5201 TERMINAL
REPAIR MANUAL

Volume I

INFOREX

FE. 0255-1
Printed in U.S.A.

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

TABLES

CHAPTER 1
INTRODUCTION

1.0 GENERAL

This chapter defines the scope and objectives of the manual and provides a list of related documents. This list provides supplemental information to aid in performing the tasks of terminal repair.

1.1 OBJECTIVES OF MANUAL

This manual is intended as a guide for field engineers and other technical personnel directly involved in troubleshooting and maintaining the principal logic boards of the 5201 Terminal. This repair manual consists of two volumes: Volume I contains the theory and related troubleshooting information, and Volume II contains all the figures referenced in Volume I. Figure numbers in Volume I refer to illustrations in Volume II. To facilitate ordering these publications, a single document number is used with a "1" or "2" suffix to indicate the volume, e.g.,

FE.0255-1 for Volume I
FE.0255-2 for Volume II

1.2 SCOPE OF MANUAL

This manual is divided into six chapters. Each chapter contains information on the Monitor, Keyboard, Terminal Control board, T/R Feature Interface board, T/R Control board, Refresh Storage board and Power Supply when applicable to the coverage provided by the chapter. The chapters are as follows:

1. INTRODUCTION
2. PRIMARY BLOCK DIAGRAM DESCRIPTION
3. FUNCTIONAL BLOCK DIAGRAM DESCRIPTION
4. LOGIC DIAGRAM DESCRIPTION
5. FUNCTIONAL LOGIC DESCRIPTION
6. TERMINAL MAINTENANCE

Chapter 1 provides an Introduction to the manual. It includes the scope, objectives and related documents applicable to this manual.

Chapter 2 provides a general description of the 5201 Terminal to include the important data flows within the Terminal. It is supported by a primary block diagram.

Chapter 3 provides a functional block diagram description of each major assembly of the 5201 Terminal. These assemblies are the Monitor, Keyboard, T/R Feature Interface board, Refresh Storage board, Terminal Control board, T/R Control board and Power Supply. Each of the assemblies is supported by a functional block diagram.

Chapter 4 provides a logic description of functional logic areas that are expanded in detail because of the difficulty or complexity of the circuitry. Each logic description is supported by a logic diagram. Other diagrams are included as required.

Chapter 5 provides a functional logic description of the Screen Refresh function. The Screen Refresh function is the only totally hardware-generated function. All other functions are totally or in part software-generated. The Screen Refresh function is supported by a functional logic diagram and other diagrams.

Chapter 6 provides fault isolation analysis and support documentation required to perform Terminal maintenance to the component level.

Appendix A provides a discussion of the diagnostic routines generated for use in Terminal fault isolation. Listings of the routines with applicable comments are provided.

1.3 APPLICABLE DOCUMENTS

The following related documents should be used as required:

68P65130A74-1	Motorola Monitor Technical Repair Manual
68P25253A02	Motorola Monitor Service Manual
S02-210	Keyboard Specification
2010-8003*	Family Tree 5201 Terminal (115 VAC)
2010-8004*	Family Tree 5201 Terminal (220 VAC)
S-001-026	Logic and Schematic Diagram Symbology

*2010-8003 and -8004 provide a list of all applicable assemblies.

CHAPTER 2
PRIMARY BLOCK DIAGRAM DESCRIPTION

2.0 GENERAL

The 5201 Terminal (Figure 2-1) is a fully intelligent Terminal with a Keyboard for entering data into the System 5000 and a Display Module which displays data from either the Keyboard or a data record retrieved from a file in the System 5000. In addition to data retrieved and displayed, the Terminal provides the operator with a means of inputting commands to the System 5000 to perform operator-initiated functions. Besides interfacing with the System 5000, the 5201 Terminal also interfaces with a Serial Printer which allows the operator to generate a hard copy of the screen display. Terminal-to-system data flow is provided by a Key Adapter Unit (KAU) capable of handling up to eight Terminals. The data flow within the Terminal is from the KAU to the Display Monitor, from the Keyboard to the Display Monitor, and from Refresh Storage to the KAU. Data from the KAU is routed through the T/R Control, the Terminal Control and Refresh Storage to the Display. Data from the Keyboard is routed through the Terminal Control and Refresh Storage to the Display. Data from Refresh Storage for the KAU is Keyboard data that has been temporarily stored and subsequently routed through the Terminal Control and T/R Control to the KAU. The 5201 Terminal is capable of either local (hardwired) or remote (modem) transmission. The transmission between the 5201 Terminal and the KAU is serial for both the local and remote configurations. The 5201 Terminal comprises seven basic assemblies:

T/R Control Logic PC board assembly

Keyboard assembly

Terminal Control board assembly

Refresh Storage PC board assembly

14-inch Display Module

115VAC/220VAC Power Supply assembly

T/R Feature Interface board

2.1 T/R CONTROL LOGIC PC BOARD ASSEMBLY

The primary function of the T/R Control Logic is to provide interfacing between the Terminal and the KAU located in the System Control Unit card box. It converts parallel-transmitted data from the Terminal Control board to serial data for the KAU, and serial-received data from the KAU to parallel data for use by the Terminal Control board. Besides transmitting data, the T/R Control Logic also decodes the poll and address messages from the KAU to determine whether or not the terminal is being flagged by the system. The T/R Control board provides the Terminal Control board with received data status and Printer status information. The T/R Control board may be configured to transmit and receive data by means of either a twisted pair (local) or a modem (remote). When a modem is connected to the Terminal, the T/R Control board is automatically configured for remote transmission.

2.2 KEYBOARD ASSEMBLY

The primary function of the Keyboard assembly is to provide the operator with means for inputting character data directly to the Terminal, inputting command codes for use by the System 5000, and inputting control flags for use in controlling data positioning on the screen. Alphanumeric character data and command codes are in EBCDIC code. Control flags use both EBCDIC code and hardwired input. All Keyboard data is sent to the Terminal Control board, where it is decoded and used for Screen Refresh or desired control functions as determined by the key depressed.

2.3 TERMINAL CONTROL BOARD ASSEMBLY

The Terminal Control board (TC) is the heart of the 5201 Terminal. It is a microprocessor under software control, which processes data and controls all activity within the Terminal. It provides the Terminal with the ability to process all commands and messages from the KAU without interrupting the rest of the System 5000, thus making the 5201 Terminal a truly intelligent Terminal. The TC receives data from the KAU through the T/R control board assembly, from the Keyboard and from the Refresh Storage assembly. It sends data and control information to the KAU through the T/R Control assembly, to the Refresh Storage assembly, and to the T/R Feature Interface board. The TC board may be controlled and exercised by the FE Panel when desired.

2.4 REFRESH STORAGE PC BOARD ASSEMBLY

The purpose of the Refresh Storage (RFS) board is to store EBCDIC character information from either the KAU or the Terminal Keyboard and display this character data on the Display Monitor. The RFS board serves as a temporary storage area for Keyboard data destined for later transmission back to the KAU. It contains 2K bytes of semiconductor memory, which is equivalent to a full screen of characters. Besides providing storage, the RFS board also takes the EBCDIC codes stored in the memory and generates character codes which are converted to serial display data for refreshing the Display Monitor. All activity of the RFS board is under control of the microprocessor. The RFS board is also capable of generating hexadecimal code when it is selected by the FE.

2.5 DISPLAY MODULE

The monitor used in the 5201 Terminal (Rev C) is a 14-inch Motorola Monitor with an 80-character by 24-line display. This provides the Terminal with the capability of displaying up to 1920 characters in a 5 x 7 dot matrix, with a single-character cursor which may be positioned either directly by the microprocessor or from the Keyboard by the operator. Only 1840 data characters can be displayed, as the bottom line is reserved for command codes and communication between the Terminal and the KAU. The Display Monitor is selectable for either 115 or 230 volts, 50/60 Hz, and has regulated power supplies.

2.6 POWER SUPPLY ASSEMBLY

The 5201 Terminal uses either 115 VAC or 220 VAC input power. The Power Supply assembly generates various DC voltages which are applied to the various assemblies within the Terminal. Both the AC and DC voltages are regulated by the Terminal, thus providing stable operating voltages for reliable Terminal operation.

2.7 T/R FEATURE INTERFACE BOARD

The primary function of the T/R Feature Interface board is to provide interfacing between the 5201 Terminal and a Serial Printer, thus providing the operator with means for generating a hard copy of the displayed data. It receives Printer data and control information from the microprocessor, which it routes to the Serial Printer. From the Printer it receives status information, which it routes to the T/R Control Logic PC Board assembly, where it is formatted into a status word for use by the microprocessor.

CHAPTER 3
FUNCTIONAL BLOCK DIAGRAM DESCRIPTION

3.0 INTRODUCTION

This chapter contains the Terminal data flow diagram (Figure 3-1) and functional block diagram descriptions of all major assemblies in the 5201 Terminal:

a. Monitor (Motorola)
b. Keyboard (19-inch and 21-inch)
c. T/R Feature Interface (2016–5027)
d. Refresh Storage board (2010–5030)
e. T/R Control board (2010–5031)
f. Terminal Control board (2001–5008-2)
g. Power Supply (2010–5035)

Each assembly is supported by a functional block diagram and other documents as required. The level of coverage is to the functional block. Hardware boundaries of the assemblies have been preserved to ensure consistency of coverage. Where necessary for understanding, main paragraphs (3.x.x) have been further divided into subparagraphs (3.x.x.x). The subparagraph is the lowest level of discussion presented in this chapter.

3.1 TERMINAL MONITOR

The Monitor used in the 5200 System is a Motorola Model XM400-6X, Inforex PN S09-140-001. It displays up to 1920 characters (80 characters by 24 lines) on a 14-inch CRT which operates at a horizontal scan rate of 15,750 ±500 Hz and a vertical scan rate of 50–60 Hz. The data to be displayed is generated from the Keyboard by the operator or from the Terminal Control board by KAU commands. The data is applied to the monitor as a video signal from the Refresh Storage board and Vertical and Horizontal Sync signals applied from the Terminal Control board (Figure 3-2). The video signal is processed by the Video Display circuits to control the beam of the CRT. The position on the CRT where the data is displayed is controlled by the Vertical and Horizontal Sync and Deflection circuits, which deflect the beam (video) to a particular location on the screen. When the beam is on, data is painted in the area where the beam strikes the CRT screen; if the beam is off, the CRT is blank in the given area. The intensity, contrast and brightness of the data displayed are adjusted by controls on the front panel and the Power Supply assembly. Power (115/230 VAC, 60 Hz) to operate the monitor is supplied from the 5201 Terminal Power Supply assembly. The internal B+ required by the monitor is developed by the Regulated Low-Voltage power supply.

3.1.1 Video Display Circuits

The Video Display circuits process the video input signal to turn the CRT beam on or off. The video signal, applied from the Refresh Storage board, is a series of pulses which correspond to the code (1–positive, 0–gnd) for data to be displayed. The video signal is a result of serializing the Character Generator data in the Refresh Storage logic and coupling an analog signal, Monitor Video, to the display input. (The amplitude of the video signal is varied by the Intensity and Contrast controls, which vary the bias of the output stage at the

Refresh Storage board.) The Monitor Video signal is amplified and conditioned by the video circuits, which provide high gain over a wide bandwidth. The output of the video circuits is coupled to the control grid of the CRT. The overall gain of the video circuits is such that it increases the video signal amplitude to a level that turns on the CRT beam when the input goes positive. When a logical zero is being outputted from Refresh Storage, the video amplifier and the CRT beam are essentially turned off; the CRT is blank at the given point on the screen. The CRT beam is also turned off during horizontal and vertical retrace times, which are synchronized with the video data by the Vertical and Horizontal Sync inputs. These signals are applied to the Vertical and Horizontal Sync and Deflection circuits from the Terminal Control board.

3.1.2 Vertical Sync and Deflection Circuits

The Vertical Sync and Deflection Circuits deflect the CRT beam from the top to the bottom of the CRT viewing area. The signal to deflect the beam is developed by the Vertical Oscillator, Driver and Output circuits. Normally, the oscillator is cut off and a sawtooth signal is developed which subsequently drives the yoke to deflect the CRT beam vertically, top to bottom. The Vertical Sync input is applied to, and triggers the oscillator to drive the sawtooth signal toward ground. The negative-going signal is coupled through the Driver to the Output, which is cut off, initiating a vertical retrace. The field in the Vertical Deflection coils collapses, so that the deflection returns to the top of the screen. The Vertical Sync input, which is developed on the Terminal Control board immediately after the last character (79) on the last line (24) has been processed, synchronizes the data being displayed with vertical deflection. Thus, when the last bit of data is processed in the lower right corner of the CRT, the beam is deflected to the top of the CRT so that the next bit of data (character 0, line 0) is displayed in the upper left corner of the CRT. Figure 4-2 illustrates the screen organization by line, row and character. Each character is apportioned a 7 x 9 dot matrix on the screen within the 12 rows that make up a line. The beam is deflected across the screen, left to right, a row at a time by the Horizontal Sync and Deflection circuits.

3.1.3 Horizontal Sync and Deflection Circuits

The Horizontal Sync and Deflection circuits deflect the CRT beam from left to right across the CRT viewing area. The signal to deflect the beam is developed by the Horizontal Oscillator, Driver and Output circuits. The Horizontal Sync signal from the Terminal Control board synchronizes the oscillator with the data to be displayed by shifting the oscillator on/off rate and, subsequently, the scan and retrace from left to right across the CRT. The oscillator is free-running, and the Horizontal Sync signal pulses the oscillator after the data for each row in a line of characters has been shifted from the Refresh Storage board. The Sync signal is compared with a sample of the Horizontal Output signal at the input to the oscillator and, if the two signals are out of phase, the oscillator bias is varied so that the oscillator turns on faster or slower. When the oscillator turns on, the driver raises the signal to a level sufficient to force the output to turnoff, the magnetic field in the yoke collapses, and horizontal retrace starts. When the output is turned off, a signal to blank the CRT is also developed. When the oscillator turns off, the output turns on, and horizontal scan starts. Thus, as each complete row of data is processed and displayed, the horizontal retrace starts, synchronized with a pause in data bits. When the next bit of data is ready, it is displayed at the left edge of the screen.

3.1.4 Regulated Low-Voltage Power Supply

The Regulated Low-Voltage power supply provides the low voltage necessary to operate the Monitor: +30 VDC and +73 VDC. These voltages are developed by Rectifier and Regulator circuits. The AC input voltage can be either 115 volts or 230 volts; a switch at the input provides the correct connection to the input transformer. The AC is rectified and applied to the Regulator, which maintains the B+ output. Fluctuations in the B+ output voltage result in the conduction of the regulator, which counteracts the fluctuation.

3.2 KEYBOARD

The Keyboard provides the operator with the means for entering or retrieving data and commands in the 5000 System. The type of keyboard used is optional; it can be either a 21-inch, 91-key Keyboard (Inforex PN 2010-5020) or a 19-inch, 81-key Keyboard (Inforex PN 502-210-001,-002). In addition, Keyboards are available for languages other than English (Japanese, identified as Kana, is available).

The operation and function of all of the Keyboards are similar, but the location (layout) of the alphanumeric and function keys is different. For all Keyboards, the major output is an 8-bit-plus-parity EBCDIC code which is applied to the Terminal Control board, where it is subsequently routed to the System and Display (CRT). The code or data developed is generated within the Keyboard by the Key Switch modules and Encoder matrix, the Encoder Control logic, the MOS Encoder logic and the Repeat Character logic (Figure 3-3).

3.2.1 Key Switch Modules and Encoder Matrix

Data entry is initiated by the operator on the downstroke of a key (not dependent on key release). When a key is depressed, two isolated outputs are applied to the Encoder matrix, which combines the outputs from all keys on the Keyboard into 13 input lines to the MOS Encoder for encoded keys and into single-line outputs to the Encoder Control logic for function keys. The key outputs are developed by solid-state switch modules, which are the basic components of the Keyboard. Because of the circuitry used in the modules, the outputs are either ON or OFF (never in between). When a key is depressed, pulsed outputs are developed which are subsequently stored in the MOS Encoder, regardless of whether the key is released or not. The key outputs are also applied from the Encoder matrix to the Encoder control logic, which determines the output of the MOS Encoder and the operating mode of the Keyboard. In addition, the Repeat Key output is applied from the Encoder matrix to the Repeat Character logic to initiate the Repeat function of the Keyboard, and Control/Kana/Shift outputs are applied to the Terminal Control directly to indicate Keyboard operational status.

3.2.2 Encoder Control Logic

The Encoder Control logic determines the output of the MOS Encoder, which can be one of four modes. The mode is determined by the inputs from the shift keys, alpha keys, numeric pad keys and control key, along with a System Shift input. The combination of inputs is decoded to form a three-bit Control code which is inputted to the MOS Encoder logic. The System Shift input determines upper- and lower-case modes for data during System operation; when System Shift is ON it forces upper-case mode, regardless of the Shift key position. The Control and Kana function keys are used to override all modes and force the Control or Kana mode for the Keyboards. Therefore, the variations of the three-bit Control code determine the 10-bit output (KS0-7, a parity bit KS8, Repeat Enable bit 10) of the MOS Encoder.

3.2.3 MOS Encoder Logic

The MOS Encoder logic decodes the 13 inputs resulting from depressing a key, loads the correct data code into a 10-bit storage register according to the Control code input, and outputs a strobe (KBSTB) which indicates the data is available to be outputted to the Terminal Control. The circuitry of the MOS Encoder includes two-key rollover: a code is stored by an internal strobe as each key is depressed; another code, resulting from depressing another key, can be stored whether or not the first key is released. The proper codes are outputted sequentially as follows: When the first key is depressed, a strobe signal is generated and the encoded output is stored. If a second key is depressed while the first key is depressed, the output of the second key is stored. Thus, the code from each key is stored as the key is depressed: a strobe signal is generated and the encoded output of the key is stored. The outputs from the MOS Encoder are eight data bits (KS0-7), a parity bit (KS8) and a strobe signal (KSSTB); this data is applied to the Terminal Control for storage in a Random Access Memory (RAM). The strobe signal (KSSTB) is also applied to the Repeat Character logic along with a Repeat Enable (bit 10) signal to develop a Repeat Strobe signal used for the Repeat function. (The output codes for each type of Keyboard in each mode are listed in Tables 6-1 and 6-2.

3.2.4 The Repeat Character Logic

The Repeat Character logic is used to generate a repetitive strobe signal when the Repeat key is enabled. The Repeat key is enabled for all alphanumerics and symbols, and it is disabled for the CTRL key and all command keys except for Cursor Left (←) and Cursor Right (→). When the Repeat key is depressed and an enabling additional key is depressed, the Repeat Character logic is enabled by Repeat Enable (bit 10) from the MOS Encoder. The Repeat Enable is combined with KBSTB (strobe) and the Repeat key input to generate a repeating strobe signal that is outputted to the Terminal Control. The strobe signal is repeated every 100 milliseconds (10 Hz) or every 50 milliseconds (20 Hz), as determined by the timer within the Repeat logic. As each strobe pulse is applied to the Terminal Control, the EBCDIC code for the character being repeated is stored in the Terminal Control RAM.

3.3 T/R FEATURE INTERFACE (Figure 3-4)

The T/R Feature Interface provides the terminal with the capability of outputting character data to a Serial Printer for generation of a hard copy of the video display. In addition, logic on the T/R Interface board provides the Local and Remote Clocks used in the transmitting and receiving data by the T/R Control board. The T/R Interface board has three functional areas: Status and Control Interface logic, Data Register and Interface logic, and Local/Remote Clock Generation logic.

3.3.1 Status and Control Interface Logic

The Status and Control Interface Logic comprises line drivers which provide interfacing for Printer Control data being routed from the Terminal Control board to the Serial Printer. Printer Control data is generated by hardware on the Terminal Control board in response to software commands. Printer Control commands control the printing of character data by the Serial Printer. The Status and Control Interface logic also contains line drivers which provide interfacing for status information generated by the Serial Printer. The Printer Status data is routed to the T/R Control board, where it is formulated as part of a Terminal status word by the receiver multiplexer.

3.3.2 Data Register and Interface Logic

The Data Register and Interface logic receives printer character data from the B Register on the Terminal Control board. Upon command by Terminal software, the character data is loaded into an 11-bit buffer register by LD PRTR. The output of the register is routed through output line drivers to the Line Printer as Printer data. On power turn-on or a Master Reset command from the FE Panel, the Data Register and Interface logic receives a reset flag to initialize the registers.

3.3.3 Local/Remote Clock Generation Logic

The Local/Remote Clock Generation logic comprises counters and decoders which divide the basic OSC (13.487 MHz) input down to the Local/Remote Clocks. The Local/Remote Clock is sent to the T/R Control board, where, depending on whether the Terminal is in Remote or Local operation, the appropriate clock is selected for the T/R Clock (TRANSMIT/RECEIVE CLK).

3.3.3.1 Remote Clock Generation

When the Remote Enable flag from the T/R Control board is active, the Remote Clock Generation logic is active. One of two Remote Clock frequencies is selected by a baud rate switch on the T/R Feature Interface. When the baud rate switch is in the 1800-line position, the OSC Clock is divided down to provide a Remote Clock frequency of 28.8 kHz. When the baud rate switch is in the 1200-line position the OSC Clock is divided down to provide a Remote Clock frequency of 19.2 kHz.

3.3.3.2 Local Clock Generation

When the Local Enable flag from the T/R Control board is active, the Local Clock Generation logic is active. Unlike the Remote Clock, the Local Clock is not frequency-selectable. The OSC Clock is divided down to provide a Local Clock frequency of 153 kHz.

3.4 REFRESH STORAGE BOARD (Figure 3-5)

The Refresh Storage (RFS) board provides keyboard and KAU data storage for the purpose of video screen presentation and message generation for transmission to the KAU. All inputs (both control and data) are from the Terminal Control board. The RFS board provides video control and video data output to the Display Monitor and RFS data outputs to the Terminal Control board. The RFS board performs three basic functions: video screen refresh, keystroke and KAU message screen display, and outputting data to Terminal Control for message transmission. The Refresh Storage board comprises ten functional areas: Refresh Storage Address Selection, Refresh Memory, Input/Output Memory Drivers, Input Parity, Output Buffer, Output Parity, CG Control, Display Data Buffer, Character Generator and Serializer, and Video Display Control.

3.4.1 Refresh Storage Address Selection

The Refresh Storage Address Selection logic comprises multiplexers which select address information for use by Refresh Memory. Under normal conditions, Address Select provides Video Address information (VA0–10) to Refresh Memory to address the memory for the purpose of Screen Refresh. When writing data into Refresh Memory from the Terminal Control board or reading data out of Refresh Memory to the Terminal Control board, Address Select routes B Register data through the Refresh Storage Address Selection logic as

the addressing for Refresh Memory. Because of the dynamic nature of the RAM's used for Refresh Memory, when no active addressing (screen refresh, read/write data) is being performed the RAM's must be addressed. Address Select accomplishes this by selecting Refresh Address, which provides the required 32 addresses to keep the RAM's alive.

3.4.2 Refresh Memory

Refresh Memory comprises eighteen 1 x 1024 bit RAM's. These RAM's are configured to provide a 9 x 2048 bit Refresh Memory. Eight bits of refresh data plus a parity bit are stored. Addressing is provided by the Refresh Storage Address Selection logic, which routes either B Register, Video Address, or Refresh Data to Refresh Memory for memory addressing. The data to be stored in Refresh Memory (RSDI0–7) is received through Input/Output selectable Memory Drivers from the Terminal Control board. In addition to receiving address and data information, Refresh Memory also receives control signals from the Terminal Control board. Reset 1 and Reset 2 perform an initializing function on the RAM's before any operation (Screen Refresh, RAM Refresh, Write Data, Read Data) being performed. The signals CS1 and CS2 (Chip Select 1 and 2) are active each time a read or write operation is performed. The Chip Select commands enable the input/output data lines. CLK 1 and CLK 2 are the clock signals for the RAM's and are required for any read, write or RAM refresh operation. The output of Refresh Memory is sent through selectable INPUT/OUTPUT Memory Drivers to the Output Buffer for temporary storage.

3.4.3 Input Parity

Input Refresh Storage data (Refresh Data In) from the Terminal Control board is sent to a parity generator, where a parity bit is generated for storage through the Input/Output Memory Drivers in the Refresh Memory as part of the data word. This establishes the parity of the input data and is used when outputting the data from Refresh Memory to check for data storage errors.

3.4.4 Input/Output Memory Drivers

The Input/Output Memory Drivers perform two important functions. The first function is that of a multiplexer. When writing data from the Terminal Control board to Refresh Memory, Read/Write Control routes Refresh Data In through the Input/Output Memory Drivers to refresh Memory. When reading data from Refresh Memory to the Terminal Control board or Display Monitor, Read/Write Control routes Refresh Data Out from Refresh Memory through the Input/Output Memory Drivers to the Output Buffer. The second function of Memory drivers is to convert Refresh Data In to voltage levels that are consistent with storage in Refresh Memory and to convert Refresh Data Out back to digital voltage levels.

3.4.5 Output Buffer

The Output Buffer provides Refresh Memory with a temporary storage area for one character (eight bits) of data plus its associated input parity bit. When reading from Refresh Memory for Screen Refresh or as a result of a software command, a Refresh Data Out Clock is generated by Terminal Control. This clock pulse loads one character of Refresh Data Out into the Output Buffer. The output of the buffer is sent to the Terminal Control board, where it is used for data transmission and video control functions. Internally Refresh Data Out is routed through the Output Buffer to various functions for use in parity checking and Screen Refresh.

3.4.6 Output Parity

The Output Parity circuit is physically identical to the Input Parity circuit. The functional difference between the two circuits is that the Input Parity circuit is programmed for odd parity, whereas the Output Parity circuit is programmed by the parity bit associated with each Refresh Data Out character word. If the character parity bit is active high, indicating an even number of high bits in the eight-bit character word, the Output Parity circuit is programmed by the parity bit to generate odd parity on the associated character bits. In the same manner, if the parity bit is inactive low, the Output Parity circuit is programmed to generate even parity on the associated character bits. Configuring Output Parity in this manner causes the odd parity output for all characters to be active high unless the character bits have been altered either by Refresh Memory or the Output Buffer. Thus, if DOPE (Data Output Parity Error) is inactive low, this indicates an incorrect Refresh Data Out character.

3.4.7 Character Generator Control Buffer

The Character Generator (CG) Control Buffer is a six-bit buffer used to store two bits of Refresh Data Out (bits 6 and 7), four bits of Hex Control information, and the output parity bit (DOPE). A new six-bit CG control word is loaded into the buffer each time a CG Address Load command is received from the Terminal Control board. The CG Address Load command is generated each time a new Refresh Data Out word is read from Refresh Memory. The output of the CG Control Buffer (CG Select) is sent to the Character Generator and Serializer function, where it is used to select the proper Character Generator to be used for the character being processed.

3.4.8 Display Data Buffer

The Display Data Buffer is a six-bit buffer which, when commanded by CG Address Load, is loaded with Refresh Data Out (bits 0–5). The CG Address Load command is generated by Terminal Control each time a Refresh Data Out word is read from Refresh Memory. The output of the Display Data Buffer is applied to the character Generator and Serializer function as the Character Generator Address.

3.4.9 Character Generator and Serializer

The Character Generator and Serializer logic comprises two character generators capable of storing sixty-four 7 x 9 bit character matrices each and an eight-bit shift register used to convert parallel data to serial data for video generation. One character generator stores upper-case character codes, which include numerics and special characters; the other stores lower-case and hex character codes. The Serializer receives seven-bit character words from both character generators and serial-shifts them out to the Video Display Control logic.

3.4.10 Video Display Control

The Video Display Control logic comprises discrete logic which performs three basic functions. The first is the conversion of digital character, cursor and error data to analog levels for screen display. The second is to modify the serial character data according to Field Definition and Job Definition bytes stored in the Terminal Control board. Selected characters or data fields may be blinked, blanked or lowered in intensity when the applicable Video Control signal from Terminal Control is active. The third function is to control the contrast and intensity of the video display in response to external contrast and intensity controls. Output of the Video Display Control logic is applied to the Display Monitor as MON VIDEO.

3.5 T/R CONTROL BOARD (Figure 3-6)

The T/R Control board provides the communications and interfacing between the KAU and the 5201 Terminal. Its primary function is to convert and asynchronously transmit or receive serial data between the KAU and the Terminal. To perform this function a 1402A UART (LSI) is used. The T/R Control board also contains logic for enabling and controlling remote and local communications. Another important function is to generate a horizontal (Longitudinal Redundancy Check logic) parity byte on the transmitted and received data for use in checking for message errors. In addition to the communications-related functions, the T/R Control board contains the 8K of ROM storage which houses the Terminal operational program and the time base which generates the basic clock (13.487 MHz) used throughout the Terminal.

3.5.1 Local/Remote Control Logic

The Local/Remote Control logic provides the inhibit and enable signals for routing data (received and transmitted) between the KAU and the Terminal. The mode of operation is determined by whether or not a modem is connected to the Terminal for remote transmission. When a modem is connected, Remote Control from the modem applied to the Local/Remote Control logic automatically configures the Terminal for remote data transmission. This is performed by enabling the remote transmit/receive path and inhibiting the local transmit/receive path by means of Local/Remote Enable commands applied to the Transmit and Receive Data Local/Remote Select logic. The Local/Remote Control logic also provides a Serial Distributor (SD) function which allows the Terminal to be time-shared with others on the same remote transmission line. Transmit Status flags (TRE, TRANSEN) developed internally and on the Terminal Control Board are used to generate transmit gate timing for use by the Asynchronous Transmit/Receive module.

3.5.2 Terminal Time Base

The T/R Control board contains a crystal oscillator which provides the terminal with the basic timing frequency of 13.487 MHz. The 13.487 MHz clock is routed to the T/R Feature Interface board, where it is divided down to provide the time base circuits with the local and remote clocks. Depending on whether the Terminal is in a local or remote configuration, the T/R clock is a function of the local or remote clock. Besides being routed to the T/R feature interface, the basic clock (13.487 MHz) is routed to the Terminal Control board, where it is used to clock the display counters. The T/R clock is used by the UART module as the frequency for transmitting and receiving. The other clock frequency generated by the time base is a TC clock (6.74 MHz), and it is used by the Terminal Control board to generate the Processor cycle timing.

3.5.3 Transmit Data Local/Remote Select Logic

The Transmit Data Local/Remote Select logic provides the data path selection between the KAU and the Terminal. When REMOTE ENABLE is active, transmit data is routed through a current driver to a modem for remote data transmission. In addition, REMOTE EN, in conjunction with the output of the current driver from the modem receive line (inactive during transmission), is used to inhibit transmit data from being routed through the local data path. When LOCAL ENABLE is active, the transmit data is routed through a twisted pair to the KAU.

3.5.4 Longitudinal Redundancy Check Logic

The function of the Longitudinal Redundancy Check (LRC) logic is to generate a horizontal parity byte on both received data from the KAU and transmit data sent to the KAU by the Terminal. Similarly, the KAU also generates a horizontal parity byte. This byte is generated on a message basis, and it is part of the applicable transmit or receive message. In a Received Data message, the last byte of information sent by the KAU is the LRC byte. This byte is compared with the LRC byte accumulated by the LRC logic. If there is a correct comparison, it is assumed that the message sent by the KAU was received properly by the Terminal. If there is an incorrect comparison, the Terminal sends a NAK message, which instructs the KAU to retransmit the message. Similarly, when the Terminal is transmitting a message, it accumulates an LRC byte which it transmits as the last message byte to the KAU. At the KAU this LRC byte is compared with a LRC byte generated on the transmitted data by the KAU. If there is an incorrect comparison, the Terminal is flagged by the KAU to retransmit. The LRC logic operates in two modes, Receive and Transmit.

3.5.4.1 LRC Logic–Receive Mode

In the normal configuration, the Received Data (RR1 –RR8) input to the LRC multiplexer is routed to the LRC accumulator circuits. The LRC logic remains in a wait condition until the Received Data decoder detects an EOT character, indicating a beginning of KAU transmission. At this time the Received Data decoder sends a Receive Data flag to the LRC circuits to initialize and to enable the LRC accumulator. For each character following the start of text character (EOT), the LRC logic accumulates an eight-bit word which is stored in the LRC output buffer. When the Terminal detects an end-of-text character (ETX), the software generates a LRC OFF command, which disables the LRC logic. At this time the Terminal compares the accumulated LRC byte with the LRC byte sent as part of the KAU transmission to determine whether the message must be retransmitted. Once the LRC Data has been sampled by the Terminal, the software generates a CLR LRC command, initializing the LRC Logic, readying it for the next message parity check.

3.5.4.2 LRC Logic–Transmit Mode

When the Terminal is transmitting messages, the Terminal Control board commands the LRC multiplexer to route transmit data (TR1–TR8) to the LRC accumulator circuits. Unlike the Receive mode, where the LRC accumulator is enabled by the Received Data Decoder, when in Transmit mode the LRC accumulator is initialized and enabled by the Terminal Control board in response to software (LRC ON). The LRC accumulates an eight-bit LRC byte on the Transmit data, which it stores in the LRC output buffer. At the end of the data transmission, the Terminal retrieves the accumulated LRC byte and transmits it to the KAU as the last character. After transmitting the LRC byte, the software clears the LRC circuits, readying them for the next LRC check for the purpose of disabling the LRC circuits.

3.5.5 Receive Multiplexer (Table 3-1)

The Receive Multiplexer (MUX) comprises four 4:1 multiplexers and four 2:1 multiplexers which supply a 12-bit Received Data word to the Terminal Control board (RM0–11). Selection of the Receive Mux is a function of commands generated on the Terminal Control board by software. These commands are REC MS1 and REC MS2. The Receive MUX processes four types of data, depending on the value of REC MS1 and 2. Inputs to the MUX are: Received data (RR1–RR8), LRC data from the LRC logic, error and transmit status from Transmit/Receive module, Printer status from the T/R Feature Interface, and the Terminal address (TA1,2,4) from the address switch. When the software instruction is X'E0', the Terminal address and Printer status are formulated into a Received Data output. If the software instruction is an X'E1', the error and transmit status information is formulated into a Received Data input. An X'E2' instruction selects the Received Data (RR1–RR8) input-to-output path, and when an X'E3' instruction occurs the LRC data is routed to Terminal Control as Received Data. The select code and actual Received Data message are listed in Table 3-10.

Table 3-1 Receive Multiplexer (T/R Control Board) Output Selection

REC MS	1 2	RM 0	RM 1	RM 2	RM 3	RM 4	RM 5	RM 6	RM 7	RM 8	RM 9	RM 10	RM 11
	0 0	TA 1	TA 2	TA 4	0	PPR FEED RDY	CARR RDY	CHAR RDY	PPR OUT	CHECK	PRT RDY	0	0
	1 0	THRE	TRE	OE	FE	PE	0	0	0	CHECK	PRT RDY	0	0
	0 1	RR 1	RR 2	RR 3	RR 4	RR 5	RR 6	RR 7	RR 8	CHECK	PRT RDY	0	0
	1 1	LRC 1	LRC 2	LRC 3	LRC 4	LRC 5	LRC 6	LRC 7	LRC 8	RR 1	RR 2	RR 3	RR 4

3.5.6 Receive Data Local/Remote Select Logic

The function of the Receive Data Local/Remote Select logic is to route either the local or remote received data to the Transmit/Receive module (UART). The primary enabling signal is Local/Remote Enable from the Local/Remote Control logic. When Local Enable is active, the local twisted-pair input is routed to the UART and the remote received data input is inhibited. When Remote Enable is active, the remote input is enabled, and the local input is inhibited. The Terminal is normally in a Receive Data configuration. When transmitting, it sends an Inhibit command to the Receive Data Local/Remote Select logic to disable both the local and remote inputs, thus eliminating possible Transmit interference from the Receive Data lines.

3.5.7 Received Data Decoder Logic

The Received Data decoder comprises discrete logic which performs a number of decode functions on the received data. It monitors the received data for General Poll and Direct Poll messages and to take appropriate action. When the KAU sends a poll message, the decoder checks the address byte. If the address byte equals the Terminal address switch value, the Decoder logic samples the condition of the TRANS RDY command from the Terminal Control board. If the command is active (Terminal has data to send), after decoding the General Poll command the logic flags the Terminal Control board, which takes appropriate action. If a Direct Poll is detected, the Processor is interrupted regardless of TRANS RDY. If TRANS RDY is inactive, the Decoder logic initiates an automatic Acknowledge message in response to a General Poll without interrupting the Processor.

