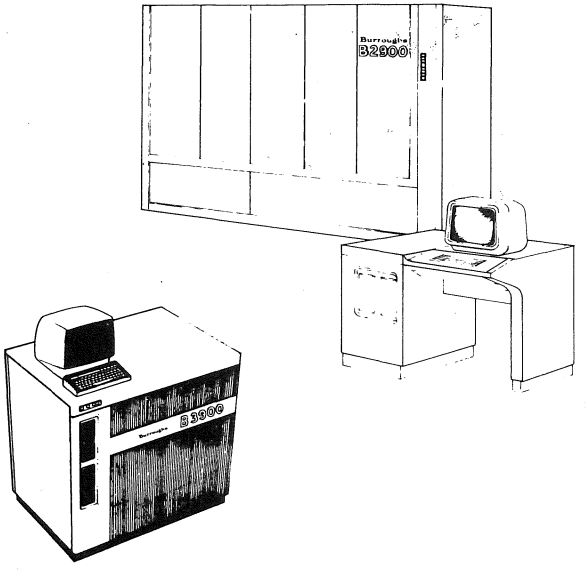
Burroughs

B 2900/B 3900 System

HANDBOOK



FIELD ENGINEERING PROPRIETARY DATA

F.E. Dist. Code CC

Printed in U.S. America June 1982 1137858

Note: This complete reprint includes information released under the following PCN's:

PCN 1137558-001

Correspondence regarding this publication should be forwarded using the Remarks form at the back of the manual, or may be addressed directly to TIO West Documentation, Burroughs Corporation, 1300 John Reed Court, City of Industry, California 91745, U.S.A.

LIST OF EFFECTIVE PAGES

Page	Issue
Title	Original
ii	Original
iii	Original
iv	Blank
v thru xvi	Original
1-1 thru 1-154	Original
2-1 thru 2-32	Original
3-1 thru 3-9	Original
3-10	Blank
4-1 thru 4-56	Original
4-57 thru 4-58	PCN-001
4-59 thru 4-72	Original
5-1 thru 5-11	Original
5-12	Blank

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont)

LIST OF ILLUSTRATIONS (Cont)

LIST OF TABLES

LIST OF TABLES (Cont)

LIST OF TABLES (Cont)

SECTION 1

GENERAL INFORMATION

Documentation

A documentation scheme called Cardmap is used to identify all of the components and interconnections between the components that exist on a printed circuit board (card). The cardmap for each card is found in the B 2900/B 3900 T&F documentation package. Each cardmap consists of four or five sections:

1. IC Documentation (figure 1-1)
2. Card Edge Connectors (figure 1-2)
3. Multipoint Circuit List (figure 1-3)
4. Discrete Component List (figure 1-4)
5. PROM Truth Tables

IC DOCUMENTATION

IC, up to 96, are identified according to their physical location on a card. The 96 locations, A0 through Q5, are used to identify the IC residing at the corresponding position on a card (see figure 1-1).

NOTES

(A) IC location specifier.

(B) IC type specifier. This includes a four digit DA mnemonic, as referenced in the Logic Device Representation Book (LDRB), referenced in the Logic Device R, and a description of the component.

(C) Schematic representation of the IC.

(D) Signal names of each input and output.

(E) Signal source. Multiple source is indicated by a + following the location. The entire signal net is listed in the Multipoint Circuit List.

(F) Additional load point for this signal.

(G) Load or sink location of signal.

(H) Load or sink location of signal. The asterisk (*) denotes multiple sinks; multiple sources are indicated by the legend, S+, following the load location.

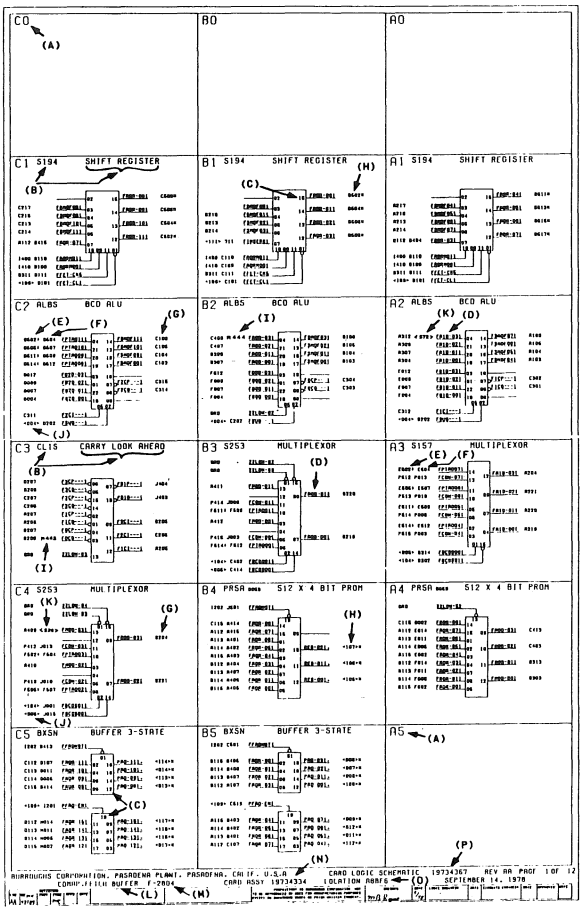


Figure 1-1. Card Map, IC Documentation

(I) Resistor specifier, see Discrete Component List.

(J) Backplane pin specifier.

(K) Frontplane pin specifier.

(L) Card name.

(M) File number of card (used only by the factory).

(N) Assembly part number of card.

(O) Location of card in the system.

(P) Part number of the Card Map.

CARD EDGE CONNECTORS

Cardmap contains a listing of the signals that are present at each back/frontplane pin of a card. Figure 1-2 illustrates a page of card edge connector documentation.

NOTES

(Q) Logic signal name.

(R) Backplane pin number.

(S) Frontplane pin number.

(T) Signal source.

(U) Signal load.

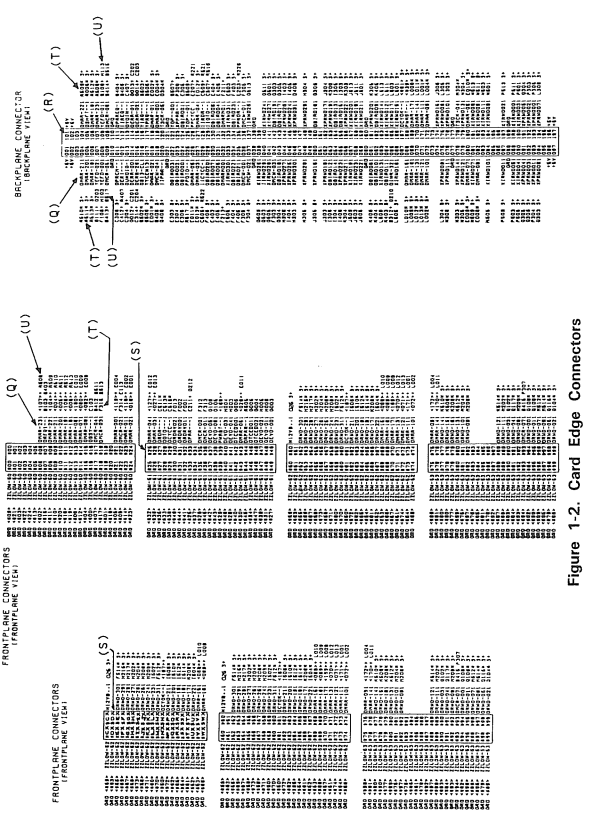


Figure 1-2. Card Edge Connectors

MULTIPOINT CIRCUIT LIST

The Multipoint Circuit List is a list of all circuits that connect two or more points on a card. An example of this list is figure 1-3.

NOTES

(AA) Signal name of circuit. Circuits are listed in alphabetical order.

(AB) Circuit list for signal. It lists all sources and loads for the signal on the card.

(AC) Source designator. It lists the location of a signal source.

(AD) Load designator. It gives the location of the signal load.

(AE) Bidirectional designator. The listed location is both a source and a load for the signal.

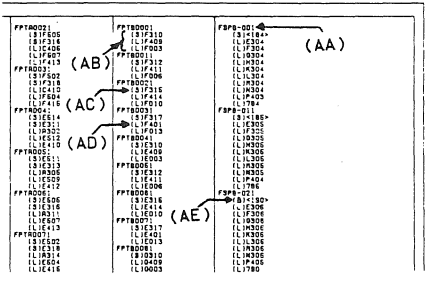


Figure 1-3. Multipoint Circuit List

DISCRETE COMPONENT LIST

The Discrete Component List (figure 1-4) of the cardmap contains all circuit information relative to each of the discrete components on a card. This list also contains any notes that describe features of the card that may not otherwise be described in the cardmap, such as jumper options.

NOTES

(BA) Component specifier.

(BB) Description of component.

(BC) Logic signal name.

(BD) Component locator, location of the component. Physical location cross reference to location call outs is found in the LDRB.

(BE) Circuit tie point, the location of a source or a load of the signal of the card.

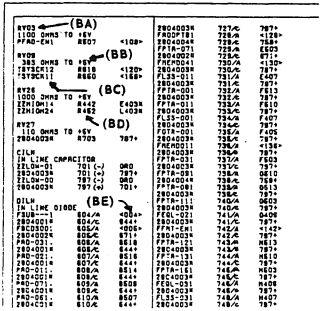


Figure 1-4. Discrete Component List

PROM TRUTH TABLE

Cards with ROM have their bit patterns represented in the last portion of the cardmap.

Block Diagram

Figure 1-5 is a block diagram of the processor, IOT, and memory.

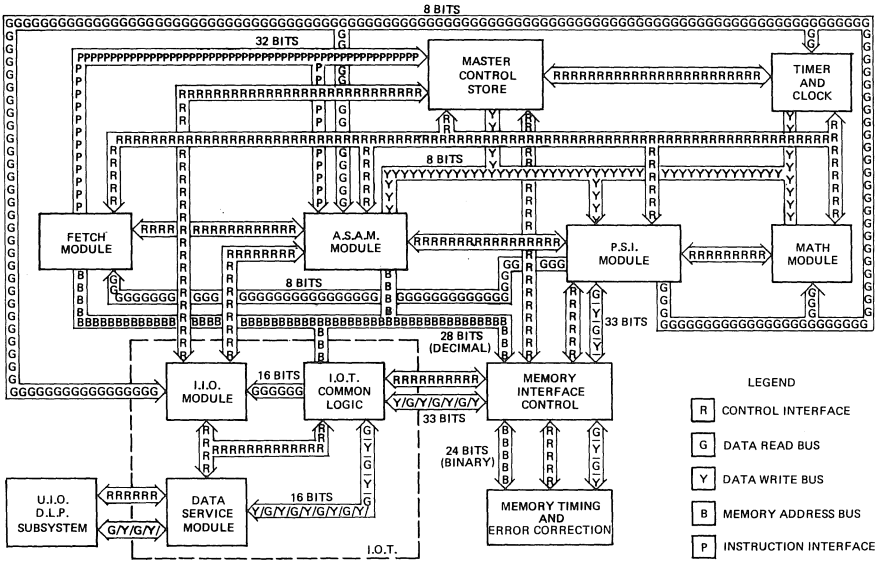


Figure 1-5. B 2900 Block Diagram

Control Store RAM Maps

The following tables are maps of the control store RAM. These tables list micro-code identifiers, bit numbers, chip locations, and card locations. If a failing micro-operator or control is known, failing RAM chips can be located by using the tables.

Table 1-1A. Math Module (QVMC) (Card Location: ABBG6)

Chip Location	Bit Number	Micro Code	Description
G2	67	Parity	
H2	66	Parity	
H1	66	Parity	
H0	64		(enable undigit check)
I2	63		(most significant bit of Branch Condition)
I1	62	QVMAD	(most significant two bits
I0	61	QVMAD	of next address)
J0	60		(clear all registers)
J1	59	QBQP-MO	(count BQP up)
J2	58	OUT1-EN	(enable of VL4 mux)
K2 K1	57 56		(select AQ, ABQ, or CQ through VL4 mux)
K0	55	CAVL	(enable counting without EDP or CHREQ)
L0	54	OUT2-EN	(enable of VL3 mux)
L1 L2	53 52		(select AQ, ABQ, or CQ through VL3 mux)
M1	51	WCALL	(WCAVL to PSI)
M0	50	VMRC	(MWBY mux select)
N1	49	VMRC	(MWBY mux select)
N0	48	VMRC	(MWBY mux select)
P1	47	QVMAD	(next address)
P0	46	QVMAD	(next address)
Q0	45	QVMAD	(next address)
Q1	44	QVMAD	(next address)
G1	43	QVMAD	(next address)
G0	42	QVMAD	(next address)

Table 1-1A. Math Module (QVMC) (Card Location: ABBG6) (Cont)

Chip Location	Bit Number	Micro Code	Description
F0	41	QVMAD	(next address)
F1	40	QVMAD	(next address)
F2	39	QCQP	(count CQP up)
E2	38		(set CQP to FF)
E1	37		(set BQP to FF)
E0	36		(set AQP to FF)
D0	35	QAIN	(set AQP to FF)
D1	34		(set CQW to FF)
D2	33		(set BQW to FF)
C0	32		(set AQW to FF)
C1	31		(carry in for binary ALU)
C2	30		(count CQP or set COML)
B2	29		(count BQP or set sign neg)
B1	28		(count AQP or set sign plus)
B0	27	RAMDIS	(ALU in Binary Mode)
A2	26		(set Overflow or count CQW)
A1	25		(set Com Strobe or count BQW)
A0	24		(set ComH or count AQW)

Table 1-1B. Math Module (QVMC) (Card Location: ABBG4)

Chip Location	Bit Micro Number	Code	Description
F1	23		(select binary ALU or Accm open)
F2	22		(select binary ALU or Accm open)
E0	21		(select binary ALU or Accm open)
E1	20		(select binary ALU or Accm open)
E2	19		(enable wrap around from MWBX to CQI)
D0	18		(subtract Mode for BCD ALU)
D1	17		(carry in for BCD ALU)
D2	16		(translate control on incoming BQP)
C0	15		(put incoming BQP in 00 direction)
C1	14		(count AQW up)
C2	13		(count BQW up)
B0	12		(count CQW up)
B1	11		(least significant four bits of branch condition)
B2	10		(least significant four bits of branch condition)
A0	9		(least significant four bits of branch condition)
A1	8		(least significant four bits of branch condition)
I0	7		(hard logic operators)
J0	6		(hard logic operators)
K0	5		(hard logic operators)
L0	4		(hard logic operators)
F0	3		(load AQ from MRR)
G1	2		(load BQ from MRR)
G0	1		(load BQ from MRR)
H0	0	QAQP-M0	(queue A address pointer)

Table 1-2. IIO Module (IIRM) (Card Location: ABBC4)

Chip Location	Bit Number	Micro Code	Description
D0	55	IIPAR	(parity)
P0	54	IICR	Not Used
P0	53 for 2	IMOP	(UOP field)
N0	51 for 4	IMOP	(UOP field)
M0	47 for 2	IMOP	(UOP field)
M0	45 for 2	IMSL	(UOP select
I0	43 for 4	ICLT	(ALU literal field)
K0	39 for 4	ICLT	(ALU literal field)
J0	35 for 4	IIIN	(ALU instruction)
H0	31 for 4	IIIN	(ALU instruction)
G0	27	IIIN	(ALU instruction)
G0	26 for 3	IIAA	(ALU A address)
F0	23	IIAA	(ALU A address)
F0	22 for 3	IIAB	(ALU B address)
E0	19	IIAB	(ALU B address)
E0	18 for 3	ICSN	(Sequencer address)
D0	15 for 4	ICSN	(Sequencer address)
C0	11	ICSN	(Sequencer address)
C0	10 for 2	IIRM	(Sequencer stack control)
C0	8	ITASL	(Test section A)
B0	7 for 2	ITASL	(Test section A)
B0	5 for 2	ITBSL	(Test section B)
A0	3	ITBSL	(Test section B)
A0	2 for 3	ITSL	(Sequencer instruction)

Table 1-3A. Service Module (ISRM) (Card Location: ABBB6)

Chip Location	Bit Micro Number	Operator	Description
Q3	71	ISPAR	(parity <even> bit)
Q2	70		Not Used
Q1	69		Not Used
Q0	68	ISXSL	(test cond expand bit)
P3	67	IUOP	(micro-operator field)
P2	66	IUOP	(micro-operator field)
P1	65	IUOP	(micro-operator field)
P0	64	IUOP	(micro-operator field)
N3	63	IUOP	(micro-operator field)
N2	62	IUOP	(micro-operator field)
N1	61	IUOP	(micro-operator field)
N0	60	IUOP	(micro-operator field)
M2	59	IUOP	(micro-operator field)
M1	58	IUOP	(micro-operator field)
M0	57	IUOP	(micro-operator field)
L2	56	IUOP	(micro-operator field)
L1	55	IUSL	(micro-operator group)
L0	54	IUSL	(micro-operator group)
K2	53	ISAL	(A test condition select)
K1	52	ISAL	(A test condition select)
K0	51	ISAL	(A test condition select)
J2	50	ISBL	(B test condition select)
J1	49	ISBL	(B test condition select)
J0	48	ISBL	(B test condition select)
I2	47	ISSN	(Sequencer direct input)
I1	46	ISSN	(Sequencer direct input)
I0	45	ISSN	(Sequencer direct input)
H2	44	ISSN	(Sequencer direct input)
H1	43	ISSN	(Sequencer direct input)
H0	42	ISSN	(Sequencer direct input)
G2	41	ISSN	(Sequencer direct input)
G1	40	ISSN	(Sequencer direct input)
G0	39	ISSN	(Sequencer direct input)
F2	38	ISSN	(Sequencer direct input)
F1	37	ISRM-FE	(Sequencer FE)
F0	36	ISRM-PP	(Sequencer PUP)

Table 1-3B. Service Module (ISRM) (Card Location: ABBB4)

Chip Location	Bit Number	Micro Operator	Description
Q2	35	ITSS	(Sequencer instruction)
Q1	34	ITSS	(Sequencer instruction)
Q0	33	ITSS	(Sequencer instruction)
P2	32	ISDI	(ALU direct input)
P1	31	ISDI	(ALU direct input)
P0	30	ISDI	(ALU direct input)
N2	29	ISDI	(ALU direct input)
N1	28	ISDI	(ALU direct input)
N0	27	ISDI	(ALU direct input)
M2	26	ISDI	(ALU direct input)
M1	25	ISDI	(ALU direct input)
M0	24	ISDI	(ALU direct input)
L2	23	ISDI	(ALU direct input)
L1	22	ISDI	(ALU direct input)
L0	21	ISDI	(ALU direct input)
K2	20	ISDI	(ALU direct input)
K1	19	ISDI	(ALU direct input)
K0	18	ISDI	(ALU direct input)
J2	17	ISDI	(ALU direct input)
J1	16	ISIN	(ALU destination, function, source)
J0	15	ISIN	(ALU destination, function, source)
I2	14	ISIN	(ALU destination, function, source)
I1	13	ISIN	(ALU destination, function, source)
I0	12	ISIN	(ALU destination, function, source)
H0	11	ISIN	(ALU destination, function, source)
G0	10	ISIN	(ALU destination, function, source)
F0	19	ISIN	(ALU destination, function, source)
E0	8	ISIN	(ALU destination, function, source)
D0	7	ISAA	(ALU A address)
D1	6	ISAA	(ALU A address)
C0	5	ISAA	(ALU A address)
C1	4	ISAA	(ALU A address)
B0	3	ISAB	(ALU B address)
B1	2	ISAB	(ALU B address)
A0	1	ISAB	(ALU B address)
A1	0	ISAB	(ALU B address)

Table 1-4A. Fetch Module – 60-Bits (FFCS) (Card Location: ABBD0)

Chip Location	Bit Number	Micro Operator
A3	59	Next address
B5	58	Next address
C4	57	Next address
C5	56	Next address
B3	55	Next address
D0	54	Next address
D1	53	Next address
D2	52	Next address
E1	51	Next address
E0	50	Next address
B0	49	USL
C2	48	USL
A0	47	
C3	46	FE
C0	45	PUP
C1	44	Write enable
B1	43	Write enable
C1	44	Write enable
A4	41	Write enable
B4	40	Write enable
A1	39	Write enable
B2	38	Write enable
A2	37	Write enable
F0	36	Eight-way select
E2	35	Four-way select
G2	34	Four-way select
F1	33	Two-way select
F2	32	Test select
G0	31	Test select
G1	30	Test select
I2	29	Two-way select
J2	28	
I1	27	Two-way select
J1	26	Memory counter load
Q0	25	Buffer address
Q1	24	Buffer address
P1	23	Buffer address
J0	22	Buffer address
I0	21	Buffer address
P0	20	Buffer address
N0	19	Buffer address
L0	18	Buffer address
L2	17	Shift
L1	16	Parity

Table 1-4B. Fetch Module – 60-Bits (FFCS) (Card Location: ABBD0)

Chip Location	Bit Number	Micro Operator 1 = Format Mode	0 = Calculate Mode
N2	15		
M1	14	Get IX flag	Base MUX select
M2	13	Clear flags	Base MUX select
N1	12	INX Reg sign load	Base Request load
M0	11	INX Reg sign load	Single instruct flag
Q2	10	INX Reg sign load	Load Address Reg
P3	9	ADD on Data Bus	Maintenance flag
P2	8	Load OP Register	FTOE signal
Q3	7	Formatter MUX select	FTOE signal
K0	6	Formatter MUX select	Load Literal Reg
K1	5	Formatter MUX select	ASAM out/in flag
K2	4	CXC load	Load error flag
Q4	3	BFF load	ALU A MUX select
Q5	2	BXC load	ALU A MUX select
P5	1	AFF load	0 = add; 1 = sub
P4	0	AXC load	ALU B MUX select

Table 1-5. Address Store & Manipulate Module (ASAM) – 60-Bits (AACS)

(Card Location: ABBF0)

Chip Locations Address 000 - 0FF	Address 100 - 1FF	Bit Number	Micro Operator
Q0	Q1	59 for 4	Next
D0	P1	55 for 4	Next
N0	N1	51 for 2	Next
N0	N1	49 for 2	BIN-BCD and Test TRUE
M0	M1	47 for 4	SPA
L0	L1	43 for 4	SBP
K0	K1	39	SPL
K0	K1	38	CSR
K0	K1	37	BL Select
K0	K1	36	PRA/BL Select
J0	J1	35 for 4	Write Enable
H0	H1	31	BLS/ALO Select
H0	H1	30 for 2	Adder MUX
H0	H1	28	SP INPUR MUX
G0	G1	27	SP INPUR MUX
G0	G1	26	Adder Load
G0	G1	25	Subtract Mode
G0	G1	24	SPA Load

Table 1-5. Address Store & Manipulate Module (ASAM) – 60-Bits (Cont) (AACS)

(Card Location: ABBF0)

Chip Locations			
Address		Bit	
000 - 0FF	100 - 1FF	Number	Micro Operator
F0	F1	23	SPB Load
F0	F1	22	ASAM Idle
F0	F1	21	
F0	F1	20	Literal
E0	E1	19 for 3	Literal
E0	E1	16	MRB char select/MRB driver
D0	D1	15 for 2	MRB char select/MRB driver
D0	D1	13	IOT GO
D0	D1	12	Data flags
C0	C1	11 for 2	Data flags
C0	C1	9	WRT ad ready
C0	C1	8	
B0	B1	7 for 4	Test select
A0	A1	3 for 2	Test select
A0	A1	1 for 2	

Table 1-6A. MCS (CCS) – 37-Bits
(Card Location: ABBF4)

Chip Locations				
Address			Bit	
000 - 3FF	100 - 7FF	800 - FFF	Number	Micro Operator
E3	F3	D3	36	Parity
A0	A1	A2	35	S1PL
B0	B1	B2	34	S0PL
C0	C1	C2	33	SEL
D0	D1	D2	32	SEL
E0	E1	E2	31	SEL
F0	F1	F2	30	Wait
G0	G1	G2	29	FE-NOT
H0	H1	H2	28	PUP
I0	I1	I2	27	DEST
J0	J1	J2	26	DEST
K0	K1	K2	25	DEST
L0	L1	L2	24	MOP
M0	M1	M2	23	MOP

Table 1-6A. MCS (CCS) – 37-Bits (Cont)
(Card Location: ABBF4)

Chip Locations				
Address			Bit	
000 - 3FF	100 - 7FF	800 - FFF	Number	Micro Operator
N0	N1	N2	22	MOP
P0	P1	P2	21	MOP
Q0	Q1	Q2	20	MOP
N3	N4	N5	19	MOP
P3	P4	P5	18	MOP
Q3	Q4	Q5	17	MOP

Table 1-6B. MCS (Cont) – 37-Bits (CCS) (Card Location: ABBF8)

Chip Locations				
Address			Bit	
000 - 3FF	100 - 7FF	800 - FFF	Number	Micro Operator
G0	G1	G2	16	NWAY
L0	L1	L2	15	NWAY
M0	M1	M2	14	NSL
N0	N1	N2	13	NSL
P0	P1	P2	12	NSL
H0	H1	H2	11	NXT
I0	I1	I2	10	NXT
J0	J1	J2	9	NXT
K0	K1	K2	8	NXT
A0	A1	A2	7	NXT
A3	B3	C3	6	NXT
B0	B1	B2	5	NXT
C0	C1	C2	4	NXT
D0	D1	D2	3	NXT
E0	E1	E2	2	NXT
F0	F1	F2	1	NXT
H3	G3	I3	0	NXT

Table 1-7. PSI Write (PWCS) (Card Location: ABBE4)

Chip Loc	Bit Number	Micro Operator	Description
Q1	31	WCSPAR	(WCSPAR parity bit)
Q1	30	Not Used	
Q1	29	MOP-1	(write error report)
Q1	28	WBF-LD	(a WBF character will load)
P1	27	HCHRW	(half character will be written)
P1	26	HLD-EN	(enable WMI hold when char not available)
P1	25	WIDLE	(write C/S idle)
P1	24	WBSL	(write control store branch select)
N1	23	WBSL	(write control store branch select)
N1	22	MWRWBF	(MWR gets WBF)
N1	21	WRQBY1	(set Write Request F/F)
N1	20	GETCHR	(get another character)
H1	19	M21-LD	(M21 load)
H1	18	M43-LD	(M43 load)
H1	17	M65-LD	(M65 load)
H1	16	M87-LD	(M87 load)
J1	15 for 4	ASL	(align select for write)
K1	11 for 4	SCR	(select character for write)
L1	7 for 4	WNA	(write next address)
M1	3 for 4	WNA	(write next address)

Table 1-8. PSI Read (PRCS) (Card Location: ABBE6)

Chip Location	Bit Number	Micro Operator	Description
H0	31	RCSPAR	(RCS Parity Bit)
H0	30		Not Used
H0	29	LSLEFT	(LSD Digit Left)
H0	28	PREQEN	(PREQ Enable)
G0	27 for 2	LSL	(MRB L.S.L. Select)
G0	25	RIDLE	(RCS Idle)
G0	24	LSDMR2	(LSD gets MRD digit 2 instead of 1)
F0	23	DETECT	(Detect sign or hex F)
F0	22 for 2	RBSL	(Read C/S branch address select)
F0	20	MRDMTY	(MRD register empty)
E0	19	CHRAVL	(Character available)
E0	18 for 3	SDP	(Sign digit pointer for read)
D0	15 for 2	MSL	(MRB M.S.D. select)
D0	13 for 2	CSR	(CHR select for read)
C0	11 for 4	CSR	(CHR select for read)
B0	7 for 4	RNA	(RMI next address)
A0	3 for 4	RNA	(RMI next address)

Table 1-9. Fetch Look-Up Table (FLUT) (Card Location: ABBD4)

Chip Location	Bit Number	Micro Operator
E1	11	LIT-OK
E1	10 for 3	Class and Length
D1	7	FL-VAR
D1	6 for 3	SCLS
C1	3	Variant
C1	2 for 3	AST

Glossary of Terms

The following are definitions of the terms used in system maintenance logic. The logic terms are listed according to Module and String location in the B 2900 system. Each term name appears as displayed on the ODT by the System Maintenance Vehicle. A cardmap term name, card location, chip location and backplane pin (if applicable) is obtained by referencing the term in the String List.

If the term name is followed by *****, such as ABLE *****, the term represents an error condition in the B 2900.

In some definitions a special format is used to specify bit position and number of bits used by the particular term. For example, in the following table, the [3:1] indicates bit position 3 for one bit is a spare bit (not used).

[3:1] = Spare.
[2:1] = VMR not equal to AEQB output.
[1:1] = Carry out of VL2 ALU.
[0:1] = Spare.

ASAM MODULE (STRING 5)

ABLANK6
Spare.

ABLANK5
Spare.

ABLANK4
Spare.

ALENMSZ
ALEN most significant digit = ZERO (ALEN 04-07 = ZERO).

ALEN
Length Unit (PSI Character Counter).

ATOG
Start BASE/LIMIT checking Toggle.

ATOG2
Defines BASE or LIMIT check. ATOG2 = 1 check limit, ATOG2 = 0 check base.

ABLE *****
ASAM BASE/LIMIT error.

ABLANK1
Spare.

ABLR
ASAM BASE/LIMIT register.

AEQL--F
ASPR port (B) data is equal to ASPR port (A) data or ABLB port (B) data.

ALSS--F
ASPR port (B) data is less than ASPR port (A) data or ABLB port (B) data.

AGTR--F
ASPR port (B) data is greater than ASPR port (A) data or ABLB port (B) data.

NOTE
Determination of which data (ASPR port A or ABLB port B) is compared to ASPR port B data is the function of ABLS. When ABLS = 0, ASPR port A data is compared. When ABLS = 1, ABLB port B data is compared.

ALSTWDF
Last word of a transfer from MCS.

AINSTRF
Instruction ready for transfer from FETCH.

ASRAR7F
ASAM scratchpad (A) address, spare condition bit.

ASPAR6F
ASAM scratchpad (A) address, spare condition bit.

ASPAR5F
ASAM scratchpad (A) address, spare condition bit.

ASPAR3F
ASAM scratchpad (A) address, spare condition bit.

ASPAR2F
ASAM scratchpad (A) address; spare condition bit.

APROAKF
Processor acknowledge from IOT.

ASPAR8F
ASAM scratchpad (A) address, spare condition bit.

ASPAR0F
ASAM scratchpad (A) address, spare condition bit.

