IBM Field Engineering

Maintenance Manual

2065 Processing Unit

SY27-2270-0

Preface

This manual, Form Y27-2270-0, is a completely revised edition of the maintenance manual for the 2065 Processing Unit and obsoletes the previous edition, Form Y27-2039-1. The following CE memorandum service aids (SA's) have been incorporated into this manual:

SA Category	SA No.
ROS	A1,A2,A3,A4,A6,A7,A8,A9,A11,A13,A14
DIAG	A7,A8,A9,A10,A11,A12,A13,A14,E3
PWR	5,11,12
GEN	11,12

The information in this manual is presented in six chapters: Chapter 1 contains reference data; Chapter 2 describes operating procedures, maintenance features, and diagnostic programs; Chapter 3 contains preventive maintenance information; Chapter 4 contains maintenance procedures and service aids which are to be used in conjunction with the diagnostic techniques diagrams (category 1) in the 2065 FEMDM (Form Y27-2038-0); Chapter 5 contains power supply maintenance information; and Chapter 6 identifies frame locations.

Companion and prerequisite manuals to the 2065 FEMM are:

2065 Processing Unit, FETOM (Volume 1), Form Y27-2036-0

2065 Processing Unit, FETOM (Volume 2), Form Y27-2037-0

2065 Processing Unit, FEMDM, Form Y27-2038-0

It is assumed that the reader has been trained on the 2065 and that he is familiar with ALD interpretation and the use of oscilloscope and related tools.

First Edition (October, 1969)

Information in this manual is subject to change from time to time. Any such change will be reported in subsequent revisions or FE Supplements.

Text for this manual has been prepared with the IBM SELECTRIC ® Composer.

A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be addressed to: IBM Systems Development Division, Product Publications, Dept 520, Neighborhood Road, Bldg. 960-1, Kingston, N.Y., 12401.

Contents

Illustrations

Tables

Abbreviations

ABC	AB register byte counter
ac	alternating current
ACC	additive card code
adr	address, addressed, addressing
ALD	automated logic diagram
amp	ampere
ASC	address store compare
ATN	alternate test number
BCD	binary-coded decimal
BCU	bus control unit
C	capacitor
CAS	control automation system
CAW	channel address word
CB	circuit breaker
CC	condition code
CCC	channel control check
CCW	channel command word
CE	customer engineer
CEDA	CPU error detection and analysis
charistic	characteristic
C/I	converter/inverter
CLD	control automation system logic diagram
CPU	Central Processing Unit
CR	diode
CROS	capacitive read-only storage
CSW	channel status word
CT	conditional terminate
dc	direct current
dec div	decimal divide
dec ovflo	decimal overflow
DM	diagnostic monitor
DSBL	disable
DX	first byte in a series of destination bytes
DX + 1	second byte in a series of destination bytes
DX + 2	third byte in a series of destination bytes
end op	end operation
EPO	emergency power off
ERSLT	expected result
exp ovflo	exponent overflow
exp unflo	exponent underflow
F	fuse
FEMDM	Field Engineering Maintenance Diagrams Manual
FEMM	Field Engineering Maintenance Manual
FETOM	Field Engineering Theory of Operation Manual
fix-pt div	fixed-point divide
fix-pt ovflo	fixed-point overflow
FLT	fault locating test
flt-pt div	floating-point divide
FLUT	Fault Locating Utility program
FPR	floating-point register
fract	fraction
GCT	gate control trigger
G/F	good/fail
GIS	general initialization sequence
GPR	general-purpose register
hex	hexadecimal
HSS	high-speed storage
Hz	Hertz
IC	instruction counter
I-Fetch	instruction fetching
ILC	instruction length code
INTRM	intermittent
I/O	input/output
IPL	initial program load
K	kilo
K	relay
kHz	kilohertz
LAL	local storage address latches
LAR	local storage address register
LCS	large capacity storage
LS	local storage
LSWR	local storage working register
MAR	memory address register
max	maximum
MCW	maintenance control word
mHz	megahertz
MMSC	maintenance mode stop clock
MP	multiprocessor
MPAS	multiprocessor additional storage (5-8) feature
ms	millisecond
multisys	multisystem
no op	no operation
ns	nanosecond
oc	overcurrent
op code	operation code
oper	operation
opr	operand
ov	overvoltage
P	parity
PAA	parallel adder A-side
PAB	parallel adder B-side
PAL	parallel adder latch
pf	picofarad
PK	power contactor
PN	part number
PP	partial product
PQ	partial quotient
PREV ADR A	previous address in ROSPARA
priv oper	privileged operation
prot	protection
PS	power supply
PSW	program status word
R	resistor
ROS	read-only storage
ROSAR	read-only storage address register
ROSBR	read-only storage backup register
ROSDR	read-only storage data register
ROSPARA	read-only storage previous address register A
ROSPARB	read-only storage previous address register B
RPQ	request for price quotation

SAA	serial adder A-side
SAB	serial adder B-side
SAB	storage address bus
SAL	serial adder latch
SAP	storage address protect
SAR	storage address register
SBA	serial adder bus A
SBB	serial adder bus B
SBBD	select bus base drive
SBED	select bus emitter drive
SCOPEX	scoping index
SCR	silicon-controlled rectifier
SDBI	storage data bus in
SDBO	storage data bus out
signif	significance
SILI	suppress incorrect length indication
SIP	Systems Integration Program
SLT	solid logic technology
SMS	standard modular system
SOROS	scan out read-only storage
spec	specification
SRL	Systems Reference Library
STAT	status trigger
STC	ST register byte counter
stg	storage
STOR CHK	storage check
SW BD	switch board
sync	synchronizing
T	transformer
TB	terminal board
T(DX)	table byte specified by DX
T(DX + 1)	table byte specified by DX + 1
TIC	transfer in channel
TN	test number
UDC	unit data check
uf	microfarad
usec	microsecond
UT	unconditional terminate
V	volt
VFL	variable-field length
$\geq$	greater than or equal to
$\geqslant$	greater than or equal to
$\leq$	less than or equal to
$\leqslant$	less than or equal to
=	equal to
$\neq$	not equal to
&	and

Safety

Safety cannot be overemphasized. Always be familiar with and heed all safety precautions. Know the procedures for artificial respiration. These precautions and procedures are outlined on IBM Form 229-1264-1.

Before making any hardware changes, resistance measurements, or replacements, be sure all power is off and all capacitors are fully discharged; do not rely on bleeder resistors. Never work alone. If in doubt: Don't!

The following safety points are repeated for emphasis:

DANGER
The internal circuitry of the converter/inverter is not isolated from the power source; therefore, a lethal potential to ground is present whenever power is on. Exercise extreme caution. This potential exists on the SMS cards, heat sinks, and terminals to the regulators.

DANGER
The converter/inverter assembly weighs approximately 150 pounds. To avoid personal injury, three men are required to remove and replace it. The area must be clear to permit easy and safe manipulation of the assembly.

DANGER
The input terminals of the regulators are not isolated from the power source; therefore, a lethal potential to ground is present whenever power is on. Exercise extreme caution.

This chapter contains reference information needed for servicing the System/360 Model 65. Information is in tabular form where possible. Diagrams are largely reserved for the FEMDM but are cross-referenced here when appropriate.

1.1 DATA FLOW AND CONTROL

Data flow diagrams are found in the *2065 Processing Unit FEMDM,* Form Y27-2038. These diagrams fall into the following categories:

System data flow	Diagram 3-1
CPU data flow	Diagram 3-2
Data flow by instruction class	Diagrams 3-3 to 3-8
Scan data flow and control	Diagram 8-101

Data flow is discussed in Chapter 1, Section 1, of the *2065 Processing Unit FETOM,* Form Y27-2036.

1.2 SYSTEM CODING

The codes and formats used for instructions and data within the Model 65 are described in the subparagraphs that follow.

1.2.1 Hexadecimal/Decimal Conversion

Powers of 2 and 16 are listed in Figure 1-1. The charts and procedures for converting binary, hexadecimal, and decimal numbers and fractions are shown in Figure 1-2.

2^n	n	2^{-n}	16^n	n
1	0	1.0	1	0
2	1	0.5	16	1
4	2	0.25	256	2
8	3	0.125	4 096	3
			65 536	4
16	4	0.062 5	1 048 576	5
32	5	0.031 25	16 777 216	6
64	6	0.015 625	268 435 456	7
128	7	0.007 812 5	4 294 967 296	8
			68 719 476 736	9
256	8	0.003 906 25	1 099 511 627 776	10 = A
512	9	0.001 953 125	17 592 186 044 416	11 = B
1 024	10	0.000 976 562 5	281 474 976 710 656	12 = C
2 048	11	0.000 488 281 25	4 503 599 627 370 496	13 = D
			72 057 594 037 927 936	14 = E
4 096	12	0.000 244 140 625	1 152 921 504 606 846 976	15 = F
8 192	13	0.000 122 070 312 5		
16 384	14	0.000 061 035 156 25		
32 768	15	0.000 030 517 578 125		
65 536	16	0.000 015 258 789 062 5		
131 072	17	0.000 007 629 394 531 25		
262 144	18	0.000 003 814 697 265 625		
524 288	19	0.000 001 907 348 632 812 5		
1 048 576	20	0.000 000 953 674 316 406 25		
2 097 152	21	0.000 000 476 837 158 203 125		
4 194 304	22	0.000 000 238 418 579 101 562 5		
8 388 608	23	0.000 000 119 209 289 550 781 25		
16 777 216	24	0.000 000 059 604 644 775 390 625		
33 554 432	25	0.000 000 029 802 322 387 695 312 5		
67 108 864	26	0.000 000 014 901 161 193 847 656 25		
134 217 728	27	0.000 000 007 450 580 596 923 828 125		
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5		
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25		
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625		
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5		
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25		
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125		
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5		
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25		

Decimal Values

Figure 1-1. Powers of 2 and 16

Hexadecimal and Decimal Integer Conversion Table

HALFWORD								HALFWORD							
BYTE				BYTE				BYTE				BYTE			
0123		4567		0123		4567		0123		4567		0123		4567	
Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	268,435,456	1	16,777,216	1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	536,870,912	2	33,554,432	2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	805,306,368	3	50,331,648	3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	1,073,741,824	4	67,108,864	4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	1,342,177,280	5	83,886,080	5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	1,610,612,736	6	100,663,296	6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	1,879,048,192	7	117,440,512	7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	2,147,483,648	8	134,217,728	8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	2,415,919,104	9	150,994,944	9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	2,684,354,560	A	167,772,160	A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	2,952,790,016	B	184,549,376	B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	3,221,225,472	C	201,326,592	C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	3,489,660,928	D	218,103,808	D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	3,758,096,384	E	234,881,024	E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	4,026,531,840	F	251,658,240	F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
	8		7		6		5		4		3		2		1

Hexadecimal and Decimal Fraction Conversion Table

HALFWORD																	
BYTE								BYTE									
	0123				4567				0123				4567				
Hex	Decimal			Hex	Decimal			Hex	Decimal			Hex	Decimal Equivalent				
.0	.0000	.00	.0000	0000	.000	.0000	0000	0000	.0000	.0000	0000	0000	0000				
.1	.0625	.01	.0039	0625	.001	.0002	4414	0625	.0001	.0000	1525	8789	0625				
.2	.1250	.02	.0078	1250	.002	.0004	8828	1250	.0002	.0000	3051	7578	1250				
.3	.1875	.03	.0117	1875	.003	.0007	3242	1875	.0003	.0000	4577	6367	1875				
.4	.2500	.04	.0156	2500	.004	.0009	7656	2500	.0004	.0000	6103	5156	2500				
.5	.3125	.05	.0195	3125	.005	.0012	2070	3125	.0005	.0000	7629	3945	3125				
.6	.3750	.06	.0234	3750	.006	.0014	6484	3750	.0006	.0000	9155	2734	3750				
.7	.4375	.07	.0273	4375	.007	.0017	0898	4375	.0007	.0001	0681	1523	4375				
.8	.5000	.08	.0312	5000	.008	.0019	5312	5000	.0008	.0001	2207	0312	5000				
.9	.5625	.09	.0351	5625	.009	.0021	9726	5625	.0009	.0001	3732	9101	5625				
.A	.6250	.0A	.0390	6250	.00A	.0024	4140	6250	.000A	.0001	5258	7890	6250				
.B	.6875	.0B	.0429	6875	.00B	.0026	8554	6875	.000B	.0001	6784	6679	6875				
.C	.7500	.0C	.0468	7500	.00C	.0029	2968	7500	.000C	.0001	8310	5468	7500				
.D	.8125	.0D	.0507	8125	.00D	.0031	7382	8125	.000D	.0001	9836	4257	8125				
.E	.8750	.0E	.0546	8750	.00E	.0034	1796	8750	.000E	.0002	1362	3046	8750				
.F	.9375	.0F	.0585	9375	.00F	.0036	6210	9375	.000F	.0002	2888	1835	9375				
1		2			3				4								

Example of use of fraction table. Convert hex .ABC to decimal value:

Find: .A (highest order) = .6250
.0B (next order) = .0429 6875
.00C (next order) = .0029 2968 7500
Add all ordered values: = .6708 9843 7500 (decimal equivalent)

Note: If the problem were to convert hex 1B6.ABC to decimal value, all that would be necessary has been shown above and in the first example (consider the two examples combined). The decimal equivalent is: 438.6708 9843 7500.

Figure 1-2. Binary and Hexadecimal Conversion Charts (Part 1 of 2)

TO CONVERT HEXADECIMAL TO DECIMAL

1. Locate the column of decimal numbers corresponding to the leftmost digit or letter of the hexadecimal; select from this column and record the number that corresponds to the position of the hexadecimal digit or letter.
2. Repeat step 1 for the next (second from the left) position.
3. Repeat step 1 for the units (third from the left) position.
4. Add the numbers selected from the table to form the decimal number.

EXAMPLE	
Conversion of Hexadecimal Value	D34
1. D	3328
2. 3	48
3. 4	4
4. Decimal	3380

TO CONVERT DECIMAL TO HEXADECIMAL

1. (a) Select from the table the highest decimal number that is equal to or less than the number to be converted.
   (b) Record the hexadecimal of the column containing the selected number.
   (c) Subtract the selected decimal from the number to be converted.
2. Using the remainder from step 1 (c), repeat all of step 1 to develop the second position of the hexadecimal (and a remainder).
3. Using the remainder from step 2, repeat all of step 1 to develop the units position of the hexadecimal.
4. Combine terms to form the hexadecimal number.

EXAMPLE	
Conversion of Decimal Value	3380
1. D	-3328
	52
2. 3	-48
	4
3. 4	-4
4. Hexadecimal	D34

To convert integer numbers greater than the capacity of table, use the techniques below:

HEXADECIMAL TO DECIMAL

Successive cumulative multiplication from left to right, adding units position.

Example: $D34_{16} = 3380_{10}$

D = 13
x16
208
3 = + 3
211
x16
3376
4 = + 4
3380

DECIMAL TO HEXADECIMAL

Divide and collect the remainder in reverse order.

Example: $3380_{10} = X_{16}$

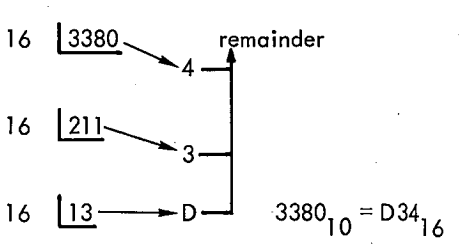


Figure 1-2. Binary and Hexadecimal Conversion Charts (Part 2 of 2)

1.2.2 Eight-Bit Zoned Character Codes (USASCII-8 & EBCDIC)

The two standard data codes are the Extended Binary-Coded-Decimal Interchange Code (EBCDIC) and the USA Standard Code for Information Interchange extended to eight bits (USASCII-8). Both EBCDIC and USASCII-8 provide for 256 possible characters. Each character is composed of eight bits (one byte) and each bit position is assigned a number. The numbering conventions differ for each code. The conventions are:

Code	Bit Position
EBCDIC	0 1 2 3 4 5 6 7
USASCII-8	8 7 6 5 4 3 2 1

In addition the seven-bit USASCII code can be imbedded, or recoded, into USASCII-8 as follows:

Code	Bit Position
USASCII	7 6 7 5 4 3 2 1
USASCII-8	8 7 6 5 4 3 2 1

The EBCDIC codes are shown in Figure 1-3. The USASCII-8 codes are shown in Figure 1-4.

In Figure 1-3, the 256-position chart outlined by the heavy black lines, shows the graphic characters and control character representations for EBCDIC. The bit-position numbers, bit patterns, hexadecimal representations and card hole patterns for these and other possible EBCDIC characters are also shown.

Bit Positions 4,5,6,7	Second Hexadecimal Digit	Digit Punches	00				01				10				11			
			00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
			12				12	12	5	12	12	12		12	12			
				11				11	11	11		11	11	11		11		
					0		0		0	0	0		0	0			0	
			9	9	9	9	9	9	9	9								
0000	0	8-1	(1) NUL	(2) DLE	(3) DS	(4)	(5) SP	&	-	(8)					(9)	(10)	(11)	(12) 0	8-1
0001	1	1	SOH	DC1	SOS				(13) /		a	j			A	J	(14)	1	1
0010	2	2	STX	DC2	FS	SYN					b	k	s		B	K	S	2	2
0011	3	3	ETX	TM							c	l	t		C	L	T	3	3
0100	4	4	PF	RES	BYP	PN					d	m	u		D	M	U	4	4
0101	5	5	HT	NL	LF	RS					e	n	v		E	N	V	5	5
0110	6	6	LC	BS	ETB	UC					f	o	w		F	O	W	6	6
0111	7	7	DEL	IL	ESC	EOT					g	p	x		G	P	X	7	7
1000	8	8		CAN							h	q	y		H	Q	Y	8	8
1001	9	8-1		EM							i	r	z		I	R	Z	9	9
1010	A	8-2	SMM	CC	SM		¢	!	(15)	:									8-2
1011	B	8-3	VT	CU1	CU2	CU3		$	,	#									8-3
1100	C	8-4	FF	IFS		DC4	<	*	%	@									8-4
1101	D	8-5	CR	IGS	ENQ	NAK	(	)	_	'									8-5
1110	E	8-6	SO	IRS	ACK		+	;	>	=									8-6
1111	F	8-7	SI	IUS	BEL	SUB	|	¬	?	"									8-7
			12				12				12	12		12	12	12		12	
				11				11				11	11	11		11	11	11	
					0				0		0		0	0	0		0	0	
			9	9	9	9									9	9	9	9	

Bit Positions 0,1
Bit Positions 2,3
First Hexadecimal Digit
Zone Punches
Zone Punches
Digit Punches

Card Hole Patterns

(1)	12-0-9-8-1	(5)	No Punches	(9)	12-0	(13)	0-1
(2)	12-11-9-8-1	(6)	12	(10)	11-0	(14)	11-0-9-1
(3)	11-0-9-8-1	(7)	11	(11)	0-8-2	(15)	12-11
(4)	12-11-0-9-8-1	(8)	12-11-0	(12)	0		

Control Character Representations

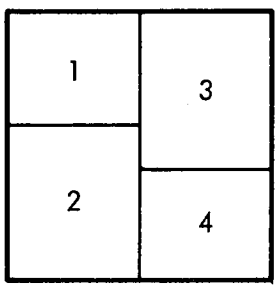
ACK	Acknowledge	EOT	End of Transmission	PF	Punch Off
BEL	Bell	ESC	Escape	PN	Punch On
BS	Backspace	ETB	End of Transmission Block	RES	Restore
BYP	Bypass	ETX	End of Text	RS	Reader Stop
CAN	Cancel	FF	Form Feed	SI	Shift In
CC	Cursor Control	FS	Field Separator	SM	Set Mode
CR	Carriage Return	HT	Horizontal Tab	SMM	Start of Manual Message
CU1	Customer Use 1	IFS	Interchange File Separator	SO	Shift Out
CU2	Customer Use 2	IGS	Interchange Group Separator	SOH	Start of Heading
CU3	Customer Use 3	IL	Idle	SOS	Start of Significance
DC1	Device Control 1	IRS	Interchange Record Separator	SP	Space
DC2	Device Control 2	IUS	Interchange Unit Separator	STX	Start of Text
DC4	Device Control 4	LC	Lower Case	SUB	Substitute
DEL	Delete	LF	Line Feed	SYN	Synchronous Idle
DLE	Data Link Escape	NAK	Negative Acknowledge	TM	Tape Mark
DS	Digit Select	NL	New Line	UC	Upper Case
EM	End of Medium	NUL	Null	VT	Vertical Tab
ENQ	Enquiry				

Special Graphic Characters

¢	Cent Sign	-	Minus Sign, Hyphen
.	Period, Decimal Point	/	Slash
<	Less-than Sign	,	Comma
(	Left Parenthesis	%	Percent
+	Plus Sign	_	Underscore
|	Logical OR	>	Greater-than Sign
&	Ampersand	?	Question Mark
!	Exclamation Point	:	Colon
$	Dollar Sign	#	Number Sign
*	Asterisk	@	At Sign
)	Right Parenthesis	'	Prime, Apostrophe
;	Semicolon	=	Equal Sign
¬	Logical NOT	"	Quotation Mark

Figure 1-3. EBCDIC Code

To find the card hole patterns for most characters, partition the 256-position chart into four blocks as follows:



Block 1: Zone punches at top of chart; digit punches at left
Block 2: Zone punches at bottom of chart; digit punches at left
Block 3: Zone punches at top of chart; digit punches at right
Block 4: Zone punches at bottom of chart; digit punches at right

Fifteen positions in the chart are exceptions to the above arrangement. These positions are indicated by small numbers in the upper right corners of their corresponding boxes, and the card hole patterns for these positions are given at the bottom of the chart. Bit-position numbers, bit patterns, and hexadecimal representations for these positions are found in the usual manner.

Following are some examples of the use of the EBCDIC chart:

Character	Type	Bit Pattern	Hex	Hole Pattern	
				Zone Punches	Digit Punches
PF	Control Character	00 00 0100	04	12 - 9	- 4
%	Special Graphic	01 10 1100	6C	0	- 8 - 4
R	Upper Case	11 01 1001	D9	11	- 9
a	Lower Case	10 00 0001	81	12 - 0	- 1
	Control Character, function not yet assigned	00 11 0000	30	12 - 11 - 0 - 9	- 8 - 1

Bit Positions
01 23 4567

1.2.3 Instructions and Instruction Formats

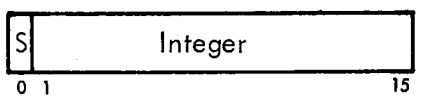
Tables 1-1 through 1-7 list the 2065 instructions according to instruction class. Model 65 instruction formats are shown in Figure 1-5.

1.2.4 Data Formats

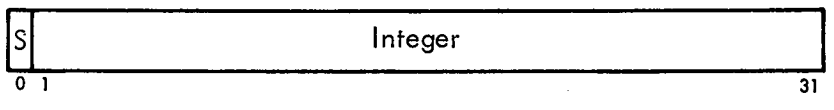
The data may be divided into four classifications:

1. Fixed-point numbers, having a binary radix and a fixed length:

   Halfword Operand



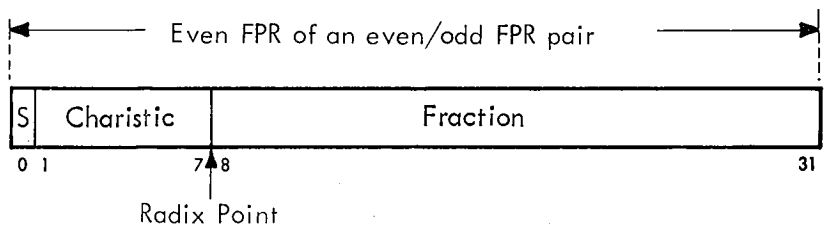
   Word Operand



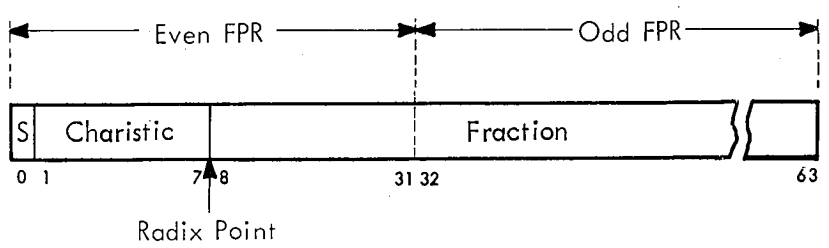
   Doubleword Operand

2. Floating-point numbers, represented by a seven-bit characteristic and a signed hex fraction:

   Short Operand

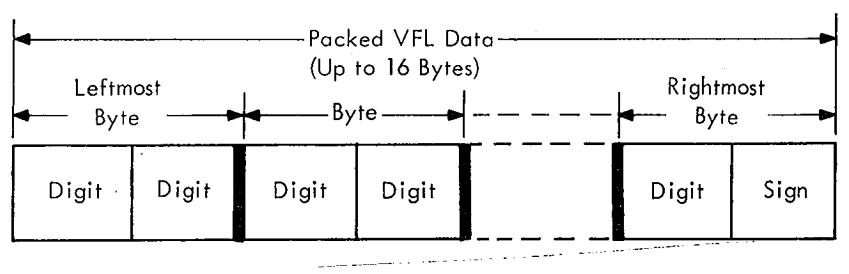


   Long Operand

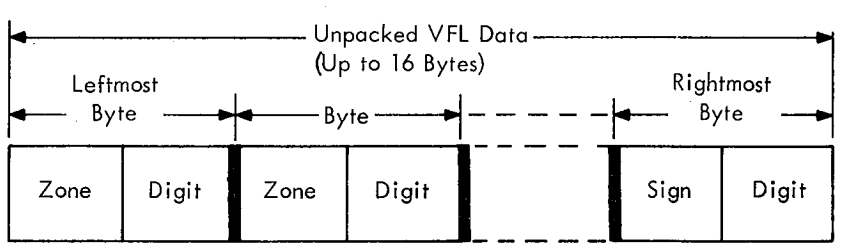


3. Decimal numbers, represented by four-bit binary-coded-decimal (BCD) digits:

   Packed:

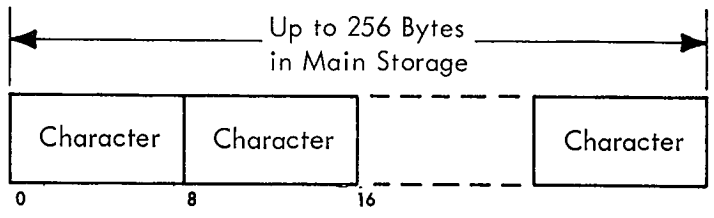


   Unpacked (Zoned)

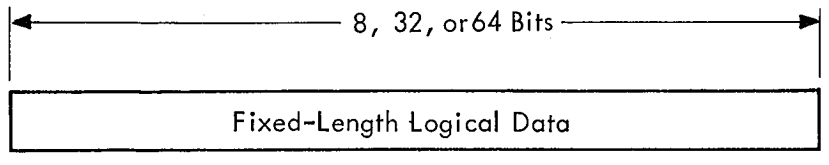


4. Logical information, represented by eight-bit zoned character codes:

   VFL Format:



   Fixed-Length Format



The main storage integral boundaries for data and instructions are shown in Figure 1-6.

Bit Positions 4,3,2,1	Second Hexadecimal Digit	00				01				10				11				Bit Positions 8,7
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 6,5
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit
0000	0	NUL	DLE			SP	0					@	P			`	p	
0001	1	SOH	DC1			!①	1					A	Q			a	q	
0010	2	STX	DC2			"	2					B	R			b	r	
0011	3	ETX	DC3			#	3					C	S			c	s	
0100	4	EOT	DC4			$	4					D	T			d	t	
0101	5	ENQ	NAK			%	5					E	U			e	u	
0110	6	ACK	SYN			&	6					F	V			f	v	
0111	7	BEL	ETB			'	7					G	W			g	w	
1000	8	BS	CAN			(	8					H	X			h	x	
1001	9	HT	EM			)	9					I	Y			i	y	
1010	A	LF	SUB			*	:					J	Z			j	z	
1011	B	VT	ESC			+	;					K	[			k	{	
1100	C	FF	FS			,	<					L	\			l	¦	
1101	D	CR	GS			-	=					M	]			m	}	
1110	E	SO	RS			.	>					N	^②			n	~	
1111	F	SI	US			/	?					O	—			o	DEL	

① If IBM equipment implementing USASCII-8 is provided, the graphic | (Logical OR) will be used instead of ! (Exclamation Point).

② If IBM equipment implementing USASCII-8 is provided, the graphic ¬ (Logical NOT) will be used instead of ^ (Circumflex).

NOTE: Current activities in committees under the auspices of the United States of America Standards Institute may result in changes to the characters and/or structure of the eight-bit representation of USASCII devised by the Institute. Such changes may cause the eight-bit representation of USASCII implemented in System/360 (USASCII-8) to be different from a future USA Standard. Since a difference of this nature may eventually lead to a modification of System/360, it is recommended that users avoid: (1) operation with PSW bit 12 set to 1, and (2) the use of any sign codes in decimal data other than those preferred for EBCDIC.

Control Character Representations

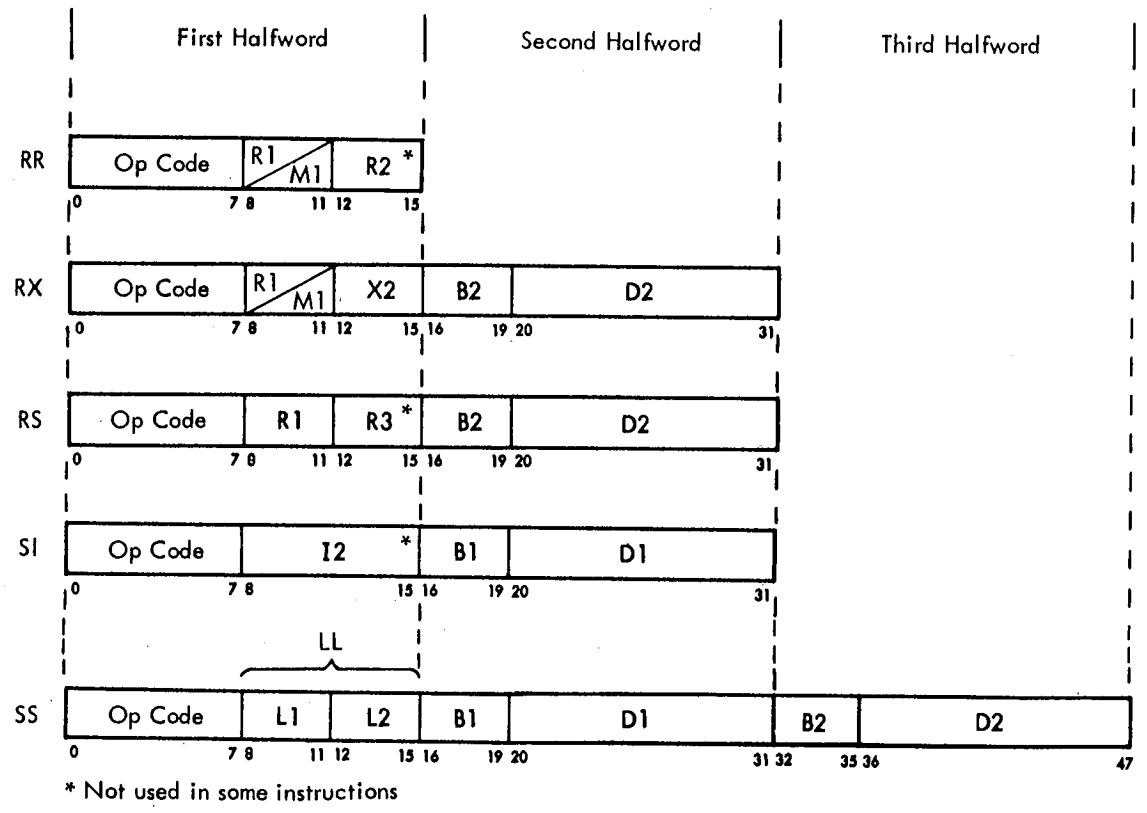
NUL	Null
SOH	Start of Heading (CC)
STX	Start of Text (CC)
ETX	End of Text (CC)
EOT	End of Transmission (CC)
ENQ	Enquiry (CC)
ACK	Acknowledge (CC)
BEL	Bell
BS	Backspace (FE)
HT	Horizontal Tabulation (FE)
LF	Line Feed (FE)
VT	Vertical Tabulation (FE)
FF	Form Feed (FE)
CR	Carriage Return (FE)
SO	Shift Out
SI	Shift In
DLE	Data Link Escape (CC)
DC1	Device Control 1
DC2	Device Control 2
DC3	Device Control 3
DC4	Device Control 4
NAK	Negative Acknowledge (CC)
SYN	Synchronous Idle (CC)
ETB	End of Transmission Block (CC)
CAN	Cancel
EM	End of Medium
SUB	Substitute
ESC	Escape
FS	File Separator (IS)
GS	Group Separator (IS)
RS	Record Separator (IS)
US	Unit Separator (IS)
DEL	Delete

(CC) Communication Control
(FE) Format Effector
(IS) Information Separator

Special Graphic Characters

SP	Space	<	Less Than
!	Exclamation Point	=	Equals
|	Logical OR	>	Greater Than
"	Quotation Marks	?	Question Mark
#	Number Sign	@	Commercial At
$	Dollar Sign	[	Opening Bracket
%	Percent	\	Reverse Slant
&	Ampersand	]	Closing Bracket
'	Apostrophe	^	Circumflex
(	Opening Parenthesis	¬	Logical NOT
)	Closing Parenthesis		Underline
*	Asterisk	`	Grave Accent
+	Plus	{	Opening Brace
,	Comma	¦	Vertical Line (This graphic is stylized to distinguish it from Logical OR)
-	Hyphen (Minus)		
.	Period (Decimal Point)		
/	Slant	}	Closing Brace
:	Colon	~	Tilde
;	Semicolon		

Figure 1-4. USASCII-8 Code



* Not used in some instructions

Legend:

R1, R2, and R3: 4-bit address of an LS register containing the first, second, and third operands, respectively.

M1: 4-bit mask used in some branching instructions.

X2: 4-bit address of an LS register containing the index value used in generating the effective second operand address.

I2: 8-bit byte of immediate data (second operand).

L1 and L2: 4-bit length (up to 16 bytes) of first and second decimal VFL operands, respectively.

LL: 8-bit length field (up to 256 bytes) for logical VFL operands.

B1, B2: 4-bit address of a base register.

D1, D2: 12-bit displacement.

B1+D1 or B2+D2 = Effective storage operand address.

Figure 1-5. Instruction Formats

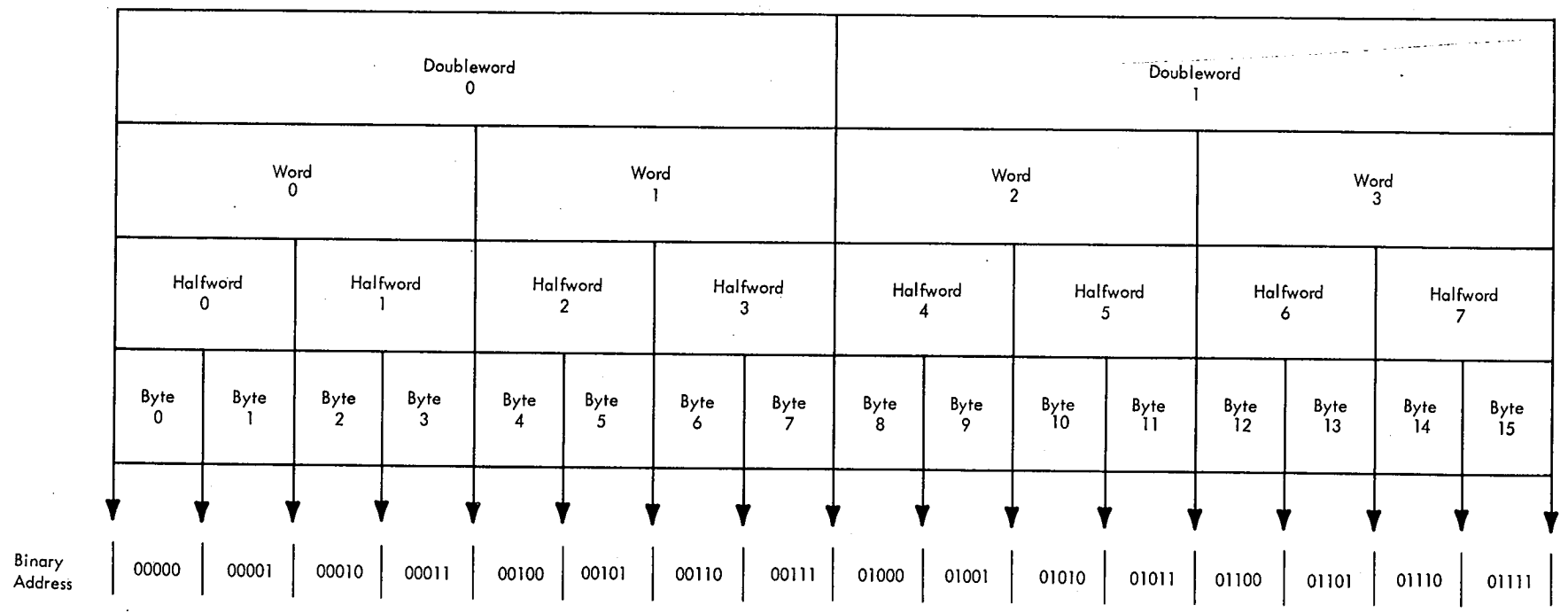


Figure 1-6. Main Storage Integral Boundaries

Table 1-1. Fixed-Point Instructions

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Add	A	5A	RX	R1 D2(X2, B2)	Algebraically add 2nd opr (in stg) to 1st opr (in GPR per R1) & place result into 1st opr location. D(21) determines which word of doubleword from stg is 2nd opr: if 1, right word; if 0, left word.	Prot (F) Adr Spec Fix-Pt Ovflo	0 : Sum = 0 1 : Sum < 0 2 : Sum > 0 3 : Overflow
Add	AR	1A	RR	R1 R2	Algebraically add 2nd opr (in GPR per R2) to 1st opr (in GPR per R1) & place result into 1st opr location.	Fix-Pt Ovflo	0 : Sum = 0 1 : Sum < 0 2 : Sum > 0 3 : Overflow
Add Halfword	AH	4A	RX	R1 D2(X2, B2)	Algebraically add halfword 2nd opr (in stg) to 1st opr (in GPR per R1) & place result into 1st opr location. 1. D(21) determines which word of doubleword from stg contains halfword 2nd opr: If 1, right word; if 0, left word. 2. D(22) determines which half of word is halfword 2nd opr: If 1, right half; if 0, left half. 3. Halfword 2nd opr is expanded to full word before addition by propagating sign bit through 16 high-order bits.	Prot (F) Adr Spec Fix-Pt Ovflo	0 : Sum = 0 1 : Sum < 0 2 : Sum > 0 3 : Overflow
Add Logical	AL	5E	RX	R1 D2(X2, B2)	Algebraically add 2nd opr (in stg) to 1st opr (in GPR per R1) & place result into 1st opr location. 1. D(21) determines which word of doubleword from stg is 2nd opr: if 1, right word; if 0, left word. 2. Sign bit of result is treated as high-order integer & is tested for carry to determine CC.	Prot (F) Adr Spec	0 : Sum = 0 (no carry) 1 : Sum ≠ 0 (no carry) 2 : Sum = 0 (carry) 3 : Sum ≠ 0 (carry)
Add Logical	ALR	1E	RR	R1 R2	Algebraically add 2nd opr (in GPR per R2) to 1st opr (in GPR per R1) & place result into 1st opr location. Sign bit of result is treated as high-order integer & is tested for carry to determine CC.	None	0 : Sum = 0 (no carry) 1 : Sum ≠ 0 (no carry) 2 : Sum = 0 (carry) 3 : Sum ≠ 0 (carry)
Compare	C	59	RX	R1 D2(X2, B2)	Algebraically compare 1st opr (in GPR per R1) with 2nd opr (in stg) & set CC according to result. D(21) determines which word of doubleword from stg is 2nd opr: if 1, right word; if 0, left word.	Prot (F) Adr Spec	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Compare	CR	19	RR	R1 R2	Algebraically compare 1st opr (in GPR per R1) with 2nd opr (in GPR per R2) & set CC according to result.	None	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Compare Halfword	CH	49	RX	R1 D2(X2, B2)	Algebraically compare 1st opr (in GPR per R1) with halfword 2nd opr (in stg) & set CC according to result. 1. D(21) determines which word of doubleword from stg contains halfword 2nd opr: if 1, right word; if 0, left word. 2. D(22) determines which half of word is halfword 2nd opr: if 1, right half; if 0, left half. 3. Halfword 2nd opr is expanded to full word before comparison by propagating sign bit through 16 high-order bits.	Prot (F) Adr Spec	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Convert to Binary	CVB	4F	RX	R1 D2(X2, B2)	Convert radix of 2nd opr (in stg) from decimal to binary & place result into 1st opr location (in GPR per R1). 1. 2nd opr is doubleword in packed format. 2. High-order word is converted first. 3. Max positive integer that can be converted is +2,147,483,647. 4. Max negative integer that can be converted is -2,147,483,648.	Prot (F) Adr Spec Data Fix-Pt Div	Unchanged

Table 1-1. Fixed-Point Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Convert to Decimal	CVD	4E	RX	R1 D2(X2, B2)	Convert radix of 1st opr (in GPR per R1) from binary to decimal & place result into 2nd opr location (in stg). 1. Result is in packed format on doubleword boundary. 2. Low-order 4 bits of field are sign. 3. If PSW(12) = 1, use USASCII-8 code for sign; if PSW(12) = 0, use EBCDIC code.	Prot (S) Adr Spec	Unchanged
Divide	D	5D	RX	R1 D2(X2, B2)	Divide 1st opr (in GPR per R1 & R1 + 1) by 2nd opr (in stg) & place result into 1st opr location (remainder in GPR per R1; quotient in GPR per R1 + 1). 1. R1 must be even adr. 2. D(21) determines which word of doubleword from stg is divisor: if 1, right word; if 0, left word. 3. Relative value of opr's must result in quotient expressible in 32-bit signed integer. 4. Sign of quotient is determined algebraically, except 0 quotient is positive. 5. Sign of remainder is same as sign of dividend, except 0 remainder is positive.	Prot (F) Adr Spec Fix-Pt Div	Unchanged
Divide	DR	1D	RR	R1 R2	Divide 1st opr (in GPR per R1 & R1 + 1) by 2nd opr (in GPR per R2) & place result into 1st opr location (remainder in GPR per R1; quotient in GPR per R1 + 1). 1. R1 must be even adr. 2. Relative value of opr's must result in quotient expressible in 32-bit signed integer. 3. Sign of quotient is determined algebraically, except 0 quotient is positive. 4. Sign of remainder is same as sign of dividend, except 0 remainder is positive.	Spec Fix-Pt Div	Unchanged
Load	L	58	RX	R1 D2(X2, B2)	Load 2nd opr (in stg) into 1st opr location (in GPR per R1). 1. D(21) determines which word of doubleword from stg is to be stored: if 1, right word; if 0, left word. 2. 2nd opr is unchanged.	Prot (F) Adr Spec	Unchanged
Load	LR	18	RR	R1 R2	Load 2nd opr (in GPR per R2) into 1st opr location (in GPR per R1). 2nd opr is unchanged.	None	Unchanged
Load & Test	LTR	12	RR	R1 R2	Load 2nd opr (in GPR per R2) into 1st opr location (in GPR per R1) & set CC according to result. 2nd opr is unchanged.	None	0 : Result = 0 1 : Result < 0 2 : Result > 0
Load Complement	LCR	13	RR	R1 R2	Load 2's complement of 2nd opr (in GPR per R2) into 1st opr location (in GPR per R1) & set CC according to result. Overflow occurs only if max negative number is 2's complemented.	Fix-Pt Ovflo	0 : Result = 0 1 : Result < 0 2 : Result > 0 3 : Overflow
Load Halfword	LH	48	RX	R1 D2(X2, B2)	Load halfword 2nd opr (in stg) into 1st opr location (in GPR per R1). 1. D(21) determines which word of doubleword from stg contains halfword 2nd opr: if 1, right word; if 0, left word. 2. D(22) determines which half of word is halfword 2nd opr: if 1, right half; if 0, left half. 3. Halfword 2nd opr is expanded to full word before loading by propagating sign bit through 16 high-order bits.	Prot (F) Adr Spec	Unchanged

Table 1-1. Fixed-Point Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Load Multiple	LM	98	RS	R1 R3 D2(B2)	Load 2nd opr (as many words as required; in stg) into GPR's, in ascending order, starting with 1st opr location (per R1) & ending with 3rd opr location (per R3). 1. 2nd opr is unchanged. 2. If R1 = R3, only 1 word is loaded. 3. If R3 < R1, GPR adr's wraparound from 15 to 0. 4. D(21) determines which word of doubleword from stg is to be loaded into LS: if 1, right word; if 0, left word.	Prot (F) Adr Spec	Unchanged
Load Negative	LNR	11	RR	R1 R2	Load 2nd opr (unchanged if negative, 2's complemented if positive; in GPR per R2) into 1st opr location (in GPR per R1). If 2nd opr = 0, unchanged with plus sign.	None	0 : Result = 0 1 : Result < 0
Load Positive	LPR	10	RR	R1 R2	Load 2nd opr (unchanged if positive, 2's complemented if negative; in GPR per R2) into 1st opr location (in GPR per R1). Overflow occurs only if max negative number is 2's complemented.	Fix-Pt Ovflo	0 : Result = 0 2 : Result > 0 3 : Overflow
Multiply	M	5C	RX	R1 D2(X2, B2)	Multiply 1st opr (in GPR per R1 + 1) & 2nd opr (in stg) & place 64-bit result into 1st opr location (in GPR per R1 & R1 + 1). 1. R1 must be even adr. 2. D(21) determines which word of doubleword from stg is 2nd opr: if 1, right word; if 0, left word.	Prot (F) Adr Spec	Unchanged
Multiply	MR	1C	RR	R1 R2	Multiply 1st opr (in GPR per R1 + 1) by 2nd opr (in GPR per R2) & place 64-bit result into 1st opr location (in GPR per R1 & R1 + 1). R1 must be even adr.	Spec	Unchanged
Multiply Halfword	MH	4C	RX	R1 D2(X2, B2)	Multiply 1st opr (in GPR per R1) & halfword 2nd opr (in stg) & place low-order 32 bits of result into 1st opr location. 1. D(21) determines which word of doubleword from stg contains halfword 2nd opr: if 1, right word; if 0, left word. 2. D(22) determines which half of word is halfword 2nd opr: if 1, right half; if 0, left half. 3. Halfword 2nd opr is expanded to full word before multiplication by propagating sign bit through 16 high-order bits.	Prot (F) Adr Spec	Unchanged
Shift Left Double	SLDA	8F	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1 & R1 + 1) left number of bit positions specified by low-order 6 bits of 2nd opr adr & place result into 1st opr location. 1. R1 must be even adr. 2. High-order bits of 1st opr are shifted out & lost; low-order vacated bits are made 0's. 3. If bit unlike sign bit is shifted out of bit position 1 of even register, fixed-point overflow occurs.	Spec Fix-Pt Ovflo	0 : Result = 0 1 : Result < 0 2 : Result > 0 3 : Overflow
Shift Left Single	SLA	8B	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1) left number of bit positions specified by low-order 6 bits of 2nd opr adr & place result into 1st opr location. 1. High-order bits of 1st opr are shifted out & lost; low-order vacated bits are made 0's. 2. If bit unlike sign bit is shifted out of bit position 1 of even register, fixed-point overflow occurs.	Fix-Pt Ovflo	0 : Result = 0 1 : Result < 0 2 : Result > 0 3 : Overflow
Shift Right Double	SRDA	8E	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1 & R1 + 1) right number of bit positions specified by low-order 6 bits of 2nd opr adr & place result into 1st opr location. 1. R1 must be even adr. 2. Low-order bits of 1st opr are shifted out & lost; high-order vacated bits are made equal to sign bit.	Spec	0 : Result = 0 1 : Result < 0 2 : Result > 0

Table 1-1. Fixed-Point Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Shift Right Single	SRA	8A	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1) right number of bit positions specified by low-order 6 bits of 2nd opr adr & place result into 1st opr location. Low-order bits of 1st opr are shifted out & lost; high-order vacated bits are made equal to sign bit.	None	0 : Result = 0 1 : Result < 0 2 : Result > 0
Store	ST	50	RX	R1 D2(X2, B2)	Store 1st opr (in GPR per R1) into 2nd opr location (in stg). 1. PAL(61) determines into which word of doubleword in stg 1st opr is to be stored: if 1, right word; if 0, left word. 2. 1st opr is unchanged.	Prot (S) Adr Spec	Unchanged
Store Halfword	STH	40	RX	R1 D2(X2, B2)	Store halfword 1st opr (in GPR per R1) into 2nd opr location (in stg). 1. ABC selects 16 low-order bits of 1st opr for storage; high-order bits are ignored. 2. STC [D(21–23)] positions 16 low-order bits of 1st opr into doubleword 2nd opr location. 3. 1st opr is unchanged.	Prot (S) Adr Spec	Unchanged
Store Multiple	STM	90	RS	R1 R3 D2(B2)	Store into 2nd opr location (as many words as required; in stg) contents of GPR's, in ascending order, starting with 1st opr location (per R1) & ending with 3rd opr location (per R3). 1. GPR adr's wrap around from 15 to 0. 2. D(21) determines into which word of doubleword in stg contents of 1st GPR are to be stored: if 1, right word; if 0, left word. 3. If R1 = R3, 1 word is stored.	Prot (S) Adr Spec	Unchanged
Subtract	S	5B	RX	R1 D2(X2, B2)	Algebraically subtract 2nd opr (in stg) from 1st opr (in GPR per R1) & place result into 1st opr location. D(21) determines which word of doubleword from stg is 2nd opr: if 1, right word; if 0, left word.	Prot (F) Adr Spec Fix-Pt Ovflo	0 : Dif = 0 1 : Dif < 0 2 : Dif > 0 3 : Overflow
Subtract	SR	1B	RR	R1 R2	Algebraically subtract 2nd opr (in GPR per R2) from 1st opr (in GPR per R1) & place result into 1st opr location.	Fix-Pt Ovflo	0 : Dif = 0 1 : Dif < 0 2 : Dif > 0 3 : Overflow
Subtract Halfword	SH	4B	RX	R1 D2(X2, B2)	Algebraically subtract halfword 2nd opr (in stg) from 1st opr (in GPR per R1) & place result into 1st opr location. 1. D(21) determines which word of doubleword from stg contains halfword 2nd opr: if 1, right word; if 0, left word. 2. D(22) determines which half of word is halfword 2nd opr: if 1, right half; if 0, left half. 3. Halfword 2nd opr is expanded to full word before subtraction by propagating sign bit through 16 high-order bits.	Prot (F) Adr Spec Fix-Pt Ovflo	0 : Dif = 0 1 : Dif < 0 2 : Dif > 0 3 : Overflow
Subtract Logical	SL	5F	RX	R1 D2(X2, B2)	Algebraically subtract 2nd opr (in stg) from 1st opr (in GPR per R1) & place result into 1st opr location. 1. D(21) determines which word of doubleword from stg is 2nd opr: if 1, right word; if 0, left word. 2. Sign bit of result is treated as high-order integer & is tested for carry to determine CC.	Prot (F) Adr Spec	1 : Dif ≠ 0 (no carry) 2 : Dif = 0 (carry) 3 : Dif ≠ 0 (carry)
Subtract Logical	SLR	1F	RR	R1 R2	Algebraically subtract 2nd opr (in GPR per R2) from 1st opr (in GPR per R1) & place result into 1st opr location. Sign bit of result is treated as high-order integer & is tested for carry to determine CC.	None	1 : Dif ≠ 0 (no carry) 2 : Dif = 0 (carry) 3 : Dif ≠ (carry)

Table 1-2. Floating-Point Instructions

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Add Normalized (long)	AD	6A	RX	R1 D2(X2, B2)	Algebraically add 2nd opr (in stg) to 1st opr (in FPR per R1 & R1 + 1) & place normalized result into 1st opr location. 1. Low-order fraction of 1st opr must be fetched from LS. 2. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Add Normalized (long)	ADR	2A	RR	R1 R2	Algebraically add 2nd opr (in FPR per R2 & R2 + 1) to 1st opr (in FPR per R1 & R1 + 1) & place normalized result into 1st opr location. 1. Low-order fractions of 1st & 2nd opr's must be fetched from LS. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Add Normalized (short)	AE	7A	RX	R1 D2(X2, B2)	Algebraically add 2nd opr (in stg) to 1st opr (in FPR per R1) & place normalized result into 1st opr location. 1. Low-order half of FPR is ignored & unchanged. 2. D(21) determines which half of doubleword from stg is 2nd opr; if 1, right half; if 0, left half. 3. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Add Normalized (short)	AER	3A	RR	R1 R2	Algebraically add 2nd opr (in FPR per R2) to 1st opr (in FPR per R1) & place normalized result into 1st opr location. 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Add Unnormalized (long)	AW	6E	RX	R1 D2(X2, B2)	Algebraically add 2nd opr (in stg) to 1st opr (in FPR per R1 & R1 + 1) & place unnormalized result into 1st opr location. 1. Low-order fraction of 1st opr must be fetched from LS. 2. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Add Unnormalized (long)	AWR	2E	RR	R1 R2	Algebraically add 2nd opr (in FPR per R2 & R2 + 1) to 1st opr (in FPR per R1 & R1 + 1) & place unnormalized result into 1st opr location. 1. Low-order fractions of 1st & 2nd opr's must be fetched from LS. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Add Unnormalized (short)	AU	7E	RX	R1 D2(X2, B2)	Algebraically add 2nd opr (in stg) to 1st opr (in FPR per R1) & place unnormalized result into 1st opr location. 1. Low-order half of FPR is ignored & unchanged. 2. D(21) determines which half of doubleword from stg is 2nd opr: if 1, right half; if 0, left half. 3. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Add Unnormalized (short)	AUR	3E	RR	R1 R2	Algebraically add 2nd opr (in FPR per R2) to 1st opr (in FPR per R1) & place unnormalized result into 1st opr location. 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Compare (long)	CD	69	RX	R1 D2(X2, B2)	Algebraically compare 1st opr (in FPR per R1 & R1 + 1) with 2nd opr (in stg); CC indicates result. 1. Low-order fraction of 1st opr must be fetched from LS. 2. Opr's remain unchanged.	Prot (F) Adr Spec	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Compare (long)	CDR	29	RR	R1 R2	Algebraically compare 1st opr (in FPR per R1 & R1 + 1) with 2nd opr (in FPR per R2 & R2 + 1); CC indicates result. 1. Low-order fractions of 1st & 2nd opr's must be fetched from LS. 2. Opr's remain unchanged.	Spec	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2

Table 1-2. Floating-Point Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Compare (short)	CE	79	RX	R1 D2(X2, B2)	Algebraically compare 1st opr (in FPR per R1) with 2nd opr (in stg); CC indicates result. 1. Low-order half of FPR is ignored. 2. D(21) determines which half of doubleword from stg is 2nd opr: if 1, right half; if 0, left half. 3. Opr's remain unchanged.	Prot (F) Adr Spec	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Compare (short)	CER	39	RR	R1 R2	Algebraically compare 1st opr (in FPR per R1) with 2nd opr (in FPR per R2); CC indicates result. 1. Low-order halves of FPR's are ignored. 2. Opr's remain unchanged.	Spec	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Divide (long)	DD	6D	RX	R1 D2(X2, B2)	Divide 1st opr (in FPR per R1 & R1 + 1) by 2nd opr (in stg) & place normalized quotient into 1st opr location. 1. Low-order fraction of 1st opr must be fetched from LS. 2. Opr's are prenormalized. 3. Remainder is not saved.	Prot (F) Adr Spec Exp Ovflo Exp Unflo Flt-Pt Div	Unchanged
Divide (long)	DDR	2D	RR	R1 R2	Divide 1st opr (in FPR per R1 & R1 + 1) by 2nd opr (in FPR per R2 & R2 + 1) & place normalized quotient into 1st opr location. 1. Low-order fractions of 1st & 2nd opr's must be fetched from LS. 2. Opr's are prenormalized. 3. Remainder is not saved.	Spec Exp Ovflo Exp Unflo Flt-Pt Div	Unchanged
Divide (short)	DE	7D	RX	R1 D2(X2, B2)	Divide 1st opr (in FPR per R1) by 2nd opr (in stg) & place normalized quotient into 1st opr location. 1. Low-order half of FPR is ignored & unchanged. 2. D(21) determines which half of doubleword from stg is 2nd opr: if 1, right half; if 0, left half. 3. Opr's are prenormalized. 4. Remainder is not saved.	Prot (F) Adr Spec Exp Ovflo Exp Unflo Flt-Pt Div	Unchanged
Divide (short)	DER	3D	RR	R1 R2	Divide 1st opr (in FPR per R1) by 2nd opr (in FPR per R2) & place normalized quotient into 1st opr location. 1. Low-order halves of FPR's are ignored & unchanged. 2. Opr's are prenormalized. 3. Remainder is not saved.	Spec Exp Ovflo Exp Unflo Flt-Pt Div	Unchanged
Halve (long)	HDR	24	RR	R1 R2	Divide 2nd opr (in FPR per R2 & R2 + 1) by 2 & place normalized quotient into 1st opr location (in FPR per R1 & R1 + 1). Low-order fraction of 2nd opr must be fetched from LS.	Spec Exp Unflo	Unchanged
Halve (short)	HER	34	RR	R1 R2	Divide 2nd opr (in FPR per R2) by 2 & place normalized quotient into 1st opr location (in FPR per R1). Low-order halves of FPR's are ignored & unchanged.	Spec Exp Unflo	Unchanged
Load (long)	LD	68	RX	R1 D2(X2, B2)	Load 2nd opr (in stg) into 1st opr location (in FPR per R1 & R1 + 1).	Prot (F) Adr Spec	Unchanged
Load (long)	LDR	28	RR	R1 R2	Load 2nd opr (in FPR per R2 & R2 + 1) into 1st opr location (in FPR per R1 & R1 + 1). Low-order fraction of 2nd opr must be fetched from LS.	Spec	Unchanged
Load (short)	LE	78	RX	R1 D2(X2, B2)	Load 2nd opr (in stg) into 1st opr location (in FPR per R1). 1. D(21) determines which half of doubleword from stg is 2nd opr: if 1, right half; if 0, left half. 2. Low-order half of FPR is ignored & unchanged.	Prot (F) Adr Spec	Unchanged