Control inputs from the Transmit/Receive Module (TRE, DR), and the Terminal Time Base (T/R CLK) provide the Decoder logic with synchronization for message decoding. In addition to the previously mentioned outputs, the Decoder logic also generates Receive Data flags for use by the LRC logic, and the inhibit command, which is used by the Received Data Local/Remote Select logic. For a more detailed discussion, refer to Chapter 4.

3.5.8 Transmit/Receive (T/R) Module

The Transmit/Receive (T/R) module (UART) is an LSI logic module which converts parallel Transmit data (TR1–TR8) to serial data for transmission to the KAU and converts serial Received data from the KAU to parallel Received data (RR1–RR8) for use by the Terminal. Besides performing data conversion, the T/R module also provides vertical parity (PE), frame parity (FE) and overrun parity (OE) information, and transmit status information (TRE, DR, THRE) for use by the Terminal. Control information is received from the Local/Remote Control logic (TRANS·GATE, SD+REM) and the Terminal Control board (–LDTR, –CLR DR). Timing is provided by the Terminal time base T/R CLK). For a more detailed discussion of the T/R module refer to Inforex Logic and Schematic Diagram Symbology, S001-026.

3.5.9 Microcode Instruction Storage

The Microcode Instruction Storage (MIS) comprises four 8 x 2048 bit ROM's which provide a total storage of 8K eight-bit instructions. The addressing for the ROM is generated on the Terminal Control board and sent to the MIS as ROA 0–12; ROA 0–10 provide the addressing, while ROA 11 and 12 select the ROM to be addressed. When ADDRESS READ is active, the MIS is enabled, and the eight-bit instruction (RODO 0–7) addressed by ROA 0–12 is applied to the Terminal Control board, where it is decoded for use by the Processor. Processor.

3.6 TERMINAL CONTROL BOARD (Figure 3-7)

The Terminal Control board is a special-purpose microprocessor which monitors and controls all Terminal activity and performs processor functions in response to an 8K operational program stored in a ROM memory on the T/R Control board. A 175-instruction set (Table 3-10) allows the programmer to:

Select data paths for moving data to different functions in the Terminal.

Perform compares between known values and the actual values stored in registers.

Perform tests both on general values of A Register and for status conditions of a Status register.

Branch both unconditionally and conditionally.

Control Terminal operation by setting or clearing conditioning flags.

Perform error reporting.

Basically, the processor performs two types of routine, KAU-initiated and Terminal-initiated. KAU-initiated routines are in response to a Read/Write command from the KAU. Upon receiving a KAU Read/Write command, the Terminal program branches to the appropriate routine to handle the KAU command and performs the operations requested by the KAU in the Read/Write message initiated by the KAU and sent to the Terminal. The Terminal initiates four primary routines. The first is a Terminal-initializing routine used in power turn-on. This routine clears the screen and initializes the Terminal, readying it for operation. The second is a keystroke-gathering and -processing routine, which is entered after a key has been depressed on the keyboard. It checks that the key is valid and, if so, performs the required processing, depending on whether the key depressed is a character or function key. The third routine is a Printer routine which is entered when an operator requests it. It is responsible for controlling and supplying data to a Serial Printer for generation of hard copy. The fourth routine is an Error routine. If, during processing either KAU- or Terminal-initiated routines, the Processor detects a prescribed invalid condition in the Terminal, the Terminal branches to an Error routine to further test and display the detected error. The microprocessor contains a software-addressable RAM storage for storing a maximum of 16 different values of character data, count information, or control or status information. Besides containing a microprocessor, the Terminal Control board also contains the logic for generating and controlling Screen Refresh. This is the only hardware-controlled section of the Terminal Control board.

3.6.1 Terminal Display Counters and Decoders

The Terminal Display counters and decoders generate all the timing required for printing character information on the video screen. The screen displays 24 lines of character data plus a partial line used for Terminal mode information (DATA, INSERT, DELETE, TRANSMITTING). Each line consists of 12 rows for normal data display, or 13 rows if the cursor is to be positioned on the line. In either case, row 12 of each line is blanked, thus providing spacing between lines of characters. On each line a maximum of 80 characters can be displayed. Eighty valid character times are generated for each row; in addition, 17 more character times are generated for each row for horizontal retrace. Each character block comprises a 9 x 12 dot matrix (9 dots per row, 12 rows). For vertical retrace from the bottom to the top of the screen, the Display counters generate one extra line at line 25 time.

To generate the timing for the described screen presentation, the Terminal Display Counter and Decoder logic consists of four unique counters: Dot counter, Character counter, Row counter and Line counter. All four counters use the OSC clock (13.482 MHz) from the T/R Control board, thus making the counters synchronous with each other. The Dot counter generates nine dot counts for painting the required dots in a character block. The Character counter, which is enabled one time for each nine dot counts, generates 97 character counts. Of these character counts, 80 are required to display the 80 valid characters; the other 17 provide the horizontal retrace time from the end of one row to the beginning of the next row. The Row counter is enabled one time for each 97 character counts, and it generates either 12 row counts if Cursor Enable is inactive, or 13 row counts if Cursor Enable is active. The Line counter is enabled one time for every 12 or 13 row counts, and it generates, by means of a decoder network, 25-7/12 line counts, which is the required number of lines to paint one complete screen presentation. The Terminal Display counters are synchronous counters that continuously provide Display timing to Terminal functions involved in generating the video presentation on the screen whenever power is applied to the Terminal. For a more detailed discussion of the Display counters refer to Chapter 4.

3.6.2 Refresh Storage Control Logic

The Refresh Storage Control logic performs four functions:

Generate the timing for Screen Refresh.

Generate the video addressing for Screen Refresh.

Generate the timing to Read/Write data from/to Refresh Storage.

Generate the timing for RAM Refresh when Screen Refresh is inhibited.

3.6.2.1 Screen Refresh Timing Generation

The Refresh Storage Control logic decodes the Display timing from the Terminal Display counters and decoders to provide the Refresh Storage board with the Clock, Read/Write Control and Strobe commands needed to refresh the video screen. Under normal conditions (VAR INH inactive low), Refresh timing refreshes the screen presentation approximately once every 20 milliseconds. For each character displayed, the Refresh Storage Control logic generates RESET CLK, –RS CLK, –READ EN, –RSDO STB and –CGAL (Refresh timing) 11 times (once for each row). Signals –RESET CLK and –RS CLK provide the timing to clock character data out of Refresh Memory. Signal –READ EN enables the Refresh-Memory-to-Output-Buffer path of the memory drivers, RSDO STB is the Load command for the Output Buffer, and –CGAL is the Load command for the Display Buffer. For a more detailed discussion, refer to Chapter 4.

3.6.2.2 Video Address Generation

In synchronization with the Refresh Timing generation, the Refresh Storage Control logic generates Video addressing to address Refresh Memory. To address all locations in Refresh Memory, 1920 sequential addresses are generated. To provide the needed count, three 4-bit binary counters are connected in sequence with this configuration; a maximum count of 2048 can be obtained. Each counter is parallel-loaded by a preset value counter which is incremented by 80 at the beginning of each new line display. Because a line of characters is painted by row, Refresh Storage Control generates the same 80 addresses eleven times (11 rows). (Addressing is inhibited during rows 12 and 13.) The preset value counter provides Refresh Storage Control with the starting address, thus enabling it to remember which line of characters is being addressed. Refresh Storage Control also sends Video Address to the Cursor Buffer and Compare logic for determining the cursor line position. For a more detailed discussion of the video addressing circuits, refer to Chapter 4.

3.6.2.3 Read/Write Data

When instructed by software, the Refresh Storage Control logic generates the timing and control commands to read or write data from or to Refresh Memory. The Read/Write operations are normally performed during horizontal retrace (Screen Refresh dead time), so there is no interference with Screen Refresh. Under special conditions, such as reading a whole screen of characters to Terminal control for transmission, or writing a character record from the KAU into Refresh Memory, Screen Refresh is inhibited by VAR INH, and Read/Write operations have priority. When performing a Read/Write operation, Refresh Storage Control inhibits Video Address output to Refresh Memory and enables the address location to be selected by the B Register.

3.6.2.4 Random Access Memory Refresh

When VAR INH is active, inhibiting Screen Refresh, and no Read/Write operation is being performed, Refresh Storage Control generates Random Access Memory (RAM) Refresh timing,which is required to prevent the RAM's used in Refresh Memory from "running down." If the RAM is not accessed within the allotted time, its storage ability deteriorates and character data is lost.

3.6.3 Cursor Buffer and Compare Logic

The primary function of the Cursor Buffer and Compare logic is to store the cursor address value received from the B Register and to compare this value with the Video Address information received from Refresh Storage Control. When the two values match, a Cursor Video signal (CV) is generated, which is sent through the Error Display logic to the video screen for presentation. It is important to note that the Cursor Video is delayed by two character times. This delay is necessary because character display painting is slow, and video addressing is actually two character times ahead of the actual character position on the screen. Unlike characters, the cursor is painted immediately after being generated. To correct for the difference in timing, the Cursor Video signal is delayed by two character counts so that the cursor is painted in the correct character position. The Cursor Buffer and Compare logic also generates a Cursor Enable flag, which is sent through Video Control to the Terminal display counters to enable the Row counter to count one additional row (row 13). To ensure that the actual Cursor Video output does not occur during character painting, the Video Control logic sends a Cursor Video Enable, which is activated at the

beginning of row 13 and which enables Cursor Video to be outputted only during row 13 time. In addition to the primary function of generating and controlling Cursor Video, the Cursor Buffer and Compare logic generates blinking information which it sends to Video Control.

3.6.4 Video Control Logic

The Video Control logic's primary function is to provide the Terminal with the video control signals for controlling the character data on the video screen. The generation of these control signals is determined by the Job Definition byte stored in the Job Definition register and the Field Definition byte associated with each data field (fixed/variable) which is stored as part of the character data stream in Refresh Memory. The Video Control logic continuously monitors the character data stream that is refreshing the video screen. When a data byte with bits 6 and 7 inactive low is detected, the control logic decodes this as the control code associated with a new data field. The control logic also decodes bits 3 and 4 of the control code to determine if the new data field is a fixed-data (FORMAT) or a variable-data (operator-changeable data) field. With the type of field determined, the Video Control logic examines the Job Definition byte and the other control bits of the control code to determine what, if any, video control flags are to be generated to modify the presentation of the associated data field. The Video Control logic generates control signals to blank fields, to blank control code display (normally blanked), to blink fields, to blank the Operator Command Area (OCA), and to lower intensity on variable- and fixed-data fields. Any combination of these control signals may be applied to the data stream being displayed. To modify the intensity of a data field or to blank a data field, the Terminal must be in Data mode (JDR0 active high). Another important control signal generated by the Video Control logic is –SYSTEM SHIFT. This signal is a function of the Job Definition byte, indicating Data mode (JDR0 active high), and the Field Definition byte stored in the Field Definition register, indicating lower alphanumeric characters allowed (FDR5 active high).

NOTE

The Field Definition byte (FDB) stored in the Field Definition register is the last FDB seen by the cursor. The FDB's stored in Refresh Memory are part of the data stream and are constantly accessed from Memory during Screen Refresh. These two should not be confused.

When –SYSTEM SHIFT is active low, the operator may enter both upper- and lower-case alpha characters from the Terminal keyboard. When –SYSTEM SHIFT is inactive high, only upper-case alpha characters may be entered at the keyboard. Another function of the Video Control logic is to generate cursor control flags. Upon detecting the beginning of row 13, the Video Control logic generates a Cursor Video Enable, which enables the generation of Cursor Video (CV) by the Cursor Buffer and Compare logic. The Video Control logic also routes the Cursor Enable flag from the Cursor Buffer and Compare logic to the Terminal Display counters to enable the row counter to count one additional time. All functions of the Video Control logic are synchronized with Screen Refresh by means of the Display timing received from the Terminal Display counters.

3.6.5 Character Generator Timing and Control Logic

The Character Generator Timing and Control logic performs two functions. One is to generate the Character Generator Control timing for converting EBCDIC character data from Refresh Memory to character data for display on the screen, and for serializing this character data for video generation. The second function is to generate the timing and control for hex character generation. Under control of the Display timing from the Terminal Display counters, the Character Generator Timing and Control logic generates a sequence of timing in synchronization with Screen Refresh timing from Refresh Storage Control. The timing sequence is such that EBCDIC character data is converted, one character at a time, to dot patterns which are loaded into a serializer and serially shifted out to the Refresh Storage D/A converter. To perform this function four timing signals are required. Signals UC CLK (upper-case clock) and LC CLK (lower-case clock) are generated once per row. These signals increment address storing counters internal to the character generators (two generators, lower-case and upper-case), which store character patterns as a function of rows. At row 11 time (last dot display per character), several additional clocks are generated to clear the internal counters. Signal SER LD (serial load) and SER CLK (serial clock) are generated to load the outputs of the character generator into an eight-bit shift register.

Signal SER LD parallel-loads one character at a time from the Character Generator into the shift registers. Immediately following SER LD, eight SER CLK's are generated to shift the character out of the shift register to the Video Generation logic. The hex function of the Character Generator control is initiated by the FE Panel. When the HEX switch on the FE Panel is active, the Character Generator Timing and Control logic generates the timing and control signals to configure the lower-case Character Generator to generate hexadecimal (hex) values for video display. A two-character stacked hex display is generated for each EBCDIC character read from Refresh Storage, and the Control logic alternately sets the Character Generator to a Superscript and then to a Subscript mode and blanks one row time between modes. This allows the screen to display characters two high by 80 on a line. A Hex mode flag is sent to the Character Generators to inhibit the upper-case Generator and at the same time to address the lower-case Generator to the hex code storage area of the Character Generator memory.

3.6.6 Random Access Memory Multiplexer Select Buffer

The RAM Multiplexer (MUX) Select buffer is a two-bit latch which stores the RAM Select value as defined by software. Whenever an X'Dx' (IRM13) instruction is executed, the buffer is loaded, at time T12, with the value of x ($IRL2^2$–2^3). This value may be 0 (both $IRL2^2$ and $IRL2^3$ inactive low), 4 ($IRL2^2$ active high, $IRL2^3$ inactive low), 8 ($IRL2^2$ inactive low, $IRL2^3$ active high) or C (both $IRL2^2$ and $IRL2^3$ active high). Depending on the value of x, RAM Select is in one of four conditions.

3.6.7 Random Access Memory Multiplexer

The RAM MUX comprises twelve 4:1 multiplexers which are indirectly, as a function of RAM Select, under software control. Depending on the value of RAM Select, one of four data paths is routed to the RAM's. If RAM Select is a function of storing an X'D0', X'D1', X'D2' or X'D3' instruction in the RAM MUX Select buffer, the MUX 1 data input is routed through the RAM MUX to the RAM's. (The value of the LSB of the instruction indicates what constitutes MUX 1 data: 0 = A REG, 1 = B REG, 2 = JDR, and 3 = FDR.) If RAM Select is a function of an X'D4' instruction stored in the buffer, keystroke data from the keyboard is routed through the RAM MUX to the RAM's. An X'D8' instruction stored

in the RAM MUX Select buffer routes Receive data from the T/R Control board through the RAM MUX to the RAM's, and an X'DC' instruction stored in the buffer routes Refresh Data Out through the RAM MUX to the RAM's. RAM Select is stable until a new X'Dx' instruction is stored in the RAM MUX Select buffer, so the RAM MUX path selected is stable until a new X'Dx' instruction is executed. On power turn-on, the MUX 1 data path is selected.

3.6.8 Random Access Memories

Random Access Memory storage comprises three 4 x 16 bit RAM's providing storage for sixteen 12-bit data words. The data inputs from the RAM MUX are routed so bits 8-11 are stored in the high RAM, bits 4-7 are stored in the middle RAM and bits 0-3 are stored in the low RAM. Besides the data inputs, each RAM receives a four-bit address from the RAM Address buffer and a RAM Enable from the RAM Enable logic. All three RAM's or any one RAM may be enabled, depending on the RAM Enable decoded. The sequence of events for writing into the RAM's is:

a. At T12 time RAM data is selected by the RAM MUX.

b. At T13 time the RAM Address is selected.

c. At T14 time a RAM Enable is generated to write the RAM data into the high RAM, middle RAM, low RAM or all three.

To read data from tthe RAM's, only a RAM Address need be selected.

3.6.9 Random Access Memory Enable Logic

The RAM Enable logic decodes Write instructions at T14 Processor time and generates RAM Enable, which writes all or selected RAM Data bits into the RAM's. Depending on software, one of five different instructions generates a RAM Enable. When either an X'C8' or an X'08' instruction is decoded, a RAM Enable is applied to all three RAM's, allowing RAM Data 0-11 to be stored. When an X'04' instruction is decoded, the high RAM only is enabled, allowing only RAM Data 8-11 to be stored. When an X'02' instruction is decoded, only the middle RAM is enabled, allowing only RAM Data 4-7 to be stored. When an X'01' instruction is decoded, only the low RAM is enabled, allowing only RAM Data 0-3 to be stored. All RAM Enable commands are active at time T14 of the Processor cycle. The RAM Enable logic receives selected Instruction Register information to decode the RAM Enable instructions.

3.6.10 Random Access Memory Address Buffer

The RAM Address buffer comprises four D flip-flops which receive low-byte Instruction Register data (IRL 2^0–2^3). When an X'3x' instruction is executed, the buffer is loaded at time T13 of the memory cycle with the low-byte information. The outputs of the RAM Address buffer are applied as the RAM Address (up to 16 addresses) to the RAM's.

3.6.11 A/C Register Multiplexer (Table 3-2)

The A/C Register MUX consists of 12 parallel 4:1 multiplexers. Its function is to route one of four types of data to the A or C Register as ACM 0-11. The primary input is the output of the RAM's (RADO 0-11). Other inputs are from the Instruction register (IRL 2^0–2^3, IRM 2^0–2^3, IRH 2^0–2^3) and Instruction Register decoder (±IRL 0 – IRL 11). A/C Register MUX data input selection (A/C MUX Select) is a function of the Microcode and Branch Control. Normally, the A/C Register MUX is routing RAM data to the A and C Registers. When an X'10' through X'1B' instruction is executed, the X'1x' indicates that it is a "clear select bit of A Register instruction." The least significant bit of the instruction indicates the bit to be cleared. An X'1x' instruction logically ANDs the RADO 0-11 input with the –IRL 0-11 input to clear the selected bit of the A Register. When an X'2x' instruction is executed, it indicates that the selected bit, as defined by the least significant bit of the instruction, is set. An X'2x' instruction logically "ands" RADO 0-11 A input with the IRL 0-11 input to set a selected bit in the register. Besides the instruction select of the A/C Register MUX, a command from Branch Control (AR/CRM) is used to select the Instruction Register input. The Branch Control command loads a set of program constants into either the A Register or the C Register, according to the instruction X'9E' (A Register constants) or an X'9F' (C Register constants) being executed. The A/C Register MUX is dynamic and is always selecting RAM data input unless instructed to select other input data.

Table 3-2 A/C Register Multiplexer Output Selection

A/C MUX SEL	ACM 0	ACM 1	ACM 2	ACM 3	ACM 4	ACM 5	ACM 6	ACM 7	ACM 8	ACM 9	ACM 10	ACM 11
00	RADO 0	RADO 1	RADO 2	RADO 3	RADO 4	RADO 5	RADO 6	RADO 7	RADO 8	RADO 9	RADO 10	RADO 11
01	IRL 2^0	IRL 2^1	IRL 2^2	IRL 2^3	IRM 2^0	IRM 2^1	IRM 2^2	IRM 2^3	IRH 2^0	IRH 2^1	IRH 2^2	IRH 2^3
10	RADO 0· $\overline{\text{IRL0}}$	RADO 1· $\overline{\text{IRL1}}$	RADO 2· $\overline{\text{IRL2}}$	RADO 3· $\overline{\text{IRL3}}$	RADO 4· $\overline{\text{IRL4}}$	RADO 5· $\overline{\text{IRL5}}$	RADO 6· $\overline{\text{IRL6}}$	RADO 7· $\overline{\text{IRL7}}$	RADO 8· $\overline{\text{IRL8}}$	RADO 9· $\overline{\text{IRL9}}$	RADO 10· $\overline{\text{IRL10}}$	RADO 11· $\overline{\text{IRL11}}$
11	RADO 0· IRL0	RADO 1· IRL1	RADO 2· IRL2	RADO 3· IRL3	RADO 4· IRL4	RADO 5· IRL5	RADO 6· IRL6	RADO 7· IRL7	RADO 8· IRL8	RADO 9· IRL9	RADO 10· $\overline{\text{IRL10}}$	RADO 11· IRL11

3.6.12 Job Definition Register

The Job Definition Register (JDR) is a 12-bit parallel-loaded register which stores RAM data (RADO 0-11) when commanded by software. The function of the JDR is to store a Job Definition Byte (see Table 3-3), which the Terminal receives as part of Read/Write messages from the KAU. The Job Definition Byte defines the job that the Terminal is to perform on the message data. As part of the Receive message sequence that the Terminal program performs on the received data, when the JDB is received, the program generates an X'7C' instruction. The X'7C' instruction is decoded as a JDR load, and at time T12 of the Processor cycle, the JDB, which was previously stored in the RAM's, is loaded into the Job Definition register. Job Definition Register information is applied to various other functions of Terminal Control. Signal JDR 0 (LSB of JDB) is applied to the Test Bit logic to test whether the Terminal is in Operator or Format mode. Several Job Definition bits are applied to the Video Control logic for generating Video Control signals used in Refresh Storage. The entire Job Definition Byte is applied to MUX 1, where, when instructed by software, it is incorporated as part of a data message transmitted back to the KAU.

Table 3-3 Job Definition and Field Definition Code Designations

Job Definition Register			Field Definition Byte in Data Stream		
Bit 7	0 =	Hold Bit Clear	Bit 7,6	00 =	Distinguishes the EBCDIC control code from A/N or function key
	1 =	Hold Bit On			
Bit 6	0 =	Lockout Clear			
	1 =	Lockout Set	Bit 5	0 =	Upper Alpha and Numeric
				1 =	Lower Alpha and Numeric
Bit 5	0 =	Fixed Data Field Low Intensity			
	1 =	Fixed Data Field High Intensity	Bit 4,3	00 =	Variable Data
Bit 4	0 =	Variable Data Field Low Intensity		01 =	Spare
	1 =	Variable Data Field High Intensity		10 =	Fixed Data
				11 =	Auto Mode
Bit 3		Not Used	Bit 2	1 =	Blink this Field
Bit 2	0 =	Error Light Off	Bit 1	1 =	Blank this Field
	1 =	Error Light On	Bit 0	1 =	Lockout (software only)
Bit 1	1 =	Blank OCA			
	0 =	Display OCA			
Bit 0	0 =	Format Mode			
	1 =	Operator Mode			

3.6.13 Field Definition Register

The Field Definition Register (FDR) is a 12-bit parallel-loaded register which stores RAM data (RADO 0-11) when commanded by software. The function of the FDR is to store the Field Definition Byte (FDB). The FDB is applied by the KAU as part of the data stream and defines the type of data (variable, fixed, fixed auto) and how the data for a specific field is to be handled by the Terminal (blank field, allow lower-case characters). The number of FDB's is dependent on the number of fields to be displayed on the video screen. Unlike the Job Definition register, the Field Definition register is loaded with a new FDB each time the operator moves the cursor to a new data field on the screen. Data fields exist only when the Job Definition byte defines the mode of operation as Data mode. When the Terminal detects the presence of a new FDB during cursor movement, the program initiates an X'7D' instruction. The X'7D' is decoded as an FDR load, and at time T12 of the processor cycle the FDB, which has been moved from Refresh Memory to RAM storage, is loaded into the Field Definition register. The output of the FDR is applied to MUX 1 for distribution and to the Video Control logic for determining System Shift configuration.

3.6.14 B Register

The B Register comprises a three-stage up/down binary counter with parallel load capability. It is a versatile, general-purpose register which can be loaded in parallel with RAM data (RADO 0-11), incremented, decremented or cleared by software instruction. The Processor uses the B Register for holding the cursor position, the Read/Write character address, printer data when outputting to a Serial Printer, and position data for characters on the video screen. RAM data (RADO 0-11) is parallel-loaded into the B Register at time T13 of the Processor cycle when Terminal Control is executing an X'52' instruction. It is incremented by COUNT-UP CLK at time T14 of the Processor cycle when an X'54' instruction is decoded. Decrementing the B Register is performed by COUNT-DOWN CLK, which is a function of executing an X'58' instruction during time T15 of the Processor cycle. Clearing the B Register is performed at time T12 of the Processor cycle in response to an X'51' instruction being decoded. B Register data (BR 0-11) is applied to various functions of Terminal Control. It is applied to the Cursor Buffer and Compare logic when BR 0-11 is the Cursor Position address, to Refresh Storage when BR 0-11 is the Refresh Memory address, to T/R Feature Interface when BR 0-11 is Printer data, and to the Test Latch logic when BR 0-11 is Character Position count. The Test Latch logic tests BR 0-11 to determine if the character position is the first (BR = 0) or the last (BR = 1840). The B Register output is also applied to the FE Indicator MUX, which, when selected at the FE Panel, routes the B Register output to the panel for display.

3.6.15 A Register

The A Register comprises a three-stage up/down binary counter with a parallel load (3 = 74193). It receives parallel data from the A/C MUX (ACM 0-11). Control of the A Register is provided by the Processor program and Branch Control. The A Register is a versatile, general-purpose register which is capable of holding data from the A/C Register MUX or a binary counter which may be incremented or decremented by the program. The A Register is used by the program (1) to hold character data returned from Refresh Storage for testing, (2) to hold character data (from keyboard and KAU) to be written into Refresh Storage and (3) to hold Error information for testing and for writing into the Error registers. A/C Register MUX data (ACM 0-11) is loaded into the A Register by the Load RAM command generated by the instruction set or Branch Control. RAM data from the A/C Register MUX (ACM 0-11) is parallel-loaded into the A Register at time T13

of the Processor cycle when an X'42' instruction is decoded. When either an X'1x' or X'2x' instruction occurs, at time T15 of the Processor cycle, a selected bit in the A Register is cleared (X'1x') or set (X'2x'), as defined by the least significant bit of the instruction in response to the conditioning of A/C Register MUX and the generation of Load RAM command. A third method of generating a Load RAM command is in response to an X'C8' (RFS to A Reg) instruction. Prior to an X'C8' instruction, an X'DC' instruction selects the Refresh Storage Input at time T12 of the Processor cycle, and an X'34'/X'35'/X'36' instruction selects a location in the RAM's at time T13 of the same cycle. At time T14 an X'C8' instruction writes the Refresh Storage data into the RAM's. The RAM output is routed through the A/C Register MUX to the A Register, where, at time T15 of the Processor cycle, X'C8' generates a load command, and the RAM data is loaded into the A Register. The last method of loading the A Register from the A/C Register MUX is by an X'9E' Branch instruction which configures the A/C Register MUX to select Instruction Register data for presentation to the A Register. At time T14 of the Processor cycle, a BLAR command from Branch Control loads the constants from the Instruction register through the A/C Register mux into the A Register. The A Register also operates as an up/down binary counter. The A Register is incremented in response to an X'44' instruction occurring at time T14 of the Processor cycle. It is decremented at time T15 of the Processor cycle, when an X'48' instruction is decoded. The A Register is cleared by either a RSTI from Timing Counter and Control, a BCAR command at time T13 of the Processor cycle from Branch Control, when an X'9E' instruction is being executed, or at time T13, when an X'1X', X'2X' or X'C8' instruction is being executed. The output of the A Register (AR is applied to:

FE Indicator MUX for display at the FE Panel when selected.

The RFS MUX to be stored in RFS memory as RSDI 0-11 when selected.

The Error registers for error display.

The MUX 1 for distribution.

The Test Bit logic for testing.

The Test logic for FDC, VDF detection.

3.6.16 C Register

The C Register comprises a three-stage up/down decade counter and a JK flip-flop with a parallel load function. The C Register is a general-purpose register which may be loaded in parallel with A/C Register MUX data (ACM 0-11), incremented or decremented. When the C Register is used as a counter, its output is a decimal count (BCD). Unlike the A and B Registers, the C Register generates a 13-bit parallel output. The 13th bit is provided by the JK flip-flop. The parallel data input received from the A/C Register MUX is either RAM data or a set of constants from the I Register. Parallel load of the C Register is controlled by the program when loading RAM data and by the Branch Control function when loading constants. When an X'62' instruction occurs at time T13 of the Processor cycle, the C Register loads RAM data from the RAM's through the A/C Register MUX. When a BLCR (Branch Load C Register) command from Branch Control is received, constants from the I Register through the A/C Register MUX are loaded into C Register at time T14 of the Processor cycle. The C Register is incremented by COUNT-UP CLK, which is activated at time T14 of the Processor cycle when an X'64' instruction is executed. The C Register is decremented by COUNT-DOWN CLK, which is activated at time T15 of the Processor cycle when an X'68' instruction is executed. The C Register may be cleared in

one of three ways: (1) at time T12 of the Processor cycle when an X'61' instruction is executed, (2) at time T13 of the Processor cycle by BCCR (Branch Clear C Register), which is a function of an X'9F' (Load Constants into C REgister) instruction being executed, and (3) by an active RSTI form Timing Counter and Decoder. The output of the C Register (CR 0-12) is applied to the FE Indicator MUX for display on the FE Panel when desired, to the Refresh Storage MUX for storage as RSDI 0-7 in Refresh Storage, and to the Arithmetic Compare logic for comparison with MUX 1 data.

3.6.17 FE Indicator Multiplexer (Table 3-4)

The FE Indicator MUX, which comprises twelve 4:1 multiplexers, provides the FE with a method of displaying a selected set of data inputs on indicators mounted on the FE Panel. The data inputs are A Register data (AR 0-11) (note that bit 12 is not displayed), or Instruction Register data (IRL 2^0–2^3, IRM 2^0–2^3). Selection of the data to be displayed is a function of B0 and B1, located on the FE Panel. When neither B0 nor B1 is active, Instruction Register data is displayed. When only B1 is active, C Register data is displayed. When both B0 and B1 are active, B Register data is displayed.

3.6.18 Multiplexer 1

Multiplexer 1 (MUX 1), which comprises twelve 4:1 multiplexers, is controlled by software. The primary function of MUX 1 is to provide a data distribution path by which the contents of the A, B, FDR or JDR Register may be routed to various functions of Terminal Control. The inputs to MUX 1 are A Register data (AR 0-11), B Register data (BR 0-11), FDR data (FDR 0-11) and JDR data (JDR 0-11). When data path instruction X'D0' is executed, A Register data is routed through MUX 1 for distribution. When data path instruction X'D2' is executed, JDR data is routed through MUX 1 for distribution. When data path instruction X'D3' is executed, FDR data is routed through MUX 1 for distribution.

Multiplexer 1 contains a MUX Select buffer, which stores the least significant bit of the instruction, thus holding the multiplexer selected until a new X'Dx' instruction is generated by the program to update the buffer at time T12 of the Processor cycle. The output of MUX 1 (M1 0-11) is distributed to various functions. The most-used distribution path is to the RAM MUX. When the Terminal is transmitting data to the KAU, the active path is Refresh Storage, to A Register, to MUX 1, to Transmit MUX. When the program is testing the value stored in a register, the path from MUX 1 to Arithmetic Compare logic is used, so all 12 bits or selected groups of bits can be tested against a C Register value.

3.6.19 Transmit Multiplexer (Table 3-5)

The Transmit MUX comprises eight 8:1 multiplexers which are controlled by software and the Received Data Decoder on the T/R Control board (Automatic ACK generation). It performs two functions. One function is to route either MUX 1 data or hardwired Automatic Acknowledge data to the T/R Control board for transmission to the KAU. The other function is to route Error Display information from Error Registers 1 and 2, when in Error Display mode, to the Error Display Control and Printer logic for conversion to Error Video (EV), which is displayed on the video screen. The Transmit MUX has eight 8-bit inputs, of which four are associated with transmitting data and four are associated with error display. To select one of the eight, Transmit MUX Select is composed of three select bits (TM4, TM2, TM1). Bit TM4 (MSB) is controlled by the error display function and is inactive low when not in error display. Combinations of TM2 and TM1 select the four different Transmit Data inputs to the Transmit MUX. Software instructions route either

Table 3-4 FE Indicator Multiplexer Output Selection

B 0	B 1	REG 0	REG 1	REG 2	REG 3	REG 4	REG 5	REG 6	REG 7	REG 8	REG 9	REG 10	REG 11
0	0	IRL 2^0	IRL 2^1	IRL 2^2	IRL 2^3	IRM 2^0	IRM 2^1	IRM 2^2	IRM 2^3	IRH 2^0	IRH 2^1	IRH 2^2	IRH 2^3
0	1	CR0	CR1	CR2	CR3	CR4	CR5	CR6	CR7	CR8	CR9	CR10	CR11
1	0	AR0	AR1	AR2	AR3	AR4	AR5	AR6	AR7	AR8	AR9	AR10	AR11
1	1	BR0	BR1	BR2	BR3	BR4	BR5	BR6	BR7	BR8	BR9	BR10	BR11

Table 3-5 Refresh Storage Multiplexer Output Selection

TM 4 2 1	TR1	TR2	TR3	TR4	TR5	TR6	TR7	TR8	FUNCTION
0 0 0	1	1	1	1	1	1	1	1	'FF'
0 0 1	0	1	1	1	0	0	0	0	'0E'
0 1 0	M1 0	M1 1	M1 2	M1 3	M1 4	M1 5	M1 6	M17	XMIT
0 1 1	M1 8	M1 9	M1 10	M1 11	0	0	0	0	DATA
1 0 0	0	AR 10	AR 9	AR 8	AR 7	AR 6	0	0	ERROR 1
1 0 1	AR 0	AR 1	AR 2	AR 3	AR 4	AR 5	1	0	ERROR 1
1 1 0	AR 6	AR 7	AR 8	AR 9	AR 10	AR 11	0	1	ERROR 2
1 1 1	AR 0	AR 1	AR 2	AR 3	AR 4	AR 5	1	1	ERROR 2

M1 0-7 (X'E8' decoded to generate TM2 and TM1 value) or M1 8-11 (X'EC' decoded to generate TM2 and TM1 value) from Multiplexer 1 through the Transmit MUX to the Transmit Control board for transmission to the KAU. Two sets of Transmit MUX inputs are hardwired. One is a hardwired X'FF' (pad character) code, and the other is a hardwired X'0E' (ACK character). These inputs are sequentially selected, for routing data to the T/R Control board, either by software instructions (X'E0' = transmit pad character, X'E4' = transmit ACK), or by Transmit flags received from the T/R Control board. If the Terminal is in Error Display mode, TM4 is active high, and one of four error display words, depending on the values of TM2 and TM1, is routed through the Transmit MUX to the Error Display Control logic. An inactive TM2 generates a Transmit MUX Select, which routes either the low-order bits or the high-order bits from Error Register 1, depending on TM1 value, to the Error Register Display Control logic. An inactive TM2 generates a Transmit MUX Select, which routes either the low-order or high-order bits from Error Register 2, depending on TM1 value, to the Error Register Control logic.

3.6.20 Refresh Storage Multiplexer (Table 3-6)

The Refresh Storage Multiplexer (RFS MUX) comprises eight multiplexers which are selected by software instructions. The function of the Refresh Storage MUX is to route either A Register data (AR 0-7) or portions of C Register data (CR 0-3, CR 4-7, CR 8011, CR 12) to the Refresh Storage board for storage. MUX Select is a function of decoding one of four instructions:

X'F0'	Selects A Register data
X'F1'	Selects CR 0-3 for low bits and all 1's for high bits
X'F2'	Selects CR 4-7 for low bits and all 1's for high bits
X'F3'	Selects CR 8-11 for low bits and all 1's for high bits
X'F4'	Selects CR 12 for least-significant bit and 0, 0, 0, 1, 1, 1, 1, 1 for bits, 1-7, respectively

In addition to selecting data to be stored, the Refresh Storage MUX applies a Test Bit Logic Select command to the Test Bit logic when an X'F8' instruction is decoded. This Test Bit logic command is used to select one bank of functions to be tested.

