropriate schematic.

Logic Conventions

Digital logic techniques are used to perform most functions within this oscilloscope. Function and operation of the logic circuits are represented by standard logic symbols and terms. All logic functions are described using the positive logic convention. Positive logic is a system of notation whereby the more positive of two levels is the true, or 1 state; and the more negative level is the false, or 0 state. In logic descriptions, the more positive of the two logic voltages is referred to as high, and the more negative state as low. The specific voltages that constitute a high or low state vary between different electronic devices (e.g., ECL and TTL). Active–low signals are indicated in the text by an (L) following the signal name, and by a horizontal line above the signal name when on a schematic. Signal names without indicators are considered active–high. Hexadecimal numbers are identified with a *hex* suffix. Decimal numbers have no radix suffix.

General System Description

The left, right, and center plug–in compartments allow up to three plug–in units to be installed in the 11401/11402 oscilloscope. Plug–in units are used to condition signals prior to delivery to the oscilloscope. These preconditioned signals are passed to the oscilloscope by two paths; 1) the Analog Input, and 2) the Serial Data digital interface.

Analog signals are directed through the Plug-in Interface board to the Acquisition and Time Base boards where they are changed into digital signals. The digitized signals are then stored in the Memory board RAMs. The Compressor board circuit extracts the stored digitized signals from the Memory board RAMs and conditions them for the Display controller board circuit, where they are processed by the CRT Driver board and presented on the CRT. The Compressor accomplishes this by squeezing the waveform into 500 points, which happens to be the fixed length (maximum number) of points that the Display Controller circuit can represent on the CRT screen.

Digital signals pass through the Serial Data digital interface which connects the Plug-in Interface board, via the Standard I/O board, to the Main Processor. The Serial Data bus is used for bi–directional control between plug–ins and the Main Processor, and for data communications.

Detailed Circuit Description

The Main Processor communicates with the various circuit blocks through parallel Data and Address interface busses. The Serial Interface bus that connects the Display Controller circuit to the Rear Panel board is used for diagnostic purposes only.

Refer to the schematic diagrams while reading this Detailed Circuit Description. Complete schematic diagrams are provided in this manual.

Each diagram is divided into stages that are bordered with wide shaded lines. The Detailed System Description subheadings provide schematic diamond numbers that reference you back to the appropriate schematic.

Assembly numbers (A numbers) identify components and the board these components are mounted on. The designation A7R200 identifies the board as assembly A7 (the Display Controller board), and the component number as resistor R200.

A2 Line Inverter, A3 Rectifiers and Control <39>

The Line Inverter, Rectifiers, and Control circuits provide semi-regulated DC voltages to the Regulators circuitry. A schematic diagram of the Line Inverter, Rectifiers, and Control circuitry is given on diagram 39, in Section 6, Schematic Diagrams and Circuit board Illustrations in Volume 2 of the service manual. The schematic is divided by gray shaded lines separating the circuitry into major stages.

Line Interface

Power is applied through line filter FL99, line fuse F99 and PRINCIPAL POWER SWITCH S130. The line filter FL99 prevents power-line interference from entering the oscilloscope, and noise (in the range of 1 MHz to 1 GHz) generated within the oscilloscope from entering the line.

Resistor R99 serves to discharge the line capacitance in front of the line frequency bridge rectifiers and prevents a shock hazard from contact with the power connector pins when the power cord is disconnected.

The primary line fuse F99 prevents a fire hazard resulting from an improper setting of the 115/230 line switch, or from a major fault in the line-side circuitry. The thermal cutout switch S99 will disconnect the AC line power to the Line Interface circuitry when the heat sink temperature reaches approximately +75°C.

The PRINCIPAL POWER SWITCH S130 disconnects both sides of the line. It is located on the rear panel and is not intended to be used as the primary means of turning the oscilloscope on and off, as this function is handled by the ON/STANDBY switch, located on the front panel.

Capacitors, C140 and C230, serve to bypass differential-mode noise, generated by the reverse recovery of the line-frequency rectifiers. Thermistors RT130 and RT240 limit the initial surge current when charging the line filter caps C200, C310, C200, C220, and C320. The thermal time constant of these thermistors is matched to the discharge time of the line filter caps, through bleeder resistor, R220. This ensures continued surge current limiting when the oscilloscope is switched off and on several times in succession. Once the thermistors warm up they have negligible affect on the input current.

Spark gaps E231 and E230 prevent over charging the line filter caps C200, C310, C200, C220, and C320 due to improper setting of the 115/230 switch S250 or from high energy differential-mode line transients. CR340 operates as a

full–wave rectifier when S250 is set for 230 V operation, and as a voltage doubler, when S250 is set for 115 V operation, thus maintaining 230 to 380 V DC across R220, for either setting of S250.

A neon flasher consisting of R640, C640 and DS640 indicates that hazardous voltages are present on the line filter caps. R220 discharges these capacitors in about two minutes. Transformer T440 provides the line trigger signal and power for the Standby Power circuitry, when the front panel ON/STANDBY switch is off and the rear panel PRINCIPAL POWER SWITCH is on.

Capacitors C400 and C401 provide a return path for common–mode current transients flowing in the chassis, caused by stray capacitive–coupling of the switching waveforms to the chassis. These capacitors also lower the impedance of the common–mode noise sources within the oscilloscope and power supply. T410 further decouples these noise sources. The leakage inductance of transformer T410 (appearing as a differential–mode inductance to the Pulse Width Modulator), in conjunction with C500, forces a continuous DC current to be drawn from the line storage caps C200, C310, C220, and C320.

Pulse Width Modulator

The primary means of controlling the output voltages from the Low–Voltage Rectifiers is the Pulse Width Modulator (PWM). The PWM transistors Q600 and Q601 are power MOSFET devices. They, in conjunction with L520 and CR500, form a negative buck switching regulator operating at a fixed frequency of 100 kHz. By controlling the conduction time of either transistor Q600 or Q601 the DC voltage applied to the 50 kHz Inverter (across C630 and C631) can be varied from zero volts up to the maximum voltage available across the line filter caps C200, C220, C310, and C320. Each of the PWM transistors conducts current on alternate switching cycles; they never conduct current at the same time. The control voltages for the gates of the PWM transistors are provided by T710. Resistors R600 and R610, and diodes CR600 and CR610 set the switching speed of the PWM transistors.

The 100 kHz variable–duty square wave developed at the drains of the PWM transistors is filtered by L520, C630 and C631. CR620 provides a path for the continuous DC current flowing in L520 when neither PWM transistor is conducting current.

Primary Current Sense

The primary current–sense transformer T700 samples the instantaneous current flowing in each PWM transistor. Q801 and Q800 clamp the secondaries of T700 to one base–emitter drop, giving true current–transforming operation. The current sense signal, +CS, is used to control the current flowing in the PWM transistors. It is used for both protection and feedback control.

50 kHz Inverter

The primary function of the inverter is to convert the DC voltage provided by the Pulse Width Modulator into a 50 kHz square wave. The 50 kHz Inverter MOSFET devices Q620 and Q610 are driven by a 10 V peak, 50 kHz square wave from the gate drive transformer T720. Cross conduction (i.e., both transistors conducting at the same time) is prevented by R620, R611, CR621, and CR611 working into the gate–source capacitance of Q620 and Q601. During the 100 ns when neither transistor is conducting, diodes CR601 and CR620 provide a path for current from mutual inductances and leakage of the transformer. When the transformer primary current decays to zero, the respective MOSFET is biased on, ready to conduct. Thus no switching losses are associated with the half–wave bridge.

In the event of an inverter overvoltage condition, spark gap E630 will fire, clamping the inverter input voltage to a safe level. If the overvoltage is caused by a shorted PWM transistor, the spark gap will continue to conduct current

until fuse F410 is opened and the inverter capacitors C630 and C631 discharge. If the overvoltage condition is caused by an open feedback path, the control circuit will limit the current to a level below that required to clear F410, and will then initiate a restart cycle, extinguishing the spark gap.

Rectifiers

Four separate sets of power transformer windings in T130 and T140 step the 50 kHz Inverter voltage down to the level required to generate eight semi–regulated outputs: 154, 117, 17, 15.2. Both power transformers operate in parallel but have different volt–second/turn ratio values in order to establish the proper output voltages.

After rectification and filtering, the semi–regulated outputs pass through current sense resistors, which are actually ECB (etched circuit board) traces tapped at specific lengths. The voltage across these traces is proportional to the current at the output terminals. These voltages (approximately 25 mV at the rated current) are used by the control circuit to limit fault currents in the oscilloscope plug–in units and to provide a diagnostic feature. The exception to this type of sensing is the 154 V outputs, whose currents are sensed by R535, R636 and R637. The current sense signal is then level–shifted by Q530 and Q630.

Inverter Voltage Sampler

Components CR250, CR251, C251, R240, and Q250 form a sample–and–hold circuit that provides a voltage proportional to the inverter input voltage. This voltage provides feedback to the Error Amplifier circuit.

Ramp Injection

The Ramp Injection circuit consists of Q210 and associated components. The ramp signal is generated by charging C216 through R313. This voltage ramp is converted to a current by Q210, with R312 setting the magnitude of the injected current and, therefore, the current loop gain. The ramp generator is reset periodically by either the 100 kHz clock pulse or by the end of the PWM "ON" time, whichever comes first. (Normally the PWM signal, from U200A, pin 1 acting through CR200 and Q213 occurs first.) CR311 prevents the ramp capacitor C216 from discharging completely, leaving a small DC bias current in the ramp signal to hold off the converter during the power–up sequence.

Current Limit Comparator

The primary current sense signal plus the injected ramp current are summed and converted to a voltage by R305. The voltage across R305 (proportional to the instantaneous current in the PWM switching transistors) is applied to the current–limit comparator U410C. When the current in the PWM inductor L520 rises to a level determined by the error signal, the comparator resets the latch U200A ending the switching cycle. A new switching cycle is initiated by the 100 kHz clock pulse which sets the latch, turning "on" the alternate PWM transistor, until it is again reset by the current limit comparator. The error signal at the non–inverting input to the current–limit comparator regulates the magnitude of current flowing in the PWM filter inductor L520 by controlling the on time of the PWM power switching transistors.

Error Amplifier

The +5.1 S and +5 I error signals are summed by the Error Amplifier U400B. The output terminal voltage is set by the +5.2 DC REF voltage, which is generated by U800 in the Local Power circuit and adjusted with potentiometer R800, 5.2 REF ADJ.

Soft Start

Shutdown of the converter is accomplished by ramping down the reference voltage input to the Error Amplifier. The RST(L) signal discharges the soft start capacitor C411 through CR313, Q410 and R324. All supplies will track this signal down to zero, where they will stay until the Fault Delay Latch (diagram 40) times out. The RST(L) signal also pulls down the PFSET line, latching the PWRUP signal low and turning the Fault LEDs on for the duration of the time out.

After the 200 ms time out of the Fault Delay Latch, the clamp is removed from the 5.2 V REF input to the Error Amplifier allowing the output voltage to rise at the rate determined by R323 charging the softstart capacitor C411. The output voltages will track this rise, which has a time constant of approximately 10 ms. After a delay of 20 ms, as determined by the Power Fail Detect clamping network of R326, C323, CR321 diagram 40, the power fail detect comparator U420D is reset, which turns off the LEDs and signals the oscilloscope that the output voltages are stable by activating PWRUP.

Standby Power

Components CR750 and C850 rectify and filter the secondary voltage of transformer T440 providing 17–30 V to power the remote line switch and standby control. This voltage is regulated to 16 V by a series–pass regulator composed of Q840, Q740, VR830, and associated components. Zener VR830 and divider R832–R833 are the positive and negative inputs to differential amplifier Q740. The output of the differential amplifier is the error voltage, and is applied to the base of the PNP series pass transistor Q840. Stability compensation is provided by C730.

When the power conversion circuitry is started, power is supplied by the inverter through the +17 V line, acting through CR630. This shuts off Q840, unloading the line trigger transformer T440, which would otherwise cause distortion of the Line Trigger output (LTRIG).

Power is applied to the control circuit by grounding the DCPWRSW line, which turns on Q100. This is done by the remote power switch located on the front panel. Three–terminal regulator U300 sets the control circuit voltage to +12 V.

100 kHz Clock

When power is first applied the fault delay latch is set, preventing startup of the power conversion circuitry. Meanwhile, the 100 kHz clock pulse generator, made up of the comparator U410D and associated components, is allowed to stabilize. The frequency of the clock is set by R300 charging C301 to a voltage set by the divider R316 and R301. The pulse duration is set by C312.

Gate Drive

The 100 kHz clock pulse toggles the divide–by–two flip–flop U200B. This provides the balanced 50 kHz drive signal for the 50 kHz Inverter. U100 buffers both the PWM and 50 kHz Inverter signals, providing sufficient current to drive the gate transformers as well as the Local Power charge pumps formed by CR110, CR111, CR112, CR113 and C212, C213, C214 and C215. These charge pumps establish the +22 V and P10 V supplies. The input logic in Gate Drive U200 provides the alternating drive required by the PWM gate transformer. Damping and blocking are provided by R112–R111 and C120–C121, respectively. R201 and C200 create a time delay that holds the PWM latch U200A reset, when power is first applied.

Line Trigger

The line trigger signal is applied to the differential–to–single–ended amplifier U810A via the 1U line signals. R730 and R736 set the output impedance and magnitude of the line trigger signal, LTRIG, (used by the oscilloscope and plug–in units) to less than 470 = and from 1 V to 3 V P–P, respectively.

A2A2 Control Rectifier Board <40>

Fault Detection

The Fault Detection circuit provides protection and diagnostics for the Regulators circuitry and the Line Inverter, Rectifiers, and Control circuitry. The purpose of the protection circuitry is to prevent single faults in the power supply or oscilloscope power busses from becoming multiple faults. It does this by forcing an orderly shutdown of the power conversion circuitry, followed by restart attempts at regular intervals. Since all fault conditions have the same result, (i.e., shutdown of the converter) diagnostic features have been added to assist in locating the source and cause of the fault condition. A fault is identified as a persistent condition of excessive current, voltage or temperature in the power supply, oscilloscope, plug–in units or accessories.

A schematic diagram of the Fault Detection circuitry is given on diagram 40, in Section 6, Schematic Diagrams and Circuit board Illustrations in Volume 2 of the service manual. The schematic is divided by gray shaded lines separating the circuitry into major stages.

Digital Current Sense

The overload fault sequence begins when the voltage drop across one of the Rectifiers current–sense trace resistances exceeds a reference level (about 25 mV) for several switching cycles (20 μs), tripping one of the current sense comparators (U610C, U610A, U610D, U610B, U410A, U620D, and U620B). The reference levels for the comparators are set by zener diode voltage divider networks (VR630, R635 and R630 is one example) which establishes the reference and, therefore, the current limit point for the +5.1 V outputs.

Once tripped, the first action of a comparator is to pull down the PFSET (Power Fail Set) line, which is an analog "OR" of all the comparator outputs. This signal trips the Power Fail Detect comparator U420D turning on Q730, which pulls the PWRUP line down, signaling to the oscilloscope that a power failure is imminent. This signal also turns on the Fault LED Driver Q430, supplying current to all diagnostic LEDs via the LIGHTS(L) line. Meanwhile, the current sense comparator signal fires one of the PUT devices (for example Q721) which bypasses the LED current, extinguishing only the LED associated with the particular fault condition, while all other LEDs remain on.

Tripping the current sense comparator also begins a time delay, determined individually for the various outputs by discharging capacitor C324 through the individual resistors in the fault line (for example R513). During this delay, the LEDs are lit, the PWRUP signal is low and the power conversion circuitry attempts to clear the fault by providing maximum available power to the load. This time delay is approximately 0.1 ms allowing time to charge capacitors during transient loads. If the fault is cleared within this time, the LEDs are extinguished and the PWRUP line is brought high, resuming normal operation without disturbing the output voltages.

Analog Current Sense

The Analog Current Sense circuit operates in the same manner as described for the Digital Current Sense circuit.

Primary Current Limit Detect

The primary current is limited on a cycle–by–cycle basis by simply clamping the output of the Error Amplifier. VR300 does this by setting the peak current in the switching transistors. The zener clamp current flows through R303, turning on Q400, which, after a 10 ms delay, trips the Power Fail Detect and Fault Delay Latch initiating a converter shutdown.

Fault Delay Latch

If a fault persists, the fault Delay Latch, formed by C324 and the comparator in U420C, is set, pulling down the restart line, RST(L). This signal initiates shutdown of the power conversion circuitry, which will remain off for the duration (approximately 200 ms) determined by C325, and C324 in series and R322 and R324 in parallel.

Fault LED Driver

This circuit completes the current path through Fault LED's DS533, DS630, DS530, and DS531. Transistors Q720, Q721, Q432, and Q431, respectively, determine which LEDs are illuminated.

Digital Voltage Sense

The Digital Voltage Sense circuitry detects over voltage and under voltage faults. The +5 and P5 V supplies are sensed by VR700 and VR704, respectively. Exceeding the zener voltages causes Q700 and Q701 to turn on, tripping off the DIGVF indicator, DS533. A voltage fault of the post–regulated analog supplies is sensed by the Regulator circuit Voltage Fault Detect, whose output is ORed with the Fault line by CR722 and CR721 to generate a restart.

Thermal Fault Sense

When inadequate air flow causes the thermal cutout switch S99 to reach 75°C, an over–temperature condition occurs interrupting the DC input to the PWM. The converter shutdown will continue until the sensor has cooled to 50° C, only then initiating a restart cycle.

Power Fail Detect

Power fail detection is accomplished by monitoring the duty cycle of the pulse width modulator (PWM). The PWM(L) signal is converted to a DC voltage by the integrator formed by R328, R320, R327, C320 and C321. Since the input to the integrator is an inverted PWM signal, the DC output of the integrator is directly proportional to the rectified AC line. When this voltage drops below a level set by divider R330 and R329 (corresponding to a duty cycle of 95%), the comparator U420D pulls down on the PWRUP signal, and lights all diagnostic LEDs. Power conversion continues until an under voltage condition is detected, giving the oscilloscope time to prepare for a power failure. Once the rectified line voltage has dropped below the minimum regulation range (190 V DC), the converter is shut down by the under voltage detect circuit leaving a substantial stored charge in the line storage capacitors, thus preventing large surge currents when the AC line is reapplied with the thermistors still hot.

Fan Speed Controller

Transistor Q640 and associated components comprise the fan speed modulator. A 10 Hz pulse–width modulated signal from the fan speed controller (FAN PWM) sets the DC fan speed by pulsing it on, then allowing it to freewheel through diode CR740. Fuse F740 protects these devices from any faults on the fan circuit.

The fan speed controller, made up of U730, U710B, and associated components, functions as a limited range feedback control system, which attempts to keep the temperature of the exit air stream constant over the specified operating range of ambient temperatures. The exit air temperature is sensed by U730, and is converted to a DC voltage (VTEMP) by R732. This voltage becomes the reference for a 10 Hz pulse width modulator formed by the operational amplifier U710B. The frequency of oscillation is set by C821, which, with R728, serves as an integrator, converting the fan output back into a DC voltage, which is compared to VTEMP. The action of the circuit is to maintain the two voltages at the same average DC level. When a temperature rise is sensed by U710B, the pulse width to the fan is increased, speeding it up, which in turn reduces the temperature of the exit air, maintaining closed–loop control.

A4 Regulators <41>

Local Regulator Power

The operational amplifiers used for the +50, +15, +5, P50, P15 and P5 V Regulators require that the following special voltages be generated for their operation:

1. The +20 V supply is generated from the semi-regulated +54 V supply by reference zener diode VR732.
2. The P20 V supply is generated from the semi-regulated P54 V supply by reference zener diode VR720 and transistor Q820.
3. The +10 V supply is generated from the semi-regulated +54 V supply, by zener diode VR730.
4. The P10 V supply is generated from the semi-regulated P54 V supply by zener diode VR731.
5. The +10.0 REF is used as a reference voltage.

+50 V Regulator

Semi-regulated +54 V from the Line Inverter, Rectifier and Control circuit (diagram 39) is the unregulated voltage source for this supply. Differential amplifier U220C compares the feedback voltage at pin 9 against the reference voltage at pin 10. The error output at pin 8 of U220C reflects a difference between these two inputs. A sample of the +50 V output is connected to U220C pin 9 via divider network R241 and R235. Notice that the feedback voltage of this divider is obtained from a line labeled +50 S (sense). If the feedback voltages were obtained at the supply, the voltage at the load would not stay constant, due to the voltage drop across the resistance of the cable between the supply and its load. The separate sense line overcomes this problem by sensing the voltage at the load. Because the current in the sense line is small and constant, the load voltage is held constant regardless of the load current.

Regulation of voltage occurs as follows: If the +50 V Regulator output decreases (becomes less positive) due to an increase in load or a decrease in input voltage (as a result of line-voltage change or ripple), the voltage across divider R235 and R241 also decreases. This results in a less positive level, at pin 9 of U220C, than that established by the +10.0 REF supply at pin 10 of U220C. This decreases the current through VR510 causing an increase in current through the base-emitter junction of Q501. The result is increased conduction of Q501, the series regulator device. The load current increases and, therefore, the voltage across the load also increases sufficiently to balance the input to the differential amplifier U220C. The REF ADJ, R830, on the +10 Ref supply sets the output level of this supply.

–15 V Regulator

Basic operation of in the –15 V Regulator is the same as for the +50 V Regulator. The reference level for this supply is established through R130 at pin 12 of U230D. The divider ratio of R122 and R135 sets a level of zero volts at pin 13 of U230D. Any change at the output of the –15 V supply appears at pin 13 of U230D as an error signal. The output voltage is regulated in the same manner as described for the +50 V Regulator. Diode CR433 will keep the output of this supply from going more positive than about –4.4 V if it gets shorted to one of the more positive supplies.

+5 V Regulator

The operation of the +5 V Regulator is basically the same as described for the previous supply regulators. Error voltage is provided through R134 to pin 6 of U230B, and pin 5 is referenced to the +10.0 REF supply. The divider

ratio of R138 and R139 is 2:1, so pin 5 of U230B is at 5 V when the supply is operating normally. Any change at the output of the +5 V Regulator supply appears at pin 6 of U230B as an error signal. Diode CR431 limits the output of this supply to about P0.6 V, if it gets shorted to one of the negative supplies.

+15 V Regulator

The +15 V Regulator operates in the same manner as the +50 V Regulator. Error feedback voltage to pin 13 of U220D is provided through R236. Pin 12 of U220D is referenced to the +10.0 REF supply. The divider ratio of R237 and R236 sets pin 13 of U220D at +15 V. Any change in the output level of the +15 V Regulator appears at pin 13 of U220D as an error signal. This results in an opposite change at the output, pin 14 of U220D, which is conveyed to the series regulator transistor Q400, through Q311, to correct the error in the output voltage of the supply. Diode CR420 limits the output of this supply to about 4.6 V if it gets shorted to one of the negative supplies.

–50 V Regulator

Operation of the –50 V Regulator is basically the same as described for the +50 V Regulator. Error voltage to pin 9 of U230C is provided by divider R224-R223 and is referenced to the P50 S (sense) line. The divider ratio of R224 and R223 sets the level at pin 9 of U230C to zero volts when the output of this supply is correct. Protection diode CR432 limits the output voltage of this supply to P14.4 V should the supply be shorted to a more positive supply.

Voltage Fault Detect

Over-voltage or under-voltage of any regulated supply is detected by the window comparators U220A, U220B, and associated resistors. These resistors set a hysteresis window that is 5% of the regulator sense line voltages. The output of the comparators is analog "ORed" and sent to the Regulator Fault Indicator circuitry on the Control Rectifier board, where the signal is latched into an LED indicator and initiates an immediate shutdown of the entire power supply.

A5 Acquisition Board <6> through <20>

Sampling Hybrid/Vertical Channel Select (diagram 6)

The Sampling Hybrid contains three sampling modules and a four input channel switch. The Sampling Hybrid simultaneously samples the vertical signal from each plug-in compartment. The differential outputs of the three samplers are connected to the Channel Switch IC. The Channel Switch IC converts the differential sampled signal to a single-ended signal and provides a means to select each sampled signal for output to the A/D converter. A calibration voltage (CHSWREF) used for A/D auto-calibration is also connected to the fourth channel switch input.

The Time Base board provides signals (VS0, VS1) to the Vertical Channel Select circuitry that control the Channel Switch output.

Strobe Driver/Sample Gate Generator (diagram 6)

The Strobe Driver IC provides the differential sampling strobes to the Sampling Hybrid. The Sample Gate generator creates the sampling pulse from the falling edge of SAMPLECK (20 MHz). SAMPLE_ENABLE prevents sample pulses from occurring while the A/D converter sequentially digitizes the simultaneously sampled plug-in signals.

A/D Converter (diagram 8)

The A/D converter converts the sampled signal to a 10-bit digital value. The A/D converter is a two-stage flash with error correction. The output of the first stage 5-bit A/D converter (U430) is converted to an analog voltage by the 5-bit DAC (U442) and subtracted from the sampled input signal by the Difference Amplifier (U443). The output of the Difference Amplifier is applied to the second stage which uses two stacked 5-bit A/D converters (U350, U552) to yield a 6-bit A/D converter. The upper and lower 16 codes of the second stage are used for error correction that can correct for errors in the first stage conversion. The Error Correction PROM corrects the LSB of the first stage when an error is sensed by the second stage. The Over/Under Range Logic (U266) defeats error correction when the magnitude of the input signal is greater than full scale.

Clock Generator (diagram 7)

The Clock generator circuitry receives the 200 MHz clock from the PLL and uses a twisted ring shift register counter (U810, U712 and U710) to generate the 20 MHz clock. Each successive tap of the shift register generates a 20 MHz clock that is delayed 5 ns.

The Clock generator provides 20 MHz clocks to the Sample Gate generator and the first stage, second stage and output latches of the A/D converter. Also provided are the 20 MHz clocks for the Coarse Time Interpolator, the serial data out shift register of the Fine Time Interpolator and the Master Clock (MCK) for the Time Base board. Figure 2-1 illustrates the clocks generated for the sampler and A/D.

Trigger (diagram 10)

The Trigger Hybrid (U1710) receives and processes analog signals from the left, center and right plug-in compartments. Two identical ICs within the Trigger Hybrid output trigger gate signals which are used by the Main and Window holdoff, time interpolation and acquisition circuits.

The microprocessor serially loads data to on-board shift registers which control the selection of coupling, slope and free run via U1710-48. A separate serial load clock is provided for the Main and Window shift registers at U1710-23 and U1710-47 respectively. Main and Window trigger source data is serially loaded and latched into U2030.

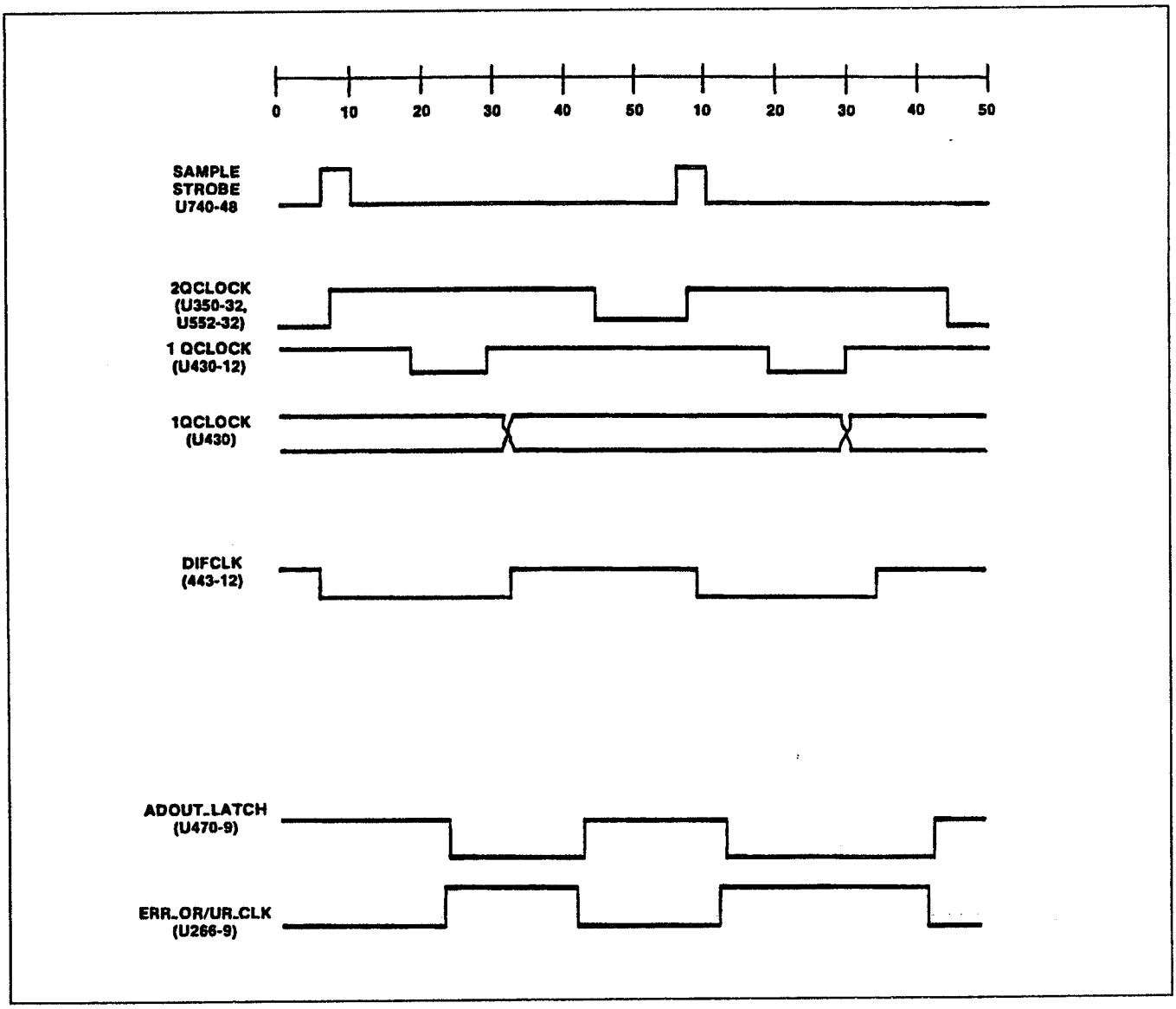


Figure 2-1. Sampler and A/D Clocks

Main Trigger Holdoff (diagrams 10 and 11)

The Main Trigger Holdoff circuitry limits the rate at which trigger gate signals can be generated and provides the user a means to trigger on the desired portion of a complex signal. Figure 2-2 illustrates the interaction between the holdoff signal and the Main Trigger Gate (TGM). The Trigger Hybrid (U1710) drives the Main Trigger Gate output high when the input signal crosses the user selected trigger level input. The Trigger gate remains high until the Main Holdoff circuit times out and then resets the Main Trigger gate. Trigger signals are rejected by the Trigger IC when the Trigger Gate is high or MTHO is high.

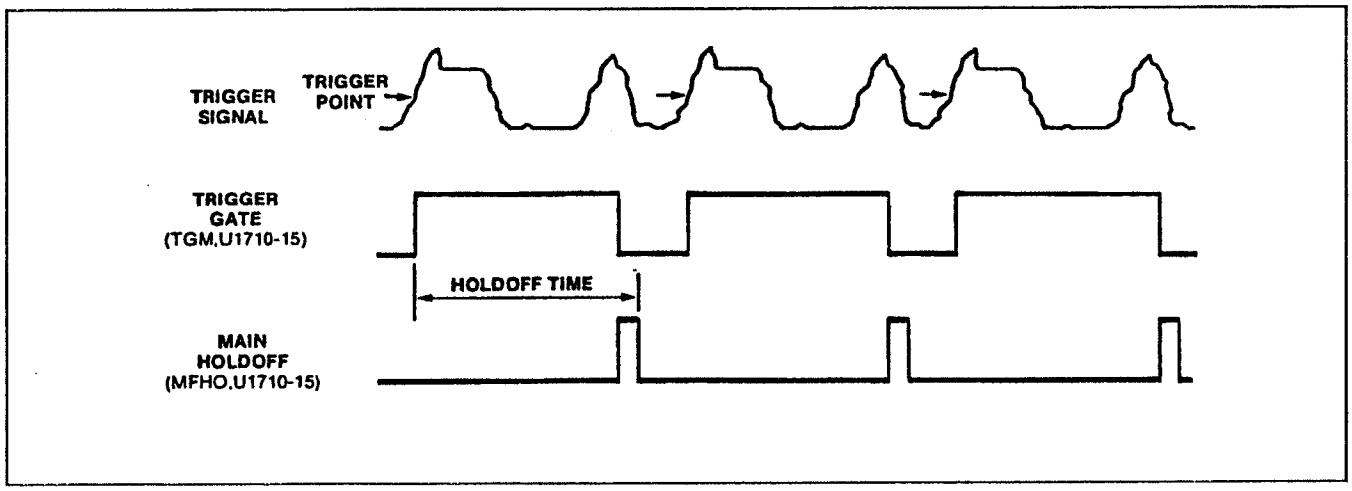


Figure 2-2. Main Trigger Gate and Holdoff Timing

The Main Holdoff circuitry is comprised of a coarse and a fine section. The coarse section contains a digital counter (U264), and and circuitry to generate the holdoff pulse. The coarse holdoff circuits are clocked from the 19.6608 MHz clock.

The Trigger Gate is synchronized to the 19.6608 MHz clock via U1546A and U1546B. Upon receipt of this synchronized trigger gate (U1546-9), the digital counter begins counting down the holdoff interval. When the digital count is completed, MHORESET(L) is asserted low which places the counter in the load state and also starts the holdoff pulse. The holdoff pulse (HOPULSE) is generated by a the Holdoff pulse shift register comprised of U1550 and U1644. The input to the shift register (U1550-3) is set to a high for normal operation. Thus following the reset of the shift register by MHORESET(L), a high is clocked through to the output by the 19.6608 MHz clock. When the holdoff interval is longer than 500 μs, the falling edge of HOPULSE resets U1832B and causes MFHO (U1832-9) to go low and thereby enables the Trigger IC to produce another trigger gate.

The first tap of the Holdoff pulse shift register enables circuitry (U1352, U1552, U1554B) which creates signals that arm the events counter, the record trigger and window holdoff circuits.

Main Fine Holdoff (diagram 11)

The Main Fine Holdoff circuitry is used in conjunction with the digital holdoff to increase the main holdoff resolution to minimum of 500 ps. Analog ramps are used to remove the uncertainty between the trigger and digital holdoff clock.

A positive going analog voltage ramp begins when the Main Trigger occurs and stops on the second holdoff clock edge following the trigger.

Since the holdoff clock occurs randomly with respect to the trigger, the time that the analog ramps runs will vary. Therefore the voltage at which the analog ramp stops is a measure of the time between the trigger and the holdoff clock.

The voltage point where the ramp stops is stored on a capacitor until the digital holdoff count is complete and then the ramp begins again. The ramp runs until it crosses the Main Ramp Threshold voltage, which is present on pin 2 of U1940A. The voltage ramp is allowed to run long enough to make up for propagation delay variations in the digital holdoff circuitry. U1940A makes a positive transition when the ramp crosses the threshold voltage. The Main Ramp Threshold voltage is controlled by the Trigger Enhancement board, A5A2.

Another analog voltage ramp begins at the positive transition of the voltage comparator and the first ramp is reset to its start position.

The second voltage ramp runs until it crosses the comparison voltage set on a second voltage comparator (U1940B). This comparison voltage is varied via software control to allow the positive transition of U1940B to vary up to one holdoff cycle (50.86 ns) in 500 ps steps.

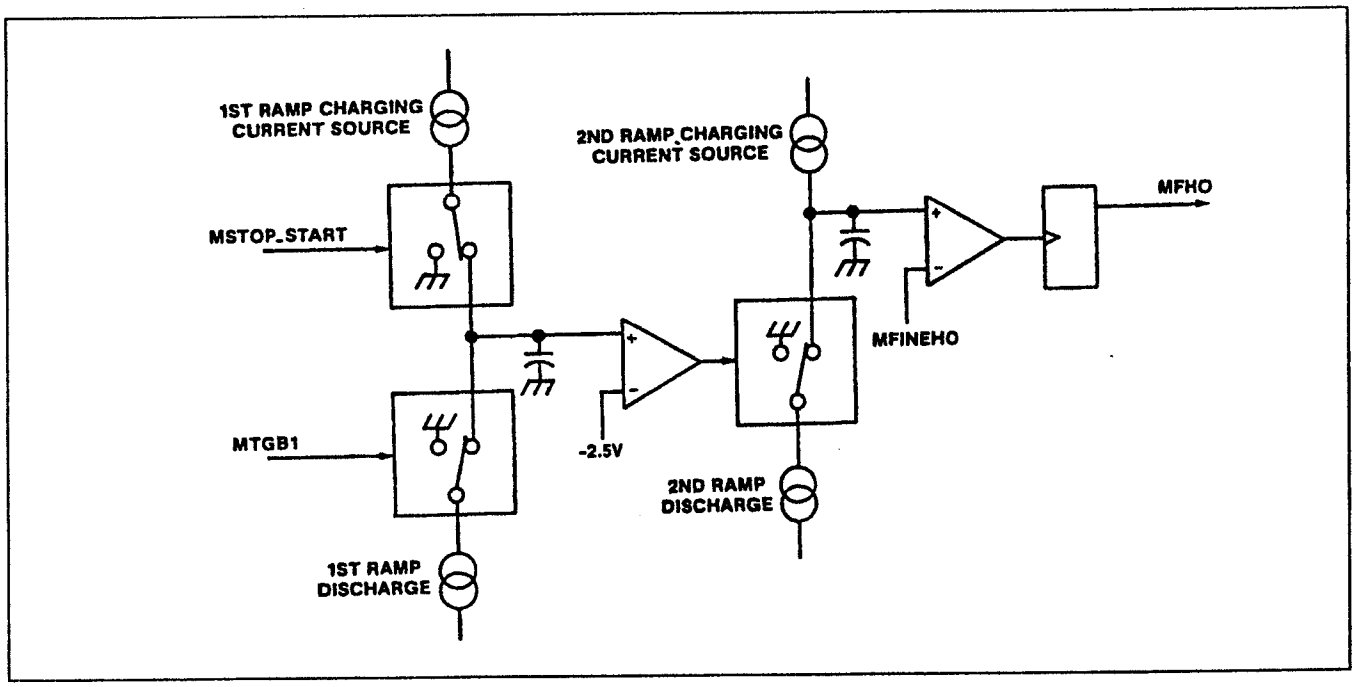


Figure 2-3. Equivalent Operation of the Main Fine Holdoff

A5A2 Trigger Enhancement Board <11A>

The comparators U210D and U210C and the flip-flop U300B effectively measure the time difference between the deassertion of MHFO (low) and the assertion of the differential pair TGM/TGM-L. If this time difference is shorter than the RC time constant determined by R200 and C200 (at the D input to flip-flop U300B), then the flip-flop clocks a low Q-bar output when pin 15 of U210 goes high.

At this point, the existing state at pin 3 of comparator U210 is high. The high output at U210, pin 15 is delayed six nanoseconds by delay line DL210 and is input to the comparator U210B at pin 7. The comparator outputs a low at pin 3. The delay line then outputs a high at pin 1, which is input to the comparator U210A at pin 5. The comparator outputs a high at pin 2.

The comparator outputs at pins 3 and 2 and the flip-flop output at pin 14 are wire-or'ed together. The low-to-high transition at the clock input of the flip-flop U300A causes the Q-bar output to change state (it doesn't matter which state the output is changing to or from). This transition changes the Main Ramp Threshold voltage, used as the reference to comparator U1940A. This effectively changes the timing of the Main Fine Holdoff generator by 1 ns.

In the above circuitry, the input to pin 10 of comparator U210 is temperature-compensated by diodes CR410, CR412, CR414, and CR16.

Delay by Events (diagram 10)

The Delay by Events counter allows the acquisition of the Window record to be delayed relative to the Main trigger point by a user specified number of events. The Delay by Events counter is made up of three sections. The first two high-speed sections reside on the Acquisition board and a low speed section is contained within U264 on the Time Base board.

When Delay by Events is selected, the Window Trigger IC ignores the window holdoff input (U1710-39) and performs as a high speed comparator. When the window trigger signal is above the trigger threshold the trigger gate (TGW) is high and vice versa. The first stage of the Events counter (U1530) is a 2-bit shift register counter which is clocked by TGW. The first stage output (U1530-15) clocks the second stage 4-bit counter which in turn clocks the slow counter (U264) on the time base board. After each stage has reached terminal count, a clock signal will be generated on the next positive transition of TGW which sends a differential trigger signal to the Window Time Interpolator via U1420A.

Time Interpolators (diagram 7)

The Time Interpolator measures the time between the trigger signal and the sampling strobe once during each acquisition cycle. Separate Time Interpolators measure the trigger to sampling strobe time for the Main and Window triggers. Each Time Interpolator consists of a coarse and a fine section.

The coarse measurement begins when the MSFR(L) output of the Main Fine Time Interpolator makes a negative transition. This occurs following the Main Trigger and enables the 200 MHz clock to pass through gate U1110B to the coarse counter (U922). The coarse counter counts until the sampling strobe (via U1010-14) occurs, yielding the trigger to strobe time with 5 ns of resolution. The Main Coarse Counter is initialized before each acquisition cycle via the Coarse_LD_CK and Coarse_LD signals generated in the holdoff circuitry. The Window Coarse Time Interpolator operates in the same manner.

The Fine Time Interpolators measure the time between the trigger and the 200 MHz clock with 10 ps of resolution. The measured time is converted to a digital count within the Fine Time Interpolator and is serially shifted into the

Fine Time Interpolator Shift Register when the measurement is complete. The data from the Main and Window Fine Time Interpolators is serially multiplexed to the DAG IC on the Time Base board.

Upon receipt of the Fine Request (FRQ(L)) signal from the Time Interpolator circuitry the DAG IC reads the Main Coarse and Fine data from the coarse multiplexer (U1244, U1246) and the fine multiplexer (U1250) respectively. After the Main data is read the coarse multiplexer input is switched to the Window coarse data and the Fine Time Interpolator Shift Register is loaded with Window Fine data upon receipt of the Fine Acknowledge (FAK(L)) signal from the DAG IC. The Window data is then loaded into the DAG IC.

ACQ Autocal and Refresh (diagram 12)

Diagram 12 contains the circuitry for automatic calibration of various digitizer circuits, including the acquisition, time base, and trigger circuits. It also contains circuitry for analog voltage refresh of various digitizer circuits, including the trigger level and holdoff, acquisition time interpolators, two-stage A/D flash converter, and sampling circuits.

Voltage Reference DAC (diagram 12)

The DAC (U1470) provides multiplexed analog output of various refresh voltages to the analog sampling bridges: Refresh Bank 1 (U960 and U970), Refresh Bank 2 (U1270 and U1272), and Refresh Bank 3 (U1072 and U1170). The microprocessor writes a digital input value to the DAC every 400 µs. Each digital input value represents a unique reference or control voltage needed for one of the circuits listed in the paragraph above.

The DAC's Offset (R1576) and Gain (R1582) settings are adjusted during diagnostics calibration initiated at the front panel (see the Section 4 Diagnostics for details).

Refresh Banks 1, 2, and 3 (diagram 12)

After the DAC presents the analog voltage to the sampling bridges of the refresh banks, the microprocessor places an address at the input of the addressable latches (U1070, U1370, and U1172). Then the SCAN(L) signal from pin 34 of J84 enables the input address to select the addressed output which will follow the level of the D input (starting with Refresh Bank 1). This, in turn, selects which switch of the sampling bridge that closes for 400 µs (all analog switches of the sampling bridge are initially open). Thus, the analog voltage from the DAC is stored on the capacitor of the now closed refresh bank sampling bridge switch. This voltage is then passed from the high impedance output buffer (of the closed switch) to the circuit it supplies the reference or control voltage for. This process is repeated every 400 µs, starting, sequentially, with each switch of Refresh Bank 1; then, each switch of Refresh Bank 2; and then, each switch of Refresh Bank 3, so that each reference or control voltage circuit is continually refreshed with the appropriate voltage.

Cal Select (diagram 12)

During the plug-in calibration mode, register U1570 receives an address from the microprocessor that indicates the appropriate settings for the Calibrator and plug-in calibration signals. The CALRANGE(L) signal from pin 26 of J84 clocks in the specified mode settings. In particular, pin 19 of U1570 drives the enabling of either the 450 Ω or 50 Ω output impedance for the Calibrator signal and decoder U1670A and B provide the appropriate enables for the Cal Gain Select stage.

Calibration Gain Select (diagram 12)

A digital value from the microprocessor provides an analog plug-in Calibrator Voltage Reference of, typically, 5 to 10 V from the DAC and Refresh Bank 2 circuitry to the input amplifier of the Cal Gain Select stage, which attenuates the voltage. Analog switches in U1672, in conjunction with part of resistor divider network R1877, provide 1X, 2X, or 4X attenuation and analog switches in U1772, in conjunction with part of resistor divider network R1877, provide attenuation of 1X, 10X, 100X,, or 1000X, depending on the decoded signal from the Cal Select stage. The voltage is then passed to the Calibrator Output buffer stage.

Calibrator Output Buffer (diagram 12)

Op amp U1870, with transistors Q1968 and Q1969, drive the calibrator output signals, including Plug-in Cal at pin 3 of J91A and Front Panel Cal to the CALIBRATOR output.

Front Panel Calibrator (diagram 12)

The front panel CALIBRATOR signal is either a DC reference level of about 6 mV to 6 V for probe calibration of gain and offset; or a 6 V, 1 kHz or 2 MHz square wave for probe compensation and deskewing. The square wave is generated by Q1668, Q1562, Q1560, and their associated circuitry whenever the CAL_ CLK signal into flip-flop U2060A (this happens during the probe deskewing and compensation operations) is enabled. The output impedance for the CALIBRATOR is, again, either 450 Ω or 50 Ω, depending on the type of 11000-series probe attached to the bnc connector.

Plug-in Ground Select (diagram 12)

Demultiplexer U1970 receives an address from the Cal Select circuit (U1570) that indicates which plug-in compartment is selected. Appropriate ground reference currents from the plug-in and the front panel are passed from op amp U1870B to the front panel and plug-in cal circuitry to offset any voltage difference.

A6 Time Base Board <A13 through A20>

The basic function of the Time Base controller is to produce waveform record lengths that have the desired length and resolution at the correct position. The time base controller uses circuitry that is optimized for high-speed operation. An 80186 microprocessor is (U224, diagram 16) is used to set up circuitry for most modes of operation. However, software is used to perform some complex functions.

The 80186 microprocessor uses 16K bytes of RAM (U281 and U283, diagram 16) and 128K bytes of ROM, along with several I/O devices to accomplish numerous functions. However, the time base system circuitry may operate independent of the 80186 for many acquisition and transfer cycles. The time base system circuitry restarts itself at the end of each cycle unless the 80186 instructs it to do otherwise.

Main and Window Records

Before we can talk about the operation of the time base system, we need to understand what a record is and what parameters and constraints are associated with it. In a conventional oscilloscope, the trigger point on the displayed waveform is very near the start of the sweep. In the 11401/11402 digital sampling oscilloscope, it is possible to position the trigger point anywhere relative to the record. However, the trigger point for the main record must occur *during* the main record. This can be at the beginning or at the end of the record, or anyplace in between. This requires that we know where the trigger is before the data is acquired, so that we can acquire enough data before the trigger point to complete the record.

By setting up a counter to count from the beginning of the sweep to a value which is equal to the time before the trigger, we can collect enough points before the trigger. Once the trigger has occurred, we can count the amount of data that occurs after the trigger to collect the proper number of points. Then, we need only count back the required number of points (as calculated by the Destination Address Generator, see U166, diagram 20 and the overview description, below, for more details) from the end of the acquired points to identify a complete record.

In addition to the main record, the 11401/11402 also allows two window records to be defined by the user, with the constraint that they must have at least one point in common with the main record (this is mainly to let the user see where the window really is). The window records use a single-trigger source, but their trigger is not confined to being on the window records. The window trigger, in most cases, is connected to the main trigger system via hardware located on the acquisition board. In that case, the windows are positioned relative to the main trigger point, but because of the above constraints, the main trigger does not have to occur during the window records. So, it is possible to have a window that is completely before or after the main trigger. These requirements cause the hardware within the time base system to be used in several different manners for different sorts of setup conditions.

Note that when looking at the time base system, one must also consider the acquisition board, as the two boards are very interrelated in their functions. In fact, in many cases, a function may be split between the two boards in order to take advantage of each board's strengths.

Functional Overview

This overview describes the events that happen during a normal acquisition cycle. Keep in mind that this not an inclusive list for all conditions, but a simple case that can be built upon.

Acquisition

The starting point for the system is the occurrence of the RESTART signal, This may be created due to the previous cycle finishing or by the microprocessor starting the digitizer system. The occurrence of the RESTART signal causes a number of things to happen (this is a general order of events):

1. The acquisition memory system switches into the acquisition (write) mode and the time base logic initializes for the next sweep.
2. The time base logic starts the PRETC counter, which is used to guarantee that at least the required number of pre-trigger points will be acquired before the trigger. With this, the rate counters for the main and window time bases begin to output time base requests. It is important to note that the rate counters both run continuously, even during the transfer phase of the system.
3. The vertical state sequencer generates the Acq_Request signal to the acquisition memory that will correspond to each of the selected vertical input signals.
4. Using the Acq_Request line, the Acquisition memory system takes the current 10-bit A/D value from the acquisition board, via the Input Data Latch, and places it into the Acquisition memory, and then, advances the memory address counters for the next sample. Along with the 10-bit A/D value that was stored into the acquisition memory, there are two groups of three bits. One of these 3-bit groups identifies which time base the Acq_Request was generated by: Main, Window1, and/or Window2. This information is used later, in keeping track of the number of main and window record points that have occurred. The other group of three bits is used to identify the vertical input that produced this A/D sample (this data comes from the Initial ID table). This is driven by inputs from the Vertical State Sequencer and the Plug-in Chop Sequencer. The Initial ID table lets the scope encode a large number of possible vertical input combinations down into the eight possible vertical sources.

Trigger

The assertion of the main trigger is held off by the PRETC signal, so that the trigger doesn't occur until enough data points have been acquired.

The value in the pre-trigger counter is determined by finding the maximum number of pre-trigger points needed by the main and window records. The output of the pre-trigger counter is fed into a section of hardware called the Randomizer. Its function is to make the total cycle time of the system more random. If the system is not random in nature, then certain frequencies of input signals will cause the system to not acquire complete records during equivalent time operation.

Once the randomizer inserts a random delay into the PRETC path and the PRETC signal arrives on the acquisition board, the acquisition system may now accept the next valid trigger condition. For the trigger to be valid, it must also satisfy the holdoff conditions that have been set up. When the first valid trigger occurs, the acquisition board passes the main trigger signal over to the Time Base board. At the same time, the window holdoff counter is started

(if it is set up) to holdoff the window trigger system, which is constrained to allow only window triggers after the main trigger has occurred. Also, the main time interpolater is now used to measure the amount of time between the trigger occurrence and the next 50 ns clock, which is the basic clock rate of the time base system. This value is sent to the DAG (Destination Address Generator), which uses this number, along with other measured and constant values, to produce the addresses for the data points acquired during this sweep.

These addresses are used to position the data points during the transition phase in the waveform memory, which is located on the MMU Board in the EXP section of the 11401/11402 system. Once the main trigger signal arrives on the time base board, it starts the main post-record counter, which is used to count the number of main record points to acquire after the trigger. Each of these points is handled in exactly the same manner as the pre-trigger points. The data points in acquisition memory do not carry any time-keeping information with them, other than their relative position to other points in the same record. Therefore, it can be seen that the flow of these points into acquisition memory must be continuous during the acquisition of the signal. Otherwise, the relative time between the points would be altered (this concept will be important during the discussion of the plug-in chop system).

Once the main post-record counter has counted down, thereby filling up the post-trigger section of the main record, it asserts its terminal count signal, which will cause several things to happen. First, the flow of rate pulses from the main time base counter will be shut off so that no more main record points will be acquired. Also, it will send a signal to the End Detect Logic that uses this signal, and others like it from the window time bases, to determine the end of the acquisition phase of the cycle.

Window Time Base

The window time base system supports two fairly independent windows. They share enough hardware and other things to make them appear as one time base at times. and, at other times, as two time bases. The windows are required to use the same trigger source and to operate at the same rate and duration. But, there are no restrictions upon the position of one window verses the other. The descriptions of window1 and window2 are interchangeable. The 1 and 2 designators are for convenience only. The window time base hardware operates in much the same manner as that of the main time base, although, there are more counters and more modes of operation. There are three types of operation modes for the window time base, they are *runs before windows*, *pre-trigger windows*, and *runs after windows*. All of these modes share elements of the next mode. In other words, the runs before mode behaves similarly to the pre-trigger mode. The runs after mode uses some common elements of the pre-trigger mode.

Pre-Trigger Mode

In the window time base, the pre-trigger mode is the most similar mode to that of the main time base's mode of operation. In the pre-trigger mode, some points are acquired before the window trigger and the remainder of the points are acquired, using the individual window post-record counter. Since there is only one pre-trigger counter in the system, both the main and window time bases must work off of the same counter. As stated previously, the window trigger is common between the two window records, but it may be that the main trigger is fed over into the window hardware or it may be a completely separate trigger signal. This, by definition, must occur after the main trigger has occurred and the window holdoff has timed out.

Runs After Mode

The runs after mode for the windows occurs when the user specifies that the desired position of the window record is some time after the occurrence of the window trigger, such that, some time must pass before the window may start acquiring data. In this case, a counter, called the window position counter, is used. Each window time base has one of these, just as each of the windows have their own post-record counter. This position counter is used to

count the number of window rate pulses from the window rate counter, after the window trigger. Once this position counter has counted down, the window's post-record counter is started, which causes the window rate requests to be sent off to the vertical state sequencer and handled in a similar manner to that of the main record. When this post-record counter finishes, it will shut off the window time base requests for that window and inform the end detect logic that the window has completed.

Runs Before Mode

For the runs before case, the entire window record is positioned ahead of the window trigger. To acquire only the desired points, we would have to know when the trigger was going to occur, since all actions enabling and disabling the acquisition of the window points have to occur before the actual trigger, even though these points are supposed to be related to the trigger.

This requires a sufficiently long pre-trigger time to acquire the entire window record and to acquire all window points that are available until the actual trigger shows up. At that time, the acquisition of the window record's points are disabled and the end detect logic is signaled to indicate the end of the that particular window record.

Note that the two window records do not have to be in the same modes, as the modes are set independently for the two window records. In the runs before case, the window post-record counter is used during the transfer phase of the cycle to count back as the DAG is reading points out of the acquisition memory, to determine the number of the particular window record points that it has encountered. Thus, it provides a way to count back in time in order to look at the data that was acquired before the trigger event took place. Once a pre-determined number of points have been counted back, the DAG can now treat these window points just as any other type of points.

Trigger-to-Trigger Counter

The trigger-to-trigger counter is used to measure the number of 50 ns intervals between the main and window triggers. Since the main and window time interpolator measure the interval between the trigger and the next 50 ns clock, this information, which is collected by the DAG shortly after the individual trigger events, can be used with the value of the trig-to-trig counter, which must be read via the microprocessor, to achieve trigger-to-trigger single shot measurements with a resolution of 10 ps and an accuracy of 100 ps.

End Detect Logic

Once all of the enabled time bases have reported their status as done (the disabled time bases are always treated as finished), the end detect logic makes sure that the vertical state sequencer has completed its last operation and, when appropriate, that the plug-in chop sequencer has completed its work (if it is in a special mode called "Single Shot Chop"). The plug-in chop sequence is described and its modes later on. Once all of these checks have been performed, the end detect logic asserts a signal, Acq/Xfer, to the Destination Address Generator (DAG) and to the Acquisition memory system. This signal is used to switch modes in both places.

For the DAG, it tells it to start trying to read the acquisition memory. To the acquisition, this signal, Acq/Xfer, switches the mode from "write" to "read" and changes the direction of the automatic address counters. Neither the time bases nor the DAG need to know the absolute address of a piece of data, only the relative position to the next sample from the time base. Because the acquisition memory uses a post-increment and post-decrement mode of operation, the hardware performs a dummy read cycle at this time to correctly position the address counters to the lastly acquired piece of data, rather than, to the next available location in acquisition memory. The acquisition memory consists of four banks of 4 k by 16-bit words for a total of 16 k words.

Destination Address Generator

Once the acquisition memory has turned around, the DAG starts to request waveform points via the Dag_Request line. It should be noted that the DAG always starts at the end of the record and works to the start of the record. For each point, the DAG will request a point from the memory, this point may contain a point for the main, window1, and/or window2 records. The DAG never sees the data contained in the waveform point, but looks directly at one of the 3-bit groups and, indirectly, at the other 3-bit group. As noted above, one of the 3-bit groups, called the horizontal tags, is sent to the DAG so that the DAG can keep track of the relative positions of each of the records. The DAG will look at the first of the waveform tags and check to see if the data point is for that particular record, if not, it will move on to the next tag bit. Otherwise, it will make the necessary calculations for that point and either generate an address for it, or discard it (for a variety of reasons). A point can be discarded: if it is part of the system overrun (more about this later); if the record, which has to be a window, is in runs before and this point is before the record; and, if the point follows a record that has already been completely sent or it has been marked Invalid by the final ID table. Under most circumstances, even though a point is discarded, the DAG actually uses this information to update internal and external counters in order to keep track of the current position.

The other group of tag bits is used to encode the vertical source of that sample data. This set of three bits is applied to the Final ID table, along with two bits from the DAG that indicate which record it is processing at the time: main, window1, or window2. These five bits are used to look into a table that will determine if that particular waveform is wanted or not. To clarify this, if we think of a user wanting the left1 and left2 plug-in channels to be displayed on the main time base, left1 to be displayed on window1 and left2 on window2 time base. Once a time base has been set up, the acquisition memory will actually contain points of all vertical sources, with all time bases, over time. So that we don't use up anymore waveform memory than we have to, the final ID table provides a means to identify the unneeded traces and to delete them (before they are sent to the waveform memory) by declaring them to be invalid.

The final ID table also helps solve a problem in calculating the addresses of the waveform points, when multiple plug-in units are being used. In this case, the system samples all of the inputs, from left, center, and right, simultaneously, then, the three inputs are sequentially applied to the A/D converter. In this setup, all three data values occurred at the same instant in time, so the position of the data within the waveforms should all be identical.

The final ID table also provides a signal, Adr_Hold, to indicate to the DAG that a new address should not be calculated for the next point. In this manner, all of the points of a group of time coincident data will be placed at a common position within a waveform. The other service that the final ID table provides is the further sorting out of what address the points go to in waveform memory. The waveform memory is overseen by the MMU system, which provides a set of TAG registers for the transfer of waveforms. These tags are used as a base address, within the waveform memory, while the address calculated by the DAG provides the offset from that base. The final ID table, using the inputs of vertical and horizontal sources of a waveform, specifies one of 14 possible waveform tags.

These tags and the addresses, which are generated by the DAG, are passed to the MMU interface. This interface is a fully "handshaken" interface with the waveform memory of the MMU. When the signal 'Dig_Request' is asserted by the interface hardware, the MMU eventually responds with A_Out, which tells the interface hardware to place the tags and address of the data point on the address/data lines. Then, after the MMU latches these addresses, the MMU asserts D_Out. This causes the interface hardware to place the actual data value for the point (after first passing through the left-justifier system, which really does nothing more than to adjust the 10-bit number into a 16-bit number that is left-justified and provides some special handling).

At this point the interface hardware removes the Dig_Request signal. While this has been happening, the DAG has been working on the next data point, right up to the point where it has to wait for the previous data to be completely transferred to the DAG. The DAG determines that it has reached the end of all of the acquired data by the calculations of the addresses of the waveforms that it has been transferring. Once the address of the waveform points falls below zero, the DAG detects that it has reached the end of the record in memory. If for some reason, due

to hardware malfunction, there are not enough data points in memory to satisfy the DAG, the DAG continues keeping the memory until the acquisition memory counters wrap around and, again, present the same data that was presented previously. If this idea is extended to the point where only one point was acquired in a waveform, which should have had 1024 points or so, the acquisition memory would cycle 1024 times. Once the end of the transfer stage has been completed, the DAG will assert the END signal for a short time. This will cause the RESTART signal to be asserted, which will cause the entire sequence to be repeated.

A7 Display Controller Board <33> through <37>

Executive Processor Parallel Interface Port (diagram 37)

The Executive Processor Parallel Interface port provides two separate functions; 1) provides a bi–directional 16–bit interface for commands and, 2) provides an interface for transferring incoming 16–bit waveform information. In the command mode this interface is a relatively straightforward 16–bit port, but in the trace data mode it performs a number of different data transformations upon the incoming data. This port operates in conjunction with U524's programmable DMA (Direct Memory Access) unit shown on diagram 33.

The Executive Processor Parallel Interface port is comprised of IC's U731 and U733. U733 is a bi–directional 8–bit latching port with a two–line handshake for each direction of data flow. U731 is basically the same port without the handshake logic.

The incoming data channel uses two lines, SENDNEW(L) and DATALATCH(L) (J52, pins 43 and 45 respectively) to implement the handshake. The SENDNEW(L) source is from the Display controller board and the DATALATCH(L) source is from the device at the other end of the interface. When the Display controller is ready for an incoming transfer to occur, SENDNEW(L) is brought low, if its not already low; the low state indicates that the display is ready. The device at the other end of the interface will bring DATALATCH(L) low and place the data to be transferred upon the 16–bit bus. When the DATALATCH(L) signal is brought high, the rising edge is used to latch the incoming data, and to set the SENDNEW(L) line high.

Since the SENDNEW(L) line is also the DMA (Direct Memory Access) request line for the incoming channel, the DMA channel will eventually read the port, and this will cause the SENDNEW(L) line to be reset to its ready' state.

The outgoing data channel uses two lines, DATARDY (Dataready) and DATAGATE(L) (J52, pins 49 and 47 respectively) to implement the handshake. The DATARDY source is from the Display Controller board and the DATAGATE(L) source is from the device at the other end of the interface. When the Display controller is ready to transfer a word across the interface, it will bring the DATARDY line high. This will signify that there is data stored in the port latches for the other device to access. When the other device on the interface brings the DATAGATE(L) line low, the port's output drivers will be enabled and thereby place the data upon the 16–bit bus. When the DATAGATE(L) line is brought high, the rising edge will clear the DATARDY line. This falling edge of the DATARDY line will cause the flip–flop to generate a DMA request to the microprocessor.

The DMA data request line for the incoming data channel (DRQ0) is connected to the data received flag (flip–flop) on the interface IC, U733, pin 5. When this flag is driven high by new data being clocked into the interface port, the DRQ0 line is driven high. This will cause U524 (diagram 33) to give up the bus at the next opportunity and perform a read of the interface port, either the command port or the trace port. When the processor reads the interface port, the data received flag will be cleared, thus removing the DMA request and signaling the device on the other end of the interface port that the word has been read.

The DMA data request line for the outgoing data channel (DRQ1) is handled somewhat different than the incoming data request line. When the Display Controller board is reset upon power–up, or at any other time the board is reset, the DRQ1 line is set to the active state (high). When the software wants to send a transmission from the Display Controller board to the device at the other end of the interface, it will enable the channel 1 DMA. The pending request will cause a word to read from memory and to be written into the interface port. In the process of writing data to the interface port the DMA request line is cleared. Once the device on the other end of the interface has acknowledged the transfer, the DMA request will be reasserted.

CPU and Ready Logic (diagram 33)

The Display Controller board's incoming and outgoing data is serviced by U524's programmable DMA (Direct Memory Access) unit. The DMA provides the following two independent channels. The DRQ0 (pin 18, Data Request line) that serves the incoming data channel, and DRQ1 (pin 19, Data Request line) that serves the outgoing data channel.

You should note that transfers from the interface port into memory is a source–synchronized transfer, while the transfer from memory to the interface is a destination–synchronized transfer. The interface port has been designed to run without any wait states. That is, its a zero wait-state device. Microprocessor U524 uses PCS4 to access this port, so currently this port is selected between 0200–027F(hex) in the I/O space.

Waveform Attribute Encoder (diagram 37)

This circuit facilitates high–speed transfers of incoming information. The digitizer and waveform processing systems in this oscilloscope dictate that the waveform data format is a left–justified 10–bit word. This 10–bit word is transmitted in a 16–bit word, with the lower six bits being considered as "don't cares", with three exceptions. These three exceptions are Null, Overrange, and Underrange, and their values are passed as 8000(hex), 7FFF(hex), and 8001(hex), respectively. The Waveform Attribute Encoder stage uses PAL (Programmable Array Logic) to decode these special cases and set the appropriate bits in the trace data word. The Null bit in the display word may be set on any data passing through the Waveform Attribute Encoder, regardless of the value of the data. This "Null force" condition is enabled by asserting bit 4 of port 1 of U523 (pin 35) on diagram 33.

The waveform data that is sent to the Display Controller board from the Compressor board is in a two's complement format. In order to support the VRS (Vertical Raster Scan) system, which requires 9–bit data ranging between 0 and 1FF(hex) inclusive, the data must be transformed and shifted. The number format transformation is accomplished by inverting the most significant bit of the incoming data. The shifting of the incoming trace data is further complicated because the VRS system can support two display modes; the single– and the dual–axis modes. When the VRS system is operating in the single–axis mode the incoming data is shifted to the right by six bits. When the display is operating in the dual–axis mode the data must be shifted to the right by seven bits. This will adjust for half the dynamic range of the individual axis. A bit coming from the MUART (Multifunction Universal Asynchronous Receiver Transmitter) circuit, port 1, bit 7 (U523, pin 32 on diagram 33) is used to supply an offset to the encoded trace data word. When this bit (offset) is cleared, the waveform will be displayed in the lower half of the display, because the resultant data values range between 0 and OFF(hex). When the offset bit is set, the waveform will be displayed in the upper half of the VRS display area, since the resultant data is within the range of 100(hex) to 1FF(hex).

ROM and Select (diagram 33)

EPROM's U602 and U612 comprise the standard firmware for the Display Controller board, with U602 being the high order byte. The standard firmware EPROM's may be configured for use by three different types of EPROM's, depending upon the amount of storage needed. Refer to Table 2–1 for further information.

EPROM's U700 and U712 comprise the optional firmware for the Display Controller board, with U700 being the high order byte. The optional firmware EPROM's may be configured for use by three different types of EPROM's, depending on the amount of storage needed. See Table 2–2 for more information.

Table 2-1
Standard EPROMs Storage Capabilties and Jumper Settings

EPROM Type	Jumper Setting J7	Size In J12	Address Bytes	Range *(hex)*
27128	128	256	32K	F8000–FFFF
27256	256	256	64K	F0000–FFFF
27512	256	512	128K	E0000–FFFF

Table 2-2
Optional EPROMs Storage Capabilties and Jumper Settings

EPROM Type	Jumper Settting J8	Size In J13	Address Bytes	Range *(hex)*
27128	128	256	32K	D8000–DFFF
27256	256	256	64K	D0000–DFFF
27512	256	512	128K	C0000–DFFF

The EPROM's are selected via the ROMCS(L) line on processor U524, pin 34. When U524 is reset, the ROMCS(L) line is the only Chip Select Unit line that is active, and only then for the address range of FFC00–FFFFF*(hex)* (the upper 1K of memory). The ROMCS(L) line also sets up a three wait state delay in every EPROM access, and it looks at the external ready line.

Microprocessor Data Latch (diagram 33)

The Microprocessor Data latch stage is comprised of data bus drivers U615, U621, gate U427C, and inverter U634F. Address and data information is multiplexed together onto a common set of pins (U523, pins 1 through 8). This necessitates locating the MUART and the microprocessor on the same local bus (before the Microprocessor Data latch stage). When the microprocessor supplied signals DT/R(L) and DEN(L) are used to control latches U615 and U621, the data bus will actually be demultiplexed, in that the address does not appear on the data bus. Since the address does not appear on the data bus, the MUART chip is the only component other than the bus drivers to reside on the local microprocessor bus, lines PD0 – PD15. In order to prevent contention with the Microprocessor Data latch stage during a MUART access, the DEN(L) signal is gated with the MUARTCS(L) line, which is the PCS(L) on the microprocessor. With this gating the Microprocessor Data latch stage is disabled during any request to the MUART.

The MUART is accessed via the microprocessor Control Bus using the MUARTCS(L) line, which is connected to the PCS0 line, pin 25 on the microprocessor. All requests to this line requires that three wait states be inserted.

The Display Controller board's microprocessor, U524, can be run in a Forced Instruction Mode. This allows a person to observe the address lines in operation, and to detect a faulty component, using signature analysis. This is the only place where diagnostics software is not used, as the address lines are assumed faulty at this point, and the software cannot run.

The Display Controller board circuitry will operate in the Forced Instruction Mode when jumpers J5, J6, and J11 are moved from the Norm position, to the Test position. This forces data bus drivers, U615 and U621, into the disabled mode. Jumpers J5 and J11 cause the two microprocessor local data bus lines, PD1 and PD9, to be pulled down to ground. This, in turn, will cause microprocessor, U524, to receive a FDFD(hex) instruction code, which is a STD instruction. Since this is in effect an NOP instruction, the microprocessor will continue to fetch this instruction and progress through the address range. A note should be made that microprocessor U524 will not sequentially go through memory, as the CS register is set to FFFF(hex) upon power up. With IP register being set to zero, this will cause the address sequence to be FFFFO(hex) → FFFFF(hex), then OOOOO → OFFEFF(hex), then back to FFFFO(hex). Further, the chip select line for the ROMs will be active during the passage through FFFFO(hex) to FFFFF(hex).

Microprocessor Address Latch (diagram 33)

The Microprocessor Address latch stage is comprised of address bus drivers U526, U614, and U620. The ALE (address latch enable) signal from U524, pin 61, is active high to latch the addresses into the three bus drivers. The rising edge of the ALE signal is generated off the rising edge of the MPUCLK signal. Addresses are valid on the trailing edge of the ALE signal.

Diagnostic Loopback Control (diagram 37)

The Diagnostic Loopback Control stage is comprised of U623A,B,C,D, U630A,C,D, and U634A. When the LOOPBACKEN(L) bit is set, it causes this stage to make the Executive Processor Parallel Interface Port stage look busy. This strobes handshake lines, SENDNEW(L) and DATARDY(L), to transfer data from one side of latching ports U731 and U733, to the other. The transfer occurs when microprocessor U524's Chip Select strobes the BMLCS(L) line, by reading from I/O address 280(hex).

General Purpose Static RAM (diagram 33)

The General Purpose Static RAM (hereafter called GPSRAM) is comprised mainly of RAMs U601 and U611. The start of the RAM address space serves as an interrupt table for microprocessor U524.

The Display controller will be provided with 4K of RAM when Jumper J9 is installed in the 4K position and both 6116P RAMs (U601 and U611) are installed at the bottom of their sockets leaving pins 1, 2, 27, and 28 open. If it becomes necessary to provide more GPSRAM space, an alternate set of 6264P RAMs may be used. When 6264P RAMs are installed in the Display controller, jumper J9 must be moved to the 16K position.

Interface Data Buffers (diagram 37)

The Interface Data Buffers stage is comprised of octal bus transceivers U713, U720, and gate U721D. This stage provides synchronous two–way communications between the interface port and the microprocessor data bus.

Waveform Data Buffers (diagram 37)

The Waveform Data Buffers stage is comprised of octal buffer/drivers U616, U622, gates U721A,B, and inverter U636F.

MUART (diagram 33)

The MUART (Multi–function Universal Asynchronous Receiver Transmitter) stage is comprised of interface IC U523. It provides 16 lines of parallel I/O, a serial port, a baud rate generator, various timers, and an interrupt controller. The display controller uses the parallel I/O to provide software control of various portions of the display hardware. The serial port is used to provide a diagnostic communication path between the waveform processor (on the A16 Compressor board) and the display controller. MUART U523 is also used to develop software for the display controller, and to provide interrupts to microprocessor U524 in the CPU and Ready Logic stage.

Because of the limited number of pins available on the U523 MUART chip, the address and data information is multiplexed together onto a common set of pins (pins 1 through 8). This necessitates locating the MUART and the microprocessor on the same local bus (before the Microprocessor Data latch stage). When the microprocessor supplied signals DT/R(L) and DEN(L) are used to control the Microprocessor Data latches U615 and U621, the data bus will actually be demultiplexed, in that the address does not appear on the data bus. Since the address does not appear on the data bus, the MUART chip is the only component other than the bus drivers to reside on the local microprocessor bus, lines PD0 – PD15. In order to prevent contention with the Microprocessor Data latch stage during a MUART access, the DEN(L) signal is gated with the MUARTCS(L) line, which is the PCS(L) on the microprocessor. With this gating the Microprocessor Data latch stage is disabled during any request to the MUART.

Serial port lines TxData(L), RxData(L), and COMCLK are connected to to interface connector J52. Only the TxData(L) is an output. These are 5 V only signals, and while the format is compatible with RS–232–C interface, a voltage converter must be set up before an RS–232–C device may be connected to the serial port. The MUART only supports one handshake line which has has been hard-wired to the active state. The MUART is accessed via the microprocessor Control Bus using the MUARTCS(L) line, which is connected to the PCS0 line, pin 25 on the microprocessor. All requests to this line require that three wait states be inserted.

Video Shifter Control (diagram 34)

The Video Shifter Control stage is comprised of gates U630B, U432A, and positive edge–triggered flip–flop U425B. The BSRLOAD (Bit Plane Shift register Load) signal is generated by the detection of the video hardware phase of the video memory. That is, when CCLK, FCLK, LCLK, and MCLK are all high, and U630B is gated by the Display Enable signal from the CRT controller and Select stage (pin 18, U515). The presence of the Display Enable signal at pin 18 of U515 indicates a valid CRT controller and Select stage video memory address. This detection signal is clocked by the rising edge of the PCLK signal into pin 11 of U425B. This allows the signal on pin 6 of U432A to be sampled on the falling edge of PCLK by the Plane 1 and Plane 2 Video Shifter stages, U220 and U221, and thereby cause them to latch the data from the Bit Plane 1 and Bit Plane 2 DRAM (video memory) stages.

CRT Controller and Select (diagram 34)

The CRT controller and Select stage consists mainly of U515 and U534. This stage is the basis for all timing in the video portion of the Display Controller board. The chip select, CRTCS(L), causes a positive–going pulse to be formed using U534, U525E, and U517D. Latch U631A is used to latch the R/W(L) signal so that it will meet the hold times of controller U515. The data bus lines (MD0–7) run to controller U515 (D7–D0). Using MA1 (U515, pin 24) as a register select signal, the command and data registers are both placed on the same side of the data bus. The CRT controller and Select stage is selected through the microprocessors PCS1 pin, named CRTCS(L). This chip select must be set up for a three wait state delay for every CRT controller and Select stage request.

On the video side of the CRT controller and Select stage, the master video clock is set at 2.0 MHz, using the CCLK line. This sets the word display time to 500 ns. CRT controller U515 generates a new address on every falling edge of its master clock. This new set of addresses is passed on to the Bit Plane Address Mapping stage and to the VRS

Plane Address MUX stage. Since the new addresses are not required until the rising edge of CCLK the CRT controller has plenty of time to generate the new addresses. The Display Enable signal (U515, pin 18) is handled in much the same way. Display Enable is used to generate a load pulse to the Video Shift Control stage.

The LPEN input (U515, pin 3) is used by the diagnostics hardware to determine where the CRT controller thinks it is, with relation to the address values being presented on the Diagnostic Timing MUX stage.

Video Memory Buffer and Select (diagram 33)

Bus driver/latches U502, U503, U512, and U513 mainly comprise the Video Memory Buffer and Select stage. The most important function of this stage is to provide a way for the slower microprocessor to latch data from the high-speed video memory. It also serves to reduce loading on the microprocessor Data Bus. The drivers or latches are enabled depending upon the direction of the data transfer and when VSEL(L) is generated from the Video Memory Interface Synchronizer stage. The 16-bit Video Data bus (VD0-15) is a local bus that is used just for transfers between the video memory and the microprocessor.

Diagnostic CRTC/MPU Address Trigger Control (diagram 37)

The Diagnostic CRTC/MPU Address Trigger Control stage is comprised of U401A,B,C,D, U402A,B, and U517A. The video address registers U500, U501, U510, and U511 capture all 14 bits of the CAS(L) addresses. They also capture the RA0, RA1, RA2, and RA3 lines from U515 in the CRT controller and Select stage. The registers are triggered in two parts so that the CAS(L) cycles may be captured, by U401, which is driven by the Diagnostic CRTC/MPU Address Trigger Control stage. The trigger signal is also sent to the LPEN (light pen) input of U515 in the CRT controller and Select stage. Only one trigger signal is generated. Another trigger will not occur until the VRLCS(L) line is pulsed, indicating that all of the previously captured data has been read. The type of trigger (CRT address or microprocessor address) is determined by the state of the trigger bit in the "write only' diagnostics register, U600, pin 16, located at I/O address 101(hex). The address latches are read only registers.

Diagnostic Bit and VRS Plane Address Feedback Latches (diagram 37)

The Diagnostic Bit and VRS Plane Address Feedback latches stage is comprised of 8-bit registers U500, U501, U510, and U511. These video address registers capture all 14 bits of the CAS(L) addresses. They also capture the RA0, RA1, RA2, and RA3 lines from U515 in the CRT controller and Select stage. The registers are triggered in two parts so that the CAS(L) cycles may be captured, by U401, which is driven by the Diagnostic CRTC/MPU Address Trigger Control stage. The trigger signal is also sent to the LPEN (light pen) input of U515 in the CRT controller and Select stage. Only one trigger signal is generated. Another trigger will not occur until the VRLCS(L) line is pulsed, indicating that all of the previously captured data has been read. The type of trigger (CRT address or microprocessor address) is determined by the state of the trigger bit in the "write only' diagnostics register, U600, pin 16, located at I/O address 101(hex). The address latches are "read only' registers.

The microprocessor address trigger will occur upon the first GATE signal generated by U426A, pin 6 of the Video Interface Synchronizer stage, after the VRLCS(L) register bank has been read. The CRT controller address trigger will occur at the first CCLK(L) time after Display Enable has gone high, and after the VRLCS(L) register bank has been read.

Diagnostic Timing MUX (diagram 37)

The Diagnostic Timing MUX stage is comprised of multiplexers U433, U530, and flip–flops U516A, U516B. The purpose of this stage is to provide a means of examining a large number of repetitive signals, and to feed a number of them into the Programmable Timers (pins 20 and 21) of the U524 microprocessor. A means is also provided to route a subset of these signals into the Data In (pin 34) of the MUART, U523. In addition to handling the signals that are too fast for the microprocessor, a subset of these signals are passed through a divide by 4 circuit to reduce their rate. The multiplexers, U433 and U530, are used to multiplex the various signals in the Display Controller board circuitry into two input streams. The two multiplexers have common select lines (pins 1, 2, and 4) that select the source for their outputs. These select lines are driven by a diagnostics write only register located at I/O address 101*(hex)*. Another line from this register provides a means to reset the divider circuitry and to force its output to the high state. In order to let the microprocessor test the timers internally, the timer input pins must be held at a high level.

The output of the low–speed signal multiplexer is also connected to the DATA IN input to MUART U523. When the proper input selection is made, the DOUT (diagnostics output), pin 59, of gate array U125 can be routed to the MUART input. This also provides a way for the software to directly find the current state of the sync signals for the timing of various display events.

System Timing Generator (diagram 34)

The System Timing generator stage is comprised mainly of oscillator Y430, divider U431, U435A, and an associated chain of gates. This stage generates all timing signals for the Display Controller board.

These timing signals are derived from a 32 MHz clock that is generated by oscillator Y430, and with one exception, do not depend upon gate delays to produce the correct signal. The 32 MHz oscillator provides the required CRT pixel rate, and is divided down to produce 16 MHz, which is the highest possible microprocessor clock rate.

In order to provide the automatic board test system with access to the timing chain, U332D and U532D were added to the basic circuit. They provide the means of halting the 32 MHz clock and of supplying a new clock signal to the System Timing generator circuitry. When pin 12 of U332D (connected to pull–up resistor R534) is brought low with a test probe, the oscillator signal will be blocked from the remainder of the System Timing generator's circuitry. The pulled–up input of U532D, pin 12, may now be toggled externally to provide a new clock signal.

The output of gate U532D, pin 11, is connected to the input of clock divider U431, pin 2, and to the input of U427B which is the only gate delay element on the Display Controller board. Gate U427B delays the 32 MHz clock signal sufficiently that the signal edges occur near the times when the outputs of divider U431 changes states. The output of U427B is inverted by U434B to provide the PCLK (pixel clock) signal to the Display Controller board circuitry. This signal is again inverted by U434C to provide the PCLC(L) signal. This signal is used only in the DRAM Control generator stage. The slight delay of the PCLK(L) signal caused by U434C gives the DRAM Control generator stage more time to set up the inputs for the various elements.

The 32 MHz signal is divided into four sub multiples by U431. These are MCLK (microprocessor clock) at 16 MHz, LCLK at 8 MHz, FCLK at 4 MHz, and CCLK (character clock) at 2 MHz. An inverted version of CCLK is generated by U434E. The CCLK signal is also used by U435A to generate LCCLK(L) (Late CCLK). The LCCLK(L) signal is used to provide a window in which the DRAM accesses may occur. A single DRAM operation may occur in either or both half cycles of the LCCLK(L) signal. This delayed signal allows the video hardware to latch the data at the very end of their memory cycle, while the DRAMs are still holding valid data at their outputs. This delayed clocking scheme also allows the Video Memory Interface Synchronizer stage time to resolve the marginally stable conditions within some of its flip–flops. The LCCLK(L) signal is also used as the video phase RASE (Row Address Strobe Enable) signal, which is the counterpart of the Gate(L) signal for the microprocessor phase of the video memory cycle.

The MCLK (Microprocessor Clock) signal is used to drive microprocessor U524. The microprocessor produces a lower–frequency clock (8 MHz) from this signal which is used as its bus clock. This 8 MHz clock signal must not be interchanged with the LCLK 8 MHz signal.

DRAM Control Generator (diagram 34)

The DRAM Control generator stage, as its name implies, generates the basic timing signals to operate the Dynamic Random Access Memory systems on the Display Controller board. The control signals used for the VRS Plane DRAMs are the same as those used for the Bit Plane DRAMs.

In addition, this stage also generates a few signals that are used in various other places on the Display Controller board. Shift register U326 is the heart of this stage, and its output looks like a delay line with 31.25 ns steps.

Together, U427A and U434A generate a combination of signals that is input to U425A to form a single pulse at the start of each memory cycle. This pulse is injected into shift register U326, pins 1 and 2. Shift register U326 is simultaneously being clocked by the PCLK(L) signal to form a "solid–state delay line.' This delay line will produce a single pulse on each of its eight output lines (Q0 through Q7) during each memory cycle. These pulses are used to control the state of the RS flip–flops that are constructed from NAND gates (U332A,B, U333A,B,C,D). The output of these RS flip–flops are the DRAM control signals. The output of the solid–state delay line is also used to generate VDLATCH (pin 10 of U732C), which is used to latch data from a microprocessor read, from the video memory. A few of the outputs of the delay line are used directly for other control lines. Q3 is used in the Video Memory Interface Synchronizer stage. Output Q6 is used by the diagnostics hardware to latch the RAS addresses from the DRAM address multiplexers. Output Q7 is used to latch the word select lines VA1 and VA1(L) sufficiently early enough to set up the DRAM control circuitry.

Diagnostic Control/Status Latch (diagram 37)

The Diagnostic Control/Status latch is comprised of registers U600, U610, gates U613D,C, and LEDs DS500, DS501. In some cases the Display Controller board circuitry may not be able to indicate a malfunction. To solve this problem, two LEDs (DS500 and DS501) have been provided to give a visual indication of the current diagnostic test status. In addition to the two LEDs, the output of U610 is connected to eight square pins (Diagnostic Status Pins, 0 through 7). The state of these LEDs and square pins is determined through software. The LEDs are controlled using two bits from the diagnostics write only register, located at I/O address 101(hex). This register, U600, is reset to all zeros when the X RESET (external reset) line on the parallel interface is driven low. The state of the Diagnostic Status Pins (0–7) is set by the value written into the write only register U610, located at I/O address 100(hex). This register is reset to all zeros when the X RESET line is driven low.

Video Memory Interface Synchronizer (diagram 34)

The Video Memory Interface Synchronizer stage is the heart of the interface between the microprocessor and the display memory. This stage is responsible first to recognize a request for access to the display memory, then to hold off the microprocessor while a request signal is generated and acknowledged by the video memory, and finally to release the microprocessor.

The operation of this stage begins when a mid–range chip select (MCS0–3) is generated by the microprocessor. The chip selects become valid sometime after the beginning of T1 of the microprocessor. The signal is latched at the falling edge of clock MPUCLK (start of T2). This stage of delay is necessary because the microprocessor generates its chip selects before the RD(L) and WR(L) strobes become valid. At the halfway point of T2 the request is passed to the next stage. This stage generates the SRDY(L) (Synchronous Ready) signal to the microprocessor. This signal

will easily be set-up in time to meet the setup times for the microprocessors SRDY line. At this point the request is passed to the video memory side of the interface.

The request from the microprocessor side of the Video Memory Interface Synchronizer stage is now present at the input of the video memory side of the interface, waiting to be accepted. At the next falling edge of the CCLK signal, the request is passed to the next stage, which is a slightly delayed version of CCLK. This delay allows two things to happen. First, any metastable condition induced by the transferring of the microprocessor request to the asynchronous video interface, will have had time to settle out. Second, this delay allows the memory cycle to have data present and valid at the very end of the video hardware display cycle when the video data is latched. Once the request is clocked through the LCCLK flip-flop, it becomes knows as GATE. The GATE signal will only go low during the MPU portion of the video memory access time, and it will appear only once during any single microprocessor request. The GATE signal is used to re-decode the chip select into a plane request. This plane request will enable the hardware of the desired plane into generating a microprocessor memory request.

As the microprocessor memory cycle proceeds, the Q3 timing signal into pin 12, U337D in conjunction with the GATE signal into pin 6, U426A, will reset the microprocessor request. This will clock a flip-flop to assure that this signal is present during the next MPUCLK rising edge. Once this rising edge occurs, the request will be removed from the input of the video memory side of the Video Memory Interface Synchronizer stage. Also, the SRDY (Synchronous Ready) signal will be removed from the microprocessor, thereby allowing the microprocessor to continue the memory cycle. The Q3 signal is timed such that the video memory request will be removed from the video side before the start of the next MPU portion of the video access window. Also, if the operation requested was a read, the reset signal to the microprocessor side of the Synchronizer stage must occur late enough for the video memory cycle to complete and latch the data into the bus interface unit. The read data is latched into the bus interface unit by a combination of timing signal Q7 and GATE.

Once the microprocessor begins to finish the memory cycle, it will remove the chip select. When the chip select signal goes high it will reset the flip-flops in the microprocessor side of the Synchronizer stage. This action will prevent the Synchronizer stage from locking out another immediately following video memory request. When a string operation is being done in the video memory, the microprocessor only holds the chip selects high for just a few tens of nanoseconds before the next cycle is begun.

Bit Plane Address Mapper (diagram 34)

The Bit Plane Address Mapper stage consists of 4-bit binary full adders U412, U424, U428, U514, U520, and U521. This stage is used to reduce the memory requirements of the two Bit Plane DRAM stages. The CRT controller and Select stage is programmed such that considerable memory between scan lines is unused if linear mapping is done between the addresses that are output by the CRT controller and Select stage and the two Bit Plane DRAM stages. The Bit Plane Address Mapper stage allows this unused memory space to be compressed to a much smaller value.

This stage condenses the offset between adjacent scan lines from 64 words down to 48 words. In the row/column mode of the CRT controller and Select stage, the row addresses are incremented by "one" every time the column addresses progress through an entire count sequence. The column count sequence is actually between only 0 and 43. The other addresses generated at the end of the scan line (during raster retrace) are not needed, so they are ignored. From word 0 in one scan line to word 0 in the following scan line the addresses will advance by 48 (or $30_{(hex)}$). If the row address is considered as the scan line count, it could be multiplied by $30_{(hex)}$ to get the proper offset for each scan line. Because $30_{(hex)}$ is not a simple shift, the column address is added back into the newly converted row address to create the new address bit. This is accomplished by the first three 4-bit adders U514, U520, and U521. A multiplication by $10_{(hex)}$ is performed by just offsetting the bits that get added to the column addresses, therefore, the lower four column address bits are passed undisturbed. Then, the final three 4-bit adders U412, U424, and U428 are used to add the new version of the row address to the proper bits of the column address. This stage

will complete the above operation in between the time that the CRT controller and Select stage generates the new addresses and the occurrence of the rising edge of the CCLK signal, which is when the new addresses are used.

Bit Plane Address MUX (diagram 34)

The Bit Plane Address MUX (Multiplexer) stage is comprised of U404, U410, U414, U421, U403, U400, U634E and U325B. This stage is used to multiplex the CRT addresses for the bit planes and the microprocessor's addresses. Within this stage there are two sets of multiplexers. The first set performs the actual address multiplexing, and the second set multiplexes the results of the first set. The output of this stage provides the multiplexed addresses required by the Bit Plane 1 and 2 DRAM (Dynamic Random Access Memory) stages.

This stage performs a fairly simple transformation, as the least significant bit of the Bit Plane Address Mapper stage output is paired up with the least significant "word" address bit from the microprocessor. These multiplexers are controlled by a buffered and slightly late version of CCLK, with the exception of the least significant multiplexer. The low order multiplexer uses an unbuffered version of CCLK and a faster chip to ensure the VA1 line sets up flip–flop U325B soon enough. These multiplexers are controlled by the COLADR (Column Address) signal from the DRAM Control generator.

VRS Plane Address MUX (diagram 34)

The VRS Plane Address MUX (Multiplexer) stage is comprised of U405, U411, U415, U423, U413, and U420. This stage is used to multiplex the CRT addresses for the VRS Max Plane DRAM and VRS Min Plane DRAM stages, and for the microprocessor's addresses. Within this stage there are two sets of multiplexers. The first set performs the actual address multiplexing, and the second set multiplexes the results of the first set. The output of this stage provides the multiplexed addresses required by the VRS Max and Min Plane DRAM (Dynamic Random Access Memory) stages.

The input selection for the multiplexers is done using the same buffered CCLK signal that the Bit Plane Address MUX stage uses. This stage performs the same function for the VRS (Vertical Raster Scan) as the Bit Plane Address MUX stage performs for the two Bit Plane DRAM stages. That is, to put the most rapidly changing address bits in the row address lines. These multiplexers are controlled by same signal (COLADR line) as the Bit Plane Address MUX.

Bit Plane 1 and 2 DRAM Control (diagram 35)

The following discussion will cover both the Bit Plane 1 DRAM Control and the Bit Plane 2 DRAM Control as one stage. The bit–plane memory system is constructed of 64K Dynamic RAMs in a 16K x 4 configuration, running at a cycle time of 250 ns. The bit planes respond to memory requests from two sources.

First, from the video hardware, which needs refreshing every 500 ns. This request is formed by using the high state of the ACCLK signal, which is an inverted version of LCCLK(L) (Late Character Clock).

Second, from the combinations of the GATE(L) signal, and from the mid–range chip select lines from microprocessor U524. The video hardware requests are always word reads, while the microprocessor requests can be read or write, either byte or word type.

Once the request has been formed it is used along with a buffered version of CAS(L), and a word select line, VA1 and VA2(L), in order to determine which set of chips to send the request. The RAS signal is combined with the RASE signal to produce a signal (Q6 of U326) to the RAMs that will only occur when there is a valid cycle to be performed. Because the least significant bit is lost due to the word select line, twice as many addresses must be presented to the DRAMs in order to keep them refreshed.

When a microprocessor request occurs, the RAS signal is also sent to the other plane. The WE(L) signal (pin 8 of U320C) is constructed using the R/W(L) signal from the microprocessor and combining it with the GATE signal, so that a WE(L) can only be created during a microprocessor memory cycle.

Bit Plane 1 DRAM (diagram 35)

The Bit Plane 1 DRAM stage is comprised of dynamic random access memories U210, U211, U212, U213, U214, U215, U216, and U217. The following discussion will cover both the Bit Plane 1 DRAM and the Bit Plane 2 DRAM stages, since implementation of the two planes are identical, except where noted. The memory system is constructed of 64K Dynamic RAMs in a 16K x 4 configuration, running at a cycle time of 250 ns. The bit planes respond to memory requests from two sources.

First, from the video hardware, which needs refreshing every 500 ns. This request is formed by using the high state of the ACCLK signal, which is an inverted version of LCCLK(L) (Late Character Clock).

Second, from the combinations of the GATE(L) signal, and from the mid-range chip select lines from microprocessor U524. The video hardware requests are always word reads, while the microprocessor requests can be read or write, either byte or word type.

Once the request has been formed it is used along with a buffered version of CAS(L), and a word select line, VA1 and VA2(L), in order to determine which set of chips to send the request. The RAS signal is combined with the RASE signal to produce a RAS(L) signal to the DRAMs that will only occur when there is a valid cycle to be performed. Because the least significant bit is lost due to the word select line, twice as many addresses must be presented to the DRAMs in order to keep them refreshed.

When a microprocessor request occurs, the RAS signal is also sent to the other plane. The WE(L) signal (pin 8 of U320C) is constructed using the R/W(L) signal from the microprocessor and combining it with the GATE signal, so that a WE(L) can only be created during a microprocessor memory cycle.

Bit Plane 2 DRAM (diagram 35)

The Bit Plane 2 DRAM stage is comprised of dynamic random access memories U310, U311, U312, U313, U314, U315, U316, and U317. Since the function of this stage is identical to that of the Bit Plane 2 DRAM stage described previously, you must refer to that subheading for a circuit description.

Plane 1 Data Buffer (diagram 35)

The Plane 1 Data Buffer stage is comprised of octal bus transceivers U200 and U201. This stage provides synchronous two-way communications between the bit 1 data bus and the video data bus.

Plane 2 Data Buffer (diagram 35)

The Plane 2 Data Buffer stage is comprised of octal bus transceivers U300 and U301. This stage provides synchronous two–way communications between the bit 2 data bus and the video data bus.

Plane 1 Video Shifter (diagram 35)

The Plane 1 Video Shifter is comprised of register U220. Bit data that is read from the Bit Plane 1 DRAM stage is latched into the video shift register U220, then shifted out serially. The shifted data is passed to the VRS Generation and Control stages custom gate array, U125, for further processing.

The bit map data is presented to the input of the 16–bit video shift register, U220, at the end of the video hardware memory cycle. This data is latched into the video shift register upon the falling edge of the PCLK signal, when the BSRLOAD signal is held high. The least significant bit of the video memory word is shifted out first. This shifted data is fed into flip–flop U230A,B, whose active high drives the CNTE (counter enable) on gate array U125.

For the current monochrome implementation of the Display Controller board circuitry, there are only two planes of video memory; the Bit Plane 1 DRAM, and Bit Plane 2 DRAM stages. Their video shifter outputs, BP1 and BP2, respectively, are fed into the corresponding inputs of the VRS Generation and Control stage, U125. The BP0 input, pin 55, of U125 is connected to ground for this implementation. If another plane was to be added, its video shifter output would be connected to the BP0 input.

Plane 2 Video Shifter (diagram 35)

The Plane 2 Video Shifter stage is comprised of register U221. Since the operation of this stage is essentially the same as that of the Plane 1 Video Shifter stage above, only that stage will be described. Refer to the Plane 1 Video Shifter stage heading.

VRS Plane DRAM Control (diagram 36)

The Video Raster Scan Plane Dynamic RAM Control stage is comprised of U222B, U223A,B,C,D, U330A,B,C,D, U334C,D,E,F, U335B,C, U336B,C, and associated components.

VRS Max Plane DRAM (diagram 36)

The VRS (Video Raster Scan) Max Plane DRAM stage is comprised of Dynamic Random Access Memory devices U120, U121, U122, and U123. The following discussion will cover both the VRS Max Plane DRAM and the VRS Min Plane DRAM stages, since they operate the same. Further, in most respects these two stages are the same as the Bit Plane 1 and Bit Plane 2 DRAM stages. The major difference between these two pairs of stages is, while the bit planes each produce a 16–bit word on each video hardware memory cycle, the VRS planes memory each produces a 32–bit word. In order to interface with the 16–bit microprocessor bus, the 64K byte VRS memory plane has been divided into two 32k–byte planes. Even though the VRS memory system requires about nine display cycles at the start of each scan line, the DRAM will generate data on every video hardware memory cycle. This eliminates the necessity of more logic to decode the valid times for the VRS.

The word select lines are used only to decode the microprocessors request into the proper bank, as opposed to the bit planes where the word select lines were used to select between banks for all memory requests. Like the bit plane

memory, the VRS memory is both byte and word addressable from the microprocessor. The same WE(L) signal is used for this memory.

VRS Min Plane DRAM (diagram 36)

The VRS Min Plane DRAM stage is comprised of Dynamic Random Access Memory devices U110, U111, U112, and U113. Since the operation of this stage is similar to that of the VRS Plane Max DRAM stage described previously, you must refer to that subheading for a circuit description.

VRS Plane Data Buffers (diagram 36)

The Vertical Raster Scan, Plane Data Buffers stage is comprised of octal bus transceivers U100, U101, U102, U103. This stage provides combined synchronous two–way communications between the Video 0 Data/Video 1 Data busses and the 16–bit Video Data output bus.

VRS Generation and Control (diagram 36)

The VRS Generation and Control stage is comprised of custom gate array U125, and associated components U124, U224A,B,C, U230A,B, U231, and U337A,B,C. Gate array U125 has a gate density of about 3,000 gates. This gate array provides the Display Controller board circuitry with a high–speed method of translating waveform data from the waveform processor circuitry into a displayable image. To this end the U125 gate array is placed between the output of the bit planes and the input to the CRT Driver board circuitry. The U125 gate array is capable of handling a color display system. The VRS (Vertical Raster Scan) system requires nine 29–bit display words to be loaded into the gate array before the VRS display time of each scan line. In the current implementation, there are 160 pixels presented on the current scan line before the VRS system is enabled. The U125 gate array combines the bit plane's video data with the data that it generates to produce the final video output.

The U125 gate array contains a color lookup table and some diagnostics support hardware. These functions are accessed using serial interfaces that are driven by microprocessor U524.

As stated previously, in the current implementation of the Display Controller board circuitry, the VRS display time starts 160 pixels after the start of the scan line. Before the start of the VRS time, the internal registers must be loaded and the device's counters must be initialized. The control signals for these operations are generated by PAL (Programmable Array Logic) U231. The register load signals CSR0(L) to CSR7(L) and CSRC(L), (pins 1–8 and 63, respectively) occur at the end of the video hardware memory cycle. These active low signals are actually generated by the 3 to 8 decoder, U124, by the Enable signal (U231, pin 19), and the low order address lines of the CRT controller. The current implementation makes use of the fact that the low order address lines always progress through the 0–F(*hex*) sequence at the beginning of each scan line, undisturbed by the actions of the mapping hardware, and independent of the actual scan line that is being displayed. In the 160 pixel period of time that the VRS system is being loaded, the video memory actually performs 11 video hardware memory cycles; 10 cycles plus another to cover the pipeline effect of the access time delay.

After all of the VRS display data has been loaded, the PAL (pin 17 of U231) generates an INIT(L) signal to reset all of the counters and state latches within the VRS system. As the VRS time is about to start, the PAL generates another signal at pin 16, U231, which triggers flip–flops U230A and U230B to generate the start signal CNTE (counter enable) at the proper clock edge. At the end of the VRS display time, U231 generates a pulse at pin 15 that resets flip–flops U230A and U230B, and removes the CNTE signal.

The MUART stage generates a variety of signals that allow microprocessor control of the VRS system. The SCALE(L) signal, U231, pin 9, that sets the current scaling factor for the incoming waveforms also drives U125, pin 39, which is the AXMD (axis mode) signal. This determines the number of axes to place upon the screen, one or two. The BLANK(L) signal, U125 pin 47, allows the microprocessor to blank the video output of the Display Controller board circuitry. If the BLANK(L) signal is active, all the video outputs will be driven low. The VRS signals CMPD and DIND (U125, pins 44 and 52) for Color Map Data and Diagnostics Input Data, respectively, are driven by the Data Out signal from the MUART stage. The CMPD and DIND signals supply input data to the color map hardware and to the diagnostic hardware, respectively. The VRS signal CMPC (color map clock), U125 pin 45, is driven by the MUART generated COLOR CLK signal to allow the data upon the CMPD (pin 44) to be clocked into the gate array. Similarly, the DINC (diagnostics input clock), pin 51, when driven by the VRS IN CLK signal from the MUART stage, will allow the data present upon the DIND (pin 52) to be clocked into the diagnostics hardware. The DOUT (pin 59) signal from gate array U125 is an output that is connected through some diagnostics hardware to the MUART stage's DATA IN (U523, pin 34). This allows the microprocessor to read the results of the diagnostics tests when the VRS input signal DOC (diagnostics output clock) U125, pin 49, is driven by the MUART stage's VRS OUT CLK signal. Finally, the VRS signal GATE is driven by the MUART stage's VRS GATE signal which provides a trigger control mechanism for the diagnostics functions.

Address line DTRG is sent from the CRT controller and Select stage to gate array, U125 pin 48, to provide a scan line position signal to the diagnostics hardware. This is connected to CR5 which will cause the diagnostics trigger to occur on scan line number 512, decimal.

The U125 gate array is clocked with a 32 MHz signal (on pin 43) which is used as the pixel rate clock. The bit plane inputs are BP0, BP1, and BP2, pins 55, 54, and 53, respectively. As mentioned previously, BP0 is not used in this implementation. There is a delay between the bit plane data entering the VRS gate array and the resultant data being placed upon the video outputs. This will always be a fixed number of pixel clock periods, as there are numerous pipelining delays throughout the VRS system. The video outputs of gate array U125, R OUT (red or normal), G OUT (green or dim), and B OUT (blue) are buffered by U337A,B, and C, and sent to the CRT Driver board circuitry. It should be noted that while this implementation of the Display Controller board circuitry can only handle a four intensity monochrome display, the video output can drive a three-gun color system, as it is only the limitation of the bit planes that prevents color displays.

Display Driver Interface (diagram 36)

The Display Driver Interface stage is comprised of U517C, U533C, and U334A. This stage provides the control and video signal interface to the CRT Driver board.

A8 CRT Driver Board <38>

A schematic diagram of the CRT Drive circuitry is given on diagram 38, in Section 6, Diagrams and Circuit board Illustrations in Volume 2 of the service manual. The schematic is divided by gray shaded lines separating the circuitry into major stages.

The CRT Driver board holds the circuitry necessary for driving the raster scan CRT. The video and sync signals from the Display Controller board are used in generating the Z–Axis, sweep signals, and grid bias voltages for the CRT. Since the CRT is operated in portrait mode rather than the more usual landscape mode in typical television and monitor applications, the descriptions for the horizontal and vertical sections will seem reversed.

Horizontal Sweep Circuit

The horizontal processor U630 generates the sweep current for the horizontal deflection yoke. The horizontal processor includes a voltage–controlled oscillator, a voltage ramp generator, a high–gain amplifier, and a flyback generator. These circuits provide sweep synchronization, horizontal deflection, and linearity.

Integrated circuit U630 synchronizes to the 59.5 Hz, TTL–level FIELD SYNC signal. The amplifier output at pin 1 provides the horizontal yoke sweep current. During the sweep, C623 charges to approximately 14 V. When the sweep reaches the right side of the screen, the voltage on C623, added to the 14 V supply at pin 11, is internally applied to the pin 1 output. This produces the +28 V flyback signal that causes the horizontal deflection beam to retrace to the left side of the screen.

The HORIZ SIZE adjustment R621 sets the magnitude of sweep current, thereby setting the amount of deflection. The HORIZ HOLD adjustment R620 is part of an RC timing circuit that produces a ramp at the oscillator input pin 6. Adjusting the ramp's frequency rate to approximately 59.5 Hz causes synchronization to occur. The HORIZ LIN adjustment R541 compensates for any nonlinearity in the ramp–generating circuitry.

Vertical Sweep Circuit

The major parts of the vertical sweep circuit are the vertical processor U420, the deflection yoke, the S–shaping capacitor C230, the retrace capacitors C231 and C131, the clamping diode CR240, and the switching transistor Q130. These parts work together to produce a deflection current that sweeps the video beam from the bottom to the top of the CRT. This circuit also produces a flyback signal to the flyback transformer T110 that is in parallel with the deflection yoke.

The vertical processor U420 provides a drive signal for the sweep circuit. This square–wave drive signal, output at pin 1, is coupled through Q320 and T330 and applied to the base of the switching transistor Q130. The transistor's output signal is synchronized to the RASTER SYNC(L) signal through an integrating circuit at pin 4. U420 internally compares the two inputs (pin 3 and pin 4) for a phase shift between the two. If a phase shift is detected, the phase detector's output at pin 5 adjusts the oscillator's output frequency at pin 1.

Transistor Q130 is turned on while the beam is sweeping toward the top of the screen. When the base drive signal turns Q130 off, the beam is at its maximum sweep (deflection) and the voltage at Q130's collector flys up to approximately 350 V. This is the flyback voltage. At this time, the yoke current quickly ramps to a negative value, causing the beam to retrace to the bottom of the screen. Diode CR240 conducts during the first half of the sweep.

Due to several factors (e.g., a flat image screen, non–linear components, etc.), a linearity coil L210 is included to provide current ramp shaping. This affects the linearity of the screen image. L210 controls image shaping on bottom of the screen. VERT SIZE adjustment, L120 (part of the deflection yoke) allows a total circuit inductance adjustment

that changes the current ramp for the desired beam deflection. The VERT HOLD adjustment, R530, is part of an RC timing circuit that produces a ramp at the oscillator's input (pin 7). Adjusting the ramp's frequency rate to approximately 35.7 kHz causes synchronization to occur. The VERT POS control R520 provides a variable phase shift to the feedback ramp at pin 4, thereby allowing a vertical shift in the screen image.

If an overvoltage condition arises due to an increase in flyback–generated supply voltages, the Zener diode VR320 increases the voltage to the oscillator within U420. The resulting increase in the output frequency to the flyback transformer causes voltage limiting.

High Voltage and Grid Voltage Generator

The 11.5 kV CRT anode potential and other bias voltages are generated by the flyback transformer from the flyback waveform. This transformer is in parallel with the yoke and also supplies some of the sweep current for the yoke winding. The following list specifies the voltages generated, and their uses.

11.5 kV	for CRT anode
400.0V	for acceleration grid and focus grid, pin 7.
65.0V	for overvoltage limiting (by VR320)
– 65.0V	for MAIN BRIGHT control

Z–Axis Amplifier

The video signals from the Display controller are decoded by U400 to give four output levels; off, dim, norm and bright. These signals are amplified by Q300. The resulting output is applied to the CRT through peaking inductor, L200. The Z–Axis amplifier Q300 operates in a common–base configuration with a 50 MHz bandwidth.

The MAIN BRIGHT adjustment R202 controls the bias voltage on grid 1 of the CRT to accommodate variations in CRT characteristics.

The intensity of the CRT image is set by the output level of the D/A converter consisting of U600 and R500 – R506. This output level sets the base voltage of Q300 via emitter follower Q510 and temperature–compensating diode CR510. The level of the D/A converter is controlled by the output of shift register U601, which is set by the serial input MON DATA and clocked in by MON CLK. On power–up, U601 is reset by R600 and C600 for maximum intensity.

A9, A10 and A11 Circuit Boards Front Panel

These descriptions are for the functional blocks that include the Touch Panel, major menu keys (hard keys), and Menu Status LED drive. These three functions are interfaced to the Executive Processor through U103 (on the front panel control board), a general purpose programmable keyboard and display controller.

The keyboard function of the IC handles the touch matrix and hard keys. The display function drives the menu LED light bars.

Throughout this document, U103 is referred to as the 8279, Intels designation number. The 8279 is programmed to be in the "Encoded Scan Sensor Matrix" mode. In this mode, an image of the touch matrix and hard keys are stored in a block of RAM referred to as "Sensor RAM." This RAM is internally organized as eight bits by eight bits. See "8279 Sensor Memory Array" table in the back of this document to see how each bit physically maps onto the front panel.

The 8279 notifies the Executive Processor that the Touch Panel or hard keys have changed states and needs servicing by asserting the IRQ interrupt output. An interrupt is generated whenever the "image" changes in any way.

Touch Panel

The infrared LEDs produce a matrix of light beams that are interrupted by the users finger when pointing to a particular touch zone. The touch panel and hard key matrix are scanned continuously until a "shadow" or keypress is detected. When a hit is detected, that scan is completed and the interrupt line is asserted by the 8279. During the time the interrupt is active, no new data is written into the sensor ram from the Touch Panel or Hard Keys even though the hardware continues to scan. This is so the data does not change in the Sensor RAM while the processor is reading it.

Only one infrared LED is turned on at a time and only the phototransistor directly opposite is selected to receive light. This prevents any crosstalk between emitter/detector pairs.

IR Detector

All 33 phototransistors are multiplexed to a common line and applied to the input of U700 and associated circuitry. R803 sinks current from the phototransistors depending on light level. R804 and C801 form a highpass filter to eliminate the bulk of the ambient light which is composed mostly of 60 Hz and 120 Hz.

CR700a clamps the large positive switching spikes generated from the multiplexers selecting a new phototransistor.

U700a and associated circuitry detects the relatively small rising transition as a result of the IRLED's coupling light to the phototransistors. Feedback diode CR700b allows U700a to have very high gain in the positive direction and unity gain in the negative direction to achieve high sensitivity and still have adequate response time.

U700b and the four associated resistor shift the 0 to P5 V levels from the previous stage to TTL compatible levels.

Touch Panel Address Generator

U213 (Front Panel Control board) generates the 6–bit address bus that is used to select an infrared LED and its compliment phototransistor on the Touch Panel board.

U211 buffers the Touch Address bus before leaving the Front Panel Control board.

Clock and Control generator U111 and associated resistors and capacitor generate a square wave at approximately 90 kHz that drives U103 and U114 clock inputs.

U210a generates a pulse called "Sync" that is used to synchronize the Touch Address bus to the 8279 (U103) on the falling edge of CNT7(L) from the select lines SL0–SL2 on U103. This trailing edge triggers the one-shot that generates a 500 ns wide pulse.

Along with clearing counters U213a and U213b, this sync pulse causes a value of 7 to be loaded into the clock generator U114, which skews the Touch Addresses ahead by approximately 70 µs. The reason for the skew is that the 8279 inherently samples the data on the selected Return Line at the CENTER of its count duration which results in wasted time following the sample that could be used to set up the analog multiplexers next selection. It also allows more time for the turn on delay of phototransistors.

InfraRed Enable Pulse (IREN(L))

This pulse ultimately turns on the selected Infrared LED on the Touch Panel board. U113a, U115b, and U212c use various clocks from U114 to generate the Infrared Enable pulse. IREN(L) goes active (low) 50 µs before U103 samples the selected Return Line. This allows ample time for the phototransistor to turn on and stabilize.

The Sync(L) inputs to U113 ensures that the flip–flop's start and stay in proper phase.

The IRDIS(L) input to U212c is used for diagnostics to disable the IREN(L) pulse which inhibits all IRLED's from turning on. This gives the ability to check that all Touch zones change states showing there are no shorted runs and/or phototransistors.

Latch Enable pulse (LEN(L))

This signal causes the incoming serial data (SDATA) to be latched into the selected registers of U1001 The 8279 requires that all eight return lines be held stable until the last return line (RL7) is sampled, at which time the byte is loaded into the Sensor RAM. Although the data is dumped into the RAM a byte at a time, each incoming bit is individually compared with the same bit already stored in RAM. If a difference is found, the 8279 sets an internal flag that enables the IRQ line to be asserted when the current scan is completed.

The rising edge of LEN(L) occurs at the same time the 8279 samples the selected return line.

Soft/Hard key Decoding

For this section, refer to the "8279 Sensor Memory Array" table in the back of this document when necessary.

The first eight bits (byte 0) are unused because of a problem inherent in the 8279. U760 and U761 on the Touch Panel board decodes the next five bytes, 1 through 5, and are mapped to the X– and Y–axes of the touch grid. Byte 6 is used for the Major Menu Hard Keys and are decoded by U761 on the Touch Panel board. Byte 7 is decoded on the Front Panel Control board by U212a and enables the 74LS151 to route the Front Panel button board Hard Key states to the 8279.

Menu Status LED Drivers

The Major Menu LED light bars are driven by the display refresh register outputs of the 8279 (on the Front Panel Control board). Internally there is a block of display RAM organized as eight by eight bits. With respect to the "8279 LED Display Memory Array" table in the back of this document, the display RAM is scanned column by column, automatically, lighting the appropriate LED bar(s) when a high bit is encountered. U200, U203, and U214 allow blanking with BD(L) signal during switching and drive the transistors in U205 and U216 that sink the LED current.

80286 to a Hardware–Software Interface

8279 Modes and Commands:
The following commands program the 8279 operating modes. All commands are sent to the control register at address 3302*hex*. All data is read from or written to address 3300*hex*.

Initial setup

Keyboard/Display Mode Set-04*hex*
Keyboard mode:

This input mode is set to "encoded scan sensor matrix."

Display mode:

This output mode is set to "eight 8–bit character display, left entry." Thus, after power up and/or RESET, a value of "04" should be written to the control register to set these modes.

Program Clock-22*hex*
This internal clock prescaler should be set to a value of 2, thus, a value of "22" should be written to the control register only after power–up and/or RESET.

Display Blanking-A0*hex*
This value should be sent to the control register initially to prevent the display outputs from powering up in the blanked state.

Clear command-C1*hex*
Writing this value to the control register clears all bits in the display RAM and resets the Sensor RAM pointer to row 0. It also resynchronizes the internal timing chain. When this command is used, no data can be written to the display RAM for 160 μs.

Touch Panel and Hard Keys

Read Sensor RAM-50*hex*
In order to read the contents of the sensor RAM, this command must first be written to the control register. Upon receiving an interrupt, writing a value of 50*hex* sets the first read to be from the first row of RAM and sets the auto–increment flag so that each successive read will be from the next row of RAM. To read all 64 bits of Sensor RAM, make eight consecutive reads from the data register after writing this command. Refer to the tables and front panel drawing in the back of this document to relate memory bits to actual physical screen locations.

Once the "image" changes and the 8279 asserts the interrupt line, further writing into sensor RAM from the Touch matrix and Hard Keys is inhibited until the interrupt line is reset.

End Interrupt-EO*hex*
Writing and EO_{hex} to the control register resets the interrupt line, usually after reading the Sensor RAM contents.

Menu Status LEDs

Write display RAM-90*hex*
The EXP sets up the 8279 for a write to the display RAM by first writing this command. Writing the value 90_{hex} to the control register places the first data write into the byte 0 column and sets the auto–increment flag such that all subsequent data writes will be to the next column of display RAM. With respect to the "8279 LED Display Memory Array" table in the back of this document, the LEDs are scanned from left to right. The LEDs turn on with a high in RAM, so, for full intensity, set all eight bits to 1 s in the appropriate row.

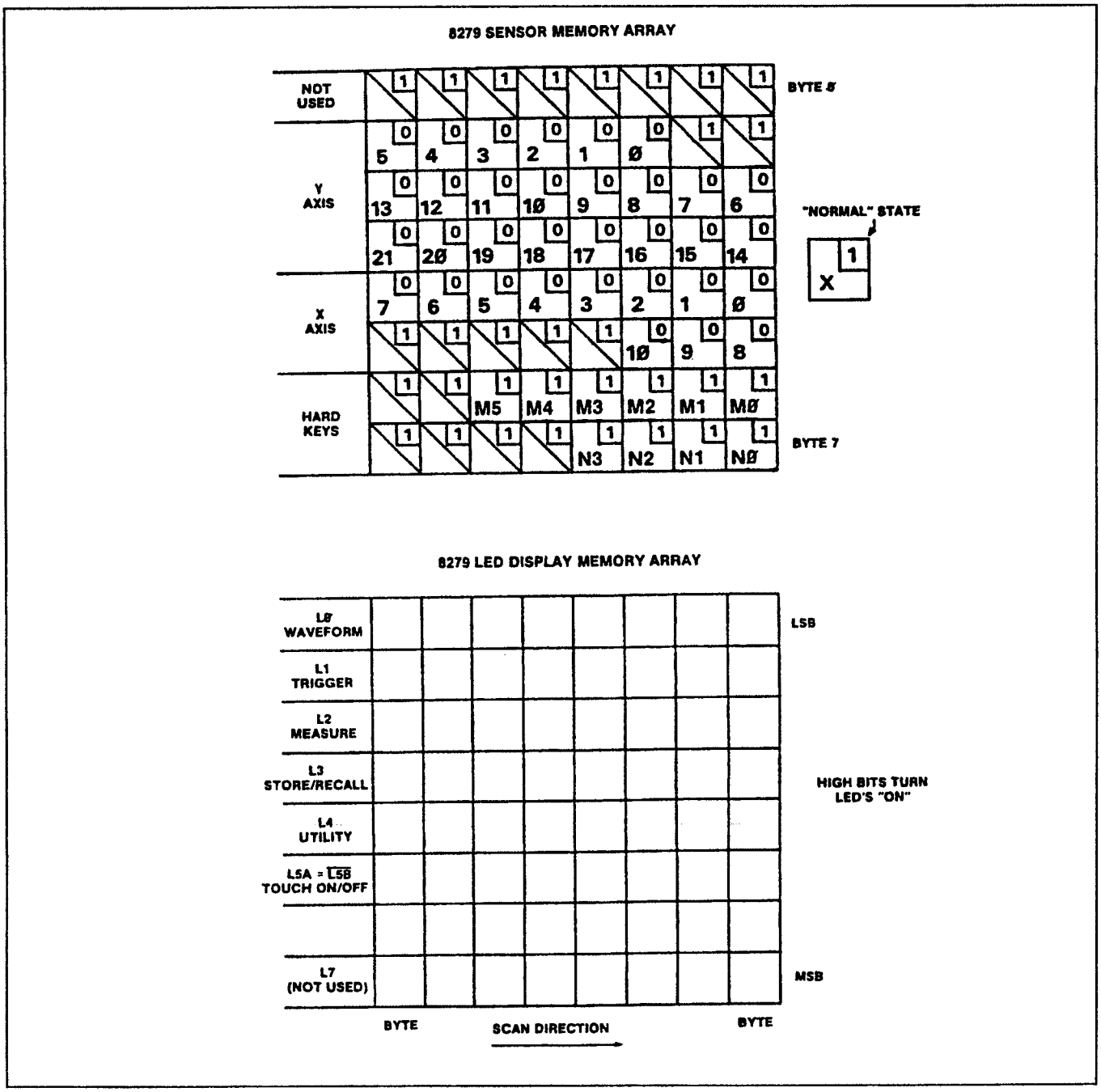


Figure 2–4. Display and Sensor Memory Arrays

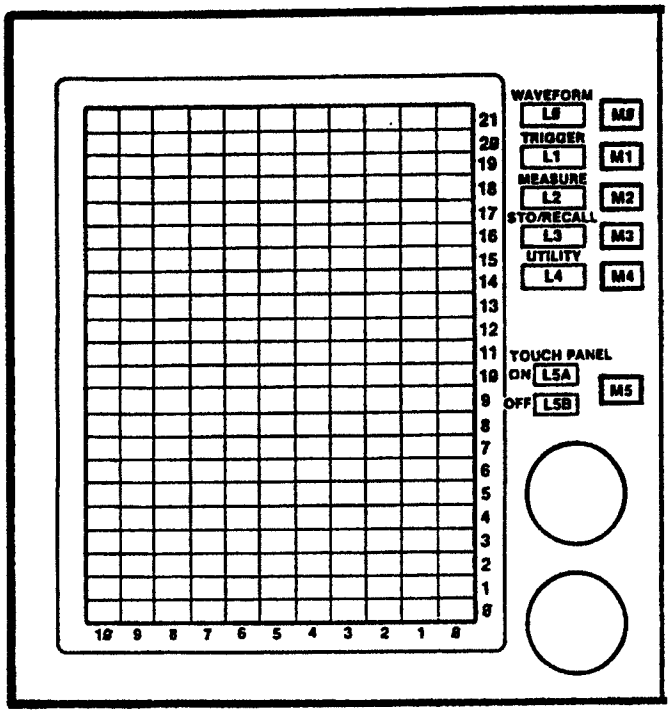


Figure 2-5. Touch Panel Layout

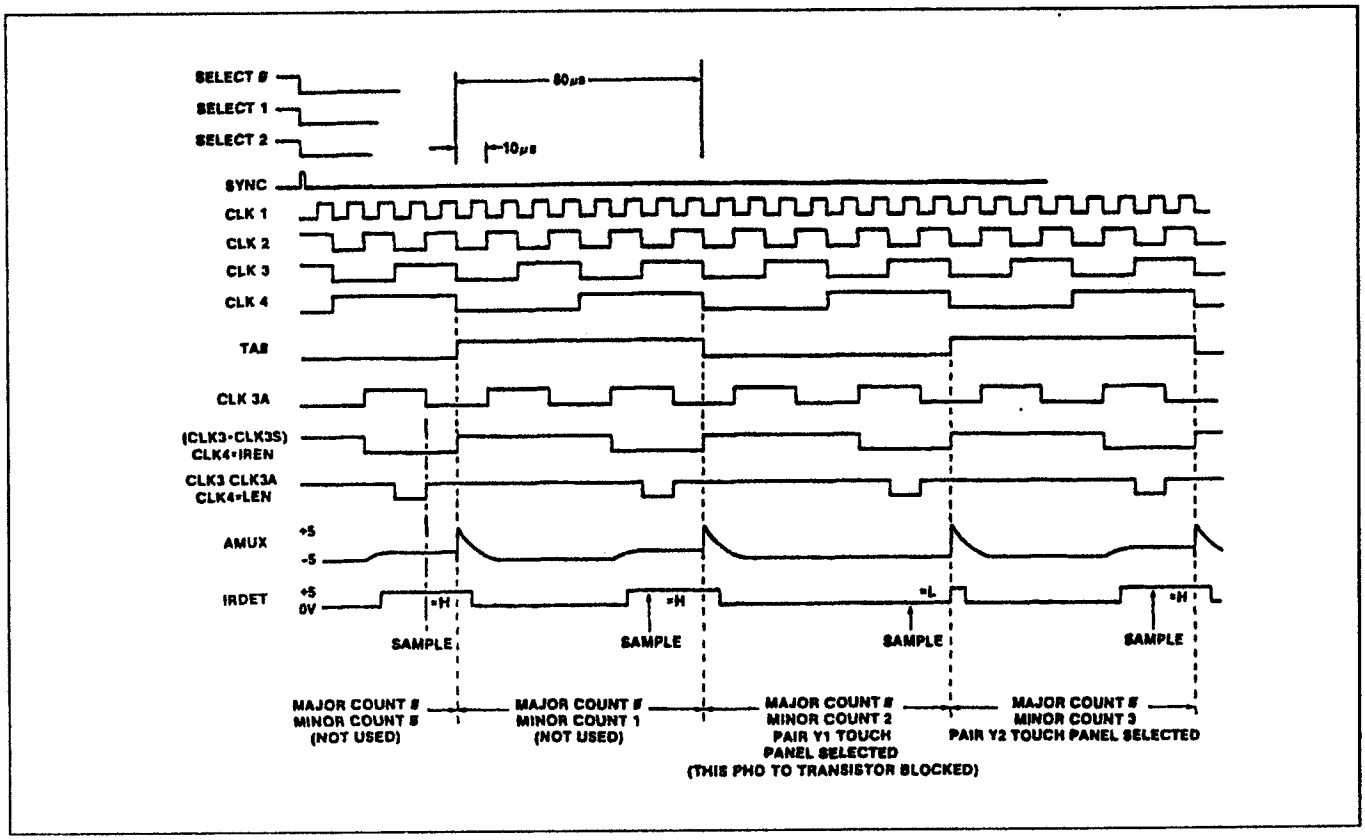


Figure 2–6. Front Panel Control Timing

A12 Rear Panel Board <3> and <4>

This board is the oscilloscope's link to the outside world. It contains connectors for one GPIB Port, one RS–232 Port, and one Printer Port (Centronics style).

The Rear Panel board is controlled from the I/O board through a 40–wire cable (J78). This cable contains an 8–bit bidirectional data bus, a 4–bit address bus, four device select lines, four interrupt lines, GPIB DMA request and grant lines, four device control lines, and assorted power supply and ground lines.

The data bus and address bus drive the GPIB controller IC (U410) directly. The microprocessor is trying to read or write it. Control signals DBIN, and WR(L) are used by U410 to determine if the current microprocessor, it pulls the GPIB INTR(L) line low. The interrupt controllers on the Main Processor board monitors this interrupt line and will signal the microprocessor to service U410. Among the services needed by U410 are: the receiver section of the IC has a byte of data (Inbyte register) from the GPIB bus that the microprocessor needs to read. The transmitter section's register (Outbyte register) is empty and is ready to receive another byte of data. The status of the GPIB bus or U410 has changed and the microprocessor has to be notified.

On the opposite sides of the GPIB controller IC is another bus system. There is an 8–bit data bus which goes to a bi–directional GPIB Data buffer U500. There is an 8–bit control bus which goes to a GPIB Control Driver (U510). The signal flow direction here is done in groups of 3 and 4. The TE (TALK ENABLE) signal from U410 controls the direction of U500 and DAV, NDAC, NRFD of U510. The CONT (Direction Control) signal from U410 controls the direction of ATN, SRQ, REN, and IFC of U510. The GPIB bus is connected to the opposite side of the U500 and U510. U500 and U510 are specially designed to be TTL signal level compatible on the U410 side and GPIB bus compatible on the bus side of the buffer driver.