Table 1-2. Floating-Point Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Load (short)	LER	38	RR	R1 R2	Load 2nd opr (in FPR per R2) into 1st opr location (in FPR per R1). Low-order halves of FPR's are ignored & unchanged.	Spec	Unchanged
Load & Test (long)	LTDR	22	RR	R1 R2	Load 2nd opr (in FPR per R2 & R2 + 1) into 1st opr location (in FPR per R1 & R1 + 1). 1. Low-order fraction of 2nd opr must be fetched from LS. 2. Set CC according to sign & magnitude.	Spec	0 : 2nd opr fract = 0 1 : 2nd opr < 0 2 : 2nd opr > 0
Load & Test (short)	LTER	32	RR	R1 R2	Load 2nd opr (in FPR per R2) into 1st opr location (in FPR per R1). 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC according to sign & magnitude.	Spec	0 : 2nd opr fract = 0 1 : 2nd opr < 0 2 : 2nd opr > 0
Load Complement (long)	LCDR	23	RR	R1 R2	Load 2nd opr (in FPR per R2 & R2 + 1) into 1st opr location (in FPR per R1 & R1 + 1) with sign complemented. 1. Low-order fraction of 2nd opr must be fetched from LS. 2. Set CC according to original sign & magnitude.	Spec	0 : 2nd opr fract = 0 1 : Orig sign + 2 : Orig sign -
Load Complement (short)	LCER	33	RR	R1 R2	Load 2nd opr (in FPR per R2) into 1st opr location (in FPR per R1) with sign complemented. 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC according to original sign & magnitude.	Spec	0 : 2nd opr fract = 0 1 : Orig sign + 2 : Orig sign -
Load Negative (long)	LNDR	21	RR	R1 R2	Load 2nd opr (in FPR per R2 & R2 + 1) into 1st opr location (in FPR per R1 & R1 + 1) with sign made minus. 1. Low-order fraction of 2nd opr must be fetched from LS. 2. Set CC according to result sign & magnitude.	Spec	0 : 2nd opr fract = 0 1 : 2nd opr < 0
Load Negative (short)	LNER	31	RR	R1 R2	Load 2nd opr (in FPR per R2) into 1st opr location (in FPR per R1) with sign made minus. 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC according to result sign & magnitude.	Spec	0 : 2nd opr fract = 0 1 : 2nd opr < 0
Load Positive (long)	LPDR	20	RR	R1 R2	Load 2nd opr (in FPR per R2 & R2 + 1) into 1st opr location (in FPR per R1 & R1 + 1) with sign made plus. 1. Low-order fraction of 2nd opr must be fetched from LS. 2. Set CC according to result sign & magnitude.	Spec	0 : 2nd opr fract = 0 2 : 2nd opr > 0
Load Positive (short)	LPER	30	RR	R1 R2	Load 2nd opr (in FPR per R2) into 1st opr location (in FPR per R1) with sign made plus. 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC according to result sign & magnitude.	Spec	0 : 2nd opr fract = 0 2 : 2nd opr > 0
Multiply (long)	MD	6C	RX	R1 D2(X2, B2)	Multiply 1st opr (in FPR per R1 & R1 + 1) & 2nd opr (in stg) & place normalized product into 1st opr location (in FPR per R1 & R1 + 1). Opr's are prenormalized.	Prot (F) Adr Spec Exp Ovflo Exp Unflo	Unchanged
Multiply (long)	MDR	2C	RR	R1 R2	Multiply 1st opr (in FPR per R1 & R1 + 1) & 2nd opr (in FPR per R2 & R2 + 1) & place normalized product into 1st opr location (in FPR per R1 & R1 + 1). Opr's are prenormalized.	Spec Exp Ovflo Exp Unflo	Unchanged

Table 1-2. Floating-Point Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Multiply (short)	ME	7C	RX	R1 D2(X2, B2)	Multiply 1st opr (in FPR per R1) & 2nd opr (in stg) & place normalized product into 1st opr location (in FPR per R1 & R1 + 1). 1. D(21) determines which half of doubleword from stg is 2nd opr: if 1, right half; if 0, left half. 2. Opr's are prenormalized.	Prot (F) Adr Spec Exp Ovflo Exp Unflo	Unchanged
Multiply (short)	MER	3C	RR	R1 R2	Multiply 1st opr (in FPR per R1) & 2nd opr (in FPR per R2) & place normalized product into 1st opr location (in FPR per R1 & R1 + 1). Opr's are prenormalized.	Spec Exp Ovflo Exp Unflo	Unchanged
Store (long)	STD	60	RX	R1 D2(X2, B2)	Store 1st opr (in FPR per R1 & R1 + 1) into 2nd opr location (in stg). 1st opr is unchanged.	Prot (S) Adr Spec	Unchanged
Store (short)	STE	70	RX	R1 D2(X2, B2)	Store 1st opr (in FPR per R1) into 2nd opr location (in stg). 1. PAL(61) determines into which half of doubleword in stg 1st opr is to be stored: if 1, right half; if 0, left half. 2. Low-order half of FPR is ignored. 3. 1st opr is unchanged.	Prot (S) Adr Spec	Unchanged
Subtract Normalized (long)	SD	6B	RX	R1 D2(X2, B2)	Algebraically subtract 2nd opr (in stg) from 1st opr (in FPR per R1 & R1 + 1) & place normalized result into 1st opr location. 1. Low-order fraction of 1st opr must be fetched from LS. 2. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Subtract Normalized (long)	SDR	2B	RR	R1 R2	Algebraically subtract 2nd opr (in FPR per R2 & R2 + 1) from 1st opr (in FPR per R1 & R1 + 1) & place normalized result into 1st opr location. 1. Low-order fractions of 1st & 2nd opr's must be fetched from LS. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Subtract Normalized (short)	SE	7B	RX	R1 D2(X2, B2)	Algebraically subtract 2nd opr (in stg) from 1st opr (in FPR per R1) & place normalized result into 1st opr location. 1. Low-order half of FPR is ignored & unchanged. 2. D(21) determines which half of doubleword from stg is 2nd opr: if 1, right half; if 0, left half. 3. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Subtract Normalized (short)	SER	3B	RR	R1 R2	Algebraically subtract 2nd opr (in FPR per R2) from 1st opr (in FPR per R1) & place normalized result into 1st opr location. 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Exp Unflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Subtract Unnormalized (long)	SW	6F	RX	R1 D2(X2, B2)	Algebraically subtract 2nd opr (in stg) from 1st opr (in FPR per R1 & R1 + 1) & place unnormalized result into 1st opr location. 1. Low-order fraction of 1st opr must be fetched from LS. 2. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Subtract Unnormalized (long)	SWR	2F	RR	R1 R2	Algebraically subtract 2nd opr (in FPR per R2 & R2 + 1) from 1st opr (in FPR per R1 & R1 + 1) & place unnormalized result into 1st opr location. 1. Low-order fractions of 1st & 2nd opr's must be fetched from LS. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0

Table 1-2. Floating-Point Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Subtract Unnormalized (short)	SU	7F	RX	R1 D2(X2, B2)	Algebraically subtract 2nd opr (in stg) from 1st opr (in FPR per R1) & place unnormalized result into 1st opr location. 1. Low-order half of FPR is ignored & unchanged. 2. D(21) determines which half of doubleword from stg is 2nd opr: if 1, right half; if 0, left half. 3. Set CC per result sign & magnitude.	Prot (F) Adr Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0
Subtract Unnormalized (short)	SUR	3F	RR	R1 R2	Algebraically subtract 2nd opr (in FPR per R2) from 1st opr (in FPR per R1) & place unnormalized result into 1st opr location. 1. Low-order halves of FPR's are ignored & unchanged. 2. Set CC per result sign & magnitude.	Spec Exp Ovflo Signif	0 : Fract = 0 1 : Fract < 0 2 : Fract > 0

Table 1-3. Decimal Instructions

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Add Decimal	AP	FA	SS	D1(L1, B1) D2(L2, B2)	Algebraically add 2nd opr (in stg) to 1st opr (in stg) & place result into 1st opr location. 1. Opr's & result are in packed format. 2. Opr fields may overlap if low-order bytes coincide. 3. Right to left, byte by byte. 4. Shorter opr is extended with high-order 0's. 5. 1st opr field must be large enough to contain all 2nd opr significant digits.	Prot (S,F) Adr Data Dec Ovflo	0 : Sum = 0 1 : Sum < 0 2 : Sum > 0 3 : Overflow
Compare Decimal	CP	F9	SS	D1(L1, B1) D2(L2,B2)	Algebraically compare 1st opr (in stg) with 2nd opr (in stg) & set CC according to result. 1. Opr's are in packed format. 2. Shorter opr is extended with high-order 0's. 3. Opr fields may overlap if low-order bytes coincide. 4. Right to left, byte by byte. 5. Result is not stored & opr fields are unchanged.	Prot (F) Adr Data	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Divide Decimal	DP	FD	SS	D1(L1, B1) D2(L2, B2)	Divide 1st opr (in stg) by 2nd opr (in stg) & place result into 1st opr location (quotient is leftmost in 1st opr location; remainder, rightmost). 1. Opr's are in packed format. 2. Dividend must contain at least 1 high-order 0. 3. Max dividend field = 16 bytes (31 digits & sign); L1 = 15. 4. Max divisor field = 8 bytes (15 digits & sign); L2 = 7. 5. Divisor field must be < dividend field (L2 < L1). 6. Max quotient field = 15 bytes. 7. Quotient field = dividend field minus remainder (divisor) field (L1 minus L2). 8. Remainder field = divisor field. 9. Opr fields may overlap if low-order bytes coincide. 10. Sign of quotient is determined algebraically, except 0 result is positive. 11. Sign of remainder is same as dividend sign.	Prot (S,F) Adr Spec Data Dec Div	Unchanged
Move with Offset	MVO	F1	SS	D1(L1, B1) D2(L2, B2)	Store 2nd opr (in stg) to left of and adjacent to low-order 4 bits of 1st opr (in stg). 1. Opr's are in packed or unpacked format. 2. If 2nd opr is shorter than 1st opr, fill 1st opr field with high-order 0's. 3. If 2nd opr is longer than 1st opr, ignore excess 2nd opr high-order digits. 4. Right to left, byte by byte.	Prot (S,F) Adr	Unchanged

Table 1-3. Decimal Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Multiply Decimal	MP	FC	SS	D1(L1, B1) D2(L2, B2)	Multiply 1st opr (in stg) by 2nd opr (in stg) & place result into 1st opr location. 1. Opr's are in packed format. 2. Product must contain at least 1 high-order 0. 3. Max multiplicand field = 16 bytes (31 digits & sign); L1 = 15. 4. Max multiplier field = 8 bytes (15 digits & sign); L2 = 7. 5. Multiplier field must be < multiplicand field (L2 < L1); max value of L2 = 7. 6. Multiplicand field initially contains high-order 0-field equal in length to multiplier field. 7. Max product field = 16 bytes (31 digits & sign). 8. Sign of product is determined algebraically, except 0 result is positive.	Prot (S,F) Adr Spec Data	Unchanged
Pack	PACK	F2	SS	D1(L1, B1) D2(L2, B2)	Convert format of 2nd opr (in stg) from zoned to packed & place result into 1st opr location (in stg). 1. 2nd opr is in zoned format. 2. No restriction on overlapping fields. 3. Extend 2nd opr with high-order 0's, if necessary. 4. If 1st opr field is too short to contain all significant digits of 2nd opr field, ignore excess 2nd opr high-order digits. 5. Right to left, byte by byte.	Prot (S,F) Adr	Unchanged
Subtract Decimal	SP	FB	SS	D1(L1, B1) D2(L2, B2)	Algebraically subtract 2nd opr (in stg) from 1st opr (in stg) & place result into 1st opr location. 1. Opr's & result are in packed format. 2. Opr fields may overlap if low-order bytes coincide. 3. 1st opr field must be large enough to contain all 2nd opr significant digits. 4. Shorter opr is extended with high-order 0's. 5. Right to left, byte by byte.	Prot (S,F) Adr Data Dec Ovflo	0 : Dif = 0 1 : Dif < 0 2 : Dif > 0 3 : Overflow
Unpack	UNPK	F3	SS	D1(L1, B1) D2(L2, B2)	Convert format of 2nd opr (in stg) from packed to zoned & place result into 1st opr location (in stg). 1. 2nd opr is in packed format. 2. No restriction on overlapping fields. 3. Extend 2nd opr with high-order 0's, if necessary. 4. If 1st opr field is too short to contain all significant digits of 2nd opr field, ignore excess 2nd opr high-order digits. 5. If PSW(12) = 1, use USASCII-8 code for zones; if PSW(12) = 0, use EBCDIC. 6. Right to left, byte by byte.	Prot (S,F) Adr	Unchanged
Zero & Add	ZAP	F8	SS	D1(L1, B1) D2(L2, B2)	Place 2nd opr (in stg) into 1st opr location (in stg). 1. 2nd opr is in packed format. 2. Opr fields may overlap if low-order byte of 1st opr coincides with or is to the right of low-order byte of 2nd opr. 3. 1st opr field must be large enough to contain all 2nd opr significant digits.	Prot (S,F) Adr Data Dec Ovflo	0 : Result = 0 1 : Result < 0 2 : Result > 0 3 : Overflow

Table 1-4. Logical Instructions

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
AND	N	54	RX	R1 D2(X2, B2)	AND 1st opr (in GPR per R1) with 2nd opr (in stg) & place result into 1st opr location. Left to right, byte by byte.	Prot (F) Adr Spec	0 : Result = 0 1 : Result ≠ 0
AND	NC	D4	SS	D1(L, B1) D2(B2)	AND 1st opr (in stg) with 2nd opr (in stg) & place result into 1st opr location. 1. Left to right, byte by byte. 2. Max number of bytes is 256.	Prot (S,F) Adr	0 : Result = 0 1 : Result ≠ 0
AND	NI	94	SI	D1(B1) I2	AND immediate opr (I2 of inst) with 1st opr (in stg) & place result into 1st opr location.	Prot (S) Adr	0 : Result = 0 1 : Result ≠ 0
AND	NR	14	RR	R1 R2	AND 1st opr (in GPR per R1) with 2nd opr (in GPR per R2) & place result into 1st opr location. Left to right, byte by byte.	None	0 : Result = 0 1 : Result ≠ 0
Compare Logical	CL	55	RX	R1 D2(X2, B2)	Binarily compare 1st opr (in GPR per R1) with 2nd opr (in stg) & set CC according to result. 1. Left to right, byte by byte. 2. Terminate on inequality or end of fields.	Prot (F) Adr Spec	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Compare Logical	CLC	D5	SS	D1(L, B1) D2(B2)	Binarily compare 1st opr (in stg) with 2nd opr (in stg) & set CC according to result. 1. Left to right, byte by byte. 2. Max number of bytes is 256. 3. Terminate on inequality or end of fields.	Prot (F) Adr	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Compare Logical	CLI	95	SI	D1(B1) I2	Binarily compare 1st opr (in stg) with immediate opr (I2 of inst) & set CC according to result. 1. Left to right. 2. Terminate on inequality or end of fields.	Prot (F) Adr	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Compare Logical	CLR	15	RR	R1 R2	Binarily compare 1st opr (in GPR per R1) with 2nd opr (in GPR per R2) & set CC according to result. 1. Left to right, byte by byte. 2. Terminate on inequality or end of fields.	None	0 : Opr 1 = Opr 2 1 : Opr 1 < Opr 2 2 : Opr 1 > Opr 2
Edit	ED	DE	SS	D1(L, B1) D2(B2)	Change format of source (2nd opr; in stg) from packed to zoned, edit source under control of pattern (1st opr; in stg), & place result into 1st opr location. 1. Left to right, byte by byte. 2. Max number of bytes is 256.	Prot (S,F) Adr Data	0 : Result = 0 1 : Result < 0 2 : Result > 0
Edit & Mark	EDMK	DF	SS	D1(L, B1) D2(B2)	Change format of source (2nd opr; in stg) from packed to zoned, edit source under control of pattern (1st opr; in stg), place result into 1st opr location, & place location of each 1st significant result digit into GPR1. 1. Left to right, byte by byte. 2. Max number of bytes is 256.	Prot (S,F) Adr Data	0 : Result = 0 1 : Result < 0 2 : Result > 0
Exclusive OR	X	57	RX	R1 D2(X2, B2)	Exclusive-OR 1st opr (in GPR per R1) with 2nd opr (in stg) & place result into 1st opr location. Left to right, byte by byte.	Prot (F) Adr Spec	0 : Result = 0 1 : Result ≠ 0
Exclusive OR	XC	D7	SS	D1(L, B1) D2(B2)	Exclusive-OR 1st opr (in stg) with 2nd opr (in stg) & place result into 1st opr location. 1. Left to right, byte by byte. 2. Max number of bytes is 256.	Prot (S,F) Adr	0 : Result = 0 1 : Result ≠ 0
Exclusive OR	XI	97	SI	D1(B1) I2	Exclusive-OR immediate opr (I2 of inst) with 1st opr (in stg) & place result into 1st opr location.	Prot (S) Adr	0 : Result = 0 1 : Result ≠ 0
Exclusive OR	XR	17	RR	R1 R2	Exclusive-OR 1st opr (in GPR per R1) with 2nd opr (in GPR per R2) & place result into 1st opr location. Left to right, byte by byte.	None	0 : Result = 0 1 : Result ≠ 0
Insert Character	IC	43	RX	R1 D2(X2, B2)	Insert 2nd opr (byte; in stg) into bits 24–31 of 1st opr location (in GPR per R1). Remaining bits in GPR are unchanged.	Prot (F) Adr	Unchanged

Table 1-4. Logical Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Load Address	LA	41	RX	R1 D2(X2, B2)	Insert 2nd opr adr into bits 8–31 of GPR specified by R1. 1. Bits 0–7 in GPR are made 0's. 2. 2nd opr is not fetched from stg.	None	Unchanged
Move	MVC	D2	SS	D1(L, B1) D2(B2)	Place 2nd opr (in stg) into 1st opr location (in stg). 1. Left to right, byte by byte. 2. Max number of bytes is 256. 3. Move operation can be high or low speed.	Prot (S,F) Adr	Unchanged
Move	MVI	92	SI	D1(B1) I2	Place immediate opr (I2 of inst) into 1st opr location (in stg).	Prot (S) Adr	Unchanged
Move Numerics	MVN	D1	SS	D1(L, B1) D2(B2)	Place numeric portion (low-order 4 bits) of each byte of 2nd opr (in stg) into low-order 4 bits of corresponding byte of 1st opr (in stg). 1. Left to right, byte by byte. 2. Max number of bytes is 256. 3. Zones (high-order 4 bits) in both opr's are unchanged. 4. No restriction on overlapping fields.	Prot (S,F) Adr	Unchanged
Move Zones	MVZ	D3	SS	D1(L, B1) D2(B2)	Place zone portion (high-order 4 bits) of each byte of 2nd opr (in stg) into high-order 4 bits of corresponding byte of 1st opr (in stg). 1. Left to right, byte by byte. 2. Max number of bytes is 256. 3. Numerics (low-order 4 bits) in both opr's are unchanged. 4. No restriction on overlapping fields.	Prot (S,F) Adr	Unchanged
OR	O	56	RX	R1 D2(X2, B2)	OR 1st opr (in GPR per R1) with 2nd opr (in stg) & place result into 1st opr location. Left to right, byte by byte.	Prot (F) Adr Spec	0 : Result = 0 1 : Result ≠ 0
OR	OC	D6	SS	D1(L, B1) D2(B2)	OR 1st opr (in stg) with 2nd opr (in stg) & place result into 1st opr location. 1. Left to right, byte by byte. 2. Max number of bytes is 256.	Prot (S,F) Adr	0 : Result = 0 1 : Result ≠ 0
OR	OI	96	SI	D1(B1) I2	OR immediate opr (I2 of inst) with 1st opr (in stg) & place result into 1st opr location.	Prot (S) Adr	0 : Result = 0 1 : Result ≠ 0
OR	OR	16	RR	R1 R2	OR 1st opr (in GPR per R1) with 2nd opr (in GPR per R2) & place result into 1st opr location. Left to right, byte by byte.	None	0 : Result = 0 1 : Result ≠ 0
Shift Left Double Logical	SLDL	8D	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1 & R1 + 1) left number of bit positions specified by low-order 6 bits of 2nd opr adr. 1. R1 must be even adr. 2. High-order bits of 1st opr are shifted out & lost; vacated low-order bits are made 0's.	Spec	Unchanged
Shift Left Single Logical	SLL	89	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1) left number of bit positions specified by low-order 6 bits of 2nd opr adr. High-order bits of 1st opr are shifted out & lost; vacated low-order bits are made 0's.	None	Unchanged
Shift Right Double Logical	SRDL	8C	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1 & R1 + 1) right number of bit positions specified by low-order 6 bits of 2nd opr adr. 1. R1 must be even adr. 2. Low-order bits of 1st opr are shifted out & lost; vacated high-order bits are made 0's.	Spec	Unchanged
Shift Right Single Logical	SRL	88	RS	R1 D2(B2)	Shift 1st opr (in GPR per R1) right number of bit positions specified by low-order 6 bits of 2nd opr adr. Low-order bits of 1st opr are shifted out & lost; vacated high-order bits are made 0's.	None	Unchanged

Table 1-4. Logical Instructions (Cont)

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Store Character	STC	42	RX	R1 D2(X2, B2)	Store bits 24–31 of 1st opr (in GPR per R1) into 2nd opr location (in stg).	Prot (S) Adr	Unchanged
Test Under Mask	TM	91	SI	D1(B1) I2	Set CC according to state of 1st opr bits (in stg) selected by mask bits (I2 of inst). 1. If mask bit = 1, test corresponding 1st opr bit; if mask bit = 0, ignore corresponding 1st opr bit. 2. Character in stg is unchanged.	Prot (F) Adr	0 : Selected bits all 0's (mask is all 0's) 1 : Selected bits mixed 0's & 1's 3 : Selected bits all 1's
Translate	TR	DC	SS	D1(L, B1) D2(B2)	Add 1st opr byte (argument; in stg) to effective 2nd opr adr, use result as stg adr, & place function byte from resulting stg adr into corresponding 1st opr byte location. 1. Effective 2nd opr adr = contents of GPR adr by B2, + D2. 2. LL = number of bytes to be translated. 3. 1st opr bytes are processed left to right.	Prot (S,F) Adr	Unchanged
Translate & Test	TRT	DD	SS	D1(L, B1) D2(B2)	Add 1st opr byte (argument; in stg) to effective 2nd opr adr, use result as stg adr, & test function byte from resulting stg adr. If 0, translate & test next argument byte; if non-0, complete operation by inserting related argument adr into GPR1 & function byte into GPR2. 1. Effective 2nd opr adr = contents of GPR adr by B2, + D2. 2. LL = number of bytes to be translated. 3. 1st opr bytes are processed left to right. 4. Set CC according to ending condition.	Prot (F) Adr	0 : All bytes tested are all 0's 1 : Non-0 byte found before last byte to be tested 2 : Non-0 byte found as last byte to be tested

Table 1-5. I/O Instructions

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Halt I/O	HIO	9E	SI	D1(B1)	Terminate current I/O operation at selected channel & I/O unit. 1. D(13–15) is channel adr. 2. D(16–23) is I/O unit adr.	Priv Oper	0 : Interruption in channel 1 : CSW stored 2 : Halted 3 : Unavailable
Start I/O	SIO	9C	SI	D1(B1)	Select specified I/O unit & initiate channel command to that unit. 1. D(13–15) is channel adr. 2. D(16–23) is I/O unit adr. 3. CAW, which specifies address of 1st CCW, is fetched from location 72 (48, hex).	Priv Oper	0 : Available 1 : CSW stored 2 : Working 3 : Unavailable
Test Channel	TCH	9F	SI	D1(B1)	Test state of selected channel & set CC accordingly. 1. D(13–15) is channel adr. 2. D(16–23) is ignored. 3. State of channel is not affected.	Priv Oper	0 : Available 1 : CSW ready 2 : Working 3 : Unavailable
Test I/O	TIO	9D	SI	D1(B1)	Clear interruption condition in addressed channel or associated I/O units, & set CC according to status of addressed channel & I/O units. 1. D(13–15) is channel adr. 2. D(16–23) is I/O unit adr. 3. CSW is stored at location 64 (40, hex) if: a. I/O unit or control unit contains pending interruption. b. I/O unit or control unit is executing previous operation, or there is pending channel-end/control unit-end for another I/O unit. c. I/O unit or its control unit detects machine error.	Priv Oper	0 : Available 1 : CSW stored 2 : Working 3 : Unavailable

Table 1-6. Branching Instructions

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Branch & Link	BAL	45	RX	R1 D2(X2, B2)	Store PSW(32–63), link information, into GPR (adr by R1) & branch to location specified by 2nd opr adr. 1. Branch is unconditional. 2. Link information is stored whether or not branch is successful.	Prot (F)†	Unchanged
Branch & Link	BALR	05	RR	R1 R2	Store PSW(32–63), link information, into GPR (adr by R1) & branch to location specified by GPR (adr by R2). 1. Branch is unsuccessful if R2 = 0; use next sequential instr adr. 2. Link information is stored whether or not branch is successful.	Prot (F)†	Unchanged
Branch on Condition	BC	47	RX	M1 D2(X2, B2)	Branch to location specified by 2nd opr adr if state of CC is as specified by M1. 1. Branch is unconditional if M1 is all 1's. 2. Branch is unsuccessful if M1 is all 0's; use next sequential instr adr.	Prot (F)†	Unchanged
Branch on Condition	BCR	07	RR	M1 R2	Branch to location specified by GPR (adr by R2) if state of CC is as specified by M1. 1. Branch is unconditional if M1 is all 1's and R2 $\neq$ 0. 2. Branch is unsuccessful if R2 = 0 or if M1 is all 0's; use next sequential instr adr.	Prot (F)†	Unchanged
Branch on Count	BCT	46	RX	R1 D2(X2, B2)	Subtract 1 from 1st opr (in GPR per R1); if result $\neq$ 0, branch to location specified by 2nd opr adr. 1. Place result of subtraction into 1st opr location. 2. Branch is unsuccessful if result = 0; use next sequential instr adr. 3. If 1st opr = 1, no branching occurs.	Prot (F)†	Unchanged
Branch on Count	BCTR	06	RR	R1 R2	Subtract 1 from 1st opr (in GPR per R1); if result $\neq$ 0, branch to location specified by GPR (adr by R2). 1. Place result of subtraction into 1st opr location. 2. Branch is unsuccessful if result = 0 or if R2 = 0; use next sequential instr adr. 3. If 1st opr = 1, no branching occurs.	Prot (F)†	Unchanged
Branch on Index High	BXH	86	RS	R1 R3 D2(B2)	Add increment (3rd opr; in GPR per R3) to 1st opr (in GPR per R1), algebraically compare result (index) with comparand (in odd-adr GPR specified by R3 or R3 + 1); if index $>$ comparand, branch to location specified by 2nd opr adr. 1. Place index into 1st opr location. 2. Branch is unsuccessful if index = or $<$ comparand; use next sequential instr adr.	Prot (F)†	Unchanged
Branch on Index Low or Equal	BXLE	87	RS	R1 R3 D2(B2)	Add increment (3rd opr; in GPR per R3) to 1st opr (in GPR per R1), algebraically compare result (index) with comparand (in odd-adr GPR specified by R3 or R3 + 1); if index = or $<$ comparand, branch to location specified by 2nd opr adr. 1. Place index into 1st opr location. 2. Branch is unsuccessful if index $>$ comparand; use next sequential instr adr.	Prot (F)†	Unchanged
Execute	EX	44	RX	R1 D2(X2, B2)	Execute subject instr at location specified by 2nd opr adr. Subject instr may be modified by 1st opr (in GPR per R1) if E(8–11) $\neq$ 0. Modification is achieved by OR'ing bits 8–15 of subject instr with bits 24–31 of 1st opr; if R1 = 0, no modification takes place.	Execute Prot (F) Adr Spec	Set by subject instr

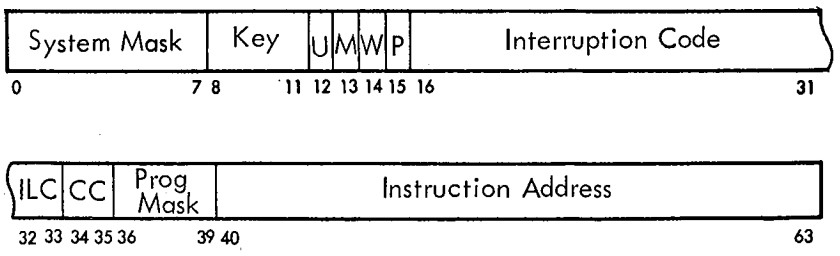
† Fetch protected: bit 4 of storage protect set.

Table 1-7. Status Switching Instructions

Instruction	Mnemonic	Op Code	Format	Operands	Function	Program Interruptions	Condition Code
Diagnose	None	83	SI	D1(B1) I2	Load word designated by stg opr adr into MCW, set or reset certain control triggers, & branch to ROS adr specified by MCW.	Priv Oper Prot (S,F) Adr Spec	Unpredictable
Insert Storage Key	ISK	09	RR	R1 R2	Insert stg protection key for 2048-byte stg block, adr by bits 8–20 of 2nd opr (in GPR per R2), into bits 24–28 of 1st opr (in GPR, per R1). 1. 1st opr: bits 0–23 are unchanged; bits 29–31 are cleared. 2. 2nd opr: bits 0–7 & 21–27 are ignored; bits 28–31 must = 0's. 3. Key is fetched twice because of 2-way interleaving.	Priv Oper Adr Spec	Unchanged
Load PSW	LPSW	82	SI	D1(B1)	Load doubleword stg opr (designated by stg opr adr) into CPU, thus replacing current PSW, & branch to new instr sequence. 1. Bits 0–15: system mask, protection key, program state. Bits 16–33: ignored. Bits 34–39: CC, program mask. Bits 40–63: instr adr. 2. If PSW(14) = 1, enter Wait state. 3. If PSW(15) = 1, enter Problem state. 4. Load PSW instr is only instr available for entering Problem or Wait state.	Priv Oper Prot (F) Adr Spec	Set by new PSW bits 34 & 35
Read Direct	RDD	85	SI	D1(B1) I2	Send 'direct control read out' signal & timing signal code (I2; in instr) to external device for about 0.6 usec; store 1 data byte from external device into stg (per stg opr adr) when 'direct control hold in' signal is absent.	Oper Priv Oper Prot (S) Adr	Unchanged
Set Program Mask	SPM	04	RR	R1	Replace CC & program mask (bits 34–39) of current PSW with bits 2–7 of 1st opr (in GPR per R1).	None	Set by opr 1 bits 2 & 3
Set Storage Key	SSK	08	RR	R1 R2	Set stg key (bits 24–28 of 1st opr; in GPR per R1) for 2048-byte stg block (adr by bits 8–20 of 2nd opr; in GPR per R2) into stg protection logic in main storage. 1. 1st opr: bits 0–23 & 29–31 are ignored. 2. 2nd opr: bits 0–7 & 21–27 are ignored; bits 28–31 must = 0's. 3. Key is set twice because of 2-way interleaving.	Priv Oper Adr Spec	Unchanged
Set System Mask	SSM	80	SI	D1(B1)	Replace system mask (bits 0–7) of current PSW with byte from location designated by stg opr adr.	Priv Oper Prot (F) Adr Multisys	Unchanged
Supervisor Call	SVC	0A	RR	I	Cause supervisor call interruption; replace old PSW(24–31) with I-field (bits 8–15) of instr, providing interruption code. 1. Clear PSW(16–23). 2. Store old PSW at stg location 32 (decimal). 3. Fetch new PSW from stg location 96 (decimal).	None	Unchanged
Test & Set	TS	93	SI	D1(B1)	Test high-order bit (bit 0) of stg opr byte (in stg), set CC according to state of tested bit, & set addressed byte back into stg as all 1's.	Prot (S,F) Adr	0 : High-order bit = 0 1 : High-order bit = 1
Write Direct	WRD	84	SI	D1(B1) I2	Send 'direct control write out' signal & timing signal code (I2; in instr) to external device for about 0.8 usec; make 1 data byte from stg (per stg opr adr) available to external device until next WRD is executed. (In multisystem mode, issue multisystem signals to other CPU.)	Oper Priv Oper Prot (F) Adr	Unchanged

1.3 PROGRAM STATUS WORDS (PSW)

The PSW has the following format:



0-7 System mask
- 0 Multiplexer channel mask
- 1 Selector channel 1 mask
- 2 Selector channel 2 mask
- 3 Selector channel 3 mask
- 4 Selector channel 4 mask
- 5 Selector channel 5 mask
- 6 Selector channel 6 mask
- 7 External mask

8-11 Protection key
12 USASCII-8 mode (A)
13 Machine check mask (M)
14 Wait state (W)
15 Problem state (P)
16-31 Interruption code
32-33 Instruction Length code (ILC)
34-35 Condition code (CC)
36-39 Program mask
- 36 Fixed-point overflow mask
- 37 Decimal overflow mask
- 38 Exponent underflow mask
- 39 Significance mask

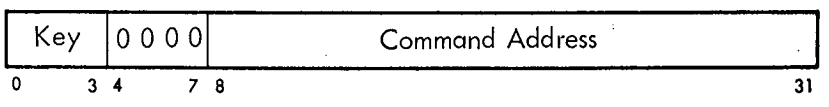
40-63 Instruction address

*A one-bit equals on, and permits an interruption

The interruption code is defined in Table 1-8. This table shows how interrupted instructions are finished. The permanent main storage assignments for the PSW's are listed in Table 1-9.

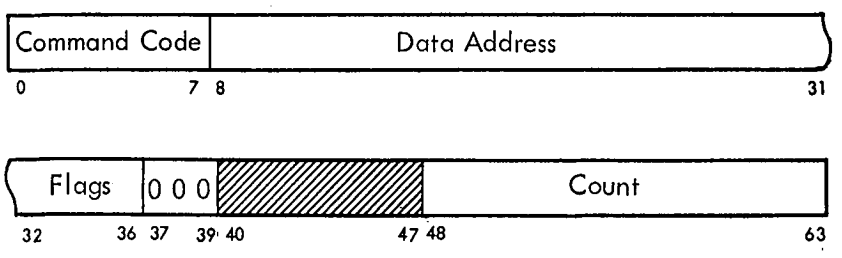
1.4 I/O CONTROL WORDS

The channel address word (CAW) has the following format:



0-3 Protection Key
4-7 Zeros
8-31 Command Address

The channel command word (CCW) has the following format:



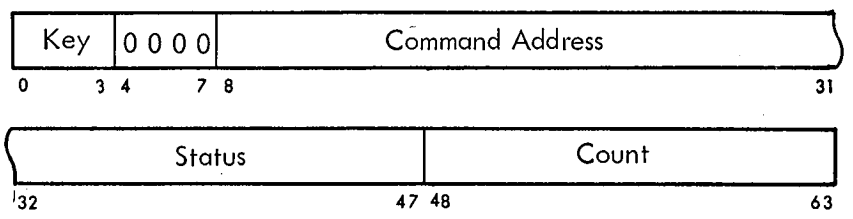
1. Bits 0–7, Command Code. The command codes are as follows (X indicates that the bit position is ignored; M denotes a modifier bit):

Names	Flags	Code
Write	CD CC SLI PCI	MMMM MM01
Read	CD CC SLI SKIP PCI	MMMM MM10
Read Backward	CD CC SLI SKIP PCI	MMMM 1 100
Control	CD CC SLI PCI	MMMM MM11
Sense	CD CC SLI SKIP PCI	MMMM 0 100
Transfer in Channel		x x x x 1 000

CD = Chain data
CC = Chain command
SLI = Suppress length indication
SKIP = Skip
PCI = Program-controlled interruption

2. Bits 8–31, Data Address. Specify the location of a byte in main storage.
3. Bits 32–36, Flags:
   - 32 Chain data flag
   - 33 Chain command flag
   - 34 Suppress length indication flag
   - 35 Skip flag
   - 36 Program-controlled interruption flag
4. Bits 37–39, Zeros.
5. Bits 40–47, Ignored.
6. Bits 48–63, Count. Specify the number of bytes in the operation.

The channel status word (CSW) has the following format:



0-3 Protection key
4-7 Zero
8-31 Command address
32-47 Status
- 32 Attention
- 33 Status modifier
- 34 Control unit end
- 35 Busy
- 36 Channel end
- 37 Device end
- 38 Unit check
- 39 Unit exception
- 40 Program-controlled interruption
- 41 Incorrect length
- 42 Program check
- 43 Protection check
- 44 Channel data check
- 45 Channel control check
- 46 Interface control check
- 47 Chaining check

48-63 Count. Form the residual count for the last CCW used.

Table 1-8. Interruptions

Interruption	Interruption Code (Old PSW Bits 16–31)	PSW Mask Bit	ILC	How Instruction Execution Is Finished
Machine Check	00000000 00000000	13	u	Terminated
Program				
Operation	00000000 00000001	–	1, 2, 3	Suppressed
Privileged operation	00000000 00000010	–	1, 2	Suppressed
Execute	00000000 00000011	–	2	Suppressed
Protection	00000000 00000100	–	0, 2, 3	Suppressed or Terminated
Addressing	00000000 00000101	–	0, 1, 2, 3	Suppressed or Terminated
Specification	00000000 00000110	–	1, 2, 3	Suppressed
Data	00000000 00000111	–	2, 3	Terminated
Fixed-point overflow	00000000 00001000	36	1, 2	Completed
Fixed-point divide	00000000 00001001	–	1, 2	Suppressed or Completed
Decimal overflow	00000000 00001010	37	3	Completed
Decimal divide	00000000 00001011	–	3	Suppressed
Exponent overflow	00000000 00001100	–	1, 2	Completed
Exponent underflow	00000000 00001101	38	1, 2	Completed
Significance	00000000 00001110	39	1, 2	Completed
Floating-point divide	00000000 00001111	–	1, 2	Suppressed
Multisystem	00000000 00010010	–	2	Suppressed
Supervisor Call	00000000 rrrrrrrr	–	1	Completed
External				
External signal 7	00000000 ppppppp1	7	u	Completed
External signal 6	00000000 pppppp1 p	7	u	Completed
External signal 5	00000000 ppppp1 pp	7	u	Completed
External signal 4	00000000 pppp1 ppp	7	u	Completed
External signal 3	00000000 ppp1 pppp	7	u	Completed
External signal 2	00000000 pp1 ppppp	7	u	Completed
INTERRUPT pushbutton	00000000 p1 pppppp	7	u	Completed
Timer	00000000 1ppppppp	7	u	Completed
I/O				
Multiplexer channel (0)	00000000 aaaaaaaa	0	u	Completed
Selector channel 1	00000001 aaaaaaaa	1	u	Completed
Selector channel 2	00000010 aaaaaaaa	2	u	Completed
Selector channel 3	00000011 aaaaaaaa	3	u	Completed
Selector channel 4	00000100 aaaaaaaa	4	u	Completed
Selector channel 5	00000101 aaaaaaaa	5	u	Completed
Selector channel 6	00000110 aaaaaaaa	6	u	Completed

Notes:
- u: Unpredictable; E(0,1)
- r: R1 and R2 fields of Supervisor Call instruction
- p: Set if pending before PSW(7) is set to a 1.
- a: I/O device address

Table 1-9. Permanent Main Storage Assignments

Main Storage Address		Length	Information Stored
Dec	Hex		
0	0	Doubleword	Initial program loading PSW
8	8	Doubleword	Initial program loading Channel Command Word 1 (CCW 1)
16	10	Doubleword	Initial program loading CCW 2
24	18	Doubleword	External interruption, old PSW
32	20	Doubleword	Supervisor call interruption, old PSW
40	28	Doubleword	Program interruption, old PSW
48	30	Doubleword	Machine check interruption, old PSW
56	38	Doubleword	I/O interruption, old PSW
64	40	Doubleword	Channel Status Word (CSW)
72	48	Word	Channel Address Word (CAW)
76	4C	Word	Unassigned
80	50	Word	Timer
84	54	Word	Unassigned
88	58	Doubleword	External interruption, new PSW
96	60	Doubleword	Supervisor call interruption, new PSW
104	68	Doubleword	Program interruption, new PSW
112	70	Doubleword	Machine check interruption, new PSW
120	78	Doubleword	I/O interruption, new PSW
128	80	22 doublewords	Diagnostic logout area

1.5 LOGIC DIAGRAMS

The logic diagrams for the Model 65 consist of automated logic diagrams (ALD's) and control automation system logic diagrams (CLS's). The following text describes the notation used in each together with an index for each.

1.5.1 ALD Notation

Detailed information on how to read ALD's may be found in General CEM number one.

1.5.2 Additive Card Codes

When the installation of a feature, or a portion of it, is made by plugging in cards, an additive card code (ACC) is printed in the logic blocks for those cards. This code appears on the second line within the block. A list of additive card codes for the Model 65 follows:

Code	Definition
ADA2	1052 Adapter Feature
CBS5	7074 Compatibility Feature
CBS6	7080 Compatibility Feature
CBS7	7090 Compatibility Feature
CH0	2870 Multiplexer Channel
CH3	2860 Selector Channel 3
CH4	2860 Selector Channel 4
CH5	2860 Selector Channel 5
CH6	2860 Selector Channel 6
DCT	Direct Control Feature
FRA6	Additional Storage Attachment Feature, 6th storage unit
FRA7	Additional Storage Attachment Feature, 7th storage unit
FRA8	Additional Storage Attachment Feature, 8th storage unit
HRT	High-Resolution Timer (RPQ)
IND4	Models IH and J
LCSA	2361 Attachment
LP	List Processing (RPQ)
MP	Multiprocessor
MPAS	Additional Storage Attachment Feature, 5th storage unit
MS, MULS	Multisystem Feature
M34	Models IH and J
RND	Rounding (RPQ)
ROSE	Read-Only Storage Extension
TIM	50-Hz Oscillator

1.5.3 ALD Index

The ALD's are divided into volumes as follows:

	Volume Number
Basic CPU	1 to 21
1052 Adapter Feature	1A
LAD's	1B
Basic CLD's	12
7074 Emulator CLD's	70
7074 Emulator ALD's	71
7080 Emulator CLD's	80
7080 Emulator ALD's	81
7090 Emulator CLD's	90
7090 Emulator ALD's	91

The following is an index of the 2065 ALD's, including the Compatibility features:

1.5.4 CLD Notation

Refer to logic M7061 in the LADS pages for an explanation of CLD notation.

1.5.5 CLD Index

The following is an index of the 2065 ROS CLD's:

Address List QZ001–QZ091
Branching Operations QE031–QE051
Decimal Operations QS001–QS121
Emulators:
- 7074:
  - Accumulator Operations QX061, QX101–QX111, QX311–QX331
  - Core-to-Core Block Transmission QX151–QX191
  - DIL QX011
  - Entries QX001
  - Index Word Operations QX301, QX341
  - Logical Operations QX121–QX131
  - Multiple-Usage Operations QX021–QX061, QX201–QX291
  - Table Lookup Operations QX351–QX371
- 7080:
  - DIL QW301
  - Entries QW311
  - General-Purpose Operations QW321–QW341
  - Indirect Address QW301
  - I/O Manipulation QW401–QW411
  - Memory-to-Memory Operations QW211–QW221
  - Memory-to-Storage Operations QW001–QW071
  - Storage Field Manipulation QW201–QW205
  - Storage-to-Memory Operations QW101–QW151
- 7090:
  - DIL QV904
  - Entries QV001–QV005
  - Error Operation QV941
  - Fixed-Point Operations QV031–QV071
  - Floating-Point Operations QV001, QV201–QV411
  - Indexing QV801–QV811
  - Manual Controls QV941
  - Miscellaneous QV001, QV021, QV701, QV901
  - Multiple-Usage Operations QV011, QV 01, QV911–QV941
  - Shift Operations QV601–QV641
  - Test Mode Operations QV941
  - Transfer Operations QV701–QV721
  - Word Transmission QV011, QV801

Fixed-Point Operations QB001–QB041, QE011–QE021, QF001, QJ001–QJ021, QJ504
Floating-Point Operations QC001–QC071, QG001–QG011
I-Fetch QT001–QU001
I/O Operations QK021
IPL QK021
Logical Operations QE001, QJ001–QJ021, QJ504–QK011, QP001–QP061
7074 Emulator (see Emulators)
7080 Emulator (see Emulators)
7090 Emulator (see Emulators)
Status Switching Operations QJ031–QJ041
Test Mode Operations QY001–QY051

1.5.6 Version Numbers

The System/360 engineering design and control scheme has led to the development of versions. The basic machine is designated version 000. Any subsequent variation (such as a feature or an RPQ) which is not an engineering change results in another version. The following is a list of the 2065 CPU versions:

Version ALD	Version CLD	Definition
000	000	Basic Machine
001	001	Special Move Instruction, 7074 Emulator (RPQ)
004	–	Model IH, Model J, or 2361 Attachment Feature
010	010	Power Failure Logout (RPQ)
018	004	Bank of America (RPQ)
022	–	High Resolution Timer (RPQ)
024	024	Edit, Normalize, and Scale (RPQ)
025	025	Shared Storage (RPQ)
026	026	Multisystem Feature
040	–	Non-IBM Meter (RPQ)
047	–	Remove Operator's Console (RPQ)
048	–	7-Bit Storage Protect (RPQ)
–	050	Invalid Sign Decimal Operation Suppression (RPQ)
051	051	Inverse Move (RPQ)
074	074	7074 Compatibility Feature
080	080	7080 Compatibility Feature
090	090	7090 Compatibility Feature
091	091	Sense Indicator Operations (RPQ)
–	180	Special Instruction, 7080 Emulator (RPQ)
A22	–	High Resolution Timer, 7090 Emulator (RPQ)
A25	A25	Shared Storage, 7090 Emulator (RPQ)
A44	–	7080 Emulator Tape Parity (RPQ)
A76	–	Combination of Versions 026 and 074
A86	–	Combination of Versions 026 and 080
A91	A91	Power Failure Logout 7090 Emulator (RPQ)
A96	–	Combination of Versions 026 and 090
B22	–	High Resolution Timer IH65, J65 (RPQ)
B25	B25	Shared Storage IH65, J65 (RPQ)
B36	–	2065/2930 Sync Pulse (RPQ)
B74	B74	Invalid Alpha, 7074 Emulator (RPQ)
C22	–	High Resolution Timer, 7080 Emulator (RPQ)
C25	C25	Shared Storage, 7074 Emulator (RPQ)
D22	–	High Resolution Timer, 7074 Emulator (RPQ)
D25	D25	Shared Storage, 7080 Emulator (RPQ)
E25	E25	Shared Storage LCS (RPQ)
K69	–	Additional Storage Attachment Feature Storage Indicators

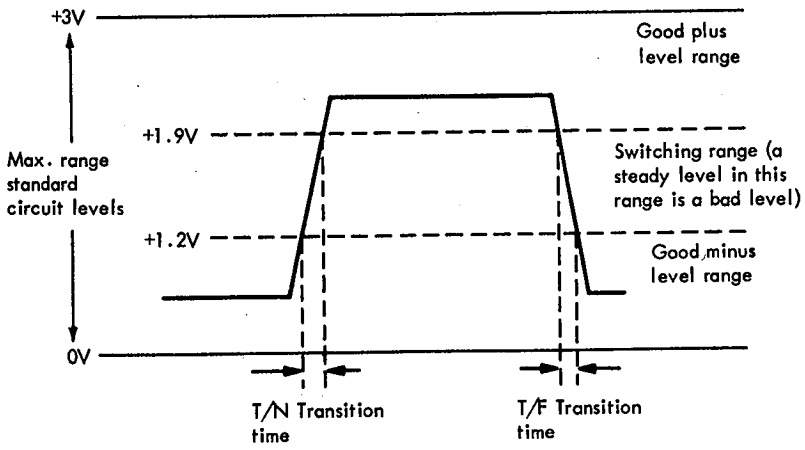
Note: Version numbers do not always agree between ALD's and CLD's because some versions do not affect both.

1.6 MODEL 65 CIRCUIT LEVELS

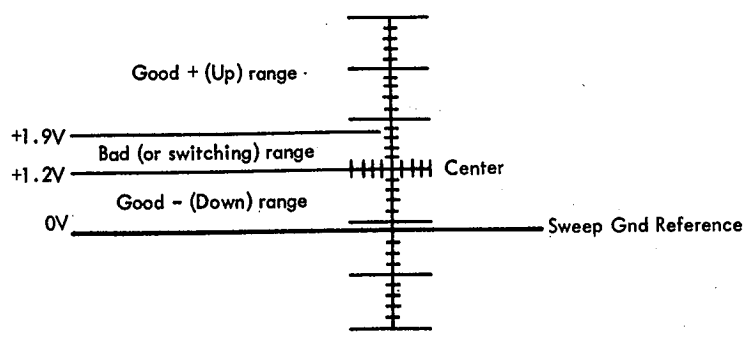
The Model 65 CPU uses the AOI (AND, OR, INVERT) family of high-speed circuits. The switching speeds of these circuits is in the order of 5 to 15 ns. The maximum range of signal voltage levels in the system is 0 to +3V.

To compare one standard level pulse to another or to measure a pulse width, an understanding of switching points is required. The switching points for high-speed circuits are shown in Figure 1-7, A. For the positive-going slope of a pulse, the switching point is +1.2V above ground (0 volts); for the negative going slope of a pulse, the switching point is +1.9V above ground (0 volts). This applies to both + pulses and - pulses. A good method to follow when scoping in CPU is:

1. Set vertical calibration to 1V/div.
2. Ground the input to be used for scoping.
3. Set the vertical position (with free-running trace) until trace is 1.2 cm (volts) below center.
4. Now while scoping, any steady level below the center line is a good "down" level, any steady level in the 1-cm division above the center line is bad or doubtful, and any steady level in the second 1-cm division above center is a good "up" level (Figure 1-7, B). Actually not all of the 1-cm division above center is in the bad level range; only 0.7 of it.



A. STANDARD LEVEL PULSE, SWITCHING POINTS



B. STANDARD SWITCHING LEVELS

Figure 1-7. 2065 Circuit Levels

Medium-speed circuits having switching speeds of approximately 30 ns are used in a few instances in the 2065. Details of individual circuit switching levels are found in General CEM Index #1 which deals with FEALD notation.

1.7 SYNC SIGNALS AND LOCATIONS

Main storage address compare sync is available whenever the machine is operating and an equal compare occurs between the storage address and the main storage address compare switches (bits 2–20 of the ADDRESS switches). This signal is available on each gate at the following points:

Gate A	01A-A4N3B06
Gate B	01B-D3N2D13
Gate C	01C-B3A5B12
Gate E	01E-E2N2D11

Note: If the multisystem feature is installed, the ADDRESS switches must be set to the relocated main storage address when prefixing is active.

ROS address compare sync is available whenever the machine is operating and an equal compare occurs between the address in ROSAR and the ROS address compare keys (bits 0–11 of the ADDRESS switches). This signal is available on gate C at 01C-A3H5B10. ROS address 150 set into the address keys provides the sync for FLT's.

Note: Both main storage and ROS address compare sync signals are available on coaxial connectors at the front of gate C if EC 705168 is installed. The ground side of these connectors is at signal ground potential, or is at the same potential as D08 on the large boards. Scope probe adapter PN 453199 should be used with these connectors.

Note: No switch is provided for ROS address compare stop such as is available for main storage compare. To stop on ROS address compare, jumper 01C-A3H5B10 to 01E-E2M4D09.

The following additional sync points are listed for reference:

Start I/O	01B-E4G7B03
Test I/O	01B-E4G7D10
Halt I/O	01B-E4G7B04
Test Channel	01B-E4G7B03

Chapter 2. Console Controls and Maintenance Features

This chapter consists of three sections:

Section 1 - describes the controls on the system control panel, CE panel, and configuration control panel.

Section 2 - describes the operating procedures for the system control panel.

Section 3 - describes specific maintenance features available to the CE: (1) diagnose instruction and associated MCW's; (2) scan-in operations; (3) logout, ROS tests, and FLT's; and (4) diagnostic programs.

Section 1. Control Panels

This section briefly describes the functions of all controls on the system control panel, CE panel, and configuration control panel.

2.1 SYSTEM CONTROL PANEL

The system control panel, in addition to its main function as the operating and monitoring center of the system, is one of the prime maintenance aids available to the CE. Using this panel, a CE can duplicate many program operations or portions of operations manually and can repeatedly exercise portions of the machine logic at a normal or a reduced rate of operation.

Some of the maintenance routines that can be performed from the system control panel include storage ripple, marginal checking, and frequency bias. Other controls allow the CE to stop the CPU at the end of the current instruction, to display main storage or LS, to store into main storage or LS, and to log out indicator status to fixed positions in main storage.

The system control panel is divided into seven separate panels (A to G), as shown in Diagram 8-1, FEMDM. The operator control section (panel G) is identical to those used on the other models of the System/360, thus providing the operators with compatibility between models. If the 2150 Console or the 2250-1 Display Unit is included in the Model 65 System, the operator control section and the EMERGENCY PULL switch may be duplicated on one of these units to provide monitor control operation.

The functions of the controls and indicators, by panel, on the system control panel are defined briefly in the ensuing paragraphs.

2.1.1 Panel A

1. DC voltmeter. Indicates the voltage levels of the marginable supplies. The particular supply indicated is determined by the MARGIN/METER SEL switch.
2. MARGIN/METER SEL switch. Has 12 positions to select the power supply to be indicated by the meter and to determine which of the attached stand-alone units may be marginally checked:
   a. STORE FRAME 1: Selects HSS unit 1.
   b. STORE FRAME 2: Selects HSS unit 2.
   c. STORE FRAME 3: Selects HSS unit 3.
   d. STORE FRAME 4: Selects HSS unit 4.
   e. CHAN FRAME 1: Selects channel 1.
   f. CHAN FRAME 2: Selects channel 2.
   g. CHAN FRAME 3: Selects channel 3.
   h. ROS LOCATE: Selects the 18V ROS bias power supply (gate D in the CPU).
   i. CPU A: Selects gate A in the CPU.
   j. CPU B: Selects gate B in the CPU.
   k. CPU C: Selects gate C in the CPU.
   l. CPU E: Selects gate E in the CPU.

   Note: When an Additional Storage Attachment feature is installed in CPU2, STORE FRAME 1, 2, 3, and 4 select HSS units 5, 6, 7, and 8, respectively.

3. MARGIN indicators:
   a. ACTIVE: Indicates that an internal power supply or an attached HSS unit or channel is being marginally checked.
   b. LOCATE: Indicates when the MARGIN/METER SEL switch is set to the position corresponding to the margined power supply, attached HSS unit, or channel.

4. POWER CHECK indicators. These eight indicators (CPU, STOR FRAME 1, 2, 3, and 4, and CHAN FRAME 1, 2, and 3) indicate an incomplete power-up status in the CPU, HSS units 1, 2, 3, and 4, and channels 1, 2, and 3, respectively.

   Note: When an Additional Storage Attachment feature is installed in CPU2, STOR FRAME 1, 2, 3, and 4 indicate an incomplete power-up status in HSS units 5, 6, 7, and 8, respectively.

5. MARGIN CHANNEL/STOR switch. Applies power to a motor in the channel or HSS unit selected by the MARGIN/METER SEL switch to lower or raise the output voltage levels from the marginable supplies in that unit or channel. The channel or storage unit must be in remote operation.

2.1.2 Panel B

The five potentiometers on this panel (ROS, +6M A GT, +6M B GT, +6M C GT, and +6M E GT) raise or lower the output voltage levels from the 18V ROS and the 6V gate A, B, C, and E supplies, respectively.

2.1.3 Panel C

The pull switch on this panel, EMERGENCY PULL, when pulled, initiates emergency power-off in the system. The switch latches in the out position and must be manually restored by the CE.

2.1.4 Panel D

This panel is blank in the basic 2065. When an Additional Storage Attachment feature is installed, a four-position STORAGE INDICATE rotary switch is added.

1. STOR 1–4: Allows the contents of the storage address registers for HSS units 1 and 2 to be displayed on the roller switches.
2. STOR 5–8: Indicates HSS units 3 and 4.
3. STOR 9–12: Indicates HSS units 5 and 6.
4. STOR 13–16: Indicates HSS units 7 and 8.

2.1.5 Panel E

1. Roller switches and indicators. This section of the panel contains six six-position roller switches, with 36 indicators (implicitly numbered 0–35) associated with each roller switch. A roll chart associated with each roller switch rotates with the roller switch to correspond with the roller position, thus showing the information being displayed for each indicator. Diagram 8-2, FEMDM, identifies the indicators for the six positions of each roller switch. These indicators are tested between positions of the switch. Position 6 of roller 6 is used to test the remaining indicators on the system control panel and on the 2150 Console.
2. DATA 0–31 and DATA 32–63 switches. These 64 switches, in hexadecimal groups, permit data to be entered manually. Correct parity is automatically generated by contacts on the switches.
3. ADDRESS switches. These 24 switches, in hexadecimal groups, select an addressable location in ROS, LS, or main storage. Correct parity is automatically generated by contacts on the switches.
4. STOR CHK (storage check) indicator: Indicates an error in the storage units.
5. PROC CHK (processor check) indicator: Indicates an error in the CPU.

2.1.6 Panel F

1. TEST MODE, REPEAT switch: Repeats the ROS test or FLT in main storage continuously.
2. TEST MODE, ROS/PROC/FLT switch. This switch has three positions:
   a. ROS: Places the CPU into ROS test mode and removes program control.
   b. PROC (process) – normal position for CPU operation.
   c. FLT: Places the CPU in the FLT mode and removes program control.
3. STORAGE INDICATE switch. This switch has two positions:
   a. STOR 1–4 - normal position: Allows the contents of the storage address registers for HSS units 1 and 2 to be displayed on the roller switches.
   b. STOR 5–8: When in this position, the operation is the same as in the normal position for all Model 65 configurations except Models IH and J. For these models, the contents of the storage address registers for HSS units 3 and 4 are displayed on the roller switches.

   Note: This switch is spare if an Additional Storage Attachment feature is installed.

4. FREQUENCY ALTERATION switch: Decreases the CPU clock period from 200 ns to 195 ns. Operates only with the CE key switch in the CE position.
5. DEFEAT INTERLEAVING switch. This switch has three positions:
   a. PROC (process) – normal position: Addressing is interleaved with no even/odd storage area reversal, unless changed by the Diagnose instruction.
   b. REV (reverse): Interleaving is disabled and the even and odd storage areas are reversed. No reversal occurs on Model G65.
   c. NO REV (no reverse): Interleaving is disabled and no even/odd storage area reversal takes place.

6. STOP ON STORAGE CHECK switch: Inhibits operation of and maintains the environment of the main storage basic storage module in which an error was detected. Other basic storage modules are not affected. (If the MP feature is installed, and storage is enabled to both CPU's, either CPU's switch will cause a stop on all errors.)
7. DISABLE INTERVAL TIMER switch: Prevents the interval timer from being decremented.

   Note: When the Multisystem feature is installed, disabling the interval timer when operating in the multisystem mode also disables the detection of CPU inactivity.