Table 3-6 Refresh Storage Multiplexer Output Selection

INST	RSDI 0	RSDI 1	RSDI 2	RSDI 3	RSDI 4	RSDI 5	RSDI 6	RSDI 7
F0	AR0	AR1	AR2	AR3	AR4	AR5	AR6	AR7
F1	CR0	CR1	CR2	CR3	1	1	1	1
F2	CR4	CR5	CR6	CR7	1	1	1	1
F3	CR8	CR9	CR10	CR11	1	1	1	1
F4	CR12	0	0	0	0	0	0	0
F8	TEST BIT MUX SELECT (TBM SELECT) AR → TBM							

3.6.21 Error Registers

The Error registers consist of two 12-bit registers (Error Register 1, Error Register 2). The purpose of the Error registers is, upon software command, to store A Register data (AR 0-11) for later use by the Error Display Control and Printer logic to display error information on the video screen. When an X'CE' instruction is detected, the Error Display Control logic generates a Load Error Register 1. When an X'CF' instruction is detected, the Error Display Control Logic generates a Load Error Register 2 command. This command loads A Register data (AR 0-11) into Error Register 2. When the Terminal is in Error Display mode, the outputs of Error Registers 1 and 2 are routed through the Transmit MUX to the Error Display logic to generate the Error Video for display on the screen. An X'CD' instruction generates an ERR REG CLR command to clear both Error Register 1 and Error Register 2. Refer to Table 3-7 for listing of errors displayed.

Table 3-7 Error Display
(on Screen Shown as Block if Function Bad)

TR1	TR2	TR3	TR4	TR5	TR6	TR7	TR8
	KEYSTROKE PARITY ERROR ER1 AR10	RFS PARITY ERROR ER1 AR9	DTP ER1 AR8	REBS ER1 AR7	WRITE ER1 AR6	0	0
RETRY ERROR ER1 AR0	TRANSMITTER TIME OUT ER1 AR1	RECEIVER TIME OUT ER1 AR2	DATA LENGTH ERROR ER1 AR3	TERMINAL STORAGE ERROR ER1 AR4	LRC ERROR ER1 AR5	1	0
INSTRUCTION ERROR ER2 AR6	WRITE VARIABLE OVERRUN ERROR ER2 AR7	DATA POSITION ERROR ER2 AR8	NO RESPONSE ER2 AR9	POINTER FAIL ER2 AR 10	PSD ER2 AR 11	0	1
TRANSMITTER NOT READY ER2 AR0	TRANSMISSION NOT COMPL. ER2 AR1	OVERRUN ERROR ER2 AR2	FRAMING ERROR ER2 AR3	PARITY ERROR ER2 AR4		1	1

3.6.22 Error Display Control and Printer Logic (Tables 3-7 and 3-8)

The Error Display Control and Printer Logic consists of a variety of logic circuits which perform four basic functions: (1) Display cursor status and, when the Display Error switch on the T/R Control board is active, error information; (2) Generate Printer Control flags in response to software commands; (3) Generate T/R Control flags in response to software commands; (4) Generate Transmit MUX Select commands for selecting the Transmit MUX output.

Table 3-8 Error Display Control and Printer Logic Output

OUTPUT	INST	FUNCTION
RESTORE	B5	DEL LPA
CHAR STROBE	B6	CLEAR LPA
CARR STROBE	BD	TRANSMIT MUX LPAL
PPR FEED STROBE	BE	TRANSMIT MUX LPAU
RIBBON LIFT	2E,2F	WHEN '2E' IS SEEN RIBBON LIFT IS ACTIVATED, '2F' + RST DEACTIVATES IT
REC MS1=0 MS2=0	E0	TRANSFER TERMINAL ADDRESS OR PAD CHARACTER TO RAM AS RECEIVED DATA
REC MS1=1 MS2=0	E1	TRANSFERS STATUS DATA TO RAM AS RECEIVED DATA
REC MS1=0 MS2=1	E2	TRANSFERS RECEIVED DATA (BIT 0-7) TO RAM
REC MS1=1 MS2=1	E3	TRANSFERS RECEIVED DATA (BIT 8-11) OR LRC 0-7 TO RAM
LRC ON	ID	AT T12 WHEN 'ID' SEEN LONGITUDINAL REDUNDANCY CKTS ACTIVATED
ERR REG CLR	CD	WHEN ACTIVE CLEARS ERROR REGS 1 AND 2
LRC OFF	IE	AT T12 TIME WHEN 'IE' OCCURS LRC CKTS DEACTIVATED
ER1 LD	CE	WHEN ACTIVE ERROR REG. 1 WILL BE LOADED FROM A REG.
ER2 LD	CF	WHEN ACTIVE ERROR REG. 2 WILL BE LOADED FROM A REG.
CLR LRC	IF	WHEN ACTIVE LRC BUFFERS CLEARED
TM 1,2,4	E4 E8 EC	THESE ARE THE TRANSMIT MUX SELECTION BITS. TM4 IS ACTIVE ONLY WHEN ERROR DATA IS BEING PROCESSED. WHEN IN XMIT–MANUAL MODE EITHER 'E4'–ACK CHAR, 'E8'–MUX 1=0-7, OR 'EC'–MUX 1=8 11 IS ACTIVE. WHEN IN AUTO XMIT A SEQUENCE OF 'FF'–PAD CHAR FOLLOWED BY '0E' IS GENERATED WITHOUT PROCESSOR INTERRUPTION.
–(CV + EV)		THE SIGNAL IS ACTUATED FOR 3 FUNCTIONS. ONE IS TO PAINT CURSOR ON SCREEN. ANOTHER IS TO PAINT FORMAT, DELETE, INSERT OR XMIT BAR ON SCREEN WHEN APPROPRIATE KEY IS DEPRESSED. THE LAST IS TO PAINT ERROR MESSAGES FROM THE ERROR REGISTER.

3.6.22.1 Cursor, Status, Error Video Generation

The term Error Display Control is misleading because the error information comprises three types of data. The first type is the real error information. When the Error Display switch on the T/R Control board is selected, the Error Display logic generates Transmit MUX Select commands, which route error information from Error Registers 1 and 2 through the Transmit MUX to the Error Display logic (TR1-TR6). In turn, Error Display logic, under control of Display timing from the Terminal Display counters, generates a 4 x 6 error display pattern which is applied to Refresh Storage for display as Error Video (EV). Twenty-four different error bits are displayed (see Table 3-7 for error information). The second type of data is Terminal Status information. The Error Display Control circuits monitor the Delete, Insert, Transmit and Data flags from the Status register (DELETE, INSERT, TRANSMIT) and Job Definition Register (DATA). If any one or a combination of the Status flags is active, Error Display Control generates Error Video (EV) to place a bar on the first seven rows of the screen display under the appropriate function title. For Data the Error Video Bar is from characters 8 to 16, for Insert from characters 24 to 32, for Delete from characters 40 to 48, and for Transmit from characters 56 to 64. The third type of data for error display is the Cursor Video (CV). Cursor Video from the Cursor Buffer and Compare logic is gated together with Error Video (EV), and is routed by Error Display Control to the Refresh Storage board for display.

3.6.22.2 Printer Control Flag Generation

The printer portion of the Error Display Control and Printer logic consists of latch circuits which monitor the program for Printer commands. When an instruction with an X'B' (11 decimal) for the most significant bit is detected, the Printer logic examines the least significant bit to determine if the instruction is a Printer Command flag. If it is, a latch is set, and the Error Display Control and Printer logic applies a Printer Control flag to the T/R Feature Interface board.

3.6.22.3 T/R Control Flag Generation

The Error Display Control and Printer logic, under software control, decodes and generates T/R control flags for controlling two functional areas of the T/R Control board. The first area is the Longitudinal Redundancy Check (LRC) function. When commanded by software, the Error Display Control and Printer logic generates commands to turn on (LRC ON), turn off (LRC OFF), and clear (CLR, LRC) the LRC circuits. These commands are active only during the transmissions of data by the Terminal. The other functional area is the Receive MUX. Instructions are decoded into a two-bit Receive MUX Select command, which is applied to the T/R Control board to select the Receive MUX output.

3.6.22.4 Transmit Multiplexer Select Generation

The Error Display Control logic generates a three-bit Transmit MUX Select command as determined by software, the Received Data decoder on the T/R Control board, and the Error Display switch on the T/R Control board. When the T/R Control board detects a General Poll command from the KAU and the Terminal has no data to send, it instructs the Error Display Control and Printer Logic to generate a sequence of Transmit MUX Select commands. Two Transmit MUX Select commands are generated. The first command is to select the hardwired Transmit MUX PAD (X'FF') character. The second command is the hardwired Transmit MUX ACK (X'0E') character. In response to software,

Transmit MUX Select commands are generated to write MUX 1 data through the Transmit MUX. When the Error Display Switch is active, the Error Display Control and Printer Logic generates Transmit MUX Select commands to route Error Register information through the Transmit MUX to itself for use in generating Error Video.

3.6.23 Arithmetic Compare Logic and Multiplexer

The Arithmetic Compare logic and MUX consists of three Arithmetic Compare units and an 8:1 multiplexer, all of which are controlled by software. The function of the Arithmetic Compare logic is to compare all or selected portions of the MUX 1 data (M1 0-11) with all or selected portions of the C Register data (CR 0-11). When a valid compare is made, a flag is generated which is applied to the Test Latch logic. Under software control, the logic makes comparisons for equal-to, greater-than and less-than conditions. Refer to the functional block diagram for the instructions and associated comparisons.

3.6.24 Status Register

The Status Register comprises various latches, flip-flops and logic circuits which monitor the status of the Terminal. Basically, three types of monitoring are performed: instruction monitoring, keyboard activity monitoring, and monitoring of the General Poll and Direct Poll flags from the T/R Control board. The instruction set is monitored for the occurrence of status flags which cause latches in the Status register to either set or reset, depending on the flag. The flags are: CLR PUL (X'AA'), CLR LOCKOUT (X'AB'), SET PUL (X'AE'), SET LOCKOUT (X'AF'), SET INSERT (X'AC'), SET DELETE (X'AD'), CLR INSERT (X'A8'), CLR DELETE (X'A9'), SET ERROR LAT (X'7A'), CLR ERROR LAT (X'7B'), SET RST INH (X'7E'), CLR RST INH (X'7F'), SET SEND (X'75'), SET OIL (X'76'), CLR SEND (X'78') and CLR OIL (X'79'). Whenever any one of the monitored flags is detected, at time T12 of the processor cycle the appropriate latch either sets or resets. In addition to the stored status control flags, there are flags that are decoded but not stored: LD PRTR (X'2B'), RST PRTR (X'2E') and AUTO (X'77). The Status register monitors the General Poll (DTP) line from the T/R Control board for the occurrence of a set DTP flag. When a set DTP command is received from the Received Data decoder on the T/R Control board, the Status register sets a latch which generates TRANS and DTP outputs. If either a Reset DTP (X'73') instruction or a RST2 command is detected, the DTP latch is cleared. Like General Poll, the Direct Poll (Set DTR) is also monitored, but for a Set DTR, only a DTR Interrupt is generated. When either a Reset DTR (X'72') or a RST2 command is detected, the DTR latch is cleared. The Status register also monitors the keyboards for keystroke generation. When a KB STB (keyboard strobe) is detected, a shift register is loaded and Keystroke Received (KSR) is set. Upon detection of a CLR KSR instruction (X'70'), the Shift register is clocked. This resets KSR and sets Keystroke Process (KSP). When a CLR KSP (X'71') is detected, KSP is cleared. In a similar manner, the same circuit also monitors KS6, KS7 and CC Store to process a control code. Whenever K6 and K7 are both low, the keystroke is a control code. The outputs of the Status Register are applied to the Test Bit logic for testing and to Branch Control, where they are used to initiate the appropriate Terminal processing subroutine to handle the status condition.

3.6.25 Test Bit Multiplexer and Test Logic

The Test Bit MUX and Test logic comprises a two-stage multiplexer circuit which outputs selected data to a test circuit which tests the data for an active condition. All functions of the Test Bit MUX and Test logic are controlled by software. The first-stage multiplexer selects either 16 Status Data (X'F0') flags from the Status register or data from the

A Register (X'F8' = AR 0-11). The data or flags are applied to a second multiplexer, where one of the 16 signals is selected for testing (X'80'–X'8F'). The signal selected is tested for an active condition. If the selected signal is active, the Test Bit MUX and Test logic applies an active low Test flag to the Test Latch Logic, where a latch is set. Refer to Table 3-9 for information concerning the flags tested and the instructions for multiplexer selection.

Table 3-9 Test Bit Logic Test Conditions

MUX SEL	'80'	'81'	'82'	'83'	'84'	'85'	'86'	'87'	'88'	'89'	'8A'	'8B'	'8C'	'8D'	'8E'	'8F'
F0	KSR	DTR	KSP	DTP	VIEW MODE	AUTO	SEND	OIL	RST INN	TRANS RDY	JDR 0	PRIOR RDY	INSERT LAT	DELETE LAT	PUL	LOCKOUT
F8	AR0	AR1	AR2	AR3	AR4	AR5	AR6	AR7	AR8	AR9	AR10	AR11	LP RDY	N/A	N/A	N/A

3.6.26 Test Latch Logic

The Test Latch logic comprises latches and gate logic circuits. One function of the Test Latch logic is to monitor the program and generate flags when commanded by the instruction being executed. The flags generated are distributed throughout the Terminal for setting hardware to the appropriate configuration to process the routine being executed. A second function is to monitor the A Register to determine whether the data field being processed is Variable or Fixed. Specifically, AR 6 and 7 are monitored to detect a Field Definition byte (FDB). If AR 6 and 7 are both inactive low, the byte being processed is a FDB. By monitoring bits 3 and 4 (AR 3 and 4) of the byte, the type of data field is decoded. When AR 3 and 4 are both inactive low in the FDB, the field is a Variable Data field. The testing of the byte is software-controlled. When a "Test AR for VDF" instruction (X'C0') is decoded AR 6 and 7 are tested to ensure the byte is a FDB, and AR 3 and 4 are tested for an inactive low condition. If the condition exists at time T14 of the Processor cycle, the VDF latch is set. When a "Test AR for FDC" instruction (X'E1') is decoded, AR 6 and 7 are tested for inactive low to determine if a Field Definition Character (FDC) is present. If the FDC exists at time T14 of the Processor cycle, the FDC latch is set. Another primary function is to test B Register data (BR 0-11) for select values of character position count. This is done under software control. When a "Test BR=0" instruction (X'C2') is detected, the Test Bit logic checks BR 0-11 for an all-zeros condition. If the condition is met, at time T14 of the Processor cycle the "BR=0" latch is set. When a "Test BR=1840" instruction (X'C3') is detected, BR 0-11 are checked for a binary 1840 value. If the condition is met, at time T14 of the Processor cycle the "BR=1840" latch is set. The zero and 1840 positions are the low and high limits of the character data presentation on the screen. Refer to Table 3-10 for more detailed information on the Test Latch logic.

Table 3-10 Test Latch Outputs

OUTPUT	INST.	TIME	WHEN AND HOW ACTIVE	DESTINATION
VDF	C0	T14	CHECKS AR6+7 = FDB='00', AR 3+4=00 = VARIABLE DATA FIELD	BRANCH CONTROL
FDC	C1	T14	CHECKS AR6+7 = 00=FDB	BRANCH CONTROL
BR=0	C2	T14	CHECKS BR0–11 = ALL ZEROS	BRANCH CONTROL
BR=1840	C3	T14	CHECKS BR0–10 = 0000110 111 = 1840 DEC/730 HEX	BRANCH CONTROL
START AUDIO	CC		CHECKS FOR CC	
–TC STOP	C7,8,9	T12	IF RFS-FETCH(C8+C9) OR RFS-STORE (C7) AT T12–TC STOP ACTIVE	TIMING COUNTER & CG CONTROL
COMP LAT			SETS AT –COMP, RESETS AT –CLR LAT	BRANCH CONTROL
TEST BIT LAT			SETS AT –TEST, RESETS AT –CLR LAT	BRANCH CONTROL
WRITE	C7	T12	SETS AT RFS STORE (C7), RESETS AT –CLR TC REQ.	RFS CONTROL
Z			SETS AT –CT0, RESETS AT –CT40	ERROR DISPLAY CONT & PRINTER LOGIC
CCR LD	C6		CHECKS FOR 'C6'	CURSOR BUFFER & COMP
TRANS RDY	B0,B1	T12	SETS AT T12–'B0', RESETS AT T12-'B1'	TEST BIT LOGIC
VAR IN H	BA,BB	T12	SETS AT T12-'BA' RESETS AT T12-'BB'	RFS CONTROL
–TRANS EN	B2,B3	T12	SETS AT T12-'B2', RESETS AT T12-'B3'	T/R CONTROL BOARD
ERE	B7,B15	T12	SETS AT T12-'B7', RESETS AT T12-'B15'	ERROR DISPLAY CONT & PRINTER LOGIC
–PRT EN	BX	T12	HIGH AT T12 WHEN 'BX' SEEN	ERROR DISPLAY CONT & PRINTER LOGIC
CLR T/R SR	B8		HIGH WHEN 'B8' SEEN	T/R CONTROL BOARD
–LD TR	B4		HIGH WHEN 'B4' SEEN	T/R CONTROL BOARD
CLR DR	B9		HIGH WHEN 'B9' SEEN	T/R CONTROL BOARD

3.6.27 Branch Control Logic

The Branch Control Logic comprises a variety of logic circuits which perform three major functions. The primary function of Branch Control is to monitor the Terminal for certain conditions which, if met, cause the ROM Address register to branch to a new address in the Terminal program. The second function of Branch Control is to load a set of constants into the A Register, and the third function is to load a set of constants into the C Register. Branch Control is under software control. To better understand the functions of Branch Control, it is first necessary to examine the structure of the microcode program. Wherever a branch condition exists in the program, three microcode instructions are associated with it. The first instruction is the branch instruction X'9x', where "9" indicates a branch test and the value of "x" indicates what is to be tested. Instructions X'90' through X'9D' are the branch instructions. An X'9E' or X'9F' instruction indicates constants are to be loaded into either the A or C Register, respectively. The second instruction is the lower eight bits of the address to which the program will branch if the branch test is successful. In the case of an X'9E' or X'9F' instruction, it is the lower eight bits of the constant to be loaded into the A or C Register. The third instruction is the upper four bits of the address for a branch and the upper five bits of the constants if an X'9E' or X'9F' instruction is being executed.

Because a Branch routine has three instructions associated with it, it takes not one, but three Processor cycles to perform the Branch function. The heart of the Branch Control logic is a shift register, which stops the Instruction Register decoder and allows the two required additional Processor cycles to occur. When an X'9x' instruction is detected by Branch Control, the "9" loads the Shift register, and the "x" value is loaded into latches for later use. With the loading of the Shift register, IRD INH is generated, which inhibits the Instruction Register decoder. At the end of the current Processor cycle, the Shift register is incremented by TC C0 (first time slot of Processor cycle). During the second Processor cycle, the ROM Address register is normally incremented, and the new microcode instruction addressed by the register (lower eight bits of address or constants depending on "x") is loaded into the low and middle Instruction registers. At the end of the second Processor cycle, the Shift register is again incremented. The low and middle Instruction registers are inhibited from loading the next program instruction by –BC3 INH being activated. At time T11 of the third Processor cycle, the high byte of the Instruction register is loaded by LD IRH. If the test condition is valid or if X'9E' or X'9F' is selected, at time T12 of the third Processor cycle, the ROM Address register is inhibited from incrementing by activating INH INC. If the branch instruction is an X'9E' or X'9F', INH INC is activated at the beginning of the third Processor cycle. At time T13 of the third Processor cycle, the ROM Address register is cleared by CLR ROA, followed at time T14 of the Processor cycle by a LD ROA command. At T12 of the third Processor cycle, the ROM Address MUX routes the Instruction register to ROM Address register (–ROAM active). When a Load ROA occurs, the value in the low, medium and high Instruction register, which under a branch test is the "Branch To" address, is loaded into the ROM Address register. In the case of an X'9E' or X'9F' instruction, the ROM Address register is not affected. For these instructions at time T13 of the third Processor cycle, either a Clear C Register (BCCR) or a Clear A Register (BCAR) is generated. The input to the ROM Address registers is Instruction Register data because, with the detection of an X'9E' or X'9F' instruction, at the beginning of the third memory cycle, the A/C Register MUX selects the I Register inputs; AR/CRM is activated. At time T14 of the third Processor cycle either the A or C Register is loaded with the generation of a BLAR or BLCR command from Branch Control. These commands load a set of constants as specified by the program.

3.6.28 Instruction Register

The Instruction register consists of two four-bit latches and one five-bit latch. The function of the Instruction register is to store microcode instructions received from Microcode Instruction Storage, located on the T/R Control board, and to distribute the instructions to the Instruction decoder and various Terminal control functions for decoding and execution. All instructions except for branch addresses and A/C Register constants are one byte (eight bits), thus requiring only the low and middle Instruction Register latches (two four-bit latches) for storage. The five-bit latch (high Instruction register) is used only when an X'9x' instruction is executed. Under normal operating conditions, a new instruction is loaded into the Instruction register each T11 time of the Processor cycle and is executed by the Processor during the current Processor cycle. Refer to Table 3-11 for a listing of the instructions executed by the Terminal.

Table 3-11 Instruction Set Summary

00	HALT	7D	LOAD FDR	B7	SET ERE
01	RAM STORE (0-3)	7E	SET RST INH	B8	CLR T/R SR
02	RAM STORE (4-7)	7F	RST RST INH	B9	CLR DR
04	RAM STORE (8-11)			BA	SET VAR INH
08	RAM STORE (0-11)	80	TEST KSR/AR0	BB	CLR VAR INH
		81	TEST DTR/AR1	BC	TRANS MUX–TRM IN
10-1C	CLR 'A' REG. BITS 0-12	82	TEST KSP/AR2	BD	TRANS MUX–LPAL
1D	LRC ON	83	TEST DTP/AR3	BE	TRANS MUX–LPAU
1E	LRC OFF	84	TEST VIEW MODE/AR4	BF	CLR ERE
1F	CLR LRC	85	TEST AUTO/AR5		
		86	TEST SEND/AR6	C0	TEST A REG = VDF
20-2B	SET 'A' REG. BIT 0-11	87	TEST OIL/AR7	C1	TEST A REG = FDF
2C	RST PRINTER	88	TEST RST INH/AR8	C2	TEST BR = 0
2D	LOAD PRINTER	89	TEST TRANS RDY/AR9	C3	TEST BR = 1840
2E	SET RIBBON LIFT	8A	TEST JDR 0/AR10	C6	LOAD CURSOR COMP.
2F	RESET RIBBON LIFT	8B	TEST RIBBON LIFT/AR11	C7	RFS – STORE
		8C	TEST INSERT/–	C8	RFS–FETCH TO 'A' REG.
30	RAM ADDRESS – KEYSTROKE (LRC) DATA	8D	TEST DELETE/–	C9	RFS–FETCH
31	RAM ADDRESS – SPP	8E	TEST PUL/–	CC	START AUDIO
32	RAM ADDRESS – DPP	8F	TEST LOCKOUT/ROM TEST	CD	CLR ERROR REG.
33	RAM ADDRESS – STATUS BITS			CE	LOAD ERROR REG. 1
34	RAM ADDRESS – GENERAL REG. 1	90	UNCONDITIONAL BRANCH	CF	LOAD ERROR REG. 2
35	RAM ADDRESS – GENERAL REG. 2	91	COMPARE		
36	RAM ADDRESS – GENERAL REG. 3	92	TEST BIT LAT	D0	A REG→MUX 1→RAM MUX
37	RAM ADDRESS – NEW SPP	93	SEND	D1	B REG→MUX 1→RAM MUX
38	RAM ADDRESS – FETCH POINTER	94	OIL	D2	JDR →MUX 1→RAM MUX
39	RAM ADDRESS – COMMAND	95	VDF	D3	FDR→MUX 1→RAM MUX
3A	RAM ADDRESS – DATA LENGTH	96	BR = 0	D4	KS→RAM MUX
3B	RAM ADDRESS – STARTING ADDRESS	97	BR = 1840	D8	REC DATA→RAM MUX
3C	RAM ADDRESS – CURSOR POSITION	98	FDC	DC	RFS→RAM MUX
3D	RAM ADDRESS – COMMAND JDR	99	–COMP		
3E	RAM ADDRESS – SEQ. COUNTER 1	9A	–TB LAT	E0	TRANS PAD/TERM ADD = REC DATA
3F	RAM ADDRESS – SEQ. COUNTER 2	9B	–VDF	E1	STATUS = REC DATA
		9C	–BR = 1840	E2	RR1–RR8 = REC DATA
41	CLR 'A' REG.	9D	–FDC	E3	LRC1-8, RR1-4 = REC DATA
42	LOAD 'A' REG. FROM RAM	9E	LOAD 'A' REG. FROM 'I' REG.	E4	ACK = TRANS DATA
44	INCREMENT 'A' REG.	9F	LOAD 'C' REG. FROM I REG.	E8	M1 0-7 = TRANS DATA
48	DECREMENT 'A' REG.			EC	M1 8-11 = TRANS DATA
		A0	CMP C = M1 (0-3)		
51	CLEAR 'B' REG.	A1	CMP C = M1 (4-7)	F0	'A' REG→RFS/STATUS→TBL
52	LOAD 'B' REG. FROM RAM	A2	CMP C = M1 (8-11)	F1	'C' REG 0-3→RFS
54	INCREMENT 'B' REG.	A4	CMP C < M1 (ALL)	F2	'C' REG 4-7→RFS
58	DECREMENT 'B' REG.	A5	CMP C > M1 (ALL)	F3	'C' REG 8-11→RFS
		A6	CMP C = M1 (ALL)	F4	'C' REG 12→RFS
61	CLEAR 'C' REG.	A8	CLR INSERT LAT	F8	'A' REG→TBL
62	LOAD 'C' REG. FROM RAM	A9	CLR DELETE LAT		
64	INCREMENT 'C' REG.	AA	CLR PUL		
68	DECREMENT 'C' REG.	AB	CLR LOCKOUT		
70	CLEAR KSR	AC	SET INSERT		
71	CLEAR KSP	AD	SET DELETE		
72	CLEAR DTR	AE	SET PUL		
73	CLEAR DTP	AF	SET LOCKOUT		
74	CLEAR LP READY				
75	SET SEND	B0	SET TRANS RDY		
76	SET OIL	B1	CLR TRANS RDY		
77	AUTO CLK	B2	SET TRANS EN.		
78	CLEAR SEND	B3	CLR TRANS EN.		
79	CLEAR OIL	B4	LOAD TR		
7A	SET ERROR	B5	DEL LPA		
7B	RESET ERROR	B6	CLR LPA		
7C	LOAD JDR				

3.6.29 Instruction Register Decoders

The Terminal Control board contains two four-bit decoders which decode instruction information from the low and middle Instruction registers. The instructions are decoded to 32 unique outputs (16 for the low Instruction register –IRL 0-15, 16 for the middle Instruction register –IRM 0-15). The Instruction Register decoders use inverters so that there are active-low and active-high decoded outputs. For each instruction, the Instruction Register decoders generate one middle Instruction register value (IRM 0-15) and one low Instruction register value (IRL 0-15).

3.6.30 ROM Address Multiplexer

The ROM Address MUX, under control of Branch Control (–ROAM), selects either a 12-bit address (AS 0-11) from the FE Panel or a 12-bit address (IRL 2^0–2^3, IRM 2^0–2^3, IRH 2^0–2^3) which is routed to the ROM Address register. In the static condition, address data from the FE Panel is routed to the ROM Address register. When executing a Branch instruction, Branch Control generates an active –ROAM, which forces the multiplexer to select Instruction Register data for routing to the ROM Address register. This data sets the register to the branch address defined by the program.

3.6.31 ROM Address Register

The ROM Address register consists of four four-bit up/down binary counters with parallel load function, connected in series. Normally the register is incremented each time T14 of the Processor cycle occurs, generating sequential addresses for accessing Microcode Instruction storage on the T/R Control board. The Parallel Load function is activated by the Load switch at the FE Panel. When the switch is depressed, the address selected at the FE Panel is routed through the ROM Address MUX to the parallel inputs of the ROM Address register. This is a manual operation which provides the field engineer with the capability of addressing desired instructions for execution or observation. The parallel Load function is automatically activated as part of executing a Branch instruction. Under this condition, the address loaded into the ROM Address register is the address associated with the Branch routine of the program. (The count-down function of the ROM Address register is not used.)

3.6.32 Timing Counter and Decoder

The Timing Counter and Decoder provides the Terminal Processor with the necessary timing for execution of the microcode instructions (program supplied by microcode storage to the Processor). The timing is divided into 15 time frames (TC C0-T15) by the decoder (note that not all times are actually used). These time frames are applied to various functions of the Terminal to control the execution of instructions by the Processor. Whenever the Timing counter is stopped, the Processor is effectively halted. The basic clock used to generate the Processor cycle times is TC CLK supplied by the T/R Control board. In addition, control inputs from the FE Panel (SC, SW, Store SW, PS STOP, STOP MR, SW) allow the FE to control the Processor operation (single cycle, repeat instruction, load address, halt). When a halt instruction (X'00') from the Instruction register is decoded, it halts the counter, thus halting the Processor operation. Two reset functions, MR (Master Reset) from the FE Panel and POR (Power On Reset) from the POR module in the power supply, are ORed together and are routed to various functions of the Terminal as RST1, RST2 and RST to initialize the Terminal. Microcode Storage Enable (–ADD Read) is generated and sent to T/R Control to enable addressing the Terminal program.

3.7 5201 POWER SUPPLY (Figure 3-8)

The 5201 Power Supply provides the Terminal with the means of using either 115 VAC or 220 VAC input power from which four DC voltages are generated for use by the Terminal. In addition to providing Terminal power, the Power Supply also provides Terminal control signals for use by the keyboard, by the Display Monitor and the audio alarm, and by the Power On Reset logic (POR). AC/DC power distribution comprises eight basic functions: AC Control, AC Selection and Rectification logic, +5/+12/−12/+22.5 VDC power supplies, Terminal controls, and POR logic.

3.7.1 AC Control

The AC Control logic consists of a line filter, which filters AC fluctuations and then couples the AC to an AC interlock which opens whenever the cover is removed, cutting off the power. The AC is routed from the AC interlock through a circuit breaker to the power ON/OFF switch, which is the operator power ON/OFF control and also provides control of the Display Monitor brightness. The AC Control output is applied to the AC Selection and Rectification logic.

3.7.2 AC Selection and Rectification Logic

The AC Selection portion is a multitap primary transformer which provides the means for selecting any AC voltage between 115 VAC and 250 VAC as the input power. The transformer secondary taps provide the various AC levels to be rectified and used by the four DC power supplies. Full-wave rectification is used to provide the +12 VDC, −12 VDC and +22.5 VDC power supplies with a DC voltage, and half-wave rectification provides pulsating DC to the +5 VDC power supply.

3.7.3 +5 VDC Regulated Power Supply

The +5 VDC power supply is a Powertec regulated supply with a maximum current output of 12 amperes. Voltage and current potentiometers are provided so that the output can be varied from 4.5 to 6.5 VDC with a current up to 12 amperes.

3.7.4 +12 VDC Regulated Power Supply

The +12 VDC power supply is a Powertec regulated supply with a maximum current output of 6 amperes. Voltage and current potentiometers are provided so that the output can be varied from 11.5 to 15.5 VDC with a current up to 6 amperes.

3.7.5 −12 VDC Regulated Power Supply

The −12 VDC power supply is the same as the +12 VDC power supply except that the positive output is grounded and voltage is taken off the negative output.

3.7.6 +22.5 VDC Regulated Power Supply

The +22.5 VDC power supply is a Powertec regulated supply with a maximum current output of 4.5 amperes. Voltage and current potentiometers are provided so that the output can be varied from 17.5 to 24.5 VDC with a current up to 4.5 amperes.

3.7.7 Terminal Controls

In addition to power circuitry, the Power Supply also has various Terminal controls. A keyboard lock switch provides a method by which the keyboard can be inhibited from functioning. Intensity and contrast controls (screwdriver adjustments) vary the CRT display and a potentiometer adjusts the alarm volume.

3.7.8 Power On Reset Logic

The Power On Reset (POR) logic provides four basic functions. The primary function is to generate a POR command at power turn-on which initializes the Terminal logic. The POR logic also provides driver logic to drive the Audio Alarm and two indicators, ERROR and POWER ON. When an error indication is received from the Terminal Control board, the ERROR indicator is illuminated, and when power is turned on, the POWER ON indicator is illuminated.

CHAPTER 4
LOGIC DIAGRAM DESCRIPTION

4.0 GENERAL

Detailed descriptions of selected portions of 5201 Terminal logic are provided to give the field engineer a better understanding of the Terminal operation. The logic diagrams contain all logic necessary to perform the function being described. Mnemonic names from the schematics have been used. The logic components have been assigned fictitious numbers (Ax) and have no actual reference designator for schematic tracing. Associated with selected logic diagrams are timing diagrams to provide an understanding of the sequence of events. The logic diagrams described are: Display Timing Generation (TC–SH 2 of 21), Screen Refresh Video Address Generation (TC–SH 4 of 21), Refresh Storage Control (TC–SH 4 of 21) and Poll Message Decode Logic T/R Control–SH 2 of 3).

4.1 DISPLAY TIMING GENERATION (Figure 4-1)

The Display Timing Generation logic generates all timing required for screen organization (Figure 4-2). The four types of timing generated are dot timing, character timing, row timing and line timing (Figure 4-3). The basic clock frequency of the Display Timing is the OSC (13.487 MHz). Referring to Figure 4-1, OSC is applied to A1 (Dot counter), which is a four-stage binary counter (74163). With the first rising edge of the OSC clock, the DC1 output is set. Counter A1 continues to increment on each OSC clock pulse until the DC8 output goes high (count of 8). Signal DC8 is applied to A3 (Character counter) as an enable and to A2 where it is inverted and applied to A1 as a reset. The outputs of A1 (DC1, DC2, DC4, DC8) are sent to a decoder which decodes DOT0-DOT8. At the next OSC clock time, A1 is reset and A3–CC1 is set. Counters A3 and A4 are a two-stage binary counter with an enable input and a carry out. Counter A3 continues to increment once each nine dots at DC8 time. When A3 has counted to 15 (CC1, CC2, CC4, CC8 high), CC15, the carry out, enables A4, the second stage of the Character counter. On the next rising edge of the OSC clock, A4-CC16 sets. The sequence just described continues until CC32, CC64 and DC8 are high (count 96), at which time the output of A5 resets A3 and A4 on the next OSC clock. The output of A3 and A4 (CC1, CC2, CC4, CC8, CC16, CC32, CC64) is applied to decoders which decode the character count (CT0-CT96). Signals CC32 and CC64 are also applied to AND gate A6, which at character 96 and DC8 enables row counter A7.

On the same OSC clock that resets the Character counter (A3 + A4), the Row counter (A7) is incremented by one. The Row counter is incremented every 97 character counts until row 11 (RC1, RC2, RC8). At row 11, if CUR is high, A12 causes A13 latch to set, which indicates that the cursor is to be painted on this line; therefore, one extra row must be generated for the cursor painting. This is done by AOI gate A8. If latch A13 is not set at row 11 time, A8 output goes high. This is applied through A9 and A10 to the reset of A7 (row counter), which resets on the OSC clock following the next count 97. The output of A8 is also applied through A15 and A16 to A17, the Line counter, which increments on the same OSC clock pulse on which the Row counter is reset, indicating that the first line is complete. If the cursor latch A13 is set, A8 is inhibited from decoding row 11 and is enabled to decode row 12 (row 8, 4), thus allowing one additional row count before resetting the Row counter and incrementing the Line counter. The sequence of events (row and line count) continues until AI gate A19 decodes line 25, row 7, which completes one screen presentation. At line 25, row 7 the Row counter A7 and the Line counter A17 are

reset and a new screen display sequence is begun. The Row counter outputs (RC1, RC2, RC4, RC8) and the Line counter outputs (LN1, LN2, LN4, LN8) are applied to decoders to decode RC0-11/12 and LN0-25, respectively. The Display Timing Generation logic is free-running in that, as long as the Terminal is energized, the count patterns are continuously generated.