U600C allows the GPIB Data buffer (U500) to be in either of two modes. With U500–11 (Pull–up Enabled) high, the GPIB bus sides of U500 has active pull–up drivers enabled for higher speed communications. This is the normal operation.

With U500–11 low, these outputs have only resistor pull–ups. The data must be transferred slower, but the resistors allow parallel polling of devices on the GPIB bus.

The state of three control signals from U410 are monitored and displayed on the rear panel of the 11401. These are SRQ (DS700), NRFD (DS710), NDAC (DS711). These LEDs (buffered by U600) show the state that U410 is in, not the state of the GPIB bus.

Two other signals of interest are GPIB RQ and GPIB GR(L). These are used by the DMA controller on the Main Processor board to communicate with the GPIB controller (U410) (if the DMA IC is installed). The DMA can be programmed by the microprocessor to service either the Inbyte register or the Outbyte register, but only one at a time. Once the DMA has its instructions from the microprocessor, it waits from GPIB RQ (GPIB Request) to go high. When the GPIB RQ goes high, the DMA request the System bus from the microprocessor by setting the microprocessor HOLD signal high. Usually within 2 to 10 μs the microprocessor finishes its current instruction an turns the System bus over to the DMA. It signals the DMA that it can use the System bus by setting the HLDA (Hold Acknowledge) line high. The DMA sets the GPIB GR(L) (GPIB Grant) line low to tell U410 the DMA is talking to it. Then U410 sets GPIB RQ low. The data transfer takes place over the System bus, either a read or a write. (At present, the DMA is only used to write to U410.) When the data transfer is complete, the DMA sets the GPIB GR(L) line high and returns the System bus to the microprocessor.

The two diode packs (CR602, CR603) are connected to each of the signal lines on the GPIB bus. They protect the GPIB bus drivers (U500, U510) from static discharge damage.

The Standard RS–232–C controller (U311) is connected to the same Data bus and Address bus as the GPIB controller. RPD to RPDO transfers data to and from the microprocessor. RPA1 to RPA4 address lines are used by the microprocessor to select individual registers in the controller. The STD RS SEL(L) line goes low when the microprocessor wants to communicate with U311. (This line drives the Chip Enable on U311.) The RD(L) and WR(L) signals are driven by the microprocessor to signal if U311 is to be read or written. U311 can request service from the microprocessor by setting the STD RS INTR(L) (interrupt) line low. The microprocessor can transmit data on the RS–232–C bus by writing a byte into the controller's transmit buffer. The microprocessor can receive data from the RS–232–C bus by reading a byte from the receiver buffer. The microprocessor can also read the status of U311. The RS–232–C controller translates the parallel data from the microprocessor to serial data on TXD RS–232–C bus. It also converts the serial data from RXD on the RS–232–C bus to parallel data for the microprocessor. U310 and U210 are transmit and receive buffers which are compatible with the RS–232–C bus. Clear To Send (CTS) and Data Set Ready (DSR) are RS–232–C control signals that can be controlled by the microprocessor. Request to Send (RTS) and Data Terminal Ready (DTR) are signals that can be ready by the microprocessor. Received Signal Detect (RSD) is always high when power is on. This RS–232–C port is a DCE type.

Two diode packs are connected to all the signal lines on the RS–232–C bus. The diodes clamp the lines to a maximum of 115 V. This is to protect U210 and U300 from static discharge to the external connector or cable.

The Printer Port (J111) is controlled by a Programmable Peripheral Interface IC (U430). This IC has all the control lines necessary to connect to a microprocessor, plus two general purpose 8–bit ports and the control signals to use them. The Rear Panel Data bus and Address bus connect to U430 and have the same function as described for the GPIB and RS–232–C controllers. The Printer Sel(L) line is set low by the microprocessor when it is communicating with U430. The RD(L) and WR(L) lines allow the microprocessor to either read or write to the registers in U430. U430 is initialized by the microprocessor for Port A to be a strobed output port and Port B to be a strobed input port. Port C provides the control signals.

To send a byte to the printer, the microprocessor writes the byte to U430's Port A. The data appears at Port A and passes to a buffer IU520) and on to the Printer Port (J111). Next the Port A Output buffer Full (OBF(L)) signal goes low and triggers a one–shot (U330A) which triggers a second one–shot (U330B). U330B pin 12 then pulses low for 1 μs. This is buffered by U541 which drives the printer Data Strobe (DS(L)) signal. When the printer receives the Data Strobe and is ready for the next byte, it returns an Acknowledge (ACK(L)) signal to U541–8. This then drives U430–11 (Port A acknowledge signal). This completes the handshake of the Printer controller with the printer. U430 now sets its pin 17 (INTR) high. This signal is inverted by U610C and becomes an interrupt to the processor to tell it the printer is ready for more data.

This Printer Port can also be put in a test mode which causes the data going out through U520 to return through U540 to U430's Port B. This is done by setting U430 pin 13 (NOR/TEST) low. This enables U540. Port B of U430 handshakes with Port A of U430 when this port is set in test mode.

When the Printer Port is in normal mode (U430 pin 13 high), the printer status is available through Port B. The status information from U541 is strobed into Port B by the microprocessor setting U430 pin 12 (SSTB/INP) low then high. This signal goes through U610B to U430–16 (STB(L) for Port B).

All the signal lines on the Printer Port (J111) are connected to two diode packs (CR605, CR606). These clamp the lines at the maximum of 0 and +5 V. This is intended to protect U520, U540, U541 from static discharge on J111.

Table 2–3
Rear Panel Board Interface J78

Pin Number	Signal Name	Full Signal Name	Definition and Use
1	RESET(L)	Reset Not	System reset from processor board. Used to initialize ICs on the R.P board.
2	D0	Data Line 0	Bit 0 of eight–bit bidirectional data bus, buffered from the microprocessor bus.
3	D1	Data Line 1	Bit 1 of data bus. See Pin 3.
4	D2	Data Line 2	Bit 2 of data bus. See Pin 3.
5	GND	Ground	Power supplies and signals return path.
6	D3	Data Line 3	Bit 3 of data bus. See Pin 3.
7	D4	Data Line 4	Bit 4 of data bus. See Pin 3.
8	D5	Data Line 5	Bit 5 of data bus. See Pin 3.
9	GND	Ground	See Pin 5.
10	D6	Data Line 6	Bit 6 of data bus. See Pin 3.
11	D7	Data Line 7	Bit 67of data bus. See Pin 3
12	4MCK(L)	4 MHz Clock	Clock to drive the GPIB IC.
13	GN	Ground	See Pin 5.
14	A1	Address Bit 1	Bit 1 of buffered and latched address bus from the microprocessor. Used for device selection.
15	A2	Address Bit 2	Bit 2 of address bus. See Pin 14.
16	A3	Address Bit 3	Bit 3 of address bus. See Pin 14.
17	GND	Ground	See Pin 5.
18	A4	Address Bit 4	Bit 4 of address bus. See Pin 14.

Table 2–4
SDI Plug-In Communications Interface J90

Pin No.	Signal Name	Full Signal Name	Definition and Use
1	+15 V	+15 volts	+15 volt power supply to the cardcage, from the Plug-in Interface board.
2	L.SDI.CK	Left SDI Clock	Clock signal that drives the Left Plug-in compartment's serial communications port. Its source is the Std. I/O board.
3	–15 V	–15 volts	–15 volt power supply to the cardcage, from the Plug-in Interface board
4	R.SDI.CK	Right SDI Clock	Clock signal that drives the Right Plug-in compartment's serial communications port. Its source is the Std. I/O board. -
5	PU/D(L)	Power Up/ Down (L)	When this line is high, the power supplies are up and regulating. When the line is low, it warns that the supplies are going down or are down. The signal source is the power supply via the Plug-in Interface board.
6	L.D. PI > MF	Left Data plug-in to oscilloscope	Serial data path. Data flows from the plug–in to the oscilloscope to request set-up info and to return its status.
7	GND	Ground	Signal and supply current return path
8	L.D. MF > PI	Left data oscilloscope to plug-in	Serial data path. Data flows from the oscilloscope to the plug-in to control it and ask for status information.
9	R.D. MF > PI	Right data oscilloscope to plug-in	Serial data path. Data flows from theoscilloscope to the plug-in to control it and ask for status information.
10	GND	Ground	Signal and supply current return path
11	A.D. PI > MF	Auxiliary data plug-in to oscilloscope	Serial data path. Data flows from the auxiliary plug-in to the oscilloscope to request set-up info and to return its status.
12	R.D. MF > PI	Right data plug-in to oscilloscope	Serial data path. Data flows from the oscilloscope to request set-up info and return status.

Table 2–4 (Cont.)
SDI Plug-In Communications Interface J90

Pin No.	Signal Name	Full Signal Name	Definition and Use
13	+15 V	+15 volts	+15 volt power supply to the card cage, from the Plug-in Interface board.
14	A.D. MF>PI	Auxilliary Data Main-frame to Plug-in	This is the serial data path. Data flows from the mainframe to the aux. plug-in to control it and ask for status information.
15	A.SDI.CK	Auxilliary SDI Clock	This clock drives the Aux. Plug-in compartment's serial communications port. Its source is the Std. I/O board.
16	GND	Ground	Signal and supply current return path
19	RD(L)	Read Not	If this is low, the selected LSI IC is being read by the microprocessor. It shouldput data on the data bus (D0–D7).
20	WR(L)	Write Not	If this is low, the selected LSI IC is being written to by the microprocessor. It should store the data from the data bus on the rising edge of WR(L).
21	GND	Ground	Signal and supply current return path.
22	DBIN	Data bus In	If this is high, the GPIB IC is being read by the microprocessor or if it is low during a DMA grant to the GPIB, the GPIB IC is being read by the DMA controller.
23	GPIB.RQ(L)	GPIB Request Not	If this is low, the GPIB IC is requesting service from the DMA controller. It is used for higher speed data transfers.
24	GPIB.GR(L)	GPIB Grant Not	When low, the DMA controller is granting service to the GPIB IC. DBIN tells the GPIB the cycle is a read or a write.
25	GND	Ground	Signal and supply current return path.
26	GPIB.SEL(L)	GPIB Select Not	When low, the GPIB is selected to communicate with the microprocessor.
27	STD.RS(L)	Standard RS232 Not Select	If this low, the standard UART is selected to communicate with the microprocessor.

Table 2–4 (Cont.)
SDI Plug-In Communications Interface J90

Pin No.	Signal Name	Full Signal Name	Definition and Use
28	OPT.RS(L) SEL	Optional RS232 Select Not	When low, the optioanl UART is selected to communicate with the microprocessor
29	Centronic(L) .SEL	Centronic Not Select	When low, the Parallel Interface IC is selected to communicate with the microprocessor
30	GND	Ground	Signal and supply current return path
31	GPIB.INTR(L)	GPIB Interrupt Not	When low, the GPIB IC is requesting service from the microprocessor via the interrupt controllers.
32	STD.RS(L) .INTR	Standard RS232 Interrupt Not	When low, the standard UART is requesting service from the microprocessor, via the interrupt controllers.
33	OPT.RS(L) .INTR	Optional RS232 Interrupt Not	When low, the optional UART is requesting service from the microprocessor, via the interrupt controllers.
34	Centronic(L) .INTR	Centronic Interrupt Not	When low, the Parallel Interface IC is requesting service from the microprocessor, via the interrupt controllers.
35	GND	Ground	Signal and supply current return path.
36	GND	Ground	Signal and supply current return path.
37	+12 V	+12 volts	The +12 volts to the RS-232-C line drivers.
38	+5 V	+5 volts	The +5 volts to the ICs on the Rear Panel board.
39	+5 V	+5 volts	The +5 volts to the ICs on the Rear Panel board.
40	–12 V	–12 volts	The –12 volts to the RS-232-C line drivers.

A14 Input/Output Board <21> and <22>

The Standard I/O board is an interface between the Executive Processor (EXP) and communications ports (i.e., RS–232–C, GPIB, etc.), devices on the Front and Rear Panel boards, and the plug–in units. It also contains the System Timer, the Real Time Clock, the Serial Data Interface device, the Temperature Sensor and the Tone generator. The EXP reads and writes to these I/O devices and the communication ports at specific I/O addresses. These I/O addresses are decoded to produce device select signals which enable the addressed device. Each I/O device is located on I/O address boundarys of at least 100hex. Except for the exec bus interface, all the interfaces are via these flexible cable connections:

- J72–Front Panel Control Board
- J77–Main Processor Board
- J78 –Rear Panel Board
- J90–Plug–In Interface Board

The lower eight bits of the Executive Data bus are used to transmit data to and from the various I/O devices. The EXP uses this byte of data to configure the various I/O devices and to read their statuses. Naturally, only one I/O device can be accessed at a time.

When the DMA controller (option 4D) is installed on the Main Processor board, the Standard I/O board handles GPIB operations differently. See the heading "GPIB Control" for details of the GPIB interface.

I/O Data Buffer (diagram 21)

The lower eight bits of the Executive Data bus from P105 are buffered by the I/O Data buffer U832. The output is the I/O Data bus, which drives data to six different on–board devices. (I/O delayed data buffer, rear panel data buffer, front panel data buffer, DAC data latch, tone/temp readback buffer, timer configuration logic [latch]).

I/O Delayed Data Buffer (diagram 21)

The I/O Delayed Data buffer interfaces between the I/O Data bus and the Write Delayed Data bus. BDEN, WDDEN(L), and OBSEL are NANDed together to enable the buffer. BDEN is a buffered version of the EXP Data Enable signal. WDDEN(L) is activated by the I/O Control circuitry to keep BDEN from immediately turning on the I/O Delayed Data buffer at the beginning of a write cycle, when the previous bus cycle was a read. The devices using the Write Delayed Data bus are MOS parts and take a long time (100 – 200 ns) to turn off their output drivers. The delay ensures that only one device will drive the data bus at a time. The other enable signal input to U520B is OBSEL, which is generated by the Address/Decode Select circuit to indicate selection of an on–board I/O device.

Timer Configuration Logic (diagram 21)

The Timer Configuration Logic is comprised of latch U720 and three 2–input data multiplexers built with discrete gates. When the EXP writes to I/O address 3200hex, LS4(L) and BIOWC(L) go active and latch the I/O data bus. Some of the latched bits are used to individually configure the way that Counters 1 and 2 are used. This lets the Timer accept different inputs for different system tasks. U712A and B and U714A determine whether 6 MHz PCLK or the

Executive bus signal DIAGNSIG is passed to the CLK input of Counter 2. DIAGNSIG is used to signal the occurrence of a diagnostic test or an operation in a test. The output of a circuit being tested is placed onto DIAGNSIG. U714D supplies the Gate input, G2, with either a constant enable or with DIAGNSIG. When DIAGNSIG is selected as the gate, it can allow PCLK pulses to clock Counter 2. This gives diagnostics the ability to determine how long a DIAGNSIG pulse lasts in that the number of PCLK pulses recorded can be used to compute the duration of the DIAGNSIG gate pulse.

The Counter 1 multiplexer consists of U714C, U712C and D and allows PCLK or the output of Counter 2 to be fed into the CLK 1 input. Cascading the two counters provides for longer counts.

The two lowest bits out of latch U720 can be used to invert the outputs of Counter 1 and Counter 2. This is necessary because the Timer can be set in different modes, and some of these result in an active low output when a counter event occurs. The counter outputs must be active high out of U820 to drive the EXP interrupt pins. The inputs and modes of the counters can be changed at any time.

Counter 0 is used by the operating system as a real–time clock based on the constantly enabled 2 MHz CLK input from the Clock generator. The Counter 0 output is used to clock U812B. On a positive–going edge, the high input (D) appears on the output (Q) as an AX interrupt to the EXP (IR2X). To clear the interrupt from U812B, the system does a read from the Timer Configuration Logic port at 3200$_{Hex}$. Counter 0 runs continuously and generates a positive edge interrupt every time it reaches a maximum count. Then Counter 0 resets to zero and continues counting while the EXP services the Timer interrupt.

Real Time Clock (diagram 21)

The Real Time Clock is comprised of U614 and its oscillator circuit. It keeps track of the current time of day, which is set and read by the EXP. The chip select LS9(L) allows access to the internal registers for I/O reads (BIORC(L)) and writes (BIOWC(L)) of date data. The WR input is driven by BIOWC(L) gated with DLYIO to satisfy timing requirement of U614. Jumper J230 is removed for shipment so the Lithium battery, BT130 which powers the Real Time Clock when the main supply is off, is disconnected if moisture condenses on the board. Oscilloscope setup instructions direct the technician to replace jumper J230. Battery backup switching is built into the chip to sense when the main supply voltage is low, and enable the battery supply. Also, if the jumper is removed the RTC IC may not operate normally even when the oscilloscope power is on.

CAUTION

Lithium batteries can be dangerous when they are discharged too fast, when more than a very small charging current is applied, or when they must be changed, (see the warning on Lithium batteries in the maintenance section of this manual).

C510 can be used to adjust the oscillator frequency. Putting a probe on the OSC IN or OSC OUT lines will cause a shift in frequency. So, to calibrate the Real Time Clock, software must set it up in a mode to produce an interrupt once every second. Then an external timer–counter is used to measure the period from a falling edge (TP310) to the next falling edge while adjusting C510 to set the period to exactly one second. Six internal registers are addressed with the address lines and loaded from the Write Delayed Data bus with time of day data. Additional registers allow an alarm function to generate an interrupt output when a specific time occurs. The user can enter the local time after battery jumper J230 is in place.

Serial Data Interface (SDI) (diagram 21)

The Serial Data Interface (SDI), U330, is a custom chip that provides serial data communication between the EXP and the three plug–in slots and with both Front Panel knobs. It is controlled by the EXP and it interrupts the EXP

when a device requires service. The EXP controls the SDI with I/O writes (BIOWC(L)) and reads (BIORC(L)) to I/O address 2000_{hex} – $203E_{hex}$. The PCLK input is clocked with the 8 MHz signal from the on–board Clock generator. Separate address and data buses are used even though the eight data lines are marked as AD.

The plug–in interface allows communication with the various plug–in units which may reside in the oscilloscope. SDINs and SDOUTs are serial data inputs and outputs, respectively. The current–limiting 100 Ω in–line resistors and the clamping diode packs CR230 and CR231 give protection from damage due to inadvertent insertion or removal of plug–in units with the power on. Pull–up resistors (10k Ω) are attached to both the inputs and outputs and tied to 5 V to pull the lines high when no plug–in is installed. These input and output lines can be read internally by the SDI to determine if a plug–in is installed in each slot.

Input SDICLK monitors CLK1OUT, which is half the frequency of the 8 MHz, Pin 2 PCLK input. CLK1OUT is used to synchronize serial data transfers and is buffered by U320C to reduce loading on the SDI chip. To keep the clock polarity the same as CLK1OUT and to keep a shorted clock line on one plug–in from disabling the other two, the clock signal is inverted and buffered separately for each plug–in channel.

I/O Address Latch (diagram 21)

Executive Address bus lines A0 – A7 are latched by U632 as ALE goes low and is used to drive various devices.

Address Decode/Select (diagram 21)

PALs U722, U732, and U730 are used to generate all chip selects for on–board devices and for devices on the Front– and Rear Panel boards. Executive address lines A0 and A8 – A15 and EXP control lines M/IO and COD/INTA are decoded to produce device select lines S0(L) – S17(L). The device select lines and their associated I/O addresses are listed in Table 2–5.

Table 2–5
I/O Addresses for Device Selects

PAL	I/O Address (hex)	Select Line	Select Name
U722	2000–203E	S1(L)	SDI Sel(L)
		S2(L)	Not Used
	3100–3106	S3(L)	Timer Sel(L)
	3200	S4(L)	Timer Confsel(L)
	3300–3302	S5(L)	Touch Panel Sel(L)
	3400	S6(L)	TP LED Sel(L)
	3500	S7(L)	not used
	3600	S8(L)	not used
	3700–373E	S9(L)	RT Clock Sel(L)
	3800	S10(L)	Tone Gen Sel(L)
U732	3900	S11(L)	Temp Sense Sel (L)
		S12(L)	not used
		S13(L)	not used
	4100–410E	S14(L)	GPIB Sel(L)
	4200–421E	S15(L)	RS-232-C Sel(L)
U730	6100	S16(L)	Opt. RS-232-C Sel(L)
	6200	S17(L)	Printer Sel(L)

The device select lines are latched by U622 and U630 with the buffered Address Latch Enable signal, BALE. The buffer outputs reflect the inputs while BALE is high, and the outputs are always enabled by the U420E output. The latched select lines go to on–board devices and to the Front– and Rear Panel boards. The Latched Select lines are also gated to produce buffer enable lines for the I/O board (OBSEL), the Front Panel board (FPSEL) and the Rear Panel board (RPSEL).

The unlatched, device select lines are gated to produce an early wait request for the EXP that will result in one to four EXP clock (PCLK) cycles being added to the current bus cycle. Later, when the latched select lines become valid, they are gated to request a specific number of wait states for the current bus cycle. The signals connected to U430 but not to U522A or U530 produce the default of four wait states inserted after the wait request. DMA0 SEL(L) acts as a chip select for DMA operations on the Rear Panel board. DMA0 SEL(L) used to request one wait state for the GPIB controller chip to give it time to export data.

On Board Power Circuits (diagram 22)

The +15 V and P15 V voltages reach the Standard I/O board via the plug–in interface connector, J90. They are fused by F600 and F602, respectively. The fused supplies are then decoupled with filter capacitors C502 and C701 and connected to the card cage Mother board to supply other Executive boards. U300 and U400 generate +12 and P12 V for the RS–232–C line drivers on the Rear Panel board. U110A and VR100 produce a precise +6.5 V, which is used as a reference voltage by the Temperature Sensor.

The +5 V current for the Front– and Rear Panel boards is routed through the Standard I/O board. Fuse F200 (1 A) protects the Rear Panel board from overloads. Likewise, F800 fuses the 5 V current for the Front Panel board. The fuses have indicators on the board that point at their output ends for easy functional checks.

Temperature Sensor (diagram 22)

The EXP uses the Temperature Sensor to get a digital reading of the temperature in the oscilloscope. When a specific temperature change has occurred, the oscilloscope can automatically recalibrate itself. The circuit consists of the thermal sensing element U130, operational amp U124 and comparator U122. U130 produces an output current that is proportional to its temperature (1 μA/° C). This current produces a voltage differential across precision resistor R134, which sets the transconductance (I →V) of U124. The resistor network on pin 3 produces a reference offset voltage (2.45 V) for U124 which can be calibrated with R110 (if present). The Temperature Sensor is calibrated at one temperature and should remain usably accurate over its range of –27 to 100° C. The accuracy of the 6.5 V supply and resistors is critical for accurate operation. As the temperature rises, the output of U124 goes more negative.

Comparator U122 senses the output of the op amp and compares it to the output of digital to analog converter (DAC), U212. The pin 4 output of U212 sinks a maximum of 1 mA, which produces –1.27 V across R130 and on pin 2 of U122. A value of 0 into the DAC causes pin 2 to sink 0 mA, which produces 0 V on R130.

The Temperature Sensor is read by the EXP with repeated writes and reads following the successive approximation algorithm. First, a byte of data is written to the DAC at I/O address 3900_{hex} with only the most significant bit (IOD7) set high. This produces a voltage which is compared with the output of sensor op amp U124. The result of the comparison is read by the EXP with the Temp/Tone Readback buffer. The result, high or low, will become the state of the last tried bit. Next, each lower bit is added in turn, converted and compared. When the least significant bit has been tried, the result is the temperature value.

DAC Data Latch (diagram 22)

Latch U610 latches data from the I/O data bus on the rising edge of BIOWC(L) when LS10(L) or LS11(L) is active. The latched data is fed directly into the D/A converter U212. The outputs are constantly enabled.

Temp/Tone DAC (diagram 22)

The Temp/Tone Digital–to–Analog Converter changes an eight–bit digital number (0 – 256) into a proportional current (0 –1 mA) on both its outputs. Both DAC outputs are activated at the same time, but the pin 4 output increases in current while the pin 2 output decreases in current. As the 8–bit digital number increases. When a value of 0 is sent to the DAC it sinks the minimum current of 0 mA on its pin 4 output, but sinks the maximum current of 1 mA on its pin 2 output. Conversely, the maximum value of 256 into the DAC causes the maximum current sink of 1 mA into pin 4 and the minimum current on pin 2.

Temp/Tone Readback Buffer (diagram 22)

Eight line buffer U612 is connected to the I/O Data bus and used by the EXP to monitor the Temperature Sensor and the Tone generator. U612 is read at either I/O address 3800_{hex} (LS10(L)) or 3900_{hex} (LS11(L)) with BIORC(L). IOD7 reads the output of the comparator to determine the fit of the last tried voltage/temperature. IOD6 reads the Tone generator enable signal. Diagnostics reads the jumpers J710 – J715 for various purposes (two jumpers are used to set the diagnostics default RS–232–C baud rate).

Tone Generator (diagram 22)

The Tone generator is based on a 555 timer, U220, with a special current driver to set its frequency. The timer puts out a square wave whose frequency is inversely proportional to the digital value written to the Temp/Tone DAC. A zero value into the DAC produces the highest tone. The Tone generator is enabled with latch U700 by driving data line D8 high to I/O address 3800_{hex} (LS10(L) active). The trailing edge of BIOWC(L) actually latches D8. A write to 3800_{hex} with D8 low resets the Tone generator and halt the output. A system reset will have the same effect.

The pin 2 output of the Temp/Tone DAC drives a set of tracking current mirrors in U214. This output has a maximum value of 1 mA when a value of 0 is written to the DAC. U214 generates two separate current flows; pin 8 produces a maximum of 1 mA and at pin 7 the maximum is 2 mA. These two currents track each other in a 1:2 ratio while driving into another pair of current mirrors in U112. Initially, the transistor with its collector tied to the Trigger/Threshold inputs of the timer is turned off, so capacitor C230 is charging at a maximum 1 mA rate. When the voltage level on the Trigger/Threshold inputs reaches the threshold it causes the Output to switch high and Discharge to go low. The low Discharge line causes the transistor (6, 7, 8) to turn off making transistor (1, 15, 16) turn on and try to sink twice the current that U214 transistor (8, 9, 10) is supplying. The result is that C230 discharges at that same rate it charged at.

The voltage level on C230 follows a symmetrical triangle pattern. The slopes of the triangle are slower and the output frequency lower for higher DAC Digital values.

The output of U220 drives through a 100 Ω resistor to the front panel connector J72 and out to the front panel speaker. The other side of the speaker is returned to pin 1 of U220 to reduce radiated signals from the speaker coil. CR121 minimizes inductive kickback spikes by clamping them to one diode drop. (Negative voltage spikes can cause the 555 timer to oscillate at a high frequency.) C226 is a large bypass capacitor that reduces noise from the high current needed to drive the square wave output.

Rear Panel Data Buffer (diagram 22)

The Rear Panel Data buffer interfaces the I/O Data bus with the Rear Panel Data bus. The outputs of buffer U100 are enabled by the rear panel bus enable signal, RPSEL, WDDEN(L) and BDEN. WDDEN(L) provides a delay to protect MOS devices on the Rear Panel board. Drive direction is controlled by DT/R(L).

Rear Panel Address Buffer (diagram 22)

Integrated circuit U200 buffers latched address lines LA1 – LA4, and control lines BIORC(L), BIOWC(L), RESET(L) and DACK0(L) to the Rear Panel. These address lines are used to select rear panel devices directly. DACK0(L) is active only when the optional DMA controller is installed on the EXP. When not used, DACK0(L) is pulled to 5 V by a 10k Ω resistor.

GPIB Control (diagram 22)

The GPIB interface operates one way with the optional DMA controller installed and a different way without it. Without the DMA controller, the EXP controls the GPIB just as it controls other I/O devices. The EXP puts the GPIB in a mode where it transfers a byte during each EXP I/O read or write to it. Input requests or "ready for more data" conditions are signaled with GPIB INTR(L), which becomes IR32(L). BIORC(L) is inverted by U820B to generate normal DBINs.

When the DMA controller (option 4D) is installed, the GPIB interface is set up by the EXP, but the actual transfer is controlled by the DMA controller. The DMA controller monitors GPIB service request line GPIB RQ after it is ANDed with DACK0(L). (DACK0(L) removes the GPIB request when the DMA controller acknowledges it to avoid erroneous multiple requests.) DACK0(L) active prevents U820B from inverting the EXP I/O read line BIORC(L), which becomes the GPIB read/write line, DBIN. Each data byte transferred can take two bus cycles (read from memory, write to GPIB). DACK0(L) is buffered by U412B to create DMA0 SEL(L) which is used to enable the Rear Panel Data buffer and to request one wait state from the DMA controller.

The GPIB interface is clocked with a 4 MHz signal from the on-board Clock generator.

Rear Panel Select and Interrupt Buffers (diagram 22)

U210A buffers the latched device select lines LS14(L) – LS17(L), which enable particular devices on the Rear Panel board.

The device interrupt lines from the Rear Panel board are buffered by U210B before being sent to the EXP interrupt controllers. If connector J78 is removed, the Rear Panel interrupt lines will be pulled high by the 10k Ω resistors tied to 5 V. This prevents the EXP from being inadvertently interrupted.

Front Panel Data Buffer (diagram 22)

Integrated circuit U600 buffers the I/O Data bus to the Front Panel board. It is enabled by BDEN and FPSEL (Front Panel Select) NANDed together. U522B produces FPSEL when any Front Panel device is selected. The inverted EXP signal DT/R(L) determines the direction of drive.

Front Panel Control Buffer (diagram 22)

Front Panel Control buffer U800 buffers the device select lines LS5(L), LS7(L), and LS8(L). It also buffers address line LA1, and control lines BIORC(L), BIOWC(L), RESET, and IRDIS(L). IRDIS(L) is a latched control signal that disables the infrared sources on the Front Panel. To disable the infrared detectors, the EXP writes a zero on I/O data bus line IOD0 to address 3400$_{hex}$. IOD0 is clocked into U700B by ORed BIOWC(L) and LS6(L). The IRDIS(L) is set inactive (LEDs ON) by writing a one to the latch or by an active RESET(L). Only during diagnostics will the infrared sources (LEDs) be disabled.

A15 Memory Management Board <27 and <28>

The Memory Management Unit (MMU) board coordinates communications among three oscilloscope subsystems: the Display, the Digitizer, and the Executive Processor (EXP). The MMU Gate Array controls each interface with a different set of handshaking and buffer control lines. In addition to the MMU Gate Array, there are sets of data buffers for each interface, two banks of DRAMS for waveform memory, address decode/select circuits and integrated diagnostic control circuitry. The EXP sets the MMU Gate Array to perform transfers by setting bits in a control register called the Status and Mode register (SMR). It must also load addresses and byte count information into either the Sequential Address generator (SAG) or the Random Address generator (RAG), which reside within the MMU.

MMU Gate Array (diagram 27)

The MMU (Memory Management Unit) Gate Array, U210, controls all data transfers to and from waveform memory. Waveform memory has two identical banks of dynamic memory (Even and Odd DRAMs), which are fully supported for simultaneous accesses. The MMU Gate Array controls high–speed transfers of waveform data and communication messages between waveform memory and the three subsystem interfaces: The Display subsystem, the Digitizer subsystem and the Executive Processor (EXP). Each subsystem interface is coordinated by a set of handshaking lines tailored to the DMA (Direct Memory Access) facilities of the subsystem. The MMU Gate Array also provides refresh addresses, strobes for waveform memory, and arbitration of access requests. Arbitration for all waveform memory accesses is ordered as follows:

1. Refresh cycles
2. Writes to the Digitizer
3. On a rotating basis, (polled every 40 ns) either
   - reads or writes by the Display
   - reads the Digitizer
   - reads or writes by the EXP.

Status and Mode Registration (SMR)

The EXP controls the MMU Gate Array with the Status and Mode register (SMR), which is located at the EXPs I/O address 1860_{hex}. Figure 2–7 shows the layout of the SMR status and control bits. Upon power up, the EXP must initialize the SMR to enable transfers between waveform memory and the Display subsystem (bits 0 and 1) and the Digitizer subsystem (bits 5 and 6). When set, bit 7 allows the EXP to access the normally inaccessible registers, which are associated with the RAG, SAG and refresh counter. Most transfers end with an interrupt bit (bits 2 – 4) being set high. The EXP responds to the interrupt by writing a one to the bit to clear it.

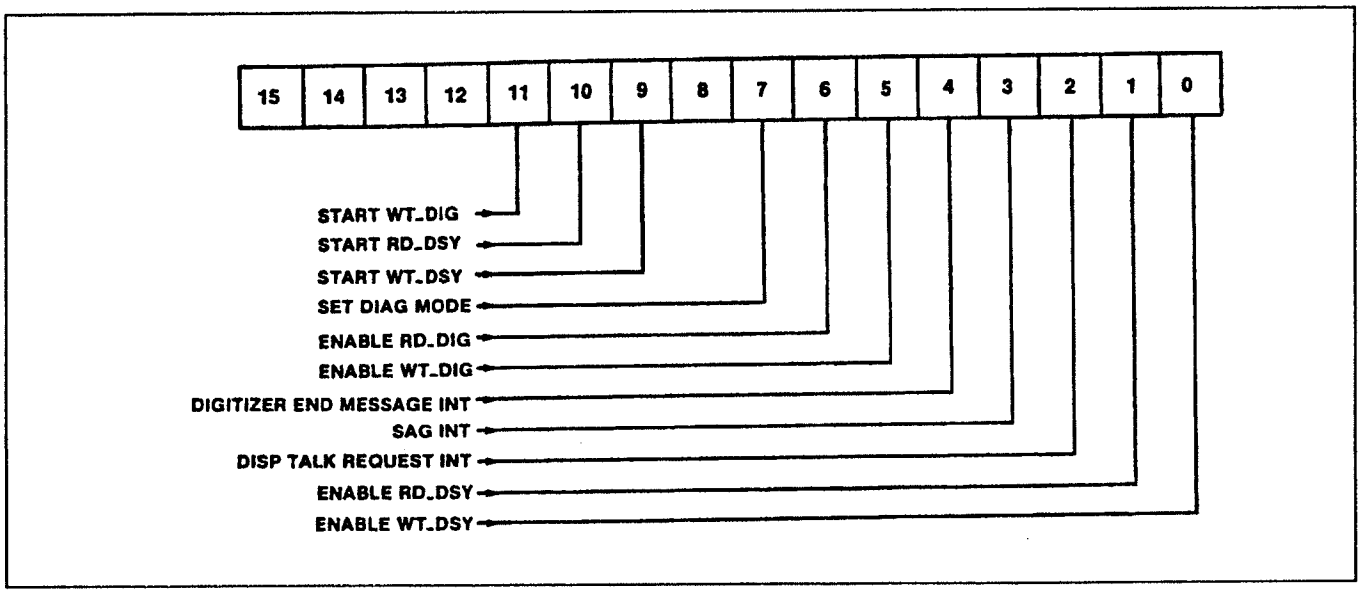


Figure 2–7. MMU Status and Mode Register

RAG and SAG Address Generators

Addresses for waveform memory accesses are generated within the MMU Gate Array by the Random Address generator (RAG) or by the Sequential Address generator (SAG). The RAG is used only to put waveform data from the Digitizer into waveform memory. Fifteen, 9–bit registers, designated RAG0 – RAG14, are the core of the RAG. Each of these, except RAG14, is loaded with a base address for a waveform data record. RAG14 is loaded with an address for communication messages from the Digitizer to the EXP. When the Digitizer is ready to send updated waveform data from an input channel, it sends a 4–bit TAG number to specify a RAG register. Along with the TAG number, an offset address is sent by the Digitizer. The RAG adds this offset to the designated RAG base address to determine the memory location for a word of data. Figure 2–8 shows a block diagram of the Random Address generator.

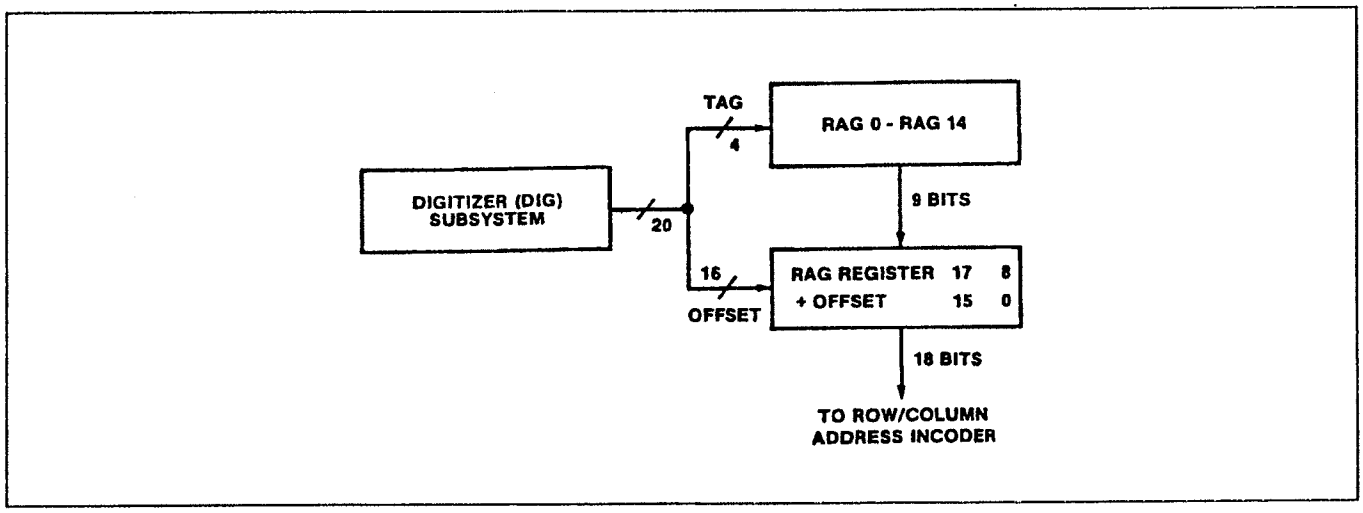


Figure 2–8. Random Address Generator

The Sequential Address generator (SAG) is used by the Display and EXP to move large blocks of data into and out of waveform memory and to send messages (from the EXP) to the Digitizer. The SAG consists of a 14–bit Message Pointer register (MPR) for the base address of a message, a 15–bit Address Counter (AC), and a Message Length register (MLR). To set the SAG to transfer a message into waveform memory, the EXP first loads the MPR with a beginning address for the waveform record. Next, the AC must be initialized to a value from 0 to 65536 to indicate where in the data block to begin an access. Finally, the EXP loads a value in the MLR that equals the message length plus the value put in the AC. The AC is incremented after each word transfer and compared with the MLR. When the MLR and AC contain equal values the transfer is complete and the SAG interrupt bit is set in the Status and Mode register. Figure 2–9 shows a block diagram of the Sequential Address generator.

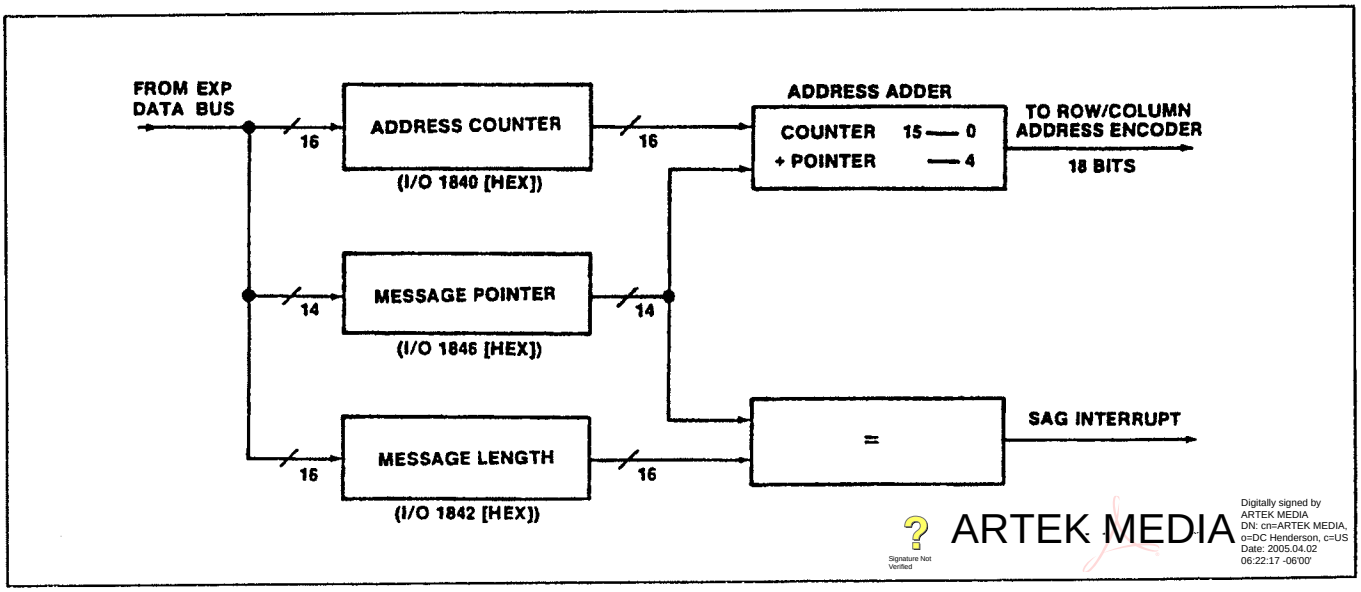


Figure 2–9. Sequential Address Generator

Display Interface

Data is transferred to and from the Display through the Compressor board on a 16–bit data bus (Y0 – Y15). These data transfers use the Sequential Address generator (SAG) to specify the destination or origin addresses in waveform memory. Data is buffered with the bi–directional Display Data buffers for Even and Odd bank waveform memory accesses. U400B and U500A buffer the four dedicated handshaking lines that synchronize the Display Interface. These buffers can be disabled for diagnostic tests. The special handshaking lines for the Display interface are:

- ENDNEW(L) Generated by Display to initiate a read of waveform memory. Its falling edge begins a read cycle.
- DATALATCH(L) Produced by the MMU to clock data from waveform memory into the Display. Data is valid on on the trailing edge.
- DATARDY(H) Generated by the Display when it is ready to send data to waveform memory.
- DATAGATE(L) Produced by the MMU to enable, on its falling edge, the Display data buffers during a transfer from Display to waveform memory. Its rising edge signals the end of the Display's write to memory.

Data transfers begin with the EXP setting up the SAG registers. The EXP must have also used the Executive buses to setup the Compressor. To enable a read of memory by the Display, the EXP must set bit 9, Start WT_DSY, in the Status Mode register (SMR). After bit 9 is set, SENDNEW(L) going active will cause the MMU Gate Array to begin a read cycle from waveform memory. Soon after the read cycle begins, the MMU Gate Array will enable the Even or

Odd Display Data buffers and drive DATALATCH(L) low. The trailing edge of DATALATCH(L) latches the data on the Compressor board or Display controller board. These handshake lines continue to toggle and the SAG continues to increment addresses until the specified number of data words have been transferred. At this time, the SAG will set the SAG interrupt bit in the SMR, which generates an interrupt to the EXP. The EXP must reset the SAG interrupt bit to enable future transfers.

A write by the Display into waveform memory follows a similar procedure. First, the SAG is set and bit 10, Start RD_DSY, of the SMR is set. When data from the Display is ready to be sent, DATARDY(H) is asserted. The MMU Gate Array drives DATAGATE(L) low to enable the data buffers at the Display. The low–to–high transition of DATAGATE(L) signals the end of the Display's write into memory. Figure 2–10 shows timing for Display Interface read and write cycles.

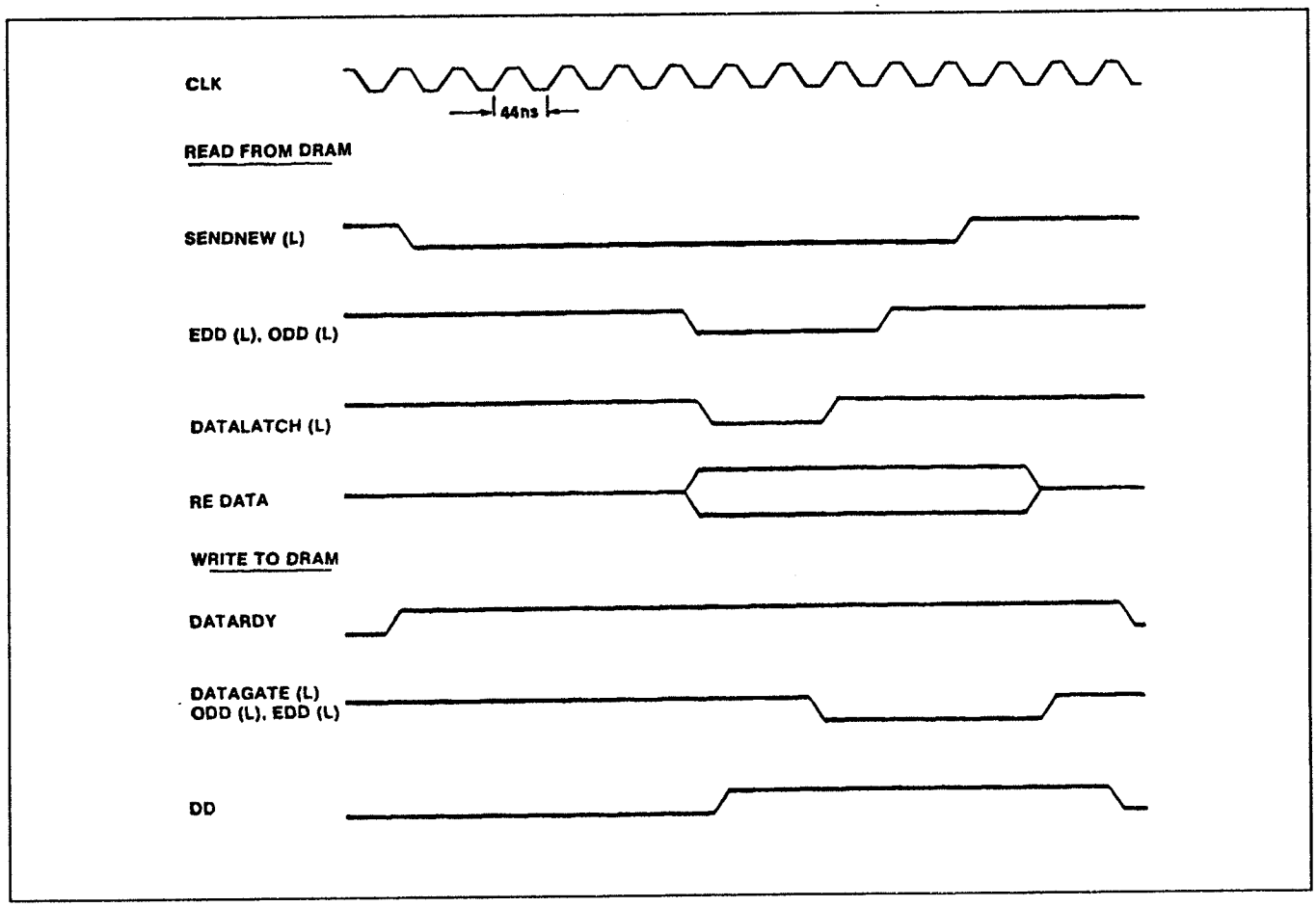


Figure 2–10. Display Interface Timing

Digitizer Interface

Data is transferred to and from the Digitizer on a 20–line multiplexed address/data bus, which is coordinated by dedicated handshaking lines. For transfers to waveform memory, the Digitizer sends a 20–bit address, then a 16–bit data word. During transfers from waveform memory to the Digitizer, only 16–bit data words are sent while the MMU Gate Array's SAG provides the addressing for waveform memory.

For transfers to waveform memory, the MMU Gate Array latches address bus (G0 – G19) information on the trailing edge of AOUT(L). The upper four bits are the TAG field, which specifies a RAG register for a particular waveform record. The lower 16–bits provide an offset address for each data word in the waveform record.

The interface timing for data transfers from the Digitizer to waveform memory requires the data to be latched by the Digitizer Data latches. Data is latched when DOUT(L) goes inactive (on the rising edge). These latches, and the Digitizer Data buffers used for waveform memory reads, are controlled by the MMU Gate Array.

When the Digitizer needs to send acquired waveform data to waveform memory, it asserts DIGREQ(L). (See the timing diagram in Fig. 1–11.) The MMU Gate Array then asserts AOUT(L) to signal the Digitizer to provide the TAG bits and offset address. The address is latched internally by the MMU Gate Array when AOUT(L) goes high. The MMU Gate Array asserts DOUT(L) to request the release of a word of Digitizer data. When DOUT(L) goes high, the data is latched by the Digitizer Data latches. Immediately, a waveform memory cycle begins to store the latched data into the waveform memory location specified by the RAG register and the offset address. The Even Bank or Odd Bank is accessed when the Digitizer address line A0 is 0 or 1, respectively. The appropriate latch output is enabled by ERG(L) or ORG(L) from the MMU Gate Array.

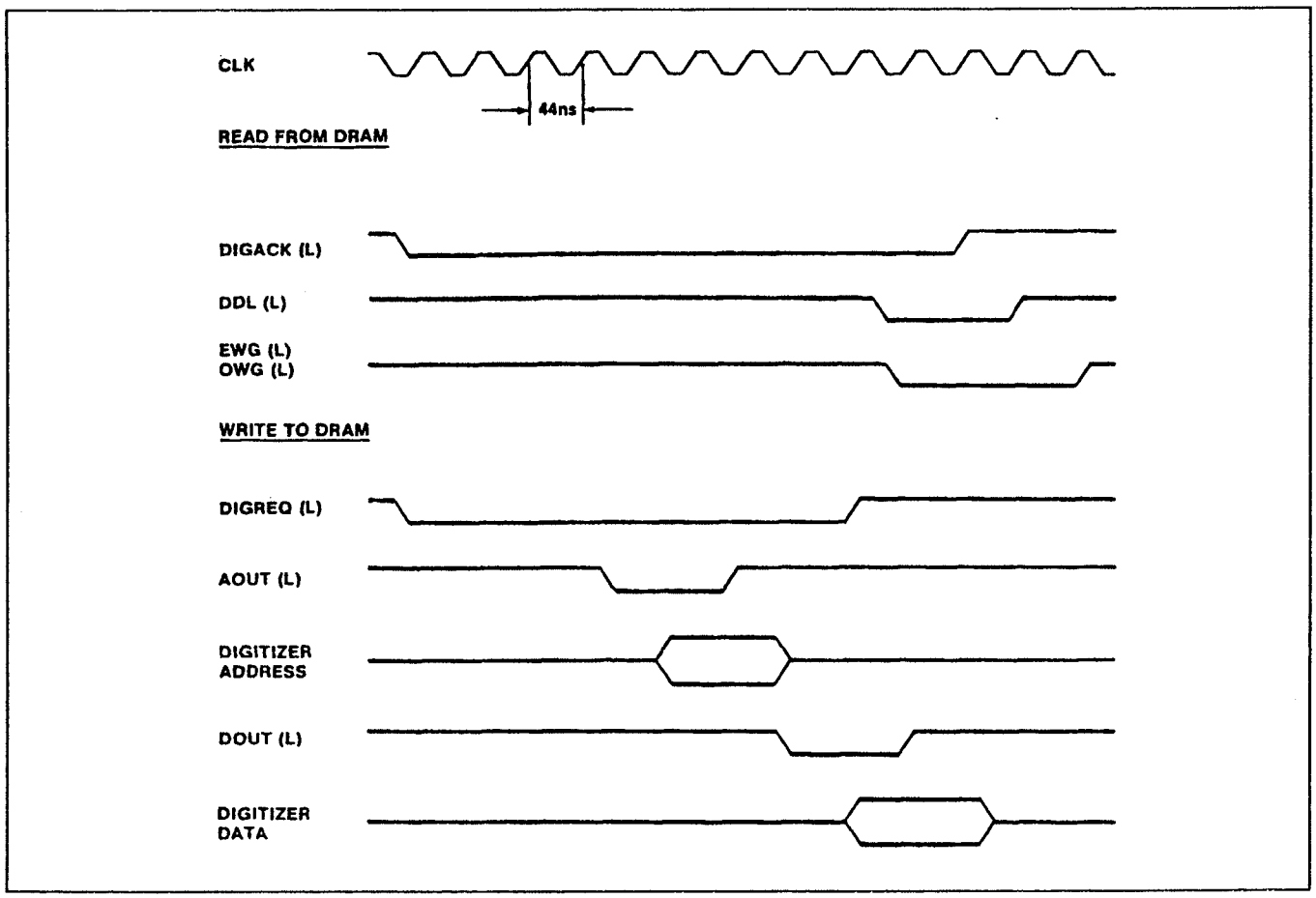


Figure 2–11. Digitizer Transfer Timing

Message transfers from the Digitizer to waveform memory are handled differently. The start of a message transfer is the same as a waveform data transfer except that the TAG field for messages is always 0E*hex* for RAG14. When the last word of the message has been sent, a dummy data transfer is sent to RAG15 (0F*hex*), which sets the Digitizer End Message Interrupt, bit 4, in the SMR. No waveform memory cycle occurs for this last RAG15 transfer. Setting bit 4

of the SMR causes the MMU Gate Array to generate an RQS interrupt to the EXP. The digitizer interface locks when another message is initiated before the SMR is cleared.

Transfers from waveform memory to the Digitizer (see the timing diagram in Fig. 1–11) operate as follows: first, the EXP must set the SAG with the location of the data and set the SMR START WT_DIG bit (bit 11). Then if the DIGACK(L) line is active or goes active, a read from Even or Odd memory will begin. When data appears on the output of the Digitizer Data buffers, the MMU Gate Array asserts DIGLATCH(L). This signals the Digitizer to latch the valid data. When, after a number of consecutive cycles, the entire block of data is transferred, the SAG interrupts the EXP.

U501C and U400A buffer the handshake lines and allow them to be disabled for diagnostics. They are disabled with ISO+ (see U400B and U501A). The upper four address lines, G16 – G19, are buffered by the multiplexer U500. U500 allows all these lines to be driven high or low by the diagnostics when ISOP is low.

Executive Processor Interface

The Executive Processor Interface performs two main functions. One is providing the EXP access to waveform memory for subsystem message passing and manipulation of waveform record data. The other main function is to allow the EXP to coordinate system operation by providing access to the Status Mode register (SMR) and to the diagnostic facilities. The interface consists of address, data, and status/control inputs, EXP interrupt outputs and a data ready output.

Six control lines are decoded by the MMU Gate Array to determine what type of bus cycle the EXP is about to perform. SYSCLK synchronizes the EXP interface, and RESET(L) provides a positive reset on power–up. The input CS(L) is driven active by the Address Decode/Select circuitry whenever the EXP is accessing memory or I/O. The other control and status inputs are decoded for the possible EXP bus cycles in Table 2–6.

Table 2–6
Types of EXP Bus Cycles

MI/O	S1(L)	S0(L)	Type of Bus Cycle
0	0	1	I/O Read
0	1	0	Not Used
1	0	1	Memory Read
1	1	0	Memory Write

BHE(L)	A0	Type of Data Transfer
0	0	Word
0	1	Upper byte valid
1	0	Lower byte valid

If the EXP is to access waveform memory (which appears in the EXPs address space), then SRDY is pulled low by the MMU Gate Array until it is finished with higher priority transfers. SRDY is released near the middle of a memory cycle. The Processor Data buffers are enabled for the proper direction, then their outputs are enabled.

When the EXP is reading waveform memory, the data is latched into the I/O Data latches with Data Latch Enable until the EXP reads it by activating DEN (Data Enable). Figure 2–12 shows the timing for EXP interface cycles.

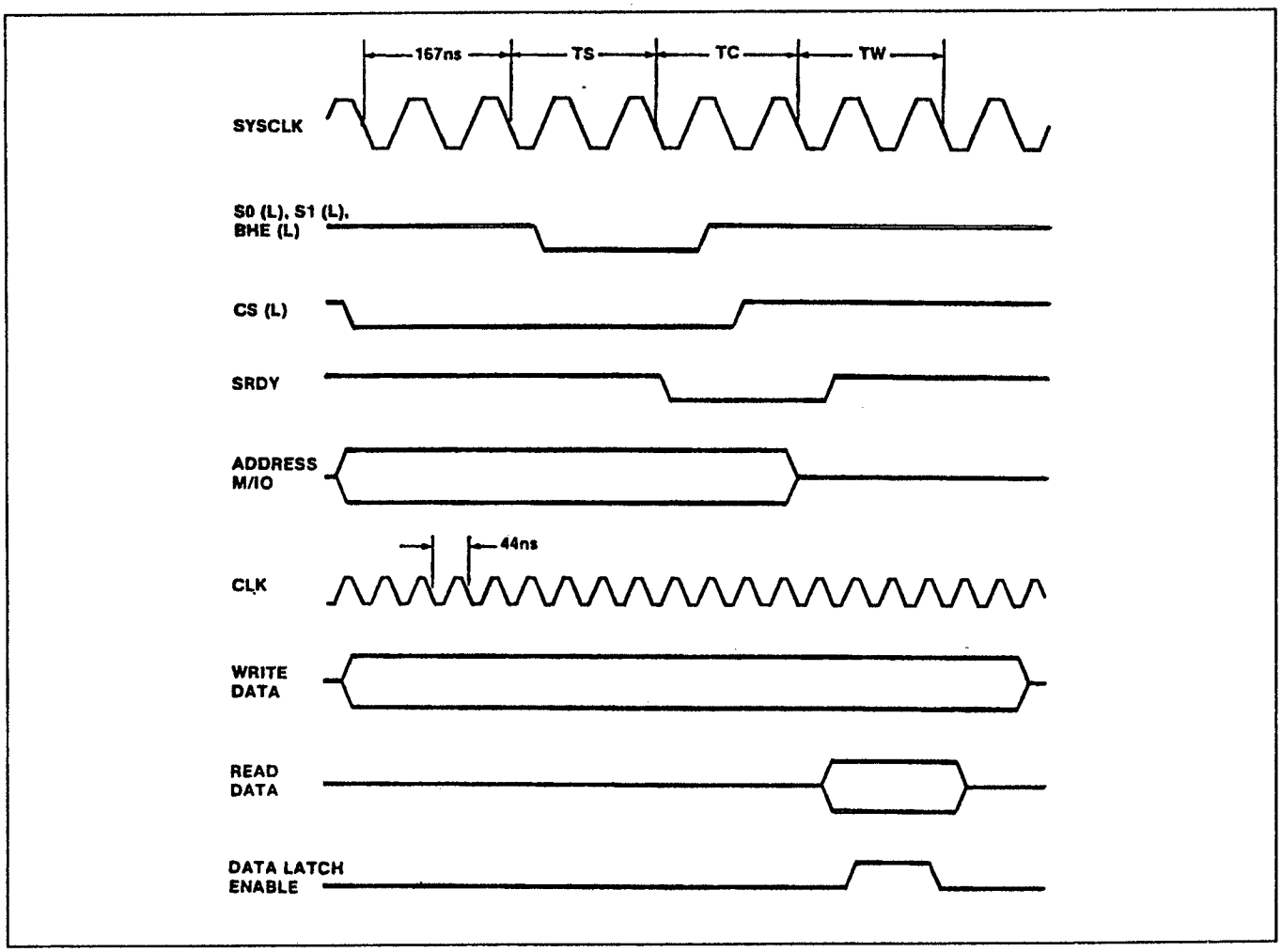


Figure 2–12. EXP Interface Timing

The EXP signal BHE(L) changes only when the next bus cycle is to transfer a different number of data bits (i.e., high byte, low byte or word). SRDY may be held much longer than shown in Figure 2–12.

When the EXP is writing directly to the MMU Gate Array, it does so with an I/O bus cycle to address range 1800 – 1FFF*hex*. The SMR is located at I/O address 186*hex*. After a transfer is complete, the MMU Gate Array asserts one of its three interrupts to the EXP. The interrupts are:

- INT0(L) SAG Interrupt
- INT1(L) Digitizer Interrupt (RAG15)
- INT2(L) Display Interrupt

The EXP responds to these interrupts by clearing the interrupt bit in the SMR and, if needed, setting the start bit for another transfer. These interrupt lines are inverted for the Executive bus by U624A–C.

DRAM Control

The DRAM is supported as two separate memory banks (Even and Odd). The MMU Gate Array generates 8–bit row and column addresses for each bank. Each DRAM Column Address Strobe (CAS) controls one byte (two DRAM chips) of memory. This allows for byte accesses by the EXP. The Row Address Strobe (RAS) for each bank goes through a delay line and an OR gate to guarantee proper RAS(L) to CAS(L) timing for the DRAMS. The falling edge of RAS(L) is delayed 10 ns by the delay lines. The rising edge is not delayed because of U110A,C. This ensures that the RAS will be inactive for the required period. The control lines and address lines go through series 33 Ω resistors to provide impedance matching for the DRAMS to protect them from undershoot.

Memory jumper J201 provides for memory expansion Option 2D, which increases memory from 128 KB to 512 KB. This is important to the MMU Gate Array when it is generating DRAM address.

Even DRAM (diagram 28)

The Even DRAM provides 64 KB of dynamic memory for waveform data and for subsystem communication messages. The Even memory is usually accessed alternately with the Odd DRAM. When A0 from either the Display or the Digitizer is low, or when A1 from the EXP is low, the Even bank of DRAM is selected. Therefore, as these address lines go high and low on consecutive addresses, the Even and Odd bank are accessed alternately. The Even bank is divided into a high bank and a low bank and each of these is further divided into a high–byte bank and a low–byte bank. The MMU Gate Array provides all control signals, refresh signals, and addresses for the Even DRAM. The MMU Gate Array also coordinates data buffers for the three DRAM interfaces (Exp, Display and Digitizer). Of these interfaces, the Executive Processor is the only device that makes byte accesses. Option 2D enlarges waveform memory to 256 KB by replacing the 16K x 4 bit DRAMs with 64K x 4–bit DRAMs.

Each Even DRAM bank, High and Low, is comprised of two Low byte chips and two High byte chips. The MMU Gate Array generates a separate enable line for each byte chip set. The enable lines connect to the Column Address Strobe, CAS(L), inputs at pin 16. The byte chip sets and their enable signals are:

- High bank High byte (U422, U420) C2E(L)
- High bank Low byte (U430, U424) C3E(L)
- Low bank High byte (U322, U320) C0E(L)
- Low bank Low byte (U330, U324) C1E(L)

In addition, the MMU Gate Array supplies the Even DRAM with a common multiplexed 8–bit address bus (J0 – J7), a Row Address Strobe, ER(L), and a write enable, EW(L). The output enables OE(L), pin 1, are tied to ground, but the outputs drive only when a RAS(L) and a CAS(L) have been received. Signal timing is shown in Figure 2–13.

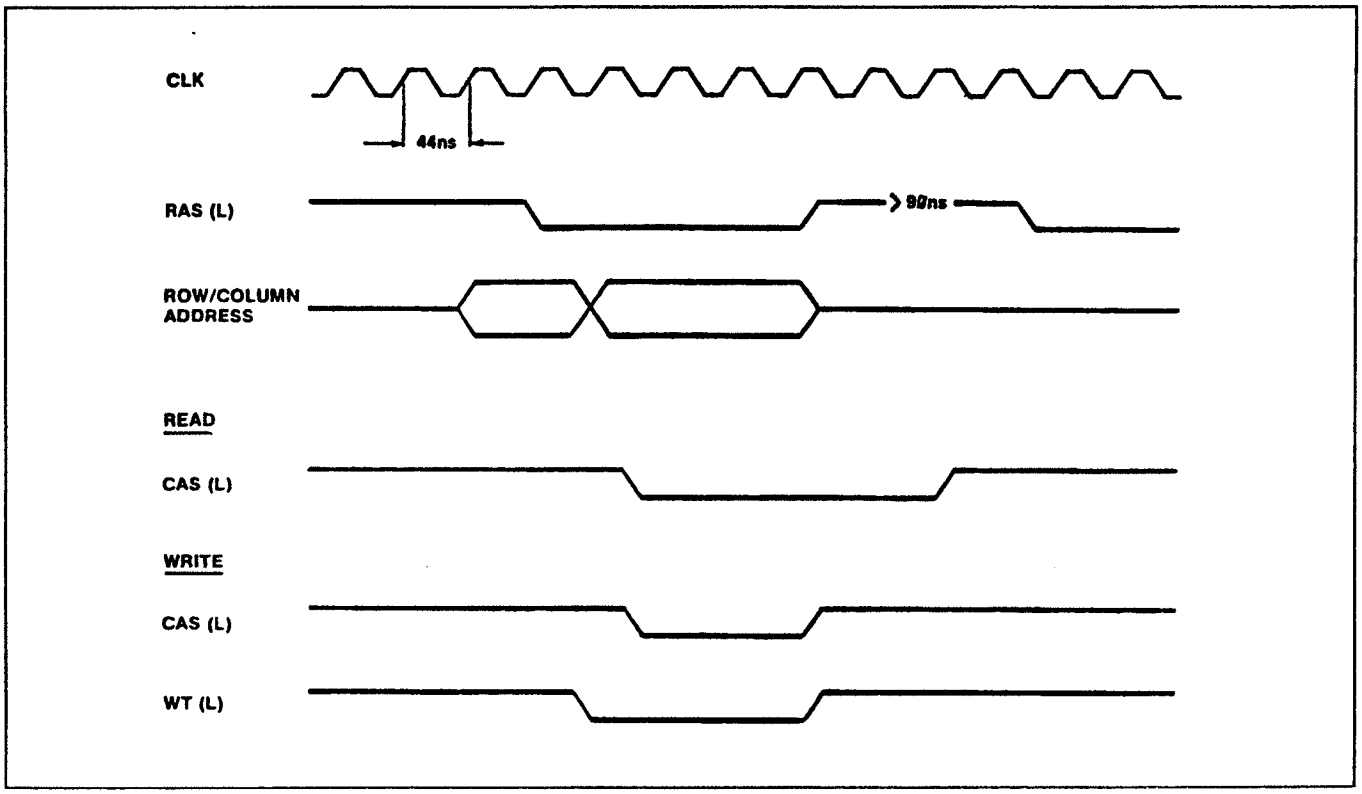


Figure 2–13. DRAM Interface Timing

Data lines E0 – E15 are shared between the High and Low banks. The Low byte banks share data lines E0 – E7. Likewise, the High byte banks share data lines E8 – E15. All data lines have series 33 Ω resistors to eliminate excessive undershoot on high–to–low transitions. Each data line is also connected through a 2.7k Ω resistor to +5 V to pull it high when all drivers are tri–stated.

To address memory locations, the MMU Gate Array sends two consecutive addresses to the RAMs. First the row address is sent on the address lines and latched by the falling edge on the RAS(L) input, pin 5. Then the column address is put on the address bus and latched by a falling edge on the CAS(L) input, pin 16. If the access is to write data into the RAM, the WE(L) input, pin 4, will be driven low soon after the RAS(L) input goes low. RAS, CAS, and WE inputs all return high at about the same time.

Odd DRAM (diagram 28)

The Odd DRAM provides 64 KB of dynamic memory (248 KB with Option 2D) for waveform data and messages. The operation and control of the Odd DRAM is almost identical to the Even DRAM (see the discussion on the Even DRAM for a detailed description). The only differences are the lines that the MMU Gate Array uses to select chips at their RAS(L) inputs:

- High bank High byte (U222, U220) C2O(L)
- High bank Low byte (U230, U224) C3O(L)
- Low bank High byte (U122, U120) C0O(L)
- Low bank Low byte (U130, U124) C1O(L). Also unique to the Odd DRAM are the Row Address Strobe input OR(L), and the write enable OW(L).

Processor Data Buffers (diagram 28)

The Processor Data buffers interface between waveform memory and the I/O Data buffers and latches, which provide data buffering for the Executive bus. The Processor Data buffers are eight–line, bi–directional tri–state drivers (the third state is high impedance). There are two sets of buffers; one set, U802 and U702, for the Even DRAM and one set, U812 and U712, for the Odd DRAM.

The buffers are enabled on pin 19 by MMU Gate Array line EEXD(L) for the Even bank and OEXD(L) for the Odd bank. Drive direction for all four buffers is controlled by MMU Gate Array line EXD on the EN inputs, pin 1. When the chips are not enabled, their outputs are set in the high impedance state.

The "B" side of the buffers is connected to the I/O Data buffers and latches. MMU Gate Array-controlled timing for transfers is explained under Figure 2-12 in the discussion of the MMU Gate Array.

Display Data Buffers (diagram 28)

The Display Data buffers provide an interface between the Even and Odd DRAMs and the Display system. These Display Data buffers are eight line bi–directional tri–state drivers (the third state is high impedance). There are two sets of buffers; one set, U610 and U602, for the Even DRAM and one set, U700 and U600, for the Odd DRAM.

The buffers are enabled on pin 19 by MMU Gate Array line EDD(L) for the Even bank and ODD(L) for the Odd bank. Drive direction for all four buffers is controlled by MMU Gate Array line DD on the EN inputs, pin 1. When the chips are not enabled, their outputs are tri–stated.

The "A" side of the buffers is connected to the Compressor Port J79. Figure 2–10, under the MMU Gate Array discussion, describes the transfer timing.

Digitizer Data Buffers (diagram 28)

The Digitizer Data buffers drive data from the DRAM memory to the Digitizer through Time Base Port J83. There are two sets of buffers; one set, U514 and U614, for the Even bank and one set, U410 and U510, for the Odd bank. The chip enable CE input, pin 19, for the even buffer set connects to MMU Gate Array signal EWG(L) and the odd buffer set connects to MMU Gate Array signal OWG(L). Figure 2–11, under the MMU Gate Array discussion, describes the transfer timing.

Digitizer Data Latches (diagram 28)

The Digitizer Data latches latch data from the Digitizer's multiplexed address/data bus and drive it to the Even and Odd DRAMs. The set of latches, U412 and U512, for the Even bank are enabled by MMU Gate Array line ERG(L) on pin 1. Likewise, the set of latches, U410 and U510, for the Odd bank are enabled by MMU Gate Array line ORG(L). Both sets of latches use the same MMU Gate Array latch enable signal DDL. The "D" inputs are connected to the Time Base Port J83.

Clock Generator (diagram 27)

The Clock generator is a clocking and a diagnostic circuit that allows the MMU Gate Array clock to be stopped and single stepped. Output of the 23 MHz oscillator Y103 is ANDed with the output of D flip–flop U300A and B. The

clock signal is inverted by U434C and used to clock in the D input, pin 12, from diagnostic control latch U530. When the D input is low, the Q output, pin 9, goes low. One clock cycle later, the Q output, pin 5, of U300A goes low. This low applied to the pin 10 input of U200C disables the clock to the MMU. While the main clock is disabled, the MMU Gate Array can be single-stepped with the pin 12 input of U110D. This single-step signal comes from diagnostic control latch U530.

Address Decode/Select (diagram 27)

The Address Decode/Select circuitry decodes addresses, control and timing signals from the Main Processor board (via the Executive bus) to produce control signals for on-board devices. These on-board control signals either coordinate transfers between the MMU Gate Array and the Executive Processor (EXP) or enable diagnostic hardware.

Digital comparator U232 has a single output that is low when the dynamic Q input matches the static P input. The comparator outputs and some Executive Address and Control lines are decoded by PAL (Programmable Array of Logic) U332 to produce four control signals:

- CS(L) Active during either an I/O or waveform memory access by the EXP.
- LOE(L) Active during memory reads by the EXP.
- DIAG(L) Active when the EXP reads from or writes to diagnostic registers external to the MMU Gate Array at I/O addresses 1900$_{hex}$ and 1980$_{hex}$.
- BUFEN(L) Active during memory and I/O operations to waveform memory or the MMU Gate Array I/O space (1800$_{hex}$–1FFF$_{hex}$).

A group of random logic gates is used to latch and further qualify the PAL control signal outputs. OR gates U432C and U432D distinguish between diagnostic I/O address 1980$_{hex}$ and 1900$_{hex}$, respectively. U334 latches the control outputs from U332, U432C and U432D when ALE (from the Main Processor board) changes from low to high. U434D enables the I/O Data buffers when DEN is high during EXP operations. U434C enables the output drivers of the I/O Data latches when the EXP is reading from waveform memory and DEN is high. U720A clocks the diagnostic control latch U530 when I/O address 1900$_{hex}$ is written to by the EXP. U432A enables diagnostic buffers U112 and U501B when the EXP reads from I/O address 1980$_{hex}$. U432B enables diagnostic control latch U524 when EXP writes to address 1980$_{hex}$.

Ready Synchronizer (diagram 27)

The Ready Synchronizer circuitry synchronizes the low-to-high transitions of the MMU Gate Array SRDY output with SYSCLK to satisfy a timing specification of the Main Processor board. SRDY will go low, then high during each Executive Processor (EXP) access of waveform memory.

When SRDY is driven low by the MMU Gate Array and the diagnostic line ISO + is inactive (low), OR gate U110B will react by driving its pin 6 output low. This low will cause the output of U612A, pin 12, to go low requesting wait states from the EXP. U814B, C and D and D flip-flops U714A and B latch the SRDY low condition.

When the MMU is ready to handle a data transfer, it sets SRDY high, which causes the output of U110B to drive the pin 1 input of U612A high. The Q output, pin 5, of flip-flop U714A will be switching at half the SYSCLK frequency because the output of U814D is high and the toggling feedback to the pin 5 input of gate U814B. When the C input, pin 11, of U714B goes high, it will latch the now high SRDY line to the Q output, pin 9. Pin 9 going high results in the output, pin 12, of U612A going high, which removes the wait request to the EXP. U612B allows any of the three control lines, RESET(L), ALE, and ISOP, to initialize the Ready Synchronizer and set its SRDY output high.

I/O Data Buffers and Latches (diagram 27)

The I/O Data buffers and latches interface the I/O data bus with the Executive Data bus. I/O reads and writes and writes to waveform memory by the Executive Processor (EXP) employ the bi–directional I/O Data buffers. The I/O Data buffers, U724 and U824, drive or receive data according to the level of the DT/R signal, which originates on the Main Processor board. Chip enable for the buffers is controlled by the Address Decode/Select circuit.

Data latches U722 and U822 are used only when the Executive Processor is reading data from waveform memory (Even or Odd bank). The MMU Gate Array and DRAM retrieve data before the EXP is ready for it so the data must be latched. Data latching is controlled by a line from the MMU Gate Array. The outputs drive data on the Executive bus when the Address Decode/Select circuit generates the output enable signal. Figure 2–12, under the MMU Gate Array discussion, shows transfer timing.

Diagnostics (diagram 27)

Diagnostic support consists of Even DRAM address feedback to the EXP, single–step capability for the MMU Gate Array, and isolation so that handshake lines won't toggle during testing. Several latches and buffers are provided to effect diagnostic control. U530 is at I/O address 1900$_{hex}$ for the EXP and must be initialized on power–up to enable normal operation. When used for diagnostics, U530 provides isolation control for the Display and Digitizer handshake lines and the upper four Digitizer address lines (TAG field). It also allows the clock to be stopped and the MMU Gate Array to be single–stepped through Clock generator control.

The other diagnostic latch and buffer is located at I/O address 1980$_{hex}$. It consists of separate write and read components that carry different information. The write port is U524 which is enabled by the ISOP line and clocked by a line from the Address Decode/Select. U524 allows the handshake lines and TAG field lines to the MMU Gate Array to be toggled for testing. The read port consists of buffers U112 and U501B. When U112 is enabled, the EXP can read the current address on the Even address lines. U501B provides a look at the actual handshake lines from the Digitizer and Display and the setting of memory jumper J201.

A16 Waveform Compressor/Adder <29>, <30> and <31>

The oscilloscope waveform display is made of 512 vertical lines. No matter what type of waveform is being displayed, it will always include 512 vertical lines. The length of the individual vertical lines depends on the change in voltage at the time represented by the horizontal location of the particular vertical line. If no input signal is present no vertical displacement of the trace is needed, so the vertical line will necessarily be longer. The length of the line will represent the FV that occurred in the time that this part of the leading edge was sampled.

The Waveform Compressor/Adder (WCA) circuit is one of the circuits the 11401/11402 oscilloscope uses to produce its display.

The waveform compressor part of this circuit provides 512 pairs of data points to the display. The MMU always provides 512 sets of data points to the compressor. Hence, the name "compressor," because the WCA reduces its sets of input data points to pairs of data points.

The pairs of points transferred to the display are the minimum and the maximum of the input set.

The adder part of the circuit provides vertical display position control for the user, by either adding a digital offset to, or subtracting the same digital offset from, the data points.

A normal, nonvectored display may have gaps, or holes, between adjacent points. if the viewer wants a continuous display, without holes, he can select the vectored display.

"Vectoring" takes place by comparing each new data point in a set with a previous pair of displayed values to see if the new point is greater than, equal to, or less than those values.

Vectoring is produced by:

1. reversing the minimum and maximum values of a pair of points, before calculating the next pair of points
2. comparing the previous group's minimum with the next group's maximum
3. comparing the previous group's maximum with the next group's minimum.

If the previous minimum is greater than the next group's maximum, the next maximum will be considered the same as the previous minimum. This will take the ends of the two adjacent vertical lines end and start at the same point.

Similarly, if the previous maximum is less than the next group's minimum, the next minimum will be considered the same as the previous maximum. As described previously, the ends of the two adjacent vertical lines will then end and start at the same point.

The WCA can transfer data representing displays of any size. The executive processor (EXP) synchronizes the number of points the WCA receives and the number of minimum–maximum (min–max) point–pairs it transmits. To do this, the WCA must receive some number of input points equal to the desired number of output point–pairs times the compression factor (CF). That is, the ratio of input points to the CF must be an integral value, which equals the number of point–pairs divided by 512.

Compressor Modes

The WCA has two modes of operation:

- Transparent (TX). In TX mode, the WCA transfers data directly from the Waveform RAM) to the Display (DSY) without altering it. The data goes through the Compressor and the Adder, and reaches the DSY exactly as it started. Application: messages, i.e., text that must not be changed.
- Compressed (N). In N mode, the WCA selects minimum and maximum values from each group of input data points. The 512 groups of data points each contain N data points, where N = the compression factor.

For example, if the WCA receives 5120 data points, the CF will be 10. This means that there will be 512 input groups of 10 data words each. The compressor will search each of the input data groups, and will produce one min–max pair for each of the 512 groups. The 512 min–max pairs are adjusted by the offset specified in the adder's data register, then sent to the DSY to produce a complete display. (The offset from the adder provides position control.)

Compression Factor (CF)

The CF is the number of data points from which each min–max pair is extracted. As shown in Table 2–7, the CF can be any listed number of 1 to 255. The maximum number of data points that can be compressed and displayed is 130,560 (where [CF][512] = maximum number of data points; and 255 x 512 = 130,560).

Table 2–7
Compression Factor

Input Points	Min-Max Points	CF
512	512	1
1024	512	2
2048	512	4
4,096	512	8
5,120	512	10
8,192	512	16
10,240	512	20
16,384	512	32
32,768	512	64
51,200	512	100
65,536	512	128
130,560	512	256

Compressing Data Points

"Compression" is the process of selecting the minimum and maximum values from a group of data points. It takes place by comparing all data points in the group and keeping only the minimum and maximum values.

Criteria for Replacing the Minimum

Input data is latched into Min register if:

1. Data is not NULLP and Min is OVER, or
2 Min is NULLP, or
3. Data is UNDER, or
4. Min is not UNDER, and Data is not NULLP and not OVER, and Min > Data, or
5. Data is last point in group to be compressed (EOC=true), and all NULLPs were found in group (ALLNULL=true), or
6. M/MUND=true.

Criteria for Replacing the Maximum

Input data is latched into Max register if:

1. Data is not NULLP and Max is UNDER, or
2. Max is NULLP, or
3. Data is OVER, or
4. Min is not OVER, and Data is not NULLP and not UNDER, and Max < Data, or
5. Data is last point in group to be compressed (EOC=true), and all NULLPs were found in group (ALLNULL=true), or
6. M/MUND=true.

At the end of the compression sequence:

1. The Min and Max values are transferred into the Adder latches
2. The role of the Min and Max registers is reversed, i.e.,
   a. the Min register will be the Max register when the next group is compressed, and
   b. the Max register will be the Min register when the next group is compressed,
3. If so specified, set the M/MUND bit true (for nonvectored).

Adder

The Adder provides an offset to vertically positioned the display on the CRT. Available offsets range from P32,768 to +32,767, as provided in a two's complement 16–bit word. The output of the Adder circuit will be one of the following four cases:

- NULLP if data is NULLP,
- UNDER if data is UNDER, or if data is negative, offset is negative, and result is positive,
- OVER if data is OVER, or if dat is positive, offset is positive, and result has carry, or
- the sum of the input and the offset.

In Compress Mode, the order within a pair is always first Min, then Max.

Address Decode/Select (U524, U430, and U424, diagram 29)

The Address Decode/Select (ADS) monitors data on the system address bus. During I/O operations, the ADS will respond to a specific address and produce latch enable (LE) signals.

Decoder U524 performs the first stage of decoding; it is wired to produce a low output when its input is 000x 0000 0xxx xxxx on lines A15 through A0, respectively.

When enabled, encoder U430 produces a low on one of its Y0–Y7 output lines, as indicated by the state of the A1, A2, and A3 input lines. Encoder U430 is enabled when:

- Decoder U524 produces a low on its output
- A12 line is high
- M/IO(L) line is low.

Latch U424 will hold the output of U430. The ALE signal activates U424.

When ALE occurs with an out–of–range address or for a memory reference, none of the eight selections is valid.

Address bit A4 is not decoded, which results in two I/O address for each selection. For example, address select 0 will be indicated for addresses 1000_{hex} and 1010_{hex}. Table 2–8 shows the address to register mapping.

Table 2–8
I/O Address Mapping

Location	Operation	Register
1000_{hex}	Write	Mode Word
1000_{hex}	Read	Status Word
1002_{hex}	Write	WCARST*
1002_{hex}	Read	ADDATA
1004_{hex}	Write	Compression Factor
1006_{hex}	Write	Offset
1008_{hex}	Write	ROMAX*

ROMAX** = no register storage

Mode Select Register (U724, diagram 29)

The WCA Mode register has three separate mode control bits, which allow eight modes. Each of the eight modes is theoretically possible, though not all are particularly useful. The three mode selections are Compress/Transparent, Vectored/ Nonvectored, and Normal/Test.

In the Compress/Transparent mode, the C/T(L) bit is used in the adder state machine and the compressor flag–decoding circuit. In compress mode the adder sends the X and Y register values, thus producing two output values for each input group. Special–case flags decoded in the compression cycle are not altered. In the transparent mode, only the X or Y value is sent; that is, just one output for each input group. The special–case flags are forced to be invalid in transparent mode.

"Vectoring" overlaps consecutive group of data. When vectoring is on, consecutive data groups are "compressed" with consideration of the previous group's min and max values. With vectoring off, the compressor's X and Y values are marked undefined before operating on a new group. The Vectored/Nonvectored bit is sent to the M/MUND decode circuit.

The Normal/Test mode specifies whether the adder output is to be transferred to the Display or presented to the Executive Processor (EXP). In normal mode, the adder state machine waits for the Display's SENDNEW(L) signal before sending the data. In test mode, the adder state machine waits for the decoded DREAD(L) (data read) signal.

Compression Factor Counter (diagram 29)

The compression factor (CF) is the number of data points from which the compressor will select a min–max pair for the adder. The CF is an 8–bit number, valid in the 1 to 255 range. A value of zero, which is specified as undefined in the HW/SW interface for WCA programming, will be equivalent to the value 256. The value stored in the CF register is loaded into an 8–bit counter (U814, U820) by the RSTCYC(L) signal. This counter consists of two 4–bit devices (74ALS191) connected as an synchronous two–stage counter with ripple carry–borrow. The RCO(L) output of the lower 4–bit counter (U814) is the input to the upper 4–bit counter (U820). Both devices are connected as down counters. The MSENDNEW(L) clocks both counters, which provide a Max/Min output when the counter reaches zero. The two Max/Min signals are ANDed to produce the EOC signal for the compressor state machine.

Reset Control (U432D, U130D, diagram 29)

This circuit decodes a Reset signal to set the compressor and adder state machines to known states, and to set, or clear, the control latches. The decoding simply selects the system Reset pulse or an output from the EXP to address 1008_{hex}.

Clock Generator (Y122, diagram 29)

A TTL–compatible crystal oscillator, Y122, produces the 20 MHz Clock signal for the compressor and adder state machines.

Compression State Machine (see the Compressor State Machine diagram)

The following are inputs to the Compressor State nmachine:

(C)DATAIN(L) (Compressor DATA IN signal, in Compressor Control on diagram 29) – When the MMU latches data into the input register, the (C)DATAIN(L) line is set to its active state (low). The MDATALATCH(L) signal from the MMU clocks D flip–flop (FF) U120A, whose Q output goes low (its D input is grounded). U120A's Q(L) output produces the MSENDNEW(L) signal, which indicates that the compressor cannot receive data while (C) DATAIN(L) is low.

WCARST (Waveform Compressor/Adder ReSeT, in Reset Control, on diagram 29) – The WCARST signal (on TP42) notifies the compressor state machine that a reset is active. In all states, the state machine will change to an initial, known, state when WCARST is asserted. The state machine will stay in that state as long as WCARST is asserted.

EOC (End Of Cycle, from TP47 in Compression Factor Counter on diagram 29) – When the Compression Factor Counter has finished compressing a group of data points, it generates the EOC signal to notify the X and Y Min/Max Latch Decoders (on diagram 2) of that completion. When the most recent input data has been processed, the state machine can follow one of two paths depending on the state of EOC. If EOC is high, the min/max is transferred to the adder input registers. (EOC causes this via U520, U320, U210A, U812, and U510.) If EOC is low, the compressor will assume its idle state, and await the next input from the MMU.

ModeWR (Mode word WRitten, U432A in Mode Select Latch on diagram 29) – Whenever a compressor mode word is written into the Mode register, the compressor will pass through its Reset state, asserting the MMUAVAIL and RSTCYC outputs. Writing a new word into the Mode register indicates the beginning of a new transmission through the WCA.

This input is tested only during the compressor's idle state, and it will remain active only for the duration of the EXPs IOWC(L) pulse. Therefore it is specified, in the HW/SW Interface, that a mode word be written only after determining that the WCA is in its idle state. This restriction requires only that the WCA not be restarted while it is transferring previous data.

ADDAV(L) (ADDer AVailable, U224 in Adder Control on diagram 31) – When asserted, ADDAV(L) signifies that the adder state machine is available to accept new data from the compressor. When the compressor has finished compressing a group of data points, it will test its ADDALV(L) input; if neither ADDAV(L) nor WCARST is asserted the compressor will stay in the "wait add" state until one of them is asserted.

The variables of the compressor state machine are used as output signals. Each state is represented in a unique combination of state variable. The outputs are:

MMUAVAIL(L) (MMU AVAILable, U222 pin 15 in Compressor Control, diagram 29) – The MMUAVAIL(L) signal has two functions:

- sets the MSENDNEW(L) signal into the state that indicates the compressor is ready for the next data element from the MMU, and
- clears the (C)DATAIN(L) input signal.

To speed the executions of the MMU (for fetching the next word) and the compressor (for processing the current input), MMUAVAIL(L) is asserted before the compressor has finished with its current data group. The MMU is specified to take at least 300 ns before it will assert the MDATALATCH(L) control. The compressor state machine will resume its idle state within this 300 ns; therefore, no overrun is possible.

LCTL (Latch ConTroL, U222 in Compressor Control on diagram 29) – Some of the compressor's internal control signals are decoded asynchronously with respect to the state machine cycling. The LCTL pulse synchronizes the ALLNULL and M/MUND signals by latching their values within a compressor state. This stabilizes the control information.

RSTCYC(L) (ReSeT CYCle, U222 in Compressor Control on diagram 29) – The Reset Cycle pulse is produced when the compressor has finished compressing a group of data points. This signal loads the compression factor counter and serves as an input to the M/MUND decoding circuit.

M/MLATCH(L) (Min/Max LATCH, U222 in Compressor Control on diagram 29) – The compressor comparators decode a latch control, and latch the decoded level into a D flip–flop. If the comparator outputs are asserted true (low) the M/MLATCH(L) signal, which is produced after the comparisons have stabilized (being combinationally produced), will cause the Cx and/or Cy LatchEnable signal to change from low to high.

M/MCLR(L) (Min/Max CLeaR, U222 in Compressor Control on diagram 29) – The M/MCLR(L) signal prepares the Cx and Cy registers to receive new data by setting their Latch Enable inputs low.

ADDLATCH(L) (ADDer LATCH, from U222 in Compressor Control, diagram 29) – When the compressor has data for the adder and the adder is available, the compressor sends ADDLATCH(L) to latch the data from the Cx and Cy registers to the Ax and Ay registers.

IDLE(L) (from U222 in Compressor Control on diagram 29) – The IDLE(L) signal indicates that the compressor is in its idle state. IDLE(L) is connected to the compressor status word registers. Indicating that the compressor is idle ensures that the compressor has finished processing the information it received from the MMU.

NOTE

When the compressor is in idle state, the (C)DATAIN input will occur asynchronously and will cause the state machine to change states. The state, to which the machine goes after idle, should be only a single bit different from idle (or an indeterminate state may exist). Therefore, the idle bit will be asserted one state after the compressor recognizes input from the MMU. This means that the idle flag will be inaccurate for one state period. Such a discrepancy will produce no erroneous operation if the idle bit is tested only after the current transmission to the Display has resulted in an interrupt from the GPAG. This restriction is acceptable because the compressor is known not to be idle between the time an MMU transfer is initiated and completed.

MMU Input Register (U500, U800, diagram 30)

Data from the MMU is latched into a 16–bit register, which consists of U500 and U800.

The latch control signal, MDATALATCH(L), is derived directly from the MMU DATALATCH(L) signal; it is asynchronous with the WCA clock.

NOTE

U120 inverts data bit 15 at the output of U500; it stays inverted until it is restored before the adder presents it to the display.

Special Case Flag Decoding

The WCA recognizes three special cases of data: Null, Over, and Under. Outside the WCA, three 16–bit words represent these three cases; these three cases are encoded by three mutually exclusive bits within the WCA. These cases directly affect the min/max latch decoding and the selection of the output value from the adder circuit. When a data value is latched into Cx, Cy, or both, the special flags are also latched. Therefore the Cx, Cy, Ax, and Ay registers are effectively 19 bits wide (16 bits of data and 3 bits of flags). This recognition circuit are combinational; it settles to the correct state after the MMU input registers is changed. The flags are decoded only in Compress Mode, as previously explained. Four 8–bit identity comparators perform the decoding.

Table 2–9
Case Flag Encoding

	Byte	
Comparator	Upper	Lower
#1 recognizes	0111 1111	—— ——
#2 recognizes	—— ——	1111 1111
#3 recognizes	1000 0000	—— ——
#4 recognizes	—— ——	0000 000x

The combination of these four comparisons and data bit 0 are used to decode the three special–case flags.

NOTE

In the compressor circuit, the flags are positive true; in the adder circuit they are negative true.

X Comparator and Y Comparator Output Latches (U510, U812, U310, U612, diagram 30)

These two registers hold the intermediate values of a compress group's minimum and maximum values. The comparator X register can serve as either the minimum or maximum register, as determined by the M/M MUXSEL signal to U320. The comparator Y register will always be the opposite of X; if X is min, Y will be max, and vice versa. The outputs of these registers are connected directly to the input of the adder registers.

X and Y Comparators (U410, U710; U302, U610, diagram 30)

These two 8–bit comparators permit the system to compare the compressors' present contents with the current input from the MMU. Each comparator produces four signals, which indicate that (P > Q) (L) and (P = Q)(L) in both the lower and upper bytes. The MMU input register is connected to the P input to the X and Y Comparators, and the X and Y Comparator Output Latches are connected to their Q inputs. The four output signals encode the relation between the values being compared; they are decoded in the min/max latch decoder (U420, U520 on diagram 30).

Table 2–10
X and Y Comparator Output Encoding

Upper (P>Q)(L)	Lower (P=Q(L)	(P>Q)(L)	(P=Q)(L)	Relation
0	0	0	0	Undefined Upper and Lower Outputs
0	0	0	1	Udefined Upper Outputs
0	0	1	0	Udefined Upper Outputs
0	0	1	1	Udefined Upper Outputs
0	1	0	0	Undefined Lower Outputs
0	1	0	1	P>Q
0	1	1	0	P>Q
0	1	1	1	P>Q
1	0	0	0	Undefined Lower Outputs
1	0	0	1	.P>Q
1	0	1	0	P+Q
1	0	1	1	P<Q
1	1	0	0	Undefined Lower Outputs
1	1	0	1	P<Q
1	1	1	0	P<Q
1	1	1	1	P<Q

X and Y Minimum/Maximum Latch Decoders (U420, U520, U320, U210A and B, diagram 30)

A combination of inputs (two sets) indicates whether the current input should be latched as the new minimum or maximum of the current group. The two sets (each set includes comparator, register, and latch) are virtually identical – they differ only in that at one time X will be min and Y will be max, or vice versa. The latch decoding takes place in a programmable array logic (PAL) IC, one for X (U520), and one for Y (U420). These PALs produce two outputs – one for minimum and one for maximum latching. The two outputs are active low, combinationally derived, and are synchronized to the compressor state machine with a D flip–flop (U210A and B). The requirements for latching are as follows:

Adder Input X and Y Latches (U712, U412, U512, U312, diagram 31)

The Input X and Y Latches hold the input values for the adder. The inputs of these latches are connected directly to the output of the compressor's Comparator Output Latches (diagram 30).

Adder Input Multiplexer (U202, U314, U414, U614, U714, diagram 31)

This multiplexer combines the output of the two Input X and Y Latches into a world–serial data stream for presentation to the Adder. The RSEL signal from the Compressor Control (diagram 29) controls the multiplexer's operation, thus designating the order in which the data from the input data latch reaches the adder. This selection process is necessary to ensure that the data always reaches the adder in minimum, then maximum order.

Adder Offset Register (U624, U722, diagram 29)

The input data to the adder can have a 16–bit offset included. This offset must be stored in twos complement to allow for both addition and subtraction. The offset is entered as the other operand to the actual adder chips.

Output Word Control (U102A and F, U112A and B, U114A and B, diagram 31)

The output of the adder is connected to an output word selector. If the input data was one of three special cases (Null, Over, or Sum) in the adder input registers, that special case will be transmitted – not the result of the addition. The output selector provides the function that selects either the sum or one of the three special cases. The following equations determine the values of S1 and S2, which are like two selector lines for a four–to–one multiplexer.

Table 2–11
Output Word Selection

S1	S2	Select	
1	1	NULL	If AX(AY) is NULL
1	0	OVER	If AX(AY) is OVER, or AX(AY) bit 15 = 1, offset = 0 and carry exists
0	1	UNDER	If AX(AY) is UNDER, or AX(AY) bit 15 = 0, offset bit 155 = 1, and sum bit 15 = 1.
0	0	Sum	Any other condition.

If the added sum forces a positive value over the OVERrange value, the output data will be set to OVER. Similarly, if the added sum forces a negative value below UNDERrange, the output value will be set to UNDER.

Adder State Machine (Refer to the Adder State Machine diagram)

The following are inputs to the Adder State machine:

Test – When the WCA is in Test Mode, the adder "expects" the EXP to read the data normally destined for the Display. The decoded IORC(L) pulse provides the read signal, via U324B on diagram 29.C/T(L) MODE (Compress/Transparent MODE, from U724 on diagram 29)When the WCA is in the compress mode, the adder state machine will transmit both its input values to U100A, B, and C (diagram 2) and U224 (diagram 31) as a min/max pair or points. In the transparent mode, only one value will be sent.

DREAD(L) (Data READ, from U324B on diagram 29) – DREAD(L) Is the decoded data read signal. It is the result of an EXP input operation from address 1002_{hex}.

RSEL (Register SELect, from U332B on diagram 29) – RSEL indicates to the adder (diagram 31) which of its two inputs to process first.

(A)DATAIN(L) (Adder DATA IN, from U330A on diagram 31) – When the compressor latches data into the adder input registers, U330A asserts the (A)DATAIN(L) signal. The ADDLATCH(L) signal from the compressor clocks U330A.

WCARST(L) (Waveform Compressor/Adder ReSeT, from U232A on diagram 29) – WCARST(L) notifies the adder state machine that a reset is active. Regardless of its present state, the state machine will change to an initial, known state when WCARST(L) is asserted. The machine will stay in that state as long as WCARST(L) remains asserted.

The following are outputs from the Adder State machine:

MUXEL (MUltipleXer SELect, from U224 on diagram 31) – MUXEL, the multiplexer control line for the adder input register, is decoded from RSEL.

ADDAV (ADDer AVailable, from U224 on diagram 31) – ADDAV indicates that the adder is idle (ADDAV high = adder idle). A high on ADDAV means that the adder will accept data from the compressor. ADDAV is also connected to the WCA Status word register.

DDATALATCH(L) (Display DATALATCH, from U224 on diagram 31) – The DDATALATCH(L) line coverts the transfer of data to the Display. DDATALATCH(L) also sets the (A)DATAIN line to a low logic level. For more detail about DDATALATCH(L), see the external specification.

DSENDNEW(L) (Display SENDNEW, from the Display (unit) – The Display asserts a low–logic level on DSENDNEW(L) when it wants the adder to send new data to the Display.

DATE(L) (DATA Output Enable, from U224 pin 17 on diagram 31) – The DATAOE(L) signal enables the cable buffers which furnish data to the Display.

Display to MMU Transmission

The MMU controls the arbitration of the bi–directional data path between it and the Display. The WCA does not interfere with that arbitration; yet it interacts with it by providing buffers which must be enabled for the Display to MMU path. The buffer enable signal is derived from the DATAGATE(L) interface line. It is delayed by U100D to ensure that the data is valid long enough for the MMU to read it.

A17 Main Processor Board <23> and <24>

After the user requests an oscilloscope operation (with a front panel control for instance), the Main Processor board (MPB) directs the oscilloscope in performing the operation. Another primary function of the MPB is to run diagnostic self–tests on the oscilloscope when powering up or when requested by the user. The MPB controls oscilloscope operations by controlling and monitoring the other circuit boards sharing the Executive System bus. Through the Executive bus boards, the MPB also indirectly controls all other oscilloscope boards. The Central Processing Unit (EXP) generates command and status signals to control on board devices (i.e., Numeric Processor Extension, Interrupt controllers, Bus controller), which help process data and help control the rest of the oscilloscope. The Central Processing Unit is referred to elsewhere in the Theory of Operation section as the Executive Processor or EXP.

The EXP effectively controls the system by running firmware routines stored in EPROMs located on the MPB and on the Memory board. Along with the Numeric Processor Extension (NPE), the EXP does all data processing not directly related to generating the display or digitizing the waveform. When power is first applied to the system, the EXP runs local and system diagnostic tests, which are located in the on–board EPROMs and on the Memory board.

To reduce the load on the EXP, the Interrupt controllers monitor the many interrupt lines for service requests from other devices in the oscilloscope. When an interrupt occurs, the Interrupt controllers notify the EXP and supply it with a vector address to a firmware routine to service the interrupting device.

As a system option (Option 4D), a DMA (direct memory access) controller can be installed on the MPB. The DMA device increases the data transfer rate for the GPIB external interface. When operating, the DMA controller has full control of the Microprocessor and Executive buses while the EXP is idle in a Hold condition.

The Bus controller runs the local Microprocessor bus and the Executive bus with optimal bus cycle timing. It adds wait states (additional clock cycles) to the current bus cycle only when they are requested by a system device.

The Clock generator provides synchronizing clock signals for the MPB and the rest of the Executive boards. It also produces device ready and system reset signals that are in sync with the clock signal.

Central Processing Unit (Exp) (diagram 23)

The Central Processing Unit, U830, referred to hereafter as the Executive Processor (EXP), controls oscilloscope operations by deciding which devices can use the Microprocessor and Executive busses and when. The EXP also processes acquired and stored waveform data. The EXP gets instructions for performing its operations from software stored in EPROMs located on the Main Processor board and the Memory board. The EXP controls all Microprocessor Bus and Executive Bus operations, through the Bus controller, except when the optional DMA controller has been granted control. Peripheral devices (such as the Memory Management Unit or external interfaces) request service from the EXP by generating an Interrupt Request.

The EXP programs and controls the Interrupt controllers, the Numeric Processor, and the optional DMA controller. This programming is done at power–up initialization and may also occur during normal operation.

The clock rate of the EXP is 6 MHz, which is derived internally by dividing the CLK input, pin 31 of U830, by two.

The EXP controls on–board devices with a variety of status and control signals The status lines (S0(L), S1(L) and M/IO start each bus cycle and determine its type. Table 2–12 lists the possible states of these signals and the names of the associated bus cycles. Bus cycles end when the input READY(L) comes active.

Table 2–12
EXP Status and Byte Types

MI/O	S1(L)	S0(L)	Type of Bus Cycle
0	0	0	Interrupt Acknowlege
0	0	1	I/O Read
0	1	0	I/O Write
0	0	1	Idle
1	0	0	Halt or shutdown
1	0	1	Memory Read (Instruction)
1	1	0	Memory Write
1	1	1	Idle

When Bus High Enable(L), BHE(L) (pin 1), is low it indicates that data will be transferred on the upper eight data lines, MD7 – MD15. BHE(L) and MA0 together signal whether the next transfer is for a word or for only the upper or lower byte. Table 2–13 shows this relationship. Control output Code/Interrupt Acknowledge, COD/INTA, distinguishes between instruction reads from memory reads. (In address decoding, it is used to distinguish between INTA and I/O bus cycles.) If M/IO, S1(L) and S0(L) are low, a low on COD/INTA indicates an Interrupt Acknowledge cycle.

Table 2–13
Data Transfer Type

BHE(L)	MA0	Type of Data Transfer
0	0	Word
0	1	Upper byte valid
1	0	Lower byte valid
1	1	Undefined

Four EXP signal lines (PEREQ, PEACK(L), BUSY(L), and ERROR(L)) provide efficient communication between the EXP and the NPE. For the functions and timing of these signals, refer to the discussion of the Numeric Processor Extension.

The Interrupt controllers monitor the system interrupt lines to ensure that the highest priority interrupt gets serviced by the EXP. When a system interrupt is received, the Master Interrupt controller asserts the INTR signal line to notify the EXP that an interrupt is pending. The EXP will finish execution of its current instruction before acknowledging and servicing the interrupt. The EXP acknowledges the interrupt by asserting INTA(L) two times in succession. The Interrupt controllers place a vector address to jump to a software routine on the Microprocessor Data Bus lines MD0 – MD7 for the EXP. The EXP runs the interrupt handling routine, then returns to its previous operations. For more information on interrupt processing, see the discussion of the Interrupt controllers.

When the optional DMA controller is installed, it uses the HOLD and HLDA lines for bus requests and bus grants, respectively. The DMA controller drives HOLD high when it needs control of the Microprocessor and Executive busses to service a peripheral device. When the EXP is ready to relinquish the busses, it responds by asserting the HLDA line and tri–stating (put in a high impedance state) its address, data and control line outputs.

Numeric Processor Extension (diagram 23)

The Numeric Processor Extension (NPE), U500, is a high–speed floating–point processor that executes instructions in parallel with the EXP. The NPE is programmed and controlled by the EXP as an I/O device at addresses $0F8_{hex}$ to $0FF_{hex}$. The NPE is enabled at the numeric processor select input, NPS1 (pin 34), by latched select line LS1.

The NPE and EXP share four dedicated signal lines which allow efficient operation and data bus sharing. When the NPE is ready to transfer data it asserts PEREQ, pin 24. The EXP asserts PEACK(L), pin 36, when it is ready to acknowledge the NPE request and generate the necessary address and bus cycle signals. While the NPE is executing a command, it keeps BUSY(L), pin 25, asserted. If the NPE encounters a problem while executing a command, it asserts ERROR(L), pin 26, which interrupts the EXP.

After the NPE has issued a PEREQ and received a PEACK(L), the Bus controller will generate an I/O read, IORC(L) (MC8), or an I/O write, IOWC(L) (MC9), command. These are received by inputs NPRD(L), pin 27, and NPWR(L), pin 28, respectively, and cause the NPE to begin a data transfer over the Microprocessor Data Bus. A data transfer bus cycle ends when input READY(L), pin 40, is asserted.

The EXP sends control commands to the NPE by enabling it with LS1(L) and by driving inputs CMD0, pin 29, and CMD1, pin 31, with latched EXP address lines LA1 and LA2, respectively. The data lines to the NPE can be driven with data to load its internal control registers or they can be read in order to get control and status register information. Most other signal inputs and outputs are also involved in controlling and sending data to the NPE. All inputs are sampled on the falling edge of CLK (MC0), pin 37.

Crystal oscillator Y700 drives the CLK input, pin 32, with an 8 MHz signal. This clock signal synchronizes internal operations of the 8 MHz NPE. Jumper J600 provides a way to stop all NPE activities by removing its clock signal for a board test.

The other inputs S0(L), pin 2, and S1(L), pin 1, and COD/INTA, pin 3, provide the NPE with current bus–cycle information.

The input Hold Acknowledge, HLDA (pin 38), line is normally low, to indicate that the EXP controls the busses. When the optional DMA controller (option 4D) is installed, HLDA tells the NPE, and other devices, that the EXP is not in control of the busses.

Bus Controller (diagram 23)

Bus controller U750 provides command and control signals for the Microprocessor and the Executive busses. It decodes EXP status and control lines to generate its command and control signals. Bus controller outputs are always enabled. U750 produces optimal bus cycle timing (minimum number of CLK cycles per bus cycle) and inserts wait cycles only while the READY(L) input remains high past the fourth CLK cycle. The timing of all bus signals is referenced to the input signal, CLK (MC0). At the falling edge of CLK, the input signals are sampled and the appropriate output signals are asserted.

Bus cycles are started by input signals, S0(L) (MC2) and S1(L) (MC3). Together with M/IO (MC4), they identify the type of bus cycle that is beginning. Table 2–14 shows this relationship and the resultant command output.

Table 2–14
Bus Cycle Type

MI/O	S1(L)	S0(L)	Type of Bus Cycle	Command Output
0	0	0	Interrupt Acknowlege	INTA(L)
0	0	1	I/O Read	IORC(L)
0	1	0	I/O Write	IOWC(L)
1	0	0	Halt or shutdown	None
1	0	1	Memory Read (Instruction)	MRDC(L)
1	1	0	Memory Write	MWTC(L)

Bus cycles end when the input, READY(L), goes low. Another active or idle bus cycle can begin immediately.

The DT/R (MC13) output signals that the EXP will either transmit or receive data. DT/R is used to control the drive direction of the Data buffers and is available on the Executive Bus from the Control buffers U751, U760. The DEN (MC12) output, which is available on the Executive Control Bus, enables the Data buffers at both ends of the Executive Bus. During consecutive write cycles, DEN remains active and DT/R remains high. If DT/R changes during a bus cycle it does so when DEN is low so that no more than one device will be driving the data bus at once.

Memory and I/O cycles have the same timing, although a different command output signal is asserted when reading or writing.

During an Interrupt Acknowledge cycle (decoded from S0, S1, M/IO), the INTA (MC10) output toggles twice. ALE and MCE (Master Cascade Enable) go high just before each INTA pulse. MCE enables the Interrupt CAS Address buffer. This allows the Cascade Address (generated by the Master Interrupt controller) to be put on the Microprocessor bus and Executive bus.

Reset Generator (diagram 23)

The Reset generator U140 provides a positive power–up reset for the Main Processor board and the entire oscilloscope. U140's output, pin 5, is gated with CLK (MC0), in the Clock generator and broadcast as RESET (MC17). In addition to generating a power–up reset, the Clock generator, U850, also produces a reset if the +5 V power supply fails.

The power–up reset remains asserted for 60 ms after the +5 V supply is stable to satisfy the reset timing requirements of the EXP and its coprocessors. The 60 ms delay is set by the 4.7 μF capacitor on pin 3 of U140.

The PWR–UP signal (generated by the power supplies to indicate proper operation) is connected to Resin, pin 2. If PWR–UP goes low due to a power supply problem, a Reset cycle will start. The 0.1 μF capacitor, C140, on pin 1 of U140 prevents transients on the +5 V supply from causing spurious resets.

Clock Generator (diagram 23)

The Clock generator circuit supplies the main clock signal, CLK (MC0), which synchronizes the Executive boards (i.e., those plugged in the Mother board). It also generates the synchronized READY(L) (MC16) and RESET (MC17) control signals. The Clock generator is comprised of the clock chip, U850, and an inverter, U550E.

The 12 MHz frequency of CLK is derived from crystal oscillator Y950 (U850, pins 7, 8). PCLK (MC1), pin 13, is one half the frequency of CLK and is synchronized with the microprocessor's internal EXP clock. Both CLK and PCLK have a 50% duty cycle.

The READY(L) (MC16) control line informs on–board chips that the current bus cycle is ending. READY(L) is asserted when the SRDY(L) input, pin 2, goes low. U550E inverts the SRDY line for the active low SRDY(L) input. The Executive bus signal, SRDY, is asserted when no additional wait states are needed by peripheral devices (e.g., memory chips, SDI chip, etc). SRDY is a Wire OR signal and is pulled high by a 300 Ω resistor connected to +5 V. This ensures that no wait states are inserted in the EXP bus cycle unless they are requested. Jumper J660 can be used to isolate the EXP from any wait state requests during troubleshooting.

The RESET (MC17) control signal initializes the

- Central Processing Unit (EXP,
- Numeric Processor Extension
- Bank Decode/Select
- DMA controller
- Diagnostic Status latch

The RESET (MC17) control signal also prevents NV RAM from being written and, after buffering, also initializes the other Executive bus boards. Note that the Interrupt controllers are not initialized by RESET (INIT by Power On or Software). RESET is a clock–synchronized version of the pin 11 input, RES(L).

Interrupt Address Buffer (diagram 23)

The Interrupt Address buffer provides the means to coordinate interrupt controllers on other boards with the Interrupt controllers on the MPB. Half of 8–bit buffer U530B drives the address lines MA1, MA2 and MA3 with the information on CAS0, CAS1 and CAS2, respectively. The Cascade Address (CAS) lines originate at the Master Interrupt controller. The Master Cascade Enable, MCE, signal from the bus controller is inverted by U550A, then used to enable the Interrupt Address buffer. J550 can disable this buffer.

Address Decode/Select (diagram 23)

Many VLSI chips on the MPB have a variety of operating modes and must be configured by the EXP to function in the desired mode. Whether initialized at power–up or reprogrammed while running, the devices must first be selected (enabled to receive configuration data), then sent specific commands or data. Device selection is done with the Address Decode/Select circuit, which is mainly composed of U710 and U711 (custom PALs) and U630 (dual 4–input NAND gates).

The EXP sends command data to an I/O or memory address reserved for the device being configured (like sending a letter to a mailbox address). The Address Decode/Select circuit decodes the address and control lines and asserts the appropriate chip–select line. The select lines are latched by the Select latches, then connected to their respective devices. Table 2–15 lists the select lines, their associated devices, and their reserved I/O or memory addresses.

Table 2–15
Select Lines Address Assignments

Select Line	Address Space	Address (*hex*)	Device
S1(L)	I/O	0FB–0FE	Numeric Processor Extension (U400)
S2(L)	I/O	200–202	Interrupt Controller Slave 1 (U321)
S3(L)	I/O	400–402	Interrupt Controller Slave 3 (U340)
S4(L)	I/O	900–902	Interrupt Controller Master (U330
S5(L)	I/O	0A00–0AFF	DMA controller (optional, U401)
S6(L)	M	30000–33FFF	Nonvolatile RAM (U200, U210)
S7(L)	M	0F8000–0FFFFF	EPROM (U211, U220)
S8(L)	I/O	8002	Diagnostic Status latch
S9(L)	I/O	8004	Wait State Diagnostic Enable (U100D)
S10(L)I/O	I/O	0000	Memory Bank Select (550, U530A)

Data Buffers (diagram 23)

The Data buffers interface the Microprocessor Data bus and the Executive Data bus. They transmit data to or receive data from the Executive Data bus as directed by the EXP or optional DMA controller. The Data buffers connect to the Executive Data bus with P104.

Data buffers U860 and U861 are 8–bit bidirectional buffers with tri–state (third state is high impedance) outputs. When disabled their output drivers are set to the high–impedance state. This isolation from the Executive bus is necessary for on–board component communication.

The chip enable inputs, pin 19, are connected to the output of the Data Buffer Enable circuit, U740, through jumper J960. Pull–up resistor R960 ensures that the Data buffers are disabled when J960 is removed.

The drive direction of the Data buffers is controlled by DT/R (MC13), which is connected to pin 1. When MC13 is high, data is driven to the Executive Data bus; when low, data is received from the Executive Data bus.

Address Buffers (diagram 23)

The Address buffers drive the Executive Address bus with the address for the current bus cycle. The address can be generated by the EXP or by the optional DMA controller. In the normal running mode, these buffers are always enabled to drive the Executive Address bus, at P104.

The 8–bit bidirectional Address buffers are U650, U651, and U660. Each chip enable, pin 19, is grounded to constantly enable the buffers. Each buffer's direction control (pin 1) is connected to a 4.7k Ω pull–up resistor and to jumper J800, pin 4. J800 allows the Microprocessor Address bus to be driven by the Executive Address bus when diagnostic tests are running. J800 is the Kernal Test connector and is discussed fully in the Diagnostics section of this manual.

Control Buffers (diagram 23)

The Control buffers provide an interface to the Executive Control bus for the 16 Microprocessor Control bus signals, MC0 – MC15. U751 and U760 are eight–line buffers. Their output enables, pins 1 and 19, are grounded to constantly enable the outputs. The buffer outputs connect to the Executive Control bus with P104.

Data Buffer Enable (diagram 23)

When communication between on-board components is in progress, the Data Buffer Enable circuit disables the Data buffers. When communication is with the Executive bus, the data buffers are enabled by DEN (MC12). DEN is generated by the Bus controller to indicate data is ready to be sent or received.

If any of U740's inputs go low, its output goes high, disabling the Data buffers. The inputs consist of five Latched Chip Select lines, one control line (DEN(, and ICEN(L)) (equivalent to a Chip Select), and two lines pulled high with a 4.7k Ω resistor connected to 5 V. The Latched Chip Select inputs are:

- LSI – Numeric Processor Select
- LS5 – Direct Memory Access (DMA) Select
- LS6 – Nonvolatile RAM Select
- LLS7 – EPROM Select
- LLS10(L) – Bank Select

The ICEN(L) input signal is asserted by the Interrupt controller circuit whenever written or read or an Interrupt Acknowledge cycle is in progress. During an Interrupt Acknowledge cycle the Master Interrupt controller puts a vector address for an interrupt routine on the lower eight lines of the Microprocessor Data bus which are read by the EXP.

The U740 output (pin 8) goes through jumper J960 before reaching the Data buffers. Jumper J960 allows the Microprocessor Data bus to be isolated from the Executive Data bus while troubleshooting and while running Diagnostics.

Address and Select Latches (diagram 24)

The 8-bit latches U610, U620, U640 and U730 latch and hold the current state of device select lines, S1(L) – S10(L), and address lines, MA1 – MA16, when ALE (MC11) or EALE (Early ALE, U340-6) goes low. The outputs, Latched Address bus and Latched Select bus, remain stable until an input changes and another EALE occurs.

Memory Read/Write Control (diagram 24)

The Memory Read/Write Control provides separate write signals for the High and Low bytes of the nonvolatile RAM. The EXP can read or write on the upper eight data lines, MD8 – MD15, the lower eight data lines, MD0 – MD7, or on all 16 lines at once. The EXP uses status line bus High Enable, BHE(L), and address line, MA0, to signal which type of data transfer is planned. The nonvolatile RAM is selected for reading or writing when LS6(L) is asserted. Table 2-16 shows the relationships between the state of BHE(L) and MA0, the type of data transfer and the enable lines asserted.

Table 2-16
Enable Scheme for Nonvolatile RAM

Transfer Type	BHE(L)	MA0(L)	Enable Lines
High Byte	0	1	WEH(L)
Low byte	1	0	WEL(L)
16-bit word	0	0	WEH(L),WEL(L)

U110A uses ALE (MC11) to latch BHE(L) (MC14) and MA0. Gate U120A ANDs latched inverted BHE(L), inverted MWTC(L), and inverted Latched Select line LS6(L) to produce the high–byte enable. For the low–byte enable, U120B ANDs latched inverted MA0, inverted MWTC(L), and inverted LS6(L).

Power Down (diagram 24)

The Power Down circuit disables the nonvolatile RAM when it detects the power supplies failing or an active MRESET signal, or PWR UP false. When the Power Supplies are good, PWR UP is high, and MRESET is low, a one second time delay starts, after which the NV RAM is enabled. The RC time of C151–R164 sets the one second delay.

If PWR UP is pulled low by the power supply logic (R152 is PULL UP), Q170 will turn on which turns off Q171. This causes a low on the chip select line for the nonvolatile RAM, disabling the RAM.

RAM Battery (diagram 24)

The RAM Battery circuit provides the nonvolatile RAM (actually they are static rams) with a constant power source if the +5.5 V power drops E+3 V. The circuit acts as a switch that connects the 3 V Lithium battery BT160 to the VCC inputs, pin 28. The switch also prevents the Lithium battery from being charged by the +5.5 V supply.

CAUTION

Recharging or improper handling of a Lithium battery is dangerous. Refer to section 3, Maintenance, of the Volume 1 Service Manual for proper handling procedures for Lithium batteries.

Nonvolatile RAM (diagram 24)

Static RAM chips, U260 and U270, are used for permanent storage of high–byte and low–byte data, respectively. The nonvolatile RAM data lines interface with the Microprocessor Data bus through the Memory Data buffers. The chips are enabled for reads or writes by latched RAMSEL(L) (LS6(L)), which can be monitored at TP230. Memory Read/Write Control enables U260 and U270, on pin 27, when data is being written to them. When memory is being read, MRDC(L) (MC6) goes low on pin 22, enabling the output drivers.

The second chip–select input, pin 26, is used to deselect U260 and U270 when the power supplies fail or when the MPB is being reset. This chip enable signal is delayed for one second after a reset or power–up.

The constant VCC power for the static RAM chips, which makes them nonvolatile, is supplied by the +5.5 V supply or by a +3 V battery backup called the RAM Battery. A RAM Battery switching circuit automatically switches the supplies (CR150, CR151).

EPROM (diagram 24)

Part of the operations and diagnostic software resides in two EPROMS (27512s), U240 and U250. They provide 128 KB of read–only, permanent memory. The EPROMs are enabled by latched select line LS7(L) on pin 20. The data outputs are enabled by memory read line, MRDC(L) (MC6). The data output lines, which are shared with the non-volatile RAM, interface with the Microprocessor Data bus through the Memory Data buffers.

The A15 input, pin 1, of the 27512s is connected to latched address, LA16, by jumper J140. J140 provides for the use of a smaller, alternate EPROM, the 27256.

Memory Data Buffers (diagram 24)

The 8–bit bidirectional buffers U220 and U230 interface the Microprocessor Data bus to the nonvolatile RAM and the EPROMs. The Memory Data buffers allow the Microprocessor Data bus and the memory devices to meet their timing specifications without causing bus contention.

Removing jumper J330 disables the Memory Data buffers, which can help isolate problems on the MPB.

Memory Data Buffer Enable (diagram 24)

The three gates of U100 and inverter U510B comprise the Memory Data Buffer Enable circuit. U510B inverts control signal DT/R (MC13) for drive direction control of the Memory Data buffers. Gate U100A ensures that the nonvolatile RAM or the EPROM is selected. Gate U100B tests that a memory write, MWTC(L) (MC7), or a memory read, MRDC(L) (MC6), is in progress. Gate U100C checks that the Bus controller has issued DEN (MC12). When all these conditions are met, the Memory Data buffers are enabled.

Jumper J330 can be removed to isolate the memories from the Microprocessor Data bus.

Interrupt Controllers (diagram 24)

The Interrupt controllers constantly monitor the Executive system interrupt lines to ensure that the highest priority interrupt gets serviced first. The Interrupt controllers provide the ability to assign priority levels to all the system's interrupt lines and to ignore (mask) any of the interrupt lines. Interrupts are service requests for peripheral devices that have data for, or that are requesting data from, the EXP.

The three Interrupt controllers are programmed by the EXP as the Master, U350, Slave 1, U360, and Slave 3, U370. Their interrupt inputs are programmed for the level–sensing mode. The programming can be done at any time using IORC(L) (MC8), IOWC(L) (MC(9), LA1, and the appropriate Latched Select line. U350 is programmed at I/O address 900–902$_{hex}$ (LS4) with the information that it is the Master, input IR1 is Slave 1, and input IR3 is Slave 3. The slaves are programmed at address 200–203$_{hex}$ (LS2) and address 400–402$_{hex}$ (LS3) for Slaves 1 and 3, respectively. They are told which slave they are so they can respond to the Cascade Address issued by the Master during an Interrupt Acknowledge cycle.

When a high appears on one of the interrupt request input pins (IR#), the interrupt output INT (pin 17) immediately goes high. If the interrupt was on one of the Master's inputs, its output immediately interrupts the EXP. If the interrupt request was on one of the slaves' inputs, then the slaves' INT output drives the Master and the Master's output drives the EXP. Nothing more happens until the EXP finishes executing its current instruction and is ready to acknowledge the interrupt.

To produce an Interrupt Acknowledge cycle, the EXP drives the M/IO, SO(L), S1(L) and COD/INTA lines low twice, just as it would for two consecutive bus cycles. Each time these lines go low the Bus controller pulls the INTA(L) (MC10) line low. During the first INTA(L) pulse, all interrupt requests are latched into the Interrupt controllers. The Master then decides which of its interrupt inputs has the highest priority, and drives the Cascade Address bus (CAS0, CAS1, CAS2) with the number for the slave that is attached to that interrupt pin. If the pin is not a slave input, then the Master will drive a zero on the Cascade Address bus to signal that it will provide the interrupt vector. Through the second INTA(L) pulse the Cascade Address remains valid so the slaves can compare it to their own numbers. If one matches the Cascade Address, it puts a vector number on the Microprocessor Data bus, which specifies an interrupt handling routine.

A too–short pulse (a glitch) on an interrupt pin will not be latched by the first INTA(L) pulse. When the Interrupt controller is signaled (by the Cascade Address) to put an interrupt vector number on the data bus, it will errone-

ously supply the vector number for its lowest priority interrupt. This can cause degraded system performance and possibly corrupted data. (It is good practice not to connect to interrupt IRX7.)

The last chip in the block is U340B, which generates ICEN(L) for the Data Buffer Enable circuit. Each Interrupt controller has an output called EN(L), pin 16, which it drives low before putting data or a vector number on the data lines. The ICEN(L) signal disables the Data buffers (U860, U861) when the microprocessor communicates with these interrupt controllers. This eliminates problems with data bus contention when an Interrupt controllers are read.

Interrupt Bus Inverters (diagram 24)

The Interrupt Bus inverters are U561, U570 and U550F. They invert the Executive Bus Interrupt lines, INT0(L) – INT12(L), for the Interrupt controllers. The input are pulled high by 4.7 k Ω pull–up resistors to ensure that no erroneous interrupts occur when boards are removed from the Executive bus.

Wait State Generator (diagram 24)

The Wait State generator (WSG) allows system devices, that need extra time for data transfers, to extend the current bus cycle by 1, 2 or 4 EXP transfer cycles (TC). The inputs to the WSG come from device Select and Latched Select lines and from the I/O board via J77. The output, SRDY, is sampled by the Clock generator(L) at the falling edge of phase 1 of the current TC and used to generate the Microprocessor Control Bus signal, READY(L).

A wait state request must be received by D flip–flop U411B before any wait states will be generated. U720 responds to active select lines with a high output to OR gate U311A. Gate U311A also monitors the I/O Wait Req signal from the I/O board, which signals that a device there is requesting wait states. The output of U311A is the wait request for the D input of U411B. This wait request is clocked in by ALE (MC11). The Q output (pin 9) of U411B is inverted by open–collector U560C to become SRDY. When a wait request is clocked through to set SRDY low, the EXP will insert at least one wait state. The RESET R(L) input, pin 13, will be high at this time.

The remaining WSG circuitry allows the required number of wait states to be selected. U511 monitors the latched version of the select line inputs. These all require one wait state. U311B has as inputs the I/O one wait request input and the output from U511. Some I/O board devices require two wait states. The two Waits input is buffered by U311C and driven to gate U320B. Four wait states is the default when a wait request is received and no number of wait states is specified.

Flip–flop U411A and counter U420 count out the requested number of wait states. When the Q output of U411B goes high (because of a wait request), it removes the reset from U441A, pin 1, and from U420, pin 9. With the reset removed, U411A will begin to divide system clock CLK by two. Flip–flop U411A's Q output, pin 5, drives the clock input of the counter, pin 8. At the third rising edge of CLK, the Q1 output (pin 4 of U420) will go high, indicating a completed count for one wait state. If two waits were requested, two clock cycles later U420's Q output, pin 5, will go high. If no number of wait states is indicated, counter U420 will continue counting until its Q4 output, pin 10, goes high. Gate U320C is used to stop SRDY low when one wait state was requested. Gate U320C stop SRDY low when two wait states were required. If the output of either of these gates goes low, or of the inverted four–count signal goes low U320A's output (pin 12) will go high. When inverted by U430D this drives the reset input U411B–13 causing the Q output to go low. This signal is inverted and SRDY is driven active, (HI) signaling the end of the wait request. (U411B–13 also resets the wait state generator, U411A and U420.) The asserted SRDY signal is used by the Clock generator to produce READY(L) (MC16), which signal the end of the current bus cycle.