ASPAR1F
ASAM scratchpad (A) address, spare condition bit.

ASPAR9F
ASAM scratchpad (A) address, spare condition bit.

ASPAR4F
ASAM scratchpad (A) address, spare condition bit.

AAC2F
Address Controller = 2. (Not used, grounded.)

AAC1F
Address Controller = 1. (Not used, grounded.)

AAC0F
Address Controller = 0. (Not used, grounded.)

AWCAVLF
Write character available from MATH.

ABLANK3
Spare.

ACSLD
ASAM Control Store Address Register.

ACSR-59
Part of next address field. (Not used.)

ANXT
Next address field.

ABINBCD
Binary to BCD conversion.

ATSTTRU
Test true to MCS.

ASPAR
Scratchpad (A) address.

ASPBR
Scratchpad (B) address.

ALE
Latches data read, during the clock period in which ALE first came true, into the output latches.

NOTE
If the system is in the maintenance mode, and it is desirable to check the contents of ASPR and then resume the active program, ALE must be checked. If ALE = 0, any readout of ASPR will corrupt data in the ASAM module. Continuation of any program will then be impossible.

AALO
ASPR port (A) output forced low, when AALO = 1.

ABL-RD
ASPAR is used as the ABLB port (B) address instead of the BASE/LIMIT address.

APTA-EN
ASPR port (A) output enable. When APTA-EN = 1, the ASPR port (A) output is the data contained in the RAM at the address indicated by ASPAR. When APTA-EN = 0 and ABLS = 1, the ASPR port (A) output is in the high impedance (open collector) state.

AWE4
Scratchpad Write Enable for the left-most character.

AWE3
Scratchpad Write Enable for the third character.

AWE2
Scratchpad Write Enable for the second character.

AWE1
Scratchpad Write Enable for the right-most character.

ABLS

HIGH to enable ABLB port (B) output. LOW to enable ASPR port (A) output.

NOTE

ABLB is written on ABL-RD * ABLS/ and AWEN only. SP-READ turns ABLS off.

ABCDS

BCD adder input select.

3 = LITERAL (LITERAL DATA FROM MICROCODE).
2 = REMAINDER (to next address or BIN-BCD).
1 = QUEUE (See AQSEL).
0 = APTAD.

ABISS

Buffer input select (MUX of data into SPR).

3 = ASAM address register.
2 = Adder output.
1 = XMWB from MCS or MATH.
0 = XPTA from FETCH.

AAD-LD

Address register LOAD or COPY high order port (B) digit when writing the adder data to ASPR.

ASUB

Control Store pipeline term that places the BCD ALU into the subtract mode.

ASPA-LD

Load scratchpad (A) address register (SPAR) on the next clock.

ASPB-LD

Load scratchpad (B) address register (SPBR) on the next clock.

AASAMRY

ASAM module is ready to accept a new CMOP.

ACSR-21

Spare.

ALIT

ASAM literal field, from micro-code.

AQSEL
QUEUE bus MUX select. Select 1 of 4 characters of port (A) data to adder or XMRB.

AP-EN
PSI Enable (enable XMRB drivers).

AIOT-GO
IO Initiate is ready to be transferred to the IOT.

ADTA-RQ
Data request to MATH module.

ADTA-FN
Data transfer complete. Drives RCOMP and WCOMP.

ADTA-RY
Data ready. Drives EDP.

AWAR
Write Access ready. Valid address in AADR.

ACSR-08
Spare.

ACSEL
6-bit multiplexor select field used for micro-sequencer test condition (S0, S1) selection.

ACSR-01
Spare.

APARITY
Parity on the 60-bit AACS word that was just read.

NOTE
Control Store parity checking does not currently exist in the ASAM module. However ASAM parity checking will exist in the B 2900 at a future date.

AADR
[31:32] ASAM Memory Address Register. 32-bit register which contains Memory Cycle Address for MATH, MCS and TIMER modules.

MCS MODULE (STRING 3)

CTRACE
Causes an interrupt and BCT after the next instruction.

CPINT
Processor Interrupt.

CTMP
Length unit output to ASAM.

CINVIO *****
Invalid I/O from IOT.

CTIMOUT *****
Instruction Timeout.

CTIME
Timer Interrupt.

CRES *****
Invalid OP sets B8 of processor RD. Can be set by an undigit detected by the MATH module during arithmetic operation, or by MCS module decoding an invalid OP received from FETCH (OP contained IN COP).

CPICTUR *****
SNAP picture taken.

CEFD *****
Memory Error Report:

1 = Corrected single bit.
2 = Double bit error.
3 = Write data bus parity.
4 = Address length error (Write).
5 = Address bus parity.
6 = Address Undigit.
7 = Memory Module not present.

CFL-LD
Load OP, AF, BF from SPTA lines.

CAC-LD
Load (A) Address Controller from XPTA lines.

CBC-LD
Load (B) Address Controller from XPTA lines.

CCC-LD
Load (C) Address Controller from XPTA lines.

CLU
Length unit output to Memory Write Bus or to CTMP.

CBFL
BF of instruction being executed. Value is in Binary.

CAFL
AF of instruction being executed. Value is in Binary.

CCC
(C) Field Address Controller.

0 = UN data; 1 = SN data; 2 = UA data.

CBC
(B) Field Address Controller.

0 = UN data; 1 = SN data; 2 = UA data.

CAC
(A) Field Address Controller.

0 = UN data; 1 = SN data; 2 = UA data.

CMPE *****
Uncorrectable Memory Error.

CADER *****
Address error from FETCH or MCS.

CRES2
Spare.

CMERRPT *****
Memory Error Report written.

COVFC
Result Field ODD flag. Control signal to MATH. During FAD, high if AF is ODD, low if AF is EVEN.

CSGNINAV
Sign invert to MATH.

CINS-ER *****
Instruction error from FETCH or MCS.

CNOBIT1
Spare.

CBCTM
Not used.

CSPARE7
Spare.

CTVW-EN
Temperature and Voltage Warning Enable. (Not used)

CFETCH
MCS is idle; Ready to receive the next instruction.

CFLAG2
Miscellaneous flag.

CSPARE9
Spare.

CTINTEN
Timer Interrupt Enable.

CPROHLT
Processor Halt.

CFLAG1
Miscellaneous flag.

CFLAG3
Miscellaneous flag.

CSPARE2
Spare.

CTRAP
Hardware trap on accumulator over/underflow.

CNOR
Normal State.

CPROWT
Not used.

COVRIDE
Not used.

CSPARE1
Spare.

CUSER
Not used.

CFGO
Signal to FETCH. Fetch module may request Memory Cycle in order to fetch the next instruction.

CERPTEN
Memory Error Report Enable.

CSNPGEN
SNAP Enable.

CCOMH
Comparison HIGH indicator.

CCOML
Comparison LOW indicator.

COVF
Overflow flip/flop.

CNOBIT2
Spare.

CYCSA
MCS Control Store Address.

CASA-OP
CMOP for ASAM module. (Low active)

CEXE-OP
CMOP for MATH module. (Low active)

CLEN-OP
CMOP for LENGTH unit. (Low active)

CPSI-OP
CMOP for PSI module. (Low active)

CFLGEN1
CMOP for FLAG unit 1.

CFLGEN2
CMOP for FLAG unit 2.

CLIT
CMOP for LITERAL unit.

CNOBIT9
Spare.

CSEL-WE
Write enable for CMAP RAM.

CERDR
Execute Read Request. To PSI and MIC.

CCSWEN
Write Enable for Control Store RAM. Generated as a result of YYWEN.

CPAR
Parity bit from MCS Control Store.

CS1PL
Output of pipeline register through 2-way branching to sequencer.

CS0PL
Output of pipeline register through 2-way branching to sequencer.

CSEL
Selection of MUX for 2-way branching.

CWAIT
Sequencer wait or used for destination to flag unit.

CFE-NOT
Sequencer file control. If high, CPUP----1 not active.

CPUP
PUSH/POP control of stack.

CDEST
Destination of CMOPS.

0 = Literal or flags unit.
1 = CFLG-EN2.
2 = CFLG-EN1.
3 = PSI module.
4 = Length unit.
5 = Math module
6 = ASAM module.

CMOP
Module Operator, points selected module (decoded from CDEST) to new address in micro-code.

CNWAY
Selects input for N-WAY branching.

3 = C4WAY--2; 2 = C4WAY--1; 1 = C8WAY--1;
0 = C16WAY-1.

CNSL
Select inputs for N-WAY branching test conditions.

CNXT
Next address to sequencer.

COP
OP code being executed.

MIC MODULE (STRING 13)

MDUM1
Spare.

MRDPAR
Single clock Read Data Parity.

MRDLCH
Read Data Latched indicates RRDL has latched single clock Read Data.

MRWDL
Single clock Read Data Latch.

MDUM2
Spare.

MIRWR
IOT Read/Write data register.

MIPARR
IOT data parity.

MDUM3
Spare.

MPRWR
Processor Read/Write data register.

MPPARR
Processor data parity.

MPPRI1
Used to extend DPPRIN (the NOR of DPPRI2 and DPPRI1) which is used by ASAM to check the Base/Limit of the address bus. It indicates that the processor rather than the IOT is using the bus.

M3SEN
In single clock mode enables tri-state data bus drivers to the IOT and PSI modules.

MRD-CL
From MCS to unlatch error registers after doing processor result descriptor store. Unlatches YEFD lines to ready for next result descriptor.

MDELPRQ
Processor request delay (active low) from PSI. Used to delay memory cycles without losing priority to allow PSI to unload read buffer.

MPPRI2
Prevents IOT from taking Memory Cycle from processor before DGTO.

MEPRI
Prevents FETCH from taking Memory Cycle from Execute before DGTD. (FETCH has higher priority but Execute was first).

MCYCLE
Maintenance bit to cycle memory.

MDUM5
Spare.

MMAR
Memory Address Register. Contains binary address generated.

MRDCY
READ cycle. If not set, it is a WRITE cycle.

MPREQ2
Processor Memory Request for error register. If not set, it was an IOT request.

MBIAD
Binary Memory Address.

NOTE
MBIAD contains the last memory address that was accessed (Read or Write Memory Cycle). MMAR is actually the address that is sent to the MT cards; however, MMAR resets after the memory cycle has completed and MBIAD does not.

MUNDGT *****
Undigit found in the BCD address given to memory.

MADPNR *****
Memory Address not present.

MEVDR *****
Error vector data from ECC. MEVD[0] indicates single bit error and MEVD[6:6] indicates bit in error.

MTCER *****
Timing card error data.

0 = No error.
1 = Single bit read error.
2 = Invalid.
3 = Multiple bit read error.
4 = Address parity error.
5 = Address length error.
6 = Write data parity error.
7 = Invalid.

MDUM6
Spare.

MSCRDA
Single Clock Read Data Available. Used to form single clock version of data output control. Acts also as the CYCC of the single clock.

MSCRDAA
Single Clock Read Data Almost Available. ORed with DSCRDA to form single clock version of data output control which enables the READ data drivers and READ data latch output of the MIC. Used in conjunction with DSC only.

MDUM7
Spare.

MGTD
Request granted to requestor.

MDUM8
Spare.

MLCDIS
Latch Clear Disable. Set only by Maintenance to prevent single clock Read Data Latch from automatically clearing at end of memory cycle after data has been transferred to requestor.

MSC
Single Clock. Set only by Maintenance to put MIC into single clock mode. This allows it to latch READ data (even though the clocks are disabled) and to simulate interface signals normally coming from the MTC.

MDUM9
Spare.

MHOLD
Register hold. Set at DGTD time and puts registers into hold mode so request data is not lost.

MERS
Error Store Cycle. Set when first of two required Error Store Cycles is completed, so that second Error Store word is selected by the write data MUX.

MERI
Error Report Inhibit. Set if an error occurs to prevent an error flag to requestor more than once on the same error.

MDUM10
Spare.

MPREQ1
Processor Request for Memory Cycle.

MIREQR
IOT Request for Memory Cycle.

MOPR
Memory OP Code register.

00 = READ; 01 = WRITE; 11 = ERROR STORE.

MLTH
Length of Write in digits. Values 0 – 7 (0 = 8 digits).

FETCH MODULE (STRINGS 1,2)

FCON
: Constant Register. An 8-bit field that is under microcode control. It is loaded by CON-LC. Data is either NI increment value or pipeline register fields SPA and SPB.

FPTB
: Two high order bits from ASPR.

FFBS
: BASE/LIMIT request to ASAM.

FADR
: 28-bit Memory Address Register. Used also as accumulator for FETCH calculations. Valid only to memory when FFREQN = 0.

FODDADR *****
: ODD address, in FADR.

FINS-ER *****
: Instruction error. Flag that FETCH sets when an invalid OP or no F at beginning of BCT address.

FBZERO
: BASE Zero flag set by microcode. 0 = BASE Zero; 1 = BASE Non-Zero.

FRD
: Bit from RD/BCT PROM that latches on each clock.

FBCT
: Latched output of RD/BCT PROM.

FAFBFER *****
: AF BF error. FETCH error detection logic has found an illegal indirect field length address or an undigit in the length.

FUD-ADR *****
: Undigit in address.

FERR *****
: FETCH error. See FAFBFER or FUD-ADR.

FIXNEG
: Index register sign flag. 1 = Negative.

FAIX
(A) Address indexed. A 2-bit field indicating which index register is to be added to (A) address.

FAC
(A) Address Controller.

FAEXT
(A) Address extended bit. Loaded from AEXTD after OP has been formatted. Used with CLASS, BEXT and CEXT to determine NI increment.

FBEXT
(B) Address extended bit. Used with CLASS, AEXT and CEXT to determine NI increment.

FCEXT
(C) Address extended bit. Used with CLASS, AEXT and BEXT to determine NI increment.

FAEXTD
(A) Address extended bit. Loaded when first 2 digits of (A) address are in DB1. Indicates (A) address is 8 digits instead of 6 digits.

FBEXTD
(B) Address extended bit. Set by FETCH detection logic. Micro-code tests this bit to determine length of instruction.

FCEXTD
(C) Address extended bit. Set by FETCH detection logic. Micro-code tests this bit to determine how much information to format.

FBIX
(B) Address indexed. A 2-bit field indicating which index register is to be added to (B) address.

FBC
(B) Address Controller.

FSTOP
System is in the single instruct mode.

FALIT
(A) Literal. Flag set from decode of (A) field length. Indicates that 1 to 6 digits of (A) address space are to be used as data.

FAIF
(A) Indirect field length. Flag set by FETCH to indicate AF [5:6] is base relative address of new field length. This length may also be indirect but may not be a literal.

FBIF

(B) Indirect field length. Flag set by FETCH to indicate BF [6:7] is base relative address of new field length. This length may also be indirect but may not be a literal.

FCIX

(C) Address indexed. A 2-bit field indicating which index register is to be added to (C) address.

FCC

(C) Address Controller.

FFREQN

FETCH Memory Request Bit (active low).

FCT

FETCH Memory Cycle counter.

FRESR

Register that captures MOD[2:2] at RES-LD time. Used in FETCH memory cycle circuits.

FASTO

Register that latches FAST at FTOE time.

1 = Store (A) address in memory.
2 = Store (B) address in memory.
3 = Store (A) and (B) addresses in memory.
4 = Store (C) address in memory.
5 = Instructions cannot be overlapped.
6 = Store (B) and (C) addresses in memory.
7 = Store (A),(B) and (C) addresses in memory.

FLIT-OK

Bit from LOOK-UP table that indicates this instruction may contain literal data in 6-digit (A) Address location instead of an address.

FCLS

CLASS and LENGTH of instruction.

7 = Invalid instruction or F.
6 = Invalid instruction.
5 = 24 digit OPS.
4 = 16 digit OPS.
3 = 12 digit OPS.
2 = 8 digit OPS.
1 = 6 digit OPS.
0 = OP 84 with 2 variants.

FFL-VAR
Used to disable error detection in BF.

FSCLS
3 bits from FETCH look-up table that further characterizes instruction being fetched. Used for branches, privileged instructions and class 1 instructions.

NOTE
FSCLS equal to 6 or 7 is an Invalid OP Decode.

FVARANT
Bit from FETCH look-up table used for searches. Indicates (B) controller is not really an address controller. Also has a few miscellaneous uses.

FAST
3-bit field whose value is stored in FETCH look-up table.

FDB4
Final data buffer before scratchpad. Output goes into formatter MUX and rearranges data into correct order.

FDB3
The third of four data buffers. Goes into formatter MUX at positions 2 and 3.

FDB2
The second of four data buffers. These buffers are required to delay data long enough to allow microcode to set flags from data, to test flags and to format data in scratchpad.

FDB1
Data buffer 1 accepts memory data from PSI. Buffer loads with pipeline register and holds when pipeline register holds. FDB1 is data input to FETCH detection logic and also addresses FLUT. Output data goes to DB2.

FNXT
Address of next microcode instruction.

FUSL
2-bits in pipeline register that control MUX inside sequencer.

FRE-NOT
Register Enable Not. Used in conjunction with PUP to control sequencer stack.

FFE-NOT
File Enable to sequencer.

FPUP
Pipeline register bit that controls sequencer stack. Used with RE-NOT.

FSPWE
8-bit field in pipeline register that controls scratchpad Write Enables. (active low)

F8WAY
Pipeline register bit that enables three of the 8-way test multiplexers.

F4WAY0
Pipeline register bit that enables two of the 4-way test multiplexers.

F4WAY1
Pipeline register bit that enables two of the 4-way test multiplexers.

F2WAY0
Pipeline register bit that enables one of the 2-way test multiplexers.

FTSEL
3-bit field in pipeline register used to select bits to be tested by NWAY branch circuits.

F2WAY1
Pipeline register bit that enables one of the 2-way test multiplexers.

F2WAY2
Pipeline register bit that enables one of the 2-way test multiplexers.

F2WAY3
Pipeline register bit that enables one of the 2-way test multiplexers.

FFMC-LD
Used to load FCT.

FSPA
4-bit pipeline register field. (A) address of scratchpad. Used also to set/reset flags and load values into FCON.

FSPB
4-bit pipeline register field. (B) address of scratchpad. Used also to load values into FCON.

FSHIFT
Pipeline register bit indicating last of valid read data is in DB1.

FPAR
FETCH Control Store parity bit.

FFMT-EN
Format Enable Bit. 1 = Format mode; 0 = calculate mode. Indicates also when pipeline and DB registers may be loaded.

FFMB
15-bit format bit field.

FDUMYCS
Spare.

FYCS
10-bit field that follows sequencer output when machine is running. Used to address Control Store array when maintenance is loading Control Store.

*IOT/SER MODULES (Strings 8,9,10,11,12)

IIAYFZ
IIO ALU output = Zero (Test condition flip/flop).

IIOT-GO
IOT start up signal from ASAM.

IIALU
IIO ALU output register (Maintenance only).

IIM3
IIO sequencer status bit. Set when branch instruction.

IIHANG *****
IIO distress flip/flop. Set when OP from ASAM = 92 or 94. Stops systems clocks. Bad OP is in IIAY locations 1 through 8.

IICPAR *****
IIO Control Store parity error flip/flop. Stops system clock.

IILOOP
IIO maintenance feature to loop on a set of instructions.

IIM2
8-bit field for IIO sequencer address of last branch. IF IICPAR= 1, then it will be the address of the parity error.

IIM1
IIO Control Store Address being executed.

IIPAR
IIO Control Store parity bit.

IICR-54
IIO spare Control Store bit.

IMOP
8-bit IIO MICRO-OP field, timeshared 4 ways.

IMSL
2-bit IIO MICRO-OP selection, to select one of four destinations for IMOP.

ICLT
8-bit IIO ALU literal.

IIIN
9-bit IIO ALU instruction.

IIAA
4-bit IIO ALU (A) address (source).

IIBA
4-bit IIO ALU (B) address (destination).

ICSN
8-bit IIO sequencer direct input (usually a branch address).

IIRM-FE
IIO sequencer FILE ENABLE bit.

IIRM-PP
IIO sequencer PUSH-POP bit.

ITASL
3-bit IIO test condition MUX selection group (A).

ITBSL
3-bit IIO test condition MUX selection group (B).

ITSI
3-bit IIO sequencer S0, S1 mode line source selection.

ISP3-03
Service module last sequencer instruction was a branch.

ISHANG *****
Service module distress flip/flop. Set for various reasons. ISM2 gives Control Store Address at which a problem is defined in the microcode listing. Stops system clocks.

ISCPAR *****
Service module Control Store parity flip/flop. When set, stops system clocks.

ISLOOP
Service module maintenance feature to loop on selected set of control store addresses.

ISM2
10-bit SER module register which contains last branch address taken by the sequencers.

ISM1
10-bit SER module register which contains current Control Store Address being executed.

ISPAR
Service module Control Store parity bit.

ISCR
2-bit Service module spare Control Store bits.

ISXSL
Service module test condition extension field.

IUOP
12-bit Service module MICRO-OP field timeshared 4 ways.

IUSL
2-bit Service module MICRO-OP selection field. Selects destination for IUOP.

ISASL
3-bit Service module test condition selection group (A).

ISBSL
3-bit Service module test condition selection group (B).

ISSN
10-bit Service module sequencer direct input. Usually used for a branch instruction.

ISRM-FE
Service module sequencer FILE ENABLE.

ISRM-PP
Service module sequencer PUSH-POP input.

ITSS
3-bit Service module sequencer S0, S1 MUX selection.

ISDI
16-bit Service module ALU direct input (literal field).

ISIN
9-bit Service module ALU instruction.

ISAA
4-bit Service module ALU (A) address (source).

ISBA
3-bit Service module ALU (B) address (destination).

IWDR
16-bit IOT Write Data Register. Used to hold or assemble least significant 16-bits of data before they go to memory. If backward (IBKF) is set, it will hold most significant 16-bits of memory data.

ILCR
2-bit DLP Strobe/Memory Sync Counter. Incremented twice after a memory read, or once after each DLP strobe. Used to keep track of number of MLI words sent or received with respect to memory reads or writes. It is used to adjust data transfer and Memory Address Register (IADR) after burst mode.

IBWA
3-bit Buffer (IBUF) Write Address. Buffer only has four MLI words of data, and therefore only needs 2-bits for an address. The top bit of address is used to detect when an address has "wrapped around' and started at zero again.

IBRA
3-bit Buffer (IBUF) Read Access. Buffer is 16X4 RAM used to buffer memory data to DLP (write). Only one address is used for read and is concatenated with IWCR to form a 32-bit register.

IYTRM
Maintenance flip/flop which follows non-shiftable TERMINATE signal, which must be maintained steady when IOT is shifting or in maintenance mode.

IYSIO

Maintenance flip/flop which follows non-shiftable HOST STROBE flip/flop. Reports status of MLI HOST STROBE signal which must be maintained steady when IOT is shifting or in maintenance mode.

IYSND

Maintenance flip/flop which follows non-shiftable SEND flip/flop. See IYTRM.

IYADSL

Maintenance flip/flop which follows non-shiftable ADDRESS SELECT flip/flop. See IYTRM.

IYBAS

2-bit Maintenance flip/flop that follows non-shiftable IBAS flip/flop. See IYTRM.

IYWENMW

Maintenance Write Enable for Memory Data Driver/Receivers.

IIWABT

Memory Write Abort flip/flop. Used to cancel memory operations when an I/O with no data transfer is requested.

IBKF

Backward flip/flop.

1. Swaps IWDR from least to most significant bits and IBUF from most to least significant bits of memory data.
2. Changes sign on all Memory Address Register count operations.

ITRF

Translate flip/flop. Removes tri-state buffer from data path and inserts tri-state translate PROM. Send or Receive.

IZZA

Spare.

IYMRWP

Maintenance Memory Read/Write parity bit.

IBUR

Burst flip/flop on 7 MHZ clock. Sets IBURSTF for burst flip/flop on 3.5 MHZ clock.

ILPE *****

MLI parity error flip/flop. Only used on receive data and is enabled whenever IOT strobe is sent to acknowledge data.

IBURSTF

Burst flip/flop used by sequencer chips on 3.5 MHZ clock.

ITIMOUT *****

DLP Timeout flip/flop. Indicates at least 1 MS has passed since last DLP strobe.

ICLTEQ2

DLP Strobe/Memory Sync counter = 2. Indicates a Memory Read has just finished or a Memory Write is required.

ILCPSTF

DLP Strobe flip/flop on 3.5 MHZ clock for sequencer tests.

ITMO

Bit 0 (LSB) = ILCPSTF1, DLP strobe sync flip/flop which is fed by R-S mode DLP strobe catcher.
Bit 1 = ITM1MS. Set when second 1 MS pulse occurs after a DLP strobe.
Bit 2 = ITMOUT. Sets when first 1 MS pulse occurs after a DLP strobe. Next 3.5 MHZ clock sets ITIMOUT.
Bit 3 = (MSB) Miscellaneous use.

ILWBPR

Send MLI data parity bit (MLI write bus).

ILWB

16-bit Send MLI data (MLI write bus).

ILST

4-bit DLP Status Register. Contains status of connected DLP, if there is one.

ISTPR

8-bit DLP Request Register. Contains emergency or interrupt requests from any possible base and is used to generate base request priority. Values in it are only valid before a base is allowed to connect as the request lines are later used for DLP status.

ISAYFZ

Service module ALU output = Zero (test condition flip/flop).

ISAY15F

Service module ALU most significant bit flip/flop.

ISALU
Service module ALU output register (maintenance only).

ISFT
Service module priority flip/flop.

IIFT
IIO module priority flip/flop.

INAE
9-bit next available entry counter (Job queue address).

INFE
9-bit next finish entry counter (Not used).

INSE
9-bit next start entry counter. Contains either job queue address of last I/O or job queue address of next I/O for service module to start.

NOTE
If ANAE = INSE, last IIO has been started before service module. IF ANAE > INSE, I/O has been in IIO module but service module has not yet done anything about it.

IIR2
8-bit IIO ALU pipeline/assembly register. Upper half. Controls also 4 to 1 memory data ALU input MUX.

IIR1
8-bit IIO ALU pipeline/assembly register, lower half.

ISPA
9-bit scratchpad memory address register. Format BBDDDXXXX where BB is BASE number; DDD is DLP number and XXXX is address of entry for BASE/DLP.

IBSY-ME
Channel busy ARAM (IBSY) maintenance Write Enable.

ISPD-ME
Scratchpad maintenance Write Enable.

IJBQ-ME
Job queue maintenance Write Enable.

IXDA-EQ
Maintenance stop condition compare flip/flop.

IEAD
28-Memory End Register.

IADR
28-Memory Address Register.

IIMOP00
Memory operation. 0 = Read, 1 = Write.

IILTH
3-bit memory length for Write OP. Only two most significant bits used as IOT does not write digits, only bytes.

ISKIP
Source of IOT clock enable signal to clock card. Normally toggles every 7 MHZ clock which causes clock card to skip every other clock to some IOT circuits. Signal to clock card is IHTG and whenever an IOT YSEL is true, IHTG is forced true and whole IOT runs at 7 MHZ.

NOTE
Regular 7 MHZ clocks are called TSCK-XX. Those controlled by IHTG are called TECK-XX.

IMAR
Memory Request.

IMPE
Memory parity even flip/flop. Set when even numbers of bits are sent on memory data interface. It is cleared by MICRO-OP ICYCRS after being tested.

IMER *****
Memory Error flip/flop. Set by memory whenever there is an irrecoverable memory error.

IPRLS
Present address less than end address flip/flop.

8 = IBUF-ME – Buffer maintenance enable.
4 = IALU-ME – Service module ALU enable.
2 = IINT – OP complete interrupt catcher for S1 mode.
1 = PRLS – Present address < end address flip/flop.

NOTE
3 high order bits were originally spare and are not reflected in screen name.

IIOTCY
IOT Memory Cycle flip/flop set when IOT has been granted memory address bus.

IPATH
IOT path test flip/flop set during path tests for two reasons:

1. Causes IHTG to be true so the IOT will run at 7 MHZ.
2. Causes some MLI signals normally sourced by a BASE to be sourced by IOT for interface testing.

IMEMRD
Memory Read Data available test condition flip/flop. Set by an OR of RDA0 and RDA1 which are on 7 MHZ clock and are not suitable for 3.5 MHZ sequencer.

IGRNTD
Memory Granted test condition flip/flop. Set by an OR of GTD0 and GTD1 which are on 7 MHZ clock and are not suitable for 3.5 MHZ sequencer.

IRDA1
Memory Read Data available sync flip/flop set by IRDA0.

IRDA0
Memory Read Data available sync flip/flop set by memory.

IGTD1
Memory Granted sync flip/flop set by IGTD0.

IGTD0
Memory Granted sync flip/flop set by memory.

PSI MODULE (STRING 4)

PWRQ
PSI Write Request to memory.

PRPE *****
PSI Read Control Store parity error.

PEDC
MATH data cycle.

PFRI
FETCH Read initiated in PSI.

PERI
MATH Read initiated in PSI.

PRPN
Read operation pending.

PFDA
FETCH data available in PMRD.

PEDA
MATH data available in PMRD.

PEGF
MATH granted flip/flop to ASAM.

PFGF
FETCH granted flip/flop to FETCH.

PDIS
Allows PEGF to disable PREQ.

PRDC
Read Data complete. Latches PRDA from MIC. To MCS.

PRIN
Processor Read initiated – Waiting for RDA to continue.

PLST
Last character. From read data available.

PFUL
PMRD valid data being put on XMRB.

PCNZ
Characters on XMRB not all Zeroes. To MCS and MATH.

PLNZ
LSD on MRB not all Zeroes. To MCS and MATH.

PNEG
Read Data has negative sign. To MCS and MATH.

PECY
MATH Memory Cycle. To MCS.

PNGT
Access not granted.

PMRD
PSI Memory Read Data Register. From memory.

PLSD
Store least significant digit. From PMRD.

PRAS
PSI Read Control Store Address on parity error.

PRAD
PSI Read Control Store Address. From MCS or FETCH.

PRCSPAR
Read Control Store parity bit.

PRSPR
Spare.

PLSLEFT
Least significant digit remaining on read.

PPREQEN
Processor Request Enable. Generates PDIS.

PLSL
PSI read PLSD mux select.

PRIDLE
Read Control Store to idle.

PLSDMR2
PLSD MUX gets PMRD digit 2 instead of 1.

PDETECT
Enables detection for (C),(D) or (F). Sets PNEG or PFDG.

PRBSL
Read Control Store branch address select.

PMRDMTY
Read Data Register (PMRD) empty.