4.2 SCREEN REFRESH VIDEO ADDRESS GENERATION

The purpose of the Video Address Generation logic is to sequentially generate Video Addresses (VA0-10) for use in addressing Refresh Storage Memory in synchronization with the display timing for Screen Refresh. Referring to Figure 4-4, A12, A14, A15, A16, A17 and A18 are the actual address generators. Counters A14, A16 and A18 form a three-stage binary counter which provides a maximum count of 2048—more than enough to generate the 1920 addresses needed to display one screen of characters. Counters A12, A15 and A17 are also a three-stage counter with a maximum count of 2048—more than enough to provide a count of 1840, the maximum count needed for the preset value. For the description of the Screen Refresh Video Address Generation logic, the starting point is assumed to be character 90 (CT90) of line 25, row 7, and all counters are reset (count 0). This point is just before the beginning of a new Screen Refresh. Referring to Figure 4-4, at CT78 latch A13 output (CTL) goes low, enabling the load function of the Video Address counter (A14, A16, A18). At the trailing edge of CT93 time, OI A11 output makes a transition from low to high, clocking the Address counter. Because load is active, the Address counter is preset to the value of the preset counter (A12, A15, A17), which for this description is zero. At CT95, CTL is disabled, thus removing the load enable from the Address counter. With CTL inactive (high), DC8 from AI, A10 is routed thrugh OI-A11 as the address clock. Since DC8 occurs once per character, the Video Address counter increments once per character for 79 more times until CT78, at which time CTL becomes active, thus inhibiting DC8. This generates 80 Video addresses, one for each valid character to be painted on the screen. At CT89 time, the video Address counter is reset in preparation for a new preset value. Because the preset value is still zero and remains zero until row 11 or row 12 is complete, the cycle repeats, thus generating the same 80 addresses for the first 12 or 13 rows of Screen Refresh, which is the first line on the screen. A1 and A2 determine whether the same addresses are to be generated for 12 or 13 rows. If the cursor is on the line, they are generated 13 times; otherwise, at CT80 of the 12th row latch A5 is set for 10 characters, thus allowing 80 DC clock pulses to clock the preset counter (A12, A15, A17). This creates a new preset value of 80, which for the next 12 or 13 rows is used as the Video Address value. The cycle just described continues for 24 lines, at which time –LN24 at A10 disables the Video Address counter from accepting a new preset value. At LN25, row 0, AND gate A7 is activated, which clears the preset counter through A8. Since line 25 has only seven rows, the preset value is not incremented by 80 at the end of line 25. Thus, at LN25, row 7, CT90 the new Screen Refresh is ready to begin.

4.3 REFRESH STORAGE CONTROL LOGIC DESCRIPTION

The Refresh Storage Control logic performs four different functions: (1) read data from storage for Screen Refresh, (2) read data from storage for transmission, (3) write data into storage from Terminal control and (4) refresh RAM's. Read data for Screen Refresh is done during the time of character 95 to character 79 (CT95-CT78). Read/write data to or from storage for Terminal control use is done during the time of character 79 to character 95, and RAM refresh is done when VAR INH is active.

4.3.1 Screen Refresh Control

Referring to Figure 4-5, Screen Refresh begins with latch A2 being activated at the end of character 94 time, which is a function of AI A1. Latch A2 active presets FF A8. The output of A8 is applied to AI A9, whose output goes low if VAR INH is not active (FF A25 set). Gate 11 A11 is a negative OR-INVERTER which, on sensing a low, goes high at the output. This high is applied to FF A15 and A14 and gates A16 and A17. With A8 set, A1 A10 has a high output which is inverted by A20 (A23=RESET) and applied to a 3-input negative OR-INVERT function whose output enables AI A22. From DOT1 to DOT3 time, A15 output (–RESET CLK) is active. From DOT5 through DOT8 time, A14 output (–RS CLK) is active. With A14 active, A22 is also active, thus generating –READ EN. At DOT7 time, A16 (RSDO STB) is active, and at DOT8 time A17 (–CGAL) is active; thus, one set of Screen Refresh control commands is generated. This cycle occurs for 79 more times until A3 decodes CT78-DC8, which is inverted and applied to the reset of latch A2 and FF A8; thus, Refresh Control command generation is inhibited until the beginning of character 95, at which time a new Screen Refresh cycle begins. One more constraint is placed on Refresh Control generation: during rows 11 and 12 character painting is not performed, as these rows are reserved for spacing and cursor presentation. Gates A5, A6 and A7 monitor the row and character counts. When rows 11 and 12 are being generated by Display Timing circuits, FF A8 is inhibited from generating Refresh Control commands. When FF A8 is set (Screen Refresh), –VA GT is generated, and addressing of Refresh Storage is a function of VA0-10.

4.3.2 Read/Write Data to Terminal Control

Software controls the reading or writing of data between Terminal Control and Refresh Control. Under normal circumstances, the actual operation is performed during horizontal screen retrace, which occurs between the time of characters 79 and 94. Referring to Figure 4-5, FF A31 has a signal RSCR as the J input of a JK FF; RSCR is generated as a result of the execution of an X'C7', X'C8' or X'C9' instruction. At DC8, following generation of RSCR, A31 is set. This is the indication to the Refresh Storage Control that either a read or write command is to be performed. The output of A31 is applied to AI A10, which during Screen Refresh is inhibited by FF A8 being set. At CT79 (CT78·DC8, Screen Refresh complete), FF A8 is cleared and A1 A10 is allowed to sample the read/write command flip-flop. If an RSCR has occurred, A10 output goes low. This low is applied to A19 through Inverter A18, and to A20. The other input of both A19 and A20 is from FF A23, which monitors WRITE., Signal WRITE is generated as the result of an X'C7' (store) instruction (X'C8' and X'C9' are read instructions). Write FF A23 is set on the next DC8 pulse after WRITE becomes active, and this indicates a WRITE command. The –WRITE EN gate A19 monitors FF A31 and A23, and if both are set, –WRITE EN is generated. Gate A20 monitors the reset side of the WRITE FF (A23), and if A31 is set and A23 is reset, a –READ EN command is generated. Associated with a WRITE/READ EN command, Refresh Storage Control must generate the other control commands to perform

the operation. The generation of these commands is initiated by the output of A10, which is routed through OI A11, where it is inverted and applied to FF A15 (–RESET CLK), FF A14 (–RSCLK), gate A16 (RSDO STB) and gate A17 (–CGAL) to generate one set of control commands (Figure 4-6) in the same way they were generated for Screen Refresh. One set of commands, instead of 80, is generated because the output of I A18 is routed, at DOT7 time, through A32, A33 and A34 to the reset side of FF A31 and FF A23, which reset. Resetting A31 and A23 inhibits further generation of Refresh Storage commands for READ/WRITE until another RSCR is generated. Only one character of data per row is read into or written out of Refresh Storage Memory to Terminal Control. During CT79 to CT94 time –VAGT is inactive, which makes addressing of Refresh Storage during a READ/WRITE operation a function of B Register.

4.3.3 RAM Refresh

The RAM's used for Refresh Storage Memory are dynamic, and they must be addressed at minimum time intervals. If a VAR INH instruction has occurred no Screen Refresh is performed; thus the RAM's are not being addressed. To ensure that the RAM's are kept refreshed when VAR INH is active, a special REF function is generated. Referring to Figure 4-5, FF A25 has VAR INH on its J input. Command VAR INH is software-initiated. At the next CT78 time following the activation of VAR INH, FF A25 sets. The reset output of A25 inhibits Screen Refresh by inhibiting A9. The set output of A25 is applied to FF A27, which actually controls the RAM refresh function. While VAR INH is active, A27 is set once per screen for one row at LNI ROW 1. The output of A27 is applied to AI A29, which is activated for 32 character times, or from CC32 to CC64 time. This output (A29) is applied to A11, which enables generation of 32 sets of Refresh Storage Control commands. These 32 commands are the required number of memory addresses needed to refresh the RAM's. The output of A29 is also inverted and applied as REF to Refresh Storage. REF is used to select the 32 addresses required for the RAM Refresh.

4.4 POLL MESSAGE DECODE LOGIC (Figure 4-7)

Poll messages received by the Terminal from the KAU are in a prescribed format. This format is EOT ('0D'), address ('00'–'07) Poll command. The Poll Message Decode logic checks for correct address and type of poll (General, Direct), and takes appropriate action depending on the status of the Terminal. The heart of the Decoder logic is Shift register A4, which provides timing for the sequence of events. Gates A1 and A 2 decode '0D', which, when the T/R Module flag (DR) indicates that one byte is complete, is applied through A5 to A6. Gate A6 has an input X4 from the SR A4, which is the earliest time for processing the received byte. Because the format is fixed, it is known that EOT will occur first. Thus, if at X4 time an 'OD' is decoded, it is assumed to be an EOT command, and A6 is activated by sending data. The generated EOT command is also applied to OR gate A27, which generates an SRCLK that is applied through A28 and A31 to the Shift register to increment the count to X3 for processing the Terminal address data.

The next information received by the Terminal is the Terminal address being polled. Gate A8 compares the Terminal address data from the KAU with the Terminal switch selected address. If there is a match, the A8 output is activated and applied to AI A9, which is activated at X3 time, the next time in the format. If there is no match at X3 time, A9 is not activated, but A10 is, thus generating a Shift register reset which is applied through A12, A26, A25 and A30 to SRA4, as SR LOAD and SRCLK to reset the Sequence counter. If the KAU message passes the address test, SR A4 is again incremented through A27, A28

and A31. This sets the SR to X2 in anticipation of a Poll command. Two types of Poll command are initiated by the KAU. These are General Poll (0C) and Direct Poll (0D). These are decoded by gates A18 and A13, respectively. When a General Poll command is detected, the Terminal replies with either an ACK or a data message. Determination of the Terminal response is a function of the Poll Message Decode logic. The General Poll command is detected by A18 at X2 time, which enables A19, A21 and A22. The output of A18 is also applied through A19 to A15, where Terminal response is tested. The Transmit Ready flip-flop FF A20 senses if the Terminal has a data message to send. If it has data to send, the detection of '0C' enables gate A15 to generate a SET DTP, which is coupled through A16 and A17 to Terminal Control, flagging it to transmit data. Also, gate A21 is enabled, thus incrementing the Shift register through A27, A28 and A31.

If, after detecting an '0C', the Terminal does not have a data response, gates A22 and A11 are enabled. Gate A22 generates a –SET, which triggers the Terminal to generate an automatic acknowledge response. In addition, –SET enables A12, which generates a Reset command through A26, A25, A30 and A31 to reset the Sequence counter. Unlike the General Poll command, when a Direct Poll command is decoded, the KAU forces the Terminal to give a data response. Gate A13 detects Direct Poll and routes it through A14 to Terminal Control as SET DTR. It also applies SET DTR through gates A26, A25, A36 and A31 to the Sequence counter to reset it. After the detection and processing of the Poll command the Poll Message Decode logic is reset to quiescent state, waiting for the next message from the KAU.

CHAPTER 5
SCREEN REFRESH FUNCTIONAL LOGIC DESCRIPTION

5.0 GENERAL

This chapter contains the functional logic analysis of the 5201 Terminal. Because the Terminal is microprocessor-controlled, most of its functions are software-originated. Screen Refresh is the only hardware-generated function; therefore it is the only function described in this chapter. The level of coverage for the Screen Refresh function is dependent on the complexity of the circuitry involved. To aid the user, back plane connector jacks and pins between the Terminal Control board and Refresh Storage board, and between the Refresh Storage board and the Display Monitor are shown on the functional logic diagram (Figure 5-1). Pin connections of individual components are shown where, for either troubleshooting or explanatory reasons, they might be beneficial to the user. In some instances not all the inputs and outputs of a module or function are shown. This is because those particular signals are not used during the Screen Refresh function.

5.1 SCREEN REFRESH

The function of Screen Refresh is to access, convert, control and display, in that order, EBCDIC character data that has been stored in a 2K Refresh Memory. One full-screen display (all 2K of Refresh Memory) is generated every 19.7 milliseconds. The frequency of the screen display is the function of the Display counters and decoders. The screen organization must be understood before a detailed functional discussion is presented.

5.1.1 Screen Organization

Referring to Figure 4-2, the screen is organized into 26 lines (lines 0-25). Line 25 is the last line to be displayed and appears at the top of the screen, where it is used to display Terminal mode (DATA, INSERT, DELETE, TRANSMITTING). Each line comprises rows of character times. Each of the first 25 lines (lines 0-24) has 12 rows, with the exception that the line displaying the cursor has 13 rows. In a purely Screen Refresh mode, the cursor is in the Home position; thus line 0 has 13 rather than 12 rows of character times. Each row comprises 97 character times (CT0-CT96); the first 80 of these character times (CT0-CT79) display valid characters. The remaining 17 character times (CT80-CT96) provide horizontal retrace times from one row to the next. In turn, each character time comprises nine dot times (DOT0-8). The first and last dot times for each character time are blanked, thus providing spacing between characters.

Screen painting (displaying characters on the screen) is performed by row. The first display point on the screen is row 0, character 0, dot 0, which is located on the far left of line 0. As this is the first dot time of a character, no valid (dot not illuminated) character data is displayed. The painting continues from left to right on the first row until character 80 time. At this time, accessing character data is inhibited, allowing a dead time of 17 character times to prepare the Screen Refresh logic and Display monitor for painting the second row (row 1). At dot 0, character 0 of the second row (row 1), the above sequence repeats. This sequence of events continues through row 10. At the beginning of row 11, character accessing is inhibited; however, the character and dot time generation continues. This is done at each row 11 time to provide a blank row for line-to-line separation of characters. Because the line being painted is line 0 and the cursor is located on this line, an additonal blank row is generated (row 12). This allows cursor video, when generated, to be displayed

without interruption of character data. Referring to Figure 4-2, observe the enlarged character block at the upper right corner of the figure. The sequence of row, dot and character times generated produces a 9 x 12 character matrix. This is referred to as a character block, and 80 valid character blocks are painted per line. The actual character to be painted is in a 9 x 7 matrix comprising rows 0-8 or rows 2-10, depending on the type of character, and dots 1-7. This matrix should be understood, as the character generators function under this concept.

Little has been said about lines 24 and 25. Line 24 is an invalid line time generated to provide vertical retrace time from the bottom right corner of the screen to the top left corner. Line 25 is actually a partial line of seven rows, and valid character data is inhibited during line 25. In place of character data, Error Video as a function of the mode of the Terminal is generated, causing the display to paint a bar under the applicable mode.

5.1.2 Refresh Memory Accessing

Accessing of Refresh Memory during Screen Refresh is controlled by Refresh Storage Control on the Terminal Control board. Because of the time involved in moving and converting data, Refresh Memory accessing is performed two character times before the character position on the Display Monitor. Thus the first character to be displayed on line 0 is accessed at character 95 time of line 25, row 7. Refresh Storage Control is under direct control of the Display counters and decoders. It detects conditions of the counters and generates the clocks, load commands and video addresses required to sequentially access Refresh Memory and move the accessed data to the Character Generators for conversion.

At the end of character 94 time (CT94-DC8), Refresh Storage Control is enabled to generate Refresh Memory access timing. On the next DOT1 (CT95 time) following CT94, a JK flip-flop (A12) is set. This activates –RESET CLOCK, which is sent to Refresh Address Select through J2 and J3, pin 55, where two reset clocks (–RESET 1, RESET 2) are generated and applied to the lower 1K and upper 1K of Refresh Memory for RAM initialization. At DOT3, –RESET CLOCK is inactivated, and –RS CLK is activated on detecting DOT5. Signal –RS CLK is routed through J2 and J3, pin 56, to Refresh Address Select, where two –RS CLK's (–RS1, –RS2) are generated and applied to the lower and upper 1K's of Refresh Memory as an enable. Signal –RS CLOCK is also applied internally to logic gate F04, where it generates a –READ ENABLE. Signal –READ ENABLE is sent through J2 and J3, pin 43, to the Output Memory Drivers. Signal –READ ENABLE selects the Refresh Memory output to the Output Buffer (Z39 and Z40) path of the Memory Drivers. Also at character 94 time of line 25, row 7, the Video Address registers are loaded from the Preset Value registers with an all-zero address. Video Address VA0-10 was sent by Refresh Storage Control through J2 and J3, pins 155-165, to Refresh Storage Select to generate Refresh Memory address –A0-A9. When Refresh Storage Control logic gate J07 generates a RSDO STB (Refresh Storage Data Out Strobe) at DOT7 and applies it through J2 and J3, pin 42, to the Output Buffer Z39 and Z40 as a load command, a 9-bit EBCDIC character code from address zero of Refresh Memory is loaded into Z39 and Z40. Because Refresh Memory is organized sequentially, address zero selects character zero of line zero.

The preceding sequence of events is repeated each character time until character 79 time, with only the Video Address incrementing by one each DOT8 time (address 0-79). At character 79 time the Screen Refresh timing (–RESET CLK, –RS CLK, –READ ENABLE, RSDO STB, Video Address incrementing) is inhibited until character 94 time, when the

same sequence of events is repeated for rows 1-10. The Video Addressing registers are reset to address zero before rows 1-10 because the same 80 characters are accessed from Refresh Memory 11 times for line 0. Before row 0 of line 1 the Preset Value register, used to determine the start address for Video Addressing, is incremented by 80, thus supplying the Video Address register with a Start address of 80 for the duration of line 1. This is the only difference in the sequence discussed between line 0 and line 1 Screen Refresh. For each line (2-23) the preset value is incremented by 80, thus allowing Refresh Storage Control to access sequentially by line the character data to be displayed on that line. The sequence is repeated until all 1920 valid character positions (80 x 24) have been accessed from Refresh Memory. At the conclusion of line 23 (last valid character line) all Refresh Memory accessing is inhibited. During line 25 time, Refresh Storage Control is initialized in preparation for painting a new screen display.

5.1.3 Parity Check

During Screen Refresh, before converting and displaying each character from Refresh Memory, a parity check is performed by Z44. Associated with each eight-bit EBCDIC character code stored in Refresh Memory is a parity bit. This parity bit is generated by an odd parity generator when the character is originally written into Refresh Memory. After the accessing of Refresh Memory by the Screen Refresh access cycle, the nine-bit character code (RDSO 0-7, RSPO) is applied to Z44. The parity bit (RSPO) configures the parity to be generated by the parity generator and checker (Z44). Configuration is such that the odd parity output of the generator is always active low if no character code error is present. This is accomplished by ROPE, which when active high, indicating an even number of ones in the character code, selects odd parity generation. When ROPE is inactive low, indicating an odd number of ones in the character code, it selects even parity generation, thus causing –DOPE at Z44–D6 to be always an active high except when the character code is in error. Signal –DOPE is applied to buffer Z42, where it is stored for later use by the character Generators.

5.1.4 Character Generator Operation

The Character Generator logic consists of two character generators (Z46, Z47) and associated logic, which receive addressing information from the Character Generator Address Buffer (Z42, Z41) and control information from the Character Generator Control located on the Terminal Control board. Associated with each Screen Refresh access cycle are Character Generator timing and control signals used to convert the accessed character from EBCDIC code to a special dot code stored in the Character Generators. For the Screen Refresh function there are two modes of character conversion: upper-case character generation and lower-case character generation. A third mode (Hex mode) of character generation is available; however, it is not covered here because, in normal Screen Refresh, hex character generation is not used. It is used only as a troubleshooting aid to the field engineer.

5.1.4.1 Upper-Case Character Conversion

Upper-case character conversion is performed by Character Generator Z46. At DOT7 time of each Screen Refresh memory access cycle, the Refresh Storage Control function generates –CGAL (Character Generator Address Load),which is sent through J2 and J3, pin 41, to the Character Generator Address Buffer, Z42 and Z41. The trailing edge of –CGAL loads the accessed character into Z42 and Z41. The six lower bits of the Character Generator Buffer (DO 0-5) are applied to the Upper-Case Character Generator as the address

of the special character code stored in a memory internal to the Character Generator. Signal DO 6 from Z42–D7 enables the appropriate character generator. If DO 6 is active high, the character to be converted is an upper-case character, and the Upper-Case Character Generator is enabled.

The Upper-Case Character Generator decodes the six-bit address (DO 0-5) and selects one of 64 character matrices stored in a 4032-bit memory located internally to the Character Generator. This 4032-bit memory is organized into 64 x 9 seven-bit segments. During row 0 the first segment (seven bits) of the segment matrix is selected; thus a seven-bit code for the character selected (one of 64 characters) is outputted to a serializer Z48 for each address (80 addresses) accessed during the Screen Refresh access cycle. The EBCDIC character codes stored in refresh Memory are actually used to address the corresponding character matrix in the Character Generator. At character 88 time (dead time of Screen Refresh) of row 0 the Character Generator control function generates a UC CLK, which is routed through J2 and J3, pin 27, to the Upper-Case Character Generator. Signal UC CLK increments a segment counter internal to the Character Generator. This counter selects the second of the nine segment character matrices. So, during row 1 of Screen Refresh, the second seven bits associated with each character accessed by each Refresh Memory access cycle are outputted by the character Generator. This sequence is repeated 80 times for each row, 2-10, with the exception of UC CLK, which occurs only once per row. At character 88 time of row 10, five additional UC CLK's are generated to reset the Segment counter to zero. After generating the reset UC CLK's, no further UC CLK's are generated until character 88 time of the next row 0. The only other input applicable to the Upper-Case Character Generator is DOPE, the output parity bit generated as a function of each EBCDIC character code accessed. When the code is correct, –DOPE is active high, and no effect is felt by the Character Generator. If –DOPE is active low, indicating an incorrect character code, the Character Generator is placed into the Complement mode. Normally the character dots painted on the screen are illuminated on a dark background. Signal –DOPE active low causes the seven-bit output of the Character Generator to be inverted. This creates a dark character on an illuminated background, providing the operator with a visual indication of incorrect parity.

5.1.4.2 Lower-Case Character Conversion

Lower-case character conversion is performed by Z47. At DOT7 time of each Screen Refresh memory access cycle, the Refresh Storage control function generates –CGAL, which is sent through J2 and J3 pin 41, to the Character Generator Address Buffer Z42-A8 and Z41-A8. The trailing edge of –CGAL loads the accessed character into Z42 and Z41. All eight bits (DO 0-7) are used for address steering of the Lower-Case Character Generator. This is required because the Lower-Case Character Generator not only contains the codes for lower-case characters, but also the codes for hex characters and control characters. In a purely Screen Refresh mode, DO 0-5 provides the address for the Character Generator. Signal DO 6, when inactive low, indicates a lower-case character. With DO 6 inactive low, the Lower-Case Character Generator is enabled and the Upper-Case Character Generator inhibited.

Internally, the two character generators are identical except for the special code character marks. The Lower-Case Character Generator decodes the six-bit address (DO 0-5) and selects one of 64 character matrices stored in a 4032-bit memory located internally to the Lower-Case Character Generator (Z47). This 4032-bit memory is divided into seven 64x9-bit segments. During row 0, the first segment (seven bits) of the matrix is selected. Thus a

seven-bit code for each character selected (one of 64) is outputted, on the same output lines as the Upper-Case Character Generator, to a serializer Z48 for each address (80 addresses) accessed during the Screen Refresh access cycle (80 cycles for each new row). At character 88 time of row 0, the Character Generator control function generates a LC CLK, which is routed through J2 and J3, pin 25, to the Lower-Case Character Generator. The LC CLK signal increments a segment counter internal to the Lower-Case Character Generator. This counter selects the second segment of a segment character matrix for character generation during row 1. The sequence repeats once each row until the end of row 10; at this time five additional LC CLK's are generated to reset the Segment counter, readying it to begin the next line of characters. At row 0, character 88 of the next line the Character Generator sequence starts again.

The one remaining valid input is the parity bit, –DOPE. When –DOPE is inactive high, there is no effect on the Lower-Case Character Generator. When –DOPE is active low, the seven-bit output of the Lower-Case Character Generator is inverted, causing the character to appear dark on an illuminated background. This alerts the operator that character parity error has been detected and appropriate action should be taken.

5.1.5 Character Serialization

The seven-bit common output of the Lower- and Upper-Case Character Generators is applied to an eight-bit parallel-to-serial converter (Z48). At DOT8 (Figure 4-6) time of character 96, row 7 of line 25, the Character Generator Control logic generates the first SER LD (serial load) of line 0, which is sent through J2 and J3, pin 21, to Z48-D1. On the positive-going edge this loads the first seven-bit segment of character 0, row 0 line 0,which was accessed from Refresh Memory and converted, during character 95 and 96 time, into Z48. During character 0 time of row 0, line 0 the Character Generator Control logic generates eight SER CLK's, which serial shift the seven bits of character 0 row 0, line 0 through the Control logic and D/A Converter logic to the Monitor to paint the seven valid dots and two invalid dot positions associated with character 0 time of row 0. A SER LD is generated at DOT8 of character 0 time to load the seven bits of character 1 into the serializer. At DOT0 of character 1 time, SER CLK's are generated, which begin to shift the seven valid dots and two invalid dot positions for painting character 1, row 0 on the display screen. The same sequence of one SERLD followed by eight SER CLK's is generated for each character time 2-79, thus painting the first nine bits of each of 80 accessed characters from Refresh Memory on row 0. At the beginning of character 80, all Screen Refresh activity is inhibited, including character serialization. During Screen Refresh dead time (characters 80–94), the Screen Refresh functions are readied for generation, conversion and serialization of the character data, stored in Refresh Memory, to be displayed during row 1 time.

5.1.6 Screen Display Control

The serialized digital data from the parallel-to-serial converter (Z43) is routed through AND gate Z53 before being converted to analog data. This gate provides display blanking control of the character data by the Video Control function located on the Terminal Control board. In addition to the blanking control provided by Z53, Video Control also applies intensity control commands to the analog data before it is sent to the Monitor for display. Three types of blanking, OCA (Operator Command Area) blanking, data blanking and control character blanking, are performed during Screen Refresh.

5.1.6.1 Operator Command Area Blanking

Video Control logic is under direct control of the Display Counters and Decoders. The OCA of the Display screen is the first 40 characters of line 23. The OCA Blanking command is initiated by bit 1 of the Job Definition byte sent from the Job Definition register to the Video Control logic. When bit 1 is active high, the Video Control logic generates a –DATA BLANK signal for the first 40 characters of line 23. The –DATA BLANK signal is sent through J2 and J3, pin 17, to AND gate Z53. This signal inhibits serialized digital data from being applied to the digital/analog conversion circuits; thus the first 40 character blocks of line 23 are blanked on the screen.

5.1.6.2 Data Blanking

Selected data fields of the Screen display may be blanked. Video Control monitors the character data (RSDO 0-7) as it is accessed from Refresh Memory. When a field definition character (RSDO 6 and 7 both inactive low) is detected, the Video Control logic examines the condition of RSDO 1, which, when active high, indicates the associated character field is to be blanked. Signal RSDO 1 high generates a –DATA BLANK enable, which remains active low until a new Field Definition character is detected. The –DATA BLANK INHIBIT is sent through J2 and J3, pin 17, to Z53-64, where serialized digital character data is inhibited from being converted to analog data. This causes the data field not to be displayed on the monitor. Also, Data Blanking Inhibit commands are generated during row 11 and 12 times to ensure that no character data is displayed at these times on the Monitor.

5.1.6.3 Control Character

Whenever a Field Definition character is detected during normal Screen Refresh, an active low –HEX BLANK command is applied to the Video Control logic. Signal HEX BLANK active generates a Control Character Blank command (–CC BLANK) for one character time each row. This –CC BLANK command is sent through J2 and J3, pin 18, to Z453-D3, where it inhibits the serialized control character (Z53-A5) from being connected to analog data and displayed on the Monitor.

5.1.7 Video Generation and Control Logic

Unblanked serialized digital character data from Z53-A8 is applied to the Video Generation and Control logic, where it is intensity-controlled and converted to Monitor Video for display on the Monitor. Three types of intensity control are performed on the character data. One type is the hardware-generated intensity control (–L1), which, in response to selected bits of the Job Definition register, lowers the intensity of variable or fixed data fields. Another type is the manual intensity control which the Video Generation and Control logic receives from the Power Supply assembly. The third type is a manual contrast control which the Video Generation and Control logic receives from the Power Supply assembly. Unlike the other intensity controls, which act on the character data, the contrast control varies the overall video level, thus affecting the total screen display.

5.1.7.1 Digital-to-Analog Conversion

Digital-to-analog data conversion of serialized character data output from Z53 is performed by applying the digital data to a "NAND" driver Z49. When the data bit is active high, the output of Z49 is low. This low is applied to amplifier Q1, which is driven towards cutoff. Amplifier Q1 near cutoff generates a large positive voltage on its collector. This positive

voltage forward-biases diode D5. With D5 forward biased, emitter follower Q3 increases conduction, causing a more positive voltage to be felt at its emitter. The emitter follower output (from emitter) is sent to the Display Monitor as Monitor Video. When the digital data bit is an inactive low, Z49 output is high. A high at Amplifier Q1 increases Q1 conduction, causing the positive voltage at the collector to decrease. Decreasing the voltage at the collector of Q1 forces diode D5 to decrease in conduction (go towards back bias), which causes a decrease in conduction of emitter follower Q3. The emitter of Q3 is forced towards a ground level, which is applied as Monitor Video to the Display Monitor.

5.1.7.2 Manual Intensity Control

The Manual intensity control is a variable 1K resistor located on the Power Supply assembly. It forms a voltage divider network in conjunction with the Q1 collector resistor. When the intensity control is turned clockwise, less voltage is applied to the anode of D5, thus causing less conduction of diode D5 in response to the digital character data bits being converted. This causes emitter follower Q3 to conduct less, thus generating a lower Monitor Video voltage, resulting in a less intensified screen display. When the intensity control is turned counterclockwise the opposite effect occurs.

5.1.7.3 Automatic Intensity Control

Besides the manual intensity control, the intensity may be automatically controlled for variable data fields or fixed data fields when commanded by bit 4 or 5, respectively, of the Job Definition byte. The Video Control function examines bits 4 and 5 from the Job Definition register. When either or both of the bits are active high and the Video Control logic detects the applicable variable or fixed data fields (bit 4 for variable data, bit 5 for fixed data), a low-intensity (–L1) command is sent through J2 and J3, pin 12, to the Video Generation and Control logic. At the Video Generation and Control logic, –L1 inhibits serial digital data from being applied to Q1, and instead routes it to the base of Q2. When the digital bit being converted is active high, Q2 is driven towards cutoff, presenting a positive voltage to the anode of D6. The positive voltage is lower than that processed by Q1, due to the action of a resistance network comprising the Q2 collector resistor and the manual setting of the contrast control. This causes emitter follower Q3 to conduct less than it would if data were processed through Q1, so that the Monitor Video produced for an active high data bit is less positive and the character displayed on the screen appears dim. Signal –L1 remains active until the detection of a new data field at this time.

Video Control determines whether the new data field should be low- or high-intensity as defined by the Job Definition Byte.

5.1.7.4 Contrast Control

When not being used for the low-intensity function, Q2 is conducting heavily. The amount of conduction is a function of the collector voltage, and the collector voltage is determined by the manual Contrast Control setting. The conduction of Q2 determines the bias voltage at the base of the output emitter follower, and the overall Monitor Video level may be lowered or raised by changing the setting of the Contrast Control. Turning the Contrast Control clockwise causes the Monitor Video static level (without data) to decrease, thus causing a decrease of the screen display contrast. Similarly, turning the Contrast Control counterclockwise increases screen contrast.

CHAPTER 6
TERMINAL MAINTENANCE

6.0 GENERAL

This chapter provides all the information required to repair the seven major assemblies of the 5201 Terminal at the Branch Office and Field Service Center (FSC) levels. The seven major assemblies are, in the order discussed:

(1) Monitor
(2) Keyboard
(3) T/R Feature Interface board
(4) Refresh Storage board
(5) T/R Control board
(6) Terminal Control board
(7) Power Supply

6.1 MAINTENANCE PHILOSOPHY

The maintenance philosophy used to troubleshoot and repair the seven major assemblies varies with the assembly being repaired. In all cases a known fault is assumed to exist, with verified information concerning the nature of the trouble written on a tag attached to the faulty assembly. The four logic boards and the keyboard are fault-analyzed and repaired, using an independent test station. This test station is a 5201 Rev A Terminal which has been modified to make it compatible with Rev C Terminals. The 5201 Rev A Terminal is used because it facilitates easy access to the logic boards and allows the field engineer to perform prepared diagnostic routines designed to aid in isolating faults to a specific component with a minimum of signal tracing. The use of the Rev A Terminal also allows the field engineer to make full use of the Terminal FE Panel in exercising the faulty assembly. Although the diagnostic routines may be accessed either from an 8K Storage board or from a PROM board, it is recommended that an 8K Storage board be used. Selected routines use automatic halt traps to detect a faulty functional area, and it is necessary to remove the halt trap to use the routine as an exerciser. In this way the field engineer can repeatedly exercise the faulty area, using a scope to isolate the bad component. Routine modification is required (refer to Appendix A) to remove the fault trap. The 8K Storage board allows the field engineer to modify the routine from the FE Panel, whereas the routines stored on a PROM board cannot be modified.

In preparing to use the test station for troubleshooting and repair of the logic boards and the keyboard, first ensure that the inverters used on the output of the ROM's located on the T/R Control board have been removed. With the inverters installed, access of the 8K Storage board or the PROM board is not possible. Place the faulty logic board on an extender at the test station, ensuring that it is well seated and the board is not touching any grounded area of either the test station or the test bench. Energize the test station, and refer to the appropriate discussion and applicable flowchart for the assembly being tested.

The bad assembly should be repaired, preferably using the bad operational Terminal. If this is not possible, a Terminal with the same Rev level must be available in which to install the defective assembly. With the defective assembly installed and energized, the appropriate discussion and flowchart or table should be referred to for troubleshooting and repair.

Each logic board is keyed to a fault isolation flowchart which provides the sequence of tasks to be performed and the expected results if no fault is present in the function being tested. Because of the unlimited fault possibilities, the flowcharts assume only that a fault exists on the logic board. The flowcharts check all the functional areas of the faulty board in a logical sequence. This ensures that any fault condition located is a valid fault for the functional area or areas being checked, and not a fault due to hardware used in detecting it. Care should be taken to follow the flowchart precisely. This ensures the integrity of the flowchart and the diagnostic routine involved. Once the faulty functional area has been located, refer to the applicable timing diagrams and schematics to isolate the defective component.

6.1.1 Test Equipment

Fault analysis of the 5201 Terminal is planned to use a minimum of test equipment. A 365 Tektronics scope or equivalent is required for monitoring pulse trains occurring in the nanosecond range. The volt/ohm meter now used at the repair facilities is also needed. A third piece of test equipment, which is recommended but not necessary, is a pulse catcher capable of switching times of 100 nanoseconds. This facilitates the isolation of faulty components.

6.2 MONITOR FAULT ISOLATION

This section provides procedures for fault isolation, component replacement, and adjustment of the 14-inch monitor (P N S09-140-001). The faults indicated on the flowcharts are not all-inclusive, but they are the common faults. Procedures are supplied for removing and replacing defective components after a fault has been isolated, and adjustments are outlined for returning the monitor to normal operation.

6.2.1 Troubleshooting Flowcharts

The flowcharts (Figures 6-2 through 6-6) are divided into five functional groups: Video, Horizontal, Vertical, Sync and Power.

6.2.2 Replacement Procedures

Procedures are given for replacing components on the PC board, the CRT, and plug-in transistors. Before replacing any part, turn off the power and remove the AC power cord from the wall outlet. The locations of the components and parts are shown in Figures 6-7 and 6-8.

6.2.2.1 PC Board Component Replacement

a. Unplug the leads external to the PC board.

b. Lift the PC board from the standoff insulators.

c. Unsolder components on the board from the plated side using a temperature-controlled iron and solder sucker.

CAUTION

Damage to the board can result from overheating.

d. Reverse the above procedure to install the PC board.

6.2.2.2 CRT Replacement

CAUTION

Use extreme care in handling the CRT. Rough handling may cause it to implode.

a. Discharge the CRT by shorting the second anode to ground.
b. Disconnect the CRT socket, yoke and second anode lead.
c. Remove the CRT from the chassis by removing the four screws at the corners.
d. Reverse the above procedure to install the CRT.