8. STORAGE SELECT switch. This switch has three positions:
   a. MAIN – normal position: Selects main storage for manually storing or displaying data.
   b. LOCAL: Selects LS for manually storing or displaying data.
   c. MAIN BYTE: Same as MAIN except that the byte selected by ADDRESS switches 21–23 is the only byte affected by a manual store operation.
9. ADDRESS COMPARE STOP switch: Stops processing if the main storage address being requested agrees with bits 2 through 20 of the ADDRESS switches.
10. CPU CHECK switch. This switch has three positions:
    a. PROC (process) – normal position: If the PSW machine check mask [PSW(13)] is a 1, the CPU stops on detection of a CPU check and the status is logged into main storage. A machine check interruption then takes place. If the mask is a 0, the result is the same as if the switch is in the DSBL position.
    b. DSBL (disable): The CPU does not stop on detection of a machine check, but the check trigger is set. No logout or interruption takes place.
    c. STOP: The CPU stops on detection of a machine check, but there is no logout of data. The check trigger is set.
11. PULSE MODE switch. This switch has three positions:
    a. PROC (process) – normal position: Does not affect CPU operation.
    b. COUNT: Provides a means of looping through a selected number of machine cycles (maximum of 2047). The number of cycles is entered into DATA switches 53–63. Each loop starts at the address contained in main storage location 0. The RATE switch must be in the PROCESS position.
    c. TIME: Provides looping when the interval timer is decremented. Each loop starts at the address contained in main storage location 0. The RATE switch must be in the PROCESS position.
12. REPEAT INSN (instruction) switch. This switch has three positions:
    a. PROC (process) – normal position: Does not affect CPU operation.
    b. SINGLE: Allows the first instruction in the DATA switches to be repeated continuously.
    c. MPLE (multiple): Allows continuous looping through the four instruction halfwords in the DATA switches.
13. REPEAT ROS ADDRESS switch: Continuously reads out the ROS address specified by ADDRESS switches 0–11. ROS TRANSFER must be depressed to start this loop.
14. DISABLE DIRECT CONTROL switch. Causes Write Direct and Read Direct instructions to become invalid instructions, and external signals to be ignored.

    Note: The DISABLE DIRECT CONTROL switch is added by the Direct Control feature.

15. RATE switch. This switch has four positions:
    a. INSN STEP (instruction step): CPU executes one machine instruction for each time START is depressed.
    b. PROCESS: Does not affect CPU operation; CPU operates at normal clock speed.
    c. SINGLE CYCLE: CPU advances by its minimum clock amount for each depression of START; all CPU operations are the same as for the PROCESS position.
    d. SINGLE CYCLE STORAGE INHIBIT: Same as SINGLE CYCLE without storage references.
16. SYSTEM RESET pushbutton: Resets main storage check triggers, on-line channels and control units, and CPU controls and check triggers to their initial state.
17. CHECK RESET pushbutton: Resets all CPU check triggers and latches to the non-error state.
18. PSW RESTART pushbutton: Loads a new PSW from main storage location 0 and starts processing if the RATE switch is in PROCESS.
19. ROS TRANSFER pushbutton: Serves to display the contents of any ROS location or to begin processing from any ROS address.
20. SET IC (instruction counter) pushbutton: Enters an address from the ADDRESS switches into bits 40–63 of the active (current) PSW. The data referenced will be loaded into Q, the first instruction will be transferred to R, and the IC will be updated by +8 bytes.
21. STORE pushbutton: Enters data from the DATA switches into the storage location specified by the STORAGE SELECT and ADDRESS switches.

22. DISPLAY pushbutton: Displays data from LS or main storage specified by the STORAGE SELECT and ADDRESS switches.

    Note: If the Multisystem feature is installed and prefixing is active when the SET IC, STORE, or DISPLAY pushbutton is depressed, the main storage address set into the ADDRESS switches is relocated.

23. START pushbutton: Starts the CPU operating in the mode selected by the RATE switch.
24. STOP pushbutton: Terminates CPU operation without changing the environment.
25. RESTART FLT I/O pushbutton: Backspaces one tape record and starts reading during ROS test or FLT operations.
26. LOG OUT pushbutton: Stores CPU status into fixed locations in main storage when the CPU is in the Stopped state.
27. Usage meters and CE Key switch: The usage meters indicate elapsed CPU running time: the Process (left) meter shows elapsed customer usage time; the CE (right) meter shows elapsed system maintenance time. The CE Key switch, when turned to the right, disables the Process meter and enables the CE meter and the FREQUENCY ALTERATION switch.

2.1.7 Panel G

1. POWER ON pushbutton: Initiates power-on sequence in the CPU and the system units. The pushbutton is back-lighted by two indicator lights. When system power is on, the pushbutton glows white; when system power is off or not correctly on, the pushbutton glows red.
2. POWER OFF pushbutton: Initiates power-off sequence in the CPU and the system units.
3. LOAD UNIT switches: These three switches select the I/O unit to be used by a load operation.
4. INTERRUPT pushbutton: Causes an external interruption in the system and sets bit 25 of the interruption code to a 1.
5. SYSTEM indicator: Indicates that a CPU usage meter is running.
6. MANUAL indicator: Indicates the CPU is in the Stopped state.
7. WAIT indicator: Indicates the CPU is in the Wait state.
8. TEST indicator: Indicates that a switch on panel F is not in the normal operating position or that a channel is in the test mode.
9. LOAD indicator: Indicates a CPU load operation. A successful load turns this indicator off.
10. LOAD pushbutton: Resets the system and starts a load operation.

2.2 CE PANEL

The CE panel is shown in Diagram 8-29, FEMDM. The controls and indicators function as follows (see Chapter 5 for details):

1. THERMAL RESET pushbutton: Resets the thermal sense relays in the CPU.
2. CPU READY/OFF switch. This switch has two positions:
   a. READY – normal position: Allows CPU power-up sequencing to continue if the thermal and overcurrent conditions are normal.
   b. OFF: Drops CPU power without affecting system power. The CPU is bypassed on a system power-on sequence.
3. CPU ON pushbutton: Starts CPU power-on sequencing if the CPU READY/OFF switch is in the READY position. Does not affect system power.
4. THERMAL TRIP indicators: These six indicators show the location of the overtemperature condition that dropped CPU power. The temperature sensors are located in gates A, B, C/D, and E, the converter/inverter, and the power supply tubs.
5. UNDER VOLTAGE CHECK switches (located on the relay gate below the CE panel in the converted units): These 15 switches isolate the power supplies from the undervoltage sensing circuits.

2.3 CONFIGURATION CONTROL PANEL (MULTISYSTEM FEATURE ONLY)

The configuration control panel is added by the multisystem feature and is shared by two CPU's. For the following discussion of the configuration control panel, refer to Diagram 7-11, FEMDM.

2.3.1 Storage Allocation Switches

A Storage Allocation toggle switch for each CPU is associated with each storage unit. Only with the switch set to ENABLE is the storage unit available to that CPU. At least one storage unit is allocated to each operating CPU. The same storage unit may be allocated to both CPU's. A change in storage allocation becomes effective when the associated CPU enters the Stopped or Wait state.

2.3.2 Floating Address Switches

One Floating Address switch is associated with each storage unit. These rotary switches control the address range of each storage unit in 262-kilobyte intervals. The first four positions are: 0 TO 262K TO 524K, 524K TO 786K, and 786K TO 1048K. If the first Additional Storage Attachment feature is installed, these switches may also be

placed in one of the next four positions: 1048K TO 1310K, 1310K TO 1572K, 1572K TO 1834K, and 1834K TO 2096K.

Contiguous addresses are usually assigned for each CPU, starting with address 0. Two storage units may not be assigned the same address interval if either is allocated to both CPU's. To provide maximum thruput, the direct-address main storage unit(s) should be as close to the CPU as possible; refer to Diagram 7-2, FEMDM. The Floating Address switches are always active.

2.3.3 PREFIX Switches

One PREFIX toggle switch is associated with each CPU. With the switch set to ENABLE, CPU references to the low-order 4096 (decimal) bytes and the high-order 4096 bytes of main storage are swapped, or relocated; also, channel references to the low-order 4096 bytes are relocated to the high-order 4096 bytes. Relocation does not occur when the PREFIX switch is set to DISABLE. Normally, neither of these two switches is set to ENABLE when no storage units are shared; only one is enabled when all storage units are shared. A change in position of a PREFIX switch becomes effective when the associated CPU (1) performs a power-on sequence, (2) has its LOAD pushbutton depressed, or (3) performs an external start operation.

2.3.4 CPU Mode Switches

One CPU Mode switch is associated with each CPU and determines the operating mode of that CPU. Multisystem mode (MS) is entered or exited when the CPU clock is stopped, according to the position of the switch. Switching between Partition (PTN) and Model 65 (65) modes is immediately effective.

2.3.5 I/O Allocation Switches

These switches enable or disable communications between the associated CPU and the control unit attached to two channels. Switches for up to 24 control units can be accommodated on the configuration control panel. These switches are labeled as a group, *I/O CONTROL*, and individually according to the control unit type. Refer to the applicable control unit FETOM for a further description of these switches.

2.3.6 VALID ADDRESS Indicators

One VALID ADDRESS indicator is associated with each CPU. When not lit, an incorrect manual assignment of an allocated storage unit has been made: either the same floating addresses have been assigned more than once or a Floating Address switch is in one of the last four positions without an Additional Storage Attachment feature being installed. The indicator lights after the needed correction is made. Not until then can the associated CPU be operated.

Section 2. Console Operating Procedures

This section describes specific operating procedures which enable the CE to monitor the system performance, to manually duplicate certain program operations, and to repeatedly exercise portions of the machine at a normal or a reduced rate.

2.4 TURNING ON SYSTEM POWER

1. Before turning on the system, check all peripheral units externally. Check that doors are properly closed, feeds are not impeded, paper and card supplies are suitable, and removed I/O units are jumpered to permit power sequencing.
2. Press POWER ON to initiate a power-on sequence. The main POWER ON indicator (panel G) will at first light up red, and then white. (If the power-on sequence has not been successfully completed, the POWER ON indicator will remain red.)
3. Turn on the power for shared/peripheral equipment not directly connected in the power-on sequence.

2.5 TURNING OFF SYSTEM POWER

If the MANUAL light is not on, press STOP. MANUAL light then goes on. Proceed as follows:

1. Check all tape drives. Put them in an unload state.
2. Check all disk drives. Put them in unload state by pressing disk START/STOP pushbutton if drives are running.

   Note: If the Multisystem feature is installed, check that all storage and I/O allocation switches for this CPU are in the DISABLE position if the other CPU is still operating.

3. Press system POWER OFF pushbutton. Power is removed automatically in an ordered sequence and POWER ON goes out.
4. Continue power-off procedures for shared/peripheral equipment not connected to the power sequence.

2.6 STOPPING AND RESTARTING THE SYSTEM

To stop all machine operations without destroying the machine environment, depress STOP. The CPU completes the instruction or I/O operation in process and enters the stopped state. To restart the system (from the point it was stopped), depress START.

2.7 RESETTING THE SYSTEM

The initial condition of the system is the stopped state, with all channels, control units, and CPU(s) controls (including machine check logic) in the normal (no error) state. This condition is automatically set during normal power-on sequencing and may be set manually by depressing SYSTEM RESET.

Note: The system reset operation does not reset the data and instruction registers in the CPU.

2.8 RESETTING CPU ERROR CHECK LOGIC

To restore the CPU error-check logic to the normal (no error) state, depress CHECK RESET. All CPU error-check logic is reset, and the check indicators are cleared. Processing resumes if it was stopped because of a machine check; the results, however, may be inaccurate.

2.9 EMERGENCY POWER OFF

The EMERGENCY PULL switch is to be pulled out by the operator in true emergency only. In this regard, a true emergency is defined as a system fire or any extreme case in which real danger threatens personnel. Once pulled out, the switch locks, and it can be reset only by CE action.

2.10 TURNING ON CONTROL UNIT POWER

1. Be sure the channels to which the control unit is attached have completed all operations (no pending interrupts existing.)
2. Stop system operation.
3. Place channel in Test Mode.
4. Place control unit in CE Mode.
5. Turn on control unit power.
6. Place control unit on-line.
7. Manually reset channel.
8. Place channel in Auto Mode.
9. Restart CPU operation by depressing START.

2.11 TURNING OFF CONTROL UNIT POWER

1. Be sure channel on which control unit is attached has completed all operations (no pending interrupts existing).
2. Stop CPU operations.

3. Place channel in Test Mode.
4. Place control unit in CE mode.
5. Turn off control unit.
6. Manually reset channel.
7. Place channel in Auto Mode.
8. Restart CPU operations.

2.12 MANUAL IPL

The IPL procedure is performed after a power-on sequence, after malfunctions that necessitate reloading the resident portion of the operating system into main storage, or for initial loading of any standalone program. To perform a manual IPL, proceed as follows:

1. Place program on desired I/O device.
2. Make I/O device ready.
3. Select desired I/O device with the three LOAD UNIT switches.
4. Press LOAD pushbutton.

The MANUAL/WAIT light goes off, the LOAD light turns on, and system reset occurs. The first 24 bytes of information are then loaded from the selected device into main storage locations 0–23. Depending on data content, the exact procedure from this point varies. A typical illustration follows:

The IPL PSW is in main storage locations 0–7, with two channel command words (CCW's) following in locations 8–23. The channel then obtains the first CCW from locations 8–15 and uses it to continue reading from the input unit, placing the data in storage. When the reading operation is completed, the LOAD light turns off, and CPU processing begins with the IPL PSW being loaded into the PSW register. If the reading operation is unsuccessful, the LOAD light remains on, and the CPU does not begin processing.

2.13 DISPLAY PSW

When the CPU is in the wait or stopped state, all but two parts of the current PSW are identified on roller 4, position 1. The two exceptions are: (1) the instruction address which is displayed in the D-register (roller 1, position 2), and (2) the instruction length code which is displayed in E-register positions 0 and 1 (roller 5, position 3).

2.14 RESTART FROM INITIAL PSW

If the CPU is in the stopped state, press PSW RESTART. The PSW is then loaded from main storage location 0 into the CPU. If the RATE switch is in the PROCESS position, the CPU will start processing at the instruction location specified by the address field of the PSW.

2.15 LOAD INSTRUCTION COUNTER

The instruction counter (Bits 40–63 of the current PSW) is always dynamically displayed by roller 6, position 3. To alter these bits without affecting the rest of the PSW:

1. Press STOP.
2. Set the desired address in ADDRESS switches 0–23.
3. Press SET IC to load the contents of the ADDRESS switches into the address field of the current PSW.
4. Press START.

2.16 INSTRUCTION STEPPING

Instruction stepping enables the system to process one instruction at a time, to service all interrupts, and to stop. Proceed as follows:

1. Press STOP; MANUAL light goes on.
2. Set RATE switch to INSN STEP; test light goes on.
3. Press START. The next instruction is processed, I/O operations are completed, and all pending interruptions are serviced. The program is run basically in the same way as during normal processing, except that the system returns to the stopped state when instruction execution is completed. The address of the next instruction to be processed is displayed in the instruction count portion of the PSW (D-register).
4. Repeat step 3 for each instruction step.
5. When instruction-step processing is completed, reset RATE switch to PROCESS.

2.17 DISPLAY LS GENERAL REGISTER

To display the contents of a general register in LS, proceed as follows:

1. Press STOP; MANUAL light goes on.
2. Set STORAGE SELECT switch to LOCAL.
3. Set ADDRESS switch 19 to center ("0") position.
4. Set ADDRESS switches 20–23 to binary address of general register to be displayed (e.g., 0000 for general register 0; 0001 for general register 1; 1111 for general register 15).
5. Press DISPLAY. The contents of the general register addressed are displayed in the T-register (roller 2, position 3). Other general registers can now be displayed by changing address switches 20–23.

2.18 LOAD LS GENERAL REGISTER

To load data into an LS general register, proceed as follows:

1. Perform the first four steps of procedure in paragraphs 2.16.
2. Set DATA switches 32–63 to generate the desired data.
3. Press STORE. The setting of the DATA switches is now stored into the selected general register. Check for accuracy by displaying the contents of this register. (This operation does not work in single cycle.)

2.19 DISPLAY LS FLOATING-POINT REGISTER

To display the contents of a floating-point register in LS, proceed as follows:

1. Press STOP; MANUAL light goes on.
2. Set STORAGE SELECT switch to LOCAL.
3. Set ADDRESS switch 19 to down ("1") position and ADDRESS switch 20 to center ("0") position.
4. Set ADDRESS switches 21–23 to binary address of floating-point register to be displayed (e.g., 000 for floating-point register 1; 001 for floating-point register 2; 111 for floating-point register 8).
5. Press DISPLAY. The contents of the floating-point register addressed are displayed in the T-register (roller 2, position 3). Other floating-point registers can now be displayed merely by changing the ADDRESS switches.

2.20 LOAD LS FLOATING-POINT REGISTER

To load data into a floating-point register in LS, proceed as follows:

1. Perform the first three steps of the procedure in paragraph 2.19.
2. Set ADDRESS switches 21–23 to binary address of first word in floating-point register to be loaded:

Floating Point Reg	Switches 21–23
0	000
2	010
4	100
6	110

3. Set first word of data to be loaded into DATA switches 32–63. (Place the exponent in byte 0.)
4. Press STORE pushbutton. Desired data is now stored in first word of selected register.
5. Set ADDRESS switches 21–23 to binary address of second word in floating-point register to be loaded:

Floating Point Reg	Switches 21–23
1	001
3	011
5	101
7	111

6. Set second word of data to be loaded into DATA switches 32–63.
7. Depress STORE. Desired data is now stored in second word of selected register.
8. Check for accuracy by displaying contents of register. The exponent is in byte 0. (This operation does not work in single-cycle.)

2.21 ADDRESS-COMPARE STOP PROCEDURE

This procedure compares the ADDRESS switch settings with the address sent either by CPU or channel to main storage. When the two are the same, the CPU enters the stopped state. (The switches in this operation are on the M panel.) Proceed as follows:

1. Set ADDRESS switches 2–20 to the desired storage address.

   Note: If the multisystem is installed, the ADDRESS switches must be set to the relocated main storage address when prefixing is active. (This is not necessary for DISPLAY and STORE functions.)

2. Set ADDRESS-COMPARE STOP switch to down (stop) position. When the address sent to main storage is the same as the address set in the ADDRESS switches, the CPU goes into the stopped state.
3. To resume processing, restore ADDRESS switches to center position, restore ADDRESS-COMPARE STOP switch to center position, and press START.

2.22 DISPLAY DOUBLEWORD FROM MAIN STORAGE

1. Press STOP; MANUAL light goes on.
2. Set ADDRESS switches to desired storage address. (This is the location of the lowest of the eight bytes to be displayed. Only doubleword locations can be addressed.)
3. Set STORAGE SELECT switch to MAIN.
4. Press DISPLAY. The contents of main storage location addressed (and the seven bytes following) are now displayed in the ST and AB registers (position 3 of rollers 1–4).

2.23 STORE DOUBLEWORD INTO MAIN STORAGE

1. Perform first three steps of procedure in heading 2.22.
2. Set all DATA switches (0–63) in doubleword configuration to be stored.
3. Press STORE. Desired data is now stored at desired address. (Check for accuracy by displaying contents of stored location.)

2.24 STORE SINGLE BYTE INTO MAIN STORAGE

1. Press STOP; MANUAL light goes on.
2. Set address of desired byte location into ADDRESS switches.
3. Set STORAGE SELECT switch to MAIN BYTE.
4. Set one byte of desired data into DATA switches.
5. Press STORE. (Check for accuracy by displaying contents of stored byte location.)

2.25 DISPLAY CAW

To display the channel address word (CAW), set the address of 48 (hex) into the ADDRESS switches, and follow the procedure in heading 2.22. (CAW is displayed in the S-register: roller 1, position 3.)

2.26 STORE CAW

To manually set a channel address word (CAW) into main storage:

1. Set desired CAW format into the DATA switches.
2. Set the address of 48 (hex) into the ADDRESS switches, and follow the procedure in heading 2.23.

2.27 DISPLAY CCW

Set the address for the desired channel control word (CCW) into the ADDRESS switches, and follow the procedure in heading 2.22.

2.28 STORE CCW

To manually set a channel control word (CCW) into main storage:

1. Set desired CCW format into DATA switches.
2. Set address of desired store location into ADDRESS switches, and follow procedure in heading 2.23.

2.29 CLEAR STORAGE PROCEDURE

The following procedure clears all main storage, as well as all general purpose and floating point registers in LS:

1. Press SYSTEM RESET.
2. Place ADDRESS switches 0, 21, and 22 in down position and all other ADDRESS switches in center position.
3. Set STORAGE SELECT switch to MAIN.
4. If Multisystem feature is installed, and processor is not in model 65 mode, disable interval timer.
5. Set roller 1 to position 2 (D-register indicators).
6. Press ROS TRANSFER; D-register will be stepping (this indicates that storage is being cleared).
7. Set STORAGE SELECT switch to LOCAL; D-register stops stepping.
8. Set STORAGE SELECT switch to MAIN.
9. Press SYSTEM RESET.

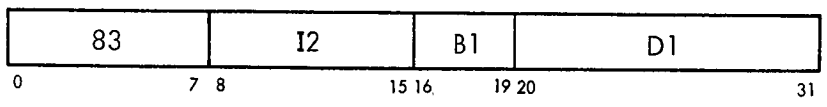
Note: Pressing SYSTEM RESET while storage is being cleared might cause invalid stores.

Section 3. Maintenance Features

This section describes the 2065 maintenance features available to the CE: (1) Diagnose instruction and associated MCW's; (2) Scan-in operations; (3) Logout, ROS tests, and FLT's; and (4) Diagnostic programs. For a discussion of marginal checking procedures, refer to Chapter 5.

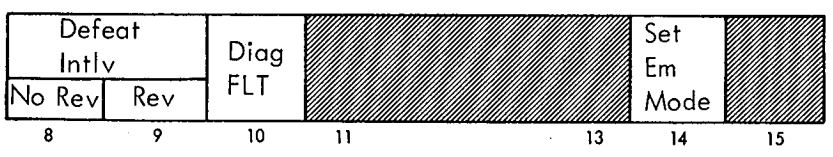
2.30 DIAGNOSE INSTRUCTION AND MCW's

The Diagnose instruction has two purposes: it is available to the diagnostic programmer as a maintenance aid and to the system programmer for emulator mode operations. The Diagnose instruction has an SI format with an op code of 83:



The operations that the Diagnose instruction can do are selected by the I2 field of the instruction and by the bit configuration of the doubleword addressed by the storage operand address (contents of GPR addressed by B1, + D1). The right half of this doubleword is the MCW, and bears the same relation to the Diagnose instruction as the CCW does to an I/O instruction. That is, the Diagnose instruction addresses the location of the MCW, and the MCW specifies the machine function. Note that an exception to this analogy is the I2 field, which can also designate certain diagnostic functions.

The bits of the I2 field have the following meaning:



1. Bit 8, Defeat Interleaving and No Reversal of Storage Addresses. Interleaving is disabled and no even/odd storage area reversal takes place.
2. Bit 9, Defeat Interleaving and Reverse Storage Addresses. Interleaving is disabled and the even and odd storage areas are reversed.
3. Bit 10, Diagnose FLT. Allows portions of FLT's to be executed under control of the Diagnose instruction. While the FLT's are being executed, special CPU functions are generated, and storage requests and clock are inhibited. This bit, which is used in conjunction with the setting of address 6B0 in ROSAR, simulates the FLT position of the TEST MODE, ROS/PROC/FLT switch on the system control panel.
4. Bits 11–13. Spares.
5. Bit 14, Set Emulator Mode. Allows the CPU to enter the emulator mode. (See Compatibility Feature FETOM's for a discussion of emulator mode operations.)
6. Bit 15. Spare.

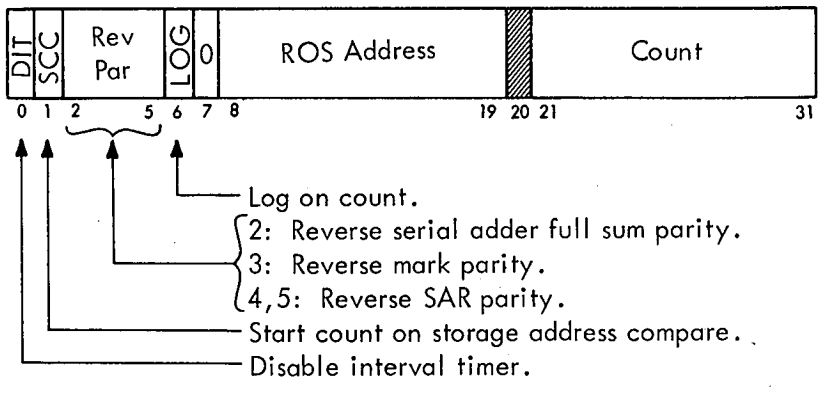
Besides the functions that can be selected by the I2 field, the Diagnose instruction can:

1. Perform channel diagnostic functions. When a channel is selected by the MCW, it is disconnected from its control units and connected to an internal (to the channel) interface simulator which consists of a one-byte register and associated controls. This simulator allows the diagnostic program to test the channel regardless of the type of attached control units that may be available.
2. Reverse the state of certain parity bits. Incorrect parity can thus be placed into the CPU or into the channel to test the error detection logic.
3. Cause a logout after a specified number of cycles (CPU only).
4. Branch into the ROS microprogram at any point.
5. Perform portions of the LS ripple tests.
6. Branch to an end-op micro-order and thus terminate normally.
7. Force PAL full-sum errors.
8. Perform scan-in operations.
9. Stop the CPU clock.
10. Read the contents of the system control panel DATA switches into storage.

The MCW is a 32-bit word used to control Diagnose-instruction and scan operations. When the Diagnose instruction is executed, the MCW is in the main storage location specified by the storage operand address and designates the diagnostic function(s) to be performed. For ROS tests and FLT's, the MCW is contained in word 1 of each test on the ROS or FLT test tape and specifies the control conditions necessary for the test, such as the expected result and the ROS word to be used for gating control. A different format is used for each MCW.

2.30.1 Diagnose Instruction MCW for CPU

When executing the Diagnose instruction and MCW(7) = 0, the MCW applies to the CPU, as follows:

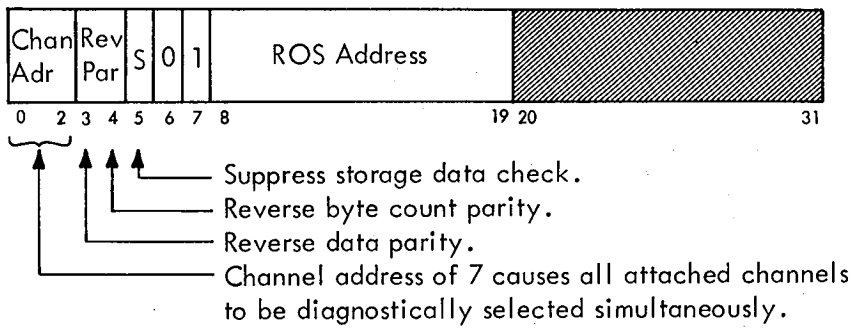


1. Bit 0, Disable Interval Timer. Disables the interval timer.
2. Bit 1, Start Count on Storage Address Compare. When used with bit 6, the FLT counter does not begin decrementing until the BCU addresses the same location as set into the MAIN STOR ADDRESS COMPARE (ADDRESS switches 2–20) switches.
3. Bit 2, Reverse Serial Adder Full-Sum Parity. Reverses the parity bit in the full-sum latch of the serial adder, thus allowing testing of the parity-checking circuits.
4. Bit 3, Reverse Mark Parity. Reverses the parity of the mark bits being sent from the BCU to main storage, thus allowing testing of the mark parity-checking circuits in main storage.
5. Bits 4 and 5, Reverse SAR Parity. Cause the parity bits which are sent to the storage address register to be reversed as follows:
   00 – No parity reversal.
   01 – Reverse low-order parity bit.
   10 – Reverse next-higher parity bit.
   11 – Reverse high-order parity bit.
6. Bit 6, Log On Count. Causes a logout to main storage when the FLT counter reaches 0. At the conclusion of the logging operation, the CPU performs a machine check interruption.
7. Bit 7. Must be a 0.
8. Bits 8–19, ROS Address. When the Diagnose instruction has completed its execution phase, these address bits are placed into ROSAR, and the operation branches to this location. The address placed into ROSAR can specify any location in ROS. Refer to Diagram 5-608, Sheet 2, FEMDM, for the most frequently used ROS addresses.
9. Bit 20. Spare.
10. Bits 21–31, Count. Specify the number of cycles (200 ns) that are to occur before the CPU enters a logout routine. The count field is made up as follows: MCW(21–25) is loaded into the address sequencer; MCW(26–29) is loaded into the FLT counter; MCW(30,31) is loaded into the FLT clock. By combining the address sequencer, the FLT counter, and the FLT clock, a maximum count of 2047 (11 bits) can be obtained. (These three counters are combined only when the MCW is used with the Diagnose instruction.) The FLT clock controls the decrementing of the FLT counter and address sequencer by 1; when the three counters combined equal 0, the logout routine is started.

2.30.2 Diagnose Instruction MCW for Channel

When executing the Diagnose instruction and MCW(7) = 1, the MCW applies to the channel. When a channel is selected by the MCW, the channel is disconnected from its control units and is connected to an internal interface simulator. The interface simulator consists of a one-byte register and associated controls. This simulator allows the diagnostic program to test the channel regardless of the type of control units that may be attached and available.

The MCW format for the channel diagnostic function is:



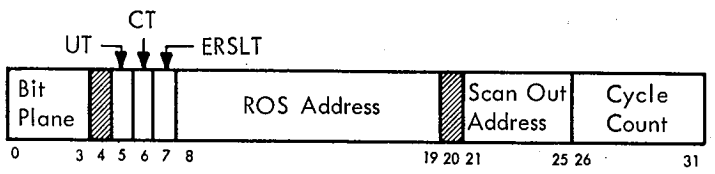
1. Bits 0–2, Channel Address. Select the channel to be diagnosed according to the following bit configuration:
   000 – Select channel 0 (multiplexer channel).
   001 – Select channel 1.
   010 – Select channel 2.
   011 – Select channel 3.
   100 – Select channel 4.
   101 – Select channel 5.
   110 – Select channel 6.
   111 – Select all channels simultaneously.
2. Bit 3, Reverse Data Parity. Causes all bytes that are read from the interface simulator to have reversed parity. This action allows testing of the storage bus-in parity check circuit.
3. Bit 4, Reverse Byte Count Parity. Provides a means of testing the byte control check circuits.
4. Bit 5, Suppress Storage Data Check. Prevents a storage data check from causing a channel data check and prevents a channel control check on a CCW fetch operation. Preventing the channel control check allows invalid CCW's to be brought into the channel to test sections of the channel check circuits.
5. Bit 6. Must be a 0.
6. Bit 7. Must be a 1.
7. Bits 8–19, ROS Address. When the Diagnose instruction has completed its execution phase, these address bits are placed into ROSAR and the operation branches to this location. The address placed into ROSAR can specify

any location in ROS. Refer to Diagram 5-608, Sheet 2, FEMDM, for the most frequently used ROS addresses.

8. Bits 20–31. Spares.

2.30.3 ROS Test MCW

The MCW for ROS tests, the right half of word 1 of each test, contains the following control information about the test:

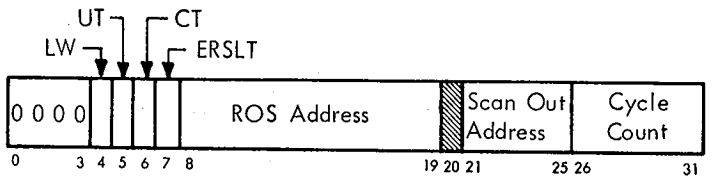


1. Bits 0–3, Bit Plane. Contain the number of the ROS bit plane tested. These bits are for display only; the ROS word is selected by the ROS address in bits 8–19, and the bit to be tested is selected by the mask.
2. Bit 4. Spare.
3. Bit 5, Unconditional Terminate (UT). If this bit equals 1, the test always causes a stop.
4. Bit 6, Conditional Terminate (CT). If this bit equals 1 and an error is encountered, testing is terminated. If, however, it equals 1 and the test does not detect an error, the CPU continues with the next ROS test. If this bit equals 0, termination is dependent upon the status of MCW(5).
5. Bit 7, Expected Result (ERSLT). Defines what the status of the ROS bit being tested should be. Comparing the ROS bit with the 1 or 0 in this bit determines whether the test passed or failed.
6. Bits 8–19, ROS Address. Contain the ROS address of the micro-instruction that contains the bit to be tested.
7. Bit 20. Spare.
8. Bits 21–25, Scan Out Address. Specify the scan-out address (roller switch position) of the portion of the ROSDR that contains the bit to be tested. This address will be either 15, 16, or 17. (See ALD M3051.)
9. Bits 26–31, Cycle Count. Determine the number of CPU clock cycles ROS must cycle to fetch the desired ROS word into ROSDR.

2.30.4 FLT MCW

The MCW in the right half of word 1 of every FLT contains control data about the test being performed. Pertinent bits of the MCW are retained in the MCW register during the running of an FLT. Other bits set the address sequencer, determine the ROS starting address, and fix the number of clock cycles the CPU will take following scan in.

Following is the format of an MCW during an FLT:



1. Bits 0–3. Contain 0's.
2. Bit 4, LW. Defines whether the left half or right half of the word to be scanned out contains the exit trigger status. If bit 4 equals 1, the left half is the desired word. If bit 4 equals 0, the right half word contains the trigger status.
3. Bit 5, UT. Always causes a stop when it equals 1.
4. Bit 6, CT. May cause a stop, depending on the outcome of the test. If this bit equals 1 and an error is encountered, testing is terminated. This bit is always set in zero-cycle and one-cycle FLT's currently in use.
5. Bit 7, ERSLT. Defines what the status of the exit trigger should be. Comparing the exit trigger state with the 1 or 0 in this bit determines whether the test passed or failed.
6. Bits 8–19, ROS Address. Contain the ROS address of the ROS word to be used for gating control.
7. Bit 20. Spare.
8. Bits 21–25, Scan Out Address. Specify the scan-out address of the scan-out word containing the status of the exit trigger.
9. Bits 26–31, Cycle Count. Fix the number of times the CPU is to cycle following scan in.

2.31 SCAN-IN OPERATIONS

The scan-in operation is the process of loading most of the CPU registers and control triggers with a predetermined bit pattern from main storage. The data-path registers are loaded by loading a word into ST, gating the contents of S (via the scan-out bus) into the parallel adder, and then into the selected register. As the data enters the scan-out bus, the original parity bits are replaced by T(0–3). To permit arbitrary parity bits to be assigned to each register byte, the ingating is done via the normal paths using ROS.

The condition for each control trigger is set by an AND using a particular bit of T (via the ST scan-out bus) and a control/timing line from the FLT control logic; e.g., T(54), in conjunction with the 'scan-in word 5' signal, sets STAT A. Special inputs are provided to reset the control triggers.

The triggers and registers that can be set by the scan-in facilities are listed on logic pages M3071 and M3081.

The Diagnose instruction may be used to perform the scan-in operations, and, in many ways, it is similar to the FLT's; e.g., the same buffer areas are used and identical sequence blocking (e.g., storage requests and CPU requests) is enabled. (The latter occurs only during the actual scan-in operation.)

In general, the procedure when using the Diagnose instruction is to set up the buffer area with a particular bit pattern and then execute the Diagnose instruction with the MCW ROS address field equal to 6B0 (hex) and with only the FLT bit (I2 bit 10) set. This procedure, under ROS control, sets the buffer information into the corresponding trigger(s) or register(s) in the CPU. Following the scan-in

operation, control is passed to a second MCW which is part of scan-in word 1. This MCW is a normal format MCW (not an FLT format MCW) and transfers to the ROS address designated and functions as the control bits indicate, if any are set.

2.31.1 Conditions for Scan-In Using Diagnose Instruction

The following conditions must be met when performing a scan-in operation using the Diagnose instruction:

1. Bit 10 of the Diagnose instruction I2 field must be a 1.
2. The ROS address in the first MCW must be 6B0 (hex).
3. The MCW must be located in main storage on a doubleword boundary.
4. No other bit in the MCW may be set.

2.31.2 Reference Data for Scan-In Facilities

The following information, including setting up the buffers, may be useful when using the scan-in facilities:

1. Buffer 1 location: 8000-807F (hex).
   Buffer 2 location: 8080-80FF (hex).
2. The buffer used is determined by the buffer 1 trigger. The state of this trigger is difficult to determine, so it is best to fill both buffers with identical information. Therefore, when scan-in operation takes place, either buffer can be used.
3. The overall scan-in buffer layout is as follows:

Location (Hex)	Scan-In Word	Left Word Data	Right Word Data
8000 and 8080	0	S-register	T-register
8008 and 8088	1	Mask bits	MCW
8010 and 8090	2	Q-register	Q-register
8018 and 8098	3	A-register	Miscellaneous
8020 and 80A0	4	B-register	Miscellaneous
8028 and 80A8	5	D-register	Miscellaneous
8030 and 80B0	6	PSW	IC
8038 and 80B8	7	E-register, R-register	External channel mask and E-register parity
8040 and 80C0	8	Test number	Alternate test number
8048-807F and 80C8-80FF	—	Unused	Unused

   Refer to logic pages M3071 and M3081 for the detailed layout.
4. For the 7080 and 7090 emulators, the test number and alternate test number are located in scan-in word 9. Refer to logic page M3081 for the data in scan-in word 8.
5. The MCW of scan-in word 1 is a normal format MCW (not an FLT format MCW).
6. Bit 62 or 63 of scan-in word 5 must be a 1.
7. When scanning into the LAR (write), the bits scanned in are inverted; i.e., a 1 results in a 0 and vice versa.
8. Any register that is not needed may be left set to 0. However, make sure that any register requiring a parity bit has the correct value.
9. If it is desirable to execute a specific number of machine cycles and then a log-on-count with a machine check, set MCW(6) of scan-in word 1 to a 1, and set in their respective MCW fields any ROS address and the desired machine cycle count.
10. If using the log-on-count facility, the count starts with the ROS address specified by the MCW of scan-in word 1.
11. It is possible to single-cycle through a scan-in sequence. (Some CPU errors in the parallel adder full-sum logic will occur during this time, but they are reset at the end of the scan-in operation.) The progress of the scan-in sequence is followed by observing the address sequencer; it is initially set to 7 and steps down to 0.
12. It is possible, by proper setup of the scan-in data, to set the desired CPU conditions and then proceed along the normal program flow. (An example would be to set one or more interrupt triggers, return to the program flow, and let the interruptions be taken normally.) Proper parity must be set for any register that is scanned into, except registers Q, R, S, and T. The ROS address that is normally used in this case is 010 (hex); the remainder of the MCW of scan-in word 1 would be 0's. Care must be taken in setting up the IC, which must address 8 bytes ahead of the instructions being scanned in or 16 bytes ahead if the IC boundary of the instruction being scanned in is 11.
13. When scan-in operation is performed with the Diagnose instruction, it is not necessary to insert the test number, alternate test number, or the mask bits in the buffer area.
14. Bit 10 of the Diagnose instruction I2 field, which must be a 1 for the scan-in operations, performs the same function as the TEST MODE switch on the system control panel when it is set to the FLT position.
15. Even though new data is scanned into E, the diagnose signal levels are held by a diagnose trigger. This trigger is reset at end op unless bit 6 of the Diagnose MCW is a 1, in which case reset is caused by the logout machine check sequence.

2.31.3 Scan-In Program Examples

The following examples demonstrate the use of the scan-in facilities: In example 1, Figure 2-1, the routine sets up three exceptional conditions at I-Fetch and then tests the priority in which they were taken. At point A, the scan-in buffers have been loaded, and execution of the Diagnose instruction is starting. In the scan-in buffer data, shown at the bottom of the figure, it can be seen that Q (scan-in word 2) is loaded with the doubleword, starting at point B, and the IC (scan-in word 6) is addressing 8 bytes ahead of point B, which would be the normal setting of the IC if this point in the program had been reached normally. R (scan-in

Storage Location (Hex)	Contents (Hex)	Pgm Tag	Symbolic Code	Comments
0500	41 F0 0 500	SETBS	LA 15,1280	Set base reg to hex
0504	82 00 F 110	STEP	LPSW INPSW	
0508	47 00 F 00C	BEGIN	BC 0,*+4	No op
050C	D2 03 0 050 F 150		MVC 80(4),EFFS	
0512	1B 00		SR 0,0	
0514	D2 07 0 058 F 100		MVC 88(8),ENPSW	
051A	D2 07 0 068 F 108		MVC 104(8),PNPSW	
0520	D2 07 0 000 F 148		MVC 0(8), RST1	
0526	58 90 F 15C		L 9,ETH	
052A	D2 3F 9 000 F 178		MVC 0(64,9),SCAN	Load scan buffer 1
0530	D2 3F 9 080 F 178		MVC 128(64,9),SCAN	Load scan buffer 2
0536	8320	(A)→	DC X'8320'	Diagnose instruction
0538	F 118		DC S(MCWS)	MCWS=006B000000000000
053A			CNOP 0,8	
053A	07 00		BCR 0,0	
053C	07 00		BCR 0,0	
053E	07 00		BCR 0,0	
0540	D5 03 0 050 F 150	EXT	CLC 80(4),EFFS	Did timer step
0546	47 80 F 046		BC 8,*	Error if equal
054A	5A 00 F 168		A 0,ONE	+1 to counter
054E	42 00 F 056		STC 0,ESV	Store count
0552	82 00 0 018		LPSW 24	
0556	00	ESV	DC XL1'0'	Save area
0558			CNOP 0,8	
0558	5A 00 F 168	PROG	A 0,ONE	+1 to counter
055C	42 00 F 064		STC 0,PSV	Store count
0560	82 00 0 028		LPSW 40	
0564	00	PSV	DC XL1'0'	Save area
0566			CNOP 0,8	
0566	07 00		BCR 0,0	
0568	47 00 F 068	(B)→SCRST2	BC 0,*	
056C	43 50 F 056		IC 5,ESV	Q reg
0570	89 50 0 008		SLL 5,8	
0574	43 50 F 064		IC 5,PSV	

Contents of Scan-In Buffers:

0000000000000000	SCAN	DC XL8-0-	Scan-in word 0
0000000000010000		DC XL8-00010000-	Scan-in word 1
4700		DC X-4700-	Scan-in word 2
F 068		DC S(SCRST2)	Scan-in word 2
4350		DC X-4350-	Scan-in word 2
F 056		DC S(ESV)	Scan-in word 2
EEEEEEEE001E0008		DC X-EEEEEEEE001E0008-	Scan-in word 3
EEEEEEEE00F00000		DC X-EEEEEEEE00F00000-	Scan-in word 4
1000000070000002		DC X-1000000070000002-	Scan-in word 5
0100000700		DC X-0100000700-	Scan-in word 6
000570		DC AL3(SCRST2+8)	Scan-in word 6
0000470000018000		DC X-0000470000018000-	Scan-in word 7

Figure 2-1. Scan-In Program, Example 1

word 7) is loaded with the tag at point B; the tag contains 4700. This tag will be gated to E by the R-to-E transfer during end op (the ROS address, 010, in scan-in word 1).

By setting up these conditions, an effective branch has been taken from point A to point B (from point A into the Diagnose instruction, which scans in register content data for point B and sets the IC with point B address to point B). Also note that program interruption triggers 1, 2, 4, and 8, the TCS trigger, and external signals 5, 6, and 7 have been set. These conditions mean that at I-Fetch of the Branch instruction at point B, a program interruption, an external interruption, and a timer exceptional condition should be pending. Note that no machine check interruptions have been taken, but a specific CPU condition has been set up and the program has been resumed.

Example 2, Figure 2-2, shows a scan-in buffer pattern loaded into main storage before execution of the Diagnose instruction. The significant point in this example is the MCW of scan-in word 1, which indicates to go to end op address 010 (hex), execute one machine cycle, and then log out the CPU status. The bottom of the figure shows the resulting logout data.

Scan-in Word	Leftword Contents	Rightword Contents
0	FFFFFFFF	FFFFFFFF
1	00000000	02010001
2	FFFF5555	FFFF5555
3	FFFFFFFF	F0000000
4	FFFFFFFF	F0800000
5	1FA5A5A5	7A002102
6	01040000	001C1C1C
7	47F09020	00018000
8	00000000	00000000

Data Scanned In

Storage Location (Hex)	Logout Word	Leftword Contents	Rightword Contents	Comments
0080	0	FDFF10FF	FAFFBFFF	D, F, ST, Q parity
0080	1	6F003FFF	FAFFBFFF	Miscellaneous; AB, R, E parity
0090	2	00000000	00000000	G; storage and LCS checks
0098	3	FFFFFFFF	80000000	STAT's; miscellaneous; external; channel
00A0	4	01040000	4C0C0000	PSW; FLT controls
00A8	5	00000000	005FBF91	MCW, scan, and FLT controls
00B0	6	FFFFFFFF	FFC4F000	Marks; ABC; STC
00B8	7	FFFFFFFF	FFFFFFFF	AB register
00C0	8	02878000	00000000	Features
00C8	9	FF470001	1C1C1C00	IC; serial adder
00D0	10	902047F0	A5A5A5A0	R, E, D, F registers
00D8	11	FFFF5555	FFFF5555	Q register
00E0	12	FFFFFFFF	00000000	PA, serial adder checks; ROS parity
00E8	13	D000F1FF	000051A0	Miscellaneous parity; LSWR
00F0	14	00000000	00000000	Gate controls
00F8	15	00000000	00021000	ROS data register
0100	16	00000000	0010010A	ROS data register
0108	17	00000000	0080E800	ROS data register
0110	18	00000000	44808050	ROSAR; PROSAR
0118	19	00000000	10100FBB	ROS; LAR
0120	20	00000000	00000000	SAR logout word
0128	21	FFFFFFFF	FFFFFFFF	ST register

Resulting Logout Data

Figure 2-2. Scan-In Program, Example 2

2.32 LOGOUT, ROS TESTS, AND FLT'S

Logout stores the status of the system control panel indicators into fixed positions of main storage when a trouble symptom occurs; the data logged out may be subsequently recalled for analysis although the status of the indicated logic is changed from what it was when the symptom appeared. ROS tests check each bit position of every ROS word. FLT's check the CPU at the logic block level.

Logout, ROS tests, and FLT's are implemented by special hardware called scan logic. Therefore, when employing these maintenance aids, the CPU is said to be in the scan mode.

The scan logic performs the following functions:

1. Controls the operation of FLT's.
2. Records the state of the CPU when a machine malfunction is detected (logout).
3. Executes the Diagnose instruction.
4. Controls the operation of the ROS tests.

2.32.1 Logout

The logout function of the scan logic stores the status of various triggers and registers, reflecting the state of the CPU, into predesignated locations of main storage. The 22 doublewords logged out (Table 2-1) are stored into main storage locations 80 through 128 (hex). (For a detailed list of the 22 doublewords, refer to ALD's M3021–M3061.)

Note: The Multisystem feature adds 16 log words (22–37, in main storage locations 180–1F8).

The status of each trigger logged out is represented by a 1 if it is set, or by a 0 if it is reset. Thus a record of the machine state, at the time that a CPU or storage error occurs, is stored unchanged in predetermined locations of main storage with a fixed format. This record can then be accessed by a program or by manual controls for analysis, printout, or display.

Table 2-1. Logout Format

Main Storage Byte Address (Hex)	Log Word	Left Half	Right Half
080	0	Parity for D-, F-, and Q-registers	Parity for ST-register
088	1	Miscellaneous controls	Parity for AB-, E-, and R-registers
090	2	G-register, LCS, and segmented clock	Storage checks
098	3	-----	Interruption, I-Fetch, and STAT's
0A0	4	PSW	BCU, FLT, and manual controls
0A8	5	-----	FLT controls, MCW, and counters
0B0	6	-----	ABC, STC, LAR, B(64–67), mark triggers, and PAL(64–67)
0B8	7	A-register	B-register
0C0	8	-----	Features
0C8	9	-----	IC and SAL
0D0	10	R- and E-registers	D- and F-registers
0D8	11	Q(0–31)	Q(32–63)
0E0	12	-----	CPU checks
0E8	13	Parity for LSWR, PSW, and IC; LSWR, PSW, IC, and checks	LSWR
0F0	14	-----	Gate control triggers
0F8	15	-----	ROSDR(70–99) and edit triggers
100	16	-----	ROSDR(37–69)
108	17	-----	ROSDR(1–35)
110	18	-----	ROSAR (3 registers)
118	19	-----	ROSAR, parity for ROSDR, LS address register, and gate control triggers
120	20	-----	Storage address, cycle, and data check(s)*
128	21	S-register	T-register

Note: PAL(64–67) and SAL(0–7) should always = 0 on logout, and the parity bits should always = 1.

*Present only when a storage check occurs and machine checks are enabled.

A logout operation can be initiated by:

1. Manually depressing the LOG OUT pushbutton on the system control panel. The system must be in manual mode.
2. Executing the Diagnose instruction when a log-on-count function is specified. Logout occurs after a predetermined number of CPU clock cycles (preset by the programmer).
3. Detecting a machine check during normal CPU operation if the CPU CHECK switch is in the PROC position and the PSW machine check mask bit is on. When a storage error is detected and the CPU is enabled for machine checks, an additional word (log word 20) is logged out. This word, from the roller switch associated with the storage in error (position 5 on rollers 1, 2, 5, or 6), contains the information, excluding storage protect indicators, that is displayed in the roller switch indicators.

2.32.2 ROS Tests

ROS tests are the principal means of testing the validity of the ROS bit planes. These tests, generated by a computer program from the tapes used in the manufacture of the ROS bit planes, are stored on magnetic tape or disks. When testing ROS, the tests are read into two main storage buffer areas, starting at locations 8000 and 8080 (hex). Under scan logic control, the ROS tests compare the value of a particular bit in a selected ROS word with its expected value as specified on the ROS test tape. Each ROS test is read into the CPU and checks one bit of a ROS word. At the same time that the test from one buffer area is being executed in the CPU, the other buffer area is being filled from the test tape via the channel.

The ROS test format consists of two doublewords: word 0 and word 1. Word 1 contains the mask and MCW. The mask, a 32-bit field, in conjunction with the scan-out address field of the MCW, selects the ROS bit to be tested from the word read out of ROS. A particular bit is tested by making all mask bits 1's except the bit that corresponds to the test bit. Word 0 contains the test number (TN) and alternate test number (ATN). The TN, a four-byte field, contains two two-byte numbers that identify the test pattern. The lower-order two bytes are the complement of the TN, and the high-order two bytes are the TN. The ATN, which refers to another ROS test, is also represented in true and complement form, with the complement and true numbers reversed from that of the TN format. The TN refers to the test being executed. The ATN refers to the test that will be executed if the tests are restarted after a failure stop (generally, it refers to the next test).

The first part of each ROS test tape contains hardcore tests to establish that the CPU is able to run ROS tests.

Testing should not proceed beyond the hardcore tests if failures are encountered.

Following the hardcore tests are the actual ROS tests. During these tests, the ROS word to be tested is selected by the ROS address in the MCW. The CPU clock is allowed to generate clock signals to cycle ROS so that the bit under test is placed into the ROSDR. The word containing this bit is defined by the scan-out address in the MCW and is transferred (scanned out) via the indicator driver logic and PAL to the T-register.

The mask is transferred from main storage to the S-register, and then the status of the bit under test is determined by comparing S with T. The result of this comparison is compared with the ERSLT bit to determine whether the test passed or failed. Pass, fail, and intermittent-fail are the three possible results of the comparison. The CT and UT bits in the MCW then determine whether to proceed with testing or to terminate.

If testing is terminated on a failure, the last two hex digits of the ROS address of the failing word are displayed in S(0–15) (roller 1, position 3), and the bit plane is displayed as the CPU TEST ADDRESS on roller 5, position 2, bits 0–3. To further define the S-register contents, S(0–7) indicates the failing word in hex, and S(8–15) indicates the failing bit in decimal.

The test is repeated until a transfer in channel (TIC) pulse notifies the CPU that the alternate buffer is filled and new test data is available.

Because FLT's cannot be run if malfunctions exist in ROS, the ROS tests should be run first, followed by the FLT's.

2.32.3 FLT's

FLT's are a unique maintenance concept in that CPU logic is checked without executing CPU instructions (i.e., without executing a program in the ordinary sense). In an FLT, fault detection is performed at the logic level. That is, FLT's are concerned with the logical function (OR, AND, INVERT) of a block rather than with its operational function (e.g., as an adder, counter, control) in the CPU.

FLT's can be divided into three categories:

1. Hardcore tests. Check the scan and normal CPU logic necessary to run FLT's.
2. Zero-cycle tests. Determine whether a trigger value can be changed by scan in and also whether the new value can be sensed. Zero-cycle tests establish the machine capability to scan in and scan out before running one-cycle FLT's.
3. One-cycle tests. During these FLT's, data is scanned into the CPU, the clock is allowed to run, and the exit trigger is scanned out and compared with a known value. One-cycle tests check combinational logic within the CPU.

Note that the terms "zero-cycle" and "one-cycle" do not refer to the number of clock cycles allowed after scan in.

2.32.3.1 Hardcore Tests

The first eight records on the FLT tape contain tests that ensure that the hardcore logic necessary to run zero-cycle and one-cycle tests is operational. Hardcore logic is defined as all the logic, scan and normal, necessary to load the FLT's into storage, to scan in, to scan out, to make decisions based on the outcome of a test, and to act on those decisions.

The CPU hardcore hardware testing is subject to the following limitations:

1. Scan hardware is not tested directly; it is, however, exercised in the hardcore portion with tests designed to isolate the trouble.
2. The BCU is not tested in the FLT's, but the BCU must be functioning properly to run FLT's. Also, main storage and storage buses must be operating. Main storage may be checked using ripple tests.
3. The channel and tape drive (or disk) used to read in the FLT's are not tested. These units must be tested using manual controls on the channel.
4. Local storage is not needed to run FLT's, and, therefore, is not tested. Local storage is checked by ripple tests.
5. The ROS microprogram and ROS must be fault-free to run FLT's. Therefore, because hardcore tests do *not* check ROS, ROS tests should be run before FLT's.

2.32.3.2 Zero-Cycle Tests

To further check the operation of the scan hardware, zero-cycle tests determine whether the scan-in and/or scan-out paths are operative. Zero-cycle tests check only those triggers displayed on the system control panel. In these tests, a pattern is scanned into the machine, the clock is not advanced, and the exit trigger is observed. If the trigger has a scan-in path, three tests are performed: one for the reset state, one for the set state, and one for the reset state again. If the trigger has no scan-in path, only one test is performed for the reset state. While one trigger is being tested, other triggers can assume various states. Whenever possible, random states are used to simulate the combinations that may be used in a normal test and to reveal interaction between triggers.

The functions checked with zero-cycle tests include:

1. Reset to triggers (this is, in effect, a scan-in-zero operation).
2. Scan-in path to triggers.
3. Ability of a trigger to hold its value in the absence of a clock signal.
4. That scan-out bus gating signals can be generated.

2.32.3.3 One-Cycle Tests

One-cycle tests are the true FLT's. The input, produced by the FLT generating system, is a test that detects and locates at least one fault. The input pattern is scanned into selected triggers, and the CPU is allowed to advance a given number of cycles. The result, which is in the exit trigger, is compared with the value expected for a correctly operating machine.

If the value in the trigger does not agree with the expected result, testing is terminated and the failing TN is displayed in S(0–15) for reference to the SCOPEX.

The SCOPEX is a series of lists, one for each test. Each list is headed by the test number in hex, followed by a row of asterisks, and consists of several lines, each line referring to a pin in the machine. If a pin in the machine is contained in a list, the net which feeds that pin is sensitive for the test pattern applied; a failure on the card, which can be observed with an oscilloscope at that pin, would cause the test to fail.

2.33 DIAGNOSTIC PROGRAMS

The Diagnostic Program System for the Model 65 comprises many programs, each of which has unique functions and uses. The principal part of the Diagnostic Program System consists of a group of sections written specifically for the Model 65 ('6' prefix series) which, when operated in the specified sequence, provide the most effective means of CPU and system error detection and isolation. In addition to the specific 2065 sections, there are sections general to some or all System/360 Models. Most of these are unit oriented. All sections are run under control of a Diagnostic Monitor, DME. A typical set of diagnostic programs for the System/360 Model 65 is listed in Table 2-2. (A diagnostic program system master tape format is shown in Figure 2-3.)

For more detailed information on the System/360 CPU and functional unit diagnostic programs, refer to:

1. *System/360 Diagnostic Program* General Reference Manual (PN 5396300).
2. *DME User's Guide* (PN 5763442).
3. The program writeup, object deck, and listing provided with each diagnostic program.

The Model 65 diagnostics provide a systematic testing of the hardware, beginning with the most basic operation and building up to complex instructions and test procedures. The diagnostics are designed to detect and isolate machine malfunctions, whether continual or intermittent. To obtain the highest degree of isolation, the diagnostics must be used in the manner and sequence they are designed for. Since FLT's provide the best means of isolating continual CPU failures, the FLT's should be run before the diagnostics. The first diagnostic program to operate is the Hard Core (not to be confused with the hardcore logic hardware used to run ROS tests and FLT's) or pre-DM test. Hard Core is designed to operate as a stand-alone program, independent of the DM. It is assumed that only the logic needed to bring in Hard Core (IPL logic) must be working. Hard Core proceeds to test all instructions used by DME plus those necessary for any diagnostics to use a DM. Basic tests of interrupt logic are included. A successful pass of the Hard Core program automatically loads and transfers control to DME. (Hard Core expects to be followed by DME from the same input unit.) Upon loading, DME first halts in the Wait state. At this point, DME and the remainder of the diagnostic programs can be operated. All Model 65 diagnostics that operate under DME provide self-explanatory printouts and looping options.