PCHRAVL
Character available on XMRB. (PFUL must be true before PEDP or PFDP are active.

PSDP
Sign digit pointer for read.

PMSL
PSI Read MSD MUX selection.

PCSR
PSI Read character select for PFRB mux.

PRNA
PSI Read Control Store next address.

PWAD
PSI Write Control Store Address from MCS.

PWCSPAR
Write Control Store parity bit.

PWSPR
Spare.

PERRPT
Write Memory Error Report.

PWBF-LD
WCS bit to load either PM87, PM65, PM43 or PM21.

PHCHRW
1/2 character to be written. Signals ASAM to downcount ALEN by 1 instead of 2.

PHLD-EN
Enable PWNI hold when character not available.

PWIDDLE
PSI Write Control Store to idle.

PWBSL
Write Control Store next address MUX select.

PWWRWBF
PMMR load from PWBF.

PWRQBY1
Write Request.

PGETCHR
Get another character from XMWB.

PM21-LD
PSI Write Memory Register load digits 2,1.

PM43-LD
PSI Write Memory Register load digits 4,3.

PM65-LD
PSI Write Memory Register load digits 6,5.

PM87-LD
PSI Write Memory Register load digits 8, 7.

PASL
Write PINP MUX select.

PSCR
Write character Most/Least (PCHM, PCHL) MUX select.

PWNNA
Write Control Store next address.

PLWW
Last word write for current PSI write operation.

PWPE *****
PSI Write Control Store parity error.

PWIN
Write OP initiated.

PWCP
Write OP complete.

PLBR
Length register busy with read operation.

PWCA
Write character available in PWBR.

PFDG
Undigit (F) detected in sign position.

PNGS
PSI Write negative sign to MATH.

PMRDPAR
Parity bit on PMRD

PLBF
Length buffer, temporarily stores number of valid digits in PMWR, PM21.

PLTH
PSI length unit. Number of bytes to be written.

PRBS
Monitors Processor Memory Read Bus (XMWB). (All modules)

PWAS
PSI Write Control Store Address for parity error.

PMSB
Stores most significant bits from XMWB.

PMWR
PSI Write Memory Data digits 8,7,6,5,4,3.

PLSB
Stores least significant bits from SMWB.

PM87
Write digits 8,7 to PMWR.

PM65
Write digits 6,5 to PMWR.

PM43
Write digits 4,3 to PMWR.

PM21
Write digits 2,1 to XPRWD.

PWBR
Data input to PSI for writing to memory.

MATH MODULE (STRING 6,7)

QWABTF *****
Write Abort to PSI memory.

NOTE
Currently set when overflow condition occurs on Arithmetic OP will set in future (LIN) when undigit is detected by Math Module during execution of Arithmetic OP Codes.

QWNEG
Negative output data.

QOVFS
Overflow.

QEXE-BN
MATH busy.

QCOM
The MATH module COM scratchpad. Loaded into CCOMH and CCOML when complete. Low active in MATH, High active in MCS.

QCOM-SI
When true, causes QCCM to be loaded into CCOMH and CCOML inverted.

QCMB--O
Sending characters during ADD. (Low active)

[0:1] = VARZ		Variant register is zero.
[1:1] = VC7		Carry out of BCD adder.
[2:1] = BMT-EQ		BQP = BQW.
[3:1] = VMX1C		Enable VL4 Multiplexer.
[4:1] = CMT-EQ		CQW = CQP.
[5:1] = VMX2C		Enable VL3 Multiplexer.
[6:1] = AMT-EQ		AQW = AQP.
[7:1] = CQZ-EQ		CQP pointer is between 4''00' and 4''77'.
[8:1] = VCO8		Wait for characters before setting write character available.
[9:1] = WCOMP		Write complete.
[10:1] = RCOMP		Read complete input interface.
[11:1] = PNF		ADD = have accumulated zero's. DIVIDE = Complementing or not.
[12:1] = BQZ-EQ		BQP is between 4''00' and 4''77'.
[13:1] = VCO13		Align of floating point output is required.
[14:1] = EDP		Input interface.
[15:1] = VCO15		Sending floating point exponent.
[16:1] = SUB		Doing subtract with BCD adder.
[17:1] = BSG		Subtract larger from smaller with BCD adder.
[18:1] = SPR [1:1]		VL4 > VL3 output from BIN ALU delayed 2.
[19:1] = AIN		Accumulator enable.

QSGN
4-bit sign register. Incoming signs right shift through register.

QACA
Accumulator RAM read/write address register (four bits).

QAL6
Digital alignment register for inserting an odd number of Zeroes into an operand while shifting during FAD.

QMRR
Register in MATH (8 bits) which reads data from PSI off the Memory Read Bus. The output of this register feeds AQ,BQ,CQ.

QVBR
QVAR one clock later decomplemented if QVAR was complemented.

QVAR
Output data of BCD ALU.

QMWBX
Data being driven towards Memory Write Bus through multiplexors.

QDVR
Quotient Prediction Data Register.

QRMR
Quotient Prediction Data Register.

QSPZ
3-bit field; The 2 most significant bits not used. Least significant bit indicates odd value in QMWBX register.

QSPR

[3:1] = Spare.
[2:1] = VMR[17:1] not equal to AEQB output of VL2 Binary ALU.
[1:1] = Carry out of VL2 binary ALU.
[0:1] = Shifting odd number of digits during FAD.

QMPR
Multiplier Partial Result Storage Register.

QCVR
Output data register for number convert hardware.

QCQW
Pointer to address in queue where QCQI on the next clock will be written at 1/2 clock time.

QBQW
Pointer to address in queue where QBQI on the next clock will be written at 1/2 clock time.

QAQW
Pointer to address in queue where QAQI on the next clock will be written at 1/2 clock time.

QAQP
Pointer to address of queue (A) from which data will be read into QAQO.

QBQP
Pointer to address of queue (B) from which data will be read into QBQO.

QCQP
Pointer to address of queue (C) from which data will be read into QCQO. in the following table, the [3:1] indicates bit position three one bit is a spare bit (not used).

QAQI
Data input to queue A.

QBQI
Data input to queue B.

QCQI
Data input to queue C.

QAQA
QAQW on previous clock. Pointer to address in queue where QAQI will be written.

QBQA
QBQW on previous clock. Pointer to address in queue where QBQI will be written.

QCQA
QCQW on previous clock. Pointer to address in queue where QCQI will be written.

QAQO
Data output from queue A.

QBQO
Data output from queue B.

QCQO
Data output from queue C.

QAMT
Not AQP.

QMT1
Spare.

QMT3
Spare.

QMT5
Spare.

QMAR
Contains the address +1 of present control store address in the pipeline register, if not in conditional branch mode. When in conditional branch mode, contains address +1 of address being branched to. Contains address +1 if CMOP is loaded (CEXE-OP1 is active low).

QVMR674

QVMR674 – [67:1] = Parity bit (ODD).
[66:3] = Spare.

QVMR634

QVMR[63:4] – [63:1] = Most significant bit of branch condition.
[62:2] = Most significant 2-bits of branch address.
[60:1] = Clear all registers.

QBQP-MO
QVMR[59:1] = Count BQP in up (00) direction.

QVMR583

QVMR[58:3] – [58:1] = OUT1-EN = Unconditional enable of VL4 MUX.
[57:2] = Select AQ,ABQ, or CQ through VL4 MUX.

QVMR554

QVMR[55:4] – [55:1] = CAVL = Enable counting without EDP or CHREQ
[54:1] = OUT2-EN = Unconditional enable of VL3 MUX.
[53:2] = Select AQ, BQ, or CQ through VL3 MUX.

QVMR511
QVMR[51:1] = WCAVL = Force WCAVL to PSI.

QVMRC02
QAVMR[50:1] = Select line for MWBY MUX.

QVMR492
QVMR[49:2] = Select lines for MWBY MUX,

QVMR478
QVMR[47:8 = Least significant 8 bits of branch address.

QVMR391
QVMR[39:1] = Count CQP in up (00) direction.

QVMR383

QVMR[38:3] – [38:1] = Set CQP to FF.
[37:1] = Set BQP to FF.
[36:2] = Set AQP to FF.

QAIN
QVMR[39:1] = Toggles accumulator on or off.

QVMR343

QVMR[34:3] – [34:1] = Set CQW to FF.
[33:1] = Set BQW to FF.
[32:2] = Set AQW to FF.

QVMR314

QVMR[31:4] – [31:1] = Carry in for binary ALU.
[30:1] = Count CQP or set COML.
[29:1] = Count BQP or set sign Neg. (active low).
[28:1] = Count AQP or set sign Plus (active low).

QVMR274

QVMR[27:4] – [27:1] = Binary ALU in binary mode and disable accumulator from writing.
[26:1] = Set overflow (low active) or count CQW.
[25:1] = Set COM strobe (low active) or count BQW.
[24:1] = Set COMH (low active) or count AQW.

QVMR234

QVMR[23:4] – Select binary ALU or accumulator operation.

QVMR198

QVMR[19:8] – [19:1] = Enable wrap around from MWBX to CQI.
Also = read from MWB.
Also = lock variant data.
Also = some miscellaneous uses.
[18:1] = Subtract mode for BCD ALU.
[17:1] = Carry in for BCD ALU.
[16:1] = Translate control on incoming BQP.
[15:1] = Put incoming BQP in 00 direction.
[14:1] = Count CQW in up (00) direction.
[12:1] = Count CQW in up (00) direction.

QVMR114

QVMR[11:4] = Least significant four bits of branch condition.

QVMR074

QVMR[7:4] = Hard logic operator (ADD, MUL.,SET1, etc.)

QVMR033

QVMR[3:3] – [3:1] = Load AQ from MRR.
[2:1] = Load BQ from MRR.
[1:1] = Load CQ from MRR.

QAQP-M0

Pointer to address of queue (A) from which the data will be read into QAQO.

TIMER MODULE (STRING 0)

TTINT
8 = instruction timeout to MCS.
4 = timer interrupt to MCS and FETCH.

TTSEQ
Timer sequencing.

TASY
Inhibit timer interrupt.

TV3
8-bit field from TV2.

TV2
8-bit field from TV1.

TV1
8-bit field from XMRB.

TT05-0
Comparator input. From TT03-0. To XMWB Bit 0 (LSB).

TT03-0
Comparator input. From TT01-0.

TT01-0
Comparator input. From TT05-0 or TT05-1.

TT05-1
Comparator input. From TT03-1. To XMWB Bit 1.

TT03-1
Comparator input. From TT01-1.

TT01-1
Comparator input. From TT05-1 or TT05-2.

TT05-2
Comparator input. From TT03-2. To XMWB Bit 2.

TT03-2
Comparator input. From TT01-2.

TT01-2
Comparator input. From TT05-2 or TT05-3.

TT05-3
Comparator input. From TT03-3. To XMWB Bit 3.

TT03-3
Comparator input. From TT01-3.

TT01-3
Comparator input. From TT05-3 or TT06-0.

TT06-0
Comparator input. From TT04-0. To XMWB Bit 4.

TT04-0
Comparator input. From TT02-0.

TT02-0
Comparator input. From TT06-0 or TT06-1.

TT06-1
Comparator input. From TT04-1. To XMWB Bit 5.

TT04-1
Comparator input. From TT02-1.

TT02-1
Comparator input. From TT06-1 or TT06-2.

TT06-2
Comparator input. From TT04-2. To XMWB Bit 6.

TT04-2
Comparator input. From TT02-2.

TT02-2
Comparator input. From TT06-2 or TT06-3.

TT06-3
Comparator input. From TT04-3. To XMWB Bit 7 (MSB).

TT04-3
Comparator input. From TT02-3.

TT02-3
Comparator input. From TT06-3 or YYDIN.

MAINTENANCE STRING (STRING 14)

YIM4
IIM4 – Data register for setting up IOT stop conditions.

NOTE
Use pin 043 card ABBC8 as sync point.

YNOUSE1
Spare.

YXSL
ISSL – Used with IXIT to select stop conditions.

YTOINH
ITOINH – Timeout Inhibit in IOT.

YXIT
IXIT – Used with IXSL to select stop conditions.

YOP
FYOP – OP equal stop logic in FETCH.

NOTE
Use pin 012 card ABBD4 as sync point.

YSAD
FYSAD – FETCH Control Store (FYCS) Address stop logic.

NOTE
Use pin 166 card ABBD0 as sync point.

YBAS
DBAS – Used with YLIM to set up either BASE/LIMIT or address to COMPARE.

NOTE
Sync points are pin 036 for Address Check and pin 109 for BASE/LIMIT Check on card ABBE2.

YLIM
DYIM – Used with YBAS to set up either BASE/LIMIT or address to COMPARE.

YBLC
DBLC/ – Used to select either BASE/LIMIT or address checking.

YRA
PYRA – PSI Read Control Store Address (Pras) stop logic.

NOTE
Use pin 108 card ABBE6 as sync point. (low active)

YWA
PYWA – PSI Write Control Store Address (PWAS) stop logic.

NOTE
Use pin 171 card ABBE4 as sync point. (low active)

YNOUSE2
Spare.

YAMCOCO
AMACD – ASAM Control Store Address (ACSLD) stop logic.

NOTE
Use pin 146 card ABBF0 as sync point.

YNOUSE3
Spare.

YCYMAD
CYMAD – MCS Control Store Address (CYCAS) stop logic.

NOTE
Use pin 085 card ABBF8 as sync point.

YCMP
PYWA – MATH Control Store Address (QMAR) stop logic.

NOTE
Use pin 059 card ABBG6 as sync point.

YCENN
YCENN – Clock Enables to string. (low active)

YSPODIS
YSPODIS – Allows console to switch from UNILINE to CONSOLE on errors.

YINH
YINH – Inhibit parity errors. (Status 6 and 7).

YINS
YINS – Inhibit SNAP and PROHLT. (Status B and C).

YSS
YSS – System STOP clock control.

0 = Do not stop any clock.
1 = Invalid.
2 = Stop Processor clock only.
3 = Stop IOT and Processor clocks.

YSOFTEN
YSOFTEN – Stop Processor on SI-FLG when soft condition met.

YGSPW
YGSRW – Enable stop on PSI Write Control Store Address.

YGSPR
YGSPR – Enable stop on PSI Read Control Store Address.

YGS7
YGS7 – Group select for maintenance panel stop logic, Row 7.

YGS6
YGS6 – Group select for maintenance panel stop logic, Row 6.

YGS5
YGS5 – Group select for maintenance panel stop logic, Row 5.

YGS4
YGS4 – Group select for maintenance panel stop logic, Row 4.

YGS3
YGS3 – Group select for maintenance panel stop logic, Row 3.

YGS2
YGS2 – Group select for maintenance panel stop logic, Row 2.

YGS1
YGS1 – Group select for maintenance panel stop logic, Row 1.

YGS0
YGS0 – Group select for maintenance panel stop logic, Row 0.

YCPE *****
YCPE – MCS Control Store parity error bit. Will give error halt (P) or (S).

YFPE *****
YFPE – FETCH Control Store parity error bit. Will give error halt (P) or (S).

YPPE *****
YPPE – PSI Read or Write Control Store parity error bit. Will give error (P) or (S).

YQPE *****

YQPE – MATH Control Store parity error bit. Will give error halt (P) or (S).

YIOTER *****

YIOT-ER – IIO or SER Control Store parity error, IIHANG or ISHANG. Will give ERROR HALT (P) or (S).

YLTCH

Spare.

YSTSFT

YSTSFT – SOFT HALT condition met. Stop on SI-FLG.

ASAM SCRATCHPAD (ASPR) DEFINES

The ASAM scratchpad is a 16X32 dual port RAM used to store the OP, AF, and BF addresses, and address information for SNAP and Memory Error Reporting.

NOTE

ALE must be checked before resuming the active program when the system is in the maintenance mode and the contents of ASPR is to be checked. If ALE = 0, any readout of ASPR will corrupt data in the ASAM Module. Continuation of any program is then impossible.

Table 1-10. ASPR Defines

Address	Definition
0	
1	R1 SNAP
2	R4 memory error report
3	
4	
5	
6	
7	
8	Working register
9	XXOPAFBF
A	A address
B	B address
C	C address
D	Working register
E	Working register
F	Working register

ASAM BASE AND LIMIT BUFFER (ABLB) DEFINES

The ASAM Base and Limit Buffer is a 16X24 dual port RAM used to store the program base and limit, and the machine base and limit.

Table 1-11. ABLB Defines

Address	Definition
0	Program limit
1	Program base
6	Machine limit
7	Machine base

FETCH SCRATCHPAD (FSPR) DEFINES

The FETCH scratchpad is 16X32 dual port RAM used to store the OP code AF BF, addresses, and other information pertinent to the FETCH module for use in address resolution.

Table 1-12. FSPR Defines

Address	Definition
0	OP OP AF BF
1	IX Address: Base + 32 for IX check
2	A Address
3	B Address
4	C Address
5	Next instruction address
6	Next next instruction address
7	AF indirect field length address
8	BF indirect field length address
9	Branch address
A	Last storing address for IX check
B	Saves OP AF BF during processor RD
C	Index register
D	Zero
E	Base
F	Stash address for processor RD

IIO MODULE ALU (IIAY) DEFINES

The IIO ALU is an ALU with a 8X16 internal RAM used to store converted descriptors, addresses and other information pertinent to the IIO module.

Table 1-13. IIAY Defines

Address	Definition
0	Working register
1	OP
2	AF
3	BF
4	C, A7
5	A6, A5 (Initiate address or
6	A4, A3 memory address)
7	A2, A1
8	Memory byte counter
9	0 0 0 B2 B1 L4 L2 L1 ***
A	C, B7 (or DLP OP)
B	B6, B5 (or BL)
C	B4, B3 (or LL)
D	B2, B1
E	B2 B1 L4 L2 L1 0 0 0 ***
F	Working Register

*** B2 B1 is binary representation of DLP base number.

*** L4 L2 L1 is binary representation of DLP number.

IOT SCRATCHPAD (ISPD) DEFINES

The IOT scratchpad is a 16x512 RAM used by both the IIO module and the SER module. Each DLP channel has 16 memory locations used to store the DLP OP and addresses,etc. The following table details the memory allocation of the scratchpad.

Table 1-14. ISPD Defines

Addresses	DLP Channel Number
00-0F	DLP 00
10-1F	DLP 01
20-2F	DLP 02
30-3F	DLP 03
40-4F	DLP 04
50-5F	DLP 05
60-6F	DLP 06
70-7F	DLP 07
80-8F	DLP 10
90-9F	DLP 11
A0-AF	DLP 12
B0-BF	DLP 13
C0-CF	DLP 14
D0-DF	DLP 15
E0-EF	DLP 16

Table 1-14. ISPD Defines (Cont)

Addresses	DLP Channel Number
F0-FF	DLP 17
100-10F	DLP 20
110-11F	DLP 21
120-12F	DLP 22
130-13F	DLP 23
140-14F	DLP 24
150-15F	DLP 25
160-16F	DLP 26
170-17F	DLP 27
180-18F	DLP 30
190-19F	DLP 31
1A0-1AF	DLP 32
1B0-1BF	DLP 33
1C0-1CF	DLP 34
1D0-1DF	DLP 35
1E0-1EF	DLP 36
1F0-1FF	DLP 37

Each DLP channel has 16 memory locations as detailed in the following table. XX = address digits decoded from DLP channel number (see table 1-14 above).

Table 1-15. DLP Channel Memory Locations

Address	Definition
XX0	OP,V1,V2,V3 DLP OP and variants
XX1	Temporary RD-IOT portion
XX2	Descriptor Link 1 and DLP address word
XX3	Descriptor Link 2 and job number
XX4	Extended RD Word 1
XX5	Extended RD Word 2
XX6	0 A7 A6 A5 (A address)
XX7	A4 A3 A2 A1 (A address)
XX8	0 B7 B6 B5 (B address)
XX9	B4 B3 B2 B1 (B address)
XXA	C6 C5 C4 C3 (C address or modified
XXB	C2 C1 0 0 end address)
XXC	0 A7 A6 A5 (Memory address of last initiate instruction is
XXD	A4 A3 A2 A1 overwritten and becomes a working register.)
XXE	Special instruction flag
XXF	Special instruction data

IOT CHANNEL BUSY RAM (IBSY) DEFINES

The IOT Channel Busy RAM is a 4X256 RAM (only 32 locations used) used to indicate that a DLP channel is busy. Each DLP channel has one memory location and the 8-bit is set to indicate channel busy. The 8-bit is cleared when the SER module passes the result descriptor to the IIO module. The following table details the memory allocation to DLP channel numbers.

Table 1-16. IBSY Defines

Address	Channel Number
00	DLP 00
01	DLP 01
02	DLP 02
03	DLP 03
04	DLP 04
05	DLP 05
06	DLP 06
07	DLP 07
08	DLP 10
09	DLP 11
0A	DLP 12
0B	DLP 13
0C	DLP 14
0D	DLP 15
0E	DLP 16
0F	DLP 17
10	DLP 20
11	DLP 21
12	DLP 22
13	DLP 23
14	DLP 24
15	DLP 25
16	DLP 26
17	DLP 27
18	DLP 30
19	DLP 31
1A	DLP 32
1B	DLP 33
1C	DLP 34
1D	DLP 35
1E	DLP 36
1F	DLP 37

SER MODULE ALU (ISAY) DEFINES

The SER module ALU is an ALU with an internal 16X16 RAM used to store descriptors, result descriptors and addresses.

Table 1-17. ISAY Defines

Address	Definition
0	Scratchpad (ISPD) address
1	Original job queue value
2	Memory address for RD
3	DLP RD word 1
4	DLP RD word 2
5	DLP RD word 3
6	IOT portion of RD
7	Longitudinal Parity Word (LPW)
8	Working register
9	Working register
A	Address digits X,7,6,5
B	Address digits 4,3,2,1
C	OP,V1,V2,V3 or special instruction flag
D	Descriptor Link word 1
E	Descriptor Link word 2
F	Working register

CMOP Description

ASAM OP

Table 1-18. ASAM OP

ASAM	(A)	[10]	A = Memory Address (AADR)
ASAM	(A<<A+1)	[12]	A replaced by A + 1
ASAM	(A<<A-2)	[13]	A replaced by A − 2
ASAM	(A-C)	[93]	A replaced by C
ASAM	(A-4SAVE)	[14]	A replaced by A − 4, A = Memory Address
ASAM	(A-INCBF)	[15]	A = Memory Address, A replaced by A + BF
ASAM	(A-INCD)	[16]	A = Memory Address, A replaced by A + D
ASAM	(A-INCE)	[17]	A = Memory Address, A replaced by A + E
ASAM	(A-INCWBB)	[18]	A = Memory Address, A repl'd by A + MWB/BI
ASAM	(A-INC1)	[19]	A = Memory Address, A replaced by A + 1
ASAM	(A-INC2)	[1A]	A = Memory Address, A replaced by A + 2
ASAM	(A-INC2BF	[1B]	A = Memory Address, A replaced by A + 2BF
ASAM	(A-INC3)	[1C]	A = Memory Address, A replaced by A + 3

Table 1-18. ASAM OP (Cont)

ASAM	(A-INC4)	[1D]	A = Memory Address, A replaced by A + 4
ASAM	(A-SAVE)	[1E]	A = Memory Address, A RESTORED
ASAM	(AOLPCHK)	[1F]	MOVE ALPHA OVERLAP Check Routine
ASAM	(B)	[20]	B = Memory Address
ASAM	(B<<B+E)	[22]	B replaced by B + E
ASAM	(B<<D)	[23]	B replaced by D
ASAM	(B<<MWB)	[24]	B replaced by MWB − 4 BYTES
ASAM	(B<<MWB4)	[25]	B replaced by MWB − 4 DIGITS
ASAM	(B-BASE0)	[2B]	B replaced by B minus BASE 0
ASAM	(BCT)	[94]	BRANCH COMMUNICATE Routine
ASAM	(B-4SAV)	[2C]	B replaced by B − 4, B = Memory Address
ASAM	(B-INCE)	[2E]	B = Memory Address, B replaced by B + E
ASAM	(B-INC2)	[2F]	B = Memory Address, B replaced by B + 2
ASAM	(B-INC2BF)	[31]	B = Memory Address, B replaced by B + 2BF
ASAM	(B-INC3)	[32]	B = Memory Address, B replaced by B + 3
ASAM	(B-INC4)	[33]	B = Memory Address, B replaced by B + 4
ASAM	(B-SAVE)	[34]	B = Memory Address, RESTORE B
ASAM	(B>C)	[35]	IF B > C, SET TSTTRU
ASAM	(B=C)	[36]	IF B = C. SET TSTTRU
	(BA0<<LIT)	[91]	= Base zero replaced by 3 bytes from MWB − MCS
ASAM	(BASESUB)	[37]	SUBTRACT the BASE from NI FOR ENTER
ASAM	(BASE0<WB)	[38]	BASE 0 replaced by MWB − 3 BYTES from MATH
ASAM	(C)	[39]	C = Memory Address
	(C<<A)	[92]	C replaced by A
ASAM	(C<<B+D)	[3A]	C replaced by B + D
ASAM	(C<<C-WBB)	[3B]	C replaced by C − MWB/BI :1 BYTE
ASAM	(C-INCWBB)	[3F]	C = Memory Address, C repl'd by C + MWB/BI
ASAM	(C-INC2BF)	[40]	C = Memory Address, C replaced by C + 2BF
ASAM	(C-INC3)	[41]	C = Memory Address, C replaced by C + 3
ASAM	(C-SAVE)	[42]	C = Memory Address, C RESTORED
ASAM	(C>B)	[43]	If C > B, SET TSTTRU
ASAM	(CHRXFR)	[44]	CHARACTER TRANSFER Routine
ASAM	(D)	[45]	D = Memory Address
ASAM	(D<<A)	[46]	D replaced by A

Table 1-18. ASAM OP (Cont)

ASAM	(D<<B)	[47]	D replaced by B
	(D<<LIT)	[4E]	D replaced by 4 bytes from MWB - MCS
ASAM	(D<<MWB)	[48]	D replaced by MWB:4 BYTES – MATH
ASAM	(D<<MWBB)	[49]	D replaced by MWB/BI :1 BYTE
ASAM	(E)	[4B]	E = Memory Address
ASAM	(E<<A)	[4C]	E replaced by A
ASAM	(E<<B)	[4D]	E replaced by B
ASAM	(E<<MWBB)	[50]	E replaced by MWB/BIN:1 BYTE – MCS
ASAM	(E-INC1)	[51]	E = Memory Address, E replaced by E + 1
ASAM	(E-INC3)	[52]	E = Memory Address, E replaced by E + 3
	(EPLUS1)	[90]	E replaced by E + 1, E = Memory Address
ASAM	(E<<LIT)	[4F]	E replaced with Literal from MWB:4 BYTES – MCS
ASAM	(F<<MWB)	[53]	F replaced with MWB:4 BYTES
ASAM	(F-INCE)	[54]	F = Memory Address, F replaced by F + E
ASAM	(F-INC16)	[55]	F = Memory Address, F replaced by F + 16
ASAM	(F-INC2BF)	[56]	F = Memory Address, F replaced by F + 2BF
ASAM	(FETCH)	[57]	Prepare for FETCH T on EXECUTE Transition
ASAM	(IOTXFER)	[5C]	Perform Transfer of Instruction to IOT
ASAM	(LIMT0<WB)	[6B]	replace LIMIT 0 with MWB:3 BYTES
ASAM	(LITB)	[6C]	B replaced with MWB:4 BYTES = Memory Addr
ASAM	(LITESAVE)	[6E]	E replaced with MWB:4 BYTES = Memory Addr, E restored
ASAM	(LM0<<LM3)	[6F]	LIMIT 0 replaced by LIMIT 3
ASAM	(MRB<<AF)	[70]	AF placed on MRB:1 BYTE
ASAM	(MRB<<B)	[71]	B placed on MRB:4 BYTES
ASAM	(MRB<<BF)	[73]	BF placed on MRB:1 BYTE
ASAM	(MRB<<C)	[74]	C placed on MRB:4 BYTES
ASAM	(MRB<<E)	[75]	E placed on MRB:4 BYTES
	(MRB<<F)	[6D]	F placed on MRB: 4 BYTES
ASAM	(MRB<<R1)	[76]	R1 placed on MRB:4 BYTES
ASAM	(MRB<<R4)	[78]	R4 placed on MRB:4 BYTES
	(NOP)	[01]	No OP
ASAM	(NOLPCHK)	[79]	MOVE NUMERIC OVERLAP Check Routine
ASAM	(NTRCALC)	[7E]	Enter CALCULATION Routine

Table 1-18. ASAM OP (Cont)

ASAM	(OFFSET)	[7F]	TRANSLATE ADDRESS Routine
ASAM	(RLIT)	[80]	BASE RELATIVE ADDRESS from MWB:4 BYTES = Memory Address
ASAM	(RLITBSAV)	[81]	BASE RELATIVE ADDRESS from MWB:4 BYTES = Memory Address, replaces B
ASAM	(RLITDSAV)	[83]	BASE RELATIVE ADDRESS from MWB:4 BYTES = Memory Address, replaces D
ASAM	(RLITESAV)	[85]	BASE RELATIVE ADDRESS from MWB:4 BYTES = Memory Address, replaces E
ASAM	(R4-SAV)	[87]	R4 = Memory Address, Base added in AADR – Relative Address in SPR
ASAM	(R1<<MWB)	[88]	R1 replaced by MWB:4 BYTES
ASAM	(R4<<MWB)	[89]	R4 replaced by MWB:4 BYTES
ASAM	(TESTCOMP)	[01]	used to Test and Reset TSTTRU
ASAM	(WOLPCHK)	[8A]	MOVE WORDS OVERLAP Check Routine

