



**CDC® BREAKPOINT CONTROLLER
AND BREAKPOINT PANEL
DT120, DT195, FC402**

GENERAL DESCRIPTION
OPERATION AND PROGRAMMING
INSTALLATION AND CHECKOUT
THEORY OF OPERATION
DIAGRAMS
MAINTENANCE
PARTS DATA

HARDWARE REFERENCE/MAINTENANCE MANUAL

||





**CDC® BREAKPOINT CONTROLLER
AND BREAKPOINT PANEL
DT120, DT195, FC402**

GENERAL DESCRIPTION
OPERATION AND PROGRAMMING
INSTALLATION AND CHECKOUT
THEORY OF OPERATION
DIAGRAMS
MAINTENANCE
PARTS DATA

HARDWARE REFERENCE/MAINTENANCE MANUAL

MANUAL TO EQUIPMENT LEVEL CORRELATION SHEET

This manual reflects the equipment configurations listed below.

EXPLANATION: Locate the equipment type and series number, as shown on the equipment FCO log, in the list below. Immediately to the right of the series number is an FCO number. If that number and all of the numbers underneath it match all of the numbers on the equipment FCO log, then this manual accurately reflects the equipment.

EQUIPMENT TYPE	SERIES	WITH FCOs	COMMENTS
DT120-A DT195-A FC402-A	01 02 02		



LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Cover	--								
Title Page	--								
ii	D								
iii/iv	D								
v/vi	D								
vii/viii	D								
ix	C								
x	C								
1-1	A								
2-1	A								
3-1	B								
3-2	B								
4-1 thru 4-6	A								
4-7	C								
4-8 thru 4-11	A								
4-12	C								
4-13 thru 4-47	A								
4-48	C								
5-1	C								
5-2	A								
5-3 thru 5-54	C								
6-1	B								
7-1 thru 7-14	C								
A-1 thru A-16	A								
B-1 thru B-3	A								
Comment Sheet	D								
Cover	--								



PREFACE

This manual is to be used in conjunction with the Control Data® Basic Micro-Programmable Processor Hardware Maintenance Manual.

It contains the theory of operation, diagrams, maintenance information, and parts data for the DT195-A, DT120-A, and FC402-A Breakpoint Controller and Breakpoint Panel.

Information presented in this manual is intended for use by maintenance personnel in training and in the field.

The Basic Micro-Programmable Processor Hardware Maintenance Manual contains a detailed description of the basic processor itself as well as a listing of applicable documents. Other documents that may be of use to the reader are listed below:

<u>Publication</u>	<u>Publication Number</u>
Basic Micro-Programmable Processor Hardware Maintenance Manual	39451400
Micro-Programmable Computer Family Hardware Reference Manual	88973400
CYBER 18 Overview Manual	60475000
CYBER 18 Computer Systems Central Processor Field Repair Guide	60475001



CONTENTS

<p>1. GENERAL DESCRIPTION 1-1</p> <p>Functional Description 1-1</p> <p>Physical Description 1-1</p> <p>Electrical Description 1-1</p> <p>Environmental Conditions 1-1</p> <p>2. OPERATION AND PROGRAMMING 2-1</p> <p>3. INSTALLATION AND CHECKOUT 3-1</p> <p>Installation 3-1</p> <p>Bit Numbering Conventions 3-1</p> <p>Checkout 3-1</p> <p>4. THEORY OF OPERATION 4-1</p> <p>Breakpoint Controller Functional Operations 4-1</p> <p>Function Control Register 4-1</p> <p>Auto-Display 4-2</p> <p>Breakpoint 4-2</p> <p>Macro Breakpoint 4-2</p> <p>Micro Breakpoint 4-3</p> <p>Breakpoint Controller Control Commands 4-3</p> <p>H Control Character 4-3</p> <p>I Control Character 4-3</p> <p>J Control Character 4-3</p> <p>K Control Character 4-3</p> <p>L Control Character 4-3</p> <p>Master Clear 4-4</p> <p>Deadstart 4-4</p> <p>Breakpoint Controller and Processor Communication 4-4</p> <p>Micro Mode 4-4</p> <p>Macro Mode 4-4</p> <p>Breakpoint Controller Hardware 4-6</p> <p>Breakpoint Controller Data Path Hardware 4-6</p> <p>Function Control Register 4-6</p> <p>Breakpoint Register 4-6</p> <p>Shift Register Input Selector 4-6</p> <p>Shift Register 4-7</p> <p>MIR Input Data Selector 4-8</p> <p>Left Shift Input Data Selector 4-8</p> <p>Hexadecimal-to-ASCII Converter 4-8</p> <p>Transmitter Input Data Selector 4-9</p> <p>Universal Asynchronous Receiver/Transmitter 4-9</p>	<p>ASCII-to-Hexadecimal Converter 4-10</p> <p>Remote/Local Data Selector 4-10</p> <p>Shift Buffer Register 4-10</p> <p>Control Character Register 4-10</p> <p>Breakpoint Controller Control Hardware 4-11</p> <p>Control Instruction Format 4-11</p> <p>Control Block Diagram Logic Description 4-12</p> <p>Control Instruction PROMs 4-13</p> <p>Control Instruction Address Counter 4-13</p> <p>Control Instruction Decode 4-14</p> <p>Breakpoint Controller Micro Instructions 4-16</p> <p>Shift Counter 4-20</p> <p>Bit Count Register 4-20</p> <p>Breakpoint 4-21</p> <p>Breakpoint Controller Timing Chain 4-21</p> <p>Breakpoint Controller and Processor Interface Signals 4-21</p> <p>Go and Stop 4-23</p> <p>Baud Rate Generator 4-23</p> <p>Breakpoint Controller Controlware Controlware Listing Description 4-23</p> <p>Basic Controlware Operations 4-24</p> <p>Typical Breakpoint Controller Operations 4-24</p> <p>Breakpoint Panel 4-25</p> <p>Switch Encoder 4-25</p> <p>Breakpoint Panel Shift Register 4-25</p> <p>Control Code Display 4-26</p> <p>Glossary 4-26</p> <p>5. DIAGRAMS 5-1</p> <p>Interconnecting Signal Diagram 5-1</p> <p>Logic Diagrams 5-1</p> <p>6. MAINTENANCE 6-1</p> <p>Special Handling 6-1</p> <p>Spares Testing 6-1</p> <p>Preventive Maintenance 6-1</p> <p>7. PARTS DATA 7-1</p>
---	---

APPENDIXES

<p>A Controlware Listing A-1</p>	<p>B UART Details B-1</p>
----------------------------------	---------------------------

FIGURES

1-1 External Interface Capability of Breakpoint Controller	1-1	4-16 Change/Fetch Operation Flow Chart (J11: Command)	4-42
2-1 Breakpoint Panel Controls and Indicators	2-1	4-17 Transmit Operation Flow Chart (J11: Command)	4-43
4-1 Function Control Register	4-1	4-18 Enter P with 234 Command (K1234:), Change/Fetch Operation	4-45
4-2 Micro-Mode Flow Diagram	4-5	4-19 Breakpoint Panel Functional Block Diagram	4-48
4-3 Macro-Mode Flow Diagram	4-6	5-1 Interconnecting Diagram	5-1
4-4 Panel Interface Data Path Block Diagram	4-7	5-2 Breakpoint Panel Logic Diagram	5-3
4-5 Control Instruction Format	4-11	5-3 Breakpoint Controller (PWA 96752176) Logic Diagram	5-5
4-6 Control Block Diagram	4-12	5-4 Breakpoint Controller (PWA 88909300) Logic Diagram	5-37
4-7 PROMs Configuration to Form 96 x 24 Bit Control Instruction	4-19	7-1 Breakpoint Controller and Panel Spare Parts List	7-3
4-8 Breakpoint Controller Micro Instructions	4-19	7-2 Breakpoint Panel Printed Wiring Assembly	7-5
4-9 PROM B7 Format	4-22	7-3 Breakpoint Controller Printed Wiring Assembly (96752176)	7-7
4-10 Panel Interface Clock	4-24	7-4 Breakpoint Controller Printed Wiring Assembly (88909300)	7-11
4-11 Controlware Bit Configuration	4-29		
4-12 Input Operation Flow Chart	4-31		
4-13 Change/Fetch Operation Flow Chart	4-36		
4-14 Transmit Operation Flow Chart	4-39		
4-15 Input Operation Flow Chart (Enter J)			

TABLES

2-1 Breakpoint Panel Switches and Indicators LED	2-1	4-6 Jump Conditions	4-15
3-1 Breakpoint Panel Display Bit Equate Matrix	3-2	4-7 Strobe Signals	4-16
4-1 Function Control Register	4-1	4-8 Display Code PROM B7	4-18
4-2 Display Code Definitions	4-2	4-9 Breakpoint Controller Micro Instructions	4-20
4-3 Corresponding Hexadecimal Numbers and ASCII Code	4-9	4-10 Eight-Input Priority Encoder Truth Table	4-25
4-4 ASCII to Hexadecimal Conversion	4-10	4-11 Binary Code of Breakpoint Panel Switches	4-26
4-5 Control Instruction Field Programmable Read-Only Memory	4-14	4-12 Glossary of Terms	4-27

GENERAL DESCRIPTION

This manual contains the functional and physical descriptions of the breakpoint controller and breakpoint panel.

FUNCTIONAL DESCRIPTION

The breakpoint controller and breakpoint panel, when used with the basic micro-programmable processor, provide a man/machine interface. The breakpoint panel is the device for inputting the control commands and displaying the responses. The breakpoint controller receives the control command, executes it in conjunction with the processor, and returns a response to the breakpoint panel. The breakpoint controller also has the capability of interfacing with the remote devices shown in figure 1-1.

PHYSICAL DESCRIPTION

The breakpoint controller consists of one 11- by 14-inch printed circuit board that plugs into a slot of the processor chassis. The breakpoint panel consists of a 16- by 4-1/2-inch printed circuit board that is mounted directly above the processor chassis. It connects to the panel interface module through a flexible ribbon cable.

ELECTRICAL DESCRIPTION

The breakpoint controller requires +5 V dc and ± 12 V dc, which are derived from the main central processing unit

(CPU) power supply via backplane pins. The breakpoint panel requires only +5 V dc, which is provided by the breakpoint controller via the flexible cable.

There are three types of logic level in the breakpoint controller:

- The breakpoint controller operation is performed using standard TTL logic level signals:

Logical zero (low)	+0.4 V dc or less
Logical one (high)	+2.4 to 5.25 V dc
- The RS232-C logical level is required for interfacing with any RS232-C-compatible device:

Logical zero (low)	-3 V dc to -12 V dc
Logical one (high)	+3 V dc to +12 V dc
- The 20 mA current loop is required for interfacing with the teletypewriter.

ENVIRONMENTAL CONDITIONS

Operational:

Temperature - 40°F to 120°F (4.4°C to 48.4°C)
(maximum thermal shock 0.2°F/min.)

Relative Humidity - 10 to 90 percent.

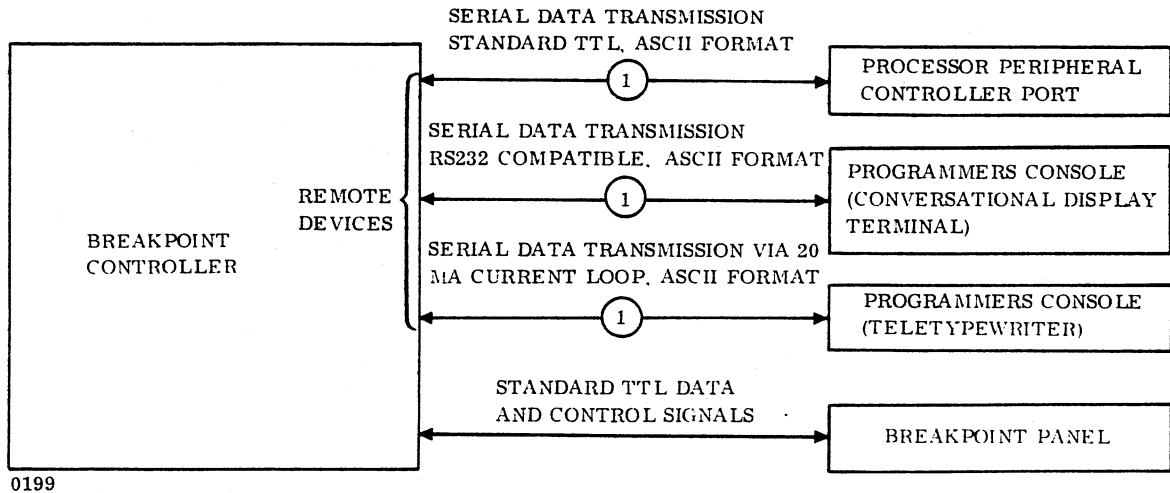


Figure 1-1. External Interface Capability of Breakpoint Controller

||

5

4

|

|

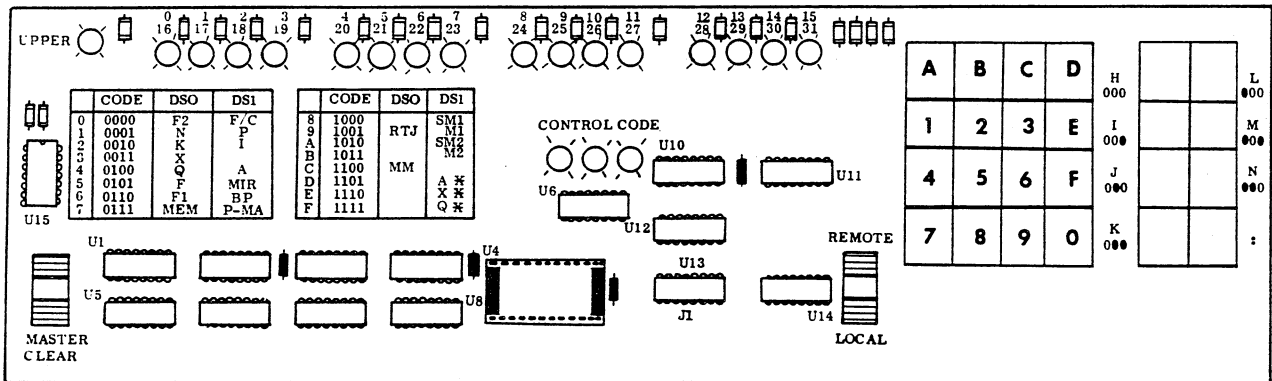


5

•

Once installed, the breakpoint controller requires no operating or programming except selecting the desired baud rate via a switch on the I/O-TTY controller. The breakpoint controller control commands are described in detail in section 4, Theory of Operation.

The breakpoint panel switches and display light-emitting diodes (LEDs) are shown in figure 2-1 and are described in table 2-1. The operation of switches is self-explanatory. The control command is entered via the switches in the same order as it is written. The response is displayed by the LEDs.



0161

Figure 2-1. Breakpoint Panel Controls and Indicators

TABLE 2-1. BREAKPOINT PANEL SWITCHES AND INDICATORS (LED)

Switch/Indicator	Type of Switch/Indicator	Function
Data display	16 indicators (LEDs)	Display current binary data as determined by specific control characters
UPPER	1 indicator (LED)	Indicates whether the display is upper or lower 16 bits; upper (0 through 15) when lit
CONTROL CODE	3 indicators (LEDs)	Indicate last control character entered as follows: 0 = H 1 = I 2 = J 3 = K 4 = L 5 = M (Undefined) 6 = N (Undefined) 7 = Error
MASTER CLEAR	Momentary switch	Master clear to CPU, memory, and peripheral controllers
REMOTE/LOCAL	Two-position switch	Enables panel or remote programmers console as follows: REMOTE position = Programmers console is enabled LOCAL position = Panel is enabled
Data entry	16 momentary pushbuttons	For entry of hexadecimal data (0 through F)
Control character	8 momentary pushbuttons	For entry of control character (H through N)
Function control definition table		Operator assistance information such as function control register definition and control character definition



INSTALLATION

The breakpoint controller is normally installed in slot U of the processor chassis. Refer to the card-slot assignment in the 1700 Enhanced Processor with MOS Memory System Overview Hardware Maintenance Manual. The breakpoint panel is mounted directly above the processor using the four screws that are provided. Install a flexible ribbon cable between the breakpoint controller and the breakpoint panel.

NOTE

Be sure that the arrows on the male connectors at both ends of the cable are pointed to pin number 1 of the female connectors on the boards.

BIT NUMBERING CONVENTIONS

NOTE

It should be noted that two different bit numbering conventions exist between the breakpoint controller and the basic micro-programmable processor. Throughout the breakpoint controller the least significant bit is 00 (LSB=00). In the basic micro-programmable processor, bit 00 is the most significant bit. The 16 data display indicators (LEDs) on the breakpoint panel are also numbered in the basic micro-programmable processor convention with the most significant bit equal to 00.

Table 3-1 provides an aid for converting from one convention to the other.

CHECKOUT

The breakpoint controller and the breakpoint panel can be checked out using the following general procedure. Refer to Breakpoint Controller Functional Operations in section 4 for details about the control commands.

1. Select a parameter in either display 1 or display 0 (i.e., the P register, micro memory, macro memory, or function control register).
2. Enter the selected parameter with the desired data.
3. Display the selected parameter.
4. Compare the displayed data with the entered data for correctness.
5. Repeat the above procedure for all parameters of displays 1 and 0. Also verify that H and I control commands are executed correctly.

Refer to section 3 of the Basic Micro-Programmable Processor Hardware Maintenance Manual for some typical panel interface operation procedures, such as enter/display a register, read/write micro memory, and read/write macro memory.

The success of the above operations indicates that the breakpoint controller and breakpoint panel are fully operational in performing the basic system communications.

TABLE 3-1. BREAKPOINT PANEL DISPLAY BIT EQUATE MATRIX

Panel Display Bit Number	Table 4-1 Bit/Digit	Function Control Bit Description	Panel Logic Term Relationship	
Lower Display				
31 (LSB)	00 (LSB)	FC00	FC00	SR00
30	01	FC01	FC01	SR01
29	02	FC02	FC02	SR02
28	03	FC03	FC03	SR03
27	04	FC04	FC04	SR04
26	05	FC05	FC05	SR05
25	06	FC06	FC06	SR06
24	07	FC07	FC07	SR07
23	08	FC08	FC08	SR08
22	09	FC09	FC09	SR09
21	10	FC10	FC10	SR10
20	11	FC11	FC11	SR11
19	12	FC12	FC12	SR12
18	13	FC13	FC13	SR13
17	14	FC14	FC14	SR14
16	15	FC15	FC15	SR15
Upper Display				
15	16	FC16	FC16	SR16
14	17	FC17	FC17	SR17
13	18	FC18	FC18	SR18
12	19	FC19	FC19	SR19
11	20	FC20	FC20	SR20
10	21	FC21	FC21	SR21
09	22	FC22	FC22	SR22
08	23	FC23	FC23	SR23
07	24	FC24	FC24	SR24
06	25	FC25	FC25	SR25
05	26	FC26	FC26	SR26
04	27	FC27	FC27	SR27
03	28	FC28	FC28	SR28
02	29	FC29	FC29	SR29
01	30	FC30	FC30	SR30
00 (MSB)	31 (MSB)	FC31	FC31	SR31

This section presents a functional description of the breakpoint controller module and the breakpoint panel. The breakpoint controller is completely described first and the breakpoint panel is described later. The theory of operation for the breakpoint controller consists of the controller functional operations and implementation of these operations using both hardware and controlware. Some typical operations are described at the end to complete the combined operation of hardware and controlware.

BREAKPOINT CONTROLLER FUNCTIONAL OPERATION

The breakpoint controller provides for all common computer control panel functions, such as enter/display registers, read/write macro memory, read/write micro memory, and breakpoint, as well as start, stop, and master clear functions via the breakpoint panel or the programmers console. The programmers console can be either a teletypewriter (20 mA current loop), a conversational display terminal (RS232-compatible), or any device with TTL logic level that has full duplex serial data transmission and ASCII format.

FUNCTION CONTROL REGISTER

A basic part of the breakpoint controller operation is the function control register (FCR). It is a 32-bit register that has access to the CPU similar to the way switches on a conventional panel have access to the CPU (see figure 4-1). FCR bit assignments are listed in tables 4-1 and 4-2.

The 32 bits of the FCR can be grouped into eight hexadecimal digits (digit 0 is the highest order) as follows:

- Display – Digits 0 to 1
- Machine Modes – Digits 2 through 5
- Machine Status – Digits 6 and 7

The display digits determine which individual register of two groups of registers (shown in table 4-2) can be displayed and/or modified. The machine modes, digits 2 through 5, are used to set such conditions as step/run mode, selective stop, selective skip, etc., in the same manner as switches on a conventional control panel. The two least significant digits (6 and 7) of the FCR indicate the status of the processor, such as overflow, parity error, protect fault, etc.

TABLE 4-1. FUNCTION CONTROL REGISTER

Bit	Digit	Bit definition	
00 00	(LSB)	Overflow	
01 01	7	Protected instruction	
02 02		Protect fault	
03 03		Parity error	
04 04		Interrupt system active	
05 05	6	Auto-restart enabled	
06 06		Micro running	
07 07		Macro running	
08 08	5	Enable auto-display	
09 09		Enable console echo	
10 0A		Enable micro memory write	
11 0B	4	Multilevel indirect address mode	
12 0C		Suppress console transmit	
13 0D		3	0 0 Breakpoint off
14 0E	0 1 Instruction reference breakpoint		
15 0F	1 0 Storage operand breakpoint		
16 10	1 1 All references breakpoint		
17 11	2	Breakpoint instruction (breakpoint stop if clear)	
18 12		Micro breakpoint, step, go, stop (macro if clear)	
19 13		Step	
20 14	1	Selective stop	
21 15		Selective skip	
22 16		Protect switch	
23 17	0	Display 1	
24 18		0	Display 0
25 19			
26 1A			
27 1B			
28 1C	(MSB)		
29 1D			
30 1E			
31 1F			

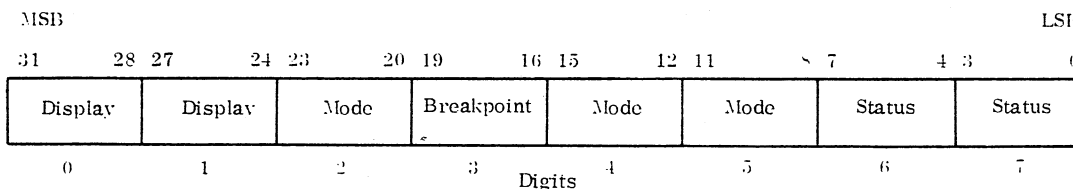


Figure 4-1. Function Control Register

TABLE 4-2. DISPLAY CODE DEFINITIONS

Code	Select Code	Display 1 (K Function)	Select Code	Display 0 (L Function)
0 0 0 0 0	J10:	FCR	J00:	F2 (Addressed by N register)
1 0 0 0 1	J11:	P [†]	J01:	N (MSBs) ^{††}
2 0 0 1 0	J12:	I [†]	J02:	K (LSBs) ^{††}
3 0 0 1 1			J03:	X
4 0 1 0 0	J14:	A [†]	J04:	Q
5 0 1 0 1	J15:	MIR	J05:	F
6 0 1 1 0	J16:	BP-P/MA	J06:	F1 { Addressed by K register Enabled by SM111
7 0 1 1 1	J17:	BP-P/MA (Display only)	J07:	MEM
8 1 0 0 0	J18:	SM1		
9 1 0 0 1	J19:	M1	J09:	RTJ
A 1 0 1 0	J1A:	SM2		
B 1 0 1 1	J1B:	M2		
C 1 1 0 0			J0C:	MM
D 1 1 0 1	J1D:	A*		
E 1 1 1 0	J1E:	X*		
F 1 1 1 1	J1F:	Q*		

[†]Used to address macro memory. The P and A registers are automatically incremented after each memory reference. The I register is not incremented after the memory reference.

^{††}The combined contents of these two registers are used to address micro memory. The K register is automatically incremented after each memory reference. The N register does not automatically increment.

The following are some restrictions for the use of FCR bits.

- FCR bits FC10 and FC11 (enable console echo and enable auto-display) are mutually exclusive; that is, the operator may select one or the other but not both simultaneously.
- Selecting BP-P/MA code 0 1 1 1 from table 4-2 results in both the breakpoint (BP) register and the page/memory address register (BP-P/MA) being displayed. BP is the leftmost 16 bits, and P/MA is the rightmost 16 bits. BP can only be modified if BP-P/MA code 0 1 1 0 is selected. P/MA cannot be modified in either case.
- Selecting the N or K register (table 4-2) results in both N and K being displayed. N is the leftmost eight bits, and K is the rightmost eight bits. However, the N register can only be modified when N is selected. Similarly, the K register can only be modified when K is selected.

AUTO-DISPLAY

When auto-display is enabled (FCR bit FC10 is set), the register selected by the control code and display codes is output to the operator's interface and continuously updated as long as the interface is a display terminal and not a teletypewriter. With auto display enabled, depressing a terminator (:, G, or @) with no control character preceding it causes a go signal. This method can be used for stepping through a micro or macro program.

BREAKPOINT

There are two types of breakpoint: micro and macro. If FCR bit FC19 is set, micro breakpoint is selected. If bit FC19 is clear, macro breakpoint is selected.

Macro Breakpoint

Bits FC16 and FC17 of the FCR are used to select three types of macro breakpoint.

Bit 17	Bit 16	
0	0	Breakpoint not selected
0	1	Instruction reference breakpoint
1	0	Store operand breakpoint
1	1	All references breakpoint

A macro breakpoint stop occurs when the breakpoint register is equal to the macro-memory address and the select conditions are met. If FCR bit FC18 is set, an interrupt rather than a stop occurs when the breakpoint conditions are met.

Micro Breakpoint

FCR bits FC16 and FC17 are used to select two types of micro breakpoint.

Bit 17	Bit 16	
0	0	Breakpoint not selected
0	1	Upper/lower micro instruction breakpoint
1	0	Micro-word breakpoint
1	1	Micro-word breakpoint

The upper/lower micro instruction breakpoint (FC17-FC16 = 01) requires that the micro memory address (P/MA) and upper/lower micro instruction selections are equal to the lower 13 bits of the breakpoint register to cause a micro stop. The micro-word breakpoint only requires that the 12 bits of the micro-memory address register (P/MA) are equal to the lower 12 bits of the breakpoint register.

BREAKPOINT CONTROLLER CONTROL COMMANDS

The control characters accepted by the breakpoint controller include: H, I, J, K, L, @, :, G, and ?. Control characters H through L identify the type of data or operation entered or returned. The at symbol (@), the colon (:), and G all perform an entry termination function. The question mark (?) generates a master clear. A normal control command consists of one control character (H, I, J, K, or L); zero, two, four, or eight hexadecimal digits (0 through F); and a terminating entry (:, @, or G) in that order.

A normal response consists of a control character identifying the data that follows and four or eight hexadecimal digits. If a transmission or operator error occurs on the control command entry, an asterisk (*) precedes the control character and the FCR is unconditionally displayed with the last legal control character. All entries except the ? cause a response, unless FCR bit FC15 (suppress console transmit) is set. The following are control character usage and examples. The colon (:) is used as the terminating entry.

H Control Character

The H control character is used to clear a specific bit in the FCR; for example, H14:.

This would clear FCR bit 14₁₆ (starting from the most significant bit (MSB) of FCR bit FC31) and the response would be a display of the updated FCR.

The H control character is also used to generate a stop if H: is entered. This is a micro stop if FCR bit FC19 is set and a

macro stop if FCR bit FC19 is clear. The response to this stop command is a display of FCR.

I Control Character

The I control character is identical to H except it sets a specific bit in the FCR. The I control character is also used to generate a go signal if I: is entered. This is a micro go if FCR bit FC19 is set. It is both a micro go and a macro go if FCR bit FC19 is clear.

J Control Character

The J control character is used to replace the contents of the FCR in a digit mode. While it may be used to change the value of any FCR digit, it is generally used to change digits 0 and 1. The value of digits 0 and 1 specifies which parameter is displayed on display requests or entered on enter requests (refer to table 4-2). The J control command always consists of J followed by two hexadecimal digits and a terminator, normally a colon (:). The first hexadecimal digit specifies one of FCR digits 0 through 5 and the second hexadecimal digit specifies the value the digit is to assume, 0 through F. The control command J14: would set FCR digit 1 to 4₁₆ (select the A register), and the response would be a display of the updated FCR.

The J control character is also used to display alternately the upper and lower 16 bits of a 32-bit register on a 16-bit breakpoint panel. The J: control command causes the other 16 bits to be displayed, and the UPPER indicator on the breakpoint panel is complemented.

K Control Character

The K control character is used to display or enter data into the parameter specified by display 1. The K control command has two formats. The first format, K:, is a request to display the parameter specified by display 1. The second format is an enter data request. It consists of K followed by four to eight hexadecimal digits followed by a terminator. The hexadecimal digits are the data to be entered. For example:

- To display the P register, enter:
 - J11: Select P register in display 1 (set FCR digit 1 to 1₁₆)
 - K: Display P register
- To enter 14FE₁₆ into the breakpoint register, enter:
 - J16: Select breakpoint register (set FCR digit 1 to 6₁₆)
 - K14FE: Enter 14FE₁₆ into breakpoint register

L Control Character

The L control character is operationally the same as the K control character, except that it is associated with display 0.

NOTE

When macro memory is displayed or entered, the register selected in display 1 is the macro-memory address. The display 1 selection must be the P, A, or I register. The selected P or A register is automatically incremented by 1 after the display. The I register is not incremented.

When micro memory is displayed or entered, the K register is the least significant eight bits of the address and N register bits N3 through N7 provide the remaining address bits. The K register is incremented by 1 after the display. The N register is not automatically incremented.

Master Clear

A master clear can be generated in one of several ways:

- A power-on master clear
- The master clear switch on the breakpoint panel
- An active low signal from a peripheral controller
- A question mark (?) code input from the programmers console

DEADSTART

The breakpoint controller also provides a deadstart capability that is nothing more than an automated version of keyboard transmission. Deadstart operation provides an automated method of loading data into micro or macro memory and all registers. During deadstart operation, the controller executes the control commands as they are continuously input from a deadstart device. The deadstart program must be written in terms of control commands that select the desired micro or macro memory, set up the initial address, load the data, and automatically generate a go signal at the end, if desired.

BREAKPOINT CONTROLLER AND PROCESSOR COMMUNICATION

During breakpoint panel operations, the processor is required by the controller to perform part of the operation. The breakpoint panel interface operation can be divided into three basic operations.

- Input operation - During this operation, the controller receives the control command from the breakpoint panel or remote devices.
- Change/fetch operation - During this operation, the control command operation is carried out by both the breakpoint controller and the processor (except that when the selected register is either the micro instruction register (MIR), FCR, or BP-P/MA, the processor is not required). The change operation changes the contents of the selected register with new entered data. The fetch operation reads out the contents of the selected register for display.
- Transmit operation - During this operation, the response of the control command is transmitted to the breakpoint panel or remote device.

The input and transmit operations are performed by the breakpoint controller itself. To perform the change/fetch operation, the breakpoint controller must generate the following basic three-step sequence:

1. Request control of the CPU. The breakpoint controller must first take over control of the processing element. The processor must also prepare itself before giving up control to the breakpoint controller.
2. Generate micro instructions to be executed by the processor for the desired change/fetch operation.

3. Exit the processor to return control. Figures 4-2 and 4-3 show the breakpoint controller and processor communication flow diagrams, depending upon the micro or macro mode of operation.

First the breakpoint controller must determine if it needs the CPU. The CPU is normally required except when MIR, FCR, or BP-P/MA is the selected register. If the processor is needed, the controller generates SETRCPU, and, depending on whether FCR bit FC19 is set or clear, the micro mode or macro mode sequence is generated.

MICRO MODE

When the processor is in micro mode, the breakpoint controller does not require any assistance from the emulator (firmware), but the processor must initially be stopped before the panel interface can take over control of the processor. To take control of the processor, the controller first generates the SETSM214 signal, which causes the processor to:

- Set the status mode register bit SM214. The SM214 pre-enables the breakpoint controller to MIR.
- Clear the micro instruction register (MIR).
- Cause the ENMM signal in the control 1 module to go low, which disables the micro memory and enables the breakpoint controller micro instructions to the micro memory three-state bus.
- Disable the strobe macro-memory address buffer register by causing GATEAB in the control 1 module to be high except for the read/write macro-memory operation.
- Save the micro-memory address by causing DECMAC to be high.
- Save test bit TB by disabling the GATET clock in the control 2 module.

The breakpoint controller now generates the CONSTART signal, which causes the processor to run in micro mode (allowing the main clock generator to run). The no operation (NOP) micro instruction, MIR = 0, is executed. The controller generates a micro instruction sequence with the M field equal to 01, which is executed by the processor to change/fetch the selected register. Prior to execution of the last breakpoint controller micro instruction, the controller generates CLRSM214, which clears status mode bit SM214 to pre-enable the micro memory to MIR. The controller now generates the last micro instruction with the M field equal to 00 and the T field equal to 000 to exit to the CPU. The M field equal to 00 causes the ENMM signal to go high, which enables the micro memory and disables the breakpoint controller to the micro-instruction register. Control is returned to the processor. The processor reads the micro instruction awaiting in MIR prior to the breakpoint controller sequence. The panel interface also stops the processor by generating the CONSTOP signal.

MACRO MODE

When the processor is in macro mode, the emulator is required to prepare the processor before relinquishing control to the breakpoint controller. When the controller determines that it needs the CPU, it generates the RCPUCPU signal which in turn generates the CONSTART signal. The CONSTART signal is required to insure that the micro code is running. Simultaneously, the breakpoint controller

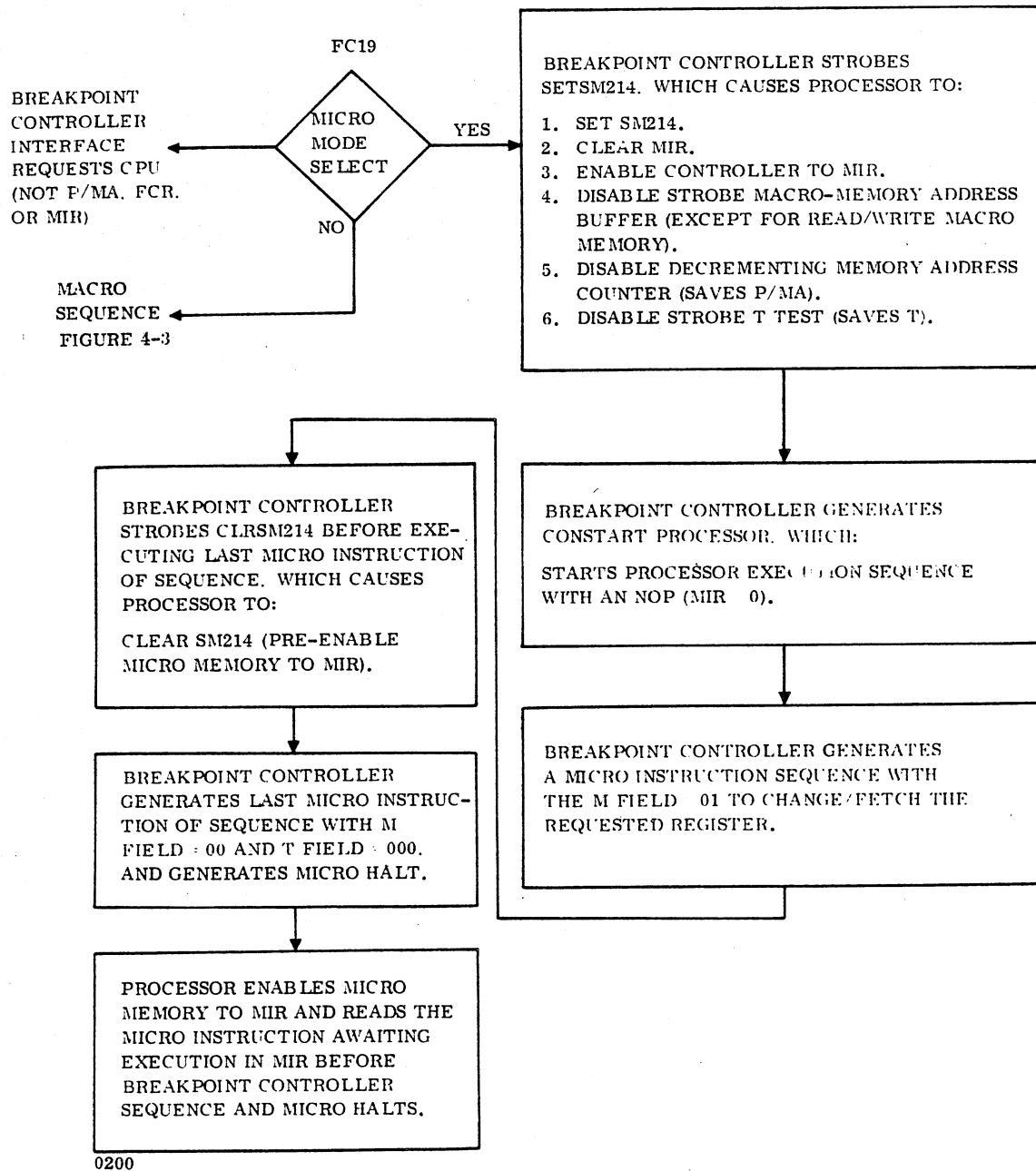


Figure 4-2. Micro-Mode Flow Diagram

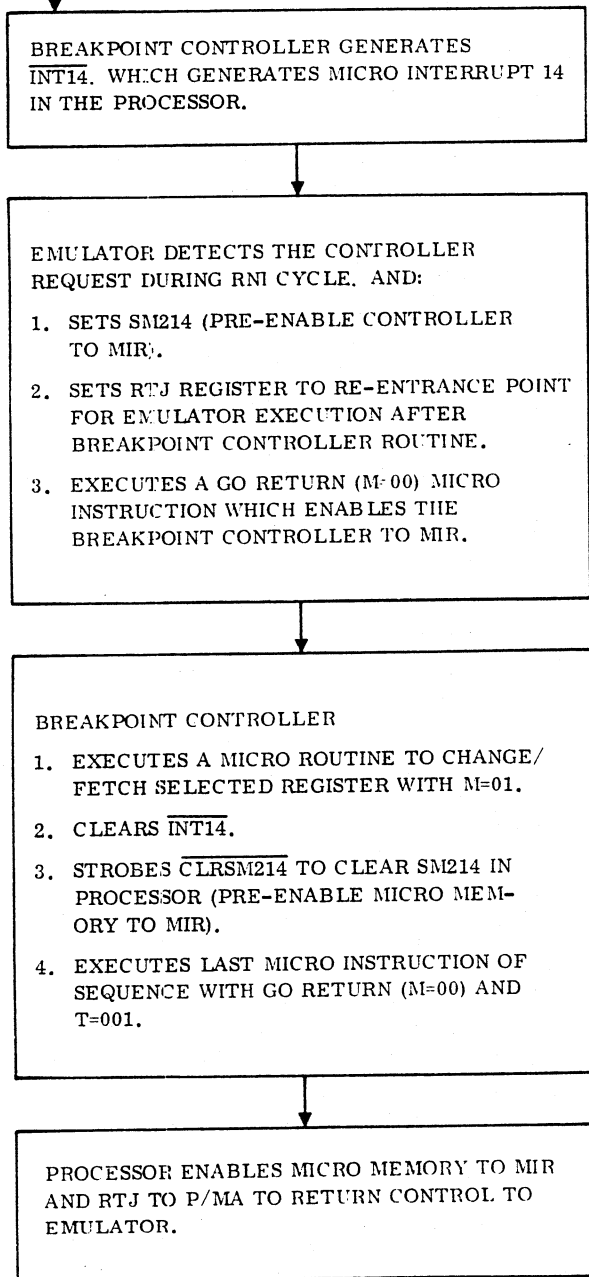
generates micro interrupt INT14, which is the panel request to CPU interrupt. The emulator performs the following operations once INT14 is recognized by the emulator during the read next instruction (RNI) cycle.

- Sets status mode interrupt bit SM214 to pre-enable the breakpoint controller to MIR.
- Sets return jump (RTJ) register to the re-entrance address for emulator execution after the breakpoint controller sequence.
- Executes a GO return (M field equal to 00) micro instruction which causes the ENMM signal in the control 1 module to go low. This disables the micro

memory and enables the breakpoint controller to the micro-memory three-state bus.

The controller now has control of the CPU. The controller generates a micro-instruction sequence with the M field equal to 01 to be executed by the CPU. After the CPU completes the required operation, the breakpoint controller exits the CPU by clearing INT14. The CLRSM214 signal is also generated to clear SM214. This pre-enables the micro memory to the micro-memory bus. The last controller micro instruction is executed with the M field equal to 00 and the T field equal to 001 to exit the CPU. The M field equal to 00 causes the ENMM signal to go high, enabling the micro memory and disabling the breakpoint controller to the

MICRO SEQUENCE
FIGURE 4-2



0201

Figure 4-3. Macro-Mode Flow Diagram

micro-memory three-state bus. The M field equal to 00 also selects the contents of the RTJ register as the address of the next micro instruction pair. Note that the RTJ register contained the re-entrance micro-memory address for the emulator. The T field equal to 001 selects the upper micro instruction at the address specified by the RTJ register. Control is now returned to the emulator and it continues at the point that it stopped prior to the controller sequence.

BREAKPOINT CONTROLLER HARDWARE

The breakpoint controller operations are performed by combining hardware and controlware. The breakpoint controller hardware is divided into two types of functional hardware for clarity: data path hardware and control hardware.

BREAKPOINT CONTROLLER DATA PATH HARDWARE

The breakpoint controller data path consists of the selectors, registers, shift register, universal asynchronous receiver/transmitter (UART), etc., that are interconnected as shown in figure 4-4. The data is manipulated by controlware via various data paths.

Function Control Register

The function control register (FCR) is a basic part of the breakpoint controller operation. It is a 32-bit register that has access to the CPU similar to the way switches on a conventional panel have access to the CPU. Table 4-1 shows the FCR bit assignments. The FCR consists of four HEX-D-type flip-flops, D10, F10, G10, and A11, which store the 24 FCR bits, FC8 through FC31. These FCR bits can be altered externally via control characters; FC00 through FC07 reflect processor status and cannot be changed externally. The FCR is cleared by the master clear signal going low. The FCR and the shift register form a complete feedback loop by having the output of one connected to the input of the other. This allows the FCR contents to be altered via the shift register. The FCR is clocked by the STROBE F/C signal, which is generated by controlware. Refer to the paragraph on Control Instruction Decode below.

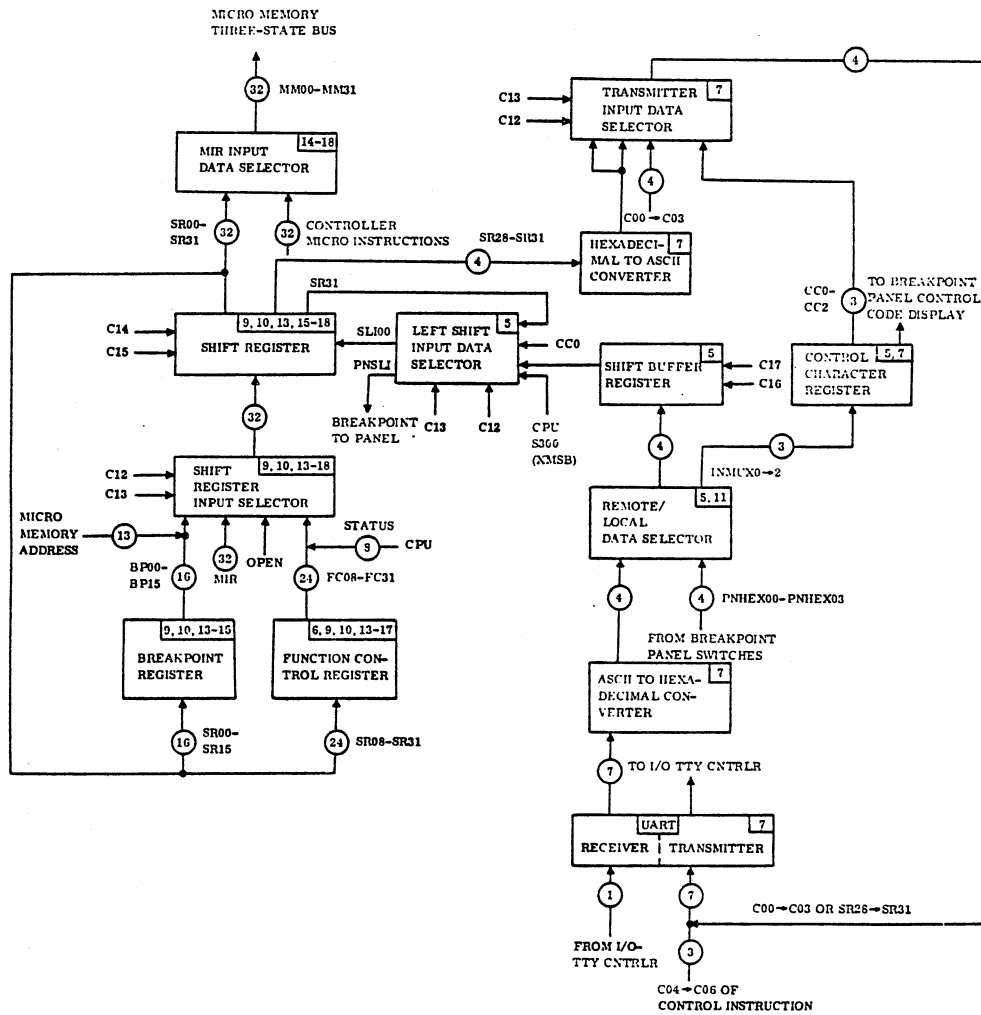
Breakpoint Register

This is a 16-bit register that contains the breakpoint address. When the macro- or micro-memory address is equal to the breakpoint address, the computer is stopped if the breakpoint select conditions are met. The breakpoint register consists of three HEX-D type flip-flops, B10, G11, and K10. The breakpoint register receives its input from shift register output SR00 through SR15, and its output is in turn connected to the input of the shift register. This allows the contents of the breakpoint register to be changed via the shift register. The output of the breakpoint register is also connected to breakpoint comparators to be compared with the macro- or micro-memory address.

Shift Register Input Selector

This selector consists of 4-to-1 multiplexers that select one of the three sources to be parallel-loaded into the shift register for modifying or transmitting. Select signals C12 and C13 are bits 12 and 13 of the 24-bit control instructions. Refer to the paragraphs on Control Instruction Format below. The selector selects the sources as follows:

C13	C12	Selected Source
0	0	Open
0	0	Function control register FC00-FC31
1	0	Micro-instruction register MIR00-MIR31
1	1	Breakpoint register BP00-BP15 and micro-memory address PG0-PG3, MA0-MA7, and BTLW



NOTE: THE NUMBERS IN THE UPPER RIGHT CORNER OF EACH BLOCK CORRESPOND TO THE LOGIC DIAGRAM SHEET NUMBER

0202

Figure 4-4. Panel Interface Data Path Block Diagram

Shift Register

The 32-bit shift register consists of eight universal bidirectional shift registers (G12, A12, B12, H12, D12, E12, J12, and M12), which handle all the data transfers serially in and out of the breakpoint controller module. The following is typical of shift register usage:

- During the input operation, the data entered from the breakpoint panel or remote device is shifted left serially into the shift register for storing via the shift buffer register.
- The data stored in the shift register can be shifted into the arithmetic/logical unit (ALU) X register, which in turn is loaded into any selected parameter of the processor.
- The contents of the ALU X register can be shifted left into the shift register and then shifted to the breakpoint panel via SR31 or to the remote device for display via the UART.

The shift register holds the FCR for modification. The control signals of the shift register are bits C14 and C15 of the panel interface control instructions; therefore, shift register operation is totally a function of controlware. Shift register operation is controlled by C14 and C15 as follows:

C15	C14	Operations
0	0	Do nothing
0	1	Left shift
1	0	Right shift
1	1	Parallel loading

The shift number is determined by controlware and tracked by a shift counter. A description of the shift counter is given in the control hardware section. During the left shift operation, shift left serial input data SL100 is derived from the output of the left shift input selector. The output of shift register SR00 through SR31 is sent to the micro-instruction register via the MIR input data selector and also to the function control and breakpoint registers. This allows modification of these registers by loading the desired data from the shift register. During the display operation, the most significant bit (MSB) SR31 of the shift register is sent out to the breakpoint controller via the left shift input selector if the breakpoint panel is selected. However, if the remote device is selected, the most significant four bits of the shift register, SR28 through SR31, are converted to ASCII code before sending out to the remote device via the universal asynchronous receiver transmitter (UART).

MIR Input Data Selector

The MIR input data selector selects either the output of the shift register, SR00 through SR31, or the panel interface-generated micro instructions to be input to MIR. The selector consists of six 2-to-1 multiplexers, A13, B13, G9, D13, M10, and M11, and four 4-to-1 multiplexers, E11, D11, H9, and H10. The input selection is determined by the RCPU signal. When the RCPU signal is high, it indicates the processing element is under the control of the breakpoint controller; therefore, the controller-generated micro-instruction, MM00 through MM31, is selected. When the RCPU signal is low, SR00 through SR31 is selected to allow MIR to be modified. The MIR input data selector is also a part of controller micro-instruction generation hardware; it is described in detail under Breakpoint Controller Micro Instructions below. The output of the selector is connected to the micro-memory three-state bus so that it can be loaded into the micro-instruction register.

Left Shift Input Data Selector

The left shift input data selector, 4-to-1 multiplexer A6, selects the left shift serial input data SL100 for the breakpoint controller shift register and the left shift serial input data, PNL5I, for the breakpoint panel shift register.

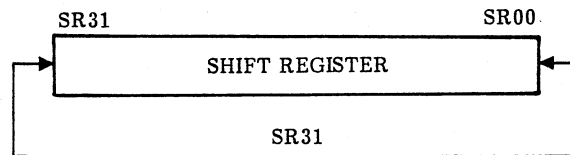
The select signals of the selector are bits C12 and C13 of the control instruction. The input sources are selected as follows:

C13	C12	SL100 for Controller Shift Register	PNL5I for Breakpoint Panel Shift Register
0	0	CC0	Upper flip-flop output (K85)
0	1	SR31	SR31
1	0	S300	SR31
1	1	Output of shift buffer register	Output of shift buffer register

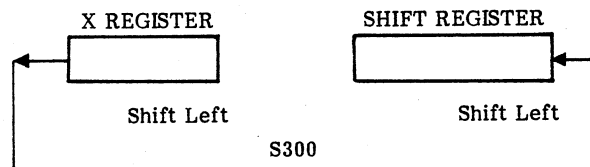
The controlware selects the left shift serial input data for the breakpoint controller shift register according to its desired operation.

- CC0 is the least significant bit (LSB) of the control character code. CC0 is low for the H control character and high for the I control character; therefore, CC0 is selected to clear or set a specific bit in the FCR for the H or I control command, respectively.

- SR31 is the most significant bit of the shift register and is selected when an end-around shift left operation is desired:



- S300 is the most significant bit of central processor ALU selector S3. When X is enabled to the ALU, S300 is the most significant bit of the ALU X register. Therefore, S300 is selected when data transfer from X register to shift register is desired:



- The output of the shift buffer register is selected to transfer data from the shift buffer register to the shift register. This operation is normally required to store data coming in from the breakpoint panel or the remote device via the remote/local data selector during an input operation. It is also used to modify the FCR contents in the digit mode (refer to the paragraph on the control character under Breakpoint Controller Control Commands).
- The left shift serial input data PNL5I for the breakpoint panel is used solely for display purposes.
- The upper flip-flop output at K8-5 is selected to indicate that either the upper or lower 16 bits of data are being displayed via the UPPER indicator on the breakpoint panel. The upper flip-flop output K8-5 is complemented if a J: command is entered, since the J code can be used to display alternately the upper or lower 16-bit data. However, the upper flip-flop output can be forced low by setting bit C12 of the control instruction to high.
- SR31, the most significant bit of the shift register, is selected to transfer the contents of the panel interface shift register to the breakpoint panel shift register to drive the data display LEDs.
- The output of the buffer shift register (left shift serial input data) is required for displaying incoming codes and data during the input operation.

Hexadecimal-to-ASCII Converter

Each hexadecimal digit, which consists of the four most significant bits SR31 through SR28 of the shift register, must be converted into ASCII code before transmitting out to remote devices via the UART. The conversion is accomplished using both hardware and controlware. Table 4-3 shows the ASCII code of the corresponding hexadecimal numbers.

First SR31 through SR28 are decoded to generate the less than or equal to 9_{10} (LE9) condition:

$$LE9 = (\overline{SR29} \cdot SR31) + (SR30 \cdot \overline{SR31})$$

TABLE 4-3. CORRESPONDING HEXADECIMAL NUMBERS AND ASCII CODE

Hexadecimal Number (SR31 - SR28)	ASCII Code						
	6	5	4	3	2	1	0
0 0 0 0	0	1	1	0	0	0	0
0 0 0 1	0	1	1	0	0	0	1
0 0 1 0	0	1	1	0	0	1	0
0 0 1 1	0	1	1	0	0	1	1
0 1 0 0	0	1	1	0	1	0	0
0 1 0 1	0	1	1	0	1	0	1
0 1 1 0	0	1	1	0	1	1	0
0 1 1 1	0	1	1	0	1	1	1
1 0 0 0	0	1	1	1	0	0	0
1 0 0 1	0	1	1	1	0	0	1
1 0 1 0	1	0	0	0	0	0	1
1 0 1 1	1	0	0	0	0	1	0
1 1 0 0	1	0	0	0	0	1	1
1 1 0 1	1	0	0	0	1	0	0
1 1 1 0	1	0	0	0	1	0	1
1 1 1 1	<u>1</u>	<u>0</u>	<u>0</u>	0	1	1	0
	C06 - C04						

The LE9 condition is checked by controlware. If the LE9 condition is true, the controlware sets bits 4 through 6 of the ASCII code to 011 via three bits, C04 through C06 of the control instruction. Bits 0 through 3 of ASCII remain the same as SR28 through SR31. If the hexadecimal digit is greater than 9, the controlware sets bits 4 through 6 of ASCII code to 100 via bits C04 through C06 of the control instruction. Bits 0 through 3 of ASCII code are converted by hardware from SR28 through SR31 according to the following equations:

- ASCII code Bit 0 = $\overline{\text{SR28}}$
- ASCII code Bit 1 = $\overline{\text{SR28} + \text{SR29}}$
- ASCII code Bit 2 = $\text{SR30} \cdot (\text{SR29} + \text{SR28})$
- ASCII code Bit 3 = Low

Transmitter Input Data Selector

This four-bit selector consists of multiplexers E4 and F2 that select one of four data sources for the UART. The sources are selected by bits C12 and C13 of the control instruction as follows:

C13	C12	Selected Data
0	0	Bits 0 through 4 of the ASCII code correspond to greater than 9 ₁₀ (GR9) hexadecimal digit.
0	1	Bits 0 through 4 of the ASCII code correspond to less than or equal to 9 ₁₀ (LE9) hexadecimal digit.
1	0	Four least significant bits (C00 through C03) of the control instruction
1	1	Control character code CC0 through CC2

Bits C04 through C06 of the control instruction combine with one of the above four data bits to form the complete seven-bit ASCII code applied to the transmitter register data input of the UART. If the control character code is selected, the controlware sets bits C04 through C06 to 100 to generate the proper ASCII code for H, I, J, K, and L characters. If select signals C13 and C12 equal 10, the least significant four bits, C00 through C03, of the control instruction are sent out to the programmers console to perform specific operations, such as carriage return, space, line feed, or type an asterisk as indicated by controlware.