Start

Hard Core	DME	Utility	MCK Analysis Program	Interrupt Test	Basic CEDA	CEDA 1
601F	F040	FOFF, FLUT	6108	6105	6109	610A

CEDA 2	CEDA 3	CEDA 4	CEDA 5	Model 65 Log Out Area Test	Reliability Test
610B	610C	610D	610E	6115	E330

End

Model 65 Diagnose	Timer and Meter Test	2365 Memory	2365 Memory Error Check	Storage Protect Test	Channel and I/O Programs	SIP
6347	E381, E382, F38F	E3A1	E3A2	E3CB	E431-F902	EFC0-EFDF

Figure 2-3. Diagnostic Program System Master Tape Format

Table 2-2. Typical Set of Diagnostic Programs for System/360 Model 65

Diagnostic Program Number	Description
601F	Hard Core
F040	DME, Diagnostic Monitor
6103	ROS Test
6105	Interrupt Test
6109	Basic CEDA (CEDA 0)
610A	CEDA 1
610B	CEDA 2
610C	CEDA 3
610D	CEDA 4
610E	CEDA 5
610F	CEDA 6
6115	Model 65 Log Out Area Test
6347	Model 65 Diagnose
E38F	Elapsed Time Meter
E3A1	2365 Memory
E3A2	2365 Memory Error Check
E3CB	Storage Protect Test
E3E2 and E3E3	2361 Storage
E431–E437	2870 Multiplexer Channel
E450–E457	2860 Selector Channel
F501–F509	Tape Functional
F519–F51B	Tape, CRC, and Error Check
F521 and F522	Tape IRG
F54A	2816 Tape Switch
F600–F609	2841 Storage Control for 2311
F610–F613	2311 Disk
F620–F629	2841 Storage Control for 2302
F630	2302 Disk
F680–F684	2841 Two-Channel Switch, 1 CPU
F688–F68C	2841 Two-Channel Switch, 2 CPU
E691–E697, E699, and E69B	2820 Drum Storage Control
E698	2820 Two-Channel Switch
E69A	2301 Drum Surface Analysis
F6FE and F6FF	2973-2 Disk Switch
F700–F709	2260 Local
F750–F75B	2250-1 Unbuffered
F760–F76C	2250-1 Buffered
F770–F773, F77A, F77C–F77E, F780–F782, F786–F788, and F78C	2840/2250 Display and Control
F7F0	2814 Display Two-Channel Switch
F804–F806, F808, F809, F80C, and F80D	2821 Control Unit, 1403 and 2540
F860–F864	2821 Two-Channel Switch, 1 CPU
F868–F86C	2821 Two-Channel Switch, 2 CPU
F810, F811, F815, F820, F821, F823, and F825	2540 Reader-Punch
F830–F832, F836, F838, F839, and F83F	1403 Printer
F900–F902	1052 Printer-Keyboard, 2150 Console
FDB0–FDB7	2260 Remote
6EA0–6EA3	Multisystem Feature
6F21–6F2B	7074 Emulator Diagnostics
6F31 and 6F32	7080 Emulator Diagnostics
6F41 and 6F42	7090 Emulator Diagnostics
6F81	SEREP 65 (Utility)
EFB0	SIP Control
EFC3	SIP, Instruction Mix
EFC4	SIP Matrix
EFC5	SIP Memory Sort
EFC6	SIP Binomial Expansion
EFC7	SIP Random CPU
EFD0	SIP 2400 Tape
EFD1	SIP Multi-File
EFD4	SIP 2701
EFD5	SIP 2702
EFD7	SIP DASD1
EFDA	SIP 2250
EFDB	SIP 2260
EFDC	SIP 2821
FFDE	SIP 1443

One only

Note: For current diagnostic levels, refer to System/360 Diagnostic Service Aids.

In addition to the basic interrupt tests run in Hard Core, tests of the logout function and interrupt logic are performed, including a test of interruption priority and sequencing on multiple interruptions. These tests use scan-in facilities under control of the Diagnose instruction. The interrupt test uses only instructions which have been previously tested by Hard Core.

Thus far the testing has been for basic and continual failures. A complete test of all CPU instructions follows the interrupt test. These programs, called CPU Error Detection and Analysis (CEDA), start with basic tests and build up to a point where instructions are used to solve complex problems. These CPU programs detect and isolate intermittent errors. Isolation to the ROS word is accomplished by the Program Micro Trace (PMT) program.

Upon a successful pass of the CEDA programs, the validity of the CPU is ensured. To complete CPU testing, the Logout, Diagnose, and Storage diagnostics are run. A storage diagnostic, designed to detect and isolate intermittent storage errors, is available to supplement the standard storage diagnostic. This program is not necessarily run during normal maintenance but is useful for known customer problems that are not detected by other means.

To complete testing of the units, all I/O adapter and unit diagnostics are run. Upon successful completion of these diagnostics, the validity of the complete system, as a system, is ensured, using the Systems Integration Program (SIP). Successful operation of this program provides considerable confidence regarding total system performance. The SIP should be run a minimum of 15 minutes to obtain best results; an extended run of an hour or more is advisable when time permits.

All diagnostics written specifically for the Model 65 are hardware-dependent and are likely to be affected by engineering changes. The listing for each program section defines the EC level of the hardware required to operate that specific section.

2.33.1 Hard Core

The Hard Core program (program number 601F), a basic test to validate DM operation, operates as a stand-alone program, using a minimum of logic. Assuming that only the logic that is necessary to load this program and to give it control is working, the program proceeds to test all instructions used by the DME, plus those necessary for any diagnostic to use a DM. Errors are indicated by entering the Wait state, thus requiring the CE to look up a displayed address to determine the nature of the error. A loop or a continue option is provided. Comments in the listing provide the CE with likely starting points for isolating errors. The Hard Core performs a basic test of the primary input unit and calls in DME to run the rest of the diagnostic programs.

The Hard Core contains special routines to supplement the ROS programs by checking all micro-orders connected with the J and K fields of ROS. This scheme ensures that the CPU is able to branch to the correct ROS word when ROSAR is modified by the CPU environment. Also included is a basic test of whether the Model 65 interruptions and exceptional conditions can be recognized on an individual basis. In most cases, these conditions are set up by the scan-in facility, using the Diagnose instruction. Where necessary or practical, the normal setup of these conditions is used. The following conditions are tested:

1. Machine check interruption.
2. Supervisor call interruption.
3. Program interruptions.
4. External interruptions.
5. Timer-at-limit exceptional condition.
6. Timer-step exceptional condition.
7. Program-store-compare exceptional condition.
8. Invalid-instruction-address exceptional condition.

Throughout the program, the CLD blocks and micro-orders being tested for the first time are printed in the listing with the appropriate test. This printout does not necessarily indicate that a particular CLD block or micro-order is causing the error, but it is an attempt to narrow the field of search and provide a starting point.

The following instructions are tested by Hard Core; they appear in the approximate order in which they are tested, top-to-bottom, left-to-right (No instruction is used before it is tested.):

LPSW	CR	CH	OI	SSM
BC	A	IC	TM	ISK
CL	S	STC	MVI	SSK
LA	AR	STH	CLI	TCH
ST	SR	BCT	SRL	TIO
L	OR	BCTR	SLL	SIO
LR	O	BALR	LM	
NR	N	BAL	STM	
BCR	LH	NI	SVC	

2.33.2 Diagnostic Monitor E (DME)

A Diagnostic Monitor (DM) is a control program that provides:

1. A set of standard operating procedures.
2. A standard means of communication with the CE (via the system control panel and/or the console).
3. A standard method of communication between the diagnostic program and the DM.
4. Common subroutines to diagnostics for print, code conversion, debugging aids, and other control functions.
5. Relief for diagnostic programs of the burden of adjusting to various user environments.

The DM for the Model 65 is DME (program number F040). (DME is similar to DMA8.) The DME provides additional functions, namely:

1. 16-bit I/O addresses for use by diagnostic sections.
2. The ability of a single CPU to address and to mask more than seven I/O channels.
3. Provides the sections with information that enables them to correlate the different paths to an I/O device.
4. 2250 Display Unit feature.
5. Expanded English printouts.
6. Diagnostic program debugging aids.

The DME is loaded by an IPL operation or automatically upon successful completion of Hard Core. Except for the SIP sections, only one section at a time is loaded for execution, and no I/O operations are overlapped with section execution.

2.33.3 Machine Check Analysis

The purpose of the Machine Check Analysis (MCK) program (6101) is to format and print an "Error Summary" of the logout information which accompanies a machine check interrupt on a Model 65 Processor.

To utilize existing F0FF and machine level control programs, this program has been assigned an identity of 6101. (It replaces the existing F101 section of the "F" series. The functions accomplished by the replaced F101 section will be incorporated as part of a new section.)

The format used to present the Error Summary information is illustrated in the following example:

*** MODEL 65 MACHINE CHECK ERROR SUMMARY ***

STG ADR CKS	0000 0000	STG DATA CKS	0000 0000	STG PROTECT CKS	0000
LCS UNIT CKS	0000	LCS SAR/MRK	0/0	LCS SPF/SDR	0/0
PAR ADD HS CK	0000 0000	PAR ADD FS CK	0000 0000 0	SER ADD HS/FS CK	1/0
ROS PAR CKS	000	TIME CLK STEP MACH CK	0		
E REG PAR CK	0	H REG PAR CK	0	MPR DECODE PAR CK	0

When a machine check interrupt occurs, this program scans the logout area and formats the error indicators for an Error Summary Printout. The Error Summary is developed one line at a time. After a line is formatted in core storage, the program requests the control program to print that line. When the print operation is complete, the next line is formatted. This sequence is repeated until the entire Error Summary has been formatted and printed. This format is followed by the normal DME HEX dump of the logout.

The MCK program is initially loaded via the normal DME loading procedures. Once loaded it remains in core until the machine check occurs. DME then passes control to this program for a printout of the machine check logout area.

2.33.4 Interrupt Test

The Interrupt Test (program number 6105) checks whether the Model 65 interruptions and exceptional conditions can be recognized and handled in the proper priority. In most cases, these conditions are set up by means of scan-in facilities, using the Diagnose instruction. Where necessary or practical, the normal setup of these conditions is used. A complete description of test follows.

It is initially assumed that Hard Core runs that ensure interrupts can be taken individually. The first three routines are related and work in a building block fashion.

Routine 01 simply forces each of the following conditions and checks that the correct forced address is brought up. They are TCS, PROG, SVC, EXT, I/O, WAIT, PSC, INV ADD, and STG PROT. In Routine 02, these conditions are combined in pairs (STG PROG is not included in Routine 02) to see if the higher priority address is forced. Then Routine 03 uses from two to five of these conditions in five different combinations to test for priority being correct and for certain conditions such as Prog-SVC to be reset by an MCK. The five groups are shown below.

1	2	3	4	5
TIMER	TIMER	TIMER	TIMER	TIMER
MCK	MCK			PSC
PROG	SVC	PROG	SVC	
TCS	TCS	TCS	TCS	
IO	IO	IO	IO	

If an error does occur in Routine 03, the necessary data is printed and then the program will try to determine which condition or conditions are causing the failures. Each condition is eliminated one at a time until a successful pass is encountered; then the eliminated conditions are reinserted, and, as long as no failure occurs, are left in. At the end of this test, the condition(s) that had to be eliminated are printed out. For best results any errors in RT01 and RT02 should be corrected before running Routine 03.

Routines 04–07 check whether the interrupt code in the old PSW contains the correct information. The routines test External, SVC, Program, and IO respectively.

Routine 08 checks special case program interrupts and cancel pulse operations.

Routine 09 is basically the same as Routine 03 but it allows the operator the option of selecting the exceptional conditions he wants via the console data keys. Any combination of the following conditions may be used (The printout is the same as Routine 03 except that the old PSW's are not printed out.):

TIMER STEP, MCK, PROG OR SVC, EXT, IO

Routine 10 checks the MCK and external mask bits in the PSW to inhibit respective interrupts when they are zero.

Routine 11 tests the interrupt priority of all available channels. The interrupt request priority should not be confused with data request priority: the interrupt request priority is 0, 1, 2, 3, 4, 5, 6; the data request priority is 1, 2, 0, 3, 4, 5, 6.

Routine 12 is a test which generates conflicts between all available channels and the CPU. All channels used will be in diagnose mode, doing writes from storage, while the CPU does fetches and stores. The only indication of an error will be a hang condition either in CPU or channel or an unexpected interrupt.

Routine 13 is a program interrupt test using actual instructions to generate the interrupt conditions instead of scan-in as in some of the other routines. If a particular interrupt has a PSW mask bit associated with it, the interrupt will be tried with and without the bit present to check for proper operation.

Routine 14 performs branch instructions, with the branch address being an invalid address. Each of the three branches will be done directly and then via an execute instruction. The three branches used are BC, BAL, and BCT.

2.33.5 CPU Error Detection and Analysis (CEDA)

Although FLT's isolate a CPU failure, such as a faulty module or a wiring error made during an engineering change, and a continual machine failure, such as an incorrect transfer path or a faulty register bit, they sometimes fail to isolate an intermittent failure. The primary purpose of the CPU Error Detection and Analysis (CEDA) program (program numbers 610A through 610E) is the isolation of intermittent failures. Continual failures are isolated using CEDA 0 (6109). CEDA 1-5 exercises the CPU using groups of instructions to create an intermittent failure and then isolates the failure.

Isolation is first accomplished to the instruction level; further isolation results from the CE's interpretation of the ROS analysis tables presented by the Program Micro-Trace (PMT). The use of random data in conjunction with the various instructions allows isolation of data-dependent intermittent failures for CEDA 1-5.

CEDA provides a complete test of CPU instruction fetching, decoding, and execution hardware, including the checking of all CPU data paths. This test is accomplished by an instruction chain that consists of the following combinations of instructions: the instruction under test repeated 16 times with different data each time, the instruction under test interspersed with previously tested instructions, and the instruction under test with specific conditions set up to force normal interruption conditions. The failing instruction is isolated from the chain and a ROS analysis is performed to further isolate the failure.

CEDA contains six sections, a basic section and five instruction chain sections. The order of testing instructions is determined by their complexity, section 5 having the most complex instruction. Instruction simulators are used periodically throughout all sections.

The following instructions are not tested by CEDA: Start I/O, Halt I/O, Test I/O, Test Channel, Write Direct, Read Direct, and Diagnose.

2.33.5.1 Instruction Isolation

The failing instruction isolation routine is entered whenever an error is encountered during execution of the instruction chain. This routine isolates to the instruction as follows:

1. Execute the instruction chain from the beginning through the eighth instruction (assuming that the maximum of 16 instructions are in the chain).
2. If step 1 fails, the instruction chain is divided in half and rerun, starting from instruction 1. This operation continues until only one instruction remains or until the remaining chain is successful. If the remaining chain runs successfully, the chain, plus one-half of the previous reduction, is executed. This procedure continues until one instruction is isolated.
3. If step 1 ran successfully, the instruction chain is increased to the maximum number (in this case, 16) and retried. Isolation to one instruction is then the same as in step 2.
4. If step 1 ran successfully (which means instructions 1–8 did not fail) and step 3 ran successfully (which means instructions 9–16 did not fail), an error message is printed to the effect that, although an error was encountered in the initial run, the isolation routine was unable to reproduce it and isolate it.

This routine, although used only to isolate 1 of 16 instructions, can isolate from any length instruction chain that is a multiple of two. Upon isolating to a failing instruction, the routine generates a check or proof of this instruction before reporting the instruction as a definite failure. The reasons for this approach are as follows:

1. If, within a specified number of tries (10,000 at present), a specific chain length did not produce a failure (but should have), the routine assumes a successful pass and continues its analysis. Therefore, subsequent test failure indications could be misleading.
2. Because execution of the instruction chain always starts from the first instruction, the instruction before the failing instruction will run successfully if the isolated instruction is the true failing instruction.
3. Therefore, upon isolation of an instruction, execution of the chain from instruction 1 through the instruction previous to that isolated is retried and checked for valid operation. If the execution is successful, the isolated instruction is the true failing instruction and is reported as such. But if the execution fails, the chain is reduced by one instruction and is retried. This procedure continues until a successful run of the instruction chain, and 1 is then added to this instruction chain to identify the true failing instruction, which is reported.

This routine, after isolating the true failing instruction, presents to the CE the failing instruction and its position in the chain, the original operands, and the expected and received data. The CE has the option of continuing with the program, looping on this test, or going into an Intermittent Micro-Trace Analysis routine to further isolate the failure.

2.33.5.2 CLD Isolation

The Intermittent Micro-Trace Analysis routine is entered whenever an instruction has been isolated as failing within the chain. The routine uses the PMT (discussed below) as a nucleus. The intent is to isolate intermittent failures to an exact microword.

Entry into this routine initiates a microtrace of the failing instruction, using the log-on-count feature of the

Diagnose instruction. The count is set to log out after the first ROS word of the failing instruction. The logout data is saved in a predetermined area. The failing instruction is then repeated (10,000 times at present), and each logout is checked against the first result.

If the comparison is successful, the counter of the first result is stepped by 1; if the comparison is unsuccessful, the new logout is saved. The action continues with each logout being checked against the previous results until the count goes to 0, at which time the log-on-count is increased by 1 and the process repeated. This procedure allows a ROS history table to be generated and presented to the CE. The table contains the logout received and the number of times it was received.

A message is printed to signify when analysis has begun, because the analysis may be lengthy (5 to 10 minutes). The microtrace is stopped after five errors on the same logout cycle or after the instruction is completed.

The program allows the isolation of those data-dependent or sequence-dependent failures that are difficult and time-consuming to locate.

2.33.5.3 Program Micro-Trace

The Program Micro-Trace (PMT) is an integral part of CEDA and is the basic means of error isolation. There is also a stand-alone (IPL) version of the PMT (program number 6345) that isolates failures detected by diagnostics other than the Model 65 disgnostics.

To observe the operation of the system at the micro-instruction level, turn the RATE switch on the system control panel to SINGLE CYCLE, thus placing the system in the single-cycle mode. When in this mode, each depression of START causes execution of one micro-instruction. The roller switches indicate the effect of that micro-instruction on the internal data. Repeating this process leads to a trace on a cycle-by-cycle basis.

This method of tracing instructions is possible because, with few exceptions, the data available to the roller switches is also available in the logout area. Instructions may therefore be traced by forcing a logout operation after each cycle of the instruction and then editing and printing out the contents of the logout area.

Three types of errors are handled by the PMT: type 1 errors continually occur in both single-cycle mode and at machine speed; type 2 errors occur at machine speed but not in single-cycle mode; type 3 errors occur intermittently at machine speed. To determine the type of error, loop on the instruction sequence that caused the error and see if the error recurs. The error will be either continual or intermittent. If it is continual, single-cycle through the instruction. A continual error in single-cycle mode (type 1) may be readily isolated. If the error is intermittent or does not occur in single-cycle mode (type 2), the PMT will permit it to be reduced to a type 1 error. The PMT allows these errors to be isolated faster than manual single-cycling because the PMT records the system conditions at the time of each error and provides the records for off-line analysis. Note that this procedure permits system operation with programs not affected by the errors.

Type 3 errors, which are the most difficult to isolate manually, are isolated by the intermittent error PMT sequence. This operation provides a record of the system conditions at the time of the error as well as an error-free record. Basically, the following occurs:

1. A search is made for the first cycle of the failing instruction, and the cycle count is then stopped. The system conditions are then logged out.
2. The first logout of the first cycle is printed out. This logout is used as a standard for subsequent logout operations of the same cycle.
3. The PMT then performs a logout operation on cycle 1, arbitrarily 10,000 times, each time comparing the new logout with the standard logout.
4. If, after 10,000 logout operations, no discrepancy is detected, the cycle count (in the MCW cycle counter) is stepped by 1, causing a logout operation in cycle 2 of the failing instruction for the next 10,000 passes.
5. The process is repeated until either a discrepancy is detected in the second or subsequent cycles or until all cycles of the failing instructions have been tested.
6. If a discrepancy is detected, it is printed out and saved with the standard logout of the cycle. All following logout operations of that cycle are compared with these two logouts, and, if an additional discrepancy is found, it is also printed out and saved.
7. After 10,000 passes of the cycle containing the error have been executed, a summary statement is printed, showing how often the standard and each of the discrepancy logouts occurred. In most cases, the logout with the highest count is the error-free logout, whereas logouts with the smaller counts contain the error. Note that a discrepancy logout may be the error-free logout. Differences in the logouts are signified on a word basis with asterisks.

2.33.5.4 Extended Print and Error Utility

The Extended Print and Error Utility program prints both error and operational messages for the CE and provides operational flexibility within some of the diagnostic programs. The utility is incorporated in the diagnostics that are unique to the Model 65 CEDA programs to provide consistency between the messages.

This utility makes available to the CE several program or section sense switches that are additional to the standard DM sense switches. The section sense switches are located in the section preface at the beginning of each program. They may be set by (1) manually storing the set information with the DATA and ADDRESS switches and the STORE pushbutton, (2) a message via the data entry keyboard, or (3) the DATA switches when section sense

switch 7 is set. When the DATA switches are used, DATA switches 0–31 correspond to the 32 section sense switches.

To use the DATA switches without a manual store operation, section sense switch 7 must first be set either by manual storage of the set information or by entry of a set section sense switch 7 message on the data entry keyboard. With section sense switch 6 set, DATA switches 0–31 are continually read by the program; however, for the program to move them into the section preface for program use, DATA switch 63 must be set. This switch establishes the validity of the setting of DATA switches 0–31. The validity switch permits the CE to set DATA switches 0–31 for the desired functions and then have the program automatically move the new information into the section preface when DATA switch 63 is set (section sense switch 7 having been previously set).

Any section sense switch, once set, continues to perform its function until specifically reset. The following defines the function of these switches:

Sense Switch	Position Function: Set	Position Function: Normal (reset)
0	Scope loop on error	Bypass scope loop
1	Bypass PMT	Perform PMT
2	Print long format of PMT	Print short format of PMT
3	Loop or lock on error	No loop or lock on error
4	Multi-instruction trace from address B through C	Single instruction trace from address C
5	Bypass intermittent mode	Use intermittent mode
6	Spare	Spare
7	Read DATA switches (via Diagnose instruction)	Ignore DATA switches
8–31	Unique section sense switches as defined by the program writeup; not used in the Model 65 CEDA programs.	

Section sense switches 0 and 3 loop the failing chain with the data that caused the failure. Switch 3 is interrogated upon detection of the first error. If the error is continual and the switch is set, a program loop is provided. If the error is intermittent, a lock-on-error loop is provided. Switch 0 should be set after an error has been detected using switch 3. A short program loop is provided, all printing is bypassed, and no other switches are interrogated. With section switch 0 off, interrogation of the other section sense switches is again provided.

Section sense switch 1, when set, omits the microtrace of the failing instruction. Section sense switch 5 causes PMT to log out each machine cycle of the failing instruction several thousand times, searching for the intermittent failure, unless the switch is set. It it is set, the failing instruction is traced only once.

2.33.6 Model 65 Logout

The Logout Program (program number 6115) primarily checks the validity of the information that is recorded in the logout area. The program uses the log-on-count feature of the Diagnose instruction exclusively. Following the occurrence of the forced logout in each test, the logout area is compared with a set of predetermined constants for the particular test, and any errors are indicated.

In some routines, the counts are chosen for a particular reason, such as logging out during an MVC instruction just before or just after the IC is stored in the LSWR. The selected counts are useful in troubleshooting certain problems and may constitute a secondary use of the program.

2.33.7 Diagnose Test

The Diagnose program (program number 6347) has two purposes: to check the CPU features of the Diagnose instruction and to check the error detection circuits in the CPU.

The following CPU features are tested:

1. Reverse mark parity.
2. Reverse SAR 1 and SAR 2 parity.
3. Reverse serial adder full sum parity.
4. Log-on-count: verifies the count placed in the MCW and provides the proper number of cycles.
5. Log on address compare.

The following error logic is tested:

1. Multiply decode.
2. E-register parity 0–7 and 8–15.
3. Parallel adder full sum 32–39, 40–47, 48–55, 56–63.
4. ROS parity.
5. Serial adder half sum.
6. Parallel adder half sum for bits 32–63.

2.33.8 Storage Diagnostic

This program (program number E3A1) tests the 2365 Processor Storage when it is attached to Model 65. The storage unit is tested by means of various standard data patterns, including all 1's and/or 0's and delta noise patterns. Special techniques, such as a random data/random instruction routine and relocation of the exercise routine into the tested array, are also used to make the test more stringent.

The first errors are printed as they occur, using an immediate print format. At the end of the run, the error data is summarized and broken down according to the logic of the storage unit. This data is then analyzed to discover patterns that are unique to a given circuit failure.

Provisions are made for shmooing, for selecting the even or odd storage area, and for loading into a selected storage area(s), using the DEFEAT INTERLEAVING switch. The program uses the DM Storage Definition Table to automatically configure itself to a particular storage configuration.

2.33.9 Intermittent Storage Error Detection

This unique storage program (program number E3A3) uses random data while simulating customer operation. The program locates intermittent storage errors that go undetected by normal program means.

Errors are detected by means of hardware parity detection. No comparisons are made between correct and incorrect words; thus, no slowdown occurs in storage testing.

Errors are isolated by means of a program-generated Error Correction Code which is attached to each random data word used, thus isolating the failing bit and address.

Testing is accomplished by using all randomly generated data words. Specific addresses, blocks of contiguous storage, or the entire storage may be tested at one time. The CE has the option of a test mode of operation. SORT routines, such as Exchanging, Pair Interchanging, and Shifting, are used to work storage in a manner similar to customer operation.

Addresses may be determined by the CE via the ADDRESS switches or may be randomly established from a generated random number. Either choice is used to establish a block of 1024 words with an origin at the nearest 4096 boundary that encompasses the chosen address.

The program occupies a maximum of 3400 bytes of core storage, and can be relocated into the odd storage area (upper) when the troublesome address is in the even storage area (lower).

The permanent storage area, addresses 0–327, cannot be checked. The error check logic in the storage unit is checked by program number E3A2.

2.33.10 Storage Protect Test

The Storage Protect Test (program number E3CB) performs functional and diagnostic tests of the storage-protect storage and logic used in the Model 65. It tests the operation of the storage-protect logic while the CPU is issuing storage requests and fetches, and the Test and Set, Insert Storage Key, and Set Storage Key instructions. The SP4I in the 2365 Processor Storage and the M8 in the 2361 LCS are tested by this program. No attempt is made to perform a complete test of the channel storage protect functions.

The 2365 Processor Storage portion consists of six routines:

1. Routine 1, Functional Test: Tests the logical functions of the storage-protect logic. A single location is used in the test. All combinations of storage keys are placed in the location, with all combinations of PSW keys being used against each of these storage keys. Tests are made to check that proper protective action occurs on both storage requests and fetches, and that all basic storage modules of main storage can be cancelled when a storage address protect interruption occurs.

   A test is also made of I-Fetch recovery on a Set Storage Key instruction for all CPU's that perform a recovery on that instruction. This is the case where a Set Storage Key instruction changes the key of the block from which the CPU is fetching instructions. The I-Fetch must refetch these instructions to know if a protection violation exists on these instructions after the Set Storage Key instruction is completed.
2. Routine 2, Basic Data Patterns: Tests the ability of the storage-protect storage to store and fetch 0's and 1's.
3. Routine 3, Floating Zero Pattern: A 0 is floated through a field of 1's to check for improper bit wiring and induced noise. This routine also performs the functions of an addressing test.
4. Routine 4, 64/256 Checkerboard: This routine is the worst-case pattern for the M8 storage when used as the storage-protect storage in the 2361 LCS. This routine writes the pattern into storage, then cycles each location through a three-cycle sequence of: (1) write the opposite pattern, (2) read the opposite pattern, and (3) write the original pattern. After performing this sequence on all locations, the contents of the storage-protect storage are read out and checked.
5. Routine 5, SP4 Core Beat Routine: The SP4 storage protect does not have a delta-noise routine. There are no half-selected cores during the read cycle. During the write cycle, the bit drivers supply half-write current to all cores on the associated bit line, with the word line supplying the other half-write current. This routine tries to set non-selected bits to the 1 state with this half-write bit current.
6. Routine 6, Random Key Mix: Randomly selected instructions are written into all storage protect blocks above the program. Random keys are generated and placed into all storage protect locations. Program control is then given to the random instructions, and they are executed to exercise the random keys and to try to cause storage protect errors. The program takes a random path through the various blocks of storage protect storage and maintains a PSW key that matches the key of the block from which the instructions are being taken. The simulate-I/O interface is turned on while executing these random instructions to cause I/O data fetches, and is also run with a non-zero protect key. Any storage protect errors that are sent to the channel are handled and analyzed in the normal fashion.

Sequential LCS and units 1 and 3 of interleaved LCS can be tested in a normal fashion, in the same manner as the

2365 storage protect. In units 2 and 4 of interleaved LCS, however, the keys cannot be fetched back to the CPU because of the System/360 requirement that bit 20 of the address of an Insert Storage Key or Set Storage Key instruction be zero. The program depends on the hardware check circuits to check the keys in these units. For this reason, however, the recovered key data from these units is unavailable.

2.33.11 Channel Programs

The 2860 Selector Channel is checked with diagnostics which do not require devices on the channel being tested. About 95% of the channel can be tested without devices attached. These programs, which are functional in nature, should be run before I/O device diagnostics are requested. The Selector Channel diagnostics are:

1. E450, Functional tests.
2. E451, Bad parity detection tests.
3. E457, Addressing and protect key tests.

In the 2870 Multiplexer Channel diagnostics, testing is accomplished using channel diagnose lines; thus, actual I/O unit response cannot be tested. The multiplexer channel diagnostics are:

1. E431, Tests local storage in the 2870 channel.
2. E432, Tests basic channel functions and responses.
3. E433, Tests data transfers and chaining operations.
4. E434, Tests channel error-checking circuits.
5. E435, Tests the ability to address all of core storage, including LCS.
6. E436, Tests the ability to operate two interfaces simultaneously.

2.33.12 Systems Integration Program (SIP)

As noted in the previous paragraphs, the unit diagnostics and multi-unit diagnostics are designed to test specific units and subsystem features. The Systems Integration Program, SIP (SIP is EFB0 and program numbers EFB1–EFDF are subprograms), is designed to test the Model 65 operating as a system. This program is concerned with interaction problems caused by simultaneously running all units. It is designed to operate in conjunction with DME for system definition and program loading. SIP is composed of a basic control program that executes and controls SIP modules for I/O units and CPU's. These SIP modules are run in a multi-programming and multi-processing environment. The tested system can vary from a minimum configuration to a maximum configuration.

The SIP CPU tests include Matrix, Instruction Mix, Memory Sort, Binomial Expansion, and Random. The program exercises the I/O units in various combinations while these CPU tests are being executed.

Because SIP has been tailored to the FE's needs for large systems, it is a most useful system test program. When used properly, SIP has also proved to be a valuable diagnostic tool. Its use is strongly recommended to establish the overall status of a system in the following situations:

1. When the customer has turned the system over to IBM with an undefined or poorly defined problem.
2. *Before* and *after* the installation of EC's.
3. At the beginning of "PM," when the time available will probably exceed the time required to correct known customer complaints.
4. Before turning the system back to the customer.

The SIP program, if running successfully, provides considerable confidence regarding total system operation.

2.33.13 System Error Recording and Editing

The Model 65 contains extensive error-checking logic. To make the best use of this logic and of the machine environment saved (logged out) when an error is detected, two recording programs and two editing programs are available.

2.33.13.1 Recording Programs

At system generation time, the customer has the option of including in his operating system either of two automatic recording modules or of relying on a manual means of error-recording via the System Environment Recording, Edit, and Print (SEREP) program. The two automatic modules are System Environment Recording 0 (SER0) and 1 (SER1). They are basically identical in that they record by writing the error logout on the system residence (disk), but are different as to size of resident core storage required and as to system recovery after the recording. SER0 resides in a small amount of core storage but requires that the system be reinitialized (reload of OS/360) upon completion of the recording. SERI, which also resides in core storage, analyzes the error to determine whether the damage to OS/360 is sufficient to abort the system. In some instances, depending on the error, only the job is aborted, and OS/360 continues.

2.33.13.2 Editing Programs

Two editing programs are available: Error Record Edit and Print (EREP) and SEREP. EREP edits and prints the logouts stored on the disk by SER0 or SER1 and runs under control of OS/360. SEREP is a stand-alone (IPL) program which edits and prints the logout taken directly from core storage. SEREP requires that the system be reloaded after its operation, whereas EREP does not. SEREP requires information regarding the output device address, emulators, and features to be punched in a control card prior to loading.

2.33.14 Diagnostic Sections (Emulator Features)

The diagnostic programs for the three emulators (7074, 7080, and 7090) are written to enable test of the emulator

hardware only. The assumption in connection with all three emulator diagnostic programs is that the diagnostic programs for the basic CPU hardware have already run successfully. Therefore, the emulator diagnostics are written with the assumption that any errors they detect are errors in the emulator hardware. The special instructions for the emulators are tested by examining the CLD's and arranging the data or instructions in such a way that all passes through the CLD's are utilized or tested. This procedure ensures that all emulator hardware is tested.

A characteristic peculiar to the emulators, which affect the diagnostic programs, is that certain areas of main storage are addressed directly by the hardware. For example, the address translation hardware always addresses a specific area in main storage (usually the upper portion), which is referred to as the "image core" for the object machine. Therefore, the emulator diagnostics must ensure that they or the DM they are using cannot be located in these areas. Part of the main storage allocation for op code translation is in the lower portion of main storage. This portion is the area normally occupied by the DM. When the emulator diagnostic is running, certain routines which test op code conversion must, therefore, move the DM to another area. Consequently, the emulator diagnostics must have identified the DM they are going to run with. The specific DM model they will run with is listed in the program listing and in the program writeup.

2.33.14.1 7074 Emulator Diagnostics

The 7074 Emulator Diagnostics are in 11 sections (program numbers 6F21 through 6F2B). The first sections test common functions such as Convert Address to Binary and Enter and Leave Emulator Mode. Succeeding sections build up a greater collection of instructions until, in 6F2B, the complex table lookup instructions are tested. The instructions tested by each section are listed in the program listing and writeup. Each section occupies fewer than 4000 bytes of HSS.

The error printouts use the conventional format when the description of the error is printed out with the actual and expected data fields.

2.33.14.2 7080 Emulator Diagnostics

The 7080 Emulator Diagnostics are in two sections (program numbers 6F31 and 6F32), each occupying about 40,000 decimal bytes of HSS. Here, as in the 7074 emulator diagnostics, the basic instructions are tested first, followed by the more complex instructions. If an error is detected, an error number is printed with the identity of the routine that detected it, the actual data, and the expected data. At the end of the program listing, a cross reference of the error numbers is provided. A description of the test run and the meaning of the actual and expected data fields are associated with each error number. The program listing is referred to from time to time to clarify the function of certain routines.

2.33.14.3 7090 Emulator Diagnostics

The 7090 Emulator Diagnostics are in two sections (program numbers 6F41 and 6F42), each occupying about 40,000 decimal bytes of HSS. Here, as in the 7074 emulator diagnostics, the basic instructions are tested first, followed by the more complex instructions. Also, as in the 7074 Emulator Diagnostics, the error printouts follow the conventional format in that the description of the error is printed out with the actual and expected data fields.

2.33.15 Diagnostic Sections (Multisystem Feature)

Four diagnostic sections are available for testing the Multisystem feature and the other functions unique to the Multiprocessing System/360 Model 65. The testing assumes that all the diagnostic sections for the individual units and the other features in the system run successfully. Therefore, an error detected by the four Multisystem feature diagnostics indicates a faulty component unique to the multiprocessing system. The four diagnostic programs are:

1. 6EA0, Shared function (bring-up) tests (2 CPU's required).
2. 6EA1, Simplexed CPU tests (1 CPU required).
3. 6EA2, Duplexed CPU tests (2 CPU's required).
4. 6EA3, Shared storage priority tests (2 CPU's required).

Chapter 3. Preventive Maintenance

Section 1. Basic Unit

The preventive maintenance schedule (Table 3-1) provides a checkout routine designed to keep the CPU performance at original specifications. Several of these procedures permit anticipation of a failure for some of the critical circuits, providing ample warning for replacement planning. The accuracy of these predictions depends upon the frequency with which the checkout procedure is run. The frequency and thoroughness of each procedure should be increased if warranted by the conditions.

Table 3-1. Preventive Maintenance Schedule

Item	Procedure	Frequency (months)
ROS bit-plane pressure	Heading 4.13.3, steps 9 through 11	1
Lamp test	Heading 4.7	3
Air flow	Check for proper air flow through entire CPU; all fans running, filters clean (replace filters every 3 months) and not obstructed, and exhaust areas not blocked.	1
Marginal checks	Heading 5.3	3
Diagnostics	Heading 4.5	3
Voltage and power checks	Heading 5.4	6
Grounds	Heading 4.8	6
Timing checks	Heading 4.6	6
ROS optimization	Heading 4.9	6
Elapsed Time Meter calibration	Heading 4.10	6
Cleaning	Vacuum and dust inside CPU. Check air flow paths. Clean covers. Clean switch contacts.	12
Mechanical	Check for loose screws, bolts, and nuts; cable wear, overheating; and broken connections.	12

Section 2. Features

There are no unique preventive maintenance procedures for the features.

Chapter 4. Maintenance Procedures

This chapter encompasses all procedures for trouble-shooting and checkout with the exception of power. All procedures relating to power are in Chapter 5.

Section 1. Service Checks and Repair Procedures

4.1 MAIN STORAGE RIPPLE TESTS

4.1.1 Write All 1's

1. Depress ADDRESS keys 0, 21, and 22.
2. Set all DATA switches to 1's.
3. Set Storage Select switch to MAIN.
4. Set RATE switch to PROCESS.
5. Depress ROS TRANSFER.
6. Run at nominal voltage for 1/2 minute.
7. To stop test, place STORAGE SELECT switch in LOCAL.
8. Depress SYSTEM RESET.

4.1.2 Read All 1's

1. Set ADDRESS key 0 to 1. Set all other ADDRESS keys to 0.
2. Set CPU CHECK switch to STOP.
3. Depress ROS TRANSFER.
4. Run at nominal voltage for 1/2 minute.
5. To stop test, set STORAGE SELECT switch to LOCAL.
6. Depress SYSTEM RESET.

4.1.3 Write/Read All 0's

Perform procedures in headings 4.1.1 and 4.1.2 with DATA switches set to 0.

4.1.4 Write/Read Alternate Pattern

1. Repeat procedures in 4.1.1 and 4.1.2 with each byte of DATA switches set to 10101010 (AA in hex).
2. Repeat procedures in 4.1.1 and 4.1.2, with each byte of DATA switches set to 01010101 (55 in hex).

4.1.5 No Parity Bit Pattern

Repeat procedures in 4.1.1 and 4.1.2 with each byte of the DATA switches set to 00000001 (01 in hex). This test will determine if the parity bit for all storage addresses can be reset. In all previous tests the parity bit is set.

4.2 LOCAL STORE RIPPLE TESTS

4.2.1 Write All 1's

1. Set ADDRESS switches 0, 21, and 22 to 1's.
2. Set DATA switches 32–63 to 1's.
3. Set STORAGE SELECT switch to LOCAL.
4. Set RATE switch to PROCESS.
5. Depress ROS TRANSFER.
6. Run at nominal voltage for 15 seconds.
7. To stop test, depress SYSTEM RESET.

4.2.2 Read All 1's

1. Set ADDRESS key 0 to 1. Set all other ADDRESS keys to 0.
2. Set CPU CHECK switch to STOP.
3. Depress ROS TRANSFER.
4. Run at nominal voltage for 15 seconds.
5. To stop test, depress SYSTEM RESET.

4.2.3 Write/Read All 0's

Repeat procedures in 4.2.1 and 4.2.2 with DATA switches 32–63 set to 0.

4.2.4 Write/Read Alternate Pattern

1. Repeat procedures in 4.2.1 and 4.2.2 with DATA switches 32–63 set to 10101010 in bytes 4, 5, 6, and 7.
2. Repeat procedures in 4.2.1 and 4.2.2 with DATA switches 32–63 set to 01010101 in bytes 4, 5, 6, and 7.

4.2.5 No Parity Bit Pattern

Repeat procedures in 4.1.5, making sure that the STORAGE SELECT switch is in LOCAL.

4.3 ROS TESTS

The ROS tests are explained in Chapter 2, heading 2.31.2. Operating procedure is found on logic pages M8005.

The ROS tests provide a thorough check of the ROS portion of the CPU. These tests do not require the

execution of a program; special hardware (hardcore) is used to run portions of the tests. This hardcore is tested first to ensure that hardware testing can proceed properly. To provide background information for troubleshooting, a discussion of ROS parity checking follows.

4.3.1 ROS Parity Checking

The ROS word is divided into four groups: bits 0–5, 6–42, 43–68, and 69–99. Bit 0–5 are ignored and are not parity-checked. The remaining three groups are parity-checked independently. Bit 20 is the parity bit for bits 6–42; bit 85, for bits 43–68; and bit 91, for bits 69–99.

If a parity error is detected, the reset to the part of the data register or indicator backup latches containing the failing group is blocked. The new data is not gated into this group. The group or groups not in error are reset and have their part of the next ROS word gated in. For example, if bit 87 fails, the reset to bits 69–99 is blocked. The other two groups, bits 6–42 and 43–68, receive their portion of the next ROS word. Bits 0–5 always get the next word data.

The group in error may be determined by observing roller switch 2, position 2, the first three of the last five indicators (ROS PTY: 6–42, 43–68, and 69–99). The indicator(s) that is on indicates the failing group(s). The failing bit(s) may be identified by comparing the indicated bits of the failing group (roller switches 2, 3, and 4, position 4) with the bits listed in the ROS address list (QZ logic) for that ROS word.

The address of the failing ROS word may be found in the indicated ROS Previous Address Register (ROSPAR). [Roller switch 3, position 4, the previous address indicator (PREV ADR A), provides the register indication. If the indicator is on, use ROSPARA register; if off, use ROSPARB register.] The alternate ROSPAR contains the address of the word that accessed the failing word. ROSAR contains the address of the next word. A ROS parity error prevents these registers from advancing from this state. The registers are indicated by roller switch 1, position 4.

4.3.2 ROS All 0's, All 1's Word Tests

Stored in ROS are two words that may be used to check the ROS sense amplifiers, sense latches, and indicators. One word contains all 0's, except bit 2, and has correct parity in total, but each section has incorrect parity. The other word contains all 1's except bits 1–5, 47–56 (NA field), and bits 36, 37, 81, and 96 (spares), and has incorrect parity in the 69–99 section. To use these words, proceed as follows:

1. Depress SYSTEM RESET.
2. Set CPU CHECK switch to STOP.
3. Set REPEAT ROS ADDRESS switch.
4. Set roller switches 2, 3, and 4 to position 4.
5. Set address 000 (hex) in ADDRESS switches 0–11.
6. Depress ROS TRANSFER.
7. Check the indicators for zeros in all locations. (The basic unit does not display bits 0–5.) The repeat loop should be stopped, and all three parity errors should be indicated (roller switch 2, position 2).
8. Depress SYSTEM RESET.
9. Set address 801 (hex) in ADDRESS switches 0–11.
10. Depress ROS TRANSFER.
11. Check the indicators for 1's in all locations, except bits 0–5, 47–56, 36, 37, 81 and 96. The repeat loop should be stopped, and a 69–99 parity error should be indicated (roller switch 2, position 2).
12. Depress SYSTEM RESET.

If incorrect indications are present (see steps 7 and 11 for correct indications), loop on adjacent addresses and compare the indicated data with the data listed in the ROS address list (QZ logic) for the word. If the same bits fail, proceed as in heading 4.3.5. If the adjacent address data is correct, run the ROS word tests (heading 4.3.3).

4.3.3 ROS Word Tests

The ROS word tests check the ROS hardcore and each bit of each ROS word. Each ROS word bit is checked by comparing it with an expected value for it. The expected value used for the comparison is obtained from the manufacturing interface tape from which the ROS bit planes are manufactured. To perform the comparison, the ROS address is loaded into ROSAR by way of S. Each address is used repetitively until each bit in the ROS word is checked. If there is no comparison, the test stops with the failing bit and its address displayed on the system control panel.

The operating procedure for the ROS tests using the ROS test tape is found on logic page M8005. If an error is encountered during steps 1–8, proceed in accordance with heading 4.3.4. If an error is encountered during steps 9–11, refer to heading 4.3.5 for additional information and proceed in accordance with 4.3.6.

There is no analysis documentation to support failure isolation because all of the necessary information is displayed on the system control panel.

4.3.4 ROS Hardcore Repair

This repair procedure consists of a series of observations and tests to locate a failure in the ROS hardcore. The correct operation of the hardcore is necessary for successful testing of the ROS word bits.

If any of the following failures occur, proceed as directed:

1. Failure to store 1's successfully in ST. This failure may be identified by inspecting the indicators (roller switches 1 and 2, position 3).
   a. Check that TEST MODE switch is set to ROS; if it is not, set it, and restart the tests.
   b. Trace the failing bit(s) using logic, and repair.
2. Failure to complete IPL (first depression of LOAD).

This failure may be identified by not finding the indications specified in the test procedure (logic page M8005), or by the LOAD indicator (panel G) or the Timing Gate indicator (roller switch 4, position 5) remaining on.

a. Check for the following:
   (1) LOAD UNIT switches are set to the correct channel and I/O unit address.
   (2) CPU CHECK switch is set to DSBL.
   (3) Channel is in Automatic mode.
   (4) I/O unit is ready and not in Test mode.
   (5) Unit data check (UDC) or channel control check (CCC) (roller switch 5, position 2) indication. If either is on, inspect appropriate unit for details.

b. Inspect storage location 0–23 (hex) for correct IPL data and location 80-A7 (hex) for Loader record. Compare displayed data with data specified on logic page M8006.

c. If tape runs away, a failure of the test number compare operation is indicated. Either the test is not being performed at all or it is not successful. Use the following procedure to isolate the trouble:
   (1) Clear main storage.
   (2) Store all ones in 8000 and 8080 of main storage. This will set up the buffers to scan in all bits to S and T.
   (3) Store FF FF FF FF 03 00 05 41 (hex) at main storage locations 8008 and 8088.
   (4) Set the following switches:
       CPU CHECK to DSBL
       TEST MODE to FLT with REPEAT on
   (5) Set up ROS address 6B0 in ROS ADDRESS COMPARE switches and depress ROS TRANSFER.
   (6) If the pass trigger comes on, the test number compare is functioning correctly. The trouble must be something which prevents the test from being made; e.g., a missing TIC pulse.
   (7) If the fail trigger comes on, a missing bit from either S or T is indicated. Sync on ROS address 150 while observing "Scan out S + T" for reference. Scope "PAL=0" which should be active during "Scan out S + T". Scope back to the failing bit.

3. Storage check. This failure may be identified by the STOR CHK indicator being on. Use the ripple tests (heading 4.1) with STOP ON STORAGE CHECK depressed to locate the failing address. Roller switch 6, position 4, indicates the type of error.

4. Failure to stop with the specified indications after depression of RESTART FLT I/O. The indications are listed on logic page M8005. (If tape is used, it may not stop running. Note that each depression of RESTART FLT I/O causes a backspace and read operation.) This failure is caused by the inability to make test number comparison. The expected condition is all 1's in S and a zero result from PAL. Use the following procedure to isolate the failure:

a. Disable interval timer.
b. Set CPU CHECK switch to DSBL.
c. Add the following jumpers to ground:
   (1) 01B-C2G4B10 ('Enable scan bypass' signal on ALD AP821).
   (2) 01E-B3F6D07 ('Scan Out ST' on KU491).
d. Depress SYSTEM RESET.
e. With the data switches all up, depress STORE. Verify that S and T contain all 0's. PAL(32–63) should contain all 1's (Roller 6, position 1).
f. Depress DATA switches 0–31 and depress STORE. Verify that S contains all ones and that both T and PAL contain all zeros (except parity bits).

   Note: While in the stop loop, ROS address 8AC [and 893 if IC (21,22) = 11] will be gating IC and a constant through the adder. This may cause some indicators to glow dimly and may generate a pulse at the output of the zero detect circuits. To prevent confusion in steps f and g, eliminate these indications by setting all ADDRESS switches to zero and depressing SET IC.

g. Return DATA switches 0–31 to up position and depress DATA switches 32–63. Depress STORE and verify that S contains all 0's and T contains all 1's. PAL should again contain all 0's.
h. During steps f and g, the output of the zero detection circuits should indicate a zero result. This may be verified by checking 01E-B3C6D07 for a minus level (net AZ4 on logic KU311).
i. If correct indications are not obtained, check zero detection circuitry. Figure 4-1 shows a representative bit position in the circuitry used for test number and result comparisons in ROS word tests and FLT's. Note that a bit in S (Roller 1) or its corresponding bit position in T (Roller 2) will result in a zero output from PAL. PAL is set to 1 only if S and T both contain 0 in that position. Also note that indicators are numbered 0 to 35; e.g., S(31) and T(63) are roller bit 35.
j. Remove ground jumpers and return switches to normal.

5. Failure to complete IPL (second depression of LOAD). Refer to failure 2, above, for directions; the symptoms are similar. If necessary, use the following procedure to single cycle through the ROS operation:
   a. Set up buffers A and B at main storage locations 8000 and 8080 respectively. Doublewords at 8000 and 8080 contain TN and ATN. Doublewords at 8008 and 8088 contain mask and MCW. Refer to Chapter 2 for details.

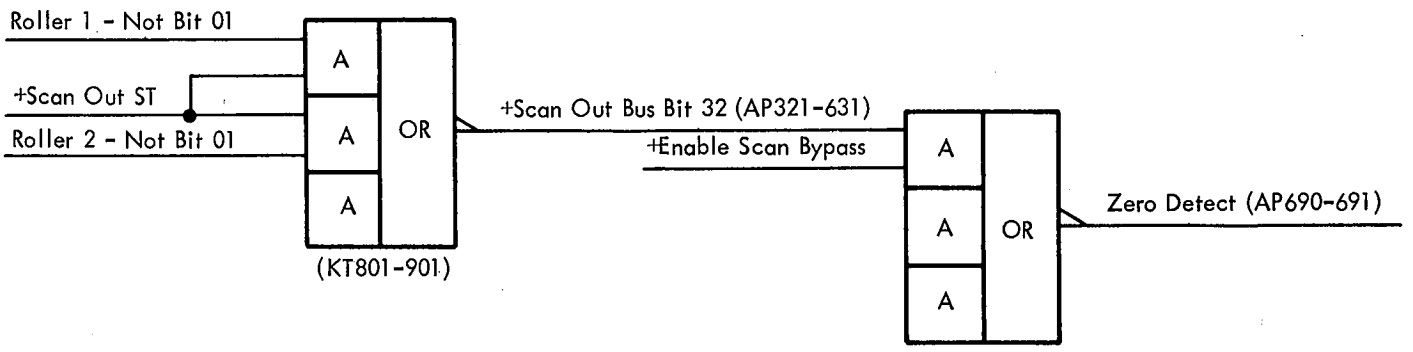


Figure 4-1. Test Number Comparison Circuit

b. Store all 1's in S and T.
c. Set the following switches:
   CPU CHECK to DSBL
   TEST MODE to ROS
   RATE to SINGLE CYCLE
d. Observe roller 5, position 2 for ROS Test Sequencer.
e. Depress RESTART FLT I/O and follow flowcharts (FEMDM, Diagram 8-116). Step through operation with START key until ROS Test Sequencer = 6.
f. Momentarily ground E-B3G07B04, which will turn on TIC trigger.
g. Continue to single cycle, following flowcharts until ROS Sequencers loop on 3, 2, 1. Pass trigger should then be on and this test should loop until a TIC pulse is generated by repeating step f. Step ROS sequencers to 0 and proceed with test.

6. Failure to reach successful completion of the tests, with the specified indications, after second depression of RESTART FLT I/O. The indications are listed on logic page M8005. Error analysis data is also provided on the same page and on page M8006.

4.3.5 ROS Hardware Tests

The ROS hardware tests provide for the isolation and repair of a failing ROS bit or bits if the failure is in the ROS hardware (up to and including the sense latches). To aid in understanding the adjustments and restrictions of ROS, the following background information is provided.

The ROS uses the capacitive coupling between a drive/balance line pair and a differentail sense line pair to generate a signal. The nominal signal level from a sense line pair is on the order of 1 mv. To amplify this low-level signal to SLT levels, a sense amplifier is used; a clipping network squares the signal and limits the amplitude.

The sense amplifier is essentially two differential amplifiers, a linear amplifier, and a clipping network. Each amplifier differentially senses the signal from comparable sense line pairs on each gate (C and D). The two differential amplifiers are then dot-OR'ed into the linear amplifier. (Since only one side of one gate is driven at a time, only one of the two differential amplifiers senses a signal.) The output of the linear amplifier is then fed through the clipping network to the sense latch. Figure 4-2, A, shows a typical 1 and 0 at the output of the sense amplifier (after clipping). Reference time T_0 may be found at location 01C-E3A2D06. Note that the output is always a bipolar signal; the difference between a 1 and a 0 is the sequence in which the positive and negative voltage swings occur. A 1 is first positive, then negative; a 0, negative, then positive. (The initial voltage swing is called the "signal"; the following kickback, the "recovery.")

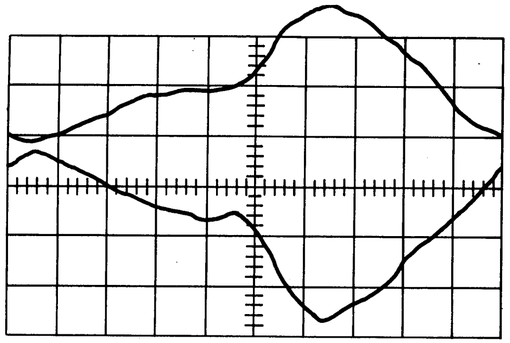
When observing a string of 1's and 0's while in a particular loop (e.g., the stop loop), the 1's may be distinguished from the 0's by looking at the fall time of the positive voltage swing. A 1 has a sharp rise and fall; a 0 has a sharp rise and a slower fall. See Figure 4-2, A and B.

The drive line of the drive/balance line pair is driven by a transistor in a 1408 transistor array drive matrix, 88 transistors per bit plane. One side of the matrix (the base of the transistors) is driven by 22 'select bus base drive' (SBBD) signals, common to all bit planes. The other side of the matrix (the emitter of the transistors) is driven by 64 'select bus emitter drive' (SBED) signals, four per bit plane. The transistor in the matrix at the intersection of the active SBBD and SBED signals drives the selected drive line. To prevent more than one transistor from being gated on in each cycle, the SBBD precedes the SBED signal. See Figure 4-2, C and D.

4.3.6 ROS Hardware Repair

With the preceding background information in mind, the results of the ROS tests may be properly interpreted. The following discussion provides techniques for ROS hardware tests. The procedure on logic page M8005, steps 9–11, is used to run the tests and to set up scoping loops when failures occur. Additional troubleshooting techniques and procedures, not based on the ROS tests, have been included in this discussion where they have been found to be of value to the CE.

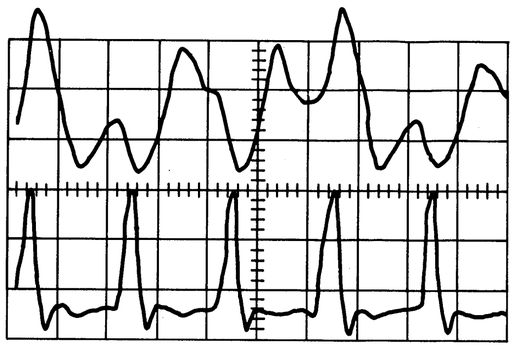
A flowchart for ROS troubleshooting is given in the Diagnostic Techniques section of the 2065 FEMDM. ROS timing information is found on logic page M8003. A chart of sense amplifier and sense latch locations is provided on



20ns/div; .5v/div.

Top Trace = 1 Bit
Bottom Trace = 0 Bit

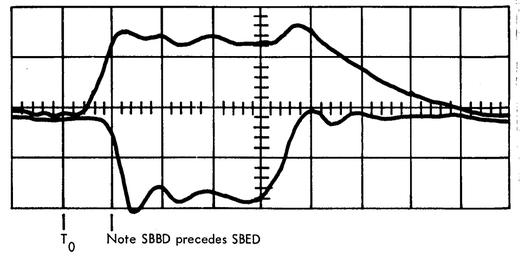
(A) Sense Amplifier Output, Typical 1 and 0



20ns/div; .5v/div.

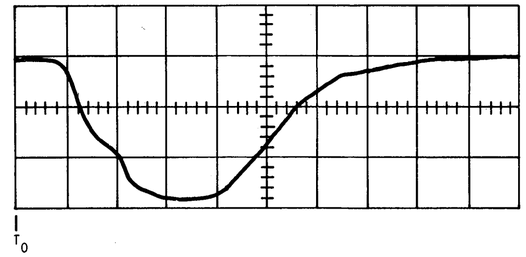
Top Trace = Bit Pattern (10010)
Bottom Trace = Strobe

(B) Sense Amplifier Output, Five Cycles



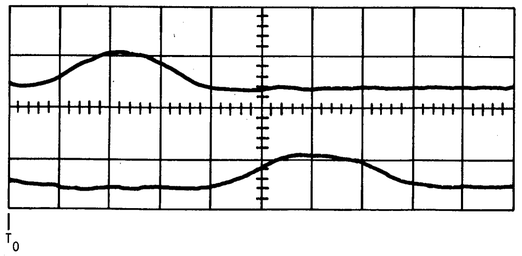
Upper waveshape: Typical SBBD
Lower waveshape: Typical SBED
Scope settings:
V 2v/cm
H 20ns/cm

(C) Signals to Array Drive Transistor



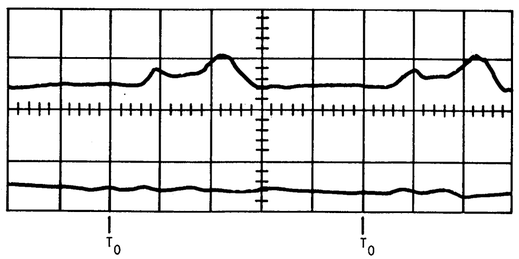
Scope settings:
V 2v/cm
H 20ns/cm

(D) Typical Array Drive Transistor Output



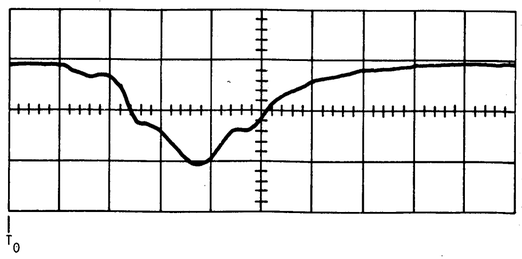
Upper waveshape: Weak 1
Lower waveshape: Weak 0
Scope settings:
V 1v/cm
H 20ns/cm

(E) Sense Amplifier Output, Weak 1 and 0



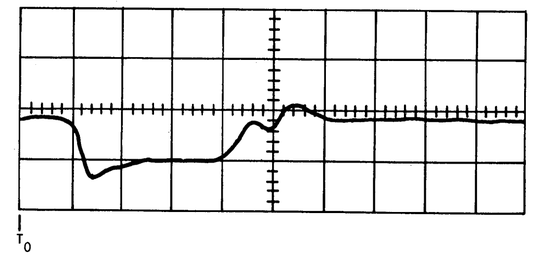
Upper waveshape: Distorted Bit
Lower waveshape: No Bit (Select Noise Present)
Scope settings:
V 1v/cm
H 40ns/cm

(F) Sense Amplifier Output, Distorted Bit



Scope settings:
V 2v/cm
H 20ns/cm

(G) Slow or Low Array Drive Transistor Output



Scope settings:
V 2v/cm
H 20ns/cm

(H) Array Drive Transistor with Open Drive Line

Figure 4-2. ROS Signals

logic page EF501. Refer to heading 4.9 for strobe timing and bias optimizing procedure.

4.3.6.1 Weak Sense Amplifier Output

A low or weak signal that appears with slow transitions and low amplitudes (Figure 4-2, E) may be caused by low torque on the pressure plates, dirt on the bit plane, or a poor drive signal.

Whenever a poor 1 or 0 is found, the first requirement is to substitute another sense amplifier card to be sure the card is not at fault. The card used, PN 5801524, has four amplifier channels on it.

On a few occasions poor sense amplifier operation has been traced to a missing −18V bias voltage. This causes the sense amplifier to behave as if it were biased at −7V. The bits appear much wider than normal. Check for poor solder connections to voltage buses and for loose voltage connectors.

If the weak output is a result of low torque, the torque should be corrected using the procedure under heading 4.13.3. Contamination on the bit plane may be sufficient to increase the distance between the drive line tabs and the sense lines causing reduced output. If this is the case, the bit plane must be cleaned. The procedure for this is under heading 4.13.2.