Wait State Diagnostic Enable (diagram 24)

The Wait State Diagnostic Enable (WSDE) circuit allows the device ready signal (SRDY) to be monitored on the Executive bus. An inverted SRDY signal is taken from U411B–9 in the Wait State generator. SRDY is enabled to the Executive bus by setting latch output U110B–11 to a high state. When an I/O write (IOWC(L)) occurs and diagnostic select (LS9) is set low, the output of U130D–11 will go low. This low output will be inverted, enabling data bit MD0 to be latched into U110B. If MD0 is high, any change in SRDY will be reflected on the DIAGNSIG(L) line of Executive bus. If MD0 is low, the SRDY signal will not appear on DIAGNSIG(L).

Diagnostic Status Latch (diagram 24)

The Diagnostic Status latch latches diagnostic test error codes from the EXP. The latch drives two LEDs and six Error Status Test Points. The error code on MD0 – MD7 will only be latched at the end of an I/O write bus cycle when LS8(L) is active. The reset input, pin 1, ensures that the latched outputs are all low on power–up or after a system reset.

DMA Controller (Optional) (diagram 24)

The Direct Memory Access controller option, U800, provides high–speed data transfer capabilities for some communications ports. When the DMA is installed, J800 jumper must be moved to pins #1 and #2. (Without the DMA option it goes on.) It is wired in parallel with the Microprocessor Address, Data and Control busses and can use the Executive buses. The DMA controller requests control of the busses with the HOLD line when one of its DREQ(L) lines is pulled low by a peripheral device and potentially at other times. The EXP grants control with the Hold Acknowledge, HLDA, line. Timing for DMA bus cycles is the same as that of the EXP. For a timing diagram, see the discussion on the Central Processing Unit (EXP). LS5(L), on pin 8, enables the DMA controller when the EXP is instructing it for its next task.

When the EXP is in the Hold mode, it tri–states all its outputs so that the DMA can drive them. The DMA controller is programmed through I/O addresses 0A00 – 0AFF$_{hex}$, to let it know where its sequence of instructions is located in memory. The DMA controller is directed by the EXP to start and to run at a specific maximum number of bus cycles per second. It has three channels to three different devices. Each channel has a set of DMA Request (DREQ) and DMA Acknowledge (DACK) lines for peripheral–device–initiated transfers. The channel two End of DMA (EOD) line has been programmed as a general all–channel interrupt to the EXP. This is used to tell the EXP that the DMA is finished with its assigned task.

When the DMA controller is not installed, the EXP handles the peripheral device requests with interrupt service routines.

A18 Memory Board <25> and <26>

The Memory board provides the Executive Processor (EXP) with DRAM and EPROM for most operations. Support circuitry for the memories and diagnostic circuitry for troubleshooting are located on–board. All accesses to DRAM or EPROMs are initiated by the Main Processor board (MPB) and specifically by the 80286 Executive Processor or the optional DMA controller. See the discussion of the Main Processor board in this section for bus cycle timing information.

Test points TP200, TP500 and TP800 provide ground connections for test oscilloscope probes.

Address Latches (diagram 25)

Address latches U540, U640 and U642 buffer and hold the Executive Address lines and BHE(L) for the EPROMS, DRAM controller and other on–board devices. With their outputs always enabled and LE HIGH, these latches are transparent, allow addresses to pass through as soon as they are available. The addresses are latched when the LE input is driven low by MRDC(L), MWTC(L), IORC(L), or IOWC(L) from the MP. At the end of a memory cycle, LE goes high and the latches again are transparent. LE (An Early ALE) can be monitored on test point TP801.

Address Decode (diagram 25)

Programmable array of logic (PAL) U542 decodes the address lines to produce RAM select signal RAM SEL(L) and four EPROM select signals, CS1(L) – CS4(L). RAM SEL(L) enables the DRAM controller to begin a DRAM access cycle. Each EPROM select line is latched, along with four Executive address lines, by U540. The latch outputs are always enabled. While latch enable (LE) is high the latch outputs are responsive to changes on the inputs. Table 2–17 shows the addresses that U542 requires to produce the select outputs. Note, A20 – A23 are Bank–Select Information.

Table 2–17
PAL Address Decoding

Select	Address Range	*hex*		Chips Enabled
RAM SEL(L)	Low Bank Switch	X	0–01FFFF	All RAM
		1	20000–3BFFF	All RAM
CS1(L):	High Bank Switch	0	C0000–DFFFF	U630, U730
CS2(L)		1	80000–9FFFF	U620, U720
CS3(L)		1	A0000–BFFFF	U612, U712
CS4(L)		1	C0000–DFFFF	U600, U700

The real address line inputs A14 – A19 are generated by the EXP on the MPB. Lines A20 – A23 are memory bank–select lines, which are encoded by the Bank–Select circuitry on the MPB. (This is true when J541 on the MPB is in position.)

EPROM Select (diagram 25)

The EPROM Select circuitry provides latched EPROM select lines and an enable signal for the EPROM Data buffers. Transparent latch U540 latches and buffers the EPROM select lines and drives them on its constantly enabled outputs. The outputs reflect changes on the inputs until the latch enable (LE) input goes low, at which time the current inputs are latched onto the outputs. At the end of a memory cycle, LE becomes high causing U540 to become transparent and ready for another memory cycle. The LE signal can be monitored on TP801.

When gate U632A detects an active EPROM line latched Select(L), it generates EPROM SEL, which enables the EPROM Data buffers when BDEN is true on a read cycle. EPROM SEL also signals the Memory Wait State generator to request wait states from the EXP. EPROM SEL can be monitored on test point TP806. Jumper J543 allows the EPROM Data buffers to be disabled during board testing. The EPROM select lines can be monitored on one of the test points TP802 – TP805.

EPROM (diagram 25)

The EPROMs contain most of the operating system code and diagnostics code for the EXP. All the EPROMs share the latched address bus. (These address lines are just buffered until the middle of the access cycle when they get latched. This is to allow the address information to be available to the EPROMs as early in the bus cycle as possible.) The EPROMs are organized into high– and low–byte pairs. Each pair is selected by a separate latched chip–select signal, which is generated by the Address Decode PAL. The output data drivers of an EPROM pair are enabled when the pair is selected. The high–byte chips share data lines ED8 – ED15 and the low–byte chips share data lines ED0 – ED7. These data lines are buffered to the Executive Data bus by the EPROM Data buffers.

The EPROMs installed can be either 27256s or 27512s. The circuit is configured by moving jumper J541 and by changing the PAL U542. The standard oscilloscope is supplied with 27512s. In the standard configuration, J541 should be in the A position.

EPROM Data Buffers (diagram 25)

The EPROM Data buffers drive data from the EPROMs on the Executive Data bus. Both 8–bit buffers are enabled by the output of U830A when all its inputs are high. The enable line can be monitored on test point TP809. The inputs to U830A are: EPROM SEL, which signals that an EPROM pair is selected; BDEN, which signals that the EXP is ready to read data; and inverted DT/R, which signals whether the memory operation is a read or write.

Bank Enable (diagram 25)

The Bank Enable circuitry buffers address lines A20 – A23 to produce bank address lines BA0 – BA3. The lower two, BA0 and BA1, contain the RAM bank address. The upper two, BA2 and BA3, contain the EPROM bank address (but the EPROMs get their bank address information through PAL U542 pins 1 and 2). These lines can be monitored on test points TP501 – TP504, respectively. U530 gates BA0 and BA1 with latched address line LA17 to produce address signals (BA17, BA18) that are dependent on which RAM bank is selected. These go through jumpers to the DRAM controller as AH6 and AH7 (only with 16 RAM, J501).

Memory Wait State Generator (diagram 25)

When a memory access starts, the Memory Wait State generator signals the EXP to extend the bus cycle a specific number of clock cycles (PCLK).

When an EPROM pair is selected, the EPROM SEL line goes high at the D input of U820B. If jumper J800 is not set on zero waits, the high will be clocked by bale to the Q output. The high Q output is SRDY(L), which is inverted by U552 to drive the system signal SRDY. When SRDY is driven low, it signals the EXP to begin inserting wait states. The high Q output also removes the reset condition from U820A. U820A divides the S CLK signal by two and clocks shift register U810. Jumper J800 determines which output of U810 gets inverted by U732A, and therefore, how long SRDY remains low. When the selected output of U810 goes high, it causes U820B to be reset, along with rest of the EPROM wait circuit, ending the wait request condition. Test point TP808 can be used to monitor SRDY(L), which signals EPROM wait requests. SRDY(L) also can be read by the diagnostics with AND gate U452B.

Wait requests from the DRAM controller, AACKA(L), are ANDed with L RAMSEL, which is latched–inverted RAM SEL(L). The output of gate U530A is inverted and used to pull SRDY low, requesting wait states from the EXP. The DRAM controller determines the number of wait states requested. Test Point TP206 can be used to monitor this line.

Memory Diagnostic Signal Select (diagram 25)

This diagnostic circuit causes either of two signals generated on the board to be driven on the Executive bus line DIAG SIG(L). One signal is SRDY(L), which is generated by the EPROM Wait State generator. The other is a combination of the DRAM row address strobe lines. (This is a RAM refresh.) Reading either of these signals requires a write to diagnostic I/O address 8020$_{hex}$. The address is decoded and the resultant device select active low signal) is latched into U352A by BALE. The latched output is gated with IOWC(L) (I/O write) by U450D. When this output is high the latch U352B is enabled and data lines D0 and D1 are latched. D0 (HI) enables the read of SRDY(L) and D1 (HI) enables the read of the RAM refresh. This last signal can be monitored on test point TP204.

Memory Configuration Readback (diagram 25)

This circuit allows the diagnostics to read the position of the RAM memory configuration jumpers and the Bank address lines. The Executive Processor does an I/O read at address 8040$_{hex}$ to read the information on the lower eight data lines. U350A decodes the address and produces a low signal, which is latched by U352A as BALE goes low. The latched device select signal is gated with IORC(L) (I/O read) and used to enable the read buffer U832.

DRAM controller (diagram 26)

DRAM controller U410 is configurable for DRAMs of different sizes and speeds in a two– or four–bank arrangement. It provides high speed access to the standard two banks of 64K x 4–bit DRAM chips, and it also automatically provides refresh signals. Though the DRAM controller supports dual–port access to the DRAM, this circuit uses only the port A interface. In addition to the DRAM controller chip, a set of initialization shift registers and a high/low byte enable circuit are in this block.

The EXP cannot access the DRAM on power–up or on reset until the DRAM controller has been configured. The DRAM controller is configured while the rest of the oscilloscope is in its 45 ms reset state. The on–board Reset generator produces a 10 ms reset pulse when the system reset goes active. During this time, shift registers U320 and U420 are loaded via jumpers J322 and J422 (jumpered with ECB runs) with the configuration data for the DRAM controller. After 10 ms the reset signal is removed from the DRAM controller. The DRAM controller then produces

16 clock pulses from its MUX/PCLK output that shifts the 16 configuration data bits into the PDI input. This takes about 20 μs. About 30 ms later the rest of the system will come out of reset and the EXP will start running.

The system status lines S0(L) and S1(L) are inverted and used to drive the RDA(L) and WRA(L) inputs. These signals are generated by the EXP. RAM SEL(L), generated by Address Decode, enables port A of the DRAM controller at input PEA(L) to allow DRAM accesses.

The WRA(L) and RDA(L) inputs determine whether a read or write access to RAM is requested. Address inputs AL0 – AL8 are used to generate the row address, and inputs AH0 – AH8 are used to generate the column address.

The output AACKA(L) signals the EXP that additional time is needed for a DRAM access.

Outputs WE and PSEN are gated by U332B, C and U340B, C with latched inverted versions the EXP signals BHE and A0(L). The gates produce write enable signals HBYTE WE(L) and LBYTE WE(L). BHE and A0(L) are also gated with latched RAM SEL to produce the DRAM Data buffers enable signals HBYTE OE(L) and LBYTE OE(L). These enable signals can be monitored on test points TP300 – TP303. In addition, RAMSEL can be monitored on test point TP205.

Also in this circuit block, the EXP transmit and receive signal DT/R is buffered. When the EXP is reading from DRAM, the output enables of the DRAM are activated by BDT/R. Inverted DT/R is used to select the drive direction for the DRAM Data buffers and can be monitored on test point TP207.

DRAM (diagram 26)

In the standard configuration, the DRAM is comprised of two groups of four 64K x 4-bit DRAMs. The Memory board is designed to allow two more groups of RAM to be added later. The two groups used are:

- Group 0

  U110, U112, U120 and U122,

- Group 1

  U210, U212, U220 and U222.

The DRAM controller and the buffered EXP control line DT/R control the DRAM. The DRAM may be written to with just a high or low–byte of data. The high and low write–enable and the high and low output–enable lines provide this function. Reading just a high or low byte is controlled by enabling one of the DRAM Data buffers with the output–enable lines (although this is not necessary). Data to and from the DRAM chips uses the RD data lines, which are paralleled to each group of DRAMs. The RD data lines are connected by the DRAM Data buffers to the Executive Data bus. Access timing is shown in this section in the DRAM controller discussion.

DRAM Data Buffers (diagram 26)

The DRAM Data buffers provide an interface between the DRAM data bus and the Executive Data bus. Drive direction is controlled by the inverted EXP control signal DT/R. Each buffer chip can be enabled separately to allow just high or low byte reads and writes. Signal HBYTE OE(L) enables high–byte buffer U740; likewise, LBYTE OE(L) enables low–byte buffer U742. The in–line resistor packs provide impedance damping of ringing (especially negative voltages), on the DRAMs sensitive inputs.

DRAM Configuration (diagram 26)

The DRAM Configuration jumpers allow the DRAM to be arranged in different ways to facilitate various customer options and design considerations. The jumpers select which address and bank–address lines will be applied to the DRAM controller. Jumper J521 provides for use of 64K x 4- or 256K x 4-bit DRAMs. J501 allows two additional groups of DRAM to be added. J520 provides for switching to the virtual addressing mode of the EXP. (Eliminates bank switching of the DRAM.)

DRAM Controller Reset Generator (diagram 26)

The DRAM controller Reset generator monitors the system RESET line and uses it to generate a shorter reset pulse for the DRAM controller than for the system reset. The 45 ms system RESET pulse is buffered, then differentiated. The resultant quick pulse is applied to the input RESIN. U430 produces a 10 ms reset pulse, which appears on the output where it gets inverted and applied to the DRAM controller. It gets inverted again and applied to the shift–load input of shift registers U320 and U420. When power is first applied to the oscilloscope, U430 monitors the +5 V line. When the +5 V line reaches 4.7 V, the RESET(L) output is driven active and remains active for 10 ms. After this delay the RESET(L) output is brought back high, initializing the DRAM controller.

Contents

Section 3—Maintenance

Section 3
Maintenance

This section of the manual contains information for performing preventive maintenance and corrective maintenance for the 11401/11402 Digitizing Oscilloscopes.

Preventive Maintenance

Preventive maintenance, performed regularly, can prevent instrument breakdown and may improve the reliability of the instrument. The severity of the environment to which the instrument is subjected will determine the frequency of maintenance. A convenient time to perform preventive maintenance is preceding electrical adjustment of the instrument.

Cabinet Panel Removal

WARNING

Dangerous potentials exist at several points throughout this instrument. When the instrument is operated with the covers removed, do not touch exposed connections or components. Some transistors have voltages present on their cases. Disconnect power before cleaning the instrument or replacing parts.

The top and bottom cabinet panels (or covers) provide protection from operating potentials present within the instrument. In addition, they reduce radiation of electromagnetic interference from the instrument. Fasteners retain the cabinet panels. To remove the panels, loosen the fasteners and lift the panels off. Operate the instrument with the panels in place to protect the interior from dust.

Cleaning

The 11401/11402 should be cleaned as often as operating conditions require. Accumulation of dirt in the instrument can cause overheating and component breakdown. Dirt on components acts as an insulating blanket and prevents efficient heat dissipation. It also provides an electrical conduction path which may result in instrument failure. The side panels reduce the amount of dust reaching the interior of the instrument. Operation without the panels in place necessitates more frequent cleaning.

CAUTION

Avoid the use of chemical cleaning agents which might damage the plastics used in this instrument. Exercise care when cleaning Hypcon connectors; see cleaning instructions under Hypcon Connectors in this section. Use a nonresidue type of cleaner, preferably isopropyl alcohol or totally denatured ethyl alcohol. Before using any other type of cleaner, consult your Tektronix Service Center or representative.

Exterior

Loose dust accumulated on the outside of the instrument can be removed with a soft cloth or small brush. The brush is also useful for dislodging dirt on and around the front-panel controls. Dirt which remains can be removed with a soft cloth dampened in a mild detergent and water solution. Do not use abrasive cleaners.

Crt

Clean the crt faceplate with a soft, lint-free cloth dampened with denatured alcohol.

Interior

Cleaning the interior of the instrument should seldom be necessary. The best way to clean the interior is to blow off the accumulated dust with dry, low-velocity air (approximately 5 lb/in^2). Remove any dirt which remains with a soft brush or a cloth dampened with a mild detergent and water solution. A cotton-tipped applicator is useful for cleaning in narrow spaces, or for cleaning more delicate circuit components.

CAUTION

Do not use cotton-tipped applicators for cleaning on, or near, Hypcon connectors. Cotton fibers will stick to edges and surfaces causing open or intermittent connections. See the instructions in step 4 of the Hypcon Connector removal procedure (listed under Semiconductors).

Circuit boards and components must be dry before applying power to prevent damage from electrical arcing.

The high-voltage circuits should receive special attention. Excessive dirt in this area may cause high-voltage arcing and result in improper instrument operation.

Visual Inspection

The 11401/11402 should be inspected occasionally for such defects as broken connections, improperly seated semiconductors, damaged or improperly installed circuit boards and heat-damaged parts. The corrective procedure for most visible defects is obvious; however, particular care must be taken if heat-damaged parts are found. Overheating usually indicates other trouble in the instrument; therefore, correcting the cause of overheating is important to prevent recurrance of the damage.

Semiconductor Checks

Periodic checks of semiconductors are not recommended. The best check of semiconductor performance is actual operation in the instrument. More details on semiconductors are given later in this section.

Periodic Electrical Adjustment

To ensure accurate measurements, check the electrical adjustment of this instrument after each 2000 hours of operation, or every 24 months if used infrequently. In addition, replacement of components may necessitate adjustment of the affected circuits. Adjustment instructions are given in Section 2—Checks and Adjustments of the 11401 and 11402 Service Reference manual.

Corrective Maintenance

Corrective maintenance consists of component replacement and instrument repair. Special techniques required to replace components in the 11401/11402 Digitizing Oscilloscope mainframes are given here.

Power Supply Voltage Hazard

If you must work inside the power supply with power applied, do not touch any metal-faced part on the A2A1 Line Inverter board. The metal frame of the power supply chassis itself should be safe to touch. (However, a live circuit could short to the chassis, under certain conditions, and then the chassis would be dangerous to handle).

WARNING

All metal components, including any metal-faced ones, on the A2A1 Line Inverter board should be considered hazardous. *This is because these components are at the AC line voltage potential.*

Always remove the line power cord before any disassembly.

An electric-shock hazard exists when the 11401/11402 is NOT grounded. Do not remove the ground wire (green-yellow) that connects the power supply chassis to the mainframe.

Manually discharge the line storage capacitors on the A2A1 Line Inverter board before beginning any work inside the A2 power supply. *(See the Access to Components in the A2 Power Supply procedure, in this section.)*

Selected Components Criteria Table

At the time of printing, the 11401/11402 uses no selected components.

Obtaining Replacement Parts

All electrical and mechanical part replacements for the 11401/11402 can be obtained through your Tektronix field office or representative. However, many of the standard electronic components can be obtained locally in less time than is required to order them from Tektronix, Inc. Before purchasing or ordering replacement parts, check the parts list for value, tolerance, rating, and description.

NOTE

When selecting replacement parts, remember that the size and shape of a component may affect its performance in the instrument. All replacement parts should be direct replacements unless you know that a different component will not adversely affect instrument performance.

Special Parts

Some parts are manufactured or selected by Tektronix, Inc. to satisfy particular requirements. Some are manufactured for Tektronix, Inc. to satisfy our specifications. Most of the mechanical parts used in this instrument have been manufactured by Tektronix, Inc. To determine the manufacturer of parts, refer to the Parts List, Cross Index Mfr. Code Number to Manufacturer.

Ordering Parts

When ordering replacement parts from Tektronix, Inc., include the following information:

1. Instrument type
2. Instrument serial number
3. A description of the part (if electrical, include circuit number)
4. Tektronix Part Number

Component Color Coding

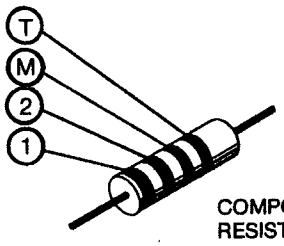
This instrument contains composition resistors, metal-film resistors, and wire-wound resistors. The resistance values of wire-wound resistors are usually printed on the component body. The resistance values of composition resistors and metal-film resistors are color coded on the components, using the EIA color code (some metal-film resistors may have the value printed on the body). The color code is read starting with the stripe nearest the end of the resistor. Composition resistors have four stripes, which consist of two significant figures, a multiplier, and a tolerance value (see Fig. 3-1).

Metal-film resistors have five stripes consisting of three significant figures, a multiplier, and a tolerance value.

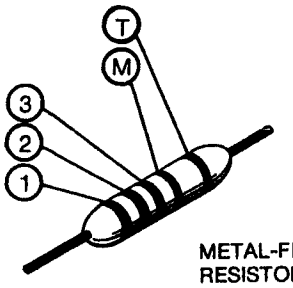
The values of common disc capacitors and small electrolytics are marked on the side of the component body.

The cathode end of glass-encased diodes is indicated by a stripe, a series of stripes, or a dot. The cathode and anode ends of metal-encased diodes can be identified by the diode symbol marked on the body.

COLOR CODE



COMPOSITION RESISTORS



METAL-FILM RESISTORS

(1)(2) and (3) - 1ST, 2ND, AND 3RD SIGNIFICANT FIGS.

(M) - MULTIPLIER (T) - TOLERANCE;

COLOR	SIGNIFICANT FIGURES	RESISTORS	
		MULTIPLIER (OHMS)	TOLERANCE
BLACK	0	1	——
BROWN	1	10	±1%
RED	2	10^2 or 100	±2%
ORANGE	3	10^3 or 1 K	±3%
YELLOW	4	10^4 or 10 K	±4%
GREEN	5	10^5 or 100 K	±1/2%
BLUE	6	10^6 or 1 M	±1/4%
VIOLET	7	——	±1/10%
GRAY	8	——	——
WHITE	9	——	——
GOLD	——	10^{-1} or 0.1	±5%
SILVER	——	10^{-2} or 0.01	±10%
NONE	——	——	±20%

7386-301

Figure 3-1. Color code for resistors.

Static-Sensitive Device Classification

CAUTION

Static discharge can damage any semiconductor component in this instrument.

This instrument contains electrical components that are susceptible to damage from static discharge. See Table 3-1 for the relative susceptibility of various classes of semiconductors. Static voltages of 1 kV to 30 kV are common in unprotected environments.

TABLE 3-1
Relative Susceptibility to Damage from Static Discharge

Semiconductor Classes	Relative Susceptibility Levels[1]
MOS or CMOS microcircuits, and discrete or linear microcircuits with MOS inputs (most sensitive)	1
ECL	2
Schottky signal diodes	3
Schottky TTL	4
High-frequency bipolar transistors	5
JFETs	6
Linear microcircuits	7
Low-power Schottky TTL	8
TTL (least sensitive)	9

[1]Voltage equivalent for levels.
1 = 100 to 500 V
2 = 200 to 500 V
3 = 250 V
4 = 500 V
5 = 400 to 600 V
6 = 600 to 800 V
7 = 400 to 1000 V (est.)
8 = 900 V
9 = 1200 V

(Voltage discharged from a 100 pF capacitor through a resistance of 100 ohms.)

Observe the following precautions to avoid damage:

1. Mimimize handling of static-sensitive components.
2. Transport and store static-sensitive components or assemblies in their original containers on an anti-static tube rail, or conductive foam. Label any package that contains static-sensitive assemblies or components.
3. Discharge the static voltage from your body by wearing a wrist strap while handling these components. Servicing static-sensitive assemblies or components should be performed only at a static-free work station by qualified service personnel. We recommend use of the Static Control Mat, Tektronix Part Number 006-3414-00, and Wrist Strap, Tektronix Part Number 006-3415-00.
4. Allow nothing capable of generating or holding a static charge on the work station surface.
5. Keep the component leads shorted together whenever possible by storing them in conductive foam or rails.
6. Pick up components by the body, never by the leads.
7. Do not slide the components over any surface.
8. Avoid handling components in areas that have a floor or work-surface covering capable of generating a static charge.
9. Use a soldering iron that is connected to earth ground.
10. Use only special anti-static suction-type desoldering tools.

Soldering Techniques

WARNING

To avoid electric-shock hazard and instrument damage, disconnect the instrument from its power source before soldering.

The reliability and accuracy of the 11401/11402 can be maintained only if proper soldering techniques are used when repairing or replacing parts.

The desoldering and removal of parts is especially critical and should be done only with an anti-static vacuum solder extractor; preferably, one approved by a Tektronix, Inc. Service Center.

Use wire solder with rosin core, 63% tin, 37% lead. Contact your local Tektronix, Inc. representative or field office for approved solders.

All circuit boards used in the 11401/11402 are multilayer. Conductive paths between the top and bottom layers may connect with one or any number of inner layers. If this inner conductive path is broken (due mainly to poor soldering practices), the board is unusable and must be replaced. **Damage can void warranty.**

CAUTION

Only an experienced maintenance person, proficient in the use of vacuum-type desoldering equipment, should attempt repair of any board in this instrument.

When soldering on circuit boards or small wiring, use only a 15-watt, pencil-type soldering iron. A higher wattage soldering iron can cause the etched circuit wiring to separate from the circuit board material. It can also melt the insulation from small wiring. Always keep the soldering-iron tip properly tinned to ensure the best heat transfer to the solder joint. Apply only enough heat to make a good solder joint. To protect heat-sensitive components, hold the component lead with a pair of long-nose pliers between the component body and the solder joint.

The following technique should be used to replace components on the circuit boards.

Touch the tip of the vacuum desoldering tool directly to the solder to be removed.

CAUTION

Excessive heat can cause the etched circuit wiring to separate from the circuit board material.

Never allow the solder extractor to remain on the board for more than three (3) seconds. Solder wick, spring-actuated or squeeze-bulb solder suckers, and heat blocks (for multi-pin components) must not be used. **Damage can void warranty.**

NOTE

Machine insertion of some components places a bend in both of the leads. These bent leads hold the component in position during a flow-solder manufacturing process which solders all of the components at once. Some components are difficult to remove due to their bent leads. To make removal of machine-inserted components easier, first remove the solder from the joint. Then straighten the leads of the components on the back of the circuit board, using a small screwdriver or pliers. Next, remove the component.

When removing multi-pin components, do not heat adjacent conductors consecutively (see Fig. 3-2). **Pause a moment to allow the circuit board to cool before proceeding to the next pin.**

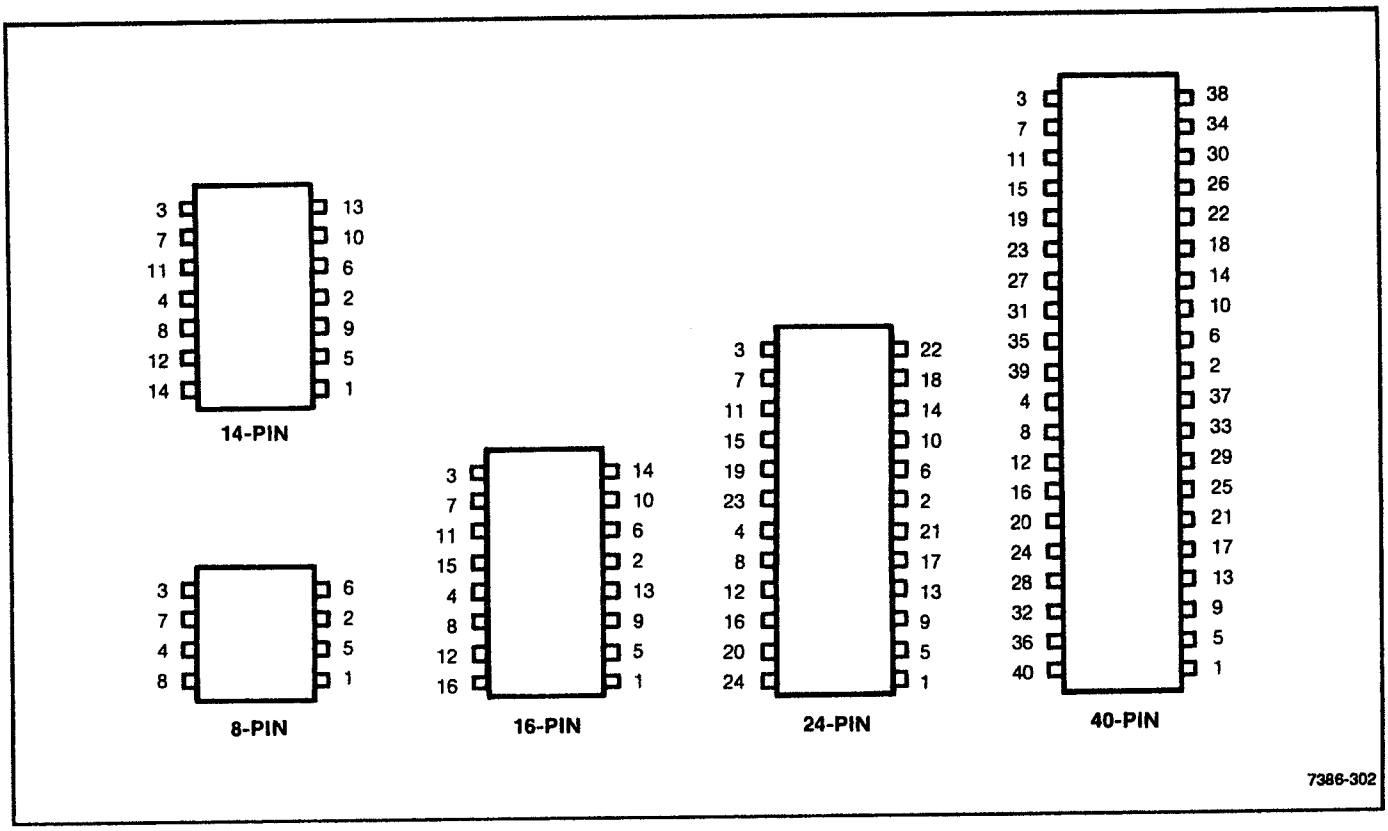


Figure 3-2. Recommended IC desoldering sequence.

Bend the leads of the replacement components to fit the holes in the circuit board. Insert the leads into the holes in the board, or as originally positioned.

Touch the iron to the connection and apply enough solder to make a firm solder joint.

Cut off any excess leads protruding through the board.

Clean the areas around the solder connection with a flux-removing solvent. Do not remove the information printed on the circuit board.

CAUTION

Before cleaning around Hypcon connectors, read the instructions about them. See the Semiconductors procedure, in this section.

TABLE 3-2
Part Number Reference For Support Items

Purpose	Item	Quantity Required	Location in Manual
Standard Accessories	User's Reference Manual	1	Appendix A— Accessory List "Standard Accessories"
Optional Accesories	Service Manual	1	Volume 2, Section 7, Replaceable Mechanical Parts "Optional Accessories"
Lubricant for Power Transistors	Silicone grease, Tektronix Part 006-1353-01	1 tube	Removing and Replacing Parts, "Semiconductors"
Discharge Static Voltage to Prevent Damage of Static-Sensitive Components	Static Control Mat Tektronix Part 006-3414-00 Wrist strap Tektronix Part 006-3415-00	1 1	"Static-Sensitive Device Classification"
Test Equipment for General Trouble-shooting	Sony/Tektronix 370 Programmable Curve Tracer Tektronix 577/177 Curve Tracer	1	Volume 2, "Equipment Required"
	Remote Diagnostic Operation Test Terminal, ANSI 3.64 Compatible with RS-232-C Interface Tektronix 7904A Oscilloscope with 7A26 Dual Trace Amplifier, 7S14 Dual Trace Delayed Sweep Sampler, and 7B10 and 7B15 Time Bases, or equivalent	1	

TABLE 3-2 (cont)
Part Number Reference For Support Items

Purpose	Item	Quantity Required	Location in Manual
Test Equipment (cont)	Technipower Autovariac W10MT3W, or equivalent	1	
	Tektronix DM 501A Digital Multimeter	1	
	Tektronix DC 510 350 MHz Universal Counter/Timer	1	
	Tektronix TM 506 Power Module Mainframe	1	
	Tektronix 1240 Logic Analyzer	1	
	Tektronix 067-1264-00 11000 Extended Diagnostics Unit	1	
	Tektronix 067-1267-00 Calibration Fixture, Troubleshooting Aid Extender Card with Cables	1	
	Tektronix P6401 Logic Probe	1	
	Tektronix 067-0587-02 Calibration Fixture (Signal Standardizer) modified for 11000 series.	1	
Integrated Circuit Extracting Tool	Tektronix Part 003-0619-00	1	Removing and Replacing Parts, "Semiconductors"
Sockets for Integrated Circuits	Number of Pins	Tektronix Part Number	
	8 14 16 20 24 40	136-0727-00 136-0728-00 136-0729-00 136-0752-00 136-0751-00 136-0757-00	

TABLE 3-2 (cont)
Part Number Reference For Support Items

Purpose	Item	Quantity Required	Location in Manual
Circuit Board Removal Tools	Straight-slot screwdriver, large Torx screwdrivers, T-7, T-8, T-10, T-15, T20, T-25	1 1 ea.	Removing and Replacing Parts, "Circuit Boards"
	Allen (Hex) Wrench, 1/16-inch	1	
	Nutdrivers, 3/16", 1/4", 7/16"	1 ea.	
	Needle-nose pliers	1	
	Soldering iron, 15 W	1	
	Vacuum solder extractor, anti-static, Pace PPS-4A or equivalent	1	
	Torque wrench, 0-10 inch-pound range	1	
	Shorting resistor, 100-200Ω	1	Removing and Replacing Parts. See step 2c of the "Access to Components in the A2 Power Supply" procedure.
Optional Tools	Flashlight, penlight-size	1	
Check GPIB Transfer Capability	IBM PC Model XT/AT Hewlett Packard 200/300 Series Desktop computer	1	User's Reference Manual, Tektronix Part 070-5791-00
Test Equipment for Checks and Adjustment Procedures	Refer to Test Equipment Table in the 11401 and 11402 Service Reference manual.		

Service Kits

1. **Extended Diagnostics 11000-series Power Supplies Part 067-1264-00.**

   The Extended Diagnostics unit (EDU) is a power supply troubleshooting tool for the 11301, 11302, 11401, 11402, and future 11000-series oscilloscopes. Using 21 LEDs, the EDU indicates the type of faults (such as current limit, voltage fault, or temperature fault) that may cause the power supply to shut down.

   The EDU connects to the J44 Diagnostic Port on the A2A2 Control Rectifier Board for testing.

2. **Calibration Fixture: Troubleshooting Aid Extender Card with Cables Part 067-1267-00.**

   This calibration fixture is a troubleshooting aid for the 11401 and 11402 oscilloscopes. It includes an extender card (with its cables) and a RS-232-C loopback device. The extender card enables plug-on circuit boards to be raised from the card cage to be probed. It also acts as a kernel stimulus allowing troubleshoting of the EXP kernel circuitry. The RS-232-C loopback device permits troubleshooting the circuits from the EXP μP out to the standard RS-232-C port.

Removing and Replacing Parts

WARNING

*To avoid electric-shock hazard and instrument damage, **always disconnect the instrument from its power source before removing or replacing components or plug-in units.***

The exploded-view drawings associated with the Replaceable Mechanical Parts List may be helpful in the removal or disassembly of individual components or subassemblies. (See Volume 2, "Diagrams and Parts Lists.")

The top and/or bottom covers will need to be removed for most repairs. Such removal is not mentioned in each procedure. As the covers would need to be removed before the individual circuit boards are located, it is assumed that they were off the instrument. Use a coin or a straight-slot screwdriver with a large-sized tip to loosen the cover fasteners. Rotating the fastener a quarter turn counterclockwise will loosen it.

Whenever a specific area is mentioned (such as the right side), it will usually be in reference to the front of the 11401/11402. If another reference is intended, it will be so described (such as the left side, as facing the rear).

NOTE

Refer to the "Adjustments Required After Circuit Board or Module Replacement" table at the end of this section.

Electrical Lock-on of the ON/STANDBY Power Switch

Some applications of the 11401/11402 Digitizing Oscilloscope may require that the power remain on. To electrically lockon the power (disable the ON/STANDBY Power Switch), use the following procedure:

1. Switch the PRINCIPAL POWER SWITCH to OFF (rear panel). Remove the AC power cable.
2. Position the instrument on its left (handle) side, as facing the front.
3. Remove the bottom cover.
4. Locate the **A4 Regulator board** as shown in Figure 3-8.
5. Locate the **J820 jumper** on the A4 Regulator board. (See Figure 3-14.)
6. Reposition the J820 jumper from its two outer (right side) pins to its two inner (left side) pins. Do not drop the jumper while moving it.
7. Remove the bottom cover. Turn the instrument right side up.
8. Reconnect the AC power cable and switch the PRINCIPAL POWER SWITCH to ON.

   CAUTION

   The power will now remain on regardless of the setting of the ON/STANDBY Power Switch.

   ***Do not install or remove a plug-in unit while the power is on.** Doing so may damage the mainframe and/or the plug-in.*
9. To turn the power off while the ON/STANDBY Power Switch is disabled, use the PRINCIPAL POWER SWITCH.

To return to normal operation of the ON/STANDBY Power Switch, follow the preceding instructions in reverse order.

A2 Power Supply Removal

The power supply can be slid out of the rear of the 11401/11402 for maintenance and troubleshooting. It may also be removed to gain better access to the A1 Plug-In Interface board, the A2A1 Line Inverter board, the A2A2 Control Rectifier board or the A4 Regulator board.

To remove the power supply from the mainframe, proceed as follows:

- Turn the mainframe on its left side (as viewed facing the rear panel). The power supply will now be at the bottom of the instrument.
- Remove the eight Torx screws that secure the power supply (see Fig. 3-3).

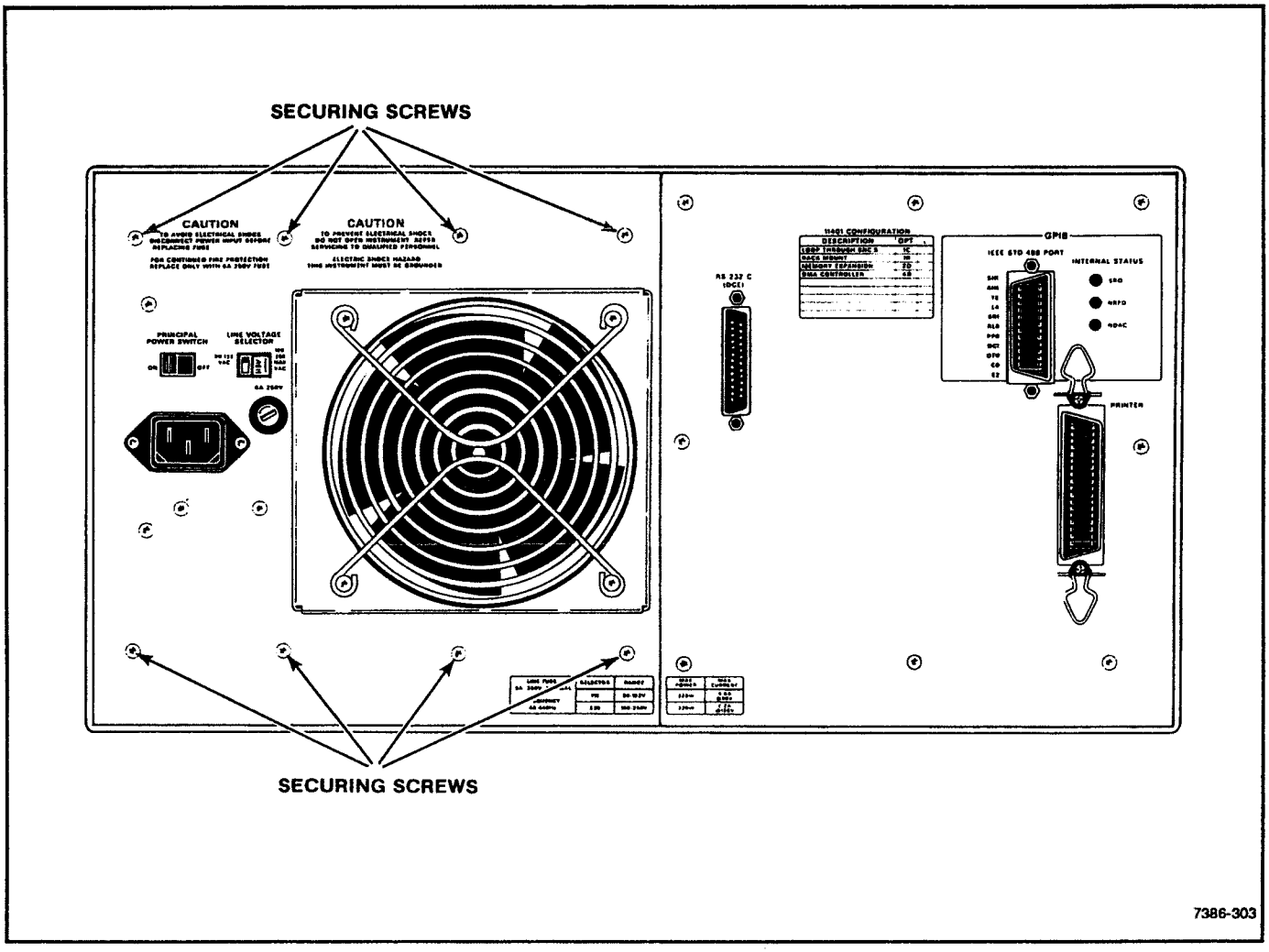


Figure 3-3. Rear panel location of screws securing the power supply.

- Carefully pull the power supply part way out of the mainframe. Stop before the wires (to the A2A2 Control Rectifier board connectors) begin to stretch taut or bind.

CAUTION

Excessive pulling on the power supply beyond this point may damage connector pins.

- The wires should be long enough to permit the power supply to be partially removed from the mainframe without disconnecting any wire connectors.

NOTE

Should disconnection of any wires be necessary, record their location for correct replacement. (See Fig. 3-4.)

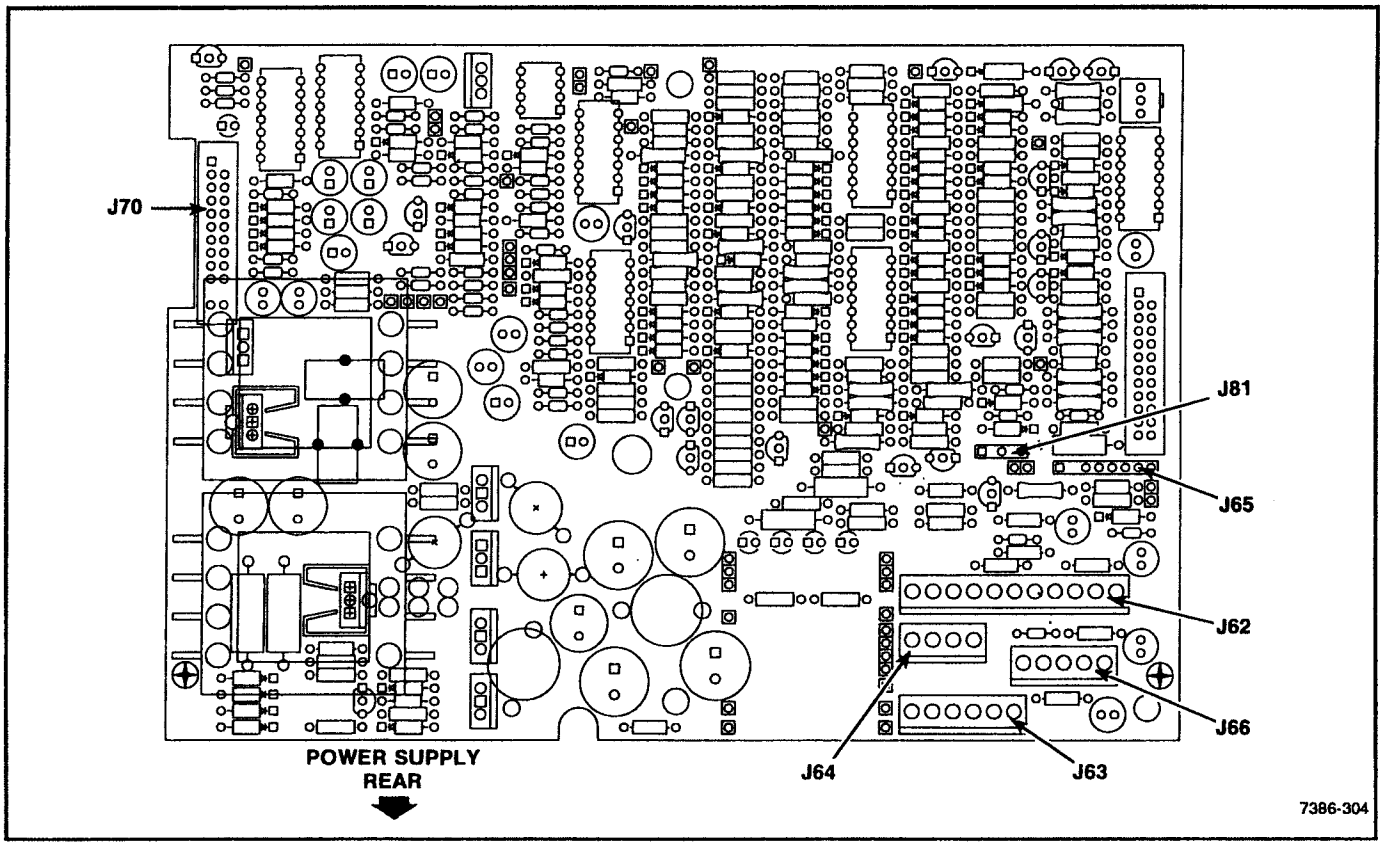


Figure 3-4. Bottom view of the A2A2 Control Rectifier board showing connector locations.

- Do not remove the chassis ground (green-yellow) wire. It should stay attached between the power supply and the mainframe.

  The power supply may now be removed from the mainframe. First remove the wire connectors from their pins on the A2A2 Control Rectifier board. Depending on the repair work to be done, it may be appropriate to set both the detached power supply and the mainframe right-side up.

- To apply power, attach the AC power cord to the power plug receptacle.

NOTE

The fan will revolve slightly slower than normal when the instrument is connected this way.

This connection will allow troubleshooting the power supply and/or power-related problems in the mainframe.

NOTE

*Before removing any of the power supply covers, read the warnings in the **Access to Components in the A2 Power Supply** procedure of this section.*
An extended diagnostics unit is available for troubleshooting the power supply. For a brief description and the Tektronix Part umber of this tool, see "Service Kits" near the beginning of this section.

To replace the power supply, follow the previous procedure in reverse order.

NOTE

Align the metal guides on the top of the power supply with the grooves inside the upper part of the opening in the mainframe.

Be careful not to pinch any wires or interconnecting cables while installing the power supply.

Replace the eight Torx screws into the rear plate and tighten them securely.

Access to Components in the A2 Power Supply

To reach the components inside the power supply chassis for maintenance or repair, use the following procedure:

WARNING

*Disconnect the 11401/11402 from the power source. **Allow the line storage capacitors to discharge before removing any cover from the power supply.** Unless they are manually discharged, these capacitors remain charged with a high DC voltage for several minutes after the line power is disconnected. **A warning-indicator DS640 (neon bulb) located on the A2A1 Line Inverter board flashes when this stored voltage exceeds about 80 V. Do not remove the power supply covers while this light is flashing.***

1. Remove the power supply as previously described.

   a. Disconnect the AC power cord before proceeding.

2. To reach the A2A1 Line Inverter board, first remove the protective cover from the top of the power supply chassis. This is done by removing the four screws securing the cover.

   a. Carefully slide the protective cover off the power supply chassis.

   b. The A2A1 Line Inverter board is now accessible for maintenance or repair. However, if the 11401/11402 is to be operated with the cover removed, heed this:

WARNING

***All metal components on the A2A1 Line Inverter board should be considered hazardous.** Such components remain at the AC line voltage potential.*

Once the power cord has been disconnected, manually discharge the line storage capacitors. (See Fig. 3-5).

   c. Before starting maintenance or taking resistance measurements, manually discharge the line storage capacitors (C200, C220, C310 and C320) as follows:

      - Remove the protective cover as previously described.

      - Locate the line storage capacitors on the A2A1 Line Inverter board. Place a 100-200 Ω, 400 V, 1-watt insulated resistor in parallel with A2A1R220 (470 kΩ) to discharge the capacitors, see Figure 3-5. (Glue a plastic rod to the shorting resistor to act as a safety handle.)

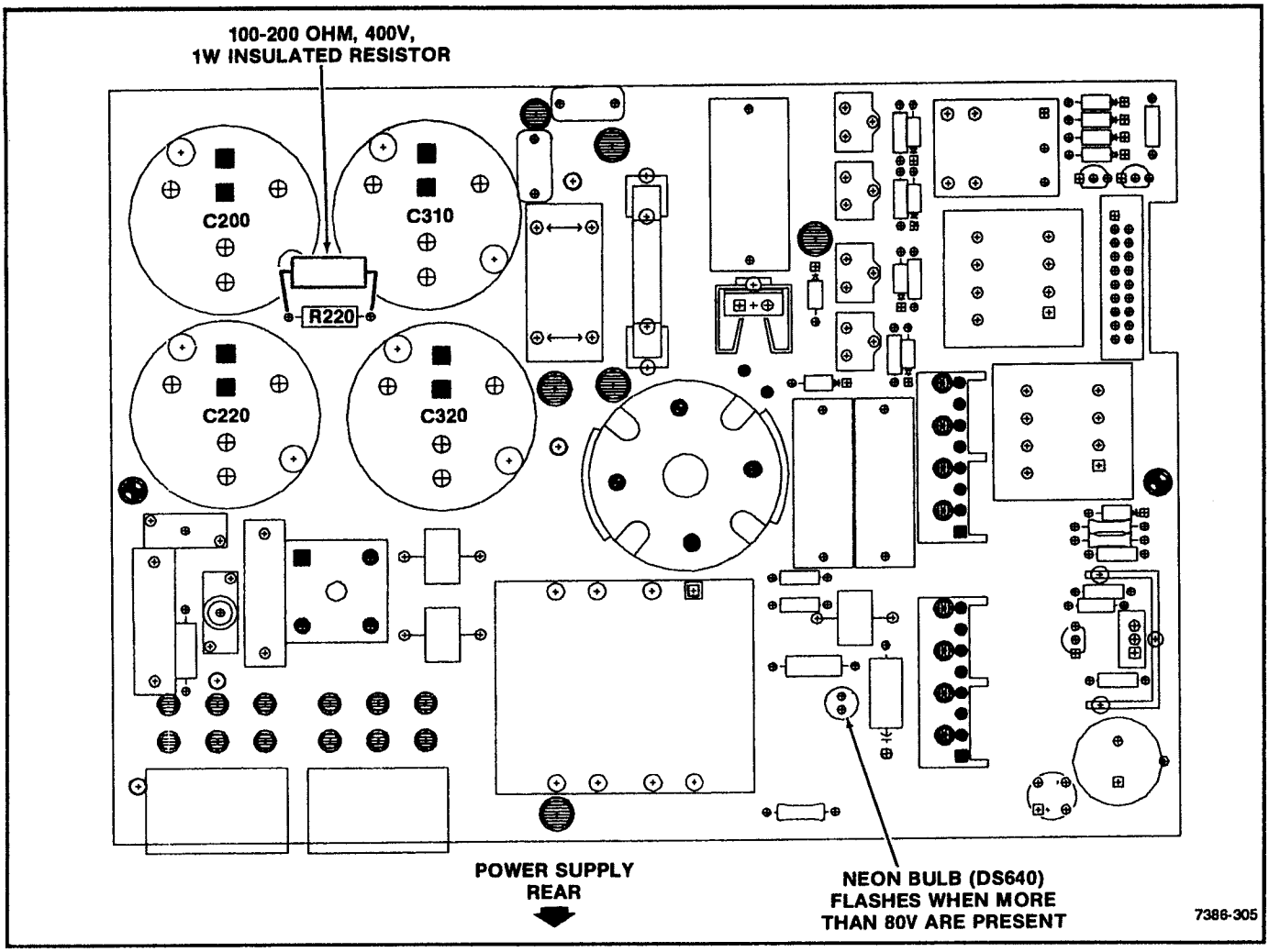


Figure 3-5. Manually discharging the line storage capacitors on the A2A1 Line Inverter board.

3. To replace the power supply protective cover, follow the removal procedure in reverse order.

4. To reach the line voltage circuitry (PRINCIPAL POWER SWITCH, Line Voltage Selector, fuseholder, etc.), proceed as follows:

   a. Remove the two long screws from the right side of the fan housing (as viewed facing the rear panel). (See Fig. 3-3.)

   b. Remove the screw above and left of the PRINCIPAL POWER SWITCH. Then remove the screw below and left of the power plug receptacle (the one closest to the left edge).

   c. Follow step 3 of the A2A1 Line Inverter Board Removal procedure, later in this section.

      - Remove the overall cover from the power supply.

   d. To replace the overall cover, follow this procedure in reverse order.

5. Complete access to the A2A2 Control Rectifier board or to the A2A1 Line Inverter board may be obtained by following the procedures to remove these boards. (See "Circuit Board Removal" in this section.)

6. Replace the power supply as previously described.

NOTE

Check that no wires contact the fan blades.

Fan Motor Removal

Remove and replace the fan motor as follows:

1. Mark the top of the fan motor housing. (It reassembles only one way.) Remove the four screws holding the assembly together. (See Fig. 3-3.)

   - Hold the housing as the last screws are removed.

2. Separate the grill and the housing from the motor.

3. Remove the two wires at their motor connections.

   - Note that the red wire is (+) and the brown wire is (–).

4. Remove the fan motor.

NOTE

Observe the position in which the motor was mounted. Replace it the same way, or the fan wires may not reach.

5. To replace the fan motor, follow the removal procedure in reverse order.

   - Don't pinch the wires under the fan housing.
   - Tighten screws securely. Remove the marking from the top of the housing.
   - Check that no wires contact the fan blades.

Cathode-Ray Tube Removal

Remove the cathode-ray tube (crt) as follows:

WARNING

*The crt may retain a dangerous electrical (12 kV) charge. Before removing the crt, the anode must be fully discharged. **Short the anode lead from the crt to the chassis. Wait approximately ten minutes and again firmly short this lead to the chassis.** Then, remove the crt.*

Use care when handling. Breakage of the crt causes a high-velocity scattering of glass fragments (implosion). Protective clothing and safety glasses should be worn. Avoid striking the crt on any object which might cause it to crack or implode. When storing a crt, place it in a protective carton. Otherwise, set it face down in a protected location. Be certain it is on a smooth surface with a soft mat under the crt faceplate.

1. Turn the 11401/11402 so that its front is at your right. Remove the two flat, Torx screws holding the crt shield to the chassis.

2. Lift up the outside of the shield.

   - The inner edge of the shield is held in place by two tabs. These fit into slots in the chassis beneath the inner edge of the A7 Display Controller board.

     As the shield is lifted, its bottom will clear the frame behind the instrument's handle. At that point, remove the shield carefully. Don't allow it to strike the crt.

WARNING

Crt anode voltage is 12 kV. Ground the lead to the chassis to short any stored charge remaining in the crt.

Wait about ten minutes and ground the anode lead again.

3. Use a non-conducting tool to pry up the rubber cap of the anode lead. (It is on the upper part of the crt behind the front casting. See Fig. 3-6.) Release the spring clip inside the cap and in the crt opening to remove the anode. Ground the anode to the chassis by inserting a screwdriver blade tip against the anode and touching the blade to the top of the front casting.

4. Remove the base-pin socket from the rear of the crt.

5. Disconnect connector J54 from the A8 CRT Driver board. It is located on the right front side, as viewed facing the crt. (See Fig. 3-6.)

6. Turn the mainframe on its right side. The crt will be at the top.

7. Remove the two flat-head screws from the bottom of the bezel. Turn the mainframe right side up.

8. Remove the two Control knobs from the front of the bezel. (Use an Allen wrench to loosen the knob setscrews.)

9. Lift up on the bottom of the bezel and swing it outward.

NOTE

The upper part of the bezel is held by two tabs. These tabs fit into two slots inside the front casting.

The wire cable from P73 on the A10 Front-Panel Control board may be removed (see Fig. 3-19). Disconnect connector P73 from the A10 Front-Panel Control board. Note the position of connector index triangles in order to correctly replace the connector. Carefully remove the wire cable through the slot provided in the front casting.

10. Remove the bezel.

    - Remove the wire connector from the quick-disconnect contact on the upper right corner of the front casting. Move the wire aside.

CAUTION

Be careful not to damage the interconnecting cable to P73, if it remains connected.

To protect it from being scratched or marred, cover the front of the bezel with some protective material.

11. Remove the four Torx screws and washers from the corner prongs of the band fastened to the faceplate.

CAUTION

Don't allow the crt to drop when loosening screws.

12. Hold one hand on the faceplate. Gently push forward on the crt base with the other hand. Slowly remove the crt from the front of the 11401/11402.

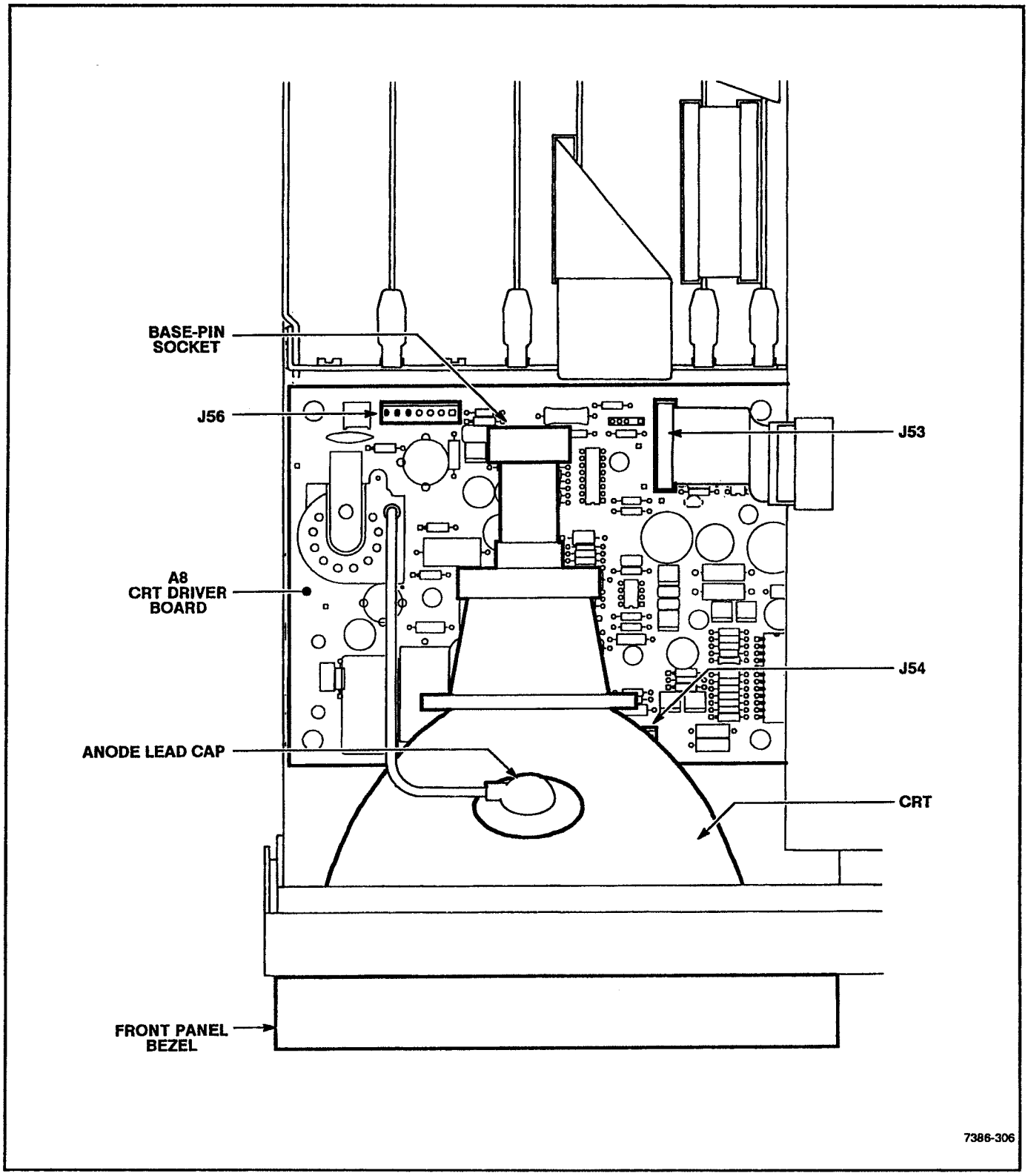


Figure 3-6. Top view of the Crt and A8 Crt Driver board.

Cathode-Ray Tube Replacement

Replace the crt as follows:

1. Replace connector J54 on the A8 Crt Driver board (see Fig. 3-6). Replace the wire connector to the quick-disconnect contact on the front casting.

2. Insert the crt into the front casting with the anode opening towards the top. The crt fits in downward and is pushed toward the left side. Align the corner prongs of the faceplate band around the four screwholes near the faceplate corners. Replace the four Torx screws and washers. Tighten securely.

3. Clean the crt faceplate with a soft lint-free cloth dampened with denatured alcohol. Be careful not to scratch the glass.

NOTE

If the P73 connector was disconnected when the front-panel bezel was removed, reconnect it. Route the connector and cable through the slot in the front casting. Match the index triangles of the connector and its holder on the A10 Front Panel Control board. Reattach the connector. Dress any cable slack toward the center of the chassis.

4. Replace the front-panel bezel. Insert the two tabs at the top of the bezel into the slots inside the front casting, above the upper edge of the faceplate. At the same time, center the two holes (near the bottom right) of the bezel around the shafts for each control knob. Push the bottom of the bezel backwards until it fits flush against the casting and in the side grooves of the casting. Don't pinch the interconnecting wire cable.

5. Replace the two Control knobs on their shafts. Tighten their setscrews securely.

6. Turn the mainframe on its right side. (The crt will be at the top.) Replace the two flathead screws in the bezel and tighten securely.

7. Turn the mainframe right side up.

8. Remove the protective cap from the crt base pins. Install the crt base-pin socket. Align the keyway of the socket with the gap between the pins on the base. Push the socket over the crt pins until it is seated.

9. Install the anode lead in the hole near the top of the crt. Inside the rubber cap is a spring clip. Put one side of the clip into the crt hole, then push the other side in. Check that the clip is connected by lightly tugging on the cap.

10. Replace the crt shield. Insert the two tabs on the shield's inner edge into their respective slots in the chassis. These slots are underneath the A7 Display Controller board.

    - Slide the outer edge of the shield behind the handle and inside the chassis. Align the countersunk holes with the threaded openings on the inside of the shield.

11. Reinstall the two flathead Torx screws. Tighten them securely.

NOTE

Replacing the crt will require that the instrument be readjusted. Refer to the "Checks Required after FRU Replacementi 'in the 11401 and 11402 Service Reference manual.

Lithium Battery Handling, Disposal, and First Aid

Two lithium batteries are connected to circuit boards in the 11401/11402. BT130 is soldered to the A14 Input/Output board and BT160 to the A17 Main Processor board.

Removal and Handling

WARNING

Wear safety glasses when desoldering or soldering a lithium battery.

To remove the battery from a board, apply a vacuum solder extractor to the joint where the battery is connected. Touch the solder extractor only on the tabs welded to the battery terminals.

WARNING

Never touch the battery itself with a hot solder extractor or soldering iron. Do not desolder or solder directly on the battery terminals.

Use the standard desoldering method covered in the "Soldering Techniques" procedure in this section. Once the solder is removed from the connection, remove the battery by pulling it free from the circuit board.

Do not drop a battery. Short circuits or other problems can be caused by dropping. Do not place a battery on any metal surface.

WARNING

*To avoid personal injury, observe proper procedures for the handling of lithium batteries. **Improper handling may cause fire, explosion, or severe burns.** Do not recharge, crush, disassemble, heat the battery above 212° F (100° C), incinerate, or expose the contents to water.*

Storage

When storing a battery, separate it from all metal or conductive surfaces. Do not store batteries loosely nor dump them into bins. Store them in their original shipping container or in individually wrapped plastic.

Board Cleaning

Battery salts, if present, will usually appear like a white powder upon contact with a circuit board. These salts may react with one or more layers of the board material, resulting in permanent damage and/or open traces.

Clean the salts from the board immediately with a brush and water. Any boards which were in contact with battery salts for more than a few minutes must be thoroughly tested for possible damage.

Discharging

WARNING

Never touch a battery directly with a hot soldering iron. Wear safety glasses when soldering or desoldering.

To discharge, solder a 10 Ω resistor across the battery to the tabs welded at each end. (Use the drawing shown in Fig. 3-7
for an example.)

After discharging for 24 hours, test the battery with an ammeter. If the reading is no more than 10 mA of current, the battery is considered fully discharged. (The resistor can be left attached for disposal.)

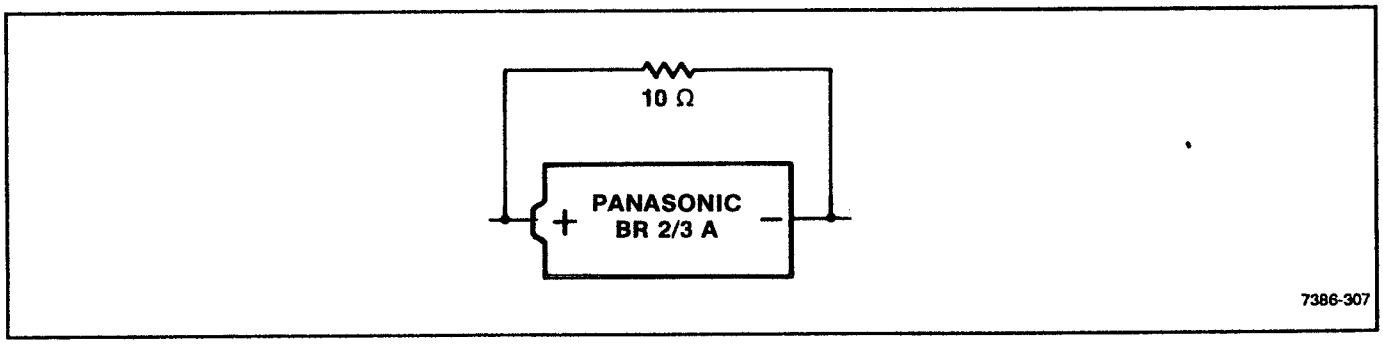


Figure 3-7. Lithium battery discharging through 10 Ω resistor.

Disposal

After the battery has been discharged, dispose of it according to local, state and federal regulations.

NOTE

Typically, small quantities (less than 20) can be safely disposed of with ordinary garbage in a sanitary landfill.

Larger quantities must be sent by surface transport to a Hazardous Waste Disposal Facility. The batteries should be individually packaged to prevent shorting. Then, pack them into a sturdy container that is clearly labeled, "Lithium Batteries — DO NOT OPEN".

Replacement

Re-install the new battery in the circuit board. Use the standard method to solder the connection, as covered in the "Soldering Techniques" procedure in this section.

NOTE

The preceding WARNINGs apply to soldering as well as to desoldering methods.

Emergency & First Aid Information

Manufacturer:	**Panasonic**
Battery Type:	**Lithium Poly-Carbon monoflouride, BR 2/3 A**
Solvent (electrolyte):	**Gama Butyrlactone is of low toxicity. It can cause some eye and respiratory irritation. The solvent may be released during venting, according to the manufacturer. (Venting is an outgassing of battery material.) This is usually caused by short circuiting (for more than a few seconds) or by overheating.**
Solute:	**LIBF4**

Should you come in contact with battery solvent

By	Do This
Contact with skin	Wash promptly with plenty of water.
Contact with eyes	Flush immediately with plenty of water and use an emergency eye wash, if available. Report to a medical professional for treatment.
Inhalation	Leave the area and get fresh air. Report to a medical professional for treatment.
Ingestion	Non-toxic according to laboratory testing. However, report to a medical professional for advice.

In case of venting, clear the immediate area. Venting will usually last only a few seconds.

Circuit Board Removal

The top and/or bottom covers will need to be removed for most repairs. Such removal is not mentioned in each procedure. As the covers would need to be removed before the individual circuit boards are located, it is assumed that they were off the instrument. Use a coin or a straight-slot screwdriver with a large-sized tip to loosen the cover fasteners. Rotating the fastener quarter-turn counterclockwise will loosen it.

All circuit boards used in the 11401/11402 Oscilloscope are multilayer. Conductive paths between the top and bottom layers may connect with one or any number of inner layers. If this inner layer conductive path is broken (due mainly to poor soldering practices), the board is unusable and must be replaced. **Damage can void warranty.**

If a circuit board is damaged beyond repair, replace the entire board assembly. Part numbers are given in the Replaceable Electrical Parts List in the Volume 2 Extended Service Manual.

To determine the location of a circuit board, see the exploded view in Figure 3-8.

Most circuit boards in the 11401/11402 are mounted on the chassis. Pin connectors are used for electrical interconnection with chassis-mounted components and other circuit boards. Five boards (A14 I/O, A15 MMU, A16 Waveform Compressor, A17 Main Processor, and A18 Memory) plug onto the top of the A13 Mother board. Feed-through connectors join the plug-on boards to the A13 Mother board.

CAUTION

After removing a circuit board from the instrument, place it upon a non-conducting surface. This will minimize the chance of static charge damage to the integrated circuits and/or related circuitry.

Some parts mounted on a board must be retained for use with the new assembly. These parts would include interconnecting plugs, supports posts, and some wiring.

Chassis-Mounted Boards

A1 Plug-In Interface Board

Remove and replace the A1 Plug-in Interface board as follows:

1. Remove the plug-in units.

2. Remove the nine screws that fasten the three interface connector receptacles to the chassis. (See Fig. 3-9).

3. Remove all the connectors from the A7 Display Controller board. Note the position of their index triangles so that the connectors can be correctly replaced. Remove the six Torx screws that fasten the board to the chassis. Remove the A7 Display Controller board. (See Fig. 3-17 and "A7 Display Controller board" later in this section.)

4. Remove the A2 Power Supply (see "A2 Power Supply Removal," in this section). Remove the connectors from the A2A2 Control Rectifier board, except for J70 and J81 (see Fig. 3-4 in this section).

NOTE

Record the positions of the connectors for correct replacement.

Return the 11401/11402 to its right side up position (if the instrument is on its side).

5. Disconnect connectors J57 and J60 from the A4 Regulator board (see Fig. 3-14). Remove the two Torx screws from the metal heatsink at the rear of the board.

NOTE

The A4 Regulator board is now unfastened from the chassis. However, it still remains connected to the A1 Plug-In Interface board.

Carefully disconnect the J95 and J96 interconnecting pins from the A1 Plug-In Interface board by pulling the A4 Regulator board towards the rear. Remove the A4 Regulator board.

6. Position the instrument so that the A1 Plug-In Interface board can be removed through the top of the mainframe chassis.

7. Remove the A1 Plug-In Interface board. Note the locations of the coaxial end-lead connectors and the interconnecting plugs so that they can be correctly replaced.

8. To replace the A1 Plug-In Interface board, follow the removal procedure in reverse order. Match the index triangle on the pin connectors to the corresponding triangle on the circuit boards or connectors. Correct location of the pin connectors is shown in the circuit board illustrations.

NOTE

To ease replacement of the screws in the A1 Plug-In Interface connector receptacles (see Fig. 3-9), start the screws when the A1 Plug-In Interface board is replaced into the instrument. Then, tighten all nine screws in the connector receptacles.

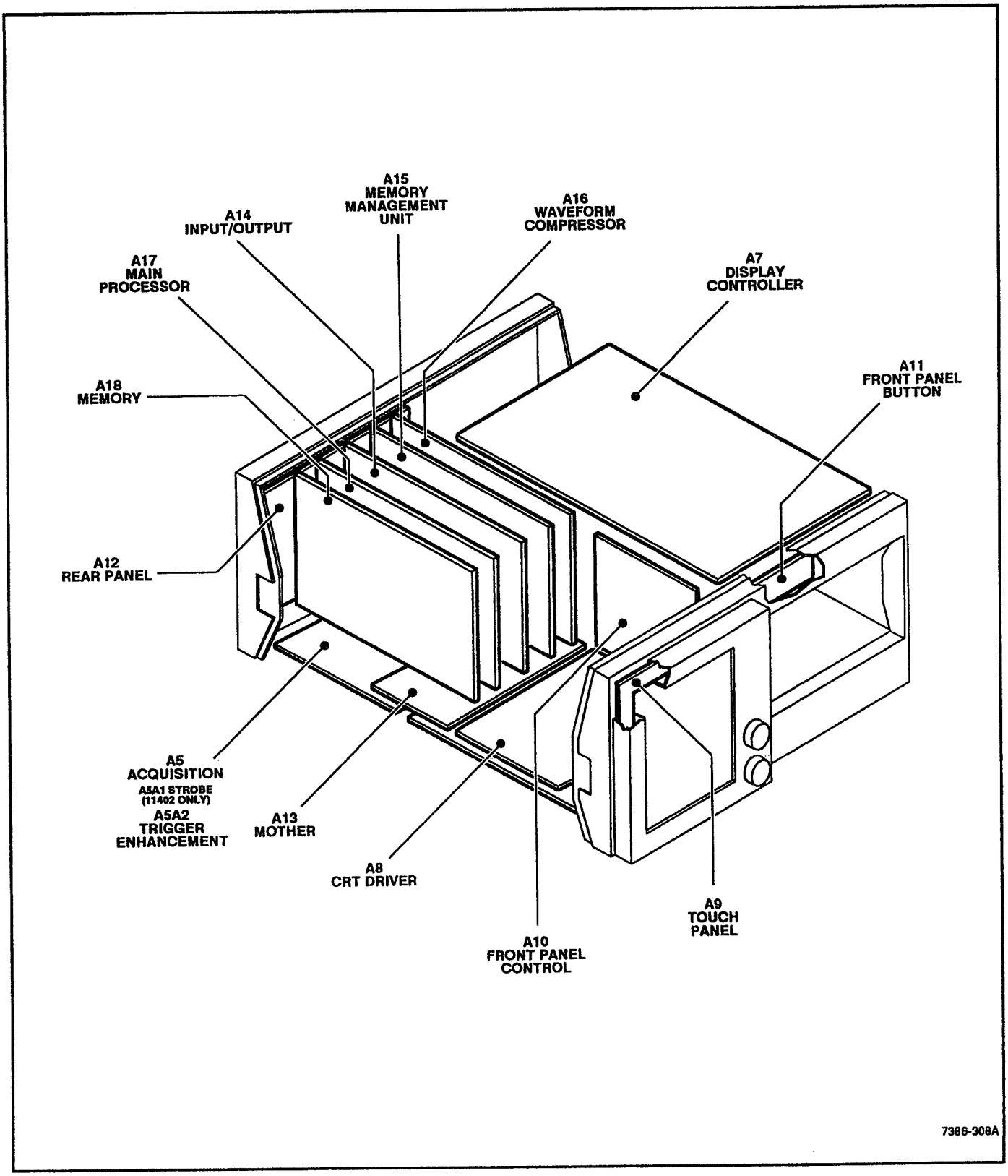


Figure 3-8. Location of circuit boards in the 11401/11402.

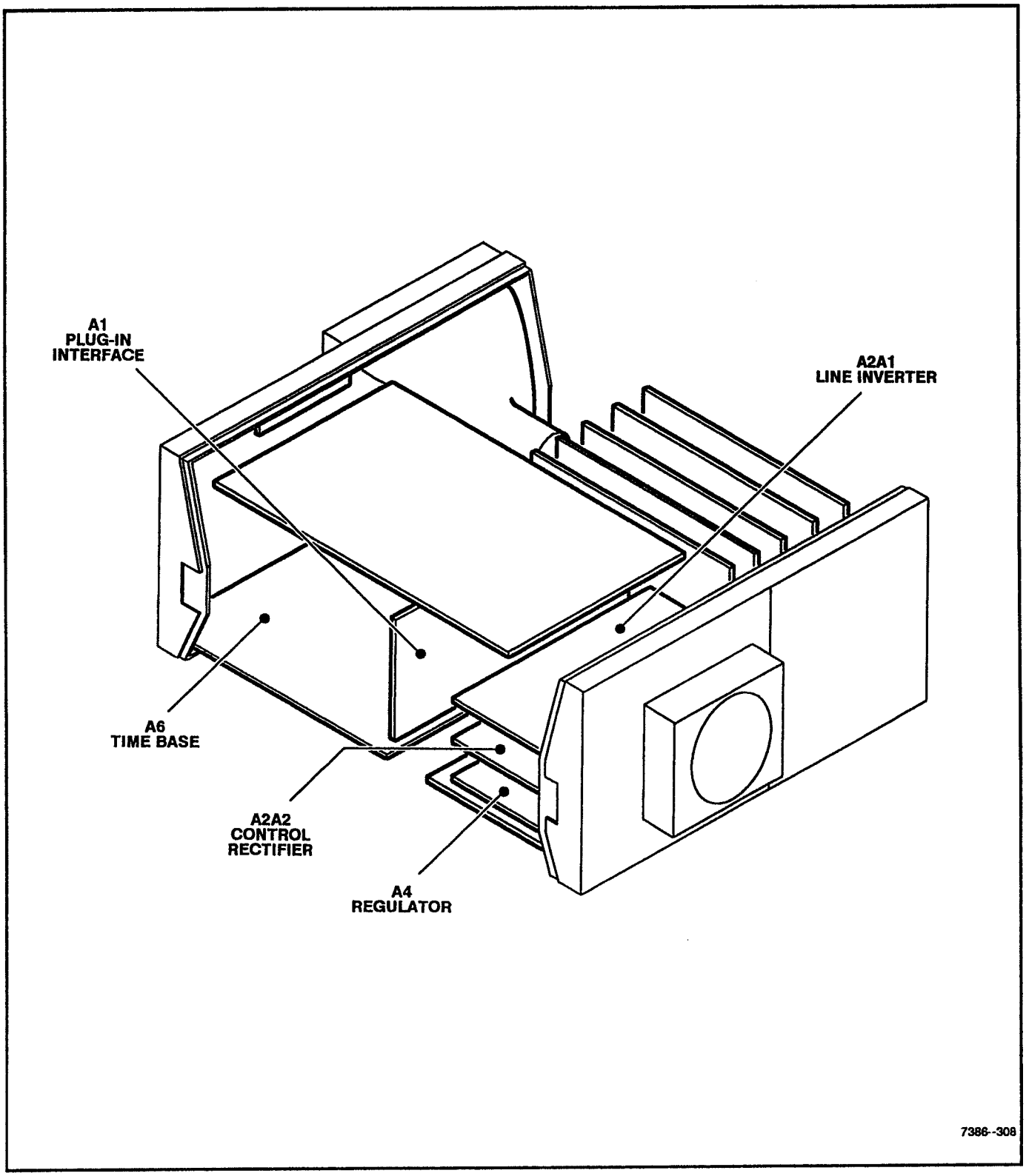


Figure 3-8 (cont). Location of circuit boards in the 11401/11402.

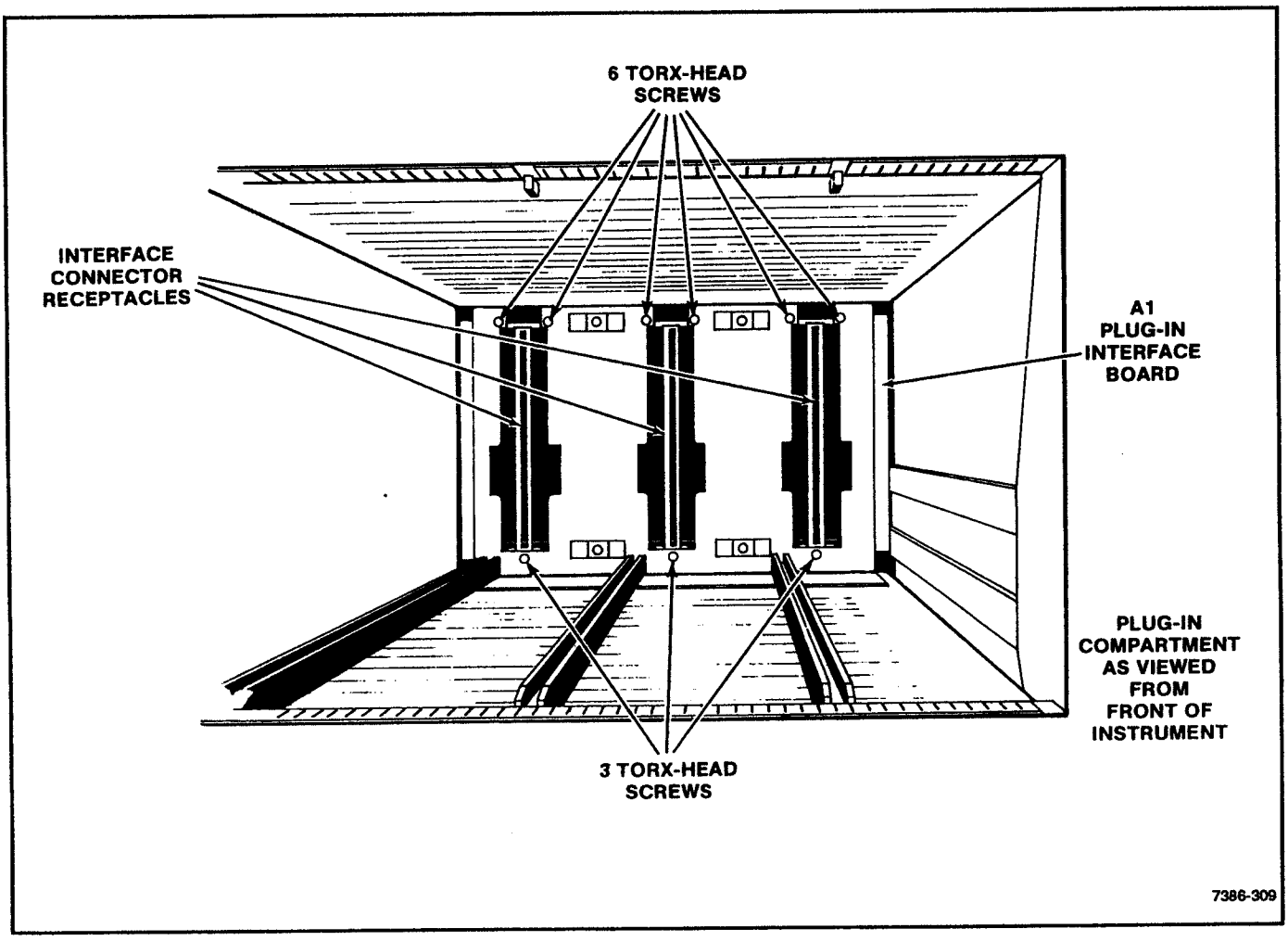


Figure 3-9. Plug-in compartment showing mounting screws for the A1 Plug-In Interface board.

A2A1 Line Inverter Board

Remove and replace the A2A1 Line Inverter board as follows:

1. Remove the power supply (see "A2 Power Supply Removal," in this section).

2. Remove the protective cover from the power supply. (See "Access to Components in the A2 Power Supply," in this section. Follow the WARNING and steps 1 through 2a.)

3. Remove the two screws that secure the A2A1 Line Inverter board (see Fig. 3-10). Then remove the screw holding the front of the overall cover to the A2A1 Line Inverter board. Remove the two screws from the bottom front of that cover.

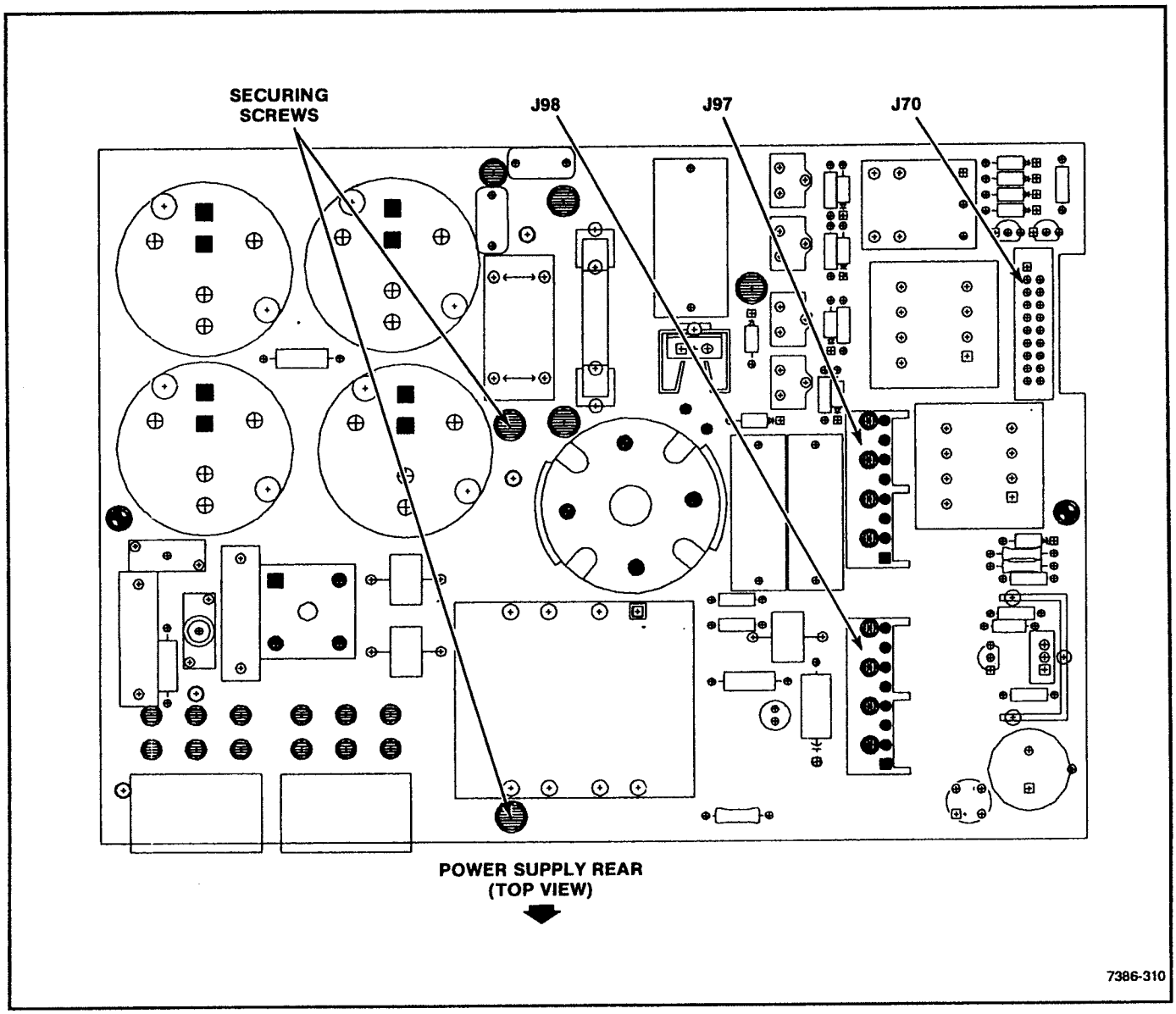


Figure 3-10. Connector locations for removal of the A2A1 Line Inverter board.

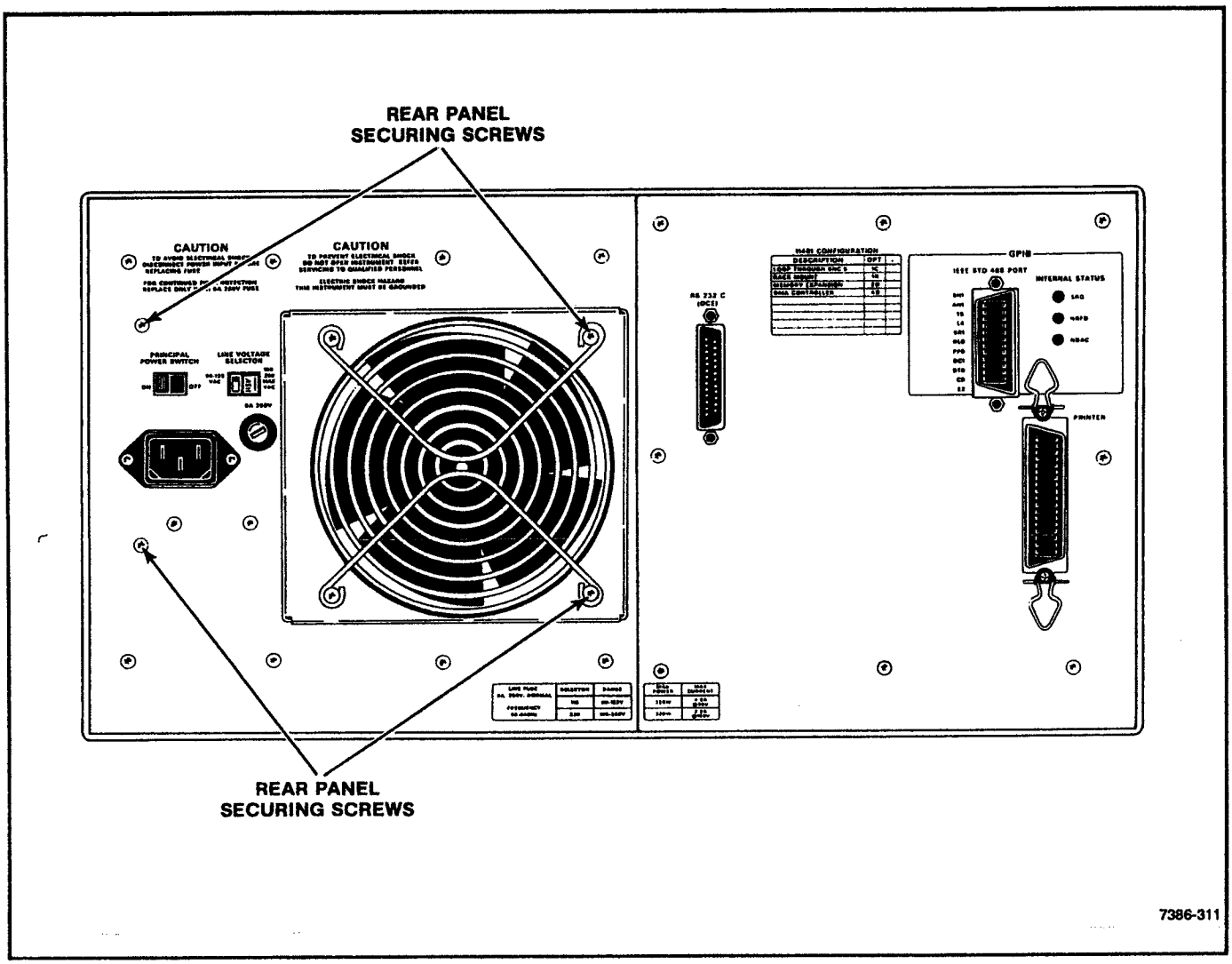


Figure 3-11. Location of rear panel screws securing the power supply cover.

4. Remove the two screws from the left side of the power supply rear plate, above and below the power plug receptacle. (See Fig. 3-11.)

   a. Remove the two long screws from the right side of the fan motor housing. (See Fig. 3-11.)

   b. Remove the overall cover from the power supply unit.

5. Remove connector J70 from the right front edge of the A2A1 Line Inverter board (see Fig. 3-10). Gently separate the A2A1 Line Inverter board from the A2A2 Control Rectifier board. (The interconnecting pins of J97 and J98 hold them together.)

6. Unsolder the (white) wire connecting the A2A1 Line Inverter board to the line filter at the filter.

   - Unsolder the (white) wire connecting the A2A1 Line Inverter to the fuseholder, at the fuseholder.

NOTE

When removing a wire from a circuit board, always tag the wire and the corresponding connection point on the circuit board.

7. Remove the A2A1 Line Inverter board.

8. To replace the A2A1 Line Inverter board, follow the removal procedure in reverse order. Match the index triangle on the pin connectors with the corresponding triangle on the board. Correct location of the pin connectors is shown in the circuit board illustrations.

CAUTION

Check that the circuit board is held in place by its plastic guides on both edges. These guides are mounted inside the overall power supply cover.

A2A2 Control Rectifier Board

Remove and replace the A2A2 Control Rectifier board as follows:

1. Remove the power supply. (See "A2 Power Supply Removal," in this section.)

2. Remove the protective cover from the power supply. (See "Access to Components in the A2 Power Supply," in this section).

3. Follow steps 3 through 4b of the "A2A1 Line Inverter Board Removal" procedure (in this section) to remove the overall cover from the power supply chassis.

4. With reference to Figure 3-12, remove connector J70 connector at the upper front edge of the A2A2 Control Rectifier board. Remove connector J81 (to the fan motor) which is located near the lower right.

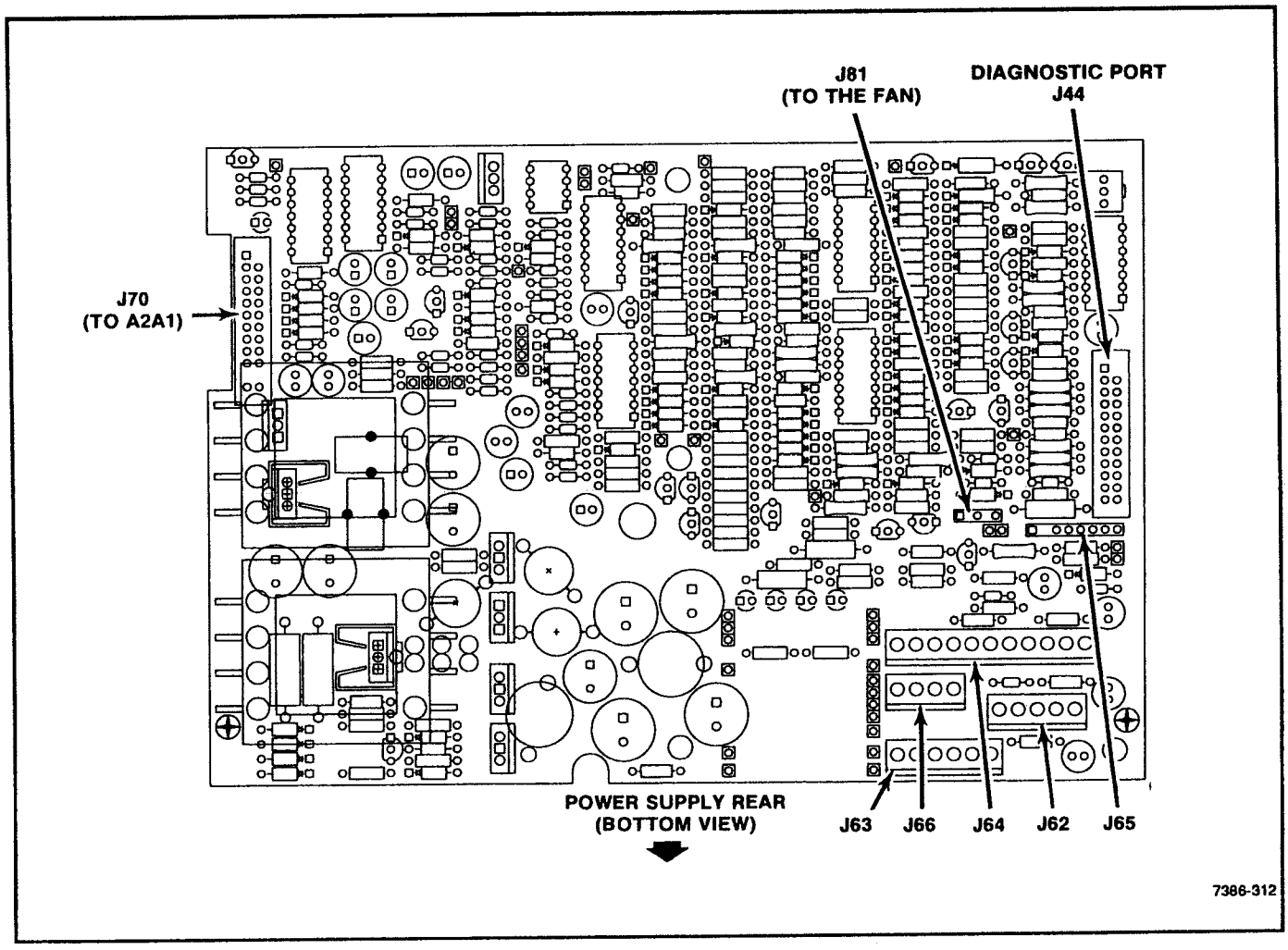


Figure 3-12. Connector locations for removal of the A2A2 Control Rectifier board.

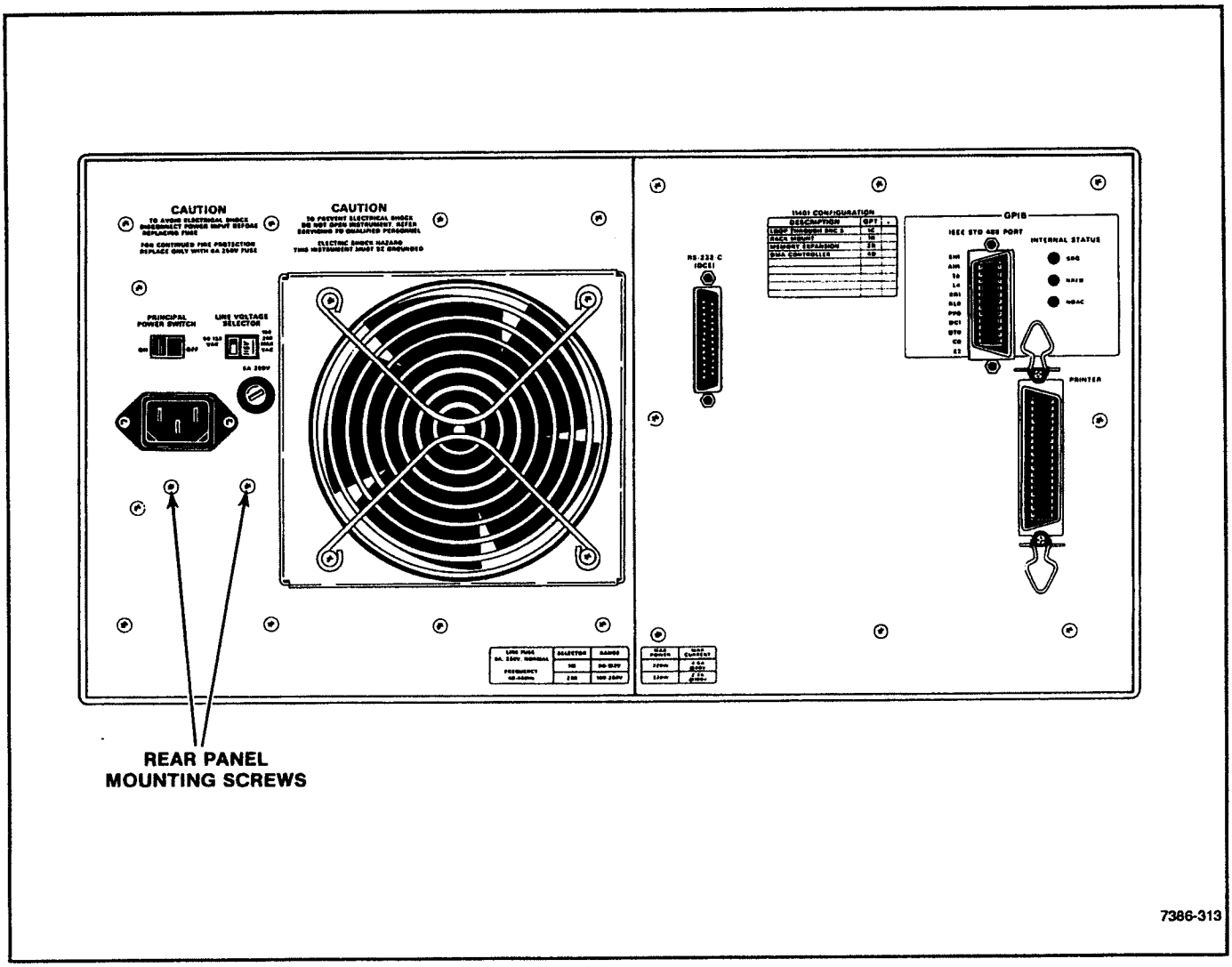


Figure 3-13. Rear panel mounting screws for the A2A2 Control Rectifier board.

5. Remove the two screws below and to the right of the power plug receptacle on the 11401/11402 rear panel (see Fig. 3-13). Gently separate the A2A2 Control Rectifier board from the A2A1 Line Inverter board. The interconnecting pins in J97 and J98, on the A2A1 Line Inverter board, hold both boards together (see Fig. 3-10).
6. Remove the A2A2 Control Rectifier board.

CAUTION

The A2A1 Line Inverter board is now unfastened and requires support so it does not fall and become damaged.

7. To replace the A2A2 Control Rectifier board, follow the removal procedure in reverse order. Match the index triangles on the pin connectors with the corresponding triangles on the board. The correct location of the pin connectors is shown in the circuit board illustrations.

CAUTION

Check that the circuit board is held in place by its plastic guides on both sides. These guides are mounted inside the overall power supply cover.

A4 Regulator Board

Remove and replace the A4 Regulator board as follows:

1. Remove the power supply. (See "A2 Power Supply Removal," in this section.)
2. If it is on its side, set the 11401/11402 right side up.
3. Disconnect connectors J57 and J60 from the A4 Regulator board (see Fig. 3-14). Note the connector index triangle locations for correct reconnection.
4. Remove the two Torx screws from the metal heat-sink attached to the rear of the board (see Fig. 3-14).

NOTE

The A4 Regulator board is now unfastened from the chassis. However, it remains connected to the A1 Plug-In Interface board through interconnecting pins.

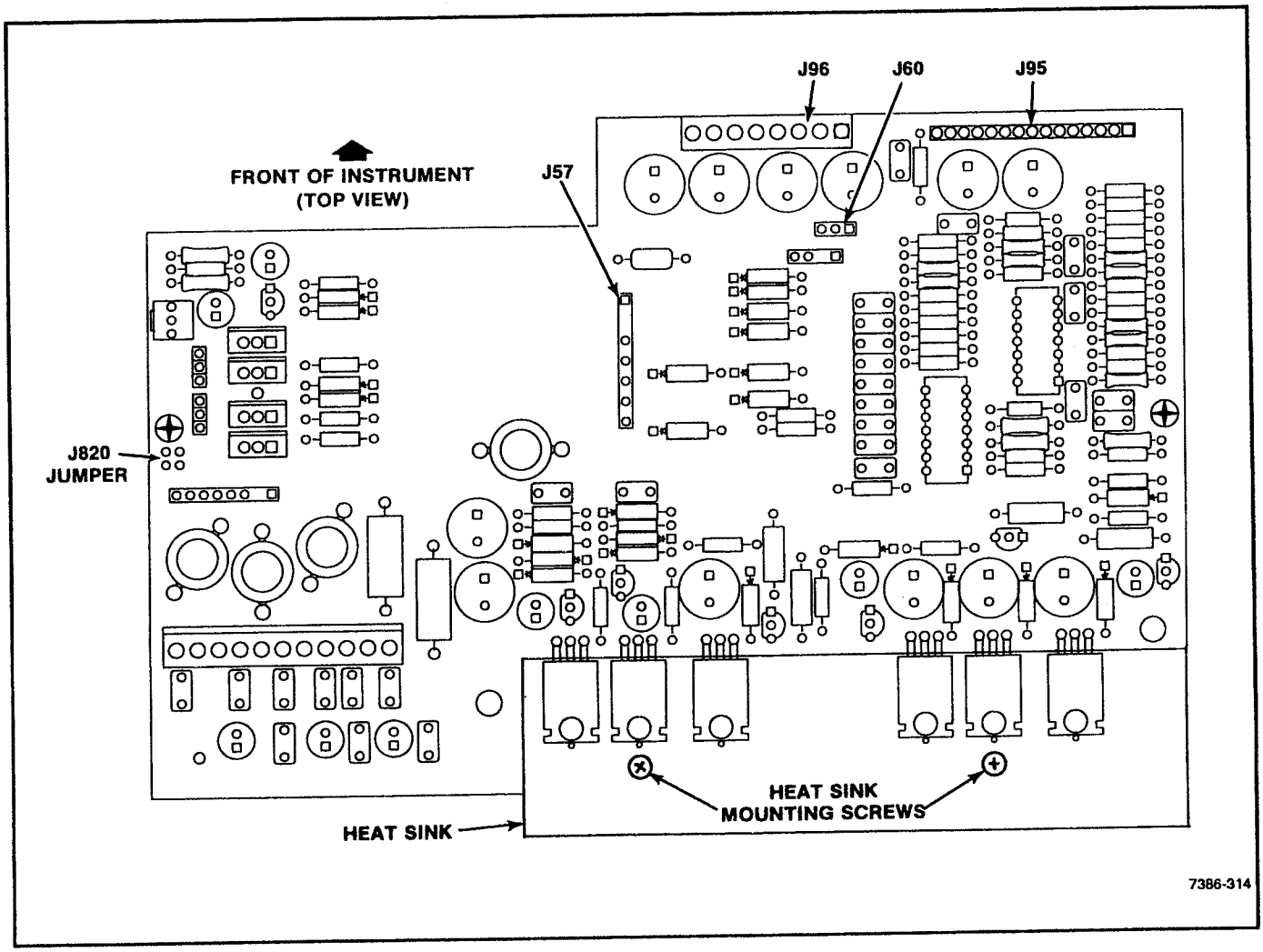


Figure 3-14. Connector locations for removal of the A4 Regulator board.

5. Carefully disconnect the J95 and J96 pins from the A1 Plug-In Interface board by pulling the A4 Regulator board toward the rear.

6. Remove the A4 Regulator board.

7. To replace the A4 Regulator board, follow the removal procedure in reverse order.

CAUTION

Use care when reconnecting the J95 and J96 pins to the A1 Plug-In Interface board. (Should it become necessary, the A5 Acquisition board may be removed to view these pins through the mainframe chassis. See the following A5 Acqquisition board procedure.)

NOTE

Match the index triangle on the pin connectors with the corresponding triangle on the circuit board. Correct locations of the pin connectors are shown in the circuit board illustrations.

A5 Acquisition Board

Remove and replace the A5 Acquisition board as follows:

1. Turn the instrument on its right side (as viewed facing the front). The board is located beneath the card cage and the power supply compartments and beside the A6 Time Base board.

2. Remove the six Torx screws from the board.

   Remove the long Torx screws from the support pivots at the front edges of the board.

3. Move the rear side of the board outward. Position the board so that its outside edge is about perpendicular to the bottom of the oscilloscope. Do not stress the wire bundles.

NOTE

Record the positions of the connectors and the receptacles to aid in their correct reinstallation.

4. Disconnect the Peltola connectors from the center and board edge areas. (See Fig. 3-15.)

5. Disconnect connectors J85, J66B, J91C, J86, J84, and J91A from along the inside edge of the board.

6. Remove the A5 Acquisition board.

7. To replace the A5 Acquisition board, follow the removal procedure in reverse order.

CAUTION

Don't pinch any interconnecting wires underneath the board. Arrange the wires away from the posts to which the Torx screws will be fastened.

NOTE

On an 11402 Digitizing Oscilloscope there is a small circuit board attached to the A5 Acquisition board. This is the A5A1 Strobe board. It is located near the Sampler Hybrid IC and is connected by Berg sockets. The Strobe board slides into place and can be easily removed.

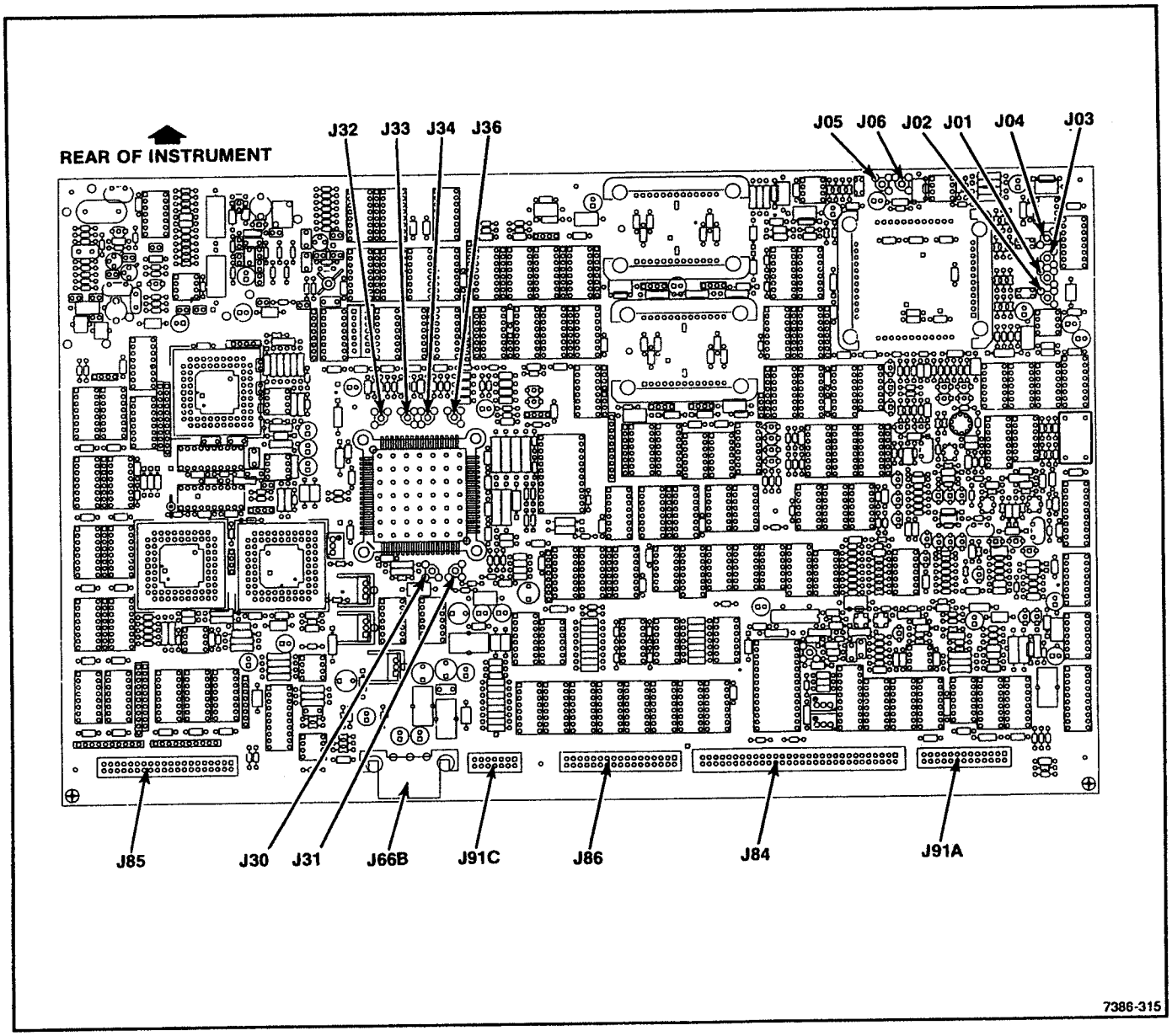


Figure 3-15. Connector locations for removal of the A5 Acquisition board.

A6 Time Base Board

Remove and replace the A6 Time Base board as follows:

1. Turn the instrument on its right side (as viewed facing the front). The board is located next to the Crt bottom and beside the A5 Acquisition board.

   - The A5 Acquisition board overlaps the rear edge of the A6 Time Base board. Remove the board-retaining screws from the A5 Acquisition board. (See the preceding "Circuit Board Removal" procedure.) **Do not remove its connectors.** Move the A5 Acquisition board away from its location to remove the A6 Time Base board.

NOTE

Record the positions of each connector to aid in their correct replacement.

2. Remove connector J09 (see Fig. 3-16).

3. Remove connectors J85, J66A, J83, J86, J84, and J91.

4. Remove the spacer post and five Torx screws from the board.

5. Remove the A6 Time Base board.

6. To replace the A6 Time Base board, follow the removal instructions in reverse order.

CAUTION

The 11401/11402 has static-sensitive components. Be sure to observe all special precautions mentioned under the heading "Static-Sensitive Device Classification".

Don't pinch any interconnecting wires.

A7 Display Controller Board

Remove and Replace the A7 Display Controller board as follows:

1. Remove connectors J57, J53, J52, and J63C from the board (see Fig. 3-17). Note the index triangles on each connector so they can be replaced correctly.

2. Remove the six Torx screws.

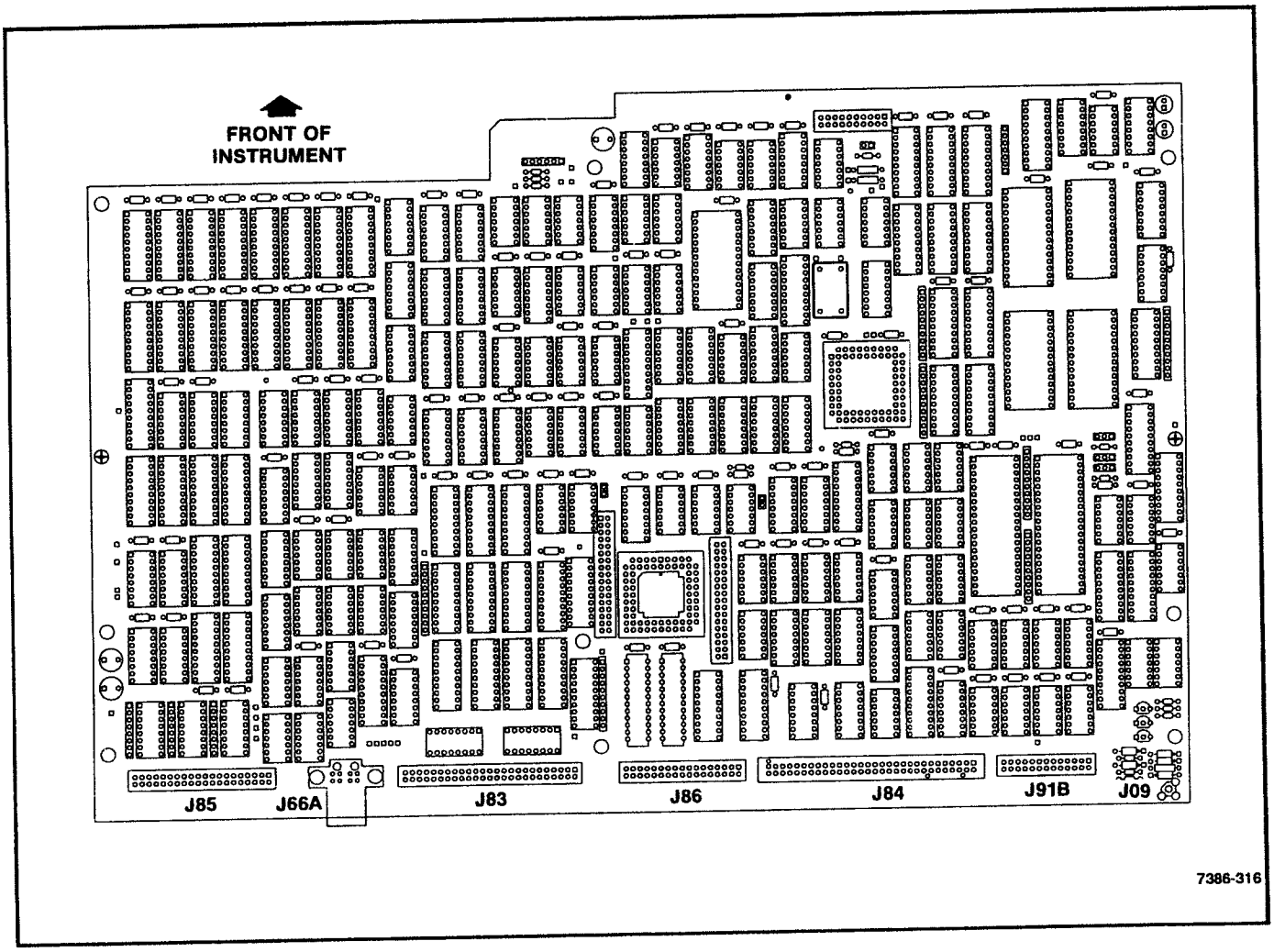


Figure 3-16. Connector locations for removal of the A6 Time Base board.

3. Remove the A7 Display Controller board. Lift and extract the board toward the right side of the 11401/11402 (as viewed facing the front).

   - Notice that the inside edge is held fast by slots in each bottom edge of the circuit board guides. (These guides secure the circuit boards within the card cage compartment.)

4. To replace the A7 Display Controller board, follow the removal procedure in reverse order.

CAUTION

Observe the routing of wires underneath the board. Do not pinch any interconnecting wires when replacing the board.

NOTE

Insert the inner edge of the board back into each of the slots of the board guides. (Be certain the guides are seated correctly on top of the circuit boards in the card cage).

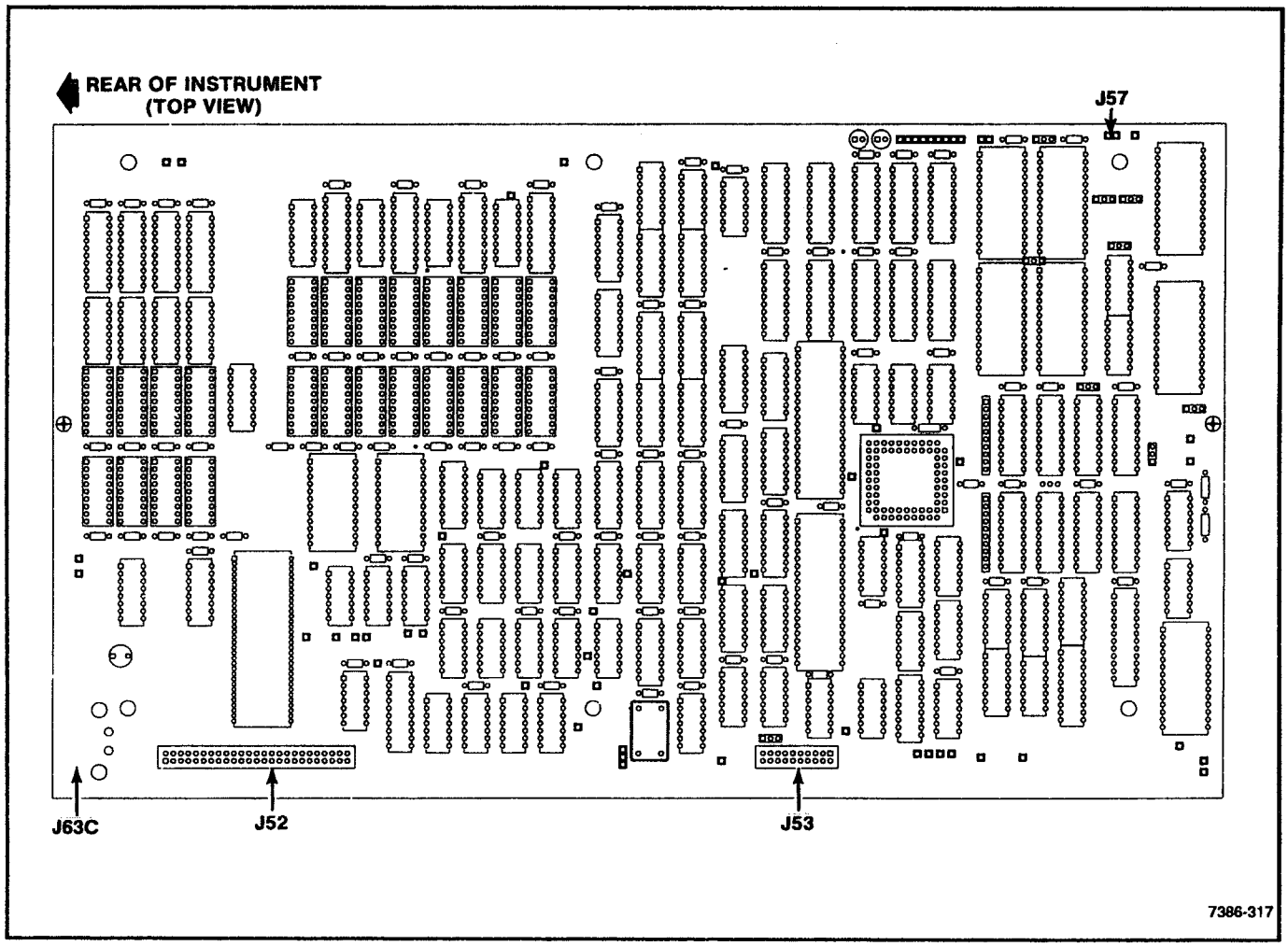


Figure 3-17. Connector locations for removal of the A7 Display Controller board.

A8 CRT Driver Board

Remove and replace the A8 Crt Driver board as follows:

1. Remove the crt shield. (See the "Cathode Ray Tube Removal," in this section.)

2. Remove all connectors from the A8 Crt Driver board (see Fig. 3-18) Note the position of connectors' index triangles so the connectors can be correctly replaced.

3. Remove the Torx screws from the (left side) front and rear decorative trim covers. Do NOT attempt to remove the trim covers until after reading the following CAUTION.

CAUTION

Do not lift the trim covers to remove them. They will break. There is a clip on the inside of the trim cover which slides over the end of the side frame section.

To remove the trim covers, move each cover towards the end of the instrument where it is located. (The front cover moves foward and the rear cover moves backward.) Moving the clip about 1/8-inch will release the cover. Then, the cover can be removed from the instrument.

4. Remove the trim covers.

5. Remove the single Torx screw from the center of the (left side) frame section.

   - Remove the two screws from the ends of the frame section.
   - Remove the frame section.

6. Remove the Torx screws from each corner of the board.

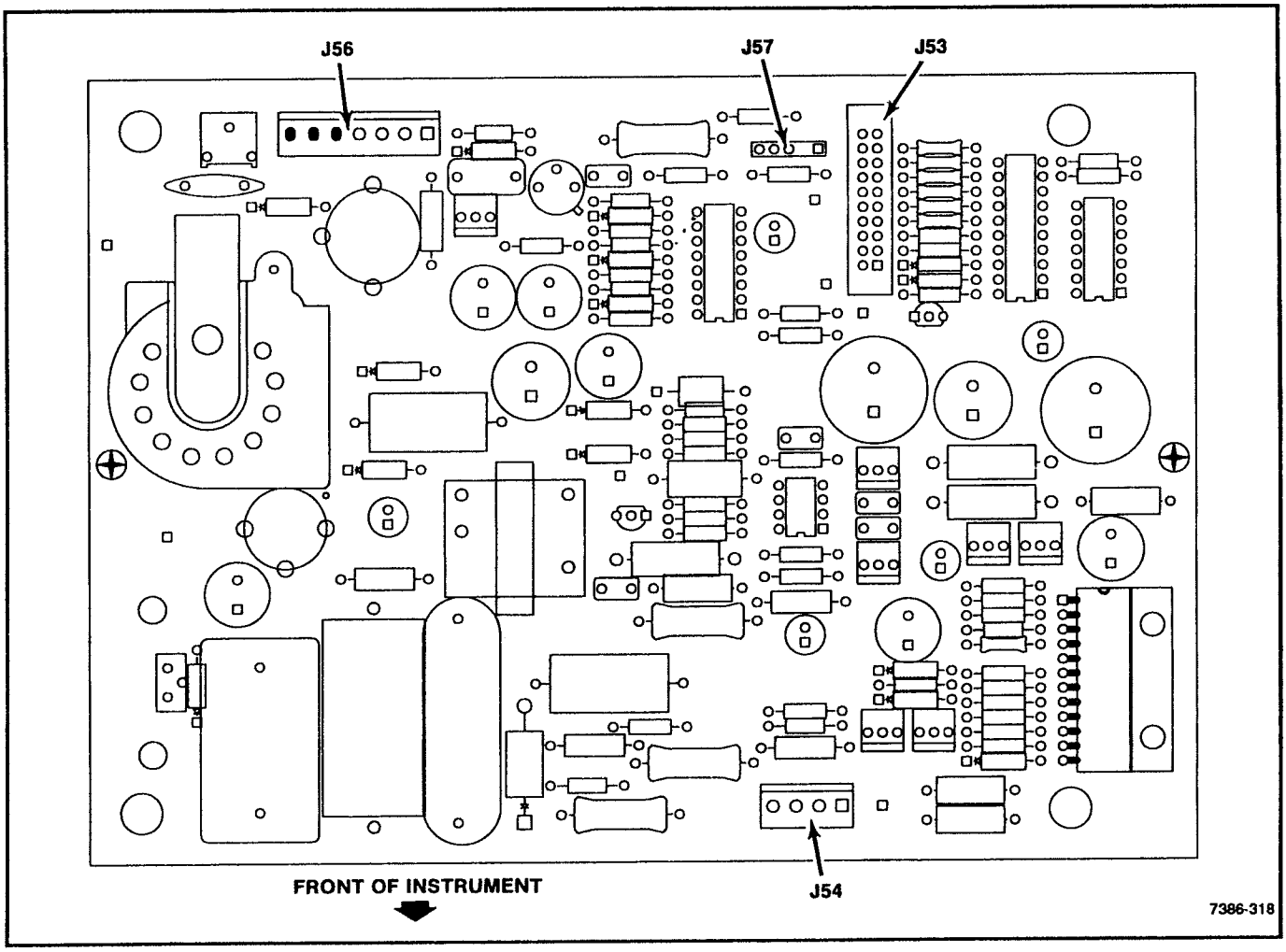


Figure 3-18. Connector locations for removal of the A8 CRT Driver board.