Universal Asynchronous Receiver/Transmitter

The detailed block diagram, pin designations, and descriptions of the UART are included in Appendix B. Only a brief description of the UART and how it is used in this breakpoint controller module is given here.

The UART has the capability of simultaneously converting the asynchronous serial binary character to a parallel format (receiver) and parallel characters to serial, asynchronous output (transmitter) with start, parity, and stop bits added or verified.

Since the UART is not required during the time the breakpoint panel is being used (remote/local switch panel is at LOCAL position), the UART is disabled by the LOCAL + MC signal that is derived from the panel switch PNL0CAL signal.

An active high master clear signal also clears the UART. To transmit the character out to the remote devices, the controlware generates the TBRLOAD signal at pin 23 to load the corresponding seven-bit ASCII code into the transmit buffer register. This data is shifted out serially at pin 25 (TRO) via the UART internal transmit register at the rate determined by the clock (PNUARTCLK) at pin 40. Note that the clock frequency is 16 times the desired transmitter shift rate. The transmitter output data is transmitted out to the remote devices via the 20 mA current loop driver for I/O-TTY, an RS232 driver (for the conversational display terminal), and a standard TTL level driver.

The receiver portion of the UART asynchronously receives the serial input data from remote devices at pin 20 (RRI) and converts them into parallel format at the rate determined by the clock, PNUARTCLK. The receiver clock frequency is also 16 times the desired receiver shift rate. The serial data from remote devices is converted to TTL logic level data by a 20 mA current loop receiver, an RS232 receiver, or a standard TTL receiver before being applied to the UART receiver input. Whenever one complete character has been received and assembled, the data ready (RDR) line at pin 19 goes high. The controlware resets the data ready line by generating a DRRESET signal at pin 18 when the received character has been completely processed. Since the data received from remote devices is in ASCII code, it must be converted into hexadecimal for manipulation by the breakpoint controller and the processor.

ASCII-to-Hexadecimal Converter

The conversion from ASCII to hexadecimal is accomplished via the preprogrammed programmable read-only memory (PROM) D1. The conversion is limited to those legal characters shown in table 4-4. The four least significant bits, RR0 through RR3, and the most significant bit RR6 of ASCII code from the UART receiver data outputs are used to address one of 32 code locations. The selected location contains the hexadecimal code corresponding to the ASCII code of the character received.

The characters 0 through F are indicated by bit 7 of the PROM output being a 1. Similarly, the terminators (:), @, and G are indicated by bit 6 of PROM output being 1. These conditions are then used by controlware to determine if a hexadecimal digit, a terminator, or a control character has been received. A code is legal only if bits RR6 through RR4 of the ASCII code are equal to 011 or 100 or if bit 8 of PROM output is a 1.

Remote/Local Data Selector

The 2-to-1 multiplexer C1 selects either the hexadecimal data from the output of the ASCII to hexadecimal converter or the breakpoint panel switches in accordance with the state of the REMOTE signal. The REMOTE signal is high when the remote/local switch of the breakpoint panel is in the REMOTE position or during a deadstart operation (SM204 is low). The output of C1 can be loaded into the shift buffer register and/or to the control character register as determined by controlware. Similarly, the 2-to-1 multiplexer D4 selects the conditions such as the hexadecimal digit receiver (HEXRCVD), terminator code received (TERMCODE), data ready (DRDY) or RESTART for remote or local operation.

Shift Buffer Register

The shift buffer register consists of one four-bit universal bi-directional shift register, C2, that receives its input from the remote/local data selector C1. The control signals of the shift buffer register are bits C17 and C16 of the control instruction, which control the parallel loading and shift operations. The main function of the shift buffer register includes:

- Temporarily store the incoming four-bit data before shifting it into the breakpoint controller face shift register during input operation.
- During the J control command that modifies the FCR in digit mode, the shift buffer register contains the value that the FCR digit is to assume.
- The shift buffer register can also be shifted left to the breakpoint panel for display.

Output pin 15 of the shift buffer register is connected back to its left shift serial data input pin 7 to perform end-around shift left operation which saves the original data.

Control Character Register

When the character entered determines that the controlware is a control character instead of a hexadecimal digit or a terminator, the three least significant bits from the output of the remote/local data selector are loaded into the quadruple D-type flip-flop C3 by the controlware-generated

TABLE 4-4. ASCII-TO-HEXADECIMAL CONVERSION

Address RR6 and ¹⁶ RR3-RR0	ASCII Code (RR6-RR0)	Characters	PROM D1 Output (Bits 8-1)
00	0 1 1 0 0 0 0	0	C0
01	0 1 1 0 0 0 1	1	C1
02	0 1 1 0 0 1 0	2	C2
03	0 1 1 0 0 1 1	3	C3
04	0 1 1 0 1 0 0	4	C4
05	0 1 1 0 1 0 1	5	C5
06	0 1 1 0 1 1 0	6	C6
07	0 1 1 0 1 1 1	7	C7
08	0 1 1 1 0 0 0	8	C8
09	0 1 1 1 0 0 1	9	C9
0A	0 1 1 1 0 1 0	: Terminator	A0
0B	Illegal code		00
0C	Illegal code		00
0D	Illegal code		00
0E	Illegal code		00
0F	0 1 1 1 1 1 1	? Master clear	90
10	1 0 0 0 0 0 0	@ Terminator	A0
11	1 0 0 0 0 0 1	A	CA
12	1 0 0 0 0 1 0	B	CB
13	1 0 0 0 0 1 1	C	CC
14	1 0 0 0 1 0 0	D	CD
15	1 0 0 0 1 0 1	E	CE
16	1 0 0 0 1 1 0	F	CF
17	1 0 0 0 1 1 1	G Terminator	A0
18	1 0 0 1 0 0 0	H	80
19	1 0 0 1 0 0 1	I	81
1A	1 0 0 1 0 1 0	J	82
1B	1 0 0 1 0 1 1	K	83
1C	1 0 0 1 1 0 0	L	84
1D	1 0 0 1 1 0 1	M [†]	85
1E	1 0 0 1 1 1 0	N [†]	86
1F	Illegal code		00

[†] M and N control characters are undefined.

clock, STRBCCR. The output of the control character register is decoded by decoder D5 as follows:

CC2	CC1	CC0	Control Characters
0	0	0	H
0	0	1	I
0	1	0	J
0	1	1	K
1	0	0	L
1	0	1	M (not used)
1	1	0	Open
1	1	1	Open

The output of the control character register is either gated out to the breakpoint panel for driving CONTROL CODE LED displays or transmitted to remote devices via the UART.

BREAKPOINT CONTROLLER CONTROL HARDWARE

The breakpoint controller operations are controlled by a control instruction program (controlware) that is stored in PROMs. The breakpoint controller control hardware is required mainly to execute the control instruction program and to perform other control functions as follows:

- Address the PROMs to select one of 96 control instructions.
- Decode the control instruction to generate the control signals for selectors, shift registers, etc.
- Generate micro instructions to be executed by the CPU.
- Implement miscellaneous logic circuits such as breakpoint control, controller timing chain, baud rate generator, etc.

Control Instruction Format

The panel interface controlware is contained in nine 32-bit PROMs which form 96-bit by 24-bit control instructions. Controlware, in this manual, is defined as a set of 24-bit control instructions that control and sequence controller operations. The control instruction is shown in figure 4-5.

The P0 field specifies the jump mode:

Bit C23 = 1 Jump if condition is true
 Bit C23 = 0 Jump if condition is false

The jump condition is selected by the P1 field and the jump address is specified by the P7 field.

The P1 field consists of five bits, C18 through C22, that select a jump condition to be tested according to jump mode P0.

The P2 field consists of two bits, C16 and C17. The P2 field controls two separate functions:

- It selects one of four types of breakpoint controller generated micro instructions, referred to as LOC0 through LOC3. Refer to the paragraph on Breakpoint Controller Micro Instructions below.
- It selects one of four modes of operation for the shift buffer register:

C17	C16	Mode of Operation
0	0	Inhibit clock (NOP)
0	1	Right shift
1	0	Left shift
1	1	Parallel load

The P3 field consists of bits C14 and C15, which select one of four modes of operation for the shift register.

The P4 field consists of bits C12 and C13, which control four separate functions:

- It selects parallel input data for loading into the shift register. Either FCR00 through FCR31, MIR00 through MIR31, or breakpoint/micro-memory address can be selected. Refer to the Shift Register Input Selector description above for details.
- It selects input data for the UART. Refer to the paragraph on the Transmitter Input Data Selector above.
- It selects serial left shift input data for the breakpoint panel interface shift register and breakpoint panel shift register. Refer to the paragraph on the Left Shift Input Selector above.
- It selects the micro-memory upper/lower 16 bits and also forces display 1. Display 1 is forced by controlware to generate the appropriate micro instructions:

C13	C12	Functions
0	0	Force display 1 and set MML at STRBMMX
0	1	Force display 1 and set MMU at STRBMMX
1	0	Set MML at STRBMMX
1	1	Set MMU at STRBMMX

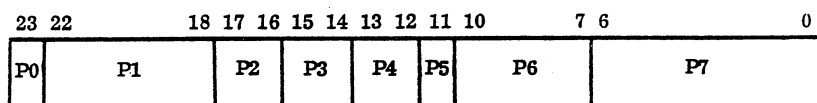


Figure 4-5. Control Instruction Format

The P5 field consists of only one bit, C11, which enables/disables the shift counter. The shift counter keeps track of how many times the shift register is shifted:

- C11 = High Enable the shift counter
- C11 = Low Disable the shift counter

The P6 field consists of four bits, C7 through C10, that generate the strobe signals for registers, counters, etc.

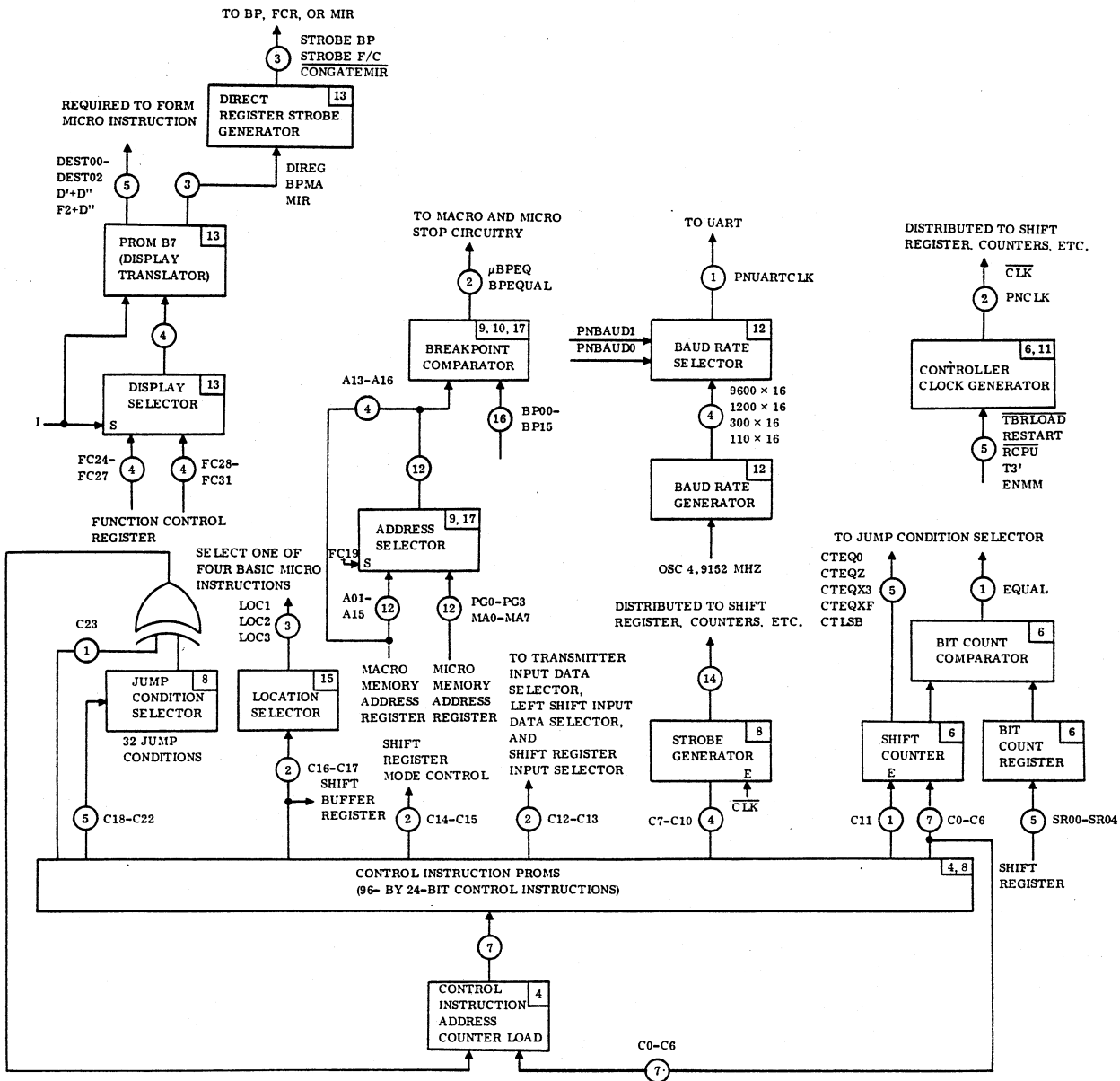
The P7 field, also called the address field, consists of seven bits, C0 through C6. The P7 field can be used as follows:

- The jump address is the address of the next control instruction if the jump condition is met.

- The shift counter load value allows the shift counter to be preset to any desired value.
- The ASCII transmit character is the ASCII code of any desired character, such as the asterisk, line feed, carriage return, etc., and can be set into the P7 field and transmitted out to the remote devices via the UART.

Control Block Diagram Logic Description

Figure 4-6 shows the control block diagram of the break-point controller.



NOTE: THE NUMBERS IN UPPER RIGHT CORNER OF EACH BLOCK CORRESPOND TO THE LOGIC DIAGRAM SHEET NUMBERS

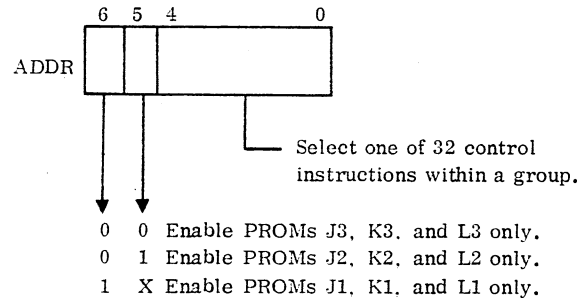
0203

Figure 4-6. Control Block Diagram

Control Instruction PROMs

The PROMs contain a total of 96 control instructions that, when executed, generate all necessary control signals for controller operation. Each PROM is organized as 32 words with eight bits per word. Therefore, a total of nine PROMs are required to form 96- by 24-bit control instructions. Figure 4-7 shows the arrangement of the PROMs. The output of PROMs J1, J2, and J3 are OR-wired to form control instruction bits C16 through C23. Similarly, the control instruction bits C8 through C15 and C0 through C7 are formed from the output of PROMs K1, K2, and K3 and L1, L2, and L3, respectively. The 96 control instructions are divided into three groups of 32 control instructions. PROMs J3, K3, and L3 form the first group of 32 control instructions, address 0₁₆ through 1F₁₆. The second group of 32 control instructions address 20₁₆ through 3F₁₆, are formed by PROMs J2, K2, and L2. The third group of 32 control instructions, address 40₁₆ through 5F₁₆, are formed by PROMs J1, K1, and L1. Table 4-5 lists the content of PROMs. The control instruction is selected by the control instruction address counter.

To select one of 96 control instructions, seven address bits, ADDR0 through ADDR6, are required. The lower five address bits, ADDR0 through ADDR4, select one of 32 control instructions within a group. The other two bits select the 32 control instruction groups by enabling the desired PROM groups:



Control Instruction Address Counter

The control instruction address counter generates address bits ADDR0 through ADDR6 to select the next control instruction. The control instruction address counter consists of two 4-bit synchronous counters, J5 and K5. The counter is cleared by the master clear signal to allow the controller to be executed from the beginning. The address counter input is connected to the output of the PROM, bits C0 through C6. If the jump condition is met, the address counter is loaded with jump addresses C0 through C6 of the previous control instruction. If the jump condition is not met, the address counter is incremented by 1 to select the next sequential control instruction.

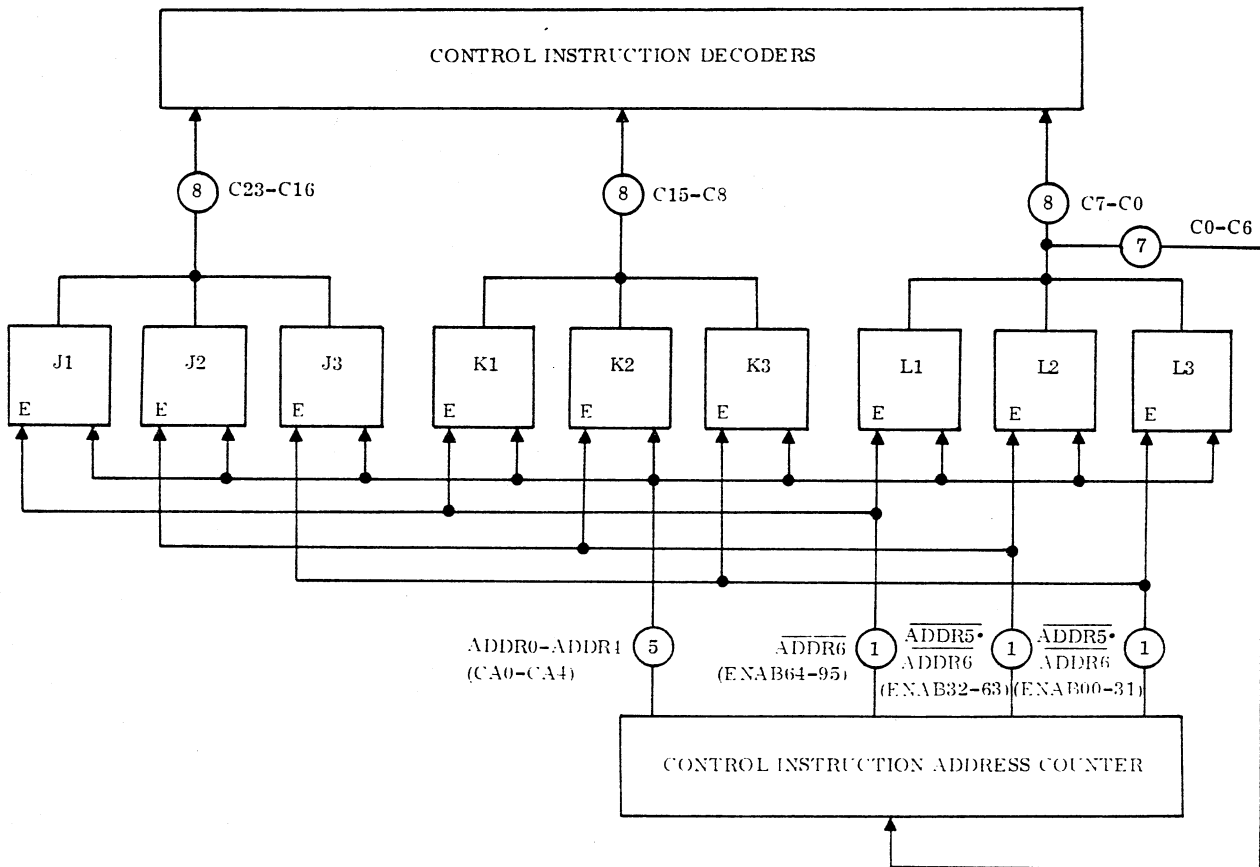


Figure 4-7. PROMs Configuration to Form 96 x 24 Bit Control Instruction

TABLE 4-5. CONTROL INSTRUCTION FIELD
PROGRAMMABLE READ-ONLY MEMORY

Location Address	Control Instruction Bits C23-C00								
	Word 64-95			Word 32-63			Word 0-31		
	J1	K1	L1	J2	K2	L2	J3	K3	L3
00	0C	00	42	58	1C	20	90	00	06
01	80	05	91	28	00	23	E8	00	00
02	FC	07	50	80	F3	28	30	36	12
03	80	03	44	2C	00	11	32	15	1E
04	44	A8	44	80	E3	28	7C	00	15
05	60	03	43	4C	03	28	80	00	11
06	4C	00	43	EC	00	28	84	00	08
07	82	26	43	00	07	90	80	01	08
08	44	A8	48	84	00	2B	C8	00	10
09	5C	03	4C	40	98	29	BC	00	0F
0A	80	16	43	80	06	BA	D3	00	0E
0B	81	36	4C	A0	00	2A	FC	00	0D
0C	E6	00	4E	00	21	0D	80	05	8F
0D	82	00	4E	68	00	2F	80	02	0F
0E	4C	00	53	00	21	0A	56	B8	0E
0F	80	00	52	3C	00	31	80	01	BB
10	4C	00	52	00	21	2A	3C	01	82
11	00	07	90	00	31	40	80	D3	28
12	40	A8	D2	00	21	20	68	00	11
13	5C	36	57	24	06	B6	FC	00	1D
14	80	16	55	00	11	30	58	5D	14
15	81	36	56	80	00	37	80	D3	16
16	83	03	57	00	01	40	9C	00	18
17	83	16	58	54	98	37	80	98	1B
18	80	00	59	74	00	32	B0	00	1A
19	44	A8	D9	00	21	20	82	B8	1B
1A	4C	03	5E	80	03	3B	80	88	1B
1B	60	06	5E	84	00	00	74	00	16
1C	83	16	5D	90	00	3C	80	04	11
1D	80	00	5E	80	01	3E	80	05	27
1E	44	A8	DE	E8	00	00	6C	00	40
1F	81	03	A5	80	15	00	FC	00	21

Control Instruction Decode

The control instruction is decoded to generate the control signals required for breakpoint controller operations.

P1 field decode - The P1 field of the control instruction selects a jump condition. Table 4-6 lists all the jump conditions and their descriptions. The P1 field decoder

consists of two 16-to-1 multiplexers, E5 and G3. Bits C18 through C21 of the control instruction select one of 16 conditions for each multiplexer, and bit C22 enables/disables the multiplexer as follows:

C22 = Low Enables multiplexer G3 and disables multiplexer E5

C22 = High Enables multiplexer E5 and disables multiplexer G3

The selected jump condition is exclusively ORed with bit C23 of the control instruction to determine if a jump condition is met. A jump condition is met if:

- The condition is true and jump-if-true mode is selected (C23 = high).
- The condition is false and jump-if-false mode is selected (C23 = low).

If the jump condition is met, PNTEST4 at L7-11 goes low to load the address field P7 (bits C0 through C6) of the control instruction into the control instruction address counter. This allows the internal processor to select the control instructions required to perform a specific operation.

P2, P3, P4, and P5 field decode - These fields do not require much decoding, and therefore have been fully described in the paragraphs under Control Instruction Format above.

P6 field decode - The P6 field is decoded to generate the necessary strobe signals for registers and counters. Table 4-7 lists all the strobe signals and their functions. The P6 field consists of two 8-to-1 decoders K6 and J6. If bit C10 is low, K6 is enabled and J6 is disabled; the opposite is true if C10 is high. Note that the CLK signal is connected to allow input pin 5 to enable these decodes only when the CLK signal is low. This causes the strobe signals to have the same pulse width as the CLK signal. In addition to the strobe signals listed in table 4-7, the P6 field is also decoded by decoder H6.

C10C7	Signal Names	Function
0 0 0 1	\overline{XENAB}	Enable S300 to be MIR31 of shift X to left micro instruction (LOC0) if X is operator selected register.
0 1 1 0	\overline{RDSTRB}	Form read macro-memory code in S field of read macro-memory micro instruction.
0 1 1 1	\overline{EXIT}	Generate last micro instruction with M field equal to 00 to return control to the CPU.

The above signals are required in generating the micro instructions. Refer to Controller Breakpoint Micro Instructions below.

P7 field decode - The P7 field does not require any decoding. Refer to control instruction format for P7 field usage.

TABLE 4-6. JUMP CONDITIONS

C22-C18	Conditions/ Mnemonics	Descriptions
0 0 0 0 0	High	Unconditional true
0 0 0 0 1	Remote	Remote/local switch on panel is set to REMOTE.
0 0 0 1 0	N/C	Not used
0 0 0 1 1	Running	Micro running and micro mode
0 0 1 0 0	DRDY	Data is ready from the UART, or a breakpoint panel key is depressed.
0 0 1 0 1	COND.5	PINNED-out sense (not used)
0 0 1 1 0	N/C	Not used
0 0 1 1 1	EQUAL	Bit count register equals shift counter
0 1 0 0 0	FC15	Function control register bit 15 (suppress console transmit)
0 1 0 0 1	LE9	Transmitted hexadecimal digit is less than or equal to 9.
0 1 0 1 0	BPMA	BP-P/MA is selected.
0 1 0 1 1	MIR	MIR is selected.
0 1 1 0 0	J	Not J control character
0 1 1 0 1	H + I	H or I control character
0 1 1 1 0	N/C	Not used
0 1 1 1 1	ERROR	Error flip-flop is set.
1 0 0 0 0	CTEQXF	Shift counter equals 31.
1 0 0 0 1	CTEQXF	Shift counter equals 15.
1 0 0 1 0	TERMCODE	Terminate code received (@, G, or :).
1 0 0 1 1	CPU 16	16-bit CPU
1 0 1 0 0	HEXRCVD	Hexadecimal digit received
1 0 1 0 1	CTEQX3	Least significant two bits of shift counter equal 11.
1 0 1 1 0	CTLSB	Least significant bit of shift counter
1 0 1 1 1	MEM	Macro memory is selected.
1 1 0 0 0	μMEM	Micro memory is selected.
1 1 0 0 1	F2	File 2 is not selected.
1 1 0 1 0	NOT AUTO	Busy or not auto-display
1 1 0 1 1	DIREG	Direct registers (BP, FCR, MIR) selected
1 1 1 0 0	N/C	Not used
1 1 1 0 1	CTEQ0	Lower five bits, 0 through 4, of shift counter equal zero.
1 1 1 1 0	N/C	Not used
1 1 1 1 1	CTEQZ	Lower six bits, 0 through 5, of shift counter equal zero.

TABLE 4-7. STROBE SIGNALS

C10-C7	Strobe Signals	Location	Functions
0 0 0 0	N/C	K6-15	Not used
0 0 0 1	Test III (CLRSM214)	K6-14	Clear status mode bit 214.
0 0 1 0	TBRLOAD	K6-13	Load transmit buffer register of the UART.
0 0 1 1	DRRESET	K6-12	Reset data ready line of the UART and set busy flip-flop high.
0 1 0 0	STRBCCR	K6-11	Strobe control character register.
0 1 0 1	CLRSR	K6-10	Clear shift register.
0 1 1 0	CLRCNT	K6-9	Clear shift counter; strobe bit count register.
0 1 1 1	EXIT	K6-7	Set M field of last micro instruction to 00 to return control to the CPU.
1 0 0 0	SSREG	J6-15	Generate strobe signals for selected direct register MIR, BP, or FCR.
1 0 0 1	N/C	J6-14	Not used
1 0 1 0	STRBUL	J6-13	Strobe upper/lower flip-flop K8.
1 0 1 1	SET ERROR	J6-12	Set error flip-flop high.
1 1 0 0	STRBMMX	J6-11	Strobe micro-memory upper/lower flip-flop.
1 1 0 1	CLRBUSY	J6-10	Clear busy flip-flop and error flip-flop.
1 1 1 0	SETRCPU	J6-9	Set RCPU to take over control of CPU; also disable controller free-running clock.
1 1 1 1	LOADCT	J6-7	Load shift counter with address field P7, bits C0-C6.

Breakpoint Controller Micro Instructions[†]

After taking control of the processor, the breakpoint controller develops micro instructions and forces the processor to execute these instructions. These instructions are developed and loaded into the micro instruction register via the micro-memory three-state bus. There are four

[†]The descriptions of these micro instructions are typical for a 16-bit machine.

^{††}For a 32-bit machine, the 32-bit X register and the 32-bit panel interface shift register can be considered as one 64-bit shift register.

basic micro instructions: LOC0, LOC1, LOC2, and LOC3. They are selected by the P2 field (bits C17 and C16) of the control instruction. A combination of these micro instructions is normally required to perform breakpoint controller operations. Descriptions of the basic controller micro instructions are as follows:

LOC0 (Shift ALU X Register to Left) - This instruction shifts the ALU X register to the left one bit and is used to transfer data from the controller shift register to the X register and vice versa. The 16-bit X register and the 32-bit panel interface shift register can be considered as one 48-bit shift register.^{††} The most significant bits are contained in the X register, and the least significant bits are contained in the controller shift register.

The micro instruction LOC0 contains the following:

- F = 01010 (Select B input)
- A = Don't care
- B = 011 (X register is the B input)
- D = 110 (X register is the destination register)
- S = 0000 (NOP)
- C = 111010Y (Shift destination X register to the left one bit, and enter Y in to the least significant bit position of X register.)

To transfer the contents of the X register to the lower 16 bits of the shift register and the most significant 16 bits of the shift register to the X register, the LOC0 micro instruction is executed 16 times with SR31 of the shift register selected as the Y bit of the C command. Simultaneously, the shift register is shifted left 16 times with the most significant bit of the X register (XMSB) as left shift serial input data. Since the X register is used by the breakpoint controller as a temporary storage register, the original contents of the X register must be restored at the end of the operation if X is not the selected register. The controlware generates the XENAB signal from the P6 field of the control instruction to allow either S300 (X is the selected register) or SR31 (X is not the selected register) to be MIR31 of the shift X to left micro instruction.

LOC1 (Increment micro or macro-memory pointer register by 1) - During enter/display macro memory, this micro instruction is used to increment the selected P or A register of display 1 by 1 and store it back into the selected register and macro-memory address buffer. The I register is not incremented even though this micro instruction is executed.

During enter/display micro memory, the K register is incremented by the INCK command in the C field of this micro instruction. This LOC1 micro instruction allows the macro or micro memory to be sequentially entered/displayed without re-entering the address; that is, the address is required to be input only once.

F = 11010 (ADD+) - This F code is used to increment the contents of the display 1 selected register by 1.

F = 01111 (Select A input) – This F code is used mainly to transfer the contents of the display 1 selected register to macro memory address buffer register AB.

A = The display 1 register previously selected by the operator via FCR bits FC24 through FC27

B = 001 and MIR28-29 = 11 (B source is all zeros)

D = DEST00 through DEST02 – These bits are derived from the output of PROM B7, which translates the display code into the corresponding destination register code in the D field. Refer to the description of Control Instruction Format above for PROM B7.

S = 000 (NOP)

C = Bits 28 and 29 must be 1 1 to set the B input to all zeros when required.

C = 1000101 and bit MIR19 = 0 (C' code) – Increment the K register by 1.

LOC2 (X to selected register) – This micro instruction transfers the contents of the X register to an operator selected register. During the entering operation, the input data is first shifted from the shift register to the X register. The LOC2 micro instruction is then used to store the contents of the X register in the operator selected register.

F = 01010 (Select B input)

A = 011 (X register as A input) A source is not important, since the F field always selects the B input.

B = 011 (X register is B input)

D = Operator selected destination register DEST00 through DEST02. Refer to the PROM B7 descriptions under Control Instruction Format above for details of DEST00 through DEST02.

S = 0000 (NOP) – This indicates that the destination registers are those in the D code; e.g., ALU registers P, I, Q, F1, A, X, and F.

= 1001 (D' code) – The D' code indicates that the destination register is either a micro memory or one of status mode registers M1, M2, SM1, or SM2.

= 1011 (D" code) – The D" code indicates that the destination register is one of the double-precision registers, A*, Q*, and X*.

= 0100 (WRITE) – This code is generated when writing into macro memory is required.

= 0110 (F2WR) – This code is generated when writing data into the file 2 registers.

C = TK/O (Transform K register at selector S8 position 0) The C field must contain 0110000 and bit MIR19 = 1 (C" code). This TK/O transform allows the lowest eight bits, eight through 15, of selector S2 to be loaded into the K register.

= TN/O (Transform N register at selector S8 position 0) – The C field must contain 1000000 and bit MIR19 = 1 (C" code). This TN/O transform allows the lowest eight bits, eight through 15, of selector S2 to be loaded into the N register. Note that

these instructions are the only means of loading the K and N registers.

LOC3 (Selected register to X) – During the display operation, this micro instruction is used to transfer the contents of the operator selected register to the X register. The X register is then shifted out serially to the breakpoint controller via selector S3 and displayed.

F = 01111 (A input is selected) – This F code is used when the selected register is in display 1.

= 01010 (B input is selected) – -- This F code is used when the selected register is in display 0. This F code is also required for read micro-memory operation.

A = Operator selected register in display 1 via FCR bits FC24 through FC26

B = Operator selected register in display 0 via FCR bits FC28 through FC30

D = 110 (X register is the destination register)

= 000 (NOP) – Required for read micro-memory instruction

S = 0000 (NOP) – This S code indicates that the A and B fields of the micro instruction contain the A and B codes, respectively (not A' or B' code). In other words, the operator selected register of display 1 or display 0 is a register in A or B code.

= 0111 (A' code) – The selected register of display 1 is a register in A' code; that is, SM1, SM2, M1, M2, A*, Q*, and X*.

= 1000 (B' code) – This S code is used when micro memory is the selected register and read micro-memory operation is required.

= 0011 (READ) – This S code is used to read macro memory.

C = Don't care

Table 4-8 summarizes the four basic types of micro instructions and all variations in each field of micro instruction.

The following are descriptions of important hardware required to generate the micro instructions.

MIR31 Generation – As indicated in figure 4-8, MIR31 of the shift X to left (LOC0) micro instruction is used to transfer data from the breakpoint controller shift register to the X register if MIR31 is substituted by the most significant bit, SR31, of the shift register.

If the MIR31 is substituted by the most significant bit of the X register (S300), the X register is shifted with end-around. This is required when X is the operator selected destination register.

The MIR31 is selected by 4-to-1 multiplexer H7 as follows:

C17	(X) (FETCH)	MIR31
0	0	LOC1 + SR31
0	1	S300
1	0	Low
1	1	Low

TABLE 4-8. DISPLAY CODE PROM B7

Display	Address	Selected Registers	B7 PROM Output	PROM Output (in Hexadecimal)
Display 0 (L Function)	0 0 0 0 0	F2	0 0 0 0 1 1 1 1	0F
	0 0 0 0 1	N	0 0 0 0 0 0 0 0	00
	0 0 0 1 0	K	0 0 0 0 0 0 0 0	00
	0 0 0 1 1	X	0 0 0 0 0 1 1 0	06
	0 0 1 0 0	Q	0 0 0 0 0 0 1 1	03
	0 0 1 0 1	F	0 0 0 0 0 1 1 1	07
	0 0 1 1 0	F1	0 0 0 0 0 1 0 0	04
	0 0 1 1 1	MEM	0 0 0 0 0 0 0 0	00
	0 1 0 0 0	Not Used	0 0 0 0 0 0 0 0	00
	0 1 0 0 1	RTJ	0 0 0 0 0 0 0 0	00
	0 1 0 1 0	Not Used	0 0 0 0 0 0 0 0	00
	0 1 0 1 1	Not Used	0 0 0 0 0 0 0 0	00
	0 1 1 0 0	MM	0 0 0 1 0 0 1 0	12
	0 1 1 0 1	Not Used	0 0 0 1 0 0 1 1	13
0 1 1 1 0	Not Used	0 0 0 0 0 0 0 0	00	
0 1 1 1 1	Not used	0 0 0 0 0 0 0 0	00	
Display 1 (K Function)	1 0 0 0 0	FCR	1 0 0 0 0 0 0 0	80
	1 0 0 0 1	P	0 0 0 0 0 0 0 1	01
	1 0 0 1 0	I	0 0 0 0 0 0 1 0	02
	1 0 0 1 1	Not Used	0 0 0 0 0 0 0 0	00
	1 0 1 0 0	A	0 0 0 0 0 1 0 1	05
	1 0 1 0 1	MIR	1 0 1 0 0 0 0 0	A0
	1 0 1 1 0	BP-P/MA	1 1 0 0 0 0 0 0	C0
	1 0 1 1 1	BP-P/MA Display only	1 1 1 0 0 0 0 0	E0
	1 1 0 0 0	SM1	0 0 0 1 0 1 0 1	15
	1 1 0 0 1	M1	0 0 0 1 0 1 0 0	14
	1 1 0 1 0	SM2	0 0 0 1 0 1 1 1	17
	1 1 0 1 1	M2	0 0 0 1 0 1 1 0	16
	1 1 1 0 0	Not Used	0 0 0 0 0 0 0 0	00
	1 1 1 0 1	A*	0 0 0 1 1 1 0 1	1D
1 1 1 1 0	X*	0 0 0 1 1 1 1 0	1E	
1 1 1 1 1	Q*	0 0 0 1 1 1 1 1	1F	

Bit C17 selects the type of micro instruction.

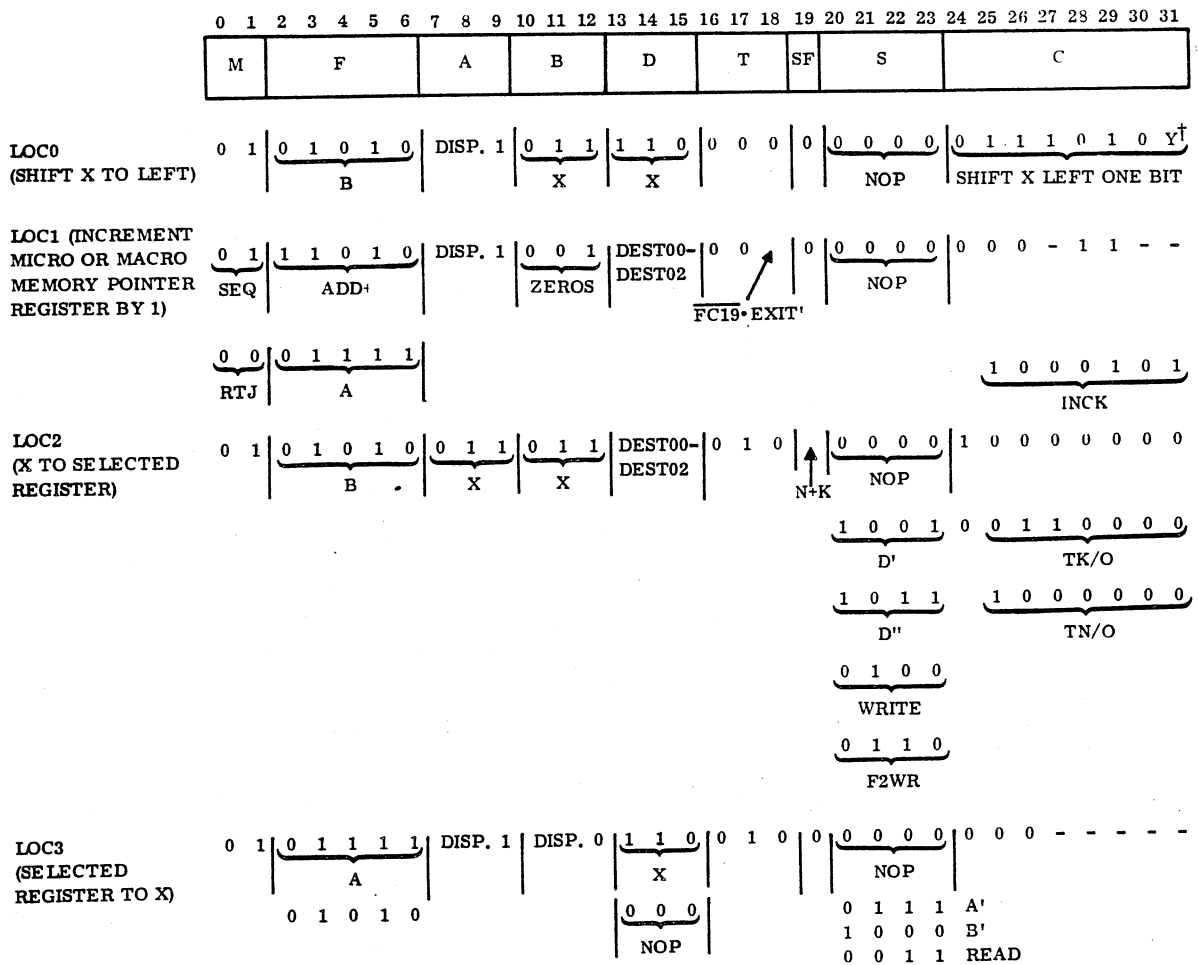
C17 = Low LOC0 or LOC1
C17 = High LOC2 or LOC3

(X)-(XENAB) selects the proper source for MIR31 depending upon if the X register is the selected destination register or not.

Display Code Translator - Table 4-2 shows the display code definition for display 1 and display 0. The selected register from either display 1 or display 0 must be translated into the

corresponding code for the micro-instruction D field. The display 1 and display 0 codes consist of FCR bits FC24 through FC27 and FC28 through FC31, respectively. The display 1 and display 0 codes are first selected by 2-to-1 multiplexer B8, which is enabled only when the L or K control function is entered. Display 1 or display 0 is selected via the L signal decoded from the control character decoder.

L = Low Selects display 1, FC24 through FC27
L = High Selects display 0, FC28 through FC31



† WHEN ENABLED BY XENAB FROM THE CONTROLWARE, Y IS EITHER SR31 (IF X IS NOT THE SELECTED REGISTER) OR S300 (IF X IS THE SELECTED REGISTER).

0215

Figure 4-8. Breakpoint Controller Micro Instructions

Display 1 can be force-selected if bit C13 of the control instruction is low and the STRBMMX strobe signal is generated by controlware. The actual display code translation is performed by the 32- by eight-bit PROM B7. Table 4-8 lists the contents of PROM B7. The format of PROM B7 is shown in figure 4-9.

Bits 0 through 2 (DEST00 through DEST02) are used to form the destination (D) field of the micro instruction. Bits 3 and 4 (D' + D'' and F2 + D'') are used to modify the S field of the

generated micro instruction. Bits 5 through 7 are used to indicate which of the direct registers, BP-P/MA, MIR, or FCR, is selected and generate the strobe signals STROBE F/C, STROBE BP, or CONGATEMIR accordingly.

MIR Input Data Selector - The MIR input data selector consists of 2-to-1 multiplexer DM8123 and 4-to-1 multiplexer DM8214 with three-state outputs to interface directly with the micro-memory three-state bus. The selector is enabled only when ENMM=ECPI is low (processor

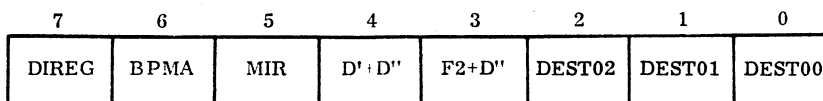


Figure 4-9. PROM B7 Format

micro memory is disabled). The selector selects either the outputs of the shift register or the breakpoint controller micro instructions to be loaded into MIR. The selector is also a part of hardware that generates the breakpoint controller micro instructions. Table 4-9 lists all the equations for each micro-instruction bit and the positions of the multiplexers that select the micro instruction type.

Shift Counter

This is a six-bit counter (J4 and K4) whose main function is to track how many times the shift register has been shifted. The shift counter consists of two four-bit synchronous counters that are enabled by bit C11 of the control instructions:

C11 = High Enables shift counter
C11 = Low Disables shift counter

The outputs of the counter are used to generate jump conditions. The controlware monitors the jump conditions listed below to determine if the desired number of shifts has been attained.

- CTEQX3 - The lower two bits of the counter equal 11_{16} . This jump condition is used to shift the shift register four times (one hexadecimal digit is shifted).
- CTEQXF - The lower four bits of the counter equal F_{16} . This jump condition is to be used to shift the shift register 16 or 32 times.
- CTEQ0 - The lower five bits of the shift counter equal zero.
- CTEQZ - All six bits of the shift counter equal zero.

The shift counter can be preset by the $\overline{\text{LOADCT}}$ signal to any desired value (0 through 31) via bits C0 through C5 of the control instruction. The shift counter can also be cleared by a master clear or by a $\overline{\text{CLR CNT}}$ signal from controlware.

Bit Count Register

This is a five-bit register (L10) that stores the lower five bits, SR00 through SR04, of the breakpoint controller shift register. The lower five bits of the shift register indicate the specific bits of the function control register that need to be modified. During the H or I control command, the bit count register is loaded with the bit position of the FCR bit to be cleared or set. For example in the H16: command, $16_{16} = 10110$, is loaded into the bit count register.

The FCR is shifted left via the shift register, and; simultaneously, the output of the bit count register is compared with the lower five bits of the shift counter by comparator M9. When they are equal, the controlware sets or clears that FCR bit depending upon the I or H control function. During the J control command that replaces the contents of the FCR in digit mode, the digit position is first converted into the bit position by shifting to the right two places before being loaded into the bit count register. The bit count register now indicates the first bit position of the digit to be modified. For example in the J11: command, $11_{16} = 00010001$ is shifted to the right two places before being loaded into the bit count register. The bit count register now contains 00100, which is the first bit position of digit 1. Only the upper three bits of the bit count register (SR02 through SR04) are compared to the upper three bits of the shift counter. The lower two bits of the bit count register are ignored and considered to be equal to the lower

TABLE 4-9. BREAKPOINT CONTROLLER MICRO INSTRUCTIONS

Micro-Memory Data Bits	RCPU = Low		RCPU = High	
MM00	SR31		Low	
MM01	SR30		$\overline{\text{EXIT}}$	
MM02	SR29		$\overline{\text{MEM}} \cdot \overline{\text{EXIT}}$	
MM03	SR28		RCPU = High	
MM04	SR27		$\overline{\text{MEM}} \cdot \overline{\text{EXIT}}$	$\text{LOC1} + \text{LOC3} \cdot \overline{\text{L}}$
MM05	SR26		RCPU = High	
MM06	SR25		$\overline{\text{MEM}} \cdot \overline{\text{EXIT}}$	$\text{LOC1} + \text{LOC3} \cdot \overline{\text{L}}$
MM07	SR24		$\text{FC26} \cdot \text{LOC2}$	
MM08	SR23		$\text{FC25} + \text{LOC2}$	
MM09	SR22		$\text{FC24} + \text{LOC2}$	
MM10	SR21		$\text{FC30} \cdot \text{LOC3}$	
MM11	SR20		$\text{LOC} \cdot \overline{\text{DIS01}} + \overline{\text{CT16}}^\dagger$	
MM16	SR15		Low	
MM17	SR14		C17	
MM18	SR13		$\overline{\text{FC19}} \cdot \overline{\text{EXIT}}$	
MM19	SR12		$\text{LOC2} \cdot (\text{N} + \text{K})$	
MM24	SR07		C17	
MM25	SR06		$\text{LOC2} \cdot \text{NREG} + \text{LOC1} \cdot (\mu\text{M} + \text{F1} + \text{F2}) + \text{C11}$	
MM26	SR05		$\text{C11} + \text{KREG} \cdot \text{LOC2}$	
MM27	SR04		$\text{C11} + \text{KREG} \cdot \text{LOC2}$	
MM28	SR03		$\text{LOC1} \cdot (\text{F2} + \text{MEM})$	
MM29	SR02		$\overline{\text{C17}}$	
MM30	SR01		Low	
MM31	SR00		INST. LSB	
Select Signals S1 and S0 = RCPU and FC ^{††}				
Micro Memory Data Bits	0 0	0 1	1 0 (LOC0 + LOC3)	1 1 (LOC1 + LOC2)
MM12 SR19	SR19	SR19	$\overline{\text{DIS00}} + \overline{\text{LOC3}}^{\dagger\dagger\dagger}$	$\overline{\text{DIS00}} + \overline{\text{LOC3}}^{\dagger\dagger\dagger}$
MM13 SR18	SR18	SR18	$\overline{\text{LOC3}} \cdot \mu\text{MEM}$	DEST02
MM14 SR17	SR17	SR17	$\overline{\text{LOC3}} \cdot \mu\text{MEM}$	DEST01
MM15 SR16	SR16	SR16	Low	DEST00
Select Signals S1 and S0 = C17·RCPU and RCPU· $\overline{\text{LOC2}}$				
Micro Memory Data Bits	0 0	0 1 (LOC0 or LOC1)	1 0 (LOC2)	1 1 (LOC3)
MM20 SR11	SR11	Low	$\text{D}' + \text{D}''$	$\text{FC31} \cdot \text{L}$
MM21 SR10	SR10	Low	$\text{F2} + \text{MEM}$	$\text{FC27} \cdot \overline{\text{L}}$
MM22 SR09	SR09	Low	$\text{F2} + \text{D}''$	$\text{FC27} \cdot \overline{\text{L}} + \text{RDSTRB}' \cdot \text{MEM}$
MM23 SR08	SR08	Low	$\text{D}' + \text{D}''$	$\text{RC27} \cdot \overline{\text{L}} + \text{RDSTRB}' \cdot \text{MEM}$
[†] $\overline{\text{DIS01}} = \overline{\text{FC29}} \cdot (\overline{\text{FC31}} \cdot \overline{\text{FC30}} \cdot \overline{\text{FC29}} \cdot \overline{\text{FC28}}) = \overline{\text{FC29}} \cdot \text{KREG}$ ^{††} $\text{FC} = \text{LOC1} + \text{LOC2}$ ^{†††} $\overline{\text{DIS00}} = \overline{\text{FC28}} + \overline{\text{KREG}} + \overline{\text{MML}}$				

two bits of the shift counter by forcing the input pin M5-3 to be high (since \bar{J} at H3-13 is low). This allows the first bit of the selected digit to be recognized, and the next four consecutive bits of FCR (one digit) to be modified once the upper three bits of the bit count register is equal to the upper three bits of the shift counter. Refer to the description of J11: operation under Typical Breakpoint Controller Operations below.

Breakpoint

There are two types of processor breakpoints: macro breakpoint and micro breakpoint. For macro breakpoint, the macro-memory address is compared with the contents of the breakpoint register; for micro breakpoint the micro-memory address is used. The macro-memory or micro-memory address is selected by 2-to-1 multiplexers C12, J13, and K11 according to FCR bit FC19 as follows:

- FC19 = High (micro BP) Select micro-memory address (PG0, PG3, MA0, MA7, and \bar{BTLW})
- FC19 = Low (macro BL) Select macro-memory address (A01 through A12)

The macro and micro breakpoint conditions are selected by FC16 and FC17 via 4-to-1 multiplexer J7:

FC17	FC16	Macro Breakpoint (J7-1)	Micro Breakpoint (J17-14)
0	0	Low: breakpoint not selected	Breakpoint not selected
0	1	SELGETMAK: instruction reference breakpoint	(\bar{BTLW} + BP12)· μ BPEQ
1	0	CPU WRITE: store operand breakpoint†	μ BPEQ
1	1	High: all reference breakpoint	μ BPEQ

The contents of the breakpoint register are compared to the macro- or micro-memory address by 4-bit magnitude comparators: C10, C11, H13, and K9. The comparator is enabled by the \bar{RCPU} at K9 3 when it is high. This indicates that the breakpoint operation is performed only when the breakpoint controller is not in control of the processing element. The μ BPEQ or BPEQUAL signal is generated when the micro-memory address is equal to the breakpoint register.

For macro breakpoint, if the breakpoint register is equal to the macro-memory address and the breakpoint select conditions are met, the processor is stopped. If bit FC18 of the FCR is set, it causes the SETSM104 signals at M6-2 to generate a macro breakpoint interrupt (SM104) instead of a stop. For micro breakpoint, if the micro-memory address PG0 through PG3, MA0 through MA7, and the upper/lower micro-instruction selection signal \bar{BTLW} are equal to breakpoint register bits BP00 through BP12 and the combination of FCR bits 17 and 16 is not zero, then a micro stop occurs (MICROSTOP at M8-8 goes to low). If FCR bit FC17 is set, then comparison of breakpoint bit 12 and the upper/lower micro-instruction selection signal \bar{BTLW} is not required.

† In basic MOS processor, the CPUWRITE signal is replaced by CPUBKPWRT signal from ADDR/CONTROL board.

Breakpoint Controller Timing Chain

The breakpoint controller clock signal (\bar{CLK}) that is used for controller shift registers, breakpoint panel shift registers, control instruction address counter, and shift counter is comprised of two independent clocks: the breakpoint controller interface internal clock and the processor T3 clock. Figure 4-10 shows the breakpoint controller clock.

Breakpoint Controller Internal Clock – The controller internal clock is used for execution of internal operations independently of the processor.

The controller internal clock generator consists of the dual retriggerable one-shot, A4, that generates approximately 120 nanoseconds symmetrical clock pulses. The clock generator can be stopped and started by controlling the triggering input signal to the first one-shot as follows:

- The breakpoint controller internal clock generator is free-running only when the \bar{RCPU} at G4-4 is high, which indicates internal operation. However, when breakpoint controller operation is executed in conjunction with the processor (\bar{RCPU} is low), the trigger input fed back from the second one-shot is disabled, causing the clock generator to stop. The clock generator is restarted after the breakpoint controller exits from the processor.
- The breakpoint controller clock generator can also be stopped and restarted via the $\bar{TBRLOAD}$ signal. During local operation (the remote/local switch of the breakpoint panel is in the LOCAL position), the universal asynchronous receiver/transmitter (UART) is disabled. This allows the strobe transmitter buffer register signal $\bar{TBRLOAD}$ of the UART to be used for debouncing the breakpoint panel keys. If any key of the breakpoint panel is depressed, the $\bar{TBRLOAD}$ signal is generated to set flip-flop B6-9 to high. This in turn stops the internal clock to allow the key to be debounced. The $\bar{TBRLOAD}$ signal also triggers one-shot C5 to produce approximately a 15 millisecond RESTART signal. The trailing edge of the RESTART signal resets flip-flop B6-9 and allows the internal clock generator to run again. The controller internal clock generator is also stopped by master clear.