6.2.2.3 Plug-In Transistor Replacement

NOTE

When replacing any plug-in transistor, be sure the mounting screws are tight in order to ensure adequate cooling.

a. Holding the transistor, remove the screws securing it.
b. Apply silicone grease evenly to both sides of the mica insulator before installation.
c. Align the socket with the heat sink and fasten the socket and transistor in place.

NOTE

Do not strip the mounting screws. A poor electrical and mechanical connection can result.

6.2.3 Adjustments

The normal adjustments to be made on the Monitor are: the regulator, the video bias, the horizontal oscillator, and vertical size and linearity. Before any adjustments are made, check the power (AC input, B+).

6.2.3.1 Regulator Adjustment

CAUTION

Maladjustment of low-voltage regulator may result in damage to horizontal output transistor.

a. Connect the Monitor to AC outlet 120/240 volts; be sure SW1 is in the correct position.
b. Be sure the normal operating signals (video, sync) are connected to the Monitor.

c. Connect a digital voltmeter or other accurate meter to pin 2 (+73V) on the PC board.

d. Adjust R74 for an output of +73VDC.

CAUTION

Do not run control R74 through its range—the Monitor can be damaged.

6.2.3.2 Video Amplifier Bias

a. Connect the test equipment (meter, scope) to measure +30VDC to the junction of R12 and R18.

b. Adjust R10 for a reading of +30VDC.

c. Disconnect the video input lead, if necessary, to eliminate noise.

6.2.3.3 Horizontal Oscillator Adjustment

a. Apply normal operating signals to the Monitor.

b. Adjust the core of coil L1 (horizontal set) until horizontal blanking lines are vertical.

c. Center the display, if necessary, with the magnets on CRT neck.

6.2.3.4 Vertical Size and Linearity

a. Apply normal operating signals to the Monitor.

b. Adjust the vertical size, R65, and vertical linearity, R59, alternately to obtain a raster which fills the screen to the correct height with uniformly spaced horizontal lines.

c. Center the display, if necessary, with magnets on CRT neck.

6.3 KEYBOARD REPAIR AND TROUBLESHOOTING

6.3.1 General

The following paragraphs describe procedures and techniques for maintaining, repairing and troubleshooting the microswitch Keyboard (P N 2010-5009 or P N 2010-5020). The procedures are planned to aid in making quick diagnoses, repairs, removals and replacements, or in performing detailed troubleshooting.

6.3.2 Troubleshooting

When troubleshooting the Keyboard, use the keystroke gathering routine (Terminal Control Fault Isolation), loaded or called by means of the FE Panel, to test the Keyboard. After running the routine and identifying the problem, follow the procedures in the following paragraphs. Problems originating at the Keyboard can be classified in the following general categories:

a. No character code input to Terminal Control board.

b. Incorrect character code input to Terminal Control board.

c. Continuous character codes to Terminal Control board.

6.3.2.1 No Character Code Input to Terminal Control

Use the following procedure when the Keyboard trouble is suspected to be no character code input to Terminal Control.

a. Most probable cause of trouble:

(1) Inoperative strobe circuit IC33.

(2) No +5VDC or −12VDC input to Keyboard, or open GND.

b. Remedy

(1) With the Keyboard connected, with power on, and with a letter key held down, measure voltages at pins P1-11 to P1-3 (bits 0 to 8).

(2) If all pins measure +2.55 (minimum) VDC, check for open GND (P1 pin K to all data bit pins).

(3) If all pins do not measure +2.55VDC in step (2), proceed to paragraph 6.3.2.2.

(4) If GND is good in step (2), check power connections to IC33 +5VDC (pin 37) and −12 VDC (pin 39).

(5) If the correct voltages are measured in step (4), replace IC33.

6.3.2.2 Continuous Character Code Input to Terminal Control

When the Keyboard trouble is suspected to be that the same character code is being sent to the Terminal Control board, regardless of what key is depressed, the following procedure can be used to localize the faulty key or switch module. For this problem, no consideration is given to the validity of the character: the character code may be valid or invalid.

a. Most probable cause of trouble:

(1) Keyboard locked up by faulty key or switch module.
(2) Shorted IC outputs.
(3) Shorted output connector P1.

b. Determine cause of trouble:

(1) With Keyboard connected and power on, measure voltages on the following pins of connector P1:

Pin 4 – bit 7
Pin 5 – bit 6
Pin 6 – bit 5
Pin 7 – bit 4
Pin 8 – bit 3
Pin 9 – bit 2
Pin 10 – bit 1
Pin 11 – bit 0

NOTE

Inactive high is +2.55 (min) VDC. Active low is 0.6 (max), VDC.

(2) Refer to Table 6-1 or 6-2, and Figure 6-10 or 6-11, as applicable, and ascertain the corresponding key for the character code in step b(1).
(3) Remove and replace entire switch module (see NOTES).

EXAMPLE:

Pin 3–bit 8 = 1
Pin 4–bit 7 = 1
Pin 5–bit 6 = 0
Pin 6–bit 5 = 0
Pin 7–bit 4 = 0
Pin 8–bit 3 = 0
Pin 9–bit 2 = 0
Pin 10–bit 1 = 0
Pin 11–bit 0 = 1

} Code for lower-case.
Remove "a" key module.

Table 6-1 21-Inch Keyboard Codes

Key	DATA OUTPUT MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
1	BE	BE	BE	E	23
2	CTRL	CTRL	CTRL	CTRL	CTRL
3	—	—	—	—	—
4	F1	5A	F1	5A	43
5	F2	7C	F2	7C	44
6	F3	7B	F3	7B	45
7	F4	5B	F4	5B	46
8	F5	6C	F5	6C	47
9	F6	4A	F6	4A	48
10	F7	50	F7	50	49
11	F8	5C	F8	5C	52
12	F9	4D	F9	4D	53
13	F0	5D	F0	5D	42
14	60	6D	60	6D	54
15	7E	4E	7E	4E	10
16	A0	5F	A0	5F	10
16A	FA	BA	FA	BA	10
17	11	09	11	09	ED
18	EB	EB	EB	EB	10
19	DB	DB	DB	DB	10
20	—	—	—	—	—
21	BF	BF	BF	BF	24
22	BB	BB	BB	BB	20
23	9B	8B	9B	8B	10
24	D8	13	98	D8	10
25	E6	13	A6	E6	55
26	C5	13	85	C5	56
27	D9	13	99	D9	57
28	E3	13	A3	E3	58
29	E8	13	A8	E8	59
30	E4	13	A4	E4	62
31	C9	13	89	C9	63
32	D6	13	96	D6	64

Table 6-1 21-Inch Keyboard Codes (continued)

KEY	DATA OUTPUT MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
33	D7	13	97	D7	65
34	6A	A1	6A	A1	66
35	AA	4F	AA	4F	67
36	CF	CF	CF	CF	10
37	11	09	11	09	EE
38	CB	CB	CB	CB	10
39	DC	DC	DC	DC	10
40	—	—	—	—	—
41	8C	8C	8C	8C	25
42	BC	BC	BC	BC	21
43	SHIFT LOCK	SHIFT LOCK	SHIFT LOCK	SHIFT LOCK	SHIFT LOCK
44	C1	13	81	C1	68
45	E2	13	A2	E2	69
46	C4	13	84	C4	72
47	C6	13	86	C6	73
48	C7	13	87	C7	74
49	C8	13	88	C8	75
50	D1	13	91	D1	76
51	D2	13	92	D2	77
52	D3	13	93	D3	78
53	5E	7A	5E	7A	79
54	7D	7F	7D	7F	30
55	8A	9A	8A	9A	31
55A	B8	B9	B8	B9	10
56	—	—	—	—	—
57	CD	CD	CD	CD	10
58	CA	CA	CA	CA	10
59	CE	CE	CE	CE	10
60	—	—	—	—	—
61	8D	8D	8D	8D	26
62	BD	BD	BD	BD	22
63	—	—	—	—	—
64	SHIFT	SHIFT	SHIFT	SHIFT	SHIFT

Table 6-1 21-Inch Keyboard Codes (continued)

	DATA OUTPUT				
KEY	MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
65	E9	13	A9	E9	32
66	E7	13	A7	E7	33
67	C3	13	83	C3	34
68	E5	13	A5	E5	35
69	C2	13	82	C2	36
70	D5	13	95	D5	37
71	D4	13	94	D4	38
72	6B	4C	6B	4C	39
73	4B	6E	4B	6E	3A
74	61	6F	61	6F	3B
75	SHIFT	SHIFT	SHIFT	SHIFT	SHIFT
76	–	–	–	–	–
77	DD	DD	DD	DD	10
78	CC	CC	CC	CC	10
79	DE	DE	DE	DE	10
80	–	–	–	–	–
81	40	40	40	40	10
82	–	–	–	–	–
83	–	–	–	–	–
84	–	–	–	–	–
85	–	–	–	–	–
86	–	–	–	–	–
87	–	–	–	–	–
88	–	–	–	–	–
89	8E	8E	8E	8E	27
90	REPEAT	REPEAT	REPEAT	REPEAT	REPEAT
91	–	–	–	–	–
92	–	–	–	–	–
93	–	–	–	–	–
94	–	–	–	–	–
95	–	–	–	–	–
96	–	–	–	–	–
97	EA	EA	EA	EA	10

Table 6-1 21-Inch Keyboard Codes (continued)

	DATA OUTPUT				
KEY	MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
98	DA	DA	DA	DA	10
99	EC	EC	EC	EC	10
100	—	—	—	—	—
101	F7	F7	F7	F7	10
102	F8	F8	F8	F8	10
103	F9	F9	F9	F9	10
104	F4	F4	F4	F4	10
105	F5	F5	F5	F5	10
106	F6	F6	F6	F6	10
107	F1	F1	F1	F1	10
108	F2	F2	F2	F2	10
109	F3	F3	F3	F3	10
110	F0	F0	F0	F0	10
111	70	70	70	70	10

	Control Input IC-33		
	Pin 17 S1	Pin 18 S2	Pin 19 S3
Mode 1 — Upper-case and numerics	0	0	0
Mode 2 — Upper-case and numerics shifted	1	0	0
Mode 3 — Lower-case and numerics	0	1	0
Mode 4 — Lower-case and numerics shifted	1	1	0
Mode 5 — Control	1	1	1
	0	0	1
	0	1	1
	1	0	1

Bit Weight Example

Bit No.	7	6	5	4	3	2	1	0
Level	1	0	0	0	0	0	0	1

81 Hex Code

Table 6-2 19-Inch Keyboard Codes

KEY	DATA OUTPUT MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
4	F1	5A	F1	5A	98
5	F2	7C	F2	7C	9F
6	F3	7B	F3	7B	81
7	F4	5B	F4	5B	83
8	F5	6C	F5	6C	84
9	F6	4A	F6	4A	85
10	F7	50	F7	50	A9
11	F8	5C	F8	5C	AA
12	F9	4D	F9	4D	AC
13	F0	5D	F0	5D	BC
14	60	6D	60	6D	A3
15	7E	4E	7E	4E	A2
16	A0	5F	A0	5F	60
16A	REPEAT	REPEAT	REPEAT	REPEAT	REPEAT
17	ED	ED	ED	ED	ED
18	EB	EB	EB	EB	EB
19	DB	DB	DB	DB	DB
23	FUNCT. (1)	FUNCT. (1)	FUNCT. (1)	FUNCT. (1)	FUNCT. (1)
25	E6	E6	A6	E6	94
26	C5	C5	85	C5	82
27	D9	D9	99	D9	8E
28	E3	E3	A3	E3	86
29	E8	E8	A8	E8	BD
30	E4	E4	A4	E4	96
31	C9	C9	89	C9	97
32	D6	D6	96	D6	AD
33	D7	D7	97	D7	8F
34	6A	A1	6A	A1	BE
35	AA	4F	AA	4F	BF
36	CF	CF	CF	CF	CF
37	EE	EE	EE	EE	EE
38	CB	CB	CB	CB	CB
39	DC	DC	DC	DC	DC

Table 6-2 19-Inch Keyboard Code (continued)

KEY	DATA OUTPUT MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
43	9B	8B	9B	8B	BB
44	C1	C1	81	C1	92
45	E2	E2	A2	E2	95
46	C4	C4	84	C4	8D
47	C6	C6	86	C6	9D
48	C7	C7	87	C7	87
49	C8	C8	88	C8	C8
50	D1	D1	91	D1	A4
51	D2	D2	92	D2	9A
52	D3	D3	93	D3	AE
53	5E	7A	5E	7A	BA
54	7D	7F	7D	7F	89
55A	AF	9C	AF	9C	9C
57	CD	CD	CD	CD	CD
58	CA	CA	CA	CA	CA
59	CE	CE	CE	CE	CE
64	SHIFT	SHIFT	SHIFT	SHIFT	SHIFT
65	E9	E9	A9	E9	93
66	E7	E7	A7	E7	8C
67	C3	C3	83	C3	90
68	E5	E5	A5	E5	9E
69	C2	C2	82	C2	8A
70	D5	D5	95	D5	A5
71	D4	D4	94	D4	A8
72	6B	4C	6B	4C	99
73	4B	6E	4B	6E	AF
74	61	6F	61	6F	A7
75	SHIFT/ KANA	SHIFT/ KANA	SHIFT/ KANA	SHIFT/ KANA	SHIFT/ KANA
77	DD	DD	DD	DD	DD
78	CC	CC	CC	CC	CC
79	DE	DE	DE	DE	DE
81	40	40	40	40	40
97	EA	EA	EA	EA	EA

Table 6-2 19-Inch Keyboard Codes (continued)

KEY	DATA OUTPUT MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
98	DA	DA	DA	DA	DA
99	EC	EC	EC	EC	EC
101	F7	F7	F7	F7	F7
102	F8	F8	F8	F8	F8
103	F9	F9	F9	F9	F9
104	F4	F4	F4	F4	F4
105	F5	F5	F5	F5	F5
106	F6	F6	F6	F6	F6
107	F1	F1	F1	F1	F1
108	F2	F2	F2	F2	F2
109	F3	F3	F3	F3	F3
110	F0	F0	F0	F0	F0
111	DF	DF	DF	DF	DF

	Control Input IC–33		
	Pin 17	Pin 18	Pin 19
	S1	S2	S3
Mode 1 – Upper-case and numerics	0	0	0
Mode 2 – Upper-case and numerics shifted	1	0	0
Mode 3 – Lower-case and numerics	0	1	0
Mode 4 – Lower-case and numerics shifted	1	1	0
Mode 5 – KANA	1	1	1
	0	0	1
	0	1	1
	1	0	1

Bit Weight Example

Bit No.	7	6	5	4	3	2	1	0
Level	1	0	0	0	0	0	0	1

81 Hex Code

NOTES

1. No spare switch modules are provided on the Keyboard.
2. When replacing the solid-state switch module, the entire assembly must be replaced according to paragraph 6.3.3. This is a factory-calibrated assembly. The magnets, Hall generator, and plunger travel-distance are matched and cannot be interchanged.
3. If the problem is not found in steps b(1) through b(3), replace IC33.

(4) Check between P1–pins J,K (GND) and other data bit pins for shorts.

6.3.2.3 Incorrect Character Code Input to Terminal Control

When the Keyboard trouble is suspected to be an incorrect character code input to the Terminal Control board, proceed as follows:

a. Probable Cause:

(1) Common ground to all data bit pins.
(2) Defective MOS Encoder circuit.
(3) Defective switch module.

b. Remedy:

(1) With power off, check for ground between P1–pin K and all data bit pins. There should be no shorts.
(2) With the keyboard connected and power on, depress and release several letter keys and measure the strobe pulse at P1–12.
(3) The strobe pulse in step (2) should go to logic "1" with a pulse width of 10–200 microseconds each time the key is depressed.
(4) If the correct strobe pulse is not measured in step (3), check the output of IC33–pin 33.
(5) If the strobe pulse is correct in step (4), replace IC30.
(6) If the strobe pulse is incorrect in step (4), replace IC33.
(7) If the strobe pulse is correct in step (3), check the power input to IC33 (+5VDC at pin 37, –12VDC at pin 39).
(8) If the power is correct, check the data bit code at the output of IC33 for a particular key against the codes listed in Table 6-1 or 6-2.
(9) If the code in step (8) is incorrect for several keys, replace IC33.
(10) If the code in step (8) is incorrect for a particular key, recheck the code; if it is still incorrect, replace the particular switch module.

6.3.3 Keyboard Corrective Maintenance

This section describes repair, removal, replacement and cleaning procedures for the Keyboard. These corrective maintenance procedures are required to restore a Keystation to operational status after fault isolation. After repairs have been made to a Keyboard, it is recommended that some operator functions be performed to verify system operations with the repaired Keyboard.

6.3.3.1 Keyboard Switch Module Removal and Replacement Procedure

Proceed as follows when removing and replacing solid-state switch modules on the microswitch Keyboard:

a. Removal:

(1) Lift Keyboard from the desk.

(2) Disconnect ground cable and cables from J1 and J3. Place Keyboard on a flat surface.

(3) Remove button from module being replaced by pulling upward, or prying upward with padded tool on its under surface.

(4) Remove as many adjacent buttons as required to furnish adequate work space.

CAUTION

Remove buttons from alternate-action keys only when they are in free position. Failure to do this results in damage to the module.

(5) Slip module extraction tools (in replacement kit) down the inside face of each "D" bracket (see Figure 6-12). Place extraction tools through each window so that the tip of each tool rests against the outside bottom of the mounting rail.

(6) With extraction tools still in position, grip the switch module plunger with pliers and pull straight out.

(7) Unsolder (see Figure 6-12) and remove the black plastic lead-frame package from the termination board. When unsoldering terminals, use solder sucker to remove all solder from pin holes.

CAUTION

Use 15-watt soldering iron in step (7). Do not allow soldering iron to remain on terminal for more than 4 seconds, as excessive heat can cause damage to the lead-frame package.

b. Installation or replacement:

CAUTION

When installing a switch module, the entire assembly must be replaced, as the chips and plunger are matched. DO NOT DISASSEMBLE.

(1) Ascertain that switch mounting frame is not damaged or bent. If frame is bent, form it back to original shape before installing switch module (see Figure 6-13).

(2) Determine type of plunger (black or gray) and assemble return spring (see Figure 6-13) on the spring boss located on the bottom of switch plunger.

(3) Insert module in Keyboard switch mounting frame (see Figure 6-13) so that the orientation arrow (top of module housing) is pointing in same direction as other modules in same row.

(4) Ascertain that the return spring seats on the the raised boss (bottom of mounting frame) and the four terminals extend through the PC board.

(5) Check for the following conditions:

- Plunger moves freely and noiselessly.
- Top surface of module does not extend above those of other modules on same row.
- Module is locked securely in place by "D" bracket on mounting frame.
- All terminals extend through PC board.

NOTE

If it is necessary to remove module to correct any of above conditions, be extremely careful not to damage or disfigure any portion of the plunger, since it will affect module operation and/or the retention of the button.

(6) Solder terminals in place using 60/40 tin-lead solder.

NOTE

To ensure a good solder joint, apply a minimum amount of 0.050-inch-diameter solder 90 degrees to the terminal. Excess solder can cause a terminal-to-frame short.

CAUTION

If the lead frame is to be returned for evaluation, apply soldering iron for a period no greater than 4 seconds, as excessive heat can damage the lead-frame package. To prevent damage to wire bonds between the chip and the lead-frame, handle the lead-frame carefully. It is potted with a resilient material which completely surrounds the chip, resulting in a chip that is not rigidly mounted, but rather encapsulated in a flexible compound. If pressure is applied to the potting or if the lead-frame is flexed, the wire bond can be broken. The failure mode can be either a complete or an intermittent "open." The latter is caused by further separation due to expansion as the result of temperature variations. If the lead-frame must be handled, hold it by the edges. Apply no pressure to the chip area, and do not flex the lead-frame.

(7) Reassemble buttons on all switch modules, being careful to place the buttons on the correct switches.

6.3.3.2 Space Bar Assembly Removal and Replacement Procedure

Removal and replacement of the space bar assembly is different from that of the conventional switch module, because of the difference in its construction. The space bar assembly includes two module housings as mounting supports (excluding chip and magnets), a torsion wire and a noise suppressor. The steps for removing and replacing this assembly are described in the following procedure.

a. Removal:

(1) Remove space bar mounting module by placing the module extraction tools (in the replacement kit) down the inside face of the "D" brackets on the mounting frame.

(2) When the extraction tools release the space bar from the mounting frame "D" brackets (see Figure 6-14), pull upward until it is free of "D" brackets.

(3) Repeat steps (1) and (2) for the remaining space bar mounting modules.

(4) When both sides of the space bar are released, pull evenly on both ends to lift it free of the Keyboard, being careful not to damage or disturb the switch module under the space bar.

(b) Installation or replacement:

(1) Compare removed "D" brackets with unused "D" brackets in space bar row to determine if they are damaged, bent or misshapen. If so, re-form them to their original shape.

(2) Remove old noise suppressor (see Figure 6-15) from switch module operated by space bar.

(3) Install new noise suppressor by pushing down with finger, using moderate pressure (see Figure 6-15).

(4) Position new space bar assembly in proper location and press down (applying even pressure) on both ends of space bar until support modules are locked in place by "D" brackets on mounting rail (see Figure 6-16).

(5) If it is necessary to replace space bar torsion wire, position mounting modules at a convenient angle and insert torsion wire (see Figure 6-17).

6.3.3.3 Keyboard and Switch Cleaning Procedure

Use the following procedure when drinks or other gummy liquids are spilled on the keyboard. It is not recommended for periodic cleaning unless conditions warrant it.

a. Keyboard cleaning. (This procedure is intended for microswitch Keyboards only. It is not recommended for other Keyboards, particularly those using mechanical switches with ferrous components.

(1) Lift Keyboard from desk enclosure, and remove buttons to be cleaned separately (see key switch cleaning procedure).

(2) Immerse Keyboard in a solution of lukewarm water and mild dishwashing detergent.

(3) Use a soft brush to remove any stubborn deposits.

(4) Actuate plunger to remove gummy deposits from plunger bearing surface.

(5) Allow Keyboard to soak in the solution for approximately 1 hour. If solution becomes dirty, remove Keyboard and soak again in a fresh detergent solution, actuating plungers periodically while soaking.

(6) Rinse Keyboard in clean lukewarm water to remove all traces of detergent. Actuate plungers during rinsing.

(7) Air-dry thoroughly for approximately 6 hours before reassembling switches and placing Keyboard back in service. A low-pressure, oil-free air blast aids in removing excess water and promotes faster drying.

CAUTION

Solvent-type cleaners are NOT recommended because they attack the plastic parts of the switch. Clean Keyboard as soon as possible after accidental spills.

b. Key switch button cleaning. Clean the individual key switch buttons in the following manner:

(1) Remove the button from the Keyboard and clean it in a solution of lukewarm water and a mild dishwashing detergent.

(2) Rinse button in clean lukewarm water to remove detergent.

(3) Wipe outside surface of button dry to avoid water-spotting.

(4) Wait until inside mounting area is dry before reassembling button on Keyboard plungers. A low-pressure, oil-free air blast speeds the drying.

6.4 T/R FEATURE INTERFACE FAULT ISOLATION

A flowchart fault isolation analysis (Figure 6-18) keyed to diagnostic routine 20 is provided for fault isolation to the faulty functional area of the T/R Feature Interface board. Because of the simplicity of the board, the individual signals are checked for failure. Routine 20 exercises all areas of the board with the exception of the status drivers, which provide isolation between the Serial Printer and Terminal for status information being sent back to the Terminal. To check the status drivers it is necessary to have available a Serial Printer which may be connected to a Terminal with a resident operation program. Actual printer information must be outputted to the Serial Printer to check the printer status information being returned by the Serial Printer. In most instances, if all other functions check out, it is easier to replace the IC chips (Z16, Z17) used as status drivers rather than attempt to isolate the fault.

To use the flowchart, start with the instructions on the far left of sheet 1. Following the flow, sequentially answer the decision questions. Where an answer to a decision indicates a discrete faulty function, a list of possible IC's is provided, together with the signal at fault and a page reference to the T/R Feature Interface board schematics.

6.5 REFRESH STORAGE FAULT ISOLATION

A flowchart fault analysis (Figure 6-19) keyed to diagnostic routines 17 and 21 is provided for fault isolation to the faulty functional area on the Refresh Storage board. Finding the faulty component is relatively easy because the functions of the board are oriented to a sequential progression of events. Therefore a definite functional flow exists, allowing the use of half-splitting techniques early in fault isolation. More scoping of signals is necessary, however, due to the minimal availability of visual indications.

Two visual indications are available for use in fault isolation on the Refresh Storage board: the Monitor Display and the C Register used by the diagnostic routine. The C Register, under routine 21 control, provides a visual indication of the character data being stored and retrieved from Refresh Memory. This effectively splits the Refresh Storage board in half. By observing the Display Monitor for the proper display, as indicated on the flowchart, it is possible, in many instances, to determine the faulty functional area by the pattern or lack of pattern on the Display Monitor. Visual indications to be inspected and scoping information are supplied on the flowchart, placed wherever possible near the applicable decision.

To use the flowchart, start with the instructions on the far left of sheet 1. Following the flow, sequentially answer the decision questions. At the point where an answer to a decision indicates a discrete faulty function, a list of possible defective IC's is provided, together with the signal or signals at fault and a page reference to the Refresh Storage board schematics. The routine in use continues to exercise the board in the manner last indicated by the flowchart. This makes it possible to scope points specified by the flowchart and the schematics to locate the faulty component.

Table 6-3 T/R Feature Interface Board Signal Distribution

NAME	PAGE	JACK–PIN	BACK PANEL B	BACK PANEL A	TC	T/R	PRINTER	RFS
OSC	SH 1	J5-11						
REMOTE EN	SH 1	J5-167				J4-167		
LOCAL EN	SH 1	J5-170				J4-170		
REMOTE CLK	SH 1	J5-168				J4-168		
LOCAL CLK	SH 1	J5-169						
–PRTR RDY	SH 1	J5-45	J1-45	J7-7				
–CHECK	SH 1	J5-46	J1-46	J7-6				
–PPR OUT	SH 1	J5-47	J1-47	J7-5				
–CHAR RDY	SH 1	J5-37	J1-37	J7-8				
–CARR RDY	SH 1	J5-36	J1-36	J7-9				
–PPR FEED RDY	SH 1	J5-35	J1-35	J7-10				
PRTR RDY	SH 1	J5-43				J4-43		
CHECK	SH 1	J5-44				J4-44		
PPR OUT	SH 1	J5-42				J4-42		
CHAR RDY	SH 1	J5-40				J4-40		
CARR RDY	SH 1	J5-39				J4-39		
PPR FEED RDY	SH 1	J5-38				J4-38		
+5V	SH 1	J5-87-90						
GND	SH 1	J5-177-180						
BR 0-10	SH 2	J5-14-18, 25-29	J1-65-75		J2(65-75)			J3(65-75)
LD PRTR	SH 2	J5-24	J1 85		J2-85			
–RESET	SH 2	J5-91	J1-91		J2-91	J4-91		J3-91
–CHAR STROBE	SH 2	J5-70	J1-64		J2-64			
PPR FEED STROBE	SH 2	J5-64	J1-62		J2-62			
CARR STROBE	SH 2	J5-66	J1-175		J2-175			
RESTORE	SH 2	J5-68	J1-170		J2-170			
RIBBON LIFT	SH 2	J5-72	J1-169		J2-169			
PRTR SEL	SH 2	J5-75	J1-84		J2-84			
SET DTR	SH 2	J5-93	J1-93		J2-93	J4-93		
–SET TGL	SH 2	J5-13				J4-13		
–TGL	SH 2	J5-12				J4-12		
–SEL RDY LNS	SH 2	J5-73	J1-163	J7-35				
–PRTR SEL	SH 2	J5-74	J1-164	J7-36				
–RIBBON LIFT	SH 2	J5-71	J1-161	J7-34				
–RESTORE	SH 2	J5-67	J1-157	J7-2				
–CARR STROBE	SH 2	J5-65	J1-156	J7-3				
–PPR FEED STROBE	SH 2	J5-63	J1-154	J7-4				
–CHAR STROBE	SH 2	J5-69	J1-159	J7-1				
–DATA 1	SH 2	J5-104	J1-104	J7-11				
–DATA 2	SH 2	J5-103	J1-105	J7-12				
–DATA 4	SH 2	J5-101	J1-103	J7-13				
–DATA 8	SH 2	J5-106	J1-107	J7-14				
–DATA 16	SH 2	J5-105	J1-106	J7-15				
–DATA 32	SH 2	J5-100	J1-101	J7-16				
–DATA 64	SH 2	J5-102	J1-13	J7-17				
–DATA 128	SH 2	J5-23	J1-23	J7-50				
–DATA 256	SH 2	J5-22	J1-22	J7-49				
–DATA 512	SH 2	J5-21	J1-21	J7-48				
–DATA 1024	SH 2	J5-20	J1-20	J7-47				

Table 6-4 Refresh Storage Board Signal Distribution

NAME	PAGE	JACK–PIN	BACK PANEL A	TC	BACK PANEL B	POWER SUPPLY	T/R	MONITOR	T/R FEAT. INT.	PRINTER	KEYBOARD
			J13-3			POR	J4 4,5				
+12V	SH 1	J3 4,5	J1 3,5			TB2 3					
SER LD	SH 1	J3 21	J14 8	J2 21							
–12V	SH 1	J3 1,2									
–LN23	SH 1	J3 26		J2 26							
UC CLK	SH 1	J3 27		J2 27							
–HEX MODE	SH 1	J3 23		J2 23							
LH	SH 1	J3 24		J2-24							
LC CLK	SH 1	J3 25		J2 25							
SER CLK	SH 1	J3 20		J2 20							
–BLINK	SH 1	J3 19		J2 19							
–CC BLANK	SH 1	J3 18		J2 18							
–DATA BLANK	SH 1	J3 17		J2 17							
–ERR DIS	SH 1	J3 14		J2 14							
(CV+EV)	SH 1	J3 13		J2 13							
–LI	SH 1	J3 12		J2 12							
START AUDIO	SH 1	J3 76		J2 76							
INTENSITY	SH 1	J3 10	J11 1			R5 1&2					
MON VIDEO	SH 1	J3 8	J12 5					P19 8			
CONTRAST	SH 1	J3 9	J11 3			R6 1&2					
AUDIO OUT	SH 1	J3 7	J13 5			POR					
VA 0-10	SH 2	J3-155-165		J2(155-165)							
BR 0-10	SH 2	J3-65-75		J2(65-75)	J1(65-75)				J5 (14–18,25-29)		
–VA GT	SH 2	J3 63		J2 63							
REF	SH 2	J3 53		J2 53							
RA 0 (CC1)	SH 2	J3 57		J2 57							
–RS CS	SH 2	J3 54		J2 54							
–RESET CLK	SH 2	J3 55		J2 55							
–RS CLK	SH 2	J3 56		J2 56							
+22.5	SH 2	J3 172 175	J14 9			TB2 5				J7 4	J8 7
			J13 5			POR					
+5V VSS	SH 2	J3 84,85									
GND	SH 2	J3 177 180									
+5V	SH 2	J3 87 90		J2(87 90)	J1(87 90)						
RSDI 0-7	SH 3	J3 52 45		J2(52 45)							
RSDO STB	SH 3	J3 42		J2 42							
–WRITE EN	SH 3	J3 44		J2 44							
–READ EN	SH 3	J3 43		J2 43							
–CC HEX	SH 3	J3 38		J2 38							
LCM1	SH 3	J3 39		J2 39							
LCM2	SH 3	J3 40		J2 40							
–CGAL	SH 3	J3 41		J2 41							
RSDO 0-5	SH 3	J3 28 33		J2(28 33)							
RSPO	SH 3	J3 36		J2 36							
RSDO 6 & 7	SH 3	J3 34 35		J2 34&35							
–RSPE	SH 3	J3 37		J2 37							

6.6 T/R CONTROL BOARD FAULT ISOLATION

A flowchart fault analysis (Figure 6-20) keyed to diagnostic test routine 22 provides fault isolation to the faulty functional area of the T/R Control board. To test the T/R Control board, a routine was written which, in conjunction with a minor hardware change (one jumper), makes the Terminal act like a KAU sending General Poll messages to the T/R Control board. At the same time the Terminal also monitors the messages and displays the information on the Display Monitor, along with a longitudinal redundancy check value. In this manner the major functions of the T/R Control board are exercised, and a visual indication is provided which aids in isolating faults to a functional area. Before installing a T/R Control board for testing, ensure that the inverters on the ROM outputs are removed. The routine does not operate while they remain in the board. A combination of signal scoping and visual indications analysis is used in the flowchart fault analysis, and applicable scope and visual display information is included, placed whenever possible near the applicable decision box.

To use the flowchart, start with the instructions at the far left of sheet 1. Following the flow, sequentially answer the decision questions. At the point where an answer to a decision indicates a discrete faulty function, the bad function or functions are indicated, together with a list of possible defective IC's and the signal or signals at fault, with a page reference to the T/R Control board schematics. The routine continuously exercises the T/R Control board, making it possible to signal trace, using a scope or pulse catcher, to locate the faulty component.

6.7 TERMINAL CONTROL FAULT ISOLATION

A flowchart fault analysis (Figure 6-21), keyed to diagnostic routines 1 through 20, is used to fault isolate to a functional area of the Terminal Control board. Because the Terminal Control board is a highly complex logic board having many independent functions, various methods of trouble analysis are required to check the integrity of the board. To exercise and check for faults, a series of diagnostic routines using various indicators have been generated and applied to the flowchart to logically eliminate good functions and detect bad functions.

6.7.1 Terminal Control Diagnostic Routines

Two basic types of diagnostic routine are required to check the Terminal Control board. The first gives a visual indication, either at the FE Panel or on the Display Monitor, of the results of the test. The second repeatedly exercises various signals on the Terminal Control board, allowing the signals to be scoped for the proper waveform indication.

6.7.1.1 Visual Indication Routine

Most of the diagnostic routines used to test the Terminal Control board give some visual indication, and they may be divided into two classes. The one most widely used is the automatic trap routine, which exercises functional areas in a prescribed manner and then tests the function being exercised for a known condition. If the test succeeds, the routine proceeds to test a second function. If the test fails, the routine halts. On the FE Panel the HALT light is illuminated, and the Address register indicates the halt address where the routine failed. The second class of diagnostic routine runs continuously in a loop, and instructions in the routine provide indications by means of registers which may be monitored at the FE Panel, or by means of a defined character pattern displayed on the CRT. In both types of routine, the flowchart shows the results to be expected and what to do if the tests fail.