7. Remove the A8 Crt Driver board by sliding it under the neck of the crt and out of the left side of the instrument.

8. To replace the A8 Crt Driver board, follow the removal procedure in reverse order.

A9 Touch Panel Subassembly

Remove and replace the A9 Touch Panel subassembly as follows:

1. Remove the front-panel bezel. (Refer to "Cathode-Ray Tube Removal," in this section.) Begin with step 6 and proceed through step 9.

NOTE

The wire cable from P73 on the A10 Front Panel Control board is removed with the A9 Touch Panel subassembly. (See Fig. 3-19.) Disconnect connector P73 from the A10 Front Panel Control board. Note the position of connector index triangles in order to correctly replace the connector. Carefully remove the wire cable through the slot provided in the front casting.

Protect the front of the bezel after it is removed. Since the plastic exterior may scratch, cover it with protective material.

2. To replace the A9 Touch Panel subassembly, route the P73 wire cable back through the slot in the chassis. Attach the connector to the board after checking the index triangles for correct orientation.

3. Replace the bezel. (Refer to "Cathode-Ray Tube Replacement," in this section.) Use steps 4 through 7.

NOTE

Feed any slack cable from P73 to inside the chassis (near the A10 Front Panel Control board). Be careful not to pinch the interconnecting cable when the bezel is replaced.

A10 Front Panel Control Board

Remove and replace the A10 Front Panel Control board as follows:

1. Remove the crt shield. (See "Cathode-Ray Tube Removal," in this section.) Follow steps 1 and 2.

2. Remove connector J53 from the A7 Display Controller board. (See Fig. 3-17.)

3. Remove connectors P72, P75, P73 and P74. (See Fig. 3-19.) Note the position of their index triangles for correct replacement.

4. Remove the two Torx screws at the upper edge of the board. (See Fig. 3-19.)

5. Lift the board away from the guides at its bottom and remove it.

6. To replace the A10 Front Panel Control board, follow the removal procedure in reverse order.

A11 Front Panel Button Board

Remove and replace the A11 Front Panel Button board as follows:

1. Remove the A7 Display Controller board. (See "A7 Display Controller Board," in this section.)
2. Remove the crt shield. (See "Cathode-Ray Tube Removal," in this section.) Follow steps 1 and 2.
3. Remove connector P75 from the A10 Front Panel Control board (see Fig. 3-19). Note the position of connector index triangles for correct replacement.
4. Remove the two Torx screws from the A11 Front Panel Button board, which is located at the top and near the inside center of the front casting.
5. Remove the A11 Front Panel Button board.
6. To replace the A11 Front Panel Button board, follow the removal procedure in reverse order.

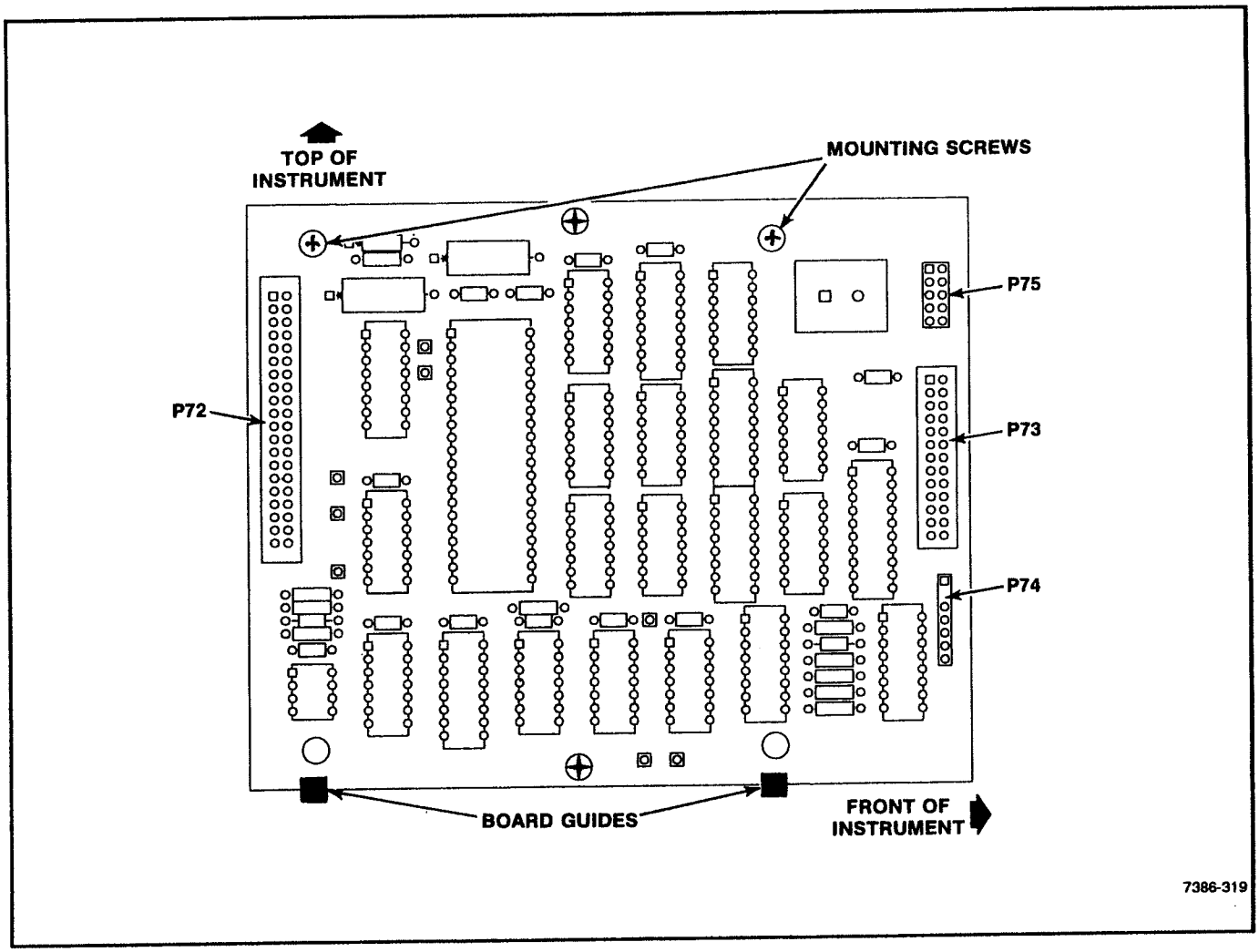


Figure 3-19. Connector locations for removal of the A10 Front Panel Control board.

Remove and replace the A12 Rear Panel board as follows:

1. Remove the eight Torx screws from the outer edges of the rear panel plate. (See Fig. 3-20.)

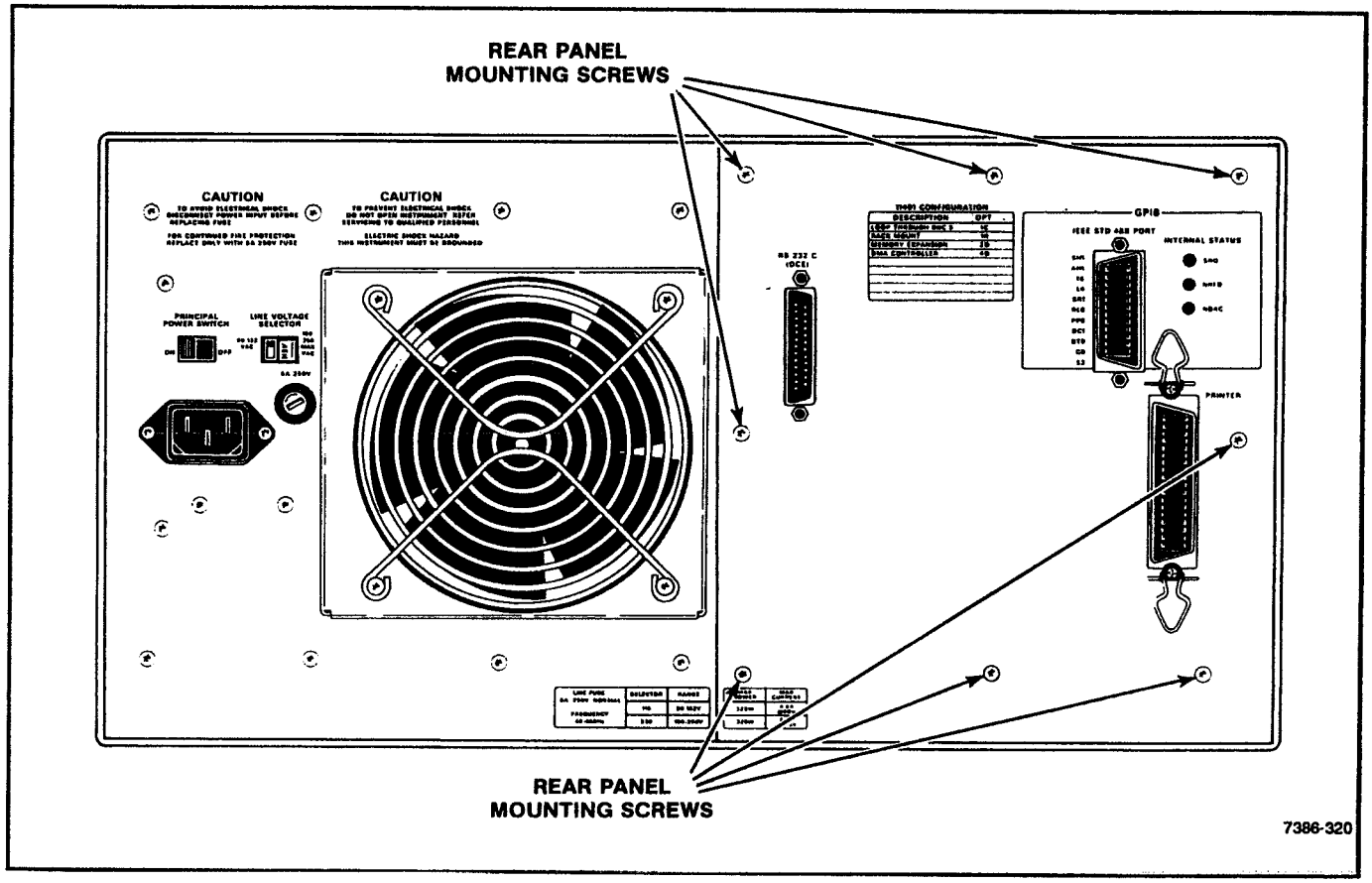


Figure 3-20. Location of rear panel mounting screws.

2. Tilt the plate back from the mainframe. Remove connector J78 from the top of the A12 Rear Panel board. Note the position of the connector index triangles for correct replacement.

   - Remove the connectors from the RS-232-C, GPIB, and PRINTER connector holders. (See Fig. 3-21.)

3. Remove the rear panel plate and its attached A12 Rear Panel board.

4. Remove the following items from the rear panel plate:

   - Two bail brackets, screws, and washers from the PRINTER connector,
   - Two posts from the GPIB connector,
   - Posts, lockwashers, and flat washers from the RS-232-C connector(s),
   - Torx screw and washer (at lower left, if present).

5. Remove the A12 Rear Panel board from the plate.

CAUTION

The metal covers on the PRINTER and on the GPIB connectors are loose. If the board is inverted, they will fall off.

6. To replace the A12 Rear Panel board, follow the removal procedure in reverse order.

NOTE

Replacement of connector J78 will be easier if the connector is replaced before the plate is reinstalled.

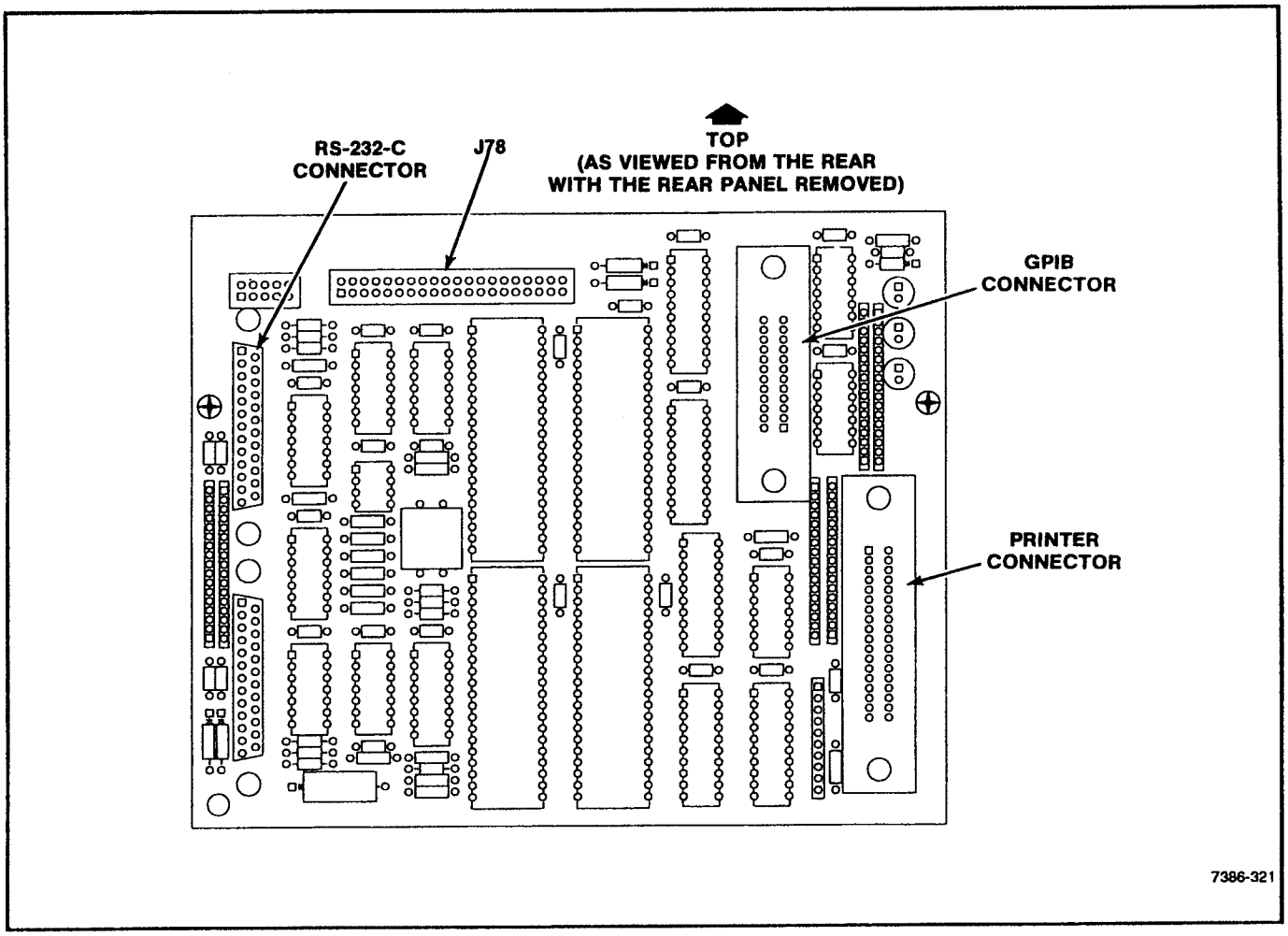


Figure 3-21. Connector locations for removal of the A12 Rear Panel board.

A13 Mother Board

Remove and replace the A13 Mother board as follows:

1. Remove both circuit board guides from atop the card cage. The guides are retained by two small catches, located in two holes in the left side bracket of the card cage. The other ends of the guides contain slots which attach to the edge of the A7 Display Controller board. Both ends of the guides can be pried loose for removal.

2. Remove the A14 I/O, A15 MMU, A16 Waveform Compressor, A17 Main Processor, and A18 Memory Circuit boards. (See "Plug-On Boards," in this section.)

NOTE

Tag the interconnecting plugs and mark the board locations so that they can be replaced correctly.

3. Remove connector J63B from the A13 Mother board.

4. Remove the six Torx screws.

5. Remove the A13 Mother board.

6. To replace the A13 Mother board, follow the removal procedure in reverse order.

NOTE

Don't pinch the wires along the inside edge of the board when replacing it.

Plug-On Boards

All circuit boards inside the card cage plug onto the A13 Mother board. Feed-through connectors join the plug-on boards to the Mother board. Figure 3-22 shows the location of each board within the card cage.

NOTE

An extender card is available for troubleshooting the card cage circuit boards. For a brief description and the Tektronix Part Number of this troubleshooting aid, see "Service Kits" near the beginning of this section.

A14 Input/Output (I/O) Board

Remove and replace the A14 I/O board as follows:

1. Remove both circuit board guides from atop the card cage. These guides are retained by two small catches located in two holes in the left bracket of the card cage. The other ends of the guides contain slots which attach to the edge of the A7 Display Controller board. Both ends of the guides can be pried loose for removal.

2. Remove connectors J78, J77, J90 and J72. Note the position of connector triangles for correct replacement. (See Fig. 3-22 for the location of the A14 Input/Output board in the card cage and Fig. 3-23 for connector locations.)

3. Lift the white, hinged tab at the upper front edge of the board. Pull the tab upward until the A14 I/O board separates from the A13 Mother board.

4. Remove the A14 I/O board.

WARNING

A lithium battery (BT130) is mounted on the A14 I/O board. This battery requires special handling for disposal. Read the instructions on "Lithium Battery Handling, Disposal, and First Aid," in this section.

5. Replace the A14 I/O board by following the removal procedure in reverse order.

NOTE

Insert the board edges into the plastic guides at each end of the card cage. Lower the board into position.

Check that connector P105 is seated on the A13 Mother board connector (see Fig. 3-22). Push down firmly on the A14 I/O board to connect it.

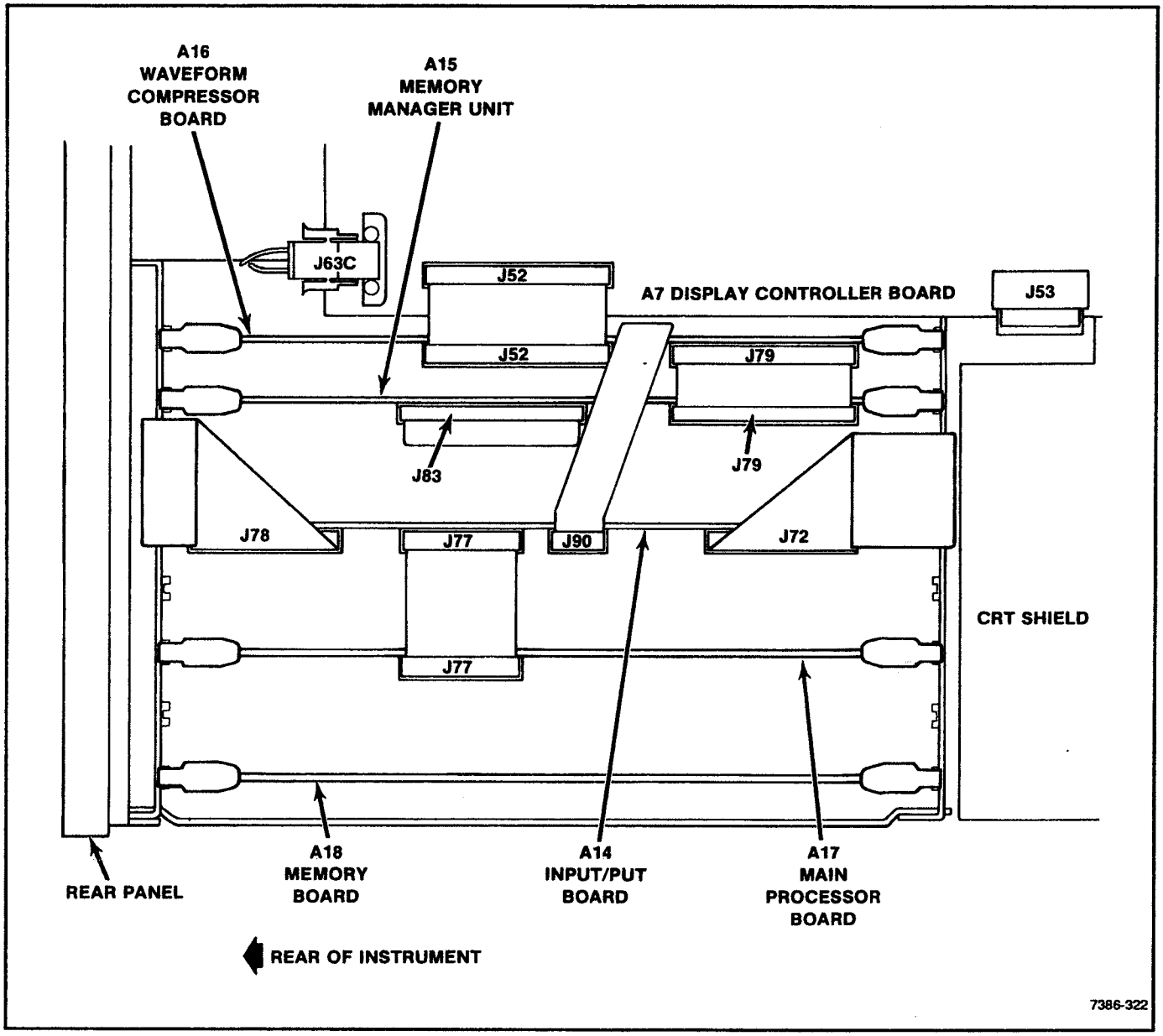


Figure 3-22. Top view of the card cage showing circuit board locations.

A15 Memory Manager Unit (MMU) Board

Remove and replace the A15 MMU board as follows:

1. Remove both circuit board guides from atop the card cage. The guides are retained by two small catches located in two holes in the left bracket of the card cage. The other ends of the guides contain slots which attach to the edge of the A7 Display Controller board. Both ends of the guides can be pried loose for removal.

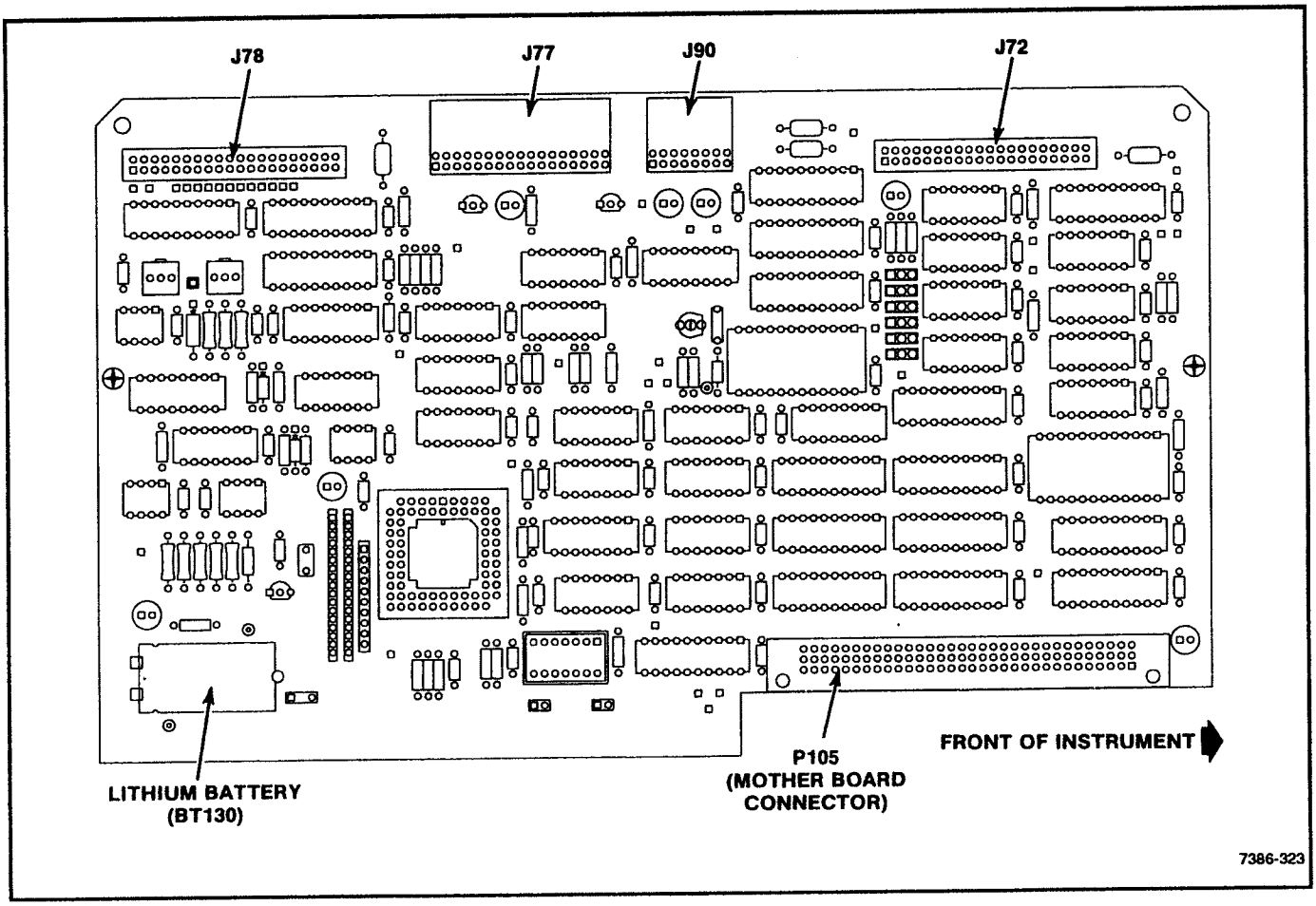


Figure 3-23. Connector locations for removal of the A14 Input/Output board.

2. Remove connectors J83 and J79 (see Fig. 3-22 and 3-24). Note the position of connector index triangles for correct replacement.

3. Remove J90 from the A14 I/O board.

4. Lift the white, hinged tabs at the front and rear edges of the board. Pull the tabs upward until the A15 MMU board separates from the A13 Mother board.

5. Remove the A15 MMU board.

6. Replace the A15 MMU board by following the removal procedure in reverse order.

NOTE

Insert the board edges into the plastic guides at each end of the card cage. Lower the board into position.

Check that connector P101 is seated onto the A13 Mother board connector. Push down firmly on the A14 MMU board to connect it.

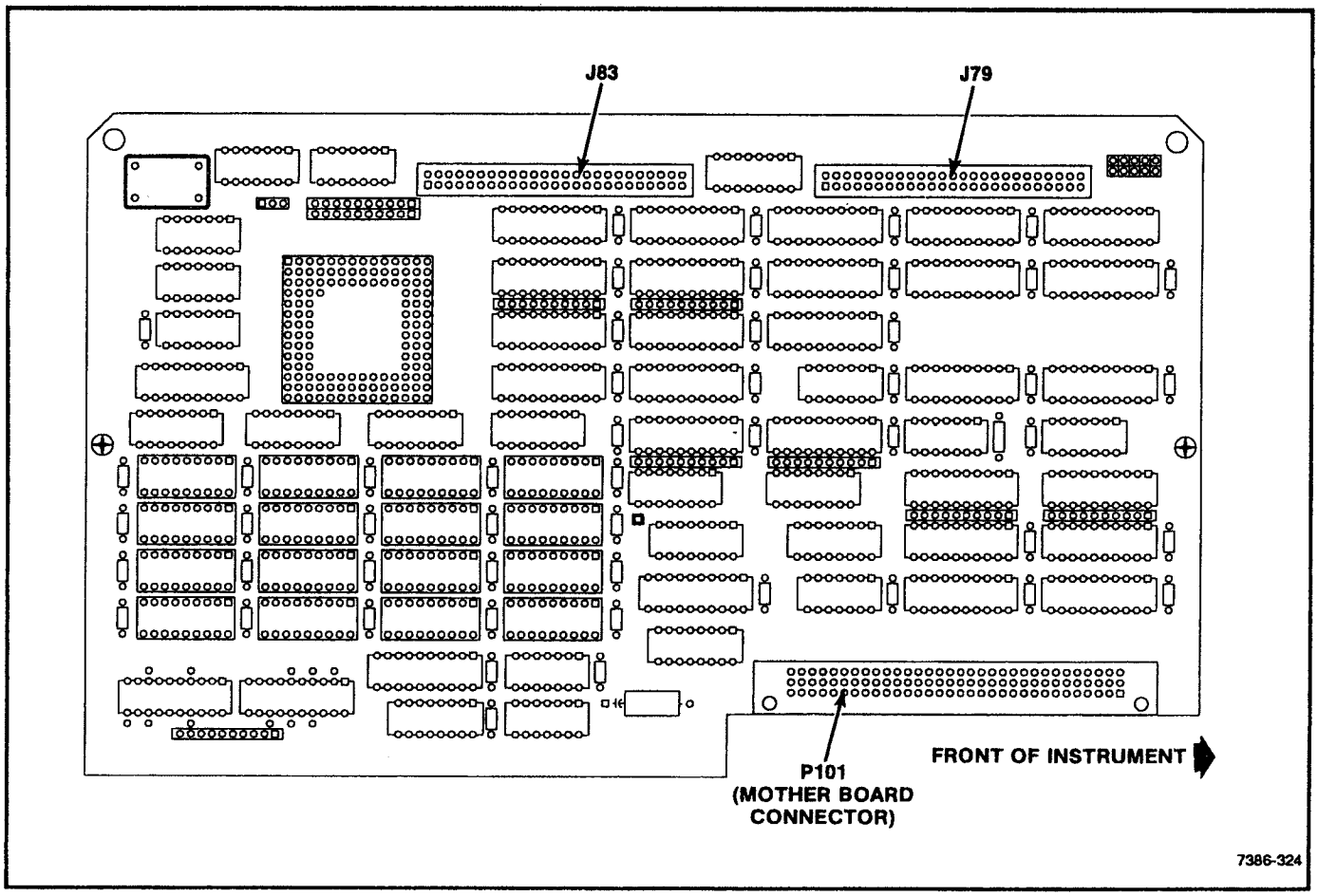


Figure 3-24. Connector locations for removal of the A15 Memory Manager Unit board.

A16 Waveform Compressor Board

Remove and replace the A16 Waveform Compressor board as follows:

1. Remove both circuit board guides from atop the card cage. The guides are retained by two small catches located in two holes of the left bracket of the card cage. The other ends of the guides contain slots which attach to the edge of the A7 Display Controller board. Both ends of the guides can be pried loose for removal.

2. Remove connectors J52 and J79. (See Fig. 3-22 and 3-25). Note the position of connector index triangles for correct connector replacement.

3. Remove J90 from the A14 I/O board. (See Fig. 3-23.)

4. Lift the white, hinged tabs at the front and rear edges of the board. Pull the tabs upward until the A16 Waveform Compressor board separates from the A13 Mother board.

5. Remove the A16 Waveform Compressor board.

6. Replace the A16 Waveform Compressor board by following the removal procedure in reverse order.

NOTE

Insert the board edges into the plastic guides at each end of the card cage. Lower the board into position.

Check that connector P100 is seated onto the A13 Mother board connector. Push down firmly on the A16 Waveform Compressor board to connect it.

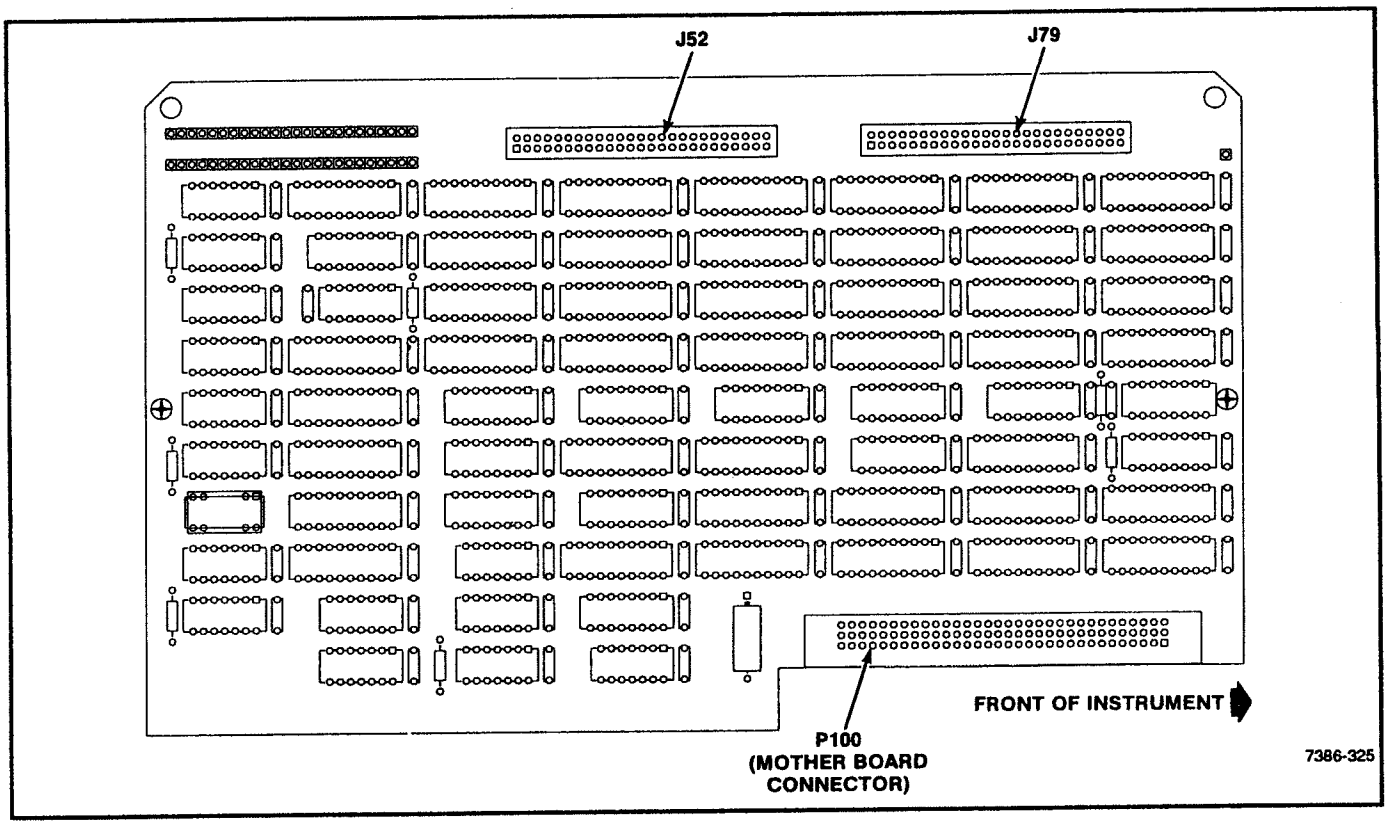


Figure 3-25. Connector locations for removal of the A16 Waveform Compressor board.

A17 Main Processor Board

Remove and replace the A17 Main Processor board as follows:

1. Remove both circuit board guides from atop the card cage. The guides are retained by two small catches located in two holes in the left bracket of the card cage. The other ends of the guides contain slots which attach to the edge of the A7 Display Controller board. Both guide ends can be pried loose for removal.

2. Remove connector J77. (See Fig. 3-22 and 3-26).

3. Lift the white, hinged tabs at the front and rear edges of the board. Pull the tabs upward until the A17 Main Processor board separates from the A13 Mother board.

4. Remove the A17 Main Processor board.

WARNING

A lithium battery (BT160) is mounted on the A17 Main Processor board. ***The battery requires special handling for disposal.*** *Read the instructions on the "Lithium Battery Handling, Disposal, and First Aid," in this section.*

5. Replace the A17 Main Processor board by following the removal procedure in reverse order.

NOTE

Insert the board edges into the plastic guides at each end of the card cage. Lower the board into position.

Check that connector P104 is seated onto the A13 Mother board connector. Push down firmly on the A17 Main Processor board to connect it.

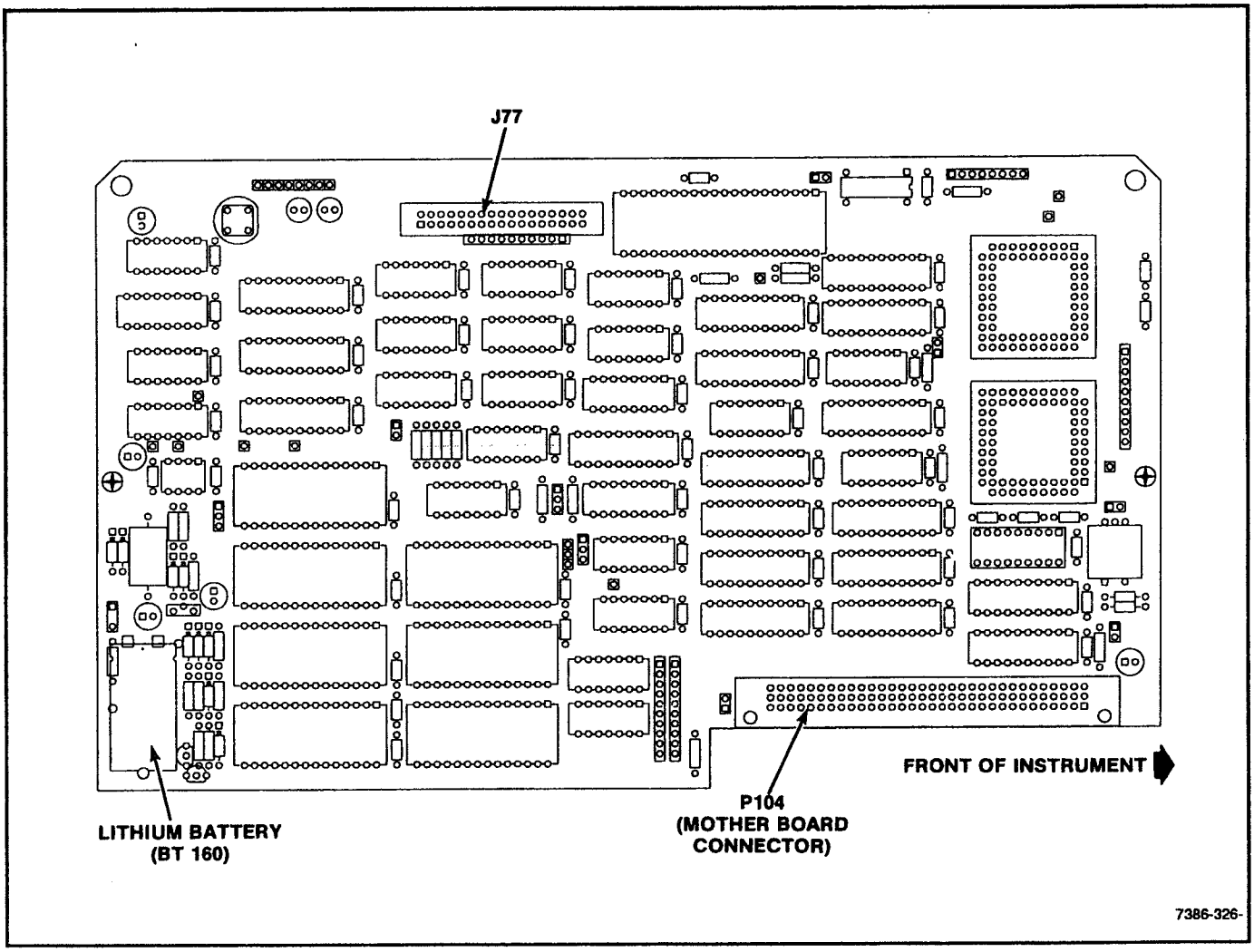


Figure 3-26. Connector locations for removal of the A17 Main Processor Board

A18 Memory Board

Remove and replace the A18 Memory board as follows:

1. Remove both circuit board guides from atop the card cage. The guides are retained by two small catches located in two holes in the left bracket of the card cage. The other ends of the guides contain slots which attach to the edge of the A7 Display Controller board. Both ends of the guides can be pried loose for removal.

2. Lift the white, hinged tabs at the front and rear edges of the board. Pull the tabs upward until the A18 Memory board separates from the A13 Mother board. (See Fig. 3-22 for the location of A18 within the card cage).

3. Remove the A18 Memory board.

4. Replace the A18 Memory board by following the removal procedure in reverse order.

NOTE

Insert the edges of the board into the plastic guides at each end of the card cage. Lower the board into position.

Check that connector P106 is seated onto the A13 Mother board connector. Push down firmly on the A18 Memory board to connect it.

Front-Panel Bezel Removal

(See the A9 Touch Panel Subassembly Removal)

S74 and S75 Encoder Removal

To remove and replace the S74 and S75 Encoder, proceed as follows:

1. Remove the front-panel bezel and the crt. (Refer to "Cathode Ray Tube Removal," in this section.)
2. Remove the plastic shaft extender by pulling it off.
3. Remove the 7/16" nut and the washer.
4. Remove the Encoder(s) from the casting by pulling toward the interior of the instrument. (Notice the way its key fits into the casting keyway.)
5. Unsolder the wires from the Encoder(s).

NOTE

When removing wires from a component, record their locations for replacement.

6. Remove the Encoder(s).
   - Connector P74 can be disconnected (from the A10 Front-Panel Control board) if both Encoders are removed.
7. To replace the S74 or S75 Encoder, follow the removal procedure in reverse order. For the crt, refer to the "Cathode-Ray Tube Replacement" procedure.

NOTE

Be certain that the wires are correctly replaced on the Encoder(s).

Align the key on the Encoder(s) with the keyway(s) in the front casting before tightening the nut.

If it was removed, replace connector P74 on the A10 Front-Panel Control board.

Rear Panel Removal

To remove the rear panel, proceed as follows:

1. Remove the rear panel plate and board. (Refer to the A12 Rear Panel board removal procedure, in this section). Follow steps 1 through 5.

   - The A12 Rear Panel board is now accessible for troubleshooting and/or repair.

   CAUTION

   The metal covers on the Printer and on the GPIB connectors are loose. They will fall off if the board is inverted.

2. To replace the rear panel, follow the removal procedure in reverse order.

Semiconductors

Semiconductors should not be replaced unless actually defective. If removed from their sockets during routine maintenance, return them to their original sockets. Unnecessary replacement of two facing pages semiconductors may affect the adjustment of the instrument. When semiconductors are replaced, check the operation of circuits that may be affected.

WARNING

To avoid an electric-shock hazard, always disconnect the instrument from its power source before removing or replacing components.

Replacement semiconductors should be of the original type or a direct replacement. (Refer to Figure 3-27 for semiconductor indexing and lead configurations.) If a replacement transistor is made by a different manufacturer than the original, check the manufacturer's basing diagram for correct basing. When removing soldered-on transistors or integrated circuits, use an anti-static vacuum solder extractor (see the Soldering Techniques procedure in this section) to remove the solder from the circuit board pads. Transistors which have heat radiators or are mounted on the chassis use silicone grease to increase heat transfer. Replace the silicone grease on both sides of the insulating washer when replacing these transistors.

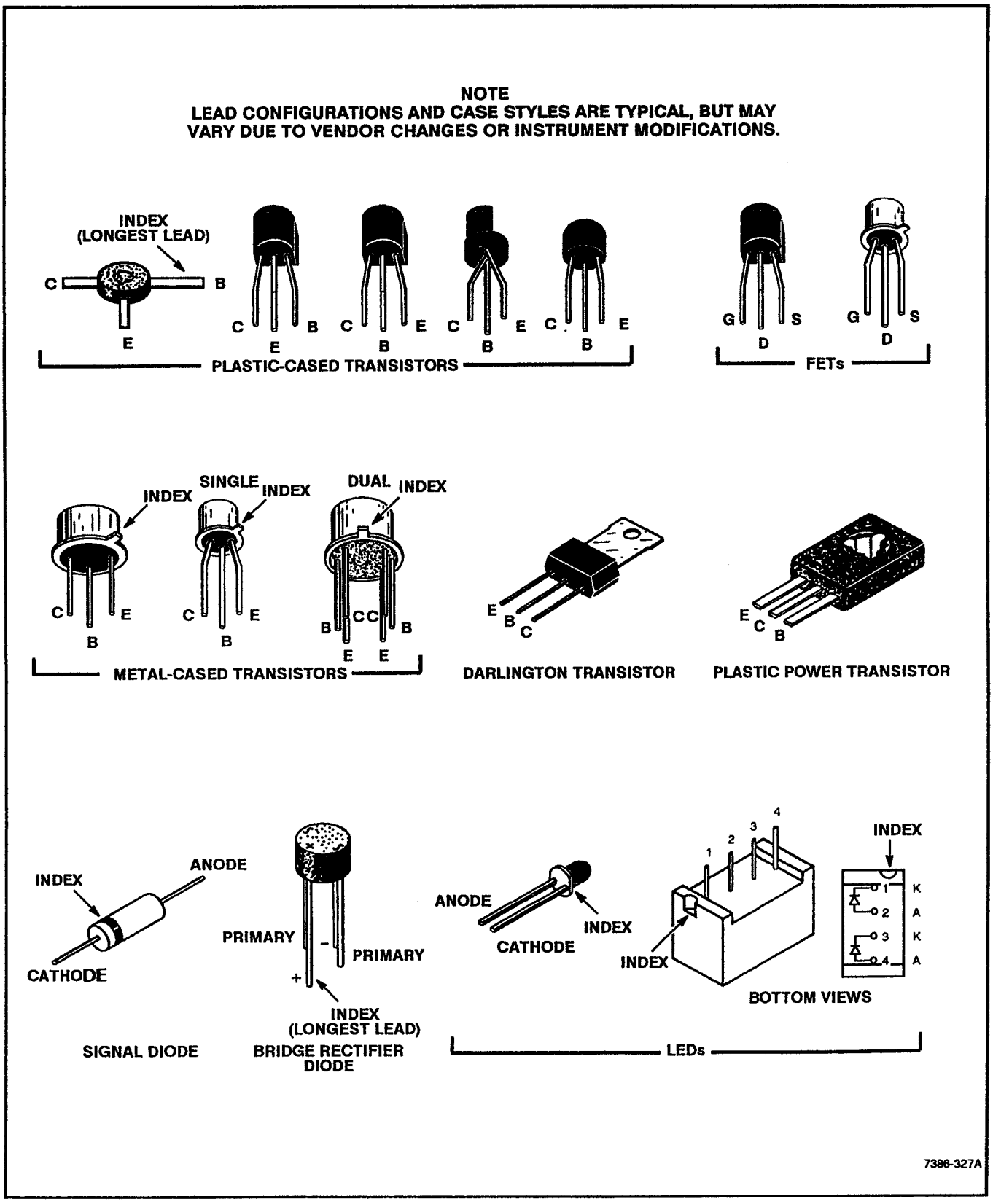


Figure 3-27. Semiconductor Lead Chart.

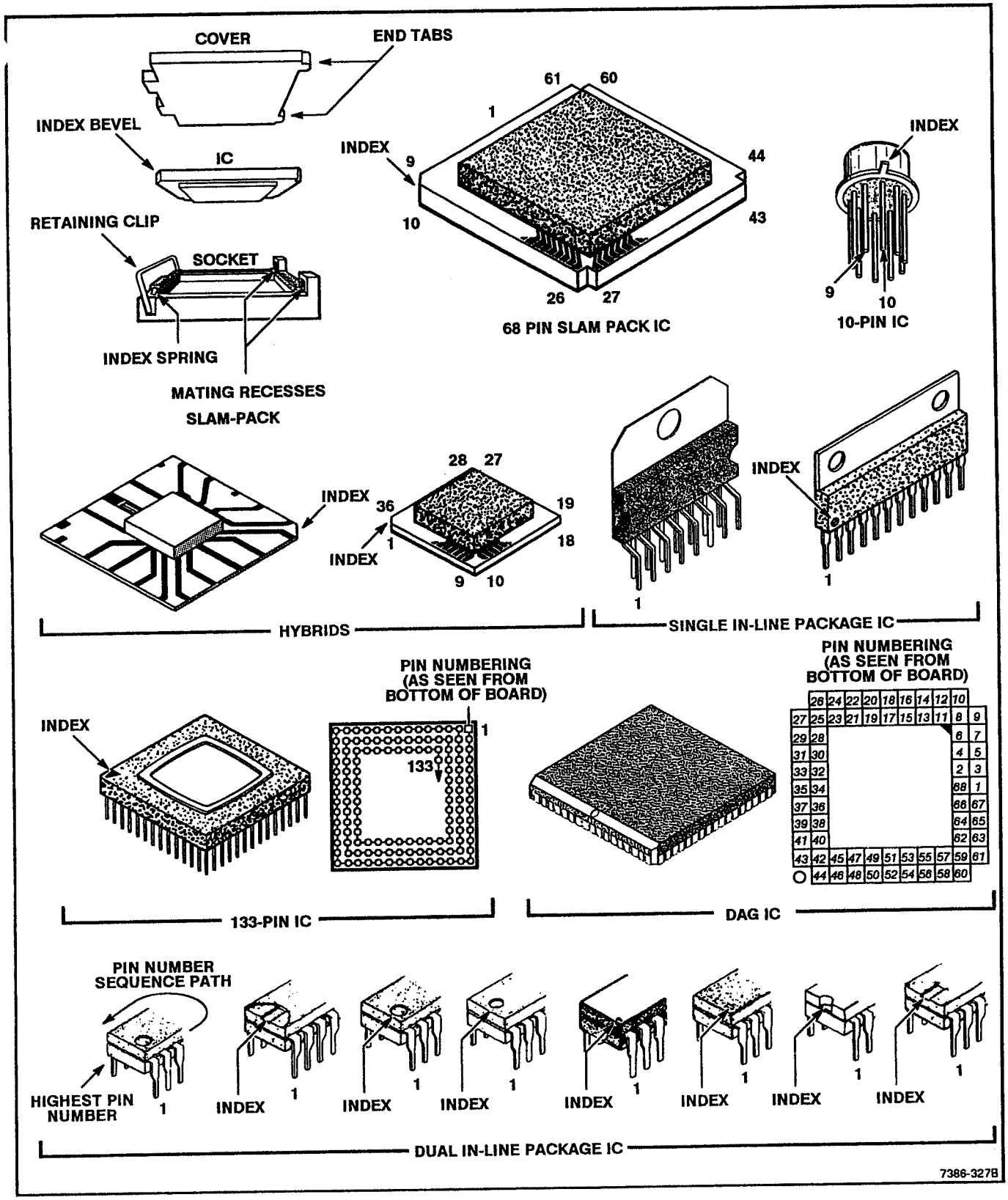


Figure 2-27 (cont). Semiconductor Lead Chart.

WARNING

Handle silicone grease with care. Avoid getting the silicone grease in your eyes. Wash hands thoroughly after use.

Most integrated circuit pin sockets are indexed by a square-shaped socket for the number 1 pin. The reverse view of a circuit board shows the pins are numbered in a clockwise direction starting at the index. (An example of a pin number sequence is shown in the Semiconductor Lead Chart, Figure 3-27.)

To replace one of the power transistors mounted on the metal heat-sink at the rear of the power supply (A4 Regulator board), first remove the board. (See the A4 Regulator Board Removal procedure in this section.) Then unsolder the transistor leads, remove the mounting screw from its heat radiator, and remove the transistor and its insulating washer. When replacing the transistor, be sure to reinstall the insulating washer between the transistor and the heat sink (use silicone grease as previously described). Reinstall the mounting screw and tighten it just enough to hold the transistor in place. Then solder the replacement transistor to the A4 Regulator board. Replace the board by following the removal procedure in reverse order.

An extracting tool should be used to remove in-line integrated circuits to prevent damaging the pins. This tool is available from Tektronix, Inc.; see Table 3-2 for part number If an extracting tool is not available, use care to avoid damaging the pins. Pull slowly and evenly on both ends of the integrated circuit (IC). Try to avoid disengaging one end from the socket before the other end. **(Most ICs will be soldered into the circuit board, rather than in sockets).**

CAUTION

Do not remove stickers affixed to the top of EPROMs. Removing such stickers will admit light into the chip, and may cause partial erasure of its data.

Hypcon Connectors

The Hypcon (hybrid-printed connector) is a precision-made connector designed to provide low loss electrical and thermally-efficient connection between the printed circuit board and hybrid integrated circuits. An exploded view of the Hypcon connector is shown in Figure 3-28. When replacing the hybrid IC's, be careful not to touch the elastomer gold-plated contacts with your fingers, or to use a cleaner which will degrade contact reliability. **If it becomes necessary to use a cleaning solvent near the connector when replacing adjacent (within 1/2") circuit board components, the Hypcon connector and hybrid IC should be removed.**

IMPORTANT: Remove all traces of solder flux or foreign material contamination from the circuit board contact area before replacing the connector. **Contamination usually takes place during the soldering and cleaning process.** Even when the soldering is done carefully, flux, oil, or other contaminants can be carried under the connector during the cleaning operation. When the solvent evaporates, nonconductive contaminants may remain on or near the contact interfaces.

The cleaning process (either hand cleaning with a solvent or machine cleaning in an automatic detergent wash) is not recommended for boards containing Hypcon connectors.

If a component near a Hypcon connector must be replaced, the following steps are recommended:

1. Remove the hybrid IC and Hypcon connector (see Disassembly and Removal Instructions) before any soldering or cleaning and store in a dirt-free covered container. When several hybrids and Hypcon connectors are to be removed, keep parts together and replace as sets; do not interchange parts.

2. Hand Soldering:

   a. Use small diameter solder (0.030 inch-0.040 inch).

   b. Use low wattage soldering irons (15-20 watts).

   c. Use care with solder and placement.

3. Remove solder flux and contact contamination with isopropyl alcohol, denatured ethyl alcohol, or a FREON TF cleaner such as SPRAY-ON #2002.

4. Flush the hybrid and Hypcon connector mounting area with isopropyl alcohol. Do not scrub with a cotton-tipped applicator, as cotton fibers will adhere to edges and surfaces of contact area and cause open or intermittent connections. The elastomer should be examined under light for dust, hair, etc., before it is reinstalled. If the etched circuit board surfaces require more cleaning, scrub with a soft rubber eraser and blow or vacuum clean while dusting the surface with a small clean brush.

5. If the hybrid IC and elastomer contact holder are contaminated, clean by flushing or spraying with alcohol and oven dry at 50° C. Do not scrub with a cotton-tipped applicator or similar device. If the contact holder is excessively contaminated, replace it with a new one.

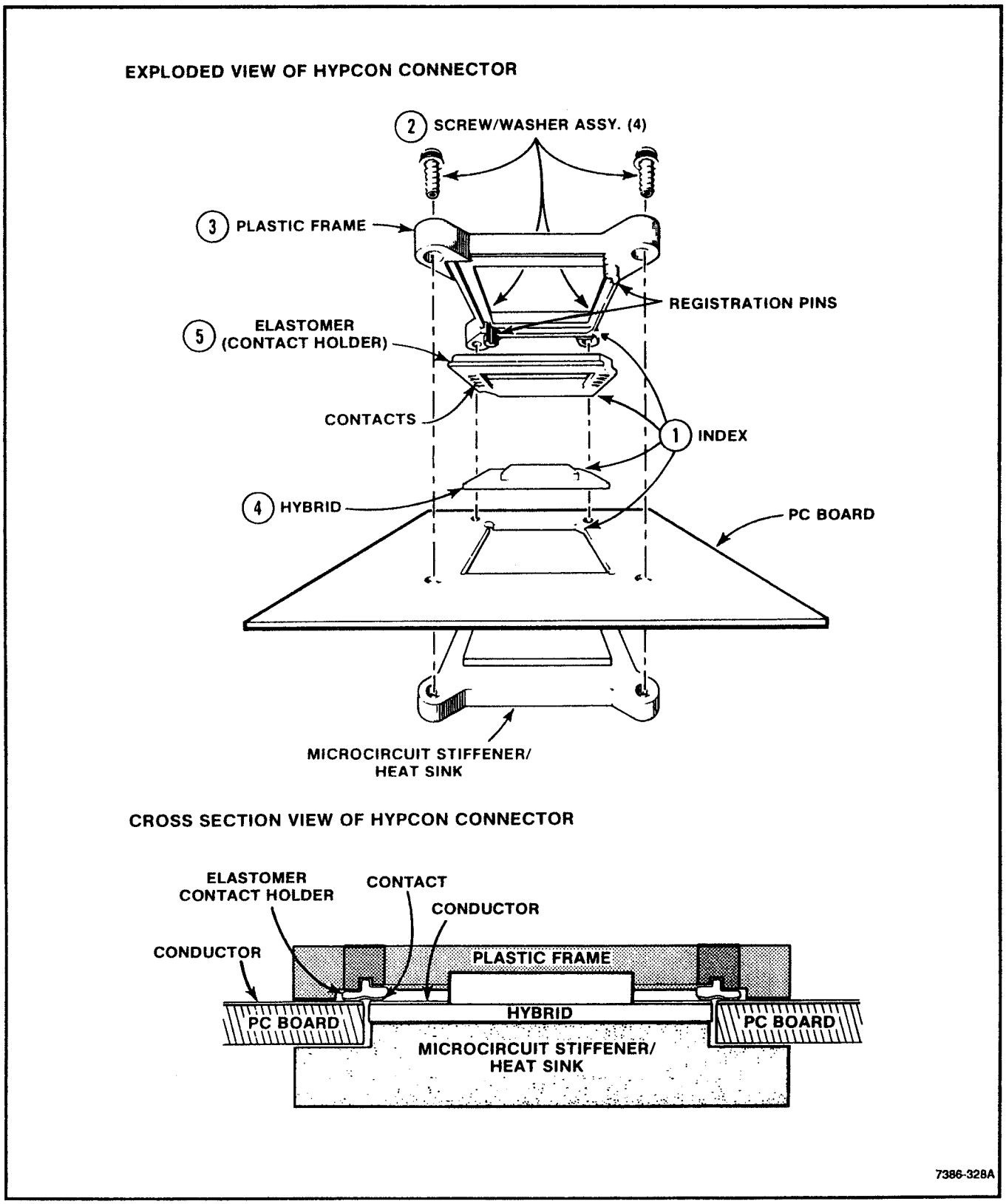


Figure 3-28. Hypcon assembly removal and replacement.

DISASSEMBLY AND REMOVAL

1. Notice index on circuit board (arrow) and plastic frame (pointed tab).
2. Unscrew and remove the 4 screw/washer assemblies.
3. Lift plastic frame from board.
4. Notice index location of hybrid and remove from circuit board with tweezers.

NOTE

Step 5 describes the removal of the elastomer from the plastic frame. This step is not necessary when replacing only the hybrid.

5. Notice index location of elastomer contact holder and remove from plastic frame by grasping corner with tweezers and lifting up. Do not touch the gold-plated contacts with your fingers.

REASSEMBLY AND REPLACEMENT

a. Grasp a corner of the elastomer with tweezers and place it into plastic frame. Align keyed corner of elastomer with keyed corner of plastic frame. Tamp elastomer into plastic frame uniformly.

NOTE

Cleanliness is very important. Small hairs and elastomer flash under the contacts, which are almost invisible to the naked eye, will prevent good electrical contact. Most apparent failures of the hybrid are actually due to contamination of the Hypcon. Do not touch the gold-plated contacts with fingers.

b. Place the hybrid into the square hole in circuit board. The hybrid is keyed so that it will fit into the circuit board in only one orientation. When the back of the hybrid rests on the heat sink pedestal, the top of the hybrid should be flush with the top of the circuit board.

c. Place the plastic frame with elastomer installed over hybrid such that key (pointed tab) align with the corner arrow on the circuit board.

d. Insert mounting hardware and apply two inch-pounds of torque (2.3 cm-kg) to secure connector assembly. Do not overtighten. To do so will strip the microcircuit stiffener/heat sink mounting threads.

7386-328B

Figure 3-28 (cont). Hypcon assembly removal and replacement.

Tighten the Hypcon mounting screws to two inch-pounds of torque.

Make sure that the elastomer is properly seated in the contact holder before remounting the assembly to the circuit board. Exercise care when mounting the frame—elastomer contact holder—hybrid IC assembly to the circuit board to prevent misalignment between the connector and board.

CAUTION

Because of the close tolerances involved, special care must be taken to ensure correct index alignment of each Hypcon part during reassembly. Failure to do so can crack the hybrid substrate. (See Figure 2-27 for index locations).

If your instrument contains both the flush and stepped type of Hypcon connectors, be careful not to mix the elastomer contact holders during reassembly. Flush Hypcon connectors have green elastomer contact holders and the plastic frame is marked FLUSH. Stepped Hypcons have neutral-colored elastomer contact holders with a slight ridge or step on the contact surface, the large frames are marked STEPPED. The registration pins on the stepped plastic frame are slightly longer than those on the flush frame. The elastomer contact holder in the small stepped connectors is indexed differently than the large connectors. Look for a small gold arrow in one corner of the holder instead of a flat corner. Match this corner arrow with the pointed corner of the plastic frame. Give close attention to this indexing, because elastomer contact holders can easily be inserted incorrectly.

Differences also exist between the large flush and the large stepped Hypcon circuit board receptacles. Figure 3-27 shows the cross-sectional differences which must be observed when working with an instrument that contains both types of Hypcon connectors.

CAUTION

Damage to the elastomer contact holder can result if the connectors are not mated properly with the board receptacle.

When replacing the hybrid, insert it into the board opening and then position the Hypcon connector in the board registration holes for perfect alignment. The outer portion of the Hypcon frame should be flush with the circuit board before the four mounting screws are tightened. Avoid touching the hybrid and elastomer contact with your fingers; finger oils can degrade reliability.

A procedure for removal and replacement is included in Figure 3-27.

Hybrid Integrated Circuits

The A5 Acquisition board has a Hybrid IC (U1710). The IC is mounted to the inside of its heatsink cover. The cover is oriented to the circuit board in two ways. First, two of the socket screws are offset. This allows the cover to be reinstalled only one way. Second, one corner is flat rather than rounded as are the other corners. This flat corner matches an etched design on the board.

To remove a Hybrid IC, proceed as follows:

1. Remove the four 1/4-inch retaining nuts from the heatsink cover.

2. Lift the cover (and the IC) from its socket.

CAUTION

Avoid touching the IC or the socket contacts with your fingers. Finger oils can degrade reliability.

Replace the Hybrid IC as follows:

1. Orient the replacement IC to fit the socket. Align the flat corner of the cover to the flat corner etched design on the circuit board.

2. Press the IC toward the board to feel the spring tension of the contacts. Move the cover around slightly until the IC seats flush against its socket.

3. Hold the IC in place and install the four retaining nuts. Tighten them **fingertight.**

4. Use a torque wrench to tighten the nuts in a **diagonal method only** to 3-1/2 to 4 inch-pounds. Refer to the following CAUTION for diagonal tightening instructions.

CAUTION

Do not tighten these nuts clockwise. *That will crack the heatsink cover.*

Instead, ***use the following diagonal method to tighten them:*** *Begin at the lower-right nut and tighten it. Go diagonally across the cover to the upper-left nut. Tighten it. Next, drop down to the lower-left nut and tighten it. Then, go diagonally across the cover to the upper-right nut. Tighten it.*

Diagonally check that all four nuts are tightened to the recommended torque. (See step 4.)

Microcircuit Integrated Circuits

There are two microcircuit ICs on the A5 Acquisition board. They are the U1200 and U1220 Interpolators.

These microcircuit ICs have integral heatsinks. The IC is oriented to the circuit board in two ways. First, three locator posts (inside the cover) bracket the IC. Two posts are at one end and the third at the other end. Second, the cover has a flat-edged corner which matches with an etched design on the board.

To remove the microcircuit IC, proceed as follows:

1. Remove the four 1/4" retaining nuts from the cover.
2. Lift the IC (and attached cover) from its socket.

CAUTION

Avoid touching the IC or the socket contacts with your fingers. Finger oils can degrade reliability.

Replace the microcircuit IC as follows:

1. Align the flat edge corner of the cover with the matching design on the board. This orients the IC correctly.
2. Check that the locator pins fit in their respective locations.
3. Press the IC toward the board to feel the spring tension of the contacts.
4. Hold the IC in place and install the four retaining nuts. Tighten them **finger tight.**
5. **Tighten the nuts using a diagonal method only.** See the following CAUTION for diagonal tightening instructions.

   a. Use a torque wrench to tighten the nuts to 3-1/2 to 4 inch-pounds.

CAUTION

Do not tighten these nuts clockwise. That will crack the heatsink cover.

Use a diagonal nut-tightening procedure. See the Hybrid Integrated Circuit Replacement procedure, immediately preceding this part. (Follow the instructions in the CAUTION.)

Chip Carrier ("Slam-Pack") Integrated Circuits

Several circuit boards have these ICs. They include the A5 Acquisition board, the A6 Time Base board, the A7 Display Controller board, the A14 Input/Output board, the A17 Main Processor board, and the A18 Memory board.

Some Chip Carrier ICs have raised, ridged heatsink covers. Others may have flat covers. The IC is oriented to its socket by a flat corner. The other corners are notched to fit the edges of the socket. (See Figure 3-27.) The flat-edged corner of the IC aligns with a "spring" (small metal tab) at one corner of the socket.

To remove a Chip Carrier IC, proceed as follows:

1. Hold the cover in place and unfasten the holding clip by pushing it aside. It may help to push down slightly on the cover.

2. Remove the cover slowly to prevent the IC from falling out.

NOTE

Observe the index of the IC before removing it.

3. Remove the IC with tweezers.

CAUTION

Avoid touching the IC or the socket contacts with your fingers. Finger oils can degrade reliability.

Replace the Chip Carrier IC as follows:

1. Using tweezers, put the flat edge of the replacement IC against its index spring.

CAUTION

Check that the spring in the corner does not get bent by the flat corner index, which may short the corner two contacts.

2. Arrange the other corners with tweezers to fit evenly at the edges of the socket.

3. Set the cover flat on the chip with the end tabs properly aligned with the mating recesses in the socket.

4. Push down on the cover, keeping it flat on the chip, and slide the cover into place. Hold it there while moving the holding clip over the tabs on the opposite end of the cover.

5. Check that the cover is secure.

Interconnecting Pins

Two methods of interconnection are used to electrically connect circuit boards with other boards and components. When the interconnection is made with a coaxial cable, a special end-lead connector plugs into a socket on the board. Other interconnections are made with a pin soldered into the board.

Two types of connectors are used for these interconnecting pins. If the connector is mounted on a plug-on board, a special socket is soldered into the board. If the connector is on the end of a lead, an end-lead pin connector is used which mates with the interconnecting pin. The following information provides the removal and replacement procedure for the various types of interconnecting methods.

Coaxial-type End Lead Connectors (Peltolas)

Replacement of coaxial-type end-lead connectors requires special tools and techniques. Only experienced maintenance personnel should attempt to remove or replace these connectors. We recommend that the damaged cable or wiring harness be replaced as a unit. For cable or wiring harness part numbers, see the Replaceable Mechanical Parts List. An alternative solution is to refer the replacement of the defective connector to your local Tektronix Field Office or representative. Figure 3-29 gives an exploded view of a coaxial end-lead connector assembly.

Color coding of wires may be helpful to connect a Peltola connector to its socket on a circuit board. The wire insulation's color, or its colored stripe, is the same as the color represented by the last digit of the JXX component number. (EXAMPLE: a green wire would connect to a J05 socket.)

Circuit Board Pins

Replacing circuit board pins on multilayer boards is not recommended. (All circuit boards in the 11401 are multilayer boards.)

Circuit Board Pin Sockets

The pin sockets on the circuit boards are soldered to the back of the board. To remove or replace one of these sockets, first unsolder the pin (use an anti-static vacuum-type desoldering tool to remove excess solder). Then straighten the tabs on the socket and remove the socket from the board. Place the new socket in the circuit board hole and press the tabs down against the board. Solder the tabs of the socket to the circuit board; be careful not to get solder inside the socket.

CAUTION

The spring tension of the pin sockets ensures a good connection between the circuit board and the pin. This spring tension can be destroyed by using the pin sockets as a connecting point for spring-loaded probe tips, alligator clips, etc.

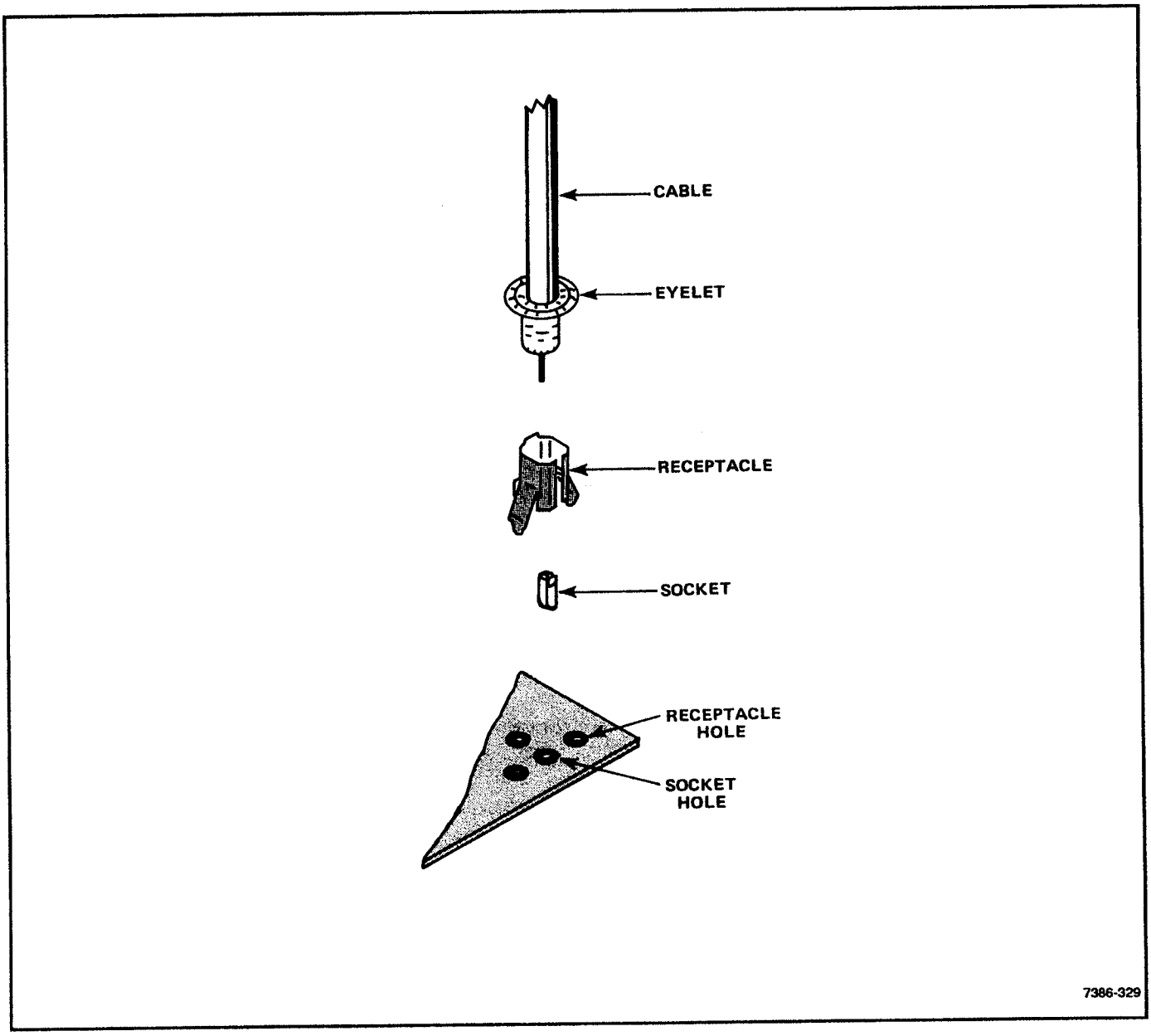


Figure 3-29. Coaxial end-lead connector assembly.

Multi-Pin Connectors

The pin connectors used to connect the wires to the interconnecting pins are clamped to the ends of the associated leads.

Some of the pin connectors are grouped together and mounted in a plastic holder. The overall result is that these connectors are installed and removed as a multi-pin connector. If the individual end-lead pin connectors are removed from the plastic holder, note the order of the individual wires for correct replacement into the holder.

To remove or replace damaged multi-pin connectors, remove the old pin connector from the end of the lead and clamp the replacement connector to the lead.

Arrangement of Pins in Multi-Pin Connectors

Pin 1 on multi-pin connectors is designated with a triangle (or arrowhead). A triangle, dot, or square printed on circuit boards denotes pin 1. When a connection is made to a circuit board, the orientation of the triangle on the multi-pin holder is determined by the index (triangle, dot or square) printed on the circuit board. (See Fig. 3-30.) Most circuit-board, mounted connectors have a triangle index mark.

NOTE

Some multi-pin connectors are keyed by a gap between the pin 1 and 3 positions in the holder. (A small plastic plug covers the pin 2 position on the end of the holder.) There is a corresponding gap between pins 1 and 3 on the circuit board.

Align the gap in the multi-pin holder with the gap between the circuit board pins. The connector is then ready to be installed.

Many of the larger, gray-colored, multi-pin ribbon connectors have a red line along one side of their attached wire cables. This red line indicates the location of pin 1 and 2 or the location of the triangle index mark.

Some of the gray-colored ribbon cables may have the number of their connectors stamped on them.

The ribbon connectors have two functions. The first is to provide a strain release for the wire connections. The wire ribbon is wrapped around a bar in between the wire connections and the top of the connector. Strain is then felt between the wires and the top of the connector. This releases most of the strain which would otherwise be felt on the wire connections.

The second function on most of the ribbon connectors is to provide a pull-tab to ease disconnection. A white-colored pull-tab is attached inside the connector. When the tab is pulled, even pressure is applied across the connector. The connector separates from its holder easily.

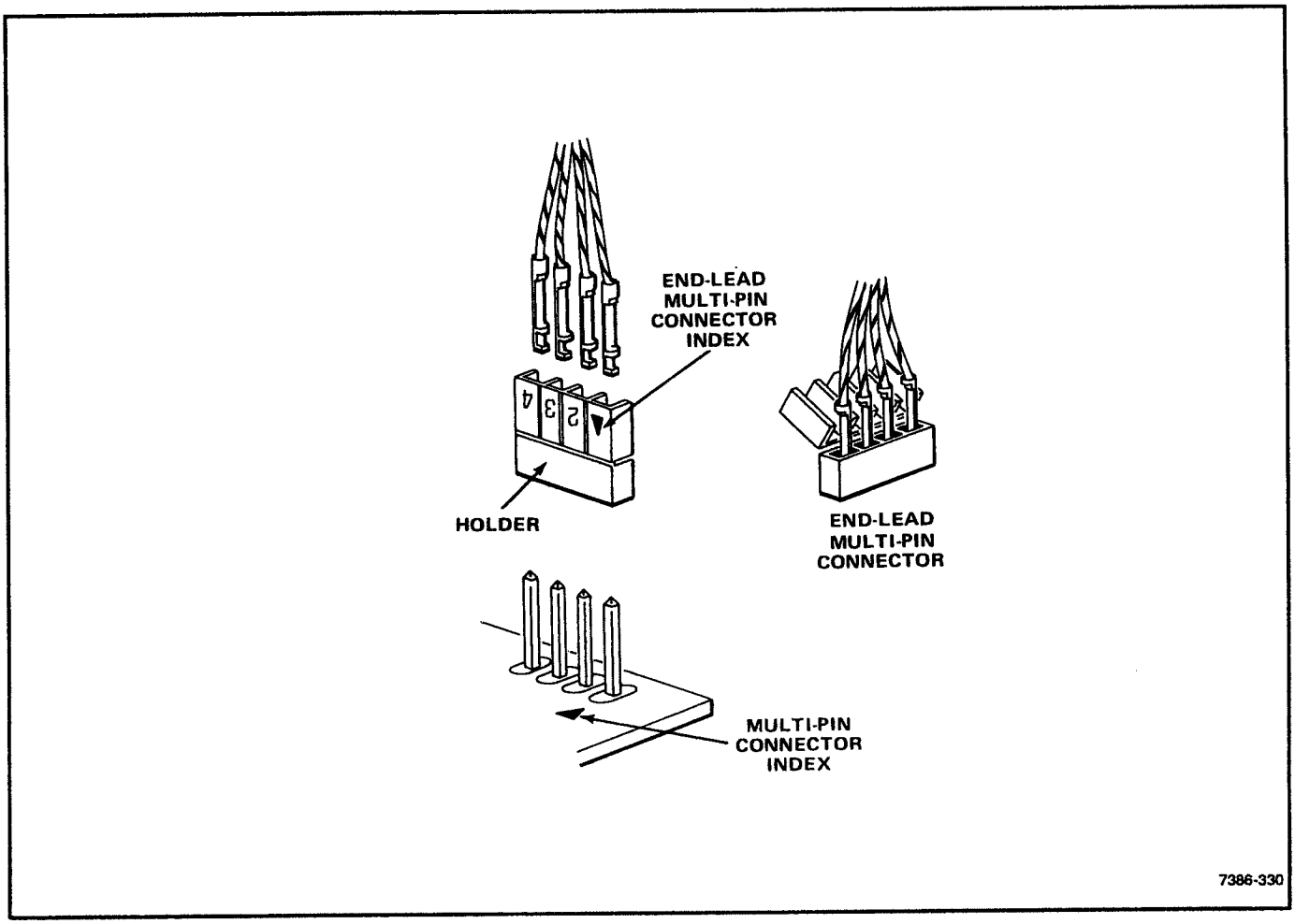


Figure 3-30. Orientation of multi-pin connectors.

NOTE

To remove these gray-colored ribbon connectors, grasp the white pull-tab (fastened into the connector, if there) and pull it loose from the holder.

If there isn't a white pull-tab present in the connector, grasp the ends of the connector instead. Pull it straight out from the connector socket.

If a ribbon connector is found with an open or shorted wire, individual wires cannot be removed or replaced. The ribbon connector and cable must be replaced as a unit.

Plug-in Interface Connectors

If connectors or contacts are damaged, we recommend replacing the entire A1 Plug-In Interface board. See the "A1 Plug-In Interface board Removal" proceedure that is located in the "Circuit Board Removal" part of this section. An alternate solution is to have your local Tektronix Field Office repair or replace the damaged A1 Plug-In Interface board.

Power Transformer

Replace the power transformer only with a direct replacement Tektronix transformer. Remove and replace the power transformer as follows:

1. Remove the A2A2 Control Rectifier board as described under Circuit Board Removal in this section.

NOTE

Record the position of the transformer leads so they may be correctly replaced.

2. Unsolder the transformer leads from the A2A2 Control Rectifier board. Remove any excess solder from the circuit-board pads (see Soldering Techniques, in this section).

3. Remove the transformer.

4. Place the new transformer in position and solder the leads to the A2A2 Control Rectifier circuit-board pads.

5. Replace the A2A2 Control Rectifier in the power supply as described under Circuit Board Removal in this section.

Line Fuse

The Line fuse used in the 11401/11402 is located on the rear panel of the power supply. Replace the line fuse (F99) with one of proper type and rating.

NOTE

Line voltage fuse F99 is used for both 115 and 230 V operation. No change in the fuse is necessary when switching the LINE VOLTAGE SELECTOR switch between 115 V and 230 V.

TABLE 3-3
Adjustments Required after Circuit Board or Module Replacement

Circuit Board or Module Replaced	No Adjustment Required	Adjustments Required	Information
A1 Plug-In Interface	✓		
A2A1 Line Inverter		+5 V Ref (A3R800) Reg Ref (A4R830)	
A2A2 Control Rectifier		+5 V Ref (A3R800) Reg Ref (A4R830)	
A4 Regulator		Reg Ref (A4R830)	
A5 Acquisition		Offset (A5R1576) Gain (A5R1582)	
A6 Time Base*	✓		
A7 Display Controller*	✓		
A8 Crt Driver		Main Brite (A8R202) Horiz Hold (A8R620) Vert Hold (A8R530) Vert Pos (A8R520) Vert Size (A8L120) Horiz Lin (A8R541) Horiz Size (A8R621) Horiz Pos (A8R540) Focus (A8R100)	
A9 Touch Panel	✓		
A10 Front-Panel Control	✓		
A12 Rear Panel	✓		
A13 Mother	✓		
A14 I/O		Time/Date	Refer to the 11401/11402 User's Reference Manual, under "Utilities."
A15 MMU	✓		

TABLE 3-3 (cont)
Adjustments Required after Circuit Board or Module Replacement

Circuit Board or Module Replaced	No Adjustment Required	Adjustments Required	Information
A16 Waveform Compressor	✓		
A17 Main Processor*		STEP 1.	If the firmware version of the new board is different from that of the old board, you must update the A18 Memory Processor board's firmware to the same version.
		STEP 2. Clear NV RAM	Remove the Battery Backup Disable jumper J150, located on the A17 Main Processor board. Connect a shorting wire across capacitor C150 (also located on the A17 Main Processor board) for approximately 15 seconds. Remove shorting wire and replace the Battery Backup Disable jumper on J150.
		STEP 3. Install Unit ID	Refer to the 11401/11402 User's Reference Manual, under "Command Set."
		STEP 4. Install Cal Constant	Refer to Step D, Sampler and Digitizer in the Adjustment Procedure.

TABLE 3-3 (cont)
Adjustments Required after Circuit Board or Module Replacement

Circuit Board or Module Replaced	No Adjustment Required	Adjustments Required	Information
A18 Memory*	✓		If the firmware version of the new board is different from that of the old board, you must update the A17 Main Processor board's firmware to the same version.
Crt		Main Brite (A8R202) Horiz Hold (A8R620) Vert Hold (A8R530) Vert Pos (A8R520) Vert Size (A8L120) Horiz Lin (A8R541) Horiz Size (A8R621) Horiz Pos (A8R540) Focus (A8R100)	
Power Module		+5 V Ref (A3R800) Reg Ref (A4R830)	

***These circuit boards contain firmware. If the board is replaced, refer to the firmware update instructions that accompany the replacement board.**

Contents

Section 4—Diagnostics

Section 4
Diagnostics

This section describes the diagnostic tests and tools designed for the 11401 and 11402 Digitizing Oscilloscopes. The automatic and manual diagnostic tests that help verify instrument operation are described here as well as methods for using them to troubleshoot an instrument to the component level. Other diagnostic facilities, both internal and external to the instruments, are also discussed in this manual. Plug-in unit diagnostics are covered in their respective service manuals.

The information in athis manual should be used in conjunction with the theory of operation and schematic diagrams in order to be the most useful.

Overview

The 11401 and 11402 run Diagnostic Self Tests at power-up to verify most hardware functions. Each installed plug-in unit also performs an independent set of self tests at power-up and reports the results to the mainframe. When the low-level Self Tests start at power-up the message **Diagnostics in Progress** is displayed. Shortly afterward, the message **Comm Test in Progress** is also displayed. When the high-level Self Tests start, the message **Self Test in Progress** replaces both previous messages. These tests run automatically and return the instrument to normal operation when no faults are found.

NOTE

Pressing front panel buttons, turning knobs, or touching the Touch Panel while tests are running may cause a diagnostic failure.

If a mainframe or plug-in unit Self Test fails, the instrument emits two beeps. It then enters the Extended Diagnostics mode and displays a menu showing the status of all test blocks. Test blocks with failures will have an error index code next to the block name. For low-level test failures, the error index codes are read using procedures described later in this section.

Before the power-up Self Tests can run, the power supply must be operating properly. Power supply troubleshooting is discussed in documentation provided with the Extended Diagnostics 11000-Series Power Supplies Troubleshooting Fixture (067-1264-00).

Kernel Diagnostics

At power-up, and only at power-up, the Executive, Display, and Digitizer subsystem processors run their local set of Kernel Diagnostics tests. These low-level tests verify the kernel processors, their necessary support circuitry (i.e., ROM, RAM, clocks, etc.), and their ability to communicate with each other. Any failure of the Kernel Diagnostics will cause a single beep (Executive subsystem only) and the failing subsystem kernel to loop on the failed test and indicate the failure with circuit board, status LEDs and pins. Kernel processors that cannot run Kernel Diagnostics can be troubleshot with the procedures described under Troubleshooting Non-Functioning Subsystem Kernels.

Self Test Diagnostics

If the Executive subsystem runs all of its Kernel Diagnostics without failure it will start Self Test Diagnostics (possibly only locally), even though the Display and Digitizer may have failed to establish communication with the Executive subsystem. The Self Test Diagnostics, which are run at power-up and can also be run at other times, help to verify overall instrument functionality in a short period of time.

Extended Diagnostics

Any failures in the Self Test Diagnostics cause the instrument to enter the Extended Diagnostics mode, after the remaining Self Tests have run. The Extended Diagnostics display shows a test menu with the names of major circuit blocks and the test status of each. The test status will be either a "pass" indicator, a "****" indicator for tests that have not run or which do not have current status information, a "????" indicator for optional hardware which is not present or subsystems whose communication paths are not functional, or a five-digit error index code that refers to a test description in this manual. See the Error Index Overview section under Extended Diagnostics on how to access a test description for a given error index code.

Extended Diagnostics consists of all Self Tests plus a number of interactive tests used to troubleshoot, verify, or calibrate specific circuits. In addition, there may be some automatic tests which were excluded from being run in Self Test, such as NVRAM tests or lengthy RAM tests, to protect the instrument from undesirable events or to reduce the Self Test execution time. In the Extended Diagnostics mode, one or more tests can be selected and run with the chosen test loop conditions. These tests can also be run remotely with a terminal/controller connected to the RS-232-C or GPIB interface. See the Remote Diagnostics section under Extended Diagnostics for detailed information on external diagnostics control.

Running Self Tests/Extended Diagnostics From Normal Operation

Self Test or Extended Diagnostics can be selected and run at any time during normal instrument operation. To run the tests, press the UTILITY button, then select either the Self Test or the Extended Diagnostic label. When the Self Tests are invoked in this way, the instrument will not run the Kernel Diagnostics. Self Test Diagnostics contain equivalent Kernel Diagnostic tests, therefore circuit coverage is not being reduced by not performing the actual Kernel Diagnostic tests. All Self Tests will run to verify instrument functionality, then return the instrument to normal operation if no faults occur. Once again, a failure of the Self Tests will automatically invoke the Extended Diagnostics mode.

Disabled Front Panel Buttons (Except Hardcopy)

While in Self Test or Extended Diagnostics, most normal functions of the front panel buttons are suspended. Pressing a front panel button or touching the Touch Panel during some tests will cause an apparent failure. When an Extended Diagnostics menu is displayed, you can press the front panel Hardcopy button for a permanent record of test failures (a compatible printer must be connected). See the Hardcopy section under Extended Diagnostics for detailed information on diagnostic Hardcopy. For information on connecting a printer, refer to the User's Reference Manual.

Diagnostic Options Jumpers

The Diagnostic Options jumpers J710 -J715 on the Standard I/O Board (A14) provide a way to separately enable or disable the Kernel Diagnostics (Executive Subsystem only), Self Tests, and Extended Diagnostics at power-up. Refer to Figure 4-1 for jumper location. Also, the default RS-232-C baud rate for external control of the Diagnostics can be changed from the factory setting of 9600. One jumper setting for the Kernel Diagnostics causes the instrument to continuously loop on the RS-232 loopback test to aid in troubleshooting the RS-232-C port. For some functions, two jumpers are used together to specify the setting. Note that only jumper J710 is labeled on the circuit board.

Each jumper is set (true) when the two pins on the right are jumpered. In Table 4-1, a set jumper is indicated by a "1" and an unset (false) jumper by a "0." Factory settings are indicated by a "Normal" entry in the Description column.

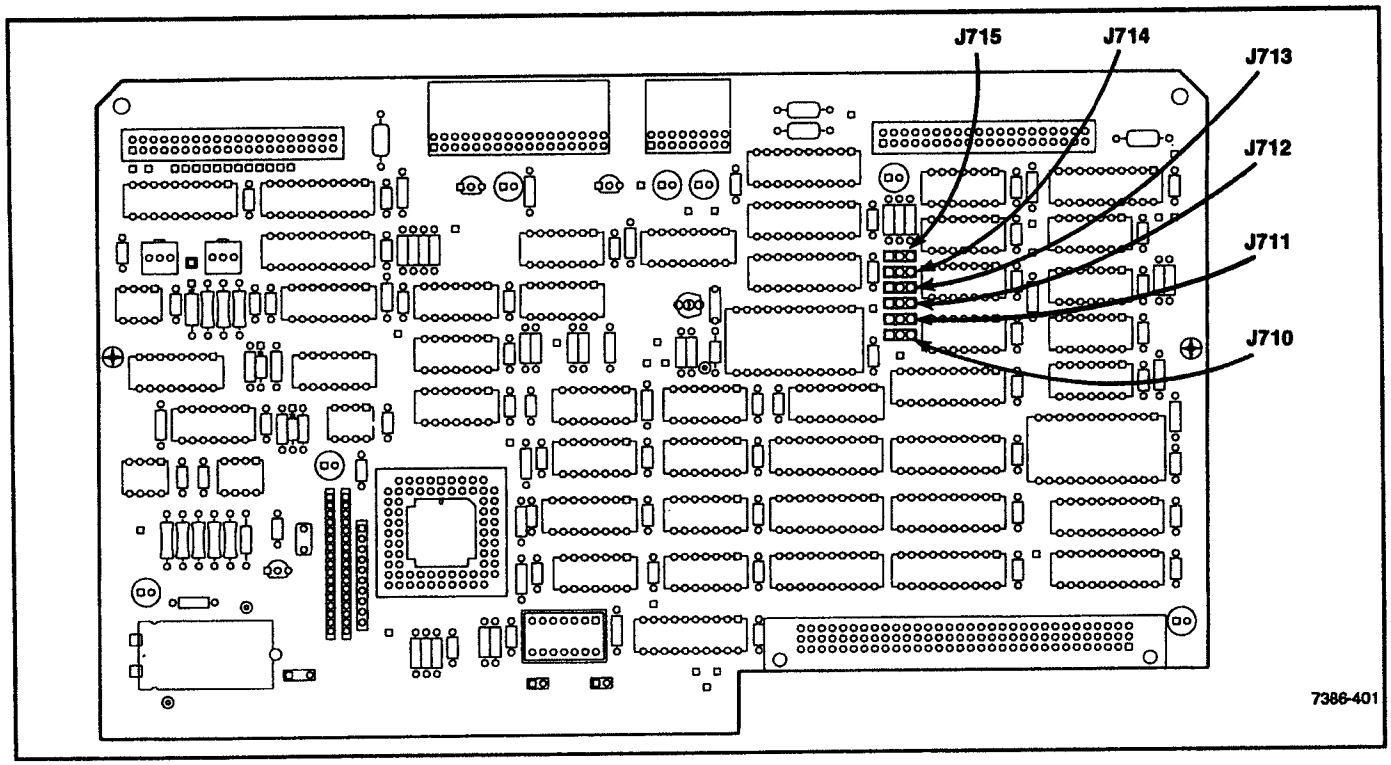


Figure 4-1. Location of the Diagnostic Options Jumpers on the Standard I/O Board.

TABLE 4-1
Jumper Settings of the Diagnostic Options Jumpers

Controlled Parameter	Jumper(s)		Description
Executive Kernel Diagnostics:	J711	J710	
	0	0	Bypass Kernel Diagnostics.
	0	1	Loop on RS-232 test after running Kernel Diagnostics.
	1	0	Run Kernel Diagnostics but do not loop on a failed test.
	1	1	Normal; run Kernel Diagnostics and loop on the first failed test.
Self Test Diagnostics:	J712		
	0		Bypass Self Test Diagnostics at power-up.
	1		Normal; run Self Tests at power-up.
Extended Diagnostics:	J713		
	0		Bypass Extended Diagnostics after any Self Test.
	1		Normal; run Extended Diagnostics after a Self Test failure.
RS-232-C Baud Rate during Extended Diagnostics:	J715	J714	
	0	0	300 baud.
	0	1	1200 baud.
	1	0	Normal; 9600 baud.
	1	1	19,200 baud.

Equipment Required

The following equipment is recommended for use in the troubleshooting the 11401/11402 with the Diagnostic procedures presented in this section. For a complete list of all recommended maintenance equipment, refer to the 11401 and 11402 Service Reference Manual.

- TEKTRONIX 067-1267-00 Calibration Fixture: Troubleshooting Aid Extender Card with Cables.
- TEKTRONIX P6401 Logic Probe.
- TEKTRONIX 1240 Logic Analyzer.
- TEKTRONIX 7904A Oscilloscope with 7A26 Dual Trace Amplifier, 7S14 Dual Trace Delayed Sweep Sampler, and 7B10 and 7B15 Time Bases, or equivalent.
- Test terminal, ANSI 3.64 compatible, with RS-232-C interface.

Circuit Coverage

The following list contains those circuits in the 11401/11402 Digitizing Oscilloscope mainframe that are less than 50% functionally verified (directly or indirectly) by the Extended Diagnostics tests. Functional verification is 0% unless the item is classified as "partial", which indicates it is 1-50% tested. Numbers within "<>" indicate the associated schematic numbers. Circuits not on this list are 50 to 100% functionally tested by the Diagnostics with or without installed plug-in units.

To ensure good functional verification of the Digitizer subsystem and plug-in unit interface, install three, 11000 series, generic plug-in amplifiers (i.e., 11A32, 11A34, 11A52) before testing. Circuit entries that depend on installed plug-in units for proper testing, include the phrase "with plug-in units" and the degree of coverage.

I/O Board (A14)
- I/O-Internal & Control <21>
  - Real Time Decode/Request—partial
  - Realtime Clock—partial

Rear Panel Board (A12)
- Rear Panel—GPIB & RS-232-C <3>
  - GPIB Controller—partial
  - GPIB Data Buffers
  - GPIB Control Driver
  - GPIB Status

Plug-in Interface Board (A1)—Partial with plug-in units.

Acquisition Board (A5)

- Acq Sampler <6>
  - Sample Gate Generator—Good with plug-in units.
  - Strobe Generator—Good with plug-in units.
  - Strobe Driver—Good with plug-in units.
  - Sampler—Good with plug-in units.
  - Vertical Channel Select—Good with plug-in units.
- Acq Clock Generators & Time Interpolators <7>
  - Fine Time Interpolators
  - Fine Time Interpolators Mux
  - Fine Time Interpolators Shift Register
  - Course Time Interpolators
  - Course Time Interpolators Mux
- Acq Trigger & Holdoff Logic <10>
  - Trigger Source Register—partial
  - Trigger Signal Buffers
  - Trigger Hybrid—partial
  - Buffer—partial
  - Trigger Signals Conditioner—partial
  - Delay By Events—partial
  - Holdoff Logic—partial
- Acq Main & Window Fine Holdoff Generators <11>
  - Acquisition Control Register—partial
  - Main Fine Holdoff Generator
  - Window Fine Holdoff Generator
- Acq Autocal & Refresh <12>
  - Plug-in Ground Select—Good with plug-in units.

Time Base Board (A6)

- TB A/D Interface & System Clock <14>
  - Invalid Point Detect
- TB Control Latches & Plug-in Sequencer <15>
  - Timebase Status Readback Buffer—partial
  - Last Request Control—partial
  - Chop Controller & Plug-in Sequencer—partial
  - Chop Transition Blanking—partial
  - Aux Z-Axis Blanking
- TB MPU Kernel & Address Decoding <16>
  - Wfm ID Tables Select—partial
- TB Main Counters <18>
  - Main Request Logic
  - Last Request Generation
  - Window 2 Seek Control
  - Window 2 Request Logic
- TB Window Counters <19>
  - Window 1 Seek Control
  - Window 1 Request Logic
- TB DAG Interface <20>
  - Final Wfm ID Encoding Table—partial
  - Destination Address Generator—partial
  - Restart Control—partial

Kernel Diagnostics

Kernel Diagnostics normally run at power-up to verify the local processor hardware and to establish communication with the other subsystems. The local hardware includes the kernel microprocessor, clocks, ROM, RAM, address decoders, data buses, address buses and control signals to the ROM and RAM, and other hardware necessary for test functions (e.g., DMA channels, interrupts, etc.). A properly operating power supply is also necessary for these tests to run. If a mainframe subsystem is unable to run its Kernel Diagnostics, refer to Troubleshooting Non-Functioning Subsystem Kernels. Errors found during Kernel Diagnostics produce error index codes on indicators within the instrument (and externally when possible) as discussed below.

Error Index Overview

Kernel Diagnostics produce error index codes which point to test descriptions. The test descriptions, which are named for the circuits being tested, are found in the Test Descriptions portion of this manual. Each subsystem's Kernel Diagnostics generates error index codes in different ways and with different formats. The following sections will describe in detail the ways the error index codes are generated and how and where to read them for each subsystem.

Mainframe Test Status Indicators

All mainframe subsystem kernels in the instrument have similar kernel test status indicators. One circuit board in each subsystem kernel contains two status LEDs that indicate the onset, failure, and completion of testing, and five or more status pins from which an error index code can be read. The status pins are located on each board near the status LEDs. Illustrations in the Troubleshooting section of the Service Reference Manual show the locations of the kernel status LEDs and pins. These status indicators are used for other purposes during normal instrument operation, hence they are valid only while Diagnostics are running. Component locations are shown in Figures 4-2, 4-3, and 4-4.

The instrument will need some disassembly in order to see the status LEDs and to read the on-board status pins. Refer to the Service Reference Manual for cabinet and board removal information, including important warnings and cautions.

Operation of Status LEDs and Pins

The status LEDs and pins for all subsystems operate the same, though the Kernel Diagnostics in each subsystem run independently . At power-up, both status LEDs are lit and all test status pins are set high (logic 1) by a hardware reset line. When a kernel processor starts executing diagnostic code from the ROMs, it turns one LED off. At the start of the second test, and of those that follow, the count on the status pins is decremented by one. Thus, a unique error code is registered on the status pins at the start of each test. When all Kernel Diagnostic tests in a subsystem pass, both LEDs are turned off and Self Test Diagnostics begin. If a test fails, but is able to finish execution, the lit LED will be turned off and the other LED will be lit. If possible, a single beep will be generated (Executive Kernel only) and the kernel processor will begin looping on the failed test. To identify the failed test, read the error index code on the status pins. The Executive and Display kernels have additional ways of displaying failure information which are discussed under under their respective headings.

If a kernel processor locks up and is unable to finish running a test routine, the LEDs will not trade states and the kernel processor will not be able to loop on the failed test. To determine whether the kernel processor is looping or locked up, turn the ON/STANDBY switch to Standby then back on and observe the status LEDs. When the tests start running, one LED will be lit and one will be off. Now, notice whether or not the status LEDs change state (i.e., the lit one goes off and the off LED gets lit). If one LED comes on and stays on, the instrument has locked up on the test indicated on the status pins. This condition generally indicates a problem with on of the kernel diagnostic ROMs or associated circuitry. If the LEDs change state after testing begins, then the kernel processor has begun looping on the failed routine.

Start/Stop Trigger Pin

When a Kernel processor is looping on a failed test, the start/stop trigger pin transitions high at the start of the test and low at the completion. This signal may be useful for triggering test equipment during troubleshooting.

Executive Kernel

The Executive Kernel Diagnostics normally run at power-up as do tests in the other subsystems. The sequence of events for testing and the status indicators have been described above. After the Kernel tests have run and passed, the Executive processor checks the Diagnostic Options Jumpers on the Standard I/O Board. If the jumpers are set to loop on the RS-232 test, the Executive processor proceeds to do so. If the Diagnostic Options Jumpers are in the normal position, the Executive processor attempts to communicate with the other subsystems. The order of establishing subsystem communication is Display, Digitizer, then, Left, Center, and Right plug-in units. If the Display or Digitizer is unable to communicate, it goes into an echo mode in which it inverts, then retransmits any data received from the Executive Subsystem. Regardless of communication problems with the other subsystems, the Executive will continue on to Self Test if none of its Kernel Diagnostics have failed.

A failure of the Executive Kernel Diagnostics is indicated by one beep, if possible, from the instrument. If the Executive processor is unable to complete a kernel test and has locked up, then no beep will be heard and the internal status LEDs and Pins must be examined.

The status of the Executive Kernel Diagnostics can be read in two ways. When signal paths to the front panel are good, the error index code for the failed test is registered on the MENUS LEDs with on LEDs (logic 1) and off LEDs (logic 0). Also, during Kernel Diagnostics, the TOUCH PANEL ON and the DIGITIZER RUN status LEDs mirror the states of the Executive Kernel status LEDs. At power-up, these front panel button LEDs are both lit then one is turned off when the first test begins. Table 4-2 gives the bit encoding used for the MENUS LEDs.

TABLE 4-2
MENUS LEDs Encoding

	MENUS Button LED	Bit Number
Low hex digit		
	WAVEFORM	1-LSB (least significant bit)
	TRIGGER	2
	MEASURE	3
	STORE/RECALL	4
High hex digit		
	UTILITY	5-MSB (most significant bit)

If the Executive Kernel is unable to write to the MENUS LEDs, the error index code can be read from the on-board status pins. See Figure 4-2 to gain access to the status pins. Remove the top cabinet cover and locate the Main Processor Board (A17) in the card-cage. Figure 3-8 in the Maintenance section shows the location of most mainframe circuit boards. When reading an error index code with a logic probe or oscilloscope, carefully note the pin location for each bit. On the status pins, a TTL high is a logic one and a TTL low is a logic zero. The LSB and next three pins/bits (TP205 - TP202) make up the low-hex digit. The MSB (TP201) is the only bit used for the high hex digit, so the digit will be one or zero.

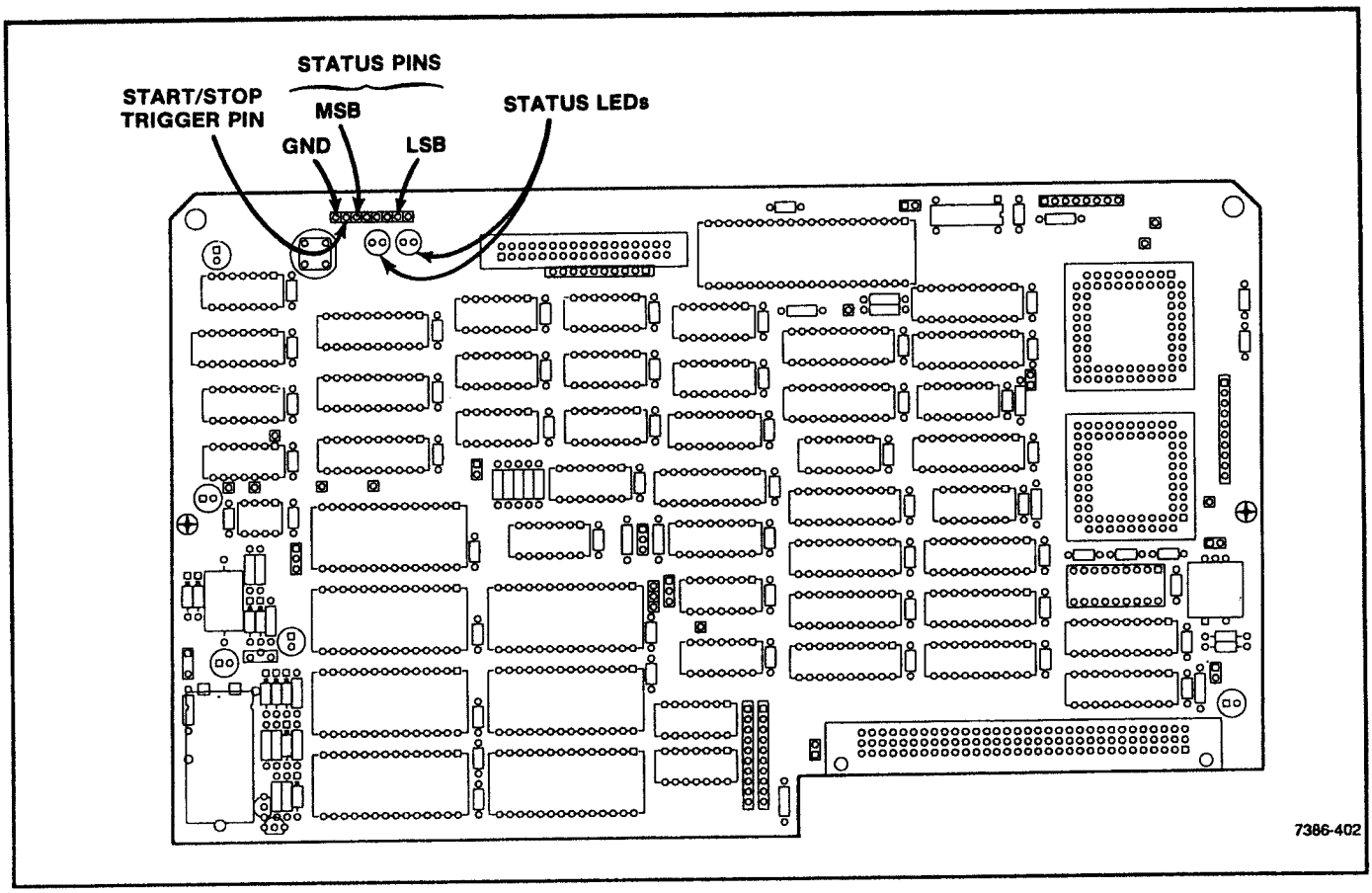


Figure 4-2. Main Processor board (A17) showing Executive Kernel pin locations.

Power-Up Options

Executive Kernel Diagnostics may be bypassed or forced to loop on the power-up RS-232 test with the Diagnostic Options Jumpers on the Standard I/O Board. The Kernel Diagnostics should be bypassed only in special circumstances because the condition of the instrument is unknown when the Diagnostics are bypassed. See the discussion in the Diagnostic Options Jumpers section under Overview near the beginning of this manual.

Display Kernel

The Display Kernel Diagnostics run at power-up as do tests in the other subsystems, but test failures are also written to the display as failure messages, if possible. Because the display may be defective, error index codes are still generated on the Display Controller Board (A7) and can be read from the status pins. The common sequence of events for testing and the status indicators is described above under Operation of Status LEDs and Pins.

At power-up, the Display Kernel first clears the display, then writes **Diagnostics in Progress** to the display. Next, the Kernel Diagnostics are started with the usual lighting of a status LED and setting the status pins all high by a hardware reset line. If a failure occurs, the status LEDs change state and a failure message such as

Dsy Kernel Failure
RAM Data Bit

is written on the display. Once again, if the display is defective the error index code can be read from the status pins on the Display Controller Board (see Fig. 4-3). See Figure 3-8 in the Maintenance section for circuit board locations. To get to the status pins, remove the top cabinet cover and locate the Display Controller Board lying just under the cover. When reading an error index code with a logic probe or oscilloscope, carefully note the pin location for each bit . On the status pins, a TTL high is a logic one and a TTL low is a logic zero. Pins 0 (LSB - TP H) to 3 (TP E) make up the low hex digit and pins 4 (TP D) to 7 (MSB - TP A) make up the high hex digit.

When the Kernel Diagnostics are finished and the communication test is about to start, the Display subsystem writes the message **Comm Test in Progress** to the display. If communication is successful, the Display proceeds to Self Test (assuming that the Diagnostic Option Jumpers are set to not bypass Self Test). A communication failure causes the Display Kernel to enter an echo mode in which it receives data from the Executive subsystem, inverts it, and sends it back to the Executive. This echo mode provides a means to troubleshoot communication problems.

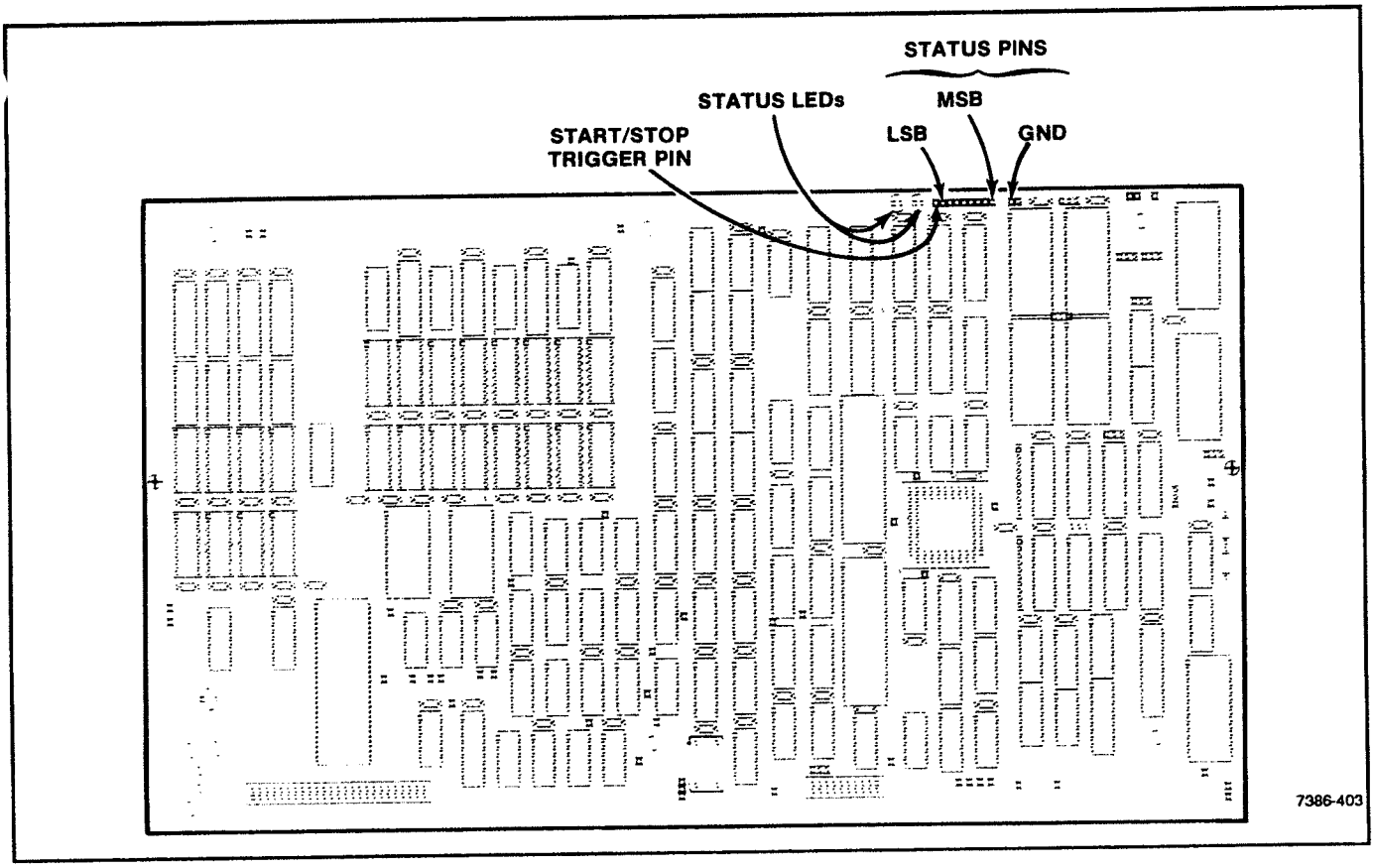


Figure 4-3. Display Controller board (A7) showing Display Kernel pin locations.

Digitizer Kernel

The Digitizer Kernel Diagnostics run at power-up as do tests in the other subsystems, and indicates test failures with status LEDs and pins on the Time Base Board (A6). The sequence of events for testing and for the status indicators is described above under Operation of Status LEDs and Pins. Unlike the other subsystems, the Digitizer Kernel has only one way to display error information.

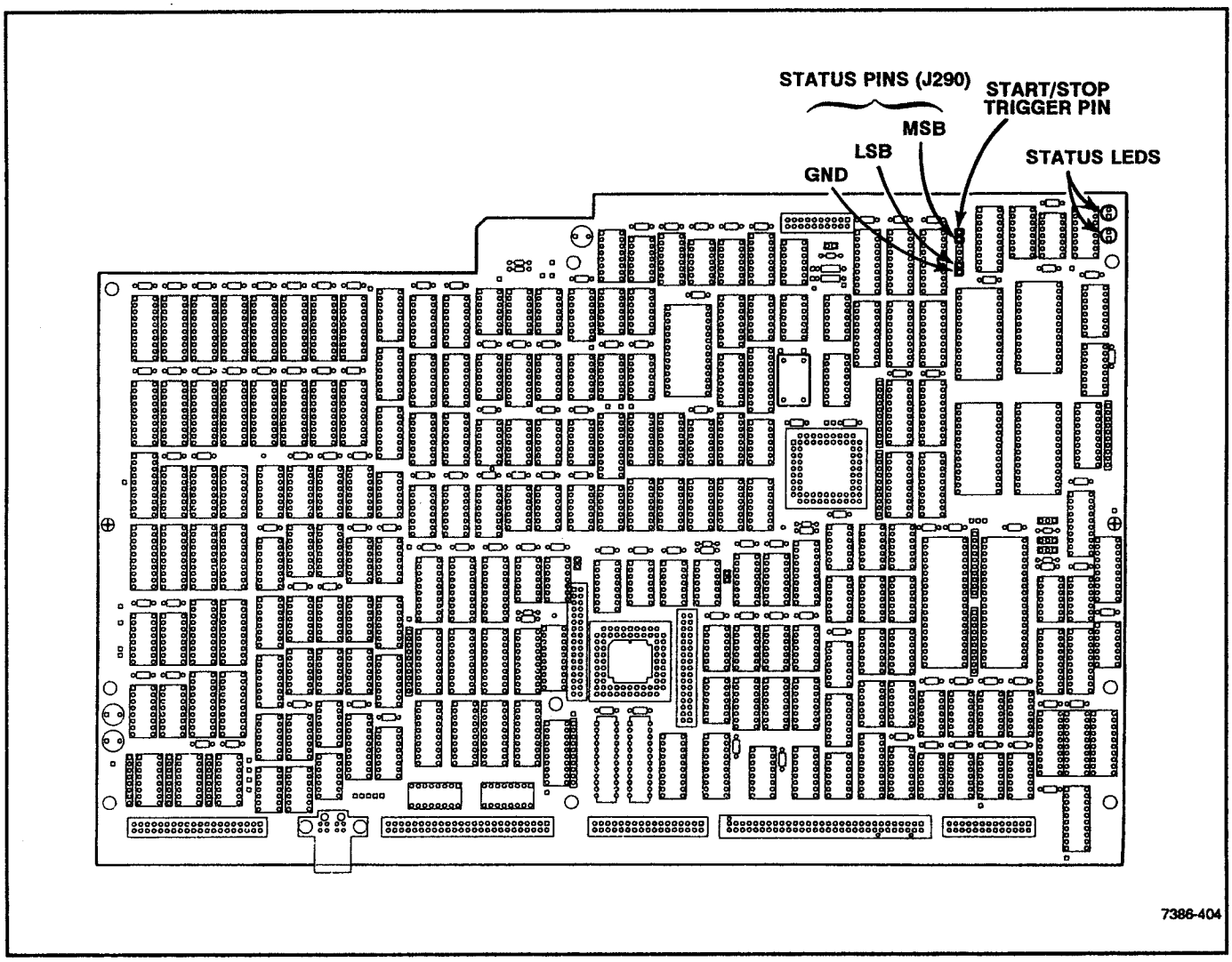


Figure 4-4 . Time Base board (A6) showing Digitizer Kernel pin locations.

When reading an error index code with a logic probe or oscilloscope, carefully note the pin location for each bit. On the status pins, a TTL high is a logic one and a TTL low is a logic zero. The LSB and next three pin/bits make up the low-hex digit. The MSB is the only bit used for the high hex digit, so the digit will be one or zero. Pins J290-2 (LSB) to J290-5 make up the low hex digit and pin J290-6 (MSB) makes up the high hex digit.

Power-Up Mainframe Subsystem Communications

The Subsystem Comm (Communications) tests run after the Kernel Diagnostics to verify communication lines between the Executive and the Display and between the Executive and the Digitizer. Each subsystem kernel's communications test, except for the Executive, is documented as the last Kernel Diagnostics routine. The Display's Executive Comm test (F4*hex*) and the Digitizer's Executive Comm test (16*hex*) are the last test descriptions under their respective Kernel Diagnostics Test Descriptions.

Executive Subsystem Routines

The Executive side of the communications tests are actually the first part of two high-level Self Tests described in the MainFrm Comm area test descriptions for the Display (E561X) and Digitizer (E562X). These routines are in the Subsys Comm block. The communication routines have a "Subsystem Not Present" part followed by a "Subsystem Present" part. The "Not Present" part that runs at power-up is a low level routine that sends and receives 16 bit words to determine if any communication is possible. If the mainframe subsystems pass the first communications tests, the second part of the MainFrm Comm tests (i.e., Subsystem Present) is executed during Self Test Diagnostics to verify high-level block data transfers. Refer to the test descriptions mentioned in this discussion for detailed descriptions of the communications tests.

Indications of Subsystem Communication Failures

If the Display or Digitizer fails the first communications test, it will enter an echo mode in which it receives, inverts, and returns any data sent by the Executive. The status pins for the failed subsystem will indicate the Executive Comm test and the status LEDs will change state. However, the Start/Stop trigger pin will not toggle. The Executive will proceed into the Self Tests and label the failed subsystem as "Not Present". Review the Error Index description for the displayed Executive error index code to learn the actual failure mode. Refer to the Display and Digitizer Kernel sections and Display and Digitizer Exec Communication test descriptions for more information on the status pins and LEDs, and the Start/Stop trigger pin.

If the Display subsystem fails to communicate properly, it will also attempt to display the message:

Dsy Kernel Failure
Exec Communication

Troubleshooting A Failed Subsystem Communication Path

To troubleshoot a Subsys Comm test failure, select the Routine menu for the failed routine and set **Terse** and **Loop** "On" and **Stop on Err** "Off". Then select the failed routine and **Run**. Connect a logic analyzer to the data lines at the failed end and look for the series of data words listed in the Error Index portion of the test descriptions. Note that the first data pattern received by the Executive (after being inverted by the subsystem) that fails to match the expected pattern will cause the next pattern sent to the failing subsystem to be DEAD*hex* and not the next pattern in the Error Index list. The DEAD*hex* pattern is unexpected so the failing subsystem immediately enters the echo mode. The mainframe subsystem would also enter the echo mode automatically if the first pattern sent by the Executive was corrupted. After DEAD*hex* is sent, the Executive resumes sending the data words from the Error Index list. Check the data path to the subsystem and the data path back to the Executive to locate the corrupted data bit(s) or control signal line.

Troubleshooting Non-Functioning Subsystem Kernels

This section describes how to troubleshoot the essential circuitry necessary to run the Kernel Diagnostics in each subsystem. The following procedures provide assistance in troubleshooting circuit faults that keep a kernel processor from running the Kernel Diagnostic tests in the on-board ROM. With this type of fault, the kernel is unable to give any indication of the fault location with either status LEDs or error index codes on the status pins. Some of the following discussion may also help in troubleshooting a subsystem that is able to run Kernel Diagnostics. Refer to the earlier discussion for troubleshooting using the Kernel Diagnostics.

The troubleshooting procedures involve setting jumpers in the kernel circuitry that isolate a processor from the rest of the system and force it to loop on a no-operation instruction. This stimulates the kernel circuitry by performing read operations throughout a kernel processor's address range. Once the kernel circuit is setup, the test equipment recommended below is used, along with the Theory of Operation and schematics, to find the defective component. A special extender card is available for the Executive Kernel that raises it out of its card-cage for easy access and provides hardware stimulation for troubleshooting. The extender board can also be configured for use in troubleshooting the other card-cage boards for any Diagnostics troubleshooting.

Refer to the Service Reference Manual for information on board removal, component handling and replacement, adjustments required after board repair or battery replacement, and safety precautions. If circuit boards are repaired and need adjustment, refer to appropriate Checks and Adjustments section in the Service Reference Manual.

Troubleshooting the Executive subsystem is covered first, including a complete discussion on using the Kernel Extender Service Kit (067-1267-00). Next, troubleshooting the Display and Digitizer kernels is discussed. The two subsystem kernels are discussed together because their kernel circuitry is very similar.

Equipment Required

The following equipment or its equivalent is necessary when troubleshooting the mainframe subsystems' kernel circuitry.

- TEKTRONIX 067-1267-00 Calibration Fixture: Troubleshooting Aid Extender Card with Cables.
- TEKTRONIX P6401 Logic Probe.
- TEKTRONIX 1240 Logic Analyzer.
- TEKTRONIX 7904A Oscilloscope with 7A26 Dual Trace Amplifier, 7S14 Dual Trace Delayed Sweep Sampler, and 7B10 and 7B15 Time Bases, or equivalent.

Executive Kernel

The Executive Kernel circuitry resides primarily on the Main Processor Board (A17), which plugs into the card-cage mother board. The Main Processor Board has no facilities to stimulate the kernel circuitry, so a disabled Executive Kernel must be troubleshot with the Kernel Extender Service Kit. The service kit includes a special extender card that can be configured to exercise the kernel circuitry. The Main Processor Board has two jumpers and one connector that configure it to work with the extender card. Their settings are discussed under the Mode 1 and Mode 2 sections below. When troubleshooting the Executive Kernel, refer to the Theory of Operation and to schematic diagrams 23 and 24.

Read the Kernel Extender Card description below and follow the setup instructions to troubleshoot the Executive Kernel.

Kernel Extender Service Kit

The Kernel Extender Service Kit (067-1267-00) provides facilities for troubleshooting the Executive card-cage boards and the RS-232-C interface. Included in the kit are an extender card and the longer cables needed when using the the extender card. An RS-232-C loopback device is included in the kit to provide external feedback of test signals generated by Diagnostics. Use of the RS-232-C loopback device is discussed after the Extender Card.

The service kit contains the following:

Service Kit	067-1267-00
1 Extender/kernel troubleshooting board	380-8673-XX
6 Diagnostic Lead set cables	012-1144-00
1 50 pin 3M card-cage replacement cable	175-9919-00
1 50 pin 3M card-cage extender cable	175-9918-00
1 34 pin 3M card-cage replacement cable	179-0207-00
1 40 pin 3M card-cage replacement cable	179-0208-00
1 40 pin 3M card-cage extender cable	175-9917-00
1 8 pin/wire ribbon cable	174-0262-00
1 Edge card support guide	386-5418-00
1 RS-232-C Loopback Connector	013-0198-00
1 Ziplock circuit board pouch	

Using the Kernel Extender Card

The extender card raises the Executive card-cage boards above the card-cage to permit easy access for troubleshooting. It plugs into the Executive card-cage mother board to connect the system bus signals, which then become available on groups of test pins on the extender card, to the extended board. To use the extender card, remove the defective board and insert the extender board into the slot. Then plug the Executive board into the top of the extender card. For most boards, the overhead cables will need to be changed as discussed later under Overhead Cable Replacement.

The extender card can operate in either of two modes. The first is as a simple extender card that raises any of the Executive card-cage boards for troubleshooting with little or no performance degradation. The second mode allows the Executive kernel circuitry to be exercised for troubleshooting problems in which the Executive processor can not run its Kernel Diagnostics. Movable jumpers on the Extender card allow the card to be configured for either mode of operation. (see Fig. 4-5) Procedures for configuring the Extender card and the Executive boards are given in the Mode 1 and Mode 2 sections below. The jumpers and their functions are as follows:

J122 Used in conjunction with J320 to specify the number wait states. Jumper across pins 1-2 for one wait and across 2-3 for zero wait states.

J220 Jumper across pins 1 and 2 to latch the mother board addresses on connectors J100, J102, and J104.

J320 Puts the extender card in the kernel stimulus mode when jumpered across pins 1, 2, 3, and 4. Normally, the four-pin jumper is left on the two holding pins adjacent to pins 1 and 4. **Note:** This mode should be used only with the Main Processor Board and then only when it is configured as described below.

In either mode of operation, the state of any of the card-cage bus signals can be monitored on groups of test pins at the top left of the extender card. The diagnostic lead sets included in the service kit are designed for quick connection of these groups of test pins to a Tektronix 1240 logic analyzer. In addition, a jumper (J220) can be set to provide latched addresses on connectors J100, J102, and J104. The addresses are latched with the Executive kernel signal ALE. The pin-outs for the test connectors are given later in this discussion.