Processor T3 Clock – The T3 clock is derived from the processor odd/even clock generator. It is required to synchronize the operation in the breakpoint controller with the processor operation; for example, shift the shift register once every time the processor executes the micro instruction LOC0 that shifts the X register to the left one time. The T3 clock is enabled only when the breakpoint controller operation is executed in conjunction with the processor. During this time, the controller internal clock is stopped. The equation that generates the controller timing chain, \bar{CLK} , is as follows:

$$\bar{CLK} = (\overline{ENMM} + \overline{RCPU}) \cdot \overline{CLOCK} + T3 \cdot (\overline{ENMM} + \overline{RCPU})$$

Breakpoint Controller and Processor Interface Signals

Before the breakpoint controller can force the processor to execute the controller micro instructions required for a specific operation, it must take over processor control by generating SETSM214 or INT14 (see figures 4-2 and 4-3).

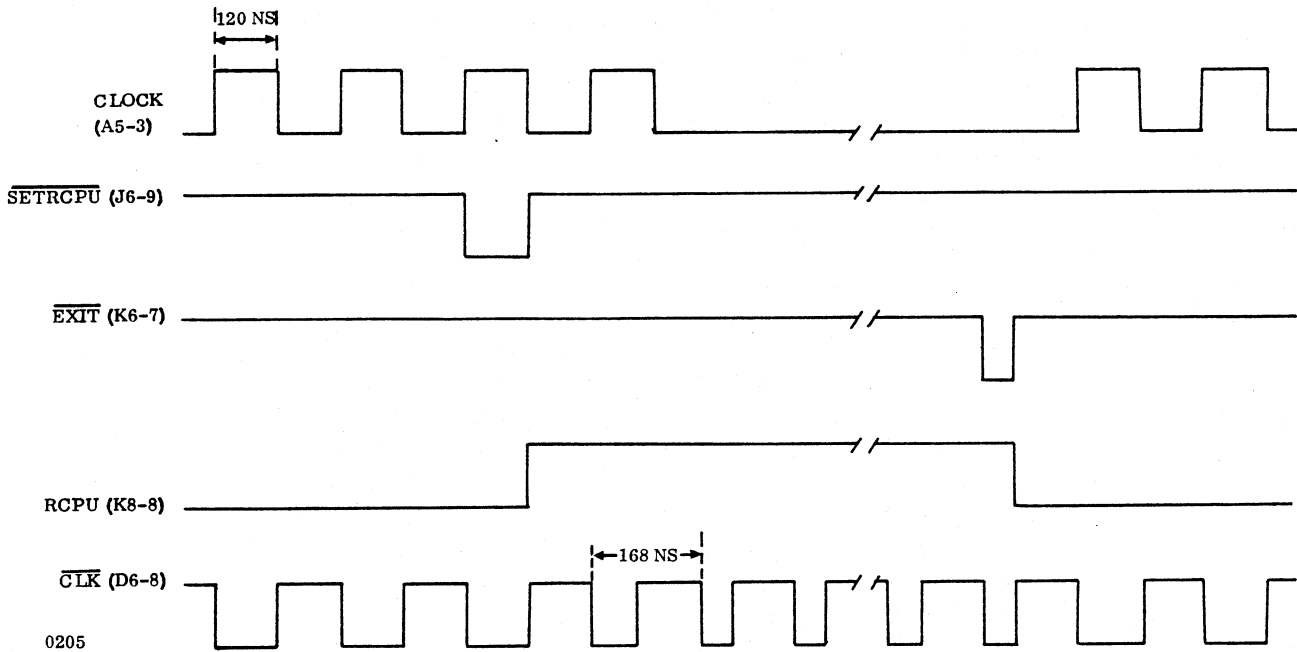


Figure 4-10. Panel Interface Clock

SETSM214 - If the processor is in micro mode, the breakpoint controller must generate the SETSM214 signal, which prepares the processor before it gives control to the breakpoint controller:

$$\text{SETSM214} = \overline{\text{FC19}} \cdot \text{SET RCPU}$$

FC19 = 1 indicates that the processor is in micro mode. The SETRCPU signal is generated by controlware whenever it determines that the processor is required to change/fetch the selected register.

INT14 - If the processor is in macro mode, INT14 is generated to signal the emulator that breakpoint controller operation is requested. Once INT14 is recognized (the breakpoint panel requests CPU interrupt), the emulator processes this interrupt to prepare the processor before giving up control to the breakpoint controller. INT14 at K8-8 is set low by SETRCPU from controlware. After the desired operation is completed, the breakpoint controller exits the CPU by generating the EXIT signal which in fact clears the INT14 signal. The RCPU signal is also generated as a complement of INT14 by the same logic circuit.

Go and Stop

The breakpoint controller has the capability to start and stop the processor. Depending upon the processor mode of operation, micro or macro, the corresponding micro or macro go or stop signal is generated via the control function, H or I. The processor mode of operation is determined by FCR bit FC19.

FC19 = High Micro breakpoint, step, go and stop
 FC19 = Low Macro breakpoint, step, go and stop

CONSTART - The CONSTART signal is generated at flip-flop J8-5 to cause a micro go if either:

- It is entered regardless of FCR bit FC19.
- Auto-display is enabled (i.e., FC10 is set) and any terminator, :, @, or G, is entered with no characters preceding it.
- The CONSTART signal is generated by the breakpoint controller SETRCPU signal. This allows the breakpoint controller to start the processor when it is in micro mode and to initially stop.

The above three conditions form the clock input for flip-flop J8-3.

$$\text{J8-3} = \overline{\text{SETRCPU} + (\text{I} + \text{AUTO}) \cdot \text{CTEQZ} \cdot \text{TERM CODE} \cdot \text{DRREST} \cdot \text{J}}$$

The CONSTART signal is sent to the control 1 module to allow the processor main clock generator to run.

EXGO - The EXGO signal is generated at F6-11 to cause a macro go. EXGO is sent to the status mode interrupt module to set the SM215 bit. When the emulator detects SM215 being set, it causes a macro go. EXGO is generated under the same condition 1 or 2 of CONSTART generation but FCR bit FC19 must be cleared.

$$\text{EXGO} = \overline{(\text{I} + \text{AUTO}) \cdot \text{CTEQZ} \cdot \text{TERM CODE} \cdot \text{DRRESET} \cdot \text{J} \cdot \text{FC19}}$$

CONSTOP - The CONSTOP signal is generated at M8-8 to cause a micro stop if FCR FC19 is set and either:

- The H: command is entered.
- The breakpoint controller exits from the processor.

These two conditions form the clock input for flip-flop J8-11.

$$J8-11 = \overline{[(H \cdot \text{NOT } \text{AUTO}) \cdot \text{CTEQZ} \cdot \text{TERMCODE} \cdot \text{DRRESET} \cdot \text{J}] + \text{EXIT}}$$

If either of the above conditions is true, flip-flop J8-3 goes low to generate CONSTOP.

- The micro step operation executes:

$$J9-10 = \overline{\text{FC20} \cdot \text{RCPU}}$$

- The micro-memory address is equal to the contents of the breakpoint register, and the micro breakpoint condition is met.

All of the above conditions are gated by FC19. The CONSTOP signal is sent to the control 1 module to stop the odd/even time generator. Refer to the control 1 module description in the Basic Micro-Programmable Processor Hardware Maintenance Manual for more details.

EXSTOP - The EXSTOP signal is generated at F6-6 to cause a macro stop. EXSTOP is sent to the status mode interrupt module to clear the SM215 bit. When the emulator detects that the SM215 is clear, it causes a macro stop. Macro stop is generated when FCR bit FC19 is clear and either:

- The H: command is entered:

$$J9-4 = \overline{[(H \cdot \text{NOT } \text{AUTO}) \cdot \text{CTEQZ} \cdot \text{TERMCODE} \cdot \text{DRRESET} \cdot \text{J}]}$$

- The macro step operation executes:

$$J9-3 = \overline{\text{FC20} \cdot \text{CPUEDS} \cdot \text{SEL} \cdot \text{GETMAK}}$$

- The macro-memory address is equal to the contents of the breakpoint register, and the macro breakpoint condition is met.

All of the above conditions are gated by FC19.

Baud Rate Generator

The baud rate is selectable through a baud rate selection switch in the I/O-TTY controller.

The baud rate generator provides four different UART baud rates: 9600, 1200, 300, and 110 baud. The receiver/transmitter clock frequency is 16 times the desired baud rate; therefore, the actual clock frequencies corresponding to the above baud rates are 9,600 times 16, 1,200 times 16, 300 times 16, and 110 times 16, respectively. The baud rate generator consists mainly of three four-bit synchronous binary counters, M1, M2, and M3, which are cascaded by using the positive overflow carry output of the first counter to enable the successive cascaded stage. This arrangement forms a 12-bit frequency divider that divides the 4.9152 MHz oscillator signal from the control 2 module down to the desired clock frequencies.

UART Clock Frequencies	4.9152 MHz Input Clock Divided by	Location
9,600 x 16	32	M2 - 14
1,200 x 16	256	M2 - 11
300 x 16	1024	M3 - 13

A clock frequency of 75 times 16 Hz would be generated at M3-11 if straight division were used. However, by allowing the counter to skip 653 (28D₁₆) clock pulses twice, a 110 times 16 clock frequency is generated at M3-11.

These four baud rates are selectable by PN BAUD1 and PN BAUD0 from the I/O-TTY controller in the 4-to-1 multiplexer M4.

PN BAUD1 PN BAUD0 UART Clock Frequencies

PN BAUD1	PN BAUD0	UART Clock Frequencies
0	0	110 x 16 = 1,760 Hz
0	1	300 x 16 = 4,800 Hz
1	0	1,200 x 16 = 19,200 Hz
1	1	9,600 x 16 = 153,600 Hz

When the 110 baud rate is selected, the SBSELECT signal at M4-15 is high to select two stop bits to be transmitted by the UART. For the other baud rates, 9,600, 1,200, and 300, a single stop bit is selected.

BREAKPOINT CONTROLLER CONTROLWARE

The controlware is defined in this manual as a set of 96- by 24-bit control instructions that sequences and controls controller operations when executed. Refer to Control Instruction Format. The panel controlware is preprogrammed in the control instruction PROMs. An actual listing is included in appendix A for reference.

Controlware Listing Description

The controlware listing includes the definitions of all control instruction fields, P1 through P7. The listing format is explained using an example from the listing.

```

          P1   P2   P3   P4   P5   P6   P7
CHKREM J TRUE REMOTE , , , , , , CHARDY JUMP IF REMOTE
0684
0008
  
```

Where:

- CHKREM is the name of the subroutine.
- J TRUE is the jump mode as indicated by bit 23 of the control instruction being set.
- REMOTE is the jump condition selected by the P1 field. Note that the blank space between commas represents a field of the control instruction as shown above.
- CHARDY is the name of the jump location when the jump condition is met.
- JUMP IF REMOTE is the general remark of the control instruction.
- The most significant two hexadecimal digits, 06₁₆, represent the control instruction address in the PROMs.
- The six hexadecimal digits, 840008₁₆, represent the control instruction itself.

Figure 4-11 shows the controlware bit configuration.

There are four main types of control instructions:

- Jump if condition true (J TRUE) – Execute the control instruction specified by address field P7 (C0 through C6) if the jump condition is true. If not, select the next sequential control instruction.
- Jump if condition false (J FALSE) – Execute the control instruction specified by address field P7 if the jump condition is false. If not, select the next sequential control instruction.
- Unconditional jump (JUMP) – Unconditionally execute the control instruction specified by address field P7. The jump condition is always satisfied in this case.
- Sequential execution (SEQ) – The next sequential control instruction is always executed. In this case, address field P7 may contain the address of the next sequential control instruction or transmit data to the UART.

Basic Controlware Operations

The breakpoint controller controlware performs three basic operations: input, change/fetch, and transmit.

- Input – The controlware is continuously executed in the idle loop consisting of the first two control instructions. As soon as data is ready from the UART or a breakpoint panel switch is depressed, the controlware goes into input operation to process the incoming data. For a typical control command, the controlware first stores the control character code at the control character register if it is a legal code, then stores the hexadecimal data (four-bit data) one after another at the shift register while monitoring for the terminator code. When the terminator code is received, the controlware goes into the change/fetch operation.
- Change/fetch – In this operation, the breakpoint controller requests the CPU (if required) and generates the micro instructions required for the operation indicated by the controller control command. These micro instructions are executed by the CPU. If the

desired operation is an enter operation, the contents of the selected register are first loaded with new data (change operation), and then they are fetched out (fetch operation) to the X register and the shift register (SR). However, if the operation is a display operation, the change operation is omitted and only the fetch operation is performed. After the fetch operation is completed, the breakpoint controller exits and returns control to the CPU before going into the transmit operation.

- Transmit – The transmit operation is used to send the response to the remote device or breakpoint panel for display.

Figures 4-12, 4-13, and 4-14 show flow charts of the input operation, the change/fetch operation, and the transmit operation, respectively.

Typical Breakpoint Controller Operations

The following typical breakpoint controller operations are chosen to illustrate the combined operation of the hardware and controlware. The operations are explained in terms of flow charts and general remarks for each step. Refer to the controlware listing, the panel interface data path block diagram (figure 4-4), and control block diagram (figure 4-6) as necessary.

J11: Control Command

The J11: control command selects the P register by changing digit 1 of the FCR to a value of 1_{16} .

For clarity, assume that the FCR originally contains 12345678_{16} . Digit 1, containing a value of 2_{16} , is changed to a value of 1_{16} . Also assume that a remote device (programmers console) is used and that FCR bit FC15 is not set (the console transmit is not suppressed).

The J11: operation is explained using three major flow charts. The input operation is shown in figure 4-15, the change/fetch operation in figure 4-16, and the transmit operation in figure 4-17. The input operation is divided into three separate flow charts: enter J control character, enter hexadecimal digit 1, and enter terminator .:

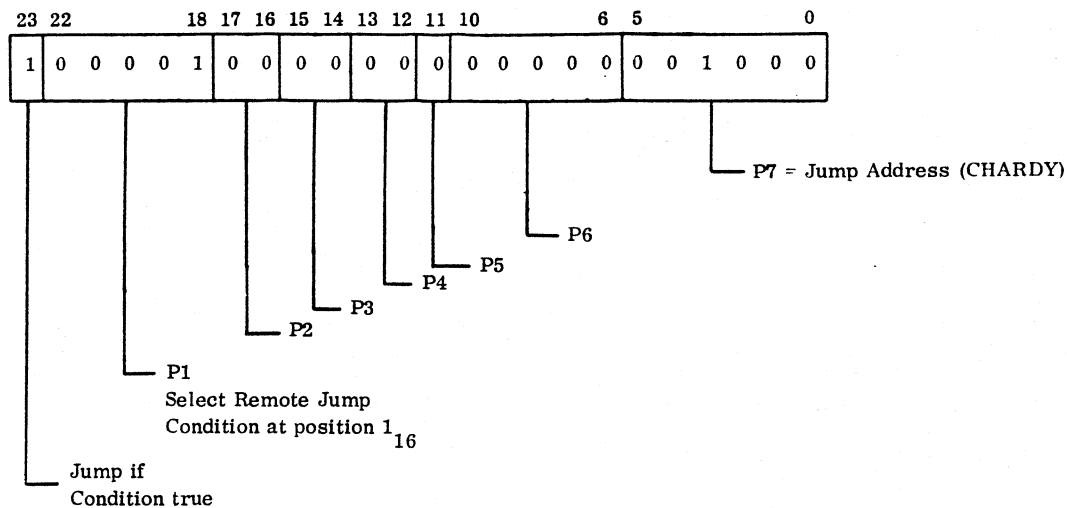


Figure 4-11. Controlware Bit Configuration

K1234: Command

Since the P register is already selected (FC27 through FC24 equal to 0001) in the above example, the next logical operation would be to load the P register with 1234₁₆. The K1234: command is required for this operation. The input and transmit operations are quite similar to the above example. Figure 4-18 is the change/fetch operation for the enter P with 1234 command (K1234:).

BREAKPOINT PANEL

The breakpoint panel is the device used for the man/machine interface via the breakpoint controller module. The breakpoint panel provides a means for entering breakpoint controller control commands and displaying the response. In panel mode, the breakpoint panel is capable of doing any operation that any remote device can do. The breakpoint panel switches and display LEDs are shown and described in figure 2-1 and table 2-1.

Figure 4-19 shows the functional block diagram of the breakpoint panel. The breakpoint panel consists of:

- Momentary switches for control character and data selection
- Switch encoder
- Panel shift register
- Control code display
- Data display and UPPER Indicator display
- LOCAL/REMOTE switch
- MASTER CLEAR switch

Only the switch encoder and panel shift register of the block diagram are described in detail here. The rest of the breakpoint panel block diagram is either self-explanatory or has been described in table 2-1.

SWITCH ENCODER

The switch encoder is used to encode any depressed control character switch or data switch into the corresponding binary code to be used by the breakpoint controller. The switch encoder consists of three priority encoder chips: U10, U12, and U13. The control character switches are encoded by U10. The data switches are encoded by U12 and U13. The truth table of these eight-input priority encoders is included for reference (table 4-10). The control character encoder U10 is always enabled except when a control character switch is being pressed. This makes the control character switches higher priority than the data switches when they are pressed simultaneously.

TABLE 4-10. EIGHT-INPUT PRIORITY ENCODER TRUTH TABLE

Inputs								Outputs					
5	4	3	2	1	13	12	11	10	6	7	9	14	15
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	L	X	X	X	X	X	X	X	L	L	L	L	H
L	H	L	X	X	X	X	X	X	L	L	H	L	H
L	H	H	L	X	X	X	X	X	L	H	L	L	H
L	H	H	H	L	X	X	X	X	L	H	H	L	H
L	H	H	H	H	L	X	X	X	H	L	L	L	H
L	H	H	H	H	H	L	X	X	H	H	L	L	H
L	H	H	H	H	H	H	L	L	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L

The PNLCC signal at U10-14 goes low if any of the control character switches are pressed to indicate to the breakpoint controller that a control character is entered and not hexadecimal data. If any of the breakpoint panel switches are pressed, the PNLKYDEP signal at U13-15 goes active. This signal indicates to the breakpoint controller that data is ready. (Breakpoint panel data ready is processed in the same manner as the data ready line from the UART during remote operation.) The outputs of priority encoders U10, U12, and U13 are ORed together to form the binary codes PNHEX00 through PNHEX03. Table 4-11 shows the binary code of the breakpoint panel switches. The terminator code received condition is generated in the breakpoint controller by ANDing PNLCC and PNHEX00 through PNHEX03.

BREAKPOINT PANEL SHIFT REGISTER

The breakpoint panel shift register converts serial data PNLCLI from the breakpoint controller to parallel format. The 16-bit output of the shift register is used to drive the data display LEDs. The breakpoint panel shift register consists of four universal bidirectional shift register chips: U5, U6, U7, and U8. The shift register mode of operation is controlled by single bit C15 (PNLSHIFT) of the control instruction; therefore, it only operates in left shift and hold modes. The shift register is cleared by the PNLCLR signal

from controlware. The clock PNLCLK is the same breakpoint controller CLK as the breakpoint controller shift register for synchronization (see Breakpoint Controller Internal Clock above). The serial left-shift input data, PNLCLI, is the most significant bit (SR31) of the breakpoint controller shift register. This allows the contents of the breakpoint controller shift register to be displayed. LEDs DS02 and DS17 correspond to the most and least significant data bits of the breakpoint controller shift register. PNLCLI is also used to drive the UPPER indicator when the output of breakpoint controller upper flip-flop K8 is selected. LED DS01 is the upper indicator.

CONTROL CODE DISPLAY

Control code LEDs DS18 through DS20 are driven by signals PNLITE0 through PNLITE2. These signals are the outputs (CC0 through CC2) of the control character register gated by the output of the error flip-flop H1 in the breakpoint controller. Therefore, when there is an error, all control code LEDs are illuminated.

GLOSSARY

Table 4-12 is a tabulation of abbreviations and mnemonics used throughout the text and figures of this manual.

TABLE 4-11. BINARY CODE OF BREAKPOINT PANEL SWITCHES

Switches	PNLCC	PNHEX03	PNHEX02	PNHEX01	PNHEX00
0	1	0	0	0	0
1	1	0	0	0	1
2	1	0	0	1	0
3	1	0	0	1	1
4	1	0	1	0	0
5	1	0	1	0	1
6	1	0	1	1	0
7	1	0	1	1	1
8	1	1	0	0	0
9	1	1	0	0	1
A	1	1	0	1	0
B	1	1	0	1	1
C	1	1	1	0	0
D	1	1	1	0	1
E	1	1	1	1	0
F	1	1	1	1	1
H	0	1	0	0	0
I	0	1	0	0	1
J	0	1	0	1	0
K	0	1	0	1	1
L	0	1	1	0	0
M (not used)	0	1	1	0	1
N (not used)	0	1	1	1	0
:	0	1	1	1	1

TABLE 4-12. GLOSSARY OF TERMS

Term	Description	Term	Description
A (A*)	A register (A register double-precision)	DRDY	Data ready
ADDR	Address bits (00 through 15)	DREADY	Data ready
ALU	Arithmetic logical unit	DRRESET	Data ready reset
A01-A15	Address bits (0 through 15) macro memory	ECPI	Enable breakpoint controller micro memory bus
BKPT	Breakpoint	ENABFC	Enable function control
BP	Breakpoint	ENDARND	Shift the shift register end-around
BPEQUAL	Breakpoint equal	ENMM	Enable micro memory
BPMA	Breakpoint memory address registers	ENTERHEX	Enter hexadecimal number
BP-P/MA	Breakpoint page/memory address registers	EXGO	External go
BP00-BP15	Breakpoint bits (00 through 15)	EXSTOP	External stop
BTLW	Bit test lower/upper micro instruction	F	F register
CCR	Control character register	F1	File no. 1
CHAR	Character	FCR	Function control register
CHARDY	Character ready	FC15	Function code bit (15)
CHKREM	Check remote	F/F	Flip-flop
CKBPMA	Check breakpoint memory address	GATEAB	Gate CPU memory address register clock
CKFC16	Check function control bit 16	GATET	Gate test bit
CKF2	Check field F2	HEXRCVD	Hexadecimal digit received
CKZERO	Check for zero	I	I register
CLK	Clock	INCK	Increment K register
CLR	Clear	INT	Interrupt
CLRCT	Clear counter	K	K register
CLRUL	Clear upper limit	LED	Light emitting diode
CONGATEMIR	Control gate micro-instruction register	LE9	Less than or equal to 9
CONSTART	Control start	LOC0,1,2,3	Location number (0 through 3)
CONSTOP	Control stop	LSB	Least significant bit
CPU	Central processing unit	MEM	Memory
CPUEDS	Central processing unit early data strobe	MIR	Micro-instruction register
CT	Counter	MM	Micro memory
CTEQXF	Count equals 15 hexadecimal	MML	Micro-memory lower 16 bits (16 through 31)
CTEQX3	Count equals 3 hexadecimal	MMU	Micro-memory upper 16 bits (00 through 15)
CTEQZ	Count equals zero	MM00-MM31	Micro-memory instruction bits (00 through 31)
CTEQ0	Count equals zero	MSB	Most significant bit
CTLSB	Counter least significant bit	N	N register
C00-C23	Control instruction bits (firmware)	P	P register
DECMAC	Decrement micro memory address counter	P/MA	Page/memory address registers
DES00-DES02	Destination selection code bits (00-02)		
DIREG	Direct registers selected		
DIS	Display		
DIS0MU	Display 0 memory upper		
DIS1ML	Display 1 memory lower		

TABLE 4-12. GLOSSARY OF TERMS (Contd)

Term	Description	Term	Description
PNBAUD0	Panel baud rate selection bit 0	SSREG	Strobe select for direct register
PNBAUD1	Panel baud rate selection bit 1	STRBCCR	Strobe control character register
PNHEX00- PNHEX15	Panel hexadecimal number bits (00 through 15)	STRBMMX	Strobe micro memory bit
PNKEYDEP	Panel key depressed	STRBUL	Strobe upper/lower
PNLCLK	Panel clock	TB	Test bit
PNLCLR	Panel clear	TBRLOAD	Load transmit buffer
PNLOCAL	Panel local	TERM	Terminate
PNSLI	Panel left shift serial input	TERMCODE	Terminate code
PNTST	Panel test	TMRCVD	Terminate received
PNUARTCLK	Panel UART clock	TRO	Transmit register output
PROM	Programmed read-only memory	TTL	Transistor transistor logic
Q	Q register	UART	Universal asynchronous receiver transmitter
Q*	Q register double precision	U/L F/F	Upper/lower flip-flop
RCPU	Request central processing unit	X	X register
RDR	Data ready	X*	X register double-precision
RDSTRB	Read stroke	XCR	Transmit carriage return
RNI	Read next interrupt	XENAB	X register enable
RRI	Receive register input	XHEXGR9	Transmit
RR0-RR7	Receive register data bits (0 through 7)	XHEXLE9	Transmit hexadecimal number less than or equal to 9
RS232	Type of transmission line characteristics	XLF	Transmit line feed
RTJ	Return jump	XMIT	Transmit
SBSELECT	Strobe select	XMITCCR	Transmit control character
SELGETMAK	Select GET MAK	XMITFCR	Transmit function control register
SETCPU	Set central processing unit	XMSB	X register most significant bit
SR	Shift register	XSPACE	Transmit space
SRL	Shift the shift register left	μ BPEQ	Micro memory address equals breakpoint register
SRLOAD	Load shift register		
SRR	Shift the shift register right		
SR00-SR31	Shift register bits (00 through 31)		

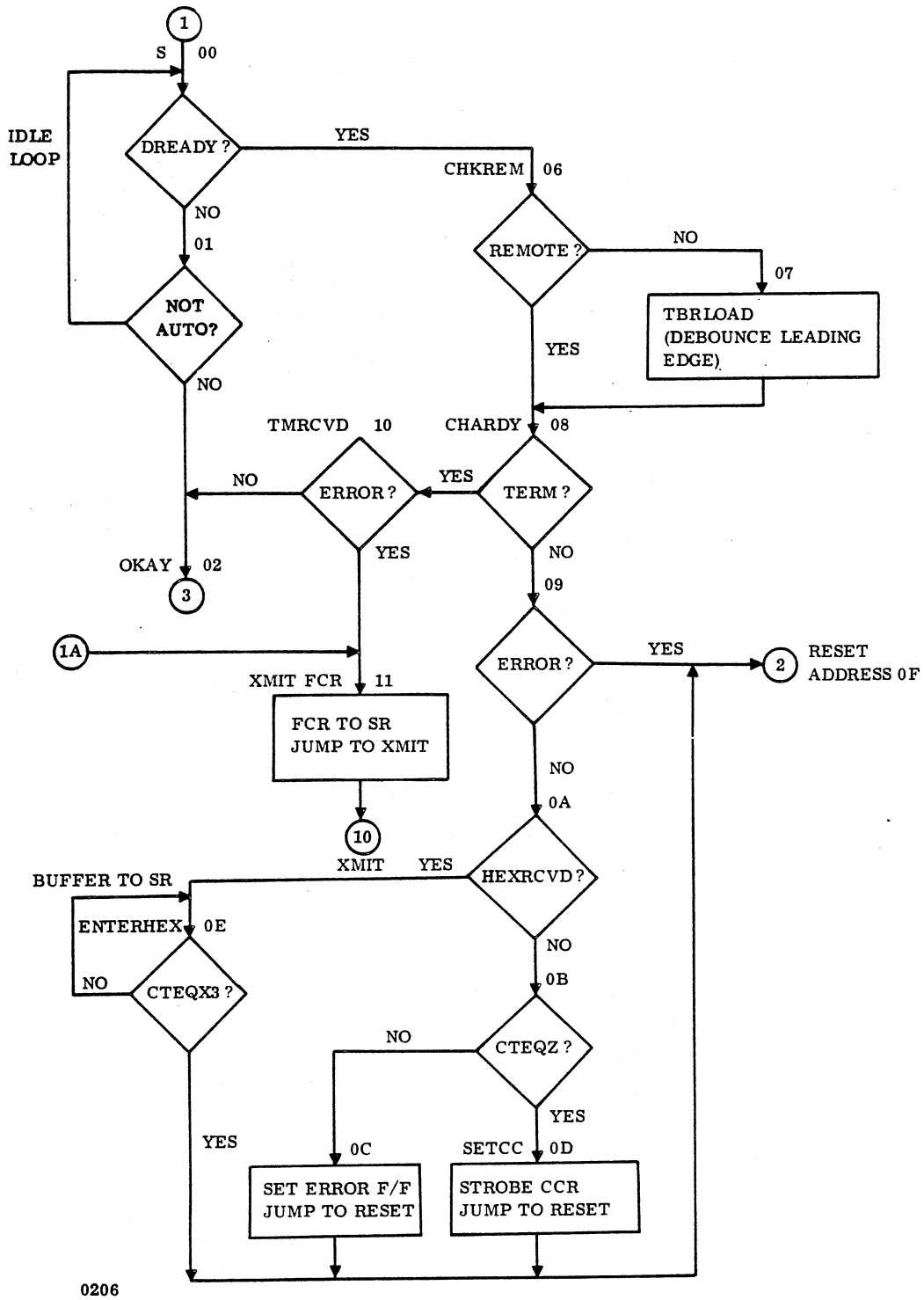
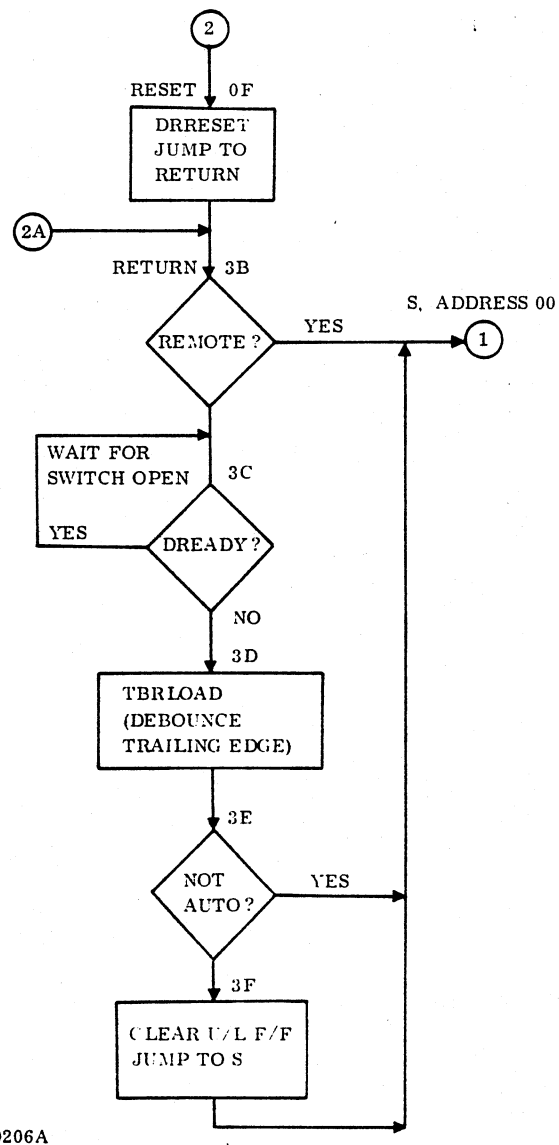
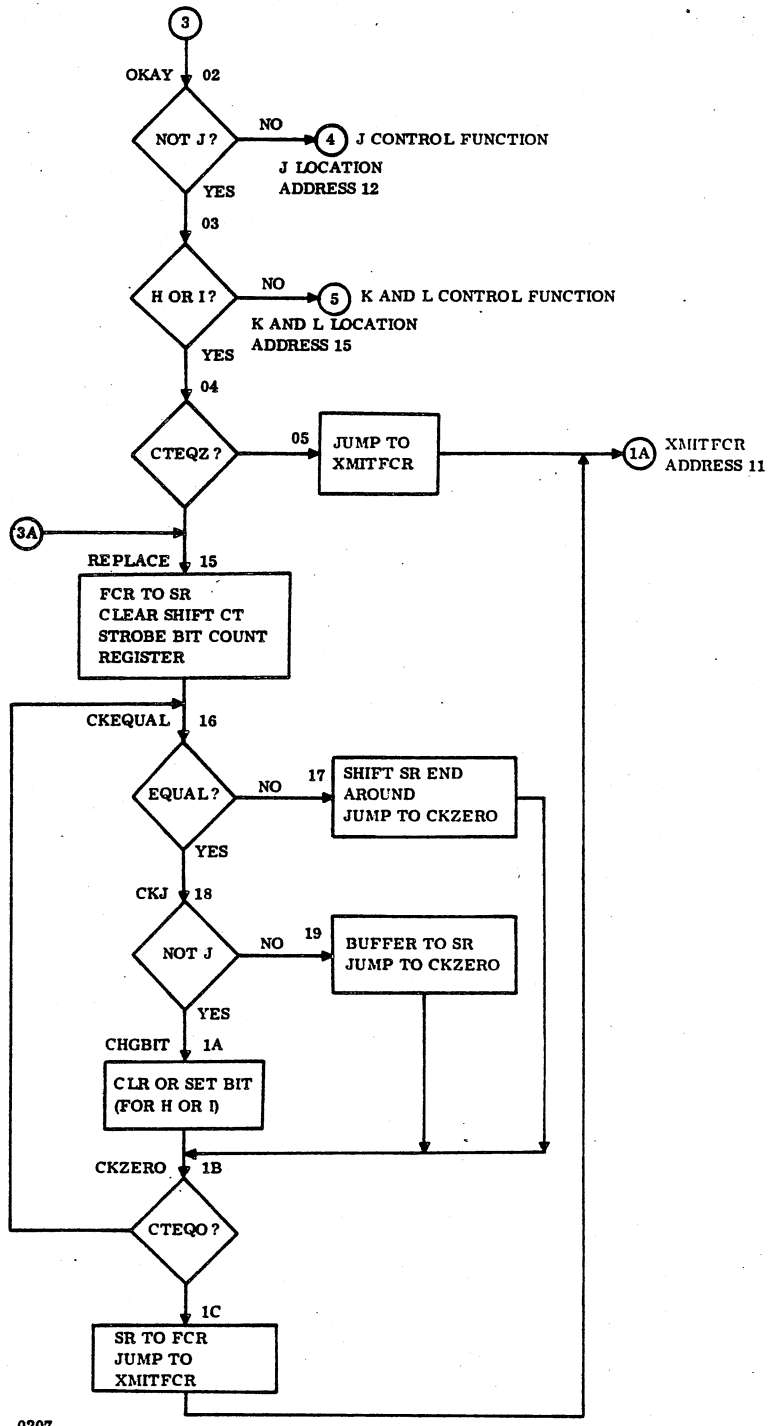


Figure 4-12. Input Operation Flow Chart (Sheet 1 of 2)



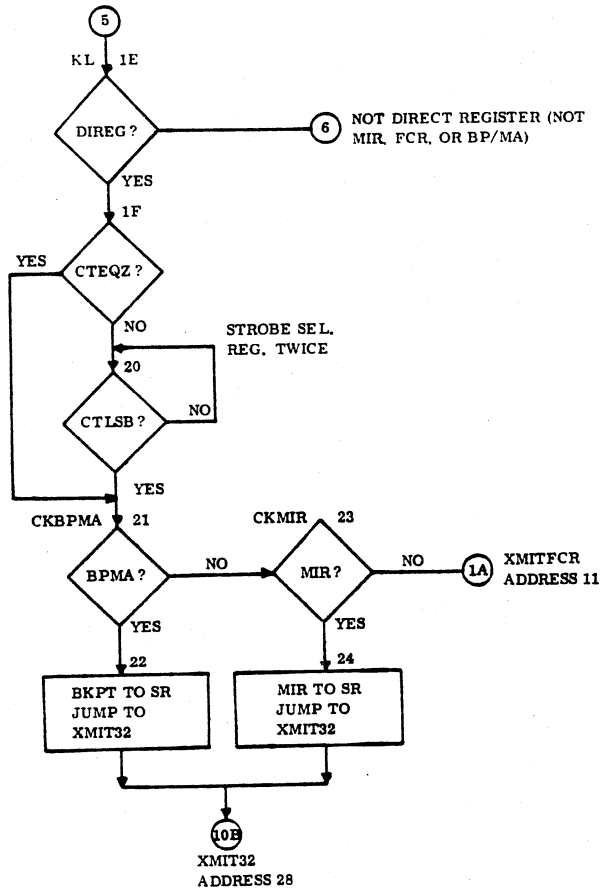
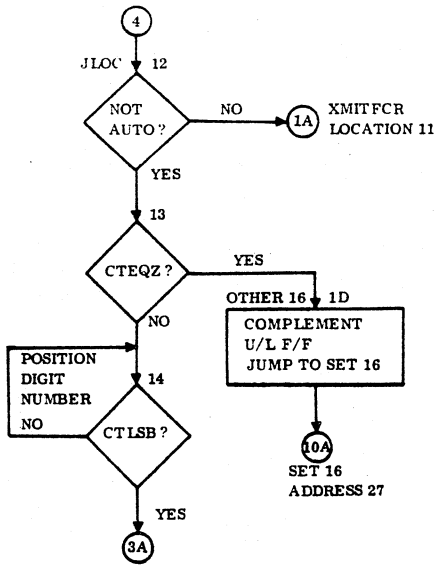
0206A

Figure 4-12. Input Operation Flow Chart (Sheet 2 of 2)



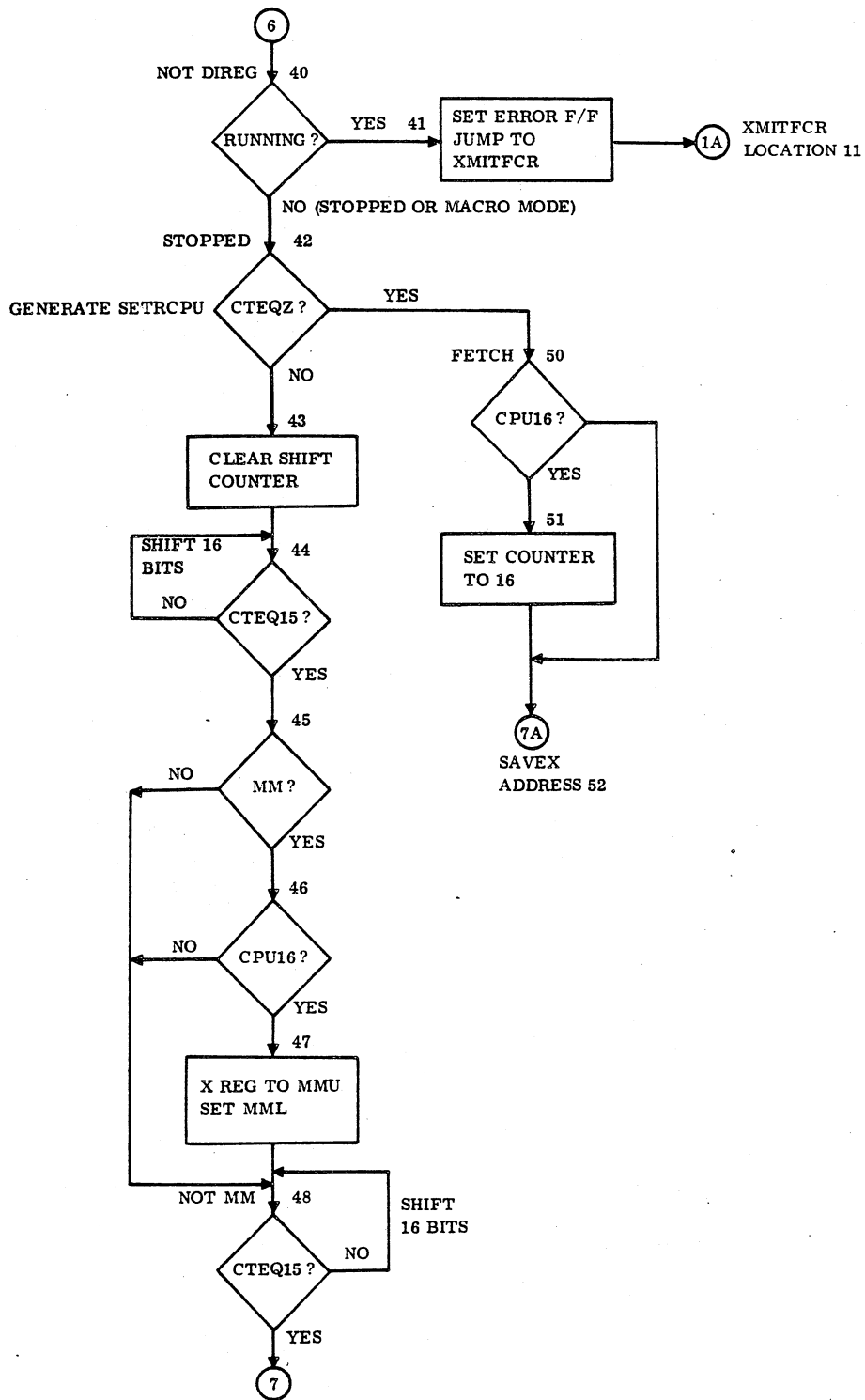
0207

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 1 of 5)



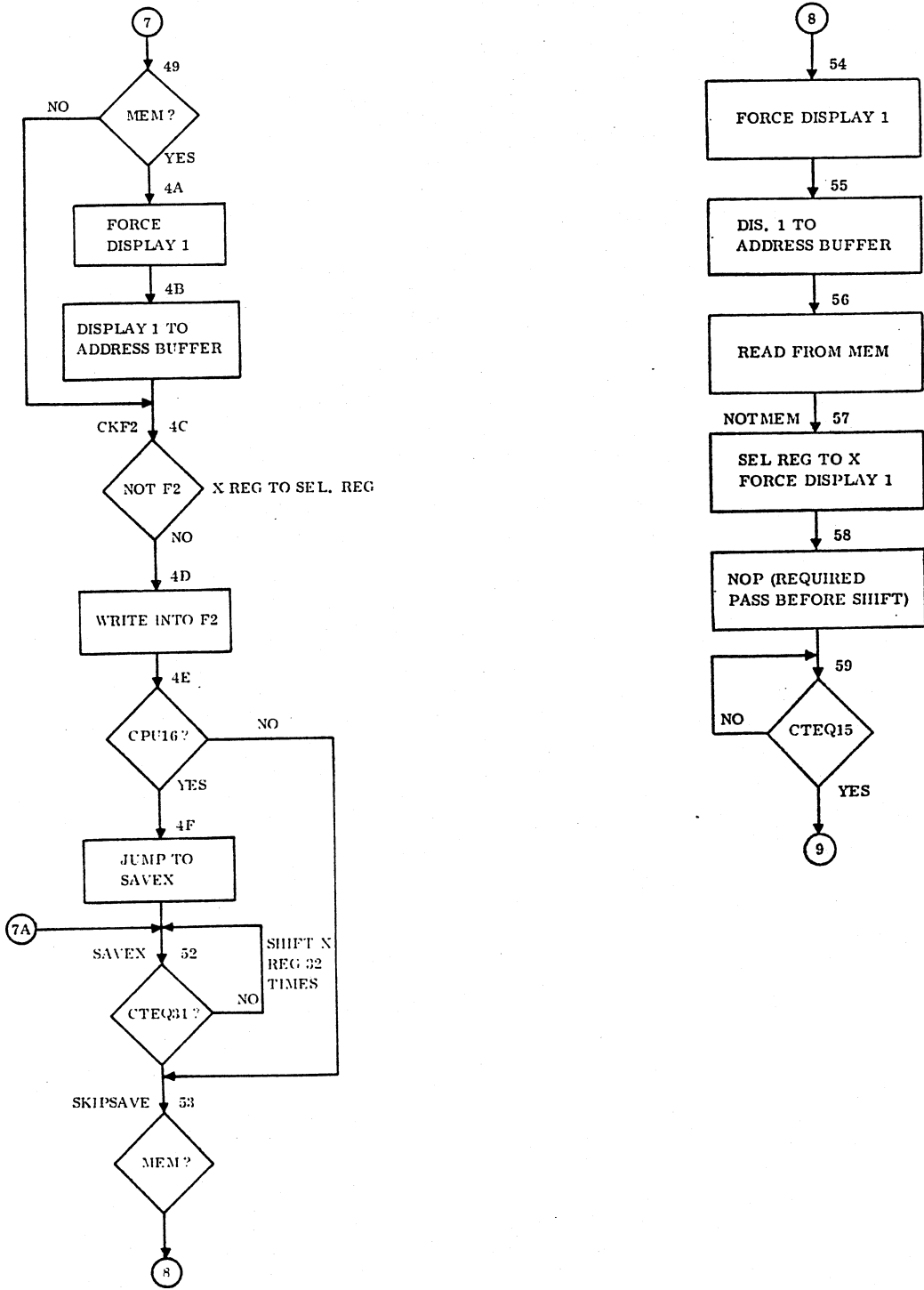
0207A

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 2 of 5)



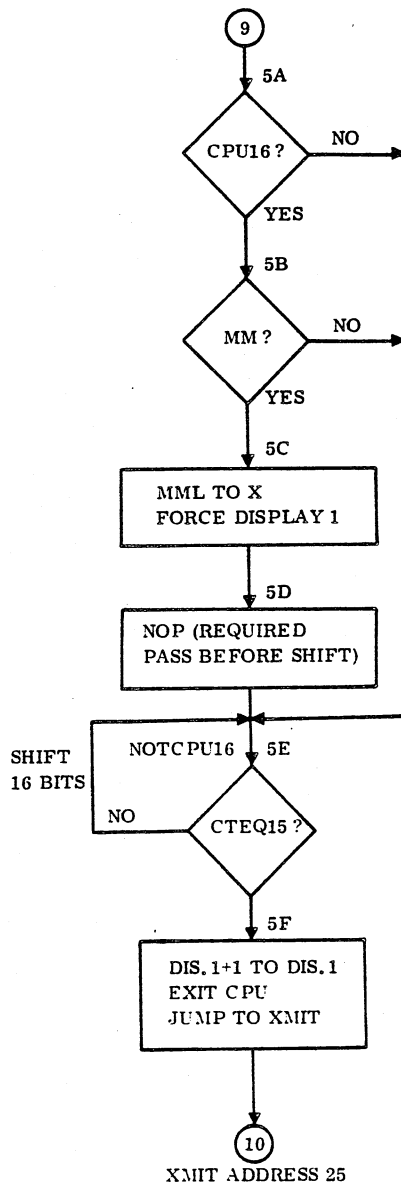
0208B

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 3 of 5)



207C

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 4 of 5)



0207D

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 5 of 5)

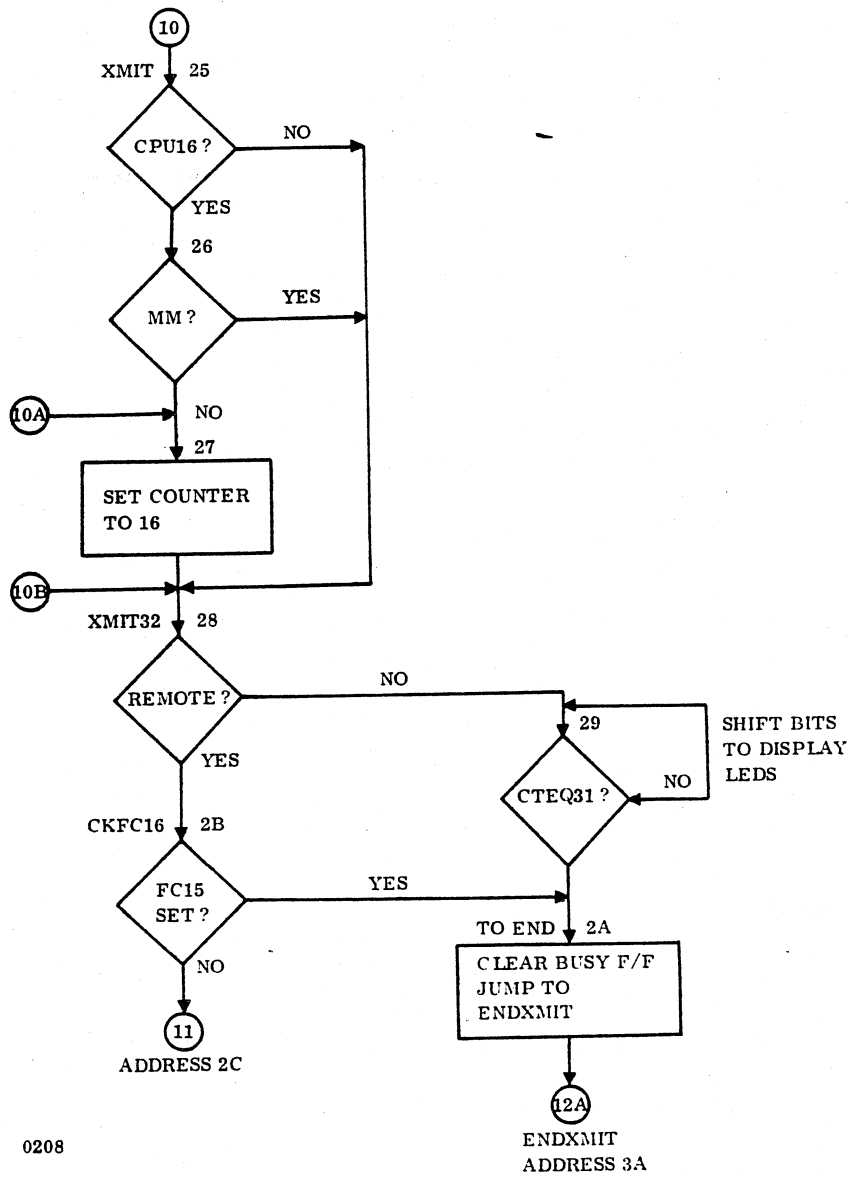
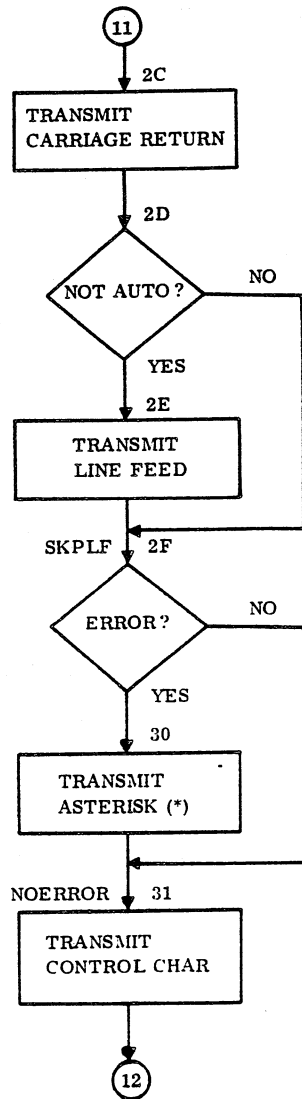
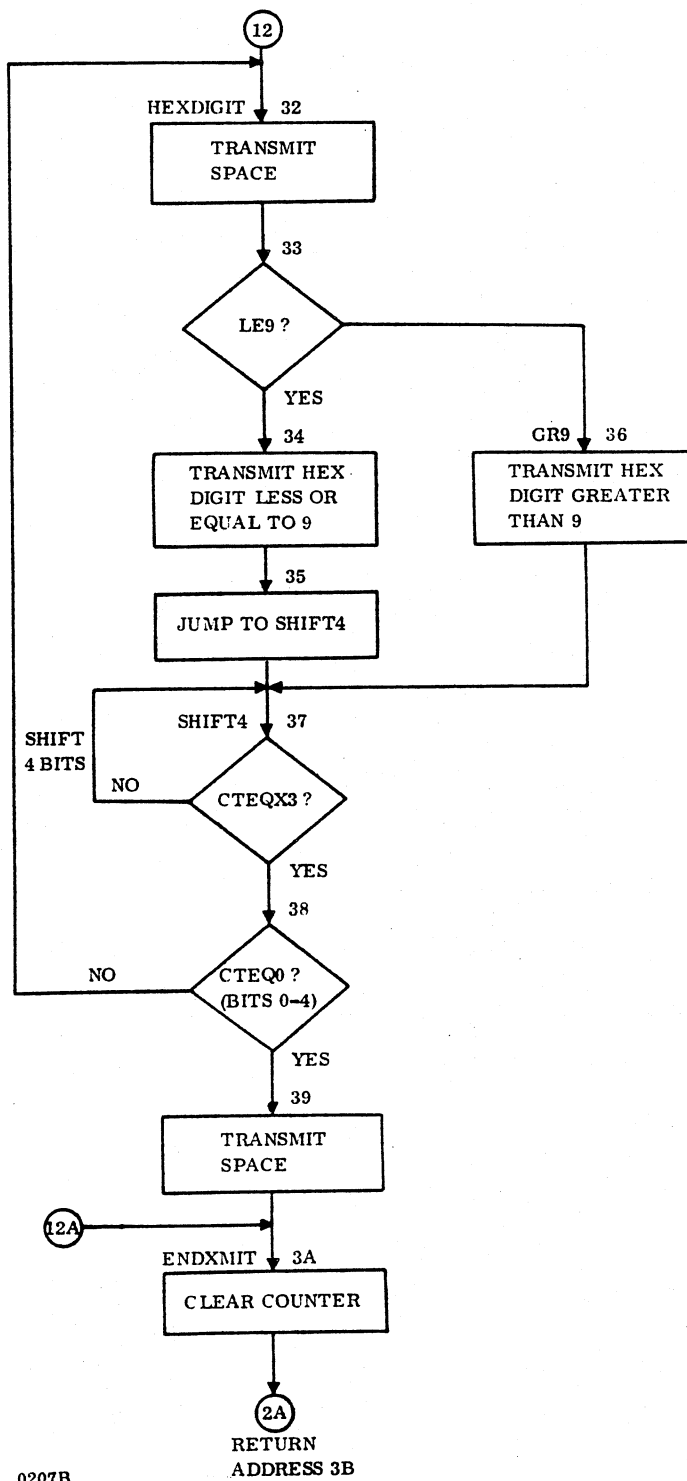


Figure 4-14. Transmit Operation Flow Chart (Sheet 1 of 3)



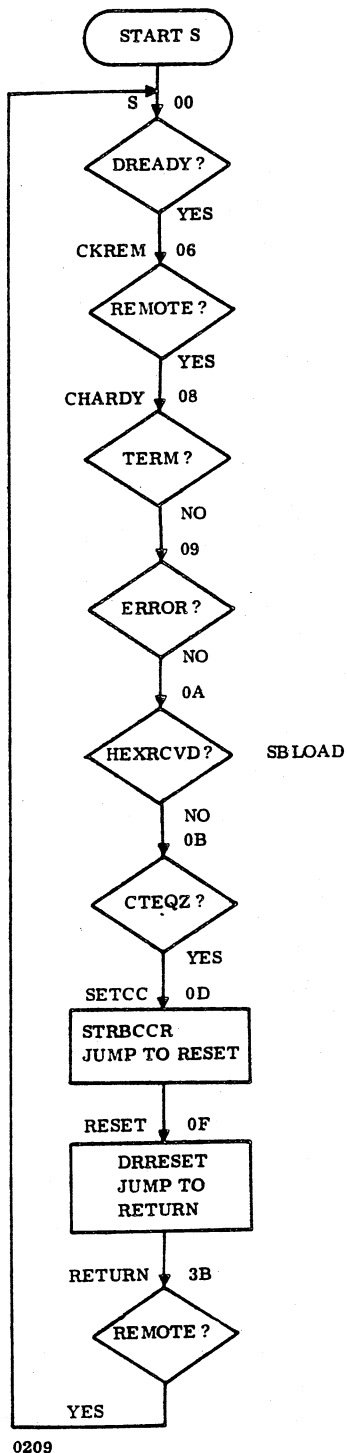
0208A

Figure 4-14. Transmit Operation Flow Chart (Sheet 2 of 3)



0207B

Figure 4-14. Transmit Operation Flow Chart (Sheet 3 of 3)



REMARKS

Data ready (DREADY) condition is true when one complete character has been received at the UART or any panel key is depressed.