Table 6-5 T/R Control Board Signal Distribution

NAME	PAGE	JACK–PIN		T/R FEAT		RFS	BACK PANEL A	POWER SUPPLY	MONITOR	MODEM	KEYBOARD	PRINTER	KAU
RODO 0	SH 1	J4-77											
RODO 1	SH 1	J4-78	71-78		J2-78								
RODO 2	SH 1	J4-79	J1-78										
RODO 3	SH 1	J4-80	J1-78										
RODO 4	SH 1	J4-81	J1-78										
RODO 5	SH 1	J4-82	J1-78										
RODO 6	SH 1	J4-83	J1-83		J2-83								
RODO 7	SH 1	J4-86	J1-86		J2-86								
REM REQ TO SEND	SH 1	J4-53	J1-53	J5-143			J6-7			▬			
LOCAL REQ TO SEND	SH 1	J4-140	J1-51	J5-50			J9/J10-6						LOCAL
OSC	SH 1	J4-11	J1-11	J5-11	J2-11								
–ERR DISP	SH 1	J4-95	J1-94		J2-94								
TC CLOCK	SH 1	J4-154	J1-155		J2-154								
DATA TERM RDY	SH 1	J4-57	J1-56	J5-147			J6-5			▬			
SD+REM	SH 1	J4-167		J5-167									
–SD+REM	SH 1	J4-170		J5-170									
DATA CARRIER DETEC.	SH 1	J4-58	J1-57	J5-148			J6-4			▬			
LOCAL CLK	SH 1	J4-169		J5-169									
+12V	SH 1	J4-04-05	J1-03&05			33-04&05	J13-3	TB2-3					
							J14-8	POR					
REMOTE CLR TO SEND	SH 1	J4-56	J1-55	J5-146			J6-6			▬			
LOCAL CLR TO SEND	SH 1	J4-55	J1-48	J5-48			J10/J9-7						LOCAL
–REMOTE GATE	SH 1	J4-60	J1-59	J5-150			J6-3			▬			
DATA SET RDY	SH 1	J4-61	J1-60	J5-151			J6-2			▬			
REM VIEW	SH 1	J4-85											
REMOTE CLK	SH 1	J4-168		J5-168									
ROA 0-11	SH 1	J4-142-153	J1-142-153		J2 142-153								
ROA 12	SH 1	J4-172	J1-172		J2-172								
–ADD READ	SH 1	J4-173	J1-173		J2-173		J13-2	POR					
+5V	SH 1	J4-87-90	J1-87-90	J5 87-90	J2 87-90	J3-87-90	J8 4&5				PI-F&H		
GND	SH 1	H4-177-180	J1-177-180	J5 177-180			J14-1,2,3	TB2-2					
TRANS RDY	SH 2	J4-137	J1-137		J2-137								
–MR	SH 2	J4-91	J1-91	J5-91	J2-91	J3-91							
–VIEWER MODE	SH 2	J4-96	J1-4		J2-4								
–TLR T/R SR	SH 2	J4-138	J1-138		J2-138								
–TRANS EN	SH 2	J4-121	J1-31		J2-121								
SET DTP	SH 2	J4-92	J1-92		J2-92								
–SET TGL	SH 2	J4-183		J5-13									
–AC8	SH 2	J4-176	J1-176		J2-176								
–SET	SH 2	J4-13		J5-12									
AC4	SH 2	J4-171	J1-171		J2-171								
SET DTR	SH 2	J4-93	J1-93	J5-93	J2-93								
–LDTR	SH 3	J4-122	J1-381		32-122								
–CLR DR	SH 3	J4-122	J1-32		J2-122								
TRI-TR8	SH 3	J4-112-119	J1-112-119	J5 107-109 114-118	J2 112-119								
RECEIVED DATA	SH 3	J4-62	J1-61	J5-152			J6-1			▬			
+TP	SH 3	J4-30	J1-120				J9/J10-1						LOCAL
–TP	SH 3	J4-31	J1-121				J9/J10-2						LOCAL
–12V	SH 3	J4-01-02	J1-1&2			J3 1&2	J14-7,J8-9	TB2-4			PI-I		
TRANSMITTED DATA	SH 3	J4-49	J1-50	J5-139			J6-8			▬			
REC MS 1	SH 3	J4-123	J1-33		J2-123								
REC MS 2	SH 3	J4-124	J1-34		J20124								
CLR LRC	SH 3	J4-10	J1-10		J2-10								
LRC ON	SH 3	J4-08	J1-08		J2-8								
LRC OFF	SH 3	J4-09	J1-09		J2-9								
CHAR RDY	SH 3	J4-40		J5-40									
PPR OUT	SH 3	J4-42		J5-42									
PPR FEED RDY	SH 3	J4-38		J5-38									
CARR RDY	SH 3	J4-39		J5-39									
TA 4	SH 3	J4-34	J1-124		S1-D2								
PRTR RDY	SH 3	J4-43		J5-43									
CHECK	SH 3	J4-44		J5-44									
TA 2	SH 3	J4-33	J1-123		SI-D3								
TA 1	SH 3	J4-32	J1-122		SI-D4								
RM 0-9	SH 3	J4-125-134	J1-125-134		J2 125-134								
RM 11-12	SH 3	J4-135-136	J1-135-136		J2 135-136								

Table 6-6 Terminal Control Board Signal Distribution

NAME	PAGE	JACK–PIN	BACK PANEL A	BACK PANEL B	RFS	T/R CONTROL	T/R FEAT	KEY-BOARD	PRINTER	FE PANEL	MONITOR	PS
OSC	SH 2	J2-11		J1-11								
CC8	SH 2	J2-60			J3-60							
CC4	SH 2	J2-59			J3-59							
CC2	SH 2	J2-58			J3-58							
CC1	SH 2	J2-57			J3-57							
RSDO7	SH 3	J2-35			J3-35							
–LN23	SH 3	J2-26			J3-26							
–LN24	SH 3	J2-7	J12-4								P19-9	
–LC INH	SH 3	J2-16			J3-16							
RSDO3	SH 3	J2-31			J3-31							
RSDO4	SH 3	J2-32			J3-32							
RSDO6	SH 3	J2-34			J3-34							
R/W STG INITIATE	SH 4	J2-141										
VA0-10	SH 4	J2-155-165			J3-(155-165)							
RESET CLK	SH 4	J2-55			J3-55							
RS CLK	SH 4	J2-56			J3-56							
RSDO STB	SH 4	J2-42			J3-42							
–CGAL	SH 4	J2-41			J3-41							
–WRITE EN	SH 4	J2-44			J3-44							
–VA GT	SH 4	J2-63			J3-63							
REF	SH 4	J2-53			J3-53							
–READ EN	SH 4	J2-43			J3-43							
RSDO2	SH 6	J2-30			J3-30							
RSDO1	SH 6	J2-29			J3-29							
SYSTEM SHIFT	SH 6	J2-111	J8-3					P1-D				
–BLINK	SH 6	J2-19			J3-19							
–LI	SH 6	J2-12			J3-12							
–CUR EN	SH 6	J2-15			J3-15							
–DATA BLANK	SH 6	J2-17			J3-17							
–CC BLANK	SH 6	J2-18			J3-18							
–HEX SW	SH 7	C38										
–CC HEX SW	SH 7	C35										
LCM2	SH 7	J2-40			J3-40							
–CC HEX	SH 7	J2-38			J3-38							
–HEX MODE	SH 7	J2-23			J3-23							
LH	SH 7	J2-24			J3-24							
LCM1	SH 7	J2-39			J3-39							
LC CLK	SH 7	J2-25			J3-25							
UC CLK	SH 7	J2-27			J3-27							
SER LD	SH 7	J2-21			J3-21							
–H	SH 7	J2-6	J12-3								P19-7	
SER CLK	SH 7	J2-20			J3-20							
TC CLK	SH 8	J2-154		J1-155		J4-154						
–POR	SH 8	J2-2	J13-8									POR
–ADD READ	SH 8	J2-173		J1-173		J4-173						
–RSTI	SH 8	J2-91		J1-91	J3-91	J4-91	J5-91					
R/W STG PWR CLR	SH 8	J2-140										
READ/WRITE	SH 8	J2-139										
–SC SW	SH 8	C 32										
STORE SW	SH 8	C 34										
–PS STOP	SH 8	C 3										
–STOP SW	SH 8	C 36										
–MR	SH 8	C 28										
STOPPED	SH 8	C 22										
–REP RC	SH 9	C 33										
–LOAD SW	SH 9	C 30										
AS 0	SH 9	C 5										
AS 1	SH 9	C 7										
AS 2	SH 9	C 9										
AS 3	SH 9	C 11										
AS 4	SH 9	C 13										
AS 5	SH 9	C 15										
AS 6	SH 9	C 17										
AS 7	SH 9	C 19										
AS 8	SH 9	C 21										
AS 9	SH 9	C 23										
AS10	SH 9	C 25										
AS 11	SH 9	C 27										
ROA 0	SH 9	C 8/J2-142		J1-142		J4-142						
ROA 1	SH 9	C 6/J2-143		J1-142		J4-142						
ROA 2	SH 9	C 4/J2-144		J1-142		J4-142						
ROA 3	SH 9	C 2/J2-145		J1-142		J4-142						
ROA 4	SH 9	C 1/J2-146		J1-142		J4-142						
ROA 5	SH 9	C 3/J2-147		J1-142		J4-142						
ROA 6	SH 9	C 20/J2-148		J1-142		J4-142						
ROA 7	SH 9	C18/J2-149		J1-149		J4-149						
ROA 8	SH 9	C16/J2-150		J1-150		J4-150						
ROA 9	SH 9	C14/J2-151		J1-150		J4-150						
ROA 10	SH 9	C12/J2-152		J1-150		J4-150						
ROA 11	SH 9	C10/J2-153		J1-153		J4-153						
ROA 12	SH 9	C26/J2-172		J1-172		J4-172						
–ROA 12	SH 9	J2-5										

Table 6-6 Terminal Control Board Signal Distribution (continued)

NAME	PAGE	JACK–PIN	BACK PANEL A	BACK PANEL B	RFS	T/R CONTROL	T/R FEAT	KEY-BOARD	PRINTER	FE PANEL	PS
RSDO 0	SH 10	J2 28			J3 28						
RSDO 5	SH 10	J2 33			J3 33						
RSPO	SH 10	J2 36			J3 36						
RSPE	SH 10	J2 37			J3 37						
RM0	SH 10	J2 125		J1 125		J4 125					
RM1	SH 10	J2 126		J1 125		J4 125					
RM2	SH 10	J2 127		J1 125		J4 125					
RM3	SH 10	J2 128		J1 125		J4 125					
RM4	SH 10	J2 129		J1 125		J4 125					
RM5	SH 10	J2 130		J1 125		J4 125					
RM6	SH 10	J2 131		J1 125		J4 125					
RM7	SH 10	J2 132		J1 125		J4 125					
RM8	SH 10	J2 133		J1 125		J4 125					
RM9	SH 10	J2 134		J1 125		J4 125					
RM10	SH 10	J2 135		J1 125		J4 125					
RM11	SH 10	J2 136		J1 136		J4 136					
KS0	SH 10	J2 101	J8 18					P1 11			
KS1	SH 10	J2 102	J8 17					P1 10			
KS2	SH 10	J2 103	J8 16					P1 9			
KS3	SH 10	J2 104	J8 15					P1 8			
KS4	SH 10	J2 105	J8 14					P1 7			
KS5	SH 10	J2 106	J8 13					P1 6			
KS6	SH 10	J2 107	J8 12					P1 5			
KS7	SH 10	J2 108	J8 11					P1 4			
KS8	SH 10	J2 109	J8 10					P1 3			
KB SHIFT	SH 10	J2 110	J8 1					P1 B			
BR0 10	SH 12	J2 65 75		J1(65 75)	J3(65 75)		J5(14 19) (25 29)				
DM LED	SH 12	J2 98									
START AUDIO	SH 13	J2 76			J3 76						
–TRANS EN	SH 13	J2 120		J1 30		J4 30					
–CLR T/R SR	SH 13	J2 138		J1 138		J4 138					
–LD TR	SH 13	J2 121		J1 31		J4 31					
–CLR DR	SH 13	J2 122		J1 32		J4 32					
–B1 SEL	SH 14	C 29								⊟	
–B0 SEL	SH 14	C 31								⊟	
REG 0	SH 14	C 37								⊟	
REG 1	SH 14	C 39								⊟	
REG 2	SH 14	C 41								⊟	
REG 3	SH 14	C 43								⊟	
REG 4	SH 14	C 45								⊟	
REG 5	SH 14	C 47								⊟	
REG 6	SH 14	C 49								⊟	
REG 7	SH 14	C 50								⊟	
REG 8	SH 14	C 48								⊟	
REG 9	SH 14	C 46								⊟	
REG 10	SH 14	C 44								⊟	
REG 11	SH 14	C 42								⊟	
ROM TEST	SH 15	J2 22									
TRANS RDY	SH 15	J2 137		J1 137		J4 137					
–SET DTP	SH 16	J2 92		J1 92		J4 92					
–SET DTR	SH 16	J2 93		J1 93		J4 93	J5 93				
KB STB	SH 16	J2 95	J8 19					P1 12			
DELETE LED	SH 16	J2 99									
INSERT LED	SH 16	J2 100									
ERR LED	SH 16	J2 96	J13 6								POR
LD PRTR	SH 16	J2 85		J1 85			J5 24				
TRANS	SH 16	J2 97									
–VIEW MODE	SH 16	J2 4	J1 4			J4 96					
CC STORE	SH 16	C								⊟	
–VIEW SW	SH 16	C24								⊟	
RSDI0-7	SH 17	J2(52 45)			J3(52 45)						
RODO 0-7	SH 19	J2 (77 83,86)	J1(77 83) (86)			J4(77 83) (86)					
–ERR DIS	SH 20	J2 94		J1 94		J4 95					
–AC8	SH 20	J2 176		J1 176		J4 176					
AC4	SH 20	J2 171		J1 171		J4 171					
PRTR SEL	SH 20	J2 84		J1 84			J5 75				
–(CV+EV)	SH 20	J2 13			J3 13						
–ERR DIS	SH 20	J2 14			J3 14						
RESTORE	SH 20	J2 170		J1 170			J5 68				
CHAR STROBE	SH 20	J2 64		J1 64			J5 70				
CARR STROBE	SH 20	J2 175		J1 175			J5 66				
PPR FEED STROBE	SH 20	J2 62		J1 62			J5 64				
RIBBON LIFT	SH 20	J2 169		J1 169			J5 72				
REC MS1	SH 20	J2 123		J1 33		J4 123	J4 123				
REC MS2	SH 20	J2 124		J1 34		J4 124	J4 124				
LRC ON	SH 20	J2 8		J1 8		J4 8					
LRC OFF	SH 20	J2 9		J1 9		J4 9					
CLR LRC	SH 20	J2 10		J1 10		J4 10					
TR1-TR8	SH 21	J2 112-119		J1(112-119)		J4(112-119)	J5(107-109, 114,118)				

6.7.1.2 Exerciser Routines

Exerciser routines exercise various functions of the Terminal Control board which cannot be checked by visual indications. The exercisers are loop routines which allow scoping of the signals associated with the function being tested to determine the faulty area. All signal waveforms required are either on the flowchart, or in the manual as timing diagrams. An example of an exerciser is routine 20. This routine exercises all the printer functions generated by the Terminal Control board. Because there are no visual indications, these functions must be individually scoped when it is believed that a possible faulty printer function exists.

6.7.2 Flowchart Organization

The Terminal Control board flowchart fault analysis is organized to check out sequentially all functions of the Terminal Control board. The most basic functions, which control overall operation of the board, are checked first. Next to be checked are the registers and logic required to execute the instructions, compares and tests. This sequence is continued, creating a pyramid effect. The approach is to ensure the integrity of the hardware required to perform the next routine, in a previously performed routine, thus guaranteeing that a detected failure is the result of the hardware involved in performing the function, and not in the hardware required to check the function. Each function is keyed on the flowchart with a circled number and an associated title. The flowchart comprises instruction, decision and result boxes. The instruction boxes indicate what is to be done to check the function. The decision boxes are based on visual indicators or scope indications and show the direction to be taken in further defining the faulty functions. The result boxes provide a list of defective functions and associated components, together with page references to the Terminal Control board schematics for further fault isolation. A result box is the last box in a flow, and it indicates that, as a result of the decision leading to the box, a discrete functional area was determined faulty.

6.7.3 Flowchart Use

To use a flowchart, start with the instructions on the far left of sheet 1. Following the flow, sequentially answer the decision questions. At the point where an answer to a decision indicates a discrete faulty function, the function or functions are shown, together with a list of possible defective IC's. Where applicable, the signal or signals involved in the faulty functional area are included. Page references to the schematics are also provided. The routine in use continues to exercise the board in the manner last indicated by the flowchart. In the case of a trap routine, some modification is necessary. This makes it possible to scope points specified by the flowchart and schematic to locate the faulty component.

6.8 5201 POWER SUPPLY

The procedures for fault isolation and adjustment of the 5201 Power Supply (P N 2010-5035-1) and POR board (P N 2010-5001) are contained in the following paragraphs. Problems and possible faults are listed in Table 6-7, together with adjustment and replacement procedures.

6.8.1 Troubleshooting

Troubleshooting the Power Supply is limited to identifying the problem, referring to Table 6-7 for a similar problem, and then checking the possible faults.

6.8.2 Adjustments

The four regulated power supplies can be adjusted by V ADJ resistor R8 on each supply. Measure the output of each supply and make adjustments as necessary so that the outputs of the supplies are: +5.4VDC at TB2-1, +12VDC at TB2-3, −12VDC at TB2-4 and +22.5VDC at TB2-5. The current should not require adjustment; however, R6 adjusts the current for each supply as follows: +5 volt supply to 12 amperes, ±12 volt supplies to 6 amperes, and +22.5 volt supply to 4.5 amperes.

Table 6-7 Common Power Supply Problems

Problem	Possible Fault
NO POWER ON indication	a. POWER ON/OFF BRIGHTNESS switch off. b. Circuit breaker, CB1, open. c. AC interlock, S1, open. d. AC power cord disconnected. e. +5VDC supply defective; check TB2 pin 1. f. Power on LED, D2 defective.
DC Power malfunction	a. Defective power supply: Check voltages at TB2 pin 1 (+5.4VDC), pin 3 (+12VDC). pin 4 (−12VDC), pin 5 (+22.5VDC). b. Improperly adjusted supply: Adjust supply by V ADJ (R8) on regulated supply. c. Improper AC input to supply: Check T1 jumpers correct. d. Check cable and Terminal board connections.
No Power On Reset	a. No +12VDC: Check +12VDC supply at TB2-3. b. Defective relay K1, K1 not closing: Check for ground at P13-8 at power on. c. Defective Q-1: Check for ground at Q1-collector at power on.
No Audio Tone	a. Check volume adjustment R7. b. No audio in pulse: Check P13-5 with command entry error. c. Defective IC Z01: Check for low output at Z01 with command entry error.
No ERROR Indication	a. No error input: Check for ground at P13-6 with command entry error. b. Faulty +5VDC supply: Check +5.4 VDC at TB2 pin 1.

6.8.3 Component Replacement

With the Terminal cover removed, the four regulated power supplies, POR board, and other components are accessible (see Figure 6-7). Each of the regulated supplies can be replaced by removing four nuts (one each corner), that secure the supply to the chassis, and unsoldering the input and output leads. The POR board and its components can be replaced by removing the four screws securing the board to the chassis and then unsoldering the leads or components, using a solder sucker and low-wattage iron.

CAUTION

Excess heat can damage the PC board.

APPENDIX A
DIAGNOSTIC ROUTINES

A.0 GENERAL

Twenty-two diagnostic routines have been developed to provide a method for exercising and fault isolating functional areas of the 5201 Terminal. These routines provide a predictable sequence of events which ensure exercising all functions of the Terminal. Routines 1 through 20 exercise the Terminal Control board, Routine 21 exercises the Refresh Storage board, and Routine 22 exercises the T/R Control board. Because of the structure of certain routines, they can be used for testing more than one board. Routine 20, besides exercising all the Printer Control functions of the Terminal Control board, is structured to exercise the T/R Feature Interface board as well. Routine 17 (Keystroke Gathering Routine), in addition to exercising the keystroke processing functions of Terminal Control, is structured to exercise the Keyboard, Refresh Storage board and Display Monitor.

A.1 ROUTINE ORGANIZATION

The diagnostic routines are organized in a logical sequence to ensure that hardware being used to test a new functional area has itself been tested in a previous routine. This sequence applies basically to all the logic boards, but is more critical in checking the complex Terminal Control (Processor) board, where visual indications and automatic traps are more widely used. All routines are set up to loop on themselves. This allows the Field Engineer to use scoping and half-splitting techniques to isolate a fault to a specific component, once the faulty functional area has been located. Some minor program modification is required when troubleshooting a faulty functional area which was located using a diagnostic routine with halt traps for fault isolation. When a trap condition is encountered and it is desired to run the routine in a loop to enable use of a scope for faulty component isolation, proceed as follows: Examine the address of the halt and then, referring to the appropriate routine listing, replace the X'00' halt instruction associated with the fault with the same instruction as the one following the halt. This eliminates the halt condition and allows the routine to continue to its end, at which time it loops back.

CAUTION

Once the fault has been located and the routine is no longer required, ensure the correct instruction is again loaded into the routine. If not, the next time this area is faulty, no indication will be observed because the trap has been eliminated.

A.2 DIAGNOSTIC ROUTINE FAULT INDICATION

Three types of fault indication are used, the most frequent being the HALT light located on the FE Panel. Compare and test instructions are used to test known conditions which, if good, create a branch to the next function to be tested. If the test condition is not met an X'00' halt instruction halts the routine at the point where the failure occurred and causes the HALT light to be illuminated. By observing the address displayed on the Instruction Register Address lights and referring to the listing, it is possible to determine which function was being tested when the failure occurred.

Another frequently used indication is the FE Panel Register Indicators, which provide monitoring of the A Register, B Register, C Register and Instruction Register outputs at the FE Panel. By exercising a known sequence of conditions for a functional area, one of the four registers can be observed for a resultant good indication. If a good indication is not obtained, the functional area being exercised is faulty. The third type of visual indication is the Display Monitor. This lends itself particularly well to checking the Screen Display and Refresh Timing and Control logic located on the Terminal Control board. It is also used extensively in fault isolation of the Refresh Storage and the T/R Control board.

A.3 TYPES OF ROUTINE

There are two basic types of diagnostic routine. The commoner type gives some visual indication, either at the FE Panel or on the Display Monitor, which makes it possible to determine the faulty functional area. With this type it may be necessary only to place the FE Panel in the RUN condition and observe a visual indication, or it may be necessary to single-cycle through the program, observing a sequence of visual indications. The second type of diagnostic routine is strictly an exerciser. This type provides no fault indication, but rather exercises a certain functional area, which allows monitoring of the function with a scope to determine if a fault exists. Scope points and associated indications when using this type of routine are provided in the flow chart trouble analysis.

A.4 DIAGNOSTIC ROUTINE OPERATION PROCEDURE

The following procedure for use of the diagnostic routines should be followed, with certain modifications as specified in the comments listed with each routine and on the flowcharts.

Step 1 At the FE Panel, place RUN/STOP to STOP.

Step 2 At the FE Panel, depress MASTER RESET (MR). Once Routine 21 has begun, do not depress MR again unless it is desired to begin the routine from the start address.

Step 3 At the FE Panel, place the ROM Address switches to the hex value of the start address for the routine to be performed.

Step 4 At the FE Panel, ensure HEX mode is off.

Step 5 At the FE Panel, depress LOAD. The start address value should appear on the ROM Address lights. If not, repeat from step 1.

Step 6 At the FE Panel, place the RUN/STOP switch to the position specified in the comments or on the flowcharts.

Step 7 At the FE Panel, place the REGISTER SELECT switches to the desired position for selecting register data as indicated on the FE Panel. This step is necessary only when the comments or the flowcharts indicate monitoring a register output for a fault condition.

Step 8 Make the desired observation as indicated by the comments and flowcharts.

When using a trap routine that must be modified for looping, perform the following:

Step 1 At the FE Panel, place RUN/STOP to STOP.

Step 2 At the FE Panel, depress MASTER RESET (MR).

Step 3 At the FE Panel, place the ROM Address switches to the hex value of the address where the instruction is to be changed.

Step 4 At the FE Panel, ensure HEX mode switch is in the OFF position.

Step 5 At the FE Panel, depress LOAD and observe the register lights for the correct address.

Step 6 At the FE Panel, place the ROM DATA switches to the value of the instruction to be loaded.

Step 7 At the FE Panel, depress STORE and observe the Instruction register for the correct value. If not present, repeat from step 1.

Step 8 Place RUN/STOP to RUN and perform the required fault isolation.

Step 9 Repeat steps 1 through 7 to reload the correct instruction back into the routine. This should probably be an X'00' instruction (HALT).

A.5 ROUTINE LISTINGS ORGANIZATION STRUCTURE

As an example, refer to Diagnostic Routine 9. The first column of double-digit numbers is the absolute address (always starting at address 00). The hex numbers in parentheses are the relative address locations. These are the addresses of the instructions if all the routines are loaded into storage sequentially, starting at address 00. The next column of double-digit numbers is the hex codes of the operational instruction (refer to Table A-1). The numbers in parentheses next to some instruction codes are the hex code that must be used for the instruction when relative addressing is used. To the right of the op codes are descriptions of the function being performed by a group of instructions. To the far right, under Comments, is a description of the routine, the indications to be expected, and information on how to use the routine.

1. Halt Test Routine

Address	Op Code
00	41
01	61
02	51
03	00

Comments

Select starting address from FE Panel. Place RUN/STOP to RUN. If the Halt function is working, the program halts and displays ROM address 04 on the FE Panel.

2. Unconditional Branch Routine

Address	Op Code
00 (04)	90
01 (05)	00 (04)
02 (06)	00
03 (07)	00

Comments

Select starting address from FE Panel. Place RUN/STOP to RUN. Program loops and address lights are dimly lighted. If not working, halt lights, and address indicates a strong 04.

3. Increment A Register Routine

Address	Op Code
00 (08)	44
01 (09)	90
02 (0A)	00 (08)
03 (0B)	00
04 (0C)	00

Comments

1. Select start address from FE Panel.
2. Place RUN/STOP to RUN.
3. Select A Register for monitoring.
4. On FE Panel, REG's 0-10 are dimly lighted and bit 11 is flickering. If run in STOP and SINGLE CYCLE is repeatedly depressed, REG lights increment.

4. Decrement A Register Routine

Address	Op Code
00 (0D)	48
01 (0E)	90
02 (0F)	(0D) 00
03 (10)	00
04 (11)	00

Comments

1. Select start address from FE Panel.
2. Place RUN/STOP to RUN.
3. Select A Register for monitoring.
4. On FE Panel REG's 0-10 dimly lighted and 11 11 is flickering. If in STOP and SINGLE CYCLE is repeatedly depressed, REG decrements.

5. Increment B Register Routine

Address	Op Code
00 (12)	54
01 (13)	90
02 (14)	(12) 00
03 (15)	00
04 (16	00

Comments

1. Select start address from FE Panel.
2. Place RUN/STOP to RUN.
3. Select B Register for monitoring.
4. On FE Panel REG's 0-10 are dimly lighted and 11 is flickering. If in STOP and SINGLE CYCLE is repeatedly depressed, REG increments.

6. Decrement B Register Routine

Address	Op Code
00 (17)	58
01 (18)	90
02 (19)	(17) 00
03 (1A)	00
04 (1B)	00

Comments

1. Select start address from FE Panel.
2. Place RUN/STOP to RUN.
3. Select B Register for monitoring.
4. On FE Panel REG's 0-10 are dimly lighted and 11 is flickering. If in STOP and SINGLE CYCLE is repeatedly depressed, REG decrements.

7. Increment C Register Routine

Address	Op Code
00 (1C)	64
01 (1D)	90
02 (1E)	(1C) 00
03 (1F)	00
04 (20)	00

Comments

1. Select start address from FE Panel.
2. Place RUN/STOP to RUN.
3. Select C Register for monitoring.
4. On FE Panel REG's 0-11 are dimly lighted. If in STOP and SINGLE CYCLE is repeatedly depressed, REG increments in BCD code.

NOTE: A and B REG's are binary counters, C REG is a decade counter.

8. Decrement C Register Routine

Address	Op Code
00 (21)	68
01 (22)	90
02 (23)	(21) 00
03 (24)	00
04 (25)	00

Comments

1. Select start address from FE Panel.
2. Place RUN/STOP to RUN.
3. Select C Register for monitoring.
4. On FE Panel REG's 0-11 are dimly lighted. If in STOP and SINGLE CYCLE is repeatedly depressed, REG decrements in BCD code. (See NOTE on Routine 7.)

9. Load A/C Reg with Constants Routine

Address	Op Code	
00 (26)	9E	
01 (27)	FF	Loads constants
02 (28)	FF	FFF in A and C REGS
03 (29)	9F	
04 (2A)	FF	
05 (2B)	FF	
06 (2C)	D0	
07 (2D)	A6	
08 (2E)	91	Checks comp branch
09 (2F)	(32) 0C	
0A (30	00	
0B (31)	00	
0C (32)	41	
0D (33)	A6	
0E (34)	99	
0F (35)	(26) 00	Checks comp branch
10 (36	00	
11 (37)	00	

Comments

This is an automatic multifunction routine.

1. Select starting address from FE Panel.
2. Place RUN/STOP to RUN.
3. Observe A register. REG's 0-11 dimly lighted.
4. Observe C Register. REG's 0-11 solid light.
5. If program halts, run in single-cycle mode and observe A Register and C Register. Both should be all 1's until address '0C', when A Register is cleared for the the noncompare test. At '0C', A Register is all 0's. If A and C REG's appear as above, observe ROM address where program halted.

   '0C' halt = bad compare logic, A MUX 1 compare path, compare branch.

   '11' halt = bad noncompare branch logic, compare latch.

10. A, B, C, JDR, FDR and Compare Routine

Address	Op Code	Comments
00 (38)	9E	This is a multifunction automatic test. If a fault is detected the program halts and the ROM address lights indicate the area of the fault.
01 (39)	AA	
02 (3A)	AA	
03 (3B)	D0	
04 (3C)	30	
05 (3D)	08	Compares
06 (3E)	52	C, REG to A REG
07 (3F)	7C	10 halt indicates (examine C REG and A REG values):
08 (40	7D	1. RAM Address Select 30 fault.
09 (41)	62	2. RAM 0 fault.
0A (42)	A6	3. RAM C REG fault (A/C MUX, C load).
0B (43)	91	
0C (44	(47) 0F	
0D (45)	00	
0E (46)	00	
0F (47)	00	
10 (48)	A0	
11 (49)	91	
12 (4A)	(51) 19	
13 (4B)	00	Compares
14 (4C)	00	C REG (0-3) = A REG (0-3)
15 (4D)	A1	'15' halt indicates bad compare logic.
17 (4F)	00	
18 (50)	00	
19 (51)	A2	Compares C REG (4-7 = A REG (4-7)
1A (52)	91	'1E' halt indicates bad compare logic.
1B (53)	(56) 1E	
1C (54)	00	
1D (55)	00	
1E (56)	D1	Compares B reg = C Reg
1F (57	A6	'24' halt indicates:
20 (58)	91	1. Bad B REG (examine B REG for FFF at FE Panel).
(21) (59)	(5C) 24	2. Bad RAM B REG path
22 (5A)	00	3. Bad B REG MUX 1 path

10. A, B, C, JDR, FDR and Compare Routine (continued)

Address	Op Code	
23 (5B)	00	
24 (5C)	D2	
25 (5D)	A6	Compares JDR = C REG
26 (5E)	91	'2A' halt indicates:
27 (5F)	(62) 2A	1. Bad job definition register.
28 (60)	00	2. Bad RAM JDR path.
29 (61)	00	3. Bad JDR MUX 1 path.
2A (62)	D3	
2B (63)	A6	Compares FDR = C REG
2C (64)	91	'30' halt indicates:
2D (65)	(68) 30	1. Bad field definition register.
2E (66)	00	2. Bad RAM FDR path.
2F (67)	00	3. Bad FDR MUX 1 path.
30 (68)	D0	
31 (69)	41	Compares C > A REG
32 (6A)	A5	'37' halt indicates bad compare logic.
33 (6B)	91	
34 (6C)	(6F) 37	
35 (6D)	00	
36 (6E)	00	
37 (6F)	61	
38 (70)	42	Compares C < A REG
39 (71)	A4	'3E' halt indicates bad compare logic.
3A (72)	91	
3B (73)	(76) 3E	
3C (74)	00	
3D (75)	00	
3E (76)	9E	
3F (77)	55	
40 (78)	55	
41 (79)	90	
42 (7A)	(3B) 03	
43 (7B)	00	
44 (7C)	00	

11. Test RAM's Routine

Address	Op Code	
00 (7D)	9E	
01 (7E)	FF	
02 (7F)	FF	Checks RAM 1 and
03 (80)	D0	RAM enable
04 (81)	31	function.
05 (82)	07	Improper indi-
06 (83)	62	cation Bad RAM 1
07 (84)	32	or enable logic.
08 (85)	08	
09 (86)	62	Checks RAM 2.
0A (87)	33	
0B (88)	08	Checks RAM 3.
0C (89)	62	
0D (8A)	34	
0E (8B)	08	Checks RAM 4.
0F (8C)	62	
10 (8D)	35	
11 (8E)	08	Checks RAM 5.
12 (8F)	62	
13 (90)	36	
14 (91)	08	Checks RAM 6.
15 (92)	62	
16 (93)	37	
17 (94)	08	Checks RAM 7.
18 (95)	62	
19 (96)	38	
1A (97)	08	Checks RAM 8.
1B (98)	62	
1C (99)	39	
1D (9A)	08	Checks RAM 9.
1E (9B)	62	
1F (9C)	3A	
20 (9D)	08	Checks RAM 10.
21 (9E)	62	

Comments

This is a multifunction test; its purpose is to check all RAM locations. To be effective, this test must be run single-step, as there are no automatic traps. A logic probe should be used to monitor the RAM address selection latch (N12, page 10 of 21), as only the storage is checked by this test.

1. Select start address from FE Panel.
2. Place RUN/STOP to STOP.
3. Select C REG for monitoring.
4. Single-cycle routine and monitor C REG and logic probe. Sequence on probe: A2, A7, D7, D2. Display binary count A2 = LSB, D2 = MSB as program is sequenced.

   C REG for all addresses = all 1's until A2, A7, D7, D2 are all active, after which time address branches to either '03' or '00'. For the next program cycle C REG is all 0's. The C REG value alternates each program cycle.

11. Test RAM's Routine (continued)

Address	Op Code	
22 (9F)	3B	
23 (A0)	08	Checks RAM 11.
24 (A1)	62	
25 (A2)	3C	
26 (A3)	08	Checks RAM 12.
27 (A4)	62	
28 (A5)	3D	
29 (A6)	08	Checks RAM 13.
2A (A7)	62	
2B (A8)	3E	
2C (A9)	08	Checks RAM 14.
2D (AA)	62	
2E (AB)	3F	
2F (AC)	08	Checks RAM 15.
30 (AD)	62	
31 (AE)	9E	
32 (AF)	00	
33 (B0)	00	
34 (B1)	A6	
35 (B2)	91	Loads alternate
36 (B3)	(7D) 00	bit pattern in RAM's.
37 (B4)	00	
38 (B5)	90	
39 (B6)	(7D) 00	
3A (B7)	00	

12. Set and Clear A Register Bits Routine

Address	Op Code	
00 (B8)	00	
01 (B9)	41	
02 (BA)	30	
03 (BB)	08	
04 (BC)	20	
05 (BD)	08	
06 (BE)	21	
07 (BF)	08	
08 (C0)	22	
09 (C1)	08	
0A (C2)	23	
0B (C3)	08	
0C (C4)	24	Sequentially
0D (C5)	08	sets A
0E (C6)	25	Register bits.
0F (C7)	08	
10 (C8)	26	
11 (C9)	08	
12 (CA)	27	
13 (CB)	08	
14 (CC)	28	
15 (CD)	08	
16 (CE)	29	
17 (CF)	08	
18 (D0)	2A	
19 (D1)	08	
1A (D2)	2B	
1B (D3)	9F	
1C (D4)	FF	
1D (D5)	FF	Testing for
1E (D6)	A6	a fault.
1F (D7)	91	
20 (D8)	23	
21 (D9)	00	'23' halt indicates bad A/C MUX.
22 (DA)	00	

Comments

This is a multifunction automatic test. It sequentially sets each bit of the A Register and then checks if the A Register is all on E's after setting bit 11. If not, it halts, indicating one of the bits did not set. If the check is a good indication, it then clears sequentially all bits of the A REG. After bit 11 is cleared it checks A reg for all 0's. If the check is good, it branches back to the beginning. If a halt occurs, observe A REG for bad bit.

Test Sequence:

1. Select start address at FE Panel.
2. Place RUN/STOP to RUN.
3. Observe FE Panel for a halt condition (halt fault).
4. If halt occurs, monitor A REG for faulty bit.

NOTE: A Register may be monitored and the program run in single-cycle mode if actual observation of A Register bits setting and clearing is desired.