LENGTH UNIT OP

Table 1-19. Length Unit OP

LGTH	(AFAF)	[8D]	LU<<AF, TEMP << AF	
LGTH	(AFAFWB)	[84]	LU<<AF, TEMP<< AF,	MWB<<LU
LGTH	(AFAF1)	[8E]	LU<<AF, TEMP<<AF +1	
LGTH	(AFAF1WB)	[85]	LU<<AF, TEMP<< AF+1,	MWB<<LU
LGTH	(AF1AF)	[90]	LU<<AF+1, TEMP<< AF	
LGTH	(AF1AFWB)	[87]	LU<<AF+1, TEMP << AF,	MWB<<LU
LGTH	(AF1AF1)	[91]	LU<<AF+1, TEMP << AF+1	
LGTH	(AF1AF1WB)	[88]	LU<<AF+1, TEMP << AF+1,	MWB<<LU
LGTH	(AF12AF)	[92]	LU<<AF+1, TEMP << 2AF	
LGTH	(AF2AF)	[8F]	LU<<AF, TEMP<<2AF	
LGTH	(BF<<LU)	[A4]	BF replaced by LU	
LGTH	(BF<<0)	[00]	BF replaced by ZERO	
LGTH	(BF<<BF+1)	[01]	BF replaced by BF +1	
LGTH	(BF<<BF–1)	[02]	BF replaced by BF – 1	
LGTH	(BFBF)	[96]	LU<<BF, TEMP<<BF	
LGTH	(BFBF1)	[97]	LU<<BF, TEMP<<BF +1	

Table 1-19. Length Unit OP (Cont)

	(BF2BF)	[98]	LU<<BF, TEMP<<2BF	
LGTH	(BF1BF)	[99]	LU<<BF+1, TEMP<< BF	
LGTH	(BF1BF1)	[9A]	LU<<BF+1, TEMP << BF+1	
LGTH	(BF12BF)	[9B]	LU<<BF+1, TEMP << 2BF	
LGTH	(LU<AF+BF)	[B1]	LU<<AF+BF, TEMP unchanged	
LGTH	(LU<BF−AF)	[B3]	LU<<BF − AF. TEMP unchanged	
LGTH	(LU<AQL+1)	[BC]	LU<<AQL+1, TEMP<<AQL+1	
LGTH	(L<2AQL+2)	[BD]	LU<<2AQL+2, TEMP<<2AQL+2	
LGTH	(LU<<1)	[C4]	LU<<LITERAL, TEMP unchanged	
LGTH	(LU<<2)	[C5]	. .	
LGTH	(LU<<4)	[C6]	. .	
LGTH	(LU<<6)	[C8]	. .	
LGTH	(LU<<7)	[C9]	. .	
LGTH	(LU<<8)	[CA]	. .	
LGTH	(LU<<10)	[D1]	. .	
LGTH	(LU<<12)	[CB]	. .	
LGTH	(LU<<16)	[CC]	. .	
LGTH	(LU<<20)	[CE]	. .	
LGTH	(LU<<100)	[CF]	LU<<LITERAL, TEMP unchanged	
LGTH	(LU<<200)	[D0]	.	
LGTH	(LU<LU+BF)	[AE]	LU replaced by LU +BF	
LGTH	(LU<LU−AF)	[AF]	LU replaced by LU − AF	
LGTH	(LULU)	[B4]	LU<<LU, TEMP<<LU	
LGTH	(LULU1)	[B5]	LU<<LU, TEMP<<LU +1	
LGTH	(LULUWB)	[BA]	LU<<LU, TEMP<< LU,	MWB<<LU
LGTH	(LU1LU)	[B7]	LU<<LU+1, TEMP<< LU	
LGTH	(LU1LU1)	[B8]	LU<<LU+1, TEMP << LU+1	
LGTH	(LU2LU)	[B6]	LU<<LU, TEMP<<2LU	
LGTH	(LU2LUWB)	[BB]	LU<<LU, TEMP<< 2LU,	MWB<<LU
LGTH	(L2AFAF)	[93]	LU<<2AF, TEMP<< AF	
LGTH	(L2AFAFWB)	[8A]	LU<<2AF, TEMP<< AF,	MWB<<LU
LGTH	(L2AFAF1)	[94]	LU<<2AF, TEMP<< AF+1	
LGTH	(L2AFAF1W)	[8B]	LU<<2AF, TEMP<< AF+1,	MWB<<LU
LGTH	(L2AF2AF)	[95]	LU<<2AF, TEMP<< 2AF,	

Table 1-19. Length Unit OP (Cont)

LGTH	(L2BFBF)	[9C]	LU<<2BF, TEMP<< BF
LGTH	(L2BFBF1)	[9D]	LU<<2BF, TEMP<< BF+1
LGTH	(L2BF2BF)	[9E]	LU<<2BF, TEMP<< 2BF
LGTH	(MWB<<LU)	[03]	MWB<<LU:1 BYTE

Literal OP

Table 1-20. Literal OP

LIT	(L00)
LIT	(L01)
LIT	(L02)
LIT	(L04)
LIT	(L06)
LIT	(L08)
LIT	(L10)
LIT	(L12)
LIT	(L14)
LIT	(L16)
LIT	(L24)
LIT	(L38)
LIT	(L40)
LIT	(L46)
LIT	(L48)
LIT	(L61)
LIT	(L63)
LIT	(L64)
LIT	(L77)
LIT	(L80)
LIT	(LC0)
LIT	(LF0)
LIT	(LFB)

MATH OP

Table 1-21. Math OP

MATH	(ACC<<0)	[13]	SET ACCUMULATOR to ZERO
MATH	(ADD)	[03]	ADD Routine
	(ALD)	[AB]	ACCUM LOAD – Move ACCUM to Q
MATH	(AND)	[2C]	Logical AND Routine
MATH	(AND/OR)	[47]	Logical AND Routine for SEARCHES
	(AQ<<BQ)	[20]	Moves BQ to AQ
MATH	(AQRP<MWB)	[17]	Load AQ READ pointer from MWB:1 BYTE

Table 1-21. Math OP (Cont)

MATH	(AQRP+1)	[33]	Increment the AQ READ pointer
	(AQRP-1)	[33]	Increment the AQ READ pointer
MATH	(BCT1)	[11]	1st BCT Routine
MATH	(BCT2)	[12]	2nd BCT Routine
MATH	(BOT)	[13]	BIT ONE Test Routine
MATH	(BQRP<MWB)	[34]	Load the BQ READ pointer from MWB:1 BYTE
MATH	(BREB)	[48]	BRE ASSEMBLE the BASE Routine
MATH	(BREL)	[4A]	BRE ASSEMBLE the LIMIT Routine
MATH	(BRT)	[39]	BIT RESET Routine
MATH	(BST)	[3A]	BIT SET Routine
MATH	(BZT)	[27]	BIT ZERO Test Routine
MATH	(CLEAR)	[10]	CLEAR the Q READ & WRITE pointers
MATH	(CLULDAQ)	[56]	CLEAR and UNLOAD the AQ
	(CLULDCQ)	[16]	CLEAR and UNLOAD the CQ
MATH	(COMPARE)	[24]	Compare the AQ to the BQ
MATH	(ConVERT)	[49]	Convert REAL to Integer
MATH	(CPA)	[09]	COMPARE ALPHA Routine
MATH	(CPN)	[08]	COMPARE NUMERIC Routine
	(CQCLR)	[21]	CLEAR CQ pointers
MATH	(CQ<<AQCH)	[31]	Transfer A Character from the AQ to the CQ
MATH	(CQ<<BQCH)	[32]	Transfer A Character from the BQ to the CQ
MATH	(CQ<<BQRB)	[4E]	EDIT MOVE SUPPRESS Routine
MATH	(CQ<<MWB)	[36]	Load the CQ with A Character from the MWB
MATH	(DIV)	[07]	DIVIDE Routine
MATH	(ELD)	[4C]	REAL EXPONENT LOAD Routine
MATH	(EXPADD)	[OD]	REAL INCREMENT the EXPONENT Routine
MATH	(EXPSUB)	[OC]	REAL DECREMENT the EXPONENT Routine
MATH	(FAD)	[3E]	FLOATING POINT ADD Routine
MATH	(FDV)	[3C]	FLOATING POINT DIVIDE Routine
MATH	(FLD)	[38]	FLOATING POINT LOAD Routine
MATH	(FMP)	[3B]	FLOATING POINT MULTIPLY Routine
MATH	(IAD)	[3D]	Integer ADD Routine
MATH	(ILD)	[3F]	Integer LOAD Routine
MATH	(IMD)	[40]	Integer Memory DECREMENT Routine
MATH	(IMI)	[41]	Integer Memory INCREMENT Routine
MATH	(IMP)	[57]	Integer MULTIPLY Routine
MATH	(IST)	[43]	Integer STORE Routine
MATH	(LOADAQ)	[14]	LOAD the AQ via the MRB
MATH	(LOADBQ)	[19]	LOAD the BQ via the MRB
MATH	(LOADCQ)	[1C]	LOAD the CQ via the MRB

Table 1-21. Math OP (Cont)

MATH	(M<<MRB)	[30]	DUMMY LOAD via the MRB
MATH	(MOT)	[44]	MASK OR Routine for HALT BREAKPOINT
	(MMV)	[58]	MOVE MANTISSA to ACCUM from QUE
	(MMVZ)	[59]	
MATH	(MSIG <<C)	[22]	COMPLEMENT the MANTISSA SIGN of ACCUM.
MATH	(MSIG <<M)	[55]	SET the ACCUMULATOR MANTISSA SIGN to (-)
MATH	(MSIG <<P)	[54]	SET the ACCUMULATOR MANTISSA SIGN to PLUS
MATH	(MULT)	[06]	MULTIPLY Routine
MATH	(MVA)	[0B]	MOVE ALPHA Routine
MATH	(MV)	[0A]	MOVE NUMERIC Routine
MATH	(MVR)	[0E]	MOVE REPEAT Routine
MATH	(NOOP)	[00]	No Operation
MATH	(NORACC)	[53]	NORMALIZE the ACCUMULATOR
MATH	(NOT)	[28]	Logical EXCLUSIVE OR Routine
MATH	(NTR)	[37]	ENTER Routine
MATH	(OR)	[2A]	Logical OR Routine
MATH	(RAA)	[4D]	REAL ADD Routine
MATH	(RDOR)	[1F]	Result Descriptor OR
MATH	(RDV)	[4F]	REAL DIVIDE Routine
MATH	(RELOADAQ)	[25]	CLEAR the AQ pointers & LOAD via MRB
MATH	(RELOADBQ)	[26]	CLEAR the BQ pointers & LOAD via MRB
MATH	(RELOADCQ)	[45]	CLEAR the CQ pointers & LOAD via MRB
MATH	(RLD)	[50]	REAL LOAD Routine
MATH	(RMU)	[52]	REAL MULTIPLY Routine
MATH	(RST)	[51]	REAL STORE
MATH	(SCANEQL)	[2E]	SCAN EQUAL Routine
MATH	(SCANNQL)	[2F]	SCAN NOT EQUAL Routine
MATH	(SHFT)	[OF]	for BRE Shift one DIGIT to LEFT
MATH	(TRAPLD)	[46]	TRAP LOAD Routine
MATH	(TRNNEXT)	[35]	TRANSLATE Routine
MATH	(UNLOADAQ)	[18]	place the contents of the AQ on the MWB
MATH	(UNLOADBQ)	[1B]	place the contents of the BQ on the MWB
MATH	(UNLOADCQ)	[1E]	place the contents of the CQ on the MWB

MCS FLAGS

Table 1-22. MCS Flags

MCSFLAG	(COMH<0)	[1D]	Comparison HIGH TOGGLE
MCSFLAG	(COMH<1)	[2D]	
MCSFLAG	(COML<0)	[1E]	Comparison LOW TOGGLE
MCSFLAG	(COML<1)	[2E]	
MCSFLAG	(CSZB<0)	[15]	CONTROL STATE BASE ZERO TOGGLE
MCSFLAG	(CSZB<1)	[05]	
MCSFLAG	(ERPTEN<0)	[13]	MEMORY ERROR REPORT ENABLE
MCSFLAG	(ERPTEN<1)	[23]	
MCSFLAG	(FETCH<0)	[18]	FETCH FLAG
MCSFLAG	(FETCH<1)	[08]	
MCSFLAG	(FRST<0)	[19]	FIRST FLAG
MCSFLAG	(FRST<1)	[29]	
MCSFLAG	(FGO)	[12]	FETCH GO FLAG
MCSFLAG	(FGO)	[22]	
MCSFLAG	(INVINS<0)	[1B]	INVALID INSTRUCTION FLAG
	(MWB<<TO6)	[1D0]	COMs and overflow on to MWB (onto LSD of MWB)
	(SPACER)	[200]	THE TIMER
MCSFLAG	(INVINS<1)	[0B]	
MCSFLAG	(INVSGN<0)	[1A]	INVERT the SIGN FLAG
MCSFLAG	(INVSGN<1)	[2A]	
MCSFLAG	(MERRPT<0)	[1C]	MEMORY ERROR REPORT FLAG
MCSFLAG	(MERRPT<1)	[0C]	
MCSFLAG	(MVW<0)	[11]	MOVE WORDS FLAG
MCSFLAG	(MVW<1)	[1FO]	
MCSFLAG	(MWB<RD1)	[1EO]	MWB << First BYTE of PROCESSOR R/D
MCSFLAG	(MWB<RD2)	[FE]	MWB << Second BYTE of PROCESSOR R/D
MCSFLAG	(NOIX2<0)	[19]	IX2 not required FLAG
MCSFLAG	(NOIX2<1)	[29]	
MCSFLAG	(NOR<0)	[15]	NORMAL STATE TOGGLE
MCSFLAG	(NOR<1)	[25]	
MCSFLAG	(OVF<0)	[1F]	OVERFLOW TOGGLE
MCSFLAG	(OVF<1)	[2F]	
MCSFLAG	(PINT<0)	[12]	PROCESSOR INTERRUPT TOGGLE
MCSFLAG	(PINT<1)	[02]	
MCSFLAG	(PLUS<0)	[19]	Remember the SIGN FLAG
MCSFLAG	(PLUS<1)	[29]	
MCSFLAG	(PROHLT<0)	[14]	PROCESSOR HALT TOGGLE
MCSFLAG	(PROHLT<1)	[04]	
MCSFLAG	(Q<0)	[1A]	EDIT Check PROTECT FLAG

Table 1-22. MCS Flags (Cont)

MCSFLAG	(Q<1)	[2A]	
	(BPLUS<0)	[2A]	Remember SIGN FLAG
	(BPLUS<1)	[1A]	
	(TINTEN<0)	[03]	TIMER INT. ENABLE
	(TINTEN<1)	[13]	
	(RDCL)	[100]	
MCSFLAG	(OVFC<0)	[16]	RESULT FIELD ODD FLAG
MCSFLAG	(OVFC<1)	[26]	
MCSFLAG	(SGNINV<0)	[1A]	SIGN INVERT FLAG
MCSFLAG	(SGNINV<1)	[0A]	
MCSFLAG	(SMRF<0)	[19]	SMEAR CASE FLAG
MCSFLAG	(SMRF<1)	[29]	
MCSFLAG	(SNPGEN<0)	[14]	SNAP-GATE ENABLE
MCSFLAG	(SNPGEN<1)	[24]	
MCSFLAG	(T<0)	[11]	EDIT Significance FLAG (FLAG2<0)
MCSFLAG	(T<1)	[01]	EDIT Significance FLAG (FLAG2<1)
MCSFLAG	(TEMP<0)	[19]	TEMPORY STORAGE TOGGLE (FLAG1<0)
MCSFLAG	(TEMP<1)	[29]	TEMPORY STORAGE TOGGLE (FLAG1<1)
MCSFLAG	(TRACE<0)	[16]	TRACE TOGGLE
MCSFLAG	(TRACE<1)	[06]	TRACE TOGGLE
MCSFLAG	(TRAPF<0)	[1C]	TRAP FLAG
MCSFLAG	(TRAPF<1)	[2C]	TRAP FLAG
MCSFLAG	(TVWEN<0)	[17]	TEMP & VOLTAGE WARNING ENABLE
MCSFLAG	(TVWEN<1)	[07]	TEMP & VOLTAGE WARNING ENABLE

PSI READ OP

Table 1-23. PSI Read OP

PSIR	(A<<CH)	[10]	READ, MRB << CHARACTERS
PSIR	(A<<F0)	[F0]	READ, MRB << LITERAL 'F0' CHARACTERS
PSIR	(A<<M0)	[D0]	READ, MRB << CHARACTERS, LEAST DIGIT SET TO ZERO
PSIR	(A<<N<<UA)	[B0]	READ, MRB << CHARACTERS, ZONE DIGIT FORCED to 'F'
PSIR	(A<<SN)	[90]	READ, MRB << CHARACTERS, DETECT SIG ZONE DIGIT ADDED
PSIR	(A<<UN)	[80]	READ, MRB << CHARACTERS, ZONE DIGIT ADDED
PSIR	(N<<FN)	[30]	READ, MRB << DIGITS, 'F' DETECTED

Table 1-23. PSI Read OP (Cont)

PSIR	(N<<FUN)	[C0]	READ, MRB << LEADING 'F', the DIGITS
PSIR	(N<<SA)	[50]	READ, MRB << DIGITS, STRIP ZONES, DETECT SIGN
PSIR	(N<<SN)	[30]	READ, MRB << DIGITS, DETECT SIGN
PSIR	(N<<UA)	[40]	READ, MRB << DIGITS, STRIP ZONES
PSIR	(N<<UN)	[10]	READ, MRB << DIGITS
PSIR	(N<<0SA)	[E0]	READ, MRB << LEADING '0', the DIGITS, STRIP ZONES, DETECT SIGN
PSIR	(N<<0SN)	[60]	READ, MRB << LEADING '0', the DIGITS, DETECT SIGN
PSIR	(N<<0UA)	[70]	READ, MRB << LEADING '0', the DIGITS, STRIP ZONES
PSIR	(N<<0UN)	[A0]	READ, MRB << LEADING '0', the DIGITS

PSI WRITE OP

Table 1-24. PSI Write OP

PSIW	(CH<<A)	[10]	WRITE CHARACTERS
PSIW	(ERRPT)	[80]	WRITE MEMORY ERROR REPORT
PSIW	(SA<<A)	[70]	WRITE CHARACTERS, STORE SIGN
PSIW	(SA<<N)	[40]	WRITE CHARACTERS, ADD ZONES, STORE SIGN
PSIW	(SA<<0N)	[E0]	WRITE CHARACTERS, ADD ZONES, STORE SIGN STRIP LEADING DIGIT
PSIW	(SN<<N)	[20]	WRITE DIGITS, STORE SIGN
PSIW	(S <<0)	[A0]	WRITE DIGITS, STORE SIGN, STRIP LEADING DIGIT
PSIW	(UA<<N)	[30]	WRITE CHARACTERS, ADD ZONES
PSIW	(UA<<0N)	[60]	WRITE CHARACTERS, ADD ZONES, STRIP LEADING DIGIT
PSIW	(UN<<A)	[D0]	WRITE DIGITS, STRIP MOST SIGNIFICANT DIGIT
PSIW	(UN<<N)	[10]	WRITE DIGITS

Processor Result Descriptors (R/D)

The Processor R/D is 16 bits long and is written into absolute memory address 80. The 16 bits are numbered (for explanation purposes only) from bit 1 through bit 16. Bit 1 is the most significant bit in the least significant digit of the Processor R/D; i.e., the 8-bit of the digit contained at absolute memory address 80. Bit 16 is the least significant digit of the Processor R/D; i.e., the 1-bit of the digit contained in absolute memory address 83.

Various system conditions are reported in the Processor R/D. Several of these conditions cause the Interrupt Toggle to set and a Branch Communicate to occur after the R/D is written to memory. What occurs is determined by what state the processor is in at the time the R/D is written; i.e., Control State Base = 0, Control State Base = 0, Normal State Base = 0, or Normal State Base = 0. In addition, not all Processor R/D are written by the processor. Some R/D are written into address 80 by the Maintenance System.

The following table shows the bit assignments and various conditions that occur.

Table 1-25. Result Descriptor Bit Assignments

BIT	CONDITION	CTRL STATE BASE = 0	CTRL STATE BASE ≠ 0	NORM STATE BASE = 0	NORM STATE BASE ≠ 0	SNAP PICTURE	SPECIAL NOTES
1	Descriptor Present						
2	Exception Condition						
3	Invalid Data	BR	BR	BR	BR	S	
4	Invalid I/O Descriptor	RP	BR	XX	XX		1
5	Invalid Instruction	BR	BR	BR	BR	S	
6	Uncorr Mem Parity Error	BR	BR	BR	BR	S	
7	Address Error	BR	BR	BR	BR	S	
8	Instruction Timeout	BR	BR	BR	BR	S	
9	Timer Interrupt	RP	BRP	BRP	BRP		1
10	Reserved						
11	Memory Error Report	0	0	0	0		2
12	Reserved						
13	Air Loss/ Over Temp	RP	BR	BR	BR		1,2
14	SNAP Picture Report	BM	BM	BM	BM		2

Table 1-25. Result Descriptor Bit Assignments (Cont)

BIT	CONDITION	CTRL STATE BASE = 0	CTRL STATE BASE ≠ 0	NORM STATE BASE = 0	NORM STATE BASE ≠ 0	SNAP PICTURE	SPECIAL NOTES
15	Temperature Warning *	O	O	O	O		
16	Voltage Warning **	O	O	O	O		
	I/O Complete	P	P	BP	BP		

B = BCT to 94 (OP 30)
M = Maintenance written R/D
R = R/D written
O = ORed into R/D when any other R/D is written
S = Snap picture if enabled
P = Set PINT
X = Don't care
* = B 3955 only
** = B 2900/B 3900 only

NOTES

1. If Timer Interrupt, Air Loss/Over Temp or IOC is sensed in control state, base = 0, the machine needs to BCT (OP 30) as soon as it leaves this state. This BCT is remembered by the flip-flop CPINT. CPINT is reset by SRD (OP 91).

2. BCT and R/D are only executed if these functions are enabled by the Set Mode command. The enable is turned off when the R/D is written into memory.

EXCEPTION CONDITION (BITS 1 & 2)

Bits 1 and 2 indicate that some exception condition exists which is reported in bits 3 thru 16. Bit 1 and bit 2 always occur together.

INVALID ARITHMETIC DATA (BIT 3)

Bit 3 is set when an undigit has been detected in an arithmetic operand other than the sign digit. Invalid Instruction, bit 5, is also set on this condition.

INVALID I/O DESCRIPTOR (BIT 4)

Bit 4 is set for an Invalid I/O Descriptor. This bit indicates that an I/O descriptor OP code is invalid for the DLP present on the indicated channel. The Interrupt toggle is set.

INVALID INSTRUCTION (BIT 5)

Bit 5 is used to indicate that an Invalid Instruction has been detected. A SNAP picture is taken if enabled by the Set Mode instruction.

Invalid instructions are:

1. Not assigned operator codes.
2. Privileged instructions, if not base zero.
3. Invalid Halts per Halt Execution digit (absolute address 77).
4. Invalid address (non-decimal digits) specified by a Branch Communicate instruction.
5. Invalid Branch Communicate address (the high order digit is not equal to F or undigits in the address).
6. Invalid instruction usage as specified in the individual instructions.
7. Invalid arithmetic data (non-decimal digits).

Execution of the invalid instruction is replaced with a Branch Communicate (OP=30) to absolute address 94. The absolute address of the Invalid Instruction is stored in lieu of the next instruction address.

UNCORRECTABLE MEMORY PARITY ERROR (BIT 6)

Bit 6 is set to indicate the detection of an Uncorrectable ''double-bit' Memory Parity Error or Address Bus Parity Error. An Uncorrectable Memory Parity Error during the execution of a Processor instruction terminates the instruction without writing into memory at the location the error was detected. A Branch Communicate (OP=30) to absolute address 94 is initiated.

ADDRESS ERROR (BIT 7)

Bit 7 is used to indicate the detection of an Address Error. A SNAP picture is taken if enabled by the Set Mode instruction.

Address Errors refer to the following conditions.

1. Base/Limit Error (Address less than Base or greater than or equal to Limit).
2. Non-decimal Digit contained in Instruction Addresses.
3. Odd Instruction Address.
4. Improper literal or literal not allowed.

The instruction is terminated immediately and a Branch Communicate to 94 is executed.

Detection of address errors is made on the addresses as they are initially read from memory.

INSTRUCTION TIME OUT (BIT 8)

Bit 8 is set to indicate that an Instruction Time Out has occurred. A timer of approximately 250 milliseconds is started with the execute of each instruction. Timing out causes a Result Descriptor to be written and a Branch Communicate to absolute address 94. A SNAP picture is taken if enabled by the Set Mode (OP=47) instruction.

TIMER INTERRUPT (BIT 9)

Bit 9 is set to indicate a Timer Interrupt (not considered to be an error). This occurs when the Real Time Timer counts to a preset value and the system is in Control State and the Base is not equal to zero or in Normal State.

UNUSED (BIT 10)

Bit 10 is not used.

MEMORY ERROR REPORTED (BIT 11)

This function must be enabled by the Set Mode instruction (OP=47). Bit 11 is set to indicate that a memory error report has been written in the memory location specified by Register R4. The enable is reset upon detection. This bit does not cause an interrupt.

UNUSED (BIT 12)

Bit 12 is not used.

AIR LOSS/OVER TEMPERATURE (BIT 13)

This function must be enabled by the Set Mode (OP=47) instruction. Bit 13 indicates an air loss (blower failure) or an over-temperature condition in the system. After detection, an 8-second delay occurs before a power-off is initiated. The enable function is reset after the R/D is written. This condition causes an interrupt.

SNAP GATE PICTURE REPORTED (BIT 14)

This function must be enabled by the Set Mode (OP=47) instruction. Bit 14 is set to indicate that a Snap Gate report has occurred. The contents of the registers and toggles are stored in the memory location as specified by register R0. The interrupt toggle is set which causes a BCT to 94 when in normal state. The enable function is disabled after the R/D and picture are written.

The detection of an invalid instruction, address error, or instruction timeout causes a Snap Gate picture to occur.

TEMPERATURE WARNING (BIT 15) B 3955 ONLY

This function must be enabled by the Set Mode (OP=47) instruction. Bit 15 indicates that the system temperature has exceeded a preset value. The detection of a Temperature Warning does not cause a power-off cycle to occur. The detection of this condition sounds the audible alarm. The status of this bit is copied for all R/Ds as long as the temperature warning exists. This bit does not cause an Interrupt.

VOLTAGE WARNING (BIT 16) B 2900/B 3900 ONLY

This function must be enabled by the Set Mode (OP=47) instruction. Bit 16 indicates that a low voltage condition has been detected. This does not cause a power-off cycle to occur. If the voltage drops too low for system operation to continue, an immediate power-off cycle occurs. The status of this bit is copied for all R/Ds as long as the voltage warning exists. This bit does not cause an Interrupt.

Interrupts

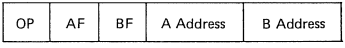
Interrupts occurring in Control State (NOR = 0) are ignored except Bits 3,5,6,7, and 8 which cause a Branch Communicate to absolute address 94 in all cases.

Bits 3,5,6,7,8,9 and 13 cause a Branch Communicate to absolute address 94 in Control State if the Base is not equal to zero.

Processor Instructions

The following are detailed descriptions of each processor instruction. For detailed description of all OP codes, see B 2900 Reference Manual #1115458.

INCREMENT (INC)



OP = 01

AF = Field Length of the A operand. Can be indirect or literal flag.
BF = Field Length of the B operand. Can be indirect.
A = Address of the A field (addend) operand. This field can be a literal with a maximum field size of 6 UN, 5 SN, or 3 UA.
B = Address of the B field (augend) and the address of the sum field.

The A and B Field addresses can be indexed, indirect or extended.

The INC instruction algebraically adds the contents of the A field to the contents of the B field and stores the sum in the B field. The addition is accomplished most significant digit first.

Example 1: Increment an Alpha Field to a Signed Field

OP	AF	BF	A	B
01	02	04	201000	102000

	Before	After
1000	C1E7	C1E7
2000	+0257	+0274
Comparison	xx	HIGH
Overflow	xx	unchanged

Example 2: Increment with Overflow Condition

OP	AF	BF	A	B
01	02	03	001000	002000

	Before	After
1000	18	18
2000	987	987
Comparison	xx	unchanged
Overflow	xx	On

ADD (ADD)

OP	AF	BF	A Address	B Address	C Address

OP = 02

AF = Field length of the A field. Can be indirect or literal flag.
BF = Field length of the B field. Can be indirect.
A = Address of the A field (addend) operand. This field can be a literal with a maximum field size of 6 UN, 5 SN, or 3 UA.
B = Address of the B field (augend) operand.
C = Address of the C field (sum).
The A,B and C address fields can be indexed, indirect or extended.

An ADD instruction algebraically adds the contents of the A field to the contents of the B Field and stores the sum in the C field. Addition is accomplished most significant digit first.

Example 1: ADD UN field to an SN field giving an SN sum.

OP	AF	BF	A	B	C
02	02	05	001000	102000	10300C

	Before	After
1000	20	20
2000	+00015	+00015
3000	xxxxxx	+00035
Comparison	xx	HIGH
Overflow	xx	unchanged

Example 2: Add a UN field to an SN field giving a UA Field.

OP	AF	BF	A	B	C
02	02	05	001000	102000	203000

	Before	After
1000	10	10
2000	−00050	−00050
3000	xxxxx	F0F0F0F4F0
Comparison	xx	LOW
Overflow	xx	unchanged

Example 3: Add two fields with an overflow condition.

OP	AF	BF	A	B	C
02	02	02	001000	002000	203000

	Before	After
1000	61	61
2000	53	53
3000	xx	unchanged
Comparison	xx	unchanged
Overflow	xx	On

DECREMENT (DEC)

OP	AF	BF	A Address	B Address

OP = 03

AF = Field length of the A operand. Can be indirect or literal flag.
BF = Field length of the B operand. Can be indirect.
A = Address of the A field (subtrahend) operand. This field can be a literal with a maximum field size of 6 UN, 5 SN, or 3 UA.
B = Address of the B field (minuend) operand and the result (difference) operand.

The A and B Field addresses can be indexed, indirect or extended.

A DEC instruction algebraically subtracts the contents of the A field from the contents of the B field and stores the result in the B field. Subtraction is accomplished most significant digit first.

Example 1: Decrement an SN field from a UN field.

OP	AF	BF	A	B
03	02	03	101000	002000

	Before	After
1000	−71	−71
2000	121	192
Comparison	xx	HIGH
Overflow	xx	unchanged

Example 2: Decrement an SN field from an SN field causing Overflow.

OP	AF	BF	A	B
03	03	03	101000	102000

	Before	After
1000	−311	−311
2000	+942	+942
Comparison	xx	unchanged
Overflow	xx	On

SUBTRACT (SUB)

OP	AF	BF	A Address	B Address	C Address

OP = 04

AF = Field length of the A field operand. Can be indirect or literal flag.
BF = Field length of the B field operand. Can be indirect.
A = Address of the A field (subtrahend) operand. This field can be a literal with a maximum field size of 6 UN, 5 SN, or 3 UA.
B = Address of the B field (minuend) operand.
C = Address of the C field (difference).
The A, B, and C address fields can be indexed, indirect or extended.