The remote/local switch on the panel is in the REMOTE position.

The J control character is not a terminator.

The error flip-flop is not set (no transmit error or no illegal code).

The J control character is not a hexadecimal digit. Generate SBLOAD to load the shift buffer register.

Bits 0 through 5 of the shift counter must equal 0. This step assures that no other character has been entered prior to control character J.

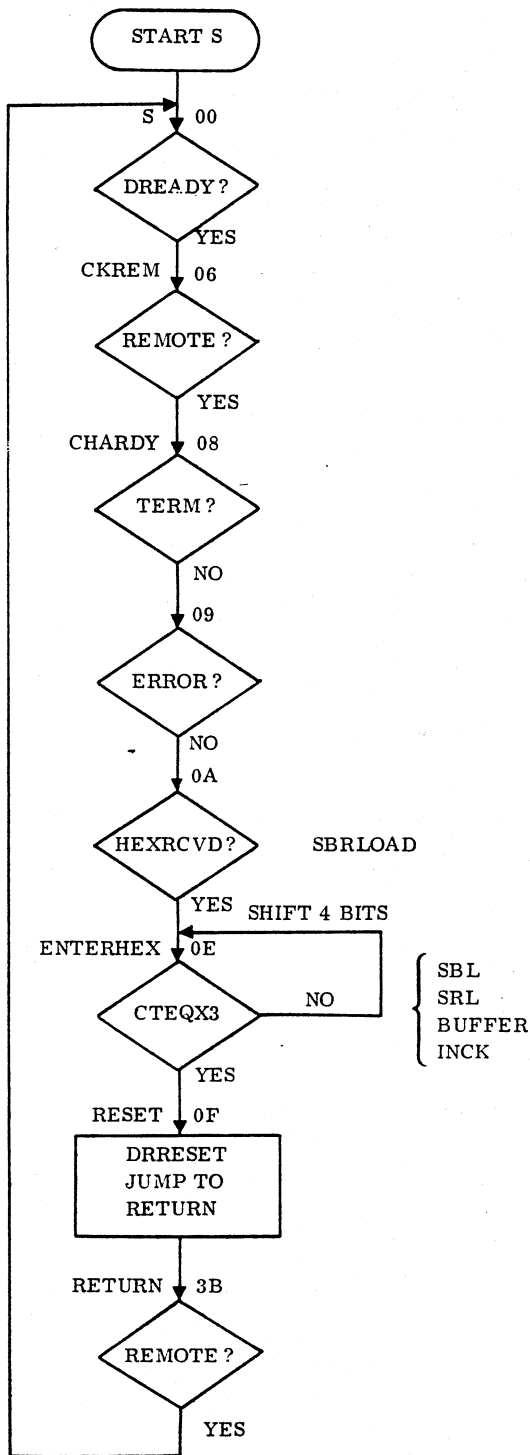
Generate STRBCCR to load the control character register with J code. Jump to the RESET control instruction.

Generate DRRESET to reset the data ready line of the UART. Jump to RETURN.

The REMOTE condition is true; go back to S.

0209

Figure 4-15. Input Operation Flow Chart (Enter J) (Sheet 1 of 3)



REMARKS

1₁₆ is the hexadecimal digit. SBLOAD = Load the shift buffer.

Transfer the contents of the shift buffer to the shift register by:

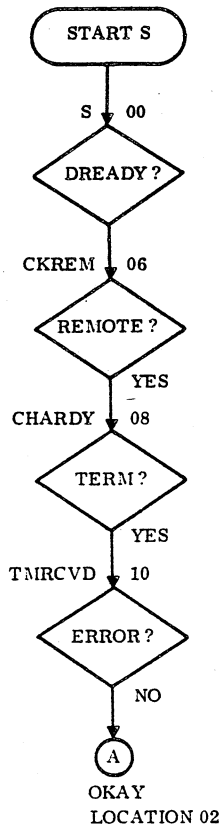
- SBL = Shift shift buffer to left one bit.
- SRL = Shift shift register to left one bit.
- BUFFER = Select output of shift buffer to be left shift serial input to shift register.
- INCK = Enable shift counter.

Repeat the above control instruction four times.

Reset the data ready line from the UART.

Figure 4-15. Input Operation Flow Chart (Enter J) (Sheet 2 of 3)

REMARKS

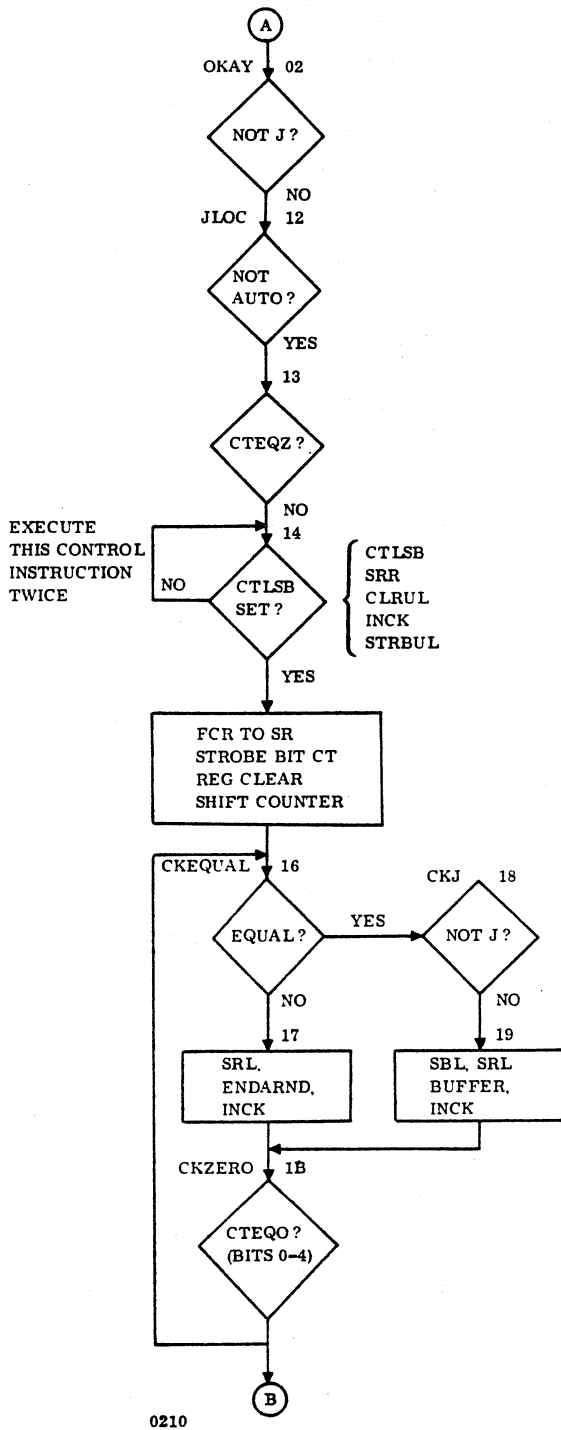


The colon (:) is a terminator.

DRRESET = Reset the data ready line of the UART.
Jump to the OKAY control instruction, which is the beginning of the change/fetch operation.

0209B

Figure 4-15. Input Operation Flow Chart (Enter J) (Sheet 3 of 3)



REMARKS

J control command

Bits 0 through 5 of the shift counter equal 001000. The CTEQZ condition is not true.

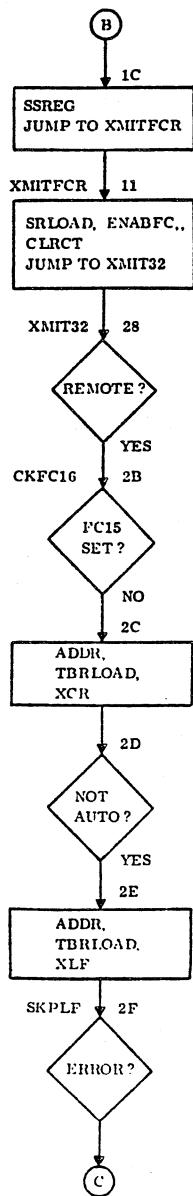
This control instruction is executed twice to shift the shift register to the right two places.

Initially	After two instruction executions
LSB of SR = 00010001	to 01000100
Shift counter = 001000	to 001010

Bit count register = 00100. Load the shift register with the contents of FCR, 12345678₁₆. Clear the shift counter (shift counter = 000000).

Note that during this J operation, the lower two bits of the bit counter register and the lower two bits of the shift counter are not compared and logically forced to be equal. (Refer to the hardware description of the bit counter register.) Initially, the shift register is shifted left four bits with end-around (shift register = 23456781₁₆). The EQUAL condition is now true. (Bit counter register = shift counter = 001xx). The contents of the shift buffer are transferred to the shift register by shifting left four times (the EQUAL condition is true four times). The shift register now contains 3456781₁₆. The shift register continues to be shifted left with end-around until the CTEQ0 condition is true (bits 0 through 4 of the shift counter = 00000). The shift register now contains 11345678₁₆.

Figure 4-16. Change/Fetch Operation Flow Chart (J11: Command)



0211

REMARKS

Generates SSREG, which in turn generates STROBE F/C signal to load the FCR with the contents of shift register.

FCR to SR
Clear shift counter
Jump to MIT 32

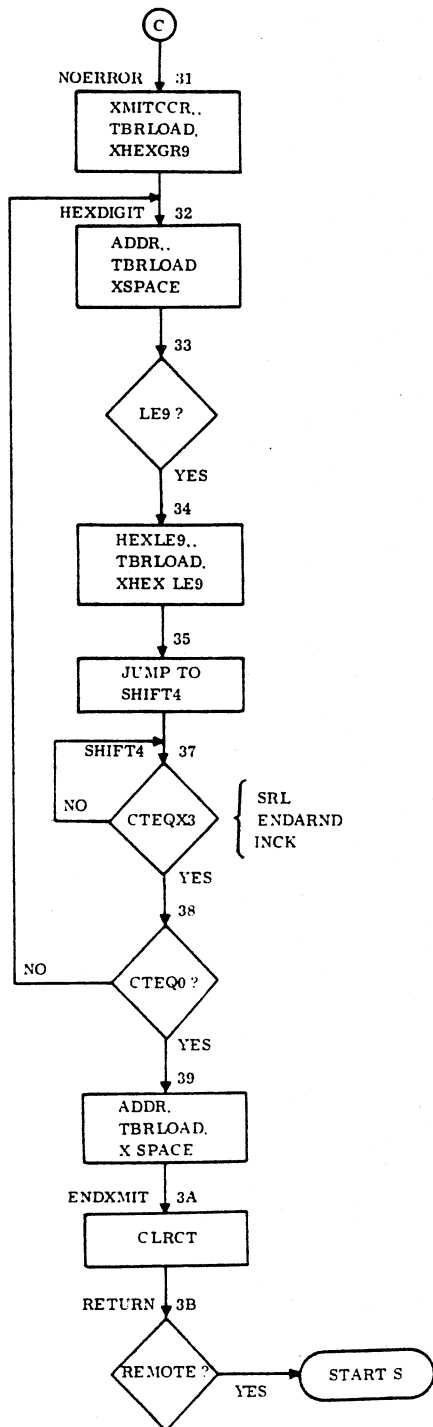
Function control register bit FC15 is previously assumed not set (not suppressing console transmit).

Transmit carriage return ASCII code by loading the transmit buffer register of the UART with address field P7 equal to 0001101.

Assume not auto display mode.

Transmit line feed ASCII code by loading the transmit buffer register of the UART with address field P7 equal to 0001010.

Figure 4-17. Transmit Operation Flow Chart (J11: Command) (Sheet 1 of 2)



0211A

REMARKS

Transmit control character J in ASCII code.

Transmit a space in ASCII code.

Check if the hexadecimal digit is less than 9, and clear the busy flip-flop.

Transmit any hexadecimal digit less than 9 in ASCII code.

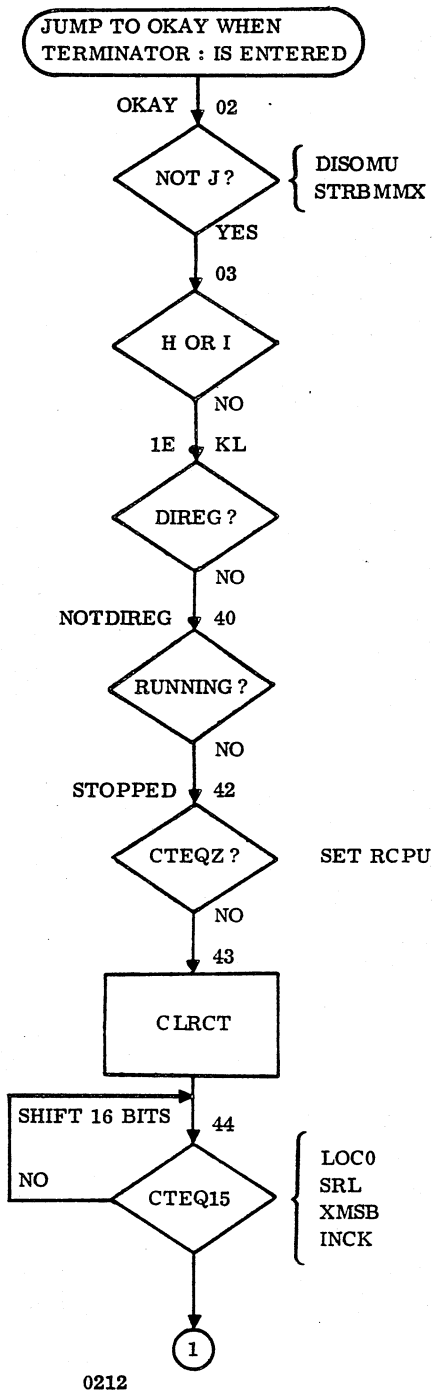
Shift the shift register to the left, end-around, four bits. This step selects the next hexadecimal digit for transmission.

Check if bits 0 through 4 of the shift counter equal 0. If not, repeat the transmit hexadecimal digit less than 9 subroutine (all hexadecimal digits of the FCR in this example are less than 9).

Transmit a space in ASCII code.

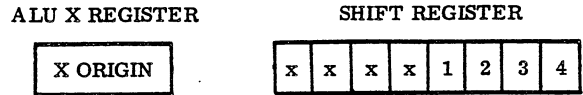
Clear the counter.

Figure 4-17. Transmit Operation Flow Chart (J11: Command) (Sheet 2 of 2)



REMARKS

After the input operation (K1234: command has been entered) the ALU X register contains its original value and the shift register contains 1234₁₆.



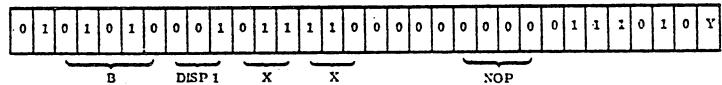
P is not the direct register.

Assume that the processor is in micro mode and not running. Otherwise an error occurs.

SETRCPU from controlware sets SM214 to 1 to take over control of the CPU. SETRCPU also generates the RCPU signal, which conditions the panel interface hardware to processor timing.

Clear the shift counter.

LOC0 micro instruction:



This micro instruction shifts the X register left one bit with SR31 as left shift input data. The shift register is also shifted left with XMSB as the left shift input data. After 16 shifts:

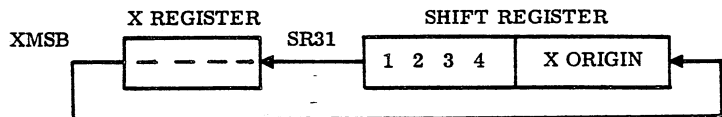
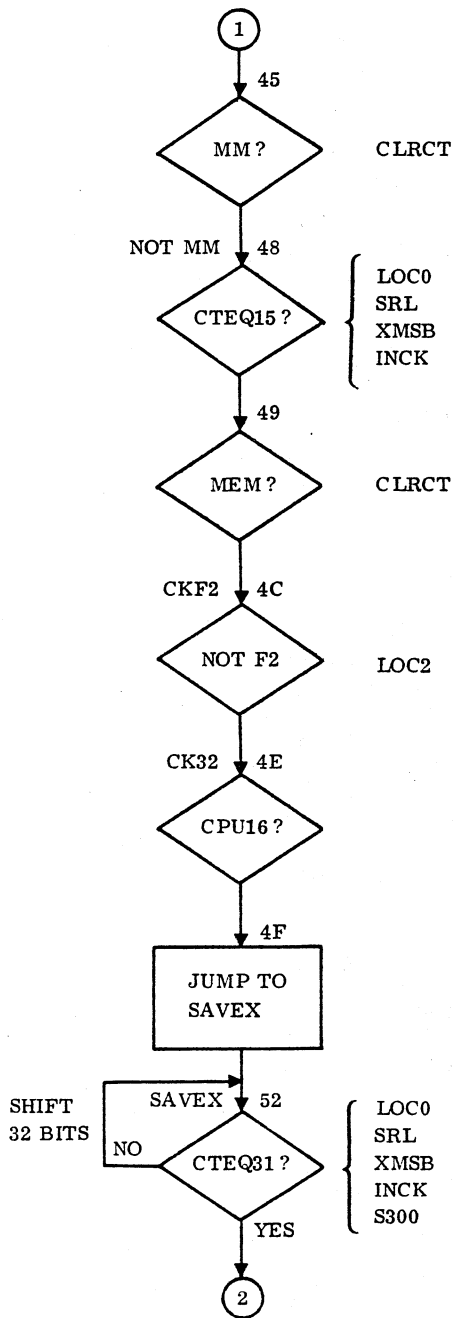


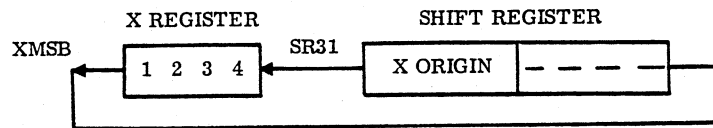
Figure 4-18. Enter P with 1234 command (K1234:), Change/Fetch Operation (Sheet 1 of 3)



REMARKS

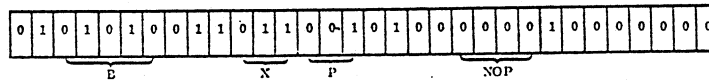
Clear the shift counter. Jump to NOTMM.

Shift the X register and shift register 16 bits.



Clear the shift counter. Jump to CKF2.

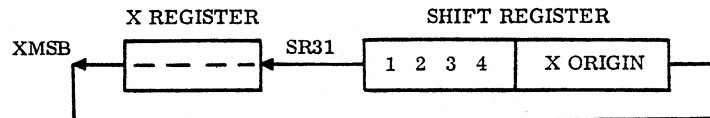
The LOC2 micro instruction transfers the contents of the X register to the selected register, P:



The P register now contains 1234₁₆.

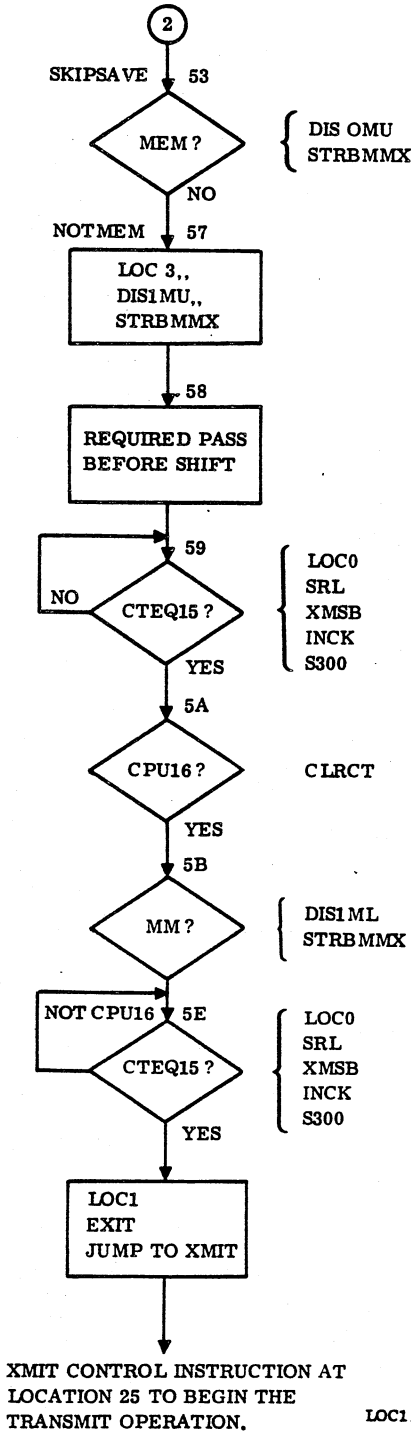
Do not jump if there is a 16-bit CPU.

The X register and shift register are both shifted left 32 bits. Since X is not the selected register, its original contents must be saved. The P6 field equal to 0001 is decoded into the XENAB signal, which selects SR31 (save X since it is not the selected register) as the least significant bit of micro instruction LOC0.



0212A

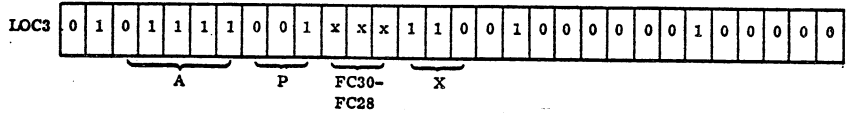
Figure 4-18. Enter P with 1234 command (K1234:), Change/Fetch Operation (Sheet 2 of 3)



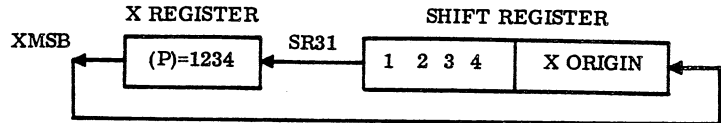
REMARKS

Set MMU at STRBMMX.

The LOC3 micro instruction transfers the contents of the P register to the X register.

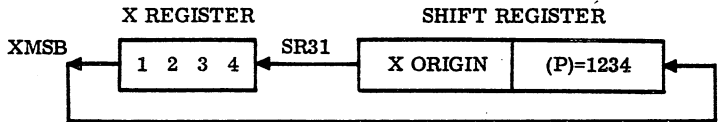


The X register and shift register now contain:



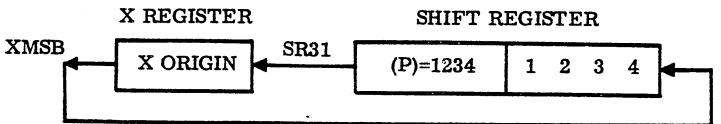
Also force display 1.

Shift 16 bits.



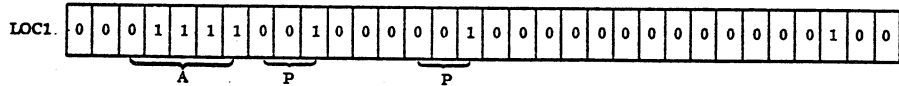
Clear the shift counter.

Shift 16 bits.



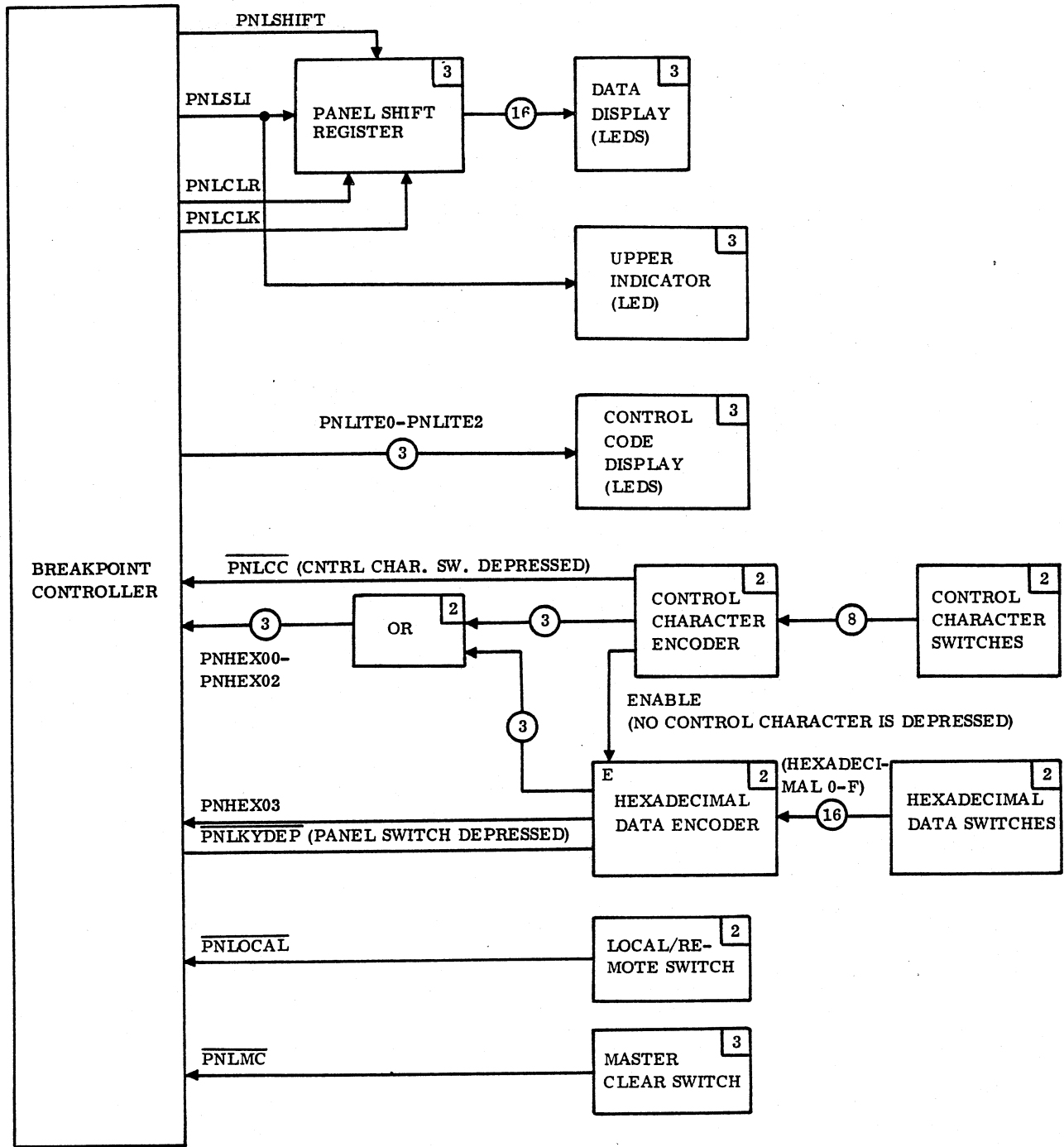
The S300 from this control instruction also clears the SM214.

The breakpoint controller exits and returns control to the CPU by generating the EXIT strobe, which sets the M field of this LOC1 micro instruction to 00 (RTJ mode). M field equal to 00 causes the ENMM signal in the control 1 module to be high, disabling the controller and enabling the micro memory to the MIR. The T field of the micro instruction is equal to 000 for micro mode.



0212B

Figure 4-18. Enter P with 1234 command (K1234:), Change/Fetch Operation (Sheet 3 of 3)



NOTE: THE NUMBER IN THE UPPER RIGHT CORNER OF EACH BLOCK CORRESPONDS TO THE LOGIC DIAGRAM SHEET NUMBER

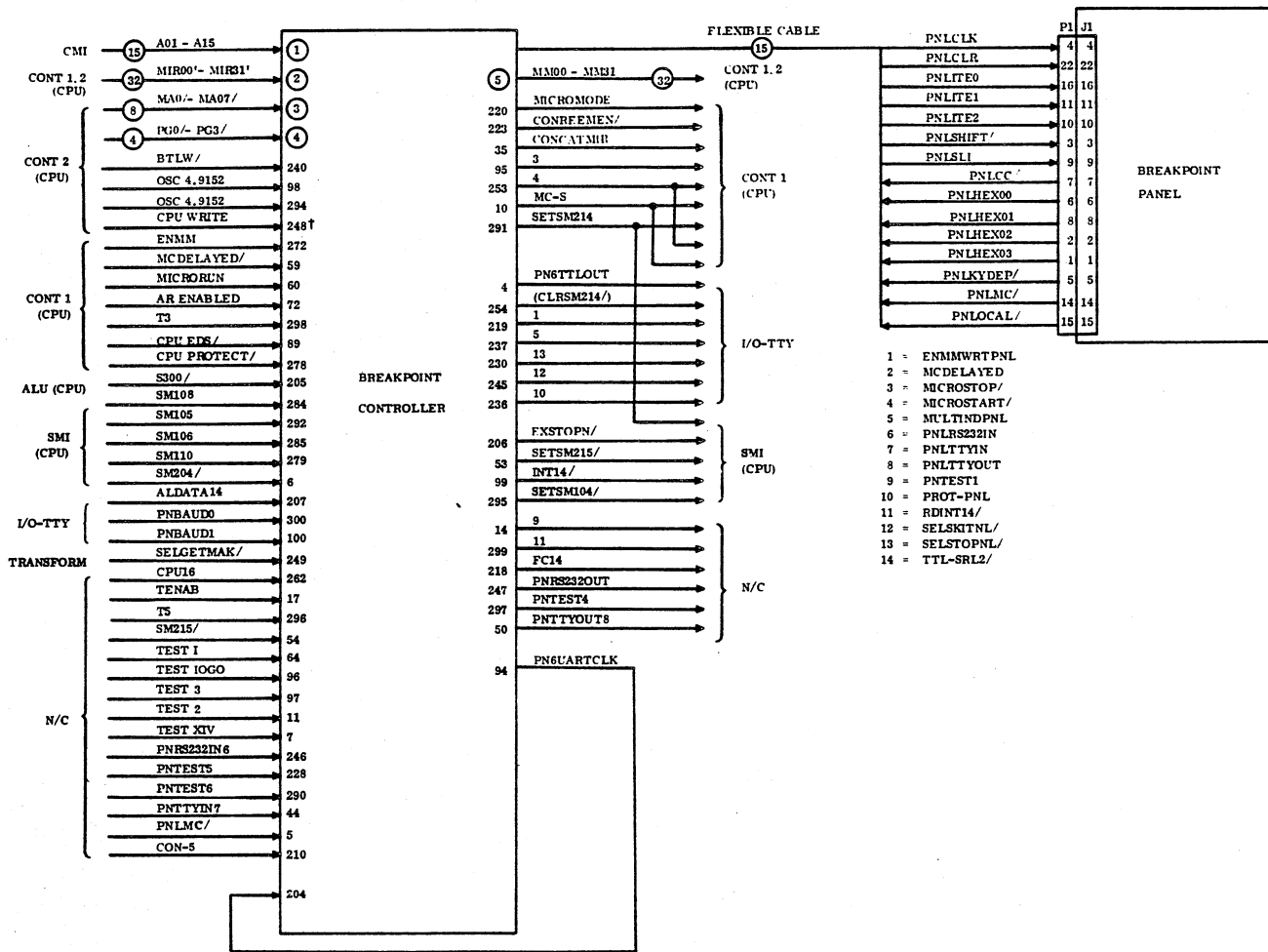
Figure 4-19. Breakpoint Panel Functional Block Diagram

INTERCONNECTING SIGNAL DIAGRAM

Figure 5-1 shows all the interface signals between the processor, breakpoint controller, and the breakpoint panel. Signal names and logic board pin numbers are included. This diagram should be useful for troubleshooting to the board level.

LOGIC DIAGRAMS

Figure 5-2 is the logic diagram for the breakpoint panel. Figures 5-3 and 5-4 are logic diagrams for the breakpoint controllers 96752176 and 88909300 respectively.



† IN BASIC MOS PROCESSOR, THE CPUWRITE SIGNAL IS REPLACED BY THE CPUBKWPRT SIGNAL FROM THE ADDR/ CONTROL BOARD.

0214

Figure 5-1. Interconnecting Diagram (Sheet 1 of 2)

PIN NUMBERS

① A01 - A15

A01	271
A02	273
A03	275
A04	277
A05	263
A06	265
A07	267
A08	269
A09	225
A10	229
A11	260
A12	258
A13	259
A14	233
A15	234

③ $\overline{MA0}$ - $\overline{MA7}$

$\overline{MA0}$	256
$\overline{MA1}$	19
$\overline{MA2}$	87
$\overline{MA3}$	92
$\overline{MA4}$	281
$\overline{MA5}$	283
$\overline{MA6}$	280
$\overline{MA7}$	86

④ $\overline{PG0}$ - $\overline{PG3}$

$\overline{PG0}$	36
$\overline{PG1}$	39
$\overline{PG2}$	38
$\overline{PG3}$	29

② MIR00' - MIR31'

MIR00'	213	MIR21'	241
MIR01'	209	MIR22'	43
MIR02'	212	MIR23'	244
MIR03'	12	MIR24'	62
MIR04'	9	MIR25'	61
MIR05'	214	MIR26'	287
MIR06'	16	MIR27'	286
MIR07'	215	MIR28'	288
MIR08'	250	MIR29'	293
MIR09'	252	MIR30'	75
MIR10'	231	MIR31'	88
MIR11'	232		
MIR12'	261		
MIR13'	268		
MIR14'	270		
MIR15'	69		
MIR16'	21		
MIR17'	15		
MIR18'	224		
MIR19'	8		
MIR20'	243		

⑤ MM00 - MM31

MM00	28	MM21	46
MM01	31	MM22	45
MM02	27	MM23	42
MM03	26	MM24	77
MM04	25	MM25	78
MM05	24	MM26	79
MM06	23	MM27	80
MM07	22	MM28	81
MM08	55	MM29	82
MM09	56	MM30	83
MM10	57	MM31	84
MM11	58		
MM12	63		
MM13	65		
MM14	66		
MM15	70		
MM16	40		
MM17	41		
MM18	49		
MM19	48		
MM20	47		

Figure 5-1. Interconnecting Diagram (Sheet 2 of 2)

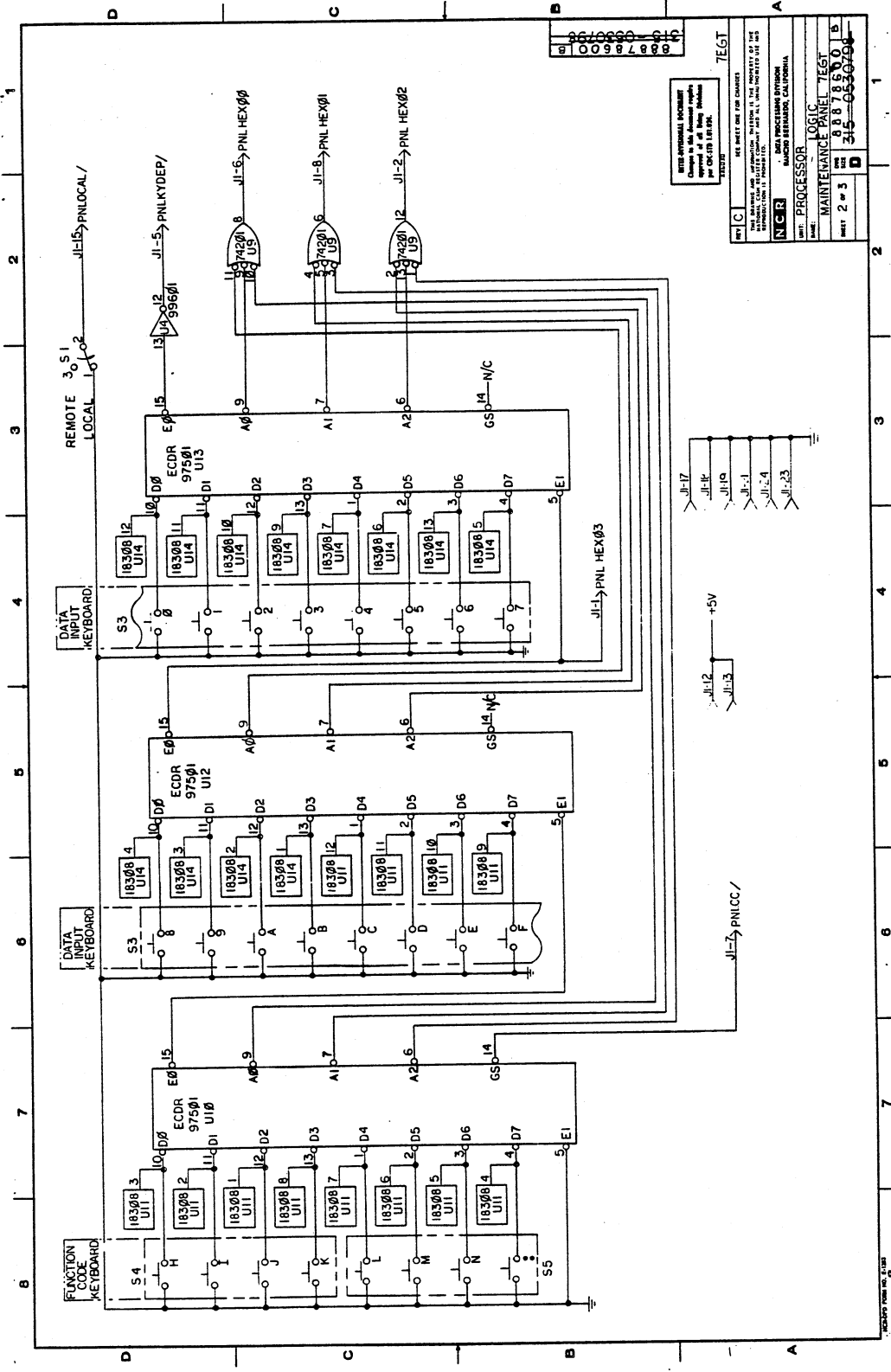


Figure 5-2. Breakpoint Panel Logic Diagram (Sheet 1 of 2)

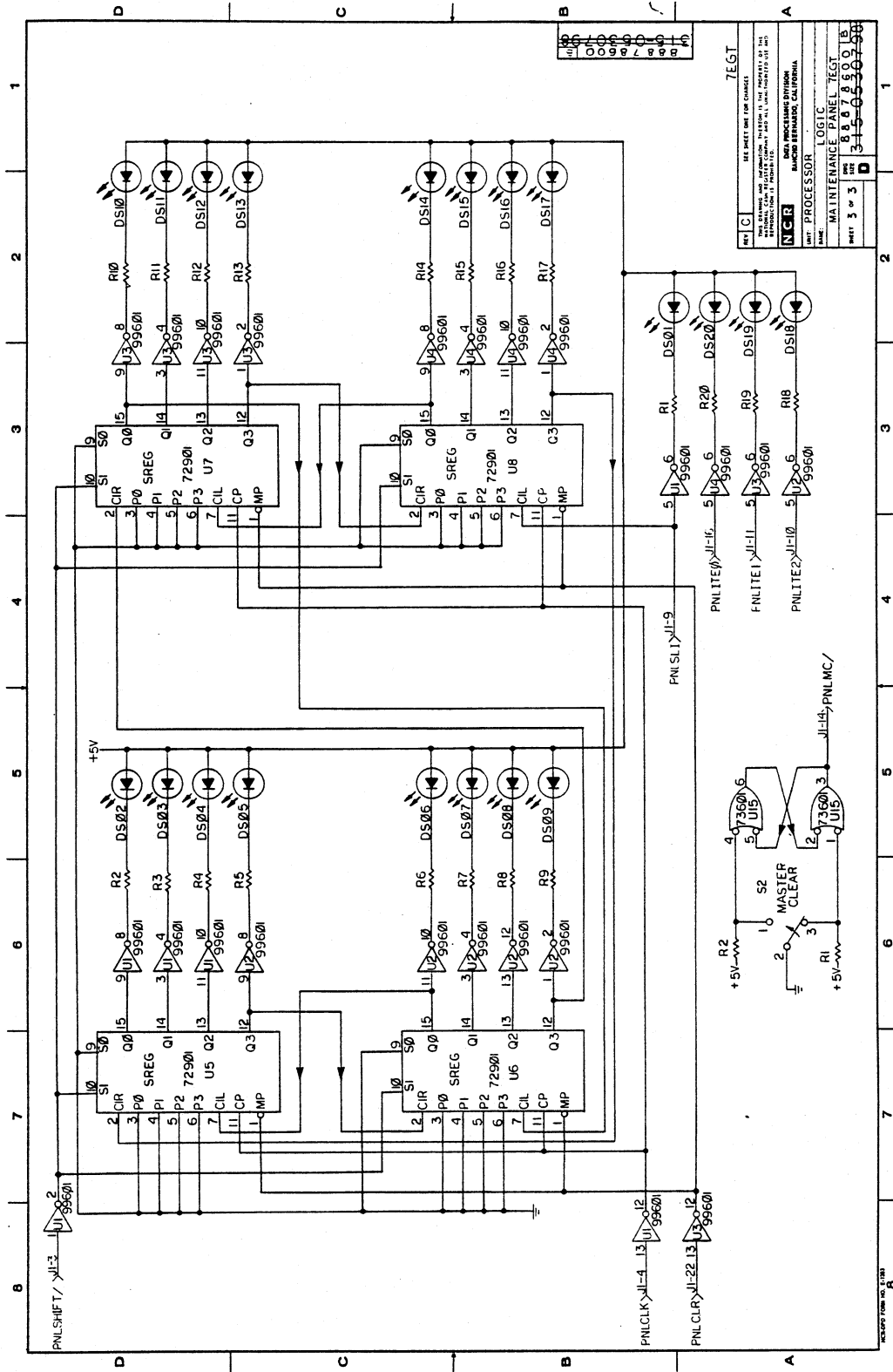
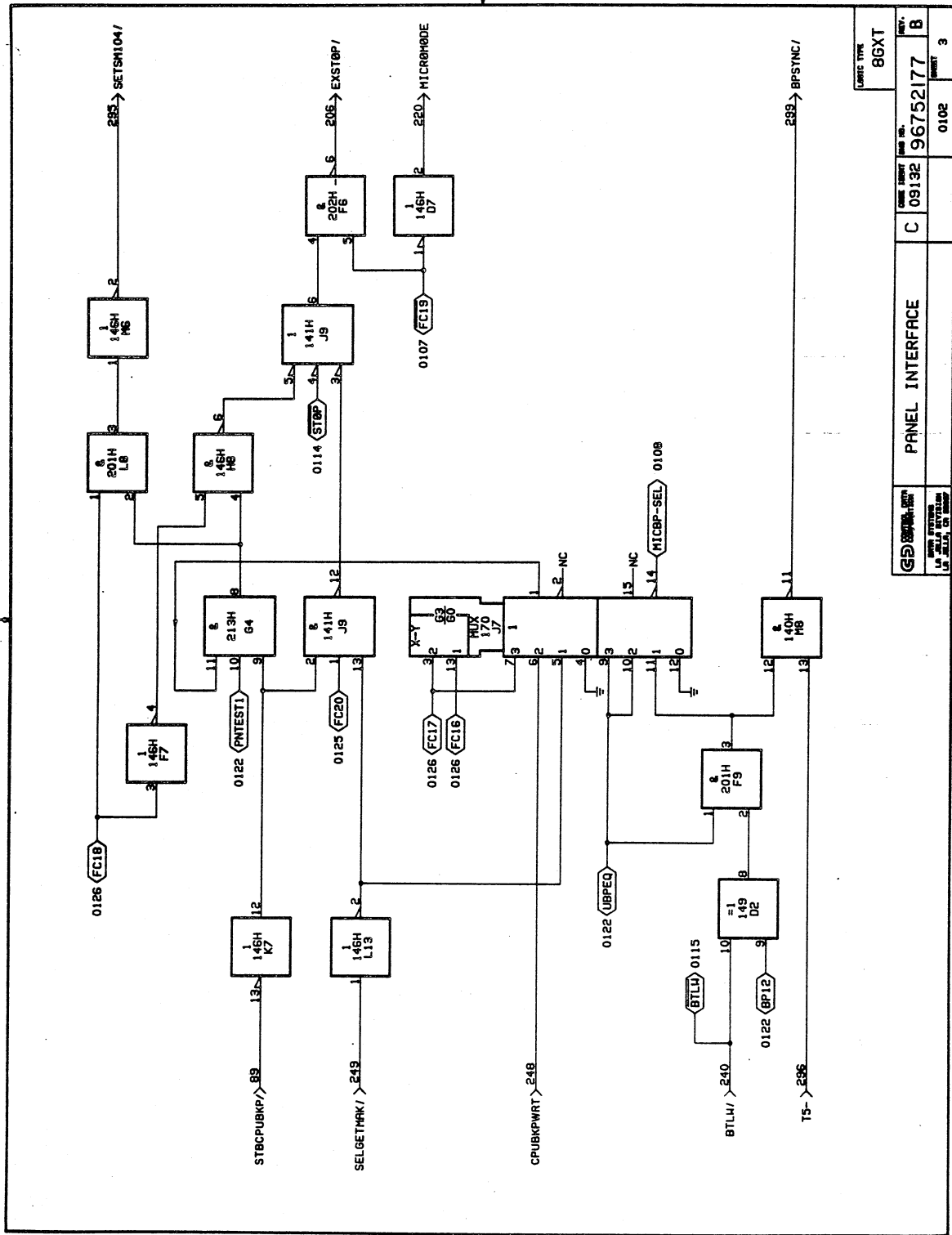
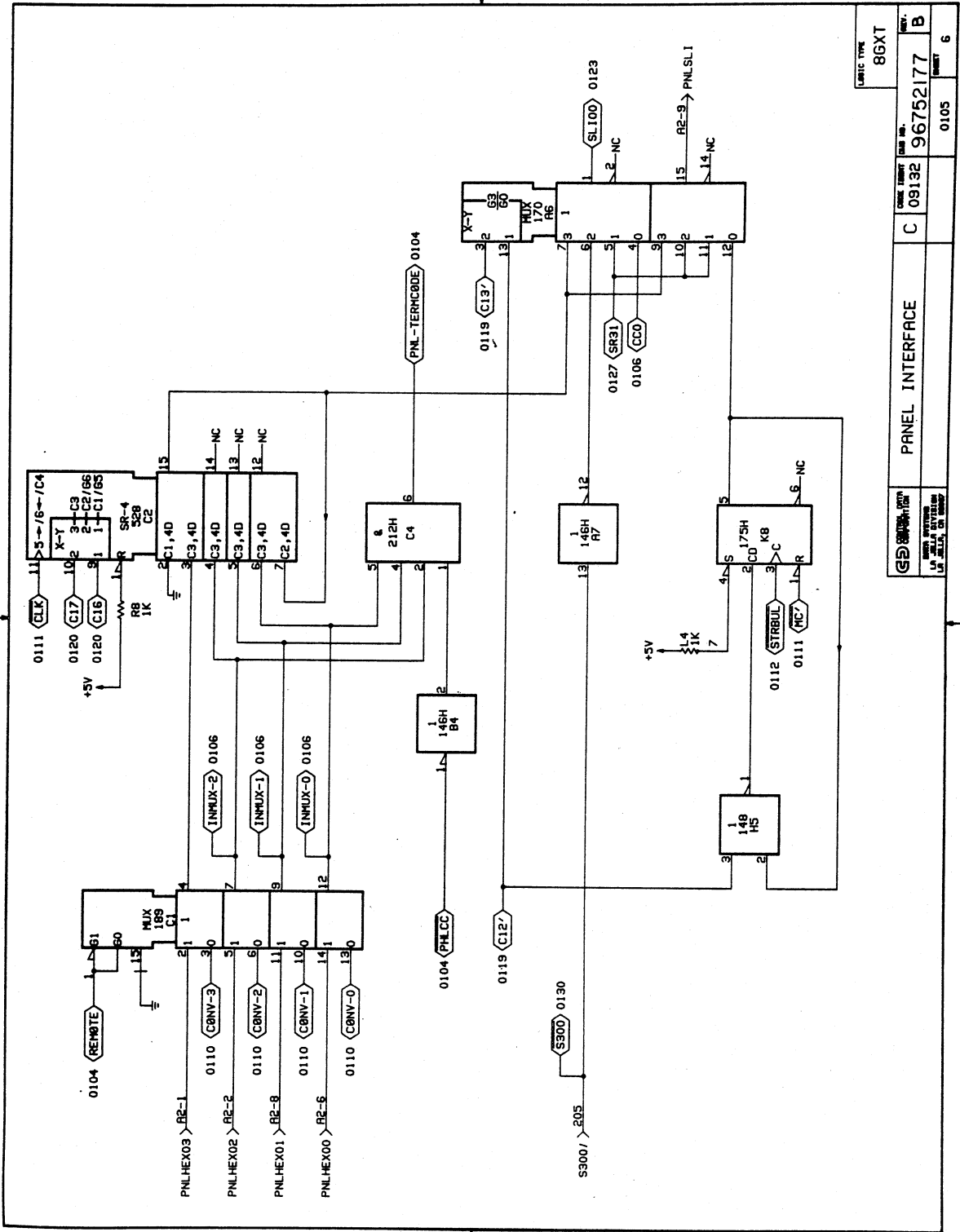


Figure 5-2. Breakpoint Panel Logic Diagram (Sheet 2 of 2)



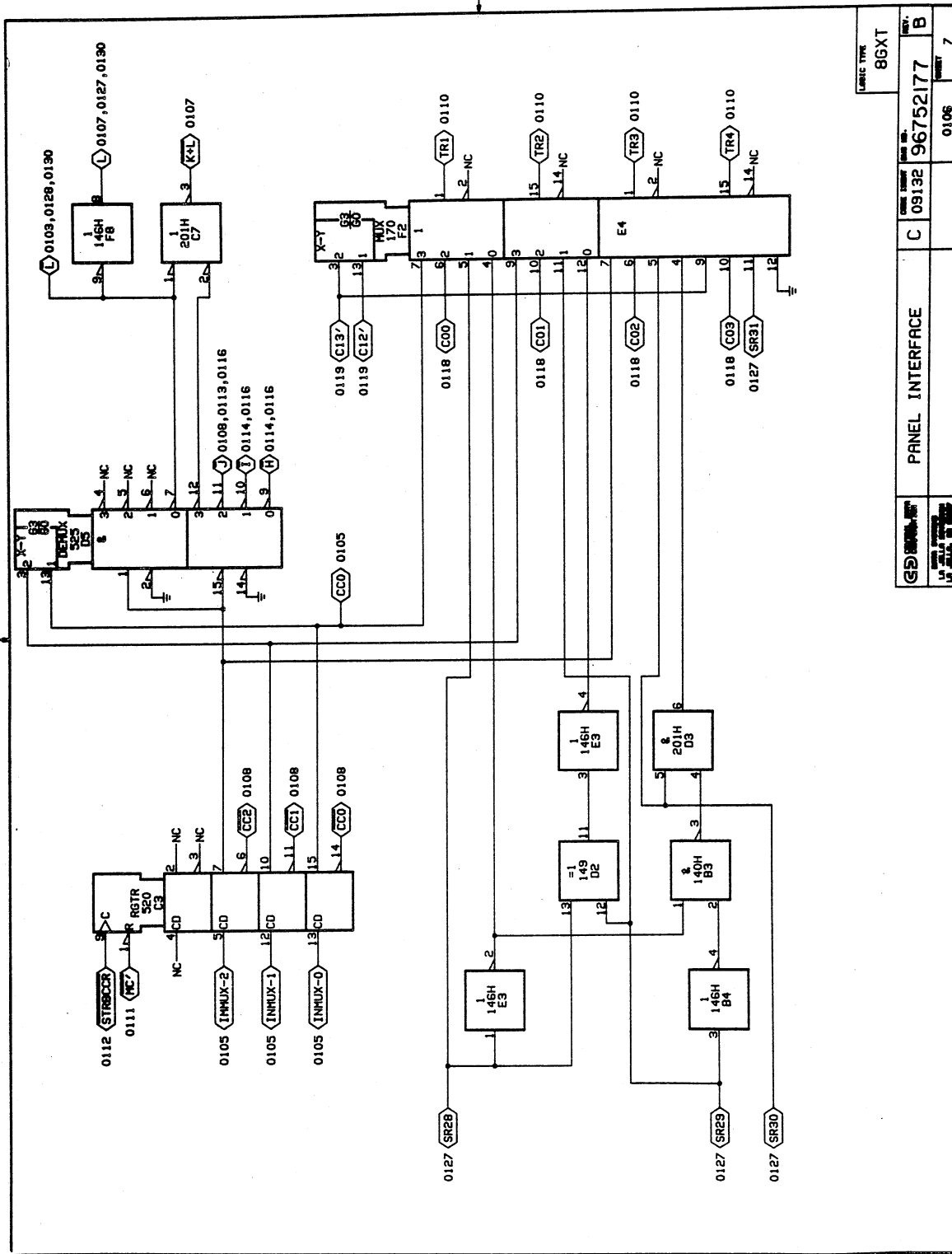
 <small> GE GENERAL ELECTRIC MADE IN THE U.S.A. U.S. PAT. & TRADEMARK OFF. </small>	PANEL INTERFACE		C 09132	96752177	B
			0102	0102	3
LOGIC TYPE 8GXT					