A slow or low array drive transistor output results in reduced sense amplifier output. Figure 4-2, G, shows this condition.

4.3.6.2 Distorted or Missing Sense Amplifier Output

A distorted or missing bit (Figure 4-2, F) may be caused by either an open or a short circuit. Either the sense line or the drive line could be open. If the drive line is open, the output of the driver transistor will appear as shown in Figure 4-2, H.

Possible shorted conditions are: a sense line shorted to ground, to an adjacent sense line, or to a drive line. A sense line to drive line short is the most probable. A detailed troubleshooting procedure for this condition follows.

The sense lines are isolated from the bit plane drive lines by a sheet of 1 mil Mylar*. If this sheet is punctured or has a piece of conductive material embedded in it, a short may occur. When this happens, failures can occur in all bit planes but will be limited to one bit position in either the upper or lower word (assuming only one puncture).

If a solid short exists, the input to the sense amplifier will be at approximately 6V. This can be scoped with the machine in SINGLE CYCLE after depressing START.

An alternative procedure, which allows a complete check for leakage between drive lines and sense lines or ground, is as follows:

1. Remove pressure bar from bottom of bit plane *without* removing pressure from pressure plates. This exposes the overlapped area of the bit plane tabs and the drive line tabs.
2. A red wire supplying +6V is connected to each edge of the drive line area of the bit plane. Isolate these +6V tabs from their mating bit plane tabs by slipping a small piece of insulating material between them. A piece of punch card stock can be used.

   CAUTION
   Be very careful not to damage the bit plane when inserting the insulator.

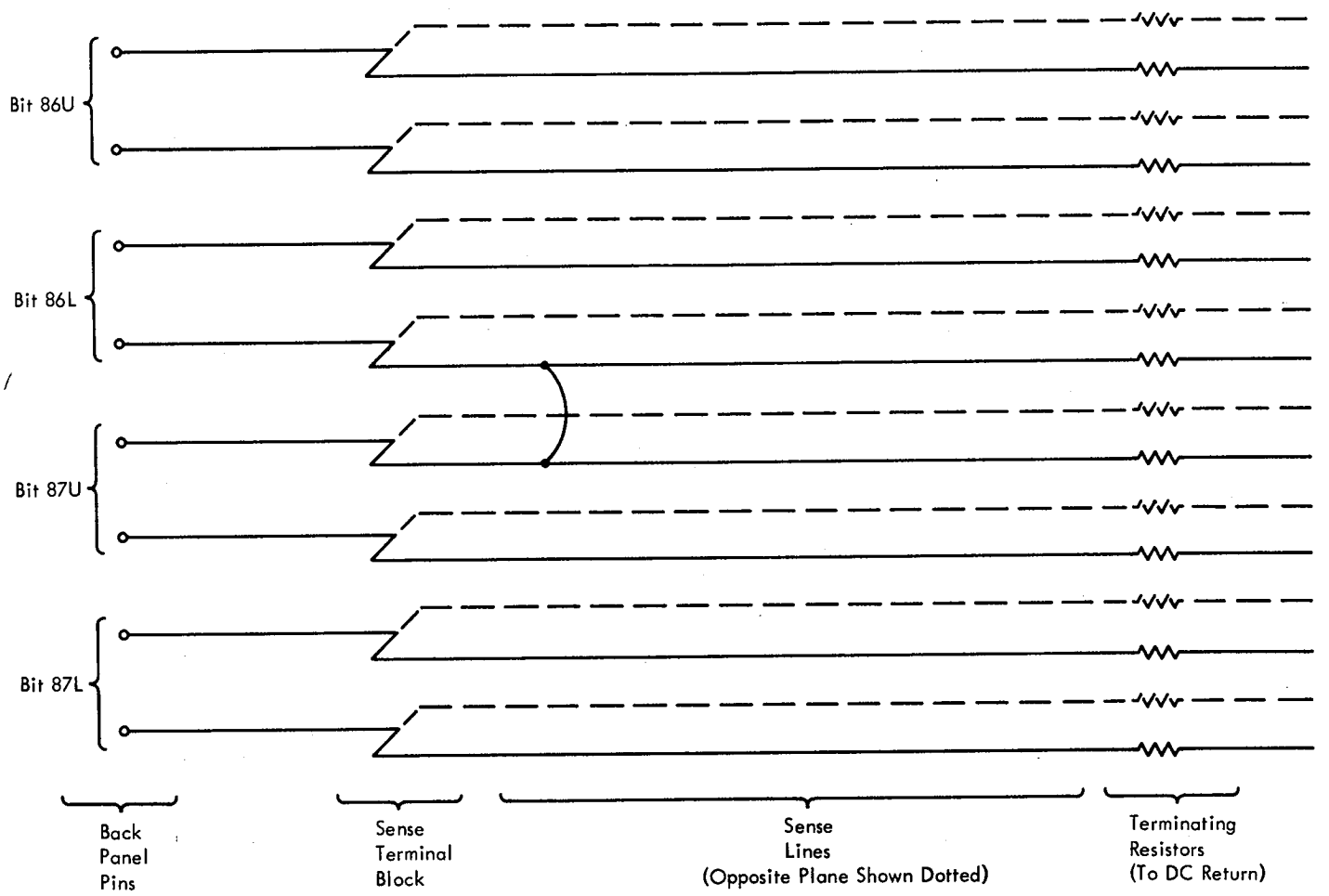
3. Check for infinite resistance between drive lines and ground. This is best done by connecting an ohmmeter across small bypass capacitor at top of plane. There are two capacitors, one for each half of the plane. Both must be checked. Connect + meter lead to + side of capacitor. The resistance should become infinite after 30 to 45 seconds when capacitor has fully charged.
4. If reading is infinite, proceed to next plane and repeat steps 1 through 3 above. Continue until all planes are checked or defective plane is located. If a short is indicated, proceed to step 5.
5. Relax pressure on pressure plates one at a time. Meter should show a higher resistance when pressure is reduced near defect.

   Note: If the resistance does not change when the pressure plates are relaxed, examine the cabling from the 1/3 boards to the array. The short could also be in this area.

6. Remove defective plane using procedure under heading 4.13.1. Inspect area indicated by failing bit position and pressure plate that affected resistance reading. If a piece of foreign material is found, carefully remove it. If a hole is found, check for a burr or foreign substance on sense plate which might have caused hole. A new bit plane must be ordered to replace any found not to have complete isolation from sense lines. Use steps 1 through 3 to verify condition of new bit plane when it is installed.
7. A temporary repair may be made while awaiting parts by using a piece of plastic wrap or transparent wrapper from a cigarette package as an insulator. This additional thickness of insulating material will reduce the output from that plane, however. Often a defective bit plane will work for a time after having been removed and reinstalled. This should not be considered a permanent repair.

If unable to isolate the cause of a distorted or missing bit using the preceding procedure, one of the less probable short or open conditions exists. The trouble may often be isolated by checking physically adjacent locations for signs of distortion. An example is shown in Figure 4-3. Assume

*Trademark of E. I. du Pont de Nemours & Co. (Inc.)



Should an open or a short occur, check the indicated areas (a short from bit 86L to 87U is indicated here).

Resistance Readings *

Condition	From Sense Amp Input to	Resistance
Normal	DC return. Any other sense amp input. Any drive/balance line.	16.5 ohms 33.0 ohms Open
Sense - sense line short	DC return. Input of sense amp to which it is shorted. Any other sense amp input. Any drive/balance line.	8.25 ohms 0 ohm 24.75 ohms Open
Sense - DC return short	DC return. Any other sense amp input. Any drive/balance line.	0 ohm 16.5 ohms Open
Sense - drive/balance line short	DC return. Drive/balance line to which it is shorted. Any drive/balance line in same or opposite plane, except line to which it is shorted. Any drive/balance line not in same or opposite plane.	16.5 ohms 0 ohm ** 200 ohms plus short resistance Open

* Remove sense amp and array drive cards and voltage jumpers in 1/3 boards.
** 0 ohm if dead short, may be a high impedance.

Figure 4-3. Sense Amplifier Input Resistances

that bit 87U is distorted. Bit 87L is normal but bit 86L shows distortion. A short probably exists between bits 87U and 86L.

A complete resistance check of the ROS array may be made which will isolate any of the possible short or open conditions. Proceed as follows:

1. Remove the following from the area of concern:
   a. Sense amplifier cards.
   b. Driver cards.
   c. Bus-to-board voltage jumpers.
   d. Select cables.
2. Make resistance readings between points indicated in the table of Figure 4-3.
3. Isolate the trouble and make the repair.

4.3.6.3 Extra or Missing Bits

Extra (picked) or missing (dropped) bits can result from electrical noise, late ROS branching, or multiple drive line selection. Most of these problems will be intermittent. To determine which is the most probable cause, refer to 4.3.6.6.

4.3.6.3.1 Electrical Noise. Noise problems are usually characterized by intermittent extra or missing bits and by failure of the ROS unit to operate error-free to the full −7V bias limit. Possible sources of electrical noise include: poor cable dress within the ROS unit; ground loops within the system; and defective switches, capacitors, and cooling fan motors. A troubleshooting procedure is given in 4.3.6.3.2.

Cable dress within the ROS unit is critical. Due to the high gain in the sense amplifiers, any noise picked up during strobe time is set into the latches as a bit. It is therefore imperative that all cables be carefully routed to minimize noise coupling and pickup. Figure 4-4 shows the cable dressing in a typical E1 board. Note how the shielded input cables are placed to act as a shield between the input and output pins of the sense amplifiers; also note that the individual leads are placed down between the pins. Figure 4-5 shows the cable routing between the E1, E2, and E3 boards. No significant departure from the routing shown should be made; the exposure to noise cannot be tolerated.

Noise can also result from ground loops within the system. The ROS frame, including the spiders, is at dc ground, not at frame ground. The dc return line for the ROS unit should be isolated from the machine except at the common frame ground connection in frame 02. Follow the procedure in heading 4.8 to determine if ground loops do exist. If extraneous signal to frame grounds are present, they must be removed.

Another source of noise may be the capacitor in the voltage crossovers on gate C, boards E1 and E2. These capacitors must be correctly wired. Check that they are present, not open, and not wired backwards. If the part number (2391011) printed on the capacitor appears upside down, the capacitor is backwards. Be sure that the correct crossover (PN 813080) is used. If crossover (PN 811483) is used erroneously, it will not properly accept the capacitor inside the crossover housing. Earlier machines had these capacitors on all boards on gate C. They are needed only on the E1 and E2 boards. Defective capacitors on other boards may be removed and need not be replaced.

The susceptibility of the ROS unit to noise can be minimized by adjusting the strobe accurately. The following criteria should be met:

1. ROS strobe width, measured at the sense amplifier should not exceed 30 ns at 50% amplitude. Strobe width as low as 20 ns is acceptable if the ROS bias limits can be reached. Since the strobes are generated by many drive signals, several sense amplifiers should be checked for minimum width.
2. ROS strobe timing should fall in the center of a no-bit (negative) pulse. A bit position in the center of the plane should be used for this measurement. Measured at 50% amplitude, the no-bit pulse should overlap the strobe pulse by 8 ns on rise and fall.

Note: Noise problems are intensified by low humidity. The recommended humidity for proper operation is 45 to 50%.

4.3.6.3.2 Troubleshooting Noise Problems. The following procedure should be used in troubleshooting noise problems:

1. Stop ROS clock by putting RATE switch to SINGLE CYCLE.
2. Depress START. The sense amplifiers are now gated.
3. With bias at normal setting, observe failing sense amplifier outputs with an oscilloscope. Failing and non-failing bits can be compared.
4. Check for noise pulses of sufficient amplitude to sync the scope. Anything over 200 mv could indicate a shorted sense and drive line.
5. Slowly reduce the bias toward −7V. Noise resulting from incorrect cable dress usually shows up between −9V and -7V. It will appear as a 20-MHz signal having an amplitude of about 1/2V. The coaxial cables from the array to the sense amplifiers should be carefully dressed to correct this. Refer to the discussion in 4.3.6.3.1 for correct cable dress. When correctly dressed, the cables should be laced into position so that the trouble will not recur.

Noise from other sources must be found by systematically eliminating possible causes one at a time while observing the oscilloscope. Cooling fans can be disconnected one at a time, for example. Refer to the discussion in 4.3.6.3.1 for additional information.

4.3.6.3.3 Late ROS Branching. When the branch portion of the next ROS address is data-dependent, it is not available until late in the machine cycle. Due to a malfunction or a timing problem, the correct next ROS address may be decoded late. ROSAR will usually be set correctly but the ROS word will not be driven out of the array correctly. This condition is described as late ROS branching. It can produce ROS parity checks in patterns which are difficult to distinguish from those caused by electrical noise. The following method may be used to differentiate between the two.

After the machine has stopped with a ROS parity check indication, one additional word will have been read out of ROS. The address of this additional ROS word will be indicated in ROSAR. It may or may not be in good parity. Since the ROS clock was stopped when this word was read out, no parity check indications pertaining to it will be on.

The ROS word which caused the error will have its

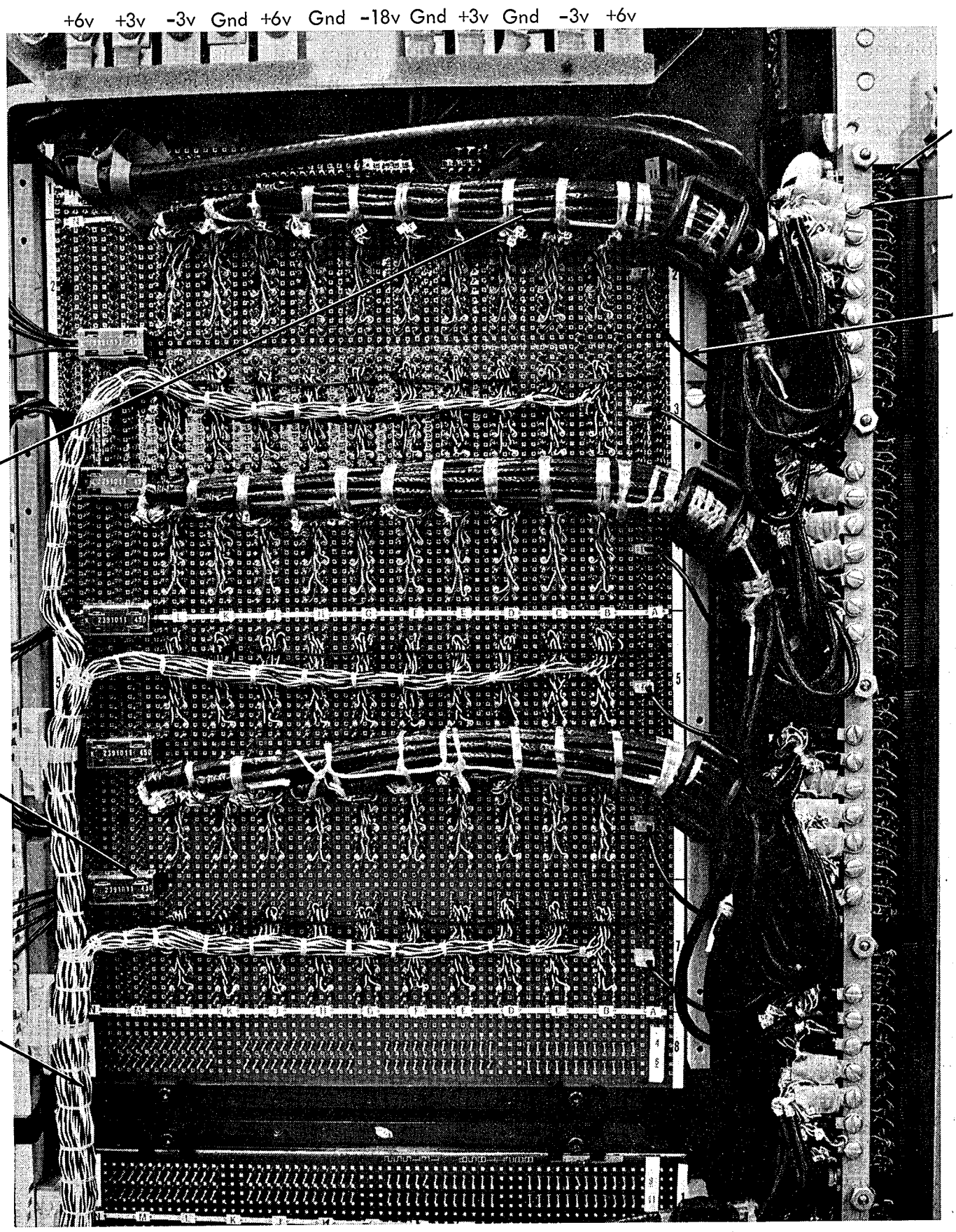


Figure 4-4. ROS E1 Board

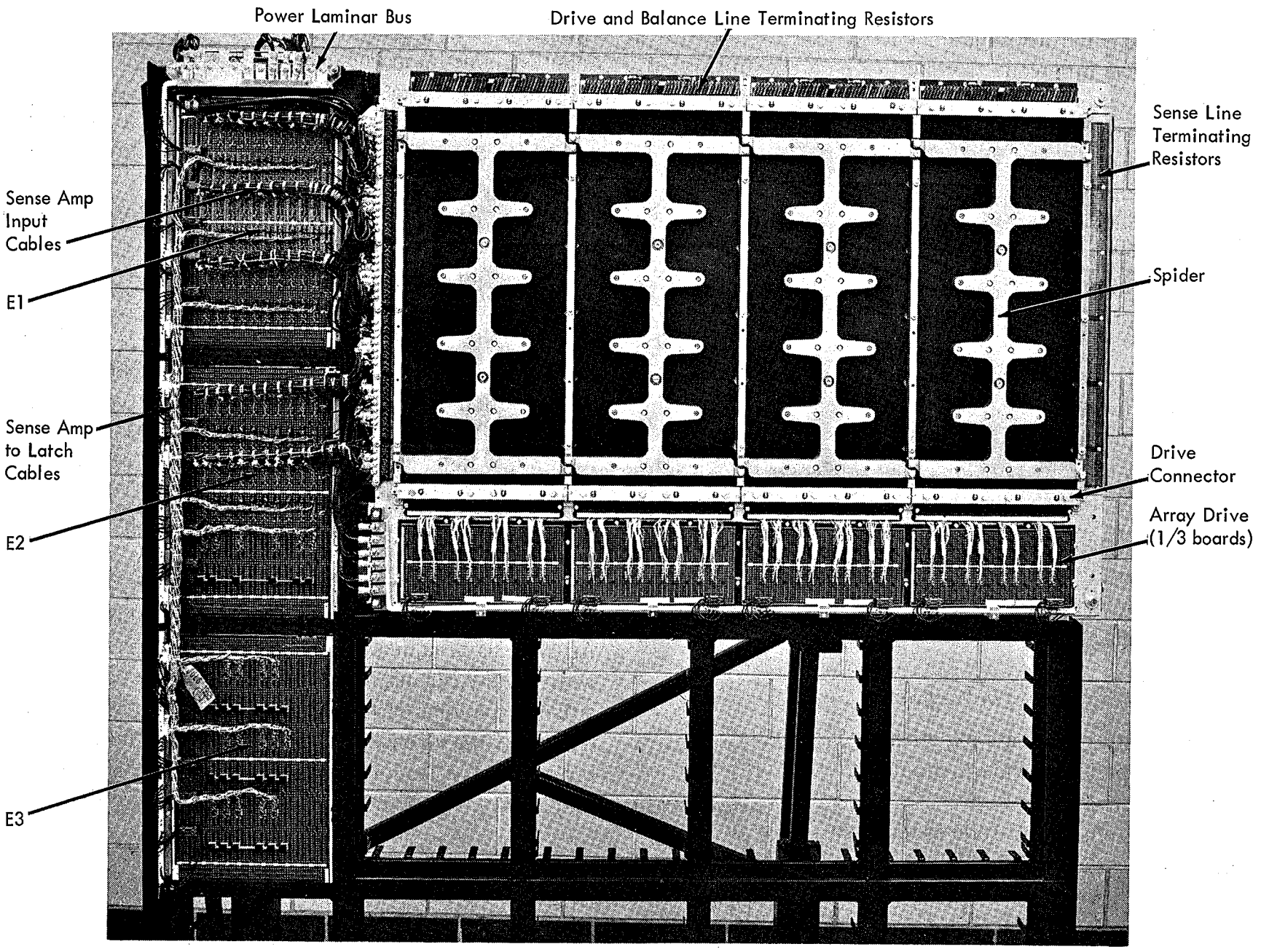


Figure 4-5. Overall ROS Layout

address in ROSPARA or ROSPARB, depending on the state of the PREV ADR A trigger. The section of this word which was out of parity will be saved in ROSDR. The remainder of ROSDR will contain bits from the word read out after the error.

Because electrical noise bursts are usually several milliseconds long, the second word will usually be out of parity as well as the first if noise is the cause. Look up the address of the last word read out (the address in ROSAR) in the ROS address list and determine whether it is in good parity. If it is in good parity, late ROS branching is the probable trouble; if not, noise should be suspected.

4.3.6.3.4 Multiple Drive Line Selection. Failures in which some bits are picked while others are dropped may result from more than one drive line being active at a time. When there is insufficient information to establish a pattern which might indicate an incorrect decode of plane, quarter plane, or drive, the following procedure may be used to find the second active drive line:

1. ROS repeat an address that fails.
2. Pull the voltage connectors from the driver boards on at a time (leaving the one for the plane being addressed) until the failure disappears. This will be the plane with the second conducting drive line. Reconnect all voltage connectors.
3. Set up the oscilloscope as follows:
   a. Form a small coil of insulated wire. Use one or two turns only. Connect this coil between the probe tip and the probe ground connection.
   b. Use a low-voltage setting on the oscilloscope.
4. Move the coil slowly across the bottom of the plane near the drive lines. As the conducting drive line is approached, a voltage will be induced in the coil. This will isolate the trouble to about three drive lines which can then be scoped normally.

4.3.6.4 Extra Bits (Bit 99 only)

On 2065 Processor Units serial number 60200 and below, ROS parity checks in which bit 99 is picked could be caused by the drive line connector board being too thick. This connector board is at the bottom of the ROS bit plane where it connects the 1/3 boards to the ROS array. The

maximum thickness of the connector board is 23 thousandths of an inch. To check the thickness:

1. Assure proper bit plane torque. (See procedure in 4.13.3.)
2. Scope the sense amplifier output for bit 99 lower, using the following addresses:

X04	XC2	
X08	XC6	
X0C	XCA	
X10	XCE	
X14	XD2	
X18	XD6	where X=Plane number

3. Look for a weak 0 bit output having a peak-to-peak swing of between 0V and slightly less than 1/2V.
4. If this weak output is present, verify that the connector board is too thick by removing it and checking it with a micrometer (or similar measuring device).

Connector boards thicker than 0.023 inch should be replaced. The entire assembly, including connector boards and 1/3 SLT boards, must be replaced. The assembly, PN 5400758, is available from Mechanicsburg, Code 34.

4.3.6.5 ROS Parity Checks in Wait State

When ROS parity checks are experienced in wait state, the ROS clock is operating when it should not. To verify this:

1. Disable interval timer.
2. Replace system in wait state as follows:
   a. Depress SYSTEM RESET.
   b. Set DATA key bit 14 to a 1.
   c. Depress STORE.
   d. Depress PSW RESTART.
3. Scope O1C-D3M6 (see logic RX003, net CD4 for pin location). There should be no clock pulses if machine is operating correctly.
4. Depress SYSTEM RESET; P0 clock pulses should be seen.

4.3.6.6 Intermittent ROS Failures

Intermittent ROS failures cannot always be resolved by card replacement. An effort should be made to gather information from logouts and indicator lights to establish the failing pattern. One of the following conditions may be found:

1. Failures in one bit in either the upper or lower word:
   A short may exist between a drive line and a sense line. The shorted sense line can cause failures in all planes, but only the shorted bit line will fail. Refer to 4.3.6.2 for troubleshooting information.
2. Failures in multiple bits in one plane:
   This is probably due to low sense amplifier output, usually caused by plane contamination or low torque problems. The same bit positions in good addresses and failing addresses should be compared for amplitude at the output of the sense amplifiers. Varying ROS bias voltage will usually make low output problems more apparent. Refer to 4.3.6.1 for troubleshooting information.
3. Failures in one bit in one plane:
   Contamination or a bad plane is indicated. When the plane must be replaced, bias should be tried as a temporary repair until parts arrive. This failure pattern could also be an early indication of 2 above.
4. Failures in multiple bits in multiple planes (extra bits):
   May be due to noise problems. Refer to 4.3.6.3.1 for additional information. Can be caused by gating both upper and lower word together, producing an effective OR'ing of the bits of the two words. This is usually a result of late ROS branching in which bit 11 of ROSAR is set too late to control gating of the upper and lower word strobe. Consult CLD's and ROS address listing to verify this. Refer to 4.3.6.3.3 for additional information on late ROS branching.
5. Failures in multiple bits in multiple planes (missing *and* extra bits):
   May indicate ROS power supply irregularities. Multiple driver selection also causes both extra and missing bits. An analysis of addresses which fail and addresses which work usually isolates this problem to incorrect decode of either plane, quarter plane, or drive. The one which does *not* fail is the erroneous decode. A more comprehensive procedure for troubleshooting this trouble is given in 4.3.6.3.4.

4.4 FLT'S

A discussion of FLT's is given in Chapter 2, heading 2.31.3. The operating procedure for the Fault Location Tests is on logic page M8005. If an error stop is encountered, proceed in accordance with the following: heading 4.4.1 for hardcore errors, 4.4.3 for zero-cycle errors, and 4.4.4 for one-cycle errors.

4.4.1 FLT Hardcore Repair

This repair procedure consists of a series of observations and tests to locate a failure in the FLT hardcore. Correct operation of the hardcore is necessary for successful testing of the remainder of the CPU hardware. Before proceeding with the FLT hardcore repairs, be sure that ROS is operating properly (heading 4.3). If any of the following failures occur, proceed as directed:

1. Failure to store 1's successfully in ST. This failure may be identified by inspecting the indicators (roller switches 1 and 2, position 3):
   a. Check that TEST MODE switch is set to FLT position; if it is not, set it and restart the tests.
   b. Trace the failing bit(s), using logic, and repair.
2. Failure to complete IPL (first depression of LOAD). Refer to heading 4.3.4, failure 2, for further details.
3. Storage check. This failure may be identified by the STOR CHK indicator being on. Use the ripple tests

(heading 4.1) to locate the failing address. Roller switch 6, position 4, indicates the type of error.

4. Failure to stop with the specified indications after first depression of RESTART FLT I/O. Refer to heading 4.3.4, failure 4, for further details.
5. Failure to complete IPL (second depression of LOAD).
   a. Check that MANUAL and LOAD indicators are on. The status of these indicators generates the 'IPL or restart I/O' signal, which inhibits the next ROS address and forces ROSAR(11) to be set.
   b. Single-cycle through the hardcore tests. (The contents of the tests are irrelevant when single-cycling to check the FLT controls. However, any value may be stored in the MCW, mask, and data areas to test specific functions.) The single-cycle procedure is as follows:
      (1) Store all 1's in T or store all 1's in main storage at addresses 8040 and 80C0 (hex). All 1's in T will allow single-cycling through the test in buffer 1 only one time. Storing 1's in main storage at 8040 and 80C0 will allow single-cycling through the test in both buffers continuously.
      (2) Depress SYSTEM RESET.
      (3) Set TEST MODE switch to FLT.
      (4) Set RATE switch to SINGLE CYCLE.
      (5) Depress RESTART FLT I/O.
      (6) Depress START. Each depression of START executes one micro-instruction (operation is almost identical with normal single-cycle operation).

Observations and hints:

(1) After RESTART FLT I/O is depressed, the operation stops at a four-way branch on TIC and TN comparison.
(2) When a micro-instruction that contains an enable-scan-bypass operation is being executed, all indicators will light except those associated with the roller switch position(s) being scanned. (See service aid, heading 4.23.)
(3) The immediate test is executed once, and then the test in the other buffer is executed once, etc.
(4) Note that during a storage-request sequence more than one clock cycle is taken with a single depression of START.
(5) When the test stops at the first stop, check the scan mode trigger; it should be set. This trigger gives a different meaning to ROS word bits 17–19 and 25–35. Some bits in the 25–30 field may retain their non-FLT meaning.
(6) Single-cycle operations are useful in checking the scan-in facility. By storing a test of all 1's and observing the indicators during single-cycle operation, the scan-in hardware can be checked.
(7) Various values may be stored in the Mask and MCW before single-cycling to check the operation of suspect controls. The MASK and MCW are in 8008 for buffer 1 and in 8088 for buffer 2. The remainder of the test may be left all zeroes.

6. Failure to loop on failing test. The 'stop scan' line (KU291) normally terminates data transfers from channel to storage and allows the failing test to remain in storage. If this line fails, the CPU stops with the failing test correctly indicated in S but, when attempting to repeat this test, another test will be run. One test in every 180 (the last in a tape record) will loop properly, however. To prevent this failure from going undetected until an FLT-detected CPU error occurs, the following procedure should be used:
   a. Choose a specific FLT and look up the first G/F point in SCOPEX.
   b. Jumper this point to ground.
   c. Run FLT's and loop on the test.
   d. Repeat steps a, b, and c for at least one more test because the chosen test could have been the last in a tape record.

4.4.2 FLT Zero-Cycle Tests

The zero cycle or "scan in-scan out" FLT's check each trigger by first resetting it, then setting it, and then resetting it again. The following areas are checked with these tests:

Reset to triggers.
Scan-in paths to triggers.
Zero and one clock advance.
Ability of triggers to retain their value in the absence of clock.
Scan-out matrix.

4.4.3 FLT Zero-Cycle Repair

When an error stop occurs, the following information about the failure is available in the console indicators:

1. Test number of the failing test (S9–15). This number refers to zero-cycle SCOPEX which gives the name of the trigger and its correct state, set or reset. Refer to logic page M8005 for an explanation of the notation used in SCOPEX.
2. Whether the trigger is actually set or reset. In some cases this is not possible; e.g., ROSAR and ST.
3. Whether failure was solid or intermittent.
4. Whether the stop was caused by an I/O or Storage error.

To isolate the trouble:

1. Set TEST MODE switch to REPEAT.
2. Depress START.
3. Sync the scope on the 'ROS address compare' signal with the address of the last micro-instruction (see Figure 4-6 for scan-in word and ROS sync address)

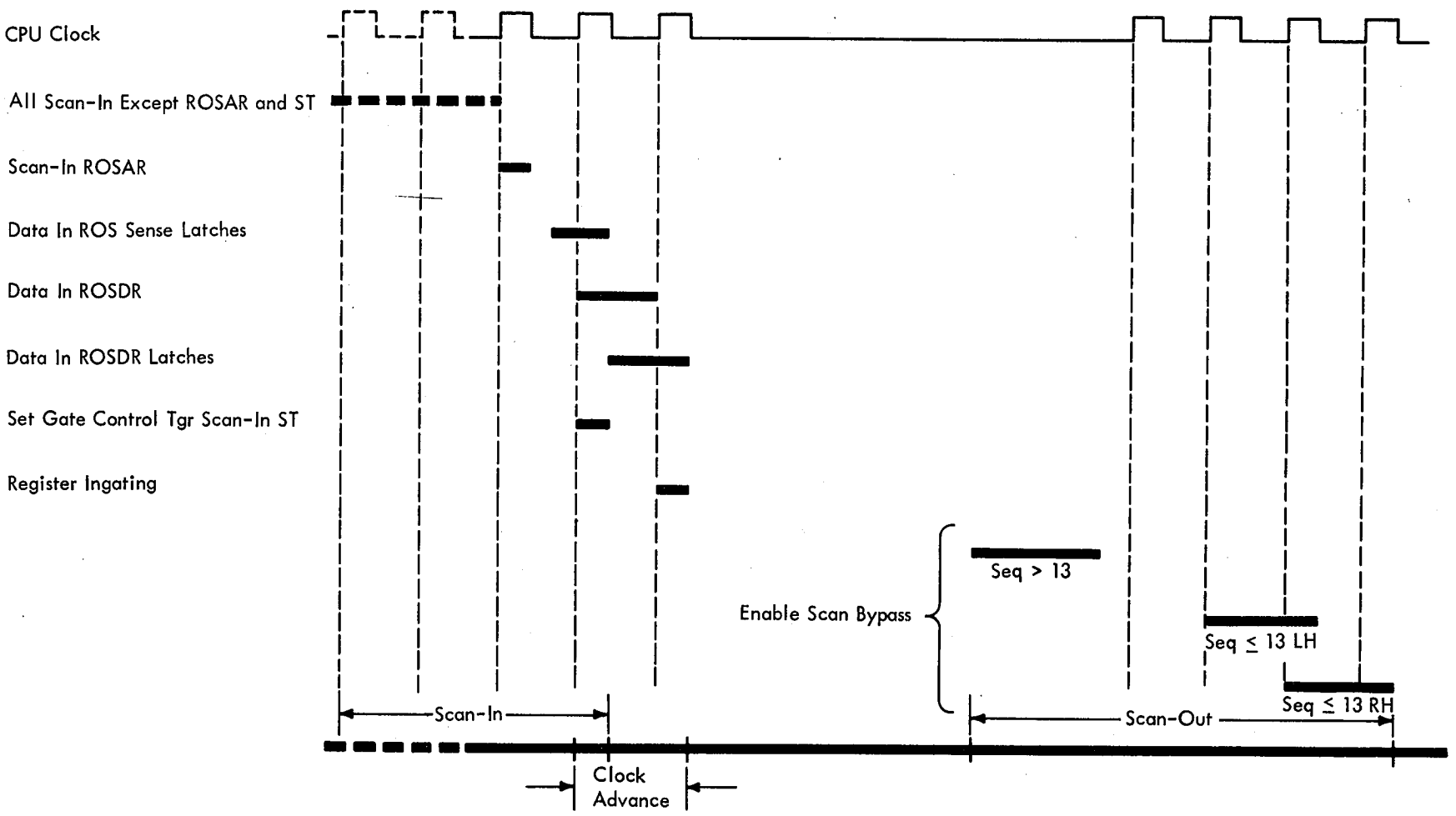


Figure 4-6. Sync Points for FLT Zero-Cycle Troubleshooting

before the test cycle count set in the ADDRESS switches and scope the point called out in the zero-cycle listing.

4. For use as a timing reference during the scan-out operation, display the 'enable scan bypass' signal on the scope; refer to Figure 4-7.
5. SCOPEX indicates whether the trigger is set (S) or reset (R) and whether the point called out in the listing is a 0 or a 1 (0 is minus, 1 is plus).
6. Divide the failure three ways:
   a. The trigger is not being scanned to the proper value (step 7).
   b. The trigger is being set or reset correctly at scan-in time but does not retain this value until scan-out time (step 8).
   c. The trigger is at the correct value during scan-out so the error must occur during scan-out and result comparison (step 9).
7. If the trigger is not being scanned to the correct state, the cause could be bad scan-in or a faulty trigger. If so:
   a. Go to the ALD's at the page called out in SCOPEX and scope the inputs from scan (gate and bit). This is sometimes through normal data paths; e.g., R and E register triggers. If the gate and bit are present, replace the card(s) in the trigger. This applies also to a reset, except that it must be determined that none of the inputs are present to override the reset.
   b. If the gate or the data bit is missing, trace back while syncing on the micro-instruction that scans into the trigger in question (Figure 4-6). The word containing the bit may be found in scan-in and logout bit assignments shown on logic pages M3021-M3061 and M3071-M3081.
8. If the failing trigger is initially set to the correct value and changes before it is scanned out, it could be caused

Scan-In Word	Register Scanned	ROS Address (QY021)	Comments
7	R and E	9B0	Storage request
		846	Q(0-15)→R (Normal data paths are used.)
		845	R→E (Normal data paths are used.)
		844	Q(16-31)→R (Normal data paths are used.)
6	PSW and IC	845	
5	D, Stats and Misc	844	
4	B and Misc	843	
3	A and Misc	842	
2	Q	842	Two cycles before ingate.
1	MCW	840	
0	S and T	150	

Figure 4-7. FLT Timing

by erroneous interaction between triggers. In all cases, where possible, random values have been placed in the other triggers.

a. Go the ALD's at the page called out in SCOPEX and scope the inputs and resets to the trigger. The time period of interest is after scan-in and until after scan-out.
b. This is best done by syncing on the ROS address that performs the scan-in. This may be determined from scan-in and logout bit assignments shown on logic pages M3021-M3081.

9. If the trigger is set to the correct state at scan-out, do the following:
   a. Go to the logic page called out in SCOPEX and scope the output of the trigger. If the point called out is the output of the indicator driver and the indicator is correct, it is not necessary to scope this point.
   b. Go to the KT page fed by the indicator driver and scope the AND fed by bit and roller position or scan word. This scoping should be done with the roller containing the trigger in question turned to another position.
   c. If the bit is not present, trace back through the circuitry to the indicator driver and make the repair.
   d. If the gate is not present, trace back to the address sequencer. The address sequencer and left half bit are set from word 1 of the test data during scan-in with address sequencer equal to zero.
   e. If the gate is present, go forward to the correct KT8XX page and check the output on the right side of the page.

4.4.4 FLT One-Cycle Repair

When a one-cycle error occurs, the test number, shown in S(0–15), refers to the one-cycle listing of SCOPEX; see logic page M8005. In addition, the nature of the test failure is indicated:

1. CPU failure. Continual (FAIL indicator on) or intermittent (PASS, FAIL, or INTRM indicators on).
2. Storage failure. STOR CHK indicator on.
3. I/O failure. Channel (CCC indicator on) or I/O unit (UDC indicator on).

When a failure occurs, proceed as follows:

1. Locate test in SCOPEX and determine number of cards signified as G/F, G/FI, and those listed at end of test:
   a. If four or fewer cards are signified, replace them as described in step 2 below.
   b. If five or more cards are signified, scoping is required. The technique for scoping is described on logic page M8005.
   c. If the timing information contains an * or a ► in the value column, refer to the logic because the feedback loop of the trigger or latch involves two or more cards.
2. If four or fewer cards are involved, replace them as follows:
   a. Determine part number of each card from ALD or card. Obtain one new card of each type.
   b. Change each card, one by one:
      (1) Do not turn off power.
      (2) After changing a card, set TEST MODE, REPEAT switch and depress START. This operation resets the pass and fail triggers and repeats the failing test.
      (3) If the action taken in step (2) did not correct the trouble (PASS indicator on and FAIL indicator off), reset TEST MODE, REPEAT switch and replace the original card.
      (4) Repeat steps (2) and (3) until all suspected cards have been changed or the trouble is corrected. If card replacement does not correct the trouble, proceed with step 3 below. If replacements made this test pass, restart FLT's from the beginning.
3. If five or more cards are involved or card replacement does not correct the trouble, isolate the trouble by using the scoping technique described on logic page M8005. Note that the scan-out operation occurs either at 800 ns after the last clock cycle or after the clock has been started and three cycles have passed. The latter is true for all triggers that are scanned out by the address sequencer being equal to 13 or less. The triggers that are scanned out by the address sequencer being equal to 14 or more are ROSAR, ROSDR, GCT's, PAL, and ST. These triggers will change state if the normal clock is allowed to start before they are sampled. Use the 'enable scan bypass' signal for reference; refer to Figure 4-7.

4.4.5 2065 SCOPEX Timings

This discussion is to facilitate the CE's understanding of the SCOPEX timings shown on logic page M8005. It consists of an explanation of times A, B, C, and D with reference to what is actually happening in the machine. Throughout this write-up, the term "Data" shall mean either a 1 or a 0.

The 2065 works on the basic principle that data flows from trigger to latch to trigger, and that the triggers are set at clock time and latches at not-clock time. The FLT's generally test a data path from one trigger, through an associated latch, and into another trigger. From here it is scanned out to "T" and checked. Only two clock pulses are required to perform this function. One clock pulse is needed to set it into the first trigger and address some ROS word to bring up appropriate gates to pass the data to the next trigger. The not-clock pulse passes the data from the first trigger to its latch. The second clock pulse sets the data into the trigger to be tested.

When synced on ROS address 150 and displaying three clock pulses, the first pulse seen on the scope is actually the one at address 150 and scan-in is just being completed. The micro-orders to scan into S & T are in CAS block 150, but these triggers are not actually set until the next clock pulse; i.e., the second clock pulse seen. This second clock pulse is the one of interest. It is at this time that the first trigger in the path will have data set into it (as in the case of S & T) unless it has been previously scanned into.

Now the first timing in SCOPEX can be discussed. It is the last shown on logic page M8005, time D. When SCOPEX states a line should be up during D, it must be up shortly into the second clock pulse and remain up until almost the same time in the third clock pulse. It may have been up solid at the start of the second clock pulse if it was scanned into previously, but it must remain up at least during the second clock and not-clock pulse to transfer the data to the latch.

The second time is C. This is the time the data is passing through the latch. The data must be up at least by the second not-clock pulse and extend through the third clock pulse to allow it to gate into the next trigger.

The third time is A. This means that the line should come up sometime during the third clock pulse and remain up. It is from here that the data will be scanned out.

The fourth time is time B. This is usually a gating line and may have a clock pulse associated with it, but it must be up during the third clock pulse. It is the gate which transfers the latch data into the trigger.

Note that when working with certain test segments there will be only two clock pulses showing. When this is true, times A and B pertain to the third pulse shown, time C to the first not-clock pulse, and time D to the first clock pulse. The CE should follow the SCOPEX with the ALD's open, whenever possible, so he may more intelligently decide what is correct.

Figure 4-8, A, is a pictorial summary similar to that shown on logic page M8005. It illustrates a typical FLT that will test the ability of the machine to transfer bit 63 of A, through the parallel adder, into position 63 of T. To do this, bit 63 of B will be set on during the scan-in operation. When scan-in has been completed, the CPU clock will be permitted to run for two cycles.

During the first clock cycle (Figure 4-8, B), the ROS word contains micro-orders necessary to gate bit 63 of B through the adder and into position 63 of T, where it will be decoded and executed. The second clock cycle is needed to accomplish ingating to T.

During the second clock cycle, the ROS sense latches are held reset, thereby preventing decode of the second ROS word. This prevents destroying the data set up in the previous cycle.

4.5 DIAGNOSTICS

After a successful run of the ROS tests and FLT's, run diagnostic programs with the marginal power supplies in the CPU, storage units, and channels set from nominal to higher or lower output. (Refer to Chapter 5, heading 5.3, for marginal checking procedure.) If a potential trouble spot is known, run the diagnostics for that area; if not, run Hard Core, DME with the CEDA sections, and SIP. SIP should be run for a minimum of 15 minutes and preferably for 1 hour without failure.

A discussion of the diagnostics is given in Chapter 2, heading 2.32. For detailed operating procedure, refer to:

1. *System/360 Diagnostic Program General Reference Manual,* PN 5396300.
2. *DME User's Guide,* PN 5763442.
3. The program writeup, flowcharts, object deck, and listing provided with each diagnostic program.

4.6 TIMING CHECKS

The timing procedure and checks for the CPU delay lines and singleshots are found on logic pages M8001, M8002, and M8008. ROS timing procedure is on logic page M8003. Timing procedure for LCS, 2065 IH or J, and the multiprocessor delay line oscillator is on logic page M7999.

4.7 LAMP TESTS

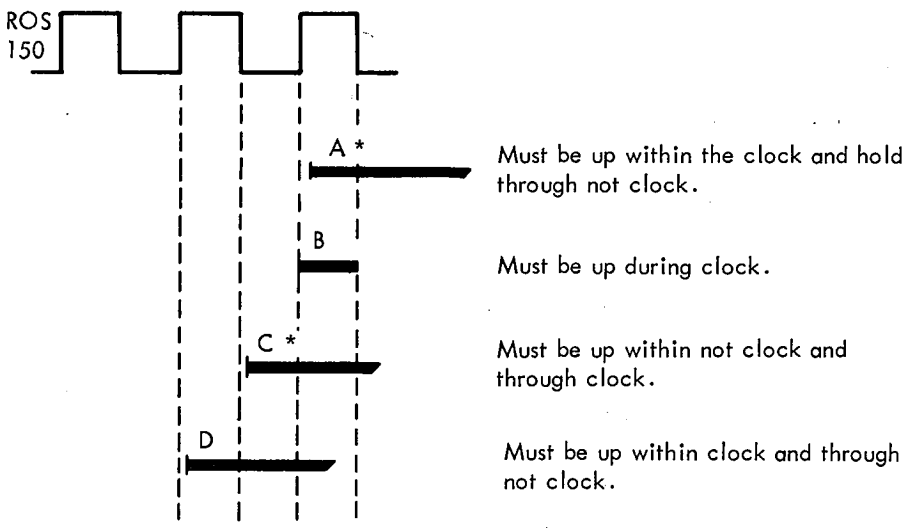
All lamps on the system control panel associated with a particular roller switch light when the roller switch is between detents. All lamps on the system control panel not associated with the roller switches or power light when roller switch 6 is at position 6.

The POWER CHECK lamps (including the red lamps backlighting the POWER ON pushbutton) normally light during power-on sequencing and go out after normal power is on. The MARGIN ACTIVE lamp lights whenever a marginable power supply or unit is being marginally checked. The MARGIN LOCATE lamp lights whenever the MARGIN/METER SEL switch is set to the power supply or unit being marginally checked.

The lamps are pluggable and may be removed from the front of the panel with the lamp tool (PN 461420).

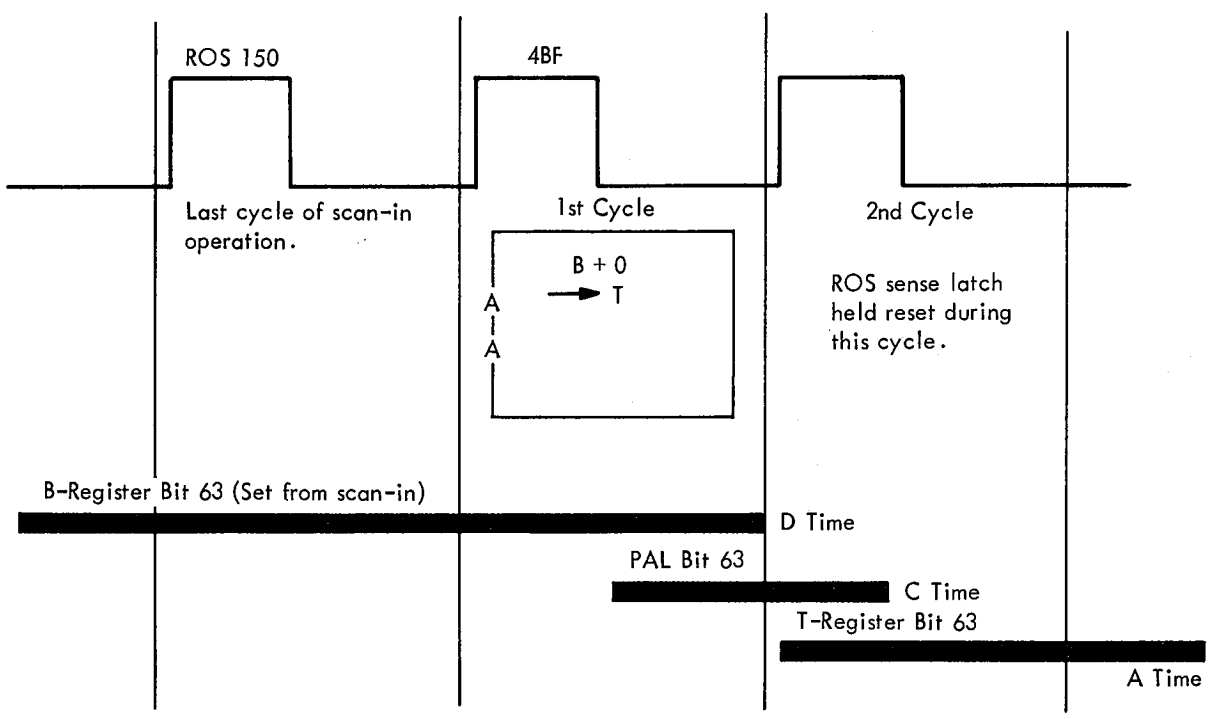
4.8 SIGNAL TO FRAME GROUND SHORTS; SERVICE CHECK

There should be a single frame ground connection for the Model 65 CPU, channels, and storage. That is, all dc returns in the CPU, storage, and channels are isolated from each other except at one common frame connection. This connection between electronic ground and the machine frame is located in frame 02 at the dc return bus



Note: * Denotes triggers or latches where portions of the associated "latch back" circuitry will be found on more than one card.

a. Understanding of Scopex Timings



b. Typical Example

Figure 4-8. SCOPEX Timing

(green/yellow wire). Additional frame grounds cause unexplained errors; for example, ROS parity checks or storage checks.

Note: On a two-processor system, one processor ground is not connected.

The system should be checked for signal-to-frame ground shorts every six months as well as during and after installation. The following procedure should be used:

1. Disconnect the green/yellow wire from the common frame ground in frame 02.
2. Connect an ohmmeter, set to the highest range, between the green/yellow wire and the machine frame.
3. Disconnect all I/O interface cables.
4. Disconnect all EPO cables from channels to control units.
5. Check that the meter indicates a resistance approaching infinity.
6. If extraneous signal-to-frame grounds are present, they must be removed. With the same setup as described in steps 1–5, first uncable the channels. If the short still exists, begin removing the dc returns for each gate in the CPU until the short is isolated to a gate. Cable clamps are common sources of signal-to-frame ground shorts. The clamp screw or a burr on the clamp may cut through cable insulation and short to the cable shield. The clamp may be excessively tight, causing the insulation to migrate away, thus permitting the cable shield and clamp to short together. These clamps can be loosened one at a time until the short is found. Especially check the thermal cable that runs from the top of gate D to the card section of gate C below the ROS unit.
7. Reconnect all cables after the short has been removed.

4.9 ROS OPTIMIZATION

The ROS timing and bias are optimized by a procedure called "shmooing". A ROS shmoo is a two-dimensional plot (bias vs time) of the failure points of ROS. The shmoo identifies the optimum operating point of both the bias voltage and the time of the leading edge of the strobe signal.

4.9.1 Adjustment Procedure

The adjustments and shmooing procedures are found on logic pages M8003 and M8004. Figure 4-9 shows the waveshapes for a typical 1 and 0 bit. Basically, the shmooing procedure is as follows:

1. Adjust timing in accordance with the procedure in steps I through VII on logic page M8003. Prepare a plot of the

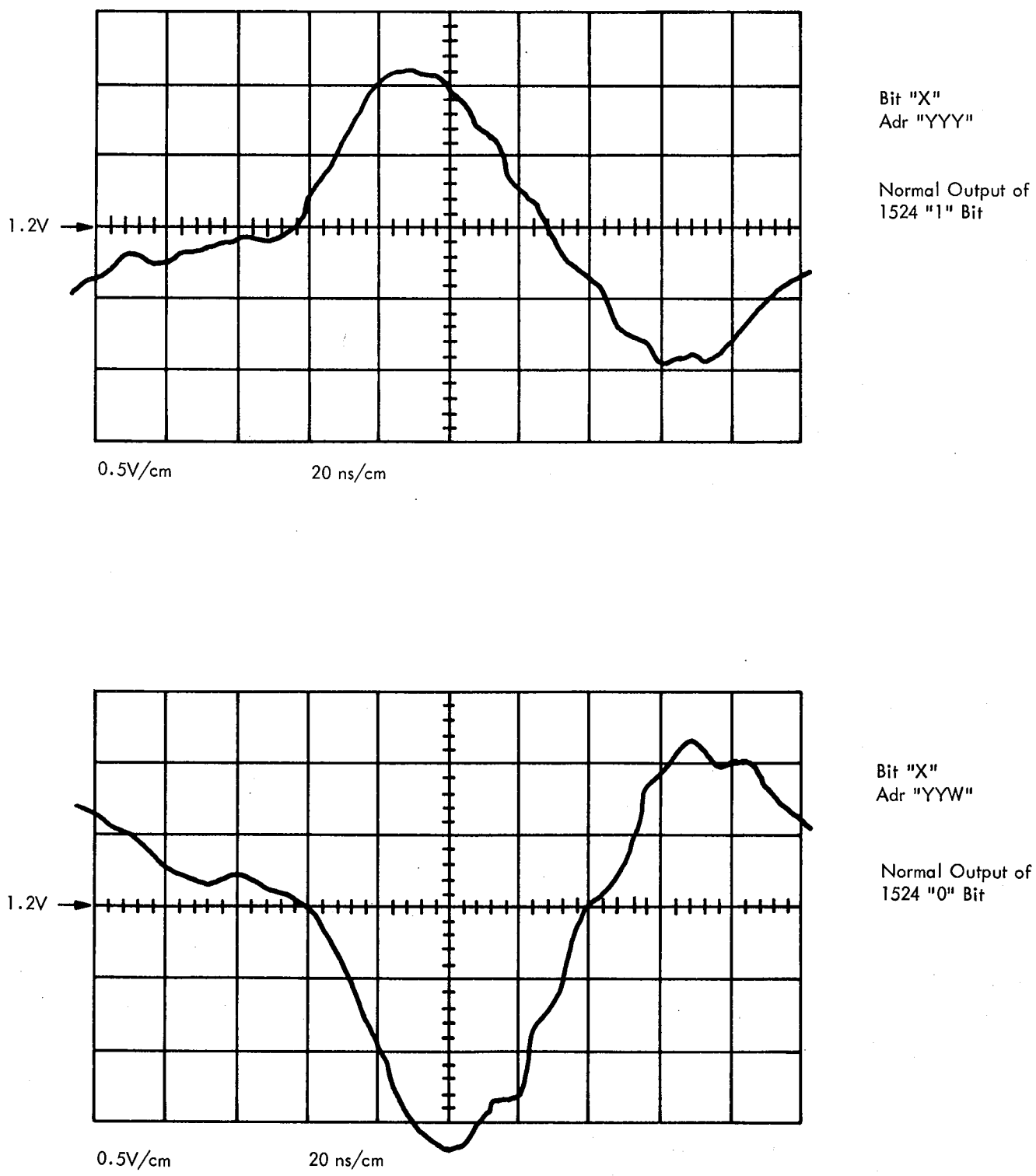


Figure 4-9. ROS Sense Amplifier 1 and 0 Bit Waveforms

bias vs time failure points while running diagnostic program number E330.

2. Vary the strobe position about its initial time value in 5-ns increments while the ROS is being exercised (E330). At each time setting, vary the ROS marginal panel control from –7V to –20V, recording both the upper and the lower V_M (–18 variable supply) values at which ROS error occur. At each failure point, also record the failing ROS address and bit. This data should be plotted on a graph (shmoo). A typical shmoo is illustrated by Figure 4-10.
3. The optimum value of strobe timing and V_M is determined by placing a rectangle of area 6V by 15 ns within the area defined by the failure points of the shmoo. The rectangle should be vertically centered within the shmoo while maintaining the rectangle area error-free. It may be to the left or right of center, depending on the available delay taps. See Figures 4-11 and 4-12.
4. If the rectangle cannot be placed within an error-free area as defined above, the ROS must be debugged. This will require bit and sense plane cleaning. Heading 4.9.2 defines the prescribed debugging procedure.
5. When the machine has been debugged satisfactorily to allow successful completion of steps 2 and 3, the optimum strobe timing and ROS marginal panel control setting will be the center point of the rectangle. If this point does not lie on a value of strobe timing attainable with the –5-ns increments of the delay line, the operating point should be on an incremental value as close as possible to the indicated point. It may be to the left or right of center.
6. With ROS set at optimum strobe timing, ROS marginal panel control is to be varied ±20% of its optimum value (all other logic voltages are left at nominal); the ROS, being again exercised in all addresses, will run error-free during this ROS marginal panel control variation.
7. If step 6 is successfully completed, return ROS marginal panel control to its optimum value. The optimized values of strobe timing and ROS marginal panel control

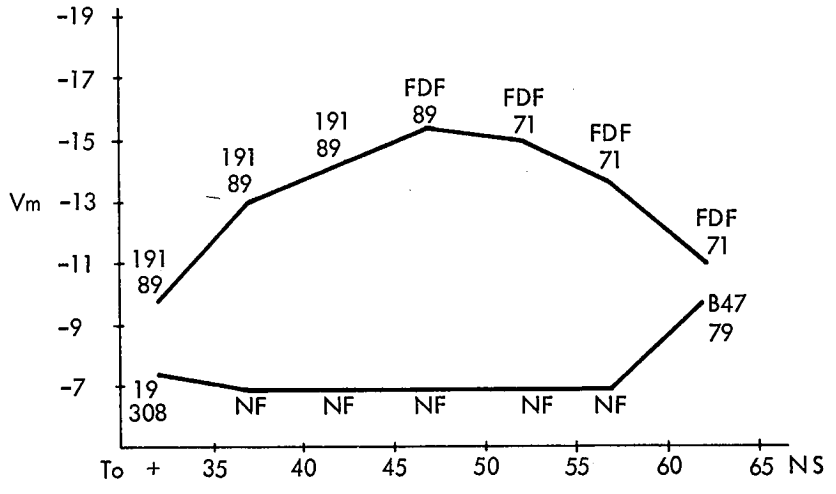


Figure 4-10. Typical Shmoo

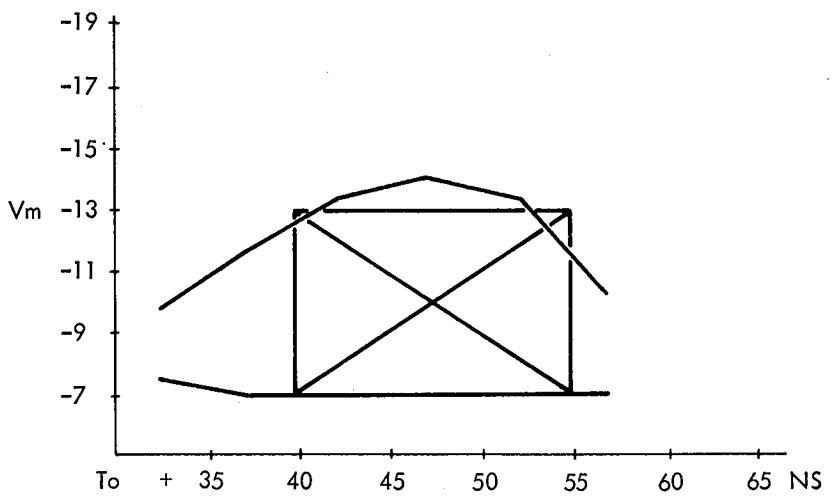


Figure 4-11. Unacceptable Shmoo

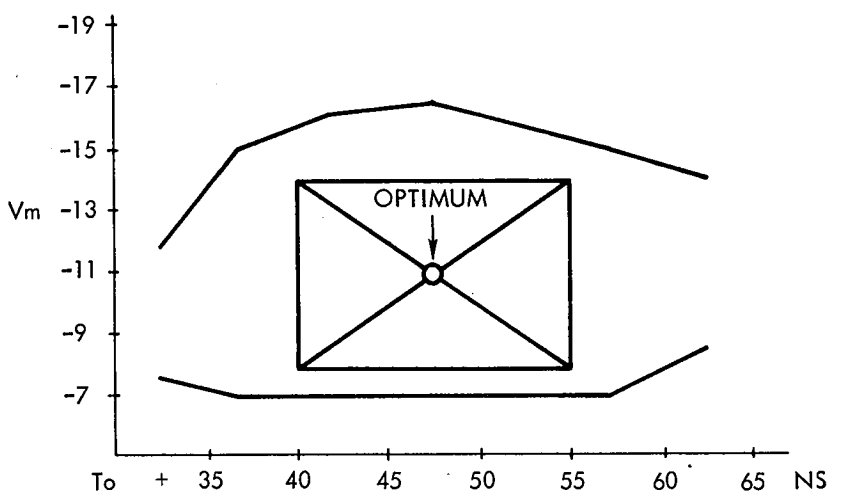


Figure 4 12. Acceptable Shmoo

are the new nominal operational settings for the ROS. If step 6 cannot be successfully completed, the optimization procedure must be repeated.