12. Set and Clear Register Bits Routine (continued)

Address	Op Code	
23 (DB)	08	
24 (DC)	10	
25 (DD)	08	
26 (DE)	11	
27 (DF)	08	
28 (E0)	12	
29 (E1)	08	
2A (E2)	13	
2B (E3)	08	
2C (E4)	14	Sequentially sets
2D (E5)	08	A Register bits
2E (E6)	15	
2F (E7)	08	
30 (E8)	16	
31 (E9)	08	
32 (EA)	17	
32 (EB)	08	
34 (EC)	18	
35 (ED)	08	'40' halt indicates
36 (EE)	19	bad A/C MUX.
37 (EF)	08	
38 (F0)	1A	
39 (F1)	08	
3A (F2)	1B	
3B (F3)	61	
3C (F4)	A6	
3D (F5)	91	Testing for a fault.
3E (F6)	(B8) 00	
3F (F7)	00	
40 (F8)	00	
41 (F9)	00	
42 (FA)	00	
43 (FB)	00	

12. Set and Clear Register Bits Routine (continued)

Address	Op Code
44 (FC)	00
45 (FD)	00
46 (FE)	00
47 (FF)	00

13. TB Logic Routine

Address	Op Code	
00 (100)	9E	
01 (101)	FF	
02 (102)	FF	Test ARO to
03 (103)	F8	test bit logic
04 (104)	80	'09' halt:
05 (105)	92	1. Bad ARO to TB.
06 (106)	(09) 09	2. Bad A REG
07 (107)	(01) 00	selection (F'8').
08 (108)	00	
09 (109)	81	3. Test logic.
0A (10A)	92	4. Test branch.
0B (10B)	(0E) DE	
0C (10C)	(01) 00	Test AR1 to
0D (10D)	00	test bit logic.
0E (10E)	82	
0F (10F)	92	Test AR2 to
10 (110)	(13) 13	test bit logic.
11 (111)	(01) 00	
12 (112)	00	
13 (113)	83	
14 (114)	92	Test AR3 to
15 (115)	(18) 18	test bit logic.
16 (116)	(01) 00	
17 (117)	00	
18 (118)	84	
19 (119)	92	Test AR4 to
1A (11A)	1D) 1D	test bit logic.
1B (11B)	(01) 00	
1C (11C)	00	
1D (11D)	85	
1E (11E)	92	Test AR5 to
1F (11F)	(22) 22	Test bit logic.
20 (120)	(01) 00	
21 (121)	00	

Comments

This routine is an automatic multifunction test. It tests all the A register to test bit logic functions and all other inputs to test bit logic except DTP, DTR, KSP, KSR, view mode. All latch functions are tested for both the active and inactive condition.

1. Select start address at FE Panel.
2. Place RUN/STOP to RUN.
3. Observe FE Panel for halt condition. If halt condition, troubleshoot faulty function—no halt indicated, no malfunction.

13. TB Logic Routine (continued)

Address	Op Code		
22 (122)	86		
23 (123)	92	Test AR6 to	
24 (124)	(27) 27	test bit logic.	
25 (125)	(01) 00		
26 (126)	00		
27 (127)	87		
28 (128)	92	Test AR7 to	
29 (129)	(2C) 2C	test bit logic.	
2A (12A)	(01) 00		
2B (12B)	00		
2C (12C)	88		
2D (12D)	92		
2E (12E)	(31) 31	Test AR8 to	
2F (12F)	(01) 00	test bit logic.	
30 (130)	00		
31 (131)	89		
32 (132)	92	Test AR9 to	
33 (133)	(36) 36	test bit logic.	
34 (134)	(01) 00		
35 (135)	00		
36 (136)	8A		
37 (137)	92	Test AR10 to	
38 (138)	(3B) 3B	test bit logic.	
39 (139)	(01) 00		
3A (13A)	00		
3B (13B)	8B		
3C (13C)	92	Test AR11 to	
3D (13D)	(40) 40	test bit logic.	
3E (13E)	(01) 00		
3F (13F)	00		
40 (140)	F0		
41 (141)	75	Test: 1.	Status select (F0).
42 (142)	86	2.	Send set.
43 (143)	92	3.	Send to test bit logic.
44 (144	(42) 47		
45 (145)	(01) 00		

13. TB Logic Routine (continued)

Address	Op Code			
46 (146)	00			
47 (147)	78			
48 (148)	86	Test:	1.	Send clear.
49 (149)	9A		2.	Test Branch.
4A (14A)	(4D) 4D			
4B (14B)	(01) 00			
4C (14C)	00			
4D (14D)	76			
4E (14E)	87	Test:	1.	Set OIL.
4F (14F)	92		2.	OIL to test bit logic.
50 (150)	(53) 53			
51 (151)	(01) 00			
52 (152)	00			
53 (153)	79			
54 (154)	87	Test:	OIL cleared.	
55 (155)	9A			
56 (156)	(59) 59			
57 (157)	(01) 00			
58 (158)	00			
59 (159)	7E			
5A (15A)	88	Test:	1.	Set RST INH.
5B (15B)	92		2.	RST INH to test bit logic.
5C (15C)	(5F) 5F			
5D (15D)	(01) 00			
5E (15E)	00			
5F (15F)	7F			
60 (160)	88	Test RST INH cleared.		
61 (161)	9A			
62 (162)	(65) 65			
63 (163)	(01) 00			
64 (164)	00			
65 (165)	B0			
66 (166)	89	Test:	1.	Set TRANS RDY.
67 (167)	92		2.	TRANS RDY to test bit logic.
68 (168)	(6B) 6B			

13. TB Logic Routine (continued)

Address	Op Code		
69 (169)	(01) 00		
6A (16A)	00		
6B (16B)	B1		
6C (16C)	89	Test TRANS RDY cleared.	
6D (16D)	9A		
6E (16E)	(71) 71		
6F (16F)	(01) 00		
70 (170)	00		
71 (171)	D0		
72 (172)	9E		
73 (173)	FF	Test JDRO to test bit logic.	
74 (174)	FF		
75 (175)	30		
76 (176)	08		
77 (177)	7C		
78 (178)	8A		
79 (179)	92		
7A (17A)	(7D) 7D		
7B (17B)	(01) 00		
7C (17C)	00		
7D (17D)	41		
7E (17E)	08		
7F (17F)	7C		
80 (180)	8A		
81 (181)	9A		
82 (182)	(85) 85		
83 (183)	(01) 00		
84 (184)	00		
85 (185)	AC		
86 (186)	8C	Test: 1.	Insert set.
87 (187)	92	2.	Insert to test bit logic.
88 (188)	(8B) 8B		
89 (189)	(01) 00		
8A (18A)	00		

13. TB Logic Routine (continued)

Address	Op Code		
8B (18B)	A8		
8C (18C)	8C	Test Insert cleared.	
8D (18D)	9A		
8E (18E) (91)	91		
8F (18F) (01)	00		
90 (190)	00		
91 (191)	AD		
92 (192)	8D	Test: 1.	Set Delete.
93 (193)	92	2.	Delete to test bit logic.
94 (194) (97)	97		
95 (195) (01)	00		
96 (196)	00		
97 (197)	A9		
98 (198)	8D	Test Delete cleared.	
99 (199)	9A		
9A (19A) (9D)	9D		
9B (19B) (01)	00		
9C (19C)	00		
9D (19D)	AE		
9E (19E)	8E	Test: 1.	Set PUL.
9F (19F)	92	2.	PUL to test bit logic.
A0 (1A0)	A3		
A1 (1A1)	00		
A2 (1A2)	00		
A3 (1A3)	AA		
A4 (1A4)	8E	Test PUL cleared.	
A5 (1A5)	9A		
A6 (1A6) (A9)	A9		
A7 (1A7) (01)	00		
A8 (1A8)	00		
A9 (1A9)	AF		
AA (1AA)	8F		
AB (1AB)	92	Test: 1.	Lockout set.
AC (1AC)	(AE) AE	2.	Lockout to test bit logic.
AD (1AD)	(01) 00		

13. TB Logic Routine (continued)

Address	Op Code		
AE (1AE)	00		
AF (1AF)	AB		
B0 (1B0)	8F	Test Lockout Reset.	
B1 (1B1)	9A		
B2 (1B2)	(B5) B5		
B3 (1B3)	(01) 00		
B4 (1B4)	00		
B5 (1B5)	2E		
B6 (1B6)	8B	Test: 1.	Ribbon Lift set.
B7 (1B7)	92	2.	Ribbon Lift to test bit logic.
B8 (1B8)	(BB) BB		
B9 (1B9)	(01) 00		
BA (1BA)	00		
BB (1BB)	2F		
BC (1BC)	8B	Test ribbon lift reset.	
BD (1BD)	9A		
BE (1BE)	00		
BF (1BF)	(01) 00		
C0 (1C0)	00		

14. Test Latch and Conditional Branch Routine

Address	Op Code	
00 (1C1)	41	
01 (1C2)	C0	
02 (1C3)	95	Test VDF and
03 (1C4)	(C7) 06	branch on VDF.
04 (1C5)	(01) 00	
05 (1C6)	00	
06 (1C7)	9D	Test branch on FDC
07 (1C8)	(CB) 0A	
08 (1C9)	(01) 00	
09 (1CA)	00	
0A (1CB)	9E	
0B (1CC)	10	Test FDF and
0C (1CD)	00	branch on FDC.
0D (1CE)	C1	
0E (1CF)	98	
0F (1D0)	(D3) 12	
10 (1D1)	(01) 00	
11 (1D2)	00	
12 (1D3)	9B	
13 (1D4)	(D7) 16	Test branch on VDF.
14 (1D5)	(01) 00	
15 (1D6)	00	
16 (1D7)	51	
17 (1D8)	C2	Test BR = 0
18 (1D9)	96	and branch on
19 (1DA)	(DD)1C	BR = 0.
1A (1DB)	(01) 00	
1B (1DC)	00	
1C (1DD)	9C	Test branch on
1D (1DE)	(E1) 20	BR = 1840 (730).
1E (1DF)	(01) 00	
1F (1E0)	00	

Comments

This test is a multifunction check. It tests various functions of the Test Latch logic and Branch control. When a malfunction is sensed, the program halts.

Test Initialize:

1. Select start address at FE Panel.
2. Place RUN/STOP to RUN.
3. Observe FE Panel for a halt condition. Halt indicates a fault.

14. Test Latch and Conditional Branch Routine (continued)

Address	Op Code	
20 (1E1)	00	
21 (1E2)	9E	
22 (1E3)	30	
23 (1E4)	07	
24 (1E5)	30	
25 (1E6)	08	Test BR = 1840
26 (1E7)	52	and branch on
27 (1E8	C3	BR = 1840.
28 (1E9)	97	
29 (1EA)	(ED) 2C	
2A (1EB)	(01) 00	
2B (1EC)	00	
2C (1ED)	75	
2D (1EE)	93	Test branch on Send.
2E (1EF)	(F2) 31	
2F (1F0)	(01) 00	
30 (1F1)	00	
31 (1F2)	76	
32 (1F3)	94	Test branch
33 (1F4)	(C1) 00	on OIL.
34 (1F5)	(01) 00	
35 (1F6)	00	

15. TC to RFS Test Routine

Address	Op Code	
00 (1F7)	F0	
01 (1F8)	9E	
03 (1F9)	FF	Write F from
03 (1FA)	FF	A REG to RFS.
04 (1FB)	C7	
05 (1FC)	54	
06 (1FD)	41	Write 0 from
07 (1FE)	C7	A REG to RFS.
08 (1FF)	9F	
09 (200)	FF	
0A (201)	FF	Write F from
0B (202)	54	C REG (0-3) to
0C (203)	F1	RFS.
0D (204)	C7	
0E (205)	54	Write F from
0F (206)	F2	C REG (4-7)
10 (207)	C7	to RFS.
11 (208)	54	Write F from
12 (209)	F3	C REG (8-11)
13 (20A)	C7	to RFS.
14 (20B)	54	
15 (20C)	64	Write F from
16 (20D)	F4	C REG (12) to RFS.
17 (20E)	C7	
18 (20F)	61	
19 (210)	DC	Read F from RFS
1A (211)	30	to A REG.
1B (212)	C8	
1C (213)	54	
1D (214)	C3	
1E (215)	9C	
1F (216)	(F7) 00	
20 (217)	(01) 00	
21 (218)	00	

Comments

This is a multifunction test for exercising the RFS MUX and writing data from A Register and C Register to RFS. It also reads data from RFS to A Register.

Test Initiate:

1. Select starting address at FE Panel.
2. Place RUN/STOP to RUN.
3. After a few seconds place RUN/STOP to STOP.
4. Observe display screen with HEX SW ON. The screen should have a repeating pattern of

   F 0 F F F F
   F 0 F F F 1.

5. Single-cycle program and monitor A REG. A REG changes from 'FFF' to '000' to '0F1'.

16. Error Logic Test Routine

Address	Op Code
00 (219)	41
01 (21A)	CE
02 (21B)	CF
03 (21C)	CD
04 (21D)	9E
05 (21E)	AA
06 (21F)	AA
07 (220)	CE
08 (221)	CF
09 (222)	CD
0A (223)	9E
0B (224)	55
0C (225)	55
0D (226)	CD
0E (227)	CE
0F (228)	CD
10 (229)	7A
11 (22A)	7B
12 (22B)	90
13 (22C)	(19) 00
14 (22D)	(02) 00
15 (22E)	00

Comments

This test is designed to test all circuitry associated with the error function.

Test Procedure:

1. On T/R control board place error display switch ON.
2. Select starting address at FE Panel.
3. Single-cycle and observe error display on CRT.
4. Place in RUN; error light is dimly lighted.

17. Keystroke Gathering Routine

Address	Op Code	
00 (22F)	D0	
01 (230)	9E	Sets JDR and FDR
02 (231)	21	so upper- and
03 (232)	00	lower-case
04 (233)	30	characters
05 (234)	08	may be displayed.
06 (235)	7C	
07 (236)	7D	
08 (237)	F0	
09 (238)	41	
0A (239)	C7	Clears data
0B (23A)	54	area of
0C (23B)	C3	screen.
0D (23C)	9C	
0E (23D)	(38) 09	
0F (23E)	(02) 00	
10 (23F)	51	
11 (240)	80	Monitor KSR
12 (241)	9A	for key
13 (242)	(40) 11	depressed.
14 (243)	(02) 00	
15 (244)	D4	
16 (245)	30	Loads keystroke
17 (246)	08	in A Register.
18 (247)	42	
19 (248)	70	
1A (249)	82	Clears KSR and
1B (24A)	92	checks KSP active
1C (24B)	(4E) 1F	high.
1D (24C)	(02) 00	
1E (24D)	00	
1F (24E)	71	

Comments

This routine monitors the keyboard input. When a key is depressed the keystroke data is entered into the A Register, where it can be observed at the FE Panel. It also displays the character on the screen for observation. Once the data area is filled (1840 characters) the screen is cleared and the routine begins painting characters on the first line again. Both upper- and lower-case characters can be displayed. The cursor position is moved in sync with the characters to allow cursor fault isolation. Halt traps are provided for checking KSR clearing, KSP setting and clearing. If a character is being moved to A Register when a key is depressed, then KSR is setting.

Test Procedure:

1. Select start address at FE Panel.
2. Depress MR and place RUN/STOP to RUN (screen is blank except OIL line).
3. Monitor A Register and depress key (EBCDIC keycode display in A Register and character appears on screen).
4. Observe screen for cursor movement when key is depressed.
5. Observe C Register for value of last character stored in RFS (first 8 bits same as A REG).

17. Keystroke Gathering Routine (continued)

Address	Op Code	
20 (24E)	82	
21 (250)	9A	Clears KSP and
22 (251)	(54) 25	checks KSP cleared.
23 (252)	(02) 00	
24 (253)	00	
25 (254)	C7	Moves character to screen.
26 (255)	DC	
27 (256)	32	Moves character on screen
28 (257)	C9	to C Register.
29 (258)	62	
2A (259)	54	Moves cursor ahead
2B (25A)	C6	on position.
2C (25B)	C3	
2D (25C)	9C	
2E (25D)	(40) 11	Check data area of screen for
2F (25E)	(02) 00	full condition; if full, clear
30 (25F)	90	screen and start over. If
31 (260)	(2F) 00	not full, look for next
32 (261)	(02) 00	keystroke.
33 (262)	00	

18. Video Control Logic Routine

Address	Op Code
00 (263)	9E
01 (264)	C1
02 (265)	00
03 (266)	F0
04 (267)	C7
05 (268)	54
06 (269)	9E
07 (26A)	24
08 (26B)	00
09 (26C)	C7
0A (26D)	54
0B (26E)	9E
0C (26F)	C1
0D (270)	00
0D (270)	00
0E (271)	C7
0F (272)	54
10 (273)	9E
11 (274)	30
12 (275)	00
13 (276)	C7
14 (277)	54
15 (278)	C3
16 (279)	9C
17 (27A)	(63) 00
18 (27B)	(02) 00
19 (27C)	00
1A (27D)	D0
1B (27E)	9E
1C (27F)	23
1D (280)	00
1E (281)	08
1F (282)	7C

Comments

This test is used to test the FDR and JDR control on the video display. Addresses 00-19 fill the screen with alternate fixed and variable fields of A's. To perform the test, follow closely the procedure below.

1. Depress MR and place RUN/STOP to RUN. Program halts at 1A–with BLANK CC/DISPCC in DISPCC, observe the first 23 lines of the video screen for $A\,{}^{2}_{4}A\,{}^{3}_{0}A\,{}^{2}_{4}A\,{}^{3}_{0}\cdot\,{}^{2}_{4}$ is a variable field and ${}^{3}_{0}A$ is a fixed field.
2. Turn display intensity down until the image is faintly visible.
3. Place RUN/STOP to STOP and, while single-cycling the program from the halt address observe the screen display.
4. At address '20' variable fields become intensified and fixed fields are dim. Also, the data light is lighted and the OCA is blanked.
5. At address '26' variable fields become dim and fixed fields intensify. Data light remains lighted but OCA returns.
6. At address '2E' all fields are intensified; the cursor moves to character 10 of the first line and begins to blink.
7. If all steps are good, then Video Control logic is functioning properly.

18. Video Control Logic Routine (continued)

Address	Op Code
20 (283)	9E
21 (284)	11
22 (285)	00
23 (286)	08
24 (287)	7C
25 (288)	52
26 (289)	54
27 (28A)	C6
28 (28B)	9E
29 (28C)	09
2A (28D)	00
2B (28E)	08
2C (28F)	7C
2D (290)	00

19. Transmit/Receive Functions Test Routine

Address	Op Code		Comments
00 (291)	1D	LRC ON.	This routine exercises all the untested transmit functions. It is used in conjunction with a scope, as no visual or halt aids are available for these functions.
01 (292)	1E	LRC OFF.	
02 (293)	1F	CLR LRC.	
03 (294)	B2	Set Trans EN.	
04 (295)	B3	Reset Trans EN.	
05 (296)	B4	LD TR.	
06 (297)	B8	CLR T/R SR.	
07 (298)	B9	CLR DR.	
08 (299)	9E		
09 (29A)	FF	Set MUX 1 to all 1's.	
0A (29B)	FF		
0B (29C)	D0		
0C (29D)	E4	Auto ACK.	
0D (29E)	E8	XMIT MUX = AR 0-7.	
0E (29F)	EC	XMIT MUX = AR 8-11.	
0F (2A0)	41		
10 (2A1)	E8	TR 1-8 = 0's.	
11 (2A2)	EC		
12 (2A3)	EO	TR 1-8 = 1's.	
13 (2A4)	E1	REC MS 1+2 = 0.	
14 (2A5)	E2	REC MS 1 = 1, MS 2 = 0.	
15 (2A6)	E3	REC MS 1 = 0, MS 2 = 1.	
16 (2A7)	90		
17 (2A8)	(91) 00		
18 (2A9)	(02) 00		

20. Printer Functions Test Routine

Address	Op Code	
00 (2AA)	2C	RST printer.
01 (2AB)	B5	Restore.
02 (2AC)	B6	CHAR Strobe.
03 (2AD)	BD	CHARR Strobe.
04 (2AE)	BE	PPR Feed Strobe.
05 (2AF)	2E	Ribbon Lift Set.
06 (2B0)	2F	Ribbon Lift Clear.
07 (2B1)	9E	
08 (2B2)	FF	
09 (2B3)	FF	
0A (2B4)	D0	
0B (2B5)	30	
0C (2B6)	08	
0D (2B7)	52	
0E (2B8)	2D	LD PRTR.
0F (2B9)	51	
10 (2BA)	2D	
11 (2BB)	90	
12 (2BC)	(AA)00	
13 (2BD)	(02) 00	
14 (2BE)	00	

Comments

This test routine exercises the printer functions on the Terminal Control board in order that they may be scoped. There are no visual or automatic halts to detect or indicate a fault.

21. Refresh Storage Exerciser Routine

Address	Op Code
00 (2BF)	F0
01 (2C0)	9E
02 (2C1)	FF
03 (2C2)	FF
04 (2C3)	C7
05 (2C4)	DC
06 (2C5)	30
07 (2C6)	C9
08 (2C7)	62
09 (2C8)	54
0A (2C9)	41
0B (2CA)	C7
0C (2CB)	C9
0D (2CC)	62
0E (2CD)	54
0F (2CE)	90
10 (2CR)	(BF) 00
11 (2D0)	(02) 00
12 (2D1)	BA
13 (2D2)	BB

Comments

This routine is a very basic program for filling the video screen with an alternate $\frac{F0}{F0}$ pattern. In addition, it reads these codes back to the C REG for observation. The RFS board is effectively half split for fault analysis, by observing what has been written into RFS.

22. T/R Control Board Exerciser Routine

Address	Op Code		Comments
00 (2D3)	9E		This routine is an exerciser for the T/R Control Board. It sends the T/R Control Board a general poll message sequence which, by means of a jumper wire, is looped back to the received data side of the UART; thus it is possible to exercise both the transmit and receive paths. To further enhance the routine, LRC sampling is performed, thus making it possible to check the LRC circuits. This routine should be run with viewer on. A screen presentation of
01 (2D4)	0D	Loads UART	
02 (2D5)	00	transmitter	
03 (2D6)	D0	with a '0D'	
04 (2D7)	E8	code (EOT).	
05 (2D8)	B4		
06 (2D9)	D1	Saves	
07 (2DA)	30	position.	
08 (2DB)	08		
09 (2DC)	51	Loop timeout	0 0 0 $^{3}6_{/}$ is displayed on the screen
0A (2DE)	54	to slow program	D F C 09 for ease of fault isolation.
0B (2DF)	C3	for timing	
0C (2E0)	9C	purposes.	
0D (2E1)	(DE) 0A		
0E (2E2)	(02) 00		
0F (2E3)	52	Recalls screen position.	
10 (2E4)	E2		
11 (2E5)	D8	Writes UART receiver	
12 (2E6)	08	value on screen.	
13 (2E7)	42		
14 (2E8)	F0		
15 (2E9)	CF		
16 (2EA)	54	Checks for full	
17 (2EB)	C3	screen.	
18 (2EC)	97		
19 (2ED)	(30) 5C		
1A (2EE)	(03) 04		
1B (2EF)	9E		
1C (2F0)	07	Loads UART transmitter	
1D (2F1)	00	with a '07' code	
1E (2F2)	D0	(address).	
1F (2F3)	E8		
20 (2F4)	B4		

22. T/R Control Board Exerciser Routine (continued)

Address	Op Code	
21 (2F5)	D1	
22 (2F6)	08	Saves screen position.
23 (2F7)	51	
24 (2F8)	54	
25 (2F9)	C3	Loop timeout to slow
26 (2FA)	9C	program for timing
27 (2FB)	(F8) 24	purposes.
28 (2FC)	(02) 00	
29 (2FD)	52	Recalls screen position.
2A (2FE)	E2	
2B (2FF)	D8	
2C (300)	08	Writes UART receiver value
2D (301)	42	on screen.
2E (302)	C7	
2F (303)	54	
30 (304)	C3	
31 (305)	9F	Checks for full screen.
32 (306)	(30) 5C	
33 (307)	(03) 00	
34 (308	9E	
35 (309)	0C	
36 (30A)	00	Loads UART transmitter with a
37 (30B)	D0	'0C' code (gen poll).
38 (30C)	E8	
39 (30D)	B4	
3A (30E)	D1	
3B (30F)	08	Saves screen position.
3C (310)	51	
3D (311)	54	
3E (312)	C3	Loop timeout to slow program
3F (313)	9C	for timing purposes.
40 (314)	(11) 3D	
41 (315)	(03) 00	

22. T/R Control Board Exerciser Routine (continued)

Address	Op Code	
42 (316)	52	Recalls screen position.
43 (317)	E2	
44 (318)	D8	
45 (319)	08	Writes UART receiver value
46 (31A)	42	on screen.
47 (31B)	C7	
48 (31C)	54	
49 (31D)	C3	
4A (31E)	9F	Checks for full screen.
4B (31F)	(30) 5C	
4C (320)	(03) 00	
4D (321)	D8	
4E (322)	E3	
4F (323)	03	Writes accumulated LRC
50 (324)	43	value on screen.
51 (325)	CF	
52 (326)	54	
53 (327)	C3	
54 (328)	97	Checks for full screen.
55 (329)	(30) 5C	
56 (32A)	(03) 00	
57 (32B)	B8	
58 (32C)	B8	
59 (32D)	90	Clears T/R Control and loops
5A (32E)	(D3) 00	to beginning of program.
5B (32F)	(02) 00	
5C (330)	41	
5D (331)	CF	
5E (332)	54	
5F (333)	C3	Clears screen.
60 (334)	9C	
61 (335)	(30) 5C	
62 (336)	(03) 00	

22. T/R Control Board Exerciser Routine (continued)

Address	Op Code	
63 (337)	51	
64 (338)	41	
65 (339)	1E	Initializes hardware and loops
66 (33A)	90	back to beginning of routine.
67 (33B)	(D3) 00	
68 (33C)	(02) 00	

Table A-1 Instruction Set (Sheet 1 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
00	HALT	TIMING COUNTER AND DECODER	At the conclusion of the time sequence in process, a halt instruction halts the counter, causing the processor to halt.
01	RAM STORE (0-3)	RAM ENABLE LOGIC	This instruction enables only the low RAM to accept inputs when T14 is active.
02	RAM STORE (4-7)	RAM ENABLE LOGIC	This instruction enables only the medium RAM to accept inputs when T14 is active.
04	RAM STORE (8-11)	RAM ENABLE LOGIC	This instruction enables only the high RAM to accept inputs when T14 is active.
08	RAM STORE ALL BITS	RAM ENABLE LOGIC	This instruction enables the high, medium and low RAM's to accept inputs when T14 is active.
10-1C	CLR AR BIT 0-12	A/C REG. MUX A REG	The "1" bit causes the A/C REG MUX to send to the A REG, as ACM0-11, –IRL0-11. The IRL bit that is low is LSB bit of the instruction. At T15 the A REG loads the 11 IRL bits because a '1X' is present.
1D	LRC ON	ERROR DISPLAY AND PRINTER LOGIC	This instruction activates the LRC CKTS on the T/R control board at T12 time.
1E	LRC OFF	SAME	This instruction deactivates the LRC CKTS on the T/R control board at T12 time.
1F	CLR LRC	SAME	This instruction clears the LRC buffers on the T/R control board when active.
20-2B	SET AR BIT 0-11	A/C REG. MUX A REG	The "2" bit causes the A/C REG MUX to send to the A REG., ACM 0-11, IRL 0-11. The IRL bit that is high is the LSB bit of the instruction. At T15 the A REG loads the 11 IRL bits because a '2X' is present.
2D	LD PRTR	STATUS REG.	This instruction sends a load command to the printer.
2C	RST PRTR	STATUS REG.	This instruction resets the printer.
2E	RIBBON LIFT	STATUS REG.	This instruction sets a ribbon lift command.
2F	RIBBON LIFT	STATUS REG.	This instruction resets a ribbon lift command.
30	KEYSTROKE DATA (LRC)	RAM ADDRESS BUFFER	'30' - '3F' are RAM addresses. They are executed at T13 time. These addresses supply 16 RAM's locations which are allocated according to data to be stored as selected by RAM MUX selection in conjunction with other microcode instructions of the subroutine being executed. For a description of the storage allocations refer to the address and mnemonic associated with it.
31	SPP	RAM ADDRESS BUFFER	
32	DPP	RAM ADDRESS BUFFER	
33	STATUS BITS	RAM ADDRESS BUFFER	
34	GP1	RAM ADDRESS BUFFER	
35	GP2	RAM ADDRESS BUFFER	
36	GP3	RAM ADDRESS BUFFER	
37	NEW SPP	RAM ADDRESS BUFFER	
38	FETCH POINTER	RAM ADDRESS BUFFER	
39	COMMAND	RAM ADDRESS BUFFER	
3A	DATA LENGTH	RAM ADDRESS BUFFER	

Table A-1 Instruction Set (Sheet 2 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
3B	TERM STG	RAM ADDRESS BUFFER	
3C	CURSOR POSITION	RAM ADDRESS BUFFER	
3D	COMMAND JDR	RAM ADDRESS BUFFER	
3E	SEQ COUNT 1	RAM ADDRESS BUFFER	
3F	SEQ COUNT 2	RAM ADDRESS BUFFER	
41	CLEAR A	A REGISTER	This instruction clears the A register at T12 time.
42	LOAD A FROM RAM	A REGISTER	This instruction loads the A register from the RAM at T13 time. When a '4' MSB is present the A/C MUX has a '00' for MUX SELECT, '00' selects RADO 0-11. The A register may be loaded by a (BLAR) from branch control, and at T15 time when a '1X', '2X' or a 'C8' instruction is being executed during the instructions A/C MUX is in different configurations.
44	INCREMENT A REG.	A REGISTER	When instruction'44' is present at T14 time, it produces a count-up CLK command and the A register is incremented. The value of the A register is a hexadecimal value.
48	DECREMENT A REG.	A REGISTER	When instruction '48' is present at T15 time, it produces a count-down command and the A register is decremented. The value of the A register is a hexadecimal value.
51	CLEAR B REGISTER	B REGISTER	Same as "Clear A Register."
52	LOAD B REGISTER FROM RAM	B REGISTER	This instruction loads the B register from the RAM at T13 time.
54	INCREMENT B REG.	B REGISTER	When instruction '54' is present at T14 time, it produces a count-up CLK command and the B register is incremented..The value of the B register is a hexadecimal value.
58	DECREMENT B REG.	B REGISTER	When instruction '58' is present at T15 time is produces a count-down command and the B register is decremented. the value of the B register is a hexadecimal value.
61	CLEAR C	C REGISTER	This instruction at T12 time clears the C register. The C register also may be cleared by a 'BCCR' command from branch control or RST2 from timing counter and decoder.
62	LOAD C FROM RAM	C REGISTER	At T13 when this command is present C register loads **RADO 0-11, C register may also be loaded** by a BLCR command from branch control. A/C MUX SELECT at '62'-T13 is '00', thus selecting RADO 0-11.
64	INCREMENT C REG.	C REGISTER	When instruction '64' is present at T14 time a count-up CLK is generated and C register is incrementing. The value is a decimal value.
68	DECREMENT C REG.	C REGISTER	When instruction '68' is present at T15 time a count-down CLK is generated and C register is decremented. The value is a decimal value.
70	CLEAR KSR	STATUS REGISTER	KSR (keystroke received) is set as a result of either a KB STB (keyboard strobe) or a FDB-CC STORE) command which generates a SR load. When a '70' occurs, the SR is clocked, resetting KSR and setting KSP (keystroke processing).

Table A-1 Instruction Set (Sheet 3 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
71	CLEAR KSP	STATUS REGISTER	When '71' occurs the SR is clocked, causing KSP to reset. The SEQ is KB STB (LOAD SR), '70' CLEAR KSR, '71' CLEAR KSP.
72	CLEAR DTR	STATUS REGISTER	DTR is set when the T/R control board detects a direct poll. Either a '72' or a RST2 clears DTR.
73	CLEAR DTP	STATUS REGISTER	DTP is set when the T/R control board detects a general poll. Either a '73' or a RST2 will clear DTP.
74	CLEAR LP READY		
75	SET SEND	STATUS REGISTER	When a '75' occurs the SEND latch is set at T12 time. This is microcode controlled.
76	SET OIL	STATUS REGISTER	When a '76' occurs the OIL latch is set at T12 time. This is microcode controlled.
77	AUTO CLK	STATUS REGISTER	When a '77' occurs a TOGGLE flop is either set or reset, depending on the previous condition.
78	CLR SEND	STATUS REGISTER	When a '78' occurs the SEND latch is reset at T12 time. This is microcode controlled.
79	CLR OIL	STATUS REGISTER	When a '79' occurs the OIL latch is reset at T12 time. This is microcode controlled.
7A	SET ERROR	STATUS REGISTER	When a '7A' occurs the ERROR latch is set at T12 time. This is microcode controlled.
7B	RST ERROR	STATUS REGISTER	When a '7B' occurs the ERROR latch is reset at T12 time, This is microcode controlled.
7C	LOAD JDR	JOB DEFINITION REGISTER	When a '7C' occurs RAM DATA (RADO 0-11) is loaded into the JDR at T12 time.
7D	LOAD FDR	FIELD DEFINITION REGISTER	When a '7D' occurs RAM DATA (RADO 0-11) is loaded into the FDR at T12 time.
7E	SET RST INH	STATUS REGISTER	When a '7E' occurs the RESET INHIBIT latch is set at T12. This is microcode controlled.
7F	RESET RST INH	STATUS REGISTER	When a '7F' occurs the RESET INHIBIT latch is reset at T12. This is microcode controlled.
80	KSR/AR0	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0', KSR is tested for a one condition. If equals "1", -TEST equals "0". If the test bit logic 1st stage MUX is selected by 'F8', AR0 is tested for a one condition. If equals "1", -TEST equals "0".
81	DTR/AR1	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0', DTR is tested for a one condition. If equals "1", -TEST equals "0". If the test bit logic 1st stage MUX is selected by 'F8', AR1 is tested for a one condition. If equals "1", -TEST equals "0".
82	KSP/AR2	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0', KSP is tested for a one condition. If equals one, -TEST equals zero. If the test bit logic 1st stage MUX is selected by 'F8', AR2 is tested for a one condition. If equals one, -TEST equals zero.
83	DTP/AR3	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0', DTP is tested for a one condition. If equals one, -TEST equals zero. If the test bit logic 1st stage MUX is selected by 'F8', AR3 is tested for a one condition.
84	VIEW MODE/ AR4	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0' / 'F8', VIEW MODE/AR4 is tested for a one condition. If equals one, —TEST equals zero.

Table A-1 Instruction Set (Sheet 4 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
85	AUTO/AR5	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0'/'F8', AUTO/AR5 is tested for a one condition. If equals one, -TEST equals zero.
86	SEND/AR6	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0'/'F8', SEND/AR6 is tested for a one condition. If equals one, -TEST equals zero.
87	OIL/AR7	TEST BIT LOGIC 2ND STAGE MUX	If the test bit logic 1st stage MUX is selected by 'F0'/'F8', OIL/AR7 is tested for a one condition. If equals one, -TEST equals zero.
88	RESET INH/ AR8	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0'/'F8', RESET INH/AR8 is tested for a one condition. If equals one, -TEST equals zero.
89	TRANS RDY/ AR9	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0', TRANS RDY from the T/R control board is tested for an active condition, if active, -TEST equals zero. If test bit logic 1st stage MUX is selected by 'F8', AR9 is tested for one condition. If equals one, —TEST equals zero.
8A	JDR0/AR10	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0', the test bit logic tests JDR0, to determine if terminal is in format mode or operator mode. If JDR0 is a one condition, -TEST equals "0" and TERM. = operator mode. If test bit logic 1st stage MUX is selected by 'F8', AR 10 is tested for a one condition. If equals one, -TEST equals zero.
8B	RIBBON LIFT/ AR11	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0'/'F8', RIBBON LIFT/AR11 is tested for a one condition. If equals one, -TEST equals zero.
8C	INSERT/—	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0', INSERT is tested for a once condition. If equals one, -TEST equals zero. 8C - F8 not used.
8D	DELETE/—	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0', DELETE is tested for a one condition. If equals one, -TEST equals zero. 8D - F8 not used.
8E	PUL/—	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0', PUL is tested for a one condition. If equals one, -TEST equals zero.
8F	LOCKOUT/ ROM TEST	TEST BIT LOGIC 2ND STAGE MUX	If test bit logic 1st stage MUX is selected by 'F0'/'F8', LOCKOUT/ROM TEST is tested for a one condition If equals one, -TEST equals zero.
90	UNCOND. BRANCH	BRANCH CONTROL	When a '9X' is seen the terminal goes into a branch mode. The '9' = branch and 'X' = type of branch. A branch causes the hardware to generate 2 additional memory cycles. Let T_0 = original cycle, T_1 = 1st extra cycle and T_2 = 2nd memory cycle. The sequence of events is as follows:
91	COMPARE	IR	
92	TB LAT	ROM ADD. MUX	T_0 Branch instruction seen. '9' loads branch SR, 'X' stored for later use. SR-A3 = 1.
93	SEND	ROM ADD. REG	T_1 Load low-order byte at T11 time into IR. SR-A4 = 1, ROM ADDRESS REG incremented at T14.
94	OIL		T_2 Low-order IR inhibited by -BC3 INH (A6 = 1).
95	VDF		T11 = Load IR high byte.
96	B = 0		T12 = Test branch 90-9D. If valid inhibit ROA increment.
97	B = 1840		T13 = Clear ROM ADDRESS register.
98	FDC		T14 = Load ROM ADDRESS register with IRL0-3, IRM 0-3, IRH 0-3 +IRA.
99	—COMP		T15 = Clear COMPARE AND TEST latch.