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 3 of 32)



DATE TYPE	86XT
CONTRACT NO.	09132
REV.	B
WORK ORDER NO.	96752177
DATE	0105
REVISION	6

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 6 of 32)



LOGIC TYPE	86XT
REV.	7
DATE	0106
DESIGN	96752177
PROJECT	B
DESCRIPTION	PANEL INTERFACE
REV.	0106
DATE	7

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 7 of 32)

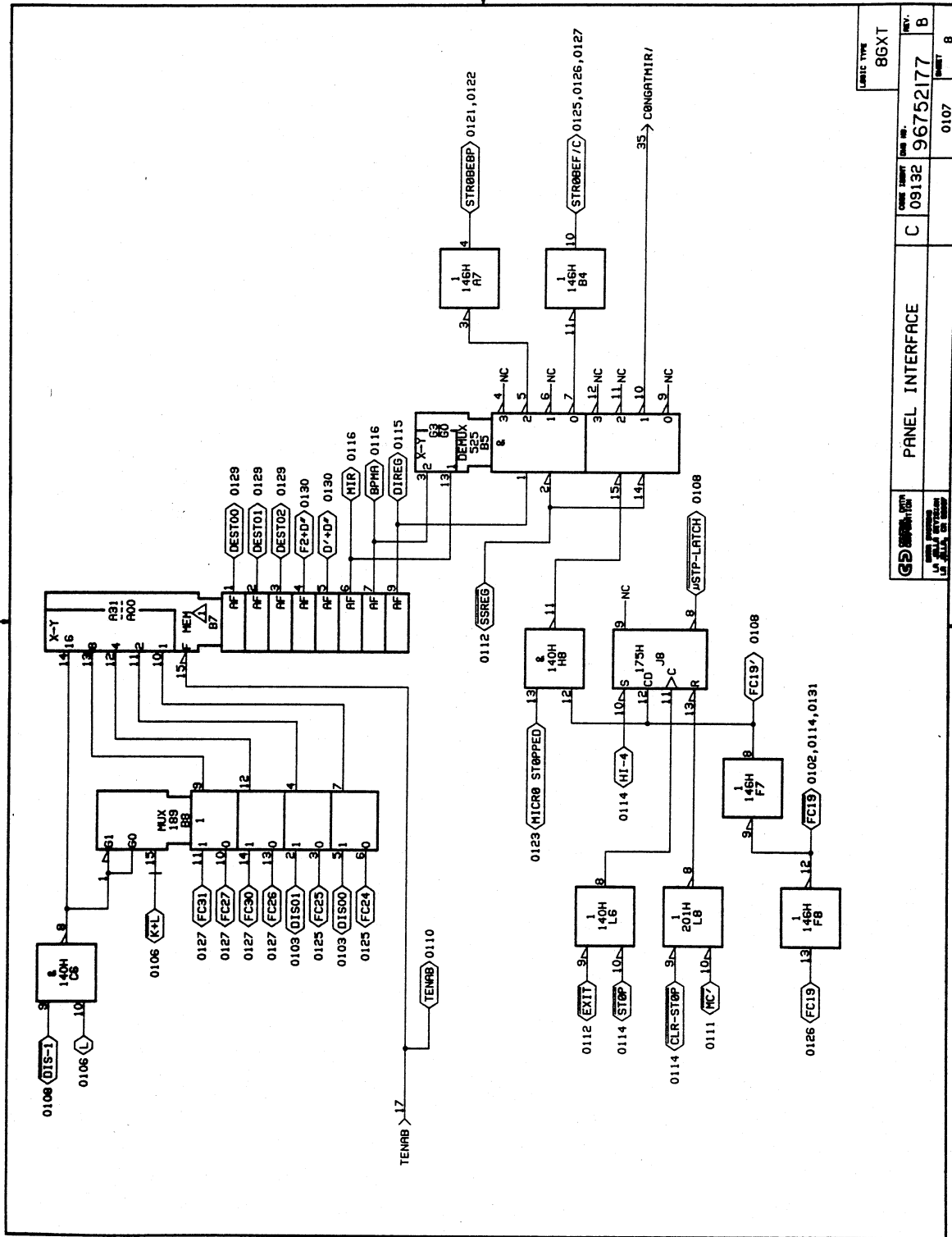
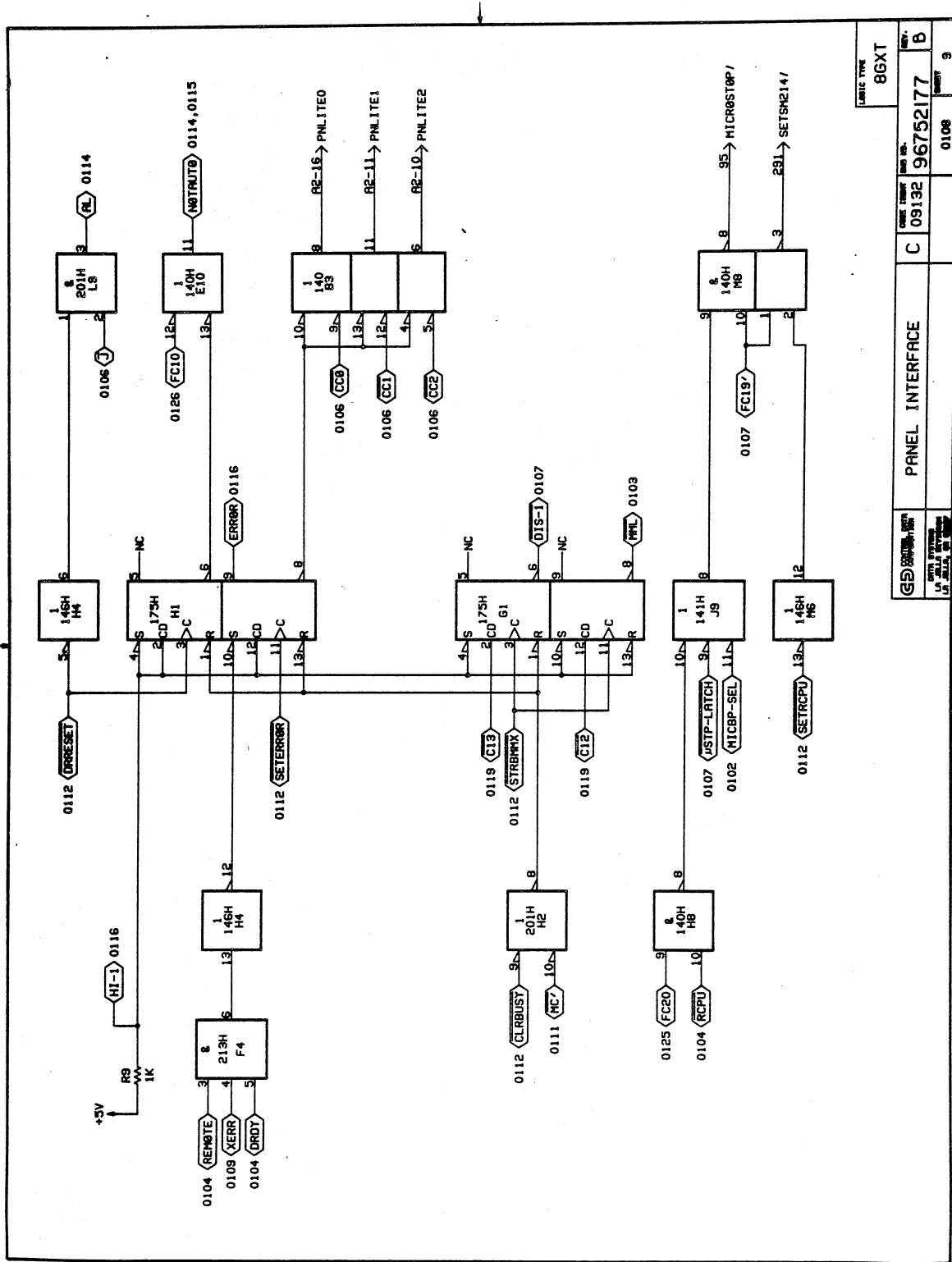


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 8 of 32)



LUMIC TYPE		8GXT	
ORDER NUMBER	REV. NO.	ORDER NUMBER	REV. NO.
C	09132	9675217	B
PANEL INTERFACE		0106	0108
		0106	0108
		0106	0108

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 9 of 32)

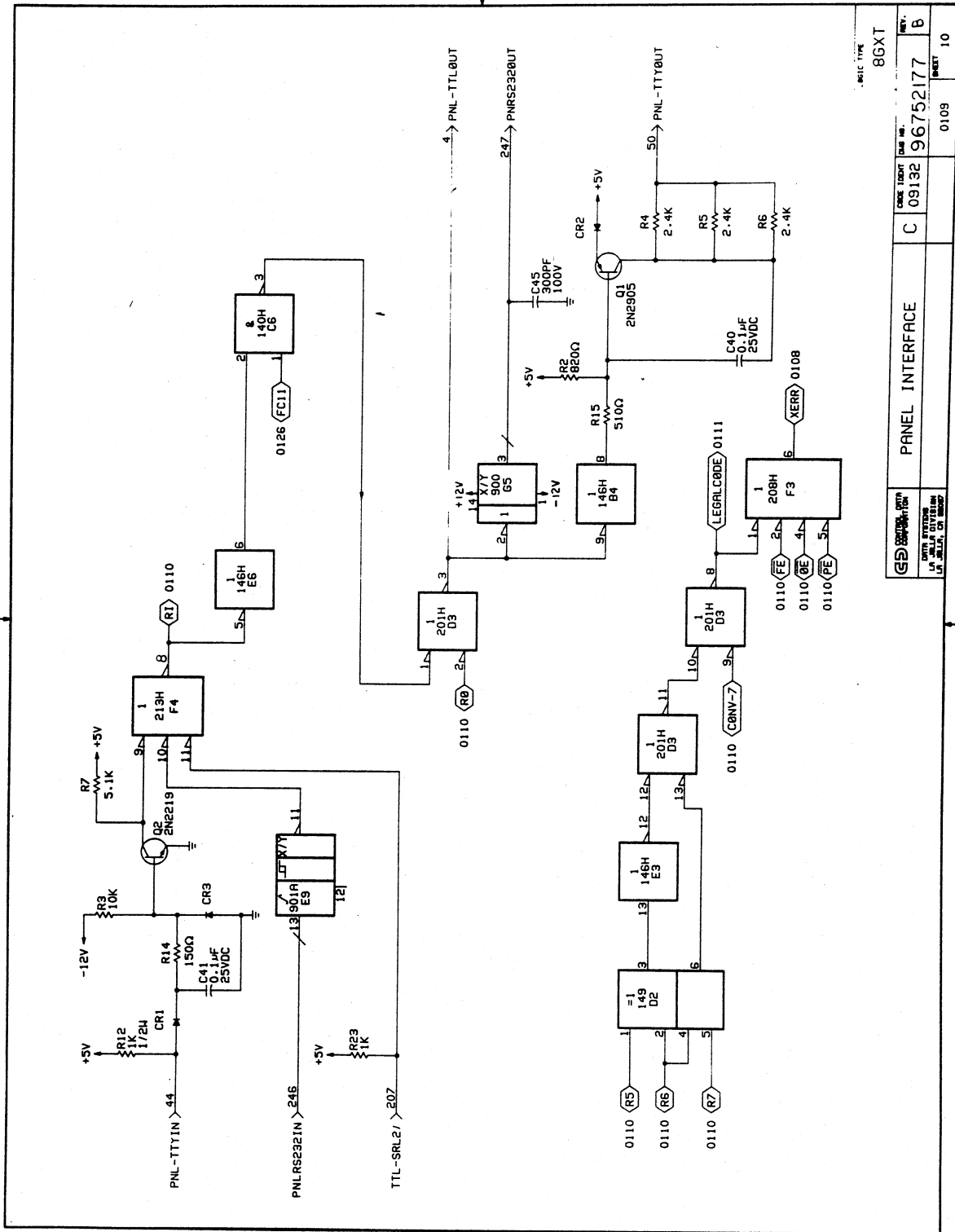


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 10 of 32)

86XT		REV. B	
C 09132		96752177	
PANEL INTERFACE		0109 10	
DATE SPECIFIED BY: J. J. WILSON		DATE CHECKED BY: J. J. WILSON	
DATE APPROVED BY: J. J. WILSON		DATE CHECKED BY: J. J. WILSON	

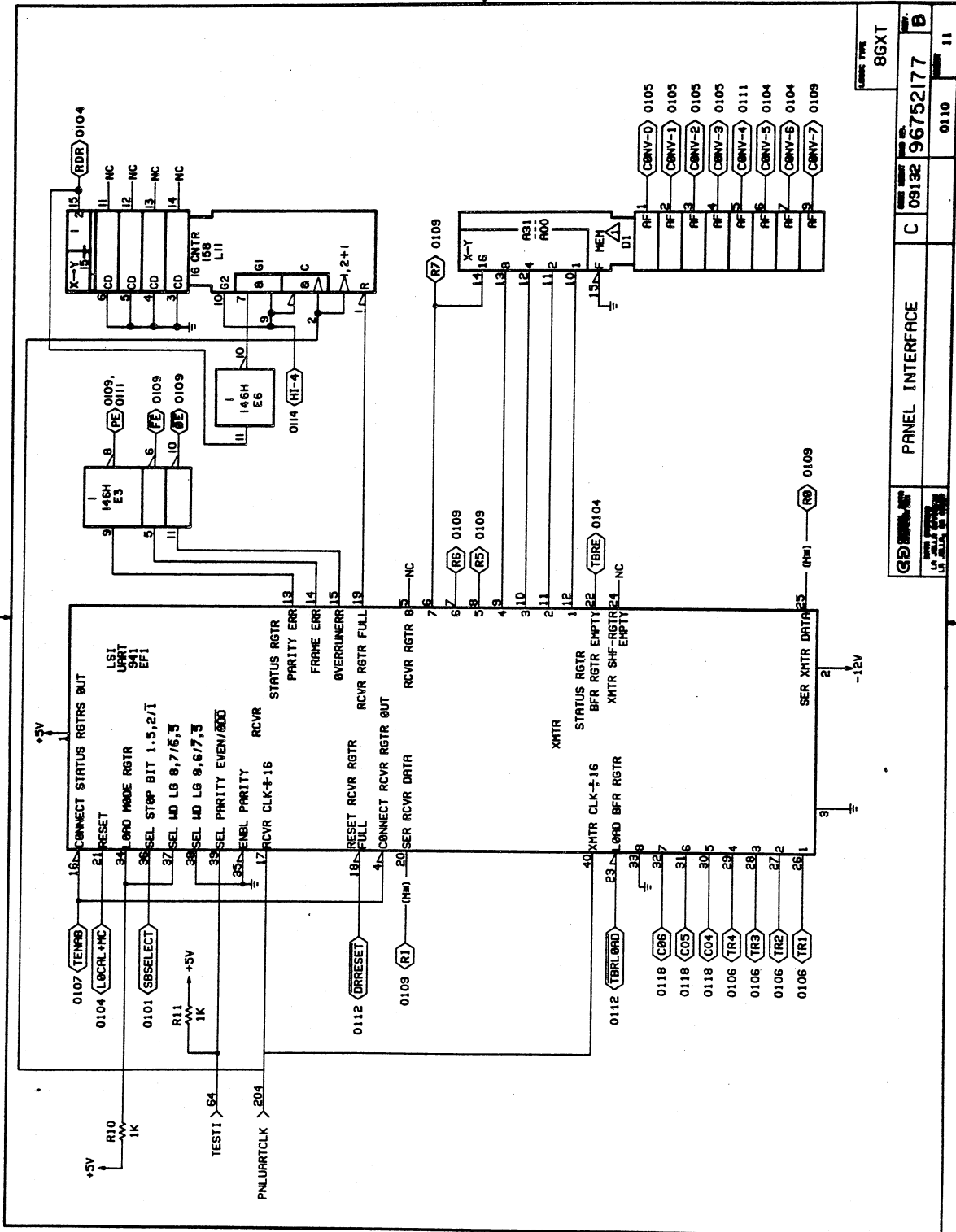
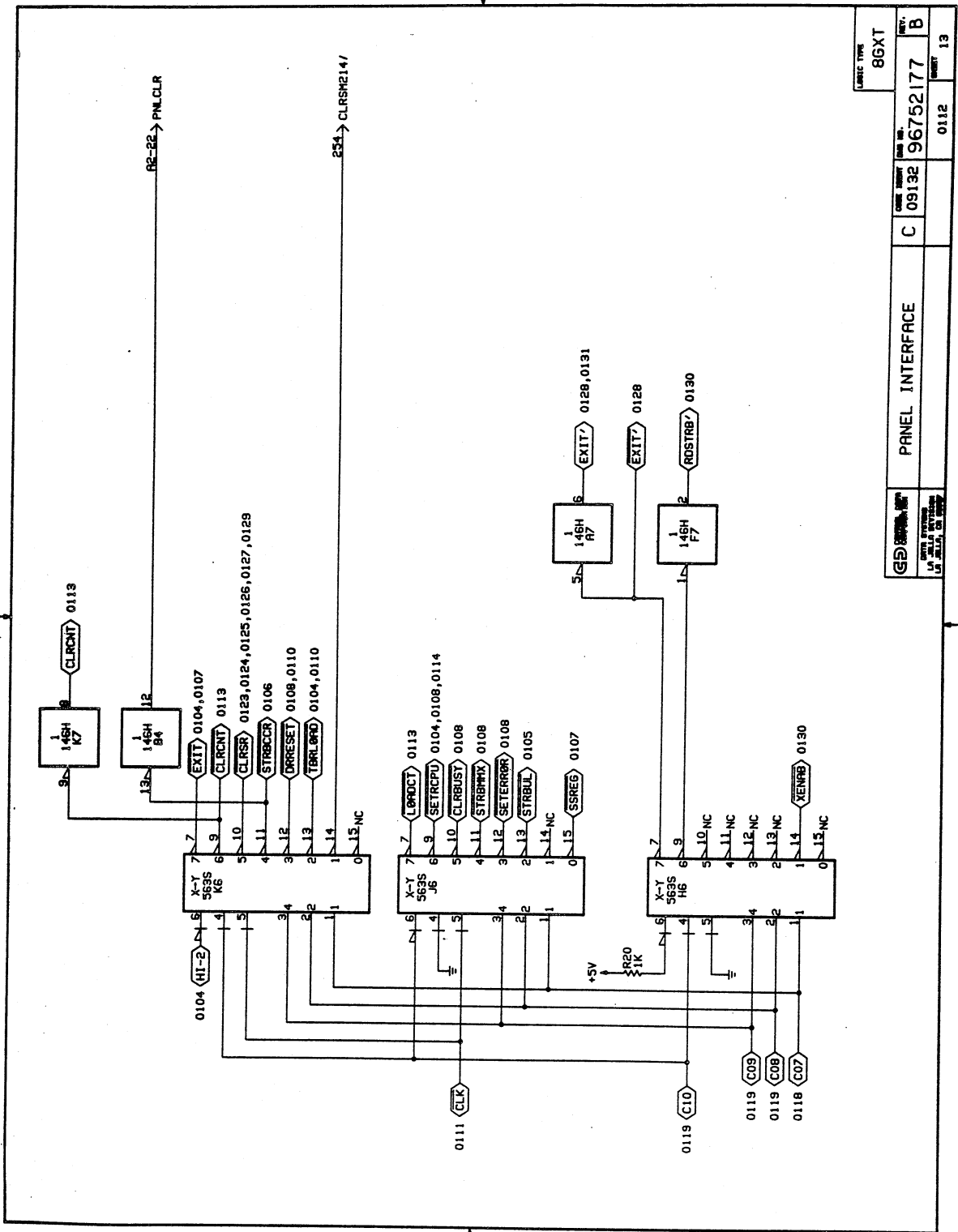
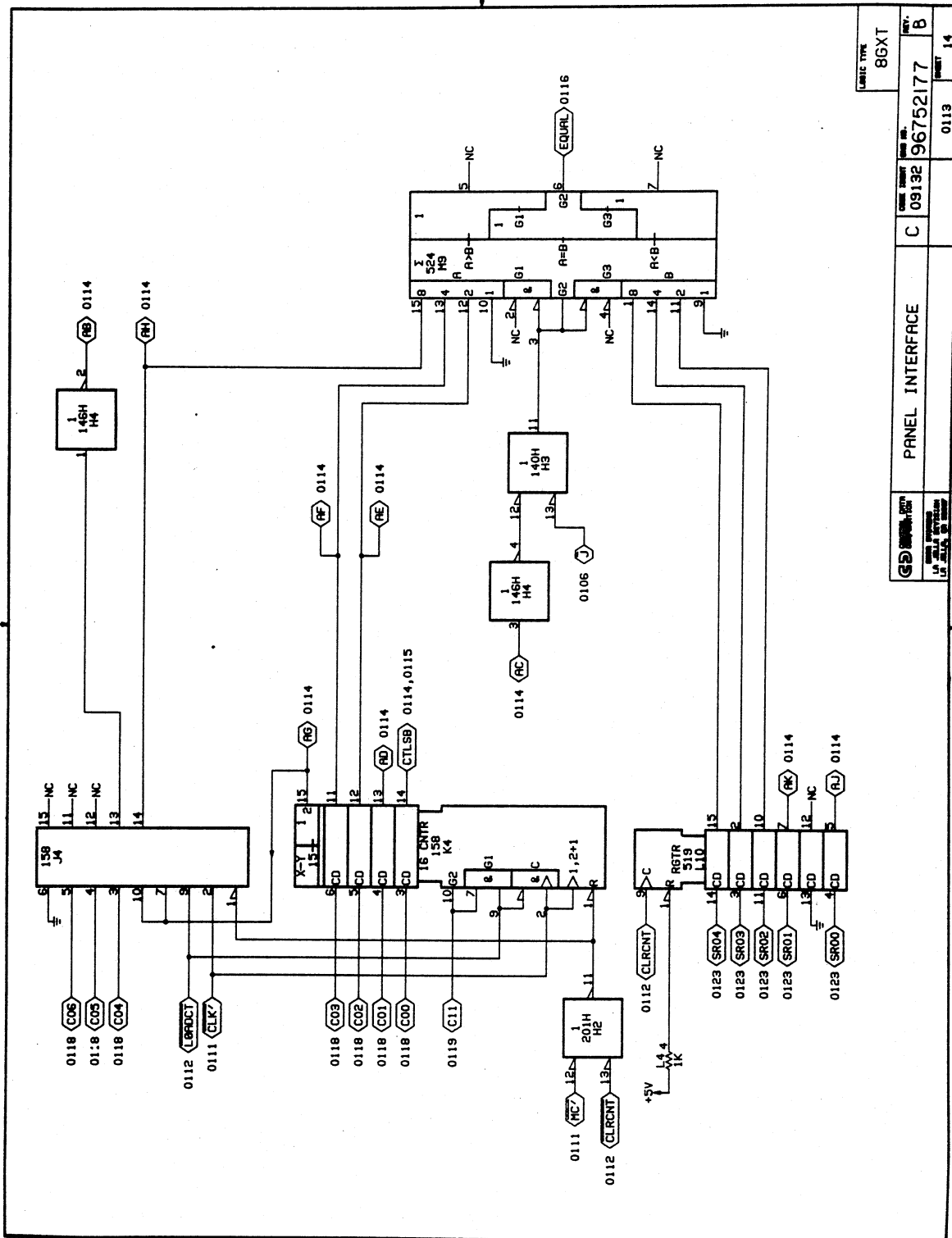


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 11 of 32)



GE	PANEL INTERFACE	C	09132	96752177	B
<small> GEORGE EASTMAN COMPANY 100 WEST 42ND STREET NEW YORK, N.Y. 10018 </small>					
<small> LOGIC TYPE 8GXT </small>				<small> REV. 0112 </small>	<small> SHEET 13 </small>

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 13 of 32)



LOGIC TYPE	8GXT
REV.	B
DATE	09132
QUANTITY	0113
REVISION	14
DESCRIPTION	PANEL INTERFACE
PROJECT	C 09132
DATE	09132
REV.	B
QUANTITY	0113
REVISION	14

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 14 of 32)

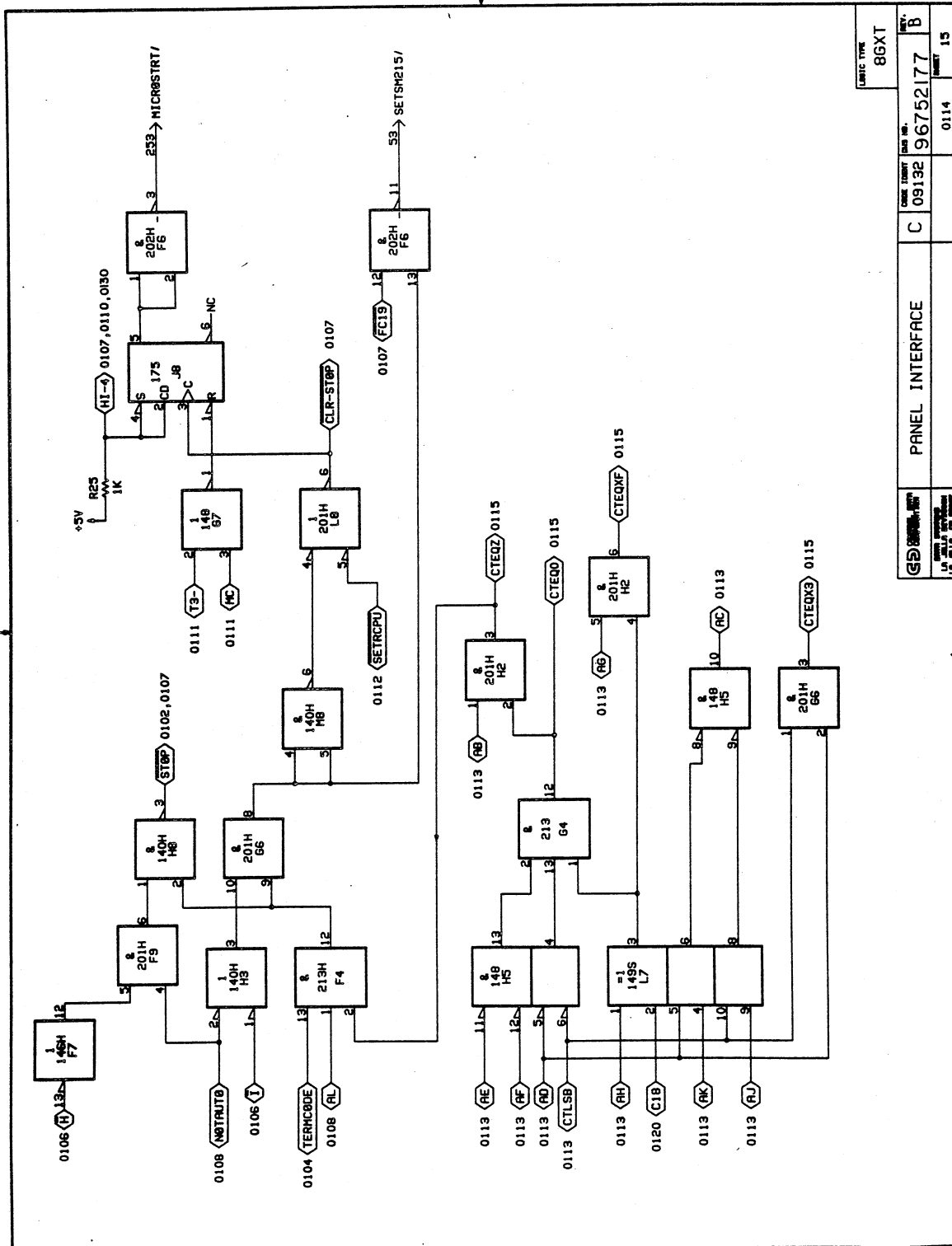


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 15 of 32)

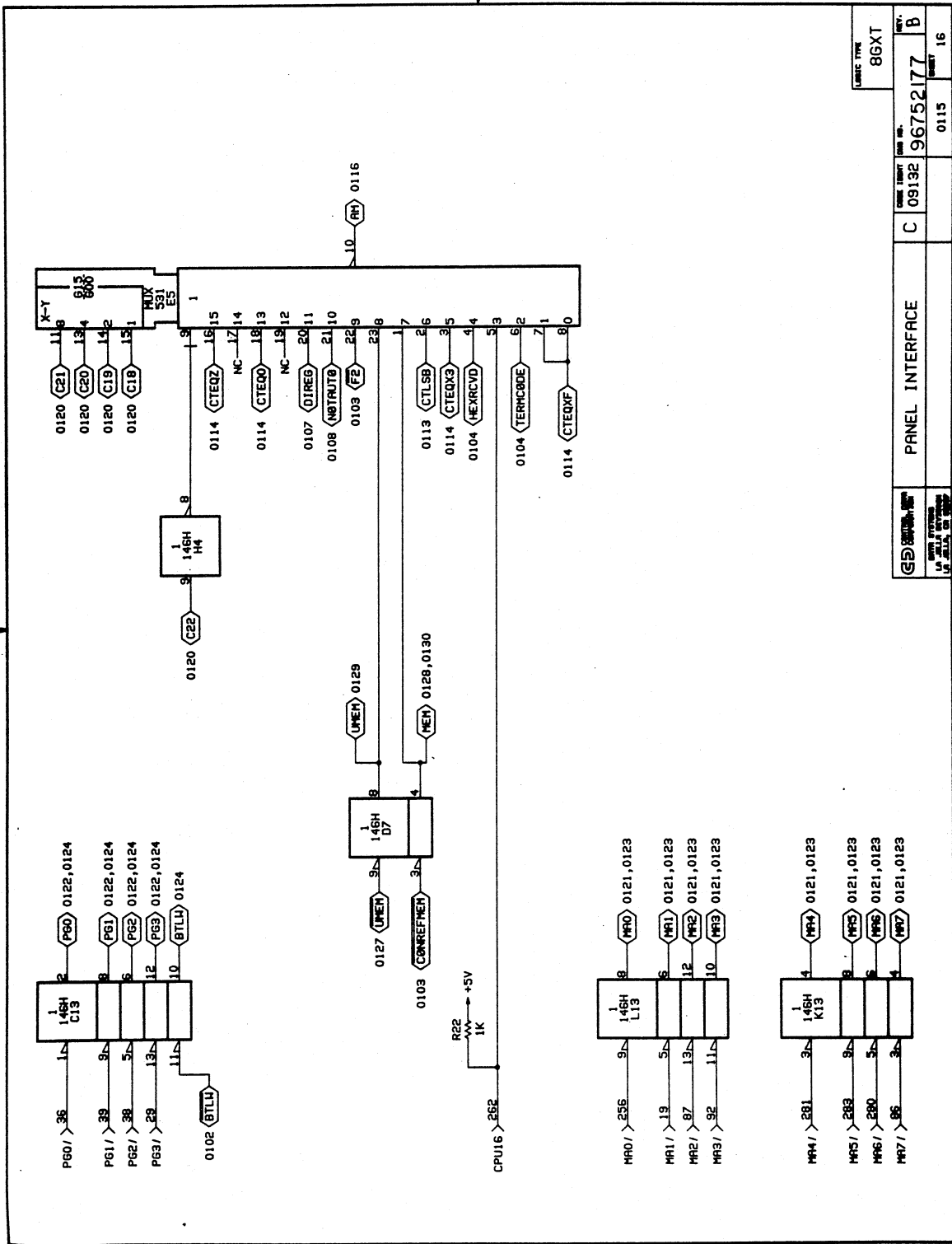


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 16 of 32)

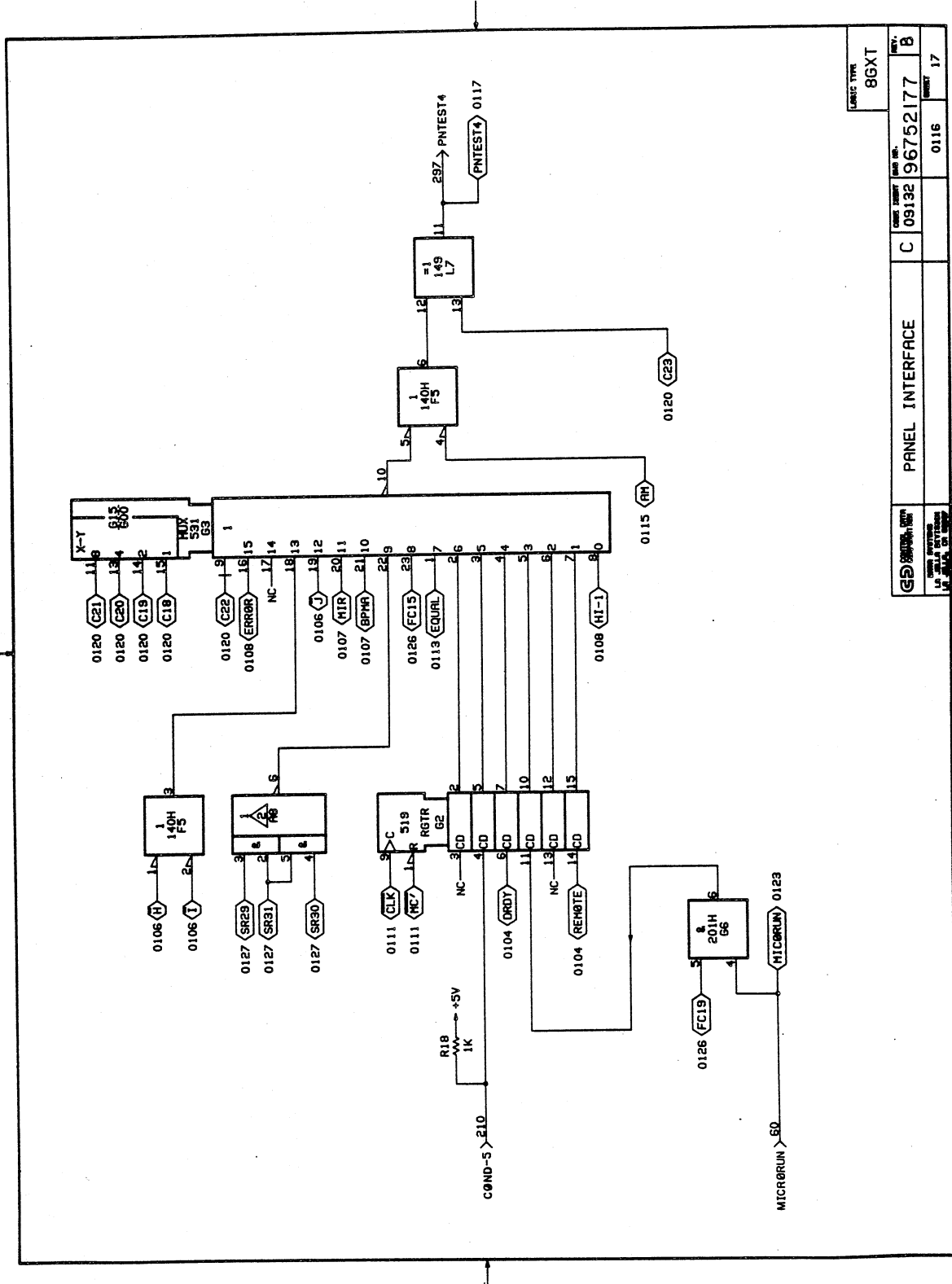


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 17 of 32)

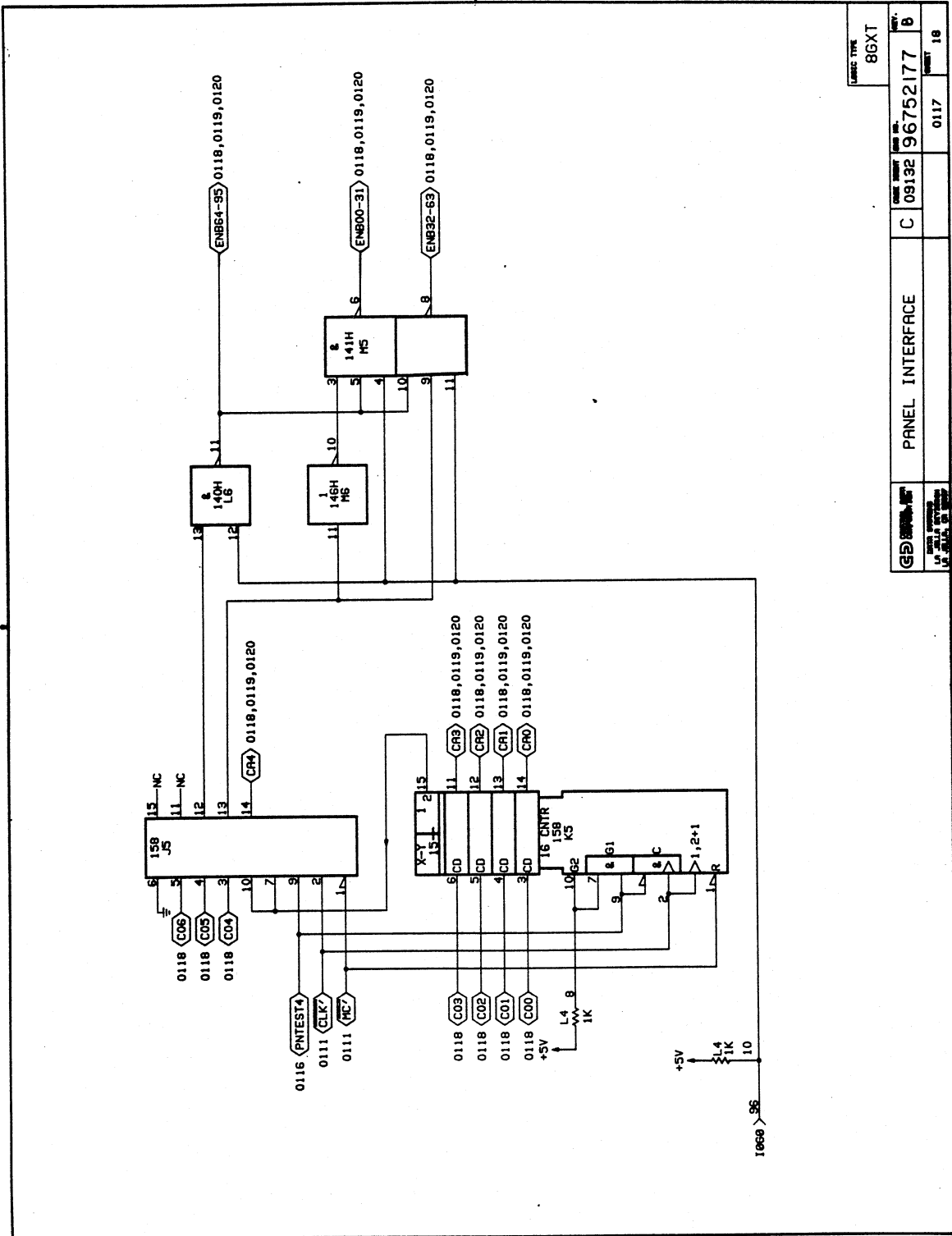
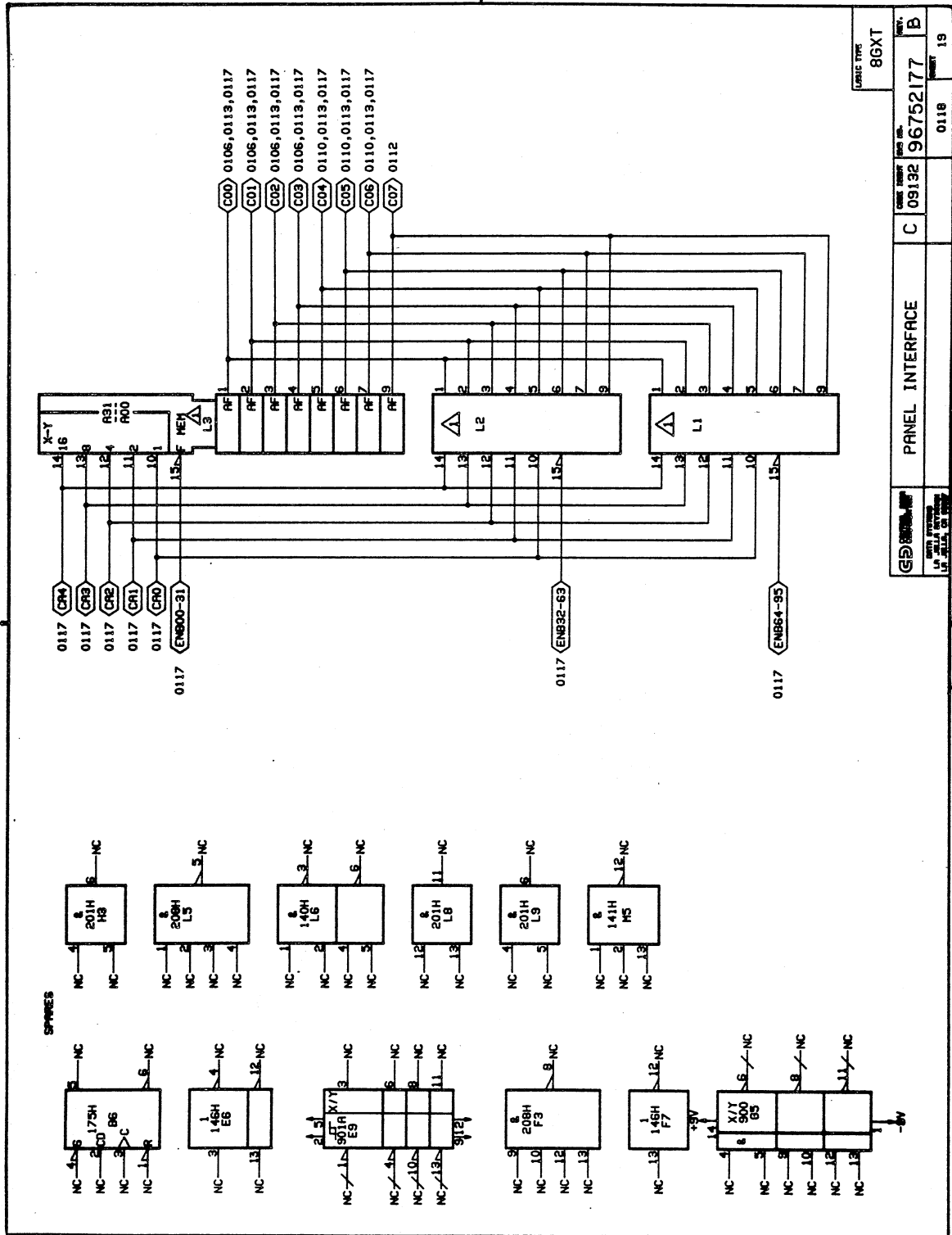


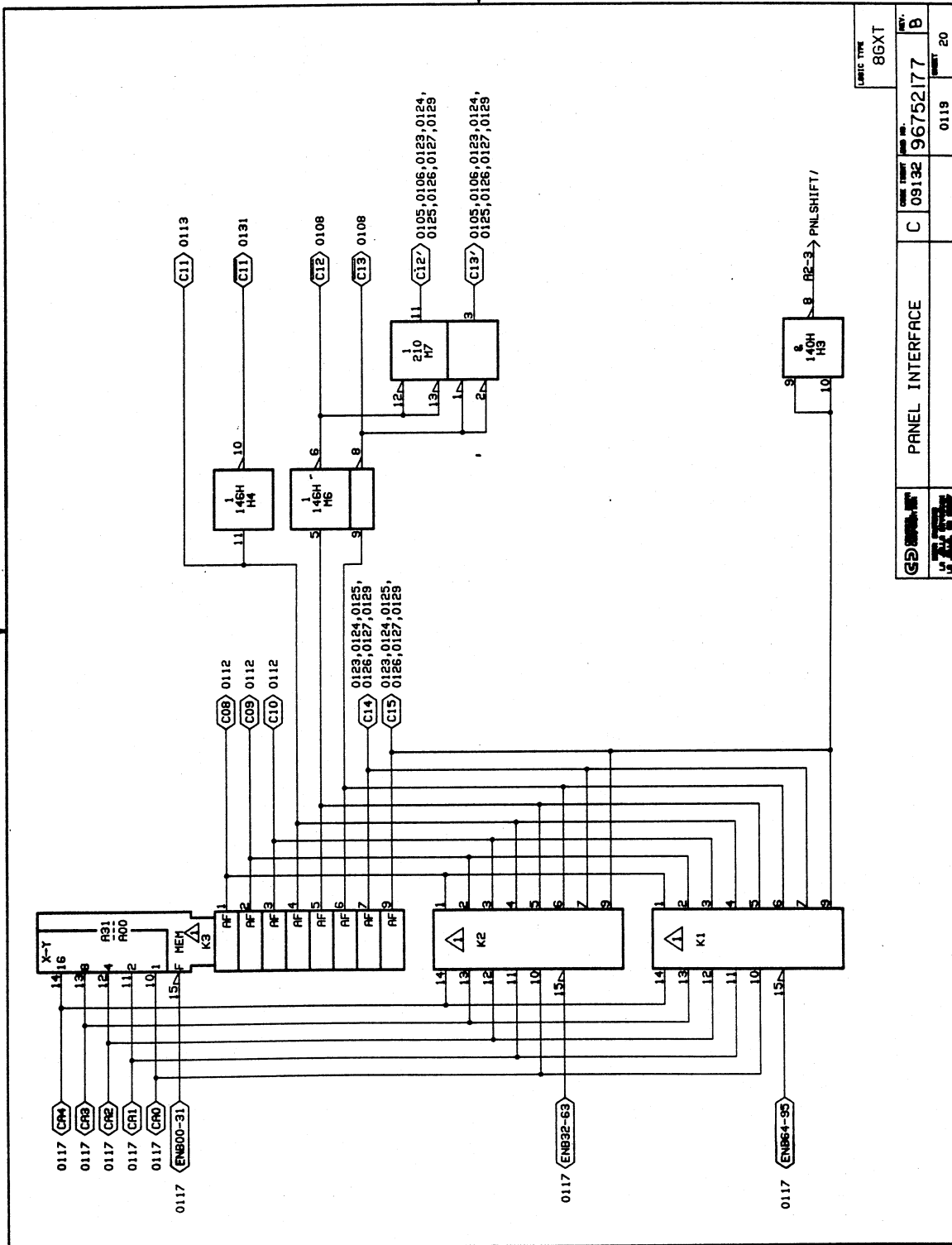
Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 18 of 32)

LOGIC TYPE	8GXT
DATE REVISED	0117
REV.	18
DATE REVISED	09132
REV.	96752177
DATE REVISED	C
REV.	PANEL INTERFACE



GE	86XT
REV. B	REV. B
DATE 09132	DATE 09132
BY 96752177	BY 96752177
0118	0118
19	19

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 19 of 32)



LOGIC TYPE		8GXT	
GD	PANEL INTERFACE	C	09132 96752177
		REV. B	
		0119	20

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 20 of 32)

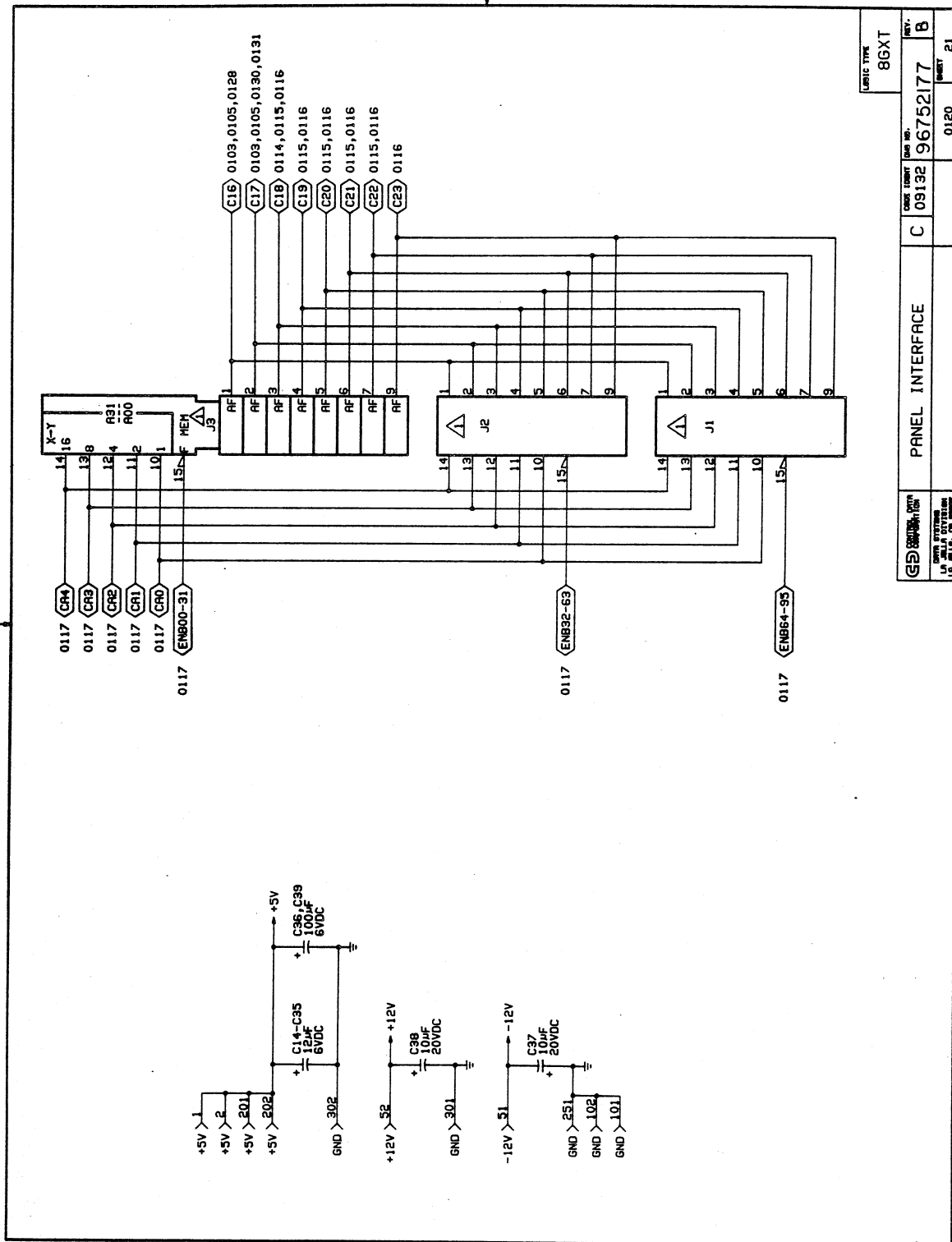


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 21 of 32)

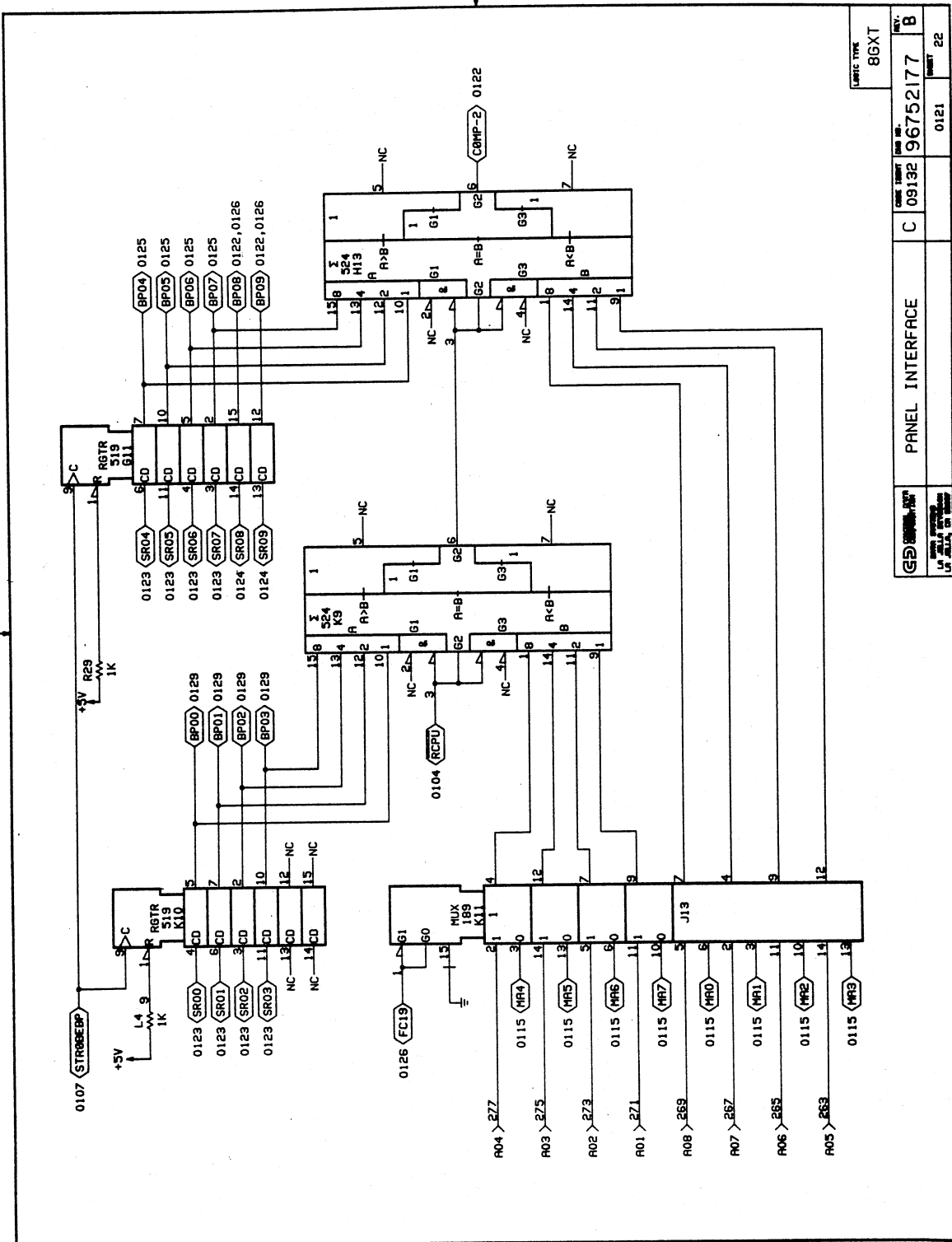


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 22 of 32)