A SUB instruction algebraically subtracts the contents of the A field from the contents of the B field and stores the result in the C field. The length of the C field is equal to the larger of the AF or BF. Subtraction is accomplished most significant digit first.

Example 1: Subtract a UN field from a UA Field giving an SN field.

OP	AF	BF	A	B	C
04	01	05	001000	202000	103000

	Before	After
1000	5	5
2000	C1C2C3C4C5	C1C2C3C4C5
3000	xxxxxx	+12340
Comparison	xx	HIGH
Overflow	xx	unchanged

MULTIPLY (MPY)

OP	AF	BF	A Address	B Address	C Address

OP = 05

AF = Field length of the A operand. Can be indirect or literal flag.
BF = Field length of the B operand. Can be indirect.
A = Address of the A field (multiplier) operand. This field can be a literal with a maximum field size of 6 UN, 5 SN, or 3 UA.

B = Address of the B field (multiplicand) operand.
C = Address of the C field (product).
The A,B and C address fields can be indexed, indirect or extended.

An MPY instruction algebraically multiplies the contents of the A field by the contents of the B field and stores the product in the C field. The C field length is the sum of AF and BF.

Example 1: Multiply a UA Field by an SN field giving a UA product.

OP	AF	BF	A	B	C
05	02	05	201000	102000	103000

	Before	After
1000	D1D2	D1D2
2000	−00011	−00011
3000	xxxxxx	F0F0F0F0F1F3F2
Comparison	xx	HIGH

Example 2: Multiply an SN field by an SN field giving an SN product.

OP	AF	BF	A	B	C
05	02	05	201000	102000	103000

	Before	After
1000	−15	−15
2000	−17	−17
3000	xxxxx	+0255
Comparison	xx	HIGH

DIVIDE (DIV)

OP	AF	BF	A Address	B Address	C Address

OP = 06

AF = Field length of the A operand. Can be indirect or literal flag.
BF = Field length of the B operand. Can be indirect.
A = Address of the A field (divisor) operand. This field can be a literal with a maximum field size of 6 UN, 5 SN, or 3 UA.
B = Address of the B field (dividend/remainder) operands.
C = Address of the C field (quotient).
The A,B and C address fields can be indexed, indirect or extended.

A DIV instruction algebraically divides the contents of the B field by the contents of the A field and stores the quotient in the C field and the remainder in the B field. The length of the dividend must be greater than the length of the divisor field (BF must be greater than the AF). The length of the quotient field is the difference in length of the A and B fields (BF − AF).

Example 1: Divide positive field into negative field giving SN result with SN remainder.

OP	AF	BF	A	B	C
06	02	05	001000	102000	103000

	Before	After
1000	12	12
2000	−00187	−00007
3000	xxxx	−015
Comparison	xx	LOW
Overflow	xx	unchanged

Example 2: Divide SN field into UN field with result length being larger than quotient field size.

OP	AF	BF	A	B	C
06	02	03	101000	002000	003000

	Before	After
1000	+12	+12
2000	134	134
3000	x	unchanged
Comparison	xx	unchanged
Overflow	xx	On

Example 3: Divide showing AF greater than BF.

OP	AF	BF	A	B	C
06	03	02	001000	002000	003000

	Before	After
1000	123	123
2000	11	11
3000	xx	unchanged
Comparison	xx	unchanged
Overflow	xx	On

MOVE DATA (MVD)

OP	AF	BF	A Address	B Address	C Address

OP = 08

AF = Forward/Backward variant. A value of 00 specifies a forward move. A value of 01 specifies a backward move. Indirect field length can be specified in AF; however, the use of the variant as a literal flag is ignored. All other values of this variant are reserved.

BF = Unused and reserved

A = Address of the A field (source). The address controller bits are ignored.

B = Starting address of the B field (destination), The address controller bits are ignored.

C = End address of the destination data field. The address controllers are ignored. The difference between the B and C addresses must be Mod 4 for proper operation of this instruction.

The A, B and C address fields can be indexed, indirect or extended.

An MVD instruction moves words (groups of four digits) from the A field to the B field until the B Address equals the C Address. The MVD is accomplished either in a forward direction or a backward direction.

NOTES

In B 3500 thru B 4800 systems the A, B, and C addresses must point to a Mod 4 boundary. In the B 3900/B 2900/B 29XX systems, there are no restrictions on which mod boundary addresses must point to. The only restriction, in order to ensure that the instruction functions properly, is that the difference between the B and C addresses must be mod 4 because of the incrementing and decrementing of addresses by a value of 4.

In B 3500 thru B 4800 systems the address controllers must be UN. In the B 3900/B 2900/B 29XX systems, the address controller bits are ignored; thus the system forces these fields to be referenced by this instruction as UN fields, regardless of what controller bits are present.

Example 1: Forward Move

OP	AF	BF	A	B	C
08	00	00	001000	002000	002012

	Before	After
1000	123456789ABC	123456789ABC
2000	xxxxxxxxxxxx	123456789ABC

The comparison and overflow toggles are not affected.

Example 2: Backward Move

OP	AF	BF	A	B	C
08	01	00	001011	002010	001998

	Before	After
999	ABCDEF012345	ABCDEF012345
1998	xxxxxxxxxxxx	ABCDEF012345

The comparison and overflow toggles are not affected.

MOVE LINKS (MVL)

OP	AF	BF	A Address	B Address	C Address

OP = 09

AF = Length of the A, B, and C fields. Indirect field length can be specified. The use of the literal flag is allowed by this system; however, it is not recommended for general use.
BF = Unused and reserved.
A = Address of the A field. The address controller bits are ignored, because the data type of this field is determined by the address controller bits in the C Address.
B = Address of the B field. The address controller bits are ignored, because the data type of this field is determined by the address controller bits in the C Address.
C = Address of the C field. Standard address controller useage applies.

The A, B and C address fields can be indexed, indirect or extended.

The MVL instruction moves the number of units of data specified by the AF variant.

NOTE

In order to preclude address errors in the operation of B 3500 through B 4800 systems, it is necessary to exercise care when specifying the C Address controller bits as UA, to ensure that the A and B field addresses point to a Mod 2 boundary.

In the B 3900/B 2900/B 29XX systems, odd boundary pointers for the A and B addresses are allowed with UA indication in the C address field, without resultant address errors.

Example 1: Move Alpha fields showing that C address controller is controlling A and B address fields.

OP	AF	BF	A	B	C
09	05	00	001000	102000	203000

	Before	After
1000	9876543210	D234516789
2000	D234516789	F0F1F2F3F4
3000	F0F1F2F3F4	9876543210

The comparison and overflow toggles are not affected.

Example 2: Two-field exchange using MVL instruction

OP	AF	BF	A	B	C
09	04	00	001000	002000	002000

	Before	After
1000	1234	5678
2000	5678	1234

The comparison and overflow toggles are not affected.

MOVE ALPHANUMERIC (MVA)

OP	AF	BF	A Address	B Address

OP = 10

AF = Length of the A field. Can be indirect or literal flag.
BF = Length of the B field. Can be indirect flag.
A = Address of the A field (source).
B = Address of the B field (destination).
The A and B Field addresses can be indexed, indirect or extended.

An MVA instruction moves either digits or characters (as specified by the address controllers) from the A field and stores them at the location specified by the B Address, left-justified.

Example 1: Move UN field to SN field causing Overflow.

OP	AF	BF	A	B
10	05	03	001000	102000

	Before	After
1000	12345	12345
2000	xxxx	+123
Comparison	xx	HIGH
Overflow	xx	On

Example 2: Move UA Field to an SN field.

OP	AF	BF	A	B
10	03	05	201000	102000

	Before	After
1000	D3F5F6	D3F5F6
2000	xxxxxx	+35600
Comparison	xx	LOW
Overflow	xx	unchanged

Example 3: Move with B address occurring in A data field.

OP	AF	BF	A	B
10	05	05	201000	201002

	Before	After
1000	F1xxxx	F1F1F1F1F1F1
1002	xxxxxx	F1F1F1F1F1F1
Comparison	xx	Equal
Overflow	xx	unchanged

MOVE NUMERIC (MVN)

OP	AF	BF	A Address	B Address

OP = 11

AF = Length of the A field. Can be indirect or a literal flag.
BF = Length of the B field. Can be indirect.
A = Address of the A field (source).
B = Address of the B field (destination).
The A and B Field addresses can be indexed, indirect or extended.

An MVN instruction moves digits or characters (as specified by the address controllers) from the A field and stores them at the location specified by the B Address, right-justified.

Example 1: Move a UN field with an Overflow condition.

OP	AF	BF	A	B
11	05	03	001000	002000

	Before	After
1000	12300	12300
2000	xxx	unchanged
Comparison	xx	unchanged
Overflow	xx	On

Example 2: Move UN field to shorter UN field.

OP	AF	BF	A	B
11	05	03	001000	002000

	Before	After
1000	00123	00123
2000	xxx	123
Comparison	xx	HIGH
Overflow	xx	unchanged

Example 3: Move SN field to a longer SN field.

OP	AF	BF	A	B
11	03	05	101000	102000

	Before	After
1000	C123	C123
2000	xxxxxx	C00123
Comparison	xx	HIGH
Overflow	xx	unchanged

Example 4: Move UA Field to a UA Field.

OP	AF	BF	A	B
11	03	03	201000	202000

	Before	After
1000	C7E8D9	C7E8D9
2000	xxxxxx	F7F8F9
Comparison	xx	HIGH
Overflow	xx	unchanged

Example 5: Move digit repeat.

OP	AF	BF	A	B
11	05	05	001000	001001

	Before	After
1000	5xxxxx	555555
Comparison	xx	HIGH
Overflow	xx	unchanged

MOVE WORDS (MVW)

OP	AFBF	A Address	B Address

OP = 12

AFBF = The length of both the A and B fields. The AFBF can specify indirect field length; however, a literal flag is ignored. A value of 0000 is equal to a length of 10,000 words (40,000 digits).

A = Address of the A field (source). The address controllers are ignored.

B = Address of the B field (destination). The address controllers are ignored.

The A and B Field addresses can be indexed, indirect or extended.

An MVW instruction moves words from the A Field to the B Field. Both addresses must be mod 4; however, only the B Address is checked for validity. The number of words moved is specified by the combination of AF and BF; 0000 specifies 10,000 words, the maximum permissible. Both address controllers must specify UN or indirect addressing; the final address must specify UN.

Example 1: Move Eight Digits

OP	AFBF	A	B
12	0002	A Field,	B Field

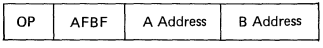
	Before	After
A Field	01020304	01020304
B Field	xxxxxxxx	01020304
Comparison	xxx	unchanged
Overflow	xxx	unchanged

Example 2: Repeat Data Field

OP	AFBF	A	B
12	0002	A Field (UN)	A Field+4(UN)

	Before	After
A Field	0123xxxxxxxx	012301230123
Comparison	xxx	unchanged
Overflow	xxx	unchanged

MOVE AND CLEAR WORDS (MVC)



OP = 13

AFBF = Length of both operands. A value of 0000 is equal to a length of 10,000 words or 40,000 digits. AFBF can specify indirect field length but a literal causes an Address Error.

A = Address of the source data field operand. Address can be indexed, indirect or extended. Final address controllers are ignored.

B = Address of the destination data field operand. Address can be indexed, indirect or extended. Final address controllers are ignored.

An MVC instruction moves words from the A Field to the B Field, clearing each A field location. Both addresses must be mod 4. The number of words moved is specified by the combination of AF and BF; 0000 specifies 10,000 words, the maximum permissible. Both address controllers must specify UN or indirect addressing; in the latter case, the final address must specify UN.

Example 1: Move Eight Digits and Clear the Source Field

OP	AFBF	A	B
13	0002	A FIELD (UN)	B FIELD (UN)

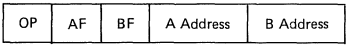
	Before	After
A Field	F1F2F3F4	00000000
B Field	xxxxxxxx	F1F2F3F4
Comparison	xxx	unchanged
Overflow	xxx	unchanged

Example 2: Justify Data Field

OP	AFBF	A	B
13	0002	A Field (UN)	A Field+4(UN)

	Before	After
A Field	1605xxxxxxxx	000000001605
Comparison	xxx	unchanged
Overflow	xxx	unchanged

MOVE REPEAT (MVR)



OP = 14

AF = Length of A operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the A address controller). AF can be an indirect or a literal flag.

BF = Number of repetitions. A value of 00 is equal to 100 repetitions. BF can be an indirect flag.

A = Address of the source data field operand. Address can be indexed, indirect or extended. The final address controller must specify UN or UA. An SN controller is treated as UN.

B = Address of the destination data field operand. Address can be indexed, indirect or extended. The final address controller must specify UN or UA. An SN controller is treated as UN.

An MVR instruction moves characters or digits from the A field to the B field. AF specifies the number of digits or characters to be moved, and BF specifies the number of times they are to be moved. Each move causes the B Address to be incremented by AF if both fields are UN, or by two times AF if either or both fields are UA. The length of the B Field is specified by the product of AF times BF if both fields are UN, or by the product of twice AF times BF if either or both fields are UA. Both address controllers can specify either UN, UA, or indirect addressing.

Example 1: Repeat a 3 digit Numeric Field 4 Times

OP	AF	BF	A	B
14	03	04	A Field (UN)	B Field (UN)

	Before	After
A Field	057	057
B Field	xxxxxxxxxxx	057057057057
Comparison	xxx	unchanged
Overflow	xxx	unchanged

Example 2: Repeat a 3 Character Alpha Field Twice in a Numeric Field

OP	AF	BF	A	B
14	03	02	A Field (UA)	B Field (UN)

	Before	After
A Field	D4D5D6	D4D5D6
B Field	xxxxxx	456456
Comparison	xxx	unchanged
Overflow	xxx	unchanged

TRANSLATE (TRN)

OP	AFBF	A Address	B Address	C Address

OP = 15

AFBF = Number of digits or characters to be translated. A value of 0000 is equal to a length of 10,000 units. AFBF can specify indirect field length or a literal.

A = Address of the source data field to be translated. The address can be indexed, indirect or extended. Address controller use is unrestricted; however, the sign of an SN field is ignored. If the

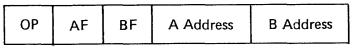
format is UN or SN, the EBCDIC numeric subset zone (F) is assumed before it is translated.

B = Address of the translate table. Address can be indexed, indirect or extended. The final address controllers are ignored.

C = Address of the destination data field operand. The address can be indexed, indirect or extended. The final address controller must be UA or UN. If the final address controller is UN, only the digit portion of each translated character is stored. If the final address controller is SN, the instruction will set the Invalid Instruction result condition.

A TRN instruction substitutes a digit or character in the B field for each digit or character in the A field and moves that substituted character to the C field. The B field address is modified by each A field digit or character so that all identical A field digits or characters cause the same B field location to be accessed. The translation table in the B field must be selected and in place prior to execution. Any code that does not exceed eight bits can be translated.

SCAN TO DELIMITER EQUAL (SDE)



OP = 16

AF = Length of A operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the A address controller). AF can be an indirect or a literal flag.

BF = Length of B operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the B address controller). BF can be an indirect flag.

A = Address of the delimiter list field. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, a zone digit in the EBCDIC numeric subset (F) is added to each digit before comparison. An SN controller is treated as UN.

B = Address of the data field to be scanned. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, a zone digit in the EBCDIC numeric subset (F) is added to each digit before comparison. An SN controller is treated as UN.

This instruction searches the B field for a digit or character equal to one of the delimiter digits or characters in the A field.

Example 1: Scan Delimiter-Equal, First digit Equal

OP	AF	BF	A	B
16	01	04	A Field (UN)	B Field (UA)

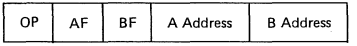
	Before	After
A Field	1	1
B Field	F1C8C4D9	F1C8C4D9
0000038	xx	00
Comparison	xxx	LOW

Example 2: Scan Delimiter-Equal, Other Than First digit equal

OP	AF	BF	A	B
16	01	04	A Field (UA)	B Field (UA)

	Before	After
A Field	E7	E7
B Field	C1C2E7F5	C1C2E7F5
0000038	xx	02
Comparison	xxx	Equal

SCAN TO DELIMITER UNEQUAL (SDU)



OP = 17

AF = Length of A operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the A address controller). AF can be an indirect or a literal flag.

BF = Length of B operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the B address controller). BF can be an indirect flag.

A = Address of the delimiter list field. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, a zone digit in the EBCDIC numeric subset (F) is added to each digit before comparison. An SN controller is treated as UN.

B = Address of the data field to be scanned. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, a zone digit in the EBCDIC numeric subset (F) is added to each digit before comparison. An SN controller is treated as UN.

An SDU instruction searches the B field for a character unequal to one of the delimiter characters in the A field.

Example 1: Scan Delimiter-Unequal, First Digit Unequal.

OP	AF	BF	A	B
17	03	04	A Field (UN)	B Field (UN)

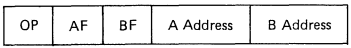
	Before	After
A Field	123	123
B Field	6123	6123
0000038	xx	00
Comparison	xxx	LOW

Example 2: Scan Delimiter-Unequal, Other Than First Digit Unequal.

OP	AF	BF	A	B
17	03	04	A Field (UA)	B Field (UA)

	Before	After
A Field	C1C2C3	C1C2C3
B Field	C1C2C3C4	C1C2C3C4
0000038	xx	03
Comparison	xxx	Equal

SCAN TO DELIMITER ZONE EQUAL (SZE)



OP = 18

AF = Length of A operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the A address controller). AF can be an indirect or a literal flag.

BF = Length of B operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the B address controller). BF can be an indirect flag.

A = Address of the delimiter list field. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, an F is compared with each zone digit of the B Field. A SN controller is treated as UN.

B = Address of the data field to be scanned. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, an F is compared with each zone digit of the A Field. A SN controller is treated as UN.

An SZE instruction is the same as an OP I6, except that 1) address controllers can specify only UA or indirect addressing (final address must specify UA) and 2) the low order digit of each character is ignored. The instruction compares only the zone bits of the characters.

Example 1: Scan Delimiter-Zone Equal, First Zone Equal

OP	AF	BF	A	B
18	02	03	A Field (UA)	B Field (UA)

	Before	After
A Field	C1D1	C1D1
B Field	D2E6C1	D2E6C1
0000038	xx	00
Comparison	xxx	LOW

Example 2: Scan Delimiter-Zone Equal, Other Than First Zone Equal

OP	AF	BF	A	B
18	02	04	A Field (UA)	B Field (UA)

	Before	After
A Field	C1D1	C1D1
B Field	E6D2C1D4	E6D2C1D4
0000038	xx	01
Comparison	xxx	Equal

Example 3: Scan Delimiter-Zone equal, No Zones Equal

OP	AF	BF	A	B
18	04	04	A Field (UA)	B Field (UA)

	Before	After
A Field	F160C1D1	F160C1D1
B Field	E6E7E8E9	E6E7E8E9
0000038	xx	03
Comparison	xxx	HIGH

SCAN TO DELIMITER ZONE UNEQUAL (SZU)

OP	AF	BF	A Address	B Address

OP = 19

AF = Length of A operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the A address controller). AF can be an indirect or a literal flag.

BF = Length of B operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the B address controller). BF can be an indirect flag.

A = Address of the delimiter list field. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, an F is compared with each zone digit of the B Field. A SN controller is treated as UN.

B = Address of the data field to be scanned. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN is specified, an F is compared with each zone digit of the C Field. A SN controller is treated as UN.

An SZU instruction is the same as an OP 17, except that 1) address controllers can specify only UA or indirect addressing (final address must specify UA) and 2) the low order digit of each character is ignored. The instruction compares only the zone bits of the data characters to the zone bits of the delimiter characters for inequality.

Example 1: Scan Delimiter-Zone Unequal, First Zone Unequal

OP	AF	BF	A	B
19	01	04	A Field (UA)	B Field (UA)

	Before	After
A Field	C1	C1
B Field	D1C1C2E7	D1C1C2E7
0000038	xx	00
Comparison	xxx	LOW

Example 2: Scan Delimiter-Zone Unequal, Other Than First Zone Unequal

OP	AF	BF	A	B
19	02	04	A Field (UA)	B Field (UA)

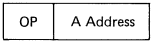
	Before	After
A Field	C1D1	C1D1
B Field	C1C2E7C3	C1C2E7C3
0000038	xx	02
Comparison	xxx	Equal

Example 3: Scan Delimiter-Zone Unequal, No Zones Unequal

OP	AF	BF	A	B
19	02	04	A Field (UA)	B Field (UA)

	Before	After
A Field	C1D1	C1D1
B Field	C3C4D4D6	C3C4D4D6
0000038	xx	03
Comparison	xxx	HIGH

BRANCH



OP = 20 through 29

A = Branch Address. Address can be indexed, indirect or extended. When not extended the final address controller bits specify the most significant digit of the address. This permits branching to any address up to and including 299,998, base relative, without indexing or extension. When the address is extended, the final address controller bits are ignored.

No Operation (NOP)

OP = 20

This instruction performs no significant action; the next instruction is fetched. The comparison indicators are not affected by NOP.

Branch Less Than (LSS)

OP = 21

An LSS branch instruction causes a branch to the A Address if the comparison indicators are low. If the comparison indicators are not low, this instruction acts as a NOP. The comparison indicators are unchanged.

Branch Equal (EQL)

OP = 22

An EQL branch instruction causes a branch to the A Address if the comparison indicators are equal. If the comparison indicators are not equal, this instruction acts as a NOP. The comparison indicators are unchanged.

Branch Less Than or Equal (LEQ)

OP = 23

An LEQ branch instruction causes a branch to the A Address if the comparison indicators are low or equal. If the comparison indicators are high, this instruction acts as a NOP. The comparison indicators are unchanged.

Branch Greater Than (GTR)

OP = 24

A GTR branch instruction causes a branch to the A Address if the comparison indicators are high. If the comparison indicators are not high, this instruction acts as an NOP. The comparison indicators are unchanged.

Branch Not Equal (NEQ)

OP = 25

An NEQ branch instruction causes a branch to the A Address if the comparison indicators are not set to equal. If the comparison indicators are equal, this instruction acts as an NOP. The comparison indicators are unchanged.

Branch Greater Than or Equal (GEQ)

OP = 26

A GEQ branch instruction causes a branch to the A Address if the comparison indicators are high or equal. If the indicators are low, this instruction acts as an NOP. The comparison indicators are unchanged.

Branch Unconditional (BUN)

OP = 27

A BUN branch instruction always causes a branch to the A Address. The comparison indicators are unchanged.

Branch Overflow (OFL)

OP = 28

An OFL branch instruction causes a branch to the A Address if the Overflow Flip/Flop is set; otherwise, it acts as an NOP. The Overflow Flip/Flop is reset by this instruction, and the comparison indicators are unchanged.

Halt Branch (HBR)

OP = 29

An HBR branch instruction causes a branch to the A Address, subject to the control imposed by the halt execution digit at address 77, absolute, and the state of the Normal Flip/Flop. The comparison indicators are unchanged by the HBR instruction.

BRANCH – COMMUNICATE (BCT)

OP	AFBF

OP = 30

AFBF = is the Communicate Address which is the low order four digits of an absolute machine address. The high order digits are equal to Zero. Indirect field length can be specified.

Typically the Branch Communicate instruction is used to branch from an object program to a control program pointer located at the Communicate Address. The control program pointer must contain a hex F as the most significant digit if the instruction is to be valid. The following processor information is stored in absolute memory locations 61 through 76.

61	62	63 THROUGH 69	70	71	72	73	74	75	76
DIGIT BASE 7	DIGIT LIMIT 7	NEXT PROGRAM INSTRUCTION ADDRESS	BASE DIGITS 6	5	4	LIMIT DIGITS 6	5	4	T* OGGLES

*Toggles: Bit 8 Unused
Bit 4 Overflow
Bit 2 Comparison LOW
Bit 1 Comparison HIGH

ENTER (NTR)

OP	AFBF	A Address	Parameters

OP = 31

AFBF = Length, in characters, of the parameter field. The maximum number of characters moved is 9,999. A length of 0000 moves no characters. AFBF can specify indirect field length but a literal will be ignored.

A = Branch address which can be indexed, indirect or extended. When not extended, the final address controller bits specify the most significant digit of the address. This permits branching to any address, up to and including 299,998 base relative, without indexing or extension. When the final address is extended, the address controller bits are ignored. When indexed by IX3, the initial contents of IX3 are used.

Parameters = Data that is to be stored in the subroutine stack. The enter instruction stores control information and parameters into a stack located in memory and then executes an unconditional branch to the instruction at the A Address.

An NTR instruction is used to branch to a subroutine; and, prior to branching, to store information required to re-establish processor conditions upon the return from the subroutine. A subroutine is a portion of a program which can be used to perform a particular function at several locations within a program. A subroutine can be entered from any predetermined point in a program by the execution of the Enter (NTR) instruction. The exiting of a subroutine is accomplished by the Exit (EXT) instruction, which has the ability to exit to either the instruction following the Enter instruction or to any other designated instruction in the program.

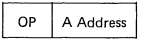
Example 1: Enter

Address	OP	AFBF	A	Parameters
003016	31	0003	020166	203010

	Before	After	
NI	003016	020166	
IX3	+0000010	+0001024	
0000040	001024	001046	Top of Stack
0001024	xxxxxx	003034	Instruction Address After Parameters
		+0000010	Value of IX3
		0	Zero Digit
		5	Value of Condition Toggles
		203010	Parameters

	Before	After
Comparison	HIGH	Clear
Overflow	On	Off

EXIT (EXT)



OP = 32

A = Return address which can be indexed, indirect or extended. When not extended, the final controller bits specify the most significant digit of the address. This permits branching to any address, up to and including 299,998 base relative, without indexing or extension. When the final address is extended, the address controller bits are ignored. The normal return address is obtained by setting the A Address to zero, indexing from IX3, and setting the address controller to indirect. When the address is indexed from IX3, the initial contents of IX3 is used.

Example 1: Exit the stack.

OP	A
32	F00000

	Before	After	
NI	xxxxxx	003034	
IX3	+0001024	+0000010	
0000040	001046	001024	
0001024	003034	unchanged	Stack
0001030	+0000010	unchanged	Stack
0001038	06	unchanged	Stack
0001040	203010	unchanged	Stack
Comparison	nnn	LOW	
Overflow	nn	On	

BIT RESET (BRT)

OP	AF	BF	A Address

OP = 33

AF = Length of the A data field. May be indirect or literal. A value of 00 is equal to a length of 100 units.

BF = 8-bit selection mask. One bits in this mask select those bit positions to be set to zero bits within each 8-bit group of the A data field. A through F can be used to specify undigits in the mask. The field is not recognized as indirect.

A = Address of the data field operand. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN format is specified and the number of digits accessed is even, the entire 8-bit mask is applied to successive groups of two digits. If the number of digits is odd, the operation is the same until the last digit is accessed. The most significant four bits of the mask are applied to this digit. If the controller specifies SN, the instruction will set Invalid Instruction.

The purpose of the BRT instruction is to reset any bit in the A field if the corresponding bit in the mask in BF is on. AF specifies the length of the A field which can be up to 100 digits or characters long. The address controller can specify UN, UA, or indirect address, with the final address controller being UN or UA.

Example 1: Bit Reset, Alpha Field

OP	**AF**	**BF**	**A**
33	03	A0	A Field (UA)

	Data	**Binary Value**
A Field	F1F2F3	111100011111001011110011
Mask	A0A0A0	101000001010000010100000
Result	515253	010100010101001001010011
Comparison	HIGH	

Example 2: Bit Reset, Numeric Field

OP	**AF**	**BF**	**A**
33	05	15	A Field (UN)

	Data	Binary Value
A Field	43105	01000011000100000101
Mask	15151	00010101000101010001
Result	42004	01000010000000000100
Comparison	Equal	

BIT SET (BST)

OP	AF	BF	A Address

OP = 34

AF = Length of the A data field. May be indirect or literal. A value of 00 is equal to a length of 100 units.

BF = Eight bit selection mask. One bits in this mask select those bit positions to be set to one bits within each eight bit group of the A data field. A through F can be used to specify undigits in the mask. The field is not recognized as indirect.

A = Address of the data field operand. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN format is specified and the number of digits accessed is even, the entire 8-bit mask is applied to successive groups of two digits. If the number of digits is odd, the operation is the same until the last digit is accessed. The most significant four bits of the mask are applied to this digit. If the controller specifies SN, the instruction is set Invalid Instruction.

The purpose of BST is to set any bit in the A field if the corresponding bit in the mask in BF is ON. AF specifies the length of the A field, which can be up to 100 digits or characters. The address controller can specify UN, UA, or indirect address; the final address controller must specify UN or UA.

Example 1: Bit Set, Alpha Field

OP	AF	BF	A
34	03	A0	A Field (UA)

	Data	Binary Value
A Field	515253	010100010101001001010011
Mask	A0A0A0	101000001010000010100000
Result	F1F2F3	111100011111001011110011
Comparison	HIGH	

Example 2: Bit Set, Numeric Field

OP	AF	BF	A
34	05	F1	A Field (UN)

	Data	Binary Value
A Field	94236	1001010000100011 0110
Mask	F1F1F	1111000111110001 1111
Result	F5F3F	1111010111110011 1111
Comparison	HIGH	

SEARCH LINK LIST (SLL)

OP	AF	BF	A Address	B Address

OP = 37

AF = Length of the A data field. May be indirect or literal flag. A value of 00 is equal to a length of 100 units.

BF = Amount of offset in units from the B Address to the field to be searched. BF is typically six digits or more to allow for the link address at B. A value of 00 is equal to a length of 100 units. BF can be indirect.

A = Address of the key to which the B data field is compared. Address can be indexed, indirect or extended. The final address controller specifies the format for both the A and B Fields and must be UN or UA. If the address controller specifies SN, Invalid Instruction is set.

B = Address of the first list entry. The initial address can be indexed or extended. The data format is that of the final A address controller. The B address controller bits determine the type of comparison to be made.