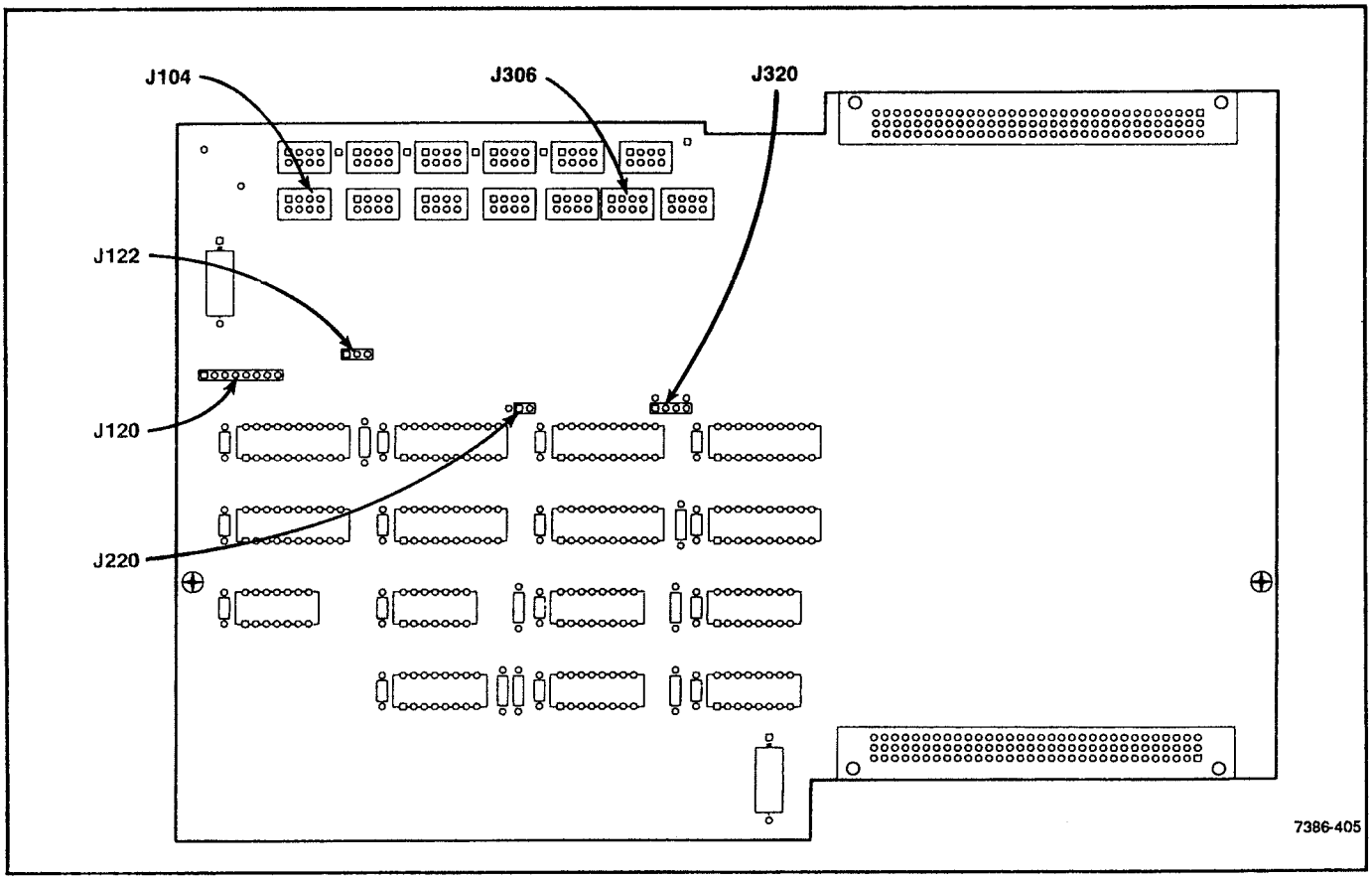


Figure 4-5. Extender Card showing configuration jumpers and connectors.

Mode 1 -- Passive Extender–As stated previously, Mode 1 operates the extender card as a simple, passive extender card that raises any of the Executive card-cage boards for troubleshooting with little or no performance degradation. A set of extender or replacement cables is provided with the Diagnostic Service kit to allow extension of Executive cards that are connected to other cards through overhead cables. This mode can be used when the Executive Kernel Diagnostics have detected a fault or when troubleshooting with the Extended Diagnostics. To operate as a simple extender card:

1. Set the following extender card jumpers:

   J320 Should be on the two holding pins adjacent to pins 1 and 4.

   J122 Either position.

   J220 Jumper pins 2 and 3 if latched addresses are desired on the extender card test pins (J100, J102, J104), otherwise jumper pins 1 and 2.

   J120 Must not be connected to the Main Processor Board.

2. Insert extender card in place of card-cage board to be troubleshot.

3. Insert card-cage board into extender board and connect appropriate overhead-cable replacements from the service kit.

Replace jumpers to original positions when troubleshooting is completed.

Mode 2 -- Active Kernel Exerciser–In Mode 2, the extender card exercises the Executive kernel circuitry. If the kernel has a serious problem, then this mode exercises the support circuitry (e.g., address, data, and control lines to essential ROM) to isolate the faults. This mode would typically be used only when none of the Kernel Diagnostic tests could run.

When both boards are properly configured and connected (see steps below), the extender card will perform sequential read operations throughout the kernel processor's address space. Address lines A1 through A23 are driven by a binary counter on the extender board. To get control of the 80286 buses the extender card asserts HOLD (J120-2) and waits for the 80286 to tri-state its outputs and assert HLDA (J204-2). When HLDA is received the extender card pulls S0(L) (J120-6) and COD/INTA (J120-7) low and generates S1(L) (J120-5) to perform memory reads. M/IO (J120-8) is held high to select the memory address space. The condition of the SRDY(L) signal line determines when S1(L) is generated. KREADADD (J120-4) is sent to the Main Processor Board to force the address buffers to reverse their drive direction while the extender card addresses are generated. The signals HLDA, SYS_CLK, SYS_RESET, and SRDY(L) that are generated by the Main Processor Board must be working for the extender card to function in this mode. To exercise the Executive Processor kernel:

1. Set the following extender card jumpers:

   J320 Jumper across pins 1 to 4 .

   J220 Jumper pins 1 and 2 if latched addresses are desired, otherwise jumper pins 2 and 3.

   J122 Jumper pins 2 and 3. This sets the number of wait states required to access Executive system ROM to zero. (Pins 1 and 2 select one wait state.)

   J120 Connect cable Tektronix part # 174-0262-00 which will later be connected to the Main Processor Board.

2. Replace Main Processor Board with extender card.

3. Remove Main Processor Board jumper J660 to disable SRDY (see Fig. 4-6).

4. Insert Main Processor Board into extender board.

5. Remove Main Processor Board jumper on J800,pins 2-3 and connect the 8-wire ribbon cable from J120 on the extender card (see Fig. 4-6).

6. Connect appropriate overhead-cable replacements from the service kit.

Replace jumpers to original positions when kernel troubleshooting is completed.

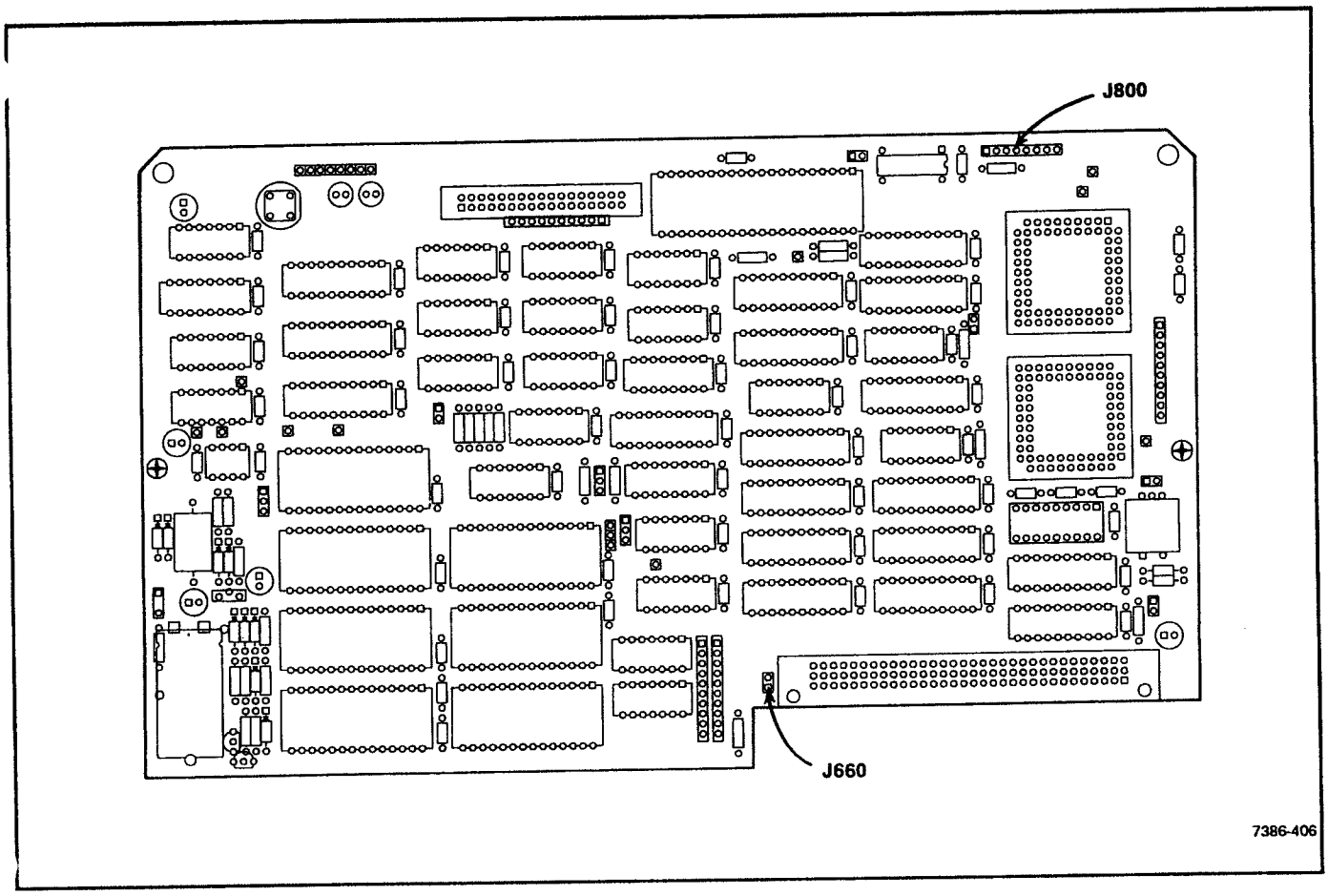


Figure 4-6. Main Processor Board (A17) showing Executive Kernel Test jumpers.

Overhead Cable Replacement–Executive card cage boards require different replacement cables for overhead cable connections. Table 4-3 defines the cables to use for raised Executive cards. The cables are identified by their Tektronix part number.

TABLE 4-3
Overhead Cable Replacement Guide

Executive Board	Extend Cable	Replace Cable	Extended Cable	Replacement Cable
Main Processor		175-9814-00		179-0207-00
Standard I/O	175-9915-00	175-9814-00 175-9854-00	175-9917-00	179-0207-00 179-0208-00
MMU	175-9808-00	175-9809-00	175-9918-00	175-9919-00
Compressor	175-9810-00	175-9809-00	175-9918-00	175-9919-00
Memory				

Extender Card Test Points–Four test points allow the mother board power and ground lines to be probed. The test points are:

TP100 Ground
TP200 +15 V
TP204 -15 V
TP300 Ground
TP400 +5 V

Extender Card Connectors–Two types of test connectors with square pins are available on the extender card. All connectors but one are for probing the mother board address, data, and control signals. Pinouts for these test signal "J" connectors are given below. Connector J120 connects by cable to the Main Processor Board connector J800 when in the kernel stimulation mode (Mode 2). This connection should only be made when both boards are properly configured. See Mode 1 and Mode 2 jumper settings earlier in this discussion.

The "J" signal connector pin numbering is:

1 3 5 7
2 4 6 8

Table 4-4 gives an alphabetical listing of all signal names and their test-connector pin numbers. Connector pins not included in Table 4-4 have no connection to any mother board or extender board signal.

TABLE 4-4
Extender Board Test Signal Connector Pinout

Signal Name	Connector-Pin Number
A0	J302-1
A1	J302-2
A2	J302-3
A3	J302-4
A4	J302-5
A5	J302-6
A6	J302-7
A7	J302-8
A8	J308-1
A9	J308-2
A10	J308-3
A11	J308-4
A12	J308-5
A13	J308-6
A14	J308-7
A15	J308-8
A16	J200-1
A17	J200-2
A18	J200-3
A19	J200-4
A20	J200-5
A21	J200-6
A22	J200-7
A23	J200-8
A24	J206-1
A25	J206-2
A26	J206-3
A27	J206-4
A28	J206-5
A29	J206-6
A30	J206-7
A31	J206-8

TABLE 4-4 (cont.)
Extender Board Test Signal Connector Pinout

Signal Name	Connector-Pin Number
ALE	J202-2
BHE(L)	J202-6
COD/INTA	J306-3
D0	J300-1
D1	J300-2
D2	J300-3
D3	J300-4
D4	J300-5
D5	J300-6
D6	J300-7
D7	J300-8
D8	J304-1
D9	J304-2
D10	J304-3
D11	J304-4
D12	J304-5
D13	J304-6
D14	J304-7
D15	J304-8
DEN	J202-7
DIAGNSIG(L)	J202-3
DT/R	J202-8
EPON	J204-1
HLDA	J204-2
INT0(L)	J204-5
INT1(L)	J204-3
INT2(L)	J106-1
INT3(L)	J106-2
INT4(L)	J106-3
INT5(L)	J106-4
INT6(L)	J106-5
INT7(L)	J106-6
INTA(L)	J202-1
IORC(L)	J204-7
IOWC(L)	J204-4
*LA1	J102-1
*LA2	J102-2
*LA3	J102-3
*LA4	J102-4
*LA5	J102-5
*LA6	J102-6
*LA7	J102-7
*LA8	J102-8
*LA9	J100-1
*LA10	J100-2
*LA11	J100-3
*LA12	J100-4

TABLE 4-4 (cont.)
Extender Board Test Signal Connector Pinout

Signal Name	Connector-Pin Number
*LA13	J100-5
*LA14	J100-6
*LA15	J100-7
*LA16	J100-8
*LA17	J104-1
*LA18	J104-2
*LA19	J104-3
*LA20	J104-4
*LA21	J104-5
*LA22	J104-6
*LA23	J104-7
M/IO	J306-4
MRDC(L)	J202-4
MWTC(L)	J204-8
PCLK	J306-7
S0(L)	J306-6
S1(L)	J306-5
SRDY	J202-5
SYSCLK	J306-8
SYSRESET	J104-8

*Only active when J220 is jumpered across pins 1 and 2.

Using the RS-232-C Loopback Device

The RS-232-C loopback device aids in troubleshooting the circuitry from the Executive Processor out to the standard RS-232-C port. Normally, the primary means of diagnostic test control is through the touch screen display. But, should the circuitry that controls the touch screen display be faulty, a test terminal can control the Diagnostics through the RS-232-C port. If the RS-232-C circuitry is also faulty, a hardware strap on the Standard I/O Board (A14) can be set to force the diagnostics to continually exercise the circuitry from the Executive out to the loopback device. This provides a troubleshooting aid in establishing RS-232-C control of the diagnostics. The loopback test can also be invoked through the Extended Diagnostics menus.

To troubleshoot the RS-232-C system at power-up (using Kernel Diagnostics), power down the instrument and remove the Standard I/O Board (A14). On the right-hand side of the board are the Diagnostics Options Jumpers labeled J710. The bottom two jumpers control Kernel Diagnostics operation and are usually both set to the right side. For this test, set the second one up from the bottom to the left. This will cause the instrument to run the kernel tests then loop on the RS-232 stimulus routine. Next, connect the loopback device on the back-panel RS-232-C port connection When the instrument is powered up the RS-232 loopback routine will run after all Kernel Diagnostics tests have run and passed. The loopback routine is described in the Executive Kernel Test Descriptions under the Std RS232 External Loopback (04) test. See the Diagnostic Options Jumpers section for further information on the options jumpers.

To run the test from Extended Diagnostics, invoke the test from the External I/O block in the RS232 area. See the RS232 Extern Loop (E425X) test description in the Executive Self Test/Extended Test Descriptions.

Display and Digitizer Kernels

The Display and Digitizer kernel circuits are quite similar, consisting of an 80186 microprocessor, its clocks, ROM and RAM, and the buses necessary to communicate with them. The Display Kernel resides on the Display Controller Board (A7) and the Digitizer Kernel resides on the Time Base Board (A6). Refer to the appropriate theory of operation discussion and the kernel schematic diagram for the Display <33> or Digitizer <16> when troubleshooting the kernel circuitry.

Setting three jumpers for either kernel processor puts it into a test mode in which it performs read cycles while looping through a certain address range. This mode is useful in helping to determine faults that prohibit the processor from executing Kernel Diagnostics code. When the kernel circuitry is defective, both diagnostic status LEDs on the associated board will be lit at power-up (by a hardware reset) and remain on. Normally, one of the LEDs would go off when the first Kernel Diagnostic test started.

Setting Kernel Test Jumpers

The usual position of jumpers is label NORM and the test position,TEST. The jumpers that should be set to TEST are as follows:

Display Controller Board	J6, J5, J11 <33>
Time Base Board	J30, J40, J50 <16>

The first jumper (i.e., J6 or J30) disables the ROM data buffers that drive the processor data lines on read cycles. The other two test jumpers (i.e., J5 and J11 or J40 and J50) work in conjunction with pull-up resistors on the data lines to cause the processor to read FDFD*hex*, which it interprets as two "STD" (byte) instructions. This is a "no operation" instruction in that the processor does not perform any external bus activity. No other jumpers need be or should be moved to initiate the kernel test loop.

There are several other signal points and jumpers that should be checked while running the test. On the Display Controller Board check the following:

- TP30 - MPUCLK, 8 MHz square wave (noisy).
- TP39 - RESET, high (5 V).
- J7 - Should be in 256 position, jumper pins 1-2.
- J10 - Should be in NORM position, jumper pins 1-2.
- J12 - Should be in 512 position, jumper pins 1-2.
- J13 - Should be in 512 position, jumper pins 2-3.
- U612-1 - A16, eight pulses, 1 µs apart with 4 ms between sets of pulses.
- U612-27 - A15, 2 MHz square wave.

On the Time Base Board check the following:

- TP14 - RESET, high (5 V).
- TP34 - CLKOUT, 8 MHz square wave (noisy).
- U283-1 - A16, eight pulses, 1 µs apart with 4 ms between sets of pulses.
- U283-27 - A15, 2 MHz square wave.

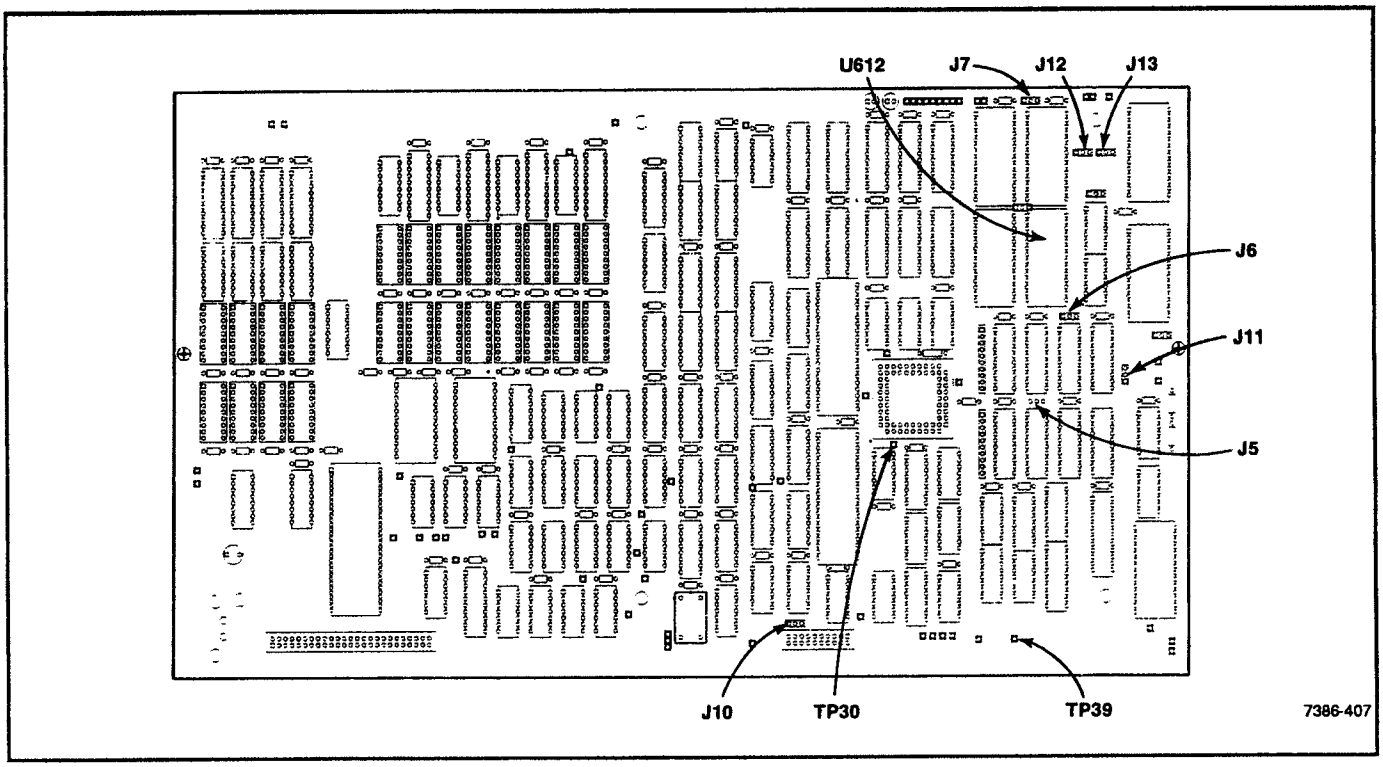


Figure 4-7. Display Controller Board (A7) showing Display Kernel Test jumpers.

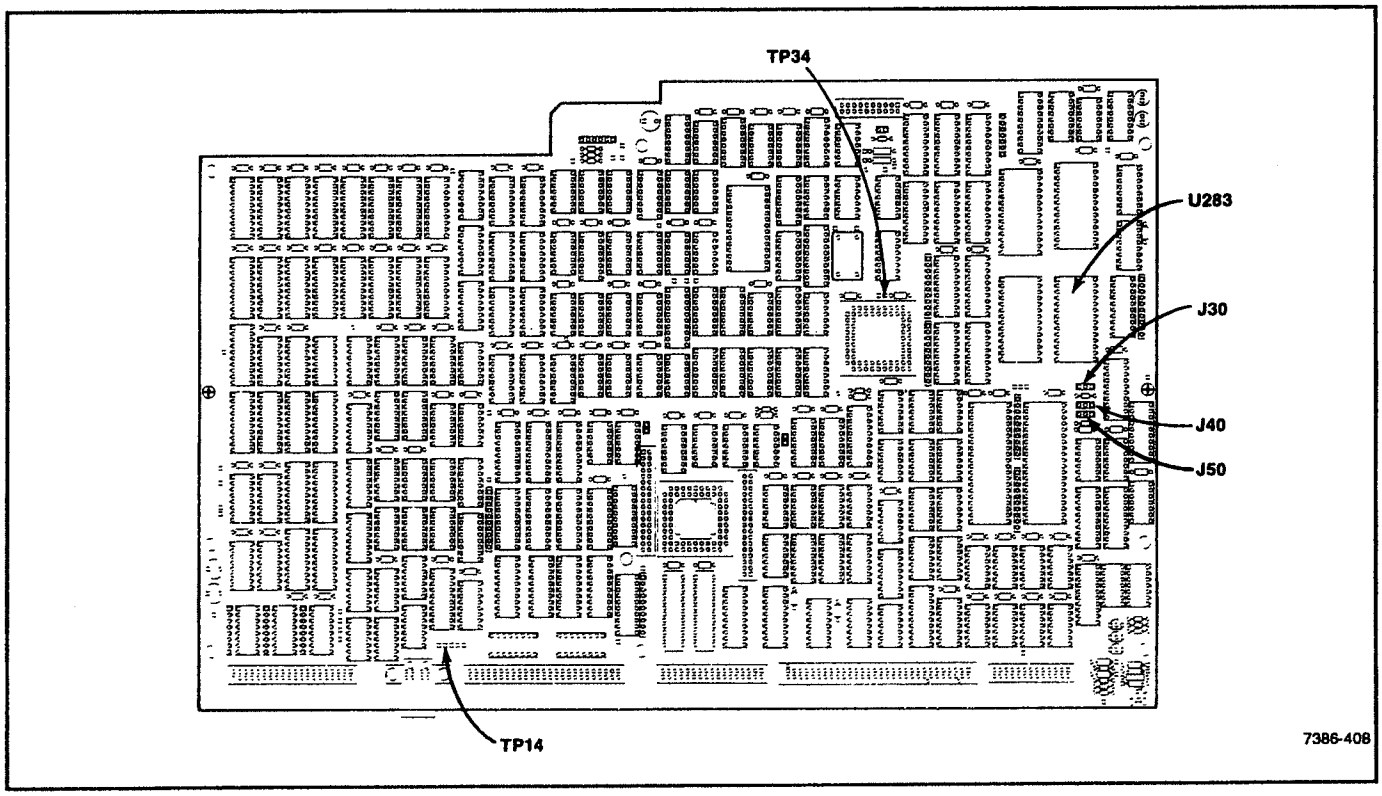


Figure 4-8. Time Base Board (A6) showing Digitizer Kernel Test jumpers.

Kernel Test Address Sequence

On power-up reset, the 80186 processor tries to read instructions from ROM, starting at address FFFF0*hex*. Instead of reading the ROM, the processor reads and executes the two "STD" instructions (FDFD*hex*), then increments the address lines to the next word address (in this case FFFF2*hex*). This read data and increment-address process then repeats indefinitely. The sequence of hex addresses that the 80186 processor will cycle through is as follows:

FFFF0	**Starting address**
FFFF2	
FFFF4	
FFFF6	
FFFF8	
FFFFA	
FFFFC	
FFFFE	
00000	
00002	
00004	
00006	
•	
•	
•	
0FFEE	**Jump to Starting address**

The lower 16 address lines increment through all values from 0000 - FFFE*hex*, while the upper four address lines go high only during eight accesses to the power-up reset vector (i.e., FFFF0-FFFFE*hex*). The non-linear address sequence is due to the internal programmable chip selects in the 80186 that assume default values on power-up. This causes some of the address range to be undefined and inaccessible.

If the processor does not cycle through the addresses, check clock and reset signals to the processor and check the data lines for the FDFD*hex* pattern. Connect an oscilloscope to the upper four address lines (only 2 are readily accessible on the boards) and use holdoff to find the pattern of eight, periodic positive pulses. This address sequence is easier to see with a 1240 logic analyzer (see the following logic analyzer set-ups).

Logic Analyzer Set-up for Display

Address Lines:	MA0 - MA7 on U614 <33> MA8 - MA15 on U620 MA16 - MA17 on U526
Clock:	ALE(H) on U526 (rising edge) <33>

Note that since MA18/MA19 is not readily accessible, the addresses acquired with the analyzer will be missing the two upper address bits. Hence, addresses such as 3FFF0*hex* will be seen instead of FFFF0*hex*.

Logic Analyzer Set-up for Digitizer

Address Lines	A0 - A7 on U252 <16> A8 - A15 on U253 A16 - A17 on U232
Clock:	ALE(H) on U232 (rising edge) <16>

Note that since A18/A19 is not readily accessible, the addresses acquired with the analyzer will be missing the two upper address bits. Hence, addresses such as 3FFF0*hex* will be seen instead of FFFF0*hex*.

Self Test Diagnostics

As stated previously in the Overview section, Self Test Diagnostics run automatically on power-up or when invoked by the user from normal operation. When Self Test begins, the message **Self Test in Progress** is shown on the display. During the majority of time that Self Test runs, subsystems perform independent testing in parallel.

Error index codes generated from Self Test failures are identical to the error index codes generated from the Extended Diagnostics. See the next section for further information on how the error index codes are generated and what they mean.

NOTE

Pressing front panel buttons, turning knobs, or touching the Touch Panel while Self Test Diagnostics are running may cause a diagnostic failure.

Extended Diagnostics

Extended Diagnostics provides the means for examining and exercising individual tests or groups of tests, primarily for troubleshooting purposes, but also as a way of increasing the user's confidence level of instrument functionality.

A Self Test failure or selecting Extended Diagnostics from the Utility menu produces a menu display showing all major circuit blocks and the plug-in units. When displayed as a result of a Self Test failure, the status of each test block will be shown next to its name. When Extended Diagnostics are invoked from the Front Panel, no tests run initially, hence there is no test status displayed.

Error Index Overview

The format of the error index codes is based on the Extended Diagnostics menu structure. The Extended Diagnostics menus (described in detail later in this section) are in a three level hierarchy with the Block menu at the highest level. Each circuit block name in the Block menu can be individually selected and tested. A selected circuit block is broken into a number of parts or circuit areas in the Area menu, the second level. Touching the Area label at the bottom of the menu displays the Area Menu for the selected block. Each circuit area has a Routine menu, the third level, associated with it that has one or more selectable routines. Routines are the smallest test units that can be selected and run. This Block, Area, and Routine menu hierarchy is used in generating the error index codes and organizing the test descriptions.

Extended Diagnostics error index codes are five digit codes whose first character indicates the subsystem or plug-in unit tested. The last four digits are hexadecimal (hex) numbers that indicate the block, area, routine, and specific failure mode. For example, E2321 is decoded as follows:

E	Subsystem - Executive
2	Block name - Front Panel
3	Area name - Soft Keys
2	Routine name - Column Open
1	Failure Identity - specific failure mode

The subsystem character of an error index code will be one of the following:

E	Executive
D	Display
G	Digitizer
L	Left
C	Center
R	Right

The failure mode digit (i.e., the right-most digit) of an error index code points to a discussion of a particular type of circuit failure. Failure mode discussions are located at the end of each test description in numerical order. Each test routine can have up to 15 (1-F*hex*) failure modes and associated discussions.

Mainframe Error Index Codes

Mainframe Self Test/Extended Diagnostics error index codes refer to test descriptions which are divided into subsystem groups under the headings Executive (E), Display (D), and Digitizer (G). The test descriptions under each subsystem heading are in numerical order. For instance, the Executive subsystem test descriptions start with test number E111X. Next in order is E112X, E113X, ..., E583X. Likewise, the Display tests start with D111X and the Digitizer tests with G111X. This order of subsystem headings and the numeric order of test descriptions reflects the actual test sequence for that particular subsystem.

Plug-in Unit Error Index Codes

Plug-in unit Self Test/Extended Diagnostics error index codes refer to test descriptions in the plug-in unit service manuals. Each type of plug-in unit (e.g., 11A34 or 11A52) has a unique set of Diagnostics and an associated set of test descriptions.

When Extended Diagnostics are entered from a Self Test failure, test status for plug-in units in the menu displays may show a "****" status. In this particular case only, and for plug-in units only, this indicates that the plug-in Self Tests actually did execute and passed successfully. Plug-in units do not save test results from Self Tests which ran successfully, and therefore have no test status to provide the mainframe for display on entry into Extended Diagnostics. The plug-in units, however, do save the first, but only the first, failure encountered during Self Tests. The first failure will therefore cause an error index code to appear in the plug-in test status, but there may be other plug-in Self Test failures not visible until the plug-in unit tests are executed again in Extended Diagnostics.

Extended Diagnostics Menu Displays

The Block menu is the top level of a three level menu structure. In the second-level Area Menu, the selected circuit block is divided into circuit areas. The Routine menu, which is the lowest level, shows the routines that make up the selected area. Routines are the smallest test unit that can be individually run. Figure 4-9 shows an example of the top-level Block menu. The menu features discussed here also appear on a remote terminal used in the front panel emulation mode.

Menu items are selected or started by touching their labels on the screen, just as in normal instrument operation. For example, to view the Area menu for a selected block, touch the Area label in the control section at the bottom of the Block menu. Most menu items are highlighted when selected. When Run is selected, the highlighted test will be performed. Touching Quit, any front panel button, or any point on the screen will halt testing after the current routine finishes.

The characters and numbers in parenthesis, preceding the selectable menu items, are the single key commands used on a remote terminal in the front panel emulation mode. Their use is covered in the Remote Diagnostics discussion, later in this manual.

Diagnostic menus are divided into three functional sections. The top two lines are for prompt and warning messages. The remainder of the menu except the bottom fourth is used for test status information. The bottom section of the menu is the test-control area. The common features and functions of each menu section will be discussed in the following Block menu description. Then, the features unique to the Area and Routine menus will be discussed.

EXTENDED DIAGNOSTICS

BLOCK	INDEX	FAULTS
a) Exec Control	pass	
b) Front Panel	E2211	1
c) Internal I/O	pass	
d) External I/O	pass	
e) Subsys Comm	pass	
f) Dsy Control	pass	
g) Video Gen	pass	
h) Dig Control	pass	
i) Timebases	pass	
j) Points Acq	pass	
k) Triggers	pass	
l) Pts/Addr Gen	pass	
m) L_11A34	****	
n) Center Slot	????	
o) Right Slot	????	

(1) Block	(2) Area	(3) Routine	(E) Exit
Front Panel	Hard Keys	Open	Diagnostic
(p) Loop	(t) Terse	(x) All	(s) Stop on Err	(r) Run
Off	Off	Off	Off	Stopped

Figure 4-9. Block Menu Example

Block Menu

The Block Menu (see Figure 4-9) is the highest level Extended Diagnostics menu and, as such, shows the status of all tests at a glance. The plug-in unit names at the bottom of the test status section may be different, depending on the configuration of the instrument.

Prompt/Warning Section–The two top menu lines display messages that prompt you to do something or warn you about a problem in operation. For instance, pressing the HARDCOPY button when no printer is connected or when the printer cannot print (e.g., off line, no paper, etc.) will cause the following warning to be displayed:

Hardcopy absent or off line.

Warning or error messages are displayed as required.

Warning or error messages (those that are shown as selectable or highlighted) can be removed by touching the screen. Other prompt-type messages (those that are not shown as selectable), which are associated with a selected routine, can only be removed by selecting a different diagnostic test.

Test Status Section–The center section of the display has three columns, labeled BLOCK, INDEX, and FAULTS, that provide the following information:

BLOCK This column shows the major circuit blocks in the order in which they are executed and any installed plug-in units. The plug-in unit names are dependent on the instrument configuration. Selecting an entry causes its name to appear below the (1)Block label in the control section of the menu. Selecting (r)Run would execute this test block.

INDEX This column gives the status of each test block as one of the following:

- **pass** The test has run and had no failure.
- PXXXX The test has failed with this error index code, which is a pointer to a test description. See the Error Index Overview discussion earlier in this section.
- "****" The test has not yet been executed, but, when run, it will give pass/fail status.
- "----" An interactive test requires some setup (e.g., install the RS-232 loop-back fixture) and, when run, it will give pass/fail status.
- (blank) The test requires some interaction with the instrument such as that required for the Front Panel Soft Keys test. No pass/fail status is generated when run.
- "????" The hardware option named by the circuit block was not installed or the communication path to a subsystem (e.g., a plug-in unit) was not working.

FAULTS This column gives the total number of failures that have occurred in the circuit block. The count is increased by one for each failed routine. Only test routines that have run and failed will increment the fault count.

Control Section–The bottom fourth of the menu provides a number of functions for controlling which test(s) will run and the operating mode while the test is running. Also, the different menu levels and Exit Diagnostics can be selected.

The control functions and their definitions are as follows:

(1) Block Selects the Block Menu for display. Shows the currently selected circuit block.

(2) Area Selects the Area Menu for display. Shows the currently selected circuit area or the first area to have a failure in the block.

(3) Routine Selects the Routine Menu for display. Shows the currently selected routine or the first routine to fail in the block/area.

(E) Exit Causes the Extended Diagnostics mode to terminate and returns the instrument to normal operation. Faults that disable the instrument will cause Exit to be non-selectable.

(p) Loop Toggles On and Off. When On, the selected test(s) run in a loop continuously until stopped. The status shows the current number of completed loops when Terse is Off. When Terse is On, the count remains at zero until the test is stopped, then the number of completed loops is displayed. Running more than 65534 loops cause the overflow status, 65535+. The Loop status is set to zero when Run is selected. When set to Off, a selected test that is run will be performed once only.

(t) Terse Toggles On and Off. When On, the display status is not updated while tests are running. When testing stops, all screen status is updated. This provides the fastest possible routine execution for troubleshooting. When Off, status is updated as tests are executed.

(x) All Toggles On and Off. When On, all tests in the current menu are selected for execution. When Off, only the selected test in the current menu will execute.

(s) Stop on Err Toggles On and Off. When On, testing stops as soon as the first failure occurs. When Off, all selected tests will run to completion or continue to loop if Loop is On.

(r) Run/(q) Quit Starts the currently selected test(s) when Run is selected and changes the status from Stopped to Running. While running tests, the control label changes to Quit. Touching Quit stops testing at completion of the current routine and changes the status to Stopped. Testing will also stop when any point on the screen is touched or a front panel button is pressed.

Area Menu

The Area Menu (see Figure 4-10) is the second level Extended Diagnostics menu. Touching the Area Menu selector at the bottom of the screen displays the Area Menu for the currently selected block. The Area Menu's Prompt/Warning Section and Control Section operate just as they do for the Block Menu. Refer to the Block Menu description for more information.

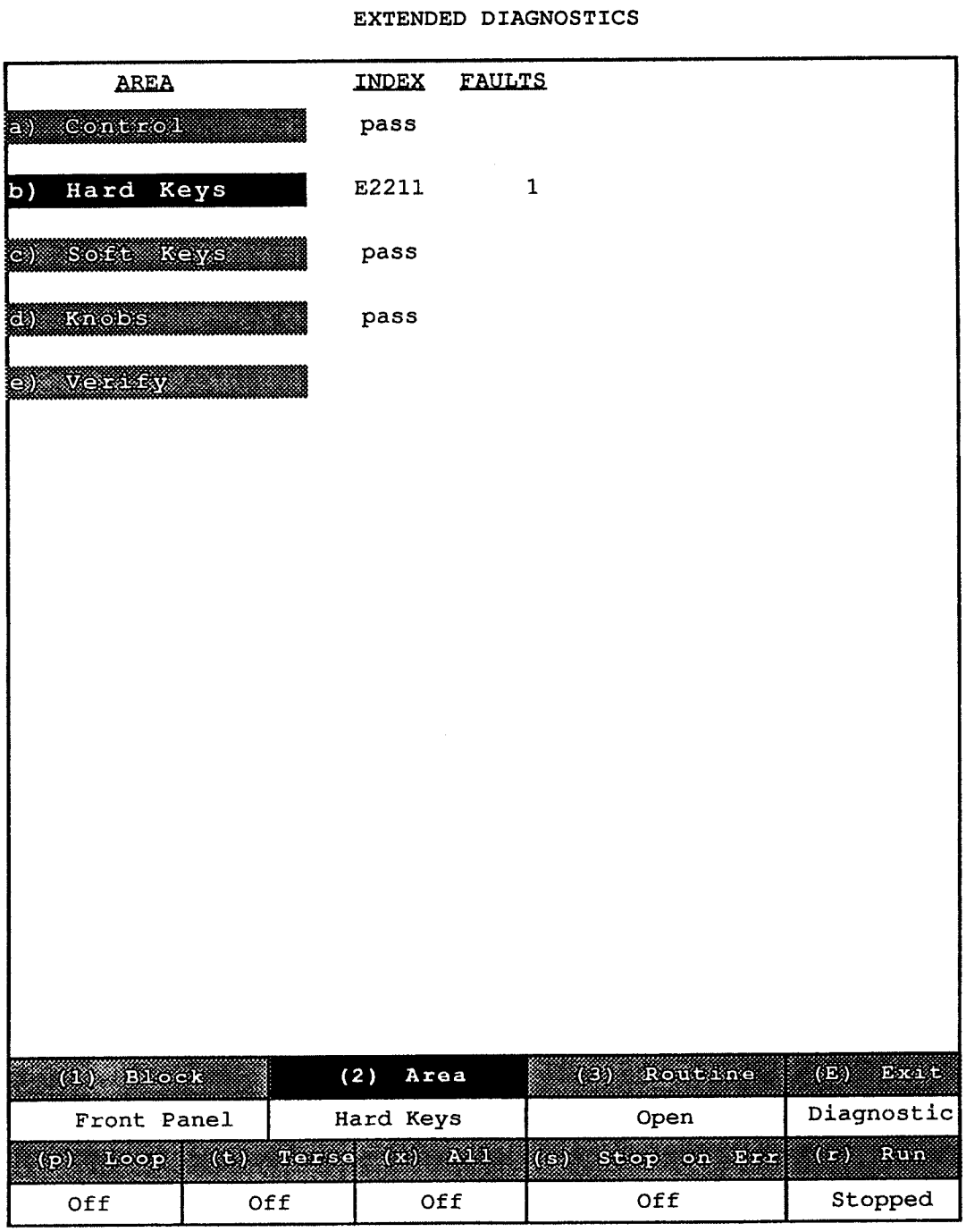


Figure 4-10. Area Menu Example

Test Status Section The center section of the display has three columns, labeled AREA, INDEX, and FAULTS, that provide the following information.

AREA This column shows the circuit areas for the selected block in the order in which they are executed. Selecting an entry causes its name to appear below the (2)Area label in the control section of the menu. Selecting (r)Run would execute this test area.

INDEX This column gives the status of each test area. The status types are the same as those in the Block Menu. Refer to the Block Menu discussion.

FAULTS This column gives the total number of failures that have occurred in the circuit area. The count is increased by one for each failed routine. Only test routines that have run and failed will increment the fault count.

Routine Menu

The Routine Menu (see Figure 4-11) is the third level Extended Diagnostics menu. Touching the Routine Menu selector at the bottom of the screen displays the Routine Menu for the currently selected area. The Routine Menu's Prompt/Warning Section operates just as it does for the Block Menu. Refer to the Block Menu description for more information.

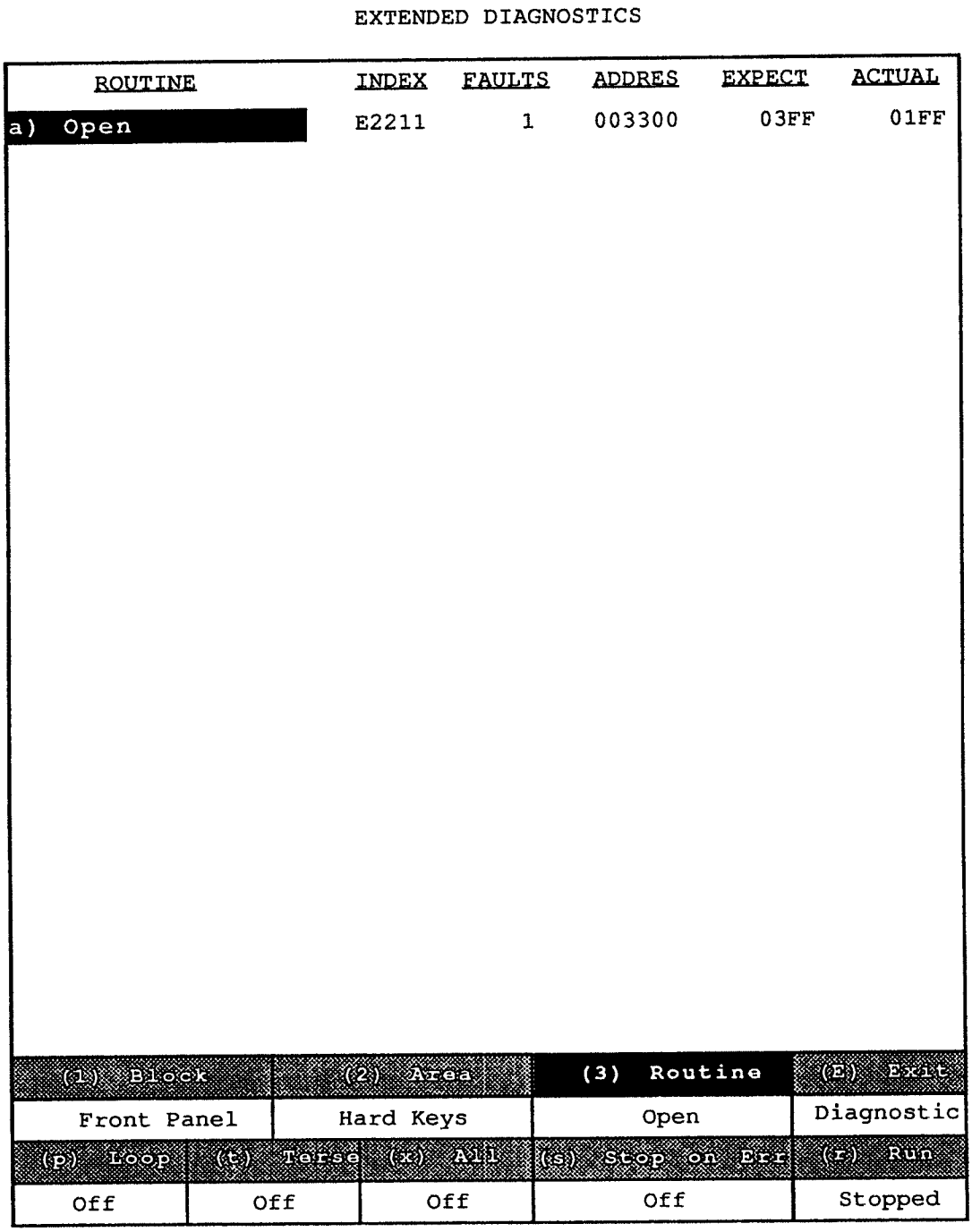


Figure 4-11. Routine Menu Example

Test Status Section–The center section of the display has six columns, with the first three labeled ROUTINE, INDEX, and, FAULT. Next are three test result columns giving either the test address, expected data, and actual data; or the acceptable data minimum, acceptable data maximum, and the actual data. This low-level information is useful for troubleshooting the failure. The columns provide the following information:

ROUTINE Shows the routines for the selected area in the order in which they are executed. Additionally, some interactive routines may be listed with their Index status shown as "----" or blank. Selecting an entry causes its name to appear below the (3)Routine label in the control section of the menu. Selecting (r)Run would execute this test routine. Interactive routines must be selected individually to be executed. When an interactive routine is selected, the (p)Loop and/or (x)All selectors will become non-selectable automatically, as this would be an invalid combination of control functions. See the Control Section discussion below for further information on interactivity between the Control selectors and the Test Status routine selectors.

INDEX Gives the status of each test routine. The status types are the same as those in the Block Menu. Refer to the Block Menu discussion.

FAULTS Gives the number of failures for the routine. The number is reset to zero when Run is selected. Exceeding 65534 failures causes the overflow status, 65535+, to be displayed.

ADDRESS Normally gives the internal memory or I/O address to which test data is written or read.

EXPECT Gives the data expected by the test as a result of test stimulus. See the appropriate test description for a listing of test stimuli.

ACTUAL Gives the actual data received by the test as a result of test stimulus.

MIN Gives the minimum acceptable limit for the Actual test result data.

MAX Gives the maximum acceptable limit for the Actual test result data.

Control Section-The following section discuss special considerations which apply only to the Routine menu level.

Interactivity With Test Status Selectors–On entry into the Routine menu, one or more of the routine selectors in the Test Status Section (specifically those that are interactive in some way), or the (p)Loop and/or (x)All selectors in the Control Section, may be non-selectable so that invalid test conditions can not be attempted. Therefore, it may be necessary to set the Loop and/or All selectors Off to allow interactive routines in the Routine menu to be selectable. Similarly, it may be necessary to select a different routine in the Test Status Section (i.e. one that is not interactive) to allow the Loop and/or All selectors in the Control Section to be selectable.

Special Combinations - Cycle & Cycle-Halt ModesTwo combinations of the Control Section test execution modes (i.e. Loop, Terse, All, and Stop on Err) significantly increases the test execution loop time for most routines. These two combinations, referred to as Cycle and Cycle-Halt modes for short, help identify intermittent faults quicker and are beneficial when troubleshooting with an oscilloscope (by providing faster test pattern stimulus). Table 4-5 shows the appropriate test execution mode set-ups for Cycle and Cycle-Halt modes.

TABLE 4-5
Cycle and Cycle-Halt Mode Set-ups

	Cycle	Cycle-Halt
Loop	On	On
Terse	On	On
All	Off	Off
Stop on Err	Off	On

As can be seen, the only difference between Cycle and Cycle-Halt is that Stop on Err is On for Cycle-Halt. This causes test execution to terminate on the first failure.

Normally, when only Loop is On, the Executive processor commands a subsystem (and itself) to execute the selected test a single time, after which the test results for that single test execution is displayed, before commanding the subsystem to execute the test again. In other words, looping is being done through the Executive processor. However, in Cycle and Cycle-Halt modes, the Executive processor commands the subsystem to loop *locally*, and for it to keep track of the number of loops it is executing and the number of faults that it has found up until the time test execution is terminated (via a test failure in Cycle-Halt mode or the normal front panel termination mechanisms in Cycle mode). Only after test execution is terminated will the test results be displayed. The net effect of this *local* looping in the subsystem is the speed-up of the test looping or cycling.

Plug-in Units Cycle & Cycle-Halt Modes—When plug-in unit routines are tested in the Cycle or Cycle-Halt modes, the following message appears in the Prompt/Warning Section of the display:

See diagnostics manual for plug-in test termination.

Test execution termination of plug-in unit routines in the Cycle or Cycle-Halt modes can not done from the touch screen or mainframe front panel buttons. Instead, plug-in unit buttons of the plug-in being tested must be used to terminate the testing. For example, to terminate testing of 11A32's or 11A34's, the CH 1 and CH 2 buttons must be pressed simultaneously.

Hardcopy Operation

The Hardcopy function of the instrument is available in a simplified form during Extended Diagnostics. All menu items, including the status of each test, are depicted on the hardcopy. Menu items that are selected are printed in boldface. An Epson compatible printer must be connected to the rear-panel PRINTER port with a standard Centronics parallel interface cable. For printer connection and configuration information, see External Interface Connection in Section 1 of the 11401/11402 User's Reference Manual.

To make a hardcopy of the screen display, press the front panel HARDCOPY button. The hardcopy of the displayed menu will include the following additional information at the top:

1. Instrument Type; either **11401 DIGITIZING OSCILLOSCOPE** or **11402 DIGITIZING OSCILLOSCOPE**;
2. Current date and time; and
3. Identification numbers for the mainframe and installed plug-in units.

Pressing the HARDCOPY button when no printer is connected or when the printer cannot print (e.g., off line, no paper, etc.) will cause the following warning to be displayed:

Hardcopy absent or off line.

Remote Diagnostics

The instrument diagnostics can be remotely controlled with either of two modes. One is the front panel emulation mode, which provides nearly 100% control of the Extended Diagnostics from a terminal connected by an RS-232-C cable. The second is the system (normal operation) mode which allows a GPIB or an RS-232-C controller to invoke Self Tests and Extended Diagnostics and to query the instrument for test status. The two modes are quite different so they are covered separately.

Front Panel Emulation Mode (RS-232-C Only)

The front panel emulation mode provides control of the Extended Diagnostics that is identical to front panel operation except that terminal key strokes are used instead of screen touches to select items. Only interactive tests such as the Front Panel Verify tests cannot be performed from the remote terminal. This mode is useful when the display or touch panel is faulty or when you want to control the Extended Diagnostics remotely with a modem and phone link.

Entry-To use the front panel emulation mode, connect an ANSI 3.64 compatible terminal to the instrument rear-panel RS-232-C port and put the instrument into the Extended Diagnostics mode. Make sure the terminal is in its ANSI compatible mode. Then, press "T" (capital T) on the terminal keyboard to transfer control to the terminal. If the display is defective, the instrument can still be put into this mode of remote operation by having a Self Test failure at power-up. The failure can be either an existing fault or an induced one (i.e., by pressing a button throughout the power-up cycle). When the Self Tests finish with a failure, indicated by two audible beeps, press "T" on the keyboard.

Upon transfer of control, the Block menu is displayed on the terminal with all information normally found on the front panel display. Each selectable menu item has an associated key stroke character in parentheses in front of it. The key stroke characters for the test status selections are reassigned when different menus are displayed while the key stroke characters for the test control selections are the same for all menus.

In this mode, the touch panel is still active and mapped to selections on the terminal screen menu. Hence, touching the screen may cause a change in the remote terminal menu selections.

Exit–To exit the front panel emulation mode, enter the "E" (capital E) character. This causes the instrument to leave Extended Diagnostics and resume normal operation. Alternately, the "T" character, as described below, will transfer control of Extended Diagnostics back to the instrument front panel.

Non-Displayed Commands-Three additional commands are available in the front panel emulation mode that do not appear in the menus on the instrument or terminal display. They are as follows:

"B"(baud rate)(CR)	This command lets you change the baud rate to one of the allowable baud rates (i.e., 110, 300 ..., 38400) from the default of 9600 set with the Diagnostic Options Jumpers on the Standard I/O Board. A carriage return must be entered after the new baud rate is typed in to complete the entry. Note that the terminal baud rate should be changed after changing the instrument baud rate.
"T"	This command character toggles control between the instrument display and the terminal display.
"H"	This command character produces a hardcopy of the menu display. Hardcopy operation was described earlier in this section.

System Mode (GPIB and RS-232-C)

The system or normal operating mode allows either the Self Tests or Extended Diagnostics to be invoked with external interface commands. The commands are as follows:

Header	Link	Argument	Notes
TEST	none	[XTND]	Set only
DIAG	none		Query only

For detailed external interface and syntax information, refer to Section 3, GPIB and RS-232-C Interfaces, in the 11401/11402 User's Reference Manual.

TEST Command-TEST without arguments initiates the Self Tests. TEST with the argument XTND initiates the Extended Diagnostics.

When the 11401/11402 receives a TEST command, the instrument goes through a pseudo power-down sequence (i.e., saves current settings), performs the testing requested, then does a power-up sequence, regardless of whether or not faults were found. In the system mode, a Self Test failure does not cause the instrument to enter the Extended Diagnostics mode.

When requested by the TEST XTND command, Extended Diagnostics are automatically configured to run all automatic tests and to exit back to normal mode operation upon completion of testing. The touch panel is disabled during Extended Diagnostics, which is indicated by a TOUCH PANEL button status of OFF and by the following message in the Prompt/Warning Section of the display:

Front panel locked out.

Completion of diagnostic testing is signaled by one of the following event codes:

Event Code	Explanation
460	Self- or extended-test diagnostics were completed successfully.
394	Self- or extended-test diagnostics were completed and failed.

If RQS is ON when the TEST command is executed, an SRQ interrupt will signal test completion to the controller. Further error information is available with the DIAG query command.

TEST deletes all stored waveforms upon execution.

DIAG? Query Command–This query returns pass/fail information from the most recent invocation of Self Tests or Extended Diagnostics. Power-up failure information can also be obtained with this query.

Failure information includes error index codes. A query after Self Tests have run will return only one error index code per subsystem. A query after Extended Diagnostics have run, however, will return up to 206 error index codes.

Pass/fail status will be in different forms depending on instrument configuration and whether or not the power-up Self Tests were bypassed via the Diagnostics Options Jumpers.

Diagnostics Passed Response–When the Diagnostics find no faults, the DIAG? response reflects special conditions or configurations present in the instrument. For example, if no faults were found and there are no 11000 series plug-in units installed, the query response will be

DIAG PASSED:"L????, C????, R????"

If no problems were found and all compartments have 11000 series plug-in units installed, the query response will be

DIAG PASSED:"NONE"

Diagnostics Failed Response–When the Diagnostics find a fault, the DIAG? response will be similar to the following example:

DIAG FAILED:"E1311, G2111, C????"

The first two failure entries are error index codes. The last entry means that no 11000 series plug-in unit is installed in the center compartment.

Diagnostics Bypassed Response–When the instrument is powered up with the Self Tests bypassed with the Diagnostics Options Jumpers the query response will be:

DIAG BYPASSED

This query response will not occur when the Self Tests are initiated during normal operation since Self Tests can not be bypassed at that time.

Test Descriptions

Kernel Page Layout

Kernel test descriptions are laid out to provide easy test identification and comprehensive test descriptions that are easy to follow. The page elements, in their order down the page, are as follows:

Page Header — Test identification is simplified by the subsystem name and routine name at the top of each page. The format is as follows:

Subsystem Name Routine Name (Kernel Error Index)

This information makes finding the test description for a given kernel error index code easier because the kernel error index codes, located at the outside edge of each page, are easy to see while thumbing through the test pages. Also, it lets you know clearly what test is being discussed on each page of a long test description.

Routine — This name is the same as that used in the page header and provides a general idea of the circuit being tested.

Overview — Provides a brief description of how the circuit was setup and exercised and it refers to schematic diagrams covering the exercised circuits.

Operator Procedure — This label is included only for interactive tests. When included in a test description, it provides setup and procedural instructions to verify, calibrate, or troubleshoot a specific circuit.

Description — Provides a detailed, step-by-step description of how the test routine exercises the circuit, including information about the operation of important signal lines, components, and circuits.

Error Index — This label is followed by a two digit error index code. The associated discussion describes the possible failure modes for the circuit tested.

See Also — Provides other references when necessary or helpful. Some examples are references to Checks and Adjustments procedures in the Service Reference Manual or to interactive diagnostic tests used to verify apparent failures.

Caveats — This entry gives other conditions that could cause the test to fail, such as improper instrument setup for the test routine.

Self Test/Extended Page Layout

Self Test/Extended test descriptions are laid out to provide easy test identification and comprehensive test descriptions that are easy to follow. The page elements, in their order down the page, are similar to those described in the Kernel Page Layout section above, with the following differences:

Page Header Test identification is simplified by the Block, Area, and Routine information at the top of each page. The format is as follows:

Block Name Area Name Routine Name (Error Index)

Routine This name is the same as that used in the Extended Diagnostics routine menus.

Error Index This label is followed by a five digit error index code. The associated discussion describes the failure mode for the circuit tested.

Page Footer The subsystem to which the test belongs is shown at the bottom of the page.

Notation Used

In the test descriptions, certain notation conventions are used to represent signal line names, component names, and circuit names. Circuit names, which coincide with names of circuit areas on schematics, are shown in bold type, as in

Printer Controller U430<4>

Signal line and component names appear as upper case, as in

DIGREQ(L) U524-12 <27> or U524-19(H):L <27>

Signal line and component names also have some other features in common. A signal's "true", "high", "active", or "on" state is in parentheses after its name or number. The dash and number following the "U" number specify a particular pin number of the component. The numbers at the end of signal names appearing in "<>" specify the schematic diagram on which the component can be found. Finally, on the right example above, the ":L" indicates the signal is being driven low. It is used as in "Set U524-19(H):L<27>"; which says that pin 19 of U524 is driven low, its non-true state.

Executive Kernel Overview

The following list gives an overview of the kernel tests that the Executive processor executes at power-up before Self Tests run. The kernel tests are shown in the order that they are executed and documented later in this manual. This overview can be used to help locate the desired test description.

1F DRAM Data Lines
1E DRAM Byte Access
1D DRAM Address/Data
1C Bank Select & Memory Configuration
1B ROM U240 Location
1A ROM U240 Checksum
19 ROM U250 Location
18 ROM U250 Checksum
17 EPROM U630 Location
16 EPROM U630 Checksum
15 EPROM U730 Location
14 EPROM U730 Checksum
13 Timer 0 Interrupt
12 Timer 1 Interrupt
11 Timer 2 Interrupt
10 80287 Math Coprocessor
0F DMA Controller Interrupt
0E Front Panel Controller Interrupt
0D Serial Data Interface Interrupt
0C Real Time Clock Interrupt
0B GPIB Controller Interrupt
0A Printer Controller Interrupt
09 Std RS232 Controller Interrupt
08 MMU Display Talk Request Interrupt
07 MMU SAG Interrupt
06 MMU RAG Interrupt
05 DMA 0 Transfer
04 Std RS232 External Loopback (forced by I/O Bd straps only)

Display Kernel Overview

The following list gives an overview of the kernel tests that the Display processor executes at power-up before Self Tests run. The kernel tests are shown in the order that they are executed and documented later in this manual. This overview can be used to help locate the desired test description.

FF	RAM Data Lines
FE	RAM Address/Data
FD	ROM U612 Location
FC	ROM U602 Location
FB	ROM U612 Checksum
FA	ROM U602 Checksum
F9	DMA 0
F8	DMA 1
F7	Timer 0
F6	Timer 1
F5	Timer 2
F4	Executive Communication

Digitizer Kernel Overview

The following list gives an overview of the kernel tests that the Digitizer processor executes at power-up before Self Tests run. The kernel tests are shown in the order that they are executed and documented later in this manual. This overview can be used to help locate the desired test description.

1F MPU RAM Data Lines
1E MPU RAM Address/Data
1D ROM U281 Location
1C ROM U283 Location
1B ROM U281 Checksum
1A ROM U283 Checksum
19 MPU DMA 0
18 MPU DMA 1
17 MPU Timer 2
16 Executive Communication

Executive Self Test/ Extended Overview

The following list gives an overview of the Executive subsystem Extended Diagnostic menus and tests. They are shown in the exact order and manner that they appear in the Extended Diagnostic menus and are documented later in this manual. This overview can be used to help locate the desired test description.

Exec Control (E1XXX)

- ROM Location (E11XX)
  - U250 (E111X)
  - U240 (E112X)
  - U630 (E113X)
  - U730 (E114X)
  - U600 (E115X)
  - U700 (E116X)
  - U612 (E117X)
  - U712 (E118X)
  - U620 (E119X)
  - U720 (E11AX)
- ROM Checksum (E12XX)
  - U250 (E121X)
  - U240 (E122X)
  - U630 (E123X)
  - U730 (E124X)
  - U600 (E125X)
  - U700 (E126X)
  - U612 (E127X)
  - U712 (E128X)
  - U620 (E129X)
  - U720 (E12AX)
- RAM Refresh (E13XX)
  - Rate (E131X)
- Dynamic RAM (E14XX)
  - Config (E141X)
  - Data Lines (E142X)
  - Address/Data (E143X)
- NVRAM (E15XX)
  - Battery (E151X)
  - Data Lines (E152X)
  - Address/Data (E153X)
- Intrrpt Ctrl (E16XX)
  - Master (E161X)
  - Slave 1 (E162X)
  - Slave 3 (E163X)

- Timers (E17XX)
  - Timer 0 (E171X)
  - Timer 1 (E172X)
  - Timer 2 (E173X)
  - Diagn Signal (E174X)
- TimerIntrpts (E18XX)
  - Timer 0 (E181X)
  - Timer 1 (E182X)
  - Timer 2 (E183X)
- MPU Waits (E19XX)
  - Zero Wait (E191X)
  - One Wait (E192X)
  - Two Wait (E193X)
  - Four Wait (E194X)
- ROM Waits (E1AXX)
  - Zero Wait (E1A1X)
- Math Coproc (E1BXX)
  - Floating Pt (E1B1X)
- DMAs (E1CXX)
  - DMA 0 (E1C1X)
  - Interrupt (E1C2X)

Front Panel (E2XXX)

- Control (E21XX)
  - RAM (EE211X)
  - RAM Control (E212X)
  - Interrupt (E213X)
- Hard Keys (E22XX)
  - Open (E221X)
- Soft Keys (E23XX)
  - Row Open (E231X)
  - Column Open (E232X)
  - Row Close (E233X)
  - Column Close (E234X)
- Knobs (E24XX)
  - Upper Knob (E241X)
  - Lower Knob (E242X)
- Verify
  - Hard Keys
  - Soft Keys
  - Knobs

Internal I/O (E3XXX)

- Temp Sensor (E31XX)
  - Comparator (E311X)
- Real Time Clk (E32XX)
  - Counting (E321X)
  - Interrupt (E322X)
  - Calibrate
- Tone Gen
  - Ramp Tone

External I/O (E4XXX)

- Printer (E41XX)
  - Loopback (E411X)
  - Interrupt (E412X)
  - Pattern
- RS232 (E42XX)
  - Loopback (E421X)
  - Baud Rate E422X)
  - Error Gen (E423X)
  - Interrupt (E424X)
  - Extern Loop (E425X)
- GPIB (E43XX)
  - Intrpt Reset (E431X)
  - Reset Status (E432X)
  - Data Lines (E433X)
  - Interrupt (E434X)

Subsys Comm (E5XXX)

- MMU Control (E51XX)
  - Status Reg (E511X)
  - Arbitration (E512X)
  - Refresh (E513X)
  - Dtalk Intrpt (E514X)
  - SAG Compare (E515X)
  - SAG Adder (E516X)
  - SAG Intrpt (E517X)
  - RAG Regs (E518X)
  - RAG Intrpt (E519X)
- Waveform RAM (E52XX)
  - Size (E521X)
  - Data Lines (E522X)
  - Address/Data (E523X)

WCA Control (E53XX)
- Reset (E531X)
- Allnull (E532X)
- M/Mund (E533X)
- Idle (E534X)
- Comprss Null (E535X)
- Xparent Null (E536X)
- Comprss Over (E537X)
- Xparent Over (E538X)
- Comprss Undr (E539X)
- Xparent Undr (E53AX)
- Non Special (E53BX)

WCA Cmprssor (E54XX)
- Max Special (E541X)
- Max DataLine (E542X)
- Min DataLine (E543X)
- Min Special (E544X)

WCA Adder (E55XX)
- Offset (E551X)
- Data Paths (E552X)
- Overrange (E553X)
- Underrange (E554X)

MainFrm Comm (E56XX)
- Display (E561X)
- Digitizer (E562X)

SDI (E57XX)
- Left Loop (E571X)
- Center Loop (E572X)
- Right Loop (E573X)
- Interrupt (E574X)

Plug-in Comm (E58XX)
- Left (E581X)
- Center (E582X)
- Right (E583X)

Display Self Test/ Extended Overview

The following list gives an overview of the Display subsystem Extended Diagnostic menus and tests. They are shown in the exact order and manner that they appear in the Extended Diagnostic menus and are documented later in this manual. This overview can be used to help locate the desired test description.

Dsy Control (D1XXX)

- ROM Location (D11XX)
  - U612 (D111X)
  - U602 (D112X)
- ROM Checksum (D12XX)
  - U612 (D121X)
  - U602 (D122X)
- Static RAM (D13XX)
  - Data Lines (D131X)
  - Address/Data (D132X)
- Timers (D14XX)
  - Timer 0 (D141X)
  - Timer 1 (D142X)
  - Timer 2 (D143X)
- DMAs (D15XX)
  - DMA 0 (D151X)
  - DMA 1 (D152X)
- Exec Intrfce (D16XX)
  - Command Port (D161X)
  - DMA Access (D162X)
  - Wavefrm Port (D163X)
  - Attributes (D164X)

Video Gen (D2XXX)

- Timing (D21XX)
  - Trace CAS (D211X)
  - BSRLOAD (D212X)
  - Bit1 CAS (D213X)
  - ACCLK (D214X)
  - Trace RAS (D215X)
  - CRTC R/W (D216X)
  - B2HE (D217X)
  - B1LE (D218X)
  - VOLE (D219X)
  - Crastersync (D21AX)
  - Cfieldsync (D21BX)
  - Dispen (D21CX)

Address Mux (D22XX)
- MPU Address (D221X)
- CRTC Address (D222X)
- CRTC R/W (D223X)

Bit Plane 1 (D23XX)
- Data Lines (D231X)
- Address/Data (D212X)

Bit Plane 2 (D24XX)
- Data Lines (D241X)
- Address/Data (D242X)

Trace Plane (D25XX)
- Data Lines (D251X)
- Address/Data (D252X)

VRS Gen (D26XX)
- Single Axis (D261X)
- Dual Axis (D262X)
- Color Map (D263X)
- Video Shfter (D264X)
- Priority (D265X)

CRT Driver
- Stimulus

Digitizer Self Test/ Extended Overview

The following list gives an overview of the Digitizer subsystem Extended Diagnostic menus and tests. They are shown in the exact order and manner that they appear in the Extended Diagnostic menus and are documented later in this manual. This overview can be used to help locate the desired test description.

Dig Control (G1XXX)

- ROM Location (G11XX)
  - U281 (G111X)
  - U283 (G112X)
- ROM Checksum (G12XX)
  - U281 (G121X)
  - U283 (G122X)
- Static RAM (G13XX)
  - Data Lines (G131X)
  - Address/Data (G132X)
- Timers (G14XX)
  - Timer 2 (G141X)
- DMAs (G15XX)
  - DMA 0 (G151X)
  - DMA 1 (G152X)
- Mesg Intrfce (G16XX)
  - Data (G161X)
  - Address/Tag (G162X)

Timebases (G2XXX)

- Clocks (G21XX)
  - 20.0000 MHz (G211X)
  - 19.6608 MHz (G212X)
- Main (G22XX)
  - Rate (G221X)
  - Post Record (G222X)
  - Pretrigger (G223X)
- Window (G23XX)
  - Rate (G231X)
  - 1 Pst Record (G232X)
  - 1 Position (G233X)
  - 2 Pst Record (G234X)
  - 2 Position (G235X)
  - No Trigger (G236X)

Points Acq (G3XXX)

- Acq Memory (G31XX)
  - Accumulator (G311X)
  - Input Contrl (G312X)
  - Data Lines (G313X)
  - Address/Data (G314X)
- Initl Wfm Id (G32XX)
  - Data Lines (G321X)
  - Address/Data (G322X)
- Plg Seq Ctrl (G33XX)
  - Vert Src Seq (G331X)
  - Tbase Req Id (G332X)
  - Seq Cntr Clk (G333X)
- Cal Refs
  - FP –10.0000V
  - FP –5.0000 V
  - FP –2.5000 V
  - FP –1.0000 V
  - FP –100.00mV
  - FP 0.0000 V
  - FP +99.951mV
  - FP +999.51mV
  - FP +2.4988 V
  - FP +4.9976 V
  - FP +9.9951 V
  - FP 1 KHz
  - FP 1 MHz
  - PI –10.0000V
  - PI +9.9951 V
- A/D Convrter (G35XX)
  - Over Range (G351X)
  - Under Range (G352X)
  - Err Corr ROM (G353X)
- A/D Voltages (G36XX)
  - +0.800 V (G361X)
  - 0.000 V (G362X)
  - –0.800 V (G363X)

Triggers (G4XXX)

- Main (G41XX)
  - Holdoff (G411X)
  - Delay Events (G412X)
  - Trig to Trig (G413X)

Window (G42XX)
    Holdoff (G421X)

Time Interp (G51XX)
    TI Register (G511X)

DAG Regs (G52XX)
    Data Lines (G521X)
    Address/Data (G522X)

Wfm Intrfce (G53XX)
    Data (G531X)
    Address/Tag (G532X)

Routine Name DRAM Data Lines

Overview This test verifies the data lines to **DRAM** <26> by performing a "walking one's" test on **DRAM** address 00000*hex*.

Description

1. Perform a "walking one's" test on **DRAM** address 00000*hex*.

   - Write the pattern 8000*hex* to address 00000*hex*. Read the same address and verify that the pattern was 8000*hex*. Continue this write/read/verify sequence with patterns 4000*hex*, 2000*hex*, 1000*hex*, 0800*hex*, 0400*hex*, 0200*hex*, 0100*hex*, 0080*hex*, 0040*hex*, 0020*hex*, 0010*hex*, 0008*hex*, 0004*hex*, 0002*hex*, 0001*hex* and 0000*hex*.

Error Index 1F The pattern read from **DRAM** <26> address 00000*hex* was not the pattern written.

Routine Name

DRAM Byte Access

Overview

This test verifies the **DRAM R/W Control's** <26> ability to address odd and even bytes of **DRAM** <26> memory locations separately by writing to locations 00001*hex* and 00002*hex*.

Description

1. Perform byte access test.
   - Write/read/verify pattern AA*hex* to **DRAM** address 00001*hex*.
   - Write/read/verify pattern 55*hex* to **DRAM** address 00002*hex*.
   - Write/read/verify pattern CC*hex* to **DRAM** address 00001*hex*.

Error Index 1E

The pattern read from one of the byte locations was not the same as the pattern written.

Routine Name

DRAM Address/Data

Overview

This test verifies the address lines and data integrity of DRAM <26> address range 00000-2FFFF*hex* by performing a RAM test on this address range.

Description

1. Verify address range 00000-0FFFF*hex* (64k bytes). Terminate test if any verify operation fails.

   - Fill address range 00000-0FFFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 00000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 00000*hex*. Increment address and continue this read/verify/write sequence until address 0FFFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 00000*hex* for CCCC*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

2. Repeat step 1 for address ranges 10000-1FFFF*hex* and 20000-2FFFF*hex*.

Error Index 1D

The pattern read from a DRAM address location was not the pattern written. On failure, the DRAM Data Lines test (1F) is repeatedly performed on the failed address location.

Routine Name

Bank Select and Memory Configuration

Overview

This test verifies **DRAM Configuration** <26> jumper selections and the software bank selectability through the **Bank Decode/Select** <23> circuit.

Description

1. Read the **Memory Configuration Readback** U832 <25>. Only the upper four bits of this port represent the jumper positions. The lower four bits represent the current bank selection.

2. Verify the position of the EPROM-size selection jumper J541 <25> for 27512 type EPROMs.

3. Verify the position of the virtual or real memory addressing jumper J520 <26> for real memory addressing.

4. Verify the combination of RAM device type and number of RAM banks jumpers J501 <26> and J521 <26>, respectively, for 64k x 4 RAM device type and two banks.

5. Store the current bank by reading **Bank Decode/Select** U530A <23>.

6. Verify bank selectability by doing a "walking one's" test on the 4-bit bank select register U520 <23>.

   - Write patterns 01*hex*, 02*hex*, 04*hex*, and 08*hex* to **Bank Decode/Select** U520 <23>. Read and verify the written patterns from **Bank Decode/Select** U530A <23> and **Memory Configuration Readback** U832 <25> (only the lower four bits).

7. Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 1C

The 27512 EPROM has not been selected through J541; or real mode selection has not been made through J520; or the RAM device and number of banks combination selected through J501 and J521, respectively, are not appropriate; or the bank select mechanism is not functioning properly.

Routine Name

ROM U240 Location

Overview

This test verifies that **ROM** U240 <24> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 1B

The bytes at E0009*hex* and E000B*hex* were not complementary or the location byte did not match the known value.

Routine Name

ROM U240 Checksum

Overview

This test verifies the integrity of **ROM** U240 <24> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U240 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 1A

The bytes at E0009*hex* and E000B*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name ROM U250 Location

Overview This test verifies that **ROM** U250 <24> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 19 The bytes at E0008*hex* and E000A*hex* were not complementary or the location byte did not match the known value.

Routine Name

ROM U250 Checksum

Overview

This test verifies the integrity of **ROM U250** <24> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U250 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 18

The bytes at E0008*hex* and E000A*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name EPROM U630 Location

Overview This test verifies that **EPROM** U630 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 17 The bytes at C0008*hex* and C000A*hex* were not complementary or the location byte did not match the known value.

Routine Name

EPROM U630 Checksum

Overview

This test verifies the integrity of EPROM U630 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U630 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 16

The bytes at C0008*hex* and C000A*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name

EPROM U730 Location.

Overview

This test verifies that **EPROM** U730 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0009*hex* and C000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 15

The bytes at C0009*hex* and C000B*hex* were not complementary or the location byte did not match the known value.

Routine Name

EPROM U730 Checksum

Overview

This test verifies the integrity of **EPROM** U730 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0009*hex* and C000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U730 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index 14

The bytes at C0009*hex* and C000B*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name

Timer 0 Interrupt

Overview

This test verifies the **Timer** U822 <21> COUNTER 0 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Clear the **Timer** COUNTER 0 interrupt.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21>, 34*hex* to **Timer** U822 <21> Control Word Register, and then toggle **Timer 0 Interrupt Reset** U812B-13 <21>.
2. Verify that there is no **Timer** COUNTER 0 interrupt pending at the **Interrupt Controllers** MASTER U350 <24>.
   - Read **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR2 U350-20 <24> is low.
3. If there is no **Timer** COUNTER 0 interrupt pending, then enable **Timer** COUNTER 0 interrupt by setting bit 2 of the **Interrupt Controllers** MASTER U350 <24> Mask Register low.
4. Complete the programming for **Timer** COUNTER 0 with 80*hex* as the starting count.
   - Write 80*hex* and 00*hex* to the Count Register low and high bytes, respectively. This starts the counter.
5. Perform a software delay.
6. Verify that the **Timer** COUNTER 0 interrupt did occur.
7. Disable the **Timer** COUNTER 0 interrupt by setting bit 2 of the **Interrupt Controllers** MASTER U350 <24> Mask Register high.
8. Restore the **Timer** COUNTER 0 for the normal operating firmware.

Error Index 13

The **Timer** COUNTER 0 interrupt did not occur or it could not be cleared at the **Interrupt Controllers** MASTER U350<24>.

Routine Name

Timer 1 Interrupt

Overview

This test verifies the **Timer** U822 <21> COUNTER 1 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Configure the **Timer** COUNTER 1 to count 6 MHz clock.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21> so that G1 U822-14 <21> is high and PCLK(H) is connected to CLK1 U822-15 <21>.
2. Clear **Timer** COUNTER 1 interrupt.
   - Write 70*hex* to the Control Word Register.
3. Verify that there is no **Timer** COUNTER 1 interrupt pending at the **Interrupt Controllers** SLAVE 3 U370 <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR0 U370-18 is low.
4. If there is no **Timer** COUNTER 1 interrupt pending, then enable the **Timer** COUNTER 1 interrupt by setting bit 0 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register low.
5. Complete programming the **Timer** COUNTER 1 with 80*hex* as the starting count.
   - Write 80*hex* and 00*hex* to the Count Register low and high bytes, respectively. This starts the counter.
6. Perform a software delay.
7. Verify that **Timer** COUNTER 1 interrupt did occur.
8. Disable the **Timer** COUNTER 1 interrupt by setting bit 0 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register high.

Error Index 12

The **Timer** COUNTER 1 interrupt did not occur or it could not be cleared at the **Interrupt Controllers** SLAVE 3 U370 <24>.

Routine Name

Timer 2 Interrupt

Overview

This test verifies the **Timer** U822 <21> COUNTER 2 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Configure the **Timer** COUNTER 2 to count 6 MHz clock.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 <21> is high and PCLK(H) is connected to CLK2 U822-18 <21>.
2. Clear **Timer** COUNTER 2 interrupt.
   - Write B0*hex* to the Control Word Register.
3. Verify that there is no **Timer** COUNTER 2 interrupt pending at the **Interrupt Controllers** SLAVE 3 U370 <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR0 U370-19 is low.
4. If there is no **Timer** COUNTER 2 interrupt pending, then enable the **Timer** COUNTER 2 interrupt by setting bit 1 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register low.
5. Complete programming the **Timer** COUNTER 2 with 80*hex* as the starting count.
   - Write 80*hex* and 00*hex* to the Count Register low and high bytes, respectively. This starts the counter.
6. Perform a software delay.
7. Verify that **Timer** COUNTER 2 interrupt did occur.
8. Disable the **Timer** COUNTER 2 interrupt by setting bit 1 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register high.

Error Index 11

The Timer COUNTER 2 interrupt did not occur or it could not be cleared at the **Interrupt Controllers** SLAVE 3 U370 <24>.

Routine Name 80287 Math Coprocessor

Overview This test verifies the floating point and transcendental function capabilities of the **Numeric Processor Extension** U500 <23>.

Description

1. Compute the following equation using the **Numeric Processor Extension** U500 <23> and verify that the result is within a tolerance of ±0.005 of the expected value.

$$\frac{\log_e(15.3) * \log_{10}(23.5) * \exp(0.12)}{\text{atan}[\sin(0.7071) * \cos(0.5) * \tan(0.5774)]}$$

Error Index 10 The result from the floating point calculation was not 11.8582 ±0.005.

Routine Name

DMA Controller Interrupt

Overview

This test verifies the **DMA Controller** <24> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. As the **DMA Controller** is an option in the instrument, this test checks to see if the option is present. If the option is not present, no further testing is done. Otherwise, a memory to memory DMA transfer is completed to generate the interrupt.

Description

1. Check to see if the **DMA Controller** is present.
   - Write, read, and verify 7F01*hex* to the General Mode Register of the **DMA Controller**.
2. If the **DMA Controller** is present (i.e., 7F01*hex* is verified in the step 1), program the **DMA Controller** channel 0 and clear its interrupt, if any.
   - Write 7B01*hex*, 0000*hex*, 0000*hex*, 0013*hex*, 0000*hex*, and 0018*hex* to General Mode Register, General Burst Register, General Delay Register, Command Pointer Register High, Command Pointer Register Low and General Command Register, respectively.
3. Verify that there is no DMA interrupt pending at the **Interrupt Controllers** SLAVE 1 U360 <24>.
   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR6 U360-24 is low.
4. If there is no DMA interrupt pending, construct the command block for the transfer at **DRAM** address 30000*hex*. Initialize the source and destination memory locations.
   - Write C8DD*hex*, 0013*hex*, 0020*hex*, 0013*hex*, 0040*hex*, 0000*hex*, 0010*hex*, 0000*hex*, 1401*hex* to locations starting from 30000*hex*.
   - Initialize the source **DRAM** address range 30020-3002A*hex* with patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex*.
   - Initialize the destination **DRAM** address range 30040-3004A*hex* with pattern 8000*hex*.
5. Enable the DMA interrupt by setting bit 6 of **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register and bit 1 of **Interrupt Controllers** MASTER U350 <24> Mask Register low.
6. Start the DMA transfer by writing 1A*hex* to General Command Register.
7. Perform a software delay.
8. Verify that the DMA interrupt occurred.
9. Disable the DMA interrupt by setting bit 6 of **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register and bit 1 of **Interrupt Controllers** MASTER U350 <24> Mask Register high.

Error Index 0F

The DMA interrupt did not occur or it could not be cleared at **Interrupt Controllers** SLAVE 1 U360.

Routine Name

Front Panel Controller Interrupt

Overview

This test verifies the **Front Panel Controller** <2> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The infrared LEDs are disabled and enabled to generate the interrupt.

Description

1. Initialize the **Front Panel Controller** <2>.
   - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
2. Verify that there is no front panel interrupt pending at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <24> is low.
3. If there was no pending front panel interrupt, enable the front panel interrupt by setting the **Interrupt Controllers** MASTER U350 <24> Mask Register bit 4 low.
4. Disable the front panel infrared LEDs by writing 00*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11(L) <2> is low.
5. Perform a software delay.
6. Enable the front panel infrared LEDs by writing 01*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11(L) <2> is high.
7. Perform a software delay.
8. Verify that the front panel interrupt did occur.
9. Disable the front panel interrupt by setting **Interrupt Controllers** MASTER U350 Mask Register bit 4 high.
10. Initialize the **Front Panel Controller.**
    - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.

Error Index 0E

The **Front Panel Controller** <2> interrupt did not occur or it could not be cleared at the **Interrupt Controllers** MASTER U350 <24>.

Routine Name

Serial Data Interface Interrupt

Overview

This test verifies the **Serial Data Interface** SDI IC U330 <21> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program the **Serial Data Interface** SDI IC U330 <21> to loopback mode.
   - Set bits 3 and 4 (Receiver Reset and Transmitter Reset bits) of the Miscellaneous Control Select 1 latch high.
   - Write 38*hex* and 00*hex* to the SDI Control Select Register and SDI Interrupt Output Mask Register.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.
2. Verify that there is no SDI interrupt pending at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 1 U360 <21> INTERRUPT REQUEST REGISTER and verify that IR4 U360-22 is low.
3. If there is no pending interrupt, enable the SDI interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 4 low.
4. Enable the transmitter empty interrupt at the Serial Data Interface SDI IC U330 <21> by writing 01*hex* to the SDI Interrupt Output Mask Register.
5. Perform a software delay and then verify that the SDI interrupt did occur.
6. Disable the SDI interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 4 high.
7. Program the **Serial Data Interface** SDI IC U330 <21> to normal mode.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch high.
   - Write 3C*hex* to the SDI Control Select Register.
   - Read SDI I/O Status Register to find out the state of Left Data Back (bit 5), Center Data Back (bit 3), and Right Data Back (bit 0) bits.
   - Write a value to the SDI Control Select Register so that bit 2 is high and bits 3, 4, and 5 reflect the state of Left Data Back, Center Data Back, and Right Data Back, respectively.
   - Write 0E*hex* to SDI Interrupt Output Mask Register.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.

Error Index 0D The SDI interrupt did not occur or it could not be cleared at the **Interrupt Controllers** SLAVE 1 U360 <24>.

Routine Name

Real-Time Clock Interrupt

Overview

This test verifies the **Real-Time Clock** <21> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Mask all the **Real-Time Clock** <21> interrupts by writing 00*hex* to its Mask Register.
2. Read the Interrupt Status Register of the **Real-Time Clock** <21> to clear all the interrupts.
3. Verify that there is no **Real-Time Clock** interrupt at the **Interrupt Controllers** <24>.
   - Read the **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR6 U370-24 is low.
4. If there is no pending **Real-Time Clock** interrupt at the **Interrupt Controllers** (i.e., IR6 U370-24 is low), enable the interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 6 and **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 low.
5. Enable the **Real-Time Clock's** one hundredth interrupt at the **Real-Time Clock** <21> by writing 18*hex* and 02*hex* to the Command Register and Mask Register, respectively.
6. Perform a software delay (at least 10 ms).
7. Disable the **Real-Time Clock's** one hundredth interrupt at the **Real-Time Clock** <21> by writing 08*hex* to the Command Register.
8. Verify that the **Real-Time Clock's** interrupt did occur.
9. Disable **Real-Time Clock's** interrupt at the **Interrupt Controllers** by setting **Interrupt Controllers** SLAVE 3 U370 <24> bit 6 and **Interrupt Controllers** MASTER U350 <24> bit 3 high.

Error Index 0C

The **Real-Time Clock** interrupt did not occur or it could not be reset at the **Interrupt Controllers** Slave U370 <24>.

Routine Name GPIB Controller Interrupt

Overview This test verifies that the GPIB interrupt can be cleared by resetting the **GPIB Controller** U410 <3>.

Description

1. Reset **GPIB Controller** U410 <3> and verify its reset status.
   - Write 80*hex*, 00*hex*, 00*hex*, and 00*hex* to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.
   - Read Interrupt Status Register 0 and Interrupt Status Register 1 and verify that they both read 00*hex*.
2. Verify that there is no GPIB interrupt pending at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR2 U370-20 is low.

Error Index 0B The **GPIB Controller** <3> reset did not set the Interrupt Status Register 0 to 00*hex*; or the **GPIB Controller** <3> reset did not set the Interrupt Status Register 1 to 00*hex*; or the **GPIB Controller** <3> interrupt could not be reset at the **Interrupt Controllers** SLAVE 3 U370 <24>.

Routine Name

Printer Controller Interrupt

Overview

This test verifies the **Printer Controller** U430 <4> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The interrupt is generated by looping back a byte of data.

Description

1. Program the **Printer Controller** <4> in test mode (loop back mode) by writing A6*hex* and 08*hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is low. This enables **Printer Loop Back Buffer** U540 <4> and disables **Printer Interface Control** buffer U541 <4>.

2. Verify that there is no printer interrupt pending at the **Interrupt Controllers** <24>.

   - Read the **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR6 U350-24 is low.

3. If there are no pending interrupts, enable the interrupt at the Interrupt Controllers <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 6 low.

4. Enable the interrupt at the **Printer Controller** by writing 0D*hex* to the Control Word Register.

5. Loop back a pattern.

   - Write AA*hex* to U430 Port A, perform a software delay, and read it from U430 Port B. Immediately following the read, a transmitter empty interrupt is generated, i.e., INTR(A)(H) U430-17 is high.

6. Perform a software delay.

7. Disable the interrupt at the **Printer Controller** <4> by writing 0C*hex* to the Control Word Register.

8. Disable the interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 6 high.

9. Verify that a printer interrupt did occur.

10. Program the **Printer Controller** <4> in normal mode by writing 09*hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is high.

Error Index 0A

The **Printer Controller** <4> interrupt did not occur or it could not be cleared at the **Interrupt Controllers** MASTER U350 <24>.

Routine Name

Std RS-232 Controller Interrupt

Overview

This test verifies the **Std RS-232 Controller's** U311 <3> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program **Std RS-232 Controller** U311 <3> in local loop back mode with eight bits per character, one stop bit, no parity and 9600 baud rate.
   - Write 35*hex*, 25*hex*, 1A*hex*, 13*hex*, 87*hex*, BB*hex*, and E5*hex* to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Verify that there is no pending RS-232 interrupt at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 is low.
3. If there is no pending RS-232 interrupt, enable the RS-232 interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 Mask Register bit 4 low.
4. Enable the receiver ready interrupt at the **Std RS-232 Controller** U311 <3> by writing 02*hex* to IMR.
5. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
6. Irrespective of the transmitter availability, transmit a character by writing 55*hex* to THRA.
7. Perform a software delay.
8. Read the character by reading RHRA.
9. Verify that the interrupt did occur.
10. Reset the receiver and disable the receiver interrupt by writing 25*hex* and 00*hex* to CRA and IMR, respectively.
11. Disable the RS-232 interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 4 high.
12. Program the **Std RS-232 controller** U311 <3> in normal mode with eight bits per character, one stop bit, no parity and 9600 baud rate.

- Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
- Read IPCR, write 00*hex* to IMR, and read ISR.
- Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index 09

The **Std RS-232 Controller** <3> interrupt did not occur or it could not be reset at the **Interrupt Controllers** SLAVE U370 <24>.

Routine Name MMU Display Talk Request Interrupt

Overview This test verifies the **MMU Gate Array** U210 <27> display talk request interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the **MMU Gate Array** U210 <27> to an isolated diagnostic mode by writing 1A*hex*, 007F*hex*, and 4*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.
2. Verify that there is no pending display talk request interrupt at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR2 U360-20 is low.
3. If there is no display interrupt pending, then enable display talk request interrupt by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 2 low.
4. Generate a display talk request interrupt by setting DATARDY(H) U524-6 <27> high and then low.
5. Perform a software delay and then verify that the display talk request interrupt did occur.
6. Clear the interrupt at the MMU by setting SMR bit 1 high.
7. Disable the display talk request interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 2 high.
8. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index 08 The display talk request interrupt did not occur or it could not be cleared at the **Interrupt Controllers** SLAVE 1 U360 <24>.

Routine Name

MMU SAG Interrupt

Overview

This test verifies the **MMU Gate Array** U210 <27> SAG Interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the **MMU Gate Array** U210 <27> to an isolated diagnostic mode by writing 1A*hex*, 007F*hex*, and 4*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.

2. Verify that there is no pending SAG interrupt at the Interrupt Controllers <24>.

   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR0 U360-18 is low.

3. If there is no SAG interrupt pending, enable the SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

4. Start a "write display" cycle by setting bit 9 of SMR high and setting SENDNEW(L) U524-5 <27> low and then high.

5. Perform a software delay and then verify that the SAG interrupt did occur.

6. Clear SAG interrupt at the MMU by setting bit 3 of SMR high.

8. Disable the SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 high.

9. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index 07

The SAG interrupt did not occur or it could not be cleared at the **Interrupt Controllers** SLAVE 1 U360 <24>.

Routine Name

MMU RAG Interrupt

Overview

This test verifies **MMU Gate Array's** U210 <27> RAG interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the **MMU Gate Array** U210 <27> to an isolated diagnostic mode by writing 1A*hex*, 007F*hex*, and 4*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.

2. Verify that there is no RAG interrupt pending at **Interrupt Controllers** <24>.

   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR1 U360-19 is low.

3. If there is no RAG interrupt pending, enable the RAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <24> Mask Register bit 1 low.

4. Generate an "end of digitizer message" interrupt.

   - Set DIGREQ(L) U524-12 <27> low and then high with U524-19(H):H <27>.

   - Set DIGREQ(L) U524-12 <27> high with U524-19(H):L <27>.

5. Perform a software delay and then verify that the RAG interrupt did occur.

6. Clear the RAG interrupt at the MMU by setting bit 4 of the Status and Mode Register (SMR) high.

7. Disable RAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 1 high.

8. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index 06

The RAG interrupt did not occur or it could not be cleared at the **Interrupt Controllers** SLAVE U360 <24>.

Routine Name

DMA 0 Transfer

Overview

This test verifies the **DMA Controller** <24> channel 0. As the **DMA Controller** is an option in the instrument, this test checks to see if the option is present. If the option is not present, no further testing is done. Otherwise, a memory to memory DMA transfer is performed and verified.

Description

1. Check to see if the **DMA Controller** is present.
   - Write, read, and verify 7F01*hex* to the General Mode Register of the **DMA Controller**.
2. If the **DMA Controller** is present (i.e., 7F01*hex* was verified in step 1), construct the command block for the transfer at **DRAM** <26> address 30000*hex*. Initialize the source and destination memory locations.
   - Write C8DD*hex*, 0013*hex*, 0020*hex*, 0013*hex*, 0040*hex*, 0000*hex*, 0010*hex*, 0000*hex*, and 1401*hex* to locations starting from 30000*hex*.
   - Initialize the source **DRAM** address range 30020-3002A*hex* with patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex*.
   - Initialize the destination **DRAM** address range 30040-3004A*hex* with pattern 8000*hex*.
3. Program the **DMA Controller** channel 0 to read the command block from memory and start the transfer of five words from 30020*hex* to 30040*hex*.
   - Write 7B01*hex*, 0000*hex*, 0000*hex*, 0013*hex*, 0000*hex*, and 001A*hex* to General Mode Register, General Burst Register, General Delay Register, Command Pointer Register High, Command Pointer Register Low and General Command Register, respectively.
5. Perform a software delay.
6. Clear the DMA interrupt at the **DMA Controller** and verify that the transfer was successful.
   - Write 0018*hex* General Command Register.
   - Compare the contents of the destination and source memory locations and verify that they are the same.

Error Index 05

The DMA transfer was not successful.

Routine Name

Std RS-232 External Loopback

Overview

This test verifies the **Std RS-232 Controller** U311 <3>, **RS-232 Clock Generator** <3> and **Std RS-232 Interface Drivers** <3> by externally looping back a set of patterns. Its use is only foreseen in situations where no diagnostic menus can be displayed on the screen (i.e., the operator needs to, or would like to, use the external diagnostic RS-232 terminal mode), yet the RS-232 port also contains a fault. In other words, there are multiple, non-kernel type faults in the instrument which prohibit being able to obtain diagnostic screen information.

Operator Procedure

This test only executes when the lower straps of S710 on the I/O board are set to 01*bin* (bottom strap to the right, the one above it set to the the left). This forces the test to be repeatedly executed after all other Executive Kernel tests have successfully run. The operator should connect an external loopback connector (Tek part no. 013-0198-00) to the RS232 connector on the rear panel of the instrument. External test equipment must be used to diagnose faults.

Description

1. Program the **Std RS-232 Controller** U311 <3> in normal mode with eight bits per character, one stop bit, no parity and 9600 baud rate.
   - Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
3. Irrespective of the transmitter availability, transmit pattern AA*hex* by writing it to THRA.
4. Wait for the receiver to be ready by repeatedly reading SRA until bit 0 is high or until a software timeout period has expired.
5. Irrespective of the receiver availability, receive the pattern by reading RHRA and verify it.
6. Repeat steps 2-5 for patterns CC*hex*, F0*hex*, 0F*hex*, and 55*hex*.
7. Program the **Std RS-232 Controller** U311 <3> in normal mode with eight bits per character, one stop bit, no parity and 9600 baud rate.
   - Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index 04 Not applicable since this test is user selected.

Executive Display Digitizer

Routine Name RAM Data Lines

Overview This test verifies the data lines to **General Purpose Static RAM** <33> by performing a "walking one's" test on **General Purpose Static RAM** address 00000*hex*.

Description

1. Perform a "walking one's" test on **General Purpose Static RAM** address 00000*hex*.

   - Write the pattern 8000*hex* to address 00000*hex*. Read the same address and verify that the pattern was 8000*hex*. Continue this write/read/verify sequence with patterns 4000*hex*, 2000*hex*, 1000*hex*, 0800*hex*, 0400*hex*, 0200*hex*, 0100*hex*, 0080*hex*, 0040*hex*, 0020*hex*, 0010*hex*, 0008*hex*, 0004*hex*, 0002*hex*, 0001*hex*, and 0000*hex*.

Error Index FF The pattern read from **General Purpose Static RAM** <33> address 00000*hex* was not the pattern written.

Routine Name RAM Address/Data

Overview This test verifies the address lines and data integrity of **General Purpose Static RAM** <33> address range 00000-00FFF*hex* by performing a RAM test on this address range.

Description

1. Verify address range 00000-00FFF*hex*. Terminate test if any verify operation fails.

   - Fill address range 00000-00FFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 00000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 00000*hex*. Increment address and continue this read/verify/write sequence until address 00FFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 00000*hex* for CCCC*hex*, F0F0*hex*, 5555*hex*, AAAA*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index FE The pattern read from a **General Purpose Static RAM** address location was not the pattern written. On failure, the RAM Data Lines test (FF) is repeatedly performed on the failed address location.

Routine Name ROM U612 Location

Overview This test verifies that **ROM & Select** U612 <33> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index FD The bytes at E0008*hex* and E000A*hex* were not complementary or the location byte did not match the known value.

Routine Name ROM U602 Location

Overview This test verifies that **ROM & Select** U602 <33> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index FC The bytes at E0009*hex* and E000B*hex* were not complementary or the location byte did not match the known value.

Routine Name

ROM U612 Checksum

Overview

This test verifies the integrity of **ROM & Select** U612 <33> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U612 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index FB

The bytes at E0008*hex* and E000A*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name ROM U602 Checksum

Overview This test verifies the integrity of **ROM & Select** U602 <33> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U602 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index FA The bytes at E0009*hex* and E000B*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name

DMA 0

Overview

This test verifies CPU & Ready Logic U524 <33> Programmable DMA Unit's channel 0 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 0 interrupt by writing 0004*hex* to DMA 0 Interrupt Control Register.

2. Initialize the source and destination memory locations in **General Purpose Static RAM** <33>.

   - Write patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex* and 5555*hex* to source memory locations starting at address 210*hex*.

   - Write patterns 5555*hex*, 3333*hex*, 0F0F*hex*, 00FF*hex* and AAAA*hex* to destination memory locations starting at address 200*hex*.

3. Program the DMA channel 0 to transfer the five source patterns to the destination memory locations starting at 200*hex* and generate an interrupt after completion of the transfer.

   - Write B725*hex*, 0005*hex*, 0000*hex*, 0200*hex*, 0000*hex*, 0210*hex* and B727*hex* to DMA 0 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source pointer Lower, and DMA 0 Channel Control Word registers, respectively. The last write initiates the DMA transfer.

4. Disable DMA 0 interrupt by writing 000C*hex* to DMA 0 Interrupt Control Register.

5. Read and verify the destination memory locations.

Error Index F9

One or more of the transferred patterns did not match the expected value.

Routine Name

DMA 1

Overview

This test verifies **CPU & Ready Logic** U524 <33> Programmable DMA Unit's channel 1 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 1 interrupt by writing 0004*hex* to DMA 1 Interrupt Control Register.

2. Initialize the source and destination memory locations in **General Purpose Static RAM** <33>.

   - Write patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex* and 5555*hex* to source memory locations starting at address 210*hex*.

   - Write patterns 5555*hex*, 3333*hex*, 0F0F*hex*, 00FF*hex* and AAAA*hex* to destination memory locations starting at address 200*hex*.

3. Program the DMA channel 1 to transfer the five source patterns to the destination memory locations starting at 200*hex* and generate an interrupt after completion of the transfer.

   - Write B725*hex*, 0005*hex*, 0000*hex*, 0200*hex*, 0000*hex*, 0210*hex*, and B727*hex* to DMA 1 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source pointer Lower, and DMA 1 Channel Control Word registers, respectively. The last write initiates the DMA transfer.

4. Disable DMA 1 interrupt by writing 000C*hex* to DMA 1 Interrupt Control Register.

5. Read and verify the destination memory locations.

Error Index F8

One or more of the transferred patterns did not match the expected value.

Routine Name

Timer 0

Overview

This test verifies **CPU & Ready Logic** U524 <33> Programmable Timers' timer 0 counting accuracy and its interrupt. The counting accuracy of the timer 0 is verified by counting the system clock for a short duration. If the timer 0 counts accurately, a timer 0 interrupt is generated and verified.

Description

1. Enable the timer 0 to count the system clock by setting its gate signal high.
   - Set **Diagnostic Control/Status Latch** TMR IN 0 <37> high by writing 83*hex* to U600 <37>.
2. Stop the timer 0 by writing 4000*hex* to its Timer Mode/Control Register.
3. Program the counter to count up by writing 0000*hex*, FFFF*hex*, and C000*hex* to timer 0 Count Register, Max Count Value A Register, and Timer 0 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
6. Read and verify the timer 0 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 0 is counting accurately), enable the timer 0 to count the system clock again by setting its gate signal high. If not, terminate testing.
   - Set **Diagnostic Control/Status Latch** TMR IN 0 <37> high by writing 83*hex* to U600 <37>.
8. Enable timer 0 interrupt by writing 0004*hex* to the Timer 0 Interrupt Control Register.
9. Program the counter to count up to 100*hex* and generate an interrupt when the count reaches 100*hex*.
   - Write 0100*hex*, 0000*hex*, and E000*hex* to timer 0 Max Count Value A Register, Count Register, and Timer 0 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
12. Disable timer 0 interrupt by writing 000C*hex* to the Timer 0 Interrupt Control Register.
13. Verify that the timer 0 interrupt did occur.

Error Index F7

The timer 0 count was not within the expected tolerance or the timer 0 interrupt did not occur.

Routine Name

Timer 1

Overview

This test verifies **CPU & Ready Logic** U524 <33> Programmable Timers' timer 1 counting accuracy and its interrupt. The counting accuracy of the timer 1 is verified by counting the system clock for a short duration. If the timer 1 counts accurately, a timer 1 interrupt is generated and verified.

Description

1. Enable the timer 1 to count the system clock by setting its gate signal high.
   - Set **Diagnostic Control/Status Latch** TMR IN 1 <37> high by writing A3*hex* to U600 <37>.
2. Stop the timer 1 by writing 4000*hex* to its Timer Mode/Control Register.
3. Program the counter to count up by writing 0000*hex*, FFFF*hex*, and C000*hex* to timer 1 Count Register, Max Count Value A Register, and Timer 1 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
6. Read and verify the timer 1 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 1 is counting accurately), enable the timer 1 to count the system clock again by setting its gate signal high. If not, terminate testing.
   - Set **Diagnostic Control/Status Latch** TMR IN 1 <37> high by writing A3*hex* to U600 <37>.
8. Enable timer 1 interrupt by writing 0004*hex* to the Timer 1 Interrupt Control Register.
9. Program the counter to count up to 100*hex* and generate an interrupt when the count reaches 100*hex*.
   - Write 0100*hex*, 0000*hex*, and E000*hex* to timer 1 Max Count Value A Register, Count Register, and Timer 1 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
12. Disable timer 1 interrupt by writing 000C*hex* to the Timer 1 Interrupt Control Register.
13. Verify that the timer 1 interrupt did occur.

Error Index F6

The timer 1 count was not within the expected tolerance or the timer 1 interrupt did not occur.

Routine Name

Timer 2

Overview

This test verifies **CPU & Ready Logic** U524 <33> Programmable Timers' timer 2 counting accuracy and its interrupt. The counting accuracy of the timer 2 is verified by counting the system clock for a short duration. If the timer 2 counts accurately, a timer 2 interrupt is generated and verified.

Description

1. Stop the timer 2 by writing 4000*hex* to its Timer Mode/Control Register.
2. Program the counter to count up by writing 0000*hex*, FFFF*hex*, and C000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
3. Perform a software delay.
4. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
5. Read and verify the timer 2 count against the known value for the expected tolerance.
6. If the count was within the expected tolerance (i.e., the timer 2 is counting accurately), enable the timer 2 interrupt by writing 0004*hex* to the Timer 2 Interrupt Control Register. If not, terminate testing.
7. Program the counter to count up to 100*hex* and generate an interrupt when the count reaches 100*hex*.
   - Write 0100*hex*, 0000*hex*, and E000*hex* to timer 2 Max Count Value A Register, Count Register, and Timer 2 Mode/Control Register, respectively.
8. Perform a software delay.
9. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
10. Disable timer 2 interrupt by writing 000C*hex* to the Timer 2 Interrupt Control Register.
11. Verify that the timer 2 interrupt did occur.

Error Index F5

The timer 2 count was not within the expected tolerance or the timer 2 interrupt did not occur.

Routine Name

Executive Communication

Overview

This test verifies the communication path from the Display processor to the Executive processor, through the **Interface Data Buffers** <37> and the **Executive Processor Parallel Interface Port** <37>, by inverting and echoing a specific set of "walking ones" and "walking zeros" patterns which are expected from the Executive processor. If, at any time, the Display processor receives an unexpected pattern, it enters an infinite loop in which it inverts and echo's every pattern sent thereafter.

Description

1. Set up **CPU & Ready Logic** 80186 DMA 0<33> to receive a data pattern from the **Executive Processor Parallel Interface Port** <37> and write it to **General Purpose Static RAM** <33> memory location 00210*hex* without generating any interrupt at the end of the transfer.

2. Set up **CPU & Ready Logic** 80186 DMA 1<33> to send a data pattern from **General Purpose Static RAM** <33> memory location 00210*hex* to the **Executive Processor Parallel Interface Port** <37> without generating any interrupt at the end of the transfer.

3. Initialize the **Executive Processor Parallel Interface Port** <37> by reading port latches U731 & U733 through **Interface Data Buffers** <37>.

4. Wait for a data pattern to be recevied from the Executive processor through DMA 0.

5. If the received data pattern matches the expected pattern, then invert all bits (i.e., 1's complement) and allow DMA 1 to send the inverted pattern back to the Executive processor (see Error Index section below).

6. If the received data pattern did not match the expected pattern, perform an infinite loop in which any patterns received, as in step 4, are inverted and sent back to the Executive processor, as in step 5.

7. Repeat steps 4 & 5 until all expected patterns have been received, inverted, and sent back (see Error Index section below).

Error Index F4

One of the data patterns received from the Executive processor did not match the expected pattern. The expected patterns to be received and those returned to the Executive processor are shown in the following table:

Expected Pattern Received (*hex*)	Pattern Sent (*hex*)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	EFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
EFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001

Routine Name MPU RAM Data Lines

Overview This test verifies the data lines to **MPU RAM** <16> by performing a "walking one's" test on **MPU RAM** address 00000*hex*.

Description

1. Perform a "walking one's" test on **MPU RAM** address 00000*hex*.

   - Write the pattern 8000*hex* to address 00000*hex*. Read the same address and verify that the pattern was 8000*hex*. Continue this write/read/verify sequence with patterns 4000*hex*, 2000*hex*, 1000*hex*, 0800*hex*, 0400*hex*, 0200*hex*, 0100*hex*, 0080*hex*, 0040*hex*, 0020*hex*, 0010*hex*, 0008*hex*, 0004*hex*, 0002*hex*, 0001*hex*, and 0000*hex*.

Error Index 1F The pattern read from **MPU RAM** <16> address 00000*hex* was not the pattern written.

Executive Display Digitizer

Routine Name MPU RAM Address/Data

Overview This test verifies the address lines and data integrity of MPU RAM <16> address range 00000-03FFF*hex* by performing a RAM test on this address range.

Description

1. Verify address range 00000-03FFF*hex*. Terminate test if any verify operation fails.
   - Fill address range 00000-03FFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 00000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 00000*hex*. Increment address and continue this read/verify/write sequence until address 03FFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 00000*hex* for CCCC*hex*, F0F0*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index 1E The pattern read from a MPU RAM address location was not the pattern written. On failure, the MPU RAM Data Lines test (1F) is repeatedly performed on the failed address location.

Routine Name ROM U281 Location

Overview This test verifies that **ROM & Select U281 <16>** is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index 1D The bytes at E0008*hex* and E000A*hex* were not complementary or the location byte did not match the known value.

Routine Name ROM U283 Location

Overview This test verifies that **ROM & Select U283 <16>** is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index 1C The bytes at E0009*hex* and E000B*hex* were not complementary or the location byte did not match the known value.

Routine Name ROM U281 Checksum

Overview This test verifies the integrity of ROM & Select U281 <16> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U281 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index 1B The bytes at E0008*hex* and E000A*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name ROM U283 Checksum

Overview This test verifies the integrity of ROM & Select U283 <16> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U283 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index 1A The bytes at E0009*hex* and E000B*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name

MPU DMA 0

Overview

This test verifies MPU U224 <16> Programmable DMA Unit's channel 0 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Program the DMA channel 0 to transfer the five source patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex* from consecutive EPROM memory locations to the destination memory locations in MPU RAM <16> and generate an interrupt after completion of the transfer.

   - Write B725*hex*, 0005*hex*, upper four bits of destination address, lower 16 bits of destination address (the address may be different in each version of software), upper four bits of source address, lower 16 bits of source address (again the address may be different in each version of software) and B727*hex* to DMA 0 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source Pointer Lower, and DMA 0 Channel Control Word registers, respectively. The last write initiates the DMA transfer.

2. Disable DMA channel 0 interrupt by writing B725*hex* to DMA 0 Channel Control Register.

3. Read and verify the destination memory locations.

Error Index 19

One or more of the transferred patterns did not match the expected value.

Routine Name MPU DMA 1

Overview This test verifies MPU U224 <16> Programmable DMA Unit's channel 1 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Program the DMA channel 1 to transfer the five source patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex* from consecutive EPROM memory locations to the destination memory locations in **MPU RAM** <16> and generate an interrupt after completion of the transfer.

   - Write B725*hex*, 0005*hex*, upper four bits of destination address, lower 16 bits of destination address (the address may be different in each version of software), upper four bits of source address, lower 16 bits of source address (again the address may be different in each version of software) and B727*hex* to DMA 1 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source pointer Lower, and DMA 1 Channel Control Word registers, respectively. The last write initiates the DMA transfer.

2. Disable DMA channel 1 interrupt by writing B725*hex* to DMA 1 Channel Control Register.

3. Read and verify the destination memory locations.

Error Index 18 One or more of the transferred patterns did not match the expected value.

Routine Name

MPU Timer 2

Overview

This test verifies MPU U224 <16> Programmable Timers timer 2's counting accuracy and its interrupt. The counting accuracy of the timer 2 is verified by counting the system clock for a short duration. If the timer 2 counts accurately, a timer 2 interrupt is generated and verified.

Description

1. Stop the timer 2 by writing 4000*hex* to its Timer Mode/Control Register.
2. Program the counter to count up by writing 0000*hex*, FFFF*hex*, and C000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
3. Perform a software delay.
4. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
5. Read and verify the timer 2 count against the known value for the expected tolerance.
6. If the count was within the expected tolerance (i.e., the timer 2 is counting accurately), clear any pending interrupts by writing 0000*hex* to both Interrupt Status Register and Interrupt In Service Register. Else terminate testing.
7. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
8. Enable the timer 2 interrupt by writing 0000*hex* to the Timer 2 Interrupt Control Register.
9. Program the counter to count up to 1000*hex* and generate an interrupt when the count reaches 1000*hex*.
    - Write 0000*hex*, 1000*hex*, E000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
10. Perform a software delay.
11. Disable timer 2 interrupt by writing 000F*hex* to the Timer 2 Interrupt Control Register.
12. Verify that the timer 2 interrupt did occur.

Error Index 17

The timer 2 count was not within the expected tolerance or the timer 2 interrupt did not occur.

Routine Name

Executive Communication

Overview

This test verifies the communication path from the Digitizer processor to the Executive processor, via **Digitizer/MMU Communication Port** <17>, **CPU/MMU Message Output Latch** <17>, **Digitizer/MMU Address Output Port** <17>, **CPU/MMU Message Tag Generator** <17>, and **Digitizer/MMU Handshake Logic** <31>, by inverting and echoing "walking one's" and "walking zero's" patterns which are expected to be received from the Executive processor. If, at any time, the Digitizer processor receives an unexpected pattern, it enters a "forever loop" in which it inverts and echo's every pattern sent thereafter.

Description

1. Enable data patterns to be sent from the Digitizer to the Executive by setting **Digitizer/MMU Handshake Logic** U83C-11 DIGREQ_EN(H):H <17>.

2. Set up **MPU** 80186 DMA 0<16> to receive a data pattern from the **Digitizer/MMU Communication Port** <17> and write it to **MPU RAM** <16> without generating any interrupt at the end of the transfer.

3. Set up **MPU** 80186 DMA 1<16> to send a data pattern from **MPU RAM** <16> to the **Digitizer/MMU Communication Port** <17> without generating any interrupt at the end of the transfer.

4. Wait for a data pattern to be recevied from the Executive processor through DMA 0.

5. If the received data pattern matches the expected pattern, then invert all bits (i.e., 1's complement) and allow DMA 1 to send the inverted pattern back to the Executive processor (see Error Index section below).

6. Signal the Executive processor, via the MMU, that a data pattern has been sent by performing a write to **CPU/MMU Message Tag Generator** U95 <17>. This sends a message tag of F*hex* to the MMU.

7. If the received data pattern did not match the expected pattern, perform an infinite loop in which any patterns received, as in step 4, are inverted and sent back to the Executive processor, as in steps 5 & 6.

8. Repeat steps 4-6 until all expected patterns have been received, inverted, and sent back (see Error Index section below).

Error Index 16

One of the data patterns received from the Executive processor did not match the expected pattern. The expected patterns to be received and those returned to the Executive processor are shown in the following table:

Expected Pattern Received (*hex*)	Pattern Sent (*hex*)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	EFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
EFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001

Routine Name U250

Overview This test verifies that ROM U250 <24> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1111 The bytes at E0008*hex* and E000A*hex* were not complementary.

Error Index E1112 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U240

Overview This test verifies that **EPROM** U240 <24> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1121 The bytes at E0009*hex* and E000B*hex* were not complementary.

Error Index E1122 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U630

Overview This test verifies that EPROM U630 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1131 The bytes at C0008*hex* and C000A*hex* were not complementary.

Error Index E1132 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U730

Overview This test verifies that EPROM U730 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0009*hex* and C000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1141 The bytes at C0009*hex* and C000B*hex* were not complementary.

Error Index E1142 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name

U600

Overview

This test verifies that EPROM U600 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1151

The bytes at C0008*hex* and C000A*hex* were not complementary.

Error Index E1152

The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U700

Overview This test verifies that EPROM U700 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations C0009*hex* and C000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1161 The bytes at C0009*hex* and C000B*hex* were not complementary.

Error Index E1162 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U612

Overview This test verifies that EPROM U612 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading Bank Decode/Select U530A <23>.
   - Select the EPROM bank 1 by setting Bank Decode/Select U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations A0008*hex* and A000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading Bank Decode/Select U530A <23>.
   - Select the EPROM bank 1 by setting Bank Decode/Select U520-6 <23> high and U520-11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <23>.

Error Index E1171 The bytes at A0008*hex* and A000A*hex* were not complementary.

Error Index E1172 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U712

Overview This test verifies that EPROM U712 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations A0009*hex* and A000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1181 The bytes at A0009*hex* and A000B*hex* were not complementary.

Error Index E1182 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U620

Overview This test verifies that EPROM U620 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations 80008*hex* and 8000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1191 The bytes at 80008*hex* and 8000A*hex* were not complementary.

Error Index E1192 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U720

Overview This test verifies that EPROM U720 <25> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations 80009*hex* and 8000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Read and verify the location byte against the known value.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E11A1 The bytes at 80009*hex* and 8000B*hex* were not complementary.

Error Index E11A2 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name

U250

Overview

This test verifies the integrity of ROM U250 <24> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U250 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1211

The bytes at E0008*hex* and E000A*hex* were not complementary.

Error Index E1212

The computed checksum did not match the stored checksum.

Routine Name U240

Overview This test verifies the integrity of ROM U240 <24> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U240 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1221 The bytes at E0009*hex* and E000B*hex* were not complementary.

Error Index E1222 The computed checksum did not match the stored checksum.

Routine Name

U630

Overview

This test verifies the integrity of EPROM U630 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U630 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1231

The bytes at C0008*hex* and C000A*hex* were not complementary.

Error Index E1232

The computed checksum did not match the stored checksum.

Routine Name U730

Overview This test verifies the integrity of EPROM U730 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Exclusive-or the contents of byte locations C0009*hex* and C000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 0 by setting **Bank Decode/Select** U520-6,11 <23> low.
   - Perform a checksum on all the bytes in U730 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1241 The bytes at C0009*hex* and C000B*hex* were not complementary.

Error Index E1242 The computed checksum did not match the stored checksum.

Routine Name U600

Overview This test verifies the integrity of EPROM U600 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Perform a checksum on all the bytes in U600 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1251 The bytes at C0008*hex* and C000A*hex* were not complementary.

Error Index E1252 The computed checksum did not match the stored checksum.

Routine Name U700

Overview This test verifies the integrity of EPROM U700 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations C0009*hex* and C000B*hex* and verify that the result is FF*hex*
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Perform a checksum on all the bytes in U700 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1261 The bytes at C0009*hex* and C000B*hex* were not complementary.

Error Index E1262 The computed checksum did not match the stored checksum.

Routine Name

U612

Overview

This test verifies the integrity of **EPROM** U612 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations A0008*hex* and A000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Perform a checksum on all the bytes in U612 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1271

The bytes at A0008*hex* and A000A*hex* were not complementary.

Error Index E1272

The computed checksum did not match the stored checksum.

Routine Name U712

Overview This test verifies the integrity of EPROM U712 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations A0009*hex* and A000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Perform a checksum on all the bytes in U712 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1281 The bytes at A0009*hex* and A000B*hex* were not complementary.

Error Index E1282 The computed checksum did not match the stored checksum.

Routine Name U620

Overview This test verifies the integrity of EPROM U620 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations 80008*hex* and 8000A*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Perform a checksum on all the bytes in U620 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1291 The bytes at 80008*hex* and 8000A*hex* were not complementary.

Error Index E1292 The computed checksum did not match the stored checksum.

Routine Name U720

Overview This test verifies the integrity of **EPROM** U720 <25> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Exclusive-or the contents of byte locations 80009*hex* and 8000B*hex* and verify that the result is FF*hex*.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
   - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <23> high and U520-11 <23> low.
   - Perform a checksum on all the bytes in U720 except the first two and verify it against the checksum stored in the first two bytes of the device.
   - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E12A1 The bytes at 80009*hex* and 8000B*hex* were not complementary.

Error Index E12A2 The computed checksum did not match the stored checksum.

Routine Name

Rate

Overview

This test verifies the refresh rate of **DRAM Controller** U410 <26>. A combination of refresh RAS signals is routed to the **Timer** U822 COUNTER 2 <21> through DIAGNSIG(L) <25> and is counted for a short duration.

Description

1. Inhibit the counter from counting.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16:L <21>.
2. Program the counter to count down with 6464*hex* as the starting counter value.
   - Write B0*hex*, 64*hex*, and 64*hex* to Timer Control Word Register and Count Register low and high bytes, respectively.
3. Route **Memory Diagnostic Signal Select's** <25> combined ROW COLUMN BUS <25> signal to DIAGNSIG(L) <25> on the Executive Bus by writing 02*hex* to U322B <25>.
4. Enable the counter to count DIAGNSIG(L) <21>.
   - Write 80*hex* to U720 <21>. This sets G2 U822-16:H and connects DIAGNSIG(L) <21> to CLK2 U822-18.
5. Perform a software delay loop.
6. Disable the counter from counting.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that the gate signal G2 U822-16 is low.
7. Read and verify **Timer** COUNTER 2 Count Register low and high bytes.
8. Stop **Timer** COUNTER 2 by programming its Control Word Register with B0*hex*.
9. Disable **Memory Diagnostic Signal Select's** <25> combined ROW COLUMN BUS <25> signal so that it does not appear in the Executive Bus by writing 00*hex* to U352B <25>.

Error Index E1311

The refresh rate was not within the expected tolerance.

Caveats

Other circuits that gate signals on to bus signal DIAGNSIG(L) may cause this test to fail. This test may also fail if the Main Processor Board **Clock Generator** <23> reference Y950 is out of tolerance since it is being used as the timing reference for the test via software delay mechanism.

Routine Name Config (Configuration)

Overview This test verifies **DRAM Configuration** <26> jumper selections and the software bank selectability through **Bank Decode/Select** <23> circuit.

Description

1. Read the **Memory Configuration Readback** U832 <25>. Only the upper four bits of this port represent the jumper positions. The lower four bits represent the current bank selection.
2. Verify the position of the EPROM size selection jumper J541 <25> for 27512 type EPROMs.
3. Verify the position of the virtual or real memory addressing jumper J520 <26> for real memory addressing.
4. Verify the combination of RAM device type and number of RAM banks jumper J501 <26> and J521 <26>, respectively, for 64k X 4 RAM device type and two banks.
5. Store the current bank by reading **Bank Decode/Select** U530A <23>.
6. Verify bank selectability by doing a "walking one's" test on the 4-bit bank select register U520 <23>.
   - Write patterns 01*hex*, 02*hex*, 04*hex*, and 08*hex* to **Bank Decode/Select** U520 <23>. Read and verify the written patterns from **Bank Decode/Select** U530A <23> and **Memory Configuration Readback** U832 <25> (only the lower four bits).
7. Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1411 27512 EPROM has not been selected through J541.

Error Index E1412 Real mode selection has not been made through J520.

Error Index E1413 The RAM device and number of banks combination selected through J501 and J521, respectively, are not appropriate.

Error Index E1414 The bank select mechanism is not functioning properly. The display shows the first pattern that did not verify.

Routine Name

Data Lines

Overview

This test verifies the the data lines to **DRAM** <26> by performing a "walking one's" test on **DRAM** address 3BFFE*hex*.

Description

1. Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
2. Select **DRAM** bank 1 by setting **Bank Decode/Select** U520-3:H <23>.
3. Perform a "walking one's" test on **DRAM** address 3BFFE*hex*.
   - Write the pattern 0001*hex* to address 3BFFE*hex*. Read the same address and verify that it was 0001*hex*. Continue this write/read/ verify sequence with the patterns 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*.
4. Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1421

The pattern read from **DRAM** <26> address 3BFFE*hex* was not the same pattern written. The display shows the first pattern that did not verify. The following table shows the RAM device that is probably bad.

Data Bits	Device
0–3	U222
4–7	U220
8–11	U212
12–15	U210

Routine Name Address/Data

Overview This test verifies the address lines and data integrity of the **DRAM** <26> by performing a RAM test on all **DRAM** locations.

Description

1. Store the current bank selection by reading **Bank Decode/Select** U530A <23>.
2. Select **DRAM** memory bank 1 by setting **Bank Decode/Select** U520-3:H <23>.
3. Verify **DRAM** address range 2C000-3BFFF*hex* (64k bytes). Terminate test if any verify operation fails.
   - Fill address range 2C000-3BFFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 2C000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 2C000*hex*. Increment address and continue this read/verify/write sequence until address 3BFFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 2C000*hex* for CCCC*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
4. If **DRAM** address range 2C000-3BFFF*hex* is free of faults, then test **DRAM** address ranges 00000-0FFFF*hex* (64k bytes), 10000-1FFFF*hex* (64k bytes), and 20000-2BFFF*hex* (48k bytes) separately.
   - Copy the 64k bytes or 48k bytes of **DRAM** to the uppermost 64k bytes (2C000-3BFFF*hex*).
   - Test **DRAM** locations using the procedure outlined in step 3.
   - Restore the **DRAM** locations that were tested by copying back information stored in uppermost 64k bytes.
5. Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <23>.

Error Index E1431 The pattern read from the displayed memory location was not the pattern written. The following table may be used to help isolate the bad RAM device.

	Data Bits			
Address	**0–3**	**4–7**	**8–11**	**11–15**
XXXX0*hex*	U122	U120	U112	U110
XXXX1*hex*	U122	U120	U112	U110
XXXX4*hex*	U122	U120	U112	U110
XXXX5*hex*	U122	U120	U112	U110
XXXX8*hex*	U122	U120	U112	U110
XXXX9*hex*	U122	U120	U112	U110
XXXXC*hex*	U122	U120	U112	U110
XXXXD*hex*	U122	U120	U112	U110
XXXX2*hex*	U222	U220	U212	U210
XXXX3*hex*	U222	U220	U212	U210
XXXX6*hex*	U222	U220	U212	U210
XXXX7*hex*	U222	U220	U212	U210
XXXXA*hex*	U222	U220	U212	U210
XXXXB*hex*	U222	U220	U212	U210
XXXXE*hex*	U222	U220	U212	U210
XXXXF*hex*	U222	U220	U212	U210

X—don't care.

Routine Name

Battery

Overview

This test verifies the **Non-Volatile RAM <24>** volatility (i.e., that **RAM Battery <24>** was functional on power-up) by checking the values of two confidence words located in the NVRAM.

Description

1. Read and verify that the confidence word at NVRAM location 3C000*hex* is DEAD*hex*.

2. Read and verify that the confidence word at NVRAM location 3C002*hex* is 2152*hex* (complement of DEAD*hex*).

Error Index E1511

One of the two confidence words was not correct.

The complementary confidence words are initialized by the normal operating firmware in the system. So, if the instrument has never entered normal operating mode, this test will fail. Once initialized, the confidence words should remain complementary as long as the **RAM Battery <24>** is good.

Routine Name

Data Lines

Overview

This test verifies the data lines to the **Non-Volatile RAM** <24> by performing a "walking one's" test on NVRAM location 3FFFE*hex*.

CAUTION

Turning the instrument power off during the execution of this test may result in losing some or all of the Non-Volatile RAM data (such as stored settings, calibration constants, etc.). This could affect normal instrument operation in unpredictable ways.

Description

1. Save the contents of **Non-Volatile RAM** address 3FFFE*hex*.
2. Perform a "walking one's" test on **Non-Volatile RAM** address 3FFFE*hex*.
   - Write the pattern 0001*hex* to **Non-Volatile RAM** address 3FFFE*hex*. Read the same address and verify that it was 0001*hex*. Continue this write/read/verify sequence with patterns 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*.
3. Restore **Non-Volatile RAM** address 3FFFE*hex*.

Error Index E1521

The read pattern did not match the written pattern. The following table shows the device that is probably bad:

Data bits	Device
0 – 7	U270
8 – 15	U260

Caveats

Turning the instrument power off during the execution of this test may result in losing the **Non-Volatile RAM** contents of the location being tested. This could affect normal instrument operation in ways not predictable. If the instrument is disabled due to a power down, refer to Section 3, Checks and Adjustments, for the procedures necessary to restore cal constants. Stored settings can not be recovered.

Routine Name Address/Data

Overview This test verifies the address lines and data integrity of the **Non-Volatile RAM** <24> by performing a RAM test on all NVRAM locations.