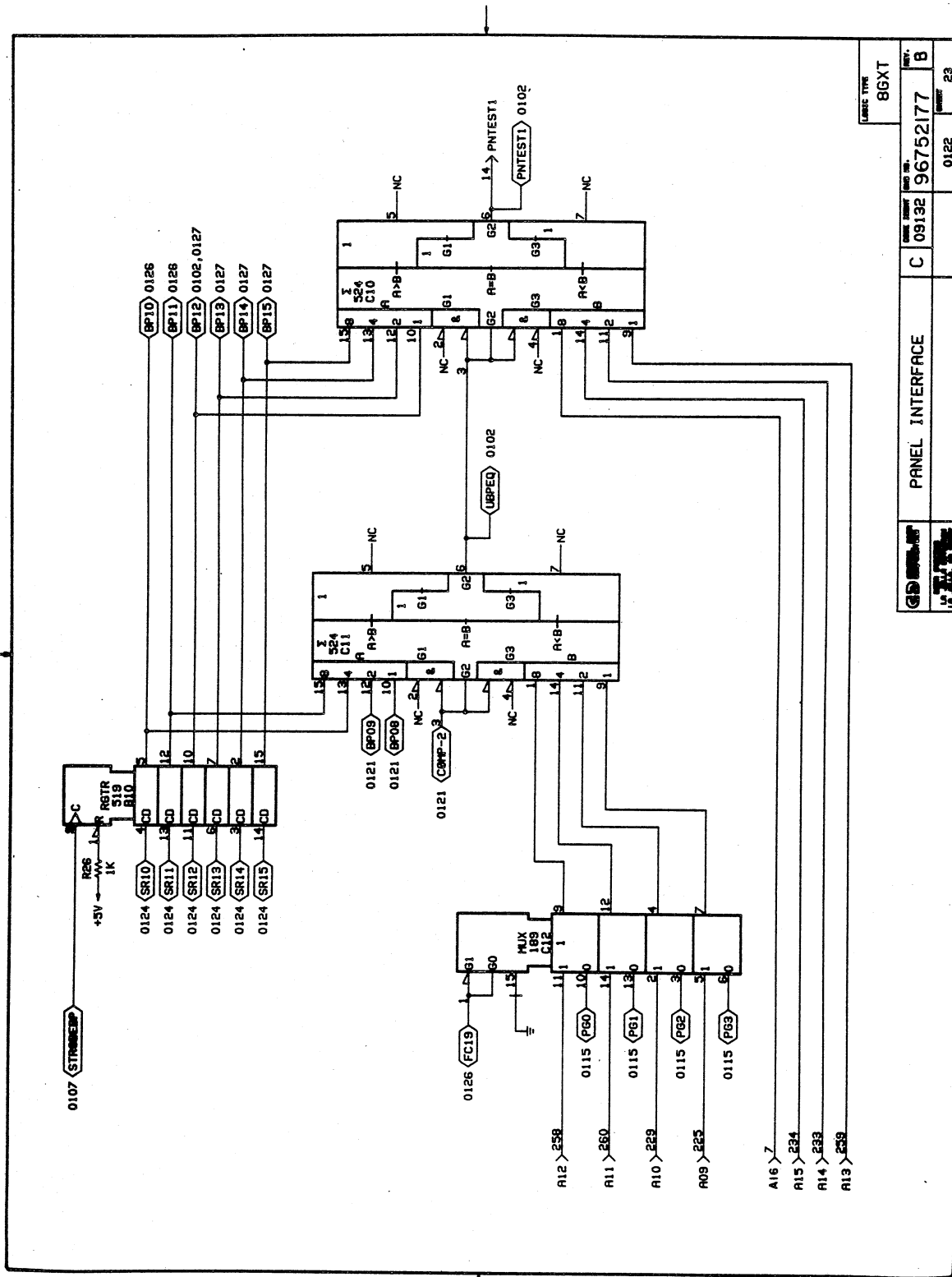


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 23 of 32)

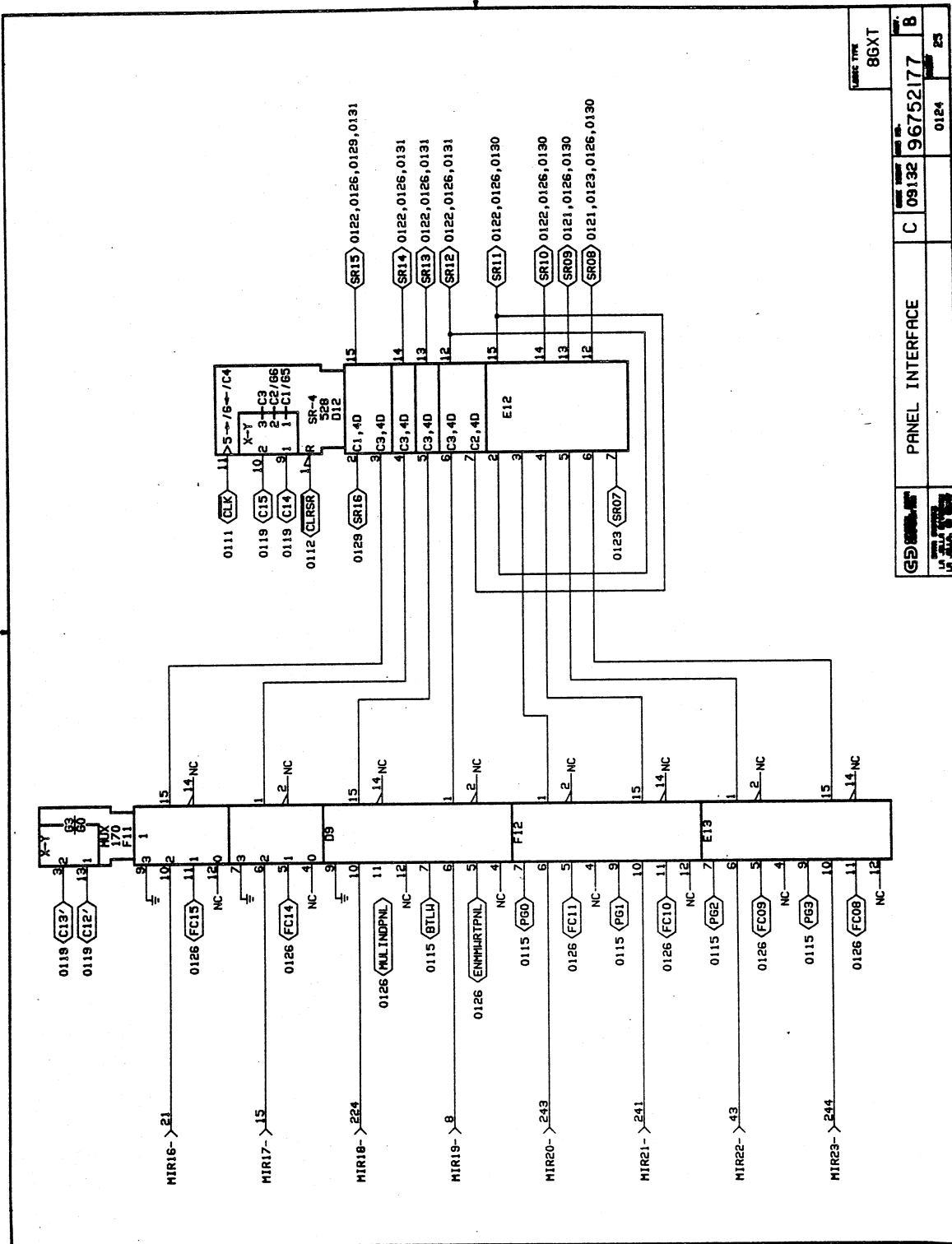
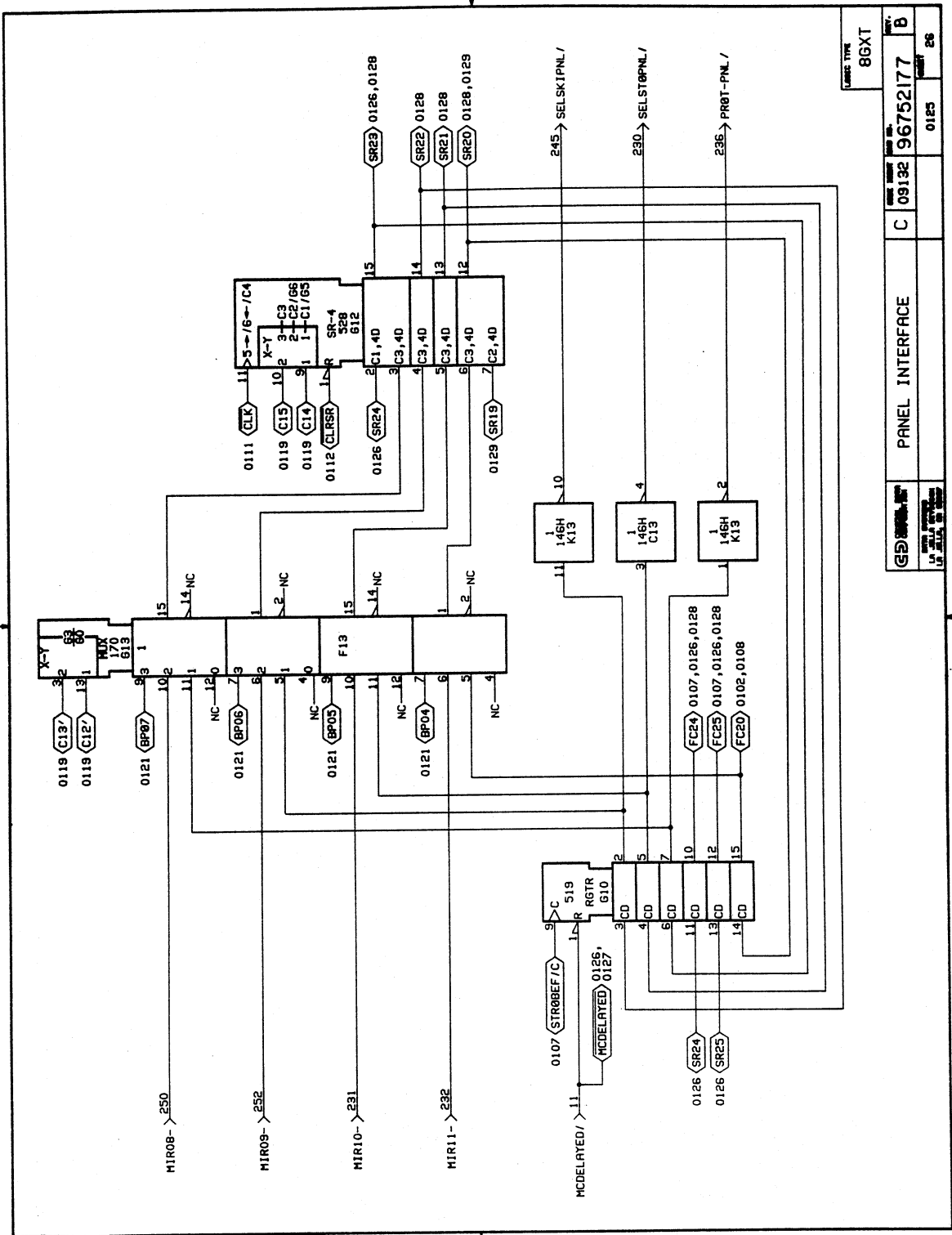
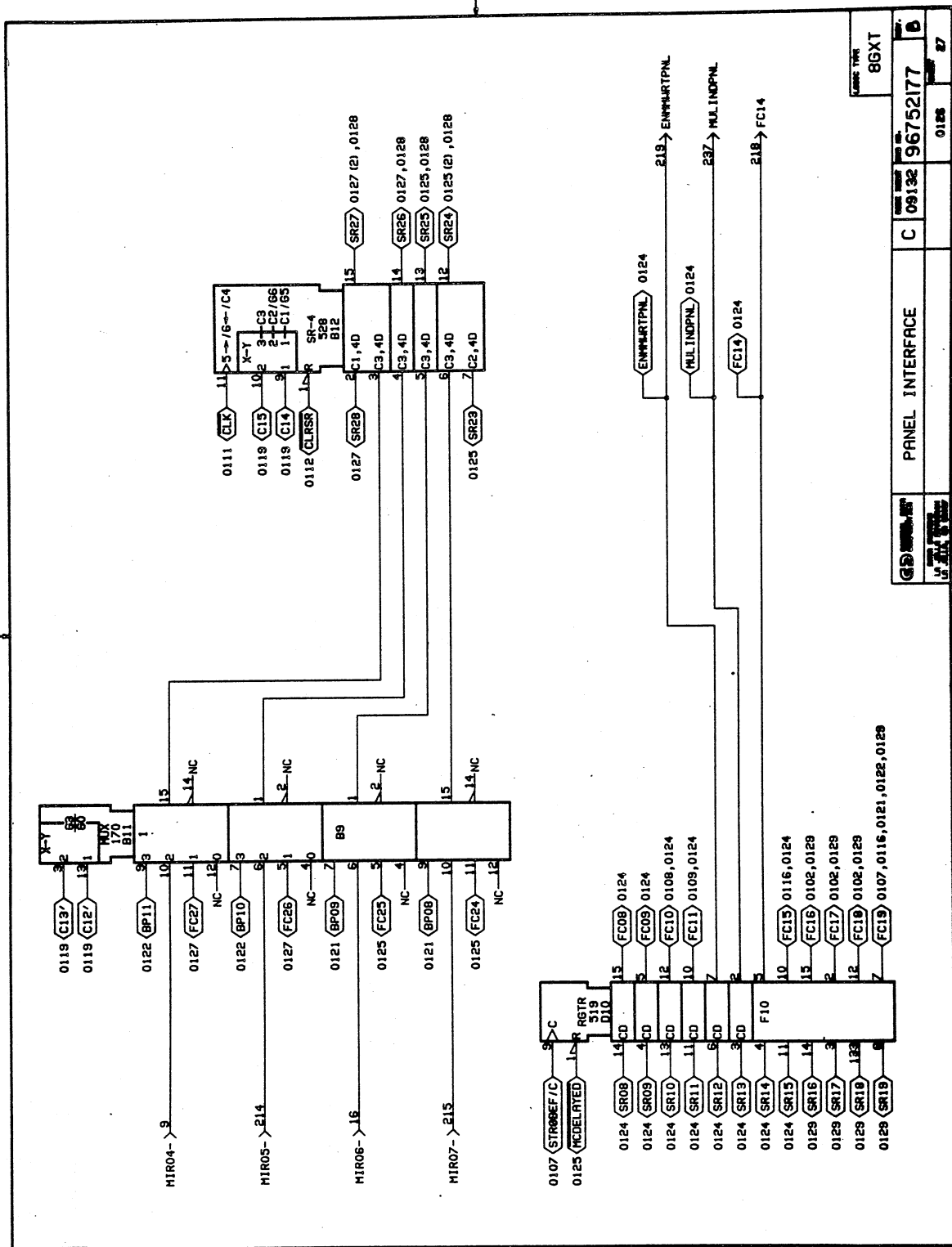


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 25 of 32)



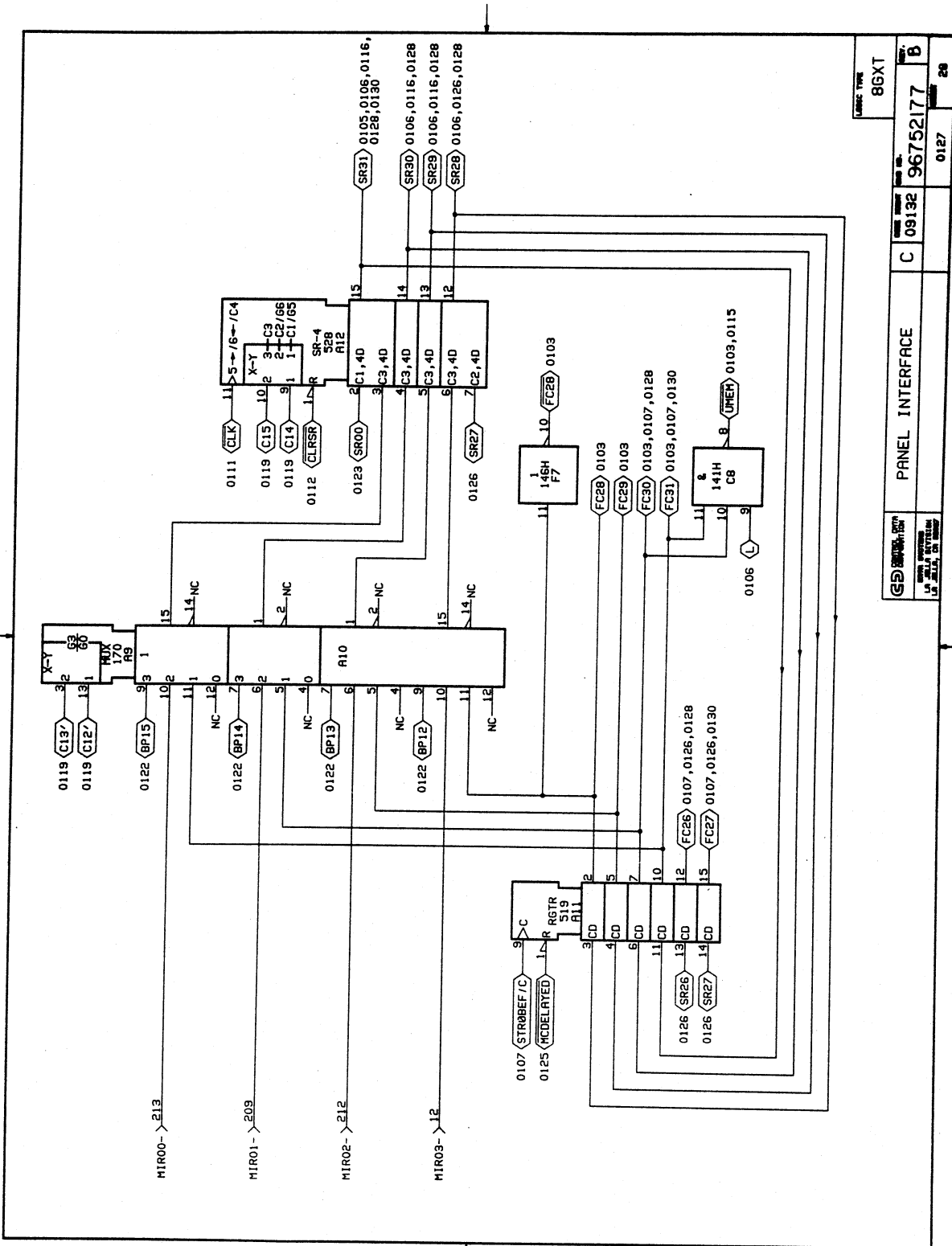
C 09132		PANEL INTERFACE		8GXT	
96752177		C 09132		96752177	
0125		0125		26	

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 26 of 32)



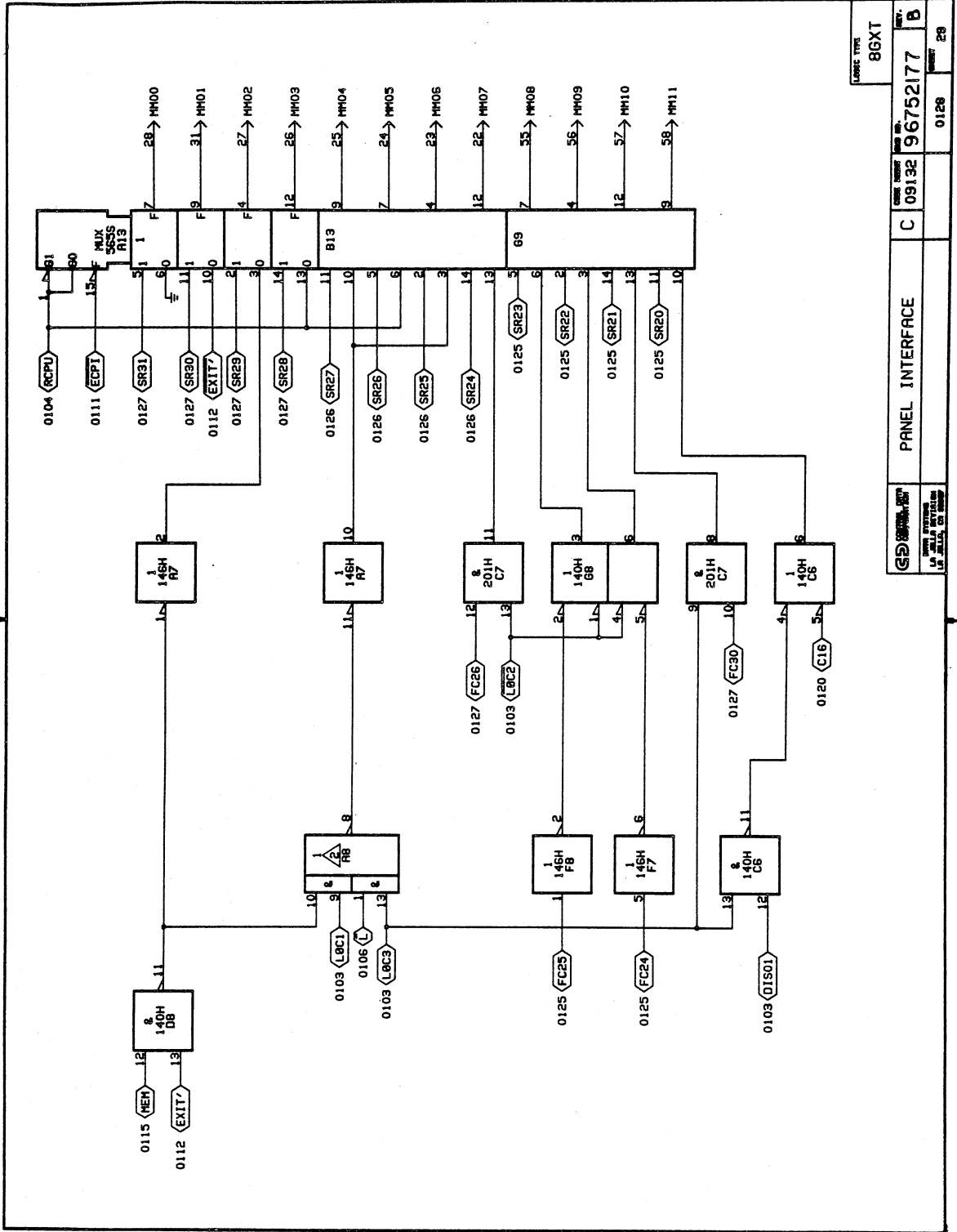
PANEL INTERFACE		C 09132		96752177		0128		27	
86XT									

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 27 of 32)



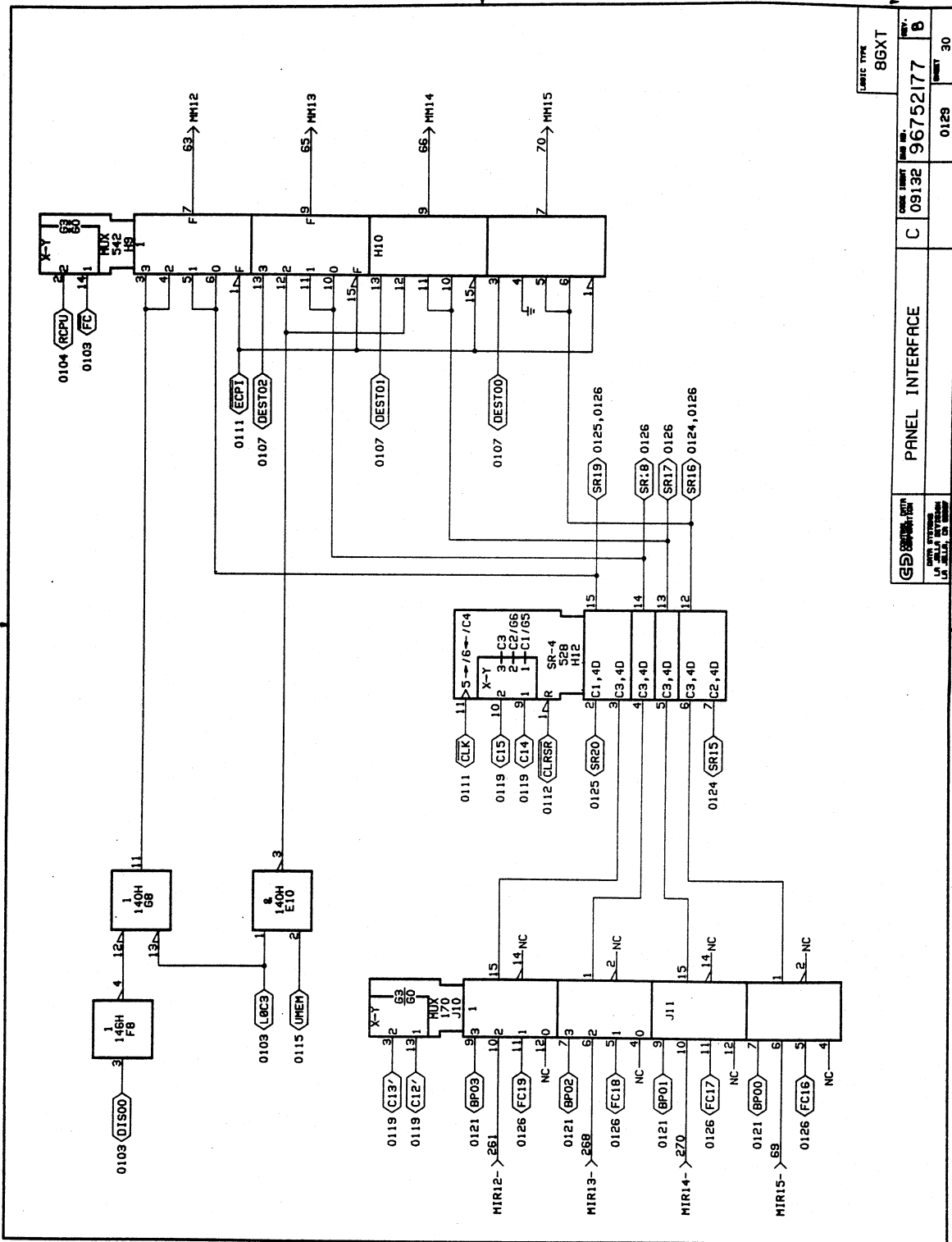
	PANEL INTERFACE	C	96752177	B	0127	28	86XT
							09132

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 28 of 32)



LOGIC TYPE		86XT	
DATE	REV.	DATE	REV.
09132	C	96752176	B
PANEL INTERFACE		0126	29

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 29 of 32)



LUBRIC TYPE		8GXT	
DATE	REV.	DATE	REV.
09132	96752177	0129	30
C		PANEL INTERFACE	
GD		PANEL INTERFACE	

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 30 of 32)

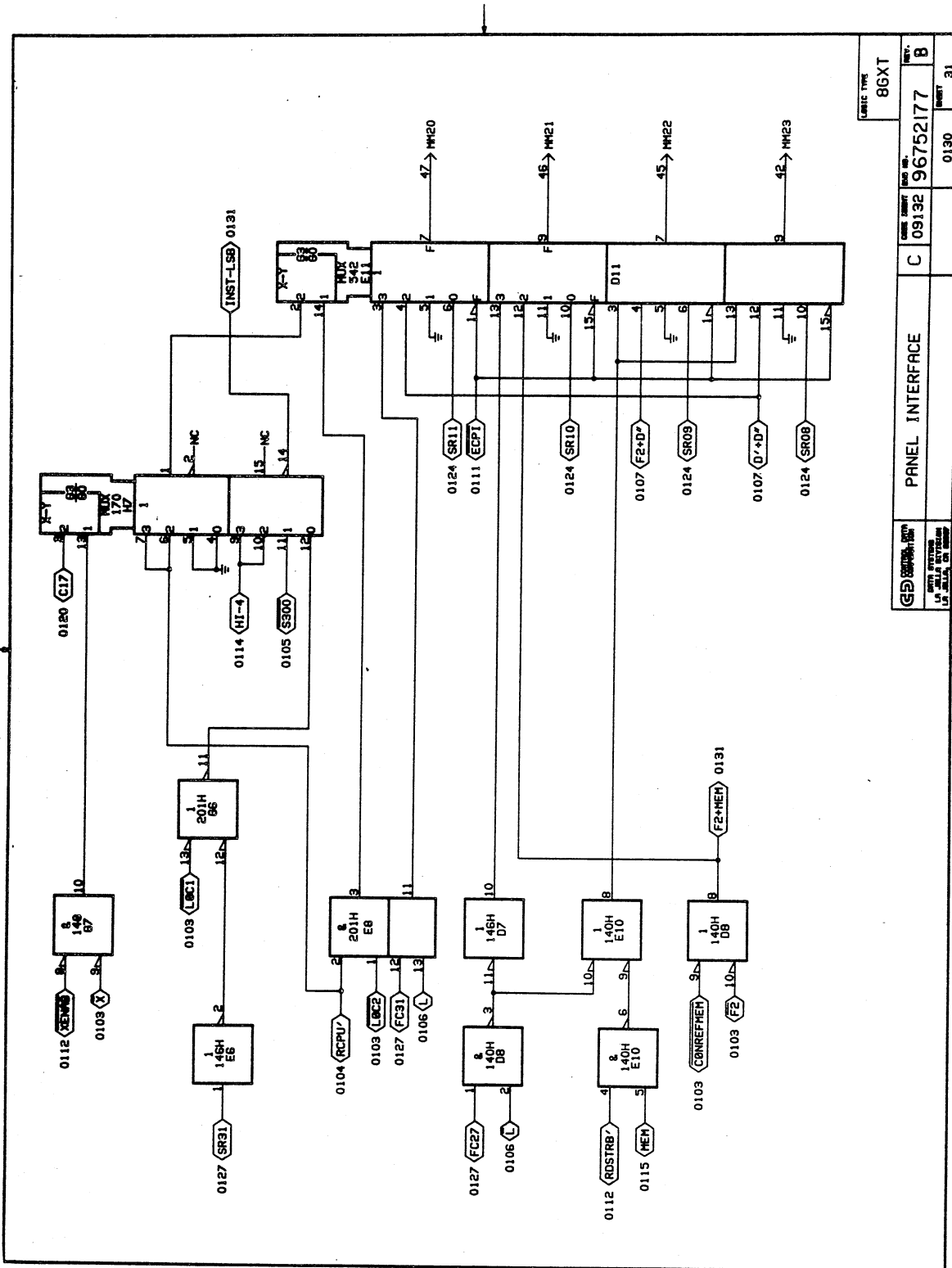
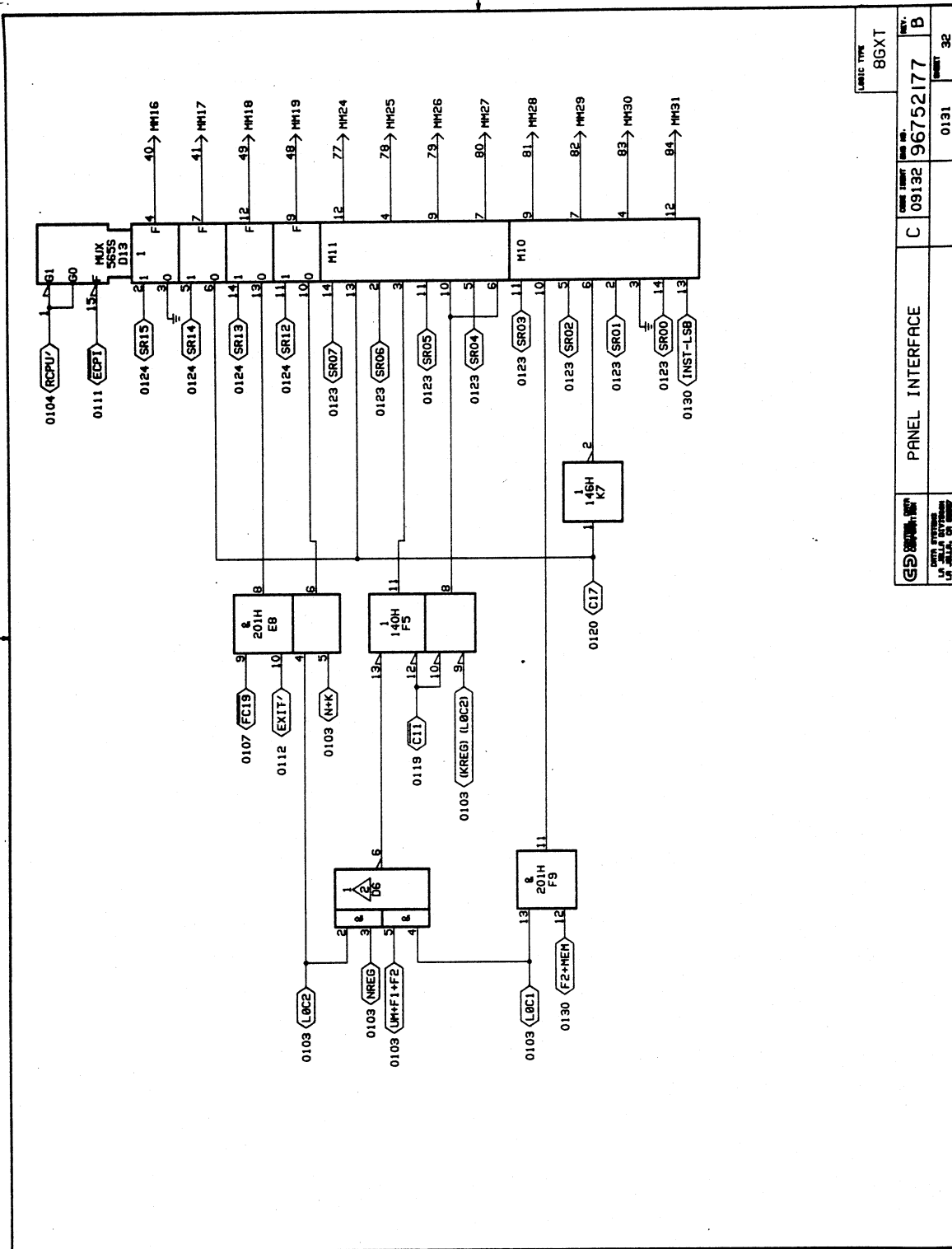


Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 31 of 32)



LOGIC TYPE	86XT
COMP. PART NO.	09132
REV.	B
QUANTITY	32
PANEL INTERFACE	C
96752176	0131

Figure 5-3. Breakpoint Controller (PWA 96752176) Logic Diagram (Sheet 32 of 32)

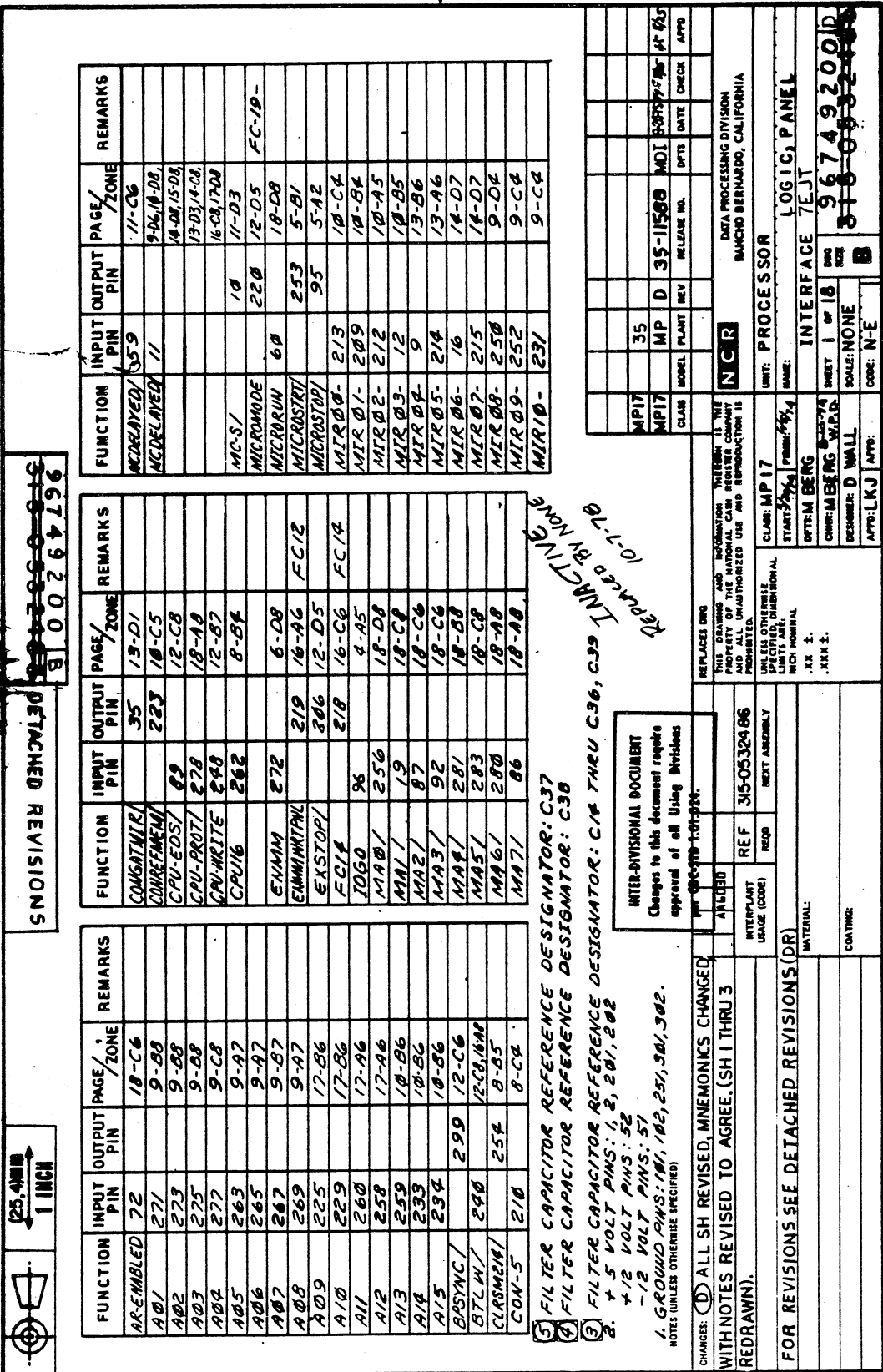


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 1 of 18)

96749200 B
315-053-2405



FUNCTION	INPUT PIN	OUTPUT PIN	PAGE/ ZONE	REMARKS
MM15	70		15-B1	
MM16	40		16-C1	
MM17	41		16-C1	
MM18	49		16-C1	
MM19	48		16-C1	
MM20	47		17-D1	
MM21	46		17-C1	
MM22	45		17-B1	
MM23	42		17-B2	
MM24	77		18-D1	
MM25	78		18-D1	
MM26	79		18-C1	
MM27	80		18-D1	
MM28	81		18-A2	
MM29	82		18-A2	
MM30	83		18-A2	
MM31	84		18-A2	
MULTINDMPL		237	16-B6	FC13

FUNCTION	INPUT PIN	OUTPUT PIN	PAGE/ ZONE	REMARKS
MIR31-	88		18-B8	
MM00	28		14-B4	
MM01	31		14-A4	
MM02	27		14-B4	
MM03	26		14-A4	
MM04	25		14-B4	
MM05	24		14-C4	
MM06	23		14-C4	
MM07	22		14-G4	
MM08	55		14-D1	
MM09	56		14-D1	
MM10	57		14-D1	
MM11	58		14-C1	
MM12	63		14-C1	
MM13	65		14-B1	
MM14	66		15-B1	

FUNCTION	INPUT PIN	OUTPUT PIN	PAGE/ ZONE	REMARKS
MIR11-	232		9-B4	
MIR12-	261		15-D6	
MIR13-	268		15-C6	
MIR14-	270		15-B6	
MIR15-	69		15-B6	
MIR16-	21		16-C7	
MIR17-	15		16-C6	
MIR18-	224		16-B7	
MIR19-	8		16-B6	
MIR20-	243		17-D6	
MIR21-	241		17-D6	
MIR22-	43		17-B6	
MIR23-	244		17-C6	
MIR24-	62		18-D8	
MIR25-	61		18-C8	
MIR26-	287		18-C6	
MIR27-	286		18-C6	
MIR28-	288		18-B8	
MIR29-	293		18-C8	
MIR30-	75		18-A8	

REV D SEE SHEET ONE FOR CHANGES

THIS DRAWING AND INFORMATION THEREON IS THE PROPERTY OF THE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. ALL RIGHTS RESERVED.
REPRODUCTION IS PROHIBITED.

NGR DATA PROCESSING DIVISION
RANCHO BERNARDO, CALIFORNIA

UNIT: PROCESSOR
NAME: LOGIC, PANEL

INTERFACE 7EJT
SHEET 2 OF 18
SCALE: NONE

96749200
315-053-2405

UNIT: SYSTEMS
SHEET: 18
INCH NOMINAL
.XX ±
.XXX ±

- 8 FULL PART IDENTITY TO BE ESTABLISHED AFTER PROGRAMMING.
7. REFERENCE DESIGNATIONS NOT USED: A1, A3, B1, B2, E1, E2, F1, F2, F3, G1, G2, H1, H2, I1, I2, J1, J2, K1, K2, L1, L2, M1, M2, N1, N2, O1, O2, P1, P2, Q1, Q2, R1, R2, S1, S2, T1, T2, U1, U2, V1, V2, W1, W2, X1, X2, Y1, Y2, Z1, Z2.
6. HIGHEST REFERENCE DESIGNATION: A13, B13, C46, C83, D13, E13, E41, F13, G13, H13, I13, K13, L13, M13, Q2, R30.

NOTES (UNLESS OTHERWISE SPECIFIED)

NC-974 FORM NO. 1-73

Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 2 of 18)

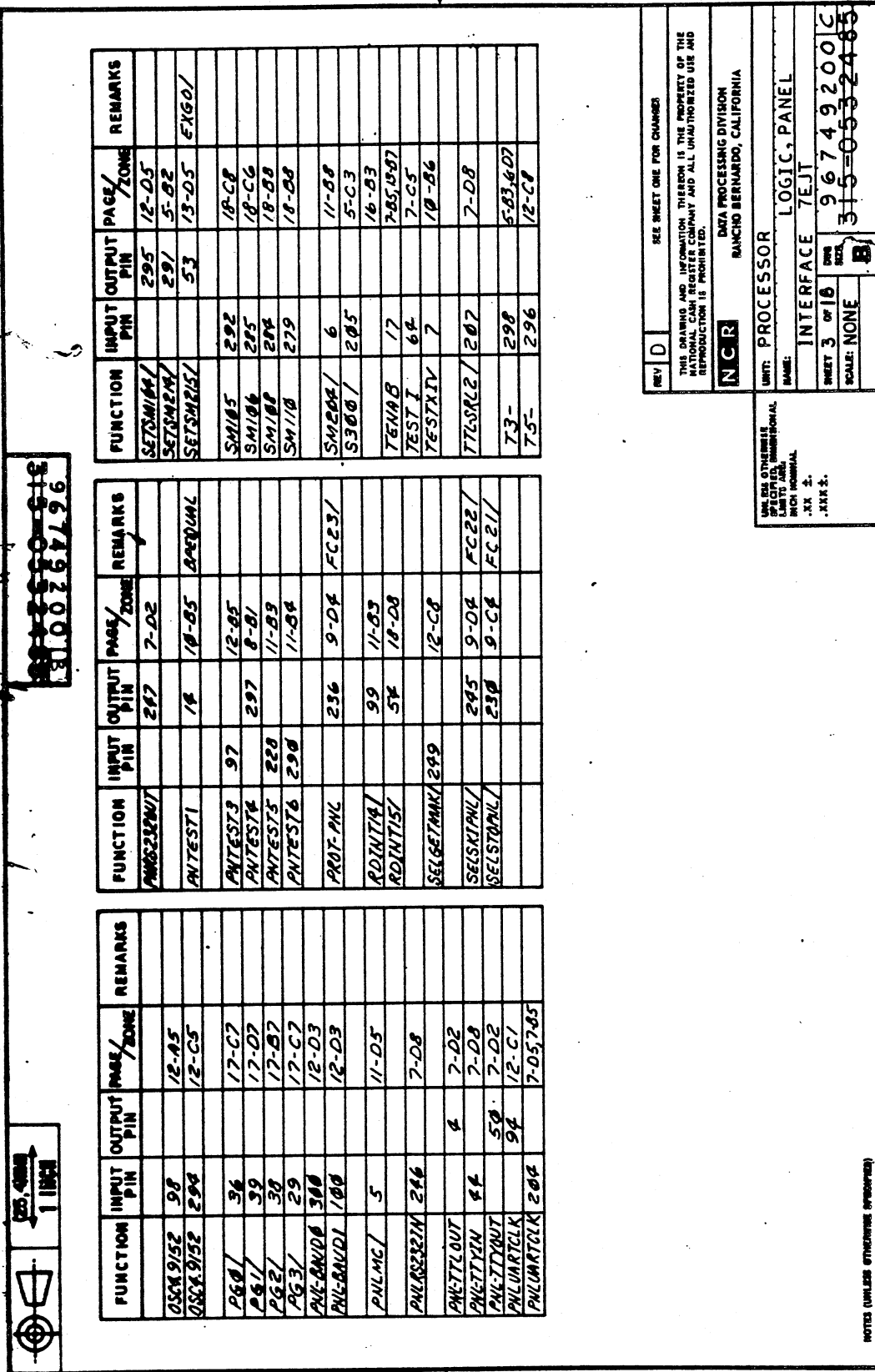


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 3 of 18)

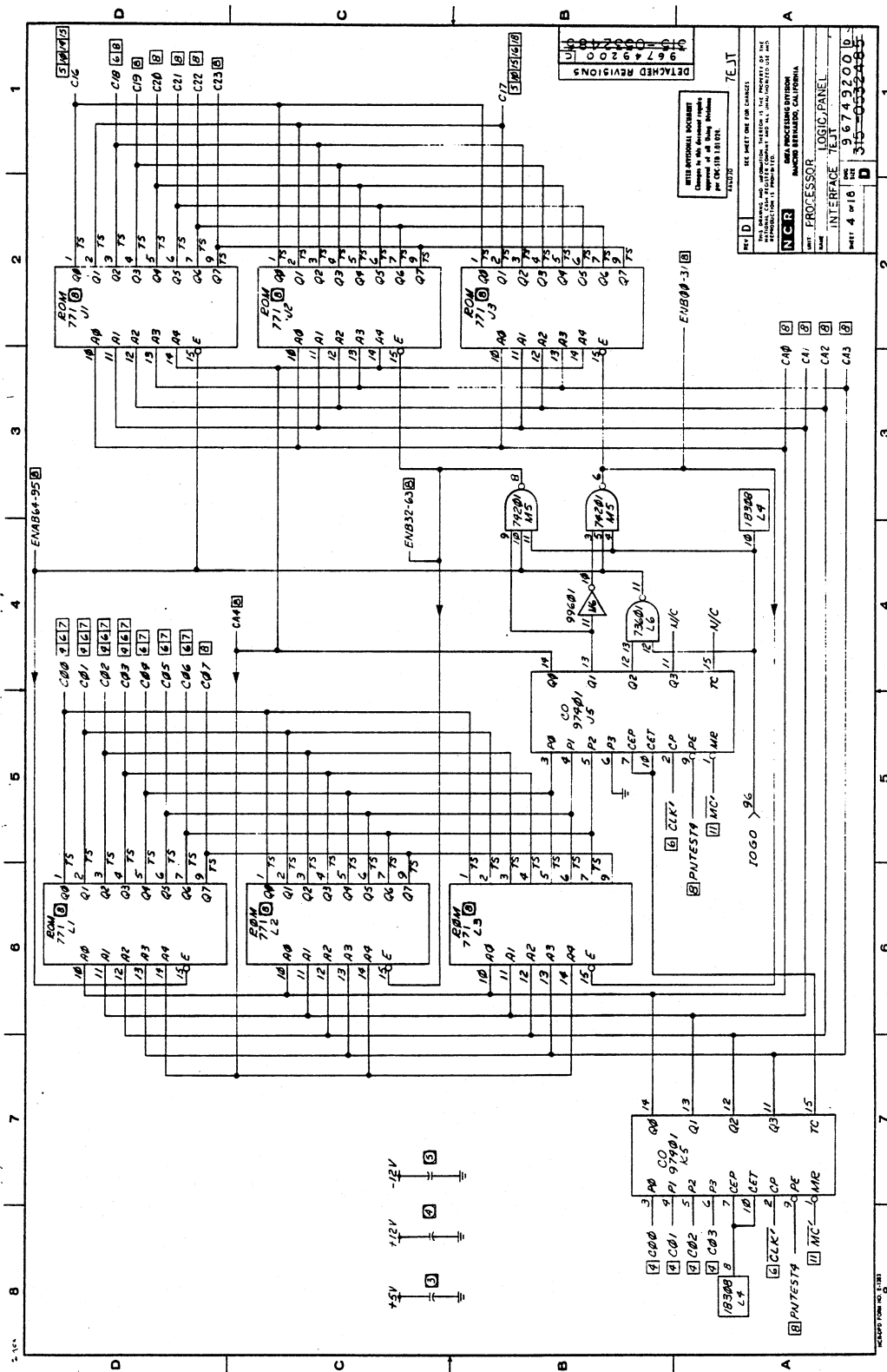


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 4 of 18)

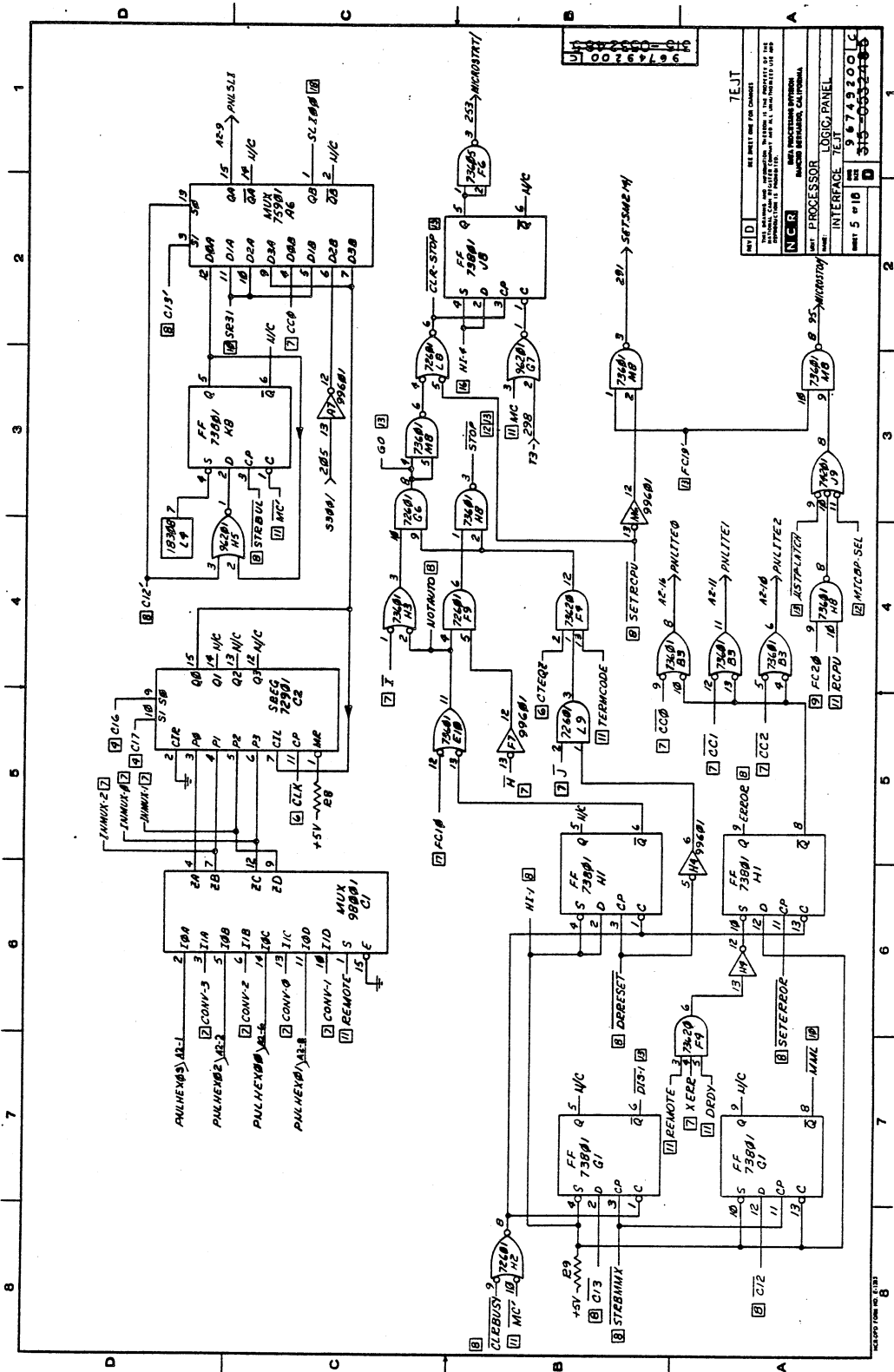


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 5 of 18)