B Address Controller		Search Type
00 (UN)	=	Search Equal
01 (SN)	=	Any Bit Equal
10 (UA)	=	Less Than or Equal To
11 (IA)	=	No Bit Equal

An SLL instruction compares data in the A Field with data in the B Field, as controlled by the AF/BF variants and the AC/BC controllers.

Example 1: Search Equal

OP	AF	BF	A	B
37	05	06	A Field (UN)	B-Field (UN)

	Before	After
A Field	12345	12345
B Field	00400012345	00400012345
IX1	xxxxxxxx	B-Field Address
Comparison	xxx	Equal

Example 2: Search Any Bit Equal, None Found

OP	AF	BF	A	B
37	01	06	A Field (UN)	B-Field (SN)

	Before	After
A Field	6	6
B Field	0000009	0000009
IX1	xxxxxxxx	unchanged
Comparison	xxx	HIGH

Example 3: Search Less than or Equal To

OP	AF	BF	A	B
37	05	06	A Field (UN)	B-Field (UA)

	Before	After
A Field	12345	12345
B Field	00400012345	00400012345
IX1	xxxxxxxx	B-Field Address
Comparison	xxx	Equal

Example 4: Search No Bit Equal

OP	AF	BF	A	B
37	01	06	A-Field (UN)	B-Field (IA)

	Before	After
A Field	6	6
B Field	0000009	0000009
IX1	xxxxxxxx	B-Field Address
Comparison	xxx	Equal

SEARCH LINK DELINK (SLD)

OP	AF	BF	A Address	B Address

OP = 38

AF = Length of the A data field. May be indirect or literal flag. A value of 00 is equal to a length of 100 units.

BF = Amount of offset in units from the B address to the field to be searched. BF is typically six digits or more to allow for the link address at B. A value of 00 is equal to a length of 100 units. BF can be indirect.

A = Address of the key to which the B data field is compared. Address can be indexed, indirect or extended. The final address controller specifies the format for both the "A' and "B' fields and must be UN or UA. If SN is specified, Invalid Instruction is set.

B = Address of the first list entry. The initial address may be indexed or extended. The data format is that of the final A address controller. The B address controller bits determine the type of comparison to be made.

Address Controller	Search Type
00 (UN)	= Search Equal.
01 (SN)	= Any Bit Equal.

The comparison toggles are set EQUAL when the entire A key field is equal to the B data field.

The comparison toggles are set EQUAL when any 1-bit of the A key field is equal to the corresponding bit of the B data field.

An SLD instruction is identical to Search Link List (OP 37) except that when a comparison condition is met and the B Address is stored in IX1, the previous B address is placed into IX2.

Example 1: Search Equal

OP	AF	BF	A	B
38	05	06	A Field (UN)	B Field (UN)

	Before	After
A Field	12345	12345
B Field	00400012345	00400012345
004000	00500012345	00500012345
IX1	xxxxxxxx	B-Field Address
IX2	xxxxxxxx	B-Field Address
Comparison	xxx	Equal

Example 2: Search Any Bit Equal, None Found

OP	AF	BF	A	B
38	01	06	A Field (UN)	B Field (SN)

	Before	After
A Field	6	6
B Field	0000009	0000009
IX1	xxxxxxxx	unchanged
IX2	xxxxxxxx	unchanged
Comparison	xxx	HIGH

Example 3: Search Less Than or Equal To

OP	AF	BF	A	B
38	05	06	A Field (UN)	B Field (UA)

	Before	After
A Field	12345	12345
B Field	00400002345	00400002345
004000	00500012345	00500012345
IX1	xxxxxxxx	B-Field Address
IX2	xxxxxxxx	B-Field Address
Comparison	xxx	Equal

Example 4: Search No Bit Equal

OP	AF	BF	A	B
38	01	06	A Field (UN)	B Field (IA)

	Before	After
A Field	6	6
B Field	0040006	0040006
004000	0050009	0050009
IX1	xxxxxxxx	00004000
IX2	xxxxxxxx	B-Field Address
Comparison	xxx	Equal

SEARCH (SEA)

OP	AF	BF	A Address	B Address	C Address

OP = 39

AF = Number of digits or characters, depending on the A address controller, to be compared between the two data fields. A value of 00 is equal to a length of 100 units. AF can be indirect or a literal flag.

BF = Number of digits or characters, depending on the B address controller, that the B Address is incremented between comparisons. This value is independent of the AF value and can be less than, equal to, or greater than the AF value. A value of 00 is equal to a length of 100 units. BF can be indirect.

A = Address of the key field operand. Address can be indexed, indirect or extended. The final address controller specifies the format for both the A and B Fields. Standard address controller usage applies.

B = Address of the first entry to be compared against the key. Address can be indexed, indirect or extended. The data format is that of the A Address controller. The B address controller bits determine the incrementation between comparisons.

B Address Controller	Increment in Digits
00 (UN)	BF
01 (SN)	BF+1
10 (UA)	2BF

C = Address of the maximum limit for the incremented value of the B Address. Address can be indexed, indirect or extended. When the incremented value of the B address equals or exceeds the value of the C Address, the instruction is terminated. A comparison of the full length of the key always takes place even if it exceeds the C Address. The final address controller specifies the type of search performed.

C Address Controller		Search Type
00 (UN)	=	Search Equal
01 (SN)	=	Search Low
10 (UA)	=	Search Lowest

An SEA instruction compares the A field with portions of the B field (modified by BF) according to CC. After each comparison, the B address is incremented by the value of BF, as modified by BC, until the search conditions are met or until the incremented B address is equal to the C Address.

Example 1: Search Equal

OP	AF	BF	A	B	C
39	01	02	A Field (UA)	1000 (UA)	1020 (UN)

	Before	After
A Field	C1	unchanged
B Field	C1F1C2F2C3F3C4F2C5F1	unchanged
IX1	xxxxxxxx	+0001000
Comparison	xxx	Equal
Overflow	xxx	Off

Example 2: Search Low, Condition Not Found

OP	AF	BF	A	B	C
39	01	01	A Field (UN)	1000 (UN)	1020 (SN)

	Before	After
A Field	2	unchanged
B Field	3459876345	unchanged
IX1	xxxxxxxx	unchanged
Comparison	xxx	HIGH
Overflow	xxx	Off

Example 3: Search Lowest

OP	AF	BF	A	B	C
39	01	01	A Field (UA)	1000 (UA)	1020 (UA)

	Before	After
A Field	C5	unchanged
B Field	C5C2C3C4C9C3C1E2C3C9	unchanged
IX1	xxxxxxxx	+0001012 (Points to letter A)
Comparison	xxx	Equal
Overflow	xxx	Off

BIT ZERO TEST (BZT)

OP = 40

AF = Length of the A data field. May be indirect or literal. A value of 00 is equal to a length of 100 units.

BF = Eight bit selection mask. One bits in this mask select those bit positions to be tested for zero bits within each eight bit group of the A data field. A through F can be used to specify undigits in the mask. The field is not recognized as indirect.

A = Address of the data field to be examined. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN format is specified and the number of digits accessed is even, the entire 8-bit mask is applied to successive groups of two digits. If the number of digits is odd, the operation is the same until the last digit is accessed. The most significant four bits of the mask are applied to this digit. If SN is specified, the Invalid Instruction is set.

The purpose of the BZT instruction is to test information in the A field with a two-digit mask (BF) and, if any A field Zero bits are found corresponding to 1-bits in the mask, to set the comparison indicators to equal. AC can specify UN or indirect address. If SN is specified, it is treated as UN and invalid instruction is set. If no comparison is made, the comparison indicators are set to high. AF specifies the length of the A field.

Example 1: Zero Test-Zero Found

OP	AF	BF	A
40	04	C0	A Field (UA)

	Data	Binary Value
A Field	C3C1E77BC4	1100001111000001111001111011
Mask	C0C0C0C0C0	1100000011000000110000011000000
Hit	1	1
Comparison	Equal	

Example 2: Zero Test-All Ones Found

OP	AF	BF	A
40	04	C0	A Field (UA)

	Data	Binary Value
A Field	C2D9C1C3C5	1100001011011001110000111000011
Mask	C0C0C0C0C0	1100000011000000110000011000000
Hit Comparison	HIGH	

BIT ONE TEST (BOT)

OP	AF	BF	A Address

OP = 41

AF = Length of the A data field. May be indirect or literal. A value of 00 is equal to a length of 100 units.

BF = Eight bit selection mask. One bits in this mask select those bit positions to be tested for one bits within each eight bit group of the A data field. A through F can be used to specify undigits in the mask. The field is not recognized as indirect.

A = Address of the data field to be examined. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If UN format is specified and the number of digits accessed is even, the entire 8-bit mask is applied to successive groups of two digits. If the number of digits is odd, the operation is the same until the last digit is accessed. The most significant four bits of the mask are applied to this digit. If SN is specified, the Invalid Instruction is set.

The purpose of the BOT instruction is to test information in the A field with two-digit mask (BF). If any A field 1-bits are found corresponding to 1-bits in the mask, this instruction sets the comparison indicators to equal. If no comparison is made, the indicator is set to high. The length of the A field is specified by AF. AC can specify UN, UA, or indirect addressing. If SN is specified, it is treated as UN but invalid instruction is set.

Example 1: Ones Test-One Found

OP	AF	BF	A
41	03	F0	A Field (UN)

	Data	Binary Value
A Field	001	0000 0000 0001
Mask	F0F	1111 0000 1111
Hit	1	1
Comparison	Equal	

Example 2: Ones Test-All Zeros Found

OP	AF	BF	A
41	02	03	A Field (UA)

	Data	Binary Value
A Field	C4C4	1100010011000100
Mask	0303	0000001100000011
Hit Comparison	HIGH	

LOGICAL AND (AND)

OP	AFBF	A Address	B Address	C Address

OP = 42

AF = Length of the A data field. May be indirect or literal. A value of 00 is equal to a length of 100 units.
BF = Length of the B data field. May be indirect. A value of 00 is equal to a length of 100 units.

NOTE

The field length of the C operand is equal to the larger of AF or BF. If the A and B Fields are not of equal length, the shorter is padded by adding trailing characters/digits of all zero bits.

A = Address of the A source data field operand. Address can be indexed, indirect or extended. The final address controller must be UN or UA and is used for all three addresses.
B = Address of the B source data field operand. Address can be indexed, indirect or extended. The final address controller is ignored.
C = Address of the result data field operand. Address can be indexed, indirect or extended. The final address controller is ignored.

An AND instruction places the logical products of the A and B fields into the C field. If a bit is set in the A field, and a corresponding bit is set in the B field, the C field bit is set. AF and BF specify the A and B field lengths, respectively. Maximum field length is 100 digits or characters, and is specified by 00. The length of the C field is the longer of AF or BF. If AF is not equal to BF, the shorter field is assumed to be filled with trailing zeros.

Example 1: And Two Numeric Fields

OP	AF	BF	A	B	C
42	02	03	A Field (UN)	B Field (UN)	C Field (UN)

	Before	After	Binary Value
A Field	F6	F6	111101100000
B Field	235	235	001000110101
C Field	xxx	220	001000100000
Comparison	xxx	Equal	

Example 2: And Two Alpha Fields

OP	AF	BF	A	B	C
42	02	03	A Field (UA)	B Field (UA)	C Field (UA)

	Before	After	Binary Value
A Field	E7E8	E7E8	111001111110100000000000
B Field	D4D8D1	D4D8D1	110101001101100011010001
C Field	xxx	C4C800	110001001100100000000000
Comparison	xxx	Equal	

LOGICAL OR (ORR)

OP	AFBF	A Address	B Address	C Address

OP = 43

AF = Length of the A data field. May be indirect or literal. A value of 00 is equal to a length of 100 units.

BF = Length of the B data field. May be indirect. A value of 00 is equal to a length of 100 units.

NOTE

The field length of the C operand is equal to the larger of AF or BF. If the A and B Fields are not of equal length, the shorter is padded by adding trailing characters/digits of all zero bits.

A = Address of the A source data field operand. Address can be indexed, indirect or extended. The final address controller must be UN or UA and is used for all three addresses.

B = Address of the B source data field operand. Address can be indexed, indirect or extended. The final address controller is ignored.

C = Address of the result data field operand. Address can be indexed, indirect or extended. The final address controller is ignored.

An ORR instruction is identical to the Logical And except that it performs the OR function and places the logical sum in the C field.

Example 1: Or Two Numeric Fields

OP	AF	BF	A	B	C
43	02	03	A Field (UN)	B Field (UN)	C Field (UN)

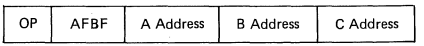
	Before	After	Binary Value
A Field	81	81	100000010000
B Field	223	223	001000100011
C Field	xxx	A33	101000110011
Comparison	xxx	HIGH	

Example 2: Or Two Alpha Fields

OP	AF	BF	A	B	C
43	03	02	A Field (UA)	B Field (UA)	C Field (UA)

	Before	After	Binary Value
A Field	C1C2C4	C1C2C4	110000011100001011000100
B Field	F2F3	F2F3	111100101111001100000000
C Field	xxx	F3F3C4	111100111111001111000100
Comparison	xxx	Equal	

LOGICAL NOT (NOT)



OP = 44

AF = Length of the A data field. May be indirect or literal. A value of 00 is equal to a length of 100 units.

BF = Length of the B data field. May be indirect. A value of 00 is equal to a length of 100 units.

NOTE

The field length of the C operand is equal to the larger of AF or BF. If the A and B Fields are not of equal length, the shorter is padded by adding trailing characters/digits of all one bits.

A = Address of the A source data field operand. Address can be indexed, indirect or extended. The final address controller must be UN or UA and is used for all three addresses.

B = Address of the B source data field operand. Address can be indexed, indirect or extended. The final address controller is ignored.

C = Address of the result data field operand. Address can be indexed, indirect or extended. The final address controller is ignored.

An NOT instruction is identical to the Logical And except that it performs the XOR function and places the result in the C field. If AF is not equal to BF, the shorter field is assumed to be filled with trailing ones.

Example 1: Exclusive Or of Two Numeric Fields-NOT Function

OP	AF	BF	A	B	C
44	03	03	A Field (UN)	B Field (UN)	C Field (UN)

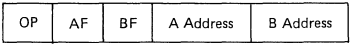
	Before	After	Binary Value
A Field	FFF	FFF	111111111111
B Field	6A1	6A1	011010100001
C Field	xxx	95E	100101011110
Comparison	xxx	Equal	

Example 2: Exclusive Or of Two Alpha Fields

OP	AF	BF	A	B	C
44	02	02	A Field (UA)	B Field (UA)	C Field (UA)

	Before	After	Binary Value
A Field	5050	5050	0101000001010000
B Field	C7D7	C7D7	1100011111010111
C Field	xxx	9787	1001011110000111
Comparison	xxx	HIGH	

COMPARE ALPHANUMERIC (CPA)



OP = 45

AF = Length of A operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the A address controller). AF can be an indirect or a literal flag.

BF = Length of B operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the B address controller). BF can be an indirect flag.

NOTE

If the field lengths are unequal and the data type is UA, the shorter field is padded with trailing blanks (40) to equal the length of the longer field. If the data type is UN the shorter field is padded with trailing zeros.

A = Address of the A data field operand. Address can be indexed, indirect or extended. The final address controller must be UN or UA. An SN controller is treated as UN.

B = Address of the B data field operand. Address can be indexed, indirect or extended. The final address controller must be the same as the A Address controller. If it is not, the Invalid Instruction result is set. An SN controller is treated as UN.

A CPA instruction compares the A field to the B field and sets the comparison indicators to high (A > B), equal (A = B), or low (A < B). AF and BF specify the length of the A and B fields, respectively. Address controllers can specify UN, SN, UA, or indirect addressing. If UA is specified, only the least significant digit of each character is compared. If AF is not equal to BF, the shorter field is assumed to be filled with trailing blanks (40).

Example 1: Compare Two Alpha Data Fields

OP	AF	BF	A	B
45	05	03	A Field (UA)	B Field (UA)

	Before	After
A Field	C1E3E24040	unchanged
B Field	C1E3E2	unchanged
Comparison	xxx	Equal

Example 2: Compare Two Alpha Data Fields

OP	AF	BF	A	B
45	02	02	A Field (UA)	B Field (UA)

	Before	After
A Field	C1D5	unchanged
B Field	C2D5	unchanged
Comparison	xxx	LOW

COMPARE NUMERIC (CPN)

OP	AF	BF	A Address	B Address

OP = 46

AF = Length of A operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the A address controller). AF can be an indirect or a literal flag.

BF = Length of B operand. A value of 00 is equal to a length of 100 units (digits or characters as specified by the B address controller). BF can be an indirect flag.

NOTE

If the field lengths are unequal, the shorter field is padded with leading zeros to equal the length of the longer field. The length does not include the sign digit of a signed numeric (SN) field.

A = Address of the A data field operand. Address can be indexed, indirect or extended. Full address controller usage applies.

B = Address of the B data field operand. Address can be indexed, indirect or extended. Full address controller usage applies.

A CPN instruction compares the A field to the B field and sets the comparison indicators to high (A > B), equal (A = B), or low (A < B). Maximum length is 100 digits, specified by 00. Address controllers can specify UN, SN, UA, or indirect addressing. If AF is not equal to BF, the shorter field is assumed to be filled with leading zeros.

Example 1: Compare a Signed Literal Field with an Unsigned Field

OP	AF	BF	A	B
46	AA	05	+20 (SL)	B Field (UN)

	Before	After
A Field	+20	unchanged
B Field	00015	unchanged
Comparison	xxx	HIGH

Example 2: Compare a Numeric Literal Field with a Signed Field

OP	AF	BF	A	B
46	A6	02	000012 (NL)	B Field (SN)

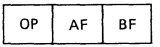
	Before	After
A Field	000012	unchanged
B Field	+25	unchanged
Comparison	xxx	LOW

Example 3: Compare a Numeric Field with an Alpha Field

OP	AF	BF	A	B
46	03	03	A Field (UN)	B Field (UA)

	Before	After
A Field	213	unchanged
B Field	D2C1D4	unchanged
Comparison	xxx	LOW

SET MODE (SMF)

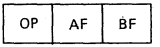


OP = 47

AFBF = A control Mask of 16 bits.

The Set Mode instruction controls the setting and resetting of various toggles within the processor.

HALT BREAKPOINT (HBK)



OP = 48

AF = Unused and ignored. May specify Indirect Field Length.
BF = Eight bit breakpoint control mask. The field will not be recognized as indirect.

In the HBK instruction, the breakpoint mask (BF) is compared to the breakpoint bit pattern located at base plus 46 and 47. If a bit is set in the bit pattern that corresponds to a set bit in the mask, the halt execution digit is accessed from address 77, absolute. The halting of the processor is then dependent upon the value of the halt execution digit. If there is no correspondence between the bits in the mask and the bits in the breakpoint bit pattern, the instruction is terminated with no significant action.

EDIT (EDT)

OP	AF	BF	A Address	B Address	C Address

OP = 49

AF = May specify a literal.
BF = Number of eight bit edit-operators and in line literals in the B-field. A value of 00 is equal to a length of 100 characters.
A = Address of the source data field operand. Address can be indexed, indirect or extended. Full address controller usage applies.
B = Address of the edit-operator field. Address can be indexed, indirect or extended. The final address controller is ignored and the data treated as UA.
C = Address of the destination data field operand. Address can be indexed, indirect or extended. The final address controller must be UN or UA. If the controller specifies SN, the Invalid Instruction is set.

An EDT instruction is normally used to edit a series of digits or characters into a readable format, adding punctuation and special symbols as required. The raw data is moved from the A field to the C field and, in the process, edited by micro-operators in the B field.

Example 1: Edit

OP	**AF**	**BF**	**A**	**B**	**C**
49	00	01	A Field (UA)	B Field (UA)	C Field (UA)

A Field	C1C2C3
B Field	02
C Field (After)	F1F2F3
Comparison (After)	HIGH

Example 2: Edit

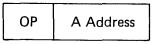
OP	**AF**	**BF**	**A**	**B**	**C**
49	00	22	A Field (SN)	B Field (UA)	C Field (UA)

A FIELD											C0	01	30	59
B FIELD				4B	D7	4B	C1	4B	E8	37	92	75	64	75
				75	75	85	93	33	01	92	5B	C3	5B	D9
TABLE (48-62)							4E	60	5C	4B	6B	5B	F0	40
							+	-	*	.	,	$	0	b
CFIELD(AFTER)	D7	C1	E8	40	5C	5C	5C	5B	F1	F3	4B	F5	F9	40	40
	P	A	Y	b	*	*	*	$	1	3	.	5	9	b	b

NOTE: b = BLANK

COMPARISON (AFTER) HIGH

INTEGER ADD (IAD)



OP = 50

A = Address of the Addend field operand. Address can be indexed, indirect or extended. The final address controller is ignored and is always treated as SN.

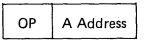
An IAD instruction algebraically adds the data in the field specified by the A Address to the data in the Accumulator, and places the sum in the Accumulator.

Example 1: Add Integer to Accumulator

OP	**A**
50	A Field

	Before	**After**
A Field	+1111111	+1111111
Accumulator	+08+01234567	+08+02345678
Comparison	nnn	HIGH
Overflow	nn	unchanged

INTEGER ADD AND STORE (IAS)



OP = 51

A = Address of the Addend and Sum field operand. Address can be indexed, indirect or extended. The final address controller is always treated as SN.

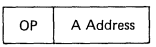
An IAS instruction is identical to the Integer Add except that the sum is placed in the location specified by the A Address. The sum is also retained in the Accumulator.

Example 1: Add Integer to Accumulator and Store

OP	A
51	A Field

	Before	After
A Field	+1111111	+2345678
Accumulator	+08+01234567	+08+02345678
Comparison	nnn	HIGH
Overflow	nn	unchanged

INTEGER SUBTRACT (ISU)



OP = 52

A = Address of the subtrahend field operand. Address can be indexed, indirect or extended. The final address controller is always treated as SN.

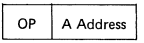
An ISU instruction algebraically subtracts the data in the field specified by the A Address from the data in the Accumulator, and places the difference in the Accumulator. The address controller can specify UN, indexing, or indirect addressing, with the final address restricted to mod 4 and specifying UN data. Literals cannot be used.

Example 1: Subtract Integer from Accumulator

OP	A
52	A Field

	Before	After
A Field	−0999999	−0999999
Accumulator	+08+02345678	+08+03345677
Comparison	nnn	HIGH
Overflow	nn	unchanged

INTEGER SUBTRACT AND STORE (ISS)



OP = 53

A = Address of the subtrahend field operand. Address can be indexed, indirect or extended. The final address controller is always treated as SN.

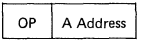
An ISS instruction is identical to the Integer Subtract except that the result is placed in the location specified by the A Address. The result is also retained in the accumulator.

Example 1: Subtract Integer from accumulator and Store

OP	A
53	A Field

	Before	After
A Field	−0999999	+3345677
Accumulator	+08+02345678	+08+03345677
Comparison	nnn	HIGH
Overflow	nn	unchanged

INTEGER MULTIPLY (IMU)



OP = 54

A = Address of the multiplier field operand. Address can be indexed, indirect or extended. The final address controller is always treated as SN.

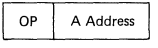
An IMU instruction multiplies the data at the A Address by the contents of the accumulator and stores the product in the accumulator. The address controller can specify UN, indexing, or indirect addressing, with the final address restricted to mod 4 and specifying UN data. Literals cannot be used.

Example 1: Multiply accumulator by Integer

OP	**A**
54	A Field

	Before	**After**
A Field	+0000003	+0000003
Accumulator	+08+01234567	+08+03703701
Comparison	nnn	HIGH
Overflow	nn	unchanged

INTEGER MULTIPLY AND STORE (IMS)



OP = 55

A = Address of the multiplier field operand. Address can be indexed, indirect or extended. The final address controller is always treated as SN.

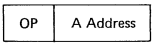
An IMS is identical to an Integer Multiply, except that the product is stored in the location specified by the A Address. The product is also retained in the accumulator.

Example 1: Multiply accumulator by Integer and Store

OP	A
55	A Field

	Before	After
A Field	+0000003	+3703701
Accumulator	+08+01234567	+08+03703701
Comparison	nnn	HIGH
Overflow	nn	unchanged

INTEGER MEMORY INCREMENT (IMI)



OP = 57

A = Address of the increment field operand. Address can be indexed, indirect or extended. The final address controller if equal to one (SN) indicates a decrement operation, and if equal to zero (UN) indicates an increment operation. Other controller values are reserved.

An IMI instruction increments or decrements the data at the A Address by 1. If AC equals 0, the data in the A field is incremented; if AC equals 1, the data is decremented. If the A controller specifies indirect addressing, the final address must have a controller equal to 0 or 1. Addresses must be mod 4; literals cannot be used.

Example 1: Memory Increment

OP	A
57	A Field (UN)

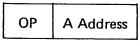
	Before	After
A Field	+1234567	+1234568
Accumulator	nnnnnnnnnnn	+08+01234568
Comparison	nnn	HIGH
Overflow	nn	unchanged

Example 2: Memory Decrement

OP	A
57	A Field (SN)

	Before	After
A Field	+1234567	+1234566
Accumulator	nnnnnnnnnnnn	+08+01234566
Comparison	nnn	HIGH
Overflow	nn	unchanged

INTEGER LOAD (ILD)



OP = 58

A = Address of the source data field operand. Address can be indexed, indirect or extended. The final address controller is always treated as SN.

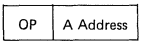
An ILD instruction loads the data at the location specified by the A Address into the accumulator.

Example 1: Load Accumulator with Integer

OP	A
58	A Field

	Before	After
A Field	−9876543	−9876543
Accumulator	nnnnnnnnnnn	+08−09876543
Comparison	nnn	LOW
Overflow	nn	unchanged

INTEGER STORE (IST)



OP = 59

A = Address of the destination field operand. Address can be indexed, indirect or extended. The final address controller is always treated as SN.

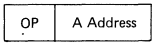
An IST instruction stores the operand and its sign from the accumulator into the 8-digit location specified by the A Address.

Example 1: Store Accumulator Integer in Memory

OP	A
59	A Field

	Before	After
A Field	nnnnnnnnnnn	−9876543
Accumulator	+08−09876543	+08−09876543
Comparison	nnn	LOW

REAL ADD (RAA)



OP = 70

A = Address of the Addend field operand. Address may be indexed, indirect or extended. When the final controller is equal to zero (UN) the data field will be Single Precision. When the final address controller is equal to one (SN) the data field is Double Precision. Address controller equal to two is treated as Single Precision.

An RAA instruction adds the data in the A field to the data in the accumulator and places the sum in the accumulator.

Example 1: Add Floating number to Accumulator

OP	A
70	A Field (UN)

	Before	After
A Field	+05+22222222	+05+22222222
Accumulator	+05+12345678	+05+34567900
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL ADD AND STORE (RAS)

OP	A Address

OP = 71

A = Address of the Addend and Sum field operand. Address may be indexed, indirect or extended. When the final address controller is equal to zero (UN) the data field is Single Precision. When the final address controller is equal to one (SN) the data field is Double Precision. Address controller equal to two is treated as Single Precision.

An RAS instruction is identical to the Real Add, except that the sum is placed into the location specified by the A Address. The sum is also retained in the accumulator.

Example 1: Add Floating Number to Accumulator and store

OP	A
71	A Field (UN)

	Before	After
A Field	+05+22222222	+05+34567900
Accumulator	+05+12345678	+05+34567900
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL SUBTRACT (RSU)

OP	A Address

OP = 72

A = Address of the subtrahend field operand. Address can be indexed, indirect or extended. When the final address controller is equal to zero (UN) the data field will be Single Precision. When

the final address is equal to one (SN) the data field will be Double Precision. Address controller equal to two will be treated as Single Precision.

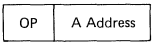
An RSU instruction causes the data in the A field to be subtracted from the data in the accumulator and the difference to be placed in the accumulator.

Example 1: Subtract Floating Number from the Accumulator

OP	A
72	A Field (UN)

	Before	After
A Field	+05+11111111	+05+11111111
Accumulator	+05+12345678	+04+12345670
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL SUBTRACT AND STORE (RSS)



OP = 73

A = Address of the subtrahend field operand. Address may be indexed, indirect or extended. When the final address controller is equal to zero (UN), the data field will be Single Precision. When the final address controller is equal to one (SN) the data field will be Double Precision. Address controller equal to two will be treated as Single Precision.

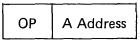
An RSS instruction is identical to the Real Subtract, except that the result is placed into the location specified by the A Address. The result is also retained in the accumulator.

Example 1: Subtract Floating Number from the Accumulator and Store

OP	A
73	A Field (UN)

	Before	After
A Field	+05+11111111	+04+12345670
Accumulator	+05+12345678	+04+12345670
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL MULTIPLY (RMU)



OP = 74

A = Address of the multiplier field operand. Address may be indexed, indirect or extended. When the final address controller is equal to zero (UN) the data field will be Single Precision. When the final address controller is equal to one (SN) the data field will be Double Precision. Address controller equal to two will be treated as Single Precision.

An RMU instruction multiplies the data in the A field by that in the accumulator and stores the product in the accumulator.

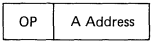
Example 1: Multiply Accumulator by Floating Point Number

OP **A**

74 A Field (UN)

	Before	**After**
A Field	+05+30000000	+05+30000000
Accumulator	+05+12345678	+09+37037034
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL MULTIPLY AND STORE (RMS)



OP = 75

A = Address of the multiplier field operand. Address may be indexed, indirect or extended. When the final address controller is equal to zero (UN) the data field will be Single Precision. When the final address controller is equal to one (SN) the data field will be Double Precision. Address controller equal to two will be treated as Single Precision.

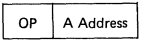
An RMS instruction is identical to the Real Multiply, except that the product is stored in the location specified by the A Address. The product is also retained in the accumulator.

Example 1: Multiply Accumulator by Floating Number and Store

OP	A
75	A Field (UN)

	Before	After
A Field	+05+30000000	+09+37037034
Accumulator	+05+12345678	+09+37037034
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL DIVIDE (RDV)



OP = 76

A = Address of the divisor field operand. Address may be indexed, indirect or extended. When the final address controller is equal to zero (UN) the data field will be Single Precision. When the final address controller is equal to one (SN) the data field will be Double Precision. Address controller equal to two will be treated as Single Precision.

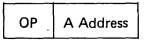
An RDV instruction divides the contents of the accumulator by the contents of the A field, placing the quotient in the accumulator.