Note: The lower operating voltage (10–12V) of the new sense amplifiers gives added importance to the adjustments in ROS and the cleanliness of the ROS planes.

4.9.2 Debugging Procedure

This procedure shall be followed if the conditions of steps 2 and 3 of heading 4.9.1 cannot be successfully met:

1. The shmoo shown in Figure 4-10 is a typical example of an area defined by failure points. Note that the recorded bits and addresses in error tend to be repetitive within the segments on either side of the peak of the shmoo. If only one or two bits are repeatedly causing failures throughout the shmoo, an attempt may be made to improve the shmoo by replacing or exchanging sense

amplifier cards. If three or more bits appear to be causing failures, or if replacing the cards (for one or two bits) does not sufficiently improve the shmoo, scope the bits to verify which, if any, is weak (see Figure 4-13). Repeatedly failing planes should be cleaned first.

2. Remove the ROS planes indicated by the shmoo failure points and clean thoroughly according to 4.13.1, 4.13.2, and 4.13.3. When the cleaned planes have been replaced, re-evaluate the shmoo according to steps 2 and 3 of 4.9.1. If these steps are successful, continue with the remaining steps of the optimization procedure.
3. If the improvement obtained by the initial cleaning is not sufficient to meet the conditions of steps 2 and 3 of 4.9.1, repeat step 2 above.
4. If, after the second cleaning, steps 2 and 3 of 4.9.1 cannot be successfully completed, again verify seating of the amplifiers. If all cards prove to be properly seated, and optimization still is not possible, replace the amplifier card(s) in the position(s) of the predominant failing bit(s), and repeat the entire optimization procedure.

4.10 ELAPSED TIME METER CALIBRATION

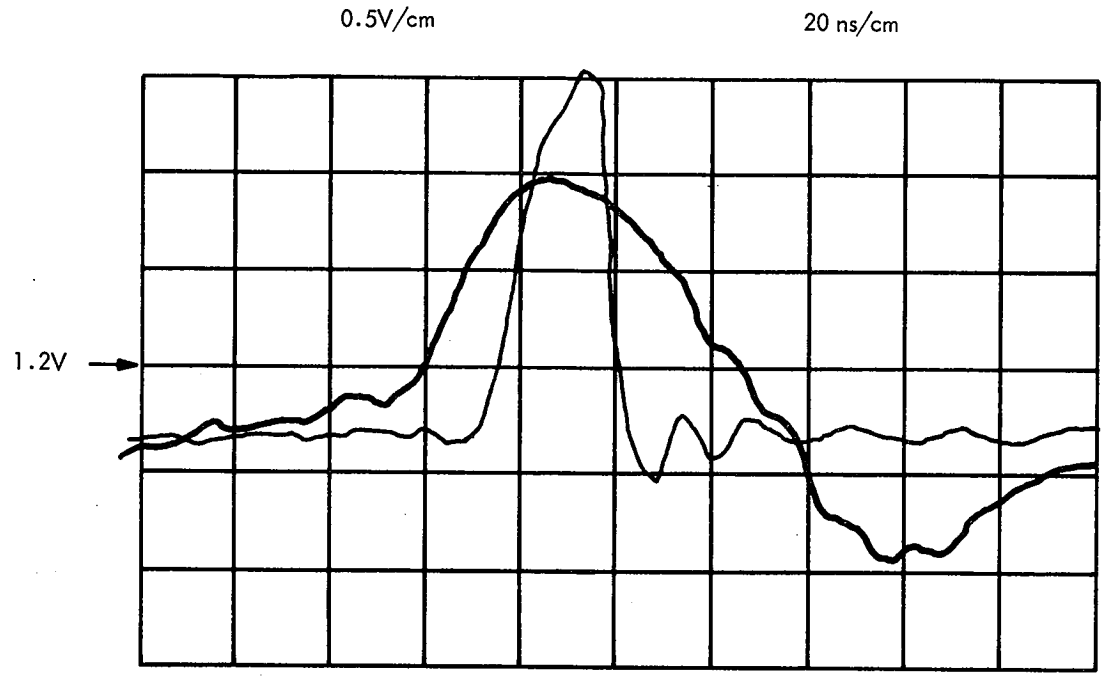
The elapsed time (usage) meters on the system control panel are calibrated by the Elapsed Time Meter diagnostic section (program number F38F). The operating procedure for this section is provided in the listing and writeup.

4.11 1052 SINGLESHOT ADJUSTMENTS (1052 Attachment Feature only)

Measure all time durations at the 1.5V level. Before adjusting singleshots 1 through 7, turn on CE MODE switch and place CONTINUOUS READ/WRITE switch in CONTINUOUS WRITE position.

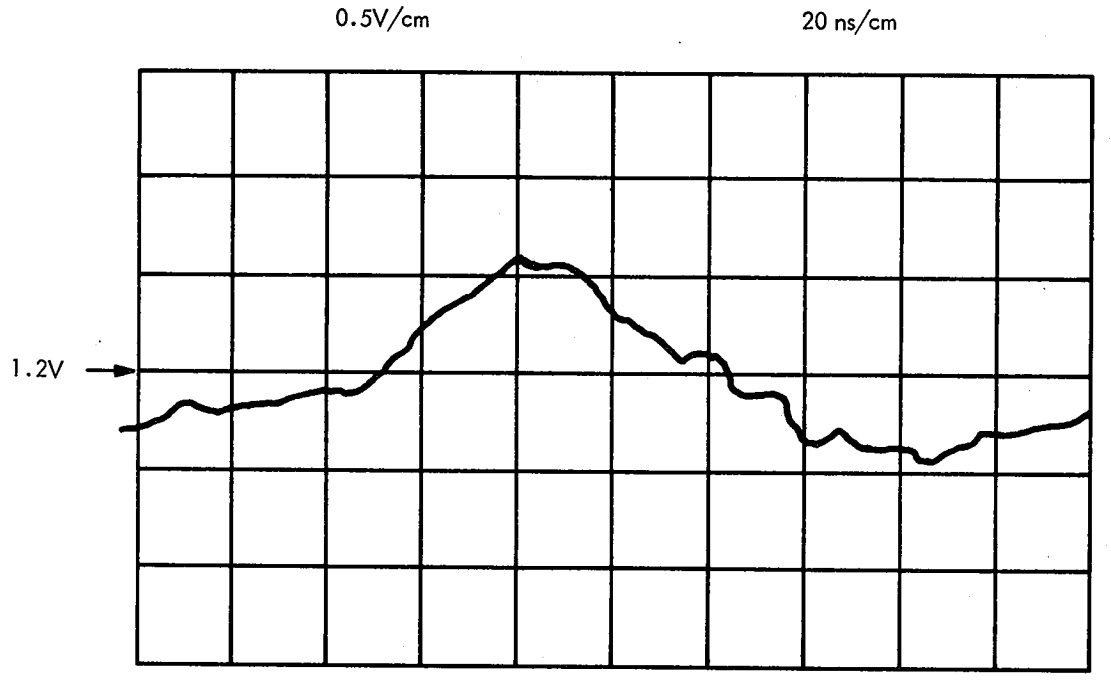
Logic Page	SS Desig	Scope Point (Note 1)	Adjust	Duration	Tolerance
PG 601	1	G7D13	Center	700ns	±35ns
PG 601	2	G7B13	Lower	200ns	±10ns
PG 601	4	E7D13	Center	200ns	±10ns
PG 601	5	D7D04	Center	28ms	+1.4 -0ms
PG 601	6	E7B13	Lower	500ns	±25ns
PG 601	7	E7D12	Upper	200ns	±10ns
PG 621	Alarm (Note 2)	D7D06	Lower	30ms	±1.5ms
PG 621	Ready (Note 3)	G7D12	Upper	200ns	±10ns

Note 1: Singleshots are located in 01E-E1 for first 1052 adapter, 01E-C1 for second 1052 adapter, and 01A-C1 for 2150.
Note 2: Key in program to ring alarm bell.
Note 3: Depress READY and then NOT READY. Depress SYSTEM RESET between each ready-not ready sequence.



Average 1524 SA Output

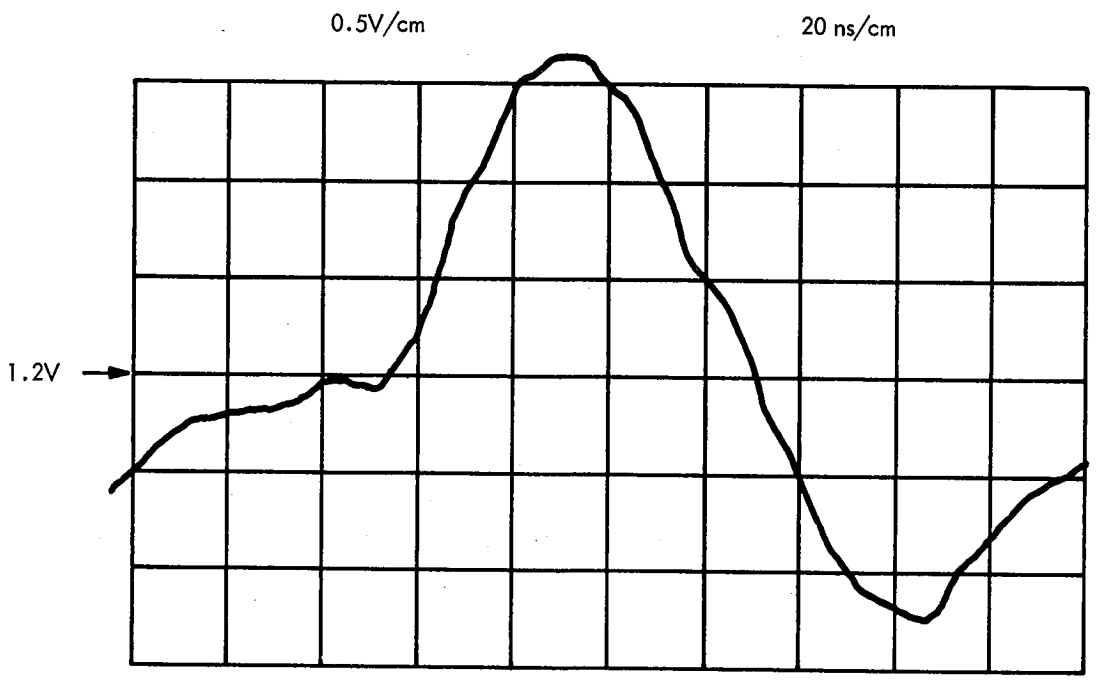
Strobe at the input to the sense latch.



Weak 1524 SA Output

Causes
- Weak sense amplifier
- Dirty bit plane
- Loose spider pressure

This bit will operate at nominal voltage.



Strong 1524 SA Output

This output is obtained when all conditions are at best case.

Figure 4-13. ROS Sense Amplifier, Average/Weak/Strong Waveshapes

Observe the following general hints on removal and replacement of assemblies:

1. Turn off power.
2. Use the right tool for the job.
3. Get help if the assembly is heavy.
4. Pay attention to the order of parts (spacers, washers, etc.) so that installation will be correct.
5. Tag wires as they are removed. Do not trust your memory.

4.12 LARGE CARD REPAIR AND REPLACEMENT

The procedure for repairing and replacing the large cards is detailed in *SLT Packaging, Tools, and Wiring Change Procedure,* FEMI, Form 223-2800.

4.13 ROS BIT PLANES

The following paragraphs discuss removal, cleaning, and installation of ROS bit planes.

4.13.1 Removal

The following procedure for removal of the ROS bit planes must be carefully followed:

1. Release pressure on the six pressure plates by loosening the 12 torque screws in the spider (Figure 4-14). Use standard bristol wrench or torque wrench (PN 461450). Do not remove torque screws; just relieve pressure. Do not loosen retaining screws.
2. Release pressure on upper and lower drive connectors by loosening pressure hex screws (four per connector). Do not remove pressure hex screws; just relieve pressure.
3. Remove the two retaining screws on both upper and lower drive connectors.
4. Remove the four corner hex screws on the spider.
5. Remove the two knurled nuts on the spider. They may be removed by hand.
6. Remove spider from two center studs. The six pressure plates are held to spider by the 12 retaining screws.
7. Remove rubber pressure pad from the two center studs.
8. Wearing lint-free gloves (PN 461421), carefully remove bit plane from alignment pins and center studs by holding at center of each horizontal edge. Plane must be held and gently lifted without being allowed to bow or touch anything.

CAUTION
The bit plane must not be creased, folded, scratched, allowed to come in contact with oil (including skin oils), or subjected to dust. The same is true of the sense lines.

4.13.2 Cleaning

The following procedure for cleaning the ROS bit planes must be carefully followed just prior to installation of these bit planes:

1. Wearing lint-free gloves (PN 461421), prepare a large flat surface, at least as large as bit plane, by washing it with lint-free cloth (PN 461622) thoroughly saturated with cleaner solution (PN 450608). Freon* should not be used.
2. Clean sense plane and upper and lower connector tabs by washing them with lint-free cloth thoroughly saturated with cleaner solution.
3. Place bit plane on prepared surface, Mylar* side up.
4. Fold new lint-free cloth into pad about 3 or 4 inches square and saturate it with cleaner solution.
5. Rub the bit plane with pad over entire surface, starting at one end and rubbing back and forth over entire length. If pad begins to dry, reapply cleaner solution.

   CAUTION
   Under no circumstances should the cleaner solution be applied directly to the bit plane or sense plane.

6. Immediately after drying, install bit plane.

4.13.3 Installation

The following procedure for the installation of clean ROS bit planes must be carefully followed:

1. Turn the 12 retaining screws on spider CCW until the six pressure plates are loose and flat against spider. Do not back screws out past this point.
2. If original bit plane is to be installed, clean both bit and sense planes, as described in 4.13.2. If a new plane is to be installed, clean just the sense plane. Then, wearing lint-free gloves, remove new bit plane from sealed bag by cutting off sealed end with scissors.

   CAUTION
   Under no circumstances should cleaning solution be applied directly to the bit plane or sense plane.

*Trademark of E.I. du Pont de Nemours & Co. (Inc.)

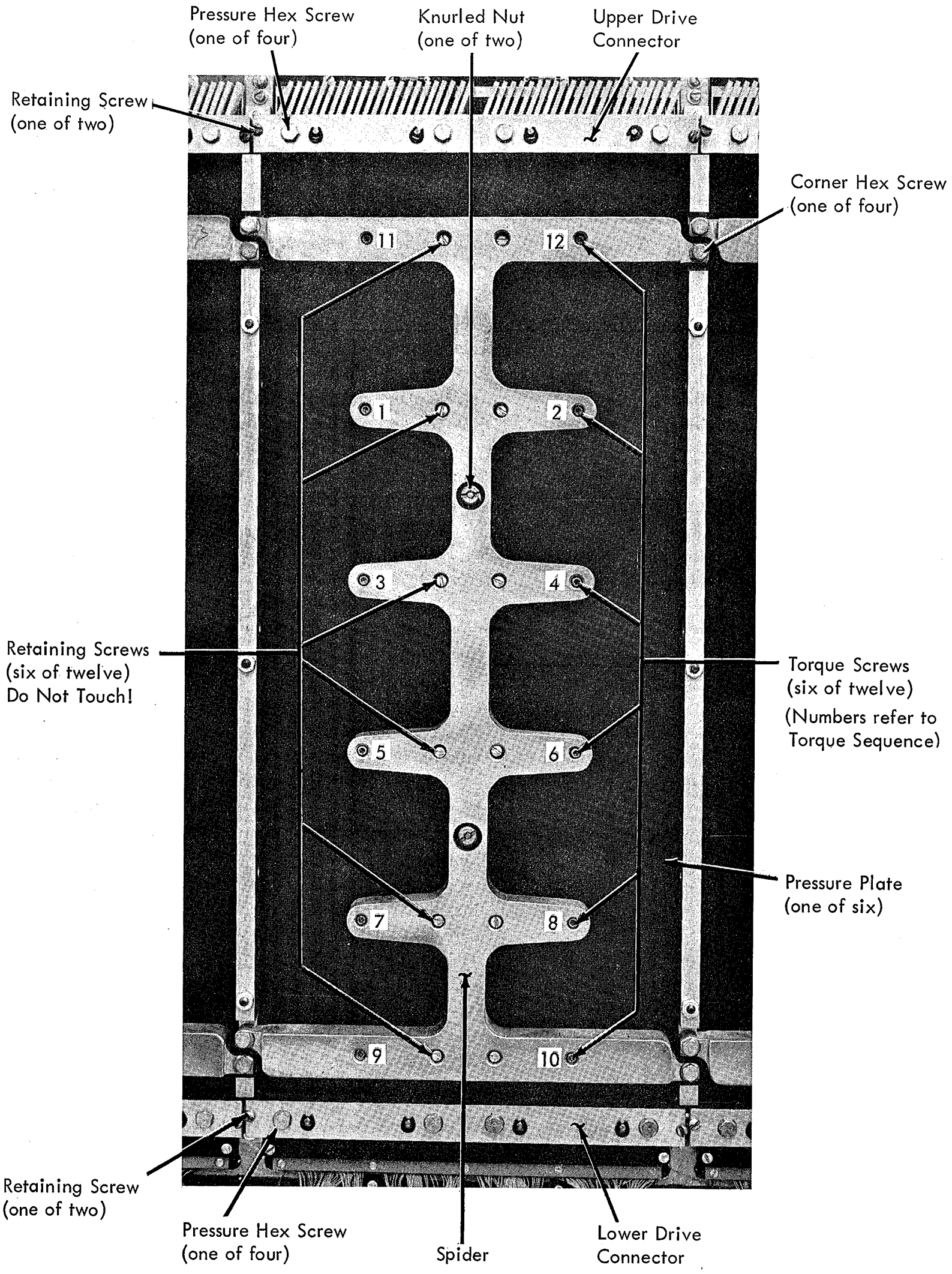


Figure 4-14. ROS Pressure Spider and Torque Sequence

3. Immediately place bit plane over the two center studs, Mylar* side against sense plane, with alignment hole up and slot down. Gently press copper land around upper alignment hole flat against sense plane and then do same for lower slot. (Upper hole provides both horizontal and vertical alignment; lower slot provides only horizontal alignment.) Bit plane must be flat against sense plane. Check registration at top and bottom of bit plane. Drive line flag must completely cover sense line at both top and bottom of bit plane. (Use sense line for bit 0 upper and bit 90 upper for this check.) If the registration is not correct, push the bit plane up as far as possible. If the registration cannot be corrected, replace the bit plane and check registration again. If it is still incorrect, contact the plant through your area office.
4. Install both upper and lower drive connector pressure bars with the two retaining screws on each. Do not tighten pressure hex screws. Recheck that bit plane is flat against sense plane and that registration is correct.

   Note: Clean the rubber surface of the pressure bars before installation.

5. Install rubber pressure pad over the two center studs.
6. While holding rubber pressure pad in position, place spider and pressure plate assembly over the two center studs.
7. With slight upward pressure on spider, align it with corner hex screw holes. Install the four corner hex screws, but do not overtighten them; just be sure they are bottomed and snug.
8. Install the two knurled nuts, but tighten them no more than fingertight.

CAUTION
The two knurled nuts must be installed before torqueing and must not be more than fingertight or the spider may fracture as it is loaded.

9. Using torque wrench (PN 461450), tighten the 12 torque screws to 2-inch pounds in the sequence shown in Figure 4-14.
10. In the same sequence, tighten torque screws 1–8, 11, and 12 to 4-1/2 inch-pounds and tighten torque screws 9 and 10 to 3-1/2 inch-pounds.
11. In the same sequence, check that torque on torque screws 1–8, 11, and 12 is within 4 to 5 inch-pounds and that torque on 9 and 10 is within 3 to 4 inch-pounds. If any torque screw is not in required range, retorque in same sequence, and recheck.
12. Remove the two retaining screws on both upper and lower drive connector pressure bars.
13. Check horizontal alignment of bit plane drive line connector tabs with tabs on terminating resistor card at top of plane and tabs of drive card at bottom of plane. Both cards may be shifted horizontally for alignment with bit plane tabs by loosening the two hold-down screws in each.
14. Problems have been encountered in alignment of bit plane connector tabs with tabs of terminating resistor card at top of array. If alignment is not possible by loosening the two screws of terminating resistor card, remove card from machine and elongate the two mounting holes. This can be done with a small, fine-tooth, rat-tail file. When filing hole, be careful not to chip or crack corner of card. A firm grip with two fingers as close to hole as possible will prevent this.
15. Check vertical position of drive connector board. Drive tabs on board must be visible below bit plane. If tabs are completely covered by bit plane, drive board should be moved down as far as possible. If this adjustment cannot meet position requirements, board should be replaced. A temporary repair can be made by placing tape over diagonal land patterns on drive board.
16. Install both upper and lower drive connector pressure bars with the two retaining screws on each.
17. Tighten the four pressure hex screws on each pressure bar until they are bottomed and snug. Do not overtighten.

Section 4. Service Aids

This section contains a collection of miscellaneous service aids which may point out a troubleshooting approach to a particular problem when FLT's and diagnostics have not isolated the trouble. To help the CE work more effectively, some common troubleshooting pitfalls are called out here together with positive suggestions for avoiding them:

1. Make full use of console indicators. Much time can be lost by scoping points which are already indicated on the console. Lamp test should be used often to avoid being misled by burned out indicator lamps.
2. Determine early whether the trouble is broad or localized so that as much circuitry as possible can be eliminated.
3. Choose scoping points so that the suspected circuitry is cut in half rather than proceeding block by block.
4. Keep the approach to a trouble as simple as possible by seeking the simplest operation that will fail. Do not key in long instruction loops by hand if such loops are available in the diagnostics.
5. When no progress is being made, recheck the most basic assumptions. For example, the original symptom may no longer be present or it may have been a result of an error in operating procedure. Some common operator errors are given here:
   a. RATE switch not in PROCESS position.
   b. Incorrect parity in general purpose or floating point registers.
   c. Wait bit on in current PSW.
   d. Program interrupt with no program new PSW stored. The IC contains 8 or A since the blank new PSW itself causes a program interrupt when no valid instructions are at location zero in storage.
   e. Program interrupt commonly caused by specification error in instruction or by incorrect protect key in current PSW while executing store-type instructions.
6. If trouble is difficult to isolate, consider the more remote possibilities. More than one trouble may be present or, less likely, more than one failing component may be responsible for a single symptom.
7. Several seemingly unrelated symptoms can result from a loose card or a loose flat cable connector.

4.14 INTERMITTENT PROCESSOR CHECKS

Unpredictable and often misleading processor checks can result from a glitch on the error trigger line on logic page KW051. This can be caused by a card with a slow switching time. The glitch may not be of sufficient duration to set the error latch and yet may be propagated into the clock circuitry on logic KC011 and KC051, causing unpredictable error indications. The following procedure may be helpful:

1. Attempt to make the trouble less intermittent by running floating-point diagnostics while very slowly biasing B-gate in the CPU.
2. Scope 01E-E2L3B03 (logic KW081) while looping on a routine or portion of customer's program that has failed at least once. The trouble will appear as a positive glitch on an otherwise steady level.
3. Trace glitch back to slow card and replace card.

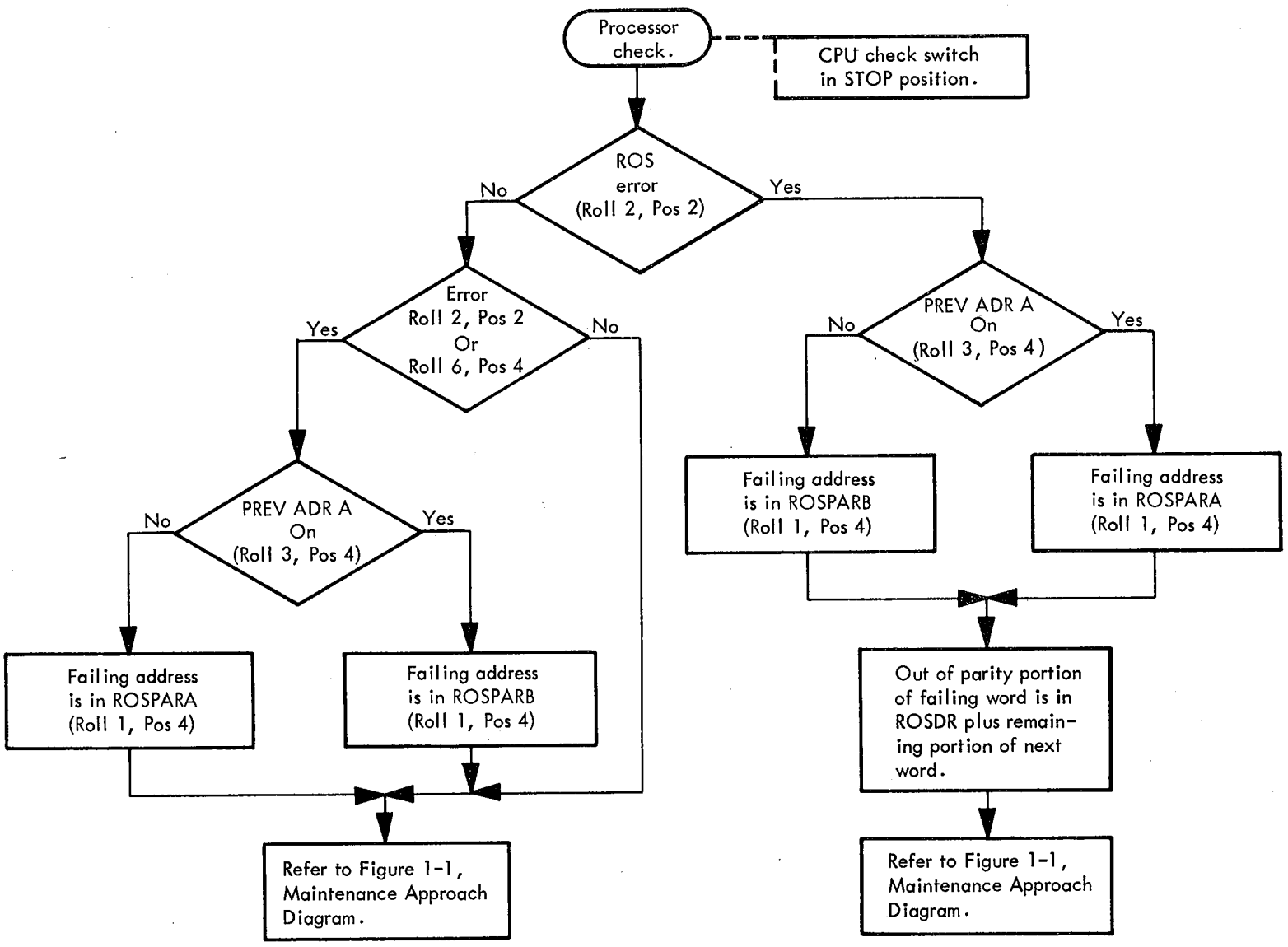
4.15 STOP-LOOP FAILURES

Failures which occur while in the stop loop may be isolated while operating at machine speed using the following procedure:

1. Insert a jumper from 01C-A3H5B10 (RX303) to 01E-E2M4D09 (KW041) to allow stop on ROS address compare.
2. Set ROS ADR COMPARE switches to address of first block of stop loop. (Refer to QY041.)
3. Depress SYSTEM RESET. ROSPARA or ROSPARB (depending on the PREV ADR A indicator) contains address stopped on. Address of next block to be executed is in ROSAR and must agree with next address shown in CLD.
4. If no error occurs, repeat steps 2 and 3 using sequential ROS addresses in stop loop until an error or wrong branch occurs.

4.16 PROCESSOR CHECKS; DETERMINING FAILING ROS CYCLE

When a processor check occurs while operating in check stop mode, the CPU clock stops after execution of the ROS cycle in which the error trigger is set. The cycle in error is indicated differently for ROS parity checks and adder checks. For example, if the previous ROS address is in ROSPARA (per the PREV ADR A trigger), the address of the failing ROS cycle is in ROSPARA for a ROS parity check but in ROSPARB for an adder check. Use Figure 4-15 to determine the ROS address corresponding to the cycle in which the failure occurred.



To Find	PREV ADR A Indicator	Look In
Next ROS Address	N/A	ROSAR
Current ROS Address	ON	ROSPARA
	OFF	ROSPARB
Previous ROS Address	ON	ROSPARB
	OFF	ROSPARA

Figure 4-15. Processor Checks; Determination of Failing ROS Cycle

4.17 CPU TROUBLESHOOTING FLOWCHARTS

Refer to the following flowcharts for troubleshooting specific areas of the CPU:

Figure 4-16 E-Register Parity Checks
Figure 4-17 Multiplier Decode Parity Checks
Figure 4-18 Serial Adder Full-Sum Checks
Figure 4-19 Parallel Adder Half-Sum Checks

4.18 REPETITION OF SELECTED ROS WORD

When the 2065 is not in the stop loop, the ROS TRANSFER pushbutton is not operative. It is often useful to repeat a ROS word when not in the stop loop; e.g., after having single cycled up to a specific ROS address. This would normally be done when the results from the ROS word to be repeated are dependent upon the functions of previous ROS words. The following procedure may be used:

1. Set up address of desired ROS word in ROS ADDRESS COMPARE switches.
2. Install a jumper from 01C-A3J4B12 to ground.
3. Place RATE switch in PROCESS.
4. Depress START. Desired ROS word is now being repeated.

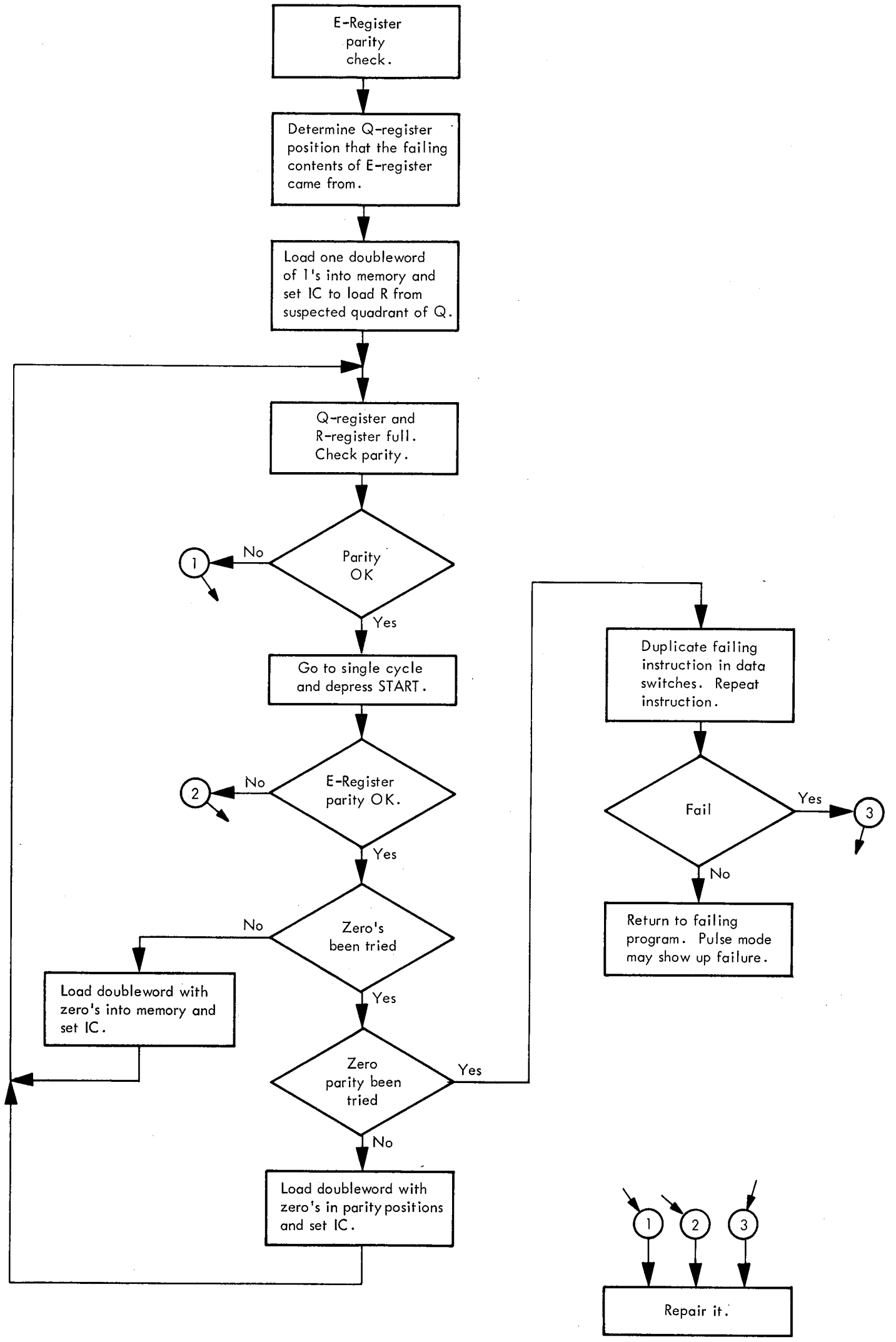


Figure 4-16. E-Register Parity Check Troubleshooting

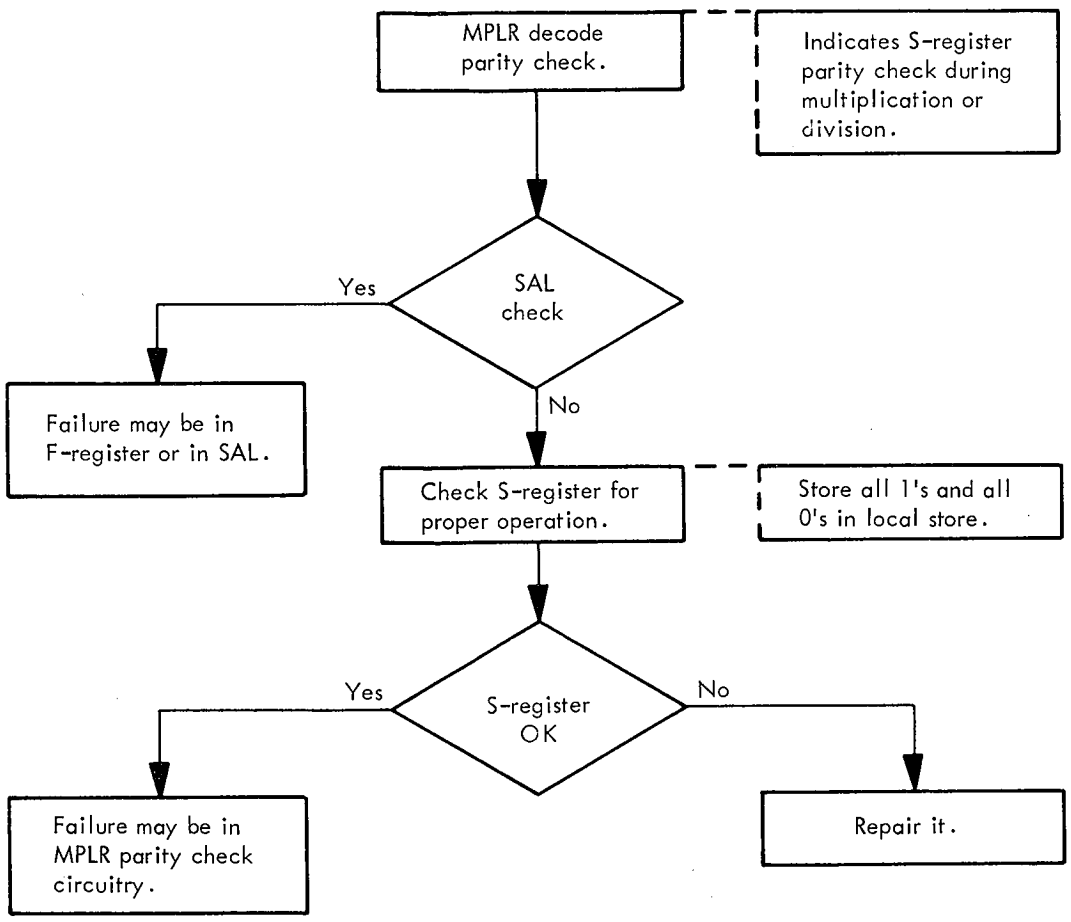


Figure 4-17. MPLR Decode Parity Check Troubleshooting

4.19 STOP ON ROS ADDRESS COMPARE

This procedure stops the 2065 when the address of the ROS word being executed is equal to the address set up in the ROS ADDRESS COMPARE switches:

1. Set up address of last ROS word to be executed in ROS ADDRESS COMPARE switches.
2. Install a jumper between 01C-A3H5B10 (logic page RX303) and 01E-E2M4D09 (logic page KW041).
3. Execute ROS routine.
4. When routine stops:
   a. Address of last ROS word executed is in ROSPARA or ROSPARB, depending on state of PREV ADR A indicator.
   b. Address of next ROS word to be executed is in ROSAR.

4.20 CLEARING WAIT BIT

Three methods of clearing the wait bit in the current PSW are given here:

1. Store all zeroes in storage location zero and depress PSW RESTART.
2. With zeroes in S, repeat ROS address 9B4 (PSW bit 0 to 39).
3. The following method clears the wait bit without executing any instructions:
   a. Depress STOP.
   b. Place RATE switch in INSTRUCTION STEP.
   c. Ensure that bit 14 of word at storage location zero is a zero.
   d. Depress PSW RESTART.
   e. Depress SYSTEM RESET.
   f. Place RATE switch in PROCESS.

4.21 DETERMINING TRIGGER BEING TESTED BY FLT

1. To determine which buffer is being used, observe the BUFFER 1 indicator:
   On = buffer at 8000
   Off = buffer at 8080
2. Observe MASK and MCW (8008 or 8088):
   a. MASK indicates which bit in scanned out word is being tested.
   b. MCW (21–25) indicates which word will scan out to T. (The field is treated as a five-bit binary counter; e.g., 10011 = word 19.) Refer to logic pages M3021–M3061 to determine which triggers will scan out.

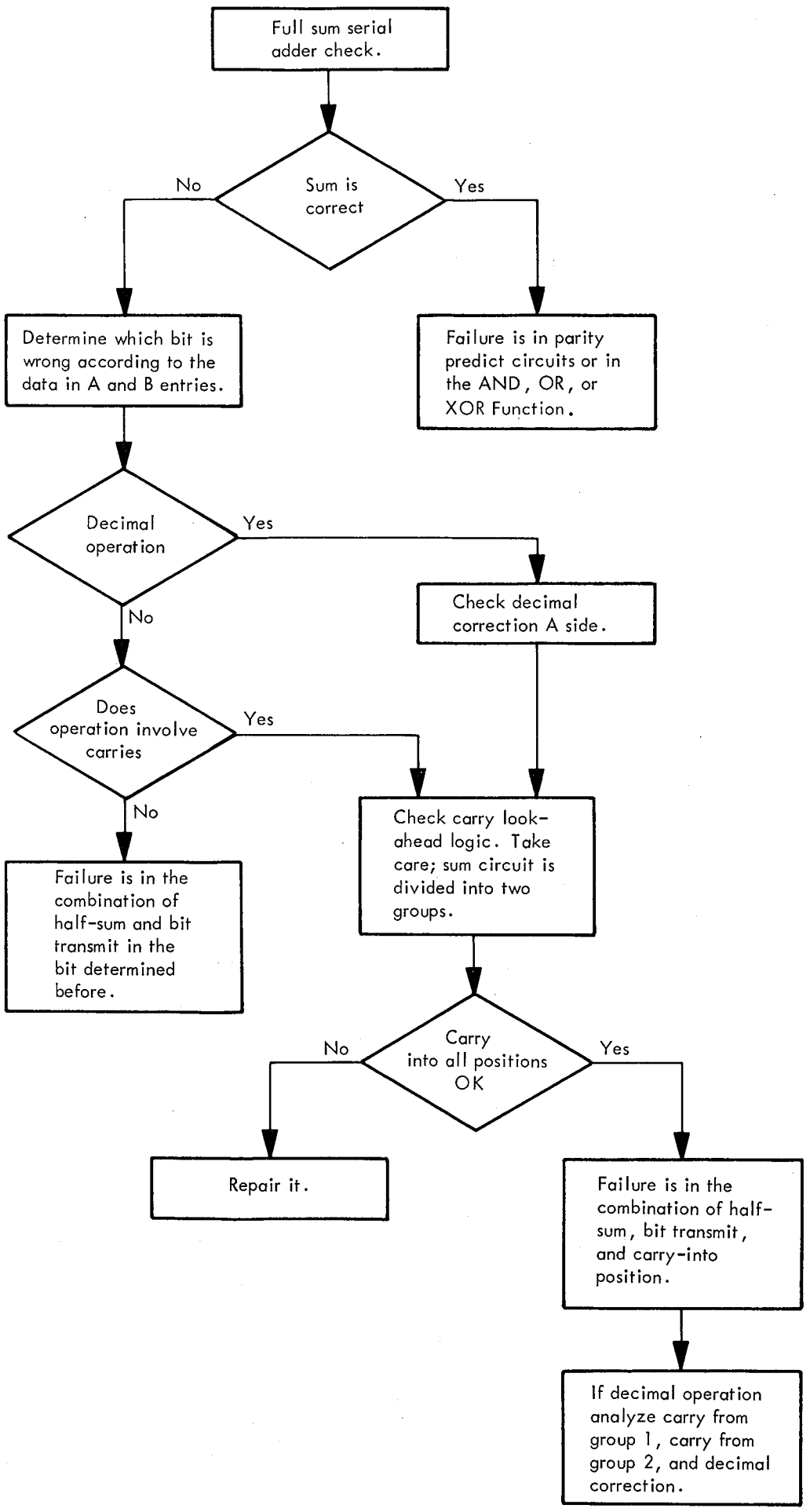
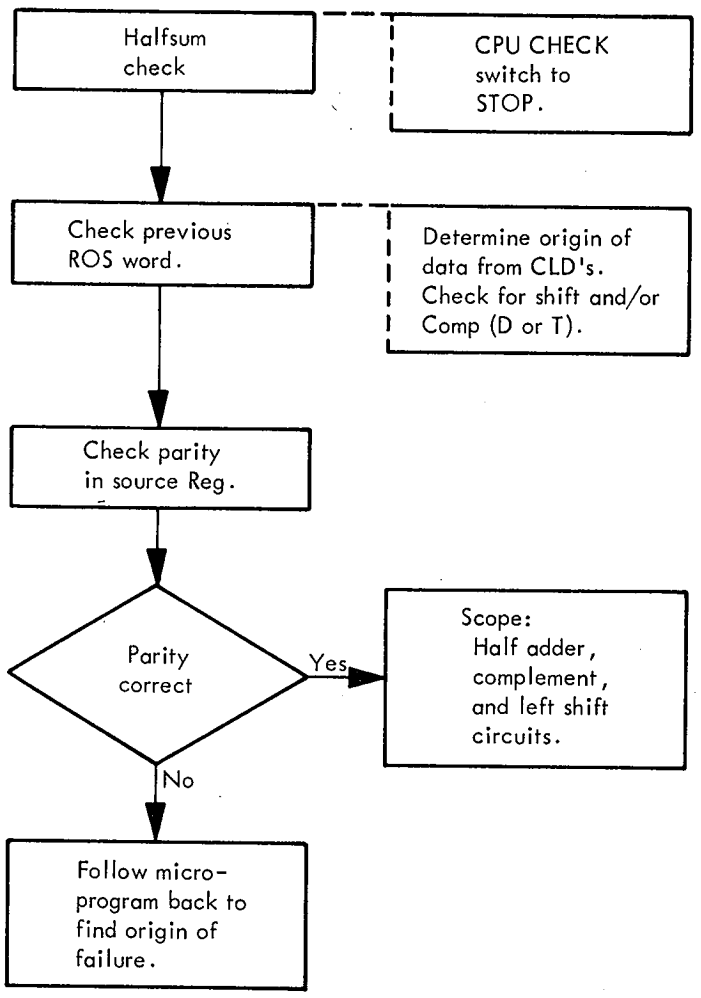


Figure 4-18. Serial Adder Full-Sum Check Troubleshooting



Possible Entries per Gate Control Triggers

Source	Adder Bits	ALD
S(00-31)	to PAA(32-63)	RT815
T(32-63)	to PAA(32-63)	RT807
T(32-63)	to PAA(31-62)	RT811
T(32-47)	to PAA(43-63)	RT813
T(48-63)	to PAA(48-63)	RT815
D(00-23)	to PAA(08-31)	RT803
D(00-23)	to PAA(07-30)	RT805
D(00-23)	to PAA(40-63)	RT801
ONES	to PAA(61-63)	RT813
IC	to PAB(40-63)	RB811
A(00-31)	to PAB(32-63)	RB805
A(04-07)	to PAB(04-07)	RB805
A(08-31)	to PAB(08-31)	RB813
B(32-63)	to PAB(32-63)	RB813
B(64-67)	to PAB(64-67)	RB813
A(06-31)	to PAB(04-29)	RB811
B(32-67)	to PAB(30-65)	RB811
B(64-67)	to PAB(28-31)	RB807
EXCS6	to PAB(28-63)	RB805
F(04-07)	to PAB(60-63)	RB807
HOT-1	to PAB 60	RB807
E(08-11)	to PAB(56-59)	RQ801
E(12-15)	to PAB(60-63)	RQ801
E(08-11)	to PAB(60-63)	RQ801
Q(04-15)	to PAB(52-63)	RQ811
Q(20-31)	to PAB(52-63)	RQ811
Q(36-47)	to PAB(52-63)	RQ811
Q(52-63)	to PAB(52-63)	RQ811

Figure 4-19. Parallel Adder Half-Sum Check Troubleshooting

c. MCW(4), the LETHF trigger, determines which word of doubleword will scan out to T:
   On = bits 0–31
   Off = bits 32–63
d. MCW(7), ERSLT trigger, determines whether scanned out trigger should be on or off.

3. The test may be restarted, after examining buffers, as follows:
   a. TEST MODE switch to FLT and REPEAT.
   b. Set up ROS address 6B0 in ROS address compare switches.
   c. Depress ROS TRANSFER.

4.22 REPETITIVE CONSOLE PUSHBUTTON OPERATION

The operation of some console pushbuttons can be made repetitive by installing a jumper from 01C-B4G3D09 (logic page DS122) to ground and holding the pushbutton depressed. The following pushbuttons are affected: STORE, DISPLAY, SET IC, START, ROS TRANSFER, and PSW RESTART.

A 16-ms pulsed operation can be obtained from these pushbuttons by adding a jumper from 01A-C2D6D04 (logic page KD601) to 01E-E2F7B03 (logic page KW011) and holding the pushbutton depressed. The last technique also applies to the LOGOUT, SYSTEM RESET, and IPL pushbuttons and is especially useful for troubleshooting an IPL hang condition.

4.23 VISUAL DISPLAY OF LOG DATA INDICATORS

During a logout, 'Enable Scan By-Pass' allows all indicators to come on. To troubleshoot a logout problem in single cycle, or a machine check hard stop during logout, it is necessary to temporarily degate the extraneous indicators. This is done by installing the following jumpers:

01E-B4K4B02 to ground
01E-B4K4D11 to ground
} KT761

These jumpers will cause data to be logged incorrectly into storage. They should be left in place only long enough to observe the indicators.

4.24 I/O SCOPING LOOP

The four-instruction loop shown in Figure 4-20 can be keyed into storage and is helpful in troubleshooting I/O problems. It provides a repetitive SIO.

If more than one operation is desired, specify chaining in the CCW's. If the problem is a control unit hang condition, it is helpful to store the address of the first instruction at location zero and use PSW RESTART. In the example shown, this address is hex 200.

Storage Location (Hex)	Assembly Language		Machine Language	Comments
	OP	Operands		
48	DC	X'00001000'	00001000	CAW
200	TIO	XXX	9D000XXX	TIO to device XXX
204	BC	7,200	47700200	Loop on TIO till available.
208	SIO	XXX	9C000XXX	SIO to device XXX
20C	B	200	47F00200	Unconditional branch
1000	CCW	----------	----------	CE may put any desired CCW's at this location.

Figure 4-20. I/O Scoping Loop

4.25 SERVICE TECHNIQUES USING OSCILLOSCOPE

The Tektronix* 453 oscilloscope can be used in combination with the CE latch card (PN 5801358) as an input to SLT logic. This is done by using the +GATE outputs from the scope as inputs to the high impedance legs of the latch card. There are two +GATE outputs, one for each trace (A and B). These gates are active during their associated sweep and produce 0 to 12V signals. The –Sync output of the CE latch card can be dotted with any normal SLT logic circuit (except long line emitter followers), permitting the 453 gate outputs to control SLT logic. Refer to Tool and Test Equipment CEM 108 for details of connection to the CE latch card.

Two typical examples of the use of this technique follow:

Example 1. To impulse an SLT line with a pulse of controlled frequency and duration. This can be used for simulating pushbuttons and resets, stepping counters, and triggering singleshots.

1. Wire output of +B Gate to +high impedance input.
2. Ground reset line of CE latch card.
3. Wire –Sync output of latch card to SLT line to be activated. Use as short a wire as possible for this connection.
4. Allow scope to auto-trace.
5. Place scope in A Intensified By B mode.

The frequency of the pulse will be determined by the A-sweep speed. The duration will be determined by B-sweep speed.

Example 2. An output of a ROS field decode is found to be intermittently coming up when it logically should not. However, the same decode line occurs legitimately three other times while cycling the failing loop. The problem is to find which ROS address is producing the extra decode.

1. Allow original sync to continue to trigger A-sweep.
2. Place scope in A Intensified By B mode.
3. Position bright spot on scope so it overlaps only the extra decode.
4. Wire output of B Gate to +high impedance input of CE latch card.
5. Wire decode line in question to another leg of AND circuit.
6. Wire –Sync output to some point that will freeze the machine. The machine will stop when the extra decode occurs, and the ROS address backup register will indicate the ROS word that developed the extra decode.

Another service technique is to use the Add Algebraic feature as an AND function for internal sync.

4.26 TURNING ON CHECK STOP AFTER HARD CORE (601F) RUNS

It is sometimes desirable to turn on check stop switches after 601F runs and before DME is loaded. The following procedure makes this possible:

1. Set up address 7090 (hex) in MAIN STOR ADDRESS COMPARE switches.
2. Turn on ADDRESS COMPARE STOP.
3. Load 601F; then depress START five times. After fifth depression, 9C 00 20 00 will appear in first four bytes of Q. (This is the SIO which brings in DME.) 9C 00 will appear in R and 41 80 will appear in E.
4. Turn off ADDRESS COMPARE STOP, turn on desired check stop switches, and depress START.

The simulated IPL will be executed and DME will be loaded.

*Trademark of Tektronix, Incorporated.

4.27 HINTS FOR RUNNING SIP

1. Normal section sense switch settings assume operation on DASDI formatted file devices. If unformatted files are being tested, set section sense switch 1 prior to loading SIP. The input message is I/LFB0/S.1/B.
2. SIP initially skips over tape labels using four forward space file operations. This normally presents no problem on unlabeled scratch tapes. However, new tapes (completely erased) or tapes with a single long record will run away. These tapes should be initialized with four short records or four tape marks.
3. Tape errors of unit check with data check set in sense byte 0, bit 4, are normally not printed. These errors will appear in the error summary at Glidedown or Termination. All tape errors may be printed by setting SIP Data Switch 4 (FB03 and above).
4. SIP I/O tasks normally depend on an incorrect ending status for error indication. An option to actually perform a program check of data is available by setting data switch 9.

4.28 ANALYZING AN UNEXPECTED WAIT STATE CONDITION

When the CPU goes to wait state, one or more of the following indications are usually present to explain the wait:

1. Peripheral equipment still operating.
2. Error code in current PSW.
3. Coded message in main storage location 000-003.
4. System operator message.
5. Valid operator intervention required.

If the wait is unexpected and unexplained, the following approach should be taken to find the cause:

1. Take a manual logout.
2. Take a storage dump.
3. Examine the program old PSW which starts at location 28 (hex). If the cause was a program interrupt, the interrupt code will show the cause. The address of the instruction which caused the interrupt can be found by subtracting the instruction length code (two high-order bits of byte at location 2C) from the instruction address (at location 2D-2F).
4. If the problem was not a program interrupt, examine the I/O and PSW at location 38 (hex). It is not unusual to use an I/O interrupt to switch the CPU to wait state.

4.29 ANALYZING I/O COMMANDS

To determine (1) the last I/O command issued, (2) the device to which it was issued, and (3) the results (status) of its execution, analyze the console or a storage dump as follows:

1. Last I/O Command Issued: This can be determined from the channel address word (CAW) which is at location 48 (hex) in storage. The last three bytes of this word contain the storage address of the channel command word (CCW) to be executed. The CCW is a doubleword which may be on any doubleword boundary in storage. Note, however, that if command-chaining is specified in the CCW, the CAW points only to the first CCW of the chain, not to the last CCW performed. The channel status word (CSW), a doubleword starting at storage location 40 (hex), normally contains the address of the last CCW executed plus eight bytes.
2. Device to which last command was issued: This information is available in the interruption code portion (bits 16–31) of the I/O old PSW at storage location 3A (hex).
3. Status of command execution: Device and channel status is contained in bits 32–47 of the CSW; i.e., the halfword at storage location 44 (hex). The byte count is available in the halfword at location 46 (hex). This, too, is an indication of the status of the command execution. The byte count should be zero unless one of the following three things occurred:
   a. A wrong-length record was encountered. If the SILI flag is set in the CCW, the wrong length was expected.
   b. The command was not actually executed for some reason; e.g., a command reject may have occurred.
   c. A check condition occurred during a read or write operation, causing data transfer to stop at the point where the error occurred. Device motion would have stopped at the end of the affected record. Channel end, device end, unit check, and incorrect length are posted in the CSW, and the residual byte count probably indicates the amount of data not stored.

For additional information, review the formats of the CAW, CCW, and CSW in *IBM System/360 Principles of Operation*, Form A22-6821.

4.30 ANALYZING IMPRECISE INTERRUPTS

An imprecise interrupt can be defined as one which does not provide all the information pertinent to the interrupt. This is normal under certain conditions. However, when imprecise program interrupts occur, it is sometimes difficult to find the instruction which caused the interrupt. The following is an example.

Given the following coding in which a store into a protected area is followed by an unconditional branch:

```
loc     coding
100     ST1,200(2)
104     BCF,300
```

The program old PSW will contain:

System mask	N/A
Key	Greater than zero
AMWP	N/A
Interruption code	00 04
ILC	00
CC	N/A
Prog Mask	N/A
Instruction address	300

In this example the instruction length code (ILC) is unknown, and the instruction address is not pointing to the instruction following the one which caused the interrupt. There is no way to trace the source of the problem from this information.

The following service technique may be used:

Float pin 01E-C3F4D04 (logic page KM831). This will cause a storage address protect (SAP) delay (ROS address 02E) every time a store is executed.

CAUTION
Do not leave this condition on the CPU as it will cause jobs to run slower than normal.

4.31 SINGLE ADDRESS BCU SCOPING LOOP

To send repeated requests to storage for a single address:

1. Place RATE switch in SINGLE CYCLE.
2. Set address 9CF into ROS ADDRESS COMPARE switches. (ROS loop is shown on page QY021).
3. Depress ROS TRANSFER.
4. Change ADDRESS switches to desired storage address. The method used to determine store and fetch is the same as that used in ripple store/ripple fetch where:
   D(21,22) = 3 results in Store
   D(21,22) ≠ 3 results in Fetch
5. Place RATE switch in PROCESS.
6. Depress START.

Note: If the address keys are changed while the machine is in the loop, parallel adder half-sum checks will occur. Depress CHECK RESET to eliminate error indications.

Several differences exist in the power section of the CPU between the converted units (i.e., a 2060 converted to a 2065) and the original units (i.e., a 2065 as originally constructed). The converted units may be identified by the relay gate hinges on the right edge of the gate; the original units, by the hinge at the bottom edge of the gate. The overvoltage assembly is not a part of the original unit power supply. This assembly must be ordered separately.

Before making changes, resistance measurements, or replacements, be sure all power is off and all capacitors are fully discharged; do not rely on bleeder resistors. Never work alone. If in doubt: Don't! Remember that normal power-off does not turn off the 28V ac, the 24V dc sequencing and EPO power, or the 25V dc bias power, nor the 115V ac to the convenience outlets. Set CB1 off (primary input) or, better, turn off the primary wall power switch to remove these voltages.

5.1 VISUAL INSPECTION AND CLEANING

DANGER
Before inspecting or cleaning the 2065 Processing Unit's power system, the primary power cable should be disconnected. Lethal potentials exist within the unit whenever power is on.

The 2065 Processing Unit's power system is inspected for:

1. Loose or damaged wiring.
2. Burned or pitted contacts on relays and contactors.
3. Loose or maladjusted cams on the MC assemblies.
4. Dust accumulation. (Dust should be removed with a vacuum cleaner.)

5.2 POWER CHECKS

Figure 5-1 shows the overall procedure for checking the power portion of the CPU. It is presumed that the primary power is within +10% or –8% of the voltage specified on the voltage plate, the room temperature is between 60°F and 90°F, and the relative humidity is less than 80%.

The power check consist of checking the momentary pick of relays K3 and K35, and the 5-second delay before transfer of K46 and K47 (or, in the converted units, K25 and K26). The check procedure is as follows:

1. Depress POWER OFF.
2. Set CB1 off.
3. Set CPU READY/OFF switch on CE power panel to OFF.
4. While observing K3, set CB1 on. K3 should pick, then drop immediately.
5. While observing K47 (or K25, if converted unit), set CPU READY/OFF switch to READY. K47 (or K25) should transfer 5 seconds later.
6. While observing K46 (or K26, if converted unit), depress CPU ON on CE power panel. K46 (or K26) should transfer 5 seconds later. (Power should be on.)
7. Immediately upon transfer of K46 (or K26), K35 should pick, then drop.

5.3 MARGINAL CHECKING

The marginal checks are performed by running the FLT's and diagnostic programs with all 6V marginable power supplies in the CPU varies within the limits prescribed below. Briefly, the procedure is as follows:

1. Load applicable FLT/ROS or diagnostic program into CPU.
2. Select a frame with the Margin/Meter SEL rotary switch.
3. Vary the +6M voltage with the individual gate ROS RAISE-LOWER potentiometer:
   Bias limits for CPU–5.5V to 6.5V.
   Bias limits for Storage–5.0V to 7.0V.
   Bias limits for Channel–5.0V to 7.0V.
   ROS limits–reduce to 80% of nominal.