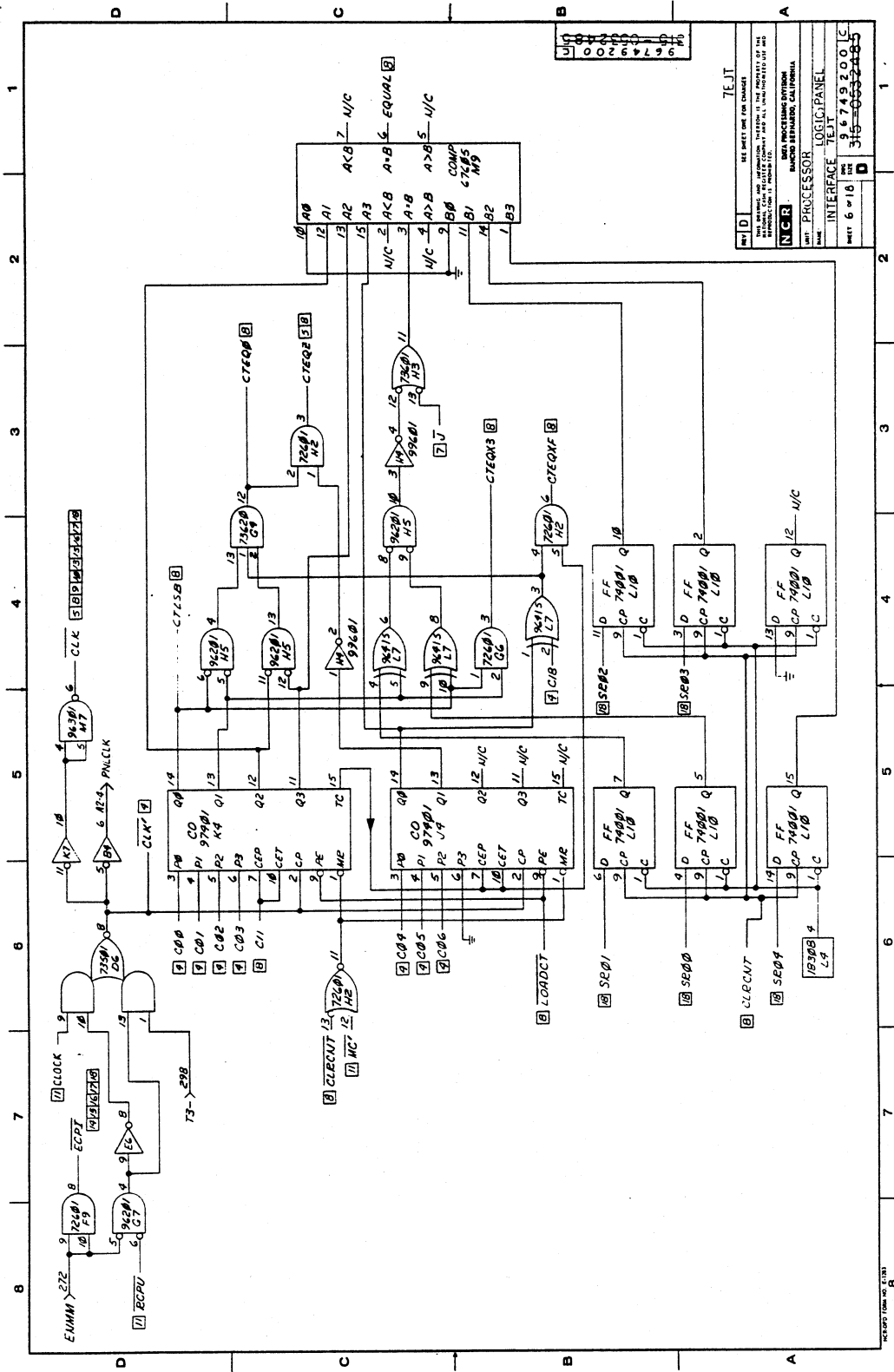


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 6 of 18)

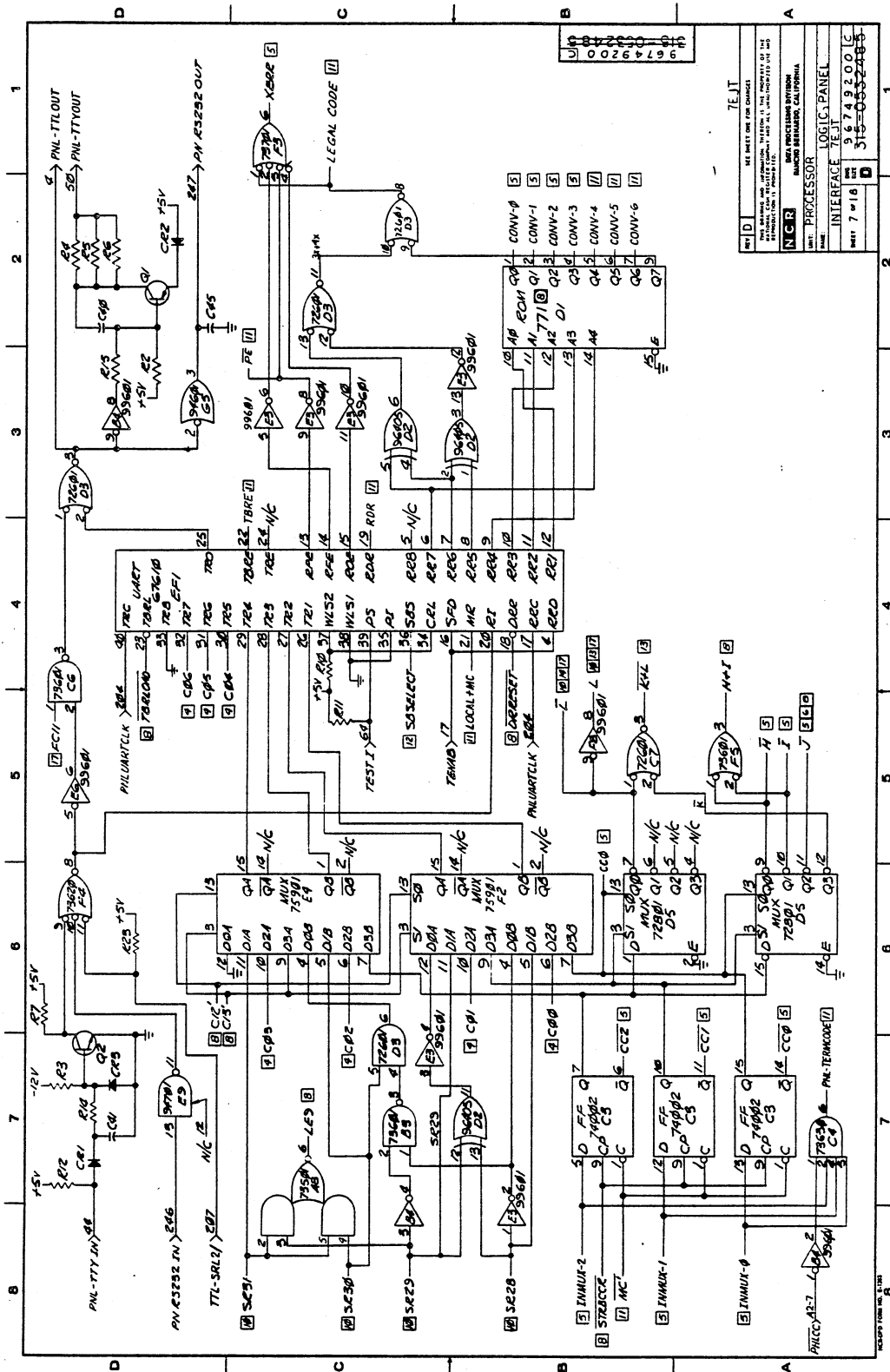


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 7 of 18)

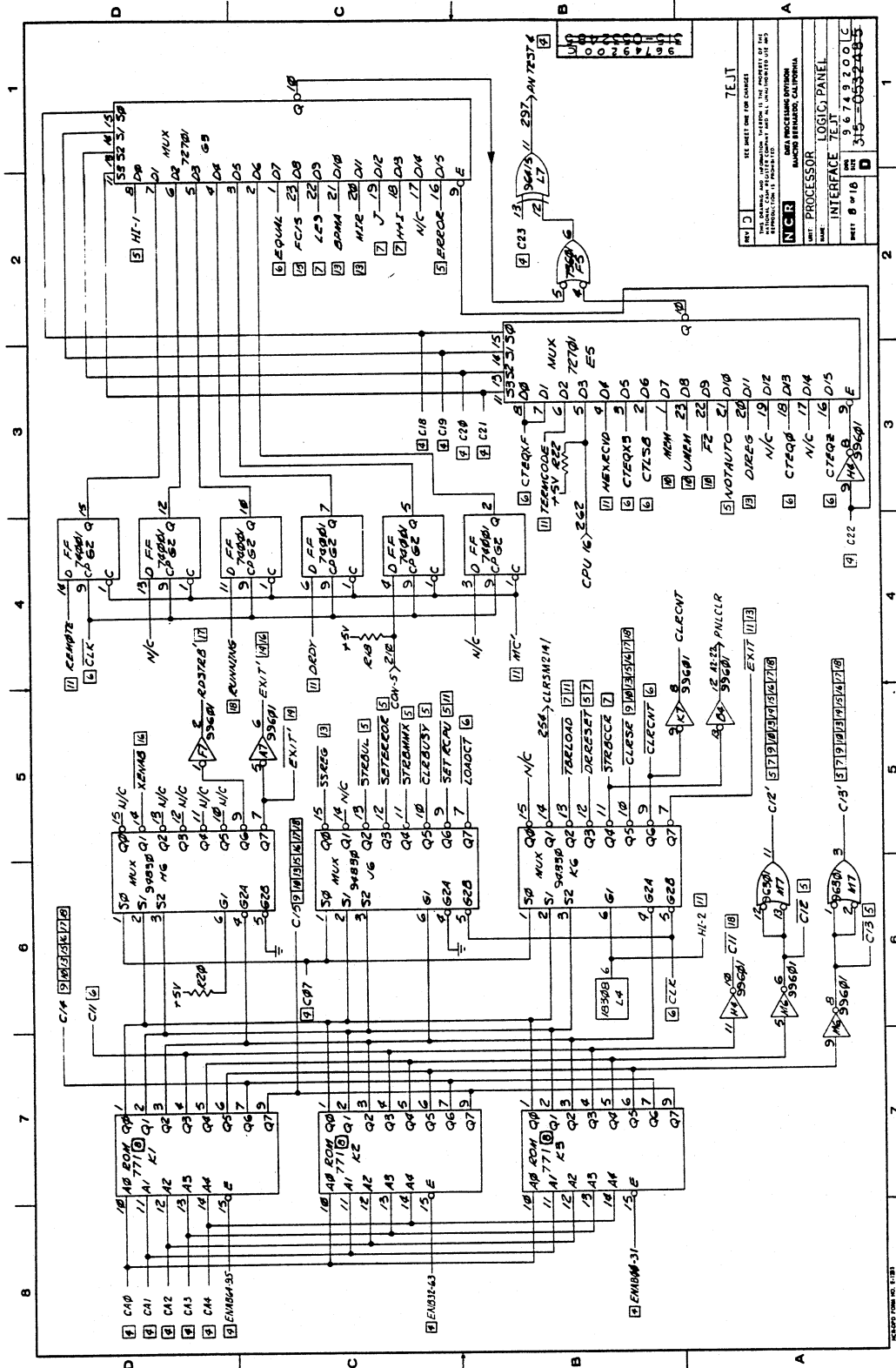
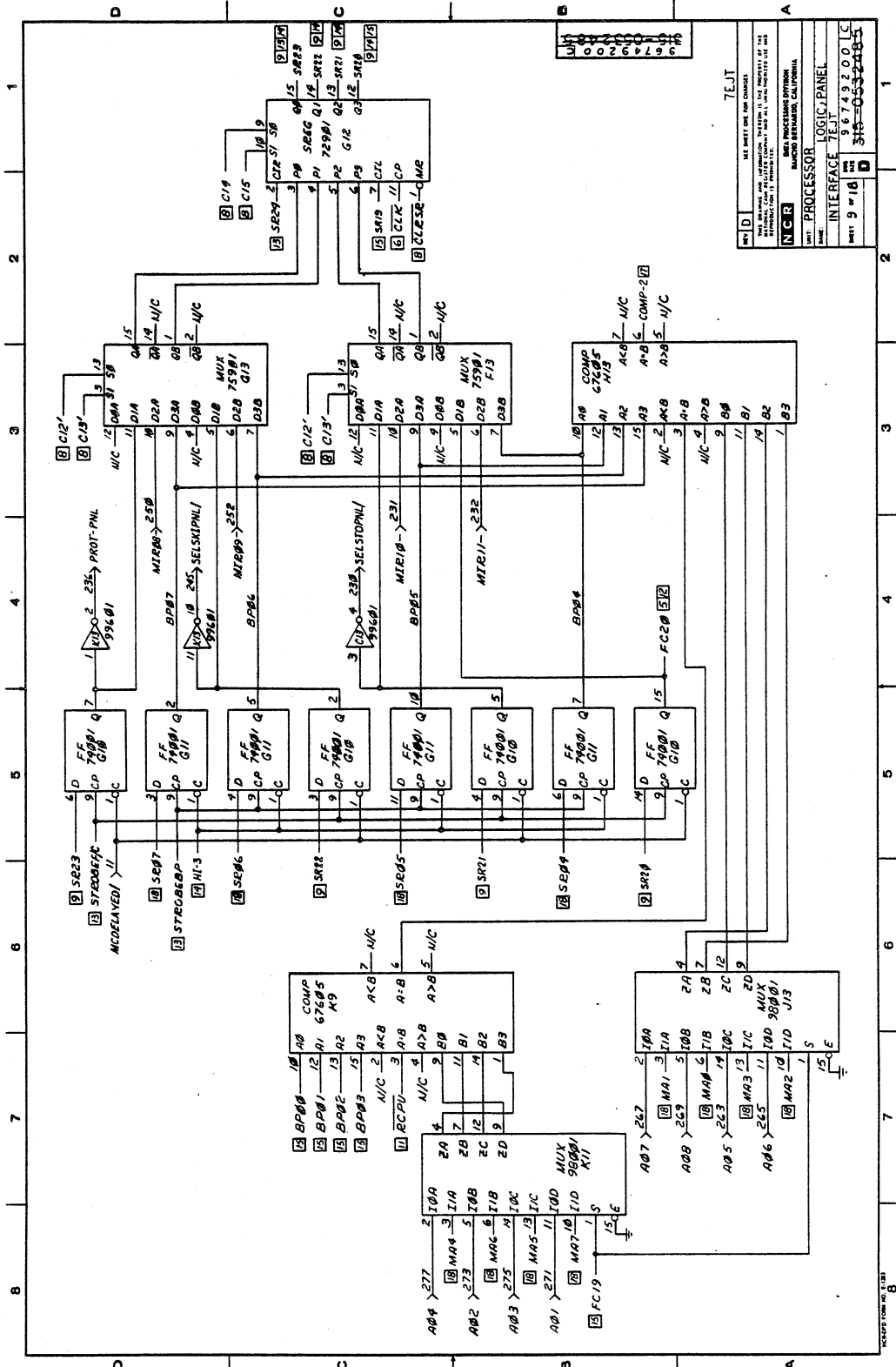


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 8 of 18)



REV	D	DATE	9 7 4 2 0 0
BY	9 18	DESIGNED BY	315-0532485
NCOR NATIONAL CORPORATION 1000 UNIVERSITY AVENUE RICHMOND, CALIFORNIA 94801			
NAME: PROCESSOR LOGIC PANEL INTERFACE 7EJT SHEET 9 OF 18			

Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 9 of 18)

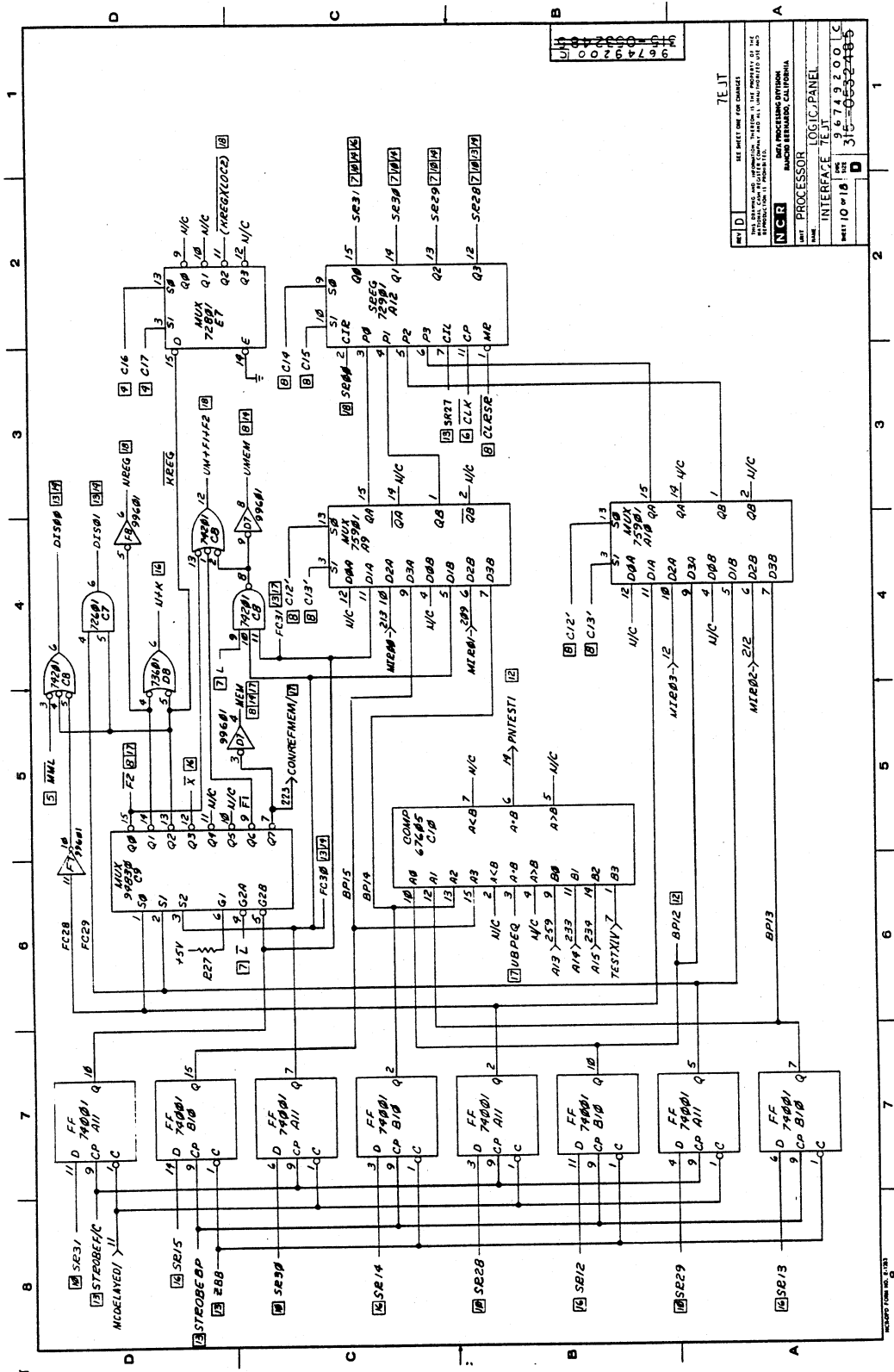


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 10 of 18)

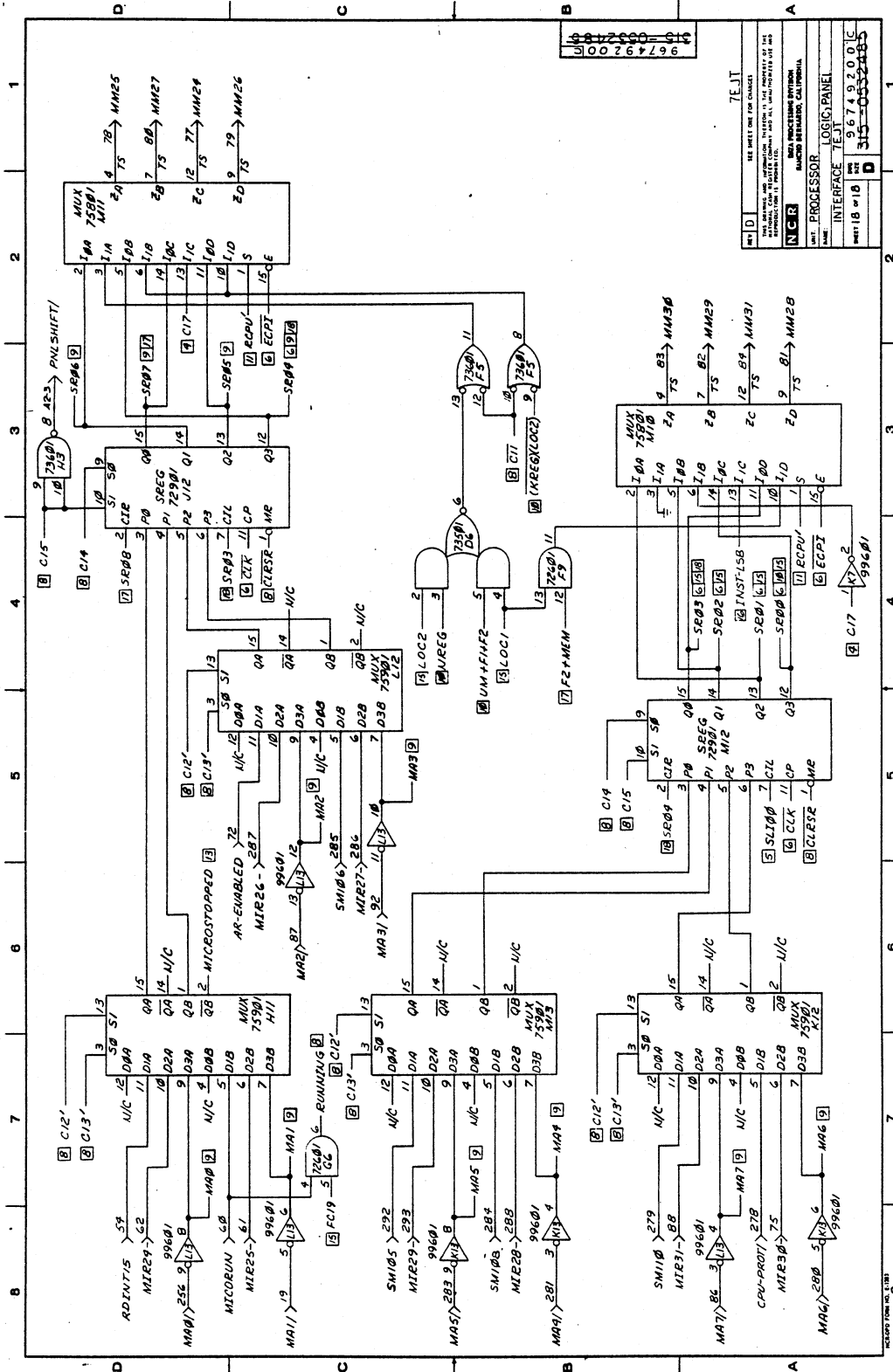


Figure 5-4. Breakpoint Controller (PWA 88909300) Logic Diagram (Sheet 18 of 18)

On-site maintenance (emergency or preventive maintenance) is limited to the diagnosis of a malfunction to the subassembly level. Repair is effected by replacing the faulty interface card or panel subassembly with a previously tested spare.

SPECIAL HANDLING

CAUTION

The breakpoint controller contains MOS-type IC devices that are subject to damage from electrostatic discharge. The controller has a red solder mask on the non-component side to identify it as being electrostatic sensitive. When installing or removing the controller, exercise extreme care in handling. Observe common practices such as touching a grounded

surface to discharge body potential before handling the assembly, turning off processor power before installing or removing the assembly, storing the assembly in the special antistatic bag when transporting, and repairing the assembly only at a properly equipped and grounded work station.

SPARES TESTING

All spare subassemblies must be tested upon receipt and retested annually.

PREVENTIVE MAINTENANCE

No preventive maintenance is required for this equipment.



PARTS DATA

7

This section contains the spare parts list and the assembly drawings applicable to the DT195, DT120, and FC402 breakpoint controller and breakpoint panel.



DWG CHKD ENG MFR APPR	1/10/78 G. J. [unclear] [unclear] [unclear]	8/11/78 [unclear]	TITLE SPL-PANEL I/F + PANEL	PREFIX SP	DOCUMENT NO. 88905000	REV. B		
FIRST USED ON			SHEET 1 OF 2					
SHEET REVISION STATUS			REVISION RECORD					
			REV	ECO	DESCRIPTION	DRPT	DATE	APP
			01		CL. B. PRE-RELEASED	LVP	8-28-74	[unclear]
			02	14058	SEE ECO	LVP	12-11-76	[unclear]
			03	14138	ADDED FIN 3	LVP	5-21-78	[unclear]
			A	14160	RELEASE CLASS A	LVP	6-30-78	[unclear]
			B	DS21059	FIN 2 WAS 88909300	DL	10-17-77	[unclear]
NOTES:								
			DETACHED LISTS					

AA8100 REV. 8/71

PRINTED IN U.S.A.



ASSEMBLY PARTS LIST

SPARE CODE
S = SPARE PARTS
N = NON SPARE PARTS

SH 2

MF

88905000		F	CLA	A	SPL-PANEL I/F + PANEL	US	MP17	08/20/74	10/11/78	1
ASSEMBLY NUMBER	REV	CLASS	BY	DT	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	PROCESSING DATE	PAGE NUMBER
FIND NUMBER	DW NUMBER	PART NUMBER	QUANTITY	UNITY	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MARK/OUT TYPE	S OR N
1 D	88878500		100		PC PWA-MAINTENANCE PANEL (TEST)	IN			AYM4	N
3 A	88879721		100		PC CABLE ASSY-IC, 2* COND, TWO CONN	IN	014138	04/10/75	PPP4	N
2 D	88909300		100		PC PWA-PANEL INTERFACE (TEST)	OUT	021059	11/23/78	AYM4	N
2 D	96752174		100		PC PWA-PANEL INTERFACE HGAT	IN	021059	11/23/78		N

NUMBER OF LINE ITEMS = 4
HIGHEST FIND NUMBER = 3

PROJECT ENGINEER

SC4D

AA 2708-1 REV 7-78

Figure 7-1. Breakpoint Controller and Panel Spare Parts List



20 (CAUM)

DESIGNER D. LINVILLE	DATE 1/21/76	CONTROL DATA	TITLE PWA- MAINTENANCE PANEL (7E8T)	PREFIX FL	DOCUMENT NO 88878500	REV N 1
CHKD W. M. GARDNER	1/21/76	SMALL COMPANY	FIRST USED ON MP 17	SHEET 1 OF 2		
ENG R.	1/21/76	DEVELOPMENT DIVISION	CODE IDENT 09132			
APPR R.	1/21/76					

SHEET REVISION STATUS		REVISION RECORD					
REV	DATE	DESCRIPTION	DRFT	DATE	APP		
H	14436	SEE ECO	JK	4-21-76	JK		
J	14733	SEE ECO	JK	1-17-77	JM		
K	DS21235	SEE ECO	ZL	7-11-77	JSH		
L	DS21358	SEE ECO	JSH	4-23-79	AW		
N	DS22040	SEE ECO	BR	3-11-81	AW		

NOTES:

INTER-DIVISIONAL DOCUMENT
Changes to this document require approval of all Using Divisions per CDC-STD 1.01.024.

DETACHED LISTS

AA6100 REV. 5-71 AA6100 PRINTED IN U.S.A.



ASSEMBLY PARTS LIST

SH 2.

88878500	N A D	PWA-MAINTENANCE PANEL (7E8T)	DSM	MP17	08/06/74	022040	02/06/81	1 / 1	MF
ASSEMBLY NUMBER	REV. CL.	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE OR. NUMBER	PROCESSING DATE	PAGE NUMBER	

LINE NUMBER	QUANTITY	PART NUMBER	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE OR. NUMBER	RATE EFFECTIVE	NAME/OUT PART TYPE	PH	PC
14	20.00	24500043		PC RES FXD .25W 150 OHMS	IN	014436	041376	PPPA	N	N
15	2.00	24500063		PC RES FXD .25W 1000 OHMS	IN	014436	041376	PPPA	N	N
2		39156102	REF	PC SPEC-MARKING/LETTERING, INK	IN	014733	041376	RFE4	N	N
13	2.00	75009901		PC RESISTOR MODULE 1K OHM 2 0/0	IN	014436	041376	PPPA	N	N
3		88878600	REF	PC LOGIC DIAG-MAINT PANEL (7E8T)	IN	014137	040975	RFE4	N	N
1	1.00	88878700		PC PWB-MAINTENANCE PANEL	IN	014137	040975	PPPA	N	N
9	4.00	88881600		PC IC 74194 TTL 4-BIT BIDIR S REG	IN	014137	040975	PPPA	N	N
11	1.00	88882100		PC IC 74H00 TTL QUAD 2-I POS NAND	IN	014137	040975	PPPA	N	N
8	1.00	88883100		PC IC 74H10 TTL TPL 3-I POS NAND	IN	014137	040975	PPPA	N	N
10	3.00	88886000		PC IC 9318 TTL 8-IN PRIOR ENCODER	IN	014137	040975	PPPA	N	N
7	4.00	88886600		PC IC 74H04 TTL HEX INVERTER	IN	014137	040975	PPPA	N	N
12	20.00	88887100		PC DIODE-LIGHT EMIT CLR NON-DIFF.	IN	014137	040975	PPPA	N	N
17	1.00	88887200		PC KEYBOARD ASSEMBLY KIT - 16 STA	IN	014137	040975	PPPA	N	N
15	1.00	88887200		PC KEYBOARD ASSEMBLY KIT - 16 STA	OUT	022040	052201	PPPA	N	N
16	2.00	88887201		PC KEYBOARD ASSEMBLY KIT - 4 STA	IN	014733	041376	PPPA	N	N
14	3.00	88887201		PC KEYBOARD ASSEMBLY KIT - 4 STA	OUT	022040	052201	PPPA	N	N
17	1.00	88887205		PC KEYBD ASSY KIT-4X4 NON LIGHTED	IN	022040	052201	PPPA	N	N
16	2.00	88887206		PC KEYBD ASSY KIT-1X4 NON LIGHTED	IN	022040	052201	PPPA	N	N
18	1.00	88889100		PC SOCKET-IC 24 PIN, REG PROFILE	IN	014137	040975	PPPA	N	N
19	1.00	88889300		PC SWITCH-SUB MINI, SPDT, ROCKER	IN	014137	040975	PPPA	N	N
20	1.00	88889301		PC SWITCH-SUB MINI, SPDT, ROCKER (M)	IN	014733	041376	RFE4	N	N
5		88889500	REF	PC SPEC-PC BOARD ASSY	IN	014137	040975	RFE4	N	N
6	4.00	88897800		PC CAP-FIXED SOLID TANT, 6VDC, 12UF	IN	014137	040975	PPPA	N	N
21	4.00	96744315		IN GASKET-STRIP ADHESIVE	IN	021358	071379	PPP3	N	N

NUMBER OF LINE ITEMS = 24
HIGHEST FIND NUMBER = 21

AA6100 REV. 5-71 SCMD MFG INFO PRINTED IN U.S.A. INTERDIVISIONAL DOCUMENT

Figure 7-2. Breakpoint Panel Printed Wiring Assembly (Sheet 2 of 2)

(TD-PB-STA-220)



ASSEMBLY PARTS LIST

96752176	E	A	D	PWA-PANEL INTERFACE RGXT	DSM	CRR 18	10/09/78	022176	02/16/82	1 / 3	MF
ASSEMBLY NUMBER	REV	CL	BY	DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORG. NUMBER	PROCESSING DATE	PAGE NUMBER	

PART NUMBER	QTY	PART NUMBER	QUANTITY	UNIT	PART DESCRIPTION	IN/OUT	CHANGE ORG. NUMBER	DATE EFFECTIVE	NAME/BUY	PC	AS
22	C	15163326	1.00	PC	IC 1489A RECEIVER RS232C DTL	IN	042540		PPP4	N	
55	C	17705905	2.00	PC	RES FXD .25W 51000 OHMS	IN			PPP4	N	
13	C	17706766	2.00	PC	CAP FIXED SOLID TANTALUM I10	IN			PPP4	N	
56	C	24500043	1.00	PC	RES FXD .25W 150 OHMS	IN			PPP4	N	
57	C	24500056	1.00	PC	RES FXD .25W 510 OHMS	IN			PPP4	N	
58	C	24500061	1.00	PC	RES FXD .25W 820 OHMS	IN			PPP4	N	
50	C	24500063	17.00	PC	RES FXD .25W 1000 OHMS	IN			PPP4	N	
51	C	24500072	3.00	PC	RES FXD .25W 2400 OHMS	IN			PPP4	N	
53	C	24500080	1.00	PC	RES FXD .25W 5100 OHMS	IN			PPP4	N	
54	C	24500087	2.00	PC	RES FXD .25W 10000 OHMS	IN			PPP4	N	
65	C	24500094	1.00	PC	RES FXD .25W 20000 OHMS	IN			PPP4	N	
52	C	24500163	1.00	PC	RES FXD COMP .50W 1K OHMS 5PCT	IN			PPP4	N	
43	C	24528618	4.00	IN	WIRE ELEC 38 GA WHITE	OUT	022129	120181	PPP1	N	
21	A	36186400	1.00	PC	IC CHIP, TYPE 1488	IN			PPP4	N	
61	A	39017400	1.00	PC	TRANSISTOR 2N219-A	IN			PPP4	N	
7	A	39186182	REF	PC	SPEC MARKING/LETTERING, INK	OUT	022129	120181	RFE4	N	
60	A	39303300	1.00	PC	TRANSISTOR-SILICON PNP 2N2905	IN			PPP4	N	
66	C	52629949	60.00	IN	WIRE ELEC 38 GA WHITE	OUT	022129	120181	PPP1	N	
49	A	75009901	1.00	PC	RESISTOR MODULE 1K OHM 2 0/0	IN			PPP4	N	
62	A	84905500	2.00	PC	MTG PAD, TRANSISTOR	IN			PPP4	N	
17	A	84996703	2.00	PC	CAP, CER 100V 22 PF	IN			PPP4	N	
15	A	84996719	2.00	PC	CAP, CER 100V 330 PF	IN			PPP4	N	
6	A	88812400	9.00	PC	RIVET SEMI-TUBULAR, BRN, 312 LG	OUT	022129	120181	PPP4	N	
14	A	88880400	2.00	PC	CAP-CERAMIC 50VDC .1UF 20PCT	IN			PPP4	N	
12	A	88880500	2.00	PC	CAP ELECT-ALUM 16VDC 100UF	IN			PPP4	N	
28	A	88880900	5.00	PC	IC 7485 TTL 4-BIT MAGN COMPAR	IN			PPP4	N	
18	A	88881000	1.00	PC	IC-TR1863A/B, UART, MOS/LSI, 40PN	IN			PPP4	N	
33	A	88881200	8.00	PC	IC 74H08 TTL QUAD 2-INPUT AND	IN			PPP4	N	
40	A	88881300	2.00	PC	IC 74150 TTL 16-BIT SEL/MUX	IN			PPP4	N	
41	A	88881400	3.00	PC	IC 74155 TTL 2/4 DECODER/MUX	IN			PPP4	N	
46	A	88881600	9.00	PC	IC 74194 TTL 4-BIT BIDIR S REG	IN			PPP4	N	
25	A	88881700	4.00	PC	IC 8214 TTL DUAL 4/1 MUX	IN			PPP4	N	
38	A	88882000	2.00	PC	IC 74H51 TTL DUAL 2-WIDE 2-INP	IN			PPP4	N	
30	A	88882100	11.00	PC	IC 74H00 TTL QUAD 2-I POS NAND	IN			PPP4	N	
31	A	88882200	1.00	PC	IC 74H01 TTL QUAD 2-I POS NAND	IN			PPP4	N	
35	A	88882300	2.00	PC	IC 74H11 TTL TRIPLE 3-INP AND	IN			PPP4	N	

SCND MFG INFO

INTERDIVISIONAL DOCUMENT



ASSEMBLY PARTS LIST

96752176	E	A	D	PWA-PANEL INTERFACE 8GXT	DSM	CBR 18	10/09/78	022176	02/16/82	2 / 3	MF
ASSEMBLY NUMBER	REV	CL	BY	DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORG. NUMBER	PROCESSING DATE	PAGE NUMBER	

PART NUMBER	QTY	PART NUMBER	QUANTITY	UNIT	PART DESCRIPTION	IN/OUT	CHANGE ORG. NUMBER	DATE EFFECTIVE	NAME/BUY	PC	AS
37	A	88882400	1.00	PC	IC 74H21 TTL DUAL 4-INPUT AND	IN			PPP4	N	
36	A	88882600	2.00	PC	IC 74H20 TTL DUAL 4-I POS NAND	IN			PPP4	N	
39	A	88882700	5.00	PC	IC 74H74 TTL DUAL D EDGE F/F	IN			PPP4	N	
44	A	88882800	9.00	PC	IC 74174 TTL HEX D F/F W/CLEAR	IN			PPP4	N	
45	A	88882900	1.00	PC	IC 74175 TTL QUAD D F/F W/CLR	IN			PPP4	N	
34	A	88883100	3.00	PC	IC 74H10 TTL TPL 3-I POS NAND	IN			PPP4	N	
20	A	88883400	22.00	PC	IC 9309 TTL DUAL 4-INPUT MUX	IN			PPP4	N	
47	A	88885100	4.00	PC	IC 74S138 TTL DECODER/MUX	IN			PPP4	N	
26	A	88885500	4.00	PC	IC 7402 TTL QUAD NOR GATE	IN			PPP4	N	
27	A	88885600	1.00	PC	IC 7437 TTL QUAD 2-INP NAND RFR	IN			PPP4	N	
29	A	88885700	1.00	PC	IC 7486 TTL QUAD 2-IN EXCL OR	IN			PPP4	N	
43	A	88885800	8.00	PC	IC 74161 TTL 4-BIT BINARY CNTR	IN			PPP4	N	
42	A	88886400	6.00	PC	IC 74157 TTL QUAD 2-INPUT MUX	IN			PPP4	N	
19	A	88886500	2.00	PC	IC 9602 TTL DUAL MONOSTAB MVB	IN			PPP4	N	
32	A	88886600	13.00	PC	IC 74H04 TTL HEX INVERTER	IN			PPP4	N	
59	A	88889100	1.00	PC	SOCKET-IC 24 PIN, REG PROFILE	IN			PPP4	N	
10	A	88889500	REF	PC	SPEC-PC BOARD ASSY	OUT	022129	120181	RFE4	N	
24	A	88896000	6.00	PC	IC 8123 TTL TRI-STATE 2-IN MUX	IN			PPP4	N	
3	C	88896600	1.00	PC	INSULATOR-CARD FRAME, UPPER	IN			PPP4	N	
4	C	88896700	1.00	PC	INSULATOR-CARD FRAME, LOWER	IN			PPP4	N	
2	C	88896800	1.00	PC	FRAME-CARD (CASTING)	IN			PPP4	N	
48	A	88897100	3.00	PC	DIODE-SILICON SWITCHING	IN			PPP4	N	
11	A	88897800	22.00	PC	CAP-FIXED SOLID TANT, 6VDC, 12UF	IN			PPP4	N	
23	A	88919501	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
67	A	88919502	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
68	A	88919503	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
69	A	88919504	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
70	A	88919505	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
71	A	88919506	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
72	A	88919507	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
73	A	88919508	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
74	A	88919509	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
75	A	88919510	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
76	A	88919511	1.00	PC	IC-TTL, ROM 256BIT PROGRAMMABLE	IN			AYN4	N	
64	A	88922900	1.00	PC	IC 74S86 QUAD 2IN EXCLUSIVE OR	IN			PPP4	N	
6	A	92030018	9.00	PC	EYELET	IN	022129	120181	PPP4	N	

SCND MFG INFO

INTERDIVISIONAL DOCUMENT

Figure 7-3. Breakpoint Controller Printed Wiring Assembly (96752176) (Sheet 2 of 3)

ASSEMBLY PARTS LIST

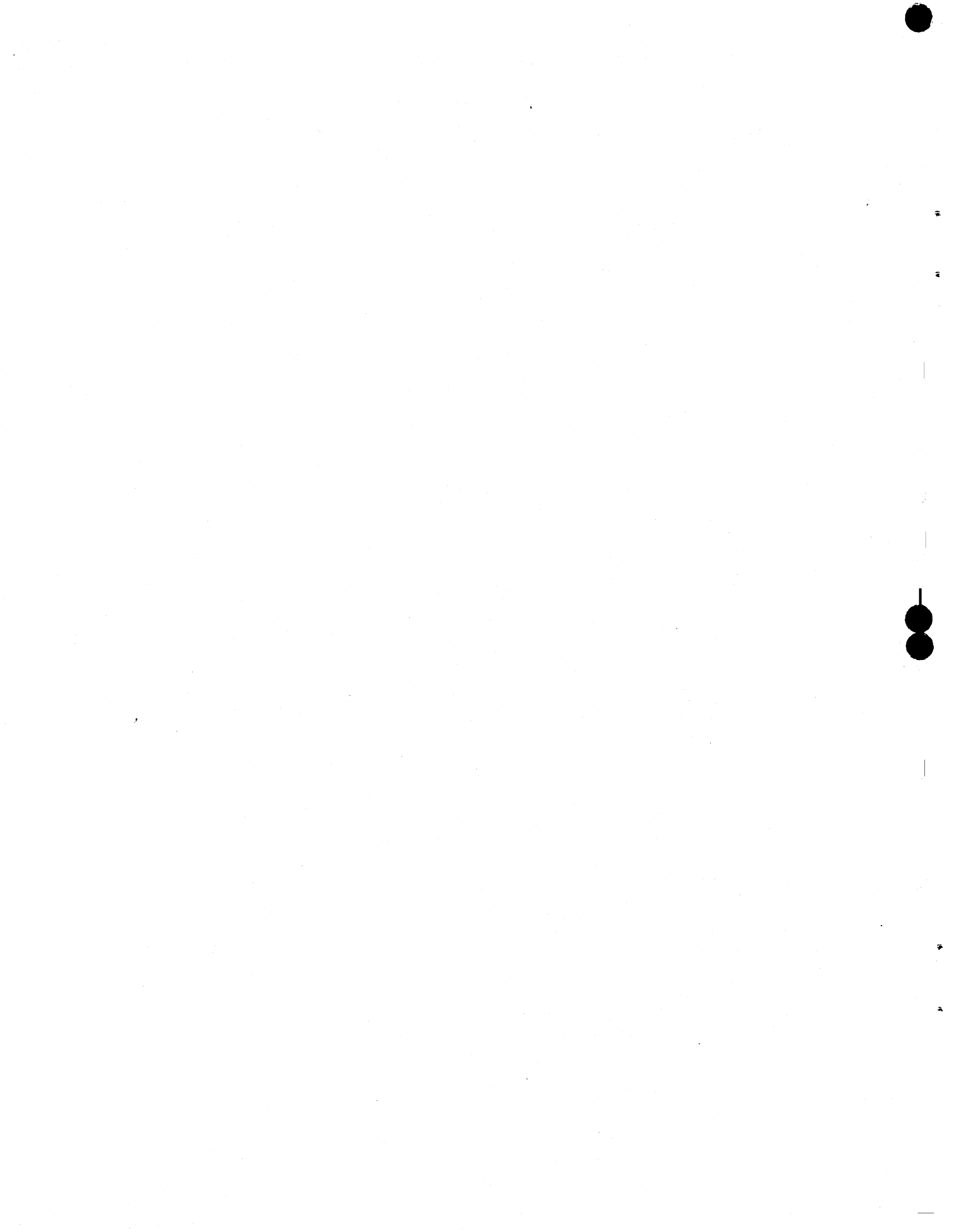
96752176	E	A	D	PWA-PANEL INTERFACE 86XT	DSM	CBR 18	10/09/78	022176	02/16/82	3/ 3	MF
ASSEMBLY NUMBER	REV	CL	REV	ASSEMBLY DESCRIPTION	DESIGN SOURCE	POST WAGE	RELEASE DATE	CHANGE ORG. NUMBER	PROCESSING DATE	PAGE NUMBER	

FIND NUMBER	REV	PART NUMBER	QTY	UNIT	PART DESCRIPTION	STATUS	CHANGE ORG. NUMBER	DATE APPROVED	NAME/ID	PC	OR
16	C	94354840	1.00	PC	CAP-CERAMIC,HIGH-K 50V 1.0UF	IN			PPP4	N	
1	A	96744100	1.00	PC	PWB-PANEL INTERFACE	IN			PPP4	N	
8	D	96752177	REF	PC	LOGIC-PANEL INTERFACE 86XT	IN			RFE4	N	
					NUMBER OF LINE ITEMS = 75						
					HIGHEST FIND NUMBER = 76						

SCHEMATIC REV 800 WFO INFO

INTERDIVISIONAL DOCUMENT

Figure 7-3. Breakpoint Controller Printed Wiring Assembly (96752176) (Sheet 3 of 3)



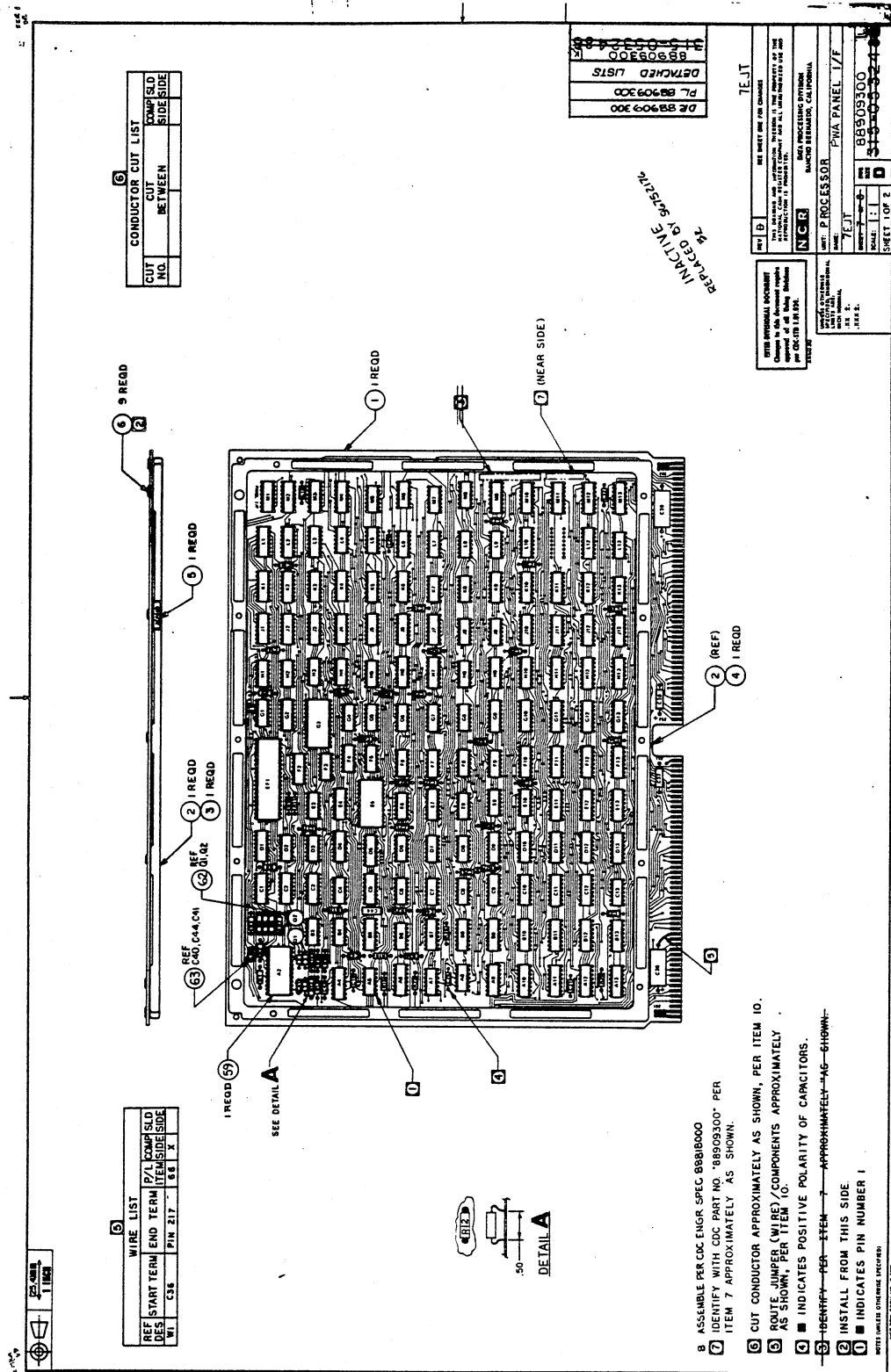


Figure 7-4. Breakpoint Controller Printed Wiring Assembly (88909300) (Sheet 1 of 4)

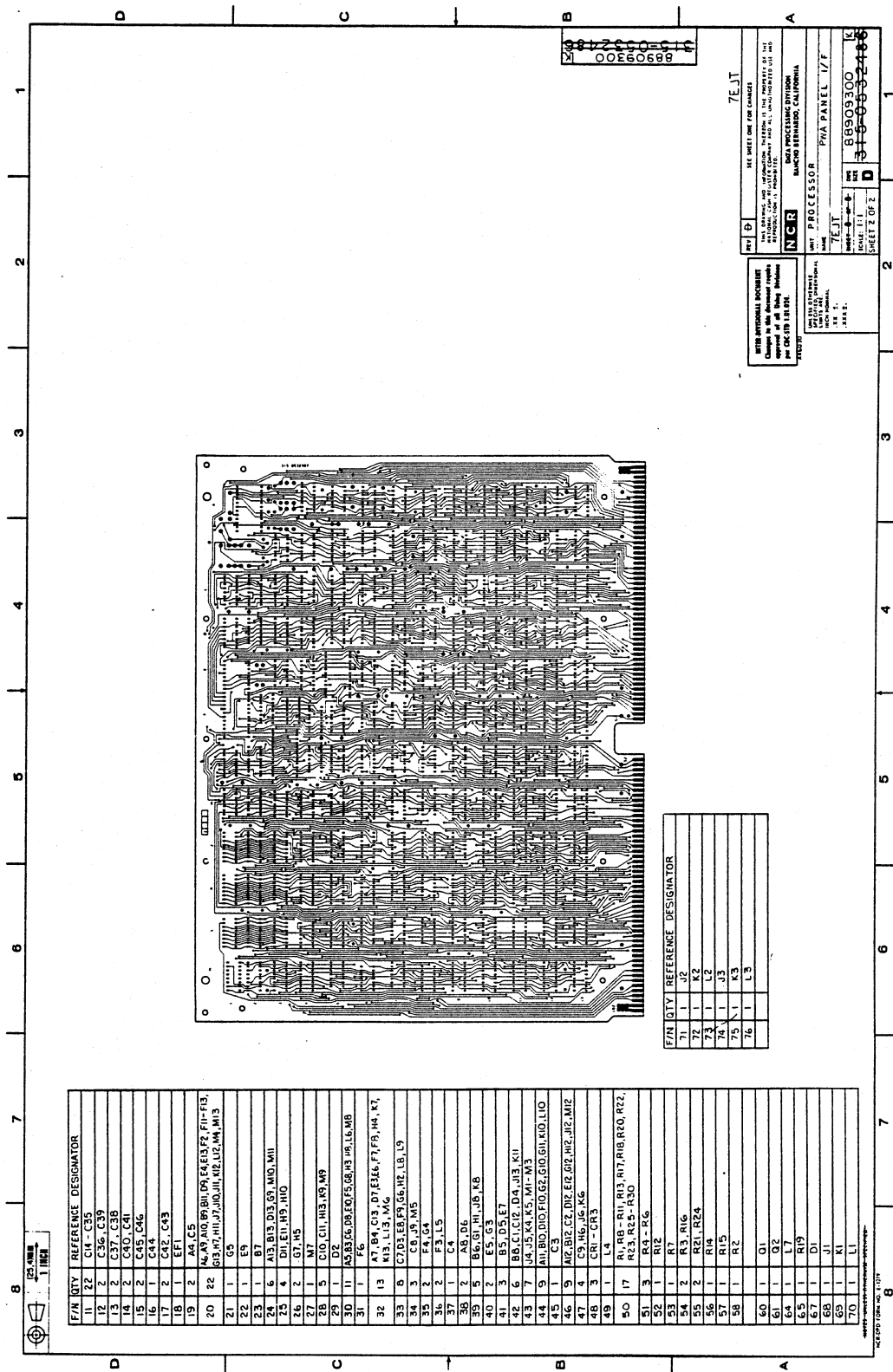


Figure 7-4. Breakpoint Controller Printed Wiring Assembly (88909300) (Sheet 2 of 4)

DRW	D. LINVILLE	DATE	4-19-74	TITLE	PWA-PANEL INTERFACE (TEJT)	PREFIX	PL	DOCUMENT NO.	88909300	REV.	L
CHKD	T. PEKKA	DATE	4-19-74	FIRST USED ON	MP 17						
ENG	W. HERBST	DATE	4-20-74	CODE IDENT	09132					SHEET	1 OF 4
MFG											
APPR	E. MUNEVAR	DATE	4-21-74								

SHEET REVISION STATUS				REVISION RECORD			
REV	ECO	DESCRIPTION	DRFT	DATE	APP		
G	14436	SEE ECO	TP	DL	4-21-74	EM	
H	14733	SEE ECO	GL	JM	1-12-77	AW	
J	DS14862	FN 18 WAS TR1602B BT		DSS	7-6-77	AWJ	
L	DS21059	INACTIVE	BR				

- INACTIVE -
REPLACED BY 96752176
BR

INTER-DIVISIONAL DOCUMENT
Changes to this document require
approval of all Sales Divisions
per CXC-STD 01.034

NOTES: AA6030

DETACHED LISTS

AS2100 REV. 8/71

PRINTED IN U.S.A.