Example 1: Divide Accumulator by Floating Point Number

OP	A
76	A Field (UN)

	Before	After
A Field	+05+20000000	+05+20000000
Accumulator	+05+12345678	+00+61728390
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL DIVIDE AND STORE (RDS)



OP = 77

A = Address of the divisor/quotient field operand. Address may be indexed, indirect or extended. When the final address controller is equal to zero (UN) the data field will be Single Precision. When the final address controller is equal to one (SN) the data field will be Double Precision. Address controller equal to two will be treated as Single Precision.

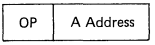
An RDS instruction is identical to a Real Divide, except that the quotient is stored in the location specified by the A Address. The quotient is also retained in the accumulator.

Example 1: Divide Accumulator by Floating Point Number and Store

OP	A
77	A Field (UN)

	Before	After
A Field	+05+20000000	+00+61728390
Accumulator	+05+12345678	+00+61728390
Comparison	nnn	HIGH
Overflow	nn	unchanged

REAL LOAD (RLD)



OP = 78

A = Address of the source data field operand. Address may be indexed, indirect or extended. When the final address controller is equal to zero the data field will be Single Precision. When the final controller is equal to one the data field will be Double Precision. Address controller equal to two will be treated as Single Precision.

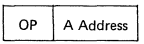
An RLD instruction loads the information from the A field into the accumulator.

Example 1: Load Accumulator with Floating Point Number

OP	A
78	A FIELD (UN)

	Before	After
A Field	+07−12345678	+07−12345678
Accumulator	nnnnnnnnnnn	+07−12345678
Comparison	nnn	LOW

REAL STORE (RST)



OP = 79

A = Address of the destination field operand. Address may be indexed, indirect or extended. A final address controller value of zero (UN) indicates Single Precision; a value of one (SN) indicates Double Precision; a value of two is treated as Single Precision.

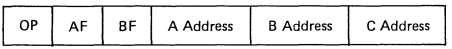
An RST instruction stores the contents of the accumulator, including the exponent and signs, in the location specified by the A address. If the A controller specifies single precision (AC = 0), the eight least significant digits in the accumulator are not stored. If AC equals 1, (double precision), the full 16 digits in the accumulator are stored.

Example 1: Store Accumulator in Memory in Floating Point Notation

OP	A
79	A Field (UN)

	Before	After
A Field	nnnnnnnnnnn	+07−12345678
Accumulator	+07−12345678	+07−12345678
Comparison	nnn	LOW

FLOATING-POINT ADD (FAD)



OP = 80

AF = Length of the A Field mantissa. The sign and exponent digits (4) are not included in this length. AF can be indirect or Literal Flag (if Literal, the mantissa length is limited to two digits). A value of 00 is equal to a mantissa length of 100 digits.

BF = Length of the B Field mantissa. The sign and exponent digits (4) are not included in this length. BF can be indirect. A value of 00 is equal to a mantissa length of 100 digits.

A = Address of the addend field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

B = Address of the augend field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

C = Address of the sum field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the data is stored in Floating Point format.

After proper alignment of operands, an FAD instruction adds the contents of the A field to those of the B field and stores the sum in the C field.

Example 1: Add Two Floating Point Numbers

OP	AF	BF	A	B	C
80	02	05	A Field (SN)	B Field (SN)	C Field (SN)

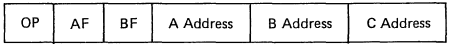
	Before	After
A Field	−04+20	−04+20
B Field	−05+67501	−05+67501
C Field	xxxxxxxxx	−05+67701
Comparison	nnn	HIGH
Overflow	nnn	unchanged

Example 2: Add Two Floating Point Numbers, Overflow Condition

OP	AF	BF	A	B	C
80	01	03	A Field (SN)	B Field (SN)	C Field (SN)

	Before	After
A Field	+99+1	+99+1
B Field	+99+999	+99+999
C Field	xxxxxxx	unchanged
Comparison	nnn	HIGH
Overflow	nnn	On

FLOATING-POINT SUBTRACT (FSU)



OP = 81

AF = Length of the A Field mantissa. The sign and exponent digits (4) are not included in the length. AF can be indirect or literal flag (if literal, the mantissa length is limited to two digits). A value of 00 is equal to a mantissa length of 100 digits.

BF = Length of the B Field mantissa. The sign and exponent digits (4) are not included in the length. BF can be indirect. A value of 00 is equal to a mantissa length of 100 digits.

A = Address of the subtrahend field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

B = Address of the minuend field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

C = Address of the difference field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the data is stored in Floating Point format.

After proper alignment of operands, an FSU instruction algebraically subtracts the contents of the A field from that of the B field and stores the difference in the C field.

Example 1: Subtract Two Floating Point Fields

OP	AF	BF	A	B	C
81	03	02	A Field (SN)	B Field (SN)	C Field (SN)

	Before	After
A Field	+02+500	+02+500
B Field	+03+20	+03+20
C Field	xxxxxx	+02−300
Comparison	nnn	LOW
Overflow	nnn	unchanged

Example 2: Subtract Two Floating Point Numbers, Overflow Condition

OP	AF	BF	A	B	C
81	03	03	A Field (SN)	B Field (SN)	C Field (SN)

	Before	After
A Field	+98−100	+98−100
B Field	+99+993	+99+993
C Field	xxxxxxx	unchanged
Comparison	nnn	HIGH
Overflow	nnn	On

FLOATING-POINT MULTIPLY (FMP)

OP	AF	BF	A Address	B Address	C Address

OP = 82

AF = Length of the A Field mantissa. The sign and exponent digits (4) are not included in the length. AF can be indirect or literal flag (if literal, the mantissa length is limited to two digits). A value of 00 is equal to a length of 100 digits.

BF = Length of the B Field mantissa. The sign and exponent digits (4) are not included in the length. BF can be indirect. A value of 00 is equal to a length of 100 digits.

A = Address of the multiplier field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

B = Address of the multiplicand field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

C = Address of the product field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the data is stored in Floating Point format.

An FMP instruction multiplies the multiplicand in the B field by the multiplier in the A field and places the product in the C field. Operands are assumed to be normalized; an unnormalized operand is treated as Zero. The result mantissa is normalized.

Example 1: Multiply Two Floating Point Numbers

OP	**AF**	**BF**	**A**	**B**	**C**
82	02	04	A Field (SN)	B Field (SN)	C Field (SN)

	Before	**After**
A Field	+01+20	+01+20
B Field	−01−5050	−01−5050
C Field	xxxxxxxx	+00−101000
Comparison	nnn	LOW
Overflow	nnn	unchanged

FLOATING-POINT DIVIDE (FDV)

OP	AF	BF	A Address	B Address	C Address

OP = 83

AF = Length of the A Field mantissa. The sign and exponent digits (4) are not included in the length. AF can be indirect or literal flag (if literal, the mantissa is limited to two digits). A value of 00 is equal to a length of 100 digits.

BF = Length of the B Field mantissa. The sign and exponent digits (4) are not included in the length. BF can be indirect. A value of 00 is equal to a length of 100 digits.

A = Address of the divisor field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

B = Address of the dividend/remainder field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the field assumed to be in Floating Point format.

C = Address of the quotient field operand. Address can be indexed, indirect or extended. The final address controller is ignored and the data is stored in Floating Point format.

An FDV instruction divides the dividend in the B field by the divisor in the A field, placing the quotient in the C field and the remainder in the B field. Operands are assumed to be in normalized form; unnormalized operands are treated as Zero. The quotient is normalized; the remainder is not, and has the same field length, sign, and exponent as the original dividend. The initial quotient exponent is the result of the dividend (B field) exponent minus the divisor (A field) exponent; it can be modified if quotient normalization is required.

Example 1: Divide Two Floating Point Numbers

OP	AF	BF	A	B	C
83	02	05	A Field (SN)	B Field (SN)	C Field (SN)

	Before	After
A Field	+00+20	+00+20
B Field	+00+60000	+00+00000
C Field	xxxxxxx	+01+300
Comparison	nnn	HIGH
Overflow	nnn	unchanged

Example 2: Divide Two Floating Point Numbers, Underflow Condition

OP	AF	BF	A	B	C
83	02	05	A Field (SN)	B Field (SN)	C Field (SN)

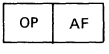
	Before	After
A Field	+50+20	+50+20
B Field	−60+60000	−60+60000
C Field	xxxxxxx	xxxxxxx
Comparison	nnn	LOW
Overflow	nnn	On

Example 3: Divide by Zero Condition

OP	AF	BF	A	B	C
83	02	02	A Field (SN)	B Field (SN)	C Field (SN)

	Before	After
A Field	+00+00	+00+00
B Field	−10+10	−10+10
C Field	xxxxxx	xxxxxx
Comparison	nnn	Equal
Overflow	nnn	On

ACCUMULATOR MANIPULATE (ACM)



OP = 84

AF = Operation variants.

An ACM instruction modifies the contents of the accumulator in the manner specified by the AF variant.

Example 1: Normalize Accumulator

OP	AF
84	00

	Before	After
Accumulator	+05+00123456 78901234	+03+12345678 90123400
Comparison	nnn	HIGH

Example 2: Convert Floating Point Number to Fixed Point Number

OP	AF
84	10

	Before	After
Accumulator	+06+12345678 90123456	+08+00123456 00000000
Comparison	nnn	HIGH

Example 3: Set Mantissa Sign to Plus

OP	AF
84	20

	Before	After
Accumulator	−05−12345678 90123456	−05+12345678 90123456
Comparison	nnn	HIGH

Example 4: Set Mantissa Sign to Minus

OP	AF
84	30

	Before	After
Accumulator	+05+12345678 90123456	+05J12345678 90123456
Comparison	nnn	LOW

Example 5: Complement Mantissa Sign

OP	AF
84	40

	Before	After
Accumulator	+05−12345678 90123456	+05+12345678 90123456
Comparison	nnn	HIGH

Example 6: Clear Accumulator

OP	AF
84	50

	Before	After
Accumulator	nnnnnnnnnnn nnnnnnnn	−99+00000000 00000000
Comparison	nnn	Equal

Example 7: Increment Exponent by 4

OP	AF
84	64

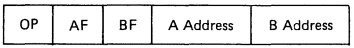
	Before	After
Accumulator	+05−12345678 90123456	+09J12345678 90123456
Comparison	nnn	LOW

Example 8: Decrement exponent by 2

OP	AF
84	72

	Before	After
Accumulator	+09+12345678 90123456	+07+12345678 90123456
Comparison	nnn	HIGH
Overflow	nnn	unchanged

DECIMAL TO BINARY (D2B)



OP = 88

AF = Length of the source data field in digits. Value can be indirect or a literal. A length of 00 is equal to a length of 100 digits.

BF = Length of the destination data field in units consisting of four binary bits. Value can be indirect. A value of 00 is equal to a length of 100 units.

A = Address of the decimal source data field. Address can be indexed, indirect or extended. The final address controller can be UN or UA. When the final controller is UA, the zone digits are ignored. A SN controller in this data field causes an Invalid Instruction error. Undigits in this field causes an undigit arithmetic error.

B = Address of the binary destination data field. Address can be indexed, indirect or extended. The final address controller can be UN or UA. When the final controller is UA, F zones are inserted. An SN controller in this data field causes an Invalid Instruction error.

The B2D instruction reads a decimal data field from a memory location (A), converts the entire value to a binary representation and stores the binary value in a second memory location (B).

Example 1: Decimal to Binary

OP	AF	BF	A	B
88	03	02	A Field (UN)	B Field (UN)

	Before	After
A Field	174	174
B Field	xx	AE
Comparison	HIGH	
Overflow	unchanged	

Example 2: Decimal to Binary – Overflow Condition

OP	AF	BF	A	B
88	03	02	A Field (UN)	B Field (UN)

	Before	After
A Field	374	374
B Field	Unchanged	
Comparison	HIGH	
Overflow	xx	On

Example 3: Decimal to Binary – Mixed Controllers

OP	AF	BF	A	B
88	02	02	A Field (UN)	B Field (UN)

	Before	After
A Field	F1F7F4	F1F7F4
B Field	xxx	OAE
Comparison	HIGH	
Overflow	Unchanged	

BINARY TO DECIMAL (B2D)

OP	AF	BF	A Address	B Address

OP = 89

AF = Length of the source data field in units of four binary bits. A value of 00 is equal to a length of 100 units.

BF = Length of the destination data field in digits. Value can be indirect. A value of 00 is equal to a length of 100 units.

A = Address of the binary source data field. Address can be indexed, indirect or extended. The final address controller can be

UN or UA. When the final controller is UA, the zone digits are ignored. A SN controller in this data field causes an Invalid Instruction error.

B = Address of the decimal destination data field. Address can be indexed, indirect or extended. The final address controller can be UN or UA. When the final controller is UA, F zones are inserted. An SN controller in this data field causes an Invalid Instruction error.

The D2B instruction reads a Binary data field from a memory location (A), converts the entire value to a decimal representation and stores the binary value in a second memory location (B).

Example 1: Binary to Decimal

OP	AF	BF	A	B
89	02	03	A Field (UN)	B Field (UN)

	Before	After
A Field	AE	AE
B Field	xxx	174
Comparison	HIGH	
Overflow	Unchanged	

Example 2: Binary to Decimal – Overflow Condition

OP	AF	BF	A	B
89	02	02	A Field (UN)	B Field (UN)

	Before	After
A Field	AE	AE
B Field	Unchanged	
Comparison	HIGH	
Overflow	xx	On

Example 3: Binary to Decimal – Zero Source Data

OP	AF	BF	A	B
89	12	01	A Field (UN)	B Field (UN)

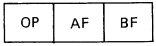
	Before	After
A Field	000000000000	000000000000
B Field	x	0
Comparison	Equal	
Overflow	Unchanged	

Example 4: Binary to Decimal – Mixed Controllers

OP	AF	BF	A	B
89	03	03	A Field (UN)	B Field (UA)

	Before	After
A Field	OAE	OAE
B Field	xxx	F1F7F4
Comparison	HIGH	
Overflow	Unchanged	

BRANCH REINSTATE (BRE)



OP = 90

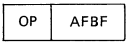
AF = The A Variant which determines the state in which the processor is to operate subsequent to the execution of this instruction (BRE). If the most significant digit of AV is equal to zero, the reinstated program is in Normal State and is responsive to interrupts. If not equal to zero, the reinstated program is in noninterruptible Control State. If the least significant digit of AV is not equal to zero, the Trace toggle is set, which causes a Branch Communicate at the end of the next instruction. Indirect Field Length can be specified.

BF = The B Variant is unused and ignored. Indirect Field Length can be specified.

The BRE instruction branches from a control program in order to reinstate another program, either in Control or Normal state. In reinstating the program, the BRE reads the following information from reserved memory addresses 61 through 76, and places it in the appropriate registers and Flip/Flops. Typically, this instruction restores the processor to the state that existed when a Branch Communicate instruction was executed. BRE is a privileged instruction which requires that the base be equal to zero. The Next Program Instruction Address, the Base, the Limit, the Overflow Toggle, and Comparison Toggles are restored as indicated by the contents of memory address 61 through 76 (refer to OP=30).

If the interrupt is set at the beginning of execution of this instruction and the AF is equal to zero, a Branch Communicate (OP=30) to memory address 94 is executed, and the contents of memory address 61 through 76 remains unchanged.

SCAN RESULT DESCRIPTOR (SRD)



OP = 91

AFBF = Low order four digits of an absolute address in memory. The high order three digits are assumed to be zero. Indirect Field Length can be specified.

The address specified is assumed to point to a 16-bit result descriptor area. The first bit of this area is examined and:

1. If it is equal to zero (no result descriptor present), the four digits (link address) immediately following the descriptor area are examined and:

   a. If they are 0000, the comparison toggles are set to EQUAL and the instruction terminates (no descriptor found).
   b. If they are not zero, they replace the original address value and the operation is repeated.

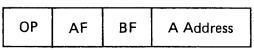
2. If it is equal to one (result descriptor present), the address of the descriptor area is stored into index register one (IX1). The next bit is examined, and the comparison toggles are set HIGH if it is a zero; otherwise, they are set LOW. The instruction then terminates.

The interrupt toggle is set to zero by this instruction.

If the initial address specified in ABBF is equal to zero, the comparison toggles are set to EQUAL, index register one (IX1) is unchanged, and the instruction terminates.

This is a privileged instruction. The base must equal zero.

READ ADDRESS MEMORY (RAD)



OP = 92

AF = Type of operation. If AF is equal to zero, the begin address of the channel specified by BF is read by the I/O Translator and stored in memory at the location specified by the A Address. If AF is equal to one, the end address of the channel specified

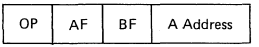
by BF is read by the I/O Translator and stored in memory at the location specified by the A Address. If AF is equal to nine, the contents of the memory location specified by the A Address is written by the I/O Translator into both the begin and end address of the channel specified by BF. AF can be indirect or specify a literal.

BF = Channel number. BF can be indirect. BF can specify any octal value from 00 to 77. Channel numbers 91 and 94 specify processor Extended Register R1 and R4. Channel numbers 90,92,93,95-9F cause Invalid Instruction to be set.

A = Address of the memory location plus two (A+2) where the eight digit begin or end address is stored. During a write operation this memory location contains the begin/end address to be written. The final address controller is ignored.

The AF specifies the type of operation. If AF is equal to Zero, the Begin address of the channel specified by BF is read by the IOT and is stored in memory at the location specified by the A Address. If AF is equal to one, the End address of the channel specified by BF is read by the IOT and is stored in memory at the location specified by the A Address. If AF is equal to nine, the contents of the memory location specified by the A Address is written into the scratchpad memory of the specified channel at the begin and end address locations.

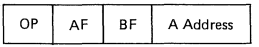
INITIATE I/O (IIO)



OP = 94

An IIO instruction causes the IOT to read a DLP descriptor from the location specified by the A Address. The IOT sends the descriptor OP code and variants to the channel number specified in BF (00-37). It also stores the Begin and End address of the data field in the Scratchpad Memory of the channel.

READ TIMER (RDT)



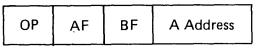
OP = 95

AF = Unused and reserved.
BF = Unused and reserved.

A = Address of the memory location where the six digit timer value is stored. Address can be indexed, indirect or extended. The final address controller is ignored.

The Read Timer instruction reads the timer value and writes it in the memory location specified by the A Address. The timer value is not affected.

READ AND CLEAR TIMER (RCT)

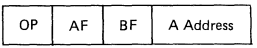


OP = 96

AF = Unused and reserved.
BF = Unused and reserved.
A = Address of the memory location where the six digit timer value is stored. Address can be indexed, indirect or extended. The final address controller is ignored.

The Read and Clear Timer instruction reads the timer value, writes that value into the memory location specified by the A address and sets the timer to zero.

SET TIMER (STT)



OP = 97

AF = Unused, but can specify literal.
BF = Unused and reserved.
A = Address of the memory location where the six digit timer value is stored. Address can be indexed, indirect or extended. The final address controller is ignored.

The Set Timer instruction sets the timer limit to the value contained at the A Address and resets the timer interrupt.

RED LIGHT HALT (RED)

OP = 98

This OP is a Red Light for the B 2900 and B 3900 Systems. It can be compared to 300094 failures on prior Medium Systems (B 3500 through B 4800). This OP is hardware forced in the Fetch Module and passed to the MCS Module, where it causes MCS Control Store to set CPROHLT (Processor Halt) and stop processor clocks.

PROCESSOR RESULT DESCRIPTOR (PRD)

OP = 99

The Processor Result Descriptor OP causes the processor to write a Processor R/D at absolute memory address 80. This OP is hardware forced by the RD/BCT PROM in the Fetch Module. The Fetch Module passes this OP to the MCS Module where the Execution Module reads the contents at absolute memory address 80. The Execution Module then ORs the error flags with this data and writes the ORed results back into absolute address 80.

Refer to Processor R/D Description for an explanation of the error flags mentioned.

SECTION 2

MAINTENANCE

Memory

FAULTY MEMORY STORAGE CHIP

A faulty memory storage chip (RAM) is located with the aid of tables 2-1 through 2-4 and a BCD or binary address. The following procedure is used to locate a faulty RAM.

1. The base containing the failing address is found by running memory test on the structure ICMD diskette or by analyzing the Memory Error Report Word. (Base 1 covers BCD addresses 000,000 through 3,407,871; Base 2 covers 3,407,872 through 6,815,743; Base 3 covers 6,815,744 through 9,961,471.)
2. The location of the module and Memory Storage Board is determined by looking at the address range of the different modules that contain the failing address.
3. The faulty RAM is located by by using the error vector code from the error report word or by analyzing the failing data pattern from the memory test. Table 2-4 lists the error vector codes and bit position numbers for the different memory storage chips on the card. (Even Mod is Mod 0 on Memory Storage Board)

Table 2-1. Base 1 Address Decode

DECIMAL ADDRESS		BINARY ADDRESS		CARD	MOD
BEGIN	END	BEGIN	END		
000000	131071	00000	1FFFF	ACBD6	0
131072	262143	20000	3FFFF	ACBD6	1
262144	393215	40000	5FFFF	ACBD4	0
393216	524287	60000	7FFFF	ACBD4	1
524288	655359	80000	9FFFF	ACBD2	0
655360	786431	A0000	BFFFF	ACBD2	1
786432	917503	C0000	DFFFF	ACBD0	0
917504	1048575	E0000	FFFFF	ACBD0	1
1048576	1173647	100000	11FFFF	ACBC8	0
1173648	1310719	120000	13FFFF	ACBC8	1
1310720	1441791	140000	15FFFF	ACBC6	0
1441792	1572863	160000	17FFFF	ACBC6	1
1572864	1703935	180000	19FFFF	ACBC4	0
1703936	1835007	1A0000	1BFFFF	ACBC4	1
1835008	1966079	1C0000	1DFFFF	ACBC2	0
1966080	2097151	1E0000	1FFFFF	ACBC2	1
2097152	2228223	200000	21FFFF	ACBC0	0

Table 2-1. Base 1 Address Decode (Cont)

DECIMAL ADDRESS		BINARY ADDRESS		CARD	MOD
BEGIN	END	BEGIN	END		
2228224	2359295	220000	23FFFF	ACBC0	1
2359296	2490367	240000	25FFFF	ACBB8	0
2490368	2621439	260000	27FFFF	ACBB8	1
2621440	2752511	280000	29FFFF	ACBB6	0
2752512	2883583	2A0000	2BFFFF	ACBB6	1
2883584	3014655	2C0000	2DFFFF	ACBB4	0
3014656	3145727	2E0000	2FFFFF	ACBB4	1
3145728	3276779	300000	31FFFF	ACBB2	0
3276780	3407871	320000	33FFFF	ACBB2	1

Table 2-2. Base 2 Address Decode

DECIMAL ADDRESS		BINARY ADDRESS		CARD	MOD
3407872	3538943	340000	35FFFF	AABD6	0
3538944	3670015	360000	37FFFF	AABD6	1
3670016	3801087	380000	39FFFF	AABD4	0
3801088	3932159	3A0000	3BFFFF	AABD4	1
3932160	4063231	3C0000	3DFFFF	AABD2	0
4063232	4194303	3E0000	3FFFFF	AABD2	1
4194304	4325375	400000	41FFFF	AABD0	0
4325376	4456447	420000	43FFFF	AABD0	1
4456448	4587519	440000	45FFFF	AABC8	0
4587520	4718591	460000	47FFFF	AABC8	1
4718592	4849663	480000	49FFFF	AABC6	0
4849664	4980735	4A0000	4BFFFF	AABC6	1
4980736	5111807	4C0000	4DFFFF	AABC4	0
5111808	5242879	4E0000	4FFFFF	AABC4	1
5242880	5373951	500000	51FFFF	AABC2	0
5373952	5505023	520000	53FFFF	AABC2	1
5505024	5636095	540000	55FFFF	AABC0	0
5636096	5767167	560000	57FFFF	AABC0	1
5767168	5898239	580000	59FFFF	AABB8	0
5898240	6029311	5A0000	5BFFFF	AABB8	1
6029312	6160383	5C0000	5DFFFF	AABB6	0
6160384	6291455	5E0000	5FFFFF	AABB6	1
6291456	6422527	600000	61FFFF	AABB4	0
6422528	6553559	620000	63FFFF	AABB4	1
6553660	6684671	640000	65FFFF	AABB2	0
6684672	6815743	660000	67FFFF	AABB2	1

Table 2-3. Base 3 Address Decode

DECIMAL ADDRESS		BINARY ADDRESS		CARD	MOD
6815744	6946815	680000	69FFFF	AABE4	0
6946816	7077887	6A0000	6BFFFF	AABE4	1
7077888	7208959	6C0000	6DFFFF	AABE6	0
7208960	7340031	6E0000	6FFFFF	AABE6	1
7340032	7471103	700000	71FFFF	AABE8	0
7471104	7602175	720000	73FFFF	AABE8	1
7602176	7733247	740000	75FFFF	AABF0	0
7733248	7864319	760000	77FFFF	AABF0	1
7864320	7995391	780000	79FFFF	AABF2	0
7995392	8126463	7A0000	7BFFFF	AABF2	1
8126464	8257535	7C0000	7DFFFF	AABF4	0
8257536	8388607	7E0000	7FFFFF	AABF4	1
8388608	8519679	800000	81FFFF	AABF6	0
8519680	8650571	820000	83FFFF	AABF6	1
8650572	8781823	840000	85FFFF	AABF8	0
8781824	8912895	860000	87FFFF	AABF8	1
8912896	9043967	880000	89FFFF	AABG0	0
9043968	9175039	8A0000	8BFFFF	AABG0	1
9175040	9306111	8C0000	8DFFFF	AABG2	0
9306112	9437183	8E0000	8FFFFF	AABG2	1
9437184	9568255	900000	91FFFF	AABG4	0
9568256	9699327	920000	93FFFF	AABG4	1
9699328	9830399	940000	95FFFF	AABG6	0
9830400	9961471	960000	97FFFF	AABG6	1

Table 2-4. Hamming Vector Decoding

Hamming Code Value	Error Vector Code	Chip Location EVEN	Chip Location ODD	Bit Number
33	43	30L	30S	0
56	71	30B	30F	1
36	49	28L	28S	2
3	07	28B	28F	3
34	45	27L	27S	4
5	0B	27B	27F	5
6	0D	25L	25S	6
7	0F	25B	25F	7
40	51	24L	24S	8
9	13	24B	24F	9
10	15	22L	22S	10
11	17	22B	22F	11
12	19	21L	21S	12
13	1B	21B	21F	13

Table 2-4. Hamming Vector Decoding (Cont)

Hamming Code Value	Error Vector Code	Chip Location EVEN	Chip Location ODD	Bit Number
14	1D	19L	19S	14
15	1F	19B	19F	15
48	61	17L	17S	16
17	23	17B	17F	17
18	25	14L	14S	18
19	27	14B	14F	19
20	29	13L	13S	20
21	2B	13B	13F	21
22	2D	12L	12S	22
23	2F	12B	12F	23
24	31	11L	11S	24
25	33	11B	11F	25
26	35	10L	10S	26
27	37	10B	10F	27
28	39	8L	8S	28
29	3B	8B	8F	29
30	3D	7L	7S	30
31	3F	7B	7F	31
1	03	5L	5S	32
2	05	5B	5F	33
4	09	4L	4S	34
8	11	4B	4F	35
16	21	2L	2S	36
32	41	2B	2F	37
0	01	0L	0S	38

MEMORY STORAGE BOARD LAYOUT

All components on the Memory Storage Board (see figure 2-1) are laid out in a 35-row (0-34) by 28-column (A through AE) matrix. Any component on the Memory Storage Board can be located by referencing the row and column designations on the card.

Example: =
= U31AB is an IC in Row 31 and Column AB.
= C2P is a Capacitor in Row 2 and Column P.
= R15Z is a Resistor in Row 15 and Column Z.
= CR17AB is a Diode in Row 17 and Column AB.

Backplane pins are numbered in two groups, 001 through 097 on the component side of the Memory Storage Board, and 101 through 197 on the solder side of the Memory Storage Board.

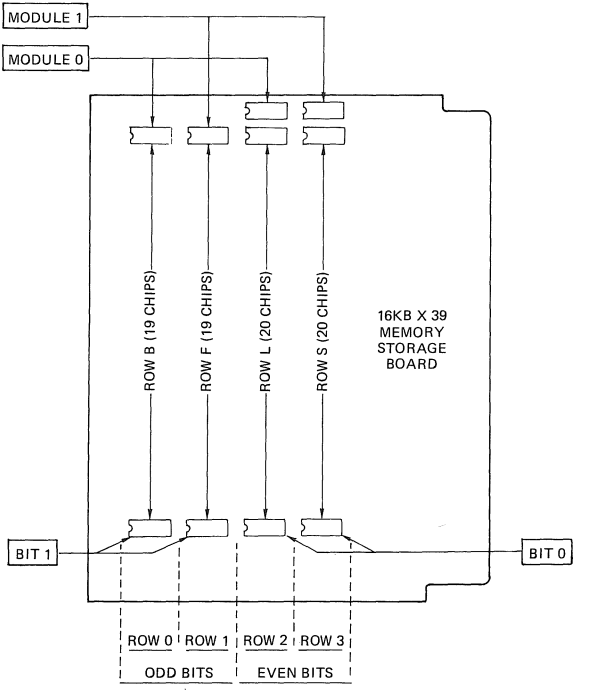


Figure 2-1. Memory Storage Board Layout

Universal Console

The Universal Console (UC) is an 8080 microprocessor based subsystem that provides operator and maintenance interfaces for the B 2900 system.

The UC incorporates an Operator Display Terminal (ODT) and an Industry Compatible Mini-Disk (ICMD) Drive that allows the operator to initialize and supervise Host system operations. All off-line maintenance operations are performed through the ODT and ICMD Drive.

NOTE

The ODT is often referred to as SPO in Medium Systems documentation. The ICMD Drive is often called Flexible Disk Drive or Floppy Disk Drive.

The UC includes hardware to perform the following functions when appropriate software is provided.

1. B 2900 Initialization.
2. B 2900 Maintenance.
3. Design Level-2 I/O Maintenance (TBI – Test Bus Interface).
4. Remote Maintenance Link (RML).
5. B 2900 System Peripherals – ODT and ICMD Drive.

Operational software for the UC is loaded from either the PANEL or PANDLP mini-disk. The PANEL mini-disk has routines for the following functions.