CAUTION

*Turning the instrument power off during the execution of this test may result in losing some or all of the **Non-Volatile RAM** data (such as stored settings, calibration constants, etc.). This could affect normal instrument operation in unpredictable ways.*

Description

1. Verify **DRAM** <26> address range 38000-3BFFF*hex* (16k bytes). Terminate test if any verify operation fails.
   - Fill address range 38000-3BFFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 38000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 38000*hex*. Increment the address and continue this write/read/verify sequence until address 3BFFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 38000*hex*, for CCCC*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
2. If **DRAM** <26> address range 38000-3BFFF*hex* is free of faults, then verify **Non-Volatile RAM** <24> address range 3C000-3FFFF*hex*.
   - Save data in **Non-Volatile RAM** address range 3C000-3FFFF*hex* (16k bytes) by copying it to **DRAM** address range 38000-3BFFF*hex*.
   - Perform the same procedure as in step 1 for the address range 3C000-3FFFF*hex*. Since the NVRAM devices are 16k X 8 devices, more patterns are required to verify them—patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, 5555*hex*, and AAAA*hex* are used.
   - Make one last read/verify pass for the pattern AAAA*hex* starting at address 3C000*hex*.
   - Copy **Non-Volatile RAM** data from **DRAM** address range 38000-3BFFF*hex* to **Non-Volatile RAM** address range 3C000-3FFFF*hex*. In order to restore the **Non-Volatile RAM** confidence words last, the copying is done from the last address to the first (i.e., from 3FFFE*hex* to 3C000*hex*).

Error Index E1531 The pattern read from the displayed memory location in the **DRAM** <26> or the **Non-Volatile RAM** <24> was not the pattern written.

If the displayed memory location was between 3C000-3FFFF*hex* and the low byte of the read pattern did not match the written pattern, then the probable faulty device is U270. If the high byte of the read pattern did not match the written pattern, then the probable faulty device is U260.

See Also

Exec Control Dynamic RAM Address/Data (E143X) if the faulty address was between 38000-3BFFF*hex*.

Caveats

Turning the instrument power off during the execution of this test may result in losing some or all of the **Non-Volatile RAM** data (such as stored settings, calibration constants, etc.). This may affect normal instrument operation. If the instrument is disabled due to a power down, refer to Section 3, Checks and Adjustments, for the procedures necessary to restore cal constants. Stored settings can not be recovered.

Routine Name

Master

Overview

This test verifies the **Interrupt Controllers** MASTER U350 <24> Mask Register. Each bit in the register is set high and low to verify the interrupt masking capability of this device.

Description

1. Disable external interrupts at the processor by setting processor interrupt enable flag low.

2. Perform a "walking one's" test on the MASTER Mask Register.

   - Write, read, and verify patterns 01*hex*, 02*hex*, 04*hex*, 08*hex*, 10*hex*, 20*hex*, 40*hex*, and 80*hex*.

3. Mask all MASTER interrupts by writing FF*hex* to its Mask Register.

4. Enable external interrupts at the processor by setting processor interrupt enable flag high.

Error Index E1611

The value read did not match the value written. The Mask Register's bits correspond to the following interrupts:

Bit 0 : Not used
Bit 1 : **Interrupt Controllers** SLAVE 1 U360 <24>
Bit 2 : **Timer** U822 <21> COUNTER 0
Bit 3 : **Interrupt Controllers** SLAVE 3 U370 <24>
Bit 4 : **Front Panel Controller** U103 <2>
Bit 5 : Not used
Bit 6 : **Printer Controller** U430 <4>
Bit 7 : Not used

If the actual value reads zero, then the interrupt represented by the set bit in the expected value is unmaskable.

If the actual value has more than one bit set, then the interrupt represented by the extra set bit(s) in the actual value is masked all the time.

Routine Name Slave 1

Overview This test verifies the **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register. Each bit in the register is set high and low to verify the interrupt masking capability of this device.

Description

1. Disable external interrupts at the processor by setting the processor interrupt enable flag low.
2. Perform a "walking one's" test on the SLAVE 1 Mask Register.
   - Write, read, and verify patterns 01*hex*, 02*hex*, 04*hex*, 08*hex*, 10*hex*, 20*hex*, 40*hex*, and 80*hex*.
3. Mask all SLAVE 1 interrupts by writing FF*hex* to its Mask Register.
4. Enable external interrupts at the processor by setting processor interrupt enable flag high.

Error Index E1621 The value read did not match the value written. The Mask Register's bits correspond to the following interrupts:

Bit 0 : **MMU Gate Array** U210 <27> sequential address generator
Bit 1 : **MMU Gate Array** U210 <27> digitizer end message
Bit 2 : **MMU Gate Array** U210 <27> display talk request
Bit 3 : Not used
Bit 4 : **Serial Data Interface** U330 <21>
Bit 5 : Not used
Bit 6 : **DMA Controller** U800 <24>
Bit 7 : Not used

If the actual value reads zero, then the interrupt represented by the set bit in the expected value is unmaskable.

If the actual value has more than one bit set, then the interrupt represented by the extra set bit(s) in the actual value is masked all the time.

Routine Name Slave 3

Overview This test verifies the **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register. Each bit in the register is set high and low to verify the interrupt masking capability of this device.

Description

1. Disable external interrupts at the processor by setting processor interrupt enable flag low.
2. Perform a "walking one's" test on the SLAVE 3 Mask Register.
   - Write, read, and verify patterns 01*hex*, 02*hex*, 04*hex*, 08*hex*, 10*hex*, 20*hex*, 40*hex*, and 80*hex*.
3. Mask all the SLAVE 3 interrupts by writing FF*hex* to its Mask Register.
4. Enable external interrupts at the processor by setting processor interrupt enable flag high.

Error Index E1631 The value read did not match the value written. The Mask Register's bits correspond to the following interrupts:

Bit 0 : **Timer** U822 <21> COUNTER 1
Bit 1 : **Timer** U822 <21> COUNTER 2
Bit 2 : **GPIB Controller** U410 <3>
Bit 3 : Not used
Bit 4 : **Std RS232 Controller** U311 <3>
Bit 5 : **Optional RS232 Controller** U331 <3>
Bit 6 : **Real Time Clock** U614 <21>
Bit 7 : Not used

If the actual value reads zero, then the interrupt represented by the set bit in the expected value is unmaskable.

If the actual value has more than one bit set, then the interrupt represented by the extra set bit(s) in the actual value is masked all the time.

Routine Name Timer 0

Overview This test verifies the **Timer** U822 COUNTER 0 <21> and the **Clock Generator** <21> by counting 2MHz clock for a short duration.

Description

1. Program the **Timer** COUNTER 0 to count down with 6464*hex* as the starting count.
   - Write 34*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
2. Perform a software delay.
3. Stop the **Timer** COUNTER 0 by programming its Control Word Register with 34*hex*.
4. Read and verify the Count Register high and low bytes.
5. Restore the **Timer** COUNTER 0 for the normal operating firmware.

Error Index E1711 The **Timer** COUNTER 0 counted faster than expected.

Error Index E1712 The **Timer** COUNTER 0 counted slower than expected.

Caveats This test may fail if the Main Processor Board **Clock Generator** <23> reference Y950 is out of tolerance since it is being used as the timing reference for the test via the software delay mechanism.

Routine Name Timer 1

Overview This test verifies the **Timer** U822 COUNTER 1 <21> by counting 6 MHz clock for a short duration.

Description

1. Configure the **Timer** COUNTER 1 to count 6 MHz clock.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21> so that G1 U822-14 is high and PCLK(H) connected to CLK1 U822-15.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write 70*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Perform a software delay.
4. Stop the **Timer** COUNTER 1 by programming its Control Word Register with 70*hex*.
5. Read and verify the Count Register low and high bytes.

Error Index E1721 The **Timer** COUNTER 1 counted faster than expected.

Error Index E1722 The Timer COUNTER 1 counted slower than expected.

Routine Name

Timer 2

Overview

This test verifies the **Timer** U822 COUNTER 2 <21> counting capability by using the gate control to stop and resume counting.

Description

1. Configure the **Timer** COUNTER 2 to count 6 MHz clock.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is high and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write B0*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Perform a software delay.
4. Disable the counter.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is low.
5. Perform a software delay.
6. Enable the counter.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is high.
7. Perform a software delay.
8. Stop the **Timer** COUNTER 2 by programming its Control Word Register with B0*hex*.
9. Read and verify the Count Register low and high bytes.

Error Index E1731

The **Timer** COUNTER 2 either does not count correctly or its gate has no effect on its counting.

Routine Name Diagn Signal

Overview This test verifies the **Timer** U822 COUNTER 2 <21> external event counting capability through DIAGNSIG(L) by using the **Wait State Generator** <24> and the **Wait State Diagnostic Enable** <24> to generate a known number of one wait states.

Description

1. Configure the **Timer** COUNTER 2 to count DIAGNSIG(L).
   - Write 80*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is high and DIAGNSIG(L) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write B0*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Enable the **Wait State Diagnostic Enable's** DIAGNSIG(L) <24> signal to the Executive Bus by writing 01*hex* to U110B <24>.
4. Generate 100*hex* one wait states by repeatedly writing to the **Timer** <21> COUNTER 2 Control Word Register. Each write to the **Timer** generates one wait state through **Wait State Generator** <24>. Each wait state generated causes DIAGNSIG(L) to toggle.
5. Disable the **Wait State Diagnostic Enable's** DIAGNSIG(L) <24> to the Executive Bus by writing 00*hex* to U110B <24>.
6. Stop the **Timer** COUNTER 2 by programming its Control Word Register with B0*hex*.
7. Read and verify the Count Register low and high bytes.

Error Index E1741 The counter counted more external events than expected.

Error Index E1742 The counter counted less external events than expected.

Routine Name

Timer 0

Overview

This test verifies the **Timer** U822 <21> COUNTER 0 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Clear the **Timer** COUNTER 0 interrupt.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21>, 34*hex* to **Timer** U822 <21> Control Word Register, and then toggle **Timer 0 Interrupt Reset** U812B-13 <21>.
2. Verify that there is no **Timer** COUNTER 0 interrupt pending at the **Interrupt Controllers** MASTER U350 <24>.
   - Read **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR2 U350-20 <24> is low.
3. If there is no **Timer** COUNTER 0 interrupt pending, then enable **Timer** COUNTER 0 interrupt by setting bit 2 of the **Interrupt Controllers** MASTER U350 <24> Mask Register low.
4. Complete the programming for **Timer** COUNTER 0 with 80*hex* as the starting count.
   - Write 80*hex* and 00*hex* to Count Register low and high bytes. This starts the counter.
5. Perform a software delay.
6. Verify that the **Timer** COUNTER 0 interrupt did occur.
7. Disable the **Timer** COUNTER 0 interrupt by setting bit 2 of the **Interrupt Controllers** MASTER U350 <24> Mask Register high.
8. Restore the **Timer** COUNTER 0 for the normal operating firmware.

Error Index E1811

The **Timer** COUNTER 0 interrupt did not occur.

Error Index E1812

The **Timer** COUNTER 0 interrupt could not be cleared at the **Interrupt Controllers** MASTER U350.

Routine Name

Timer 1

Overview

This test verifies the **Timer** U822 <21> COUNTER 1 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Configure the **Timer** COUNTER 1 to count 6 MHz clock.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21> so that G1 U822-14 is high and PCLK(H) is connected to CLK1 U822-15.
2. Clear **Timer** COUNTER 1 interrupt.
   - Write 70*hex* to the Control Word Register.
3. Verify that there is no **Timer** COUNTER 1 interrupt pending at the **Interrupt Controllers** SLAVE 3 U370 <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR0 U370-18 is low.
4. If there is no Timer COUNTER 1 interrupt pending, then enable the **Timer** COUNTER 1 interrupt by setting bit 0 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register low.
5. Complete programming the **Timer** COUNTER 1 with 80*hex* as the starting count.
   - Write 80*hex* and 00*hex* to the Count Register low and high bytes. This starts the counter.
6. Perform a software delay.
7. Verify that **Timer** COUNTER 1 interrupt did occur.
8. Disable the **Timer** COUNTER 1 interrupt by setting bit 0 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register high.

Error Index E1821

The **Timer** COUNTER 1 interrupt did not occur.

Error Index E1822

The **Timer** COUNTER 1 interrupt could not be cleared at the **Interrupt Controllers** SLAVE 3 U370.

Routine Name

Timer 2

Overview

This test verifies the **Timer** U822 <21> COUNTER 2 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Configure the **Timer** COUNTER 2 to count 6 MHz clock.
   - Write 00*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is high and PCLK(H) is connected to CLK2 U822-18.
2. Clear **Timer** COUNTER 2 interrupt.
   - Write B0*hex* to the Control Word Register.
3. Verify that there is no **Timer** COUNTER 2 interrupt pending at the **Interrupt Controllers** SLAVE 3 U370 <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR0 U370-19 is low.
4. If there is no **Timer** COUNTER 2 interrupt pending, then enable the **Timer** COUNTER 2 interrupt by setting bit 1 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register low.
5. Complete programming the **Timer** COUNTER 2 with 80*hex* as the starting count.
   - Write 80*hex* and 00*hex* to the Count Register low and high bytes. This starts the counter.
6. Perform a software delay.
7. Verify that **Timer** COUNTER 2 interrupt did occur.
8. Disable the **Timer** COUNTER 2 interrupt by setting bit 1 of **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register and bit 3 of **Interrupt Controllers** MASTER U350 <24> Mask Register high.

Error Index E1831

The **Timer** COUNTER 2 interrupt did not occur.

Error Index E1832

The **Timer** COUNTER 2 interrupt could not be cleared at the **Interrupt Controllers** SLAVE 3 U370.

Routine Name Zero Wait

Overview This test verifies the **Wait State Generator's** <24> ability to generate zero wait states. The **Timer** U822 COUNTER 2 <21> is used to count the number of wait states.

Description

1. Configure the **Timer** COUNTER 2 <21> to count 6 MHz clock.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write B0*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Enable the **Wait State Diagnostic Enable** circuit by writing 01*hex* to U110B <24> so that U411B-9 <24> is gated onto DIAGNSIG(L) <21>.
4. Generate 100*hex* zero wait requests by repeatedly writing 00*hex* to **Memory Diagnostic Signal Select** U352B <25>. The **Wait State Generator** sets DIAGNSIG(L) low when it is generating wait states. The **Timer** COUNTER 2 is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflects the number of wait states generated.
5. Disable the **Wait State Diagnostic Enable** circuit by writing 00*hex* to U110B <24> so that U411B-9 <24> is no longer gated onto DIAGNSIG(L) <21>.
6. Stop the **Timer** COUNTER 2 by programming its Control Word Register with B0*hex*.
7. Read and verify the **Timer** COUNTER 2 Count Register low and high bytes.

Error Index E1911 The number of wait states was more than zero wait states.

Routine Name One Wait

Overview This test verifies the **Wait State Generator's** <24> ability to generate one wait states. The **Timer** COUNTER 2 U822 <21> is used to count the number of wait states generated.

Description

1. Configure the **Timer** COUNTER 2 <21> to count 6 MHz clock.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write B0*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Enable the **Wait State Diagnostic Enable** circuit by writing 01*hex* to U110B <24> so that U411B-9 <24> is gated onto DIAGNSIG(L) <21>.
4. Generate 100*hex* one wait requests by repeatedly writing 70*hex* to **Timer** COUNTER 1 Control Word Register. Since the **Timer** device is a one wait state device, each access to it will generate a one wait state. The **Wait State Generator** sets DIAGNSIG(L) low when it is generating wait states. The **Timer** COUNTER 2 is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflects the number of wait states generated.
5. Disable the **Wait State Diagnostic Enable** circuit by writing 00*hex* to U110B <24> so that U411B-9 <24> is no longer gated onto DIAGNSIG(L) <21>.
6. Stop the **Timer** COUNTER 2 by programming its Control Word Register with B0*hex*.
7. Read and verify the **Timer** COUNTER 2 Count Register low and high bytes.

Error Index E1921 The **Wait State Generator** generated more than one wait states for the **Timer** <21> device.

Routine Name

Two Wait

Overview

This test verifies the **Wait State Generator's** <24> ability to generate two wait states. The **Timer** COUNTER 2 U822 <21> is used to count the number of wait states.

Description

1. Configure the **Timer** COUNTER 2 <21> to count 6 MHz clock.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write B0*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Enable the **Wait State Diagnostic Enable** circuit by writing 01*hex* to U110B <24> so that U411B-9 <24> is gated onto DIAGNSIG(L) <21>.
4. Generate 100*hex* two wait requests by repeatedly writing 60*hex* to the the **Front Panel Controller** U103 <2>. Since the **Front Panel Controller** device is a two wait state device each access to it will generate two wait states. The **Wait State Generator** sets DIAGNSIG(L) low when it is generating wait states. The **Timer** COUNTER 2 is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflects the number of wait states generated.
5. Disable the **Wait State Diagnostic Enable** circuit by writing 00*hex* to U110B <24> so that U411B-9 <24> is no longer gated onto DIAGNSIG(L) <21>.
6. Stop the **Timer** COUNTER 2 by writing B0*hex* to its Control Word Register.
7. Read and verify the **Timer** COUNTER 2 Count Register low and high bytes.

Error Index E1931

The Wait State Generator generated more than two wait states for the **Front Panel Controller** <2> device.

Routine Name

Four Wait

Overview

This test verifies the **Wait State Generator's** <24> ability to generate four wait states. The **Timer** COUNTER 2 U822 <21> is used to count the number of wait states.

Description

1. Configure the **Timer** COUNTER 2 <21> to count 6 MHz clock.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write B0*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Enable the **Wait State Diagnostic Enable** circuit by writing 01*hex* to U110B <24> so that U411B-9 <24> is gated onto DIAGNSIG(L) <21>.
4. Generate 100*hex* four wait requests by repeatedly writing A6*hex* to the **Printer Controller** U430 <4> Control Word Register. Since the **Printer Controller** is a four wait state device, each access to it will generate four wait states. The **Wait State Generator** sets DIAGNSIG(L) low when it is generating wait states. The **Timer** COUNTER 2 is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflect the number of wait states generated.
5. Disable the **Wait State Diagnostic Enable** circuit by writing 00*hex* to U110B <24> so that U411B-9 <24> is no longer gated onto DIAGNSIG(L) <21>.
6. Stop the **Timer** COUNTER 2 by programming its Control Word Register with B0*hex*.
7. Read and verify the **Timer** COUNTER 2 Count Register low and high bytes.

Error Index E1941

The **Wait State Generator** generated more than four wait states for the **Printer Controller** <4> device.

Caveats

There is no signal to represent a four wait state request coming into the **Wait State Generator** <24>. This is the default maximum number of wait states generated by this circuit.

Routine Name Zero Wait

Overview This test verifies the **Memory Wait State Gen** <25> circuit on the Memory Board. The **Timer** U822 <21> COUNTER 2 is used to count the number of wait states.

Description

1. Configure the **Timer** COUNTER 2 <21> to count 6 MHz clock.
   - Write 40*hex* to **Timer Configuration Logic** U720 <21> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464*hex* as the starting count.
   - Write B0*hex*, 64*hex*, and 64*hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Enable the **Memory Diagnostic Signal Select** circuit by writing 01*hex* to U352B <25> so that U820B-9 <24> is gated onto DIAGNSIG(L) <21>.
4. Generate 100*hex* zero wait requests by repeatedly reading an **EPROM** <24> location D0000*hex*. Each time the EPROM is selected, **Memory Wait State Gen** <25> generates zero wait states. The **Memory Wait State Gen** sets DIAGNSIG(L) low when it is generating wait states. As the **Timer** COUNTER 2 is set up to count 6 MHz clock (wait states are specified in terms of the PCLK(H) cycles which is at 6 MHz) during the time DIAGNSIG(L) is low, the number of wait states generated by the **Memory Wait State Gen** is reflected in the counter.
5. Disable the **Memory Diagnostic Signal Select** circuit by writing 00*hex* to U352B <24> so that U820B-9 <24> is no longer gated to DIAGNSIG(L) <21>.
6. Stop the **Timer** COUNTER 2 by programming its Control Word Register with B0*hex*.
7. Read and verify the **Timer** COUNTER 2 Count Register low and high bytes.

Error Index E1A11 The **Memory Wait State Gen** generated more than zero wait states.

Routine Name Floating Pt (Floating Point)

Overview This test verifies the floating point and transcendental function capabilities of the **Numeric Processor Extension** U500 <23>.

Description

1. Compute the following equation using the **Numeric Processor Extension** <23> and verify that the result is within a tolerance of ±0.005 of the expected value.

$$\frac{\log_e(15.3) * \log_{10}(23.5) * \exp(0.12)}{\text{atan}[\sin(0.7071) * \cos(0.5) * \tan(0.5774)]}$$

Error Index E1B11 The result from the floating point calculation was not 11.8582 ±0.005.

Routine Name

DMA 0

Overview

This test verifies the **DMA Controller** <24> channel 0. As the **DMA Controller** is an option in the instrument, this test checks to see if the option is present. If the option is not present, no further testing is done. Otherwise, a memory to memory DMA transfer is performed and verified.

Description

1. Check to see if the **DMA Controller** is present.
   - Write, read, and verify 7F01*hex* to the General Mode Register of the **DMA Controller**.
2. If the **DMA Controller** is present (i.e., 7F01*hex* was verified in step 1), construct the command block for the transfer at **DRAM** <26> address 30000*hex*. Initialize the source and destination memory locations.
   - Write C8DD*hex*, 0013*hex*, 0020*hex*, 0013*hex*, 0040*hex*, 0000*hex*, 0010*hex*, 0000*hex*, and 1401*hex* to locations starting from 30000*hex*.
   - Initialize the source **DRAM** address range 30020*hex*-3002A*hex* with patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex*.
   - Initialize the destination **DRAM** address range 30040*hex*-30049*hex* with pattern 8000*hex*.
3. Program the **DMA Controller** channel 0 to read the command block from memory and start the transfer of five words from 30020*hex* to 30040*hex*.
   - Write 7F01*hex*, 00*hex*, 00*hex*, 0013*hex*, 0000*hex*, and 001A*hex* to General Mode Register, General Burst Register, General Delay Register, Command Pointer Register High, Command Pointer Register Low and General Command Register, respectively.
4. Perform a software delay.
5. Clear the DMA interrupt at the **DMA Controller** and verify that the transfer was successful.
   - Write 0018*hex* General Command Register.
   - Compare the contents of the destination and source memory locations and verify that they are the same.

Error Index E1C11

The DMA transfer was not successful. The display shows the first pattern that did not verify.

Error Index ????

The DMA option was not found.

Routine Name

Interrupt

Overview

This test verifies the **DMA Controller** <24> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. As the **DMA Controller** is an option in the instrument, this test checks to see if the option is present. If the option is not present, no further testing is done. Otherwise, a memory to memory DMA transfer is completed to generate the interrupt.

Description

1. Check to see if the **DMA Controller** is present.
   - Write, read, and verify 7F01*hex* to the General Mode Register of the **DMA Controller**.
2. If the DMA Controller is present (i.e., 7F01*hex* is verified in the step 1), program the **DMA Controller** channel 0 and clear its interrupt, if any.
   - Write 7B01*hex*, 0000*hex*, 0000*hex*, 0013*hex*, 0000*hex*, and 0018*hex* to General Mode Register, General Burst Register, General Delay Register, Command Pointer Register High, Command Pointer Register Low and General Command Register, respectively.
3. Verify that there is no DMA interrupt pending at the **Interrupt Controllers** SLAVE 1 U360 <24>.
   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR6 U360-24 is low.
4. If there is no DMA interrupt pending, construct the command block for the transfer at **DRAM** address 30000*hex*. Initialize the source and destination memory locations.
   - Write C8DD*hex*, 0013*hex*, 0020*hex*, 0013*hex*, 0040*hex*, 0000*hex*, 0010*hex*, 0000*hex*, 1401*hex* to locations starting from 30000*hex*.
   - Initialize the source **DRAM** address range 30020*hex*-3002A*hex* with patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex*.
   - Initialize the destination **DRAM** address range 30040*hex*-30049*hex* with pattern 8000*hex*.
5. Enable the DMA interrupt by setting bit 6 of **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register and bit 1 of **Interrupt Controllers** MASTER U350 <24> Mask Register low.
6. Start the DMA transfer by writing 1A*hex* to General Command Register.
7. Perform a software delay.
8. Verify that the DMA interrupt occurred.
9. Disable the DMA interrupt by setting bit 6 of **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register and bit 1 of Interrupt Controllers MASTER U350 <24> Mask Register high.

Error Index E1C21 The DMA interrupt did not occur.

Error Index E1C22 The DMA interrupt could not be cleared at **Interrupt Controllers** SLAVE 1 U360.

Error Index ???? The DMA option is not present.

Routine Name RAM

Overview This test verifies the **Front Panel Controller** U103 display RAM <2> by performing a RAM test on the display RAM locations.

Description

1. Initialize the **Front Panel Controller** <2>.
   - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
2. Save the contents of the **Front Panel Controller** display RAM <2>.
   - Write 70*hex* to the Command Register and read eight bytes from the Data Register.
3. Verify the RAM locations.
   - Fill all eight locations with the pattern AA*hex*.
   - Read and verify location 0 for AA*hex*. If so, write CC*hex* to location 0. Increment the address and continue the read/verify/write sequence until location 7 is reached.
   - Read and verify location 7 for CC*hex*. If so, write F0*hex* to location 7. Decrement the address and continue the read/verify/write sequence until location 0 is reached.
   - Repeat the read/verify/write sequence starting at location 0 and 7 for alternative passes, for patterns F0*hex* and 55*hex*.
   - Make one last read/verify pass for the pattern 55*hex* starting at location 7 and also restore the saved contents of the display RAM.
4. Initialize the **Front Panel Controller** <2>.
   - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.

Error Index E2111 The pattern read did not match the pattern written from the **Front Panel Controller** display RAM <2>. The display shows the first pattern that did not verify.

Routine Name RAM Control

Overview This test verifies various display RAM controls of the **Front Panel Controller** U103 <2> by exercising these controls.

Description

1. Save the contents of the **Front Panel Controller** display RAM <2>.
   - Write 70*hex* to the Command Register and read eight bytes from the Data Register.
2. Initialize the **Front Panel Controller** <2>.
   - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
3. Verify that the display RAM can be cleared.
   - Write DD*hex* to the Command Register and read the **Front Panel Controller** Status Word Register and verify that the display RAM is unavailable (i.e., bit 7 is high).
   - Perform a software delay (approximately 300µs).
   - Read the **Front Panel Controller** Status Word Register and verify that the display RAM is available (i.e., bit 7 is low).
4. Verify that all the RAM locations may be set high.
   - Write 60*hex* to the Command Register. Read eight bytes from the Data Register and verify that each of them is FF*hex*.
5. Verify that all the RAM locations may be set low.
   - Write D3*hex* to the Command Register.
   - Write 60*hex* to the Command Register. Read eight bytes from the Data Register and verify that each of them is 00*hex*.
6. Initialize the **Front Panel Controller** <2>.
   - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
7. Restore the display RAM locations by writing back the stored contents of the display RAM.
   - Write 90*hex* to the Command Register and write eight bytes to the Data Register.

Error Index E2121 The **Front Panel Controller** display RAM <2> was available when it should have been unavailable (during the time it was being cleared).

Error Index E2122 The **Front Panel Controller** display RAM <2> was unavailable when it should have been available (150µs after it was cleared).

Error Index E2123 The **Front Panel Controller** display RAM <2> could not be set to ones, i.e., writing DD*hex* to the Command Register did not set the display RAM locations to ones.

Error Index E2124 The **Front Panel Controller** display RAM <2> could not be set to zeros, i.e., writing D3*hex* to the Command Register did not set the display RAM locations to zeros.

Routine Name

Interrupt

Overview

This test verifies the **Front Panel Controller** <2> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The infrared LEDs are disabled and enabled to generate the interrupt.

Description

1. Initialize the **Front Panel Controller** <2>.
   - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
2. Verify that there is no front panel interrupt pending at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <24> is low.
3. If there was no pending front panel interrupt, enable the front panel interrupt by setting the **Interrupt Controllers** MASTER U350 <24> Mask Register bit 4 low.
4. Disable front panel infrared LEDs by writing 00*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11(L) <2> is low.
5. Perform a software delay.
6. Enable front panel infrared LEDs by writing 01*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11(L) <2> is high.
7. Perform a software delay.
8. Verify that the front panel interrupt did occur.
9. Disable the front panel interrupt setting **Interrupt Controllers** MASTER U350 Mask Register bit 4 high.
10. Initialize the **Front Panel Controller**.
    - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.

Error Index E2131

The **Front Panel Controller** <2> interrupt did not occur.

Error Index E2132

The **Front Panel Controller** <2> interrupt could not be cleared at the **Interrupt Controllers** MASTER U350 <24>.

Routine Name

Open

Overview

This test verifies the front panel hard keys (buttons) on the I-R Touch Panel <1> and Front Panel Button boards <2> by checking that none of the front panel hard keys is stuck in the closed position.

Description

1. Disable front panel infrared LEDs by writing 00*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is low.

2. Wait for the front panel infrared LEDs to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).

   - Read the front panel sensor RAM in the **Front Panel Controller** U103 <2> by writing 60*hex* to the Command Register and reading eight bytes from the Data Register.
   - Clear the front panel interrupt by writing E0*hex* to the Command Register.
   - Perform a software delay.
   - Read the **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <24> is low.
   - Repeat the previous four steps until IR4 U350-22 <24> is low.

3. Enable front panel infrared LEDs by writing 01*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is high.

4. Perform a software delay.

5. Read the sensor RAM in the **Front Panel Controller** U103 <2> after it stabilizes.

   - Use the same procedure outlined in step 2.

6. Verify that the hard keys status in the sensor RAM is open (i.e., hard keys sensor RAM bits are all 1).

Error Index E2211

One or more hard keys was stuck in the closed position.

There are 10 hard keys in the front panel. The state of each of them is represented by a bit in the expected and actual values. A low (0) bit in the actual field represents the hard key (button) that was closed. The following table presents the bit encoding scheme for the hard keys.

Data Bit (Switch)	Hardkey
bit 0 (S100 <2>)	Digitizer Run/Stop
bit 1 (S200 <2>)	Autoset
bit 2 (S300 <2>)	Hardcopy
bit 3 (S400 <2>)	Enhanced Accuracy
bit 4 (S7 <1>)	Waveform
bit 5 (S6 <1>)	Trigger
bit 6 (S5 <1>)	Measure
bit 7 (S4 <1>)	Store/Recall
bit 8 (S3 <1>)	Utility
bit 9 (S1 <1>)	Touch Panel On/Off

See Also The Front Panel Verify Hard Keys test.

Routine Name

Row Open

Overview

This test verifies the 22 front panel infrared emitter/receptor row pairs (y-axis input sensors) in **Touch Panel Bezel Sensors** <1> by checking that all the row receptors are sensing their corresponding emitter.

Description

1. Disable front panel infrared LEDs by writing 00*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is low.

2. Wait for the sensor RAM in the **Front Panel Controller** U103 <2> to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).

   - Read the sensor RAM by writing 60*hex* to the Command Register and reading eight bytes from the Data Register.
   - Clear the front panel interrupt by writing E0*hex* to the Command Register.
   - Perform a software delay.
   - Read the **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <24> is low.
   - Repeat the previous four steps until IR4 U350-22 <24> is low.

3. Enable front panel infrared LEDs by writing 01*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is high.

4. Perform a software delay.

5. Read the sensor RAM in the **Front Panel Controller** U103 <2> after it stabilizes.

   - Use the same procedures outlined in step 2.

6. Verify that all the row receptors sense an emitter (i.e., soft key rows sensor RAM bits are 0).

Error Index E2311

One or more row receptors did not sense its corresponding emitter.

There are 22 infrared emitter/receptor row pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A high (1) bit in the actual value represents the receptor that did not sense an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <1>.

Set Bit	Emitter/Receptor Pair
bit 0	DS770/CR170
bit 1	DS760/CR160
bit 2	DS761/CR161
bit 3	DS762/CR162
bit 4	DS763/CR150
bit 5	DS750/CR151
bit 6	DS751/CR152
bit 7	DS752/CR153
bit 8	DS740/CR140
bit 9	DS741/CR141
bit 10	DS742/CR142
bit 11	DS743/CR130
bit 12	DS730/CR131
bit 13	DS731/CR132
bit 14	DS732/CR133
bit 15	DS720/CR120
bit 16	DS721/CR121
bit 17	DS722/CR122
bit 18	DS723/CR110
bit 19	DS710/CR111
bit 20	DS711/CR112
bit 21	DS712/CR113

See Also The Front Panel Verify Soft Keys test.

Routine Name

Column Open

Overview

This test verifies the 11 front panel infrared emitter/receptor column pairs (x-axis input sensors) in **Touch Panel Bezel Sensors** <1> by checking that all of the column receptors are sensing their corresponding emitters.

Description

1. Disable front panel infrared LEDs by writing 00*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is low.

2. Wait for the sensor RAM in the **Front Panel Controller** U103 <2> to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).

   - Read the sensor RAM by writing 60*hex* to the Command Register and reading eight bytes from the Data Register.
   - Clear the front panel interrupt by writing E0*hex* to the Command Register.
   - Perform a software delay.
   - Read the **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <24> is low.
   - Repeat the previous four steps until IR4 U350-22 <24> is low.

3. Enable front panel infrared LEDs by writing 01*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is high.

4. Perform a software delay.

5. Read the sensor RAM in the **Front Panel Controller** U103 <2> after it stabilizes.

   - Use the same procedures outlined in step 2.

6. Verify that all column the receptors sense an emitter (i.e., soft key columns sensor RAM bits are 0).

Error Index E2321

One or more column receptors did not sense its corresponding emitter.

There are 11 infrared emitter/receptor column pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A high (1) bit in the actual value represents the receptor that did not sense an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <1>.

Set Bit	Emitter/Receptor Pair
bit 0	DS270/CR200
bit 1	DS271/CR201
bit 2	DS272/CR202
bit 3	DS370/CR300
bit 4	DS371/CR301
bit 5	DS470/CR400
bit 6	DS471/CR401
bit 7	DS570/CR500
bit 8	DS571/CR501
bit 9	DS572/CR502
bit 10	DS670/CR600

See Also

The Front Panel Verify Soft Keys test.

Routine Name

Row Close

Overview

This test verifies the 22 front panel infrared emitter/receptor row pairs (y-axis input sensors) in **Touch Panel Bezel Sensors** <1> by checking that all the receptors are capable of sensing an input (i.e., absence of the emitter beam).

Description

1. Disable front panel infrared LEDs by writing 00*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is low.

2. Wait for the sensor RAM in the **Front Panel Controller** U103 <2> to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).

   - Read the sensor RAM by writing 60*hex* to the Command Register and reading eight bytes from the Data Register.
   - Clear the front panel interrupt by writing E0*hex* to the Command Register.
   - Perform a software delay.
   - Read the **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <24> is low.
   - Repeat the previous four steps until IR4 U350-22 <24> is low.

3. Verify that none of the row receptors sense an emitter (i.e., the soft key rows sensor RAM bits are 1).

4. Enable the front panel infrared LEDs by writing 01*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is high.

5. Wait for the the front panel sensor RAM to stabilize.

   - Use the same procedures outlined in step 2.

Error Index E2331

One or more row receptors sensed an emitter.

There are 22 infrared emitter/receptor row pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A low (0) bit in the actual value represents the receptor that sensed an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <1>.

Reset Bit	Emitter/Receptor Pair
bit 0	DS770/CR170
bit 1	DS760/CR160
bit 2	DS761/CR161
bit 3	DS762/CR162
bit 4	DS763/CR150
bit 5	DS750/CR151
bit 6	DS751/CR152
bit 7	DS752/CR153
bit 8	DS740/CR140
bit 9	DS741/CR141
bit 10	DS742/CR142
bit 11	DS743/CR130
bit 12	DS730/CR131
bit 13	DS731/CR132
bit 14	DS732/CR133
bit 15	DS720/CR120
bit 16	DS721/CR121
bit 17	DS722/CR122
bit 18	DS723/CR110
bit 19	DS710/CR111
bit 20	DS711/CR112
bit 21	DS712/CR113

See Also The Front Panel Verify Soft Keys test.

Routine Name

Column Close

Overview

This test verifies the 11 front panel infrared emitter/receptor pairs (x-axis input sensors) in **Touch Panel Bezel Sensors** <1> by checking that all the receptors are capable of sensing an input (i.e., absence of an emitter beam).

Description

1. Disable front panel infrared LEDs by writing 00*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is low.

2. Wait for the sensor RAM in the **Front Panel Controller** U103 <2> to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).

   - Read the sensor RAM by writing 60*hex* to the Command Register and reading eight bytes from the Data Register.
   - Clear the front panel interrupt by writing E0*hex* to the Command Register.
   - Perform a software delay.
   - Read the **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <24> is low.
   - Repeat the previous four steps until IR4 U350-22 <24> is low.

3. Verify that none of the column receptors sense an emitter (i.e., soft key columns sensor RAM bits are 1).

4. Enable front panel infra-red LEDs by writing 01*hex* to **Front Panel Infrared Disable Control** U700B <22> so that IRDIS(L) U212C-11 is high.

5. Wait for the front panel sensor RAM to stabilize.

   - Use the same procedures outlined in step 2.

Error Index E2341

One or more column receptors sensed an emitter.

There are 11 infrared emitter/receptor column pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A low (0) bit in the actual value represents the receptor that sensed an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <1>.

Reset Bit	Emitter/Receptor Pair
bit 0	DS270/CR200
bit 1	DS271/CR201
bit 2	DS272/CR202
bit 3	DS370/CR300
bit 4	DS371/CR301
bit 5	DS470/CR400
bit 6	DS471/CR401
bit 7	DS570/CR500
bit 8	DS571/CR501
bit 9	DS572/CR502
bit 10	DS670/CR600

See Also

The Front Panel Verify Soft Keys test.

Routine Name

Upper Knob

Overview

This test verifies the upper knob circuit in the **Serial Data Interface** U330 <21> by checking that the upper knob can be initialized and exercised properly. Exercising the knob is accomplished through special test increment/decrement operations in U330.

Description

1. Initialize knob circuits in the **Serial Data Interface** U330 <21> by writing 80*hex*, 02*hex*, and 00*hex* to Misc. Control Select 1 Register, Misc. Control Select Register and Misc. Control Select Register, respectively.

2. Perform a software delay.

3. Read the Knob Read Enable 0 Register and verify that the value is 3F*hex*.

4. Index the upper knob counter to read 00*hex*.

   - Read the Knob Read Enable 0 Register and depending on its value, manually index the counter to read 00*hex*. To increment the counter, patterns 80*hex*, A0*hex*, E0*hex*, C0*hex*, and 80*hex* are written to the Misc. Control Select 1 Register. To decrement the counter patterns 80*hex*, C0*hex*, E0*hex*, A0*hex*, and 80*hex* are written to the Misc. Control Select 1 Register.

5. Exercise the counter through its full range in both directions and verify that the counter still reads 00*hex*.

   - Decrement the counter 64 times (see step 5). Adjust the counter to 0, and increment the counter 64 times. Read the Misc. Control Select 1 Register and verify that it is 00*hex*.

Error Index E2411

The upper knob counter does not read 3F*hex* after initialization.

Error Index E2412

The upper knob counter increment/decrement test failed.

See Also

The Front Panel Verify Knobs test.

Routine Name

Lower Knob

Overview

This test verifies the lower knob circuit in the Serial Data Interface U330 <21> by checking that the lower knob can be initialized and exercised properly. Exercising the knob is accomplished through special test increment/decrement operations available in U330.

Description

1. Initialize knob circuits in the Serial Data Interface U330 <21> by writing 80*hex*, 02*hex*, and 00*hex* to Misc. Control Select 1 Register, Misc. Control Select Register and Misc. Control Select Register, respectively.

2. Perform a software delay.

3. Read the Knob Read Enable 1 Register and verify that the value is 3F*hex*.

4. Index the lower knob counter to read 00*hex*.

   - Read the Knob Read Enable 1 Register and depending on its value, manually index the counter to read 00*hex*. To increment the counter, patterns 80*hex*, A0*hex*, E0*hex*, C0*hex*, and 80*hex* are written to the Misc. Control Select 1 Register. To decrement the counter patterns 80*hex*, C0*hex*, E0*hex*, A0*hex*, and 80*hex* are written to the Misc. Control Select 1 Register.

5. Exercise the counter through its full range in both directions and verify that the counter still reads 00*hex*.

   - Decrement the counter 64 times (see step 5). Adjust the counter to 0, and increment the counter 64 times. Read the Knob Read Enable 1 Register and verify that it is 00*hex*.

Error Index E2421

The lower knob counter does not read 3F*hex* after initialization.

Error Index E2422

The lower knob counter increment/decrement test failed.

See Also

The Front Panel Verify Knobs test.

Routine Name Hard Keys

Overview This test verifies the hard keys by allowing the user to interactively press the hard keys. Both visual and audio feedback is provided.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off." Once this test is invoked, the operator may press any of the hard keys in the instrument and verify that the corresponding image of the key on the screen is highlighted, the associated LED is turned on, and an audio click is generated.

Description

1. Store the contents of the front panel display RAM in the **Front Panel Controller** U103 <2>.
   - Write 70*hex* to the Command Register and read eight bytes from the Data Register.
2. Turn all the hard key LEDs off by writing 00*hex* to the eight bytes in the display RAM.
3. Initialize the **Front Panel Controller** U103 <2>.
   - Write 04*hex* and 22*hex* to Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <2> is low.
4. Enable the front panel interrupt and the MMU interrupts. Display the image of the hard keys.
5. Wait for a hard key input.
6. When a valid hard key input is obtained, generate visual and audio feedback.
   - Highlight the image of that hard key on the screen.
   - Turn the LED associated with that hard key, if any, on by changing the contents of the display RAM in the **Front Panel Controller**.
   - Generate a click through the **Tone Generator** <22>.
7. Repeat steps 5 and 6 until an "Exit" input is received.
8. Initialize **Front Panel Controller** U103 <2>.
   - Write 04*hex* and 22*hex* to Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <2> is low.
9. Restore the front panel display RAM by writing back the saved contents of the display RAM.
   - Write 90*hex* to the Command Register and write eight bytes to the Data Register.

Error Index None.

Routine Name

Soft Keys

Overview

This test verifies the soft keys by allowing the user to interactively touch any of the soft keys. Both visual and audio feedback is provided. It is also useful in the calibration and verification of Crt Driver board adjustments.

Operator Procedure

This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off." Once this test is invoked, the operator may touch any of the soft keys in the instrument and verify that a touch box is drawn around the soft key on the screen and an audio click is generated.

Description

1. Store the contents of the front panel display RAM in the **Front Panel Controller** U103 <2>.
   - Write 70*hex* to the Command Register and read eight bytes from the Data Register.
2. Initialize **Front Panel Controller** U103 <2>.
   - Write 04*hex* and 22*hex* to the Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <2> is low.
3. Enable the front panel interrupt and the MMU interrupts. Display a grid pattern (in dim color) along the soft key boundaries on the screen to identify them.
4. Wait for a touch input.
5. When a valid touch input is obtained, draw a box of bright lines around the selected soft key and generate a click.
6. Repeat steps 4 and 5 until an "Exit" input is received.
7. Initialize Front Panel Controller U103 <2>.
   - Write 04*hex* and 22*hex* to Command Register and then write E0*hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <2> is low.

Error Index

None.

See Also

The adjustment procedure for the **CRT Driver** in the Service Reference Manual.

Routine Name Knobs

Overview This test verifies the knobs by allowing the user to interactively turn either of the knobs. Visual feedback is provided for both knob movements.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off." Once this test is invoked, the operator may turn either of the knobs on the instrument and verify that the corresponding knob pointer on the screen rotates and its associated counter value changes.

Description

1. Initialize the knob circuits in the **Serial Data Interface** U330 <21> by writing 80*hex*, 02*hex*, and 00*hex* to Misc. Control Select 1 Register, Misc. Control Select Register, and Misc. Control Select Register, respectively.

2. Enable the front panel interrupt and the MMU interrupts. Draw an image of the knobs on the screen. The display at the center of each knob image is the corresponding current knob counter value.

3. Wait for either of the knobs to be turned by reading Knob Read Enable 0 and Knob Read Enable 1 Registers in the **Serial Data Interface** <21> repeatedly and checking for any change in their values.

4. If either of the knobs is turned (i.e., the register values were different from the previous values), update the corresponding knob pointer on the screen image and update the counter value for the counter.

5. Repeat steps 3 and 4 until an "Exit" input is received.

Error Index None.

Routine Name Comparator

Overview This test verifies the sensitivity of the **Temperature Sensor** <22> by checking that its comparator U122 trips only once over the full temperature range.

Description

1. Exercise the comparator through the entire temperature range (0-100° C) and verify that it trips only once.

   - Write 00*hex* to the **DAC Data Latch** U610 <22> and check if SET HI(H) U120C-8 is high by reading the **Temp/Tone/Diagn Readback Buffer** U612 <22>. If SET HI(H) is high, then the temperature corresponding to 00*hex* (0° C) is the temperature read by the sensor.

   - Increment the previous value by 06*hex* (3° C increment), write it to the **DAC Data Latch** U610 <22>, and check if SET HI(H) U120C-8 is high. Repeat the sequence of increment/write/verify until FF*hex* (100° C) is reached. If the comparator trips only once, then SET HI(H) should be high only once for the entire temperature range.

Error Index E3111 The comparator had a sensitivity problem.

See Also The Ramp Tone manual test routine in the Tone Gen area.

Routine Name Counting

Overview This test verifies that the **Real-Time Clock** U614 <21> is running by watching the change in its internal device counters. This test does not alter or stop the clock to perform this verification.

Description

1. Read the Hundredth Counter Register and Seconds Counter Register from the **Real-Time Clock** U614 <21>.
2. Perform a software delay.
3. Read the Hundredth Counter Register and Seconds Counter Register again and verify that they are different from the previous readings.

Error Index E3211 The **Real-Time Clock** <21> is not running.

Routine Name

Interrupt

Overview

This test verifies the **Real-Time Clock** <21> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Mask all the **Real-Time Clock's** <21> interrupts by writing 00*hex* to its Mask Register.
2. Read the Interrupt Status Register of the **Real-Time Clock** <21> to clear all the interrupts.
3. Verify that there is no **Real-Time Clock** interrupt at the **Interrupt Controllers** <24>.
   - Read the **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR6 U370-24 is low.
4. If there is no pending **Real-Time Clock** interrupt at the **Interrupt Controllers** (i.e., IR6 U370-24 is low), enable the interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 6 and **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 low.
5. Enable the **Real-Time Clock's** one hundredth interrupt at the **Real-Time Clock** <21> by writing 18*hex* and 02*hex* to the Command Register and Mask Register.
6. Perform a software delay (at least 10 ms).
7. Disable the **Real-Time Clock's** one-hundredth interrupt at the **Real-Time Clock** <21> by writing 08*hex* to the Command Register.
8. Verify that the **Real-Time Clock's** interrupt did occur.
9. Disable **Real-Time Clock's** interrupt at the **Interrupt Controllers** by setting **Interrupt Controllers** SLAVE 3 U370 <24> bit 6 and **Interrupt Controllers** MASTER U350 <24> bit 3 high.

Error Index E3221

The **Real-Time Clock** interrupt did not occur.

Error Index E3222

The **Real Time Clock** interrupt could not be reset at the **Interrupt Controllers** Slave U370 <24>.

Routine Name

Calibrate

Overview

This test helps the operator to adjust the frequency of the **Real-Time Clock** oscillator <21> to the required accuracy by generating interrupts every second.

Operator Procedure

This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector set to "Off." After invoking this test, the operator may adjust the oscillator frequency by adjusting C510 <21>. Using a timer-counter (with an accuracy of at least ±1PPM) in period mode with its probe connected to TP310 <21>, C510 <21> is adjusted so that the period between two consecutive falling edges is precisely 1 s ±5PPM (±5 µs).

Description

1. Enable the front panel, and MMU interrupts, and display the "Exit" selector on the screen.
2. Enable the **Real-Time Clock's** <21> interrupt at the Interrupt Controllers <24> by setting **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 6 and **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 low.
3. Enable **Real-Time Clock's** <21> interrupt at the **Real-Time Clock** <21> by writing 18*hex* to the Command Register.
4. Until the "Exit" selection is made, repeatedly enable the **Real-Time Clock's** <21> one second interrupt by writing 08*hex* to the Mask Register.
5. Mask the **Real Time Clock's** interrupt at the **Real-Time Clock** by writing 00*hex* and 08*hex* to the Mask Register and Command Register.
6. Disable **Real-Time Clock** interrupt at the **Interrupt Controllers** by setting Interrupt Controllers SLAVE 3 U370 <24> Mask Register bit 6 and **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 high.

Error Index

None.

See Also

The adjustment procedure for the **Real-Time Clock** oscillator in the Service Reference Manual.

Routine Name

Ramp Tone

Overview

This test helps verify the **Temp/Tone DAC** U212 <22> by generating a ramp signal on the DAC output.

Operator Procedure

This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." After invoking this test in "cycle" mode (i.e., "LOOP" and "TERSE" selected), the operator is required to use external equipment to verify that the **Temp/Tone DAC** IOUT(L) U212-2 <22> is a ramp waveform with 1mA amplitude or the voltage across **Tone Generator** R214 <22> is a ramp waveform with 8.8V ±10% amplitude.

Description

1. Generate a ramp on the **Temp/Tone DAC** tone output.
   - Write 100*hex* (value corresponding to the highest tone) to DAC Data Latch U610 <22>.
   - Increment the value and write it to the **DAC Data Latch** U610 <22>. Repeat the increment/write sequence until 1FF*hex* is reached.
2. Turn tone generation off by writing 00*hex* to **DAC Data Latch** U610 <22>.

Error Index

None.

Routine Name Loopback

Overview This test verifies the **Printer Controller** U430 <4>, **Printer Data Buffer** U520 <4>, **Printer Loop Back Buffer** U540 <4>, and Printer Interface Control <4> by looping back a set of patterns.

Description

1. Program the **Printer Controller** <4> in test mode (loop back mode) by writing A6*hex* and 08*hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is low. This enables **Printer Loop Back Buffer** U540 <4> and disables **Printer Interface Control** buffer U541 <4>.

2. Transmit, receive, and verify a set of patterns.

   - Write patterns AA*hex*, CC*hex*, F0*hex*, 0F*hex*, and 55*hex* to U430 Port A.Verify the patterns by reading U430 Port B.

3. Program the **Printer Controller** <4> in normal mode by writing 09*hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is high.

Error Index E4111 The received pattern was not the same as the transmitted one. The display shows the first pattern that did not verify.

Routine Name

Interrupt

Overview

This test verifies the **Printer Controller** U430 <4> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The interrupt is generated by looping back a byte of data.

Description

1. Program the **Printer Controller** <4> in test mode (loop back mode) by writing A6*hex* and 08*hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is low. This enables **Printer Loop Back Buffer** U540 <4> and disables **Printer Interface Control** buffer U541 <4>.

2. Verify that there is no printer interrupt pending at the **Interrupt Controllers** <24>.

   - Read the **Interrupt Controllers** MASTER U350 <24> INTERRUPT REQUEST REGISTER and verify that IR6 U350-24 is low.

3. If there are no pending interrupts, enable the interrupt at the Interrupt Controllers <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 6 low.

4. Enable the interrupt at the **Printer Controller** by writing 0D*hex* to the Control Word Register.

5. Loopback a pattern.

   - Write AA*hex* to U430 Port A, perform a software delay, and read it from U430 Port B. Immediately following the read, a transmitter empty interrupt is generated, i.e., INTR(A)(H) U430-17 is high.

6. Perform a software delay.

7. Disable the interrupt at the **Printer Controller** <4> by writing 0C*hex* to the Control Word Register.

8. Disable the interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 6 high.

9. Verify that a printer interrupt did occur.

10. Program the **Printer Controller** <4> in normal mode by writing 09*hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is high.

Error Index E4121

The **Printer Controller** <4> interrupt did not occur.

Error Index E4122

The **Printer Controller** <4> interrupt could not be cleared at the **Interrupt Controllers** MASTER U350 <24>.

Routine Name Pattern

Overview This test helps the operator verify the **Printer Controller's** U430 <4> printer interface by printing a set of patterns (all printable ASCII characters).

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Before executing this test, the operator should connect a Centronics-compatible printer to the rear panel of the instrument.

Description

1. Program the **Printer Controller** U430 <4> in normal mode with its interrupt enabled by writing A6*hex*, 09*hex* and 0D*hex* to the Control Word Register.

2. Check printer availability by repeatedly reading the printer status until the printer is available or the software timeout period has expired.

   - Write 0B*hex* and 0A*hex* to the Control Word Register.
   - Read U430 Port B and verify that **Printer Interface Control** PB4(L) U541-9 <4> and PB5(H) U541-7 are high.
   - Read U430 Port C and verify that bit 3 is high.

3. Initialize the printer by writing 1B*hex*, 40*hex*, 1B*hex*, and 50*hex* to U430 Port A regardless of the printer availability.

4. Print the first line of the first character set.

   - Read the printer status using the procedure outlined in step 2. Irrespective of the printer availability, write the first pattern 20*hex* (space character) to U430 Port A. Increment the pattern and write it to U430 Port A after checking for the printer availability. Repeat the sequence of increment/read printer status/write pattern to U430 Port A until 32 patterns have been output.
   - Write 0A*hex* (line feed) and 0D*hex* (carriage return) to U430 Port A.

5. Continue printing lines of the first character set using the procedure outlined in step 4 until the last pattern 7F*hex* (tilda character) is printed.

6. Check printer availability using procedures outlined in step 2. Irrespective of the printer availability, start a new line for second character set by writing 0A*hex* (line feed) and 0D*hex* (carriage return) to U430 Port A.

7. Print the second character set.

   - Using procedures outlined in step 4 and 5 print patterns starting from A0*hex* to FF*hex*.

8. Check printer availability using procedures outlined in step 2. Irrespective of the printer availability, start a new line by writing 0A*hex* (line feed) and 0D*hex* (carriage return) to U430 Port A.

Error Index None.

Caveats This test will take longer to complete if a printer is not connected to the instrument.

Routine Name Loopback

Overview This test verifies the **Std RS-232 Controller** U311 <3> by internally looping back a set of patterns.

Description

1. Program **Std RS-232 Controller** U311 <3> in local loopback mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 35*hex*, 25*hex*, 1A*hex*, 13*hex*, 87*hex*, BB*hex*, and E5*hex* to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
3. Irrespective of the transmitter availability, transmit pattern AA*hex* by writing it to THRA.
4. Wait for the receiver to be ready by repeatedly reading SRA until bit 0 is high or until a software timeout period has expired.
5. Irrespective of the receiver availability, receive the pattern by reading RHRA and verify it.
6. Repeat steps 2-5 for patterns CC*hex*, F0*hex*, 0F*hex*, and 55*hex*.
7. Program the **Std RS-232 Controller** U311 <3> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4211 The looped back pattern did not match the pattern sent. The display shows the first pattern that failed.

Routine Name

Baud Rate

Overview

This test verifies the baud rate generator of **Std RS-232 Controller** U311 <3> and **RS-232 Clock Generator** <3> by using the Timer U822 COUNTER 2 <21> to measure the time taken to transmit and receive a character.

Description

1. Program **Std RS-232 Controller** U311 <3> in local loopback mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 35*hex*, 25*hex*, 1A*hex*, 13*hex*, 87*hex*, BB*hex*, and E5*hex* to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Enable the RS232 interrupt at the Interrupt Controllers <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 4 low.
3. Enable the receiver ready interrupt at the Std RS232 Controller U311 <3> by writing 02*hex* to IMR.
4. Set the baud rate of the **Std RS-232 Controller** <3> to 1200.
   - Write 0A*hex* , 66*hex*, and 6F*hex* to CRA, CSRA, and ACR.
5. Program the **Timer** COUNTER 2 U822 <21> to count 6 MHz clock.
   - Write 00*hex*, B0*hex*, FF*hex*, and FF*hex* to Timer Configuration Logic U720, **Timer** Control Word Register, Count Register low and high bytes, respectively.
6. Transmit a character by writing the pattern 55*hex* to THRA of U311.
7. Perform a software delay.
8. Receive the character by reading RHRA of U311.
9. Verify the **Timer** COUNTER 2 U822 <21> Count Register against the known value (the **Timer** COUNTER 2 was stopped through the RS-232 interrupt handler).
10. Repeat steps 4-9 for baud rates 9600 and 19200.
11. Disable the RS232 interrupt at the **Interrupt Controllers** U311 <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 4 high.
12. Disable all the RS-232 interrupts at the **Std RS-232 Controller** <3> by writing 00*hex* to IMR.

13. Reset the RS-232 receiver by writing 25*hex* to CRA.

14. Program the **Std RS-232 Controller** U311 <3> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.

    - Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
    - Read IPCR, write 00*hex* to IMR, and read ISR.
    - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4221

The time taken to transmit and receive a character for a baud rate was not within the expected tolerance.

Caveats

This test may fail if the Main Processor Board **Clock Generator** <23> reference Y950 is out of tolerance since it is being used as the timing reference for the **Timer** <21> COUNTER 2.

Routine Name Error Gen (Error Generation)

Overview This test verifies the error generation capability of the **Std RS-232 Controller** U311 <3> by generating an overrun error.

Description

1. Program **Std RS-232 Controller** U311 <3> in local loop back mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 35*hex*, 25*hex*, 1A*hex*, 13*hex*, 87*hex*, BB*hex*, and E5*hex* to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
3. Irrespective of the transmitter availability, transmit pattern AA*hex* by writing it to THRA.
4. Repeat steps 2 and 3 for patterns CC*hex*, F0*hex*, 0F*hex*, 55*hex*, and 56*hex*.
5. Read SRA and verify that the overrun bit is set.
6. Reset the RS232 receiver and the error status.
   - Read RHRA twice to remove the extra characters sent.
   - Write 25*hex* and 45*hex* to CRA.
7. Program the **Std RS-232 Controller** U311 <3> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4231 The **Std RS-232 Controller** <3> did not record the overrun error.

Routine Name

Interrupt

Overview

This test verifies the **Std RS-232 Controller's** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program **Std RS-232 Controller** U311 <3> in local loop back mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 35*hex*, 25*hex*, 1A*hex*, 13*hex*, 87*hex*, BB*hex*, and E5*hex* to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Verify that there is no pending RS232 interrupt at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 is low.
3. If there is no pending RS232 interrupt, enable the RS-232 interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 Mask Register bit 4 low.
4. Enable the receiver ready interrupt at the **Std RS-232 Controller** U311 <3> by writing 02*hex* to IMR.
5. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
6. Irrespective of the transmitter availability, transmit a character by writing 55*hex* to THRA.
7. Perform a software delay.
8. Read the character by reading RHRA.
9. Verify that the interrupt did occur.
10. Reset the receiver and disable the receiver interrupt by writing 25*hex* and 00*hex* to CRA and IMR, respectively.
11. Disable the RS232 interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 4 high.
12. Program the **Std RS-232 Controller** U311 <3> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.

- Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
- Read IPCR, write 00*hex* to IMR, and read ISR.
- Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4241 The Std RS-232 Controller <3> interrupt did not occur.

Error Index E4242 The Std RS-232 Controller <3> interrupt could not be reset at the Interrupt Controllers SLAVE U370 <24>.

Routine Name Extern Loop (External Loopback)

Overview This test verifies the **Std RS-232 Controller** U311 <3>, **RS-232 Clock Generator** <3> and **Std RS-232 Interface Drivers** <3> by externally looping back a set of patterns.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Before running this test, the operator should connect an external loopback connector (Tek part no. 013-0198-00) to the RS232 connector on the rear panel of the instrument.

Description

1. Program the **Std RS232 controller** U311 <3> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
3. Irrespective of the transmitter availability, transmit pattern AA*hex* by writing it to THRA.
4. Wait for the receiver to be ready by repeatedly reading SRA until bit 0 is high or until a software timeout period has expired.
5. Irrespective of the receiver availability, receive the pattern by reading RHRA and verify it.
6. Repeat steps 2-5 for patterns CC*hex*, F0*hex*, 0F*hex*, and 55*hex*.
7. Program the **Std RS-232 Controller** U311 <3> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
   - Write 1A*hex*, 13*hex*, 07*hex*, BB*hex*, and E5*hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
   - Read IPCR, write 00*hex* to IMR, and read ISR.
   - Write 01*hex*, 04*hex*, 00*hex*, 06*hex*, 30*hex*, 20*hex*, and 05*hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4251 The pattern looped back did not match the transmitted one. The display shows the first pattern that did not verify.

Caveats The test will fail if it is run without the external connector.

Routine Name Intrpt Reset (Interrupt Reset)

Overview This test verifies that the GPIB interrupt can be cleared by resetting the GPIB Controller U410 <3>.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.

Description

1. Reset GPIB Controller U410 <3> and verify its reset status.
   - Write 80*hex*, 00*hex*, 00*hex*, and 00*hex* to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.
   - Read Interrupt Status Register 0 and Interrupt Status Register 1 and verify that they both read 00*hex*.
2. Verify that there is no GPIB interrupt pending at the Interrupt Controllers <24>.
   - Read Interrupt Controllers SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR2 U370-20 is low.

Error Index E4311 The GPIB Controller <3> reset did not set the Interrupt Status Register 0 to 00*hex*.

Error Index E4312 The GPIB Controller <3> reset did not set the Interrupt Status Register 1 to 00*hex*.

Error Index E4313 The GPIB Controller <3> interrupt could not be reset at the Interrupt Controllers SLAVE 3 U370 <24>.

Caveats This test will fail if it is run with the instrument connected to a GPIB bus.

Routine Name Reset Status

Overview This test verifies that the **GPIB Controller** U410 <3> status after a reset is correct by reading its Address Status Register and Bus Status Register.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.

Description

1. Reset **GPIB Controller** U410 <3> and verify that the Address Status Register and Bus Status Register reflect an idle state.

   - Write 80*hex*, 00*hex*, 00*hex*, and 00*hex* to the Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.

   - Read the Address Status Register 0 and the Bus Status Register 1 and verify that they both read 00*hex*.

Error Index E4321 The **GPIB Controller's** U410 <3> Address Status Register was not set to 00*hex* after reset.

Error Index E4322 The **GPIB Controller's** U410 <3> Bus Status Register was not set to 00*hex* after reset.

Caveats This test will fail if it is run with the instrument connected to a GPIB bus.

Routine Name Data Lines

Overview This test verifies the data bus independence of the **GPIB Controller** U410 <3> by doing a "walking one's" test on the Data Out Register.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.

Description

1. Reset **GPIB Controller** U410 <3>.
   - Write 80*hex*, 00*hex*, 00*hex*, and 00*hex* to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.
2. Put the **GPIB Controller** U410 <3> into a "talk only" mode by writing 8A*hex* to Auxiliary Command Register.
3. Verify the data line independence of the **GPIB Controller** U410 <3>.
   - Write patterns 01*hex*, 02*hex*, 04*hex*, 08*hex*, 10*hex*, 20*hex*, 40*hex*, and 80*hex* to Data Out Register, read it back from Command Pass Through Register, and verify them.
4. Unset the **GPIB Controller's** U410 <3> "talk only" mode by writing 0A*hex* to Auxiliary Command Register.

Error Index E4331 The read pattern did not match the written pattern. The display shows the first pattern that did not verify.

Caveats This test will fail if it is run with the instrument connected to a GPIB bus.

Routine Name

Interrupt

Overview

This test verifies the **GPIB Controller's** U410 <3> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. A "buffer empty" interrupt is used for verification.

Operator Procedure

This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.

Description

1. Reset **GPIB Controller** U410 <3>.
   - Write 80*hex*, 00*hex*, 00*hex*, and 00*hex* to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.
2. Verify that there is no GPIB interrupt pending at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 3 U370 <24> INTERRUPT REQUEST REGISTER and verify that IR2 U370-20 is low.
3. If there is no pending GPIB interrupt, enable GPIB interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 2 low.
4. Enable the "byte out" interrupt at the GPIB Controller by writing 10*hex* to the Interrupt Mask Register 0.
5. Put the **GPIB Controller** in "talk only" mode by writing 8A*hex* to the Auxiliary Command Register.
6. Perform a software delay.
7. Disable the "byte out" interrupt at the **GPIB Controller** by writing 00*hex* to the Interrupt Mask Register 0.
8. Unset the **GPIB Controller's** "talk only" mode by writing 0A*hex* to Auxiliary Command Register.
9. Disable the interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <24> Mask Register bit 2 high.
10. Verify that the GPIB interrupt did occur.

Error Index E4341

The **GPIB Controller** <3> interrupt did not occur.

Error Index E4342

The **GPIB Controller** <3> interrupt could not be reset at the **Interrupt Controllers** SLAVE 3 U370 <24>.

Caveats

This test will fail if it is run with the instrument connected to a GPIB bus.

Routine Name Status Reg (Status Register)

Overview This test verifies **MMU Gate Array's** U210 <27> Status and Mode Register (SMR) by performing a pattern test on the register.

Description

1. Reset the **MMU Gate Array** U210 <27> to diagnostic mode by writing 1A*hex*, 007F*hex*, and 0*hex* to **Diagnostics** U524 <27>, SMR, and **Diagnostics** U530 <27>, respectively.

2. Perform a pattern test on SMR on bits that are both readable and writable.

   - Write, read, and verify patterns 0001*hex*, 0002*hex*, 0020*hex*, 0040*hex*, 0080*hex* and 0000*hex*.

3. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index E5111 The pattern read did not match the pattern written. The display shows the first pattern that failed.

Routine Name

Arbitration

Overview

This test verifies the arbitration capability of the MMU Gate Array U210 <27> (hereafter referred to as MMU). With the MMU in single-step mode, each arbitration combination is carefully set up and the MMU is single-stepped to the completion of that arbitration cycle. The outcome of the arbitration is verified against known results.

Description

1. Reset the MMU to diagnostic mode by writing 1A*hex*, 007F*hex*, and 0*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>.

2. Find a "refresh" arbitration cycle.

   - Pulse Clock Generator CLK(H) U110D-11 <27> (by toggling **Diagnostics** U530-7) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.

3. Irrespective of finding a "refresh" cycle, either remove the "refresh" cycle or find the next "refresh" cycle.

   - If the current arbitration group does not have "refresh" cycle in it, then pulse CLK(H) U110D-11 20 times to complete the "refresh" cycle. Else, pulse the clock 252 times to find the next "refresh" cycle.

4. Generate "read digitizer" and "read display" arbitration requests at their sync positions.

   "Read Digitizer"

   - Write 0000*hex* and 00CC*hex* to the Random Address Generator (RAG) and RAG Register 0. Set bit 7 of SMR to set the **MMU Gate Array** to diagnostic mode so that the tag bits are assured to be 0000*hex*.

   - Start the "read digitizer" arbitration by setting DIGREQ(L) U524-12 low and then high.

   - Pulse CLK(H) U110D-11 <27> appropriate number of times (six times for this combination) so that the all the other arbitration requests of this arbitration combination will coincide at the sync position.

   "Read Display"

   - Write 0000*hex*, 00A9*hex* and 00AA*hex* to SAG Message Pointer Register (MPR), Message Length Register (MLR) and Address Counter (AC) to set up the SAG to "read display." Set bit 10 of SMR to start the "read display" channel.

   - Start the "read display" arbitration by setting DATARDY(H) U524-6 high and then low.

- Pulse CLK(H) U110D-11 <27> appropriate number of times (1) so that all the other arbitration requests of this arbitration combination will coincide at their sync positions.

5. Pulse CLK(H) U110D-11 <27> appropriate number of times to read the MMU's output at **Diagnostics** U112 <27>. The low and high bytes are read separately from **Diagnostics** U112 <27>. For each arbitration request, the output is verified against a known value.

6. Complete MMU arbitration requests by pulsing CLK(H) U110D-11 <27> to the corresponding completion of each request.

7. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

8. Repeat steps 2-7 for the following arbitration combinations:

   case 2: "write display" and "read digitizer"
   case 3: "refresh" and "read display"
   case 4: "refresh" and "read digitizer"
   case 5: "refresh", "read digitizer", and "read display"
   case 6: "refresh" and "write display"
   case 7: "refresh", "write display", and "read digitizer"
   case 8: "refresh" and "write digitizer"

   The other arbitration requests are set up in the following ways:

   "Refresh"

   Since the "refresh" cycle occurs automatically every 256 clock cycles, it is only necessary to find the start of a "refresh" cycle and set up the other arbitration requests to coincide with the next "refresh" cycle's sync point.

   "Write Display"

   - Write 0000*hex*, 00A9*hex* and 00AA*hex* to SAG MPR, MLR and AC to set up the SAG to "write display." Set bit 9 of SMR to start the "write display" channel.

   - Start "write display" arbitration by setting SENDNEW(L) U524-5 low and then high.

   - Pulse CLK(H) U110D-11 <27> appropriate number of times so that all the other arbitration requests of this arbitration combination will coincide at the sync position.

   "Write Digitizer"

   - Write 0000*hex*, 00A9*hex* and 00AA*hex* to SAG MPR, MLR and AC to set up the SAG to "write digitizer". Set bit 11 of SMR to start "write digitizer" channel.

   - Start the "write digitizer" arbitration by setting DIGACK(L) U524-9 low and then high.

- Pulse CLK(H) U110D-11 <27> appropriate number of times so that all the other arbitration requests of this arbitration combination will coincide at the sync position.

Error Index E5121 Combination "read digitizer" and "read display" failed.

Error Index E5122 Combination "write display" and "read digitizer" failed.

Error Index E5123 Combination "refresh" and "read display" failed.

Error Index E5124 Combination "refresh" and "read digitizer" failed.

Error Index E5125 Combination "refresh", "read digitizer", and "read display" failed.

Error Index E5126 Combination "refresh" and "write display" failed.

Error Index E5127 Combination "refresh", "write display", and "read digitizer" failed.

Error Index E5128 Combination "refresh" and "write digitizer" failed.

Routine Name Refresh

Overview This test verifies the MMU Gate Array U210 <27> Refresh Counter by performing a "walking one's" test on this register.

Description

1. Reset the MMU U210 <27> to diagnostic mode by writing 1A*hex*, 007F*hex*, and 0*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>.

2. Find a "refresh" arbitration cycle.

   - Pulse CLK(H) U110D-11 <27> (by toggling **Diagnostics** U530-7) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU.

3. Process the "refresh" cycle by pulsing the clock 20 times.

4. Make the Refresh Counter readable at the **Diagnostics** U112 <27>. The Refresh Counter is readable only when the refresh cycle is at its sync point of arbitration.

   - Pulse CLK(H) U110D-11 <27> until the next refresh cycle (255 clocks from the previous "refresh" cycle) is found. A maximum of five attempts will be made to find two consecutive "refresh" cycles that are 255 clocks apart.

5. Irrespective of finding two consecutive "refresh" cycles, perform a "walking one's" pattern test on the Refresh Counter.

   - Write patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, and 0080*hex* to the Refresh Counter. Read and verify the patterns from **Diagnostics** U112 <27>.

6. Reset the MMU Gate Array U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index E5131 The Refresh Counter of U210 could not be accessed.

Error Index E5132 The pattern read from the Refresh Counter did not match the pattern written. The display shows the first pattern that failed.

Routine Name Dtalk Intrpt (Display Talk Request Interrupt)

Overview This test verifies the **MMU Gate Array** U210 <27> display talk request interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the **MMU Gate Array** U210 <27> to an isolated diagnostic mode by writing 1A*hex*, 007F*hex*, and 4*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.
2. Verify that there is no pending display talk request interrupt at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR2 U360-20 is low.
3. If there is no display interrupt pending, then enable display talk request interrupt by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 2 low.
4. Generate a display talk request interrupt by setting DATARDY(H) U524-6 <27> high and then low.
5. Perform a software delay and then verify that the display talk request interrupt did occur.
6. Clear the interrupt at the MMU by setting SMR bit 1 high.
7. Disable the display talk request interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 2 high.
8. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index E5141 The display talk request interrupt did not occur.

Error Index E5142 The display talk request interrupt could not be cleared at the **Interrupt Controllers** SLAVE 1 U360 <24>.

Routine Name SAG Compare

Overview This test verifies the **MMU Gate Array's** U210 <27> Sequential Address Generator (SAG) comparator by comparing a fixed value in the Address Counter (AC) with different values in the Message Length Register (MLR).

Description

1. Reset the **MMU Gate Array** U210 <27> to diagnostic mode by writing 1A*hex*, 007F*hex*, and 0*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.
2. Program the SAG to write a word to the display.
   - Write 0000*hex*, 7FFF*hex* (since the SAG pre-increments the AC, this value becomes 8000*hex* for each comparison), and 0001*hex* to Message Pointer Register (MPR), AC, and MLR, respectively.
3. Start a "write display" cycle by setting bit 9 of SMR high and setting SENDNEW(L) U524-5 <27> low and then high.
4. Pulse **Clock Generator** CLK(H) U110D-11 (by toggling **Diagnostics** U530-7) until MDATALATCH(L) U400-5 is low.
5. Pulse CLK(H) U110D-11 6 times to complete "write display" cycle.
6. Read and verify the output of the comparator. The output of the comparator is reflected in bit 3 of SMR. If the AC and MLR are equal, bit 3 will be high. The comparator output should be high only when the value in the MLR is 8000*hex*.
7. Clear SAG interrupt, if any, by setting bit 3 of SMR high.
8. Repeat steps 2-7 for values 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex* in MLR.
9. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index E5151 The SAG comparator output was incorrect. The display shows the first MLR register value for which the comparator output failed.

Routine Name

SAG Adder (Sequential Address Generator Adder)

Overview

This test verifies the SAG adder in the **MMU Gate Array** U210 <27> by adding the Address Counter (AC) and the Message Pointer Register (MPR).

Description

1. Reset the **MMU Gate Array** U210 <27> to diagnostic mode by writing 1A*hex*, 007F*hex*, and 0*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.

2. Find a "refresh" arbitration cycle.

   - Pulse **Clock Generator** CLK(H) U110D-11 <27> (by toggling **Diagnostics** U530-7) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.

3. Process the "refresh" cycle by pulsing CLK(H) U110D-11 20 times.

4. Program the SAG to read from display by writing 0002*hex*, 0001*hex* and 0001*hex* to MLR, AC and MPR.

5. Start a "read display" cycle by setting bit 10 in SMR and setting DATARDY(H) U524-6 <27> high and then low.

6. Find the "read display" cycle.

   - Pulse CLK(H) U110D-11 <27> until a SAG RAS is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.

7. Read the low byte of the adder output from **Diagnostics** U112 <27>.

8. Pulse CLK(H) U110D-11 two times.

9. Read the upper byte of the adder output from **Diagnostics** U112 <27>. Swap bits 9 and 15 of the upper byte.

10. Verify the adder output against the known output.

11. Complete the "read display" cycle by pulsing CLK(H) U110D-11 four times and clear the SAG interrupt, if any, by setting bit 3 of SMR.

12. Repeat steps 4 -11 for 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex* in AC and MPR.

13. Program the SAG to read from display by writing 0002*hex*, 0010*hex* and 3FFF*hex* to the MLR, AC and MPR.

14. Complete the "read display" cycle by following procedures outlined in steps 5-11 and verify that the adder output rippled to value 0000*hex*.

15. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index E5161 The MMU SAG adder output did not match the known value. The display shows the first output that failed.

Error Index E5162 The MMU SAG adder failed the ripple test.

Routine Name

SAG Intrpt (Sequential Address Generator Interrupt)

Overview

This test verifies the **MMU Gate Array** U210 <27> SAG Interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the **MMU Gate Array** U210 <27> to an isolated diagnostic mode by writing 1A*hex*, 007F*hex*, and 4*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.

2. Verify that there is no pending SAG interrupt at the Interrupt Controllers <24>.

   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR0 U360-18 is low.

3. If there is no SAG interrupt pending, enable the SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

4. Start a "write display" cycle by setting bit 9 of SMR high and setting SENDNEW(L) U524-5 <27> low and then high.

5. Perform a software delay and then verify that the SAG interrupt did occur.

6. Clear SAG interrupt at the MMU by setting bit 3 of SMR high.

7. Disable the SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 high.

8. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index E5171

The SAG interrupt did not occur.

Error Index E5172

The SAG interrupt could not be cleared at the **Interrupt Controllers** SLAVE 1 U360 <24>.

Routine Name RAG Regs (Random Address Generator Registers)

Overview This test verifies all the **MMU Gate Array** U210 <27> RAG registers by performing a "walking one's" test on each 9-bit RAG register.

Description

1. Reset the **MMU Gate Array** U210 <27> to diagnostic mode by writing 1A*hex*, 007F*hex*, and 0*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.
2. Find a "refresh" arbitration cycle.
   - Pulse **Clock Generator** CLK(H) U110D-11 <27> (by toggling **Diagnostics** U530-7) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.
3. Process the "refresh" cycle by pulsing CLK(H) U110D-11 20 times.
4. Start a "read digitizer" cycle by setting DIGREQ(L) U524-12 <27> low and then high.
5. Find the "read digitizer" cycle.
   - Pulse CLK(H) U110D-11 <27> until a "read digitizer" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.
6. Select the RAG Register 0 by writing the tag number of the register (0000*hex*) to RAG Tag Register in the **MMU Gate Array**.
7. Write pattern 0001*hex* to RAG Register 0.
8. Start the data handshake by setting DIGACK(H) U524-9 <27> high and then low.
9. Pulse CLK(H) U110D-11 two times and read the low byte of the RAG output from **Diagnostics** U112. Pulse CLK(H) U110D-11 two more times and read the high byte of the RAG output from **Diagnostics** U112 <27>. Swap Bit 9 with bit 16 of the RAG output.
10. Verify the RAG output against the known value.
11. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
12. Repeat steps 1-11 for patterns 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, and 0100*hex* in RAG Register 0.
13. Repeat steps 1-12 for RAG Registers 1-14.
14. Restore the RAG register 14 for normal operating firmware.

Error Index E5181 RAG register 0 failed the test.

Error Index E5182	RAG register 1 failed the test.
Error Index E5183	RAG register 2 failed the test.
Error Index E5184	RAG register 3 failed the test.
Error Index E5185	RAG register 4 failed the test.
Error Index E5186	RAG register 5 failed the test.
Error Index E5187	RAG register 6 failed the test.
Error Index E5188	RAG register 7 failed the test.
Error Index E5189	RAG register 8 failed the test.
Error Index E518A	RAG register 9 failed the test.
Error Index E518B	RAG register 10 failed the test.
Error Index E518C	RAG register 11 failed the test.
Error Index E518D	RAG register 12 failed the test.
Error Index E518E	RAG register 13 failed the test.
Error Index E518F	RAG register 14 failed the test.

Routine Name RAG Intrpt (Random Address Generator Interrupt)

Overview This test verifies **MMU Gate Array's** U210 <27> RAG interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the **MMU Gate Array** U210 <27> to an isolated diagnostic mode by writing 1A*hex*, 007F*hex*, and 4*hex* to **Diagnostics** U524 <27>, U210 Status and Mode Register (SMR), and **Diagnostics** U530 <27>, respectively.
2. Verify that there is no RAG interrupt pending at **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 1 U360 <24> INTERRUPT REQUEST REGISTER and verify that IR1 U360-19 is low.
3. If there is no RAG interrupt pending, enable the RAG interrupt at the **Interrupt Controllers** <24> by setting Interrupt Controllers MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 1 low.
4. Generate an "end of digitizer message" interrupt.
   - Set DIGREQ(L) U524-12 <27> low and then high with U524-19(H):H <27>.
   - Set DIGREQ(L) U524-12 <27> high with U524-19(H):L <27>.
5. Perform a software delay and then verify that the RAG interrupt did occur.
6. Clear the RAG interrupt at the MMU by setting bit 4 of the Status and Mode Register (SMR) high.
7. Disable RAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 1 high.
8. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

Error Index E5191 The RAG interrupt did not occur.

Error Index E5192 The RAG interrupt could not be cleared at the **Interrupt Controllers** SLAVE U360 <24>.

Routine Name Size

Overview This test verifies the waveform RAM size selection on the Memory Management Board <27> by testing one location of each 64k byte waveform RAM memory segment in **Odd DRAM** <28> and **Even DRAM** <28>.

Description

1. Reset **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register (SMR).
2. Disable "read digitizer" channel by setting bit 6 of SMR low.
3. Read jumper J201 <27> position by reading **Diagnostics** LGMEM(H) U501B-11 <27>.
4. If LGMEM(H) <27> is high (indicating 256k bytes of RAM), then write patterns 0001*hex*, 0002*hex*, 0004*hex*, and 0008*hex* to waveform RAM locations 40000*hex*, 50000*hex*, 60000*hex*, and 70000*hex*, respectively. Else, write patterns 0001*hex* and 0002*hex* to waveform locations 40000*hex* and 50000*hex*.
5. Read all the locations that were written and verify that their contents have remained the same.
6. Enable "read digitizer" channel by setting bit 6 of SMR high.

Error Index E5211 One of the patterns read from the first location in a 64k byte waveform memory segment was not the pattern written, indicating that either the J201 jumper selection was inappropriate or a waveform RAM problem was detected.

Routine Name Data Lines

Overview This test verifies the data line independence of waveform RAM by performing a "walking one's" test on an Odd DRAM <28> location.

Description

1. Reset MMU Gate Array U210 <27> to normal mode by writing 5*hex* and 007F*hex* to Diagnostics U530 <27> and U210 Status and Mode Register (SMR).
2. Disable "read digitizer" channel by setting bit 6 of SMR low.
3. Read jumper J201 <27> position by reading Diagnostics LGMEM(H) U501B-11 <27>.
4. If LGMEM(H) <27> was high, then select 7FFFE*hex* as the test location. Else, select 5FFFE*hex* as the test location. Perform a "walking one's" test on the selected test location.
   - Write, read, and verify patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*.
5. Enable "read digitizer" channel by setting bit 6 of SMR high.

Error Index E5221 The pattern read from the Odd DRAM <28> memory location was not the pattern written. The correspondence of the data bits to the Odd DRAM devices is shown in the following tables:

SMALL WAVEFORM MEMORY (J201—128k position)

Address	D0—D3	D4—D7	D8—D11	D12—D15
5FFFE*hex*	U230	U224	U222	U220

LARGE WAVEFORM MEMORY (J201—256k position)

Address	D0—D3	D4—D7	D8—D11	D12—D15
7FFFE*hex*	U130	U124	U122	U120

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of **Odd DRAM** <28> and **Even DRAM** <28> by performing a RAM test on all locations.

Description

1. Reset the **MMU Gate Array** U210 <27> to normal mode by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register (SMR).

2. Disable "read digitizer" channel by setting bit 6 of SMR low.

3. Read jumper J201 <27> position by reading **Diagnostics** LGMEM(H) U501B-11 <27>.

4. Verify waveform RAM address range 40000*hex*-4FFFF*hex* (64k bytes). Terminate test if any verify operation fails.

   - Fill the address range 40000*hex*-4FFFF*hex* with the pattern AAAA*hex*.

   - Read and verify address 40000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 40000*hex*. Increment address and continue this read/verify/write sequence until address 4FFFF*hex* is reached.

   - Repeat the read/verify/write sequence, starting again at 40000*hex* for CCCC*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

5. If LGMEM(H) <27> is high, repeat step 4 for waveform address ranges 50000-5FFFF*hex*, 60000-6FFFF*hex* and 70000-7FFFF*hex*. Else, repeat step 4 only for the address range 50000-5FFFF*hex*.

Error Index E5231

The pattern read from an **Odd DRAM** <28> or **Even DRAM** <28> memory location did not match the pattern written. The correspondence of the address and data bits to DRAM memory devices is shown in the following tables:

SMALL WAVEFORM MEMORY (J201—128k position)

Address(*hex*)	D0—D3	D4—D7	D8—D11	D12—D15
4XXX0	U330	U324	U322	U320
4XXX4	U330	U324	U322	U320
4XXX8	U330	U324	U322	U320
4XXXC	U330	U324	U322	U320
4XXX2	U130	U124	U122	U120
4XXX6	U130	U124	U122	U120
4XXXA	U130	U124	U122	U120
4XXXE	U130	U124	U122	U120
5XXX0	U430	U424	U422	U420
5XXX4	U430	U424	U422	U420
5XXX8	U430	U424	U422	U420
5XXXC	U430	U424	U422	U420
5XXX2	U230	U224	U222	U220
5XXX6	U230	U224	U222	U220
5XXXA	U230	U224	U222	U220
5XXXE	U230	U224	U222	U220

X—Don't Care.

LARGE WAVEFORM MEMORY (J201—256k position)

Address(*hex*)	D0—D3	D4—D7	D8—D11	D12—D15
XXXX0	U330	U324	U322	U320
XXXX4	U330	U324	U322	U320
XXXX8	U330	U324	U322	U320
XXXXC	U330	U324	U322	U320
XXXX2	U130	U124	U122	U120
XXXX6	U130	U124	U122	U120
XXXXA	U130	U124	U122	U120
XXXXE	U130	U124	U122	U120

X—Don't Care.

Routine Name Reset

Overview This test verifies the Waveform Compressor/Adder (WCA) **Reset Control** <29> by checking that the WCA returns to a correct state after reset.

Description

1. Reset WCA.
   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex* to the **Compression Factor Counter** Register U824 <29>, and 05*hex* to the **Mode Select Latch** U724 <29>.
2. Read WCA **Status Read Back** buffer U822 <29> and verify that it reads 04*hex*.

Error Index E5311 The WCA does not return to a correct state after reset. The information read back from the **Status Read Back** buffer U822 <29> is shown below.

bit 0 (D0) : ADDAV(L)
bit 1 (D1) : IDLE(L)
bit 2 (D2) : OVER(H)
bit 3 (D3) : UNDER(H)
bit 4 (D4) : NULL(H)
bit 5 (D5) : M/MUND(H)
bit 6 (D6) : ALLNULL(H)
bit 7 (D7) : EOC(H)

Routine Name Allnull

Overview

This test verifies the capability of **Compressor Control** <29> and **Flags Decode** <30> to recognize Null and non-Null data values. **Status Read Back** bit ALLNULL <29> is observed while sending three sets of Null and non-Null data values from the **MMU Gate Array** <27> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 4.
   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 04*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send a Null data value (8000*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that **Compressor Control** U120B-9 ALLNULL(H) <29> is high.
6. Manipulate the MMU Gate Array <27> to send two words (8200*hex*, 8000*hex*) from the waveform RAM to the WCA.
7. Perform a software delay.
8. Read **Status Read Back** buffer U822 <29> and verify that **Compressor Control** U120B-9 ALLNULL(H) <29> is low.
9. Manipulate the **MMU Gate Array** <27> to send two more words (8000*hex*, 8200*hex*) from the waveform RAM to the WCA.
10. Perform a software delay.
11. Read **Status Read Back** buffer U822 <29> and verify that **Compressor Control** U120B-9 ALLNULL(H) <29> is low.
12. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

13. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5321 The WCA did not recognize a Null data value, i.e., the ALLNULL bit read from **Status Read Back** buffer U822 <29> was low when it should have been high.

Error Index E5322 The WCA did not recognize a non-Null data value, i.e., the ALLNULL bit read from **Status Read Back** buffer U822 <29> was low when it should have been high.

Error Index E5323 The WCA did not recognize a non-Null data value in a group containing both Null and non-Null data values.

Routine Name M/Mund

Overview This test verifies that **Compressor Control** <29> is capable of selecting new minimum and maximum data points for each data group in the non-vectored mode and is capable of using the previous minimum and maximum data points for the next data group in the vectored mode. **Status Read Back** bit M/MUND <29> is observed while sending two sets of two words from the **MMU Gate Array** <27> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 3.
   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 03*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send two words (8200*hex* and 8300*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that **Compressor Control** U124B-9 M/MUND(H) <29> is high, i.e., new minimum and maximum data points need to be selected.
6. Manipulate the **MMU Gate Array** <27> to send two more words (8400*hex* and 8500*hex*) from the waveform RAM to the WCA.
7. Perform a software delay.
8. Read **Status Read Back** buffer U822 <29> and verify that **Compressor Control** U124B-9 M/MUND(H) <29> is high, i.e., new minimum and maximum data points need to be selected.
9. Repeat steps 3-8 with the WCA in vectored mode by writing 07*hex* to **Mode Select Latch** U724 <29> and verify that M/MUND(H) <29> is high in step 5 and low in step 8.
10. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

11. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5331 **Compressor Control** M/MUND(H) <29> was low when the minimum and maximum data points were undefined in non-vectored mode.

Error Index E5332 **Compressor Control** M/MUND(H) <29> was low after one cycle in non-vectored mode. In non-vectored mode, M/MUND(H) should have been high after each cycle since the old minimum and maximum data points are not remembered.

Error Index E3333 **Compressor Control** M/MUND(H) <29> was low when the minimum and maximum data points were not defined under vectored mode.

Error Index E5334 **Compressor Control** M/MUND(H) <29> was high after one cycle in vectored mode. In vectored mode, the previous minimum and maximum data points are remembered. So, they will not be undefined.

Routine Name

Idle

Overview

This test verifies that **Compressor Control** <29> and **Adder Control** <31> reflect the correct state (busy or idle) of the compressor and the adder. **Status Read Back** buffer bits IDLE(L) and ADDAV(L) <29> are observed while sending two sets of two words from the **MMU Gate Array** <27> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.

   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

2. Reset WCA to non-vectored test mode with a compression factor of 2.

   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.

   - Write 02*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.

3. Manipulate the **MMU Gate Array** <27> to send two words (8200*hex* and 8300*hex*) from the waveform RAM to the WCA.

4. Perform a software delay.

5. Verify that the compressor is idle and the adder is busy.

   - Read **Status Read Back** buffer U822 <29> and verify that **Compressor Control** U222-17 IDLE(L) <29> is low and **Adder Control** U224-19 ADDAV(L) <31> is high.

6. Manipulate the **MMU Gate Array** <27> to send two more words (8400*hex* and 8500*hex*) from the waveform RAM to the WCA.

7. Perform a software delay.

8. Verify that both the compressor and the adder are busy.

   - Read **Status Read Back** buffer U822 <29> and verify that **Compressor Control** U222-17 IDLE(L) <29> is high and **Adder Control** U224-19 ADDAV(L) <31> is high.

9. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

10. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5341 The compressor was not busy when it should have been.

Error Index E5342 The adder and/or the compressor was not busy when it should have been.

Routine Nam Comprss Null (Compress Null)

Overview This test verifies that **Flags Decode** <30> detects a Null data point appropriately in compress mode by sending the Compressor a Null data point (8000*hex*).

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 1.
   - Toggle Reset Control U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send a Null data point (8000*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that **Flags Decode** U100C-8 NULL(H) <30> is high.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5351 The Null data point is not being detected as a null condition by the WCA in compress mode.

Routine Name Xparent Null (Transparent Null)

Overview This test verifies that **Flags Decode** <30> detects a Null data point appropriately in transparent mode by sending the Compressor a Null data point (8000*hex*).

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex* and 04*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send a Null data point (8000*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that **Flags Decode** U100C-8 NULL(H) <30> is low.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5361 The Null data value is being detected as a Null condition by the WCA in the transparent mode when it shouldn't be.

Routine Name

Comprss Over (Compress Overrange)

Overview

This test verifies that **Flags Decode** <30> detects an Over data point appropriately in the compress mode by sending the Compressor an overrange value (7FFF*hex*).

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.

2. Reset WCA to non-vectored test mode with a compression factor of 1.

   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.

   - Write 01*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.

3. Manipulate the **MMU Gate Array** <27> to send an overrange data value (7FFF*hex*) from the waveform RAM to the WCA.

4. Perform a software delay.

5. Read **Status Read Back** buffer U822 <29> and verify that **Flags Decode** U100A-3 OVER(H) <30> is high.

6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5371

The overrange data value is not being decoded into the OVER flag properly in the compress mode.

Routine Name Xparent Over (Transparent Overrange)

Overview This test verifies that **Flags Decode** <30> detects an Over data point appropriately in the transparent mode by sending the Compressor an overrange value (7FFF*hex*).

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex* and 04*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send an overrange data value (7FFF*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that **Flags Decode** U100A-3 OVER(H) <30> is low.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5381 The overrange data value is being decoded into the OVER flag in the transparent mode when it shouldn't be.

Routine Name Comprss Undr (Compress Underrange)

Overview This test verifies that **Flags Decode** <30> detects an Under data point appropriately in the compress mode by sending the Compressor an underrange data value (8001*hex*).

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 1.
   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send an underrange data value (8001*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that Flags Decode U100B-6 UNDER(H) <30> is high.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5391 The underrange data value is not being decoded properly into the UNDER flag in the compress mode.

Routine Name Xparent Undr (Transparent Underrange)

Overview This test verifies that **Flags Decode** <30> detects an Under data point appropriately in the transparent mode by sending the Compressor an underrange data value (8001*hex*).

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex* and 04*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send an underrange data value (8001*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that **Flags Decode** U100B-6 UNDER(H) <30> is low.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E53A1 The underrange data value is being decoded into the UNDER flag in the transparent mode when it shouldn't be.

Routine Name Non Special

Overview This test verifies that **Flags Decode** <30> does not generate special case flags (Null, Over and Under) for non-special data by sending the Compressor non-special data values.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 1.
   - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send a non-special data value (0001*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <29> and verify that U100C-8 NULL(H), U100A-3 OVER(H), and U100B-6 UNDER(H) are all low.
6. Repeat steps 2-5 for non-special case data values 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, and 4000*hex*.
7. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
8. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E53B1 The WCA decoded one of the non-special case data values as a special case, i.e., an Under, Over or Null condition was flagged.

Routine Name

Max Special

Overview

This test verifies that the **X Comparator & Y Comparator** <30> produce the proper criterion for replacing the maximum value in **X Min Max Latch Decode** <30> & **Y Min Max Latch Decode** <30>. The comparator outputs are observed through **Status Read Back** buffer U622 <29> while special case data values (i.e., Null, Over and Under) are sent from the **MMU Gate Array** <27> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.

   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

2. Reset WCA to non-vectored test mode with a compression factor of 2.

   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 02*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.

3. Manipulate the **MMU Gate Array** <27> to send a Null data value (8000*hex*) from the waveform RAM to the WCA. This becomes the "current" data value for the comparator's subsequent comparisons since this is the first data value after reset.

4. Perform a software delay.

5. Manipulate the **MMU Gate Array** <27> to send an Under data value (8001*hex*) from the waveform RAM to the WCA. This becomes the "input" data value for the comparator's comparison.

6. Read **Status Read Back** U622 <29> and verify that it reads 55*hex*.

7. Repeat steps 2-6 for the following "current" and "input" combinations.

Current (*hex*) Data Value	Input (*hex*) Data Value	Expected Comparator Outputs (*hex*)
8000	7FFF	55
8000	8200	55
8001	8000	77
8001	7FFF	5A
8001	8200	5B
7FFF	8000	FF
7FFF	8001	F5
7FFF	8200	F5
8200	8000	DD
8200	8001	E5
8200	7FFF	5A

8. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

9. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5411 The output of the comparators did not match the known value. The display shows the first output that failed.

Routine Name Max DataLine (Max Data Lines)

Overview This test verifies the **X Comparator** <30> data lines by performing a "walking zero's" pattern test on the **X Comparator** <30> data lines.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 16.
   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 10*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send the pattern 7FFE*hex* from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read the **Status Read Back** U622 <29> and verify that it reads 55*hex*.
6. Repeat steps 2-6 for the following patterns.

Input (*hex*) Data Value	Expected Comparator Outputs (*hex*)
7FFD	75
7FFB	75
7FF7	75
7FEF	75
7FDF	75
7FBF	75
7F7F	75
7EFF	E5
7DFF	E5
7BFF	E5
77FF	E5
6FFF	E5
5FFF	E5
3FFF	E5
FFFF	E5

8. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

9. Disable SAG interrupt at the **Interrupt Controllers <24>** by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5421 The output of the comparators did not match the known value. The display shows the first output that failed.

Routine Name

Min DataLine (Min Data Lines)

Overview

This test verifies **Y Comparator** <30> data lines by performing a "walking one's" pattern test on the **Y Comparator** <30> data lines.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.

   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

2. Reset WCA to non-vectored test mode with a compression factor of 14.

   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 0E*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29> and the **Mode Select Latch** U724 <29>.

3. Manipulate the **MMU Gate Array** <27> to send the pattern C000*hex* from the waveform RAM to the WCA.

4. Perform a software delay.

5. Read **Status Read Back** U622 <29> and verify that it reads 55*hex*.

6. Repeat steps 2-6 for the following patterns.

Input (*hex*) Data Value	Expected Comparator Outputs (*hex*)
A000	5D
9000	5D
8800	5D
8400	5D
8200	5D
8100	5D
8080	5E
8040	5E
8020	5E
8010	5E
8008	5E
8004	5E
8002	5E

8. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

9. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5431 The output of the comparators did not match the known value. The display shows the first output that failed.

Routine Name Min Special

Overview This test verifies that the **X Comparator & Y Comparator** <30> produce the proper criterion for replacing the minimum value in **X Min Max Latch Decode** <30> & **Y Min Max Latch Decode** <30>. The comparator outputs are observed through **Status Read Back** buffer U622 <29> while special case data values (i.e. Null, Over and Under) are sent from the **MMU Gate Array** <27> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 2 and switch the minimum and maximum registers.
   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 02*hex* and 05*hex* to the **Compression Factor Counter** Register U824 <29>and the **Mode Select Latch** U724 <29>.
   - Set U332A-4 <29> low by writing 0000*hex* to **Address Decode/Select** port 1008*hex* <29>.
3. Manipulate the **MMU Gate Array** <27> to send a Null data value (8000*hex*) from the waveform RAM to the WCA. This becomes the "current" data for the comparator's subsequent comparisons since this is the first data value after reset.
4. Perform a software delay.
5. Manipulate the **MMU Gate Array** <27> to send an Under data value (8001*hex*) from the waveform RAM to the WCA. This becomes the "input" data value for the comparator's comparison.
6. Read the **Status Read Back** U622 <29> and verify that it reads 55*hex*.

7. Repeat steps 2-6 for the following "current" and "input" combinations.

Current (*hex*) Data Value	Input (*hex*) Data Value	Expected Comparator Outputs (*hex*)
8000	7FFF	55
8000	8200	55
8001	8000	77
8001	7FFF	A5
8001	8200	B5
7FFF	8000	FF
7FFF	8001	5F
7FFF	8200	5F
8200	8000	DD
8200	8001	5E
8200	7FFF	A5

8. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

9. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5441 The output of the comparators did not match the known value. The display shows the first comparator output that failed.

Routine Name

Offset

Overview

This test verifies the **Adder Offset Register** U624 & U722 <29> by performing a "walking one's" test on this register. The "walking one" pattern in the Offset Register is added to a 0000*hex* waveform data value and read back from **Adder Read Back** <29>.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and ModeRegister.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to Adder Decode/Select port 1002*hex*.
   - Write 01*hex* and 04*hex* to the **Compression Factor Counter** Register U824 <29> and the Mode Select Latch U724 <29>.
3. Manipulate the **MMU Gate Array** <27> to send a 0000*hex* waveform value from the waveform RAM to the WCA.
4. Perform a software delay.
5. Write 0001*hex* to **Adder Offset Register** U624 & U722 <29> of the WCA.
6. Read and verify the output (0001*hex*) of the **Adder** from **Adder Read Back** <29>.
7. Repeat steps 2-6 for patterns 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*.
8. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
9. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5511

The pattern read from **Adder Read Back** <29> port did not match the pattern written to the **Adder Offset Register** U624 & U722 <29>. The display shows the first pattern that failed.

Routine Name

Data Paths

Overview

This test verifies the data paths through **MMU Input Register** <30>, **X Comp Output Latch** <30>, **Y Comp Output Latch** <30>, **Adder Input X Latch** <31>, **Adder Input Y Latch** <31>, **Adder Input Mux** <31>, **Adder** <31>, **Adder Output Select** <31> and **Adder Read Back** <29> by performing a "walking one's" test on the data lines. In addition, the capability of **Output Word Control** <31> to recognize special data values (i.e., Null, Under and Over) is verified.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.

   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

2. Reset WCA to non-vectored test mode with a compression factor of 1 and an offset of 0.

   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Adder Decode/Select** port 1002*hex*.
   - Write 01*hex*, 05*hex* and 0000*hex* to the **Compression Factor Counter Register** U824 <29>, **Mode Select Latch** U724 <29>, and **Adder Offset Register** U624 & U722 <29>, respectively.

3. Manipulate the **MMU Gate Array** <27> to send pattern 0001*hex* from the waveform RAM to the WCA.

4. Perform a software delay.

5. Read and verify the data path through **MMU Input Register** <30>, **X Comp Output Latch** <30>, **Adder Input X Latch** <31>, **Adder Input Mux** <31>, **Adder** <31>, and **Adder Output Select** <31> by reading **Adder Read Back** <29> (value should be 0001*hex*). **Read Adder Read Back** <29> once more to verify the path through **Y Comp Output Latch** <30>, **Adder Input Y Latch** <31> and **Adder Input Mux** <31> (the value should again be 0001*hex*).

6. Repeat steps 2-5 with patterns 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, 8000*hex*, 8001*hex*, and 7FFF*hex*.

7. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.

8. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5521 The data value read from **Adder Read Back** <29> through the "X path" was not the same as the pattern sent from the MMU Gate Array <27>. The display shows the first pattern that failed.

Error Index E5522 The data value read from **Adder Read Back** <29> through the "Y path" was not the same as the pattern sent from the MMU Gate Array <27>. The display shows the first pattern that failed.

Routine Name

Overrange

Overview

This test verifies that the **Output Word Control** <31> causes **Adder Output Select** <31> to generate an overrange data value when the sum of the waveform data point sent from the **MMU Gate Array** <27> and the offset from the **Adder Offset Register** <29> exceeds an overrange value (7FFF*hex*).

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.

   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

2. Reset WCA to transparent non-vectored test mode with a compression factor of 1 and an offset of 2.

   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Adder Decode/Select** port 1002*hex*.
   - Write 01*hex*, 04*hex*, and 0002*hex* to the **Compression Factor Counter Register** U824 <29>, **Mode Select Latch** U724 <29>, **Adder Offset Register** U624 & U722 <29>, respectively.

3. Manipulate the **MMU Gate Array** <27> to send an overrange data value (7FFF*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Adder Read Back** <29> and verify that it is 7FFF*hex*.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5531

The **Adder Output Select** <31> did not generate an overrange data value when the sum of waveform data point and the offset exceeded the overrange data value.

Routine Name Underrange

Overview This test verifies that the **Output Word Control** <31> causes **Adder Output Select** <31> to generate an underrange data value (8001*hex*) when the sum of the waveform data point sent from the **MMU Gate Array** <27> and the offset from **Adder Offset Register** <29> is below an underrange value.

Description

1. Initialize the **MMU Gate Array** U210 <27> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <24>.
   - Write 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.
   - Set **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1 and an offset of – 4*hex* (FFFC*hex*).
   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.
   - Write 01*hex*, 04*hex* and FFFC*hex* to the **Compression Factor Counter** Register U824 <29>, **Mode Select Latch** U724 <29> and **Adder Offset Register** U624 & U722 <29>, respectively.
3. Manipulate the **MMU Gate Array** <27> to send a waveform data word (8002*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Adder Read Back** <29> and verify that it is 8001*hex*.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <29>.
7. Disable SAG interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 0 low.

Error Index E5541 The **Adder Output Select** <31> did not generate an underrange data value when the sum of the waveform data point and the offset is less than the underrange value.

Routine Name

Display

Overview

This test verifies the communication path from the Executive processor to the Display processor by performing a RAM test on communication message area of waveform RAM and a "walking one's" and "walking zero's" pattern test on the data path through the **Even & Odd DRAM** <28>, **Display Data Buffers** <28>, and the Waveform Compressor/Adder (WCA) <30> <31>.

On power-up, this test is used by the Executive processor, after it has successfully completed its kernel testing, to detect the "presence" of the Display subsystem. In other words, whether the Display processor has successfully reached the communication stage of its kernel testing and can functionally communicate with the Executive processor through the previously mentioned hardware. At this time, when this test executes, the Display subsystem status is "not present" and the corresponding "Display Subsystem Not Present" section in the following description is the test path executed by the Executive processor.

Description

1. Initialize **MMU Gate Array** U210 <27> by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

2. Reset the WCA.

   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.

   - Write 00*hex*, 01*hex*, and 0000*hex* to the **Mode Select Latch** U724 <29>, **Compression Factor Counter Register** U824 <29>, and **Adder Offset Register** U722 & U624 <29>, respectively.

   - Toggle **Reset Control** U130D-11 WCARST(H) <29> by writing 0000*hex* to **Address Decode/Select** port 1002*hex* <29>.

3. Verify the waveform RAM memory locations used for transmitting and receiving messages. If the waveform RAM size is 128k bytes (small memory), test address range 5F000-5FFFF*hex*; otherwise test 7F000-7FFFF*hex* (large memory). Terminate test if any verify operation failed.

   - Fill the address range (5F000-5FFFF*hex* or 7F000-7FFF*hex*) with pattern AAAA*hex*.

   - Read and verify the first address location (5F000*hex* or 7F000*hex*) for AAAA*hex*. If so, write CCCC*hex* to the first address location. Increment address and continue this read/verify/write sequence until the last address in the address range is reached.

   - Repeat the read/verify/write sequence, starting again at the first address location for CCCC*hex*, 5555*hex* and 0000*hex* (i. e., reading and verifying the previous pattern written and then writing the next pattern).

4. If there is no error in the waveform RAM, check to see if the Display subsystem is "present". If the Display subsystem was detected to be "present" (i.e., have functional communication) on power-up by the Executive processor, then skip to step 10.

"Display Subsystem Not Present"

5. Fill the first 32 locations in the Display message area in the waveform RAM (5F100-5F13F*hex* or 7F100-7F13F*hex*) with "walking one's" and "walking zero's" patterns (see Error Index section below).

6. Program the MMU Gate Array U210 <27> Sequential Address Generator (SAG) to write a pattern to the Display and start the "write display" channel.

   - Set bit 3 of the Status and Mode Register (SMR) to clear any pending SAG interrupt.
   - Calculate the appropriate values for Message Length Register (MLR), Address Counter (AC) and Message Pointer Register (MPR) so that the SAG will transfer the waveform RAM location containing the pattern to be transmitted. Write the calculated values to the corresponding registers.
   - Start the "write display" channel by setting bit 9 of SMR.
   - Wait for the SAG to complete the transfer—repeatedly read SMR until bit 3 of SMR is high or a software timeout period expires.
   - Irrespective of the completion of transfer, clear the SAG interrupt by setting bit 3 of SMR high.

7. Receive the pattern (bit-wise inversion of the transmitted pattern) from the Display subsystem.

   - Wait for the display talk request interrupt—repeatedly read SMR until bit 1 of SMR is high or a software timeout period expires.
   - If bit 1 of SMR is high, then clear the display talk request interrupt by setting bit 1 of SMR high.
   - Program the SAG to receive one word from the Display to the same location it was transmitted from.
   - Start the "read display" channel by setting bit 10 of SMR.
   - Wait for the SAG to complete the transfer—repeatedly read SMR until bit 3 of SMR is high or a software timeout period expires.
   - On completion of the transfer, clear the SAG interrupt by setting bit 3 of SMR high.

8. Read and verify the waveform memory location from which the pattern was transmitted (see Error Index section below). If the received pattern was incorrect, then set the next pattern to be sent to DEAD*hex*. This will cause the Display subsystem, upon receiving the invalid pattern, to enter a continuous loop in which it inverts and echoes every pattern sent thereafter.

9. Repeat steps 6-8 for the remaining patterns and then terminate the test.

"Display Subsystem Present"

10. Enable MMU interrupts so that normal operating communications can take place.

11. Using the normal operating communications (similar to above but with multiple words and additional information), send a block of 32 words containing "walking one's" and "walking zero's" patterns (see Error Index section below) to the Display subsystem.

12. Using the normal operating communications, receive the block of bit-wise inverted patterns from the Display subsystem and verify that they are correct.

13. Disable MMU interrupts.

Error Index E5611 The pattern received from the Display is not the bit-wise inversion of the pattern sent. The display shows the first pattern that failed.

The patterns sent and those expected to be received are shown in the following table:

Pattern Sent (*hex*)	Expected Pattern Received (*hex*)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	EFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
EFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004

FFFD	0002
FFFE	0001

Error Index E5612 No pattern was received from the Display when a pattern was being expected from it. The display shows the first pattern that was not received.

Error Index E5613 One or more of the patterns in the block of patterns received from the Display was incorrect. The display shows the first pattern that failed (see table above for pattern set).

Error Index E5615 The waveform RAM used for transferring messages to and from the Display has a fault. The pattern read was not the pattern written. The display shows the first pattern that failed.

See Also The Subsys Comm Waveform RAM Address/Data test (E523X) if the error index was E5615 in to order see the correspondence of the address and data bits to RAM devices.

Routine Name

Digitizer

Overview

This test verifies the communication path from the Executive processor to the Digitizer processor by performing a RAM test on communication message area of waveform RAM and a "walking one's" and "walking zero's" pattern test on the data path through the **Even & Odd DRAM** <28> and **Digitizer Data Latches** <28>.

On power-up, this test is used by the Executive processor, after it has successfully completed its kernel testing, to detect the "presence" of the Digitizer subsystem. In other words, whether the Digitizer processor has successfully reached the communication stage of its kernel testing and can functionally communicate with the Executive processor through the previously mentioned hardware. At this time, when this test executes, the Digitizer subsystem status is "not present" and the corresponding "Digitizer Subsystem Not Present" section in the following description is the test path executed by the Executive processor.

Description

1. Initialize **MMU Gate Array** U210 <27> by writing 5*hex* and 007F*hex* to **Diagnostics** U530 <27> and U210 Status and Mode Register.

2. Verify the waveform RAM memory locations used for transmitting and receiving messages. If the waveform RAM size is 128k bytes (small memory), test address range 5F000-5FFFF*hex*; otherwise test 7F000-7FFFF*hex* (large memory). Terminate test if any verify operation failed.

   - Fill the address range (5F000-5FFFF*hex* or 7F000-7FFFF*hex*) with pattern AAAA*hex*.

   - Read and verify the first address location (5F000*hex* or 7F000*hex*) for AAAA*hex*. If so, write CCCC*hex* to the first address location. Increment address and continue this read/verify/write sequence until the last address in the address range is reached.

   - Repeat the read/verify/write sequence, starting again at the first address location for CCCC*hex*, 5555*hex* and 0000*hex* (i. e., reading and verifying the previous pattern written and then writing the next pattern).

3. If there is no error in the waveform RAM, check to see if the Digitizer subsystem is "present". If the Digitizer subsystem was detected to be "present" (i.e., have functional communication) on power-up by the Executive processor, then skip to step 10.

"Digitizer Subsystem Not Present"

4. Fill the first 32 locations in the Digitizer transmit message area in the waveform RAM (5F100-5F13F*hex* or 7F100-7F13F*hex*) with "walking one's" and "walking zero's" patterns (see Error Index section below).

5. Set up **MMU Gate Array** <27> Random Address Generator (RAG) register 14 to receive messages in the Digitizer receive message area in the waveform RAM (5FC00-5FC3F*hex* or 7FC00-7FC3F*hex*).

6. Program the **MMU Gate Array** U210 <27> Sequential Address Generator (SAG) to write a pattern to the Digitizer and start the "write digitizer" channel.

- Set bit 3 of the Status and Mode Register (SMR) to clear any pending SAG interrupt.
- Calculate the appropriate values for Message Length Register (MLR), Address Counter (AC) and Message Pointer Register (MPR) so that the SAG will transfer the waveform RAM location containing the pattern to be transmitted. Write the calculated values to the corresponding registers.
- Start the "write digitizer" channel by setting bit 11 of SMR.
- Wait for the SAG to complete the transfer—repeatedly read SMR until bit 3 of SMR is high or a software timeout period expires.
- Irrespective of the completion of transfer, clear the SAG interrupt by setting bit 3 of SMR high.

7. Receive the pattern (bit-wise inversion of transmitted pattern) from the Digitizer subsystem.
   - Wait for the digitizer end message interrupt—repeatedly read SMR until bit 4 of SMR is high or a software timeout period expires.
   - If bit 4 of SMR is high, then clear the digitizer end message interrupt by setting bit 4 of SMR high.
8. Read and verify the waveform memory location to which the Digitizer subsystem wrote (5FC00*hex* or 7FC00*hex*). If the received pattern was incorrect, then set the next pattern to be sent to DEAD*hex*. This will cause the Digitizer subsystem, upon receiving this invalid pattern, to enter a continuous loop in which it inverts and echoes every pattern sent thereafter.
9. Repeat steps 6-8 for the remaining patterns and then terminate the test.

"Digitizer Subsystem Present"

10. Enable MMU interrupts so that normal operating communications can take place.
11. Using the normal operating communications (similar to above but with multiple words and additional information), send a block of 32 words containing "walking one's" and "walking zero's" patterns (see Error Index section below) to the Digitizer subsystem.
12. Using the normal operating communications, receive the block of bit-wise inverted patterns from the Digitizer subsystem and verify that they are correct.
13. Disable MMU interrupts.

Error Index E5621 The pattern received from the Digitizer is not the bit-wise inversion of the pattern sent. The display shows the first pattern that failed.

The patterns sent and those expected to be received are shown in the following table:

Pattern Sent (*hex*)	Expected Pattern Received (*hex*)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	EFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
EFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001

Error Index E5622 No pattern was received from the Digitizer when a pattern was being expected from it. The display shows the first pattern that was not received.

Error Index E5623 One or more of the patterns in the block of patterns received from the Digitizer was incorrect. The display shows the first pattern that failed (see table above for pattern set).

Error Index E5625 The waveform RAM used for transferring messages to and from the Digitizer has a fault. The pattern read was not the pattern written. The display shows the first pattern that failed.

See Also The Subsys Comm Waveform RAM Address/Data test (E523X) if the error index was E5625 in to order see the correspondence of the address and data bits to RAM devices.

Routine Name Left Loop

Overview This test verifies the **Serial Data Interface** SDI IC U330 <21> left channel by internally looping back "walking one" patterns using the self-test hardware.

Description

1. Program the **Serial Data Interface** SDI IC U330 <21> to loop back mode.
   - Set bits 3 and 4 (Receiver Reset and Transmitter Reset bits) of the Miscellaneous Control Select 1 latch high.
   - Write 38*hex* and 00*hex* to the SDI Control Select Register and SDI Interrupt Output Mask Register.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.
2. Read SDI I/O Status Register and verify that bit 4 (Left Receiver Ready bit) is high. Read SDI Status Register and verify that bit 0 (Transmitter Empty bit) is high.
3. Irrespective of the state of the Left Receiver Ready and Transmitter Empty bits, transmit pattern 01*hex* without the EOI (End Or Identify) bit.
   - Wait for the transmitter to be ready by repeatedly reading the SDI Status Register until Transmitter Empty bit is high or a software timeout period expires.
   - Irrespective of the state of the Transmitter Empty bit, write pattern 01*hex* to Left Transmitter Write address (201C*hex*).
4. Read SDI I/O Status Register and verify that Left Receiver Ready bit is low (i.e., the left receiver is ready). Read SDI Status Register and verify that Left EOI bit (bit 7) is high (i.e., the byte received is not the last byte of the current transmission).
5. Irrespective of Left Receiver Ready and Left EOI bits, transmit pattern 02*hex* using the same procedure outlined in step 3.
6. Read SDI Status Register and verify that Transmitter Empty bit is low (i.e., the transmitter is not empty) and Left EOI bit is high (i.e., the byte received is not the last byte of the current transmission).
7. Receive and verify the first pattern transmitted.
   - Wait for left receiver to be full by repeatedly reading SDI Status Register until Left Receiver Full bit (bit 3) is high or a software timeout period expires.
   - If Left Receiver Full bit is high (i.e., the left receiver is full) read the pattern from Left Receiver Read address. Verify that the pattern is 01*hex* and the Left EOI bit of SDI Status Register is high.

8. Repeat the process of transmit/receive/verify for the patterns 04*hex*, 08*hex*, 10*hex*, 20*hex*, and 40*hex* using steps outlined in 3 and 6. All the above patterns are transmitted without the EOI bit. This should cause the Left EOI bit of the SDI Status Register to be high.

9. Transmit pattern 80*hex* with the EOI bit.

   - Wait for the transmitter to be ready by repeatedly reading the SDI Status Register until Transmitter Empty bit (bit 0) is high or a software timeout period expires.

   - Irrespective of the Transmitter Empty bit, write pattern 80*hex* to Left Transmitter Write address with EOI bit (2014*hex*).

10. Receive and verify the pattern to be the last byte.

   - Wait for left receiver to be full by repeatedly reading SDI Status Register until Left Receiver Full bit (bit 3) is high or a software timeout period expires.

   - If Left Receiver Full bit is high, read the pattern from Left Receiver Read address. Verify that the pattern is 80*hex* and the Left EOI bit of the SDI Status Register is low.

11. Program the Serial Data Interface SDI IC U330 <21> to normal mode.

   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch high.

   - Write 3C*hex* to the SDI Control Select Register.

   - Read SDI I/O Status Register to find out the state of Left Data Back (bit 5), Center Data Back (bit 3), and Right Data Back (bit 0) bits.

   - Write a value to the SDI Control Select Register so that bit 2 is high and bits 3, 4, and 5 reflect the state of Left Data Back, Center Data Back, and Right Data Back, respectively.

   - Write 0E*hex* to SDI Interrupt Output Mask Register.

   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.

Error Index E5711 The pattern read from the left SDI channel was not the same as the pattern written or the state (EOI, receiver ready, etc.) of the IC was inappropriate.

Routine Name Center Loop

Overview This test verifies the Serial Data Interface SDI IC U330 <21> center channel by internally looping back "walking one" patterns using the self-test hardware.

Description

1. Program the Serial Data Interface SDI IC U330 <21> to loop back mode.
   - Set bits 3 and 4 (Receiver Reset and Transmitter Reset bits) of the Miscellaneous Control Select 1 latch high.
   - Write 38*hex* and 00*hex* to the SDI Control Select Register and SDI Interrupt Output Mask Register.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.
2. Read SDI I/O Status Register and verify that bit 2 (Center Receiver Ready bit) is high. Read SDI Status Register and verify that bit 0 (Transmitter Empty bit) is high.
3. Irrespective of the state of the Center Receiver Ready and Transmitter Empty bits, transmit pattern 01*hex* without the EOI (End Or Identify) bit.
   - Wait for the transmitter to be ready by repeatedly reading the SDI Status Register until Transmitter Empty bit is high or a software timeout period expires.
   - Irrespective of the state of the Transmitter Empty bit, write pattern 01*hex* to Center Transmitter Write address (2018*hex*).
4. Read SDI I/O Status Register and verify that Center Receiver Ready bit is low (i.e., the center receiver is ready). Read SDI Status Register and verify that Center EOI bit (bit 6) is high (i.e., the byte received is not the last byte of the current transmission).
5. Irrespective of Center Receiver Ready and Center EOI bits, transmit pattern 02*hex* using the same procedure outlined in step 3.
6. Read SDI Status Register and verify that Transmitter Empty bit is low (i.e., the transmitter is not empty) and Center EOI bit is high (i.e., the byte received is not the last byte of the current transmission).
7. Receive and verify the first pattern transmitted.
   - Wait for center receiver to be full by repeatedly reading SDI Status Register until Center Receiver Full bit (bit 2) is high or a software timeout period expires.
   - If Center Receiver Full bit is high (i.e., the center receiver is full) read the pattern from Center Receiver Read address. Verify that the pattern is 01*hex* and the Center EOI bit of SDI Status Register is high.

8. Repeat the process of transmit/receive/verify for the patterns 04*hex*, 08*hex*, 10*hex*, 20*hex*, and 40*hex* using steps outlined in 3 and 6. All the above patterns are transmitted without the EOI bit. This should cause the Center EOI bit of the SDI Status Register to be high.

9. Transmit pattern 80*hex* with the EOI bit.

   - Wait for the transmitter to be ready by repeatedly reading the SDI Status Register until Transmitter Empty bit (bit 0) is high or a software timeout period expires.

   - Irrespective of the Transmitter Empty bit, write pattern 80*hex* to Center Transmitter Write address with EOI bit (2010*hex*).

10. Receive and verify the pattern to be the last byte.

    - Wait for center receiver to be full by repeatedly reading SDI Status Register until Center Receiver Full bit (bit 2) is high or a software timeout period expires.

    - If Center Receiver Full bit is high, read the pattern from Center Receiver Read address. Verify that the pattern is 80*hex* and the Center EOI bit of the SDI Status Register is low.

11. Program the **Serial Data Interface** SDI IC U330 <21> to normal mode.

    - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch high.

    - Write 3C*hex* to the SDI Control Select Register.

    - Read SDI I/O Status Register to find out the state of Left Data Back (bit 5), Center Data Back (bit 3), and Right Data Back (bit 0) bits.

    - Write a value to the SDI Control Select Register so that bit 2 is high and bits 3, 4, and 5 reflect the state of Left Data Back, Center Data Back, and Right Data Back, respectively.

    - Write 0E*hex* to SDI Interrupt Output Mask Register.

    - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.

Error Index E5721 The pattern read from the center SDI channel was not the same as the pattern written or the state (EOI, receiver ready, etc.) of the IC was inappropriate.

Routine Name

Right Loop

Overview

This test verifies the Serial Data Interface SDI IC U330 <21> right channel by internally looping back "walking one" patterns using the self-test hardware.

Description

1. Program the Serial Data Interface SDI IC U330 <21> to loop back mode.
   - Set bits 3 and 4 (Receiver Reset and Transmitter Reset bits) of the Miscellaneous Control Select 1 latch high.
   - Write 38*hex* and 00*hex* to the SDI Control Select Register and SDI Interrupt Output Mask Register.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.
2. Read SDI I/O Status Register and verify that bit 0 (Right Receiver Ready bit) is high. Read SDI Status Register and verify that bit 0 (Transmitter Empty bit) is high.
3. Irrespective of the state of the Right Receiver Ready and Transmitter Empty bits, transmit pattern 01*hex* without the EOI (End Or Identify) bit.
   - Wait for the transmitter to be ready by repeatedly reading the SDI Status Register until Transmitter Empty bit is high or a software timeout period expires.
   - Irrespective of the state of the Transmitter Empty bit, write pattern 01*hex* to Right Transmitter Write address (201A*hex*).
4. Read SDI I/O Status Register and verify that Right Receiver Ready bit is low (i.e., the right receiver is ready). Read SDI Status Register and verify that Right EOI bit (bit 5) is high (i.e., the byte received is not the last byte of the current transmission).
5. Irrespective of Right Receiver Ready and Right EOI bits, transmit pattern 02*hex* using the same procedure outlined in step 3.
6. Read SDI Status Register and verify that Transmitter Empty bit is low (i.e., the transmitter is not empty) and Right EOI bit is high (i.e., the byte received is not the last byte of the current transmission).
7. Receive and verify the first pattern transmitted.
   - Wait for right receiver to be full by repeatedly reading SDI Status Register until Right Receiver Full bit (bit 1) is high or a software timeout period expires.
   - If Right Receiver Full bit is high (i.e., the right receiver is full) read the pattern from Right Receiver Read address. Verify that the pattern is 01*hex* and the Right EOI bit of SDI Status Register is high.

8. Repeat the process of transmit/receive/verify for the patterns 04*hex*, 08*hex*, 10*hex*, 20*hex*, and 40*hex* using steps outlined in 3 and 6. All the above patterns are transmitted without the EOI bit. This should cause the Right EOI bit of the SDI Status Register to be high.

9. Transmit pattern 80*hex* with the EOI bit.

   - Wait for the transmitter to be ready by repeatedly reading the SDI Status Register until Transmitter Empty bit (bit 0) is high or a software timeout period expires.

   - Irrespective of the Transmitter Empty bit, write pattern 80*hex* to Right Transmitter Write address with EOI bit (2012*hex*).

10. Receive and verify the pattern to be the last byte.

    - Wait for right receiver to be full by repeatedly reading SDI Status Register until Right Receiver Full bit (bit 1) is high or a software timeout period expires.

    - If Right Receiver Full bit is high, read the pattern from Right Receiver Read address. Verify that the pattern is 80*hex* and the Right EOI bit of the SDI Status Register is low.

11. Program the Serial Data Interface SDI IC U330 <21> to normal mode.

    - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch high.

    - Write 3C*hex* to the SDI Control Select Register.

    - Read SDI I/O Status Register to find out the state of Left Data Back (bit 5), Center Data Back (bit 3), and Right Data Back (bit 0) bits.

    - Write a value to the SDI Control Select Register so that bit 2 is high and bits 3, 4, and 5 reflect the state of Left Data Back, Center Data Back, and Right Data Back, respectively.

    - Write 0E*hex* to SDI Interrupt Output Mask Register.

    - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.

Error Index E5731 The pattern read from the right SDI channel was not the same as the pattern written or the state (EOI, receiver ready, etc.) of the IC was inappropriate.

Routine Name Interrupt

Overview This test verifies the **Serial Data Interface** SDI IC U330 <21> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program the **Serial Data Interface** SDI IC U330 <21> to loop back mode.
   - Set bits 3 and 4 (Receiver Reset and Transmitter Reset bits) of the Miscellaneous Control Select 1 latch high.
   - Write 38*hex* and 00*hex* to the SDI Control Select Register and SDI Interrupt Output Mask Register.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.
2. Verify that there is no SDI interrupt pending at the **Interrupt Controllers** <24>.
   - Read **Interrupt Controllers** SLAVE 1 U360 <21> INTERRUPT REQUEST REGISTER and verify that IR4 U360-22 is low.
3. If there is no pending interrupt, enable the SDI interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 4 low.
4. Enable the transmitter empty interrupt at the Serial Data Interface SDI IC U330 <21> by writing 01*hex* to the SDI Interrupt Output Mask Register.
5. Perform a software delay and then verify that the SDI interrupt did occur.
6. Disable the SDI interrupt at the **Interrupt Controllers** <24> by setting **Interrupt Controllers** MASTER U350 <24> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <24> Mask Register bit 4 high.
7. Program the **Serial Data Interface** SDI IC U330 <21> to normal mode.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch high.
   - Write 3C*hex* to the SDI Control Select Register.
   - Read SDI I/O Status Register to find out the state of Left Data Back (bit 5), Center Data Back (bit 3), and Right Data Back (bit 0) bits.
   - Write a value to the SDI Control Select Register so that bit 2 is high and bits 3, 4, and 5 reflect the state of Left Data Back, Center Data Back, and Right Data Back, respectively.
   - Write 0E*hex* to SDI Interrupt Output Mask Register.
   - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.