CAUTION
The marginal check drive motor assembly should not operate more then 50 seconds at a time: i.e., an off period of at least 20 seconds should follow each 50 seconds of operation.

4. If the FLT/ROS or diagnostic program does not reveal a malfunction while operating under marginal voltages, return the power supply voltages to normal. Then, use the FREQUENCY ALTERATION switch to decrease the CPU clock period from 200 ns to 195 ns, and rerun the tests.

5.4 VOLTAGE CHECKS AND ADJUSTMENTS

Apply power, and measure the dc voltages at the locations specified in Table 5-1. Use a 5% precision meter. Do not measure the voltages at the power supply regulators. (These

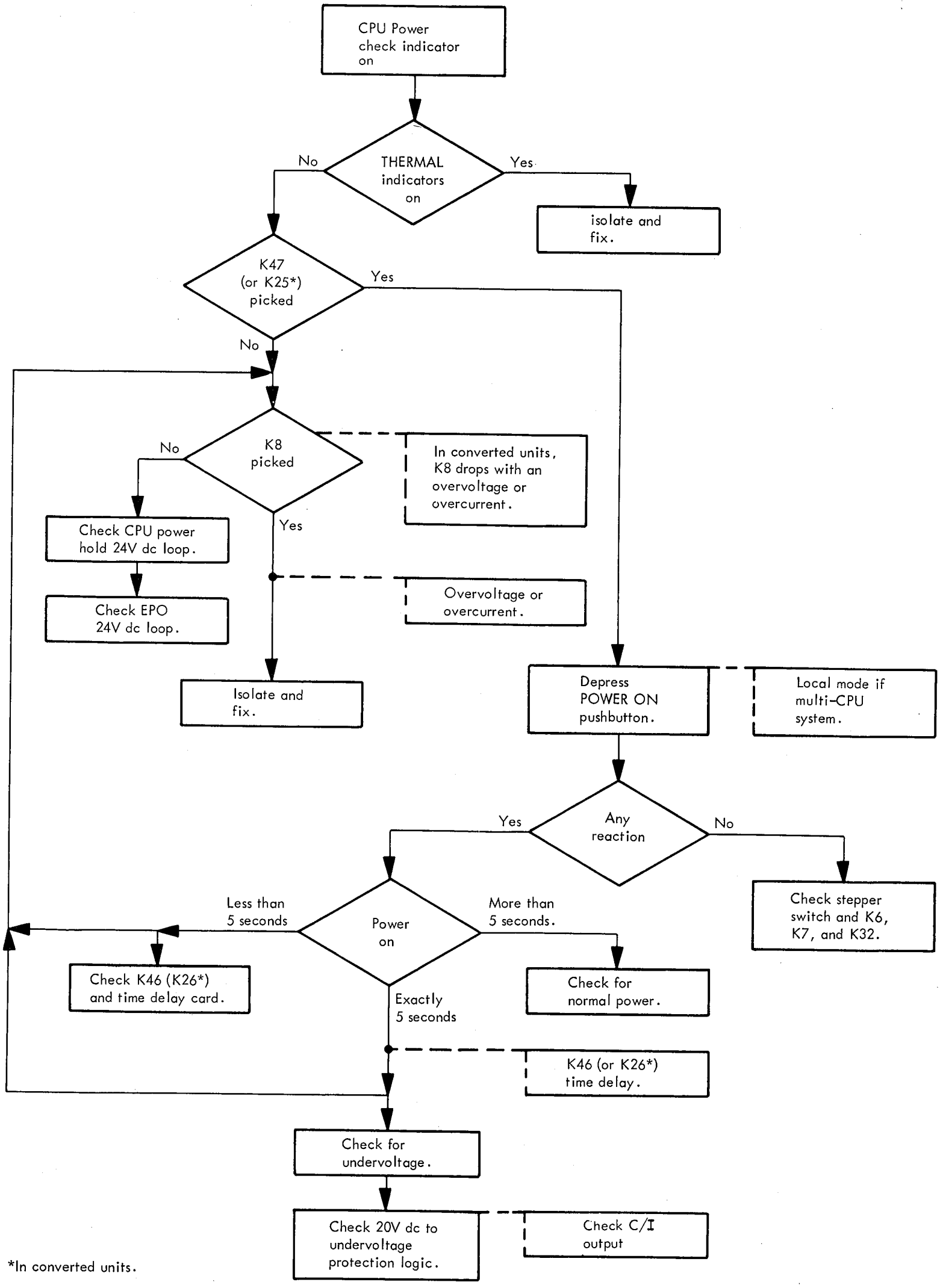


Figure 5-1. CPU Power Checkout

Table 5-1. DC Distribution

Regulator	Nominal Output	Logic Page	Laminar Bus			Voltage Measurement Location
			Gate	Pin		
				Voltage	Return	
PS1	+ 6V 25 amp	YA061	C/D upper	12	11	01C-E2G7B11
PS2	+ 3V 40 amp	YA062	B lower	8	7	01B-D3G2D03
PS3	+ 3V 40 amp	YA061	E lower	8	7	01E-D3G2D03
PS4	-- 3V 40 amp	YA062	C/D upper	10	9	01C-E2G7B06
			C/D lower	10	9	01C-D3G2B06
PS5	+ 3V 40 amp	YA062	B upper	6	5	Use gate E pin
			E upper	8	7	01E-B2G2D03
PS6	+ 3V 40 amp	YA062	C/D upper	4	3	01C-E2G7D03
			C/D lower	8	7	01C-D3G2D03
PS7 *	+ 6V 40 amp	YA061	E upper	12	11	01E-D2G6B11
			E lower	12	11	01E-B4G6B11**
PS8 *	+ 6V 40 amp	YA062	C/D lower	12	11	01C-D3G2B11 **
PS9 †	+48V 2 amp	YA141	11 BB 9	1	11-C2a	11 BB 9 - 1
PS10	-- 3V 40 amp	YA072	A upper	10	9	01A-C1G2B06
			A lower	10	9	01A-D3G2B06
			B upper	10	9	01B-B1G2B06
			B lower	10	9	01B-B3G2B06
			E upper	10	9	01E-B1G2B06
			E upper	10	9	01E-D3G2B06
PS11 *	--18V 11 amp	YA071	C/D upper	6	5	Gate C/D upper - 6 **
PS12	+ 3V 40 amp	YA072	A lower	12	11	01A-C4G6D03
			Indicators	P13	--	--
PS13	+ 3V 40 amp	YA071	B upper	8	7	01B-B1G2D03
PS14	+ 3V 40 amp	YA072	A upper	12	11	01A-C1G2D03
			Indicators	P13	--	--
			Meter	P13	--	--
PS15 *	+ 6V 40 amp	YA071	B upper	12	11	01B-A1G2B11
			B lower	12	11	01B-B4G6B11 **
PS16 *	+ 6V 40 amp	YA072	A upper	8	7	01A-C1G2B11 **
			A lower	8	7	01A-C4G6B11

* Marginable regulator.
** Use this point for voltage-limit measurement.
† PS9 may not be installed.

measurements will be greater than at the gates.) If any voltage is out of tolerance, readjust as described in 5.4.1 through 5.4.3 below. (These adjustments are applicable to all power supplies except ROS supply. For voltage adjustments on the ROS supply, refer to heading 4.9.)

After all voltages have been adjusted, the system control panel voltmeter may be calibrated. Measure all marginal power supply voltages with the system control panel voltmeter, and record in Figure B-1 of Appendix B. This information is for future reference in interpreting the system control panel voltmeter readings and should be kept with the CPU.

5.4.1 Regulator Output Voltage Adjustment

Set the dc voltages by adjusting the potentiometer on the Amplifier Assembly card or on the associated panel control of the marginable regulators. All regulators with a single logic pin location listed in Table 5-1 are measured at that location. All regulators with more than one logic pin location listed in Table 5-1 are measured and adjusted according to the following formula:

$$S = \left(\frac{H-L}{2}\right) + N$$

where: N = listed nominal voltage
H = highest measured voltage
L = lowest measured voltage
S = voltage to be set at the logic pin with the highest measured voltage

The adjustment procedure for regulators with multiple pin locations may be implemented (in Table 5-1) as follows:

1. Measure voltage at each logic pin location.
2. Subtract lowest voltage reading (L) from highest (H).
3. Divide result of step 2 by 2.
4. Add result of step 3 to nominal voltage listed (N).
5. Measure voltage at logic pin with highest voltage reading (H).
6. Adjust potentiometer on Amplifier Assembly card (of nonmarginable regulators) or on associated panel control (of marginable regulators) for voltage reading (step 5) equal to result of step 4 (S).

5.4.2 Regulator Overvoltage Trip Adjustment

5.4.2.1 Converted Units

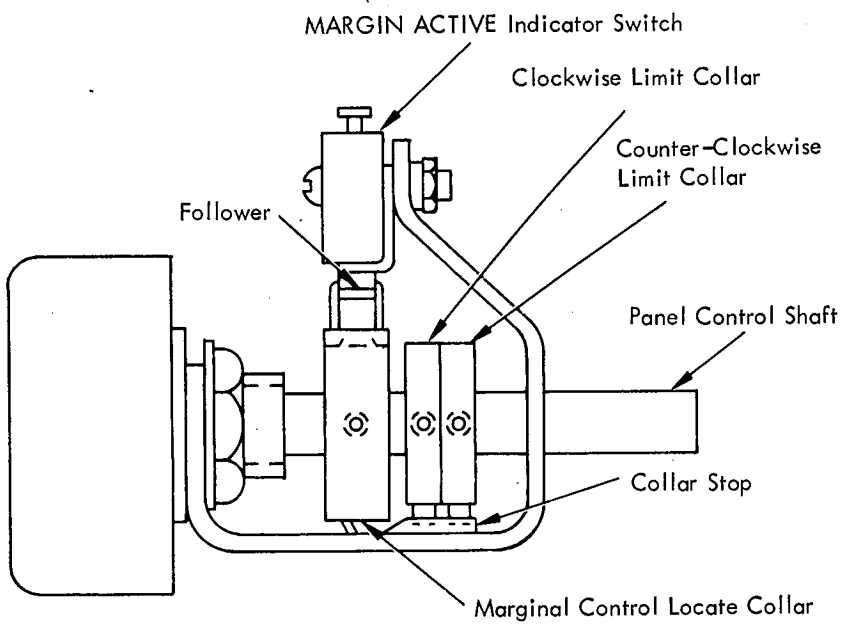
The adjustment procedure for the external overvoltage sensing circuit is described in Systems Power logic page YA121.

5.4.2.2 Original Units

Overvoltage sensing internal to each regulator is executed by the Overvoltage/Overcurrent card.

The marginal power supplies have an adjustable "ax" assembly to permit an accurate overvoltage trip point. Adjust as follows:

1. Loosen setscrew of clockwise limit collar on associated potentiometer (Figure 5-2, A).
2. Using voltmeter (0.5% or better) measure voltage at appropriate logic pin as listed in Figure 5-2, B.
3. Rotate panel control shaft until voltage listed in Figure 5-2, B under "ax" is obtained.
4. Using a small screwdriver, very slowly decrease potentiometer setting of "ax" assembly on the regulator until power drops.
5. Turn panel control to a lower setting and recycle power on.
6. Rotate panel control until voltage listed in Figure 5-2, B under "Limit Stop, Upper" is obtained.
7. Rotate clockwise limit collar clockwise until pin protruding from collar rests on right side of collar stop (as viewed from shaft end). Tighten setscrew on limit collar.
8. Proceed with undervoltage limit collar adjustment described next.



(A)

Panel Control	Regulator	Logic Pin	Voltage Setting		
			"ax"	Limit Stop	
				Upper	Lower
+6M A GT	PS16	01A-C1G2B11	6.9V	6.5V - 6.7V	5.3V - 5.5V
+6M B GT	PS15	01B-B4G6B11	6.9V	6.5V - 6.7V	5.3V - 5.5V
+6M C GT	PS8	01C-D3G2B11	6.9V	6.5V - 6.7V	5.3V - 5.5V
+6M E GT	PS7	01E-B4G6B11	6.9V	6.5V - 6.7V	5.3V - 5.5V
ROS	PS11	*	22.0V	20.0V - 20.3V	6.8V - 7.0V

*Use pin 6 on gate C/D upper laminar bus.

(B)

Figure 5-2. Marginal Check Potentiometer Adjustment Limits

5.4.3 Regulator Undervoltage Trip Adjustment

Undervoltage sensing is external to the regulators and is performed by a special sense circuit. Adjust undervoltage limit collar for marginal power supplies as follows:

1. Loosen setscrew of counterclockwise limit collar on associated panel control potentiometer (Figure 5-2, A).
2. Using voltmeter of 0.5% or better accuracy, measure voltage at appropriate logic pin listed in Figure 5-2, B.
3. Rotate panel control until voltage listed in Figure 5-2, B under "Limit Stop, Lower" is obtained.
4. Rotate counterclockwise limit collar counterclockwise until pin protruding from collar rests on left side of collar stop (as viewed from shaft end). Tighten setscrew on limit collar.

Proceed with MARGIN ACTIVE indicator switch adjustment as follows:

1. Loosen setscrew of margin control locate collar on associated panel control potentiometer (Figure 5-2, A).
2. Measure voltage as described in heading 5.4.
3. Rotate panel control potentiometer until voltage reading derived from procedure in 5.4.1 is obtained at specified logic pin. The setting for ROS panel control (PS11) is determined by ROS optimization procedure (see heading 4.9).
4. Rotate margin control locate collar until MARGIN ACTIVE indicator goes out. (The follower in the notch on the collar and the switch open.) Tighten setscrew on collar.

5.5 TROUBLESHOOTING

The troubleshooting procedures are described in three parts: Power Supply Protection Circuits, Converter/Inverter, and Regulators.

5.5.1 Power Supply Protection Circuits

The following paragraphs describe the power supply protection circuits used in the CPU: circuit interlocks, voltage sensing circuits, overcurrent sensing circuits, and thermal sensing circuits.

5.5.1.1 Circuit Interlocks

The circuit interlocks provide a power-off sequencing of the entire system if any thermal switch, regulator overcurrent circuit, or circuit breaker (except those for convenience outlets) within frames 01, 02, and 04 is opened. A power-on after any of these conditions is possible only after the affected circuit is reset and the fault corrected. The power-on sequences are shown in Diagrams 6-3, and 6-4 of FEMDM. The power-off sequence is shown in Diagram 6-8 of FEMDM.

If the sequencing is interrupted by a relay fault, check the contacts of the relay involved for dirt or oxide. Clean and, if necessary, burnish the contacts, and check whether the coil has opened or shorted. Also check the 5-second delay before transfer of K46 and K47 (or, in the converted units, K26 and K25).

Inspect stepper switch contacts for dirt or oxide, and burnish the contacts if necessary.

Inspect momentary pick relays K3 and K35. If they are not picking, check the series capacitor for an open or for not being discharged (logic YA021). If they are picked continuously, check for shorted series capacitor.

Be sure the gate is free of metal filings or solder bridges.

5.5.1.2 Voltage Sensing Circuits

The voltage sense relays (logic page YA111) are energized when all dc voltage outputs from the regulators are on. If one or more of the dc regulators is not on, the stepping switch will not reach the sequence complete position.

Overvoltage Protection. The remote sensing locations for overvoltage conditions differ between the converted units and the original units. In the original units, sensing is performed on the outside of the regulator; in the converted units, sensing is performed at the bus bars.

In converted units, the trip points of the overvoltage protection devices should be set as follows:

+6V supply	+8.0V trip
+3V supply	+4.5V trip
–3V supply	–4.5V trip

(For adjustment procedure refer to system power logic page YA121.)

In original units, overvoltage sensing internal to each regulator is executed by the Overvoltage/Overcurrent card. Refer to heading 5.4.2.2.

Undervoltage Protection. An undervoltage condition causes power to drop approximately 5 seconds after the CPU ON pushbutton is depressed or the stepper switch reaches position 1. Either K10 (positive regulators) or K11 (negative regulators) did not pick before the 5-second delay preceding transfer of K46 (or, in the converted units, K26) was completed. Use the UNDER VOLTAGE CHECK switches on the CE power panel to locate the failing regulator. Each switch isolates the indicated regulator from the undervoltage sense circuit. Switch off all UNDER VOLTAGE CHECK switches and cycle power on. Switch in one regulator at a time until power drops, indicating the faulty unit. Reset the switch to off and recycle power on. Measure the output voltage of the regulator to ensure that the regulator is not at fault. If the regulator is at fault, refer to heading 5.5.3 for repair procedures.

5.5.1.3 Overcurrent Sensing Currents

An overcurrent condition in three places in the CPU will drop power:

1. Primary Box – primary power fault.
2. Converter/Inverter – excessive regulator loading.
3. Regulator – excessive gate loading.

Inspect K8; an overcurrent (or overvoltage in the regulators) condition will pick it (except in the converted units, where K8 will drop on the overcurrent condition). K8 picks or drops on an overcurrent condition in any regulator that supplies CPU power (except PS9) or if a circuit breaker trips.

Primary Box. Any fault that trips a circuit breaker or blows a fuse may be isolated by removing the load entirely and then reloading, little by little, until the fault reappears. Remember that the convenience outlet power and all voltages below 40V remain on with normal power off. Set CB1 off or turn off the wall primary power switch to remove these voltages.

Converter/Inverter. Excessive regulator loading may be found by removing, with power off, P2, P3, P4, and P5 from the converter/inverter. Reconnect them, one at a time, cycling power each time until the fuses or CB2 in the converter/inverter (C/I-CB2) open immediately upon having power applied. Refer to 5.5.2 for the procedure to further isolate the fault. The only time an overload of this nature should take place is if a short occurs in the cables to, or in an input of, a regulator.

Regulators. Excessive gate loading may be found by inspecting the Overcurrent/Overvoltage card (or, in the converted units, the Overcurrent card) at each regulator. The overloaded unit will have the indicator lamp on. Note, however, that this indicator will also indicate an overvoltage condition. Remove the load and reset the indicator. The indicator is reset by depressing RESET on the converter/inverter with normal power off. Add the load, section by section, until the overload occurs. Continue to isolate in this manner until the fault is located.

CAUTION
Be careful that the nonstandard voltages and voltage combinations do not damage the logic circuitry.

5.5.1.4 Thermal Sensing Circuits

The CE power panel has thermal trip indicators to signify the location of an overtemperature condition. When such a condition exists, check whether there is a proper flow of air through the area, fans are running, filters are clean and not obstructed, and the exhaust area is not blocked. If the air flow is proper, locate and replace the component that is overheating. Check for the cause of the overheating. Also check the accuracy of the sensing elements; one may have changed the point at which it opens. The sensors are designed to open at 140°F.

Reset the indicators by depressing THERMAL RESET on the CE power panel. If the indicators are not reset by the pushbutton or when PK1 and then K3 picks, check K3 and the series capacitor (logic YA021). If any indicator does not reset, check the indicator relay and diode from the reset line (logic YA022).

5.5.2 Converter/Inverter

DANGER
The internal circuitry of the converter/inverter is not isolated from the power source; therefore, a lethal potential to ground is present whenever power is on. Exercise extreme caution. This potential exists on the SMS cards, heatsinks, and terminals to the regulators.

For the safety reasons stated above, maintenance is limited to the following:

1. Fuse and SMS card replacement.
2. SCR gate signal frequency adjustment.
3. Cleaning and checking for loose connections.
4. Air flow checks, including the fan.
5. Replacement of the entire converter/inverter.

Converter/inverter failures almost always show one of the following symptoms:

1. The fuses blow immediately when power is applied.
2. The input circuit breaker (C/I-CB2) trips almost immediately when power is applied.
3. The fuses blow when the load is applied (as C/I-K3 picks).
4. The fuses blow after operating briefly.

The first two symptoms imply an internal failure; the third, excessive regulator loading; and the fourth, a thermal condition. Upon the appearance of any of these symptoms, be sure the power is off and then replace the fuses.

CAUTION
Always replace both fuses at the same time with high-speed, 50-amp, form 101 fuses, PN 5261451.

With power still off, check the SCR gate signals and verify that the fan is running, the air input is not obstructed, and the exhaust area is not blocked.

The SCR gate signals are generated with the 28V ac bias power source and are observed with normal power off. Remove the cover over the SMS cards. Set the oscilloscope to add the inputs algebraically and invert input B. (Input B becomes the reference.) Observe the waveshapes shown in Figure 5-3 at the test points noted. The card in J9-2 drives SCR1, SCR5, and SCR6. The card in J9-3 drives SCR2, SCR3, and SCR4. If the frequency is not 2500 Hz (400

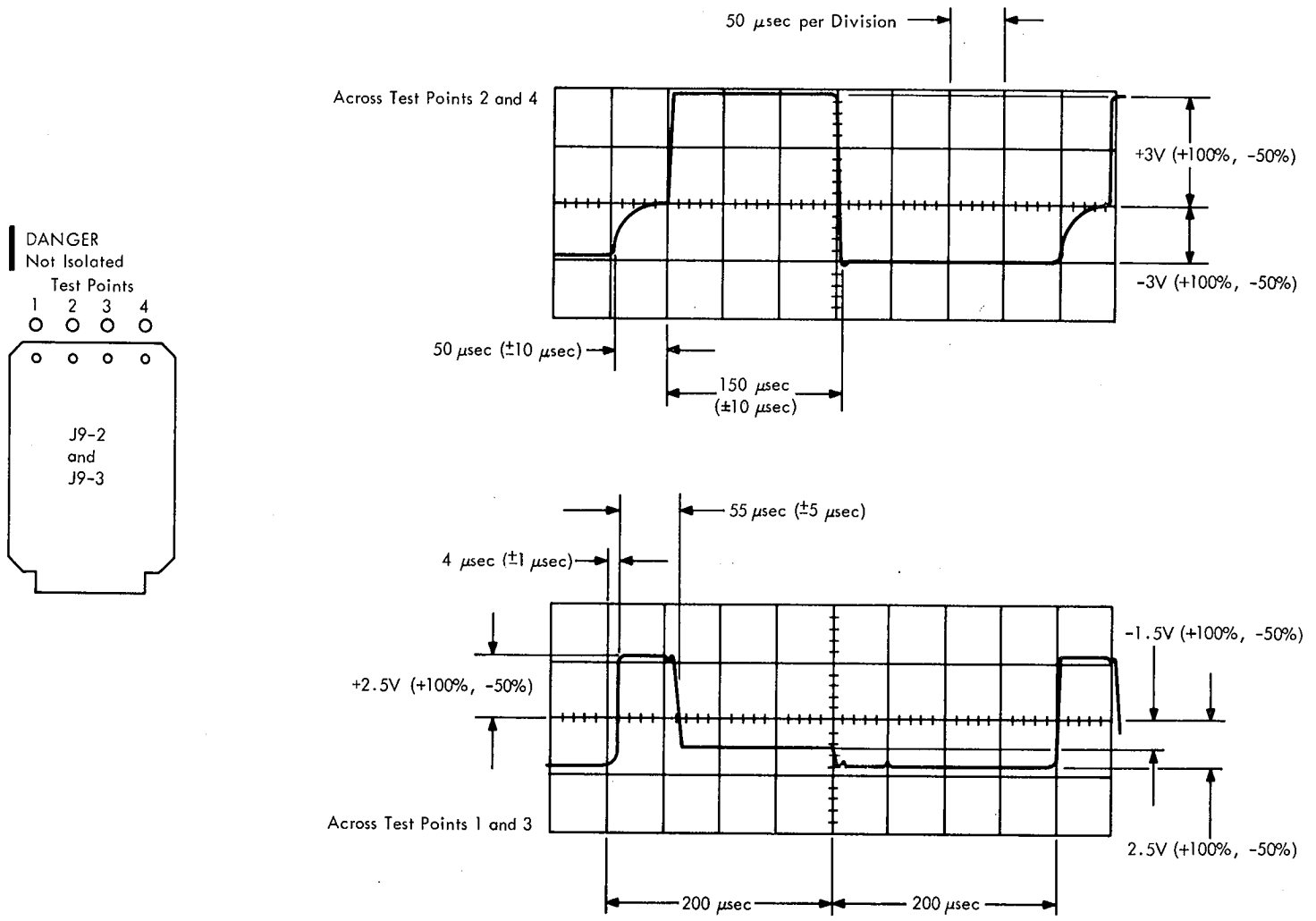


Figure 5-3. SCR Gate Signals (Converter/Inverter)

usec, leading edge to leading edge), adjust the potentiometer on the Magnetic Oscillator card in J9-1. Replace the cards if these tolerances are not met.

If the SCR gate signals are normal and the symptoms still exist, remove the entire regulator load (P2, P3, P4, and P5) and then cycle power on.

If the symptoms still exist, replace the entire converter/inverter (heading 5.6.1). If symptoms do not exist, replace the regulator connectors, one at a time, cycling power each time, until the symptoms reappear. See Figure 5-4 to determine which regulators are powered by the connectors.

Once the connector has been identified, disconnect the regulators supplied by it. Be sure all power is off. Reconnect the regulators, one at a time, cycling power each time, until the faulty regulator is isolated. Refer to heading 5.5.3 to isolate regulator malfunctions.

If trouble cannot be traced to a regulator malfunction, measure the 25V dc bias voltage [C/I-TB1-4 (positive) and -5 (negative)]. If the bias is off, check the 28V ac input, the Rectifier cards in J9-4 and J9-6, and the Contactor Driver card in J9-8. The GATES ON indicator should be on. If the bias is on, with the correct polarity, inspect the contactors for dirt or oxide; burnish the contacts, if necessary. [C/I-K2 picks when C1–C8 have charged to a total potential of approximately 150V dc. C/I-K1 and -K3 pick as C/I-K2 transfers. C1–C8 surge protection resistors (R12, R13, and R14) are shorted by C/I-K1 contacts. C/I-K3 contacts apply the load to the C/I.]

5.5.3 Regulators

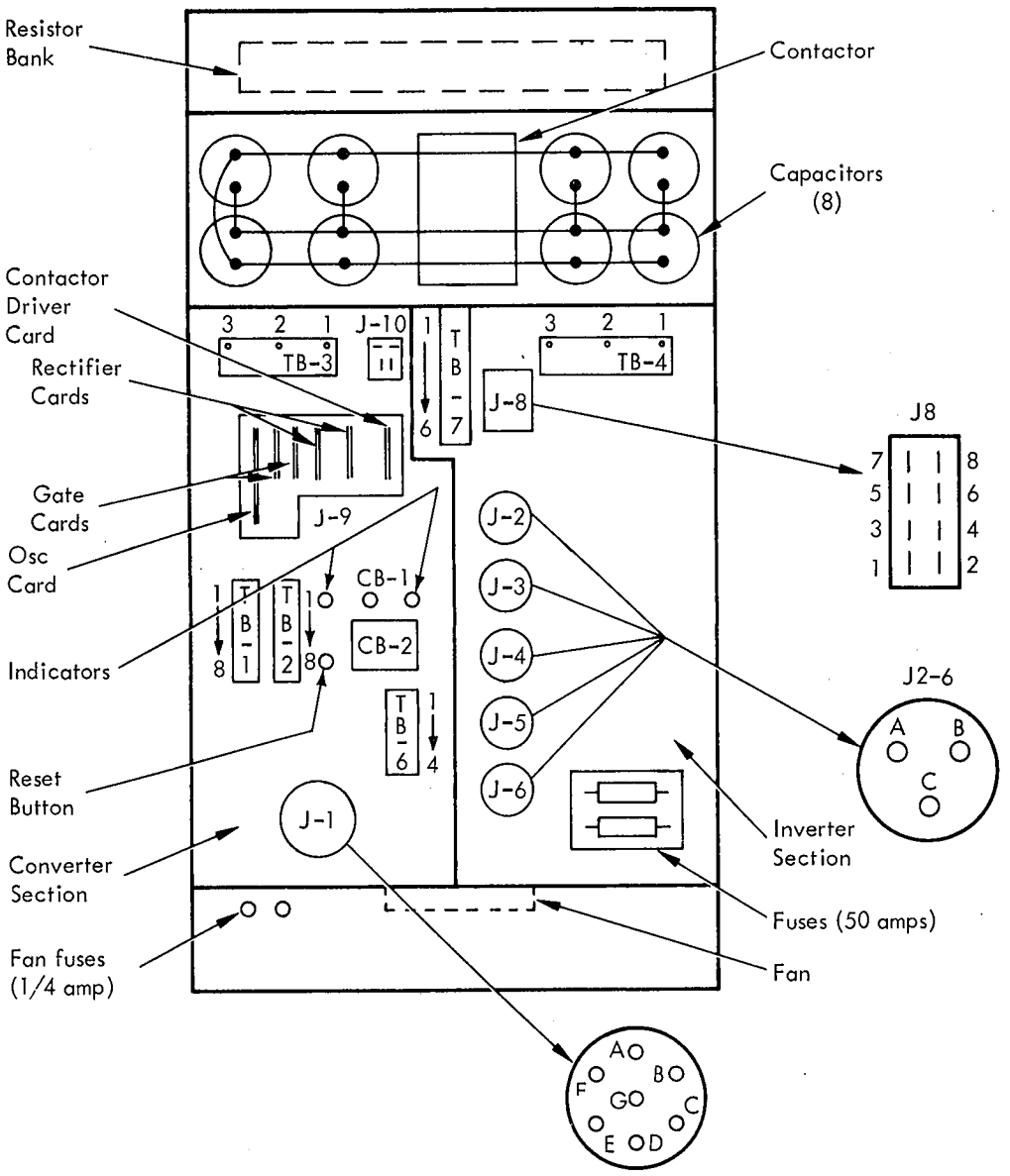
DANGER
The input terminals of the regulators are not isolated from the power source; therefore, a lethal potential to ground is present whenever power is on. Exercise extreme caution.

The power supply regulators are floating supplies and must not be grounded to the frame. The regulators are isolated by two captive mounting screws in nylon standoffs on each regulator.

If possible, limit maintenance to the following:

1. SMS card replacement.
2. Output voltage adjustment.
3. Cleaning and checking for loose connections.
4. Air flow checks.
5. Resistance measurements.

Be sure that all power is off and all capacitors are fully discharged; do not rely on the bleeder resistors. Two types



DANGER

Use extreme caution when working on the converter-inverter. High voltages are present throughout the converter-inverter tub assembly, including the heat sinks that are floating and the SMS cards.

Connector Loading

Connector	Power Supply Load
P2	PS1 PS3 PS5 PS7
P3	PS2 PS4 PS6 PS8
P4	PS9 PS11 PS13 PS15
P5	PS10 PS12 PS14 PS16

Figure 5-4. Converter/Inverter

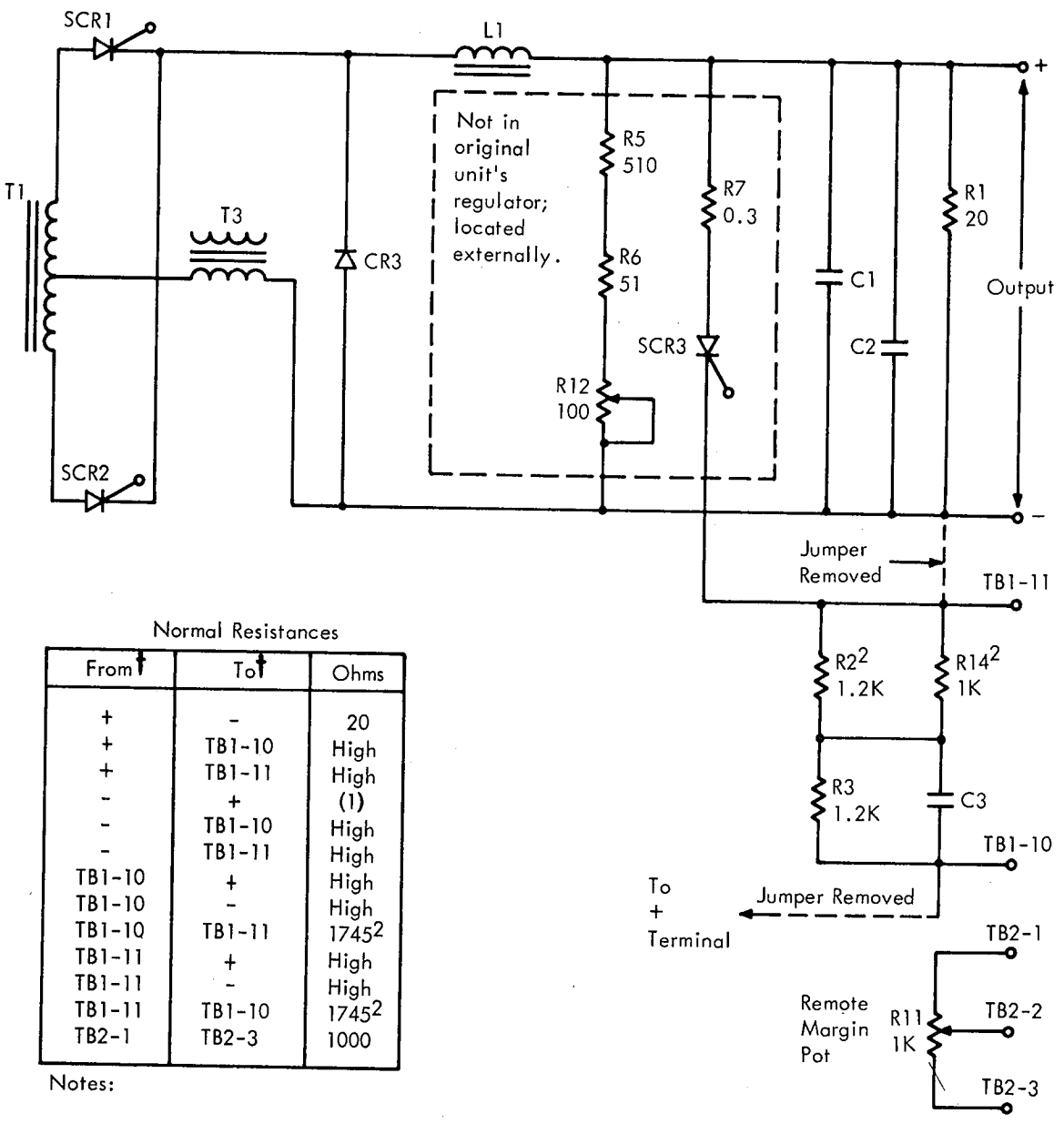
of regulators are used in the 2065 CPU. One is the SCR-controlled output and the other is the magnetic-amplifier-controlled output. The 18V dc ROS regulator (PS11) is of the first type; the others are of the second type.

5.5.3.1 SCR Regulator

A regulator malfunction may be due to an undervoltage, overvoltage, or overcurrent condition. If an undervoltage condition exists within the regulator, proceed as follows:

1. Turn off power.
2. Replace Magnetic Amplifier card (socket A), cycle power, and check for normal output.
3. Replace Differential Amplifier card (socket B), cycle power, and check for normal output.
4. Turn off power and set CB1 off.
5. Remove the three SMS cards.
6. Remove all cables at output positive and negative terminals, including jumpers to TB1.
7. Make resistance measurements between points shown in Figure 5-5. Be sure capacitors are fully discharged by measuring voltages at these points first. Use R x 1 scale for resistance measurements.

Note: Remember that a good diode has high resistance in one direction only, but a good SCR has high resistance in both directions.



Normal Resistances

From†	To†	Ohms
+	-	20
+	TB1-10	High
+	TB1-11	High
-	+	(1)
-	TB1-10	High
-	TB1-11	High
TB1-10	+	High
TB1-10	-	High
TB1-10	TB1-11	1745[2]
TB1-11	+	High
TB1-11	-	High
TB1-11	TB1-10	1745[2]
TB2-1	TB2-3	1000

Notes:

(1) Typically 15-ohms through one diode and high if through two or more series diodes (since the meter voltage is not sufficient to break down the combination).

(2) R14 is not used in the converted unit's regulator, and R2 is 560 ohms, giving a resistance of 1760 ohms between TB1-10 and TB1-11

†From: + probe
To: - probe

18V at 11 amp regulator part numbers:
Original units = 5709320
Converted units = 5244090

Figure 5-5. SCR Regulator Resistances

If an overvoltage or overcurrent condition exists within the regulator, proceed as follows:

1. Turn off power.
2. Replace Overcurrent/Overvoltage card (or, in converted units, the Overcurrent card), cycle power, and check for normal output. If power remains up and is normal, check reed relay on old card to see if improper seating and poor contact made card fail. If power drops, proceed as described for undervoltage condition above.

Replace the regulator if the failing component(s) is not found.

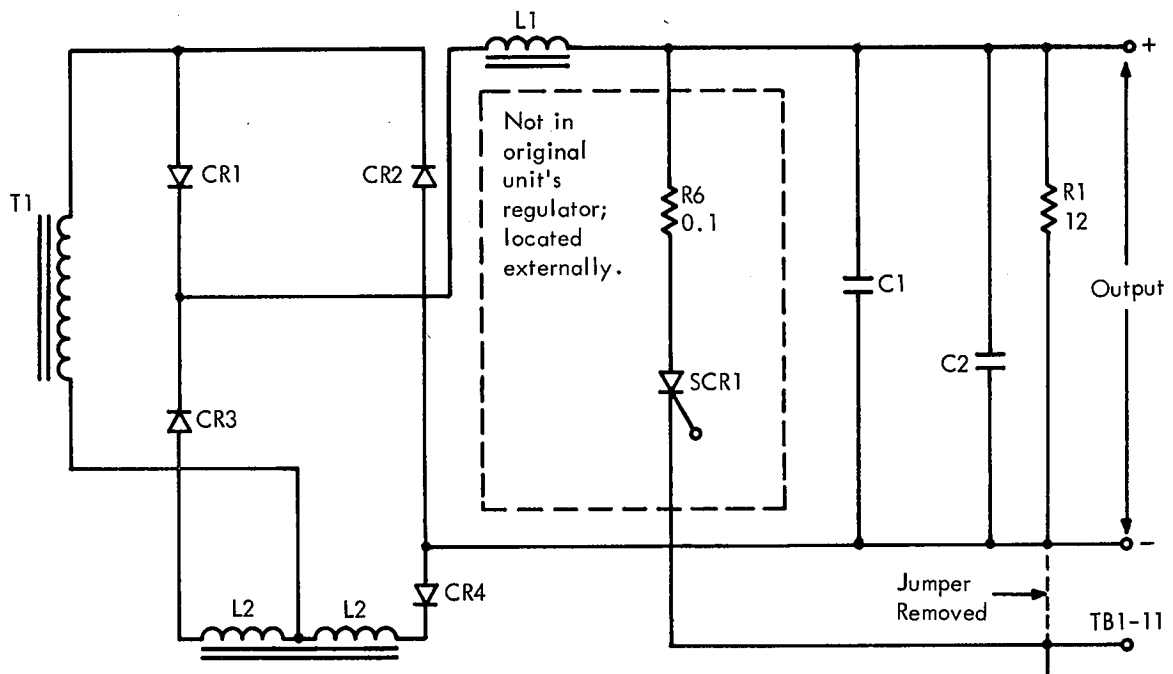
5.5.3.2 Magnetic-Amplifier Regulator

A regulator malfunction may be due to an undervoltage, overvoltage, or overcurrent condition. If an undervoltage condition exists within the regulator, proceed as follows:

1. Turn off power.
2. Replace Amplifier Assembly card, cycle power, and check for normal output.
3. Turn off power and set CB1 off.
4. Remove both SMS cards.
5. Remove all cables at output positive and negative terminals, including jumpers to TB1.
6. Remove cables to TB1-6 and -7.
7. Make resistance measurements between points shown in Figure 5-6. Use page that corresponds to type of regulator under test. Be sure capacitors are fully discharged by measuring voltages at these points first. Use R x 1 scale for resistance measurements.

If an overvoltage or overcurrent condition exists within the regulator, proceed as follows:

1. Turn off power.
2. Replace Overcurrent/Overvoltage card (or, in converted units, the Overcurrent card), cycle power, and check for



Normal Resistances

From †	To †	Ohms
+	-	12
+	TB1-10	High
+	TB1-11	High
-	+	††
-	TB1-10	High
-	TB1-11	High
TB1-10	+	High
TB1-10	-	High
TB1-10	TB1-11	High
TB1-11	+	High
TB1-11	-	High
TB1-11	TB1-10	High
TB1-6	TB1-7	.620

†From: + probe
To: - probe

††See Note 1, Figure 5-5.

3V at 40 amp regulator part numbers:
Original units = 5712020
Converted units = 5261220

Figure 5-6. Magnetic-Amplifier Regulator Resistances (Part 1 of 4)

normal output. If power remains up and is normal, check reed relay on old card to see if improper seating and poor contact made card fail. If power drops, proceed as described for undervoltage condition above.

Replace the regulator if the failing component(s) is not found.

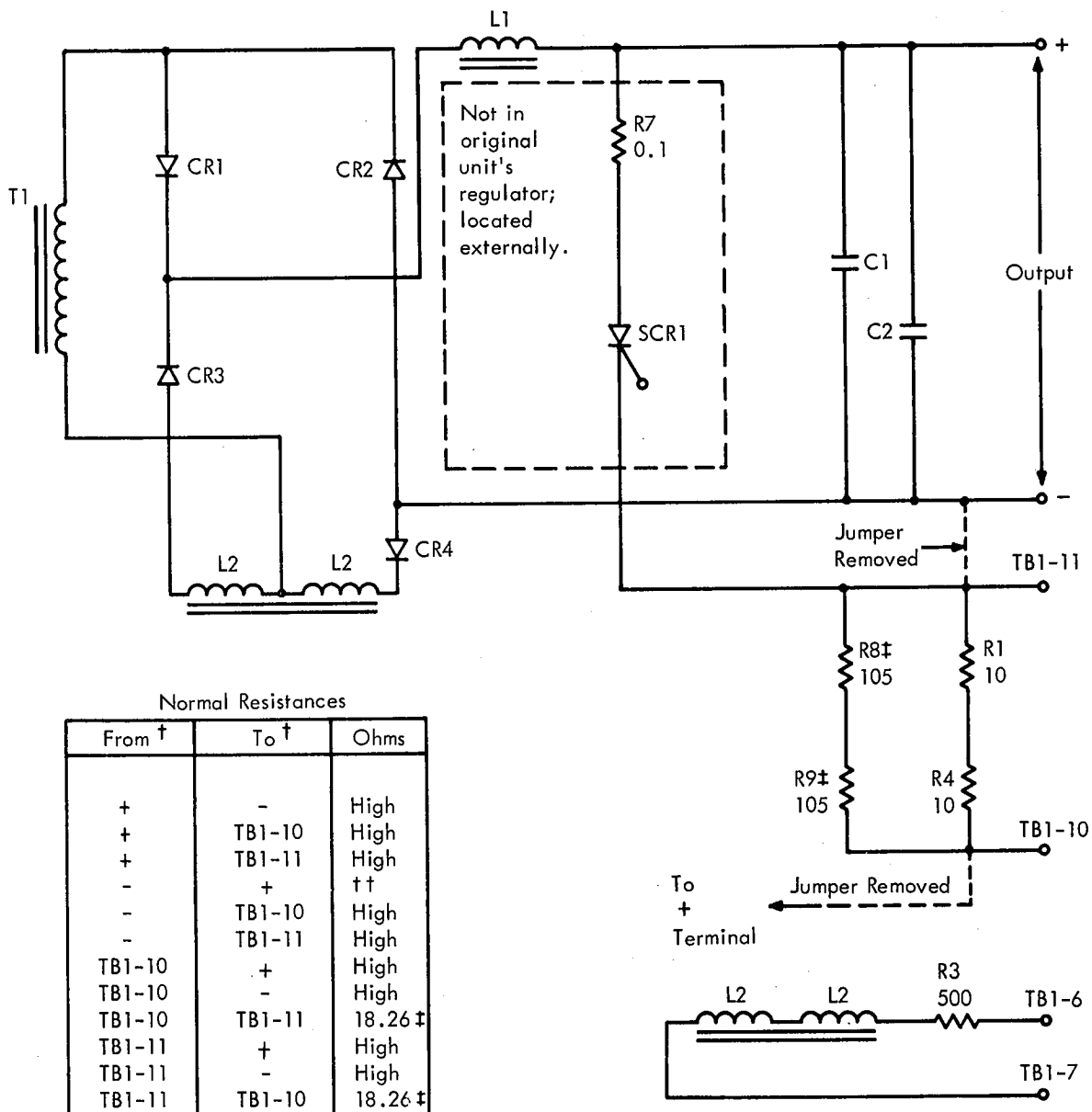
5.6 REPLACEMENT

5.6.1 Converter/Inverter Replacement

To replace the converter/inverter:

1. Turn off wall primary power switch. All power in CPU must be off.
2. As a precaution, set CB1 off.
3. Measure voltage across capacitors to be sure bleeder resistors have reduced voltage to a safe level, preferably 0V.
4. Remove P1 through P6.

DANGER
Inverter/converter power supply, PN 5703200, should not be installed or removed before partially disassembling the unit. The total weight of the supply is 153 pounds. Proper safety procedures require that the unit be disassembled to reduce the weight of each unit to be lifted to a safe amount.



Normal Resistances

From †	To †	Ohms
+	-	High
+	TB1-10	High
+	TB1-11	High
-	+	††
-	TB1-10	High
-	TB1-11	High
TB1-10	+	High
TB1-10	-	High
TB1-10	TB1-11	18.26‡
TB1-11	+	High
TB1-11	-	High
TB1-11	TB1-10	18.26‡
TB1-6	TB1-7	500

†From: + probe
To: - probe

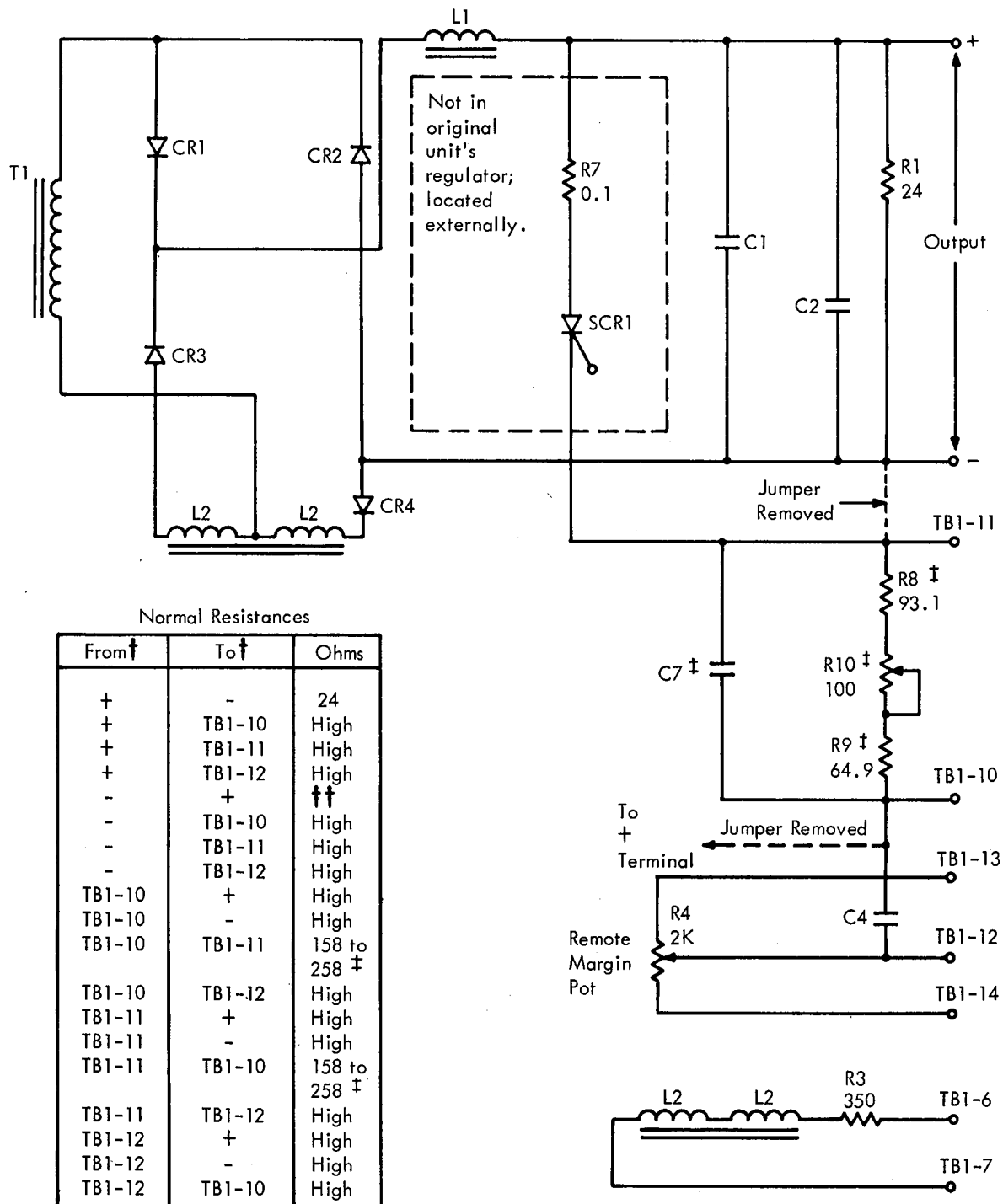
††See Note 1, Figure 5-5.

‡R8 and R9 are not used in the converted units regulator, giving a resistance of 20 ohms between TB1-10 and TB1-11.

6V at 25 amp regulator part numbers:
Original units = 5712030
Converted units = 5261230

Figure 5-6. Magnetic-Amplifier Regulator Resistances (Part 2 of 4)

5. Remove Converter section (left box) and inverter section (right box) from assembly by disconnecting necessary wiring, removing metal stops (one at each bottom and one at each top), and sliding section out of inverter/converter assembly. Always remove inverter section (right side) before converter section. Always install converter unit before inverter unit.
6. Stand assembly on a dolly and roll it away from frame. Do not attempt to carry it.
7. Reverse this procedure to install converter/inverter assembly.



Normal Resistances

From†	To†	Ohms
+	-	24
+	TB1-10	High
+	TB1-11	High
+	TB1-12	High
-	+	††
-	TB1-10	High
-	TB1-11	High
-	TB1-12	High
TB1-10	+	High
TB1-10	-	High
TB1-10	TB1-11	158 to 258 ‡
TB1-10	TB1-12	High
TB1-11	+	High
TB1-11	-	High
TB1-11	TB1-10	158 to 258 ‡
TB1-11	TB1-12	High
TB1-12	+	High
TB1-12	-	High
TB1-12	TB1-10	High
TB1-12	TB1-11	High
TB1-13	TB1-14	2000
TB1-6	TB1-7	350

†From: + probe
To: - probe

†See Note 1, Figure 5-5.

‡C7, R8, R9 and R10 are not used in the converted units regulator, giving a high resistance between TB1-10 and TB1-11.

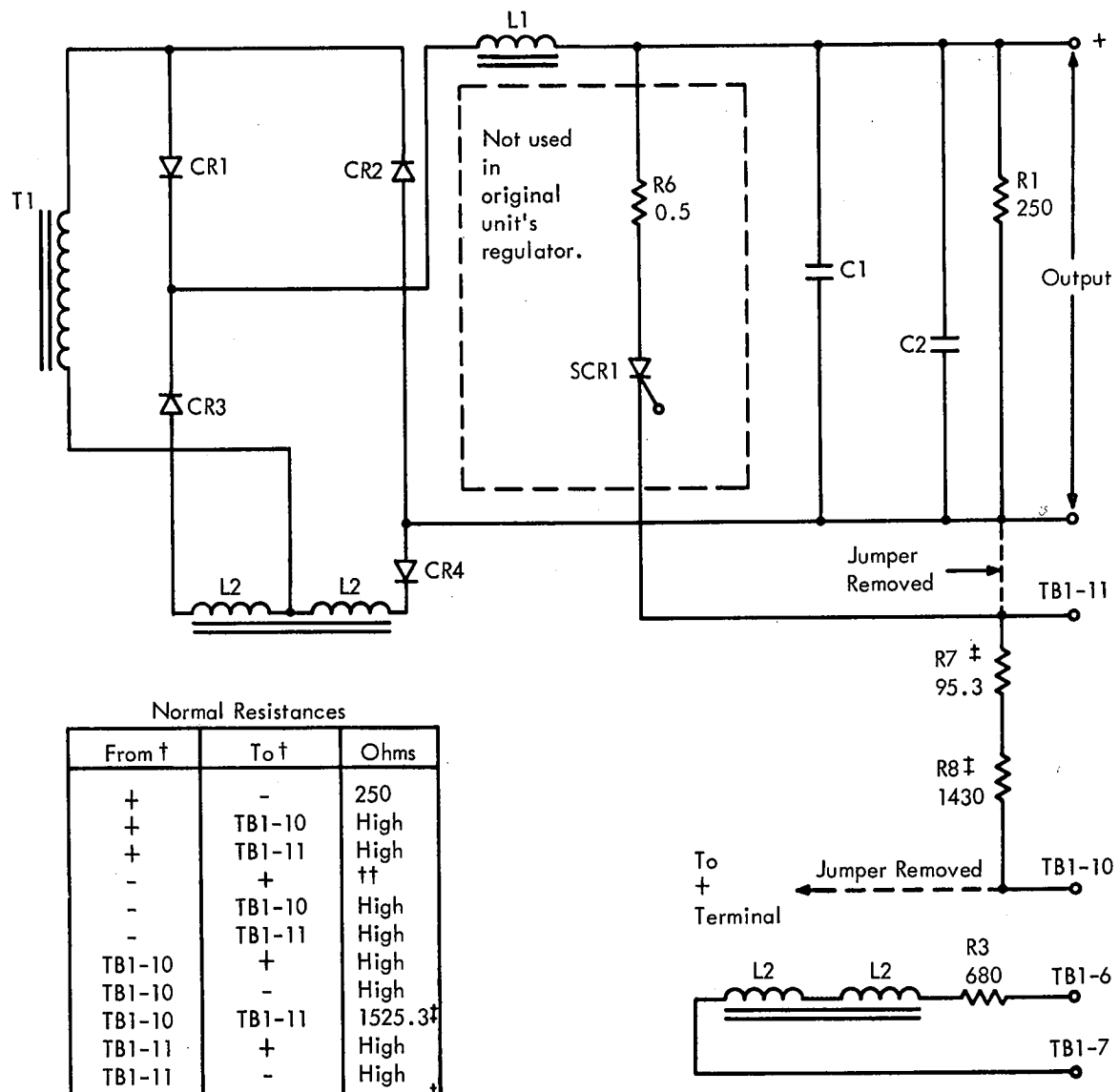
6V at 40 amp regulator part numbers:
Original units = 5712040
Converted units = 5261240

Figure 5-6. Magnetic-Amplifier Regulator Resistances (Part 3 of 4)

5.6.2 Regulator Replacement

To replace the regulator:

1. Turn off wall primary power switch.
2. As a precaution, trip CB1.
3. Remove terminal strip safety shields.
4. Bleed filter capacitors. Disconnect jumper wires connected to filter capacitors.
5. Disconnect and tag each external cable lead for reinstallation.
6. Remove captive mounting screws from nylon standoffs. Remove bottom screw first, then top. Note that these mounting screws are located at rear of regulators.
7. Lift out regulator.
8. Reverse this procedure to install regulator assembly.



Normal Resistances

From †	To †	Ohms
+	-	250
+	TB1-10	High
+	TB1-11	High
-	+	††
-	TB1-10	High
-	TB1-11	High
TB1-10	+	High
TB1-10	-	High
TB1-10	TB1-11	1525.3‡
TB1-11	+	High
TB1-11	-	High
TB1-11	TB1-10	1525.3‡
TB1-6	TB1-7	680

†From: + probe
To: - probe

††See Note 1, Figure 5-5.

‡R7 and R8 are not used in the converted unit's regulator, giving a high resistance between TB1-10 and TB1-11.

48V at 2 amp regulator part numbers:
Original units: 5712081
Converted units: 5261280

Figure 5-6. Magnetic-Amplifier Regulator Resistances (Part 4 of 4)

This chapter contains figures and references to aid the CE in locating components within the CPU.

Figures 6-1 through 6-4 identify the frames and gates of the model 65 configurations: G65, H65, I65, IH65, and J65.

Figure 6-5 identifies the system control panel sections and shows the gate and board arrangement.

Figure 6-6 shows the board locations within gates A, B, C, and E of the CPU and gives the locations of the functional elements within each gate.

Figure 6-7 shows the overall physical layout of the ROS unit, which includes part of gate C and all of gate D.

Figure 6-8 shows the CPU power tubs and power control for converted and original units and identifies the power supplies.

Figure 6-9 is a hinge end view of the four gates of the CPU. The figure shows the location and numbering of the laminar bus terminal boards.

Figure 6-10 shows the external cable routing of a Model I65 system.

Figure 6-11 identifies the components in frame 02: the right and left power tubs, right and left I/O panels, the prime power box, and panel 02X-T1.

Figure 6-12 shows the pin designations on a large board.

Figure 6-13 shows the card contact, board-pin relationship and the voltage pins for the cards.

Figure 6-14 gives the layout of the address cards for the 1052 Adapter. Two types of cards are shown to cover the two types in usage. Using this figure it is possible to determine the correct plug arrangement for any address.

Figure 6-15 shows the switch contact terminal locations for switches of the system control panel and the configuration control panel.