Table A-1 Instruction Set (Sheet 5 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
9A	–TB LAT		
9C	–B = 1840		
9D	–FDC		
9E	LOAD A REG FROM I REG.	BRANCH CONTROL IR	Like the address branch instructions (90-9D) the terminal goes into a branch mode, thus generating 2 extra memory cycles, the difference being that their purpose is to load a set of constants from the I REG into either A REG or C REG, depending on whether a '9E' (A REG) or '9F' (C REG) instruction is to be executed. The sequence is as follows:
9F	LOAD C REG FROM I REG.	ROM ADD MUX ROM ADD REG A/C MUX A REG	'9E'/ '9F' T_0 = Branch instruction seen. '9' loads branch SR, 'E' stored for later use. SR-A 3 = 1. T_1 = Load low-order byte at T11 into IR.SR-A4 = 1, ROM ADDRESS REG incremented at T14. T_2 = Low-order IR inhibited by -BC3 INH (A6 = 2). T11 = Load high-order byte. T12 = No action taken on ROM ADDRESS REG for both '9E' and '9F'. AR/CRM active to select IR output of A/C MUX. T13 = –BCCR or –BCAR active to clear the C REG or A REG. T14 = BLCR or BLAR active to load contents of I REG into either C REG or A REG.
A0	COMP C=M1(0-3)	ARITH. COMPARE LOGIC AND MUX	When an 'A0' instruction is seen the COMPARE MUX selects bit 0-3 of the C REG and MUX 1. These bits are compared in an arithmetic comparator and, if equal, –COMP = 0. M1 = 'A', 'B', JDR or FDR bits 0-3, depending on MUX SELECT. 'D0' = A REG, 'D1' = B REG, 'D2' = JDR, 'D4' = FDR.
A1	COMP C=M1(4-7)	ARITH. COMPARE LOGIC AND MUX	Same as 'A0' except bits 4-7 are compared.
A2	COMP C=M1(8-11)	ARITH. COMPARE LOGIC AND MUX	Same as 'A0' except bits 8-11 are compared.
A4	COMP C<M1(ALL)	ARITH COMPARE LOGIC AND MUX	Same as 'A0' except all bits of C REG are compared to MUX1. If C value is less than the value of M1, –COMP = 0.
A5	COMP C>M1(ALL)	ARITH COMPARE LOGIC AND MUX	Same as 'A0' except all bits of C REG are compared to MUX1, If C value is greater than the value of M1, –COMP = 0.
A6	COMP C=M1(ALL)	ARITH. COMPARE LOGIC AND MUX	Same as A0 except all bits of C REG are compared to MUX1.
A8	CLR INSERT LAT	STATUS REGISTER	This instruction at T12 time clears the INSERT latch.
A9	CLR DELETE LATCH	STATUS REGISTER	This instruction at T12 time clears the DELETE latch.
AA	CLR PUL	STATUS REGISTER	This instruction at T12 time clears the PUL latch.
AB	CLR LOCKOUT	STATUS REGISTER	This instruction at T12 time clears the LOCKOUT latch (allows operator keystroke input to resume).
AC	SET INSERT	STATUS REGISTER	This instruction at T12 time activates the INSERT function.
AD	SET DELETE	STATUS REGISTER	This instruction at T12 time activates the DELETE function.
AE	SET PUL	STATUS REGISTER	This instruction at T12 time activates the PUL function.

Table A-1 Instruction Set (Sheet 6 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
AF	SET LOCKOUT	STATUS REGISTER	This instruction at T12 time activates the LOCKOUT function (disables keystroke input processing).
B0	SET TRANS RDY LATCH	TEST LATCH LOGIC	This instruction sets TRANS RDY LATCH AT T12 time, TRANS RDY is also generated when −VIEW MODE is active. TRANS RDY informs the T/R control board that a message is awaiting transmission to the KAU.
B1	CLR TRANS RDY LAT	TEST LATCH LOGIC	This instruction clears TRANS RDY latch at T12 time.
B2	SET TRANS ENABLE	TEST LATCH LOGIC	This instruction sets TRANS ENABLE latch at T12 time. TRANS ENABLE is sent to the T/R control board to enable the UART to KAU path.
B3	CLR TRANS ENABLE	TEST LATCH LOGIC	This instruction clears TRANS ENABLE latch at T12 time.
B4	LOAD TR	TEST LATCH LOGIC	This instruction sets LOAD TR latch at T12 time. LOAD TR commands the UART on the T/R control board to load TR1-8 into the TRANS register.
B5	DEL LPA	ERROR DISPLAY CONT + PRINTER LOGIC	This instruction sets DEL LPA latch at T12 time. This command activates the line printer by generating a RESTORE command.
B6	CLR LPA	ERROR DISPLAY CONT + PRINTER LOGIC	This instruction clears the LPA latch at T12 time. A CHAR strobe is generated.
B7	SET ERE	TEST LATCH LOGIC	This instruction sets the ERE latch at T12 time. ERE controls the MSB of the TRANSMIT MUX (TM4). When set the TRANSMIT MUX is processing error information.
B8	CLR T/R SR	TEST LATCH LOGIC	This instruction generates a gate which is sent to T/R control board to clear the T/R shift register.
B9	CLR DR	TEST LATCH LOGIC	This instruction generates a gate which is sent to the T/R control board to clear the DATA RECEIVED command generated by the UART.
BA	SET VAR INH LAT	TEST LATCH LOGIC	This instruction sets the VAR INH latch at T12 time. VAR INH is sent to the video control logic to inhibit screen refresh by video addressing.
BC	TRANS MUX-TRM IN		
BD	TRANS MUX −LPAL	ERROR DISPLAY CONT +PRINTER LOGIC	This instruction generates CARR strobe at T12 time.
BE	TRANS MUX −LPAU	ERROR DISPLAY CONT + PRINTER LOGIC	This instruction generates PPR FEED strobe at T12 time.
BF	CLR ERE	TEST LATCH LOGIC	This instruction clears the ERE latch at T12 time.
C0	TEST AR= VDF	TEST LATCH LOGIC	This instruction monitors AR6,7 to detect a FDB. If a FDB is present AR3,4 are tested for '00' (variable data fleld). If a '00' is seen at T14 time when a C0 instruction is present the VDF latch is set.
C1	TEST AR= FDF	TEST LATCH LOGIC	This instruction monitors AR6,7 to detect a FDB. If a FDB is present AR3,4 are tested for '01' (fixed data field). If a '01' is seen at T14 time when 'C1' is present the FDC latch is set.

Table A-1 Instruction Set (Sheet 7 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
C2	TEST BR=0	TEST LATCH LOGIC	This instruction monitors BR0-11 for an all-zero condition. If all zeros are seen at T14 time, BR=0 latch is set.
C3	TEST BR=1840	TEST LATCH LOGIC	This instruction monitors BR0-11 for a 730 HEX or a 1840 decimal count. If a 1840 count is detected, C3 at T14 time sets the BR=1840 latch.
C6	LOAD CUR. COMP.	TEST LATCH LOGIC	This instruction generates CCR LD, which is sent to the cursor buffer and compare logic, where it loads the cursor position value from the B register into the cursor buffer.
C7	RFS-STORE	TEST LATCH LOGIC	This instruction at T12 time generates two commands. One is –TC STOP, which is sent to the timing counter and decoder, where it inhibits the timing counter, It also generates WRITE, which is sent to RFS CONTROL to enable RSD1 data from the RFS MUX to be written into RFS during screen retrace time (CT79-94). –TC STOR is also sent to CG +CONTROL to generate RSCR. RSCR performs the actual write operation.
C8	RFS-FETCH TO A REG	TEST LATCH LOGIC A REG. CG +CONTROL RAM ENABLE LOGIC RFS CONTROL	This instruction at T12 time generates –TC STOP, which sent CG +CONTROL to generate RSCR. RSCR in turn is sent to RFS CONTROL to RFS FETCH. Prior to a C8 instruction A 'DC' (RFS to RAM) and A'C5'-'C6' must be selected to open a path into the RAM from RFS. C8 at T14 time enables the RAM output, which is routed as RADO 0-11 through the A/C MUX to the A register, where at C8·T15 time the RAM data is loaded into the A REG. SEQ = T12 (–TC STOP, RSCR, 'DC'), T13 (RAM address), T14 (RAM OUTPUT ENABLE), T15 (Load into A REG).
C9	RFS-FETCH	TEST LATCH LOGIC CG +CONTROL RFS CONTROL	This instruction at T12 time generates –TC STOP, which in turn generates RSCR in CG +CONTROL, which performs the fetch by RFS CONTROL. Prior to execution of this instruction a data path must be selected.
CC	START AUDIO	TEST LATCH LOGIC	This instruction at T12 time activates START AUDIO command, which is sent to the RFS board for generation of the audio signal.
CD	CLR ERR REG	ERROR DISPLAY CONTROL AND PRINTER LOGIC	This instruction when seen clears the error registers.
CE	LOAD ERROR REG 1 FROM A REG	ERROR DISPLAY CONTROL AND PRINTER LOGIC	This instruction loads AR0-11 into Error Register 1.
CF	LOAD ERROR REG 2 FROM A REG	ERROR DISPLAY CONTROL AND PRINTER LOGIC	This instruction loads AR0-11 into Error Register 2.
D0	MUX 1-A REG IN	RAM MUX, MUX1	This instruction at T12 time selects a path from A REG output through the MUX1 and RAM MUX to the RAMs. It is primarily used to move A REG data (AR0-11) into the RAMs.
D1	MUX 1-B REG IN	MUX1, RAM MUX	This instruction is the same as 'D0' except B REG data (BR0-11) is selected by MUX1 and the RAM MUX.
D2	MUX 1-JDR in	MUX1, RAM MUX	This instruction is the same as 'D0' except job definition REG data (JDR0-11) is selected by MUX1 and the RAM MUX.

Table A-1 Instruction Set (Sheet 8 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
D3	MUX 1-FDR in	MUX1, RAM MUX	This instruction is the same as 'D0' except field definition REG data (FDR0-11) is selected by MUX1 and the RAM MUX.
D4	RAM MUX- KS IN	RAM MUX	This instruction at T12 time configures the RAM MUX to select keystroke data from the keyboard for presentation to the RAM's. This data is stored in RAM address '30'.
D8	RAM MUX-REG IN	RAM MUX	This instruction at T12 time configures the RAM MUX to select received data (RM0-11) from the T/R control board for storage in the RAMs. The address selected depends on whether the data is LRC, cursor position, starting address, data length, JDB, command or actual data.
DC	RAM MUX –RS IN	RAM MUX	This instruction configures the RAM MUX to select REFRESH STORAGE data (RSD0 0-7) from the REFRESH STORAGE board for storage in the RAMs.
E0	REC DATA –TERM ADD /TRANS-PAD	ERROR DISPLAY CONTROL + PRINTER LOGIC RECEIVE MUX- T/R CONTROL	'E0' thru 'E3' act on the RECEIVE MUX on the T/R control board by selecting the RECEIVE data by means of REC MS1 and REC MS2. When 'E0' is present the RECEIVE MUX on the T/R CONTROL board selects terminal address information for the REC data input to the TERM CONTROL –RAM MUX. REC MS1 +REC MS2 = 00. An 'E0' also generates a 000 for TX MUX SELECT, thus selecting PAD CHAR 'FF' for TR1-8.
E1	REC DATA –STATUS	SAME AS 'E0'	When 'E1' is present the RECEIVE MUX on the T/R CONTROL board selects REC DATA status information for the REC DATA input to the TERM CONTROL –RAM MUX. REC MS1 + REC MS2 = '10'.
E2	REC DATA –BITS 0-7	SAME AS 'E0'	When 'E2' is present the RECEIVE MUX on the T/R CONTROL board selects REC DATA (RR 1-RR 8) for the REC DATA input to the TERM CONTROL –RAM MUX. REC MS1 + REC MS2 = '01'.
E3	REC DATA –BITS + LRC 0-7	SAME AS 'E0'	When 'E3' is present the RECEIVE MUX on the T/R CONTROL board selects LRC 1-8 and RR1 - RR4 as the REC DATA input to the TERM CONTROL –RAM MUX. REC MS1 + REC MS2 = '11'.
E4	TRANS DATA –ACK	ERROR DISPLAY CONTROL + PRINTER LOGIC TRANSMIT MUX	'E4' - 'E8' provide TRANSMIT MUX selection by setting TM 1,2,4 to different configurations. When 'E4' is present the TRANSMIT MUX is configured to select a hardwire ACK character (0E'). TM 1,2,4 = '100'.
E8	TRANS DATA –MUX1 (4-11)	SAME AS E4	When 'E8' is present the TRANSMIT MUX is configured to select M1 4-11 from MUX1. TM 1,2,4 = '010'.
EC	TRANS DATA –MUX1 (0-3)	SAME AS E4	When 'EC' is present the TRANSMIT MUX is configured to select M1 0-3 from MUX 1. TM 1,2,4 = '110'. When either 'E0', 'E4', 'E8', 'EC' is present when error display is active (ERE set) the TRANSMIT MUX is configured to provide combinations of A REG DATA to the error display control and printer logic for use in error display.

Table A-1 Instruction Set (Sheet 9 of 9)

INSTRUCTION	MNEMONIC	HARDWARE	FUNCTION
F0	RFS-A REG IN/TEST BIT MUX-SR IN	REFRESH STORAGE MUX TEST BIT LOGIC FIRST STAGE MUX	F0,1,2,3,4 are used by the REFRESH STORAGE MUX to select the input to be sent to the REFRESH STORAGE board as RSD 1 0-7.'F8' selects test bit MUX SELECT. F0 and F8 are also used by the test bit logic 1st-stage MUX to select STATUS REG data (F0) or A REGISTER data (F8) for testing. Refer to explanation of instructions '80' – '8F' for more information.
F1	RFS-C REG (0-3)	REFRESH STORAGE MUX	F0 - When this instruction occurs RFS MUX selects A REG data (AR0-7) for RSD1 0-7, which is supplied to the RFS board. When this instruction occurs RFS MUX selects C REG data (CR0-3) for RSD1 0-3 (RSD1 4-7 are '1'), which is sent to the RFS board.
F2	RFS-C REG (4-7)	REFRESH STORAGE MUX	When this instruction occurs RFS MUX selects C REG data (CR 4-7) for RSD1 0-3 (RSD1 4-7 are'1'), which is sent to the RFS board.
F3	RFS-C REG (8-11)	REFRESH STORAGE MUX	When this instruction occurs RFS MUX selects C REG data (CR8-11) for RSD1 0-3 (RSD1 4-7 are '1'), which is sent to the RFS board.
F4	RFS-C REG (12)	REFRESH STORAGE MUX	When this instruction occurs RFS MUX selects C REG data (CR12) for RSD1 0 (RSD1 1-7 are '0'), which is sent to the RFS board.
F8	TBM-AR IN	REFRESH STORAGE MUX TEST BIT LOGIC 1ST STAGE MUX	'F8' is not used to generate RSD1 data; it generates a TBM SELECT, which enables the test bit MUX to select A REG data (AR 0-11) for testing.

CHAPTER 7

TERMINAL PROTOCOL

7.0 GENERAL

There are two distinct types of Terminal Protocol. The first and most important type is the Terminal-to-System protocol. This is the communication between the Terminal and the System 5000 and the operations which result from these communications. The second type is firmware protocol. This is essentially how the Terminal firmware is structured to control operations within the 5201 Terminal.

7.1 TERMINAL-TO-SYSTEM PROTOCOL

The 5201 Terminal cannot initiate activity between itself and the System. It is always slave to the keystation adapter unit (KAU), which is part of the System main frame. All Terminal-to-System activity, and vice versa, is accomplished with the use of predefined messages with set formats. It is these messages which control the Terminal-to-System activities. The Terminal is sampled periodically (the frequency is dependent on the number of Terminals established to be present by the System SCON) by the KAU to determine if a Terminal has requested System intervention to service it. The KAU accomplishes this by initiating a general poll message to a selected Terminal (see Figure 7-1). The selected Terminal responds in one of two ways. If it does not require System intervention (Terminal lockout not set), it merely acknowledges the receipt of the general poll message by sending a hardware acknowledge message (see Figure 7-1). If it does require System intervention (Terminal lockout set), the Terminal responds with a message containing its address, the last keystroke it processed, the contents of the job definition register, the cursor position address, and the 40 bytes of the Operator Command Area (OCA). After responding with this message, the Terminal remains in lockout until the System either has completed the operation requested or has sent a special message to clear the lockout at the Terminal. Because the time required for the System to process the Terminal request and respond may vary considerably, the Terminal, while in lockout, is configured to acknowledge general poll messages through firmware. This is to ensure that the System does not delete the Terminal from its System configuration because it did not respond.

In response to the request for System service, the System, after processing information contained in the OCA position of the Terminal response, initiates one of four operations to the Terminal: WRITE DATA, READ DATA, ERASE DATA, or BLINK SELECTED DATA. If the Terminal successfully completes the operation initiated by the System, it acknowledges completion or, in the case of READ DATA, responds with a message containing the data requested. If

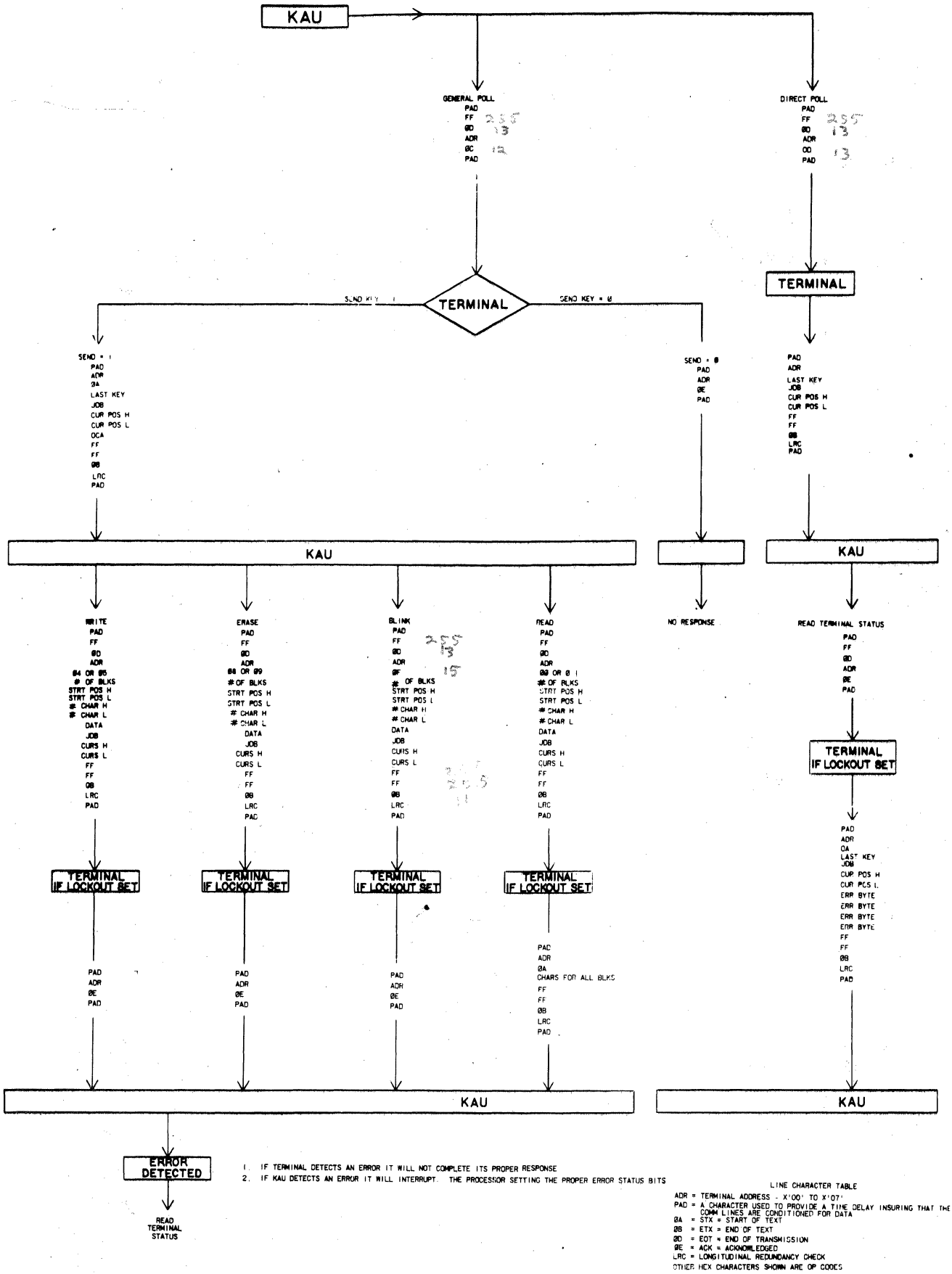


Figure 7-1 KAU Terminal Protocol

D3 GND

D4 EN OUTPUTS

D16 EN ERROR OUT

D17 16X RCV CLK

D18 Next Char. enable

D20 Recive Data in

A1 16X XMIT CLK

A2 Parity ⏚ (even)

A3 Bits/Char } + (8 Bits)

A4

A5 STOP BITS ⏚ (2)

A6 Parity EN ⏚ (none)

A7 Disable +

A8 XMIT

↓

A15 PARallel INPUT

A18 Send ⊓

D5 RCV DATA

↓

D12 OUTPUT

D13 Parity

D14 framing } ERRORS

D15 overrun

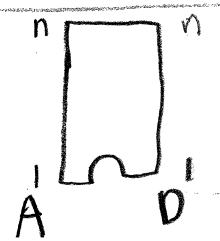
D19 Char RCVD STB.

A16 XMIT OUTPUT

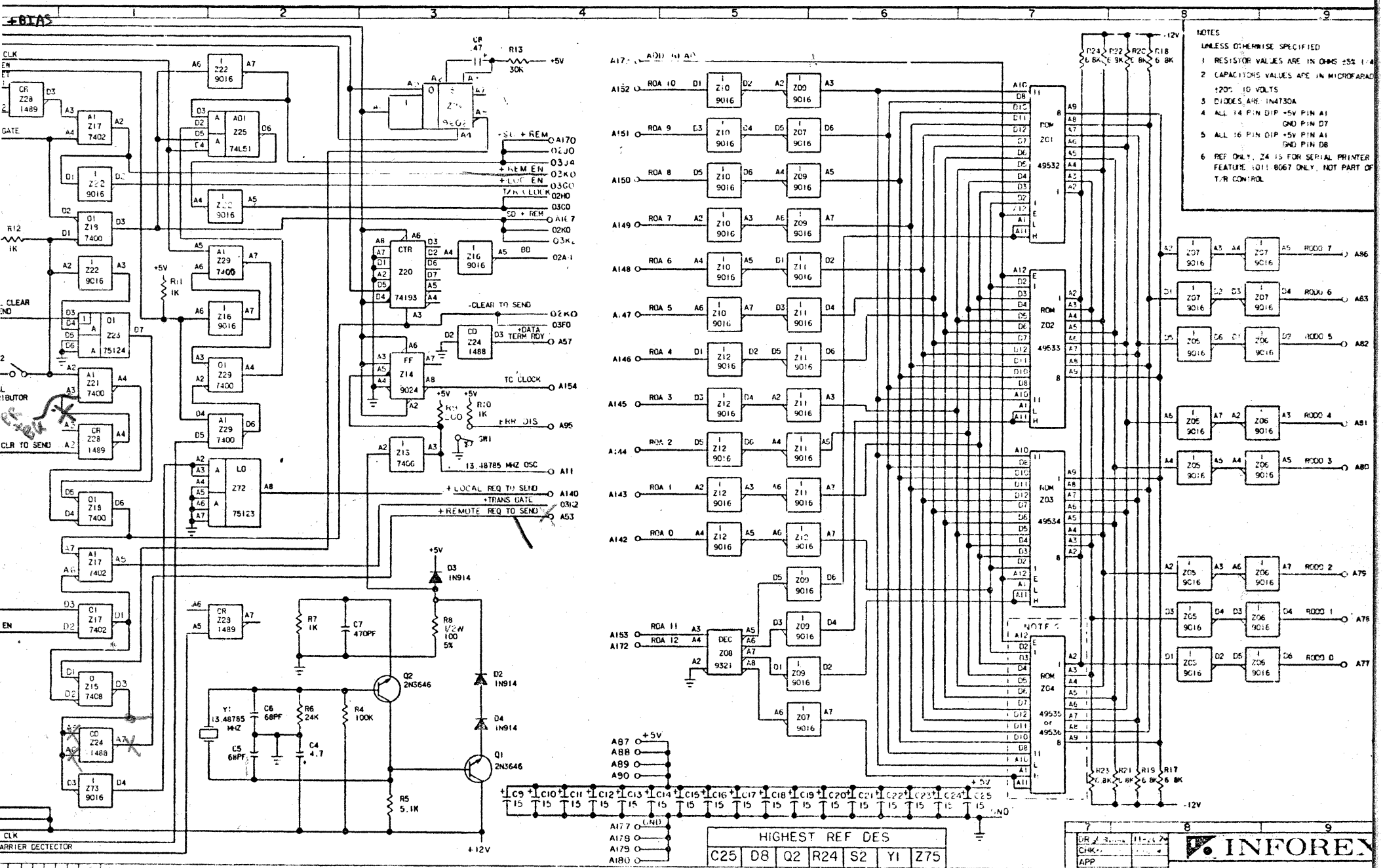
A17 complete

A19 LOAD OK

A20 Reset



228
224



NOTES
UNLESS OTHERWISE SPECIFIED
1 RESISTOR VALUES ARE IN OHMS ±5% 1/4
2 CAPACITORS VALUES ARE IN MICROFARAD ±20% 10 VOLTS
3 DIODES ARE 1N4730A
4 ALL 14 PIN DIP +5V PIN A1 GND PIN D7
5 ALL 16 PIN DIP +5V PIN A1 GND PIN D8
6 REF ONLY, Z4 IS FOR SERIAL PRINTER FEATURE (011 8067 ONLY, NOT PART OF T/R CONTROL

HIGHEST REF DES						
C25	D8	Q2	R24	S2	Y1	Z75

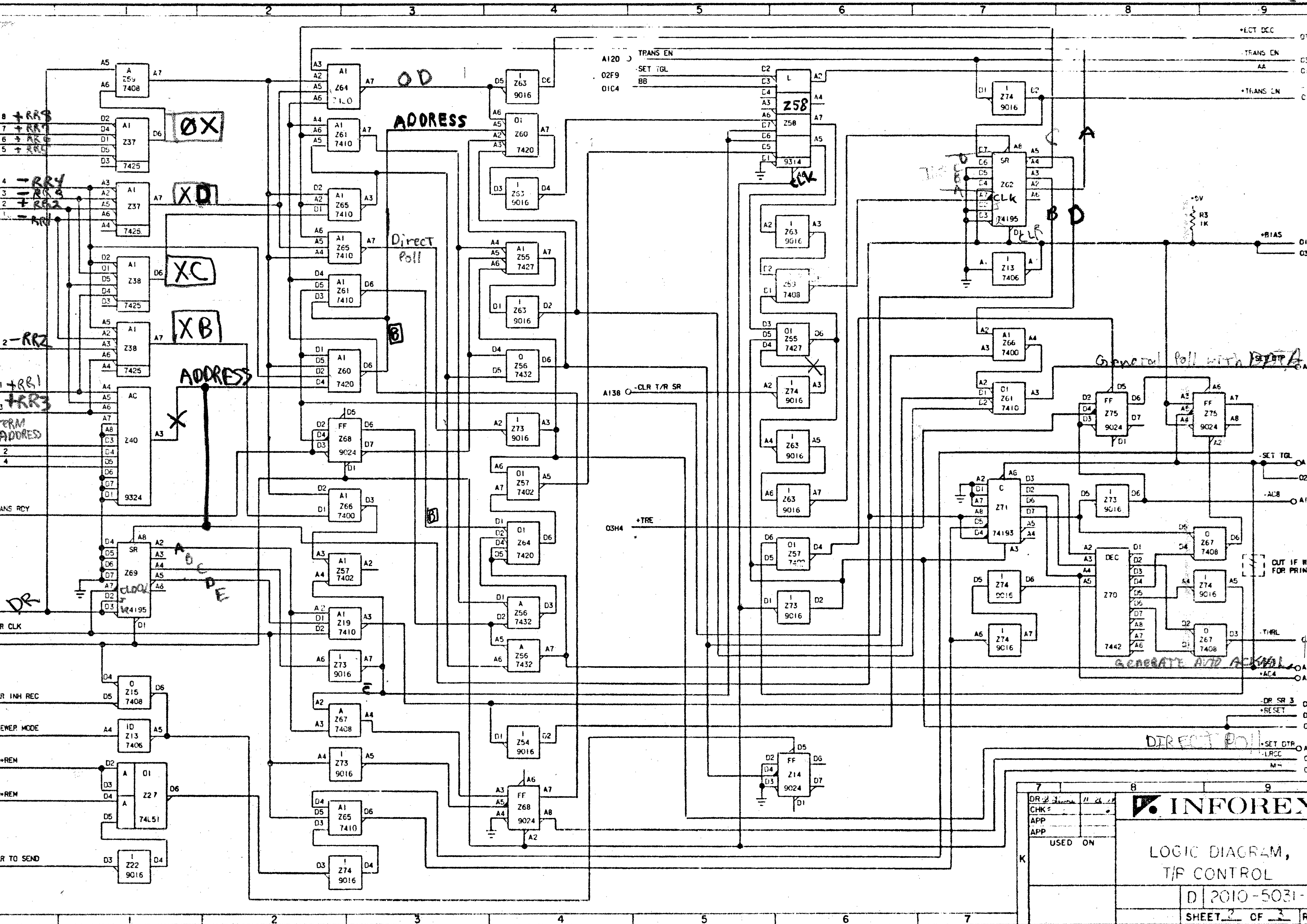
DR		
CHK		
APP		
APP		
USED ON		

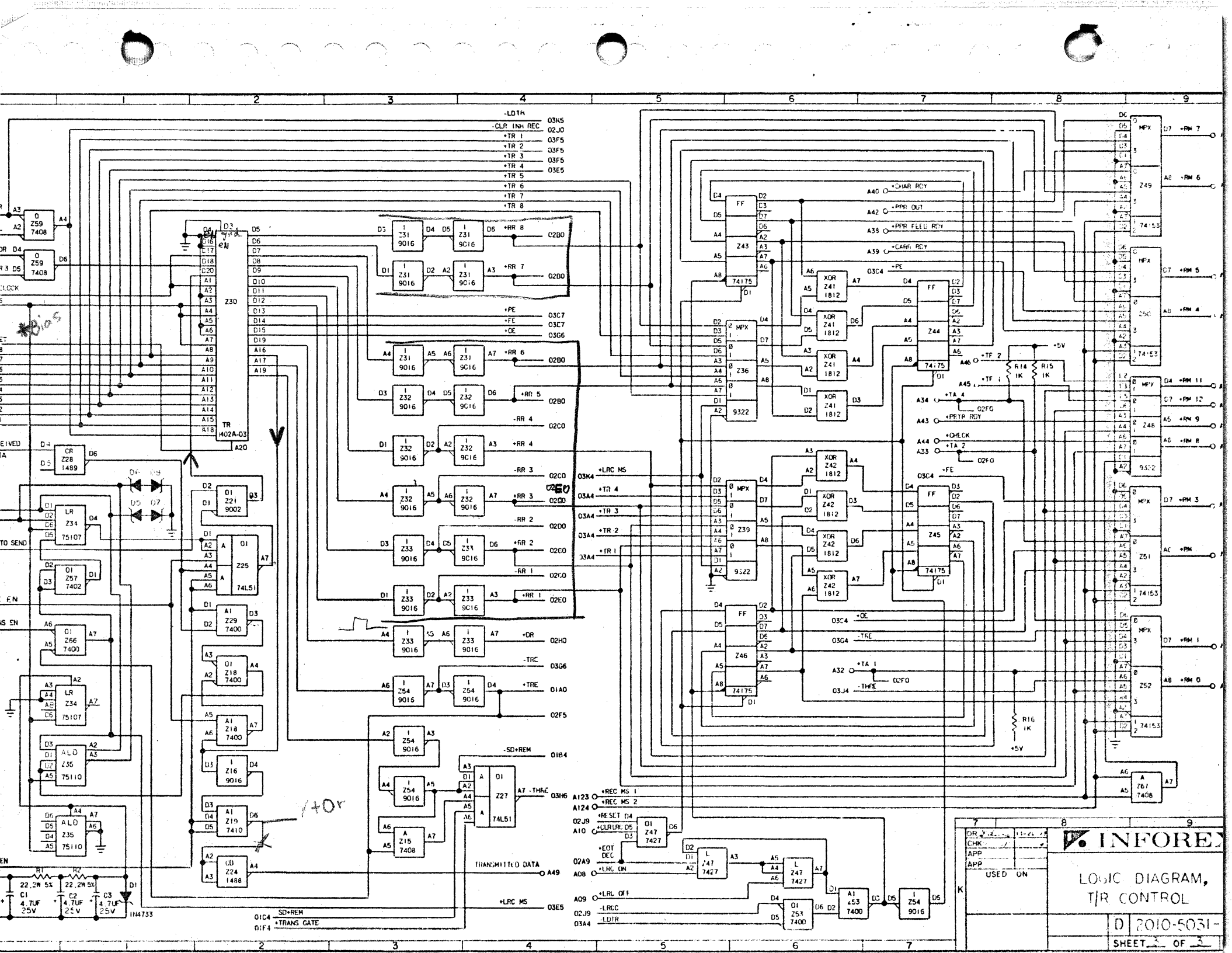
INFOREX

LOGIC DIAGRAM, T/R CONTROL

D 2010-5031-

SHEET 1 OF 2 R


+ECT DEC
TRANS EN
AA
+TRANS EN
+BIAS
SET TGL
-AC8
-THRL
+AC4
-DR SR 3
+RESET
SET DTR
-LRCC
A120
02F9
01C4
SET TGL
BB
A138
-CLR T/R SR
03H4
+TRE
+5V
R3
1K
CUT IF WI FOR PRINT
ANS RDY
R CLK
R INH REC
EWEP MODE
+REM
R TO SEND
OD
ADDRESS
ØX
XD
XC
XB
Direct Poll
General Poll with DATA
GENERATE AND ACK
DIRECT POLL
USED ON
INFOREX
LOGIC DIAGRAM,
T/P CONTROL
D 2010-5031-
SHEET 2 OF 3


-LDTR
-CLR INH REC
+TR 1
+TR 2
+TR 3
+TR 4
+TR 5
+TR 6
+TR 7
+TR 8
+PE
+FE
+OE
+RR 8
+RR 7
+RR 6
+RR 5
-RR 4
+RR 4
-RR 3
+RR 3
-RR 2
+RR 2
-RR 1
+RR 1
+DR
-TRC
+TRE
-SD+REM
+LRC MS
+REC MS 1
+REC MS 2
+RESET D4
+CLR LRC D5
+EOT DEC
+LRC ON
+LRC OFF
-LRCC
-LDTR
+TRANS GATE
TRANSMITTED DATA
+CHAR RDY
+PPR OUT
+PPR FEED RDY
+CARR RDY
+PE
+TF 2
+TF 1
+TA 4
+PETR RDY
+CHECK
+TA 2
+FE
+OE
-TRE
+TA 1
-THRE
+5V
+RM 7
+RM 6
+RM 5
+RM 4
+RM 11
+RM 10
+RM 9
+RM 8
+RM 3
+RM
+RM 1
+RM 0
TR 1402A-03
22.2W 5%
4.7UF 25V
1N4733
USED ON
INFOREX
LOGIC DIAGRAM, T/R CONTROL
D 2010-5031-
SHEET 3 OF 3