Error Index E5741 The SDI interrupt did not occur.

Error Index E5742 The SDI interrupt could not be cleared at the **Interrupt Controllers** SLAVE 1 U360 <24>.

Routine Name Left

Overview This test verifies the communication path from the Executive processor to the Left Plug-In by performing a pattern test on the data path through the **Serial Data Interface** SDI IC U330 <21>.

Description

1. Enable **Serial Data Interface** SDI IC <21> interrupt so that normal operating communications can take place between the Executive processor and the plug-in units.
2. Using normal operating communications, send a block of 18 bytes containing command headers, "walking one" patterns, and some special patterns (see Error Index section below) to the Left Plug-In.
3. Using normal operating communications, receive the block of bit-wise inverted patterns (except for the first two-byte command headers) from the Left Plug-In and verify that they are correct.
4. Disable **Serial Data Interface** SDI IC <21> interrupt.

Error Index E5811 The Left Plug-In did not respond with any message during the communication test.

Error Index E5812 The pattern received from the Left Plug-In is not the bit-wise inversion of the pattern sent. The display shows the first pattern that did not verify.

The patterns sent and those expected to be received are shown in the following table:

Pattern Sent (*hex*)	Expected Pattern Received (*hex*)
16	16
07	07
FF	00
00	FF
01	FE
02	FD
04	FB
08	F7
10	EF
20	DF
40	BF
80	7F
55	AA
AA	55
33	CC
CC	33
DD	22
BB	44

Error Index ???? The Left Plug-In is not present or is not communicating with the Executive processor.

Routine Name

Center

Overview

This test verifies the communication path from the Executive processor to the Center Plug-In by performing a pattern test on the data path through the **Serial Data Interface** SDI IC U330 <21>.

Description

1. Enable **Serial Data Interface** SDI IC <21> interrupt so that normal operating communications can take place between the Executive processor and the Plug-Ins.

2. Using normal operating communications, send a block of 18 bytes containing command headers, "walking one" patterns, and some special patterns (see Error Index section below) to the Center Plug-In.

3. Using normal operating communications, receive the block of bit-wise inverted patterns (except for the first two- byte command headers) from the Center Plug-In and verify that they are correct.

4. Disable **Serial Data Interface** SDI IC <21> interrupt.

Error Index E5821

The Center Plug-In did not respond with any message during the communication test.

Error Index E5822

The pattern received from the Center Plug-In is not the bit-wise inversion of the pattern sent. The display shows the first pattern that did not verify.

The patterns sent and those expected to be received are shown in the following table:

Pattern Sent (*hex*)	Expected Pattern Received (*hex*)
16	16
07	07
FF	00
00	FF
01	FE
02	FD
04	FB
08	F7
10	EF
20	DF
40	BF
80	7F
55	AA
AA	55
33	CC
CC	33
DD	22
BB	44

Error Index ????

The Center Plug-In is not present or is not communicating with the Executive processor.

Routine Name

Right

Overview

This test verifies the communication path from the Executive processor to the Right Plug-In by performing a pattern test on the data path through the **Serial Data Interface** SDI IC U330 <21>.

Description

1. Enable **Serial Data Interface** SDI IC <21> interrupt so that normal operating communications can take place between the Executive processor and the Plug-Ins.

2. Using normal operating communications, send a block of 18 bytes containing command headers, "walking one" patterns, and some special patterns (see Error Index section below) to the Right Plug-In.

3. Using normal operating communications, receive the block of bit-wise inverted patterns (except for the first two- byte command headers) from the Right Plug-In and verify that they are correct.

4. Disable **Serial Data Interface** SDI IC <21> interrupt.

Error Index E5831

The Right Plug-In did not respond with any message during the communication test.

Error Index E5832

The pattern received from the Right Plug-In is not the bit-wise inversion of the pattern sent. The display shows the first pattern that did not verify.

The patterns sent and those expected to be received are shown in the following table:

Pattern Sent (*hex*)	Expected Pattern Received (*hex*)
16	16
07	07
FF	00
00	FF
01	FE
02	FD
04	FB
08	F7
10	EF
20	DF
40	BF
80	7F
55	AA
AA	55
33	CC
CC	33
DD	22
BB	44

Error Index ????

The Right Plug-In is not present or is not communicating with the Executive processor.

Routine Name U612

Overview This test verifies that **ROM & Select** U612 <33> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index D1111 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Error Index ???? The bytes at E0008*hex* and E000A*hex* were not complementary.

Routine Name U602

Overview This test verifies that **ROM & Select** U602 <33> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index D1121 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Error Index ???? The bytes at E0009*hex* and E000B*hex* were not complementary.

Routine Name U612

Overview This test verifies the integrity of ROM & Select U612 <33> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U612 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index D1211 The computed checksum did not match the stored checksum.

Error Index ???? The bytes at E0008*hex* and E000A*hex* were not complementary.

Routine Name U602

Overview This test verifies the integrity of ROM & Select U602 <33> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U602 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index D1221 The computed checksum did not match the stored checksum.

Error Index ???? The bytes at E0009*hex* and E000B*hex* were not complementary.

Routine Name

Data Lines

Overview

This test verifies the data lines from the CPU , through the **µP Data Buffers** <33>, to the **General Purpose Static RAM** <33> by performing a "walking one's" test on one static RAM memory location.

Description

1. Perform a "walking one's" test on **General Purpose Static RAM** <33> address 00000*hex*. Terminate test if any verify operation fails.

   - Write the pattern 0000*hex* to address 00000*hex* Read the same address and verify that it was 0000*hex*. Continue this write/read/verify sequence with the patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, 8000*hex*, and 0000*hex*

Error Index D1311

The pattern read from **General Purpose Static RAM** <33> address 00000*hex* was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to RAM devices is shown in the following table:

D0-D7	D8-D15
U611	U601

Routine Name Address/Data

Overview This test verifies the address lines and data integrity of the **General Purpose Static RAM** <33> by performing a RAM test on all static RAM memory locations.

Description

1. Verify **VRS Min Plane DRAM** and **VRS Max Plane DRAM** <36> address range 7C000*hex* to 7CFFF*hex*. Terminate test if any verify operation fails.
   - Fill address range 7C000*hex* to 7CFFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 7C000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 7C000*hex*. Increment the address and continue this read/verify/write sequence until address 7CFFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 7C000*hex*, for CCCC*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
2. Save **General Purpose Static RAM** <33> data in address range 00000*hex* to 00FFF*hex* by copying it to VRS DRAM address range 7C000*hex* to 7CFFF*hex*.
3. Verify **General Purpose Static RAM** <33> address range 00000*hex* to 00FFF*hex*.
   - Perform the same procedures as in step 1 for the address range 00000*hex* to 00FFF*hex* using patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, 5555*hex*, and AAAA*hex*.
   - Make one last read/verify pass for the pattern AAAA*hex* starting at address 00000*hex*.
4. Copy the VRS DRAM data back from address range 7C000*hex* to 7CFFF*hex* to **General Purpose Static RAM** <33> address range 00000*hex* to 00FFF*hex*.

Error Index D1321 The pattern read from the displayed memory location in either the VRS DRAM or the **General Purpose Static RAM** <33> was not the pattern written. If the displayed address is between 7C000*hex* and 7CFFF*hex*, then the failure occurred in the VRS DRAM testing stage. Otherwise, the failure occurred while testing the static RAM. The correspondence of the address and data bits to RAM devices is shown in the following table:

General Purpose Static RAM Address	D0-D7	D8-D15
00XXX*hex*	U611	U601

X - don't care

See Also

See the Video Gen Trace Plane Address/Data (D252X) Error Index section for the correspondence of the address and data bits to VRS DRAM devices if the failure address displayed was between 7C000*hex* and 7CFFF*hex*.

Routine Name

Timer 0

Overview

This test verifies **CPU & Ready Logic** U524 <33> Programmable Timers timer 0's counting accuracy and its interrupt. The counting accuracy of the timer 0 is verified by counting the system clock for a short duration. If the timer 0 counts accurately, a timer 0 interrupt is generated and verified.

Description

1. Enable the timer 0 to count the system clock by setting its gate signal high.
   - Set **Diagnostic Control/Status Latch** TMR IN 0 <37> high by writing 83*hex* to U600 <37>.
2. Stop the timer 0 by writing 4000*hex* to its Timer Mode/Control Register.
3. Program the counter to count up by writing 0000 *hex*, FFFF*hex*, and C000*hex* to timer 0 Count Register, Max Count Value A Register, and Timer 0 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
6. Read and verify the timer 0 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 0 is counting accurately), enable the timer 0 to count the system clock again by setting its gate signal high. Else terminate testing.
   - Set **Diagnostic Control/Status Latch** TMR IN 0 <37> high by writing 83*hex* to U600 <37>.
8. Enable timer 0 interrupt by writing 0004*hex* to the Timer 0 Interrupt Control Register.
9. Program the counter to count up to 100 *hex* and generate an interrupt when the count reaches 100*hex*.
   - Write 0100*hex*, 0000*hex*, E000*hex* to timer 0 Max Count Value A Register, Count Register, and Timer 0 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
12. Disable timer 0 interrupt by writing 000C*hex* to the Timer 0 Interrupt Control Register.
13. Verify that the timer 0 interrupt did occur.

Error Index D1411

The timer 0 count was not within the expected tolerance.

Error Index D1412

Timer 0 interrupt did not occur.

Routine Name

Timer 1

Overview

This test verifies **CPU & Ready Logic** U524 <33> Programmable Timers timer 1's counting accuracy and its interrupt. The counting accuracy of the timer 1 is verified by counting the system clock for a short duration. If the timer 1 counts accurately, a timer 1 interrupt is generated and verified.

Description

1. Enable the timer 1 to count the system clock by setting its gate signal high.
   - Set **Diagnostic Control/Status Latch** TMR IN 1 <37> high by writing A3*hex* to U600 <37>.
2. Stop the timer 1 by writing 4000*hex* to its Timer Mode/Control Register.
3. Program the counter to count up by writing 0000 *hex*, FFFF*hex*, and C000*hex* to timer 1 Count Register, Max Count Value A Register, and Timer 1 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
6. Read and verify the timer 1 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 1 is counting accurately), enable the timer 1 to count the system clock again by setting its gate signal high. Else terminate testing.
   - Set **Diagnostic Control/Status Latch** TMR IN 1 <37> high by writing A3*hex* to U600 <37>.
8. Enable timer 1 interrupt by writing 0004*hex* to the Timer 1 Interrupt Control Register.
9. Program the counter to count up to 100 *hex* and generate an interrupt when the count reaches 100*hex*.
   - Write 0100*hex*, 0000*hex*, E000*hex* to timer 1 Max Count Value A Register, Count Register, and Timer 1 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
12. Disable timer 1 interrupt by writing 000C*hex* to the Timer 1 Interrupt Control Register.
13. Verify that the timer 1 interrupt did occur.

Error Index D1421

The timer 1 count was not within the expected tolerance.

Error Index D1422

Timer 1 interrupt did not occur.

Routine Name

Timer 2

Overview

This test verifies CPU & Ready Logic U524 <33> Programmable Timers timer 2's counting accuracy and its interrupt. The counting accuracy of the timer 2 is verified by counting the system clock for a short duration. If the timer 2 counts accurately, a timer 2 interrupt is generated and verified.

Description

1. Stop the timer 2 by writing 4000*hex* to its Timer Mode/Control Register.
2. Program the counter to count up by writing 0000 *hex*, FFFF*hex*, and C000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
3. Perform a software delay.
4. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
5. Read and verify the timer 2 count against the known value for the expected tolerance.
6. If the count was within the expected tolerance (i.e., the timer 2 is counting accurately), enable the timer 2 interrupt by writing 0004*hex* to the Timer 2 Interrupt Control Register. Else terminate testing.
7. Program the counter to count up to 100 *hex* and generate an interrupt when the count reaches 100*hex*.
   - Write 0100*hex*, 0000*hex*, E000*hex* to timer 2 Max Count Value A Register, Count Register, and Timer 2 Mode/Control Register, respectively.
8. Perform a software delay.
9. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
10. Disable timer 2 interrupt by writing 000C*hex* to the Timer 2 Interrupt Control Register.
11. Verify that the timer 2 interrupt did occur.

Error Index D1431

The timer 2 count was not within the expected tolerance.

Error Index D1432

Timer 2 interrupt did not occur.

Routine Name

DMA 0

Overview

This test verifies **CPU & Ready Logic** U524 <33> Programmable DMA Unit's channel 0 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 0 interrupt by writing 0004*hex* to DMA 0 Interrupt Control Register.
2. Initialize the source and destination memory locations in **General Purpose Static RAM** <33>.
   - Write patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex* and 5555*hex* to source memory locations starting at address 210*hex*.
   - Write patterns 5555*hex*, 3333*hex*, 0F0F*hex*, 00FF*hex* and AAAA*hex* to destination memory locations starting at address 200*hex*.
3. Program the DMA channel 0 to transfer the 5 source patterns to the destination memory locations starting at 200*hex* and generate an interrupt after completion of the transfer.
   - Write B725*hex*, 0005*hex*, 0000*hex*, 0200*hex*, 0000*hex*, 0210*hex* and B727*hex* to DMA 0 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source Pointer Lower, and DMA 0 Channel Control Word registers, respectively. The last write initiates the DMA transfer.
4. Disable DMA 0 interrupt by writing 000C*hex* to DMA 0 Interrupt Control Register.
5. Read and verify the destination memory locations.

Error Index D1511

One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Routine Name DMA 1

Overview This test verifies **CPU & Ready Logic** U524 <33> Programmable DMA Unit's channel 1 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 1 interrupt by writing 0004*hex* to DMA 1 Interrupt Control Register.
2. Initialize the source and destination memory locations in **General Purpose Static RAM** <33>.
   - Write patterns AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex* and 5555*hex* to source memory locations starting at address 210*hex*.
   - Write patterns 5555*hex*, 3333*hex*, 0F0F*hex*, 00FF*hex* and AAAA*hex* to destination memory locations starting at address 200*hex*.
3. Program the DMA channel 1 to transfer the 5 source patterns to the destination memory locations starting at 200*hex* and generate an interrupt after completion of the transfer.
   - Write B725*hex*, 0005*hex*, 0000*hex*, 0200*hex*, 0000*hex*, 0210*hex* and B727*hex* to DMA 1 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source Pointer Lower, and DMA 1 Channel Control Word registers, respectively. The last write initiates the DMA transfer.
4. Disable DMA 1 interrupt by writing 000C*hex* to DMA 1 Interrupt Control Register.
5. Read and verify the destination memory locations.

Error Index D1521 One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Routine Name

Command Port

Overview

This test verifies the data path and operation of the **Executive Processor Parallel Interface Port** <37> and **Interface Data Buffers** <37> by utilizing **Diagnostic Loopback Control** <37> to loop "walking one" test patterns from the output of the **Executive Processor Parallel Interface Port** <37> back to its input. Even though the **Diagnostic Loopback Control** <37> is utilized, the communication handshake mechanics are very similar to those in normal operation (i.e., as if a message was actually being sent to and from the Executive processor).

Description

1. Enable diagnostic loopback capability.
   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):L <37> (via MUART U523-27 P24 <33>).
2. Write the test pattern 0000*hex* to the **Executive Processor Parallel Interface Port** latches U731 & U733 <37>, through the **Interface Data Buffers** <37>.
3. Loop the test pattern from the **Executive Processor Parallel Interface Port** <37> outputs back to its inputs.
   - Toggle **Diagnostic Loopback Control** U623A-1 BMLCS(L). As BMLCS(L) goes low, the **Executive Processor Parallel Interface Port** outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches.
4. Read the test pattern from the **Executive Processor Parallel Interface Port** <37> through the **Interface Data Buffers** <37>.
5. Repeat steps 2-4 with the test patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*
6. Verify that each pattern read back was the same as the pattern written.
7. Disable diagnostic loopback capability.
   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):H <37> (via MUART U523-27 P24 <33>).

Error Index D1611

The pattern read from the **Executive Processor Parallel Interface Port** <37> output port did not match the pattern written to its input port. The display shows the first pattern that did not match.

Routine Name

DMA Access

Overview

This test verifies the operation of the **CPU & Ready Logic 80186 Programmable DMA Unit** <33> in reading and writing the **Executive Processor Parallel Interface Port** <37> by utilizing **Diagnostic Loopback Control** <37> to loop a test pattern from the output of the **Executive Processor Parallel Interface Port** <37> back to its input. Even though the **Diagnostic Loopback Control** <37> is utilized, the communication handshake mechanics are identical to those in normal operation (i.e., as if a message was actually being sent to and from the Executive processor).

Description

1. Enable diagnostic loopback capability.
   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):L <37> (via MUART U523-27 P24 <33>).
2. Initialize the **Executive Processor Parallel Interface Port** <37> by reading port latches U731 & U733 through **Interface Data Buffers** <37>.
3. Write test pattern 5555*hex* into **General Purpose Static RAM** <33> memory location 00210*hex* (source location) and test pattern AAAA*hex* into memory location 00204*hex* (destination location).
4. Transfer pattern from **General Purpose Static RAM** <33> to the **Executive Processor Parallel Interface Port** <37>.
   - Program the CPU DMA 1 to read the value in **General Purpose Static RAM** <33> memory location 00210*hex* and write it to the **Executive Processor Parallel Interface Port** <37>.
5. Transfer pattern from **Executive Processor Parallel Interface Port** <37> to the **General Purpose Static RAM** <33>.
   - Initialize CPU DMA 0 to read one value from the **Executive Processor Parallel Interface Port** <37> and write it to **General Purpose Static RAM** memory location 00204*hex*.
   - Toggle **Diagnostic Loopback Control** U623A-1 BMLCS(L). As BMLCS(L) goes low, the **Executive Processor Parallel Interface Port** outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches. At this point, U733-5 DRQ0(H) becomes active (high) and initiates a DMA 0 transfer.
6. Read and verify that **General Purpose Static RAM** <33> memory location 00204*hex* contains the pattern 5555*hex*.
7. Disable diagnostic loopback capability.
   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):H <37> (via MUART U523-27 P24 <33>).

Error Index D1621

The pattern read from **General Purpose Static RAM** <33> memory location 00204*hex* (using a DMA 0 transfer from **Executive Processor Parallel Interface Port** <37>), was not the pattern written (5555*hex*) to the **Executive Processor Parallel Interface Port**

<37> (using a DMA 1 transfer from **General Purpose Static RAM** <33> memory location 00210*hex*).

Routine Name

Wavefrm Port (Waveform Port)

Overview

This test verifies the data path and operation of the **Executive Processor Parallel Interface Port** <37>, **Waveform Attribute Encoder** <37>, and **Waveform Data Buffers** <37> by utilizing **Diagnostic Loopback Control** <37> to loop test patterns from the output of the **Executive Processor Parallel Interface Port** <37> back to its input. Even though the **Diagnostic Loopback Control** <37> is utilized, the waveform handshake mechanics are very similar to those in normal operation (i.e., as if a waveform was actually being sent to the Display subsystem).

Description

1. Enable diagnostic loopback capability.

   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):L <37> (via MUART U523-27 P24 <33>).

2. Initialize waveform attributes by writing 00*hex* to **MUART** U523 (pins 32-37) P10-P17. This sets **Waveform Data Buffers** U622 CD9(H):L, CD13(H):L, CD14(H):L, and CD15(H):L <37>, and **Waveform Attribute Encoder** U533D-13 NULL_FORCE(H):L <37>, U635-1 SCALE(L):L, and U632-3 OFFSET(H):L.

3. Write a test pattern (see Error Index section below) to the **Executive Processor Parallel Interface Port** latches U731 & U733 <37>, through the Interface Data Buffers <37>.

4. Loop the test pattern from the **Executive Processor Parallel Interface Port** <37> outputs back to its inputs.

   - Toggle **Diagnostic Loopback Control** U623A-1 BMLCS(L). As BMLCS(L) goes low, the **Executive Processor Parallel Interface Port** outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches.

5. Read the test pattern from the **Executive Processor Parallel Interface Port** <37> through the **Waveform Attribute Encoder** <37> and **Waveform Data Buffers** <37>.

6. Repeat steps 3-5 for the remaining test patterns.

7. Verify that each pattern read back was the expected pattern (see Error Index section below).

8. Disable diagnostic loopback capability.

   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):H <37> (via MUART U523-27 P24 <33>).

Error Index D1631

The pattern read from the **Executive Processor Parallel Interface Port** <37>, through the **Waveform Attribute Encoder** <37> and **Waveform Data Buffers** <37>, did not match the expected pattern. The display shows the first pattern that did not match. The patterns written to the **Executive Processor Parallel Interface Port** <37> and those patterns expected to be read back are shown in the following table:

Parallel Interface Port Pattern	Expected Waveform Data Buffer Pattern
AAAA*hex*	0055*hex*
CCCC*hex*	0099*hex*
F0F0*hex*	00E1*hex*
FF00*hex*	00FE*hex*
5555*hex*	01AA*hex*
7FFF*hex* (Overrange)	09FF*hex*
8000*hex* (Null)	1000*hex*
8001*hex* (Underrange)	0400*hex*

Routine Name

Attributes

Overview

This test verifies the waveform attribute control bits (color, age, and scale) from **MUART** (P10-P13) <33> through the **Waveform Data Buffers** <37> by performing a "walking one's" test on these lines. Waveform data test patterns are looped back through **Executive Processor Parallel Interface Port** <37>, **Waveform Attribute Encoder** <37>, and **Waveform Data Buffers** <37> by utilizing **Diagnostic Loopback Control** <37>.

Description

1. Enable diagnostic loopback capability.
   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):L <37> (via **MUART** U523-27 P24 <33>).
2. Write a waveform data test pattern of 5555*hex* to the **Executive Processor Parallel Interface Port** latches U731 & U733 <37>, through the **Interface Data Buffers** <37>.
3. Write a waveform attribute test pattern (see Error Index section below) to **MUART** U523 (pins 32-37) P10-P17 <33>.
4. Loop the waveform data test pattern from the **Executive Processor Parallel Interface Port** <37> outputs back to its inputs.
   - Toggle **Diagnostic Loopback Control** U623A-1 BMLCS(L). As BMLCS(L) goes low, the **Executive Processor Parallel Interface Port** outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches.
5. Read the waveform data and attribute test pattern from the **Executive Processor Parallel Interface Port** <37> through the **Waveform Attribute Encoder** <37> and **Waveform Data Buffers** <37>.
6. Repeat steps 2-5 for the remaining waveform attribute test patterns.
7. Verify that each pattern read back was the expected pattern (see Error Index section below).
8. Disable diagnostic loopback capability.
   - Set **Diagnostic Loopback Control** LOOPBACKEN(L):H <37> (via **MUART** U523-27 P24 <33>).

Error Index D1641

The pattern read from the **Executive Processor Parallel Interface Port** <37>, through the **Waveform Attribute Encoder** <37> and **Waveform Data Buffers** <37>, did not match the expected pattern. The display shows the first pattern that did not match. The waveform data patterns written to the **Executive Processor Parallel Interface Port** <37>, the waveform attribute patterns written to **MUART** P10-P17 <33>, and those patterns expected to be read back are shown in the following table:

Parallel Interface Port Waveform Data Pattern	MUART Waveform Attribute Pattern	Expected Waveform Data Buffer Pattern

Dsy Control	Exec Intrfce	Attributes (D164X)
5555*hex*	00*hex*	01AA*hex*
5555*hex*	01*hex*	21AA*hex*
5555*hex*	02*hex*	41AA*hex*
5555*hex*	04*hex*	81AA*hex*
5555*hex*	08*hex*	03AA*hex*
5555*hex*	40*hex*	00D5*hex*

Routine Name

Trace CAS

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **VRS Plane DRAM Control** U223C-8 VRSCAS(L) <36> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-20 TMR IN 0 <33>, where the number of transitions over a known period of time are counted.

VRSCAS(L) is a composite of other signals which are inherently verified as VRSCAS(L) is verified. The signals covered by VRSCAS(L) are **DRAM Control Generation** U333C-8 CAS(H) <34>, Bit Plane Address Mux U325B-8 VA1(L) <34>, **µP Address Latch** U526-3 BHE(L) <33>, **System Timing Generator** U323D-8 ACCLK(H) <34>, and **Video Memory Interface Synchronizer** U430A-3 VGATE(L) <34>.

Description

1. Enable **VRS Plane DRAM Control** U223C-8 VRSCAS(L) <36> to be counted.
   - Write 8B*hex* to **Diagnostic Control/Status Latch** U600 <37> to select VRSCAS(L) at the multiplexor inputs of **Diagnostic Timing Mux** U433 <37>. This also takes divide-by-four counter U516 out of reset.
2. Count the transitions of VRSCAS(L).
   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 0 counter to start counting.
   - Perform a software delay.
   - Disable the timer 0 counter from counting.
3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2111

The value read from the **CPU & Ready Logic** <33> 80186 timer 0 counter was not within the expected tolerance limits (i.e. the measured frequency of VRSCAS(L) was out of tolerance).

Routine Name

BSRLOAD

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **Video Shifter Control** U425B-8 BSRLOAD(H) <34> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-20 TMR IN 0 <33>, where the number of transitions over a known period of time are counted.

BSRLOAD(H) is a composite of other signals which are inherently verified as BSRLOAD(H) is verified. The signals covered by BSRLOAD(H) are **System Timing Generator** U434B-4 PCLK(H), U431-11 CCLK(H), U431-12 FCLK(H), U431-13 LCLK(H), U431-14 MCLK(H) <34> and **CRT Controller & Select** U515-18 DISPLAY ENABLE(H) <34>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.
   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Enable **Video Shifter Control** U425B-8 BSRLOAD(H) <34> to be counted.
   - Write $9B_{hex}$ to **Diagnostic Control/Status Latch** U600 <37> to select BSRLOAD(H) at the multiplexor inputs of **Diagnostic Timing Mux** U433 <37>. This also takes divide-by-four counter U516 out of reset.
3. Perform a software delay of approximately 1 millisecond to wait for the CRT CONTROLLER U515 to finish the vertical retrace period (BSRLOAD(H) is only active while not in vertical retrace).
4. Count the transitions of BSRLOAD(H).
   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 0 counter to start counting.
   - Perform a software delay.
   - Disable the timer 0 counter from counting.
5. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2121

The value read from the **CPU & Ready Logic** <33> 80186 timer 0 counter was not within the expected tolerance limits (i.e., the measured frequency of BSRLOAD(H) was out of tolerance).

Error Index D2122

A vertical sync condition in **CRT Controller & Select** U515 <34> could not be detected.

Routine Name

Bit 1 CAS

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **Bit Plane 1 & 2 DRAM Control** U320B-6 BIT1CAS(L) <35> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-20 TMR IN 0 <33>, where the number of transitions over a known period of time are counted.

BIT1CAS(L) is a composite of other signals which are inherently verified as BIT1CAS(L) is verified. The signals covered by BIT1CAS(L) are **DRAM Control Generation** U333C-8 CAS(H) <34>, **Bit Plane Address Mux** U325B-8 VA1(L) <34>, **µP Address Latch** U526-3 BHE(L) <33>, **System Timing Generator** U323D-8 ACCLK(H) <34>, and **Video Memory Interface Synchronizer** U430B-6 B1GATE(L) <34>.

Description

1. Enable **Bit Plane 1 & 2 DRAM Control** U320B-6 BIT1CAS(L) <35> to be counted.

   - Write AB*hex* to **Diagnostic Control/Status Latch** U600 <37> to select BIT1CAS(L) at the multiplexor inputs of **Diagnostic Timing Mux** U433 <37>. This also takes divide-by-four counter U516 out of reset.

2. Count the transitions of BIT1CAS(L).

   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 0 counter to start counting.
   - Perform a software delay.
   - Disable the timer 0 counter from counting.

3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2131

The value read from the **CPU & Ready Logic** <33> 80186 timer 0 counter was not within the expected tolerance limits (i.e., the measured frequency of BIT1CAS(L) was out of tolerance).

Routine Name

ACCLK

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **System Timing Generator** U323D-8 ACCLK(H) <34> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-20 TMR IN 0 <33>, where the number of transitions over a known period of time are counted.

ACCLK(H) is a composite of other signals which are inherently verified as ACCLK(H) is verified. The signals covered by ACCLK(H) are **System Timing Generator** U431-11 CCLK(H) and U431-14 MCLK(H).

Description

1. Enable **System Timing Generator** U323D-8 ACCLK(H) <34> to be counted.

   - Write BB*hex* to **Diagnostic Control/Status Latch** U600 <37> to select ACCLK(H) at the multiplexor inputs of **Diagnostic Timing Mux** U433 <37>. This also takes divide-by-four counter U516 out of reset.

2. Count the transitions of ACCLK(H).

   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 0 counter to start counting.
   - Perform a software delay.
   - Disable the timer 0 counter from counting.

3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2141

The value read from the **CPU & Ready Logic** <33> 80186 timer 0 counter was not within the expected tolerance limits (i.e., the measured frequency of ACCLK(H) was out of tolerance).

Routine Name

Trace RAS

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **System Timing Generator** U222B-6 VRSRAS(L) <36> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-20 TMR IN 0 <33>, where the number of transitions over a known period of time are counted.

VRSRAS(L) is a composite of other signals which are inherently verified as VRSRAS(L) is verified. The signals covered by VRSRAS(L) are **System Timing Generator** U332C-8 RASE(H) <36> and **DRAM Control Generation** U333A-3 RAS(H) <36>.

Description

1. Enable **VRS Plane DRAM Control** U222B-6 VRSRAS(L) <36> to be counted.

   - Write CB*hex* to **Diagnostic Control/Status Latch** U600 <37> to select VRSRAS(L) at the multiplexor inputs of **Diagnostic Timing Mux** U433 <37>. This also takes divide-by-four counter U516 out of reset.

2. Count the transitions of VRSRAS(L).

   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 0 counter to start counting.
   - Perform a software delay.
   - Disable the timer 0 counter from counting.

3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2151

The value read from the **CPU & Ready Logic** <33> 80186 timer 0 counter was not within the expected tolerance limits (i.e., the measured frequency of VRSRAS(L) was out of tolerance).

Routine Name

CRTC R/W

Overview

This test verifies **CRT Controller & Select** U631A-5 CRT R(H)/W(L) <34> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-21 TMR IN 1 <33>, where the number of transitions are counted. CRT R(H)/W(L) goes high for every read of **CRT Controller & Select** U515 <34> and goes low for every write.

CRT R(H)/W(L) is a composite of other signals which are inherently verified as CRT R(H)/W(L) is verified. The signals covered by CRT R(H)/W(L) are **CPU & Ready Logic** U524-27 CRTCS(L) <33>, U524-56 MPUCLK, and U533A-3 R(H)/W(L).

Description

1. Enable **CRT Controller & Select** U631A-5 CRT R(H)/W(L) <34> to be counted.

   - Write 93*hex* to **Diagnostic Control/Status Latch** U600 <37> to select CRT R(H)/W(L) at the multiplexor inputs of **Diagnostic Timing Mux** U530 <37>.

2. Clear **CRT Controller & Select** U631A-5 CRT R(H)/W(L) <34>.

   - Write to an internal register of U515 to set CRT R(H)/W(L) low.

3. Enable **CPU & Ready Logic** 80186 <33> timer 1 counter.

   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 1 counter to start counting from zero.

4. Generate and count occurrences of CRT R(H)/W(L).

   - Execute four read and write operations to internal registers of **CRT Controller & Select** U515 <34>. Each read operation sets U631A-5 CRT R(H)/W(L) high and the following write operation sets it back low, thereby causing the timer 1 counter to increment.

5. Disable the timer 1 counter from counting.

6. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2161

The value read from the **CPU & Ready Logic** <33> 80186 timer 1 counter did not equal the four read/write accesses made to **CRT Controller & Select** U515 <34> (i.e., CRT R(H)/W(L) did not toggle four times).

Routine Name

B2HE

Overview

This test verifies **Bit Plane 1 & 2 DRAM Control** U323F-12 B2HE(L) <35> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-21 TMR IN 1 <33>, where the number of transitions are counted. B2HE(L) toggles every time an access occurs to a high byte in **Bit Plane 2 DRAM** <35>.

B2HE(L) is a composite of other signals which are inherently verified as B2HE(L) is verified. The signals covered by B2HE(L) are **µP Address Latch** U526-3 BHE(L) <33> and **Video Memory Interface Synchronizer** U430D-11 B2GATE(L) <34>.

Description

1. Enable **Bit Plane 1 & 2 DRAM Control** U323F-12 B2HE(L) <35> to be counted.
   - Write A3*hex* to **Diagnostic Control/Status Latch** U600 <37> to select B2HE(L) at the multiplexor inputs of **Diagnostic Timing Mux** U530 <37>.
2. Enable **CPU & Ready Logic** 80186 <33> timer 1 counter.
   - Write to an internal register of the CPU & Ready Logic 80186 to enable the internal timer 1 counter to start counting from zero.
3. Generate and count occurrences of B2HE(L).
   - Execute four read operations to **Bit Plane 2 DRAM** <35> location 60000*hex* (both low and high byte). Each read operation should toggle U323F-12 B2HE(L) and cause the timer 1 counter to increment.
4. Disable the timer 1 counter from counting.
5. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2171

The value read from the **CPU & Ready Logic** <33> 80186 timer 1 counter did not equal the four read accesses made to **Bit Plane 2 DRAM** <35> location 60000*hex* (i.e., B2HE(L) did not toggle four times).

Routine Name

B1LE

Overview

This test verifies **Bit Plane 1 & 2 DRAM Control** U323B-4 B1LE(L) <35> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-21 TMR IN 1 <33>, where the number of transitions are counted. B1LE(L) toggles every time an access occurs to a low byte in **Bit Plane 1 DRAM** <35>.

B1LE(L) is a composite of other signals which are inherently verified as B1LE(L) is verified. The signals covered by B1LE(L) are **µP Address Latch** U614-19 MA0 <33> and **Video Memory Interface Synchronizer** U430B-6 B2GATE(L) <34>.

Description

1. Enable **Bit Plane 1 & 2 DRAM Control** U323B-4 B1LE(L) <35> to be counted.
   - Write B3*hex* to **Diagnostic Control/Status Latch** U600 <37> to select B1LE(L) at the multiplexor inputs of **Diagnostic Timing Mux** U530 <37>.
2. Enable **CPU & Ready Logic** 80186 <33> timer 1 counter.
   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 1 counter to start counting from zero.
3. Generate and count occurrences of B1LE(L).
   - Execute four read operations to **Bit Plane 1 DRAM** <35> location 50000*hex* (both low and high byte). Each read operation should toggle U323B-4 B1LE(L) and cause the timer 1 counter to increment.
4. Disable the timer 1 counter from counting.
5. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2181

The value read from the **CPU & Ready Logic** <33> 80186 timer 1 counter did not equal the four read accesses made to **Bit Plane 1 DRAM** <35> location 50000*hex* (i.e., B1LE(L) did not toggle four times).

Routine Name

V0LE

Overview

This test verifies **VRS Plane DRAM Control** U334D-8 V0LE(L) <36> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-21 TMR IN 1 <33>, where the number of transitions are counted. V0LE(L) toggles every time an access occurs to a low byte in **VRS Min Plane DRAM** <36>.

V0LE(L) is a composite of other signals which are inherently verified as V0LE(L) is verified. The signals covered by V0LE(L) are **µP Address Latch** U614-19 MA0 <33>, **Video Memory Interface Synchronizer** U430A-3 VGATE(L) <34>, and **Bit Plane Address Mux** U325B-9 VA1(H) <34>.

Description

1. Enable **VRS Plane DRAM Control** U334D-8 V0LE(L) <36> to be counted.

   - Write C3*hex* to **Diagnostic Control/Status Latch** U600 <37> to select V0LE(L) at the multiplexor inputs of **Diagnostic Timing Mux** U530 <37>.

2. Enable **CPU & Ready Logic** 80186 <33> timer 1 counter.

   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 1 counter to start counting from zero.

3. Generate and count occurrences of V0LE(L).

   - Execute four read operations to **VRS Min Plane DRAM** <36> location 70000*hex* (both low and high byte). Each read operation should toggle U334D-8 V0LE(L) and cause the timer 1 counter to increment.

4. Disable the timer 1 counter from counting.

5. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2191

The value read from the **CPU & Ready Logic** <33> 80186 timer 1 counter did not equal the four read accesses made to **VRS Min Plane DRAM** <36> location 70000*hex* (i.e., V0LE(L) did not toggle four times).

Routine Name

Crastersync

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **CRT Controller & Select** U515-39 CRASTERSYNC(H) <34> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-21 TMR IN 1 <33>, where the number of transitions over a known period of time are counted.

Description

1. Enable **CRT Controller & Select** U515-39 CRASTERSYNC(H) <34> to be counted.
   - Write D3*hex* to **Diagnostic Control/Status Latch** U600 <37> to select CRASTERSYNC(H) at the multiplexor inputs of **Diagnostic Timing Mux** U530 <37>.
2. Count the transitions of CRASTERSYNC(H).
   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 1 counter to start counting.
   - Perform a software delay.
   - Disable the timer 1 counter from counting.
3. Read the timer 1 counter value and verify it is within expected limits.

Error Index D21A1

The value read from the **CPU & Ready Logic** <33> 80186 timer 1 counter was not within the expected tolerance limits (i.e., the measured frequency of CRASTERSYNC(H) was out of tolerance).

Routine Name Cfieldsync

Overview This test verifies the relative frequency (with respect to the 80186 CPU) of **CRT Controller & Select** U515-40 CFIELDSYNC(H) <34> by routing it through **Diagnostic Timing Mux** <37> and into **CPU & Ready Logic** U524-21 TMR IN 1 <33>, where the number of transitions over a known period of time are counted.

Description

1. Enable **CRT Controller & Select** U515-40 CFIELDSYNC(H) <34> to be counted.
   - Write E3*hex* to **Diagnostic Control/Status Latch** U600 <37> to select CFIELDSYNC(H) at the multiplexor inputs of **Diagnostic Timing Mux** U530 <37>.
2. Count the transitions of CFIELDSYNC(H).
   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 1 counter to start counting.
   - Perform a software delay.
   - Disable the timer 1 counter from counting.
3. Read the timer 1 counter value and verify it is within expected limits.

Error Index D21B1 The value read from the **CPU & Ready Logic** <33> 80186 timer 1 counter was not within the expected tolerance limits (i.e., the measured frequency of CFIELDSYNC(H) was out of tolerance).

Routine Name

Dispen

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **CRT Controller & Select** U515-18 DISPLAY ENABLE(H) <34> by routing it through **Diagnostic Timing Mux** <37> and into CPU & Ready Logic U524-21 TMR IN 1 <33>, where the number of transitions over a known period of time are counted.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.
   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Enable **CRT Controller & Select** U515-18 DISPLAY ENABLE(H) <34> to be counted.
   - Write F3*hex* to **Diagnostic Control/Status Latch** U600 <37> to select DISPLAY ENABLE(H) at the multiplexor inputs of **Diagnostic Timing Mux** U530 <37>.
3. Perform a software delay of approximately 1 millisecond to wait for the CRT CONTROLLER U515 to finish the vertical retrace period (DISPLAY ENABLE(H) is only active while not in vertical retrace).
4. Count the transitions of DISPLAY ENABLE(H).
   - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 1 counter to start counting.
   - Perform a software delay.
   - Disable the timer 1 counter from counting.
5. Read the timer 1 counter value and verify it is within expected limits.

Error Index D21C1

The value read from the **CPU & Ready Logic** <33> 80186 timer 1 counter was not within the expected tolerance limits (i.e., the measured frequency of DISPLAY ENABLE(H) was out of tolerance).

Error Index D21C2

A vertical sync condition in **CRT Controller & Select** U515 <34> could not be detected.

Routine Name MPU Address

Overview

This test verifies the address generation from the **CPU and Ready Logic** U524 CPU <33> through the **Bit Plane Address Mux** <34> and **VRS Plane Address Mux** <34> circuits by performing a "walking one's" test on the CPU generated address lines. The addresses from the CPU are used to access both the VRS planes and the bit planes. Addresses are captured and read back through the **Diagnostic Bit & VRS Plane Address Feedback Latches** <37>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.

   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.

2. Enable **Diagnostic CRTC/MPU Address Trigger Control** <37> to latch bit plane and VRS plane addresses.

   - Set **Diagnostic Control/Status Latch** U600-16 <37> high.

3. Reset **Diagnostic CRTC/MPU Address Trigger Control** <37>.

   - Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 <37>. This drives U402A-4 low, which in turn sets U402A-5 high and allows U402B-9 to be driven high.

4. Read from an address in bit plane memory (see Error Index sections below).

   - Read from a bit plane memory location. The address generated by the CPU passes through the **Bit Plane Address Mux** <34> and **VRS Plane Address Mux** <34> and is intercepted by the **Diagnostic Bit & VRS** <37>. The address is latched in the normal row/column format. When an access is made to bit plane memory, **Diagnostic CRTC/MPU Address Trigger Control** U517A-2 GATE(H) <37> is driven high. This enables the row and column strobe signals at U401A-2 & U401D-12 <37>. U401A-3 Q6(H):L is the row strobe and U401D-11 CAS(L):L is the column strobe.

5. Read and verify latched bit plane address (see first Error Index section below).

   - Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U500 & U510 <37>, ignoring U510-8 RA0 and U510-9 RA1 (their values are unpredictable), and verify against the expected address.

6. Read and verify latched VRS plane address (see second Error Index section below).

   - Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 <37>, ignoring U511-8 RA2 and U511-9 RA3 (their values are unpredictable), and verify against the expected address.

7. Repeat steps 3-6 until all patterns have been tested or a verify operation fails.

Error Index D2211

The bit plane latched address in **Diagnostic Bit & VRS Plane Address Feedback Latches** U500 & U510 did not match the expected value. The display shows the first bit plane address pattern which did not match. The CPU addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CPU Address	Expected Bit Plane Address
50000*hex*	0000*hex*
50001*hex*	0000*hex*
50002*hex*	0000*hex*
50004*hex*	0100*hex*
50008*hex*	0200*hex*
50010*hex*	0400*hex*
50020*hex*	0800*hex*
50040*hex*	1000*hex*
50080*hex*	2000*hex*
50100*hex*	4000*hex*
50200*hex*	8000*hex*
50400*hex*	0001*hex*
50800*hex*	0002*hex*
51000*hex*	0004*hex*
52000*hex*	0008*hex*
54000*hex*	0010*hex*
58000*hex*	0020*hex*

Error Index D2212 The VRS plane latched address in **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 did not match the expected value. The display shows the first VRS plane address pattern which did not match. The CPU addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CPU Address	Expected VRS Plane Address
50000*hex*	0000*hex*
50001*hex*	0000*hex*
50002*hex*	0000*hex*
50004*hex*	1000*hex*
50008*hex*	2000*hex*
50010*hex*	4000*hex*
50020*hex*	8000*hex*
50040*hex*	0001*hex*
50080*hex*	0002*hex*
50100*hex*	0004*hex*
50200*hex*	0008*hex*
50400*hex*	0010*hex*
50800*hex*	0020*hex*
51000*hex*	0100*hex*
52000*hex*	0200*hex*
54000*hex*	0400*hex*
58000*hex*	0800*hex*

Error Index D2213 The vertical sync pulse from step 1 could not be detected within the timeout period.

Routine Name

CRTC Address

Overview

This test verifies the address generation from the **CRT Controller & Select** U515 CRT CONTROLLER <34> through the **Bit Plane Address Mapping** <34>, **Bit Plane Address Mux** <34>, and **VRS Plane Address Mux** <34> circuits by performing a "walking one's" test on the CRT CONTROLLER generated address lines. Addresses are captured and read back through the **Diagnostic Bit & VRS Plane Address Feedback Latches** <37>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.

   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.

2. Enable **Diagnostic CRTC/MPU Address Trigger Control** <37> to latch bit plane and VRS plane addresses.

   - Set **Diagnostic Control/Status Latch** U600-16 <37> low.

3. Set CRT CONTROLLER output address (see Error Index sections below).

   - Wait for a vertical sync pulse as in step 1. After it is detected (i.e., during vertical retrace), set the address to be generated by writing the address into U515 CRT CONTROLLER's "top of page" register. This will be the first address generated after the CRT CONTROLLER finishes the vertical retrace period.

4. Reset **Diagnostic CRTC/MPU Address Trigger Control** <37>.

   - Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 <37>. This drives U402A-4 low, which in turn sets U402A-5 high and allows U402B-9 to be driven high.

5. Wait for bit plane and VRS plane addresses to be latched into **Diagnostic Bit & VRS Plane Address Feedback Latches** <37>.

   - Wait for the next vertical sync pulse (as in step 1) to ensure that the bit plane and VRS plane addresses have been latched into U501, U511, U500, and U510. The bit plane address generated from the CRT CONTROLLER passes through **Bit Plane Address Mapping** <34>, **Bit Plane Address Mux** <34>, and into **Diagnostic Bit & VRS Plane Address Feedback Latches** U500 & U510 <37>. The VRS plane address passes only from the CRT CONTROLLER through the **VRS Plane Address Mux** and into latches U501 & U511 <37>. The addresses are latched in the normal row/column format. When the first address is generated by the CRT CONTROLLER, **Diagnostic CRTC/MPU Address Trigger Control** U402B-11 <37> is clocked. This enables the row and column strobe signals at U401A-2 & U401D-12 after U401B-6 CCLK(L) goes low. U401A-3 Q(H):L is the row strobe and U401D-11 CAS(L):L is the column strobe.

6. Read and verify latched bit plane address (see first Error Index section below).

- Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U500 & U510 <37>, ignoring U510-8 RA0 and U510-9 RA1 (their values are unpredictable), and verify against the expected address.

7. Read and verify latched VRS plane address (see second Error Index section below).

   - Disable **Diagnostic CRTC/MPU Address Trigger Control** <37> from latching further bit plane and VRS plane addresses by setting **Diagnostic Control/Status Latch** U600-16 <37> high.

   - Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 <37>, ignoring U511-8 RA2 and U511-9 RA3 (their values are unpredictable), and verify against the expected address.

8. Repeat steps 2-7 until all patterns have been tested or a verify operation fails.

9. Restore CRT CONTROLLER "top of page" register to its normal operating value (0000*hex*).

Error Index D2221 The bit plane latched address in **Diagnostic Bit & VRS Plane Address Feedback Latches** U500 & U510 did not match the expected value. The display shows the first bit plane address pattern which did not match. The CRT CONTROLLER addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CRT CONTROLLER Address	Expected Bit Plane Address
0000*hex*	0000*hex*
0001*hex*	0000*hex*
0002*hex*	0100*hex*
0004*hex*	0200*hex*
0008*hex*	0400*hex*
0010*hex*	0800*hex*
0020*hex*	1000*hex*
0040*hex*	0000*hex*
0080*hex*	0000*hex*
0100*hex*	8001*hex*
0200*hex*	0003*hex*
0400*hex*	0006*hex*
0800*hex*	000C*hex*
1000*hex*	0018*hex*
2000*hex*	0030*hex*
4000*hex*	0000*hex*
8000*hex*	0000*hex*

Error Index D2222 The VRS plane latched address in **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 did not match the expected value. The display shows the first VRS plane address pattern which did not match. The CRT CONTROLLER addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CRT CONTROLLER Address	Expected VRS Plane Address
0000*hex*	0000*hex*
0001*hex*	0100*hex*
0002*hex*	0200*hex*
0004*hex*	0400*hex*
0008*hex*	0800*hex*
0010*hex*	0000*hex*
0020*hex*	0000*hex*
0040*hex*	0000*hex*
0080*hex*	0000*hex*
0100*hex*	0001*hex*
0200*hex*	0002*hex*
0400*hex*	0004*hex*
0800*hex*	0008*hex*
1000*hex*	0010*hex*
2000*hex*	0020*hex*
4000*hex*	0000*hex*
8000*hex*	0000*hex*

Error Index D2223 The vertical sync pulse from step 1 could not be detected within the timeout period.

Routine Name CRTC R/W

Overview This test verifies the capability to write and read the control registers of **CRT Controller & Select** U515 CRT CONTROLLER <34> by writing and reading one pattern to one CRT CONTROLLER control register.

Description

1. Read and save U515 CRT CONTROLLER <34> "cursor position low byte" control register.

2. Write test pattern to CRT CONTROLLER control register.

   - Write 55*hex* to the CRT CONTROLLER "cursor position low byte" control register. This sets U534-1 & 2 CRTCS(L) and U631A-2 R/W(L) all low. When U631A-3 clocks, due to U534-8 MPUCLK clocking, U515-22 CRT R/W(L) goes low and the test pattern is written into the CRT CONTROLLER when U515-23 02(H) clocks.

3. Read and verify test pattern from CRT CONTROLLER control register.

   - Read from CRT CONTROLLER "cursor position low byte" control register. This sets U534-1 & 2 CRTCS(L) low and U631A-2 R/W(L) high. When U631A-3 clocks, due to U534-8 MPUCLK clocking, U515-22 CRT R/W(L) goes high and the control register is read from the CRT CONTROLLER when U515-23 02(H) clocks. The register value read is checked to see that it is 55*hex*.

4. Restore the CRT CONTROLLER's "cursor position low byte" control register with the value saved in step 1.

Error Index D2231 The pattern read back from the **CRT Controller & Select** U515 CRT CONTROLLER <34> "cursor position low byte" control register was not the pattern that was written (i.e., 55*hex*).

Routine Name

Data Lines

Overview

This test verifies the data lines from the CPU, through **Video Memory Buffer & Select** <33> and **Plane 1 Data Buffer** <35>, to the **Bit Plane 1 DRAM** <35> by performing a "walking one's" test on one bit plane 1 memory location.

Description

1. Perform a "walking one's" test on **Bit Plane 1 DRAM** <35> address 50000*hex*. Terminate test if any verify operation fails.

   - Write the pattern 0000*hex* to address 50000*hex* Read the same address and verify that it was 0000*hex*. Continue this write/read/verify sequence with the patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*,0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*

Error Index D2311

The pattern read from **Bit Plane 1 DRAM** <35> address 50000*hex* was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to DRAM devices is shown in the following table:

D0-D3	D4-D7	D8-D11	D12-D15
U213	U212	U211	U210

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of the **Bit Plane 1 DRAM** <35> by performing a RAM test on all bit plane 1 memory locations.

Description

1. Verify **Bit Plane 1 DRAM** <35> address range 50000*hex* to 5FFFF*hex*. Terminate test if any verify operation fails.

   - Fill address range 50000*hex* to 5FFFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 50000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 50000*hex*. Increment the address and continue this read/verify/write sequence until address 5FFFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 50000*hex*, for CCCC*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index D2321

The pattern read from the displayed memory location in **Bit Plane 1 DRAM** <35> was not the pattern written. The correspondence of the address and data bits to RAM devices is shown in the following table:

Bit Plane 1 DRAM Address	D0-D3	D4-D7	D8-D11	D12-D15
5XXX0*hex*, 5XXX4*hex*,	U213	U212	U211	U210
5XXX8*hex*, 5XXXC*hex*	U213	U212	U211	U210
5XXX2*hex*, 5XXX6*hex*,	U217	U216	U215	U214
5XXXA*hex*, 5XXXE*hex*	U217	U216	U215	U214

X - don't care

Routine Name Data Lines

Overview This test verifies the data lines from the CPU, through **Video Memory Buffer & Select** <33> and **Plane 2 Data Buffer** <35>, to the **Bit Plane 2 DRAM** <35> by performing a "walking one's" test on one bit plane 2 memory location.

Description

1. Perform a "walking one's" test on **Bit Plane 2 DRAM** <35> address 60000*hex*. Terminate test if any verify operation fails.

   - Write the pattern 0000*hex* to address 60000*hex* Read the same address and verify that it was 0000*hex*. Continue this write/read/verify sequence with the patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*

Error Index D2411 The pattern read from **Bit Plane 2 DRAM** <35> address 60000*hex* was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to DRAM devices is shown in the following table:

D0-D3	D4-D7	D8-D11	D12-D15
U313	U312	U311	U310

Routine Name Address/Data

Overview This test verifies the address lines and data integrity of the **Bit Plane 2 DRAM <35>** by performing a RAM test on all bit plane 2 memory locations.

Description

1. Verify **Bit Plane 2 DRAM <35>** address range 60000*hex* to 6FFFF*hex*. Terminate test if any verify operation fails.

   - Fill address range 60000*hex* to 6FFFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 60000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 60000*hex*. Increment the address and continue this read/verify/write sequence until address 6FFFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 60000*hex*, for CCCC*hex*, 5555*hex*, and 0000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index D2421 The pattern read from the displayed memory location in **Bit Plane 2 DRAM <35>** was not the pattern written. The correspondence of the address and data bits to RAM devices is shown in the following table:

Bit Plane 2 DRAM Address	D0-D3	D4-D7	D8-D11	D12-D15
6XXX0*hex*, 6XXX4*hex*,	U313	U312	U311	U310
6XXX8*hex*, 6XXXC*hex*	U313	U312	U311	U310
6XXX2*hex*, 6XXX6*hex*,	U317	U316	U315	U314
6XXXA*hex*, 6XXXE*hex*	U317	U316	U315	U314

X - don't care

Routine Name Data Lines

Overview This test verifies the data lines from the CPU, through **Video Memory Buffer & Select** <33> and **VRS Plane Data Buffers** <36>, to the **VRS Min Plane DRAM** <36> by performing a "walking one's" test on one VRS plane memory location.

Description

1. Perform a "walking one's" test on **VRS Min Plane DRAM** <36> address 70000*hex*. Terminate test if any verify operation fails.

   - Write the pattern 0000*hex* to address 70000*hex* Read the same address and verify that it was 0000*hex*. Continue this write/read/verify sequence with the patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*

Error Index D2511 The pattern read from **VRS Min Plane DRAM** <36> address 70000*hex* was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to DRAM devices is shown in the following table:

D0-D3	D4-D7	D8-D11	D12-D15
U110	U111	U112	U113

Routine Name Address/Data

Overview This test verifies the address lines and data integrity of the **VRS Min Plane DRAM <36>** and **VRS Max Plane DRAM <36>** by performing a RAM test on all VRS plane memory locations.

Description

1. Verify VRS memory address range 70000*hex* to 7FFFF*hex*. Terminate test if any verify operation fails.
   - Fill address range 70000*hex* to 7FFFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 70000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 70000*hex*. Increment the address and continue this read/verify/write sequence until address 7FFFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 70000*hex*, for CCCC*hex*, 5555*hex*, and 1000*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index D2521 The pattern read from the displayed memory location in VRS memory <36> was not the pattern written. VRS memory is divided into two banks, **VRS Min Plane DRAM <36>** and **VRS Max Plane DRAM <36>**. The correspondence of the address and data bits to RAM devices is shown in the following table:

VRS Memory Address	D0-D3	D4-D7	D8-D11	D12-D15
7XXX0*hex*, 7XXX4*hex*,	U110	U111	U112	U113
7XXX8*hex*, 7XXXC*hex*	U110	U111	U112	U113
7XXX2*hex*, 7XXX6*hex*,	U120	U121	U122	U123
7XXXA*hex*, 7XXXE*hex*	U120	U121	U122	U123

X - don't care

Routine Name

Single Axis

Overview

This test verifies the single axis operation of **VRS Generation & Control** <36>. It functionally tests the operations and circuits necessary for displaying waveforms on a single axis; for correctly resolving color attribute overlays between waveforms; and for correctly resolving the underrange, overrange, and null conditions of waveform data. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane DRAM** <36>, representing waveform data for eight separate waveforms plus the clipping waveform, and then reading the individual color pixel information out of the **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.

   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.

2. Initialize **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36> to single axis mode by setting **MUART** U523-33 SCALE(L):L <33>.

3. Initialize **VRS Min Plane DRAM** <36> and **VRS Max Plane DRAM** <36>.

   - Load **VRS Max Plane DRAM** and **VRS Min Plane DRAM** with test data (eight waveforms plus the clipping waveform). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **VRS Generation & Control** U124, U231, and U230 <36> operate in a normal manner).

4. Capture and read individual pixel color data.

   - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all bit plane data and use the data only from the VRS planes. This is accomplished by toggling **MUART** U523-28 VRS IN CLK <33> and serially clocking data into the input register through **MUART** U523-31 DATA OUT. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.

   - Wait for two vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.

   - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking MUART U523-25 VRS OUT CLK and U523-26 VRS GATE <33>, reading 16 pixels, one-by-one, from U523-34 DATA IN.

5. Repeat steps 3 and 4 with a second set of test data.

6. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2611

One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2612 A vertical sync condition in CRT Controller & Select U515 <34> could not be detected.

Routine Name

Dual Axis

Overview

This test verifies the dual axis operation of **VRS Generation & Control** <36>. It functionally tests the operations and circuits necessary for displaying waveforms on a dual axis; for correctly resolving color attribute overlays between waveforms; and for correctly resolving the underrange, overrange, and null conditions of waveform data. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane** DRAM <36>, representing waveform data for eight separate waveforms plus the clipping waveform, and then reading the individual color pixel information out of the **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.
   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36> to dual axis mode by setting **MUART** U523-33 SCALE(L):H <33>.
3. Initialize **VRS Min Plane DRAM** <36> and **VRS Max Plane** DRAM <36>.
   - Load **VRS Max Plane DRAM** and **VRS Min Plane DRAM** with test data (eight waveforms plus the clipping waveform). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **VRS Generation & Control** U124, U231, and U230 <36> operate in a normal manner).
4. Capture and read individual pixel color data.
   - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all bit plane data and use the data only from the VRS planes. This is accomplished by toggling **MUART** U523-28 VRS IN CLK <33> and serially clocking data into the input register through **MUART** U523-31 DATA OUT. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
   - Wait for two vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.
   - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking **MUART** U523-25 VRS OUT CLK and U523-26 VRS GATE <33>, reading 16 pixels, one-by-one, from U523-34 DATA IN.
5. Repeat steps 3 and 4 with a second set of test data.
6. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2621

One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2622 A vertical sync condition in **CRT Controller & Select** U515 <34> could not be detected.

Routine Name

Color Map

Overview

This test verifies the pixel color encoding operation of **VRS Generation & Control** <36>. It functionally tests the operations and circuits necessary for correctly resolving color attribute overlays between waveforms. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane DRAM** <36>, representing waveform data for eight separate waveforms plus the clipping waveform, and then reading the individual color pixel information out of the **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.
   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36> to single axis mode by setting **MUART** U523-33 SCALE(L):L <33>.
3. Initialize VRS Min Plane DRAM <36> and VRS Max Plane DRAM <36>.
   - Load **VRS Max Plane DRAM** and **VRS Min Plane DRAM** with test data (eight waveforms plus the clipping waveform). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **VRS Generation & Control** U124, U231, and U230 <36> operate in a normal manner).
4. Load the color map in the CUSTOM GATE ARRAY.
   - Toggle **MUART** U523-29 COLOR CLK <33> and serially clock data into the color map through **MUART** U523-31 DATA OUT.
5. Capture and read individual pixel color data.
   - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all bit plane data and use the data only from the VRS planes. This is accomplished by toggling **MUART** U523-28 VRS IN CLK <33> and serially clocking data into the input register through MUART U523-31 DATA OUT. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
   - Wait for 2 vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.
   - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking **MUART** U523-25 VRS OUT CLK and U523-26 VRS GATE <33>, reading 16 pixels, one-by-one, from U523-34 DATA IN.
6. Repeat steps 4 and 5 with a second color map setting.
7. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2631 One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2632 A vertical sync condition in **CRT Controller & Select** U515 <34> could not be detected.

Routine Name

Video Shfter (Video Shifter)

Overview

This test verifies **Plane 1 Video Shifter** <35> and **Plane 2 Video Shifter** <35>. **Bit Plane 1 DRAM** <35> and **Bit Plane 2 DRAM** <35> are loaded with data designed to display 16 consecutive pixels with an incremental color index. The test is designed to to drive all bits high and low through the video shifter outputs. The test also functionally verifies the color attribute overlays between bit plane color indices. The color overlay function resides in the **VRS Generation & Control** CUSTOM GATE-ARRAY U125 <36>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.

   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.

2. Initialize **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36> to single axis mode by setting **MUART** U523-33 SCALE(L):L <33>.

3. Initialize Bit Plane 1 DRAM <35> and Bit Plane 2 DRAM <35>.

   - Load **Bit Plane 1 DRAM** and **Bit Plane 2 DRAM** with test data (values AAAA*hex* and 5555*hex*, respectively). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **Plane 1 Video Shifter**<35>, **Plane 2 Video Shifter** <35>, and **VRS Generation & Control** U124, U231, and U230 <36> all operate in a normal manner).

4. Capture and read individual pixel color data.

   - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all VRS plane data and use the data only from the bit planes. This is accomplished by toggling **MUART** U523-28 VRS IN CLK <33> and serially clocking data into the input register through **MUART** U523-31 DATA OUT. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.

   - Wait for 2 vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.

   - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking **MUART** U523-25 VRS OUT CLK and U523-26 VRS GATE <33>, reading 16 pixels, one-by-one, from U523-34 DATA IN.

5. Repeat steps 4 and 5 with a second set of test data (5555*hex* for **Bit Plane 1DRAM** and AAAA*hex* for **Bit Plane 2 DRAM**).

6. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2641

One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2642 A vertical sync condition in CRT Controller & Select U515 <34> could not be detected.

Routine Name

Priority

Overview

This test verifies the priority encoding operation between bit plane and VRS plane color pixel data of **VRS Generation & Control** <36>. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane DRAM** <36>, representing waveform data for eight separate waveforms plus the clipping waveform, and by placing data in **Bit Plane 1 DRAM** and **Bit Plane 2 DRAM** <35> designed to display 16 consecutive pixels with incremental color indices. The priority encoder should override all waveform pixel data (from the VRS planes) with bit plane pixel data which contains a color index greater than zero. The individual color pixel information is read out of the **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <34>.
   - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control** CUSTOM GATE ARRAY U125 <36> to single axis mode by setting **MUART** U523-33 SCALE(L):L <33>.
3. Initialize **VRS Min Plane DRAM** <36>, **VRS Max Plane DRAM** <36>, **Bit Plane 1 DRAM** <35>, and **Bit Plane 2 DRAM** <35>.
   - Load **VRS Min Plane DRAM** <36>, **VRS Max Plane DRAM** <36>, **Bit Plane 1 DRAM** <35>, and **Bit Plane 2 DRAM** <35> with test data. The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **Plane 1 Video Shifter** <35>, **Plane 2 Video Shifter**<35>, and **VRS Generation & Control** U124, U231, and U230 <36> all operate in a normal manner).
4. Capture and read individual pixel color data.
   - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY so that it does not mask off either the VRS plane or bit plane data when capturing pixel information. This is accomplished by toggling **MUART** U523-28 VRS IN CLK <33> and serially clocking data into the input register through **MUART** U523-31 DATA OUT. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
   - Wait for two vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.
   - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking **MUART** U523-25 VRS OUT CLK and U523-26 VRS GATE <33>, reading 16 pixels, one-by-one, from U523-34 DATA IN.
5. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2651

One of the three pixel color data values did not match the expected results.

Error Index D2652 A vertical sync condition in **CRT Controller & Select** U515 <34> could not be detected.

Routine Name

Stimulus

Overview

This test verifies the **Z Axis Amplifier** U601 and U600 <38> by changing the intensity of the display CRT screen using **MUART** U523 <33>.

Operator Procedure

This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off". Once this test is invoked, the operator is required to verify that the intensity of the display CRT screen changes through four different intensity levels.

Description

1. Read and save the value of **MUART** PORT 2.
2. Set the **Z Axis Amplifier** U601 <38> (intensity or brightness register) to FF*hex*.
   - Read and save the value of **MUART** PORT 2 again.
   - Shift FF*hex* into U601 bit by bit starting with the MSB (bit 7). In order to shift a bit into U601, the value of the bit is gated onto DATA OUT U523-31 <33> (by writing it to bit 0 of **MUART** PORT 2) and at the same time BRITE CLK U523-30 <33> is set low and then high (by toggling bit 1 of **MUART** PORT 2 while maintaining the value of bit 0).
   - Restore the value of **MUART** PORT 2.
   - Perform a software delay.
3. Repeat step 2 with values 3F*hex*, 0F*hex*, and 03*hex*.
4. Set the intensity register U601 to 80*hex*.
5. Restore the value of **MUART** PORT 2 again.

Error Index

None.

Routine Name U281

Overview This test verifies that ROM & Select U281 <16> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index G1111 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Error Index ???? The bytes at E0008*hex* and E000A*hex* were not complementary.

Routine Name U283

Overview This test verifies that ROM & Select U283 <16> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
   - Read and verify the location byte against the known value.

Error Index G1121 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Error Index ???? The bytes at E0009*hex* and E000B*hex* were not complementary.

Routine Name U281

Overview This test verifies the integrity of ROM & Select U281 <16> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U281 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index G1211 The computed checksum did not match the stored checksum.

Error Index ???? The bytes at E0008*hex* and E000A*hex* were not complementary.

Routine Name U283

Overview This test verifies the integrity of ROM & Select U283 <16> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
   - Exclusive-or the contents of byte locations E0009*hex* and E000B*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
   - Perform a checksum on all the bytes in U283 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index G1221 The computed checksum did not match the stored checksum.

Error Index ???? The bytes at E0009*hex* and E000B*hex* were not complementary.

Routine Name

Data Lines

Overview

This test verifies the data lines from the MPU (U224) <16>, through the **MPU Data Buffers** (U262, U263), to the **MPU RAM** (U271, U273) by performing a "walking one's" test on **MPU RAM** location 00000*hex*.

Description

1. Perform a "walking one's" test on **MPU RAM** address 00000*hex*. Terminate test if any verify operation fails.

   - Write the pattern 0000*hex* to address 00000*hex*. Read the same address and verify that it was 0000*hex*. Continue this write/read/verify sequence with the patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*,0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*.

Error Index G1311

The pattern read from **MPU RAM** address 00000*hex* was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to RAM devices is shown in the following table:

D0-D7	D8-D15
U271	U273

Routine Name Address/Data

Overview This test verifies the address lines from the **MPU** (U224) <16>, through the **MPU Address Latch** (U232, U252, U253), to the **MPU RAM** (U271, U273), and the data integrity of the **MPU RAM** itself, by performing a RAM test on all **MPU RAM** locations.

Description

1. Verify **MPU RAM** address range 02000*hex* to 03FFF*hex*. Terminate test if any verify operation fails.
   - Fill address range 02000*hex* to 03FFF*hex* with the pattern AAAA*hex*.
   - Read and verify address 02000*hex* for AAAA*hex*. If so, write CCCC*hex* to address 02000*hex*. Increment the address and continue this read/verify/write sequence until address 03FFF*hex* is reached.
   - Repeat the read/verify/write sequence, starting again at address 02000*hex*, for CCCC*hex*, F0F0*hex*, 5555*hex*, and AAAA*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
   - Make one last read/verify pass for the pattern AAAA*hex* starting at address 02000*hex*.
2. Save **MPU RAM** data in address range 00000*hex* to 01FFF*hex* by copying it to **MPU RAM** address range 02000*hex* to 03FFF*hex*.
3. Verify **MPU RAM** address range 00000*hex* to 01FFF*hex*.
   - Perform the same procedures as in step 1 for the address range 00000*hex* to 01FFF*hex*.
4. Copy **MPU RAM** data back from address range 02000*hex* to 03FFF*hex* to **MPU RAM** address range 00000*hex* to 01FFF*hex*.

Error Index G1321 The pattern read from the displayed memory location in **MPU RAM** was not the pattern written. The correspondence of the address and data bits to RAM devices is shown in the following table:

MPU RAM Address	D0-D7	D8-D15
0XXXX*(hex)*	U271	U273

X - don't care

Routine Name Timer 2

Overview This test verifies MPU U224 <16> Programmable Timers timer 2's counting accuracy and its interrupt. The counting accuracy of the timer 2 is verified by counting the system clock for a short duration. If the timer 2 counts accurately, a timer 2 interrupt is generated and verified.

Description

1. Stop the timer 2 by writing 4000*hex* to its Timer Mode/Control Register.
2. Program the counter to count up by writing 0000*hex*, FFFF*hex*, and C000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
3. Perform a software delay.
4. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
5. Read and verify the timer 2 count against the known value for the expected tolerance.
6. If the count was within the expected tolerance (i.e., the timer 2 is counting accurately), clear any pending interrupts by writing 0000*hex* to both Interrupt Status Register and Interrupt In Service Register. Else terminate testing.
7. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
8. Enable the timer 2 interrupt by writing 0000*hex* to the Timer 2 Interrupt Control Register.
9. Program the counter to count up to 1000*hex* and generate an interrupt when the count reaches 1000*hex*.
    - Write 0000*hex*, 1000*hex*, E000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
10. Perform a software delay.
11. Disable timer 2 interrupt by writing 000F*hex* to the Timer 2 Interrupt Control Register.
12. Verify that the timer 2 interrupt did occur.

Error Index G1411 The timer 2 count was not within the expected tolerance.

Error Index G1412 Timer 2 interrupt did not occur.

Routine Name

DMA 0

Overview

This test verifies MPU U224 <16> Programmable DMA Unit's channel 0 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Program the DMA channel 0 to transfer the five source patterns — AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex* from consecutive EPROM memory locations to the destination memory locations in MPU RAM <16> and generate an interrupt after completion of the transfer.

   - Write B725*hex*, 0005*hex*, upper four bits of destination address, lower 16 bits of destination address (the address may be different in each version of software), upper four bits of source address, lower 16 bits of source address (again the address may be different in each version of software) and B727*hex* to DMA 0 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source Pointer Lower, and DMA 0 Channel Control Word registers, respectively. The last write initiates the DMA transfer.

2. Disable DMA channel 0 interrupt by writing B725*hex* to DMA 0 Channel Control Register.

3. Read and verify the destination memory locations.

Error Index G1511

One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Routine Name

DMA 1

Overview

This test verifies MPU U224 <16> Programmable DMA Unit's channel 1 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Program the DMA channel 1 to transfer the five source patterns — AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, and 5555*hex* from consecutive EPROM memory locations to the destination memory locations in MPU RAM <16> and generate an interrupt after completion of the transfer.

   - Write B725*hex*, 0005*hex*, upper four bits of destination address, lower 16 bits of destination address (the address may be different in each version of software), upper four bits of source address, lower 16 bits of source address (again the address may be different in each version of software) and B727*hex* to DMA 1 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source Pointer Lower, and DMA 1 Channel Control Word registers, respectively. The last write initiates the DMA transfer.

2. Disable DMA channel 1 interrupt by writing B725*hex* to DMA 1 Channel Control Register.

3. Read and verify the destination memory locations.

Error Index G1521

One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Routine Name Data

Overview This test verifies the **Digitizer/MMU Communication Port** <17> by performing a "walking one's" test on the port output and input latches.

Description

1. Initialize hardware for loop back operation.

   - Reset the digitizer hardware by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> low.
   - Initialize the state of the **DAG** <20> (especially pin 68, DAG_REQUEST(L) high) by setting the DAG IC U166 to diagnostic mode via setting bit 5 high in DAG IC Control Register 0 and then setting **TB Control Latch 0** U240-19 SET_RESTART(L):H and U240-18 CLR_RESTART(L):H <15>.
   - Force the system clocks into normal free-running mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):L <14>.
   - Initialize the **Initial Wfm Id Encoder Table** <14> to prevent any acquisition memory requests (i.e., force U12-6 VALIDATE_ACQ_REQUEST(H) low) by writing 0*hex* to all RAM locations in U22 and U32.
   - Initialize the state of **Acq Mem Control Logic** <13> and **Acq Mem Address Generation** <13> by setting **TB Control Latch 1** U251-14 INV_PT_RESET(L):H <15> and performing a read and write to **Acq Mem Even Bank** <13>.
   - Enable communication handshaking by setting **Digitizer/MMU Handshake Logic** U83C-11 DIGREQ_EN(H):H <17>.
   - Enable the diagnostic loop back function by setting **Diagnostic Loopback Control** U146B-19 LOOPBACK_EN(L):L <17>.
   - Force the system clocks into single-step mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):H <14>.
   - Initialize **Digitizer/MMU Communication Port** <17> to a known state by toggling **Diagnostic Loopback Control** <17> U85C-9 FORCE_DATA(L) and reading and writing the communication port latches U106 and U126.

2. Perform a "walking one's" test on the **Digitizer/MMU Communication Port**, writing patterns to communication port output latches U116 and U136, and reading the patterns back through communication port input latches U106 and U126.

   - Write the pattern 0000*hex* to the communication port output latches U116 and U136.
   - Toggle U85C-9 FORCE_DATA. When FORCE_DATA goes low, **Digitizer/MMU Handshake Logic** U76B-6 REPLY_EN(L) goes low and

enables the outputs of communication port output latches U116 and U136. When FORCE_DATA goes high, the data from U116 and U136 is latched into the communication port input latches U106 and U126.

- Read the data from the communication port input latches U106 and U126 and verify that it was the same pattern as written.
- Continue this write/read/verify sequence with patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000*hex*, 4000*hex*, and 8000*hex*.

3. Terminate loop back operation and reset digitizer hardware.

- Initialize the operating state of the communication latches and purge any pending DMA requests by writing a value (which is insignificant) to the **Digitizer/MMU Communication Port** output latches U116 and U136, toggling U85C-9 FORCE_DATA(L), and then reading the communication input latches U106 and U126.
- Disable the loop back function by setting U146B-19 LOOPBACK_EN(L):H <17>, U83C-11 DIGREQ_EN(H):L, and U166-2 WFM_REQ(H):L <20> (by setting bit 0 high in DAG IC Control Register 4).
- Reset the digitizer hardware by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> low.

Error Index G1611 - The data pattern read from the **Digitizer/MMU Communication Port** input latches U106 and U126 did not match the pattern that was written to the communication port output latches U116 and U136. The display shows the first pattern that did not match.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Digitizer busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name Address/Tag

Overview This test verifies the **CPU/MMU Message Address Output Latch** <17>, the **Digitizer/MMU Address Output Port** <17>, and the **CPU/MMU Message Tag Generator** <17> by performing a "walking one's" test on the A1-A9 CPU/MMU address lines.

Description

1. Initialize hardware for loop back operation.

   - Reset the digitizer hardware by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> low.
   - Initialize the state of the **DAG** <20> (especially pin 68, DAG_REQUEST(L) high) by setting the DAG IC U166 to diagnostic mode via setting bit 5 high in DAG IC Control Register 0 and then setting **TB Control Latch 0** U240-19 SET_RESTART(L):H and U240-18 CLR_RESTART(L):H <15>.
   - Force the system clocks into normal free-running mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):L <14>.
   - Initialize the **Initial Wfm Id Encoder Table** <14> to prevent any acquisition memory requests (i.e., force U12-6 VALIDATE_ACQ_REQUEST(H) low) by writing *0hex* to all RAM locations in U22 and U32.
   - Initialize the state of **Acq Mem Control Logic** <13> and **Acq Mem Address Generation** <13> by setting **TB Control Latch 1** U251-14 INV_PT_RESET(L):H <15> and performing a read and write to **Acq Mem Even Bank** <13>.
   - Enable communication handshaking by setting **Digitizer/MMU Handshake Logic** U83C-11 DIGREQ_EN(H):H <17>.
   - Enable the diagnostic loop back function by setting **Diagnostic Loopback Control** U146B-19 LOOPBACK_EN(L):L <17>.
   - Force the system clocks into single-step mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):H <14>.
   - Initialize **Digitizer/MMU Communication Port** <17> to a known state by toggling **Diagnostic Loopback Control** <17> U85C-9 FORCE_DATA(L) and reading and writing the communication port latches U106 and U126.

2. Perform a "walking one's" test on the **CPU/MMU Message Tag Generator** U95 and **CPU/MMU Message Address Output Latch** U124 (address lines A1-A9).

   - Write a data value (the contents of which are insignificant) to the **Digitizer/MMU Communication Port** at communication port test address (see Error Index section below for test address sequence) This write causes Digitizer/MMU Handshake Logic U76C-8 REPLY_LATCH(L) to clock the A9

address bit into **CPU/MMU Message Tag Generator** U95 and the A1-A8 address bits into **CPU/MMU Message Address Output Latch** U124.

- Toggle U85C-9 FORCE_DATA(L). This provides a clock pulse to **Digitizer/MMU Handshake Logic** flip-flop U75B, thereby causing U75B-8 REPLY_ADDR_EN(L) to go low. This enables the **CPU/MMU Message Tag Generator** and **CPU/MMU Message Address Output Latch** outputs.

- Toggle **Diagnostic Loopback Control** U85C-10 FORCE_ADDR(L). When FORCE_ADDR goes low, the **Digitizer/MMU Address Output Port** (U115, U125) is enabled. When FORCE_ADDR goes high, the message address now present on ADB0-7 of U125 is clocked into **Digitizer/MMU Communication Port** latch U126, and the message tag present on ADB16-19 of U95 is clocked into **Diagnostic Wfm Tag Readback Latch** U97.

- Read and verfiy the message address from the **Digitizer/MMU Communication Port** and the message tag from the **Diagnostic Wfm Tag Readback Latch** (see Error Index sections below for expected address and tag data).

- Repeat the previous four steps for the remaining communication port addresses (see Error Index section below).

3. Terminate loop back operation and reset digitizer hardware.

   - Initialize the operating state of the communication latches and purge any pending DMA requests by writing a value (which is insignificant) to the **Digitizer/MMU Communication Port** output latches U116 and U136, toggling U85C-9 FORCE_DATA(L), and then reading the communication input latches U106 and U126.

   - Disable the loop back function by setting U146B-19 LOOPBACK_EN(L):H <17>, U83C-11 DIGREQ_EN(H):L, and U166-2 WFM_REQ(H):L <20> (by setting bit 0 high in DAG IC Control Register 4).

   - Reset the digitizer hardware by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> low.

Error Index G1621

The message address read from the **Digitizer/MMU Communication Port** did not match the address that was written to the **CPU/MMU Message Address Output Latch**. The display shows the first address that did not match. The comunication port addresses written to and the expected message addresses read back are shown in the following table:

Port Address	Message Address
200*hex*	00*hex*
100*hex*	80*hex*
080*hex*	40*hex*
040*hex*	20*hex*
020*hex*	10*hex*
010*hex*	08*hex*

008*hex*	04*hex*
004*hex*	02*hex*
002*hex*	01*hex*
000*hex*	00*hex*

Error Index G1622 The message tag read from the Diagnostic Wfm Tag Readback Latch was incorrect. The display shows the first tag that did not match. The comunication port addresses written to and the expected message tags read back are shown in the following table:

Port Address	Message Tag
200*hex*	F*hex*
100*hex*	E*hex*
080*hex*	E*hex*
040*hex*	E*hex*
020*hex*	E*hex*
010*hex*	E*hex*
008*hex*	E*hex*
004*hex*	E*hex*
002*hex*	E*hex*
000*hex*	E*hex*

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Digitizer busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name

20.0000 MHz

Overview

This test verifies one of the **System Clock Control** <14> 20 MHz clock derivatives (MCK_TIMB_4) by using it to clock the **Window 2 Post Record Logic U264 MAIN D CTR** (window 2 post record counter) <18>. This essentially verifies that the frequency of the **Reference Oscillator** <9> 10 MHz crystal Y100 on the Acquisition board is within approximately ±0.01% and the succeeding circuits necessary to generate the Time Base board's system clocks are somewhat functional.

Description

1. Set up Reference Oscillator <9>.

   - Set R208 **FREQ DELTA** low by serial shifting a low into Acquisition Control Register U1242-12 FREQ DELTA <11>.

2. Enable clock (MCK_TMB_4) to Window 2 Post Record Logic U264 **MAIN D CTR** (window 2 post record counter) <18>.

   - Set Destination Address Generator **DAG IC** U166 <20> to diagnostic mode by setting bit 5 (DiagMpx) high in Control Register 0.

   - Set Window 2 Post Record Logic U294-17 ADV2(L):L by setting bit 3 high in DAG IC Control Register 2. This forces Window 2 Post Record Logic U287-8 <18> to go low, thereby enabling MCK_TIMB_4 through U279A.

3. Load the window 2 post record counter.

   - Reset the digitizer hardware by setting **TB Control Latch** 3 U212-12 **SYSTEM_RESET(L)**:L <15>.

   - Enable the window 2 post record counter load signal by setting TB Control Latch 0 U240-19 **SET_RESTART(L)**:L and U221B-4 **CLR_RESTART(L)**:H <15>. This causes Window 2 Post Record Logic U279B-4 RESTART(L), and therefore U279B-6 **WIND_POSTRECORD_LOAD(L)**, to go low.

   - Put the system clocks into single-step mode by setting TB Control Latch 0 U240-12 **CLOCK_SOURCE_SELECT(H)**:H.

   - Write FFFF*hex* to the 16-bit window 2 post record counter and then toggle the system clocks once, by toggling System Clock Control U67C-5 **CLOCK_STEP(L)** <14>, to actually load the value into the counter.

4. Allow the window 2 post record counter to count, then perform a software delay.

   - Remove the window 2 post record counter load signal by setting TB Control Latch 0 U240-19 **SET_RESTART(L)**:H and U221B-4 **CLR_RESTART(L)**:L <15>.

   - Put the system clocks into normal free-run mode by setting TB Control Latch 0 U240-12 **CLOCK_SOURCE_SELECT(H)**:L. This causes the counter to start counting down at the 20 MHz MCK_TIMB_4 clock rate.

   - Perform a software delay of approximately 2 milliseconds.

5. Stop the window 2 post record counter from counting.

   - Put the system clocks back into single-step mode by setting TB Control Latch 0 U240-12 CLOCK_SOURCE_SELECT(H):H.

6. Read and check the window 2 post record counter value against the test limits.

Error Index G2111 The value read from the window 2 post record counter was not within the expected tolerance limits (i.e., the measured frequency was out of tolerance).

Caveats This test may also fail if the Time Base board MPU clock Y220 <16> is out of tolerance since it is being used as the timing reference for the test via the software delay.

Routine Name

19.6608 MHz

Overview

This test verifies the **19 MHz Clock** Y2040 <10> on the Acquisition board by using it to clock the **Main Holdoff Logic** U264 MAIN A CTR (main holdoff counter) <18> on the Time Base board. The frequency is checked to within ±0.064%, which includes the tolerances of Y2040 (±0.05%) and that of the **MPU** clock Y220 (±0.01%) <16>, since it is being used as the timing reference.

Description

1. Disable the **Main Fine Holdoff Generator** <11>.
   - Set U1242-4 MFINEOFF(L):L
2. Initialize inputs to the **Holdoff Logic** <10>.
   - Set U1644-1 STRGARM(L):H and U1452-2 HOCALSEL(H):H.
   - Set U1452-4 PRETC(H):H by loading the **Pretrigger Generator** U275 WIND C CTR <19> with a value of zero and then manually clocking (via the **System Clock Control** <14>) the resulting terminal count at U275-29 through the **Pretrigger Randomizer** <19>.
   - Select the 19.6608 MHz clock inputs (U2050-2 & 5) of the **Multiplexer** <10> by setting U2030-5 low.
3. Load and start the **Main Holdoff Logic** U264 MAIN A CTR (main holdoff counter).
   - Set U236-9 MHOLOAD(H):H <18>. This sets U237-6 MHORESET(L):L, which in turn causes U247 pins 10 & 8 to go low and provides the load signal to the counter. In addition, MHORESET(L):L causes registers U1550 and 1644B <10> to be reset. The low at U1550-7 causes U1546A to be reset and its low output to be clocked through U1546B. The MHOLDOFFLD(L):L signal is the other load signal to the **Main Holdoff Logic** <18> at U247-9.
   - Write 8000*hex* to the 35-bit main holdoff counter. This causes the counter terminal count U264-19 to go low.
   - Set U236-9 MHOLOAD(H):L <18>. This, and the fact that the input of U237A should be low, sets U237-6 MHORESET(L):H. This causes one of the counter load signals, U247-10, to go high. The counter will not count until the other load signal, MHOLDOFFLD(L), goes high also. This takes place after MHORESET(L) goes high at U1550-1 <10>. When this happens, the high present at the output of U1644A is clocked through register U1550 to the output at U1550-7, thus causing the reset line of U1546A to be released and the high present at its input to be clocked through U1546A and U1546B to set MHOLDOFFLD(L) high.
4. Perform a software delay loop.
5. Stop the main holdoff counter from counting.

- Select the D19 clock inputs (U2050-2 & 5) of the **Multiplexer <10>** by setting U2030-5 high. Since the D19 signal is a software controlled signal, this turns off the main holdoff counter clock and prevents it from counting further.

6. Read and check the main holdoff counter value against the test limits.

Error Index G2121

The value read from the main holdoff counter was not within the expected tolerance limits (i.e., the measured frequency was out of tolerance).

Caveats

This test may also fail if the Time Base board MPU clock Y220 <16> is out of tolerance since it is being used as the timing reference for the test via the software delay.

Routine Name

Rate

Overview

This test verifies the **Main Rate Generator** (U264 MAIN B CTR—26 bits, count-down) <18> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Reload after zero—A check to see that if the counter counts down to zero, and is clocked once more, then it will automatically reload itself with its last loaded value.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into four bytes and written byte-by-byte in least-significant-byte order; to do a read, the counter is read in least-significant-byte order, and then the four bytes reassembled into one 26-bit value (six msb's masked out).
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the timebase counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Main Rate Generator** U264-22 SYSTEM_RESET(L) <18> is set low. This is the load control line.
   - **Main Postrecord Logic** U239D-12 RESTART(L) <18> is set low. This signal & its complement **Window 2 Post Record Logic** U287-3 RESTART(H) <18> control the clock and/or load control lines of most time base counters.
   - **Main Trigger Logic** U258-7 MCG(H) <18> is set low and U258-6 MCG(L) <18> is set high.
   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up hardware so that the strobe line (U264-26 HBSTRB(H)) is de-asserted (low).

   - Write a value to the location of the **Destination Address Generator** (U166 <20>) Control Register 0 such that bit D5 is set high; this causes the diagnostic multiplexer control bit (DiagMpx) to be asserted.

   - Write a value to the location of the **Destination Address Generator** Control Register 2 such that bit D1 (ti_avail) is set high; this causes U166-35 TI_DATA_AVAIL(H) to go low, which in turn causes **Main Rate Generator** U267C-9 TI_DATA_AVAIL(L) <18> to go high. U267C-10 should already be high due to general hardware initialization (MCG(L) = 0). The next clock to **Main Rate Generator** U297-11 will cause U264-26 HBSTRB(H) <18> to go low.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 26 times). The test pattern starts out the first time as 2000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 26 patterns is: 2000000*hex*, 1000000*hex*, 800000*hex*, 400000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.

   - Disable the load line (SYSTEM_RESET(L)) by setting **TB Control Latch 3** U260-18 <15> high and clocking the system clocks once.

   - Toggle the system clocks once more to get the test pattern actually clocked into the counter from its input latches.

   - Read and check that the terminal count bit is 0 and the counter contents matches the test pattern that was loaded.

   - Toggle the system clocks once to cause a ripple count.

   - Read and check that the terminal count bit is 0, except for the very last pass, when it should be 1 (indicates a successful count to zero). Read and check that the counter contents matches a value equal to the tests pattern minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less).

   - Shift the test pattern right one bit.

   - If not the last pass, then enable the load line (SYSTEM_RESET(L)) by setting **TB Control Latch 3** U260-18<15> low and clocking the system clocks once.

VERIFY RELOAD AFTER ZERO

4. Toggle system clocks once to cause the terminal count bit to go back to a de-asserted (low) state, and to cause the counter to be loaded (automatically) with the value of the last loaded test pattern, which should be 1.

5. Read and check that the terminal count bit is 0 and the counter contents is 1.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter to cause the counter to be loaded from its input latches.

   - Enable the load line by setting **TB Control Latch 3** U260-18 <15> low and clocking the system clocks once.
   - Load counter input latches with zero.
   - Disable the load line by setting **TB Control Latch 3** U260-18 <15> high and clocking the system clocks once.
   - Toggle the system clocks to get the zero value actually clocked into the counter from its input latches.

7. Read and check that the terminal count bit is 1 and the counter contents is 0.

8. Clock once more, then read the terminal count bit and the counter contents and check that they haven't changed.

Error Index G2211 Terminal count error just after counter was loaded with the test pattern.

Error Index G2212 Error in upper 10 bits of counter contents just after it was loaded with the test pattern.

Error Index G2213 Error in lower 16 bits of counter contents just after it was loaded.

Error Index G2214 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2215 Error in upper 10 bits of counter contents just after it was clocked to cause ripple count.

Error Index G2216 Error in lower 16 bits of counter contents just after it was clocked to cause ripple count.

Error Index G2217 Terminal count error after counter was clocked down to zero and then clocked one additional time.

Error Index G2218 Error in upper 10 bits of counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2219 Error in lower 16 bits of counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G221A Terminal count error just after counter was loaded with zero.

Error Index G221B Error in upper 10 bits of counter contents just after counter was loaded with zero.

Error Index G221C Error in lower 16 bits of counter contents just after counter was loaded with zero.

Error Index G221D Terminal count error just after counter was clocked once more after being loaded with zero.

Error Index G221E Error in upper 10 bits of counter contents just after counter was clocked once more after being loaded with zero.

Error Index G221F Error in lower 16 bits of counter contents just after counter was clocked once more after being loaded with zero.

Routine Name

Post Record

Overview

This test verifies the Main Postrecord Logic (mainly, U264 MAIN C CTR—16 bits, count-down; plus other logic) <18> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a" walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Stability at zero—A check to see if the counter remains unaffected after it has counted down to zero, and is clocked once more.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into two bytes and written byte-by-byte in least-significant-byte order; to do a read, the counter is read in least-significant-byte order, and then the two bytes reassembled into one 16-bit value.
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the time base counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Main Rate Generator** U264-22 SYSTEM_RESET(L) <18> is set low, so **Main Postrecord Logic** U236A-1 SAMPLE_M <18> is high.
   - **Main Postrecord Logic** U239D-12 RESTART(L) <18> is set low. This signal and its complement **Window 2 Post Record Logic** U287-3 RESTART(H) <18> control the clock and/or load lines of most time base counters.

- **Main Trigger Logic** U258-7 MCG(H) <18> is set low and U258-6 MCG(L) <18> is set high.

- **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up the hardware to allow clocking of this counter when RESTART(L) <18 > is set to the high state.

   - Set **TB Control Latch 3** U260-17 MSET(H):H <15>.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 16 times). The test pattern starts out the first time as 8000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 16 patterns is: 8000*hex*, 4000*hex*, 2000*hex*, 1000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.

   - Toggle the system clocks once to get the test pattern actually clocked into the counter from its input latches.

   - Disable the load line (**Main Postrecord Logic** U264-28 MAIN_COUNTER_LOAD(L)) by setting **TB Control Latch 0** U240-19 SET_RESTART(L):H <15> and **TB Control Latch 0** U221B-4 CLR_RESTART(L):L <15>, which causes **Main Rate Generator** U239D-12 RESTART(L) to go high.

   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.

   - Toggle the system clocks three times to cause a ripple count. It requires two preliminary clocks to get the clock to the counter enabled.

   - Read and check the terminal count bit. It should be 0 except for the very last pass; then it it should be 1 (indicates a successful count to zero).

   - Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex*, then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.

   - Shift the test pattern right one bit.

   - If not the last pass, then enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4

CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go low.

VERIFY STABILITY AT ZERO

4. Toggle system clocks once to see if it has any effect. It should have none.

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 1 (0), as previously explained.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go low.
   - Load counter input latches with zero.
   - Toggle the system clocks once to actually load the counter from its input latch.

7. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G2221 Terminal count error just after counter was loaded with the test pattern.

Error Index G2222 Error in counter contents just after it was loaded with the test pattern.

Error Index G2223 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2224 Error in counter contents just after it was clocked to cause ripple count DURING THE LAST PASS of the writability/countability part of this test.

Error Index G2225 Error in counter contents just after it was clocked to cause ripple count ON ALL BUT THE LAST PASS of the writability/countability part of this test.

Error Index G2226 Terminal count error after counter was clocked down to zero and then clocked one additional time.

Error Index G2227 Error in counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2228 Terminal count error just after counter was loaded with zero.

Error Index G2229 Error in counter contents just after counter was loaded with zero.

Routine Name

Pretrigger

Overview

This test verifies the **Pretrigger Generator** (mainly, U275 WIND C CTR—16 bits, count-down; plus other logic) <19> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a" walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Stability at zero—A check to see if the counter remains unaffected after it has counted down to zero, and is clocked once more.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into two bytes and written byte-by-byte in least-significant-byte order; to do a read, the counter is read in least-significant-byte order, and then the two bytes reassembled into one 16-bit value.
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the time base counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Window Rate Generator** U275-22 SYSTEM_RESET(L) <19> is set low, and so U239C-9 MAIN_RATE(L) is low.
   - **Pretrigger Generator** U239D-12 RESTART(L) <19> is set low. This signal and its complement **Window 1 Post Record Logic** U278-3 RESTART(H) <19> control the clock and/or load lines of most time base counters.

- **Window Trigger Logic** U258-2 WCG(H) <18> is set low and U258-3 WCG(L) <19> is set high.

- **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

VERIFY WRITABILITY/COUNTABILITY

2. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 16 times). The test pattern starts out the first time as 8000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 16 patterns is: 8000*hex*, 4000*hex*, 2000*hex*, 1000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.

   - Toggle the system clocks once to get the test pattern actually clocked into the counter from its input latches.

   - Disable the load line (**Main Postrecord Logic** U275-28 MAIN_COUNTER_LOAD(L)) by setting **TB Control Latch 0** U240-19 SET_RESTART(L):H <15> and **TB Control Latch 0** U221B-4 CLR_RESTART(L):L <15>, which causes **Main Rate Generator** U239D-12 RESTART(L) to go high.

   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.

   - Toggle the system clocks three times to cause a ripple count. It requires two preliminary clocks to get the clock to the counter enabled.

   - Read and check the terminal count bit. It should be 0 except for the very last pass; then it it should be 1 (indicates a successful count to zero).

   - Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex*. then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's conplement of the last value that was loaded, namely, the 1's complement of 1, which is 0.

   - Shift the test pattern right one bit.

   - If not the last pass, then enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go low.

VERIFY STABILITY AT ZERO

4. Toggle system clocks once to see if it has any effect. It should have none.

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 1 (0), as previously explained.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go low.
   - Load counter input latches with zero.
   - Toggle the system clocks once to actually load the counter from its input latch.

7. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G2231 Terminal count error just after counter was loaded with the test pattern.

Error Index G2232 Error in counter contents just after it was loaded with the test pattern.

Error Index G2233 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2234 Error in counter contents just after it was clocked to cause ripple count DURING THE LAST PASS of the writability/countability part of this test.

Error Index G2235 Error in counter contents just after it was clocked to cause ripple count ON ALL BUT THE LAST PASS of the writability/countability part of this test.

Error Index G2236 Terminal count error after counter was clocked down to zero and then clocked one additional time.

Error Index G2237 Error in counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2238 Terminal count error just after counter was loaded with zero.

Error Index G2239 Error in counter contents just after counter was loaded with zero.

Routine Name

Rate

Overview

This test verifies the **Window Rate Generator** (U275 WIND B CTR—26 bits, count-down) <19> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Reload after zero—A check to see that if the counter counts down to zero, and is clocked once more, then it will automatically reload itself with its last loaded value.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into four bytes and written byte-by-byte least-significant-byte first; to do a read, the counter is read least-significant-byte first, and then the four bytes reassembled into one 26-bit value (six MSBs masked out).
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the timebase counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Window Rate Generator** U275-22 SYSTEM_RESET(L) <19> is set low. This is the load control line.
   - **Pretrigger Generator** U239C-10 RESTART(L) <19> is set low. This signal and its complement **Window 1 Post Record Logic** U278-3 RESTART(H) <19> control the clock and/or load control lines of most timebase counters.
   - **Window Trigger Logic** <19> U258-2 WCG(H) is set low and U258-3 WCG(L) is set high.
   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up hardware so that the strobe line (U275-26 HBSTRB(H)) is de-asserted (low).

   - Write a value to the location of the **Destination Address Generator** (U166 <20>) Control Register 0 such that bit D5 is set high; this causes the diagnostic multiplexer control bit (DiagMpx) to be asserted.

   - Write a value to the location of the **Destination Address Generator** Control register 2 such that bit D1 (ti_avail) is set high; this causes U166-35 TI_DATA_AVAIL(H) to go low, which in turn causes **Trig To Trig Measurement** U267B-5 TI_DATA_AVAIL(L) <19> to go high. U267B-4 should already be high due to general hardware initialization (WCG(L) = 0). The next clock to **Main Rate Generator** U297-11 will cause U275-26 HBSTRB(H) <19> to go low.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 26 times). The test pattern starts out the first time as 2000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 26 patterns is: 2000000*hex*, 1000000*hex*, 800000*hex*, 400000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.

   - Disable the load line (SYSTEM_RESET(L)) by setting **TB Control Latch 3**-U260-18 <15> high and clocking the system clocks once.

   - Toggle the system clocks once more to get the test pattern actually clocked into the counter from its input latches.

   - Read and check that the terminal count bit is 0 and the counter contents matches the test pattern that was loaded.

   - Toggle the system clocks once to cause a ripple count.

   - Read and check that the terminal count bit is 0, except for the very last pass, when it should be 1 (indicates a successful count to zero). Read and check that the counter contents matches a value equal to the tests pattern minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less).

   - Shift the test pattern right one bit.

   - If not the last pass, then enable the load line (SYSTEM_RESET(L)) by setting **TB Control Latch 3** U260-18<15> low and clocking the system clocks once.

VERIFY RELOAD AFTER ZERO

4. Toggle system clocks once to cause the terminal count bit to go back to a de-asserted (low) state, and to cause the counter to be loaded (automatically) with the value of the last loaded test pattern, which should be 1.

5. Read and check that the terminal count bit is 0 and the counter contents is 1.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter to cause the counter to be loaded from its input latches.

   - Enable the load line by setting **TB Control Latch 3** U260-18 <15> low and clocking the system clocks once.
   - Load counter input latches with zero.
   - Disable the load line by setting **TB Control Latch 3** U260-18 <15> high and clocking the system clocks once.
   - Toggle the system clocks to get the zero value actually clocked into the counter from its input latches.

7. Read and check that the terminal count bit is 1 and the counter contents is 0.

8. Clock once more, then read the terminal count bit and the counter contents and check that they haven't changed.

Error Index G2311 Terminal count error just after counter was loaded with the test pattern.

Error Index G2312 Error in upper 10 bits of counter contents just after it was loaded with the test pattern.

Error Index G2313 Error in lower 16 bits of counter contents just after it was loaded.

Error Index G2314 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2315 Error in upper 10 bits of counter contents just after it was clocked to cause ripple count.

Error Index G2316 Error in lower 16 bits of counter contents just after it was clocked to cause ripple count.

Error Index G2317 Terminal count error after counter was clocked down to zero and then clocked one additional time.

Error Index G2318 Error in upper 10 bits of counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2319 Error in lower 16 bits of counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G231A Terminal count error just after counter was loaded with zero.

Error Index G231B Error in upper 10 bits of counter contents just after counter was loaded with zero.

Error Index G231C Error in lower 16 bits of counter contents just after counter was loaded with zero.

Error Index G231D Terminal count error just after counter was clocked once more after being loaded with zero.

Error Index G231E Error in upper 10 bits of counter contents just after counter was clocked once more after being loaded with zero.

Error Index G231F Error in lower 16 bits of counter contents just after counter was clocked once more after being loaded with zero.

Routine Name

1 Pst Record (Window 1 Post Record)

Overview

This test verifies the **Window 1 Postrecord Logic** (mainly, U275 WIND D CTR—16 bits, count-down; plus other logic) <19> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Stability at zero—A check to see if the counter remains unaffected after it has counted down to zero, and is clocked once more.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into two bytes and written byte-by-byte least-significant-byte first; to do a read, the counter is read least-significant-byte first, and then the two bytes reassembled into one 16-bit value.
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the timebase counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Window Rate Generator** U275-22 SYSTEM_RESET(L) <19> is set low.
   - **Pretrigger Generator** U239C-10 RESTART(L) <19> is set low. This signal and its complement **Window 2 Post Record Logic** U278-3 RESTART(H) <19> control the clock and/or load control lines of most time base counters.
   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up the hardware so that this counter can be clocked even when U278-3 RESTART(H) <19> is de-asserted (low).

   - Write a value to the location of the **Destination Address Generator** Control Register 2 (U166 <20>) such that bit D2 (adv1) is set high, which causes **Window 1 Post Record Logic** U294-4 ADV1(L) to go low.

VERIFY WRITABILITY/READABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 16 times). The test pattern starts out the first time as 8000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 16 patterns is: 8000*hex*, 4000*hex*, 2000*hex*, 1000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.
   - Toggle the system clocks once to get the test pattern actually clocked into the counter from its input latches.
   - Disable the load line (**Window 1 Post Record Logic** U275-14 WIND_POSTRECORD_LOAD(L)) by setting **TB Control Latch 0** U240-19 SET_RESTART(L):H <15> and **TB Control Latch 0** U221B-4 CLR_RESTART(L):L <15>, which causes **Main Rate Generator** U239D-12 RESTART(L) to go high.
   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.
   - Toggle the system clocks once to cause a ripple count.
   - Read and check the terminal count bit. It should be 0 except for the very last pass; then it it should be 1 (indicates a successful count to zero).
   - Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.
   - Shift the test pattern right one bit.
   - If not the last pass, then enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go low.

VERIFY STABILITY AT ZERO

4. Toggle system clocks once to see if it has any effect. It should have none.

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 1 (0), as previously explained.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) <18>to go low.
   - Load counter input latches with zero.
   - Toggle the system clocks once to actually load the counter from its input latch.

7. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G2321 Terminal count error just after counter was loaded with the test pattern.

Error Index G2322 Error in counter contents just after it was loaded with the test pattern.

Error Index G2323 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2324 Error in counter contents just after it was clocked to cause ripple count DURING THE LAST PASS of the writability/countability part of this test.

Error Index G2325 Error in counter contents just after it was clocked to cause ripple count on ALL BUT THE LAST PASS of the writability/countability part of this test.

Error Index G2326 Terminal count error after counter was clocked down to zero and then clocked one additional time.

Error Index G2327 Error in counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2328 Terminal count error just after counter was loaded with zero.

Error Index G2329 Error in counter contents just after counter was loaded with zero.

Routine Name

1 Position (Window 1 Position)

Overview

This test verifies the **Window 1 Position Logic** (mainly, U275 WIND E CTR—26 bits, count-down; plus other logic) <19> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Stability at zero—A check to see if the counter remains unaffected after it has counted down to zero, and is clocked once more.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into four bytes and written byte-by-byte least-significant-byte first; to do a read, the counter is read least-significant-byte first, and then the four bytes reassembled into one 26-bit value (six MSBs masked out).
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the timebase counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Window Rate Generator** U275-22 SYSTEM_RESET(L) <19> is set low, and so **Trigger Select Logic** U246C-5 WINDOW_RATE is high.
   - **Pretrigger Generator** U239C-10 RESTART(L) <19> is set low. This signal and its complement **Window 1 Post Record Logic** U278-3 RESTART(H) <19> control the clock and/or load control lines of most time base counters.

- **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up the hardware so that this load line **Window 2 Position Logic** U288A-3 WIND_POSITION_LOAD(L) can be disabled (set high) when **Main Postrecord Logic** U239D-12 RESTART(L) <18> is de-asserted (high).

   - Set **TB Control Latch 3** U260-16 WSET(H):H <15>. This guarantees that **Trigger Select Logic** U256-8 <18> will be low.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 26 times). The test pattern starts out the first time as 2000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 26 patterns is: 2000000*hex*, 1000000*hex*, 800000*hex*, 400000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.
   - Toggle the system clocks once to get the test pattern actually clocked into the counter from its input latches.
   - Set **TB Control Latch 0** U240-19 SET_RESTART(L):H <15> and **TB Control Latch 0** U221B-4 CLR_RESTART(L):L <15>, which causes **Main Rate Generator** U239D-12 RESTART(L) to go high.
   - Disable the load signal (**Window 1 Position Logic** U275-10 WIND_POSITION_LOAD(L) <19>) by toggling the system clocks once so that **Trigger Select Logic** U255A-3 <18> is clocked, which causes WIND_POSITION_LOAD(L) to go high.
   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.
   - Toggle the system clocks once to cause a ripple count.
   - Read and check the terminal count bit. It should be 0 except for the very last pass; then it it should be 1 (indicates a successful count to zero).
   - Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.
   - Shift the test pattern right one bit.

- If not the last pass, then enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L <20>, and U221B-4 CLR_RESTART(L):H <20>, which causes **Pretrigger Generator** U239D-12 RESTART(L) <18> to go low. This sets **Trigger Select Logic** U255A-6 low <18>so that WIND_POSITION_LOAD(L) goes low.

VERIFY STABILITY AT ZERO

4. Toggle system clocks once to see if it has any effect. It should have none.

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 1 (0), as previously explained.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L <20>, and U221B-4 CLR_RESTART(L):H <20>, which causes **Pretrigger Generator** U239D-12 RESTART(L) <18> to go low. This sets **Trigger Select Logic** U255A-6 low <18>so that WIND_POSITION_LOAD(L) goes low.

   - Load counter input latches with zero.

   - Toggle the system clocks once to actually load the counter from its input latch.

7. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G2331 Terminal count error just after counter was loaded with the test pattern.

Error Index G2332 Error in the upper 10 bits of the counter contents just after it was loaded with the test pattern.

Error Index G2333 Error in the lower 16 bits of the counter contents just after it was loaded.

Error Index G2334 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2335 Error in upper 10 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G2336 Error in lower 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G2337 Terminal count error after counter was clocked down to zero and then clocked one additional time.

Error Index G2338 Error in the upper 10 bits of the counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2339 Error in the lower 16 bits of the counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G233A Terminal count error just after counter was loaded with zero.

Error Index G233B Error in upper 10 bits of counter contents just after counter was loaded with zero.

Error Index G233C Error in upper 10 bits of counter contents just after counter was loaded with zero.

Routine Name

2 Pst Record (Window 2 Post Record)

Overview

This test verifies the **Window 2 Postrecord Logic** (mainly, U264 MAIN D CTR—16 bits, count-down; plus other logic) <18> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Stability at zero—A check to see if the counter remains unaffected after it has counted down to zero, and is clocked once more.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into two bytes and written byte-by-byte least-significant-byte first; to do a read, the counter is read least-significant-byte first, and then the two bytes reassembled into one 16-bit value.
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the time base counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Main Rate Generator** U264-22 SYSTEM_RESET(L) <18> is set low.
   - **Main Postrecord Logic** U239D-12 RESTART(L) <18> is set low. This signal and its complement **Window 2 Post Record Logic** U287-3 RESTART(H) <18> control the clock and/or load control lines of most timebase counters.
   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up the hardware so that this counter can be clocked even when **Window 2 Post Record Logic** U287-3 RESTART(H) <18> is de-asserted (low).

   - Write a value to the location of the **Destination Address Generator** Control Register 2 (U166 <20>) such that bit D3 (adv2) is set high, which causes **Window 2 Post Record Logic** U294-17 ADV2(L) to go low.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 16 times). The test pattern starts out the first time as 8000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 16 patterns is: 8000*hex*, 4000*hex*, 2000*hex*, 1000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.
   - Toggle the system clocks once to get the test pattern actually clocked into the counter from its input latches.
   - Disable the load line (**Window 2 Post Record Logic** U264-14 WIND_POST_RECORD_LOAD(L)) by setting **TB Control 0** U240-19 SET_RESTART(L):H <20> and **TB Control 0** U221B-4 CLR_RESTART(L):L <20>, which causes **Main Postrecord Logic** U239D-12 RESTART(L) <18> to go high.
   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.
   - Toggle the system clocks once to cause a ripple count.
   - Read and check the terminal count bit. It should be 0 except for the very last pass; then it it should be 1 (indicates a successful count to zero).
   - Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex*, then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.
   - Shift the test pattern right one bit.
   - If not the last pass, then enable the load line (**Window 2 Post Record Logic** U264-14 WIND_POSTRECORD_LOAD(L)) by setting **TB Control 0** U240-19 SET_RESTART(L):L <20> and **TB Control 0** U221B-4 CLR_RESTART(L):H <20>, which causes **Main Postrecord Logic** U239D-12 RESTART(L) <18> to go low.

VERIFY STABILITY AT ZERO

4. Toggle system clocks once to see if it has any effect. It should have none..

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 1 (0), as previously explained.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load line (**Window 2 Post Record Logic** U264-14 WIND_POST_RECORD_LOAD(L)) by setting **Restart Control** U240-19 SET_RESTART(L):L<20> and U240-2 CLR_RESTART(H):L<20>, which causes **Main Postrecord Logic** U239D-12 RESTART(L) <18> to go low.

   - Load counter input latches with zero.

   - Toggle the system clocks once to actually load the counter from its input latch.

7. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G2341 Terminal count error just after counter was loaded with the test pattern.

Error Index G2342 Error in counter contents just after it was loaded with the test pattern.

Error Index G2343 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2344 Error in counter contents just after it was clocked to cause ripple count DURING THE LAST PASS of the writability/countability part of this test.

Error Index G2345 Error in counter contents just after it was clocked to cause ripple count on ALL BUT THE LAST PASS of the writability/countability part of this test.

Error Index G2346 Terminal count error after counter was clocked down to zero and then clocked one additional time.

Error Index G2347 Error in counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2348 Terminal count error just after counter was loaded with zero.

Error Index G2349 Error in counter contents just after counter was loaded with zero.

Routine Name

2 Position (Window 2 Position)

Overview

This test verifies the **Window 2 Position Logic** (mainly, U264 MAIN E CTR—26 bits, count-down; plus other logic) <18> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Stability at zero—A check to see if the counter remains unaffected after it has counted down to zero, and is clocked once more.
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into four bytes and written byte-by-byte least-significant-byte first; to do a read, the counter is read least-significant-byte first, and then the four bytes reassembled into one 26-bit value (six MSBs masked out).
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the time base counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Window Rate Generator** U275-22 SYSTEM_RESET(L) <19> is set low, and so Trigger Select Logic U246C-5 WINDOW_RATE is high.
   - **Main Postrecord Logic** U239D-12 RESTART(L) <18> is set low. This signal and its complement **Window 2 Post Record Logic** U287-3 RESTART(H) <18> control the clock and/or load control lines of most timebase counters.

- **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up the hardware so that this load line **Window 2 Position Logic** U288A-3 WIND_POSITION_LOAD(L) can be disabled (set high) when **Main Postrecord Logic** U239D-12 RESTART(L) <18> is de-asserted (high).

   - Set **TB Control Latch 3** U260-16 WSET(H):H <15>. This guarantees that **Trigger Select Logic** U256-8 <18> will be low.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 26 times). The test pattern starts out the first time as 2000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 26 patterns is: 2000000*hex*, 1000000*hex*, 800000*hex*, 400000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.
   - Toggle the system clocks once to get the test pattern actually clocked into the counter from its input latches.
   - Set **TB Control Latch 0** U240-19 SET_RESTART(L):H <20> and **TB Control Latch 0** U221B-4 CLR_RESTART(H):L <20>, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go high.
   - Disable the load signal (**Window 2 Position Logic** U264-10 WIND_POSITION_LOAD(L)) by toggling the system clocks once so that **Trigger Select Logic** U255A-3 <18> is clocked, which causes WIND_POSITION_LOAD(L) to go high.
   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.
   - Toggle the system clocks once to cause a ripple count.
   - Read and check the terminal count bit. It should be 0 except for the very last pass; then it it should be 1 (indicates a successful count to zero).
   - Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.
   - Shift the test pattern right one bit.

- If not the last pass, then enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go low.

VERIFY STABILITY AT ZERO

4. Toggle system clocks once to see if it has any effect. It should have none.

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 1 (0), as previously explained.

VERIFY LOAD ZERO

6. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load signal by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H, which causes **Main Postrecord Logic** U239D-12 RESTART(L) to go low.

   - Load counter input latches with zero.

   - Toggle the system clocks once to actually load the counter from its input latch.

7. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G2351 Terminal count error just after counter was loaded with the test pattern.

Error Index G2352 Error in the upper 10 bits of the counter contents just after it was loaded with the test pattern.

Error Index G2353 Error in the lower 16 bits of the counter contents just after it was loaded.

Error Index G2354 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G2355 Error in upper 10 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G2356 Error in lower 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G2357 Terminal count error after counter was clocked down to zero and then clocked one additional time

Error Index G2358 Error in the upper 10 bits of the counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G2359 Error in the lower 16 bits of the counter contents after counter was clocked down to zero and then clocked one additional time.

Error Index G235A Terminal count error just after counter was loaded with zero.

Error Index G235B Error in upper 10 bits of counter contents just after counter was loaded with zero.

Error Index G235C Error in upper 10 bits of counter contents just after counter was loaded with zero.

Routine Name

No Trigger

Overview

This test verifies the **No Window Trig Detect** circuitry <19> on the Time Base board. It checks the circuitry for proper operation under all of its normal operating conditions. It does this by exercising the circuitry in several distinct functional cycles (called scenarios), each of which simulates a sequence of input combinations which can occur during normal operation.

This test supports two different versions of the Time Base board, i.e., KA-8665-01 and K-8665-02. The circuitry primarily covered for the KA-8665-01 version is U286-8,11,13, U248A, U247A, U257A. The circuitry primarily covered for the K-8665-02 version is U295, U286-8,11,13, U246D, U247A, U248A, U257A. The different versions are tested by two different and distinct algorithms. The board version is detected by reading **Timebase Status Readback Buffer** U241-3 <15>. If this bit is in the low state, then the test algorithm for the KA-8665-01 version is run; if it is in the high state, then then the algorithm for the K-8665-02 is run.

The only thing that these algorithms have in common is their preliminary hardware initialization; however, because of the hardware differences, the list of SIGNIFICANT signals for the two algorithms are different. Consequently, the descriptions of these two algorithms are separate, except for the introductory remarks at the beginning the the description section.

Description

There are some basic characteristics of how this test is executed which should be noted.

- This test makes use of the **Main Postrecord Logic** circuitry (U264 MAIN C CTR) <18> to control the state of **Main Postrecord Logic** U264-29 MAIN_POSTRECORD_TC. Details of how the counter in this circuitry is written and read are found in the test description for this circuit block, **Post Record (G222X)**.

- Five signals are monitored by this test to determine proper operation. Two of these are read from the U264 (MAIN TB IC) <18>, TC (terminal count) Register—**Main Rate Generator**, terminal count bit (bit 1) and **Main Postrecord Logic**, terminal count bit (bit 2). The other three are read from the **Destination Address Generator** U166 <20>, I/O Register 0—MCG(L) (D3), WCG(L) (D4), and NOTRIG (L) (D6). These five signals are read by the processor and concatenated together to form the 5 LSBs of a single byte of information. It is this byte containing these 5 meaningful bits (the three MSBs are zeroed out) which is compared with a set of expected values to determine proper operation of the **No Window Trig Detect** circuitry. The format of these five bits is as follows:

  D7,D6,D5: unused (always zero)
  D4: MCG(L)
  D3: WCG(L)
  D2: NOTRIG(L)
  D1: MAIN_POSTRECORD_TC
  D0: MAIN_RATE_TC

- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles U67C-5 CLOCK_STEP(L) of **System Clock Control** <14>.

1. Read **Timebase Status Readback Buffer** U241-3 <15>. If it is in the low state then execute the KA-8665-01 algorithm. If it is in the high state, then execute the K-8665-02 algorithm.

KA-8665-01 ALGORITHM

2. Hardware is initialized so that the following significant signals are set to the following states:

   - **Main Rate Generator** U264-22 SYSTEM-RESET(L) <18> is set low.
   - **No Window Trig Detect** U247A-1 MAIN_POSTRECORD_TC <19> is set low.
   - **Window Trigger Logic** U258-3 WCG(L) <19> is set high.
   - **No Window Trig Detect** U257A-1 MCG(H) is set low.
   - **No Window Trig Detect** U257-6 NOTRIG(L) is set high.
   - **No Window Trig Detect** U257-5 NOTRIG(H) is set low.
   - **Window Trigger Logic** U258-1 RESTART(L) <19> is set low.
   - **Window Trigger Logic** U268C-9 WACQTRG(H) and U268C-10 WSET(H) are set low so that U268C-8 WIND_TRIG(H) is low.

   Based on this initialization state, the five bits that are monitored by the processor during this test will read as 1D*hex*:

D4: MCG(L):	1
D3: WCG(L):	1
D2: NOTRIG(L):	1
D1: MAIN_POSTRECORD_ TC:	0
D0: MAIN_RATE_TC:	1

3. Do Scenario#1: No window trigger occurs before MAIN_POSTRECORD_TC goes high.

   - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
   - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
   - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*. This read function is done by doing the following: reading U264 (MAIN Timebase IC) TC Register <18>; masking out all but the B & C counter bits; reading **Destination Address Generator** U166 <20> I/O Register 0; inverting the value read (DAG inverts the signals); masking out all but bits MCG(L), WCG(L), and NOTRIG(L); finally rearranging the

ordering of these five bits and "or-ing" them together such that their format matches that given at the beginning of this **Description** section.

- Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
- Read the five monitored bits and verify against the expected value.
- Toggle system clocks twice more to set up the clock for the **Main Postrecord Logic** counter.
- Read the five monitored bits and verify against the expected value.
- Four more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of seven pattern verifications against seven expected values. The seven expected values are the following:

  1D*hex*, 0D*hex*, 0F*hex*, 0B*hex*, 03*hex*, 03*hex*, 03*hex*

4. Hardware is reinitialized by repeating Step 2.
5. Do Scenario#2: Window trigger occurs after the main trigger and before MAIN_POSTRECORD_TC goes high.
   - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
   - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
   - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*.
   - Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
   - Read the five monitored bits and verify against the expected value.
   - Set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCG(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once.
   - Read the five monitored bits and verify against the expected value.
   - Three more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of six pattern verifications against six expected values. The six expected values are the following:

     1D*hex*, 0D*hex*, 05*hex*, 07*hex*, 07*hex*, 07*hex*
6. Hardware is reinitialized by repeating Step 2.

7. Do Scenario#3: Window trigger occurs before main trigger and before MAIN_POSTRECORD_TC goes high.

   - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
   - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
   - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*.
   - Set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once.
   - Read the five monitored bits and verify against the expected value.
   - Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
   - Read the five monitored bits and verify against the expected value.
   - Toggle the system clocks twice.
   - Read the five monitored bits and verify against the expected value.
   - Two more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of six pattern verifications against six expected values. The six expected values are the following:

     1D*hex*, 15*hex*, 05*hex*, 07*hex*, 07*hex*, 07*hex*

8. Hardware is reinitialized by repeating Step 2.

9. Do Scenario#4: Window Trigger occurs simultaneously with main trigger and before MAIN_POSTRECORD_TC goes high.

   - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
   - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
   - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*.
   - Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H, **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H, and then toggling system clocks once.
   - Read the five monitored bits and verify against the expected value.

- Toggle system clocks twice.
- Read the five monitored bits and verify against the expected value.
- Two more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of five pattern verifications against five expected values. The five expected values are the following:

  1D*hex*, 05*hex*, 07*hex*, 07*hex*, 07*hex*

10. Hardware is reinitialized by repeating Step 2.
11. Do Scenario#5: Window Trigger occurs after main trigger and simultaneously with MAIN_POSTRECORD_TC goes high.
    - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
    - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
    - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*.
    - Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
    - Read the five monitored bits and verify against the expected value.
    - Toggle system clocks once and then set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once more.
    - Read the five monitored bits and verify against the expected value.
    - Two more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of five pattern verifications against five expected values. The five expected values are the following:

      1D*hex*, 0D*hex*, 07*hex*, 03*hex*, 03*hex*

Error Index G2361 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#1.

Error Index G2362 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#2.

Error Index G2363 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#3.

Error Index G2364 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#4.

Error Index G2365 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#5.

K-8665-02 ALGORITHM

2. Hardware is initialized so that the following significant signals are set to the folowing states:

   - **No Window Trig Detect** U295-5 SYSTEM-RESET(L) <19> is set low.
   - **No Window Trig Detect** U295-2 TC_DELAY(L) is set high.
   - **No Window Trig Detect** U295-8 TOGGLE_CNTRL (H) is set low.
   - **No Window Trig Detect** U295-12 DONOTRIG(L) is set high.
   - **No Window Trig Detect** U286-13 WCGLATE(L) is set high.
   - **Main Postrecord Logic** U264-29 MAIN_POSTRECORD_TC <18> is set low.
   - **Window Trigger Logic** U258-3 WCG(L) <19> is set high.
   - **Window Trigger Logic** U258-2 WCG(H) <19> is set low.
   - **No Window Trig Detect** U257A-1 MCG(H) is set low.
   - **No Window Trig Detect** U257-6 NOTRIG(L) is set high.
   - **No Window Trig Detect** U257-5 NOTRIG(H) is set low.
   - **Window Trigger Logic** U258-1 RESTART(L) <19> is set low.
   - **Window Trigger Logic** U268C-9 WACQTRG(H) and U268C-10 WSET(H) are set low so that U268C-8 WIND_TRIG(H) is low.

   Based on this initialization state, the five bits that are monitored by the processor during this test will read as 1D*hex*:

D4: MCG(L):	1
D3: WCG(L):	1
D2: NOTRIG(L):	1
D1: MAIN_POSTRECORD_TC:	0
D0: MAIN_RATE_TC:	1

3. Do Scenario#1: No window trigger occurs before the delayed Main Post Record terminal count (**No Window Trig Detect** U295-12 DONOTRIG(L)), i.e., **Window Trigger Logic** U258-2 WCG (H) goes high after DONOTRIG(L) goes low. U295 internal signal TOGGLE(H) is initially set to high.

   - Set **No Window Trig Detect** U295 internal signal TOGGLE(H):H by setting U295-3 TOGGLE_CNTRL(H):H, toggling system clocks once, then setting TOGGLE_CNTRL(H):L.
   - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.

- Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.

- Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*. This read function is done by doing the following: reading U264 (MAIN Timebase IC) TC Register <18>; masking out all but the B and C counter bits; reading **Destination Address Generator** U166 <20> I/O Register 0; inverting the value read (**DAG** inverts the signals); masking out all but bits MCG(L), WCG(L), and NOTRIG(L); finally rearranging the ordering of these five bits and "or-ing" them together such that their format matches that given at the beginning of this **Description** section.

- Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.

- Read the five monitored bits and verify against the expected value.

- Toggle system clocks twice.

- Read the five monitored bits and verify against the expected value.

- Toggle system clocks six times.

- Read the five monitored bits and verify against the expected value.

- Toggle system clocks once.

- Read the five monitored bits and verify against the expected value.

- Set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once.

- Read the five monitored bits and verify against the expected value.

- Two more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of eight pattern verifications against eight expected values. The eight expected values are the following:

  1D*hex*, 0D*hex*, 0F*hex*, 0F*hex*, 0B*hex*, 03*hex*, 03*hex*, 03*hex*

4. Hardware is reinitialized by repeating Step 2.

5. Do Scenario#2: Window trigger occurs after main trigger and before the delayed Main Post Record terminal count (**No Window Trig Detect** U295-12 DONOTRIG(L)), i.e., **Window Trigger Logic** U258-2 WCG (H) goes high before U295-12 DONOTRIG(L) goes low. U295 internal signal TOGGLE(H) is initially set to high.

   - Set **No Window Trig Detect** U295 internal signal TOGGLE(H):H by setting U295-3 TOGGLE_CNTRL(H):H, toggling system clocks once, then setting TOGGLE_CNTRL(H):L.

- Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
- Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
- Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*.
- Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
- Read the five monitored bits and verify against the expected value.
- Toggle system clocks twice.
- Read the five monitored bits and verify against the expected value.
- Toggle system clocks four times.
- Set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once.
- Read the five monitored bits and verify against the expected value.
- Three more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of seven pattern verifications against seven expected values. The seven expected values are the following:

  1D*hex*, 0D*hex*, 0F*hex*, 07*hex*, 07*hex*, 07*hex*, 07*hex*

6. Hardware is reinitialized by repeating Step 2.
7. Do Scenario#3: Window trigger occurs after main trigger and simultaneously with delayed Main Post Record terminal count (**No Window Trig Detect** U295-12 DONOTRIG(L)), i.e., **Window Trigger Logic** U258-2 WCG (H) goes high at the same time (clock edge) as U295-12 DONOTRIG(L) goes low. U295 internal signal TOGGLE(H) is initially set to high.
   - Set **No Window Trig Detect** U295 internal signal TOGGLE(H):H by setting U295-3 TOGGLE_CNTRL(H):H, toggling system clocks once, then setting TOGGLE_CNTRL(H):L.
   - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
   - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
   - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*.

- Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
- Read the five monitored bits and verify against the expected value.
- Toggle system clocks twice.
- Read the five monitored bits and verify against the expected value.
- Toggle system clocks five times.
- Read the five monitored bits and verify against the expected value.
- Set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once.
- Read the five monitored bits and verify against the expected value.
- Three more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of eight pattern verifications against eight expected values. The eight expected values are the following:

  1D*hex*, 0D*hex*, 0F*hex*, 0F*hex*, 07*hex*, 03*hex*, 03*hex*, 03*hex*

8. Hardware is reinitialized by repeating Step 2.
9. Do Scenario#4: Window trigger occurs after main trigger and one clock step past what would be a simultaneous event with the delayed Main Post Record terminal count (**No Window Trig Detect** U295-12 DONOTRIG(L)) if the U295 internal signal TOGGLE(H) were high; however this scenario is to always run after scenario #3 (order dependency) and on entry to this scenario TOGGLE(H) is expected to be 0.
   - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
   - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
   - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*.
   - Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
   - Read the five monitored bits and verify against the expected value.
   - Toggle system clocks twice.
   - Read the five monitored bits and verify against the expected value.