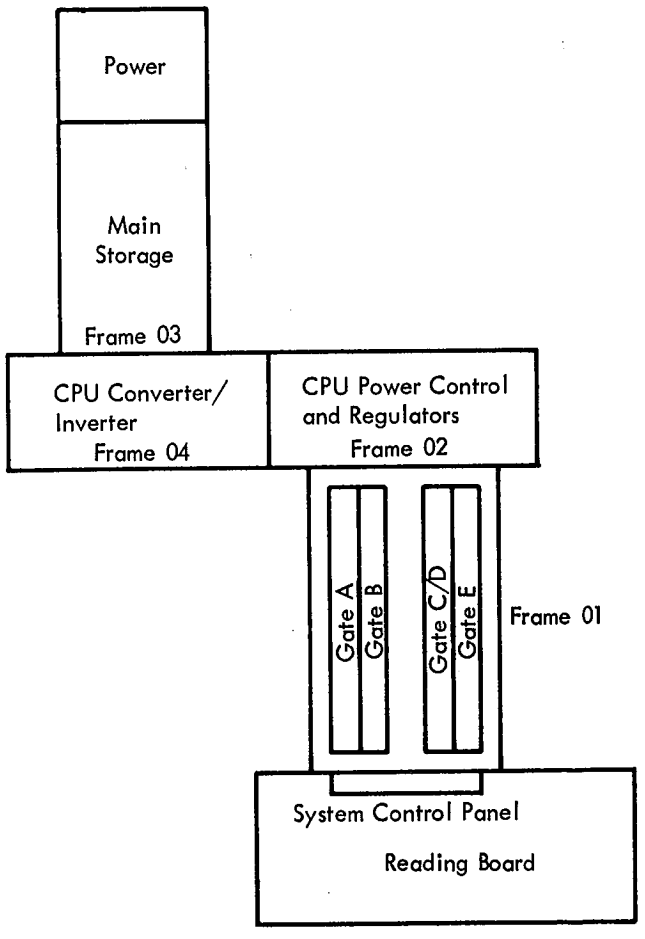


Figure 6-1. Overall CPU and Main Storage Layout, Models G65 and H65

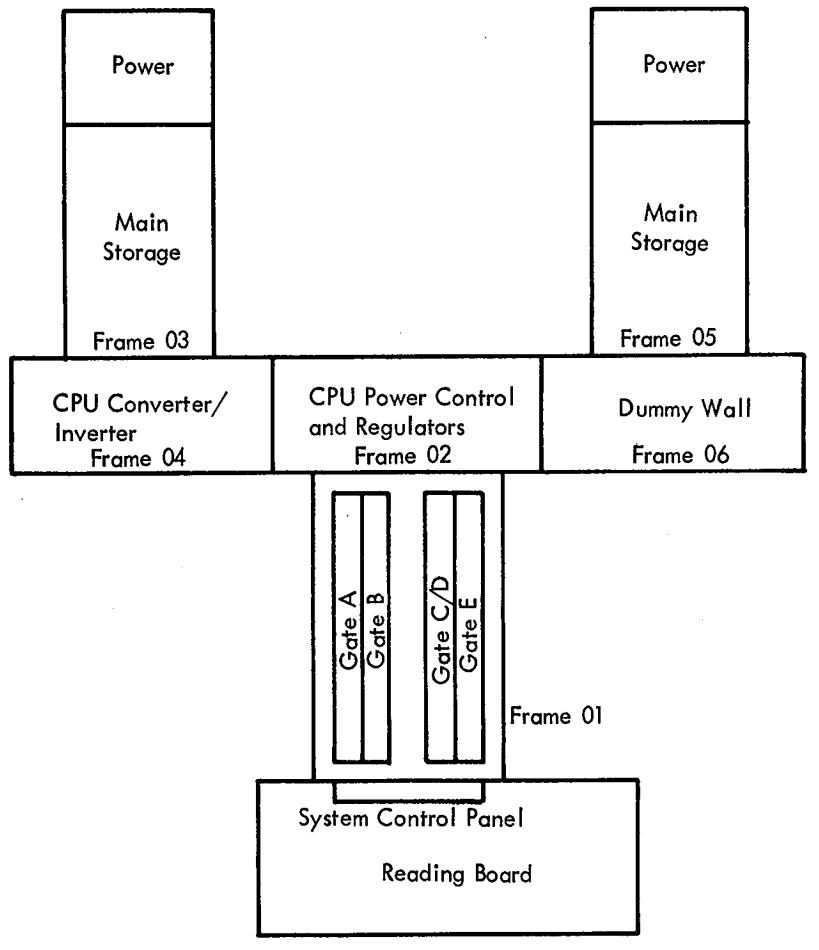


Figure 6-2. Overall CPU and Main Storage Layout, Model I65

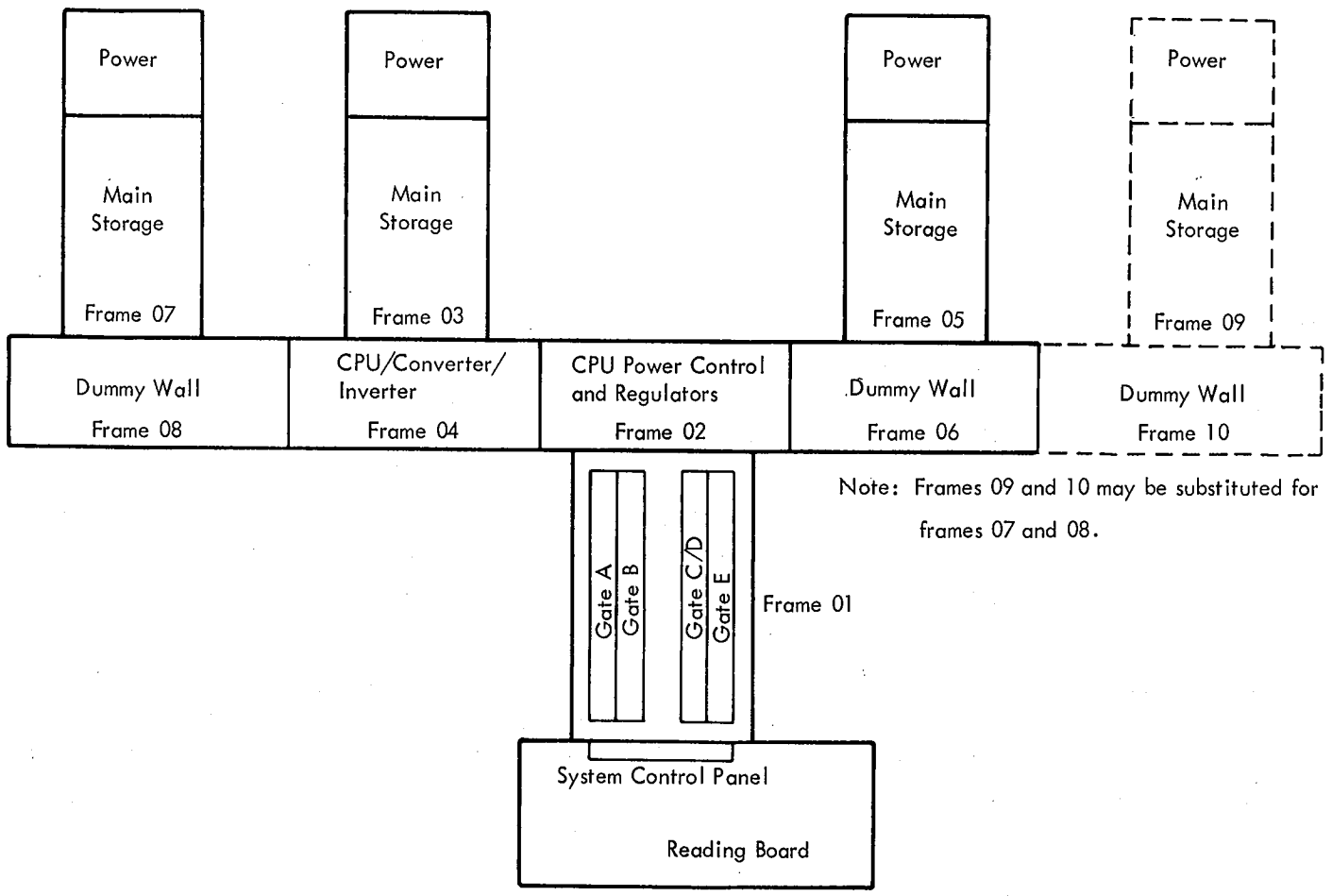


Figure 6-3. Overall CPU and Main Storage Layout, Model IH65

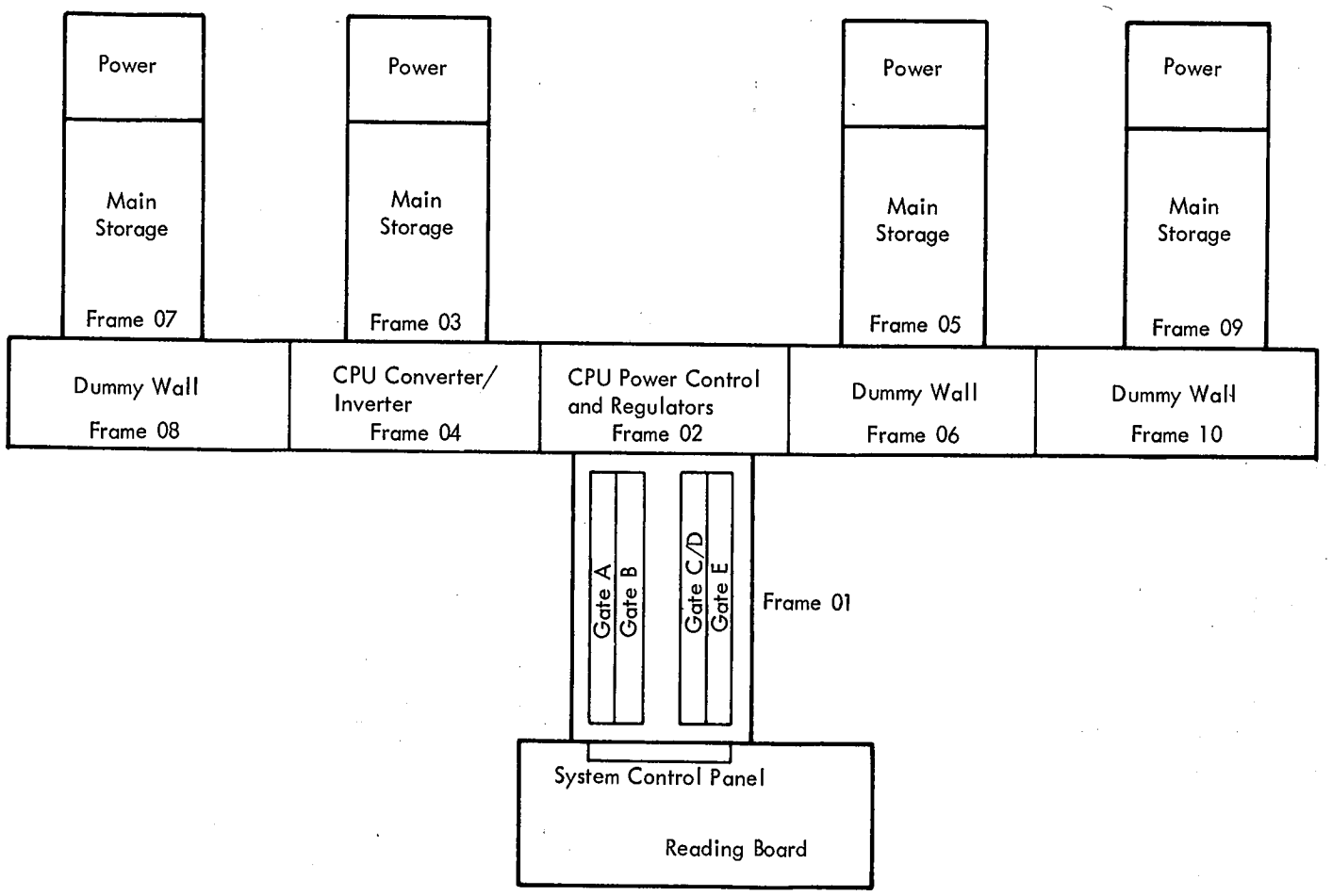


Figure 6-4. Overall CPU and Main Storage Layout, Model J65

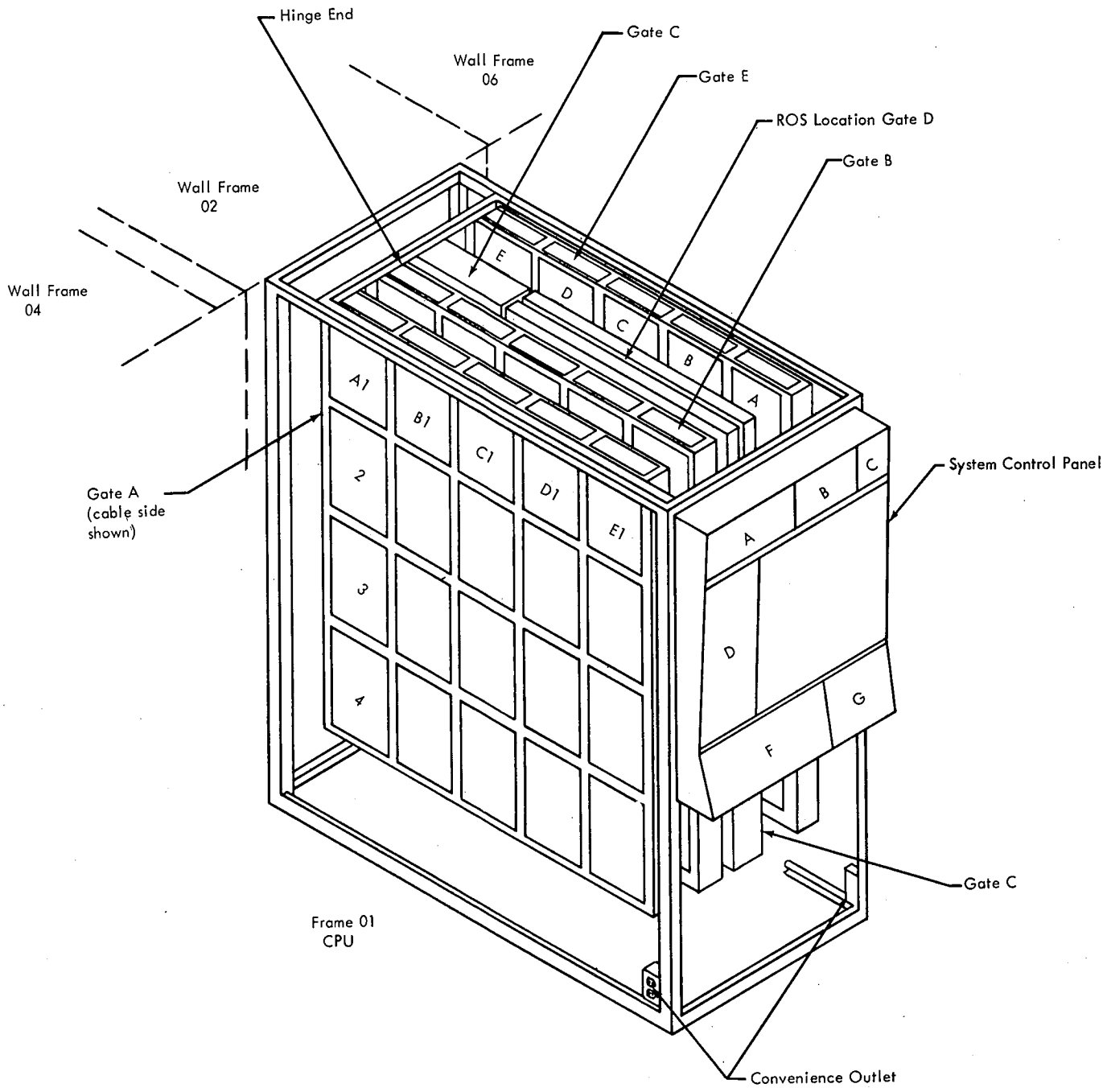
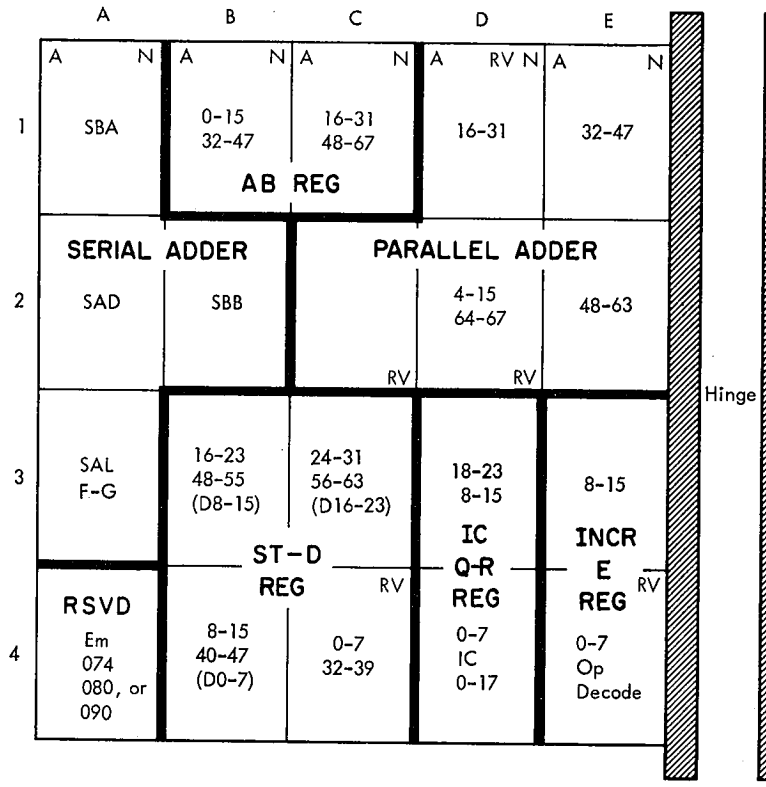
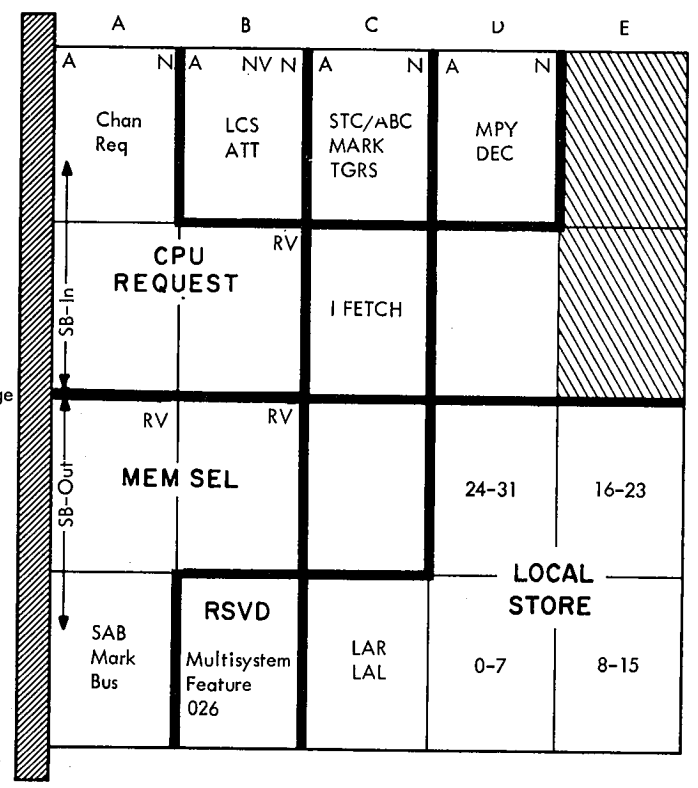


Figure 6-5. CPU Gate and Board Locations

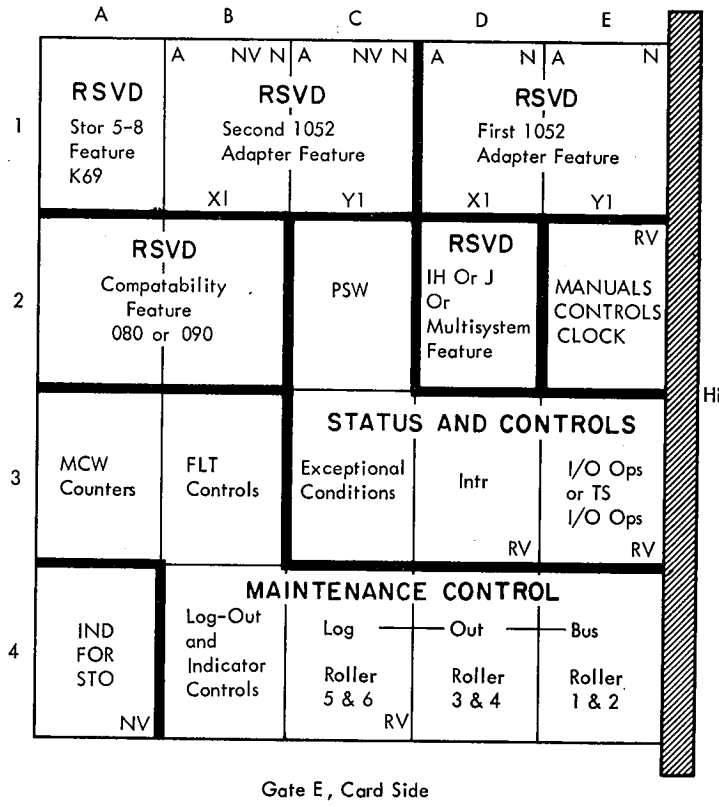


Gate B, Card Side

Hinge

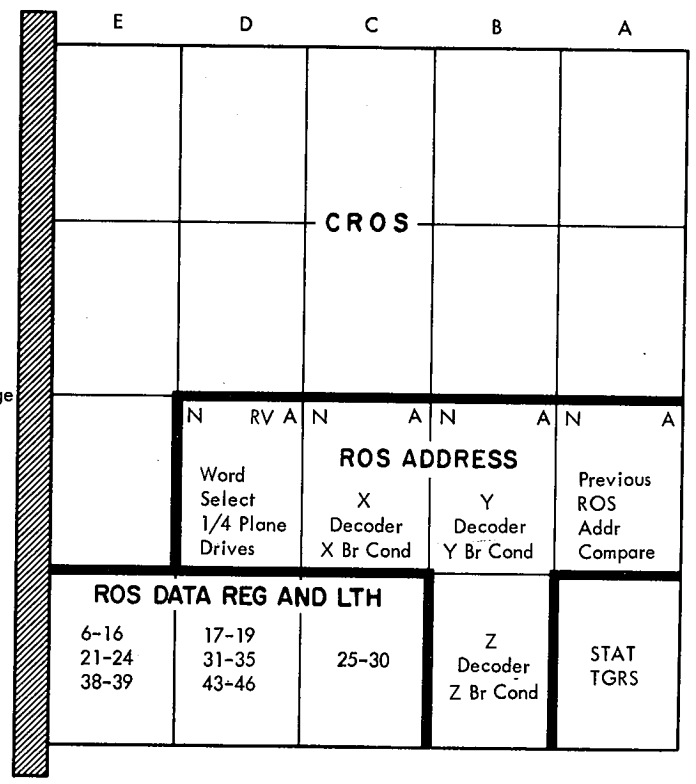


Gate A, Card Side



Gate E, Card Side

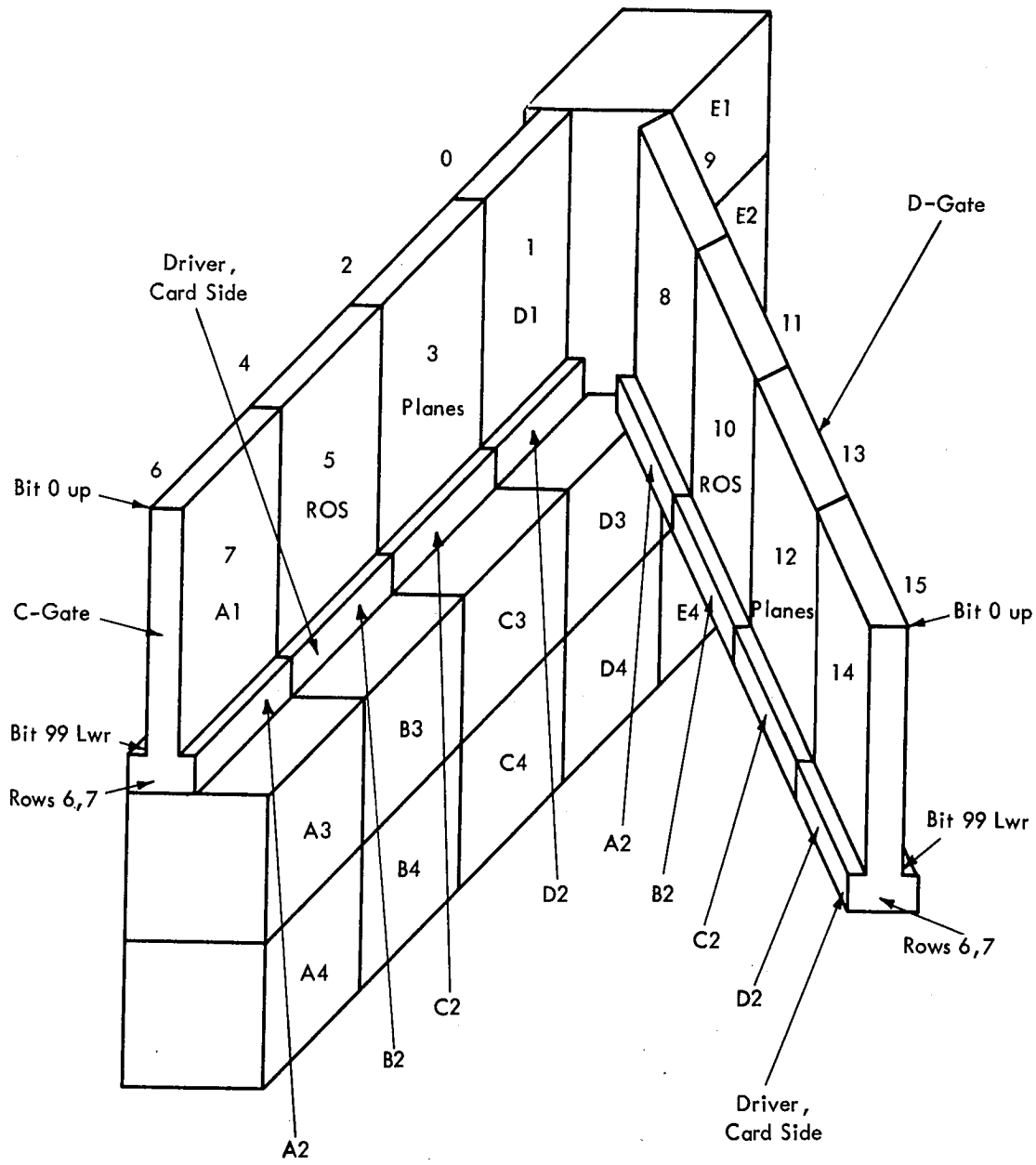
Hinge



Gate C, Pin Side

RV Replacement Version
NV New Version

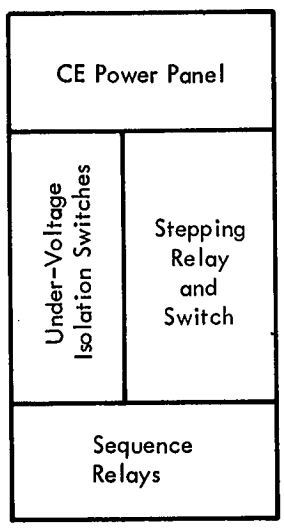
Figure 6-6. Large Board Assignment (Functional)



See ALD ED000-ED020 for detail of ROS layout.

Figure 6-7. ROS Physical Layout

Converted
Power Control



Original
Power Control

PS-1 +6 C/D Upper	PS-2 +3 B Lower	CE Power Panel and Isolation Switches	PS-9 +48 1052	PS-10 -3 A, B, E Upper & Lower
PS-3 +3 E Lower	PS-4 -3 C/D Upper & Lower	Sequence Relays	PS-11 -18 C/D Lower	PS-12 +3 A Lower
PS-5 +3 2150 B, E Upper	PS-6 +6 C/D Upper & Lower		PS-13 +3 B Upper	PS-14 +3 A Upper
PS-7 +6M E Upper & Lower	PS-8 +6M C/D Lower	Stepping Switch and Control	PS-15 +6M B Upper & Lower	PS-16 +6M A Upper & Lower

Upper and Lower refers to the upper and lower laminar bus terminal boards on the gates.

See ALD YA161-YA165 for more detail

Figure 6-8. CPU Power Layout

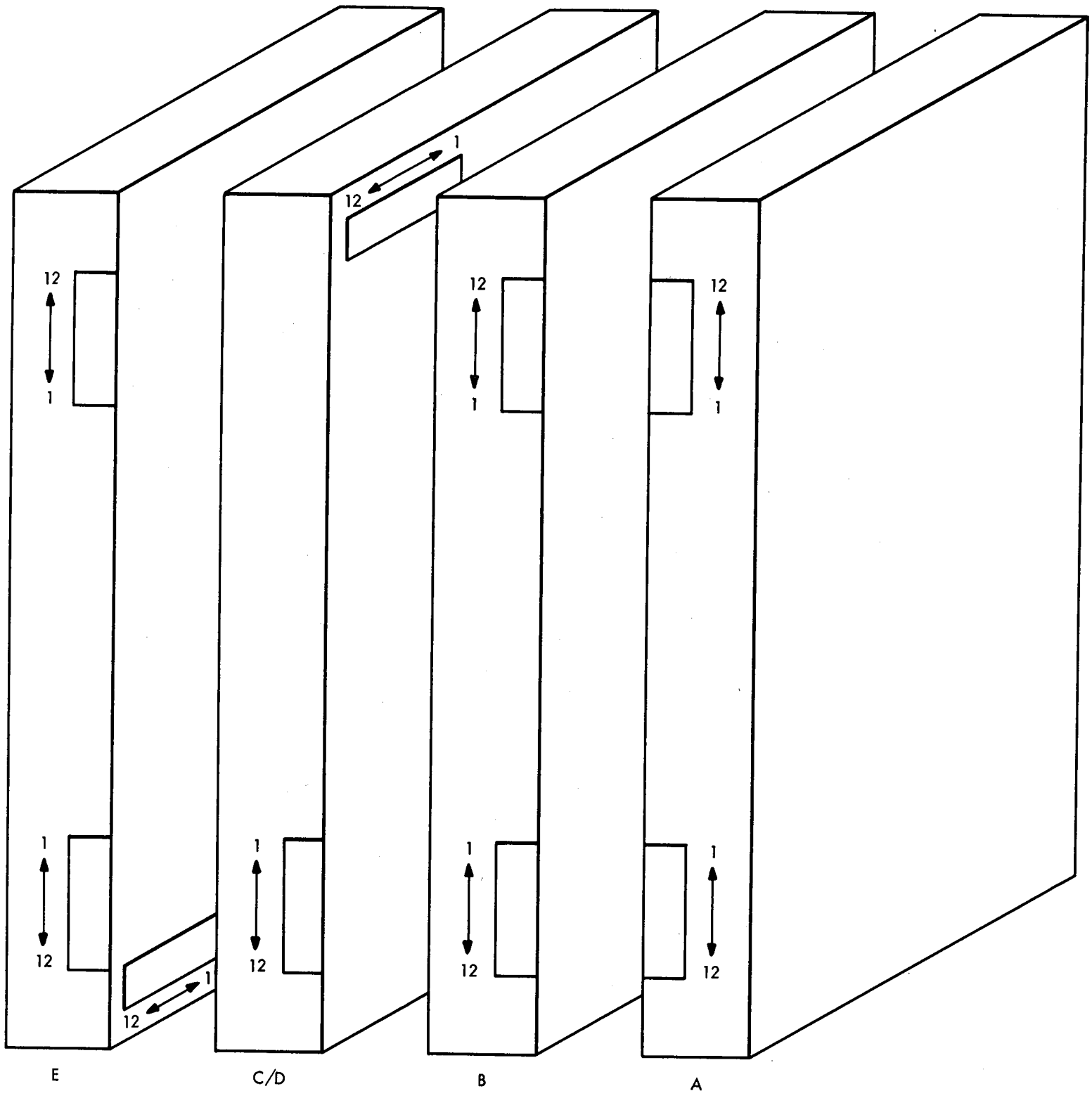
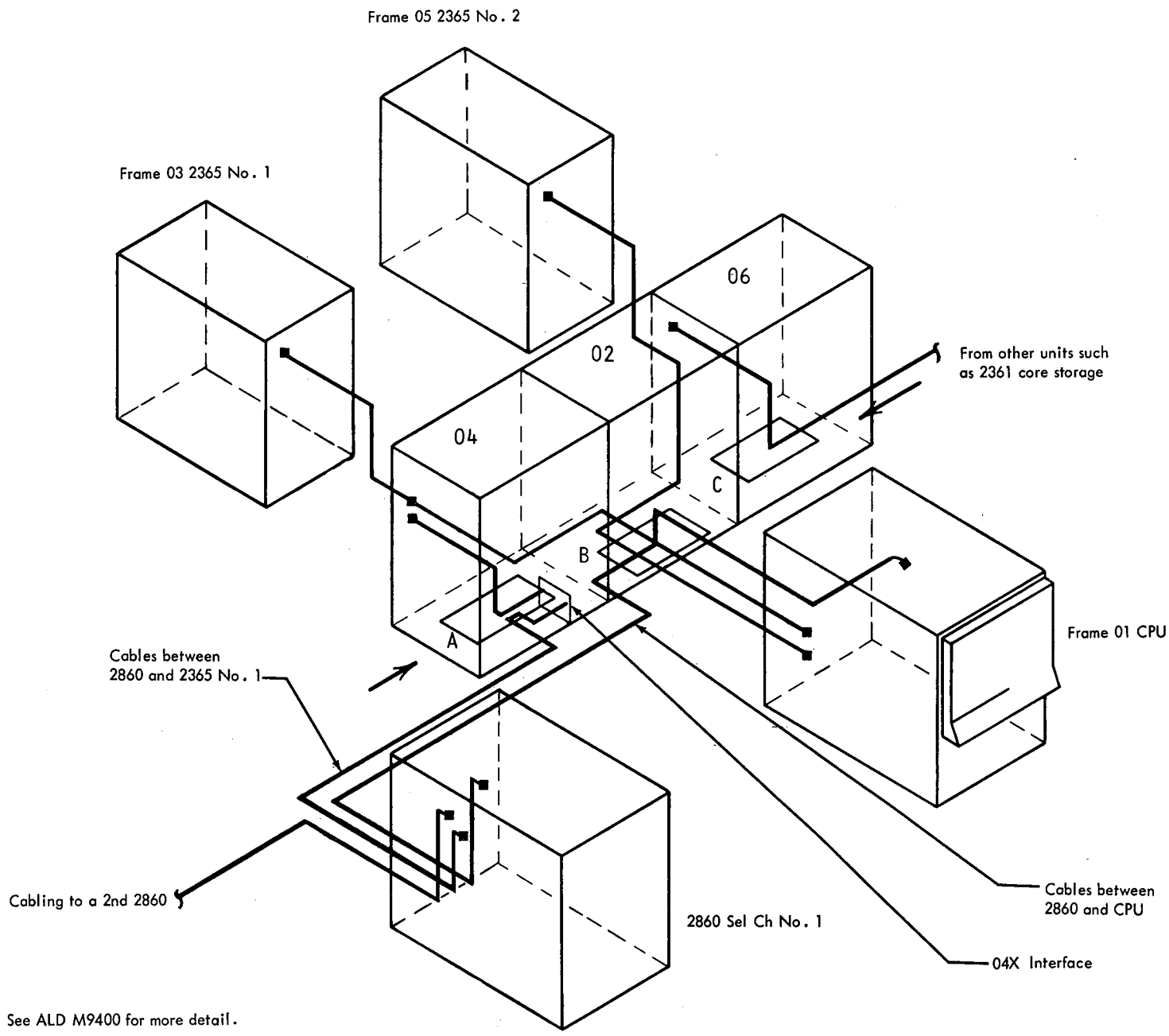


Figure 6-9. Laminar Bus Terminal Board Locations (CPU)



See ALD M9400 for more detail.

NOTES:

1. Not all cables are shown. This illustration shows only how cables enter and leave the units. Note how wall frame access holes A, B, and C are used to route cables between units. For above-floor installations, all cables enter and leave wall frames at points indicated by arrows.

Figure 6-10. Example of External Cable Routing

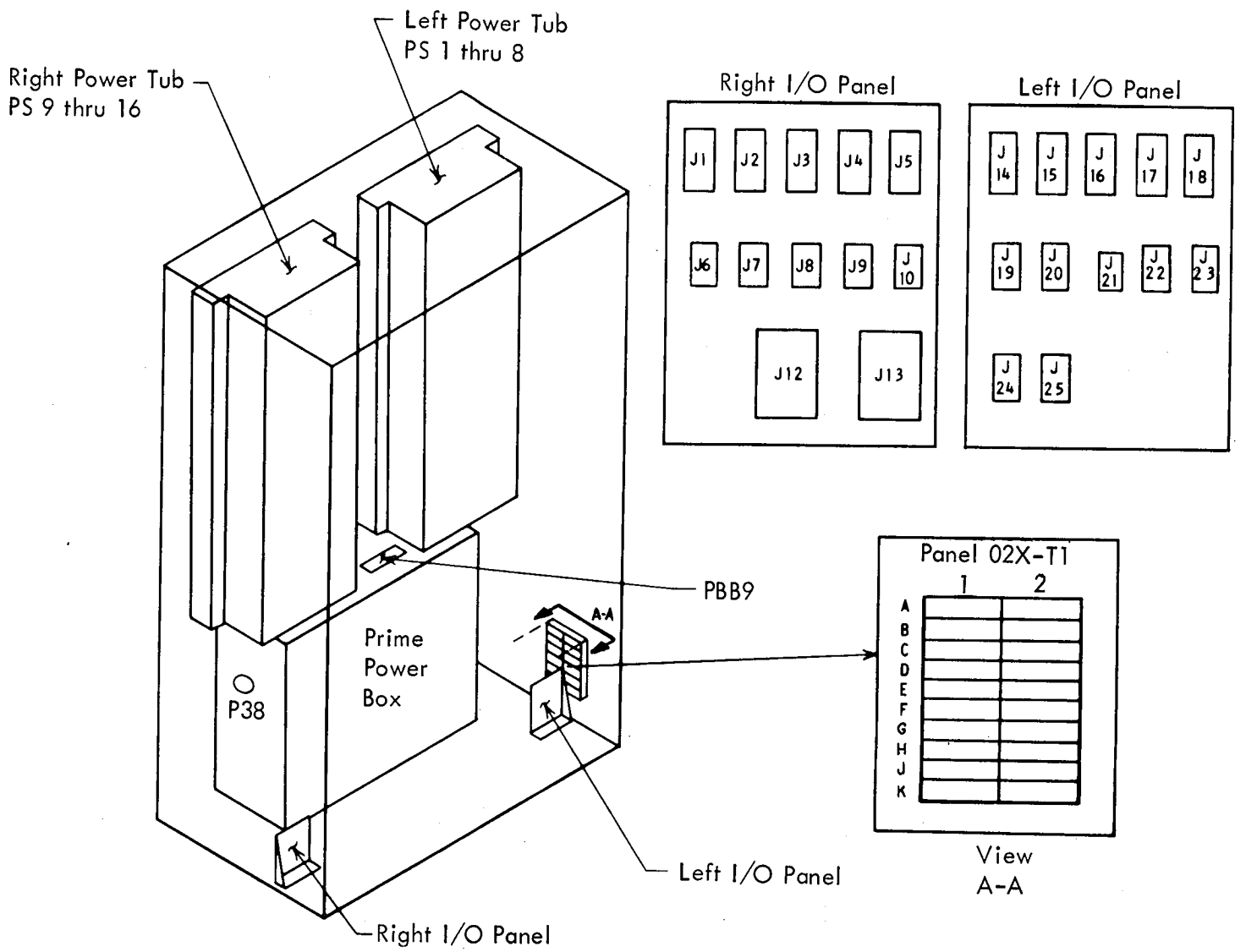


Figure 6-11. Frame 02 Component Locations

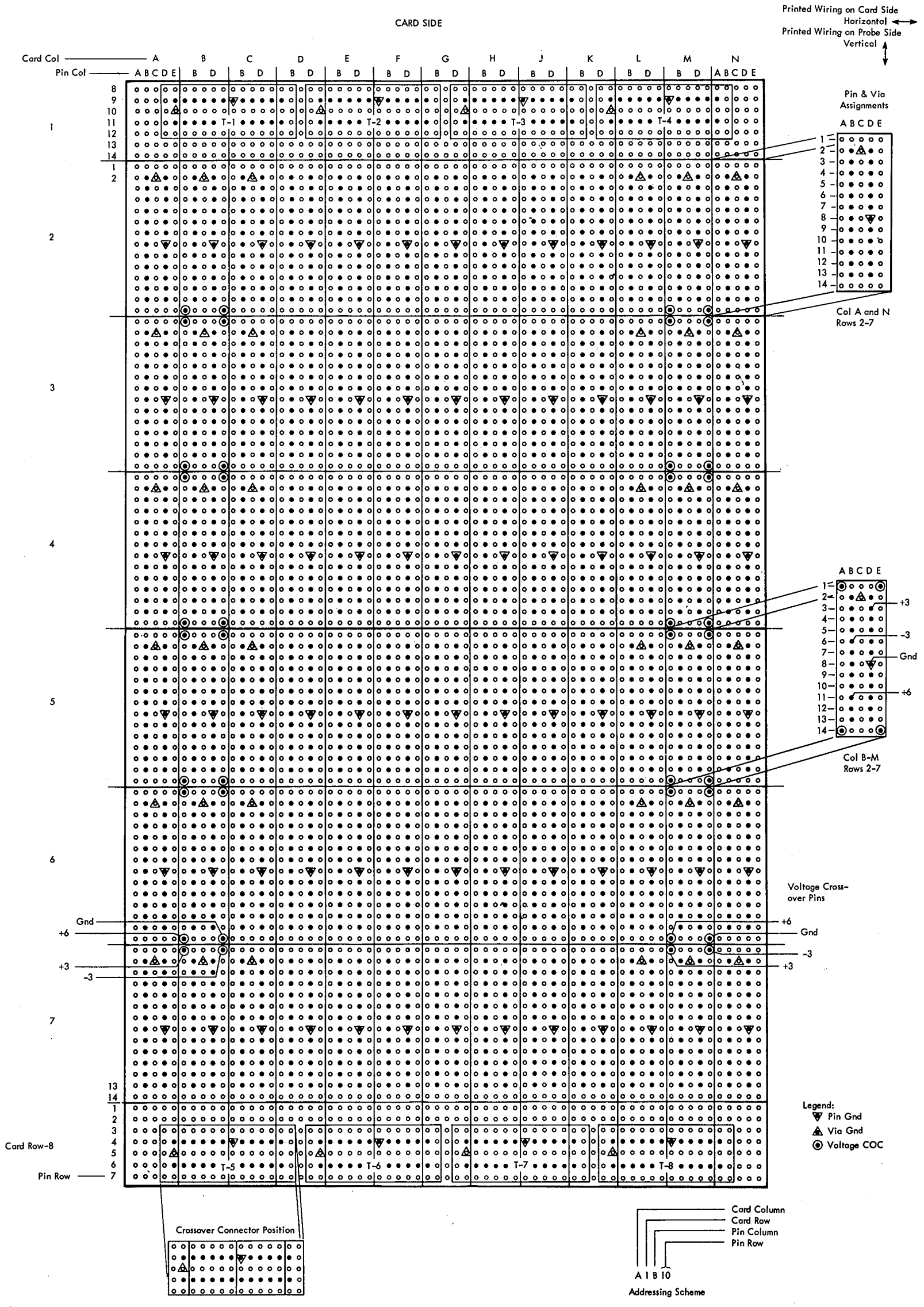
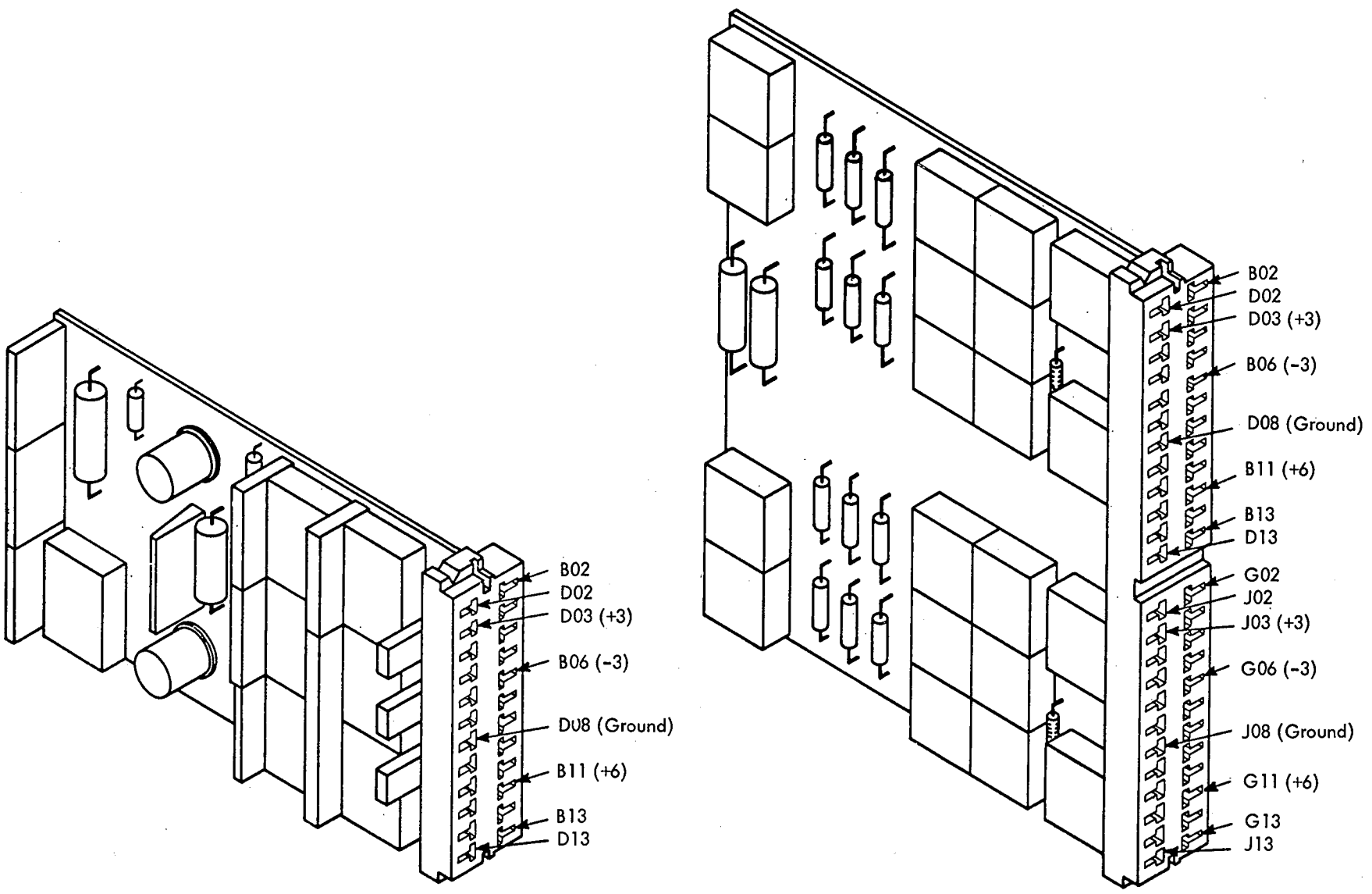


Figure 6-12. Large Board Pin Addresses



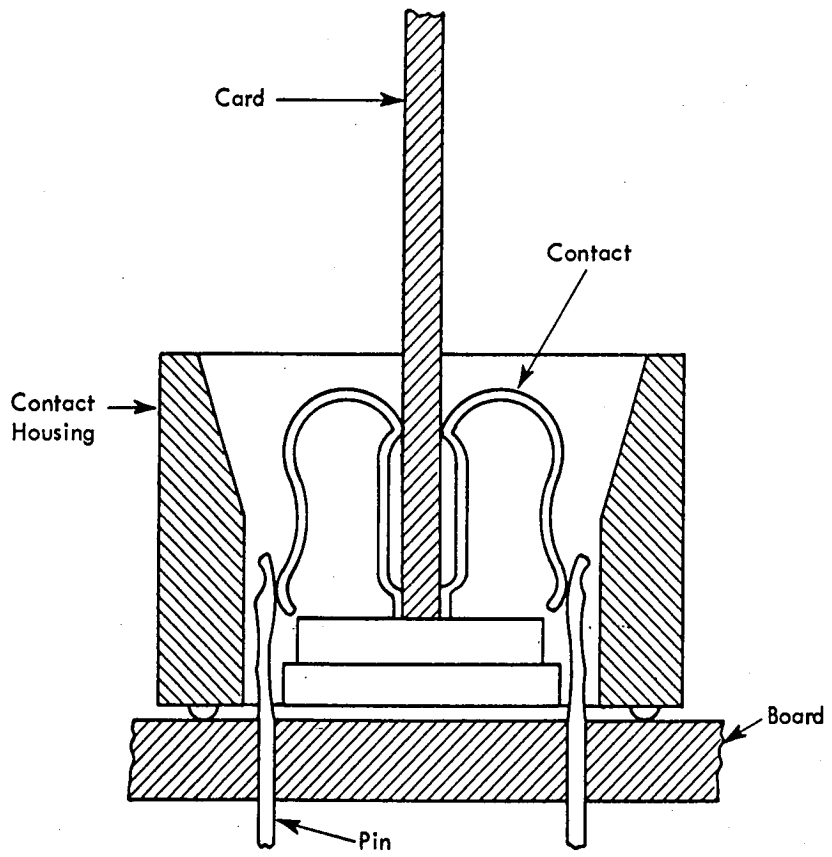
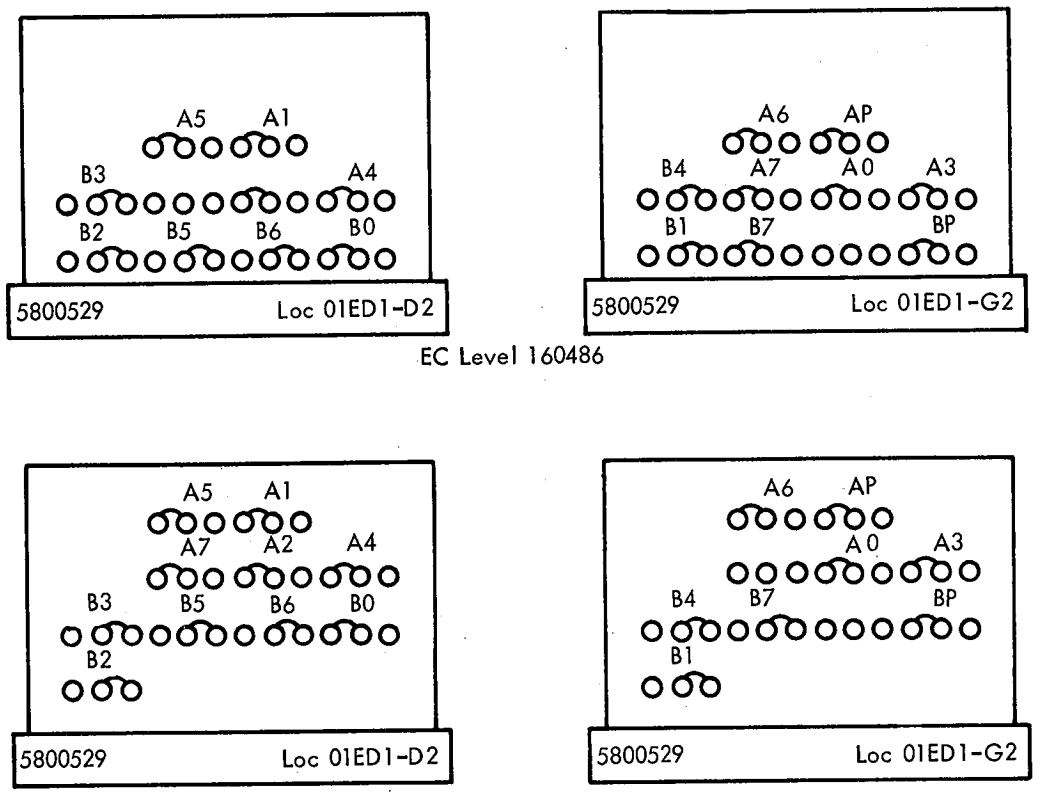


Figure 6-13. Card Contact, Board-Pin Relationship



EC Level 160486

EC Level 167197B

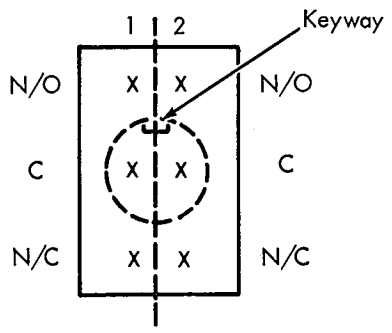
A Address Bit
B Bus Out

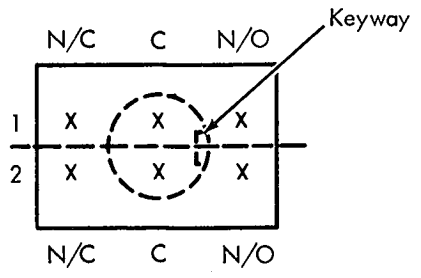
All bits are shown wired inactive.
Example: For address 09 (0000 1001P) wire as shown except reverse the wires for the active bits A4, A7, AP, B4, B7, BP; wire from center to left or center to right for each group of three terminals.

See ALD PG031, M931 for more detail.

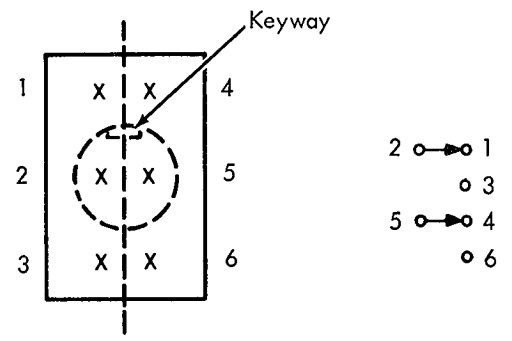
Figure 6-14. Address Card Layout for 1052 Adapter

Pushbutton Switch



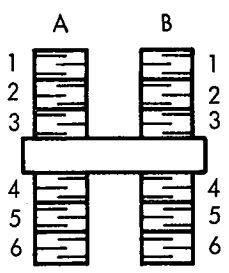


DPDT Toggle Switch

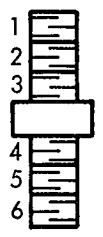


Lever Switches

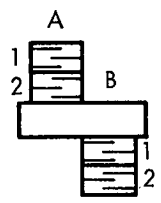
Double Lever

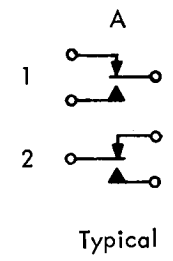


Single Lever



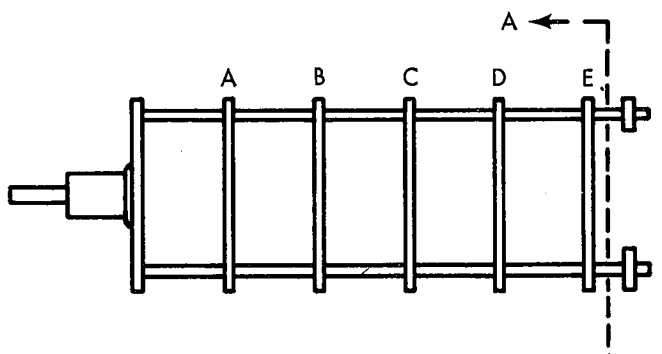
Double Lever Variation





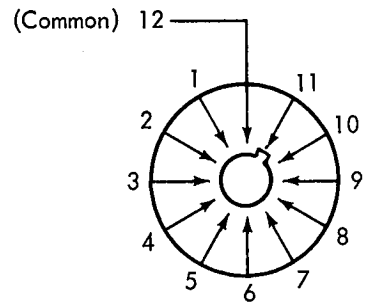
Typical

Rotary Switch



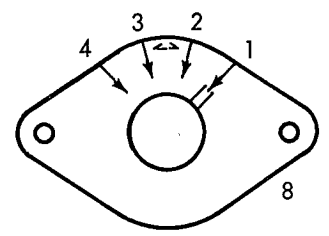
Deck E

Rotary Switch



View A-A

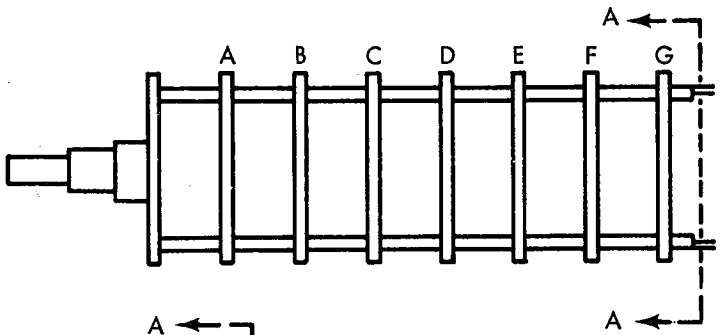
Rotary Switch

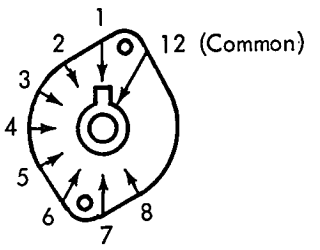


Storage 5-8 Feature

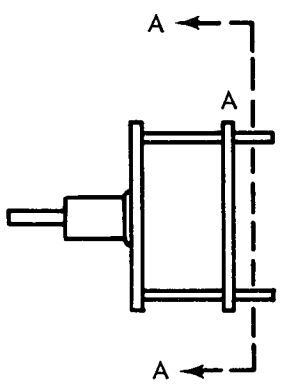
(A) SYSTEM CONTROL PANEL

Rotary Switch

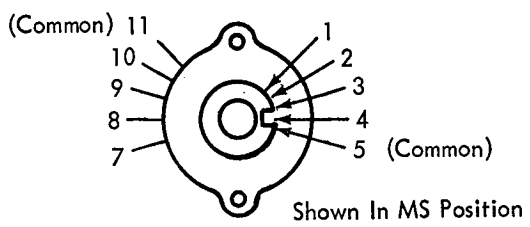




View A-A



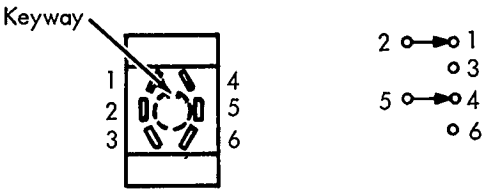
Rotary Switch



Shown In MS Position

View A-A

DPDT Toggle Switch



(B) CONFIGURATION CONTROL PANEL (MULTISYSTEM FEATURE ONLY)

Figure 6-15. Switch Contacts (Wiring Side)

Appendix A. Special Circuits

The special circuits used in the CPU (in the ROS hardware) are not field-repairable. They must be returned to the factory with a failure report.

Appendix B. Voltmeter Calibration Chart

Figure B-1 below is a removable chart that may be completed with applicable information and kept with the CPU for future reference.

MARGIN/METER SEL Switch Position	Margin Status	Actual† Voltage	Measurement Location	Indicated Voltage			
STORE FRAME 1	Nominal High Low						
STORE FRAME 2	Nominal High Low						
STORE FRAME 3	Nominal High Low						
STORE FRAME 4	Nominal High Low						
CHAN FRAME 1	Nominal High Low						
CHAN FRAME 2	Nominal High Low						
CHAN FRAME 3	Nominal High Low						
ROS LOCATE (PS11)	Nominal High Low	†† 20v 7v					
CPU A (PS16)	Nominal High Low	 6.5v 5.5v					
CPU B (PS15)	Nominal High Low	 6.5v 5.5v					
CPU C (PS8)	Nominal High Low	 6.5v 5.5v					
CPU E (PS7)	Nominal High Low	 6.5v 5.5v					
Date of Calibration							
Calibrated By							

† Measure according to text procedure, paragraph 5.4.
†† Set by optimization procedure, paragraph 4.9.

CPU Serial Number _____

Figure B-1. System Control Panel Voltmeter Calibration Chart