ASSEMBLY PARTS LIST

SPARE CODE
S = SPARE PARTS
N = NON SPARE PARTS

SH 2

88909300	L	INA	D	PWA-PANEL INTERFACE (TEJT)	DSM	MP17	09/20/74	10/09/78	1 / 3
ASSEMBLY NUMBER	REV	CLASS	HW	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	PROCESSING DATE	PAGE NUMBER

MF

P/N	BY	PART	QUANTITY	UNIT	PART DESCRIPTION	IN/OUT	CHANGE ORD.	DATE	MAKE/BUY	NO	NC
NUMBER	DES	NUMBER		MEAS		STATUS	NUMBER	EFFECTIVE	TYPE	IN	NO
55	C	17705905	200	PC	RES FXD .25W 51000 OHMS	IN	014369	020676	PPP4	N	N
13	C	17706766	200	PC	CAP FIXED SOLID TANTALUM 110	IN	014436	041376	PPP4	N	N
56	C	24500043	100	PC	RES FXD .25W 150 OHMS	IN	014369	020676	PPP4	N	N
57	C	24500056	100	PC	RES FXD .25W 510 OHMS	IN	014369	020676	PPP4	N	N
58	C	24500061	100	PC	RES FXD .25W 920 OHMS	IN	014369	020676	PPP4	N	N
50	C	24500063	1700	PC	RES FXD .25W 1000 OHMS	IN	014369	020676	PPP4	N	N
51	C	24500072	300	PC	RES FXD .25W 2400 OHMS	IN	014369	020676	PPP4	N	N
53	C	24500080	100	PC	RES FXD .25W 5100 OHMS	IN	014369	020676	PPP4	N	N
54	C	24500087	200	PC	RES FXD .25W 10000 OHMS	IN	014369	020676	PPP4	N	N
65	C	24500094	100	PC	RES FXD .25W 20000 OHMS	IN	014369	020676	PPP4	N	N
52	C	24500163	100	PC	RES FXD COMP .50W 1K OHMS 5PCT	IN	014369	020676	PPP4	N	N
63	C	24528618	600	IN	INS. SLV. ELEC-RULK	IN	014733	041376	PPP1	N	N
21	A	36186400	100	PC	IC CHIP, TYPE 148A	IN	014733	041376	PPP4	N	N
22	A	36186501	100	PC	IC CHIP TYPE 1489	IN	014436	041376	PPP4	N	N
61	A	39017400	100	PC	TRANSISTOR 2N2219-A	IN	014436	041376	PPP4	N	N
7	A	39156102	REF	PC	SPEC-MARKING/LETTERING, INK	IN	014733	041376	RFE4	N	N
60	A	39380300	100	PC	TRANSISTOR-SILICON PNP 2N2905	IN	014733	041376	PPP4	N	N
66	C	52629909	200	IN	WIRE FLECT 30 GA SOLID WHITE	IN	014733	041376	PPP1	N	N
49	A	75009901	100	PC	RESISTOR MODULE 1K OHM 2.0/0	IN	014436	041376	PPP4	N	N
62	A	84905500	200	PC	MTG PAD, TRANSISTOR	IN	014733	041376	PPP4	N	N
17	A	84996703	200	PC	CAP, CER 100V 22 PF	IN	014733	041376	PPP4	N	N
15	A	84996719	200	PC	CAP, CER 100V 330 PF	IN	014436	041376	PPP4	N	N
6	A	88812400	900	PC	RIVET-SEMI-TUBULAR, RRS .312 LG	IN	014369	020676	PPP4	N	N
14	A	88880400	200	PC	CAP-FIXED CER, 25VDC, .1UF, 20PCT	IN	014369	020676	PPP4	N	N
12	A	88880500	200	PC	CAP-FIXED AL, 16VDC, 100UF	IN	014369	020676	PPP4	N	N
28	A	88880900	500	PC	IC 7485 TTL 4-BIT MAGN COMPAR	IN	014369	020676	PPP4	N	N
18	A	88881000	100	PC	IC-TR1602A (UART) MOS DATA I/F	IN	014369	020676	PPP4	N	N
33	A	88881200	400	PC	IC 74H08 TTL QUAD 2-INPUT AND	IN	014369	020676	PPP4	N	N
40	A	88881300	200	PC	IC 74150 TTL 16-BIT SEL/MUX	IN	014369	020676	PPP4	N	N
41	A	88881400	300	PC	IC 74155 TTL 2/4 DECODER/MUX	IN	014369	020676	PPP4	N	N
46	A	88881600	900	PC	IC 74194 TTL 4-BIT BIDIR S REG	IN	014369	020676	PPP4	N	N
25	A	88881700	400	PC	IC 8214 TTL DUAL 4/1 MUX	IN	014369	020676	PPP4	N	N
38	A	88882000	200	PC	IC 74H51 TTL DUAL 2-WIDE 2-INP	IN	014369	020676	PPP4	N	N
30	A	88882100	1100	PC	IC 74H00 TTL QUAD 2-I POS NAND	IN	014369	020676	PPP4	N	N
31	A	88882200	100	PC	IC 74H01 TTL QUAD 2-I POS NAND	IN	014369	020676	PPP4	N	N
35	A	88882300	200	PC	IC 74H11 TTL TRIPLE 3-INP AND	IN	014369	020676	PPP4	N	N

INACTIVE

PROJECT ENGINEER: SCHM ID: INTDIVISIONAL DOCUMENT

Figure 7-4. Breakpoint Controller Printed Wiring Assembly (88909300) (Sheet 3 of 4)

ASSEMBLY PARTS LIST

SPARE CODE
S - SPARE PARTS
N - NON SPARE PARTS

SH 3

88909300	L	INA	D	PWA-PANEL INTERFACE (7EJT)	DSM	HP17	09/20/74	10/09/78	2 / 3
ASSEMBLY NUMBER	REV	CLASS	DW SZ	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	PROCESSING DATE	PAGE NUMBER

MF

FIND NUMBER	DW SZ	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MAKE/BUY PART TYPE	PH	NC	QTY
37	A	8882400	100	PC	IC 7421 TTL DUAL 4-INPUT AND	IN	014369	020676	PPP4	N	N	
36	A	8882600	200	PC	IC 7420 TTL DUAL 4-1 POS NAND	IN	014364	020676	PPP4	N	N	
39	A	8882700	500	PC	IC 7427 TTL DUAL D EDGE F/F	IN	014364	020676	PPP4	N	N	
44	A	8882800	900	PC	IC 74174 TTL HEX D F/F W/CLEAR	IN	014369	020676	PPP4	N	N	
45	A	8882900	100	PC	IC 74175 TTL QUAD D F/F W/CLR	IN	014369	020676	PPP4	N	N	
34	A	8883100	300	PC	IC 74110 TTL TPL 3-1 POS NAND	IN	014369	020676	PPP4	N	N	
20	A	8883300	2200	PC	IC 9309 TTL DUAL 4-INPUT MUX	IN	014369	020676	PPP4	N	N	
47	A	8883500	400	PC	IC 74513R TTL DECODER/MUX	IN	014369	020676	PPP4	N	N	
26	A	8885500	200	PC	IC 7402 TTL QUAD NOR GATE	IN	014369	020676	PPP4	N	N	
27	A	8885600	100	PC	IC 7437 TTL QUAD 2INP NAND RFR	IN	014369	020676	PPP4	N	N	
29	A	8885700	100	PC	IC 7486 TTL QUAD 2-IN EXCL OR	IN	014369	020676	PPP4	N	N	
43	A	8885800	700	PC	IC 74161 TTL 4-HIT BINARY CNTK	IN	014369	020676	PPP4	N	N	
42	A	8886400	600	PC	IC 74157 TTL QUAD 2-INPUT MUX	IN	014369	020676	PPP4	N	N	
19	A	8886500	200	PC	IC 9602 TTL DUAL MONOSTAB MVH	IN	014369	020676	PPP4	N	N	
32	A	8886600	1300	PC	IC 74004 TTL HEX INVERTER	IN	014369	020676	PPP4	N	N	
5	A	8887600	100	PC	LABEL-IDENTIFICATION,K25	IN	014369	020676	PPP4	N	N	
59	A	88889100	100	PC	SOCKET-IC 24 PIN, RFG PROFILE	IN	014369	020676	PPP4	N	N	
10	A	88889500	REF	PC	SPEC-PC BOARD ASSY	IN	014369	020676	RFE4	N	N	
24	A	8889600	600	PC	IC 8123 TTL TRI-STATE 2-IN MUX	IN	014369	020676	PPP4	N	N	
3	C	88896500	100	PC	INSULATOR-CARD FRAME,UPPER	IN	014369	020676	PPP4	N	N	
4	C	88896700	100	PC	INSULATOR-CARD FRAME,LOWER	IN	014369	020676	PPP4	N	N	
2	C	88896800	100	PC	FRAM-CARD (CASTING)	IN	014369	020676	PPP4	N	N	
48	A	88897100	300	PC	DIODE-SILICON SWITCHING	IN	014369	020676	PPP4	N	N	
11	A	88897900	2200	PC	CAP-FIXED SOLID TANT,6VDC,12UF	IN	014369	020676	PPP4	N	N	
23	A	88919501	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
67	A	88919502	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
68	A	88919503	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
69	A	88919504	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
70	A	88919505	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
71	A	88919506	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
72	A	88919507	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
73	A	88919508	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
74	A	88919509	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
75	A	88919510	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
76	A	88919511	100	PC	IC-TTL,ROM 256BIT PROGRAMMABLE	IN	014733	041376	KT44	N	N	
64	A	88922900	100	PC	IC 74586 QUAD 2IN EXCLUSIVE OR	IN	014369	020676	PPP4	N	N	

INACTIVE

AA 2709-1 REV 7-75

PROJECT ENGINEER

SCMD

INTERDIVISIONAL DOCUMENT

ASSEMBLY PARTS LIST

SPARE CODE
S - SPARE PARTS
N - NON SPARE PARTS

SH 4

88909300	L	INA	D	PWA-PANEL INTERFACE (7EJT)	DSM	HP17	09/20/74	10/09/78	3 / 3
ASSEMBLY NUMBER	REV	CLASS	DW SZ	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	PROCESSING DATE	PAGE NUMBER

MF

FIND NUMBER	DW SZ	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MAKE/BUY PART TYPE	PH	NC	QTY
16	C	94354840	100	PC	CAP-CERAMIC,HIGH-K 50V 1.0UF	IN	014436	041376	PPP4	N	N	
1	A	96744100	100	PC	PWB-PANEL INTERFACE	IN	014369	020676	PPP4	N	N	
8	D	96749200	REF	PC	LOGIC DIAG-PNL INTERFACE (7EJT)	IN	014369	020676	RFE4	N	N	

INACTIVE

NUMBER OF LINE ITEMS = 75
HIGHEST FIND NUMBER = 74

AA 2709-1 REV 7-75

PROJECT ENGINEER

SCMD

INTERDIVISIONAL DOCUMENT

Figure 7-4. Breakpoint Controller Printed Wiring Assembly (88909300) (Sheet 4 of 4)

CONTROLWARE LISTING

A

JOB: PANEL2: LOGUE
1700 MASS STORAGE OPERATING SYSTEM VERSION 4.2 DATE OF RUN: 10/16/75 SYSTEM ID: 1700 # 3 -- 32K SYSTEM (09/06/75)

PPPPPPPPPP	AAAAAAAAAA	NNN	NNN	EEEEEEEEEE	LLL	2222222222
PPPPPPPPPP	AAAAAAAAAA	NNN	NNN	EEEEEEEEEE	LLL	2222222222
PPPPPPPPPP	AAAAAAAAAA	NNN	NNN	EEEEEEEEEE	LLL	2222222222
PPP	AAA	NNNN	NNN	EEE	LLL	222
PPP	AAA	NNNNN	NNN	EEE	LLL	222
PPP	AAA	NNNNN	NNN	EFE	LLL	222
PPPPPPPPPP	AAAAAAAAAA	NNN	NNN	EEEEEEEEEE	LLL	222
PPPPPPPPPP	AAAAAAAAAA	NNN	NNN	EEEEEEEEEE	LLL	222
PPPPPPPPPP	AAAAAAAAAA	NNN	NNN	EEEEEEEEEE	LLL	222
PPP	AAA	NNN	NNNNN	EFE	LLL	222
PPP	AAA	NNN	NNNNN	EEE	LLL	222
PPP	AAA	NNN	NNNN	EFE	LLL	222
PPP	AAA	NNN	NNN	EEEEEEEEEE	LLLLLLLLLLLL	2222222222
PPP	AAA	NNN	NNN	EEEEEEEEEE	LLLLLLLLLLLL	2222222222
PPP	AAA	NNN	NNN	EEEEEEEEEE	LLLLLLLLLLLL	2222222222

*ASSEM

PANEL

PAGE 1

DATE: 10/16/75

```
0001          NAM      PANEL
0002      SEQ      MAC      P2,P3,P4,P5,P6,P7      SEQUENTIAL EXECUTION
0003          IFC      ,EQ,P7*
0004      VFD X8/*2-S,N1/1,N5/0,X2/*P2*,
0005      VFD X2/*P3*,X2/*P4*,X1/*P5*,X4/*P6*,X7/*2-1-S
0006          EIF
0007          IFC      ,NE,P7*
0008      VFD X8/*2-S,N6/0,X2/*P2*,
0009      VFD X2/*P3*,X2/*P4*,X1/*P5*,X4/*P6*,X7/*P7*
0010          EIF
0011          EMC
0012      JUMP      MAC      P2,P3,P4,P5,P6,P7      UNCONDITIONAL JUMP
0013      VFD X8/*2-S,N1/1,N5/0,X2/*P2*,
0014      VFD X2/*P3*,X2/*P4*,X1/*P5*,X4/*P6*,X7/*P7*/2-S
0015          EMC
0016      JTRUE     MAC      P1,P2,P3,P4,P5,P6,P7      JUMP IF CONDITION TRUE
0017      VFD X8/*2-S,N1/1,X5/*P1*,X2/*P2*,
0018      VFD X2/*P3*,X2/*P4*,X1/*P5*,X4/*P6*,X7/*P7*/2-S
0019          EMC
0020      JFALSE    MAC      P1,P2,P3,P4,P5,P6,P7      JUMP IF CONDITION FALSE
0021      VFD X8/*2-S,N1/0,X5/*P1*,X2/*P2*,
0022      VFD X2/*P3*,X2/*P4*,X1/*P5*,X4/*P6*,X7/*P7*/2-S
0023          EMC
```

0025		TEST EQUATES (P1 ACTUALS)	
0026	0000	EQU TRUE(0)	UNCONDITIONAL ONE
0027	0001	EQU REMOTE(01)	LOCAL/REMOTE SW SET TO REMOTE
0028	0002	EQU SWITCHUP(02)	NO PANEL SWITCH IS DEPRESSED
0029	0003	EQU RUNNING(03)	MICRO-RUNNING AND MICRO-MODE
0030	0004	EQU DREADY(04)	DATA IS READY FROM UART
0031	0005	EQU CONDS(05)	PINNF0-OUT SENSE (NOT USED)
0032	0007	EQU EQUAL(07)	COUNTER EQUALS RIT COUNT REG
0033	0008	EQU FC15(08)	RIT 15 OF FUNCTION CONTROL REG
0034	0009	EQU LF9(09)	XMIT DIGIT IS LT OR EQ 9
0035	000A	EQU BPMA(10)	BP-MA SELECTED
0036	000B	EQU MIR(11)	MIR SELECTED
0037	000C	EQU NOTJ(12)	NOT J
0038	0000	EQU HORI(13)	H = I
0039	000F	EQU FRHOR(15)	ERROR F/F
0040	0010	EQU CTEQ31(16)	COUNTER EQUALS 31
0041	0011	EQU CTEQ15(17)	COUNTER EQUALS 15
0042	0012	EQU TERM(18)	TERMINATE CODE RECEIVED (COLON)
0043	0013	EQU CPU16(19)	16 BIT CPU
0044	0014	EQU HEXRCVD(20)	HEX DIGIT RECEIVED
0045	0015	EQU CTEQX3(21)	LOWER 2 BITS OF COUNTER EQUAL 11
0046	0016	EQU CTLSB(22)	LSB OF COUNTER
0047	0017	EQU MEM(23)	MAIN MEMORY SELECTED
0048	0018	EQU MM(24)	MICRO MEMORY SELECTED
0049	0019	EQU NOTF2(25)	FILE 2 SELECTED NOT
0050	001A	EQU NOTAUTO(26)	BUSY OR NOT AUTO-DISPLAY
0051	001B	EQU DIREG(27)	DIRECT REGISTER (ON THIS MOD)
0052	001D	EQU CTEQ0(29)	COUNTER EQUALS ZERO (BITS 0-4)
0053	001F	EQU CTEQZ(31)	COUNTER EQUALS ZERO (BITS 0-5)
0055		LOC COUNT AND SHIFT BUFFER ENABLE EQUATES (P2 ACTUALS)	
0056	0000	EQU LOC0(0)	SHIFT X REG LEFT BY 1
0057	0001	EQU LOC1(1)	DIS1 + 1 TO DIS1, AR
0058	0002	EQU LOC2(2)	X TO SELECTED REG
0059	0003	EQU LOC3(3)	SELECTED REG TO X
0061	0000	EQU SRNOP(0)	SHIFT BUFFER UNAFFECTED
0062	0001	EQU SRR(1)	SHIFT BUFFER SHIFTED RIGHT
0063	0002	EQU SFL(2)	SHIFT BUFFER SHIFTED LEFT
0064	0003	EQU SRL0AD(3)	SHIFT BUFFER LOADED
0066		SHIFT REGISTER ENABLE EQUATES (P3 ACTUALS)	
0067	0000	EQU SRNOP(0)	SHIFT REGISTER UNAFFECTED
0068	0001	EQU SRR(1)	SHIFT REGISTER SHIFTED RIGHT
0069	0002	EQU SRL(2)	SHIFT REGISTER SHIFTED LEFT
0070	0003	EQU SRL0AD(3)	SHIFT REGISTER LOADED

```

0072      *      SHIFT INPUT, DATA MUX, AND XMIT MUX ENABLE EQUATES (P4 ACTUALS)
0073      0000      EQU      XIRIT(0)      ZERO FOR H, ONE FOR I
0074      0001      EQU      FNDARND(1)     END AROUND SHIFT
0075      0002      EQU      XMSR(2)      MSR OF X
0076      0003      EQU      RUFFER(3)     SHIFT BUFFER REGISTER

0078      0000      EQU      FNABDMA(0)     DMA ENABLED
0079      0001      EQU      FNABFC(1)     FUNCTION CONTROL REG ENABLED
0080      0002      EQU      FNABMIR(2)    MIR ENABLED
0081      0003      EQU      FNABPMA(3)    RP-MA ENABLED

0083      0000      EQU      HEXGR9(0)     XMIT HEX GREATER THAN 9
0084      0001      EQU      HEXLE9(1)     XMIT HEX LE 9
0085      0002      EQU      ADDR(2)      XMIT ADDRESS FIELD
0086      0003      EQU      XMITCCR(3)    XMIT CONTROL CHAR REGISTER

0088      0000      EQU      COMPUL(0)     COMPLIMENT UPPER/LOWER F/F
0089      0001      EQU      CLRUL(1)     CLEAR UPPER/LOWER F/F
0090      0000      EQU      DIS1ML(0)    FORCE DIS 1, SET MML AT STRBMMX
0091      0001      EQU      DIS1MU(1)    FORCE DIS 1, SET MMU AT STRBMMX
0092      0002      EQU      DIS0ML(2)    SET MML AT STRBMMX
0093      0003      EQU      DIS0MU(3)    SET MMU AT STRBMMX

0095      *      COUNTER ENABLE EQUATES (P5 ACTUALS)
0096      0000      EQU      SAME(0)      COUNTER UNCHANGED
0097      0001      EQU      INCK(1)      COUNTER INCREMENTED BY 1

0099      *      STROBE EQUATES (P6 ACTUALS)
0100      0000      EQU      NULL(00)     NOT USED
0101      0001      EQU      S300(01)     ENABLE S300 TO CPU LSB
0102      0002      EQU      TRLOAD(02)    LOAD XMIT BUFFER REG
0103      0003      EQU      DRRESET(03)   DATA READY RESET AND SET BUSY
0104      0004      EQU      STRBCCR(04)   STROBE CONTROL CHAR REGISTER
0105      0005      EQU      CLRSH(05)    CLR SHIFT REGISTER
0106      0006      EQU      CLRCT(06)    CLR COUNT, RD STROBE, STROBE BCR
0107      0007      EQU      EXIT(07)     EXIT FROM CPU
0108      0008      EQU      SSREG(08)    STROBE SELECTED REGISTER
0109      000A      EQU      STRBUL(10)    STROBE UPPER/LOWER F/F
0110      000B      EQU      SETERROR(11)  SET ERROR F/F
0111      000C      EQU      STRBMMX(12)   STROBE MM UPPER/LOWER F/F
0112      000D      EQU      CLRBUSY(13)  CLEAR BUSY, CLEAR ERROR
0113      000E      EQU      SETRCPU(14)   CLR TIMING CHAIN (SET RCPU)
0114      000F      EQU      LOADCT(15)   LOAD COUNTER WITH ADDRESS FIELD

```

PANEL

PAGE 4

DATE: 10/16/75

		ADDRESS FIELD EQUATES (P7 ACTUALS)		
0116		EQU	XHEXGR9(\$40)	HEX GR 9 AND CONTROL CHAR
0117	0040	EQU	XHEXLE9(\$30)	HEX LE 9
0118	0030	EQU	XSTAR(\$2A)	ASTERISK
0119	002A	EQU	XCR(\$D)	CHARRIAGE RETURN
0120	000D	EQU	XLF(\$A)	LINE FEED
0121	000A	EQU	XSPACE(\$20)	SPACE
0122	0020			

0124		S	JTRUE	DREADY.....CHKREM	JUMP IF DATA RDY OR SWITCH DOWN
0124	P0000				
0124	P0001				
0125			JTRUE	NOTAUTO.....S	JUMP IF BUSY OR NOT AUTO-DISPLAY
0125	P0002				
0125	P0003				
0126		OKAY	JFALSE	NOTJ,..DISOMJ,..STRBMMX,JLOC	SET MMU. JUMP IF CC IS J
0126	P0004				
0126	P0005				
0127			JFALSE	HORI,..CLRUL,..STRBUL,KL	JUMP IF NOT H OR I
0127	P0006				
0127	P0007				
0128			JFALSE	CTEQZ.....REPLACE	JUMP IF COUNT NOT EQ 0
0128	P0008				
0128	P0009				
0129			JUMPXMITFCR	
0129	P000A				
0129	P000R				
0130		CHKREM	JTRUE	REMOTE.....CHARDY	JUMP IF REMOTE
0130	P000C				
0130	P000D				
0131			SEQTBRLOAD	DEBOUNCE LEADING EDGE
0131	P000E				
0131	P000F				
0132		CHARDY	JTRUE	TERM.....TMRCVD	JUMP IF TERMINATE CODE (COLON)
0132	P0010				
0132	P0011				
0133			JTRUE	ERROR.....RESET	IGNORE IF ERROR F/F SET
0133	P0012				
0133	P0013				
0134			JTRUE	HEXRCVD,SBL0AD.....ENTERHEX	LD BUFFER. JUMP IF HEX
0134	P0014				
0134	P0015				
0135			JTRUE	CTEQZ.....SFTCC	SKIP IF COUNT EQUALS ZERO
0135	P0016				
0135	P0017				
0136		ERRORLOC	JUMPSETERROR,RESET	SET ERROR F/F
0136	P0018				
0136	P0019				
0137		SETCC	JUMPSTRBCCR,RESET	STORE CONTROL CHAR REGISTER
0137	P001A				
0137	P001B				
0138		ENTERHEX	JFALSE	CTEQX3,SBL,SRL,BUFFER,INCK,..*	BUFFER TO SHIFT REG
0138	P001C				
0138	P001D				
0139		RESET	JUMPDRRESET,RETURN	RESET AND JUMP TO RETURN
0139	P001E				
0139	P001F				
0140		TMRCVD	JFALSE	ERROR.....DRRESET,OKAY	JUMP IF NO ERROR
0140	P0020				
0140	P0021				
0141		XMITFCR	JUMP	,SRLOAD,ENABFC,.CLRCT,XMIT32	FCR TO SR. JUMP TO XMIT
0141	P0022				

0141	P0023	D328				
0142			JLOC	JFALSE	NOTAUTO,,,,,,XMITFCR	JUMP IF AUTO-DIS. AND NOT BUSY
0142	P0024	1268				
0142	P0025	0011				
0143			JTRUE	CTEQZ,,,,,,OTHER16		JUMP IF COUNT EQ 0
0143	P0026	13FC				
0143	P0027	0010				
0144			JFALSE	CTLSB,,SRR,CLRUL,INCK,STRBUL,*		POSITION DIGIT NUM
0144	P0028	1458				
0144	P0029	5014	REPLACE	SEQ	.SRLOAD,ENABFC,,CLRCT	FCR TO SR. STROBE BIT CT REG
0145						
0145	P002A	1580				
0145	P002B	0316	CKEQUAL	JTRUE	FQUAL,,,,,,CKJ	JUMP IF BIT COUNT EQ BIT CT REG
0146						
0146	P002C	169C				
0146	P002D	0018				
0147			JUMP	.SRL,ENDARND,INCK,,CKZERO		SHIFT END AROUND
0147	P002E	1780				
0147	P002F	981B				
0148			CKJ	JTRUE	NOTJ,,,,,,CHGBIT	SKIP IF NOT J
0148	P0030	1880				
0148	P0031	001A				
0149			JUMP	SBL,SRL,BUFFFR,INCK,,CKZERO		BUFFER TO SHIFT REG
0149	P0032	1982				
0149	P0033	881B				
0150			CHGBIT	SEQ	.SRL,HIBIT,INCK	CLR OR SET BIT
0150	P0034	1A80				
0150	P0035	881B				
0151			CKZERO	JFALSE	CTEQ0,,,,,,CKEQUAL	JUMP IF COUNT NOT EQ 0
0151	P0036	1874				
0151	P0037	0016				
0152			JUMP	,,,,SSREG,XMITFCR		STORE SR TO FCR
0152	P0038	1C80				
0152	P0039	0411	OTHER16	JUMP	..COMPUL,,STRBUL,SET16	COMPLIMENT UPPER/LOWER F/F
0153						
0153	P003A	1080				
0153	P003B	0527	KL	JFALSE	DIREG,,,,,,NOTDIREG	JUMP IF NOT DIRECT REGISTER
0154						
0154	P003C	1E6C				
0154	P003D	0040				
0155			JTRUE	CTEQZ,,,,,,CKBPMA		SKIP IF CT EQ 0
0155	P003E	1FFC				
0155	P003F	0021				
0156			JFALSE	CTLSB,,ENABFC,INCK,SSREG,*		STROBE SEL. REG. TWICE
0156	P0040	2058				
0156	P0041	1C20	CKBPMA	JFALSE	BPMA,,,,,,CKMIR	SKIP IF NOT BKPT-MA REGISTER
0157						
0157	P0042	2128				
0157	P0043	0023				
0158			JUMP	.SRLOAD,ENABPMA,,CLRCT,XMIT32		BKPT-MA TO SR
0158	P0044	2280				
0158	P0045	F328	CKMIR	JFALSE	MIR,,,,,,XMITFCR	JUMP IF NOT MIR
0159						

```

0159 P0046 232C
0159 P0047 0011
0160
0160 P0048 2480
0160 P0049 E328
0161
0161 P004A 254C
0161 P004B 0328
0162
0162 P004C 26E0
0162 P004D 0028
0163
0163 P004E 2700
0163 P004F 0790
0164
0164 P0050 2884
0164 P0051 0028
0165
0165 P0052 2940
0165 P0053 9829
0166
0166 P0054 2A80
0166 P0055 068A
0167
0167 P0056 28A0
0167 P0057 002A
0168
0168 P0058 2C00
0168 P0059 210D
0169
0169 P005A 2D68
0169 P005B 002F
0170
0170 P005C 2E00
0170 P005D 210A
0171
0171 P005E 2F3C
0171 P005F 0031
0172
0172 P0060 3000
0172 P0061 212A
0173
0173 P0062 3100
0173 P0063 3140
0174
0174 P0064 3200
0174 P0065 2120
0175
0175 P0066 3324
0175 P0067 0636
0176
0176 P0068 3400
0176 P0069 1130

```

	JUMP	.SRLOAD,ENABMIR,.CLRCT,XMIT32	MIR TO SR
XMIT	JFALSE	CPU16,....CLRCT,XMIT32	JUMP IF 32 BIT CPU
	JTRUE	MM,....XMIT32	JUMP IF MM
SET16	SEQLOADCT,16	SET COUNTER TO 16
AMIT32	JTRUE	REMOTE,....CKFC16	JUMP IF REMOTE
	JFALSE	CTEQ31,.,SRL,FNDARND,INCK,.,.	SHIFT BITS TO LITES
TOEND	JUMPCLRBUSY,FNDXMIT	CLEAR BUSY F/F
CKFC15	JTRUE	FC15,....TOFND	SUPPRESS XMIT IF FC15 SET
	SEQ	.,ADDR,.,TBRLoad,XCR	TRANSMIT CHARRIAGE RETURN
	JFALSE	NOTAUTO,....,SKPLF	SKIP LF IF AUTO-DISPLAY
	SEQ	.,ADDR,.,TBRLoad,XLF	TRANSMIT LINE FEED
SKPLF	JFALSE	ERROR,....,NOERROR	JUMP IF ERROR F/F NOT SET
	SEQ	.,ADDR,.,TBRLoad,XSTAR	TRANSMIT ASTERISK
NOERROR	SEQ	.,XMITCCR,.,TBRLoad,XHEXGR9	TRANSMIT CONTROL CHAR
HEXDIGIT	SEQ	.,ADDR,.,TBRLoad,XSPACE	TRANSMIT SPACE
	JFALSE	LE9,....CLRBUSY,GR9	JUMP IF GREATER THAN 9
	SEQ	.,HEXLE9,.,TBRLoad,XHEXLE9	TRANSMIT HEX DIGIT

PANEL

PAGE 8

DATE: 10/16/75

```

0177          JUMP      .....,SHIFT4
0177 P006A 3580
0177 P006B 0037
0178 GR9      SEQ      ..,HEXGR9,..,TBRLOAD,XHEXGR9  TRANSMIT HEX DIGIT
0178 P006C 3600
0178 P006D 0140
0179 SHIFT4  JFALSE   CTEQX3,..,SRL,FNDARND,INCK,..*   SHIFT 4 BITS
0179 P006E 3754
0179 P006F 9837
0180          JFALSE   CTEQ0.....,HEADIGIT   SEND AGAIN IF COUNT NOT EQ 0
0180 P0070 3874
0180 P0071 0032
0181          SEQ      ..,ADDR,..,TBRLOAD,XSPACE  TRANSMIT SPACE
0181 P0072 3900
0181 P0073 2120
0182 ENDXMIT  SEQ      .....,CLRCT           CLEAR COUNTER
0182 P0074 3A80
0182 P0075 0338
0183 RETURN  JTRUE    REMOTE.....,S         RETURN IF REMOTE
0183 P0076 3884
0183 P0077 0000
0184          JTRUE    DREADY.....,*        WAIT SWITCH OPEN
0184 P0078 3C90
0184 P0079 003C
0185          SEQ      .....,TBRLOAD         DEBOUNCE TRAILING EDGE
0185 P007A 3D80
0185 P007B 013E
0186          JTRUE    NOTAUTO.....,S       RETURN IF NOT AUTO-DISPLAY
0186 P007C 3EE8
0186 P007D 0000
0187          JUMP     ..,CLRUL,..,STRUL,S    CLEAR U/L F/F
0187 P007E 3F80
0187 P007F 1500
0188 NOTDIREG JFALSE   RUNNING.....,STOPPED  JUMP IF STOPPED OR MACRO-MODE
0188 P0080 400C
0188 P0081 0042
0189          JUMP     .....,SETERROR,XMITFCR  SET ERROR, JUMP TO XMITFCR
0189 P0082 4180
0189 P0083 0591
0190 STOPPED  JTRUE    CTEQZ.....,SETRCPU,FETCH  JUMP IF COUNT EQ 0
0190 P0084 42FC
0190 P0085 0750
0191          SEQ      .....,CLRCT           CLEAR COUNTER
0191 P0086 4380
0191 P0087 0344
0192          JFALSE   CTEQ15,LOC0,SRL,XMSB,INCK,..*  SHIFT 16 BITS
0192 P0088 4444
0192 P0089 AB44
0193          JFALSE   MH.....,CLRCT,NOTMM    JUMP IF NOT MICRO MEMORY
0193 P008A 4560
0193 P008B 0348
0194          JFALSE   CPU16.....,NOTMM     SKIP IF NOT 16 BIT CPU
0194 P008C 464C

```

```

0194 P008D 0048
0195                               SEQ      LOC2,,DIS0ML,,STRBMMX    X REG TO MMU, SET MML
0195 P008E 4782
0195 P008F 2648
0196 NOTMM JFALSE CTEQ15,LOC0,SRL,XMSB,INCK,,* SHIFT 16 BITS
0196 P0090 4844
0196 P0091 8488
0197 JFALSE MEM,,,,CLRCT,CKF2 SKIP IF NOT MEM
0197 P0092 495C
0197 P0093 034C
0198                               SEQ      ,,DIS1MU,,STRBMMX    FORCE DISPLAY 1
0198 P0094 4A80
0198 P0095 1648
0199                               SEQ      LOC1,,DIS0MU,,STRBMMX    DIS. 1 TO ADDRESS BUFFER
0199 P0096 48A1
0199 P0097 364C
0200 CKF2 JTRUE  NOTF2,LOC2,,,,,CK32 X REG TO SEL REG, JUMP IF NOT F2
0200 P0098 4CE6
0200 P0099 004E
0201                               SEQ      LOC2                               EXEC LOC2 AGAIN TO WRITE INTO F2
0201 P009A 4D82
0201 P009B 004E
0202 CK32 JFALSE CPU16,,,,,SKIPSAVE JUMP IF 32 BIT CPU
0202 P009C 4E4C
0202 P009D 0053
0203                               JUMP    ,,,,SAVEX
0203 P009E 4FA0
0203 P009F 0052
0204 FETCH JFALSE CPU16,,,,,SAVEX JUMP IF 32 BIT CPU
0204 P00A0 504C
0204 P00A1 0052
0205                               SEQ      ,,,,LOADCT,16 SET COUNTER TO 16
0205 P00A2 5100
0205 P00A3 0790
0206 SAVEX JFALSE CTEQ31,LOC0,SRL,XMSB,INCK,S300,* SAVE X REGISTER
0206 P00A4 5240
0206 P00A5 A8D2
0207 SKIPSAVE JFALSE MEM,,DIS0MU,,STRBMMX,NOTMEM JUMP IF NOT MEM
0207 P00A6 535C
0207 P00A7 3657
0208                               SEQ      ,,DIS1MU,,STRBMMX    FORCE DISPLAY 1
0208 P00A8 5480
0208 P00A9 1655
0209                               SEQ      LOC1,,DIS0MU,,STRBMMX    DIS. 1 TO ADDRESS BUFFER
0209 P00AA 5581
0209 P00AB 3656
0210                               SEQ      LOC3,,,,CLRCT READ FROM MEM
0210 P00AC 5683
0210 P00AD 0357
0211 NOTMEM SEQ LOC3,,DIS1MU,,STRBMMX SEL REG TO X, FORCE DISPLAY 1
0211 P00AE 5783
0211 P00AF 1658
0212                               SEQ      0 REQUIRED PASS BEFORE SHIFT

```

PANEL

PAGE 10

DATE: 10/16/75

```
0212 P0080 5880
0212 P0081 0059
0213
0213 P0082 5944
0213 P0083 A8D9
0214
0214 P0084 5A4C
0214 P0085 035E
0215
0215 P0086 5B60
0215 P0087 065E
0216
0216 P0088 5C83
0216 P0089 165D
0217
0217 P008A 5D80
0217 P0089 005E
0218
0218 P008C 5E44
0218 P008D A8DE
0219
0219 P008E 5F81
0219 P008F 03A5
0220
```

JFALSE CTEQ15,LOC0,SML,XMSB,INCK,S300,* SHIFT 16 BITS

JFALSE CPU16,....,CLRCT,NOTCPU16 JUMP IF NOT 16 BIT CPU

JFALSE MM,.,DIS1ML,.,STRMMX,NOTCPU16 JUMP IF NOT MM

SEQ LOC3,.,DIS1MU,.,STRBMMX MML TO X. FORCE DISPLAY 1

SEQ 0 REQUIRED PASS BEFORE SHIFT

NOTCPU16 JFALSE CTEQ15,LOC0,SML,XMSB,INCK,S300,* SHIFT 16 BITS

JUMP LOC1,.,EXIT,XMIT DIS.1 + 1 TO DIS.1. EXIT CPU

END

PGM= 00C0 (192) COM = 0000 (0) DAT = 0000 (0)

EQUIVALENCES

DEF.LINE	NAME	VALUE	REFERENCED AT LINE NUMBER
0000	I	00FF (000255)	
0026	TRUE	0000 (000000)	
0027	REMOTE	0001 (000001)	0130, 0164, 0183
0028	SWITCH	0002 (000002)	
0029	RUNNIN	0003 (000003)	0188
0030	DREADY	0004 (000004)	0124, 0184
0031	CUNDS	0005 (000005)	
0032	EQUAL	0007 (000007)	0146
0033	FC15	0008 (000008)	0167
0034	LE9	0009 (000009)	0175
0035	BPMA	000A (000010)	0157
0036	MIR	000B (000011)	0159
0037	NOTJ	000C (000012)	0126, 0148
003A	HORI	000D (000013)	0127
0039	ERROR	000F (000015)	0133, 0140, 0171
0040	CTEQ31	0010 (000016)	0165, 0206
0041	CTEQ15	0011 (000017)	0192, 0196, 0213, 0218
0042	TERM	0012 (000018)	0132
0043	CPU16	0013 (000019)	0161, 0194, 0202, 0204, 0214
0044	HEXRCV	0014 (000020)	0134
0045	CTEQX3	0015 (000021)	0138, 0179
0046	CTLSB	0016 (000022)	0144, 0156
0047	MEM	0017 (000023)	0197, 0207
0048	MM	0018 (000024)	0162, 0193, 0215
0049	NOTF2	0019 (000025)	0200
0050	NOIAUT	001A (000026)	0125, 0142, 0169, 0186
0051	DIREG	001B (000027)	0154
0052	CTEQ0	001D (000029)	0151, 0180
0053	CTEQZ	001F (000031)	0128, 0135, 0143, 0155, 0190
0056	LOCO	0000 (000000)	0192, 0196, 0206, 0213, 0218
0057	LOC1	0001 (000001)	0199, 0209, 0219
0058	LOC2	0002 (000002)	0195, 0200, 0201
0059	LOC3	0003 (000003)	0210, 0211, 0216
0061	SBNOP	0000 (000000)	
0062	SRR	0001 (000001)	
0063	SBL	0002 (000002)	0138, 0149
0064	SBLOAD	0003 (000003)	0134
0067	SRNOP	0000 (000000)	

SYMBOLS

DEF. LINE	NAME	ADDRESS	REFERENCED AT LINE NUMBER
0124	S	0000	0124, 0124, 0125, 0125, 0125, 0126, 0126, 0127, 0127, 0128, 0128, 0129, 0129, 0130, 0130, 0131
			0131, 0132, 0132, 0133, 0133, 0134, 0134, 0135, 0135, 0136, 0136, 0137, 0137, 0138, 0138, 0139
			0139, 0140, 0140, 0141, 0141, 0142, 0142, 0143, 0143, 0144, 0144, 0145, 0145, 0146, 0146, 0147
			0147, 0148, 0148, 0149, 0149, 0150, 0150, 0151, 0151, 0152, 0152, 0153, 0153, 0154, 0154, 0155
			0155, 0156, 0156, 0157, 0157, 0158, 0158, 0159, 0159, 0160, 0160, 0161, 0161, 0162, 0162, 0163
			0164, 0164, 0165, 0165, 0166, 0166, 0167, 0167, 0168, 0169, 0169, 0170, 0170, 0171, 0171, 0172
			0172, 0173, 0173, 0174, 0174, 0175, 0175, 0176, 0176, 0177, 0177, 0178, 0179, 0179, 0180, 0180, 0181
			0181, 0182, 0182, 0183, 0183, 0184, 0184, 0185, 0185, 0186, 0186, 0187, 0187, 0188, 0188, 0189, 0189, 0190
			0190, 0191, 0191, 0192, 0192, 0193, 0193, 0194, 0194, 0195, 0195, 0196, 0196, 0197, 0197, 0198
			0198, 0199, 0199, 0200, 0200, 0201, 0201, 0202, 0202, 0203, 0203, 0204, 0204, 0205, 0205, 0206
			0206, 0207, 0207, 0208, 0208, 0209, 0209, 0210, 0210, 0211, 0211, 0212, 0212, 0213, 0213, 0214, 0214
			0214, 0215, 0215, 0216, 0216, 0217, 0217, 0218, 0218, 0219, 0219
0124	OKAY	0004	0140
0130	CHKREM	000C	0124
0132	CHAPDY	0010	0170
0134	ERROHL	0018	
0137	SETCC	001A	0135
0139	ENTEMH	001C	0134
0139	RESET	001E	0133, 0136, 0137
0140	TMPCVD	0020	0132
0141	XMITFC	0022	0129, 0142, 0152, 0159, 0189
0142	JLOC	0024	0126
0145	REPLAC	002A	0128
0146	CKEQUA	002C	0151
0148	CKJ	0030	0146
0150	CMGRIT	003A	0148
0151	CKZEHO	0036	0147, 0149
0153	OTHEM1	003A	0143
0154	KL	003C	0127
0157	CKBPMA	0042	0155
0159	CKMIR	0046	0157
0161	XMIT	004A	0219
0163	SET16	004E	0153
0164	XMIT32	0050	0141, 0158, 0160, 0161, 0162
0164	TOEND	0054	0167
0167	CKFC15	0056	0164
0171	SKPLF	005E	0169
0173	NOERRO	0062	0171
0174	HEXDIG	0064	0180
0178	GR9	006C	0175
0179	SHIFT4	006E	0177

PANEL

PAGE 14

DATE: 10/16/75

0182	ENDXMI	0074	0166
0183	RETURN	0076	0139
018A	NOTDIR	0080	0154
0190	STOPPE	0084	0188
0196	NOTMM	0090	0193 0194
0200	CKF2	0098	0197
0202	CK32	009C	0200
0204	FETCH	00A0	0190
0206	SAVEX	00A4	0203 0204
0207	SKIPSA	00A6	0202
0211	NOTMEM	00AE	0207
021A	NOTCPU	00BC	0214 0215

*** ALPHABETICAL SORT OF SYMBOLS ***

ADDR	0085	BPMA	0035	BUFFER	0076	CHARDY	0132	CHGRT	0150	CHKPEM	0130	CK32	0202	CK9PMA	0157	CKEQUA	0146
CKF2	0200	CKFC15	0167	CKJ	0148	CKMIR	0159	CKZERO	0151	CLRHUS	0112	CLRCT	0106	CLRSR	0105	CLRUL	0089
COMPUL	0088	COND5	0031	CPU16	0043	CTE00	0052	CTEQ15	0041	CTEQ31	0040	CTEQX3	0045	CTEQZ	0053	CTLSR	0046
DIREG	0051	DISOHL	0092	DISOMU	0093	DIS1ML	0090	DIS1MU	0091	DPEADY	0030	DRRESE	0103	ENARM	0078	ENARFC	0079
ENABMI	0080	ENABPM	0081	ENDARN	0074	ENDXMI	0182	ENTERM	0138	EQUAL	0032	ERHOR	0039	ERRORL	0136	EXIT	0107
FC15	0033	FETCH	0204	GR9	0178	HEXDIG	0174	HEXGR9	0083	HFXLF9	0084	HEXRCV	0044	HIMIT	0073	HORI	0038
I	0000	INCK	0097	JLUC	0142	KL	0154	LE9	0034	LOADCT	0114	LOCO	0056	LOC1	0057	LOC2	0058
LOC3	0059	MEM	0047	MIR	0036	MM	0048	NOERRO	0173	NOTAUT	0050	NOTCPU	0218	NOTDIR	0188	NOTF2	0049
NOTJ	0037	NOTMEM	0211	NOTMM	0125	NULL	0100	OKAY	0126	OTHEP1	0153	REWOTE	0027	REPLAC	0145	RESET	0139
RETURN	0183	RUNNIN	0029	S	0124	SJ00	0101	SAME	0094	SAVFX	0206	SRL	0063	SHLOAD	0064	SBNOP	0061
SRK	0062	SET15	0163	SETCC	0137	SETERR	0110	SETRCP	0113	SHIFT4	0179	SKIPSA	0207	SKPLF	0171	SRL	0069
SRLOAD	0070	SRNOP	0067	SRK	0068	SSREG	0108	STOPPE	0190	STRACC	0104	STRAM	0111	STRRUL	0109	SWITCH	0028
TRKLOA	0102	TERM	0042	TMCVD	0140	TOEND	0164	TRUE	0020	XCP	0120	XHEXGR	0117	XHEXLE	0118	XL	0121
XMIT	0161	XMIT32	0164	XMITCC	0086	XMITFC	0141	XMSB	0075	XSPACE	0122	XSTAR	0119				

UART DETAILS

B

This appendix contains the detailed descriptions of the operating characteristics applicable to the UART terminals (table B-1), a functional block diagram of the UART (figure B-1), and the controlware program listing of the breakpoint controller. The UART provides the RS232 communication interface between the breakpoint controller

and RS232 compatible peripherals. The controlware program sequences and controls the breakpoint controller operations. For a detailed description of the controlware format, refer to the paragraph on Breakpoint Controller Controlware in section 4.

TABLE B-1. UART PIN DESIGNATIONS AND DESCRIPTIONS

Designation	Pin No.	Description	Function
RRD (RBRD)	4	Receiver register disconnect	A high-level input voltage, V_{IH} , applied to this line disconnects the receiver holding register outputs from the RR8-RR1 data outputs (pins 5-12).
RR8-RR1	5-12	Receiver buffer (holding) register data	The parallel contents of the receiver register appear on these lines if a low-level input voltage, V_{IL} , is applied to RRD. For character formats of fewer than eight bits, received characters are right-justified (RR1 = LSB) and the truncated bits are forced to a low output voltage V_{OL} .
RPE	13	Parity error	A high-level output voltage, V_{OH} , on this line indicates that the received parity does not compare to that programmed by the even parity enable (PS) control line (pin 39). This output is updated each time a character is transferred to the receiver buffer register. RPE lines from a number of arrays can be bussed together since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
RFE	14	Framing error	A high-level output voltage, V_{OH} , on this line indicates that the received character has no valid stop bit; that is, the bit following the parity bit (if programmed) is not a high-level voltage. This output is updated each time a character is transferred to the receiver holding register. RFE lines from a number of arrays can be bussed together, since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
ROE	15	Over-run error	A high-level output voltage, V_{OH} , on this line indicates that the data received flag RDR (pin 19) was not reset before the next character was transferred to the receiver holding register. ROE lines from a number of arrays can be bussed together, since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
SFD	16	Status flag disconnect	A high-level input voltage, V_{IH} , applied to this pin disconnects RPE, RFE, ROE, RDR, and TBRE, allowing them to be bus-connected.
RRC	17	Receiver register	The receiver clock frequency is 16 times the desired receiver shift rate.
DRR	18	Data received reset	A low-level input voltage, V_{IL} , applied to this line resets the DRR line.
RDR	19	Data ready	A high-level output voltage, V_{OH} , indicates that an entire character has been received and transferred to the receiver holding register.
RRI	20	Receiver input	Serial input data received on this line enters the receiver register at a point determined by the character length, parity, and the number of stop bits. A high-level input voltage, V_{IH} , must be present when data is not being received.

TABLE B-1. UART PIN DESIGNATIONS AND DESCRIPTIONS (Contd)

Designation	Pin No.	Description	Function															
MR	21	Master reset	This line is strobed to a high-level input voltage, V_{IH} , to clear the logic. It resets the transmitter and receiver registers, the receiver holding register, RFE, ROE, RPE, and DRR, and sets the serial output line to a high-level output voltage, V_{OH} .															
TBRE	22	Transmitter buffer register empty	A high-level output voltage, V_{OH} , on this line indicates that the transmitter holding register has transferred its contents to the transmitter register and may be loaded with a new character.															
TBRL	23	Transmitter buffer register load	A low-level input voltage, V_{IL} , applied to this line enters a character into the transmitter holding register. A transition from a low-level input voltage, V_{IL} , to a high-level input voltage, V_{IH} , transfers the character into the transmitter register if the register is not in the process of transmitting a character. If a character is being transmitted, the transfer is delayed until the transmission is completed. Upon completion, the new character is automatically transferred simultaneously with the initiation of the serial transmission of the new character.															
TRE	24	Transmitter register empty	A high-level output voltage, V_{OH} , on this line indicates that the transmitter register has completed serial transmission of a full character including stop bits. It remains at this level until the start of transmission of the next character.															
TRO	25	Transmitter register output	The contents of the transmitter register (start bit, data bits, parity bit, and stop (bit), are serially shifted out on this line. When no data is being transmitted, this line remains at a high-level output voltage, V_{OH} . Start of transmission is defined as the transition of the start bit from a high-level output voltage, V_{OH} , to a low-level voltage, V_{OL} .															
TR1-TR8	26-33	Transmitter register, data inputs	The character to be transmitted is loaded into the transmitter holding register on these lines with the TBRL strobe. If a character of less than 8 bits has been selected (by WLS1 and WLS2), the character is right-justified to the least significant bit, RR1, and the excess bits are disregarded. A high-level input voltage, V_{IH} , causes a transmission of a high-level output voltage, V_{OH} .															
CRL	34	Control register load	A high-level input voltage, V_{IH} , on this line loads the control register with the control bits (WLS1, WLS2, RPE, PI, and SBS). This line may be strobed or hard-wired to a high-level input voltage, V_{IH} .															
PI	35	Parity inhibit	A high-level input voltage, V_{IH} , on this line inhibits the parity generation and verification circuits and clamps the RPE output (pin 13) to V_{OL} . If parity is inhibited, the stop bits immediately follow the last data bit on transmission.															
SBS	36	Stop bit(s) select	This line selects the number of stop bits to be transmitted after the parity bit. A high-level input voltage, V_{IH} , on this line selects two stop bits; a low-level input voltage, V_{IL} , selects a single stop bit. Selection of two stop bits when programming a 5-bit word generates 1/5 bits from the UART.															
WLS2-WLS1	37,38	Word length select	These lines select the character length (exclusive of parity) as follows: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>WLS2</th> <th>WLS1</th> <th>Word Length</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>5 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>6 bits</td> </tr> <tr> <td>1</td> <td>0</td> <td>7 bits</td> </tr> <tr> <td>1</td> <td>1</td> <td>8 bits</td> </tr> </tbody> </table>	WLS2	WLS1	Word Length	0	0	5 bits	0	1	6 bits	1	0	7 bits	1	1	8 bits
WLS2	WLS1	Word Length																
0	0	5 bits																
0	1	6 bits																
1	0	7 bits																
1	1	8 bits																
PS	39	Parity select	This line determines whether even or odd parity is to be generated by the transmitter and checked by the receiver. A high-level input voltage, V_{IH} , selects even parity; a low-level input voltage, V_{IL} , selects odd parity.															
TRC	40	Transmitter register clock	The transmitter clock frequency is 16 times the desired transmitter rate.															

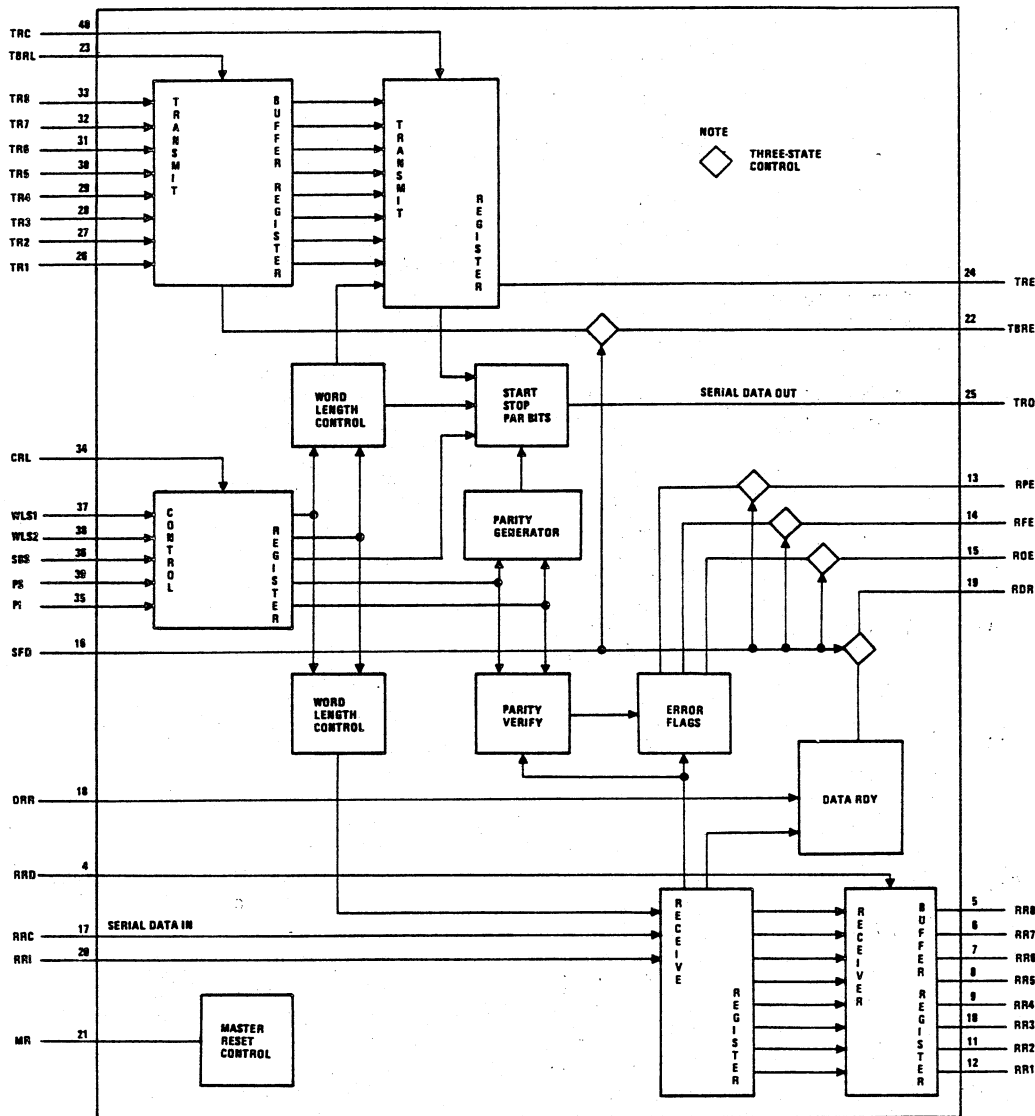


Figure B-1. UART Functional Block Diagram

1948

...

...

...

...

COMMENT SHEET

MANUAL TITLE: CDC® DT120, DT195, FC402 Breakpoint Controller and Breakpoint Panel
Hardware Reference/Maintenance Manual

PUBLICATION NO.: 96729000

REVISION: D

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ STATE: _____ ZIP CODE: _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

Please reply

No reply necessary

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND TAPE

STAPLE

STAPLE

FOLD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
PUBLICATIONS AND GRAPHICS DIVISION
4455 EASTGATE MALL
SAN DIEGO, CALIFORNIA 92121



CUT ALONG LINE

FOLD

FOLD

r
e
g



e
r

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.



CONTROL DATA CORPORATION