- Toggle system clocks five times.
- Read the five monitored bits and verify against the expected value.
- Set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once.
- Read the five monitored bits and verify against the expected value.
- Four more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of nine pattern verifications against nine expected values. The nine expected values are the following:

  1D*hex*, 0D*hex*, 0F*hex*, 0F*hex*, 07*hex*, 07*hex*, 07*hex*, 07*hex*, 07*hex*

10. Hardware is reinitialized by repeating Step 2.

11. Do Scenario#5: With U295 internal signal TOGGLE(H) assumed to be low from previous execution of Scenarios #3 and #4, Window trigger occurs after main trigger and simultaneously with the delayed Main Post Record terminal count (**No Window Trig Detect** U295-12 DONOTRIG(L)).

    - Load **Main Postrecord Logic** U264 (MAIN C CTR) <18> with 1.
    - Set **Window Trigger Logic** U258-1 RESTART(L) <19> high by setting **TB Control 0** U240-19 SET_RESTART(L):H <15> and **TB Control 0** U221B-4 CLR_RESTART(L):L <15>.
    - Read the five monitored bits and verify that their combined states form the expected value of 1D*hex*
    - Set **No Window Trig Detect** U257A-1 MCG(H):H by setting **TB Control Latch 3** U260-17 MSET:H and toggling system clocks once.
    - Read the five monitored bits and verify against the expected value.
    - Toggle system clocks twice.
    - Read the five monitored bits and verify against the expected value.
    - Toggle system clocks five times.
    - Read the five monitored bits and verify against the expected value.
    - Toggle system clocks once.
    - Read the five monitored bits and verify against the expected value.
    - Toggle system clocks once.
    - Read the five monitored bits and verify against the expected value.
    - Set **Window Trigger Logic** U258-2 WCG(H):H and U258-3 WCH(L):L by setting **TB Control Latch 3** U260-16 WSET:H and toggling system clocks once.
    - Read the five monitored bits and verify against the expected value.
    - Three more times the system clocks are toggled once and then the pattern of the five monitored bits is verified against the expected value; so for this scenario, there is a total of 10 pattern verifications against 10 expected values. The 10 expected values are the following:

      1D*hex*, 0D*hex*, 0F*hex*, 0F*hex*, 07*hex*, 07*hex*, 07*hex*, 07*hex*, 07*hex*, 07*hex*

12. Hardware is reinitialized by repeating Step 2.

13. Do Scenario#6: Window trigger occurs after main trigger and simultaneously with the delayed Main Post Record terminal count (**No Window Trig Detect** U295-12 DONOTRIG(L)). The U295 internal signal TOGGLE(H) is expected to be low due to the previous execution of Scenarios#3-#5.

    - Repeat all of the steps of Scenario#3, except do not repeat the very first step of setting the U295 internal signal TOGGLE(H). Consequently, the list of expected values for this scenario are identical to those of Scenario#3.

Error Index G2361 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#1.

Error Index G2362 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#2.

Error Index G2363 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#3.

Error Index G2364 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#4.

Error Index G2365 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#5.

Error Index G2366 The values of the five bits monitored by the processor did not match the expected values during the execution of Scenario#6.

Routine Name

Accumulator

Overview

This test verifies a set of four operations of the **Acq Mem Data Accumulator** (U14, U15, U24, U25) <14> on the Time Base board:

- A set of serial write/read operations.
- Hold control verification (ACQ_DATA_HOLD(L)).
- Mode control verification (DIAG_SERIAL_MODE).
- Invalidate point control verification (INVALIDATE_POINT(H), INV_PT_HALF(H)).

Description

There are some basic characteristics of how this test is executed which should be noted.

- Note that this 16-bit register is actually made up of four 4-bit registers (U14, U15, U24, U25); serial operations are done by doing serial operations on these registers simultaneously. The input serial data (U14-2 SERIAL_DATA_LOW_IN, & U25-2 SERIAL_DATA_HIGH_IN) come from bits D7 and D15 of the data bus, respectively; the output serial data (U15-12 ACQ_SERIAL_DATA_MID_LOW & U24-12 ACQ_SERIAL_DATA_MID_HIGH) go to bits D0 and D8 of the data bus, respectively.

- Note, however, that the accumulator registers are loaded and read in tandem with U202 in **Time base Request Pipeline** <15> and U183 in **Vertical Source Sequencer** <15>, which means that failures there could affect this test. The output serial data, for example, first pass through these other registers and actually interface to the data bus via **Vertical Source Sequencer** U183-12 ACQ_SERIAL_DATA_LOW_OUT <15> & **Time base Request Pipeline** U202-12 ACQ_SERIAL_DATA_HIGH_OUT <15>.

- Note that INVALIDATE_POINT(H) and INV_PT_HALF(H) originate in **Chop Transition Blanking** <15>.

- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:

   - **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> is set low.

   - **Restart Control** U175A-6 RESTART(L) <20> is set low. This signal and its complement U175A-5 RESTART(H) <20> control the clock and/or load control lines of most time base counters.

   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

VERIFY SERIAL WRITABILITY/READABILITY

2. Set up accumulator for right shift serial mode.

   - Set **TB Control Latch 1** U251-14 INV_PT_RESET(L):L <15>, **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:L <15> , and **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H <15>.

3. Serially write and read the following values and verify that they match: AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, 5555*hex*, AAAA*hex*

VERIFY HOLD CONTROL

4. Load the accumulator with AAAA*hex*.

5. Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):L.

6. Load the accumulator with 5555*hex*.

7. Read and verify that the accumulator contents stayed at AAAA*hex*.

8. Load the accumulator with 5555*hex* again.

9. Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H.

10. Read and verify that the accumulator is 5555*hex*.

VERIFY MODE CONTROL

11. Load the accumulator with FFFF*hex*, time base tags all with 0 (**Timebase Request Pipeline** U202-12,13,14 MAIN_ID_TAG(L), WIND_1_ID_TAG(L), & WIND_2_ID_TAG(L) <15>), and **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L) <15> with 0.

12. Parallel load the accumulator with the time base tags.

    - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):L and toggle system clocks.

    - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H , set **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:H , and toggle system clocks twice.

13. Read the accumulator and mask out all bits except those three which represent the time base tag bits. If these bits are not all zeros, then report an error.

VERIFY INVALIDATE POINT CONTROL

14. Set **Chop Controller & Plug-in Sequencer** SEQ_CLK (H) <15> to a high state.

    - Set **TB Control Latch 4** U261-15 CTR_MUX_CNTRL:H <15>, U261-(12-14) CTR_ADDR0-2 <15> all low, and **TB Control Latch 5** U250-15 SINGLE_SHOT_CHOP(L):H <15>.

- Toggle system clocks twice. This clocks a high state through **Chop Controller & Plug-in Sequencer** U201-7 <15> to **Chop Transition Blanking** U200A-5 <15>.

15. Load the accumulator as in step 11.

16. Assert hold control, and cause **Acq Mem Data Accumulator** U65C-9 INV_PT_HALF(H) <14> & **Acq Mem Data Accumulator** U67A-1 INVALIDATE_POINT(H) <14> to go high.

    - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):L.
    - Set **TB Control Latch 1** U251-14 INV_PT_RESET(L):H.
    - Toggle system clocks three times. This clocks the high at **Chop Transition Blanking** U200A-5 <15> through U210 & U150 to set INV_PT_HALF(H) & INVALIDATE_POINT(H) high.
    - Set **TB Control Latch 1** U251-14 INV_PT_RESET(L):L.
    - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H.

17. Read the accumulator and if the contents is not F00F*hex* then report an error.

Error Index G3111 The value read back during the serial write/read operation did not match the value written.

Error Index G3112 The hold control did not work.

Error Index G3113 The mode control did not work.

Error Index G3114 The mechanism for creating an invalid point indication did not work.

Routine Name

Input Contrl (Input Control)

Overview

This test primarily verifies the **Acq Mem Control Logic** <13> on the Time Base board. Specifically, it verifies the ability of this and related hardware to pass data successfully from the **Acq Mem Data Accumulator** <14> into the **Acq Mem Even Bank & Acq Mem Odd Bank** <13> . This test does not verify the hardware which controls the transfer of data out of the memory.

Description

There are some basic characteristics of how this test is executed which should be noted.

- The Acquisition Memory normally operates in a circular manner such that absolute addresses are unimportant. When it is filling with acquired waveform points it simply wraps around when it counts up to its maximum address. In normal operation the processor does not access this memory. In order to cause data to be written into a specific address range, the processor must first do a write operation to a specific address using the diagnostic addressing facilities (part of **Acq Mem Address Generation** <13>). Any subsequent writes immediately following this will begin with the very next address. This is the method used in this test.

- Normally the processor memory mapping is programmed so that the Acquisition Memory is located at address location 6000*hex*. Because of limitations in the addressing capabilities of the processor (it runs in small model), only 2k bytes of the 32k byte address space (16k 16-bit words) is accessible in this normal configuration. This is generally no problem because the processor typically does not need access during normal operation. In order to gain full processor access during this test the memory mapping is temporarily changed. The base address of the Acquisition Memory is moved to 40000*hex*, so the address range is 40000*hex* to 47FFF*hex*. This is the address range that is reported in the test results.

- The successful execution of this test is dependent on the proper operation of several other hardware blocks: **Acq Mem Address Generation**<13>, the Acquisition Memory itself, **Acq Mem Bank Select** <13>, **Initial Wfm Id Encoder Table** <14>, **Acq Mem Data Accumulator** <14>, **Acq Mem Data Input Latch** <14>, **Acquisition Request Control** <15>, and **Last Request Control** <15>.

- The entire address space of the memory is not tested here. This is done by the **Address/Data (G314X)** Acquisition Memory tests. Rather, this test checks the hardware's ability to write data into a select group of 12 address ranges. Each range is six words long and spans the transition point where the address count has all LSBs go from 1's to 0's and its MSB from 0 to 1, e.g., when the address goes from 3FF*hex* to 400*hex* (0011 1111 1111 to 0100 0000 0000).

- The **Acq Mem Data Accumulator** <14> is used to transfer data into the Acquisition Memory during this test. However it should be noted that the bit format of the data gets transposed during this transfer. In order for specific bit patterns to be written into the memory as desired, the patterns loaded into the accumulator will be such that after they are transposed by the hardware, they come out as desired for input to the memory. Following is a list showing what values are loaded into the accumulator in order to get the desired test patterns into the memory:

Accumulator Value	Acquisition Memory Value
4555*hex*	AAAA*hex*
7666*hex*	CCCC*hex*
9787*hex*	F0F0*hex*
E807*hex*	FF00*hex*
BAAA*hex*	5555*hex*

Details of how the accumulator is written and read are found in the test description **Accumulator (G312X)**.

- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:

   - The terminal counts for the time base counter IC's (U264 <18> and U275 <19>) are low (all counters loaded with FFFF*hex*).

   - **Final Wfm Id Encoding Table** U174-1 SYSTEM_RESET(L) <20> is set low. This is the select control line for **Acq Mem Control Logic** U174 <13>.

   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

   - **Last Request Control** U65C-10 INVALIDATE_POINT(H) and U65C-9 INV_PT_HALF(H) <15> are set low.

   - **Acq Mem Data Accumulator** U220A-1 RESTART(L) <15> is set low.

2. Set **Acq Mem Control Logic** U140D-13 VALID_ACQ_REQUEST(H) <13> to be always high by filling the **Initial Wfm Id Encoding Table** <14> with 08*hex*.

3. Set up hardware in **Last Request Control** <15> so that U210-10 L_LAST_REQUEST(H) can later be set high easily.

   - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H <15>.

   - Write 01*hex* to **Last Request Control** U202 <15> (written serially via the **Acq Mem Data Accumulator** <14>) such that pin 15 is set high.

- Set ACQ_DATA_HOLD(L):L.
- Toggle system clocks twice.
- Set **TB Control Latch 1** U251-12 INV_PT_RESET(L):H <15> and toggle system clocks once. This should cause **Last Request Control** U210-7 <15> to go high.
- Set **Last Request Control** U220A-1 RESTART(L):H <15> by setting **TB Control Latch 0** U240-19 SET_RESTART(L):H <15> and U221B-4 CLR_RESTART(L):L.
- Set **TB Control Latch 1** U251-12 INV_PT_RESET(L):L <15>. This should cause U220A-5 to go high.

4. Set up for easy control of **Acq Mem Control Logic** U140D-12 ACQ_REQUEST <15>.
   - Set **TB Control Latch 3** U260-15 FORCE_ACQ_REQ(L):H <15>.
   - Set **TB Control Latch 5** U250-15 SINGLE_SHOT_CHOP(L):H, U250-16 LEFT_SEL(H):H,U250-17 CENTER_SEL(H):L,U250-18 RIGHT_SEL(H):L,U250-19 CAL_SEL(H):L <15>.
   - Toggle system clocks three times. This should propagate a low signal from Vertical Source Sequencer U184 -11 REQ(L) <15> to U172B-4, which in turn will allow the state of **Acquisition Request Control** U212-9 <15> and **Acq Mem Control Logic** U140D-12 ACQ_REQUEST(H) <13> to be changed by simply setting **TB Control Latch 3** FORCE_ACQ_REQ(L) <15> and toggling the system clocks.
5. Write the first test vector (4555*hex*) into the accumulator.
   - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H <15>.
   - Write the value serially into the accumulator.
   - Set U240-6 ACQ_DATA_HOLD(L):L.
6. This long step represents the core of this test. At this point all of the preliminary operations have been done. It repetitively checks the ability of the hardware to fill the Acquisition Memory for a set of 12 address ranges. This step attempts to write six consecutive memory locations with a test pattern and then read back the results. It sets up the address range to be tested each time by doing two operations. First it starts at two locations prior to the target range and writes FFFF*hex* to nine locations. Then it sets up the **Acq Mem Address Generation** <13> hardware for specific address by writing zero to the address location which immediately precedes the six target locations. This way the target address range is initially filled with a known value, and the address put out by the address generator is set to the desired value. So after these actions, but before the attempt is made to fill the range with the test pattern, the nine locations bracketing the memory range should look like the following:

   FFFF*hex*

0000*hex* <--- address set by writing to here
FFFF*hex* <--- first location of address range to be written
FFFF*hex*
FFFF*hex*
FFFF*hex*
FFFF*hex*
FFFF*hex* <--- last location of address range to be written
FFFF*hex*

After the accumulator is filled with a test pattern, for example, 4555*hex*, and the hardware is manipulated to cause the range to be written with the transposed pattern, then the memory range (in this case) should look like the following:

FFFF*hex*
0000*hex*
AAAA*hex*
AAAA*hex*
AAAA*hex*
AAAA*hex*
AAAA*hex*
AAAA*hex*
FFFF*hex*

The entire nine-location address range is read back so as to back sure that the hardware did not erroneously fill any adjacent locations with the test pattern.

In addition to reporting the address, errors are distinguished by assigning one error index to each address range for easier identification of where the error occurred. Also there are fewer test patterns used than there are address ranges, so the series of patterns used is traversed twice. Following is a table showing the correlation of error index, test pattern, Acquisition Memory address range, and processor absolute memory address range for the 12 different iterations of this test step for the 12 different address ranges. The address ranges listed here are nine locations long, so they include the entire address space that is read back and checked. This list also indicates the order in which the address ranges are tested in this step.

Error Index	Test Pattern	Acq. Mem. Address Range	Absolute Address Range
G3121	AAAA	7FF6-0006	47FF6-40006
G3122	CCCC	3FF6-4006	43FF6-44006
G3123	F0F0	1FF6-2006	41FF6-42006
G3124	FF00	0FF6-1006	40FF6-41006
G3125	5555	07F6-0806	407F6-40806
G3126	AAAA	03F6-0406	403F6-40406
G3127	AAAA	01F6-0206	401F6-40206
G3128	CCCC	00F6-0106	400F6-40106
G3129	F0F0	0076-0086	40076-40086
G312A	FF00	0036-0046	40036-40046
G312B	5555	0016-0026	40016-40026
G312C	AAAA	0006-0016	40006-40016

- Turn on the system clocks to free run (20 MHz) by setting **TB Control Latch 0** U240-12 CLOCK_SOURCE_SELECT <15> to a low state.
- Change base address to 40000*hex*.
- Write FFFF*hex* to the nine locations of the the range.
- Write zero to the location immediately preceding the first address to be written with the test pattern, so as to set up the address.
- Change back to normal base address.
- Turn off the system clocks by setting **TB Control Latch 0** U240-12 CLOCK_SOURCE_SELECT <15> to a high state.
- Set **Final Wfm Id Encoding Table** U174-1 SYSTEM_RESET(L):H <20> by setting **TB Control Latch 3** U260-18 <15> high and clocking the system clocks once.
- Toggle the system clocks three times. This should cause **Acq Mem Control Logic** U133D-11 DUMMY_CYLE(L) <13>to toggle low and high (on 1st and 2nd clocks), **Acq Mem Address Generation** U93E-10 <13> to go low (3rd clock), and **Acq Mem Address Generation** U160-15 ACQ_MEM_R/W_MODE to go low to put the memory in write mode.
- Set ACQ_REQUEST(H):H by setting **TB Control Latch 3** U260-15 FORCE_ACQ_REQ(L):L <15> and toggling system clocks once.
- Toggle system clocks five times. This should cause the test pattern that resides in the accumulator to be written six times into the memory.
- Set ACQ_REQUEST(H):L by setting **TB Control Latch 3** U260-15 FORCE_ACQ_REQ(L):H <15> and toggling system clocks once.
- Toggle system clocks twice more to cause **Acq Mem Control Logic** U161B-6 ACQ_MEM_REQ(L) and U163-13 ACQ_DATA_BUFFER_ENABLE(L) <13> to go high.

- Set SYSTEM_RESET(L) (**Acq Mem Control Logic** U141F-13 <13> and **Last Request Control** U220A-1 <15>) to the low state again to make sure that ACQ_MEM_R/W(L)_MODE goes high again, to allow reading of the memory locations. This is dones by setting **TB Control Latch 3** U260-18 <15> low and toggling system clocks twice.
- Set the system clocks to free-run mode.
- Change base address to 40000*hex*.
- Read the address range via the diagnostic read/write facilities and compare the results for the nine locations with the expected values.
- Change base address back to normal.
- Set the system clocks to single-step mode.
- Write next test pattern into accumulator.

7. Simulate the final operations that would conclude a normal data acquisition cycle. This is not used for gathering test data for diagnostics, but may be useful for trouble-shooting the hardware (as in a terse loop).
   - Set ACQ_REQUEST(H):H by setting **TB Control Latch 3** U260-15 FORCE_ACQ_REQ(L):L<15> and toggling system clocks once.
   - Toggle system clocks once more to get the signal propagated through **Acq Mem Control Logic** U150 <13>.
   - Set ACQ_REQUEST(H):L by setting **TB Control Latch 3** U260-15 FORCE_ACQ_REQ(L):H <15> and toggling system clocks once.
   - Set **Acq Mem Control Logic** U174-10,13 L_LAST_REQUEST(H):H <13> by setting **TB Control Latch 1** INV_PT_RESET(L):H <15> and toggling system clocks once.

Error Index G3121 Value read back from address range 7FF6-0006*hex* did not match expected value.

Error Index G3122 Value read back from address range 3FF6-4006*hex* did not match expected value.

Error Index G3123 Value read back from address range 1FF6-2006*hex* did not match expected value.

Error Index G3124 Value read back from address range 0FF6-1006*hex* did not match expected value.

Error Index G3125 Value read back from address range 07F6-0806*hex* did not match expected value.

Error Index G3126 Value read back from address range 03F6-0406*hex* did not match expected value.

Error Index G3127 Value read back from address range 01F6-0206*hex* did not match expected value.

Error Index G3128 Value read back from address range 00F6-0106*hex* did not match expected value.

Error Index G3129 Value read back from address range 0076-0086*hex* did not match expected value.

Error Index G312A Value read back from address range 0036-0046*hex* did not match expected value.

Error Index G312B Value read back from address range 0016-0026*hex* did not match expected value.

Error Index G312C Value read back from address range 0006-0016*hex* did not match expected value.

Routine Name

Data Lines

Overview

This test verifies the data line integrity to **Acq Mem Even Bank** and **Acq Mem Odd Bank** <13> by performing a "walking one's" test on one acquisition memory location. The test utilizes the **Acq Mem Control Logic** <13>, Acq Mem Bank Select <13>, and **Acq Mem Address Generation** <13> circuits to access the acquisition memory.

Description

1. Force acquisition memory hardware to a known state.

   - Set **TB Control Latch** 3 U212-12 SYSTEM_RESET(L):L <15> and put the system clocks into free-run mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):L <14>.

   - Initialize the **Initial Wfm Id Encoder Table** <14> to prevent any acquisition memory requests (i.e., force U140D-13 VALIDATE_ACQ_REQUEST(H) low) by writing 0*hex* to all RAM locations in U22 and U32.

2. Perform a "walking one's" test on **Acq Mem Even Bank** <13> address 40000*hex*. Terminate test if any verify operation fails.

   - Write the pattern 0000*hex* to acquisition memory address 40000*hex* (U10, U20, U30, U40). Read the same address and verify that it was 0000*hex*. Continue this write/read/ verify sequence with the patterns 0001*hex*, 0002*hex*, 0004*hex*, 0008*hex*, 0010*hex*, 0020*hex*, 0040*hex*, 0080*hex*, 0100*hex*, 0200*hex*, 0400*hex*, 0800*hex*, 1000*hex*, 2000hex, 4000*hex*, and 8000*hex*.

Error Index G3131

The pattern read from **Acq Memory Even Bank** <13> address 40000*hex* was not the same pattern that was written. The correspondence of the data bits to the acquisition memory devices is shown in the following table:

D0-D3	D4-D7	D8-D11	D12-D15
U10	U20	U30	U40

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of **Acq Mem Even Bank <13>** and **Acq Mem Odd Bank <13>** by performing a RAM test on all acquisition memory locations. The test utilizes the **Acq Mem Control Logic <13>, Acq Mem Bank Select <13>,** and **Acq Mem Address Generation <13>** circuits to access the acquisition memory.

Description

1. Force acquisition memory hardware to a known state.

   - Set **TB Control Latch 3** U212-12 SYSTEM_RESET(L):L <15> and put the system clocks into free-run mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):L <14>.

   - Initialize the **Initial Wfm Id Encoder Table** <14> to prevent any acquisition memory requests (i.e., force U140D-13 VALIDATE_ACQ_REQUEST(H) low) by writing 0*hex* to all RAM locations in U22 and U32.

2. Verify entire acquisition RAM (address range 40000-43FFF*hex*). Terminate test if any verify operation fails.

   - Fill address range 40000-43FFF*hex* with the pattern AAAA*hex*.

   - Read and verify address 40000*hex* for AAAA*hex*. If so, write CCCChex to address 40000*hex*. Increment the address and continue this read/verify/write sequence until 43FFF*hex*.

   - Repeat the read/verify/write sequence, starting again at address 40000*hex*, for patterns CCCC*hex*, F0F0*hex*, 5555*hex*, and AAAA*hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

   - Make one last read/verify pass for the pattern AAAA*hex* starting at address 40000*hex*.

Error Index G3141

The pattern read from an acquisition memory location was not the same pattern written to that memory location. Since there are four banks of acquisition memory, every address modulo 4 represents an access to a different set of acquisition memory RAM devices. Furthermore, acquisition memory is accessed on a word-wide basis only. The correspondence of the address and data bits to acquisition RAM devices is shown in the following table:

Modulo 4 Address	D0-D3	D4-D7	D8-D11	D12-D15	
4XXX0*hex*, 4XXX8*hex*	U10	U20	U30	U40	(Bank 0)
4XXX2*hex*, 4XXXA*hex*	U50	U60	U70	U80	(Bank 1)
4XXX4*hex*, 4XXXC*hex*	U11	U21	U31	U41	(Bank 2)
4XXX6*hex*, 4XXXE*hex*	U51	U61	U71	U81	(Bank 3)

X = don't care

Routine Name Data Lines

Overview This test verifies the data lines going to the **Initial Wfm Id Encoder Table** (U72, U62, U63, U73A, U22, U32) <14> on the Time Base board by doing a "walking 1's" test on its first address location. The **Acq Mem Data Accumulator** <14> is used to read out the contents of the table.

Description There are some basic characteristics of how this test is executed which should be noted.

- The 16-bit accumulator, which is used in this test, is actually made up of four 4-bit registers (U14, U15, U24, U25); serial operations are done by doing serial operations on these registers simultaneously. The input serial data (U14-2 SERIAL_DATA_LOW_IN, and U25-2 SERIAL_DATA_HIGH_IN) come from bits D7 and D15 of the data bus, respectively; the output serial data (U15-12 ACQ_SERIAL_DATA_LOW_OUT and U24-12 ACQ_SERIAL_DATA_HI_OUT) go to bits D0 and D8 of the data bus, respectively.

- **Acq Mem Data Accumulator** U65C-10 INVALIDATE_POINT(H) and U65C-9 INV_PT_HALF(H) <14> originate in **Chop Transition Blanking** <15>.

- The accumulator registers are loaded and read in tandem with U202 in **Timebase Request Pipeline** <15> and U183 in **Vertical Source Sequencer** <15>, which means that failures there could affect this test.

- Writing to this table is straightforward—**Initial Wfm Id Encoder Table** U62-1/U63-1 SYSTEM_RESET(L) <14> is held low and a value written to the table addresses. Reading a table location, however, is not straight-forward—it involves setting SYSTEM_RESET(L) high, clocking the table output through a latch (U12) and into the accumulator, and then reading the accumulator and stripping out the table output bits from all the rest. One limitation on the coverage is that only three of the four table output bits can be read back; the MSB (ID 3) cannot be read via the accumulator. So even though the basic "walking 1's" test done on this table is effectively the same as a simple RAM test, the control aspects of achieving this are more complex.

- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:

   - **Initial Wfm Id Encoder Table** U62-1/U63-1 SYSTEM_RESET(L) <15> is set low.

   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up accumulator for right shift serial mode.

- Set **TB Control Latch 1** U251-14 INV_PT_RESET(L):L, **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:L, and **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H.

3. Set **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L):L <15>.

   - Write 00*hex* to **Vertical Source Sequencer** U183 <15>; at the same time, 0000*hex* is written to the accumulator and 0F*hex* is written to **Last Request Control** U202 <15>. This is actually a dual 12-bit serial write to the accumulator where the first 4-bits of each of the two 12-bit quantities are written to U183 and U202.

4. Put the accumulator in the hold mode.

   - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):L <15>.

5. Set table address bits A3-A5 to zero. This is accomplished by the following sequence of operations:

   - Set **TB Control Latch 0** U240-15 CHOP_SEQ_HOLD(L):H<15>.

   - Set **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:L <15>.

   - Set **TB Control Latch 3** U260-18:H <15> and toggle system clocks once (SYSTEM_RESET(L) goes high).

   - Load serially **Chop Controller Plug-in Sequencer** U180 <15> by writing **Serial Diagnostics Data and Select** U220B-9 LATCHED_SERIAL_DATA_IN <16> (location 6802*hex*), toggling system clocks, and shifting the input value left one bit, for each of the four bits of this register.

   - Set **TB Control Latch 0** U240-15 CHOP_SEQ_HOLD(L):L <15>.

6. Set SYSTEM_RESET(L):L by setting **TB Control Latch 3** U260-18:L <15> and toggling system clocks.

7. For each of the test patterns used for this "walking 1's" test (01*hex*, 02*hex*, 04*hex*, 08*hex*, 01*hex*) this test writes a test pattern to the table location and then reads it back. It then checks to see if what was read back matches the value previously written. The involved method of reading out a table location, plus the control operations required to set the read address for this table via **Chop Controller Plug-in Sequencer** U180 <15> (SEQ_CTR_BIT0-2 become A3-A5 for the table) and **Vertical Source Sequencer** U183 <15> (VS0 and VS1 become A1 and A2 for the table), cause the step-by-step processor-controlled operations to be complex.

   - Set table address bits A1 and A2 to zero. This is done by setting **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L) <15> high and serially loading VS0 and VS1 values via the accumulator.

   - Set SYSTEM_RESET(L):H by setting **TB Control Latch 3** U260-18:L <15> and toggling system clocks.

   - Set accumulator to parallel mode by setting **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:H <15>.

- Toggle system clocks twice to get output of table into the accumulator.
- Read back the 3-bit result (MSB can't be read) and check against the "walking 1's" pattern that was written. This is done by setting **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:L <15>, doing a serial read of the accumulator, and checking the three bits from it which correspond to **Acq Mem Data Accumulator** U24 INITIAL_ID_0,1, and 2 <14>.
- Set SYSTEM_RESET(L):L by setting **TB Control Latch 3** U260-18:L <15> and toggling system clocks once.
- If not last pass, then write next pattern into present table location.

Error Index G3211 The value read back from the **Acq Mem Data Accumulator** <14> did not match the value written into the **Initial Wfm Id Encoder Table** <14> (excepting the MSB).

Routine Name Address/Data

Overview This test verifies the address and data integrity of the **Initial Wfm Id Encoder Table** (U72, U62, U63, U73A, U22, U32) <14> on the Time Base board by doing a "walking 1's" test on the entire address space of the table. The **Acq Mem Data Accumulator** <14> is used to read out the contents of the table.

Description There are some basic characteristics of how this test is executed which should be noted.

- The 16-bit accumulator, which is used in this test, is actually made up of four 4-bit registers (U14, U15, U24, U25); serial operations are done by doing serial operations on these registers simultaneously. The input serial data (U14-2 SERIAL_DATA_LOW_IN, and U25-2 SERIAL_DATA_HIGH_IN) come from bits D7 and D15 of the data bus, respectively; the output serial data (U15-12 ACQ_SERIAL_DATA_LOW_OUT and U24-12 ACQ_SERIAL_DATA_HI_OUT) go to bits D0 and D8 of the data bus, respectively.

- **Acq Mem Data Accumulator** U65C-10 INVALIDATE_POINT(H) and U65C-9 INV_PT_HALF(H) <14> originate in **Chop Transition Blanking** <15>.

- The accumulator registers are loaded and read in tandem with U202 in **Timebase Request Pipeline** <15> and U183 in **Vertical Source Sequencer** <15>, which means that failures there could affect this test.

- Writing to this table is straightforward—**Initial Wfm Id Encoder Table** U62-1/U63-1 SYSTEM_RESET(L) <14> is held low and a value written to the table addresses. Reading a table location, however, is not straight-forward—it involves setting SYSTEM_RESET(L) high, clocking the table output through a latch (U12) and into the accumulator, and then reading the accumulator and stripping out the table output bits from all the rest. One limitation on the coverage is that only three of the four table output bits can be read back; the MSB (ID 3) cannot be read via the accumulator. So even though the basic "walking 1's" test done on this table is effectively the same as a simple RAM test, the control aspects of achieving this are more complex.

- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:

   - **Initial Wfm Id Encoder Table** U62-1/U63-1 SYSTEM_RESET(L) <15> is set low.

   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

2. Set up accumulator for right shift serial mode.

- Set **TB Control Latch 1** U251-14 INV_PT_RESET(L):L, **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:L, and **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):H.

3. Set **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L):L <15>.

   - Write 00*hex* to **Vertical Source Sequencer** U183 <15>; at the same time, 0000*hex* is written to the accumulator and 0F*hex* is written to **Last Request Control** U202 <15>. This is actually a dual 12-bit serial write to the accumulator where the first 4-bits of each of the two 12-bit quantities are written to U183 and U202.

4. Put the accumulator in the hold mode.

   - Set **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):L <15>.

5. Starting with the lowest address of the initial waveform table, fill it with 01*hex* (start of "walking 1's" through 4 bits of table).

6. Set **Chop Controller Plug-in Sequencer** U201-7 SEQ_CLK(H):H <15> and Set **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L):L <15>.

   - Set **TB Control Latch 4** U261-15 CTR_MUX_CNTRL:H <15>, set U261-12,13,14:L, and set **TB Control Latch 3** U250-15 SINGLE_SHOT_CHOP(L):H <15>.

   - Toggle system clocks twice. SEQ_CLK(H) and **Chop Controller Plug-in Sequencer** PLUG-IN_CLK(H) <15> should now both be high. **Chop Controller Plug-in Sequencer** U200B-9 <15> should also be high.

   - Repeat step 3 again here. This amounts to a little redundancy for setting SAMPLE_ENABLE(L) low, but does no harm.

7. For each of the test patterns used for this "walking 1's" test (01*hex*, 02*hex*, 04*hex*, 08*hex*, 01*hex*) this test reads each table location, checks to see if it matches the value previously written, then if not the last pass through the table, writes the next test pattern into that table location. The involved method of reading out a table location, plus the control operations required to set the read address for this table via **Chop Controller Plug-in Sequencer** U180 <15> (SEQ_CTR_BIT0-2 become A3-A5 for the table) and **Vertical Source Sequencer** U183 <15> (VS0 and VS1 become A1 and A2 for the table), cause the step-by-step processor-controlled operations to be complex.

- Set table address bits A3-A5. This is accomplished by the following sequence of operations:

  1) Set **TB Control Latch 0** U240-15 CHOP_SEQ_HOLD(L):H<15>.

  2) Set **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:L <15>.

  3) Set **TB Control Latch 3** U260-18:H <15> and toggle system clocks once (SYSTEM_RESET(L) goes high).

  4) Load serially **Chop Controller Plug-in Sequencer** U180 <15> by writing **Serial Diagnostics Data & Select** U220B-9 LATCHED_SERIAL_DATA_IN <16> (location 6802*hex*), toggling system clocks, and shifting the input value left one bit, for each of the four bits of this register.

  5) Set **TB Control Latch 0** U240-15 CHOP_SEQ_HOLD(L):L <15>.

- Set table address bits A1 and A2 . This is done by setting **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L) <15> high and serially loading VS0 and VS1 values via the accumulator.

- Set accumulator to parallel mode by setting **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:H <15>.

- Toggle system clocks twice to get output of table into the accumulator.

- Read back the 3-bit result (MSB can't be read) and check against the walking 1's pattern that was written. This is done by setting **TB Control Latch 0** U240-16 DIAG_SERIAL_MODE:L <15>, doing a serial read of the accumulator, and checking the three bits from it which correspond to INITIAL_ID_0,1, and2.

- Set SYSTEM_RESET(L):L by setting **TB Control Latch 3** U260-18:L <15> and toggling system clocks once.

- If not last pass, then write next pattern into present table location.

- Set SAMPLE_ENABLE(L) back to low as described in step 3.

Error Index G3221 The value read back from the **Acq Mem Data Accumulator** <14> did not match the value written into the **Initial Wfm Id Encoder Table** <14> (excepting the MSB).

Routine Name

Vert Src Seq (Vertical Source Sequencer)

Overview

This test verifies the **Vertical Source Sequencer** ROMs U184 & U173 <15> by performing a checksum on their contents.

Description

1. Force a known state for the **Vertical Source Sequencer** U184 & U173 ROM <15> address inputs.

   - Set **Main Request Logic** U277C-10 SYSTEM_RESET(L):L <18>. This forces U294-12 MAIN_REQUEST(L) to be entirely dependent on the level of U248D-13 FORCE_MAIN(H). It also forces **Chop Controller & Plug-in Sequencer** U181-15 INT_ACQ(H):L <15> which makes the **Vertical Source Sequencer** U184-1 & U173-1 PROCEED(H) entirely dependent on **Acquisition Request Control** U211B-5 <15>.

   - Set **Window 2 Request Logic** U298D-11 RBM2(L):L and U293-12 PTF2(L):H <18>. This forces U297-15 WIND_2_REQUEST(L) to be entirely dependent on the level of U293-6 FORCE_WIND_2(L).

   - Set **Window 1 Request Logic** U298C-8 RBM1(L):L and U285-11 PTF1(L):H <19>. This forces U297-12 WIND_1_REQUEST(L) to be entirely dependent on the level of U285-10 FORCE_WIND_1(L).

   - Force the system clocks into single-step mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):H <14>.

   - Set **Vertical Source Sequencer** U183-12 VS0(H):L and U183-13 VS1(H):L.

2. Increment through the **Vertical Source Sequencer** <15> ROM address space and calculate the checksum.

   - Sequence through both combinations of ROM address A6 PROCEED(H) by setting **TB Control Latch 4** FORCE_MAIN(H):L, FORCE_WIND_1(L):H, and FORCE_WIND_2(L):H <15> in one case and to the opposite logic levels in the second case. These signals control **Main Request Logic** U294-12 MAIN_REQUEST(L) <18>, **Window 2 Request Logic** U297-15 WIND2_REQUEST(L) <18> and **Window 1 Request Logic** U297-12 WIND1_REQUEST(L) <19>, which in turn control **Time base Request Pipeline** U193A-12 <15>, thus driving PROCEED(H). For each state of A6, the system clocks must be toggled five times to propagate these signals through the request logic circuits to the A6 ROM address inputs.

   - For each state of A6 in the previous step, sequence through all combinations of A2-A5 by controlling **TB Control Latch 5** U250 <15> CAL_SEL(H), RIGHT_SEL(H), CENTER_SEL(H), and LEFT_SEL(H).

- For each state of A2-A5 in the previous step, sequence through all combinations of A0-A1 (VS0, VS1) by serial shifting the appropriate combinations into **Vertical Source Sequencer** U183-12 and 13.

- For each of the above address combinations, capture the data from **Vertical Source Sequencer** U173 into parallel/serial shift register U183. Read U183 and calculate the checksum. This is accomplished by setting U183-10 DIAG_SERIAL_MODE(H):L and U183-9 ACQ_DATA_HOLD(L):H. The. serial data clock U183-11 ACQ_INTERFACE_CLOCK(H) is generated by toggling **System Clock Control** U132D-13 ASD_RD(L) which drives U133C-8 ACQ_INTERFACE_CLOCK(H).

Error Index G3311 The computed checksum did not match the expected checksum.

Routine Name

Tbase Req Id (Timebase Request Id)

Overview

This test verifies the main and window request operations of the **Time base Request Pipeline** <15> and **Acquisition Request Control** <15> by using the **Main Request Logic** <18>, **Window 2 Request Logic** <18>, and **Window 1 Request Logic** <19> circuits to generate main and window requests.

Description

1. Force the main and window request logic circuits to a known state.

   - Set **Main Request Logic** U277C-10 SYSTEM_RESET(L):L <18>. This forces U294-12 MAIN_REQUEST(L) to be entirely dependent on the level of U248D-13 FORCE_MAIN(H).
   - Set **Window 2 Request Logic** U298D-11 RBM2(L):L and U293-13 PTF2(L):H. This forces U297-15 WIND_2_REQUEST(L) to be entirely dependent on the level of U293-6 FORCE_WIND_2(L).
   - Set **Window 1 Request Logic** U298C-8 RBM1(L):L and U285-11 PTF1(L):H. This forces U297-12 WIND_1_REQUEST(L) to be entirely dependent on the level of U285-10 FORCE_WIND_1(L).
   - Force the system clocks into single step mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):H <14>.

2. Force a main and window request pattern (see Error Index section below).

   - Write time base request pattern to **TB Control Latch 4** <15>, which controls FORCE_MAIN(H), FORCE_WIND_1(L), and FORCE_WIND_2(L).

3. Clock the main and window requests through the pipeline delay.

   - Single-step the system clocks six times by toggling **System Clock Control** U67C-5 CLOCK_STEP(L) <14>. This clocks MAIN_REQUEST(L) through the **Main Request Logic** <18>, WIND_2_REQUEST(L) through the **Window 2 Request Logic** <18>, and WIND_1_REQUEST(L) through the **Window 1 Request Logic** <19>. All requests are then clocked through **Time base Request Pipeline** U203 <15> and into parallel/serial shift register U202.

4. Read and verify the main and window request pattern.

   - Read the time base request pattern serially from U202-12 ACQ_SERIAL_DATA_HIGH_OUT. This is accomplished by setting **Time base Request Pipeline** U202-10 DIAG_SERIAL_MODE(H):L and U202-9 ACQ_DATA_HOLD(L):H <15>. The serial data clock U202-11 ACQ_INTERFACE_CLOCK(H) is generated by toggling **System Clock Control** U132D-13 ASD_RD(L), which drives U133C-8 ACQ_INTERFACE_CLOCK(H).

5. Repeat steps 2-4 for the remaining main and window request patterns (see Error Index section below).

Error Index G3321

The time base request pattern read from the **Time base Request Pipeline** U202 <15> did not match the expected request pattern. The display shows the first pattern which

did not match. The time base request patterns written to **TB Control Latch 4 <15>** and the expected patterns read from U202 are shown in the following table:

TB Control Latch 4 Pattern	Expected Timebase Request Pattern
00*hex*	4*hex*
10*hex*	5*hex*
20*hex*	6*hex*
40*hex*	0*hex*
50*hex*	1*hex*
60*hex*	2*hex*
70*hex*	3*hex*

Routine Name

Seq Cntr Clk (Sequence Counter Clock)

Overview

This test verifies the clocking capability of the **Chop Controller Plug-in Sequencer** U180 <15> by passing test patterns through FPLA U181 to U180.

Description

1. Force **Chop Controller & Plug-in Sequencer** U181 & U180 <15> into initial test states.

   - Set **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):H <14> to force the system clocks into single-step clock mode. Set **Chop Controller Plug-in Sequencer** U181-23 SYSTEM_RESET(L):L. Set parallel/serial shift register U180 into parallel load mode by setting U180-10 DIAG_SERIAL_MODE(H):H and U180-9 CHOP_SEQ_HOLD(L):H.

   - Set **Chop Controller & Plug-in Sequencer** U181-23 SYSTEM_RESET(L):H.

   - Set **Chop Controller & Plug-in Sequencer** U181-13 PLUG_IN_CLK(H):H by writing 08*hex* to **TB Control Latch 5** U250 <15> and **TB Control Latch 4** U261 <15>.

2. Write a test pattern (see Error Index section below) to **Chop Controller & Plug-in Sequencer** U180 CTR_ADDR <15> inputs.

   - Write the test pattern to **TB Control Latch 4** U261 CTR_ADDR_0-CTR_ADDR_2. Keep U261-15 CTR_MUX_CTRL high so that the CTR_ADDR_0-CTR_ADDR_2 bits are passed directly through the **Chop Controller & Plug-in Sequencer** U181 FPLA to the CQ0-CQ2 outputs.

3. Read and verify the test pattern (see Error Index section below) from **Chop Controller & Plug-in Sequencer** parallel/shift register U180 <15>.

   - Set the parallel/shift register in serial mode by setting U180-10 DIAG_SERIAL_MODE(H):L and U180-9 CHOP_SEQ_HOLD(L):L. The U180-11 clock is driven by U190A-2 MCK_ACM_1 which is enabled through U190A by clocking the PLUG_IN_CLK(H) high through gate U211C and latch U200B. This results in the low enable at U190A-3.

   - Toggle the system clocks (MCK_ACM_1), via **System Clock Control** U67C-5 CLOCK_STEP(L), and read the CQ0 (SEQ_CTR_BIT_0), CQ1 (SEQ_CTR_BIT_1), and CQ2 (SEQ_CTR_BIT_2) bits back through **Serial Diagnostics Data & Select** U196B-5 CHOP_SEQ_DATA_HIGH_OUT <16>.

4. Repeat steps 2 and 3 for the remaining **Chop Controller Plug-in Sequencer** CTR_ADDR inputs (see Error Index section below).

Error Index G3331

The plug-in sequencer pattern read from **Chop Controller Plug-in Sequencer** U180 <15> did not match the pattern written to **TB Control Latch 4** CTR_ADDR outputs. The display shows the first pattern which did not match. The CTR_ADDR patterns written to **TB Control Latch 4** and the expected patterns read from U180 are shown in the following table:

TB Control Latch 4 Pattern	Expected Sequencer Counter Pattern

Points Acq	Plg Seq Ctrl	Seq Cntr Clk (G333X)
	00*hex*	0*hex*
	01*hex*	1*hex*
	02*hex*	2*hex*
	03*hex*	3*hex*
	04*hex*	4*hex*
	05*hex*	5*hex*
	06*hex*	6*hex*
	07*hex*	7*hex*

Routine Name FP -10.0000V

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating –10.0000 V through the **Front-Panel Calibrator <12>.**

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum negative value of –10.1 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to ÷1, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CF*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to ÷1 (D2-S2).
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>.**
   - Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP -5.0000 V

Overview This test helps verify the operation of **Voltage Reference DAC** <12>, **Cal Select** <12>, **Cal Gain Select** <12>, **Plug-in Ground Select** <12>, and **Cal Output Buffer** <12> by generating –5.0000 V through the **Front-Panel Calibrator** <12>.

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator** <12> output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum negative value of –10.1 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +2, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CE*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +2 (D3-S3) and +1 (D2-S2), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1**, **Refresh Bank 2**, and **Refresh Bank 3** <12>.

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP -2.5000 V

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating –2.5000 V through the **Front-Panel Calibrator <12>**.

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum negative value of –10.1 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +4, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CC*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +4 (D1-S1) and +1 (D2-S2), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>**.

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

If the **Voltage Reference DAC** <12> R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP -1.0000 V

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating –1.0000 V through the **Front-Panel Calibrator <12>.**

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator** <12> output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum negative value of –10.1 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +10, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CB*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2) and +10 (D3-S3), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3** <12>.

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP -100.00mV

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating –100.00 mV through the **Front-Panel Calibrator <12>**.

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum negative value of –10.1 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +100, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write C7*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to ÷1 (D2-S2) and ÷100 (D4-S4), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>**.

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

If the **Voltage Reference DAC** <12> R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP 0.0000 V

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating 0.0000 V through the **Front-Panel Calibrator <12>.**

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to 0.0000 volts.
3. Set **Cal Gain Select** attenuation to +1, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CF*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2).
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>.**
   - Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP +99.951mV

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating +99.951 mV through the **Front-Panel Calibrator <12>**.

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e.,a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum positive value of +10.095 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +100, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write C7*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2) and +100 (D4-S4), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>**.

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP +999.51mV

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating +999.51 mV through the **Front-Panel Calibrator <12>.**

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e.,a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum positive value of +10.095 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +10, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CB*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to ÷1 (D2-S2) and ÷10 (D3-S3), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>.**

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP +2.4988 V

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating +2.4988 V through the **Front-Panel Calibrator <12>**.

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e.,a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum positive value of +10.095 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +4, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CC*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +4 (D1-S1) and +1 (D2-S2), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>**.

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP +4.9976 V

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating +4.9976 V through the **Front-Panel Calibrator <12>.**

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e.,a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum positive value of +10.095 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +2, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CE*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +2 (D3-S3) and +1 (D2-S2), respectively.
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>.**

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP +9.9951 V

Overview This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating +9.9951 V through the **Front-Panel Calibrator <12>**.

Operator Procedure Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on **Front-Panel Calibrator <12>** output J26 on the front-panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e.,a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum positive value of +10.095 V. (The output of the **Cal Output Buffer** and **Front-Panel Calibrator** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +1, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front panel calibrator ground source.
   - Write CF*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2).
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>**.
   - Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

If the **Voltage Reference DAC** <12> R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name FP 1 KHz

Overview This test verifies that the **Front Panel Calibrator <12>, Main Holdoff Logic <18>, Holdoff Logic <10>,** and a small portion of the **Main Fine Holdoff Generator <11>** has the basic capability to generate a waveform necessary for probe compensation operations. This is done by generating an approximate 6 V p-p, 1 kHz square wave, centered around 0 V, through the **Front-Panel Calibrator <12>.**

Operator Procedure Once this test is invoked, the operator may use an oscilloscope to verify the waveform present on **Front-Panel Calibrator <12>** output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to a value of +3.00 V.
3. Set **Cal Gain Select** attenuation to +1, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to front-panel calibrator ground source.
   - Write CF*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2).
4. Enable the **Front-Panel Calibrator <12>** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):H and U1932C-9 CAL_CLK_SEL(H):L (this selects one-half the CAL_PRE_CLK rate, through U2060A, coming from the **Main Fine Holdoff Generator <11>**).
5. Disable the **Main Fine Holdoff Generator <11>.**
   - Set U1242-4 MFINEOFF(L):L
6. Initialize inputs to the **Holdoff Logic <10>.**
   - Set U1644-1 STRGARM(L):H and U1452-2 HOCALSEL(H):H.

- Set U1452-4 PRETC(H):H by loading the **Pretrigger Generator** U275 WIND C CTR <19> with a value of zero and then manually clocking (via the **System Clock Control** <14>) the resulting terminal count at U275-29 through the **Pretrigger Randomizer** <19>.

- Select the 19.6608 MHz clock inputs (U2050-2 & 5) of the **Multiplexer** <10> by setting **Trigger Source Register** U2030-5 low.

7. Load and start the **Main Holdoff Logic** U264 MAIN A CTR (main holdoff counter).

   - Set U236-9 MHOLOAD(H):H <18>, write 00002660*hex* to the 35-bit main holdoff counter, and then set U236-9 MHOLOAD(H):L <18>. This starts the counter counting down and when it reaches terminal count, the counter automatically gets reloaded via U247C-9 MHOLDOFF(L). This results in a symmetrical waveform of approximately 1 kHz frequency at **Front-Panel Calibrator** U1444C-10 <12>. When this waveform goes low, it causes the calibrator output at J26 to go approximately 6 V negative from the original positive reference of 3 V programmed above (i.e., PLUG-IN REF).

8. Wait for command from Executive processor to proceed.

9. Stop processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).

   - Close all analog switches (DG211's) in **Refresh Bank 1**, **Refresh Bank 2**, and **Refresh Bank 3** <12>.

   - Disable the periodic processor refresh interrupts.

Error Index

None.

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

The **Utility** Menu **Probes** for probe compensation waveforms in normal operation (see below).

Caveats

This test does not use the functional capabilities of the **Main Fine Holdoff Generator** <11> (i.e., U1932D-13 MFINEOFF(L) is set low) which gives it the capability of more finely setting the frequency in this test, whereas in normal operation, this may not be true. Therefore, the probe compensation waveform produced in normal operation could be bad and the waveform from this test could be good, thus indicating a problem in the **Main Fine Holdoff Generator** circuitry.

Routine Name FP 1 MHz

Overview This test verifies that the **Front Panel Calibrator** <12>, **Main Holdoff Logic** <18>, **Holdoff Logic** <10>, and a small portion of the **Main Fine Holdoff Generator** <11> has the basic capability to generate a waveform necessary for probe deskew operations. This is done by generating an approximate 600 mV p-p, 1 MHz square wave, centered around 0 V, through the **Front-Panel Calibrator** <12>.

Operator Procedure Once this test is invoked, the operator may use an oscilloscope to verify the waveform present on **Front-Panel Calibrator** <12> output J26 on the front panel of the instrument.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to a value of +300 mV.
3. Set **Cal Gain Select** attenuation to +1, **Front-Panel Calibrator** to 50Ω output impedance, and **Plug-in Ground Select** to front-panel calibrator ground source.
   - Write 4F*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):L, selects the FPCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2).
4. Enable the **Front-Panel Calibrator** <12> square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):H and U1932C-9 CAL_CLK_SEL(H):L (this selects one-half the CAL_PRE_CLK rate, through U2060A, coming from the **Main Fine Holdoff Generator** <11>).
5. Disable the **Main Fine Holdoff Generator** <11>.
   - Set U1242-4 MFINEOFF(L):L
6. Initialize inputs to the **Holdoff Logic** <10>.
   - Set U1644-1 STRGARM(L):H and U1452-2 HOCALSEL(H):H.

- Set U1452-4 PRETC(H):H by loading the **Pretrigger Generator** U275 WIND C CTR <19> with a value of zero and then manually clocking (via the **System Clock Control** <14>) the resulting terminal count at U275-29 through the **Pretrigger Randomizer** <19>.

- Select the 19.6608 MHz clock inputs (U2050-2 & 5) of the **Multiplexer** <10> by setting **Trigger Source Register** U2030-5 low.

7. Load and start the **Main Holdoff Logic** U264 MAIN A CTR (main holdoff counter).

   - Set U236-9 MHOLOAD(H):H <18>, write 00000004*hex* to the 35-bit main holdoff counter, and then set U236-9 MHOLOAD(H):L <18>. This starts the counter counting down and when it reaches terminal count, the counter automatically gets reloaded via U247C-9 MHOLDOFF(L). This results in a symmetrical waveform of approximately 1 MHz frequency at **Front Panel Calibrator** U1444C-10 <12>. When this waveform goes low, it causes the calibrator output at J26 to go approximately 600 mV negative from the original positive reference of 300 mV programmed above (i.e., PLUG-IN REF).

8. Wait for command from Executive processor to proceed.

9. Stop processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).

   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3** <12>.

   - Disable the periodic processor refresh interrupts.

Error Index

None.

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

This test does not use the functional capabilities of the **Main Fine Holdoff Generator** <11> (i.e., U1932D-13 MFINEOFF(L) is set low) which gives it the capability of more finely setting the frequency in this test, whereas in normal operation, this may not be true. Therefore, the probe deskew waveform produced in normal operation could be bad and the waveform from this test could be good, thus indicating a problem in the **Main Fine Holdoff Generator** circuitry. It should be noted, however, that the deskew waveform in normal operation is not statically generated and is not easily viewable by the operator.

Routine Name PI –10.0000V

Overview This test helps verify the operation of **Voltage Reference DAC** <12>, **Cal Select** <12>, **Cal Gain Select** <12>, **Plug-in Ground Select** <12>, and **Cal Output Buffer** <12> by generating –10.0000 V at the Plug-in Interface dc calibration source (**Plug-in Ground Select** J91A-3 CAL).

Operator Procedure This test requires that a plug-in be resident in the left compartment in order to provide the proper grounding for the calibration circuitry. Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on J91A-3 CAL of the **Plug-in Interface** <12>.

Description

1. Start processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).
   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.
   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.
   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.
2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum negative value of -10.1 V. (The output of the **Cal Output Buffer** should result in the desired voltage).
3. Set **Cal Gain Select** attenuation to +1, **Front-Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to left plug-in ground source.
   - Write 9F*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the LEFT CALCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2).
4. Disable the **Front-Panel Calibrator** square wave clock input.
   - Set U2060A-4 CAL_CLK_ENABLE(H):L.
5. Wait for command from Executive processor to proceed.
6. Stop processor refresh of programmable reference voltages.
   - Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).
   - Close all analog switches (DG211's) in **Refresh Bank 1**, **Refresh Bank 2**, and **Refresh Bank 3** <12>.

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>**.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name

PI +9.9951 V

Overview

This test helps verify the operation of **Voltage Reference DAC <12>, Cal Select <12>, Cal Gain Select <12>, Plug-in Ground Select <12>,** and **Cal Output Buffer <12>** by generating +9.9951 V at the Plug-in Interface dc calibration source (**Plug-in Ground Select** J91A-3 CAL).

Operator Procedure

This test requires that a plug-in be resident in the left compartment in order to provide the proper grounding for the calibration circuitry. Once this test is invoked, the operator may use a high accuracy digital voltmeter to verify the voltage present on J91A-3 CAL of the Plug-in Interface <12>.

Description

1. Start processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC <12>** (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).

   - Close one of the analog switches (DG211) in **Refresh Bank 1 <12>, Refresh Bank 2 <12>,** or **Refresh Bank 3 <12>** for the voltage to be programmed for approximately 400µs.

   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.

   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.

2. Set **Cal Gain Select** U1770B-5 PLUG-IN REF <12> to its maximum positive value of +10.095 V. (The output of the **Cal Output Buffer** should result in the desired voltage).

3. Set **Cal Gain Select** attenuation to +1, **Front Panel Calibrator** to 450Ω output impedance, and **Plug-in Ground Select** to left plug-in ground source.

   - Write 9F*hex* to **Cal Select** U1570. This sets U1570-19 50EN(L):H, selects the LEFT CALCOM input ground source to **Plug-in Ground Select** U1970, and sets **Cal Gain Select** dividers U1672 and U1772 to +1 (D2-S2).

4. Disable the **Front Panel Calibrator** square wave clock input.

   - Set U2060A-4 CAL_CLK_ENABLE(H):L.

5. Wait for command from Executive processor to proceed.

6. Stop processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).

   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>.**

- Disable the periodic processor refresh interrupts.

Error Index

None. Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 6 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

If the **Voltage Reference DAC** <12> R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the measured voltage will also be incorrect.

Routine Name

Over Range

Overview

This test verifies the capability of the A/D Converter <8> (more specifically the **Over/Under Range Logic** <8>) to indicate an overrange condition. The **Voltage Reference DAC** <12> is used to generate an overrange voltage, via **Refresh Bank 3** CH SWREF <12>, to **Sampler** U740 <6> (or <6A> for the 11402). The vertical select lines (VS0, VS1 of **Vertical Source Sequencer** <15>) are set up to feed the CH SWREF channel through the SAMPLER HYBRID U740 and to the input of the A/D Converter. Converted digital values are acquired and stored in **Acq Mem Even Bank** and **Acq Mem Odd Bank** <13> (just as they are in normal operation), where they can be retrieved by the processor (**MPU** U224 <16>). To help ensure that the converted value should be an overrange value, the **1st Stage A/D** <8> programmable references (Q1CALP, Q1CALN) are reduced from their calibrated values.

Description

1. Start processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).

   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.

   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.

   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.

2. Change **1st Stage A/D** <8> programmable references to induce overrange condition.

   - Set **Refresh Bank 1** U760-14 Q1CALP and U760-8 Q1CALN <12> to values which should yield reference voltages at **1st Stage A/D** U430-51,49 <8> of approximatel +0.6 V and –0.6 V, respectively. Save the previous values.

3. Set **Refresh Bank 3** U1060-1 CH SWREF <12> to a value (maximum positive **Voltage Reference DAC** <12> voltage) which should yield a voltage at **1st Stage A/D** R642 CH SW OUT <8> which is more positive than the positive A/D reference (U430-51).

4. Set up time base for acquisition.

- Load **Main Postrecord Logic** U264 MAIN C CTR <18> with a high value (00000FFFF*hex* —35 bits), set **TB Control Latch 3** U212-12 SYSTEM_RESET(H):H <15>, and clock the low non-terminal count at U264-29 MAIN_POSTRECORD_TC(H) through **Last Request Generation** <18>. This helps ensure that **Chop Transition Blanking** U210-10 L_LAST_REQUEST<15> stays low later on so that the acquisition cycle stays in write mode.

- Set **TB Control Latch 3** U260-19 SINGLE(H):H and U212-12 SYSTEM_RESET(H):L <15>.

- Put system clocks into free-run mode by setting **System Clock Control** U67B-3 CLOCK_SOURCE_SELECT(H):L <14>.

- Enable **Restart Control** <20> by setting U175A-4 SET_RESTART(L) and U175A-1 CLR_RESTART(L) both high.

- Fill all 32 locations in **Initial Wfm Id Encoder Table** <14> RAM (U22, U32) with 8*hex* in order to force U12-6 VALIDATE_ACQ_REQUEST(H) high.

- Set **TB Control Latch 4** U261-19 FORCE_LAST(H):L and U261-18 FORCE_MAIN(H):H <15>. This last bit causes **Acquisition Request Control** U193A-12 TB_REQUEST(H) <15> to go high and eventually **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L) <15> to go low, thereby enabling the **Sample Gate Generator** <6>.

- Select the CAL channel as the **Sampler** U740 <6> vertical source input by setting **Vertical Source Sequencer** U183-12 VS0 and U183-12 VS1 <15> both low. This is done by setting U183-9 ACQ_DATA_HOLD(L):H, U183-10 DIAG_SERIAL_MODE:H, and writing 88*hex* to **TB Control Latch 5** (U250-19 CAL_SEL(H):H).

- Reset **Chop Transition Blanking** <15> by setting U210-1 INV_PT_RESET(L) low, then high. This forces U210-10 L_LAST_REQUEST(H):L and will force **Acq Mem Control Logic** U160-15 ACQ_MEM_R(H)/W_MODE(L) <13> low into the write mode when SYSTEM_RESET(L) goes high.

5. Start the acquisition write cycle by writing CB*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):H.

6. Perform a software delay to allow the acquisition memory (**Acq Mem Even Bank** and **Acq Mem Odd Bank** <13>), to be filled with digitized samples.

7. Stop the acquisition write cycle by writing 8B*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):L.

8. Read one location (sample) from the acquisition memory and verify that it is an overrange value of 7FFF*hex*.

9. Restore **Refresh Bank 1** U760-14 Q1CALP and U760-8 Q1CALN <12> to their previously saved values.

10. Stop processor refresh of programmable reference voltages.

- Program the **Voltage Reference DAC <12>** for an output of +3.00 V (A60*hex*).
- Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3 <12>.**
- Disable the periodic processor refresh interrupts.

Error Index G3511 The converted A/D voltage was not an overrange value (7FFF*hex*).

Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 10 above).

See Also The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC <12>.**

Caveats If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the test could possibly fail.

Routine Name

Under Range

Overview

This test verifies the capability of the A/D Converter <8> (more specifically the **Over/Under Range Logic** <8>) to indicate an underrange condition. The **Voltage Reference DAC** <12> is used to generate an underrange voltage, via **Refresh Bank 3** CH SWREF <12>, to **Sampler** U740 <6> (or <6A> for the 11402). The vertical select lines (VS0, VS1 of **Vertical Source Sequencer** <15>) are set up to feed the CH SWREF channel through the SAMPLER HYBRID U740 and to the input of the A/D Converter. Converted digital values are acquired and stored in **Acq Mem Even Bank** and **Acq Mem Odd Bank** <13> (just as they are in normal operation), where they can be retrieved by the processor (**MPU** U224 <16>). To help ensure that the converted value should be an underrange value, the **1st Stage A/D** <8> programmable references (Q1CALP, Q1CALN) are reduced from their calibrated values.

Description

1. Start processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).

   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.

   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.

   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.

2. Change **1st Stage A/D** <8> programmable references to induce underrange condition.

   - Set **Refresh Bank 1** U760-14 Q1CALP and U760-8 Q1CALN <12> to values which should yield reference voltages at **1st Stage A/D** U430-51,49 <8> of approximatel +0.6 V and –0.6 V, respectively. Save the previous values.

3. Set **Refresh Bank 3** U1060-1 CH SWREF <12> to a value (maximum negative **Voltage Reference DAC** <12> voltage) which should yield a voltage at **1st Stage A/D** R642 CH SW OUT <8> which is more negative than the negative A/D reference (U430-49).

4. Set up time base for acquisition.

- Load **Main Postrecord Logic** U264 MAIN C CTR <18> with a high value (00000FFFF*hex* —35 bits), set **TB Control Latch 3** U212-12 SYSTEM_RESET(H):H <15>, and clock the low non-terminal count at U264-29 MAIN_POSTRECORD_TC(H) through **Last Request Generation** <18>. This helps ensure that **Chop Transition Blanking** U210-10 L_LAST_REQUEST<15> stays low later on so that the acquisition cycle stays in write mode.

- Set **TB Control Latch 3** U260-19 SINGLE(H):H and U212-12 SYSTEM_RESET(H):L <15>.

- Put system clocks into free-run mode by setting **System Clock Control** U67B-3 CLOCK_SOURCE_SELECT(H):L <14>.

- Enable **Restart Control** <20> by setting U175A-4 SET_RESTART(L) and U175A-1 CLR_RESTART(L) both high.

- Fill all 32 locations in **Initial Wfm Id Encoder Table** <14> RAM (U22, U32) with 8*hex* in order to force U12-6 VALIDATE_ACQ_REQUEST(H) high.

- Set **TB Control Latch 4** U261-19 FORCE_LAST(H):L and U261-18 FORCE_MAIN(H):H <15>. This last bit causes **Acquisition Request Control** U193A-12 TB_REQUEST(H) <15> to go high and eventually **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L) <15> to go low, thereby enabling the **Sample Gate Generator** <6>.

- Select the CAL channel as the **Sampler** U740 <6> vertical source input by setting **Vertical Source Sequencer** U183-12 VS0 and U183-12 VS1 <15> both low. This is done by setting U183-9 ACQ_DATA_HOLD(L):H, U183-10 DIAG_SERIAL_MODE:H, and writing 88*hex* to **TB Control Latch 5** (U250-19 CAL_SEL(H):H).

- Reset **Chop Transition Blanking** <15> by setting U210-1 INV_PT_RESET(L) low, then high. This forces U210-10 L_LAST_REQUEST(H):L and will force **Acq Mem Control Logic** U160-15 ACQ_MEM_R(H)/W_MODE(L) <13> low into the write mode when SYSTEM_RESET(L) goes high.

5. Start the acquisition write cycle by writing CB*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):H.

6. Perform a software delay to allow the acquisition memory (**Acq Mem Even Bank** and **Acq Mem Odd Bank** <13>), to be filled with digitized samples.

7. Stop the acquisition write cycle by writing 8B*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):L.

8. Read one location (sample) from the acquisition memory and verify that it is an underrange value of 8001*hex*.

9. Restore **Refresh Bank 1** U760-14 Q1CALP and U760-8 Q1CALN <12> to their previously saved values.

10. Stop processor refresh of programmable reference voltages.

- Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).
- Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3** <12>.
- Disable the periodic processor refresh interrupts.

Error Index G3521 The converted A/D voltage was not an underrange value (8001*hex*).

Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 10 above).

See Also The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats If the **Voltage Reference DAC** <12> R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the test could possibly fail.

Routine Name Err Corr ROM (Error Correction ROMs)

Overview This test verifies the **Error Correction ROMs <8>** on the Acquisition board by doing a checksum test.

Description There are some basic characteristics of how this test is executed which should be noted.

- Note that the most significant address line to these ROMs (A7 of U230 and U240) is the ERRCORROFF <8> signal (U251-16 of **TB Control Latch 1** <15>). All other address lines can be controlled by the processor via the **Diagnostic Register** <8> of the Acquisition board. The seven LSBs of this register form the seven LSBs of the ROMs address whereas the MSB is the test enable bit TEST. Whenever this register is written in this test the TEST bit is set to the high state. This way, the A/D outputs are disabled from the data bus which drives the ROM address inputs and the **Diagnostic Register** is free to control it.
- ERRCORROFF is in the low state on entry to this test.
- The output of the ROMs is read back via the **Acq Mem Data Accumulator** <14> on the Time Base board. For details of how this register is read refer to the Accumulator test description (G311X).
- There are certian addresses of these ROMs which by design correspond to overrange or underrange conditions for the A/D and the output from the ROMs for these addresses required different treatment than the others. This is described in step 1. The specific addresses are: 00*hex*, 7F*hex*, 80*hex* to BF*hex*, E0*hex* to FF*hex*.

1. Do a checksum algorithm on the ROMs, starting at address zero and ending at address FF*hex*.
   - Start out with a checksum value of 0.
   - When the address gets to the first location of the upper half of the ROMs, then set **TB Control Latch 1** U251-16 ERRCORROFF(H) <15> to the high state so that the upper half of the ROMs can be addressed.
   - For each address, write the seven LSBs of it to **Acq BD Data Driver** U198 <16>, latch it into the **Diagnostic Register**, and then read back the output of the ROMs via the **Acq Mem Data Accumulator** <14>.
   - If the present address is one of those known to represent an overrange or underrange condition, then mask out the four LSBs of the ROMs output.
   - Shift the checksum left one bit and add to that the present ROMs output value.
2. Check to see if the calculated checksum value matches the expected checksum for these ROMs.

Error Index G3531 The calculated checksum of the **Error Correction ROMs** <8> did not match the expected checksum.

Routine Name

+0.800 V

Overview

This test verifies the operation of the A/D Converter <8>, **Voltage Reference DAC** <12>, **Refresh Bank 1** <12>, and CH SWREF of **Refresh Bank 3** <12> on the Acquisition board. The DAC is used to generate voltages, via CH SWREF, to **Sampler** U740 <6> (or <6A> for the 11402). The vertical select lines (VS0, VS1 of **Vertical Source Sequencer** <15>) are set up to feed the CH SWREF channel through the SAMPLER HYBRID U740 and to the input of the A/D Converter. Converted digital values are acquired and stored in **Acq Mem Even Bank** and **Acq Mem Odd Bank** <13> (just as they are in normal operation), where they are retrieved and averaged by the processor (**MPU** U224 <16>) to yield a voltage measurement.

Description

1. Start processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).

   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.

   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.

   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.

2. Set **Refresh Bank 3** U1060-1 CH SWREF <12> to a value which should yield +0.800 V at **1st Stage A/D** R642 CH SW OUT <8>.

3. Set up time base for acquisition.

   - Load **Main Postrecord Logic** U264 MAIN C CTR <18> with a high value (00000FFFF*hex* —35 bits), set **TB Control Latch 3** U212-12 SYSTEM_RESET(H):H <15>, and clock the low non-terminal count at U264-29 MAIN_POSTRECORD_TC(H) through **Last Request Generation** <18>. This helps ensure that **Chop Transition Blanking** U210-10 L_LAST_REQUEST<15> stays low later on so that the acquisition cycle stays in write mode.

   - Set **TB Control Latch 3** U260-19 SINGLE(H):H and U212-12 SYSTEM_RESET(H):L <15>.

   - Put system clocks into free-run mode by setting **System Clock Control** U67B-3 CLOCK_SOURCE_SELECT(H):L <14>.

   - Enable **Restart Control** <20> by setting U175A-4 SET_RESTART(L) and U175A-1 CLR_RESTART(L) both high.

- Fill all 32 locations in **Initial Wfm Id Encoder Table** <14> RAM (U22, U32) with 8*hex* in order to force U12-6 VALIDATE_ACQ_REQUEST(H) high.

- Set **TB Control Latch 4** U261-19 FORCE_LAST(H):L and U261-18 FORCE_MAIN(H):H <15>. This last bit causes **Acquisition Request Control** U193A-12 TB_REQUEST(H) <15> to go high and eventually **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L) <15> to go low, thereby enabling the **Sample Gate Generator** <6>.

- Select the CAL channel as the **Sampler** U740 <6> vertical source input by setting **Vertical Source Sequencer** U183-12 VS0 and U183-12 VS1 <15> both low. This is done by setting U183-9 ACQ_DATA_HOLD(L):H, U183-10 DIAG_SERIAL_MODE:H, and writing 88*hex* to **TB Control Latch 5** (U250-19 CAL_SEL(H):H).

- Reset **Chop Transition Blanking** <15> by setting U210-1 INV_PT_RESET(L) low, then high. This forces U210-10 L_LAST_REQUEST(H):L and will force **Acq Mem Control Logic** U160-15 ACQ_MEM_R(H)/W_MODE(L) <13> low into the write mode when SYSTEM_RESET(L) goes high.

4. Start the acquisition write cycle by writing CB*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):H.

5. Perform a software delay to allow the acquisition memory (**Acq Mem Even Bank** and **Acq Mem Odd Bank** <13>), to be filled with digitized samples.

6. Stop the acquisition write cycle by writing 8B*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):L.

7. Read and average 64 locations (samples) from the acquisition memory to yield the final voltage measurement. Verify that this value falls within the expected limits.

8. Stop processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).

   - Close all analog switches (DG211's) in **Refresh Bank 1**, **Refresh Bank 2**, and **Refresh Bank 3** <12>.

   - Disable the periodic processor refresh interrupts.

Error Index G3611

The measured A/D voltage was outside of the acceptable limits.

Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 8 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

If the **Voltage Reference DAC** <12> R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the test may fail.

If an instrument calibration sequence has not taken place and the test uses its standard default values for the programmable reference voltages in the refresh banks, a failure may occur due to the default values not matching closely enough to instrument hardware tolerances. To ensure that good calibration constants are used for these references, exit **Extended Diagnostics**, run **Enhanced Accuracy**, and then rerun the test.

Routine Name

0.000 V

Overview

This test verifies the operation of the A/D Converter <8>, **Voltage Reference DAC** <12>, **Refresh Bank 1** <12>, and CH SWREF of **Refresh Bank 3** <12> on the Acquisition board. The DAC is used to generate voltages, via CH SWREF, to **Sampler** U740 <6> (or <6A> for the 11402). The vertical select lines (VS0, VS1 of **Vertical Source Sequencer** <15>) are set up to feed the CH SWREF channel through the SAMPLER HYBRID U740 and to the input of the A/D Converter. Converted digital values are acquired and stored in **Acq Mem Even Bank** and **Acq Mem Odd Bank** <13> (just as they are in normal operation), where they are retrieved and averaged by the processor (**MPU** U224 <16>) to yield a voltage measurement.

Description

1. Start processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).

   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.

   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.

   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.

2. Set **Refresh Bank 3** U1060-1 CH SWREF <12> to a value which should yield 0.000 V at **1st Stage A/D** R642 CH SW OUT <8>.

3. Set up time base for acquisition.

   - Load **Main Postrecord Logic** U264 MAIN C CTR <18> with a high value (00000FFFF*hex* —35 bits), set **TB Control Latch 3** U212-12 SYSTEM_RESET(H):H <15>, and clock the low non-terminal count at U264-29 MAIN_POSTRECORD_TC(H) through **Last Request Generation** <18>. This helps ensure that **Chop Transition Blanking** U210-10 L_LAST_REQUEST<15> stays low later on so that the acquisition cycle stays in write mode.

   - Set **TB Control Latch 3** U260-19 SINGLE(H):H and U212-12 SYSTEM_RESET(H):L <15>.

   - Put system clocks into free-run mode by setting **System Clock Control** U67B-3 CLOCK_SOURCE_SELECT(H):L <14>.

   - Enable **Restart Control** <20> by setting U175A-4 SET_RESTART(L) and U175A-1 CLR_RESTART(L) both high.

- Fill all 32 locations in **Initial Wfm Id Encoder Table** <14> RAM (U22, U32) with 8*hex* in order to force U12-6 VALIDATE_ACQ_REQUEST(H) high.

- Set **TB Control Latch 4** U261-19 FORCE_LAST(H):L and U261-18 FORCE_MAIN(H):H <15>. This last bit causes **Acquisition Request Control** U193A-12 TB_REQUEST(H) <15> to go high and eventually **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L) <15> to go low, thereby enabling the **Sample Gate Generator** <6>.

- Select the CAL channel as the **Sampler** U740 <6> vertical source input by setting **Vertical Source Sequencer** U183-12 VS0 and U183-12 VS1 <15> both low. This is done by setting U183-9 ACQ_DATA_HOLD(L):H, U183-10 DIAG_SERIAL_MODE:H, and writing 88*hex* to **TB Control Latch 5** (U250-19 CAL_SEL(H):H).

- Reset **Chop Transition Blanking** <15> by setting U210-1 INV_PT_RESET(L) low, then high. This forces U210-10 L_LAST_REQUEST(H):L and will force **Acq Mem Control Logic** U160-15 ACQ_MEM_R(H)/W_MODE(L) <13> low into the write mode when SYSTEM_RESET(L) goes high.

4. Start the acquisition write cycle by writing CB*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):H.

5. Perform a software delay to allow the acquisition memory (**Acq Mem Even Bank** and **Acq Mem Odd Bank** <13>), to be filled with digitized samples.

6. Stop the acquisition write cycle by writing 8B*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):L.

7. Read and average 64 locations (samples) from the acquisition memory to yield the final voltage measurement. Verify that this value falls within the expected limits.

8. Stop processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> for an output of +3.00 V (A60*hex*).

   - Close all analog switches (DG211's) in **Refresh Bank 1, Refresh Bank 2,** and **Refresh Bank 3** <12>.

   - Disable the periodic processor refresh interrupts.

Error Index G3621

The measured A/D voltage was outside of the acceptable limits.

Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 10 above).

See Also

The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

If the **Voltage Reference DAC** <12> R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the test may fail.

If an instrument calibration sequence has not taken place and the test uses its standard default values for the programmable reference voltages in the refresh banks, a failure may occur due to the default values not matching closely enough to instrument hardware tolerances. To ensure that good calibration constants are used for these references, exit **Extended Diagnostics**, run **Enhanced Accuracy**, and then rerun the test.

Routine Name

-0.800 V

Overview

This test verifies the operation of the A/D Converter <8>, **Voltage Reference DAC** <12>, **Refresh Bank 1** <12>, and CH SWREF of **Refresh Bank 3** <12> on the Acquisition board. The DAC is used to generate voltages, via CH SWREF, to **Sampler** U740 <6> (or <6A> for the 11402). The vertical select lines (VS0, VS1 of **Vertical Source Sequencer** <15>) are set up to feed the CH SWREF channel through the SAMPLER HYBRID U740 and to the input of the A/D Converter. Converted digital values are acquired and stored in **Acq Mem Even Bank** and **Acq Mem Odd Bank** <13> (just as they are in normal operation), where they are retrieved and averaged by the processor (**MPU** U224 <16>) to yield a voltage measurement.

Description

1. Start processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> (two bytes) with the voltage required for one of the reference voltages. This voltage is either a standard default value or one obtained from the Executive processor as a result of a previous calibration sequence (i.e., a calibration constant from the NVRAM).

   - Close one of the analog switches (DG211) in **Refresh Bank 1** <12>, **Refresh Bank 2** <12>, or **Refresh Bank 3** <12> for the voltage to be programmed for approximately 400 µs.

   - Open the analog switch and repeat the previous two steps for the remaining reference voltages in each bank.

   - For the duration of this test, periodically repeat this processor refresh (previous three steps) at an approximate rate of every 10 ms. This is accomplished through periodic processor interrupts.

2. Set **Refresh Bank 3** U1060-1 CH SWREF <12> to a value which should yield -0.800 volts at **1st Stage A/D** R642 CH SW OUT <8>.

3. Set up time base for acquisition.

   - Load **Main Postrecord Logic** U264 MAIN C CTR <18> with a high value (00000FFFF*hex* —35 bits), set **TB Control Latch 3** U212-12 SYSTEM_RESET(H):H <15>, and clock the low non-terminal count at U264-29 MAIN_POSTRECORD_TC(H) through **Last Request Generation** <18>. This helps ensure that **Chop Transition Blanking** U210-10 L_LAST_REQUEST<15> stays low later on so that the acquisition cycle stays in write mode.

   - Set **TB Control Latch 3** U260-19 SINGLE(H):H and U212-12 SYSTEM_RESET(H):L <15>.

   - Put system clocks into free-run mode by setting **System Clock Control** U67B-3 CLOCK_SOURCE_SELECT(H):L <14>.

   - Enable **Restart Control** <20> by setting U175A-4 SET_RESTART(L) and U175A-1 CLR_RESTART(L) both high.

- Fill all 32 locations in **Initial Wfm Id Encoder Table** <14> RAM (U22, U32) with 8*hex* in order to force U12-6 VALIDATE_ACQ_REQUEST(H) high.

- Set **TB Control Latch 4** U261-19 FORCE_LAST(H):L and U261-18 FORCE_MAIN(H):H <15>. This last bit causes **Acquisition Request Control** U193A-12 TB_REQUEST(H) <15> to go high and eventually **Vertical Source Sequencer** U183-14 SAMPLE_ENABLE(L) <15> to go low, thereby enabling the **Sample Gate Generator** <6>.

- Select the CAL channel as the **Sampler** U740 <6> vertical source input by setting **Vertical Source Sequencer** U183-12 VS0 and U183-12 VS1 <15> both low. This is done by setting U183-9 ACQ_DATA_HOLD(L):H, U183-10 DIAG_SERIAL_MODE:H, and writing 88*hex* to **TB Control Latch 5** (U250-19 CAL_SEL(H):H).

- Reset **Chop Transition Blanking** <15> by setting U210-1 INV_PT_RESET(L) low, then high. This forces U210-10 L_LAST_REQUEST(H):L and will force **Acq Mem Control Logic** U160-15 ACQ_MEM_R(H)/W_MODE(L) <13> low into the write mode when SYSTEM_RESET(L) goes high.

4. Start the acquisition write cycle by writing CB*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):H.

5. Perform a software delay to allow the acquisition memory (**Acq Mem Even Bank** and **Acq Mem Odd Bank** <13>), to be filled with digitized samples.

6. Stop the acquisition write cycle by writing 8B*hex* to **TB Control Latch 3** <15>. This sets SYSTEM_RESET(L):L.

7. Read and average 64 locations (samples) from the acquisition memory to yield the final voltage measurement. Verify that this value falls within the expected limits.

8. Stop processor refresh of programmable reference voltages.

   - Program the **Voltage Reference DAC** <12> for an output of +3.00 volts (A60*hex*).

   - Close all analog switches (DG211's) in **Refresh Bank 1**, **Refresh Bank 2**, and **Refresh Bank 3** <12>.

   - Disable the periodic processor refresh interrupts.

Error Index G3631 The measured A/D voltage was outside of the acceptable limits.

Refresh circuitry voltages can be checked at times when the processor is not periodically refreshing the programmable reference voltages, such as when no tests are running. At these times, the digitizer sets all reference voltages to +3.00 V (see step 10 above).

See Also The Calibration Output Accuracy procedure in the Checks and Adjustments section of the Service Reference Manual for calibration of the **Voltage Reference DAC** <12>.

Caveats

If the **Voltage Reference DAC <12>** R1582 GAIN and R1576 OFFSET adjustments are too far out of calibration, then the test may fail.

If an instrument calibration sequence has not taken place and the test uses its standard default values for the programmable reference voltages in the refresh banks, a failure may occur due to the default values not matching closely enough to instrument hardware tolerances. To ensure that good calibration constants are used for these references, exit **Extended Diagnostics**, run **Enhanced Accuracy**, and then rerun the test.

Routine Name

Holdoff

Overview

This test verifies the Main Holdoff Logic (mainly, U264 MAIN A CTR—35 bits, count-down; plus other logic) <18> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into five bytes and written byte-by-byte in least-significant-byte order; to do a read, the counter is read in least-significant-byte order, and then the five bytes reassembled into one 35-bit value (five MSBs masked out).
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.
- The 19 MHz clock is disabled during this test. Instead of being run full speed, it is toggled under control of the processor. It is toggled by having the processor first set **TB Control Latch 1** U251-18 D19 <15> low and then set it back high. This toggles U2050-4 19 MHz and U2050-7 19 BUF of **Multiplexer** <10>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the time base counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Main Rate Generator** U264-22 SYSTEM_RESET(L) <18> is set low.
   - **Main Postrecord Logic** U239D-12 RESTART(L) <18> is set low. This signal and its complement **Window 2 Post Record Logic** U287-3 RESTART(H) <18> control the clock and/or load lines of most time base counters.

- **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).
- **Trigger Source Register** U2030-5 19 MHz SEL(L) <10> is set high and **TB Control Latch** 1 U251-18 D19 <15> is set high (19 MHz is disabled).
- **TB Control Latch 3** U260-14 MHOLOAD(H) <15> is set high.
- **Holdoff Logic** U1546B-9 MHOLDOFFLD(L) <10> is set low.

2. Set up the hardware such that subsequent control of this counter's load signal will be enabled and as simply as possible.

   - Put main trigger into free-run mode (trigger always asserted) by serially writing a value of 03*hex* to the Main Shift Register of **Trigger Hybrid** U1710 <10>. This is only a precautionary step to help assure that **Trigger Signals Conditioner** U1452A-3 MHOTRIG(H) <10> remains asserted high during this test, which is required for control of this counter's load signal (effectively, **Holdoff Logic** U1546B-9 MHOLDOFFLD(L) <10>).
   - Set **Holdoff Logic** U1644A-1 STRGARM(L):H <10>.
   - Set **Holdoff Logic** U1452A-2 19 HOCALSEL(H):H <10>.
   - Set **Pretrigger Randomizer** U297-19 PRETC(H):H <19> by writing zero to the **Pretrigger Generator** U275 WIND C CTR <19> and toggling the system clocks 256 times. The large number of clocks is required to assure that **Pretrigger Generator** U286-12 PRETC_START(H) <19> will be propagated through the **Pretrigger Randomizer** <19>. See explanation of how **Pretrigger Generator** is written in **Pretrigger (G223X)** test description.
   - Toggle 19 MHz clock once to get a high state on **Holdoff Logic** U1644A-5 <10>.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 35 times). The test pattern starts out the first time as 400000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 35 patterns is: 400000000*hex*, 200000000*hex*, 100000000*hex*, 80000000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.
   - Toggle the 19 MHz clock once to get the test pattern actually clocked into the counter from its input latches.
   - Disable the load signal **Main Holdoff Logic** MHOLDOFFLD(L) <18> (set it high) by setting **TB Control Latch 3** U260-14 MHOLOAD(H):L <15>, toggling 19 MHz once, (**Holdoff Logic** U1550-1 MHORESET(L) <10> goes high), then toggling 19 MHz four more times.

- Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.

- In all but the last pass, toggle the 19 MHz clock once to cause a ripple count. In the last pass, a single toggle of the clock will cause the load signal to be enabled again because the terminal count bit should go high; however, this is because of the feedback of **Main Holdoff Logic** U237A-6 <18> to U247C-10 and does not involve MHOLDOFFLD(L). In order to get MHOLDOFFLD(L) back to a disabled (high) state for the rest of this test, the 19 MHz clock is toggled four more times.

- Read and check the terminal count bit. It should be 0. Normally for these time base tests it should be 1 on the last (see other test descriptions); however, this test holds an exception because of the fact that the asssertion of the terminal count itself causes the counter to automatically assert its load line again before the terminal count can be read back. When the load line is asserted low, then the terminal count will always read 0 even if the counter has internally counted down to zero. So in effect, the terminal count pulses high and low within one clock cycle of the 19 MHz clock. Consequently, the terminal count will be read back as 0 even on the last pass.

- Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.

- Shift the test pattern right one bit.

- Enable the load signal MHOLDOFFLD(L) (set it low) by setting **TB Control Latch 3** U260-14 MHOLOAD(H):H <15>.

VERIFY LOAD ZERO

4. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load signal as previously described.

   - Load counter input latches with zero.

   - Toggle the 19 MHz clock once to actually load the counter from its input latches.

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G4111 Terminal count error just after counter was loaded with the test pattern.

Error Index G4112 Error in the upper three bits of the counter contents just after it was loaded with the test pattern.

Error Index G4113 Error in the middle 16 bits of the counter contents just after it was loaded.

Error Index G4114 Error in the lower 16 bits of the counter contents just after it was loaded.

Error Index G4115 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G4116 Error in upper three bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4117 Error in middle 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4118 Error in lower 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4119 Terminal count error just after counter was loaded with zero.

Error Index G411A Error in upper three bits of counter contents just after counter was loaded with zero.

Error Index G411B Error in middle 16 bits of counter contents just after counter was loaded with zero.

Error Index G411C Error in lower 16 bits of counter contents just after counter was loaded with zero.

Routine Name

Delay Events

Overview

This test verifies the **Delay By Events Logic** (mainly, U264 MAIN F CTR—35 bits, count-down; plus other logic) <18> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into five bytes and written byte-by-byte in least-significant-byte order; to do a read, the counter is read in least-significant-byte order, and then the five bytes reassembled into one 35-bit value (five MSBs masked out).
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.
- The 19 MHz clock is disabled during this test. Instead of being run full speed, it is toggled under control of the processor. It is toggled by having the processor first set **TB Control Latch 1** U251-18 D19 <15> low and then set it back high. This toggles U2050-4 19 MHz and U2050-7 19 BUF of **Multiplexer** <10>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the timebase counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Main Rate Generator** U275-22 SYSTEM_RESET(L) <19> is set low.
   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

- **Trigger Source Register** U2030-5 19 MHz SEL(L) <10> is set high and **TB Control Latch 1** U251-18 D19 <15> is set high (19 MHz is disabled).

- This counter's load line **Delay By Events Logic** U264-3 PRETC(H) is set low.

2. Set up the hardware such that subsequent control of this counter's load signal will be enabled and as simple as possible.

   - Put Main trigger into free-run mode (trigger always asserted) by serially writing a value of 03*hex* to the Main Shift Register of U1710 **Trigger Hybrid** <10>.

   - Set **Holdoff Logic** U1644A-1 STRGARM(L):H <10>.

   - Set **Delay By Events Logic** U264-12 WINDOW_SELECT:L <18>.

   - Load serially **Delay By Events** U1630 <10> with F2*hex* so that U1530-15 will be high (which enables the clock path to U1540) and U1540 will be loaded with its maximun number (F*hex*) so that its terminal count (U1540-4) can be asserted with a single clock pulse.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 35 times). The test vector starts out the first time as 400000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 35 patterns is: 400000000*hex*, 200000000*hex*, 100000000*hex*, 80000000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.

   - Toggle the clock to this counter by toggling **Delay By Events Logic** PRETC(H) <18>. PRETC(H) is toggled by first setting it high and then setting it low again. PRETC is set high by writing zero to the **Pretrigger Generator** (U275 WIND C CTR) <19> and toggling the system clocks 256 times. The large number of clocks is required to assure that **Pretrigger Generator** U286-12 PRETC_START(H) <19> will be propagated through the **Pretrigger Randomizer** <19>. PRETC(H) is set low by writing FFFF*hex* to the same counter and toggling the system clocks two times.

   - Disable the load signal PRETC(L) by setting **TB Control Latch 2** U260-14 MHOLOAD(H):L <15>, toggling 19 MHz once, (**Holdoff Logic** U1550-1 MHORESET(L) <10>goes high), setting PRETC (H):H, and then toggling the 19 MHz clock either four times (if the first pass), or three times (for all other passes).

   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.

   - Toggle **Delay By Events Logic** U277B-4 ECLK <18> once to cause a ripple count of this counter. This is done by setting **TB Control Latch 1** U251-13 DIAG_EVENTS_CLK low and then high, then toggling the 19 MHz clock twice.

- Read and check the terminal count bit. It should be 1.
- Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.
- Shift the test pattern right one bit.
- Enable the load signal PRETC by setting MHOLDOFFLD(L):H , and then setting PRETC(H):L as previously described.

VERIFY LOAD ZERO

4. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.
   - Enable the load signal as previously described.
   - Load counter input latches with zero.
   - Toggle the PRETC once to actually load the counter from its input latches.
5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G4121 Terminal count error just after counter was loaded with the test pattern.

Error Index G4122 Error in the upper three bits of the counter contents just after it was loaded with the test pattern.

Error Index G4123 Error in the middle 16 bits of the counter contents just after it was loaded.

Error Index G4124 Error in the lower 16 bits of the counter contents just after it was loaded.

Error Index G4125 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G4126 Error in upper three bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4127 Error in middle 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4128 Error in lower 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4129 Terminal count error just after counter was loaded with zero.

Error Index G412A Error in upper three bits of counter contents just after counter was loaded with zero.

Error Index G412B Error in middle 16 bits of counter contents just after counter was loaded with zero.

Error Index G412C Error in lower 16 bits of counter contents just after counter was loaded with zero.

Routine Name

Trig to Trig (Trigger-to-Trigger)

Overview

This test verifies the **Trig To Trig Measurement** (mainly, U275 WIND A CTR—35 bits, count-down; plus other logic) <19> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into five bytes and written byte-by-byte in least-significant-byte order; to do a read, the counter is read in least-significant-byte order, and then the five bytes reassembled into one 35-bit value (five MSBs masked out).
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.
- The 19 MHz clock is disabled during this test. Instead of being run full speed, it is toggled under control of the processor. It is toggled by having the processor first set **TB Control Latch 1** U251-18 D19 <15> low and then set it back high. This toggles U2050-4 19 MHz and U2050-7 19 BUF of **Multiplexer** <10>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the time base counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - Window Rate Generator U275-22 SYSTEM_RESET(L) <19> is set low.
   - Pretrigger Generator U239C-10 RESTART(L) <19> is set low. This signal and its complement **Window 1 Post Record Logic** U278-3 RESTART(H) <19> control the clock and/or load lines of most time base counters.

- **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).
- **Trigger Source Register** U2030-5 19 MHz SEL(L) <10> is set high and **TB Control Latch 1** U251-18 D19 <15> is set high (19 MHz is disabled).
- **Trig To Trig Measurement** U245C-10 MAQRESET_BUF(L) <19> is set low.
- **Trig To Trig Measurement** U254A-2 MCG(H) is set low and U254A-1 WCG(L) is set high.
- **TB Control Latch 3** U260-14 MHOLOAD(H) <15> is set high, which assures that **Holdoff Logic** U1550-1 MHORESET(L) <10> is low.

2. Set up the hardware such that this counter's enable signal (**Trig To Trig Measurement** U275-16) will be enabled high.
   - Write a value to the location of the **Destination Address Generator** (U166 <20>) Control Register 0 such that bit D5 is set high; this causes the diagnostic multiplexer control bit (DiagMpx) to be asserted.
   - Write a value to the location of the **Destination Address Generator** Control Register 2 such that bit D1 (ti_avail) is set high; this causes U226F-13 TI_DATA_AVAIL(H) <19> to go low and U226F-12 TI_DATA_AVAIL(L) <19> to go high.
   - Set **TB Control Latch 3** U260-17 MSET(H):H.
   - Set **Window Trigger Logic** U258-1 RESTART(L):H by setting **TB Control Latch 0** U240-19 SET_RESTART(L):H <15> and **TB Control Latch 0** U221B-4 CLR_RESTART(L):L <15>.
   - Toggle system clocks once. At this point U254A-2 should go high so that now U254A-12 should be high.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 35 times). The test vector starts out the first time as 400000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 35 patterns is: 400000000*hex*, 200000000*hex*, 100000000*hex*, 80000000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.
   - Toggle the system clocks once to get the test pattern actually clocked into the counter from its input latches.
   - Disable the load signal MAQRESET_BUF(L) (set it high) by setting U260-14 MHOLOAD(H):L <15>, toggling 19 MHz once (**Holdoff Logic** MHORESET(L) <10> goes high), setting PRETC (H):H, then toggling 19 MHz four more times. PRETC is set high by writing zero to the **Pretrigger Generator** (U275 WIND C CTR) <19> and toggling the system clocks 256 times. The large number of clocks is required to assure that U286-12 PRETC_START(H) <19> will be propagated through the **Pretrigger Randomizer <19>.**
   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.
   - Toggle the system clocks once to cause a ripple count.
   - Read and check the terminal count bit. It should be 1.
   - Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.
   - Shift the test pattern right one bit.
   - Enable the load signal MAQRESET_BUF(L) (set it low) by setting U260-14 MHOLOAD(H):H <15> and setting PRETC(H):L. PRETC is set low by writing FFFF*hex* to the **Pretrigger Generator** (U275 WIND C CTR) <19> and toggling the system clocks two times to assure that U286-12 PRETC_START(H) <19> will be propagated through the **Pretrigger Randomizer <19>.**

VERIFY LOAD ZERO

4. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.
   - Enable the load signal as previously described.
   - Load counter input latches with zero.
   - Toggle the system clocks once to actually load the counter from its input latches.
5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G4131 Terminal count error just after counter was loaded with the test pattern.

Error Index G4132 Error in the upper three bits of the counter contents just after it was loaded with the test pattern.

Error Index G4133 Error in the middle 16 bits of the counter contents just after it was loaded.

Error Index G4134 Error in the lower 16 bits of the counter contents just after it was loaded.

Error Index G4135 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G4136 Error in upper three bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4137 Error in middle 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4138 Error in lower 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4139 Terminal count error just after counter was loaded with zero.

Error Index G413A Error in upper three bits of counter contents just after counter was loaded with zero.

Error Index G413B Error in middle 16 bits of counter contents just after counter was loaded with zero.

Error Index G413C Error in lower 16 bits of counter contents just after counter was loaded with zero.

Routine Name

Holdoff

Overview

This test verifies the **Window Holdoff Logic** (mainly, U275 WIND F CTR—35 bits, count-down; plus other logic) <19> on the Time Base board. It checks the following functional characteristics of the counter:

- Writability/Countability—This is a combination of a "walking 1's" test and a check to see if it will count down from each "walking 1's" value when clocked once (to see if ripple carries work for each counter bit stage).
- Load zero—A check to see if the counter can be loaded with zero successfully.
- Terminal count bit—While the previous items are being tested, the terminal count bit read out of the diagnostic terminal count register is checked every time the counter contents are read to see if it behaves properly.

Description

There are some basic characteristics of how this test is executed which should be noted.

- Any reading and writing of this counter is done one byte at a time; to do a write, the value must be broken up into five bytes and written byte-by-byte in least-significant-byte order; to do a read, the counter is read in least-significant-byte order, and then the five bytes reassembled into one 35-bit value (five MSBs masked out).
- This counter is read back as the 1's complement of the actual contents (the bits are inverted).
- If this counter 's contents are read after it has counted down to zero, you will NOT read the 1's complement of zero (1) as might be expected, but rather will read back the 1's complement of the last value loaded. The only way to detect 0 in this counter's contents is to actually LOAD zero into it first.
- System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.
- The 19 MHz clock is disabled during this test. Instead of being run full speed, it is toggled under control of the processor. It is toggled by having the processor first set **TB Control Latch 1** U251-18 D19 <15> low and then set it back high. This toggles U2050-4 19 MHz and U2050-7 19 BUF of **Multiplexer** <10>.

1. As part of the general hardware initialization that occurs on entry to this test, the following significant signals are set to the following states:
   - The terminal counts for the time base counter IC's (U264 <18> & U275 <19>) are low (all counters loaded with FFFF*hex*).
   - **Main Rate Generator** U275-22 SYSTEM_RESET(L) <19> is set low.
   - **System Clock Control** U57-4 CLOCK_SOURCE_SELECT <14> is set high (20 MHz disabled).

- **Trigger Source Register** U2030-5 19 MHz SEL(L) <10> is set high and **TB Control Latch** 1 U251-18 D19 <15> is set high (19 MHz is disabled).
- **TB Control Latch** 3 U260-14 MHOLOAD(H) is set high.
- **Window Holdoff Logic** U255B-11 MHOLDOFFLD(L) and U255B-13 MAQRESET_BUF(L) <19> are set low.

2. Set up the hardware such that subsequent control of this counter's load signal will be enabled and as simple as possible.

   - Put Main trigger into free-run mode (trigger always asserted) by serially writing a value of 03*hex* to the Main Shift Register of U1710 **Trigger Hybrid** <10>. This is only a precautionary step to help assure that U1452A-3 MHOTRIG(H) <10> remains asserted high during this test, which is required for control of this counter's load signal.
   - Set **Holdoff Logic** U1644A-1 STRGARM(L):H <10>.
   - Set **Window Holdoff Logic** U275-12 WINDOW_SELECT:L <19>.
   - Set **Trigger Signals Conditioner** U1452A-2 HOCALSEL(H):H <10>.

VERIFY WRITABILITY/COUNTABILITY

3. Do a "walking 1's" test, coupled with a ripple countability test, on the counter (so it loops 35times). The test vector starts out the first time as 400000000*hex* and then is shifted right one bit on each subsequent pass, i.e., the set of 35 patterns is: 400000000*hex*, 200000000*hex*, 100000000*hex*, 80000000*hex*, ... 8*hex*, 4*hex*, 2*hex*, 1*hex*. Each pass is done in the following manner:

   - Load test pattern into the counter input latches.
   - Toggle the 19 MHz clock once to get the test pattern actually clocked into the counter from its input latches.
   - Disable the load signal by setting **TB Control Latch 3** U260-14 MHOLOAD(H):L <15>, toggling 19 MHz once, (**Holdoff Logic** U1550-1 MHORESET(L) <10> goes high), setting PRETC (H):H, and then toggling the 19 MHz clock either five times (if the first pass), or four times (for all other passes). PRETC is set high by writing zero to the **Pretrigger Generator** (U275 WIND C CTR) <19> and toggling the system clocks 256 times. The large number of clocks is required to assure that U286-12 PRETC_START(H) <19> will be propagated through the **Pretrigger Randomizer** <19>.
   - Read the terminal count bit and check that it is 0. Read the counter contents and check that it matches the 1's complement of the test pattern that was loaded.
   - Toggle the 19 MHz clock once to cause a ripple count.
   - Read and check the terminal count bit. It should be 1.

- Read the counter contents. In all cases except the last pass, check that it matches a value EQUAL to the test pattern. Since the counter should have counted down by one, its actual contents should equal the test pattern value minus 1, e.g., if the test pattern was 800*hex* then the counter contents should be 7FF*hex* (one less); however, since the counter is read as a 1's complement, the value read back will be the 1's complement of 7FF*hex*, or 800*hex*. However, on the last pass (when the counter reaches zero), it is not possible to read back the zero, but only the 1's complement of the last value that was loaded, namely, the 1's complement of 1, which is 0.

- Shift the test pattern right one bit.

- Enable the load signal by setting MHOLDOFFLD(L):H , and then setting PRETC(H):L as previously described.

VERIFY LOAD ZERO

4. Assert the load line, write zero to the counter, de-assert the load line, and then clock the counter, to cause the counter to be loaded from its input latches.

   - Enable the load signal as previously described.

   - Load counter input latches with zero.

   - Toggle the 19 MHz clock once to actually load the counter from its input latches.

5. Read and check that the terminal count bit is 1 and the counter contents is read as the 1's complement of 0 (1), as previously explained.

Error Index G4211 Terminal count error just after counter was loaded with the test pattern.

Error Index G4212 Error in the upper three bits of the counter contents just after it was loaded with the test pattern.

Error Index G4213 Error in the middle 16 bits of the counter contents just after it was loaded.

Error Index G4214 Error in the lower 16 bits of the counter contents just after it was loaded.

Error Index G4215 Terminal count error just after counter was clocked to cause a ripple count.

Error Index G4216 Error in upper three bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4217 Error in middle 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4218 Error in lower 16 bits of the counter contents just after it was clocked to cause a ripple count during the writability/countability part of this test.

Error Index G4219 Terminal count error just after counter was loaded with zero.

Error Index G421A Error in upper 3 bits of counter contents just after counter was loaded with zero.

Error Index G421B Error in middle 16 bits of counter contents just after counter was loaded with zero.

Error Index G421C Error in lower 16 bits of counter contents just after counter was loaded with zero.

Routine Name TI Register (Time Interpolator Register)

Overview This test does a partial verification of the **Time Interpolator Data Latches** (U167, U176) <20> on the Time Base board by doing a series of serial read/write operations on it via the processor.

Description There is a characteristic of how this test is executed which should be noted.

- This 16-bit register is actually made up of two 8-bit registers; serial operations are done by doing serial operations on both 8-bit registers simultaneously. The input serial data (**Time Interpolator Data Latches** U167-11 SERIAL_DATA_LOW_IN & U176-11 SERIAL_DATA_HIGH_IN <20>) come from bits D7 and D15 of the data bus, respectively; the output serial data (**Time Interpolator Data Latches** U167-14 TID_OUT_LO & U176-14 TID_OUT_HI) go to bits D0 and D8 of the data bus, respectively.

1. Put the **Time Interpolator Data Latches** <20> into serial mode.
   - Set **Time Interpolator Data Latches** U167-23 & U176-23 TID_MODE:L <20>.
2. Load the register serially, then read it back serially and check to see if the read value matches the written value. The values used, and the order in which they are used, are as follows:

   AAAA*hex*, CCCC*hex*, F0F0*hex*, FF00*hex*, 5555*hex*, AAAA*hex*

Error Index G5111 The value read back did not match the value written.

Routine Name

Data Lines

Overview

This test verifies the data lines to the **Destination Address Generator** U166 <20> (DAG) on the Time Base board. It does this by doing a set of write/read operations on the Spcw (window spacing) value memory register, in the DAG IC.

Description

There are characteristics of how this test is executed which should be noted.

- The control registers are readable at all times, but the value memory registers are readable only if the Fine Shift Register, Read Enable bit of DAG Control Register 0 (D4) is set low. Consequently, DAG Control Register 0 is written with a value of zero at the beginning so that Spcw can be read.

- Since the value memory registers are 19-bits, and since the data bus to the DAG is only 8-bits, any write and read operations for these registers require that the 19-bit values be broken up into three 8-bit pieces before writing and that the three 8-bit pieces read back must be reconstructed into one 19-bit value. Writing and reading are done in least-significant-byte-first order.

1. As part of the general hardware initialization that occurs on entry to this test, following significant signals are set to the following states:

   - **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> is set low, so that **Destination Address Generator** U166-34 SYSTEM_RESET(L) <20> is low.

   - System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

2. Write, read, and verify the following set of patterns from the DAG Spcw value memory register:

   2AAAA*hex*, 4CCCC*hex*, 0F0F0*hex*, 0FF00*hex*, 70000*hex*, 55555*hex*, 2AAAA*hex*.

Error Index G5211

The value read back from the Spcw value memory register did not match the value written.

Routine Name

Address/Data

Overview

This test verifies the programmability of the **Destination Address Generator** U166 <20> (DAG) on the Time Base board. It does this by executing a RAM-like test on the complete set of DAG registers—this set of registers is made up of the 11 value memory registers (each 19 bits) plus the 5 DAG Control Registers (each 8 bits).

Description

There are characteristics of how this test is executed which should be noted.

- The control registers are readable at all times, but the value memory registers are readable only if the Fine Shift Register, Read Enable bit of the DAG Control Register 0 (D4) is set low. Consequently, whenever the value memory registers are to be read, DAG Control Register 0 is written with a value of zero; also, a copy of the real contents of this register must be maintained so that it also can also be tested.

- Since the value memory registers are 19-bits, and since the data bus to the DAG is only 8-bits, any write and read operations for these registers require that the 19-bit values be broken up into three 8-bit pieces before writing and that the three 8-bit pieces read back must be reconstructed into one 19-bit value. Writing and reading are done in least-significant-byte-first order.

- As in a regular RAM test, the address space spanned by all the registers of the DAG is tested basically by doing a set of write/read operations on each location. A specific test vector is used, with the individual test patterns used in the order shown here:

  2AAAA*hex*, 4CCCC*hex*, 0F0F0*hex*, 0FF00*hex*, 70000*hex*, 55555*hex*, 2AAAA*hex*.

  These patterns allow for thorough data bit testing of the 19-bit wide registers. The 8-bit control registers use the least significant 8-bits of each pattern, e.g., AA*hex* for the 2AAAA*hex* pattern.

1. As part of the general hardware initialization that occurs on entry to this test, following significant signals are set to the following states:

   - **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> is set low, so that **Destination Address Generator** U166-34 SYSTEM_RESET(L) <20> is low.

   - System clocks (20 MHz) are disabled during this test. Instead of being run full speed, they are toggled under control of the processor. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) of **System Clock Control** <14>.

2. Initialize the hardware such that SYSTEM_RESET(L) and RESTART(L) are asserted low.

   - Set U260-18 SYSTEM_RESET(L):L <15> and clock the system clocks once. System clocks are toggled by doing a processor read operation at location 4884*hex*. This toggles CLOCK_STEP(L) <14 > of **System Clock Control** <14>.

- Set **Destination Address Generator** U166-64 RESTART(L) <20> low by setting **TB Control Latch 0** U240-19 SET_RESTART(L):L <15> and **TB Control Latch 0** U221B-4 CLR_RESTART(L):H <15>.

2. Fill all registers with the first test pattern.

   - For each value memory location write the test pattern into the register as three successive bytes as previously explained.

   - For each control register location, write the least significant byte of the test pattern.

3. Now do a repetitive pass through the address space of these DAG registers. The number of passes is equal to the number of test patterns used in this test (seven). For each pass, and for each register location within the pass: first read and check the contents of the register and compare it with what was previously written; then write the next test pattern into that location and go on to the next register. Each pass begins with the lowest addressed register (Spcw—window spacing—location 4988*hex*) and counts up through the highest addressed register (DAG Control Register 4 —location 49FA*hex*).

   - If the location is the first (lowest) value memory register, then save the present value of DAG Control Register 0 in a temporary location and then write zero to that control register. This is required so that value memory registers will be readable.

   - If the location is the first (lowest) control register (its control register 0), then restore the present value of DAG control register 0 to it from its temporary location.

   - If the location is a value memory register control, then read it as three successive bytes and check it. If it's not the last pass, then write the next test pattern into this location.

   - If the location is a control register, then read it as one byte and check it. If it's not the last pass, then write the next test pattern into this location.

Error Index G5221 Error in a write/read operation in a DAG IC value memory register.

Error Index G5222 Error in a write/read operation in a DAG IC control register.

Routine Name

Data

Overview

This test verifies part of the waveform data path by loading the **Acq Mem Data Accumulator** <14> with "walking one" test patterns, clocking them through the **Acq Mem Data Input Latch** <14>, **Acq Mem Output Latch** <14>, **Acq Mem DAG Output Latch** <14>, **Left Justifier** <14>, and **Wfm Data Output Latch** <17> circuits, and then reading back waveform data patterns through the **Digitizer/MMU Communication Port** <17>. In doing so, small portions of the **DAG** <20> and **Acq Mem Control Logic** <13> circuits are tested.

Description

1. Initialize hardware for loop back operation.

   - Reset the digitizer hardware by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> low.
   - Initialize the state of the **DAG** <20> (especially pin 68, DAG_REQUEST(L) high) by setting the DAG IC U166 to diagnostic mode via setting bit 5 high in DAG IC Control Register 0 and then setting **TB Control Latch 0** U240-19 SET_RESTART(L):H and U240-18 CLR_RESTART(L):H <15>.
   - Force the system clocks into normal free-running mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):L <14>.
   - Initialize the **Initial Wfm Id Encoder Table** <14> to prevent any acquisition memory requests (i.e., force U12-6 VALIDATE_ACQ_REQUEST(H) low) by writing 0_{hex} to all RAM locations in U22 and U32.
   - Initialize the state of **Acq Mem Control Logic** <13> and **Acq Mem Address Generation** <13> by setting **TB Control Latch 1** U251-14 INV_PT_RESET(L):H <15> and performing a read and write to **Acq Mem Even Bank** <13>.
   - Enable communication handshaking by setting **Digitizer/MMU Handshake Logic** U83C-11 DIGREQ_EN(H):H <17>.
   - Enable the diagnostic loop back function by setting **Diagnostic Loopback Control** U146B-19 LOOPBACK_EN(L):L <17>.
   - Force the system clocks into single-step mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):H <14>.
   - Initialize **Digitizer/MMU Communication Port** <17> to a known state by toggling **Diagnostic Loopback Control** <17> U85C-9 FORCE_DATA(L) and reading and writing the communication port latches U106 and U126.
   - Set the **Acq Mem Data Accumulator** <14> to serial mode by setting **TB Control Latch 0** U240-6 ACQ_DATA_HOLD(L):L and U240-16 DIAG_SERIAL_MODE(H):L <15>.

- Take the digitizer hardware out of reset by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L):H <15>.

2. Set up **DAG** <20> for acquisition transfer cycle (DAG_REQUEST(L):L).

   - Enable buffer U210 by setting **Chop Transition Blanking** U210-1 INV_PT_RESET(L):H <15>.

   - Force **DAG** U175B-9 ACQ(H)/XFER(L):H <20> by setting **Restart Control** U175A-4 SET_RESTART(L):L and U175A-1 CLR_RESTART(L):H. This forces U175A-6 RESTART(L):L.

   - Set **Chop Transition Blanking** U210-11:H <15> by serially shifting a high into **Last Request Control** register U202-15, and then toggling the system clocks, via **System Clock Control** CLOCK_STEP(L) <14>, three times.

   - Release the forced high level of **DAG** U175B-9 ACQ(H)/XFER(L) <20> by setting U175A-4 SET_RESTART(L):H and U175A-1 CLR_RESTART(L):L. This forces U175A-6 RESTART(L):H.

   - Set **DAG** U175B-9 ACQ(H)/XFER(L):L <20> by toggling **Chop Transition Blanking** U210-1 INV_PT_RESET(L) <15> low to high. This drives U210-10 L_LAST_REQUEST(H) low <14>, and after one system clock **Acq Mem Control Logic** U160-15 ACQ_MEM_R(H)/W_MODE(L) <13> is driven low. This also clocks the high previously set at U210-11 through to U210-10 L_LAST_REQUEST(H). After one more system clock, the high present at U160-14 causes **DAG** U175B-11 ACQ_MEM_R(H)/W_MODE(L) to transition from low to high, thereby clocking a low into U175B-9 ACQ(H)/XFER(L).

   - Set **Acq Mem Control Logic** U161C-9 DAG_REQUEST(L) <13> low by clocking the system clocks once more. DAG_REQUEST(L) will stay low for exactly one system clock period.

3. Enable data path through **Acq Mem Data Input Latch** <14>.

   - Set **Chop Transition Blanking** U210-1 INV_PT_RESET(L):L <15> which drives U210-10 L_LAST_REQUEST low. After the next system clock pulse in step 5, the **Acq Mem Control Logic** U163-19 ACQ_DATA_BUFFER_ENABLE(L) <13> will go low and enable the outputs of the **Acq Mem Data Input Latch** <14>.

4. Write test data pattern (see Error Index section below) into **Acq Mem Data Accumulator** <14>.

   - Serially load the **Acq Mem Data Accumulator** (U15, U25, U24, U14) with a test data pattern by setting the shift registers to serial load mode, clocking the serial data in and setting the shift registers back to parallel mode.

5. Clock the data from the **Acq Mem Data Accumulator** <14> into the **Wfm Data Output Latch** <17>.

- Clock the data from the **Acq Mem Data Accumulator** through the **Acq Mem Data Input Latch** <14>, **Acq Mem Output Latch** <14>, **Acq Mem DAG Output Latch** <14>, and **Left Justifier** <14> circuits to the input of the **Wfm Data Output Latch** <17> by toggling the system clocks four times. **System Clock Control** U66A-2 DATA_AVAIL(H) <14> should be high after two clocks and stay high for only one clock period.

- Clock the data into **Wfm Data Output Latch** <17> by setting U104-11 and U114-11 WFM_REQ(H):H <17> via setting bit 0 low in DAG IC Control Register 4.

6. Clock the data from the **Wfm Data Output Latch** <17> into the **Digitizer/MMU Communication Port** <17>.

   - Toggle **Diagnostic Loopback Control** U85C-9 FORCE_DATA. When FORCE_DATA goes low, **Digitizer/MMU Handshake Logic** U76A-3 WFM_DATA_EN(L) goes low and enables the outputs of **Wfm Data Output Latch**. When FORCE_DATA goes high, the data from U104 and U114 is latched into the **Digitizer/MMU Communication Port** input latches U106 and U126.

   - Set **Wfm Data Output Latch** U104-11 and U114-11 WFM_REQ(H):L by setting bit 0 high in DAG IC Control Register 4.

7. Read the data from **Digitizer/MMU Communication Port** input latches U106 and U126 and verify that it matches the expected data pattern (see Error Index section below).

8. Repeat steps 2-7 for the remaining test data patterns (see Error Index section below).

9. Terminate loop back operation and reset digitizer hardware.

   - Initialize the operating state of the communication latches and purge any pending DMA requests by writing a value (which is insignificant) to the **Digitizer/MMU Communication Port** output latches U116 and U136, toggling U85C-9 FORCE_DATA(L), and then reading the communication input latches U106 and U126.

   - Disable the loop back function by setting U146B-19 LOOPBACK_EN(L):H <17>, U83C-11 DIGREQ_EN(H):L, and U166-2 WFM_REQ(H):L <20> (by setting bit 0 high in DAG IC Control Register 4).

   - Reset the digitizer hardware by setting **TB Control Latch** 3 U212-12 SYSTEM_RESET(L) <15> low.

Error Index G5311 The display shows the first data pattern read from the **Digitizer/MMU Communication Port** <17> which did not match the expected data pattern. The data patterns written to the **Acq Mem Data Accumulator** <14> and the expected data patterns read back are shown in the following table:

Accumulator Data Pattern	Expected Data Pattern
0000*hex*	8001*hex*
0001*hex*	8001*hex*
0002*hex*	8001*hex*
0004*hex*	8001*hex*
0008*hex*	8001*hex*
0010*hex*	8080*hex*
0020*hex*	8100*hex*
0040*hex*	8200*hex*
0080*hex*	8400*hex*
0100*hex*	8800*hex*
0200*hex*	9000*hex*
0400*hex*	A000*hex*
0800*hex*	C000*hex*
1000*hex*	0000*hex*
2000*hex*	8001*hex*
4000*hex*	8001*hex*
8000*hex*	8001*hex*

Caveats

This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Digitizer busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name

Address/Data

Overview

This test verifies part of the waveform address path by loading the **Time Interpolator Data Latches** <20> with "walking one" test patterns, clocking them to the **DAG/MMU Wfm Address Output Latch** <17>, and then reading back the waveform addresses through the **Digitizer/MMU Communication Port** <17>. Also, part of the waveform tag path is verified by loading the **Final Wfm Id Encoding Table** <20> with "walking one" test patterns, clocking them to the **Wfm Tag Output Latch** <17>, and then reading back the waveform tags through the **Diagnostic Wfm Tag Readback Latch** <17>. In conjuction, a small part of the **DAG** <20> is tested.

Operator Procedure

There are two bits, FORCE_INVALID(L) and ADR_HOLD(H), from the **Final Wfm Id Encoding Table** <20> that are not automatically verified in this test. The last two test patterns of the waveform tag test patterns (see second Error Index section below) are present in order to toggle these two bits. External test equipment can be used to verify that they are at least being driven high and low to give some confidence that they are not faulty.

Description

1. Initialize hardware for loop back operation.

   - Reset the digitizer hardware by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> low.
   - Initialize the state of the **DAG** <20> (especially pin 68, DAG_REQUEST(L) high) by setting the DAG IC U166 to diagnostic mode via setting bit 5 high in DAG IC Control Register 0 and then setting **TB Control Latch 0** U240-19 SET_RESTART(L):H and U240-18 CLR_RESTART(L):H <15>.
   - Force the system clocks into normal free-running mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):L <14>.
   - Initialize the **Initial Wfm Id Encoder Table** <14> to prevent any acquisition memory requests (i.e. force U12-6 VALIDATE_ACQ_REQUEST(H) low) by writing 0_{hex} to all RAM locations in U22 and U32.
   - Initialize the state of **Acq Mem Control Logic** <13> and **Acq Mem Address Generation** <13> by setting **TB Control Latch 1** U251-14 INV_PT_RESET(L):H <15> and performing a read and write to **Acq Mem Even Bank** <13>.
   - Enable communication handshaking by setting **Digitizer/MMU Handshake Logic** U83C-11 DIGREQ_EN(H):H <17>.
   - Enable the diagnostic loop back function by setting **Diagnostic Loopback Control** U146B-19 LOOPBACK_EN(L):L <17>.
   - Force the system clocks into single-step mode by setting **System Clock Control** U57-4 CLOCK_SOURCE_SELECT(H):H <14>.

- Initialize **Digitizer/MMU Communication Port** <17> to a known state by toggling **Diagnostic Loopback Control** <17> U85C-9 FORCE_DATA(L) and reading and writing the communication port latches U106 and U126.

2. Load **Time Interpolator Data Latches** <20> with waveform address test pattern (see first Error Index section below).

   - Set **Time Interpolator Data Latches** U167-23 & U176-23 TID_MODE(H):L to enable serial loading of the shift registers. Serially load eight bits of data into each shift register by writing the data to U167-11 SERIAL_DATA_LOW_IN & U176-11 SERIAL_DATA_HIGH_IN and clocking U167-2 & U176-2 TID_CLK.

   - Load the serial data into the parallel latches of U167 & U176 by performing the following steps. Set **Restart Control** U175A-4 SET_RESTART(L):L and U175A-1 CLR_RESTART(L):H <20>. Toggle the system clocks, via **System Clock Control** U67C-5 CLOCK_STEP(L) <14>, four times. Set **Restart Control** U175A-4 SET_RESTART(L):H and U175A-1 CLR_RESTART(L):L. Set **Main Trigger Logic** U248B-4 MSET(H):H <18> and **Window Trigger Logic** U268C-10 WSET(H):H <19>. Toggle the system clocks once to cause **DAG** U166-32 MCG(L) & U166-67 WCG(L) <20> both to go low. Toggle the system clocks twice more to cause **Time Interpolator Data Latches** U176-1 & 167-1 COEN(L) to go low. Set **Time Interpolator Data Latches** U167-23 & 176-23 TID_MODE(H):H to enable parallel mode of the registers. Set **Restart Control** U175A-4 SET_RESTART(L):L and U175A-1 CLR_RESTART(L):H and then toggle the system clocks four times to force U176-1 & 167-1 COEN(L) high, thereby clocking the internal serial data into the parallel output latches.

   - Enable the parallel outputs of **Time Interpolator Data Latches** U167 & 176 by setting U167-1 & U176-1 COEN(L):L as outlined above.

3. Load the **Final Wfm Id Encoding Table** <20> with the waveform tag test pattern (see second Error Index section below) by writing the pattern to all RAM locations in U214/U215 and U204/U205.

4. Clock the waveform address from the **Time Interpolator Data Latches** <20> into the **DAG/MMU Wfm Address Output Latch** <17> and the waveform tag from the **Final Wfm Id Encoding Table** <20> into the **Wfm Tag Output Latch** <17>.

   - Set **Wfm Tag Output Latch** U96-7 <17> and **DAG/MMU Wfm Address Output Latch** U105-11 & U135-11 <17> high by setting **DAG** U166-2 WFM_REQ(H):H <20> (via setting bit 0 low in DAG IC Control Register 4). This provides the necessary clock pulse for the latches and also causes **Digitizer/MMU Handshake Logic** U75B-9 WFM_ADDR_EN(L) <17> to go low and enable the outputs of **DAG/MMU Wfm Address Output Latch** and **Wfm Tag Output Latch**.

5. Clock the waveform address into the **Digitizer/MMU Communication Port** <17> and the waveform tag into the **Diagnostic Wfm Tag Readback Latch** <17>.

   - Toggle **Diagnostic Loopback Control** U85C-10 FORCE_ADDR(L). When FORCE_ADDR goes low, the **Digitizer/MMU Address Output Port** (U115, U125) outputs are enabled. When FORCE_ADDR goes high, the waveform address now present on ADB0-15 of U115 & U125 is clocked into

Digitizer/MMU Communication Port (U106, U126), and the waveform tag present on ADB16-19 of U96 is clocked into **Diagnostic Wfm Tag Readback Latch** U97.

6. Reset the state of **Digitizer/MMU Handshake Logic** <17>.

   - Disable the outputs of the **Digitizer/MMU Address Output Port** and **Wfm Tag Output Latch** by setting **Diagnostic Loopback Control** U74C-9 WFM_REQ(H):L (via setting bit 0 high in DAG IC Control Register 4).

7. Read and verify the waveform address and waveform tag patterns from the **Digitizer/MMU Communication Port** U106 & U126 <17> and **Diagnostic Wfm Tag Readback Latch** U97 <17>, respectively.

8. Repeat steps 2-7 for the remaining test patterns (see Error Index sections below).

9. Terminate loop back operation and reset digitizer hardware.

   - Initialize the operating state of the communication latches and purge any pending DMA requests by writing a value (which is insignificant) to the **Digitizer/MMU Communication Port** output latches U116 and U136, toggling U85C-9 FORCE_DATA(L), and then reading the communication input latches U106 and U126.

   - Disable the loopback function by setting U146B-19 LOOPBACK_EN(L):H <17>, U83C-11 DIGREQ_EN(H):L, and U166-2 WFM_REQ(H):L <20> (by setting bit 0 high in DAG IC Control Register 4).

   - Reset the digitizer hardware by setting **TB Control Latch 3** U212-12 SYSTEM_RESET(L) <15> low.

Error Index G5321

The display shows the first test pattern read from the **Digitizer/MMU Communication Port** <17> which did not match the expected test pattern. The patterns written to the **Time Interpolator Data Latches** and the expected patterns read back are shown in the following table:

Time Interpolator Pattern	Expected Pattern
0000*hex*	0000*hex*
0001*hex*	0001*hex*
0002*hex*	0002*hex*
0004*hex*	0004*hex*
0008*hex*	0008*hex*
0010*hex*	0010*hex*
0020*hex*	0020*hex*
0040*hex*	0040*hex*
0080*hex*	0080*hex*
0100*hex*	0100*hex*
0200*hex*	0200*hex*
0400*hex*	0400*hex*
0800*hex*	0800*hex*
1000*hex*	1000*hex*

Pts/Addr Gen	Wfm Intrfce	Address/Tag (G532X)
	2000*hex*	2000*hex*
	4000*hex*	0000*hex*
	8000*hex*	0000*hex*
	0000*hex*	0000*hex*
	0000*hex*	0000*hex*
	0000*hex*	0000*hex*
	0000*hex*	0000*hex*
	0000*hex*	0000*hex*
	0000*hex*	0000*hex*

Error Index G5322

The display shows the first test pattern read from the **Diagnostic Wfm Tag Readback Latch <17>** which did not match the expected test pattern. The patterns written to the **Final Wfm Id Encoding Table** and the expected patterns read back are shown in the following table:

Final Wfm Id Encoding Table Pattern	Expected Pattern
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0000*hex*	0000*hex*
0001*hex*	0001*hex*
0002*hex*	0002*hex*
0004*hex*	0004*hex*
0008*hex*	0008*hex*
0010*hex*	0000*hex*
0020*hex*	0000*hex*

Caveats

The addresses used to access the **Final Wfm Id Encoding Table <20>** to transfer data to the **Wfm Tag Output Latch <17>** are irrelevant since the table is completely filled with the same data, i.e., any address will access the same data for any given test pattern. This test verifies the data lines into and out of the **Final Wfm Id Encoding Table**, but not necessarily the addressing of the table. The integrity of some of the table RAM locations are also verified, but which ones and how many is unknown due to the lack of addressing control.

MANUAL CHANGE INFORMATION

At Tektronix, we continually strive to keep up with latest electronic developments by adding circuit and component improvements to our instruments as soon as they are developed and tested.

Sometimes, due to printing and shipping requirements, we can't get these changes immediately into printed manuals. Hence, your manual may contain new change information on following pages.

A single change may affect several sections. Since the change information sheets are carried in the manual until all changes are permanently entered, some duplication may occur. If no such change pages appear following this page, your manual is correct as printed.