1. Halt-Load, Coldstart/Warmstart.
2. Reading and writing memory.
3. Run/Stop control of Host system.
4. Loading control state programs from other mini-disks.
5. Control state program initialization.
6. Maintenance Panel (controlled by UC software).
7. Remote maintenance support (RML).

The PANDLP mini-disk has the same routines as the PANEL mini-disk, except the remote maintenance support. The PANDLP has a a routine to support the Host system use of the ICMD Drive as a peripheral. If the use of the ICMD Drive as a peripheral is desired, then the PANDLP mini-disk must be loaded into UC memory.

Console Configuration

The following section details information on card location, cabling, and card strapping.

CARD LOCATION

The UC backplane is a common backplane, therefore the cards can be installed in any order. To facilitate cabling, the recommended order of card location follows.

Table 2-5. Card Location

Backplane Location	Card Name
1	DLP/IF Card
2	FDC Card
3	Blank
4	Microcomputer Card
5	SMV Switch Card
6	Blank
7	HCP Card
8	Dual SPO/MEM Card

CONSOLE CABLING

The UC has interconnecting cables to the Processor, ICMD Drive, DLP, and the Maintenance Panel. The following table details the cable connections for each card in the UC.

Table 2-6. UC Cable Connections

Card Name	From	To	Cable Type
DLP/IF	J3	UC-DLP	Ribbon
DLP/IF	J4	TBI Panel	Ribbon *
FDC	J4	ICMD Drive	Ribbon
Microcomputer	J1	ODT	Special **
Microcomputer	J2	Maint. Panel	Ribbon
SMV Switch	J1	ABBC8 – J2	Ribbon
SMV Switch	J2	Uniline DLP	Ribbon
SMV Switch	J3	HCP – J4	Ribbon
SMV Switch	J4	ODT	Special **
HCP	J2	ABBC8 – J4	Ribbon
HCP	J4	SMV Switch – J3	Ribbon

NOTES

* The TBI Panel is located in the I/O cable junction area. The Panel connects to the Maintenance Card in the lower DLP base and is concatenated to panels for the other DLP bases.

** This cable connects the ODT and MODE switch to the SMV Switch Card (inner cable connector) and Microcomputer Card (outer cable connector).

CARD CONFIGURATION

The DLP/IF Card and the HCP Card have strappable options.

The DLP/IF Card has a jumper chip at location E3. By changing jumpers, the BAUD rate of the TBI and RML can be altered. The following tables describe the options.

Table 2-7. DLP/IF Card Jumpers for Normal Operation

E3 pin 3 to E3 pin 15	TBI BAUD rate = 19.2 K
E3 pin 5 to E3 pin 12	RML BAUD rate =
E3 pin 8 to E3 pin 9	Selects 307 KHz clock from HCP

Table 2-8. DLP/IF Card Options

E3 pin 1 to E3 pin 16	TBI BAUD Rate = 4800 or 1200
E3 pin 2 to E3 pin 15	TBI BAUD Rate = 9600
E3 pin 4 to E3 pin 13	RML BAUD Rate = 4800
E3 pin 6 to E3 pin 12	RML BAUD Rate = 1200
E3 pin 7 to E3 pin 11	Enables Ring Input from RML
E3 pin 8 to E3 pin 10	Selects 307 KHz clock (Dual SPO)

The HCP Card has jumper chips at locations J3, K3, M5, and P5. The jumpers select BAUD rate for the SPO logic and determine whether Soft or Hard SPO logic is used to communicate with the ODT.

Normally, all straps on jumper chips J3 and K3 are installed. If Hard SPO logic is desired (recommended only when Soft SPO logic is malfunctioning), remove strap J3 pin 1 to J3 pin 16. The ODT cable must then be moved from HCP Card J4 to Dual SPO J3.

The strap on jumper chip M5 (pin 6 to 11), gates 307 KHz clock to the backplane. The 307 KHz clock is used by the DLP/IF Card, therefore this strap must be installed.

The straps on jumper chip P5 select BAUD Rate for the Soft SPO logic and clock period for the Performance Monitor. The following table describes the options for jumper chip P5.

Table 2-9. HCP Card Jumpers for Normal Operation

P5 pin 1 to P5 pin 16	Selects Performance Monitor clock period = 16 μSEC.
P5 pin 3 to P5 pin 14	Soft SPO BAUD Rate = 9600
P5 pin 5 to P5 pin 11	Must be installed for B2900/B3900
P5 pin 6 to P5 pin 10	Must be installed for B2900/B3900
P5 pin 8 to P5 pin 9	Must be installed for B2900/B3900

Table 2-10. HCP Card Jumper Options

P5 pin 2 to P5 pin 15	Selects Performance Monitor clock period = 1 μSEC.
P5 pin 4 to P5 pin 13	Soft SPO BAUD Rate = 19.2 K

Universal Console Diagnostics

The UC diagnostics are divided into two major sections. Resident diagnostic programs permanently reside in UC ROM. Non-resident Diagnostic programs are loaded into UC RAM from ICMD (Industry Compatible Mini-disks).

RESIDENT DIAGNOSTICS

The Resident Diagnostics execute automatically when the UC is powered up or reset (cleared). The Microcomputer Card, Flexible Disk Control Card, and data paths are tested. The basic read circuitry of the ICMD Drive is also tested.

EXECUTING RESIDENT DIAGNOSTICS

To load UC memory and begin execution:

1. Power-up the system.
2. Place a mini-disk into the ICMD Drive.
3. Set the UC Maintenance Panel switches as follows.

Trace/Normal to Normal.
Data/MTR to MTR.
Single Cycle/Run to Run.
Panel Enable/Disable to Enable.
Processor only/Entire UC to Entire UC.
Input switches 8-15 to the down position.

Input switches 0-7 select the following options:

Switches 0-7 down – Normal testing (halt on error).
Switch 0 up – Halt after Bootstrap load.
Switch 1 up – Select ICMD Drive Exerciser.
Switch 2 up – Loop on ICMD OP.
Switch 3 up – Not used.
Switch 4 up – Loop on RAM tests.
Switch 5 up – Loop on error.
Switch 6 up – Skip resident code, go to 100 hex in RAM.
Switch 7 up – Loop on Resident Diagnostics until error.

4. Depress the RESET pushbutton (on the keyboard) or CLEAR pushbutton (on the Maintenance Panel).

   If the ERROR lamp fails to extinguish after 30 seconds, an error has been detected by Resident Diagnostics. To determine the test in error, read the hex display of the MTR register (lamps 0-7). Repeat the above procedure to ensure that a solid failure exists.

5. If test in error is 00-1F, complete the Error Reporting Form. For any other halt, the following table references the appropriate documentation.

Table 2-11. Test Documentation

Data/Mtr Lamps	Failing Test	T & F Vol.	Section
20-4F	Soft SPO Test	2	Error Halts
50-51	Soft SPO Loader	2	Error Halts
52-5F	Undefined Halt	1	MTR, Hardcore Logic
60-DD	Hard SPO Test	2	Error Description
DE-DF	Hard SPO Loader	2	Error Description
E0-E7	DOI Run-Time Error	2	Error Halts
E8-ED	Undefined Halt	1	MTR, Hardcore Logic
EE	Microprocessor Test	2	Test Desc Error Halt
EF-F0	Undefined Halt	1	MTR, Hardcore Logic
F1-F2	Bootstrap Loader	1	Bootstrap Loader
F3-F8	Undefined Halt	1	MTR, Hardcore Logic
F9-FB	ICS Loader	2	Error Halts
FC-FF	Undefined Halt	1	MTR, Hardcore Logic

RESIDENT DIAGNOSTICS ERROR REPORTING FORM

For error halts 01-1E complete the following form.

ERROR REPORTING FORM

TEST NO. ____ (HEX) IS THE PANEL ERROR LAMP ILLUMINATED?
YES ____ NO ____

READ THE FOLLOWING REGISTERS BY PLACING ADDRESS IN THE INPUT SWITCHES AND PLACING DATA/MTR SWITCH TO DATA. TO READ, DEPRESS READ FROM ADDRESS PUSHBUTTON (VALUE IN DATA/MTR LAMPS).

ABH (EF00) ____
ABL (EF01) ____
SIRR (EF0D) ____
MCR (EF0E) ____
MER (EF0F) ____

PLACE INPUT SWITCHES 0-15 TO THE DOWN POSITION. PLACE DATA/MTR SWITCH TO MTR. THE TEST IN ERROR SHOULD STILL BE IN THE MTR REGISTER. DEPRESS THE START BUTTON AND RECORD ACC (VALUE IN THE MTR LAMPS). DEPRESS THE START BUTTON AND RECORD B REGISTER. CONTINUE PROCEDURE TO RECORD VALUES FOR THE REMAINING REGISTERS.

1. ACC ____
2. B REG ____ (ERROR NO.)
3. C REG ____
4. D REG ____
5. E REG ____
6. H REG ____
7. L REG ____

Refer to UC T & F Vol. 1. Locate the test in which the error has occurred. The above form has all the information necessary to determine the fault and corrective action required.

NON-RESIDENT DIAGNOSTICS

The Non-resident Diagnostics test the UC hardware not tested by the Resident Diagnostics. Testing is accomplished by several programs, residing on the UC Non-resident Diagnostics mini-disk (DIAG41).

By resetting the UC, the Initial Code Segment (ICS) is loaded into memory. When ICS executes, it reads the jumper chip on the Host UC Port Card (HCP). The jumper chip is strapped to select Soft SPO logic or Hard SPO

logic for communication to the ODT (Operator Display Terminal). The selected logic is then tested. If no errors are encountered, the Diagnostic Operator Interface (DOI) file is then loaded into memory. The DOI allows the operator to control all other non-resident testing of the UC.

Executing Non-Resident Diagnostics

To load and begin Non-resident Diagnostics:

1. Power-up the system.
2. Insert Non-resident Diagnostic mini-disk (DIAG41) into the ICMD Drive.
3. Depress the RESET pushbutton (on the keyboard) or CLEAR pushbutton (on the UC Maintenance Panel). The ICS will be loaded and then executed. The ICS determines which SPO logic is selected and loads the proper DOI program. The DOI is then executed. If no error is encountered in the SPO logic, the DOI displays a status line on the ODT. The status line informs the operator of the next test and section to be run.

   If an error exists in the SPO logic, the ODT remains blank. The MTR lamps on the Maintenance Panel display an error number. For a description of the error, refer to UC T & F Vol. 2, Error Halts. Section 4.

4. To run all non-resident tests enter RUN on the keyboard. The tests run in the following order.

   a. LCPI (DLP/IF Card)
   b. HCP (HCP Card)
   c. FD (Flexible Disk Confidence Test)
   d. TBI (Test Bus Interface)
   e. RML (Remote Link Diagnostic)
   f. MP (Microprocessor Confidence Test)

5. If an error is found during a test, the following information is displayed on the ODT.

   a. TEST NAME
   b. SECTION NUMBER
   c. ERROR NUMBER
   d. BRIEF DESCRIPTION OF ERROR

Corrective action for errors can be obtained by referencing the section for the test (LCPI, HCP, FDC, etc.) in UC T & F Volumes 2A, 2B, 2C.

By using option commands, it is possible to run particular tests and sections, loop, read and write to mini-disk, etc. All commands are listed in the User Commands section.

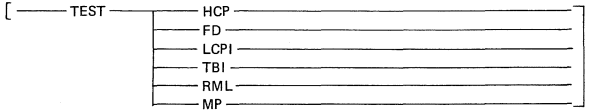
User Commands

The following commands are available for use in Non-resident Diagnostics. All commands can be entered by using the first three letters of the command. For example, the SECTION Command may be entered as SEC <number>.

Available Commands

TEST
SECTION
REPEAT
SET/RESET
DATA
BEGIN
END
INCREMENT
INITIALIZE
DRIVE
SECTOR SIZE
DATA LENGTH
OP
RUN
STOP
CLEAR
STATUE
DISPLAY
PATCH

TEST Command



TEST selects the test to be executed. More than one test can be specified and test names can be repeated. For example, TEST HCP FD RML HCP FD HCP.

SECTION Command

SECTION selects the section number (of the test specified) to be executed. More than one section number can be specified and sections can be repeated. For example, SECTION 1 2 3 1 2 4.

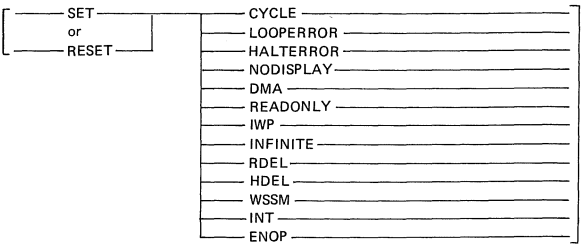
REPEAT Command

REPEAT repeats each section, as specified by SECTION the specified number of times before the next section executes.

NOTE

The BOJ and EOJ messages for each section only appears once, even if the sections repeat more times.

SET/RESET Command



SET/RESET sets or resets the following options.

CYCLE causes the list of tests specified by TEST to be repeated until CYCLE is reset.

LOOPERROR causes the current test section to be repeated if a hardware error is detected.

HALTERROR causes the test section to halt if an error is detected.

NODISPLAY when set, prevents error messages from being displayed. This allows the loop to execute faster.

DMA allows ICMD operations to use direct memory access.

READONLY prevents ICMD write operations.

IWP allows ICMD write operations to be performed without checking write protect status of the mini-disk.

CAUTION

If the IWP option is set and DIAG41 mini-disk is in the ICMD Drive, FD test overwrites the mini-disk. The mini-disk is corrupted.

INFINITE causes Section 0 of the FD test to loop continuously. Since the DOI is not accessed during this section, the operation executes faster. Oscilloscope images are then easier to interpret.

RDEL causes ICMD read operations to read the contents of a deleted sector, if one is encountered.

HDEL causes the ICMD read operation to halt if a deleted sector is encountered.

WSSM cause a Special Sync Mark (deleted data) to be written during an ICMD write operation.

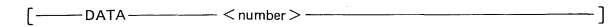
INT increment to the next track before an ICMD initialize operation is performed.

ENOP enables options for the Flexible Disk Confidence Test.

CAUTION

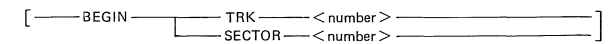
The test sections, except Section 0, execute with options specified in the test. When other options are specified, error detection routines may no longer function correctly.

DATA Command



DATA allows the user to input up to 16 bytes of data to be used by the test sections. Data is decimal unless preceded by @. For example, DATA 1 2 5 255 or DATA @FF @C2 56.

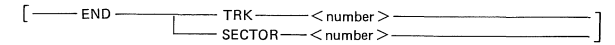
BEGIN Command



BEGIN allows the user to set the beginning track and sector to be used by the Flexible Disk Confidence Test. Beginning track and beginning sector must be specified separately.

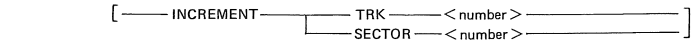
Minimum Value	Maximum Value
Track = 0	Track = 76
Sector = 1	Sector = 26

END Command



END allows the user to set the end track and sector to be used by the Flexible Disk Confidence Test. Minimum and maximum values apply as in BEGIN.

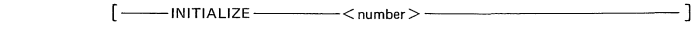
INCREMENT Command



INCREMENT allows the user to specify the amount that the track and sector is incremented after each Flexible Disk Confidence Test operation.

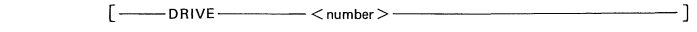
Minimum Value	Maximum Value
Track = 0	Track = 76
Sector = 0	Sector = 25

INITIALIZE Command



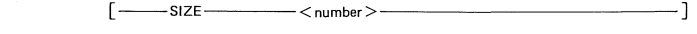
INITIALIZE specifies the sector numbers of physical sectors on every track of the mini-disk. The command requires that 26 decimal entries be made. If INITIALIZE is entered with no parameters, the mini-disk is initialized in interlaced format (1,14,2,15,3,16,etc.).

DRIVE Command



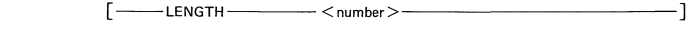
DRIVE allows the user to select ICMD Drive 1 or 2, as addressed by the DIP switches located on the drives. By default, Drive 1 is selected.

SECTOR SIZE Command



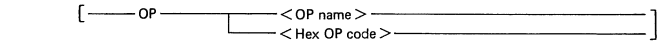
SIZE specifies the number of bytes per sector to be used by the Flexible Disk Confidence Test. Minimum value is 0, maximum value is 128. By default, Sector Size is 128.

LENGTH Command



LENGTH specifies the number of data bytes to be read or written by the Flexible Disk Confidence Test. Minimum value is 0, maximum value 512. By default, length is 128.

OP Command



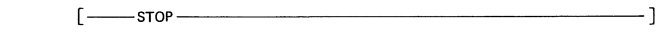
OP specifies an operation to be performed by the Flexible Disk Confidence Test. The operation can be specified by its OP name or Hex OP code.

Operation	OP Name	Hex OP Code
TEST	TEST	@2000
RESTORE	REST	@4000
READ	READ	@6000
WRITE	WRIT	@8000
INITIALIZE	INIT	@A000
RDCK	RDCK	@C000

RUN Command

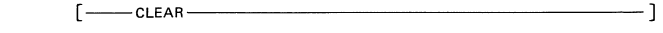
RUN causes a particular test section to begin, or if the section was interrupted by an ODT keyboard input, to continue.

STOP Command



STOP instructs the program to halt. STOP enables the operator to restart the specified execution statement.

CLEAR Command

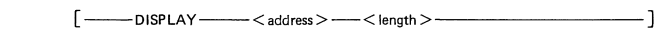


CLEAR restarts the DOI. All flags and commands are reset to their default value.

STATUS Command

STATUS displays the specified value of the REPEAT command and the number of repeats remaining. DOI SET/RESET options and command states are also displayed.

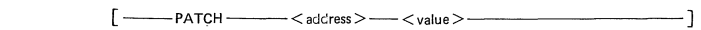
DISPLAY Command



DISPLAY allows the user to display up to 10 memory or memory-mapped I/O locations.

Minimum Values	Maximum Values
<address> = 0	<address> = @FFFF
<length> = 1	<length> = 10

PATCH Command



PATCH allows the user to write any data into any UC memory location. Address and value are decimal unless preceded by @. For example, PATCH 255 23 or PATCH @FF01 @1A

CAUTION

Data written into RAM locations can modify the DOI or test section program code. These modifications can prevent use of the DOI or cause false errors to be detected.

Special ICMD Features

The following special functions can be performed by the UC when appropriate software is loaded.

- Mini-disk initialization.
- Write protecting previously created mini-disks.
- Design Level-2 I/O maintenance.
- Remote Diagnostics.

MINI-DISK INITIALIZATION

The UC can initialize mini-disks with the system either running on-line or halted. However, on-line initialization requires the use of the ODT for approximately 3 minutes per mini-disk.

To initialize mini-disks:

1. Insert mini-disk labeled DIAG41 into the ICMD Drive.
2. Place the system in the UC mode by depressing the MODE pushbutton on the keyboard. Depress the RESET pushbutton. After approximately 20 seconds the mini-disk loads, and the ODT displays a test status line (referred to as the DOI).
3. When the DOI displays TEST HCP SECTION 0 IDLE, enter the syntax TEST FD. The Flexible Disk Confidence Test is then loaded into UC memory.
4. After the test status returns to IDLE, enter the syntax SEC 6 to select the initialize routine.

5. Remove the DIAG41 mini-disk, and insert the mini-disk to be initialized into the ICMD drive. Enter the syntax RUN. The ODT displays BOJ SECTION 6. When the ODT displays EOJ SECTION 6 the mini-disk is initialized. If more than two tracks are found in error, the mini-disk is marked bad and the operator is so informed.

WRITE PROTECTING MINI-DISKS

The following procedure provides instructions for write protecting a mini-disk. Once a mini-disk is write protected, it cannot be purged, or written to.

NOTE

Do not write protect PANEL or PANDLP mini-disks. These mini-disks contain options, written by the UC, which can require changes.

The UC can write protect mini-disks with the system running on-line or with the system halted. However, on-line write protection requires the use of the ODT for approximately two minutes per mini-disk.

To write protect mini-disks:

1. Insert mini-disk labeled DIAG41 into the ICMD Drive.
2. Place the system in the UC mode by depressing the MODE pushbutton on the keyboard. Depress the RESET pushbutton. After approximately 20 seconds the mini-disk loads, and the ODT displays a test status line (referred to as the DOI).
3. When the DOI displays TEST HCP SECTION 0 IDLE, enter the syntax TEST FD. The Flexible Disk Confidence Test then is loaded into UC memory.
4. After the test status returns to IDLE enter the syntax SEC 7. This selects the write protect routine.
5. Remove the DIAG41 mini-disk, and insert into the ICMD drive the mini-disk to be write protected. Enter the syntax RUN. The ODT displays BOJ SECTION 7. When the ODT displays EOJ SECTION 7 the mini-disk is write protected.

DL-2 I/O MAINTENANCE

I/O maintenance is performed either off-line or on-line. The off-line diagnostics are executed by loading the Test Bus Control program into UC. On-Line diagnostics are accomplished by the normal state program, DIAGNO. DIAGNO uses the Test Bus Interface Port of the UC. Consequently, the PANDLP mini-disk must be loaded into UC memory. Instructions for both diagnostics are located in the I/O section of this handbook.

Remote Diagnostics

The B 2900 System has Remote Diagnostics capabilities built-in. A link can be established by using a remote cable kit and the proper UC software (through data sets and a voice grade phone line), with a remote terminal.

After the link has been established, the remote terminal has all the capabilities of the UC ODT, with the exception of executing UC and DLP Diagnostics. The following functions can be performed remotely.

Run/stop control of system.
Clock control (Single clock, clock burst, and single instruct).
Maintenance Panel.
Viewing or setting any string (or string term) in the system.
Control state program initialization.
Reading and writing memory.
Communication between terminals by special message syntaxes.

The remote terminal should have all system T & F documentation available. Micro-code listings, test listings, and the Diagnostic Operators Listing (DOL) are essential for troubleshooting system problems.

SITE REQUIREMENTS

The following is a list of requirements for the system.

B 2900 at firmware level ABH or higher.
Anderson Jacobs 1245 or 1200 Acoustic Coupler (or equivalent).
Remote Diagnostic Cable Kit (P/N 1979 9766). This kit allows connection of the UC to a standard data set. The kit can be ordered through Product Distribution.
A standard data set cable (P/N 1144 1511).
A telephone accessible by the data set and cable.

SITE INSTRUCTIONS

To prepare the system for remote link:

1. Power down the system.
2. Remove the DLP/IF Card.
3. Remove the jumper from E3 pin 5 to E3 pin 12.
4. Install a jumper from E3 pin 6 to E3 pin 12.
5. Ensure the following jumpers are installed:

   E3 pin 3 to E3 pin 15
   E3 pin 8 to E3 pin 9

NOTE

Changing the BAUD rate of the RML port, causes non-resident diagnostics to fail in RML test, Section 5, Error 34. If error-free testing is required, reverse the above procedure.

6. Connect the ribbon cable (from kit) between the cable connector board and J2 (pins 25 to 49) of the DLP/IF board.
7. Configure the acoustic coupler as follows.

   a. AJ1200 couplers:

      1) Place DAA switch (on rear of unit) to HIGH.

   b. AJ1245 couplers:

      1) Place the MODE switch (on front of unit) to 202.
      2) Place rear switches 5 and 6 On, switches 1, 2, 3, 4, and 7 Off.

NOTE

Switches 1, 2, and 3 can be changed to correct transmission or receiving errors encountered on the telephone line.

      3) Enable internal options CTS,CR12, and MCD; disable internal options TOD. For instructions on these options refer to the AJ1245 manual.

8. Connect the data set cable between the cable connector board and the acoustic coupler.

REMOTE REQUIREMENTS

The following is a list of requirements for the remote terminal.

1. A TD830, TD850, or MT983. If the MT983 is used, it must have 2.0 ODT firmware or higher.
2. A Bell* 202C/202S type modem (or equivalent).
3. A Bell cable to connect the display unit to the modem. Part numbers of available cables are listed below.

   a. 15 foot cable P/N 1696 4975.
   b. 25 foot cable P/N 1696 4983.
   c. 50 foot cable P/N 1696 4991.
   d. 100 foot cable P/N 1696 2946.

4. A telephone accessible by the modem and Bell cable.

*Bell and design is a registered trademark of American Telephone and Telegraph Company

REMOTE INSTRUCTIONS

To prepare for connection to Remote Link:

1. Configure the display unit as follows.

   a. TD830 terminals:

      1) Remove PIA board. Lift pin 5 on IC A65. Lift pin 13 on IC A30. Disconnect resistors R27, R28, and R29. Ensure that IC A13 and IC A14 are removed. Ensure that all other IC pins are connected. Reinstall PIA board.
      2) Configure the terminal as follows.

```
0080  42  D1  00  00  17  4F  1F  1E
      00  00  00  04  09  00  04  04

0090  00  2F  CC  17  4F  FF  00  4F
      17  40  00  08  03  50  0F  FF

00A0  54
```

   b. TD850 terminals:

      1) Remove the Data Comm board. Set SW3, SW4, SW6, and SW9 of SWC8 to the On position. Set SW1, SW2, SW5, SW7, and SW8 to the Off position. Reinstall the Data Comm board.
      2) Configure the terminal firmware as follows.

```
0080  48  D1  00  00  17  4F  7B  7D
      00  00  00  04  09  00  04  04

0090  00  17  5C  17  07  80  00  4F
      17  00  00  08  00  00  07  85

00A0  D0
```

   c. MT983 terminals:

      1) Remove the SD2 board. Set SW3, SW4, SW5, and SW6 on SA01 to the On position. Set SW1, SW2, and SW7 to the OFF position. Set all switched on SA02 to the Off position. Reinstall the SD2 board.
      2) Configure the terminal firmware as follows.

```
0080
0090
00A0
```

2. Connect the Bell Cable between the display unit and the modem.

Operating Syntaxes

Syntaxes for Remote Diagnostics are in two categories. Maintenance commands, such as MP and DIS Y, are entered by the remote operator as defined in the DOL. Special syntaxes have be added for control of the RML.

All syntaxes transmitted are echoed back to the source, and are displayed on the other terminal.

The special remote diagnostic syntaxes are detailed below.

[A: <any text> – Activate the remote link.
[D: <any text> – Deactivate the remote link.
[R: <any text> – Message from UC ODT to Remote terminal.
[L: <any text> – Message from Remote terminal to UC.
[T: <any text> – Display status of the Remote Link.

ACTIVATING REMOTE LINK

The Remote Link can only be activated from the UC. After a reasonably good telephone line is established, both operators should agree on a maximum amount of time to attempt connection. If a connection is not made in the specified time, then corrective action is required (see Troubleshooting Connection Problems section).

To activate Remote Link:

1. Establish voice communication on a reasonably good telephone line.
2. Load RML Peripheral Device Firmware (PDF) by loading PANEL, STRUCT, or module test mini-disk into UC memory.
3. Place the telephone in the coupler (activate modem).
4. Activate Remote Link by system operator using the syntax, [A:. If Remote Link becomes operational, both terminals display the message RL:ACTIVE. The remote operator can then enter maintenance commands or special commands.

   If the Remote Link is not activated, the reason is displayed on the UC ODT. Ensure that all cables are installed correctly, and that the modem/coupler is connected properly. Retry the connection. If the error persists, see Troubleshooting Connection Problems section.

REMOTE LINK STATUS MESSAGES

If an error is encountered by the Remote Link, the RML PDF will retry the operation three times. If the error still exists, a status message is displayed on the UC ODT. Persistent status messages that prevent normal RML communication require corrective action as defined below.

The status messages are displayed in the following format.

RL STAT nnnn

where nnnn = status of RML

The status is in hexadecimal and each bit is defined below. Status bit 0 is the least significant bit; status bit 15 is the most significant bit.

Status bit 0 = Timeout. The remote terminal did not respond in time. Corrective action: Check remote terminal firmware configuration. If correct, replace the remote coupler and then the terminal.

Status bit 1 = Request to send. Received request to send from the remote terminal. This status is not an error condition.

Status bit 2 = Exception bit. Status bit 2, 3, 4, 8, 9, 10, 11, 12, or 15 is on.

Status bit 3 = Nak received. The remote terminal responded with a NAK. Corrective action: Ensure remote terminal is in Receive mode. If correct, replace the terminal and then the remote coupler.

Status bit 4 = BCC error. A mismatch occurred during comparison of the Block Check Character. Corrective action: Check remote terminal firmware configuration. If correct, replace the remote coupler and then the system modem.

Status bit 5 = Line discipline. The text received was framed with improper characters. Corrective action: Check the remote terminal firmware configuration. If correct, replace the remote coupler and then the system modem.

Status bit 6 = Short buffer. The message was truncated on the read operation.

Status bit 7 = Invalid OP. OP Code received by PDF is invalid. Corrective action: Run UC Non-resident diagnostics.

Status bit 8 = Transmitter empty. The hardware transmit buffer is ready. This status is not an error condition.

Status bit 9 = Character ready. The RML USART has received a character. This status is not an error condition.

Status bit 10 = Transmitter empty. All characters have been transmitted from the hardware transmit buffer. This status is not an error condition.

Status bit 11 = Parity Error. A parity error was detected by the RML USART. Corrective action: Run UC Non-resident diagnostics. If no error is found, replace the remote coupler and then the system modem.

Status bit 12 = Overrun error. A character in the RML USART was overwritten due to late service. Corrective action: Run UC Non-resident diagnostics. If no error is detected, replace the DLP/IF the Microcomputer cards.

Status bit 13 = Framing error. RML USART is not receiving valid stop bits. Corrective action: Check the remote terminal firmware configuration. If firmware is correct, replace the remote coupler. If the error is still present, replace the system modem.

Status bit 14 = Invalid data. An ETX character was detected in the write buffer. Corrective action: Run the UC Non-resident diagnostics. If no error was detected, replace the system data set and/or cables. If error is still present, replace DLP/IF and Microcomputer cards.

Status bit 15 = Data Set Ready. The terminal is ready and in the receive mode.

TROUBLESHOOTING CONNECTION PROBLEMS

If the RML is activated, and a connection is not made, the UC ODT displays the fault. The ODT displays information concerning data set status, transmit status, and timeout.

If the connection is not made, ensure that all cables are installed properly. Check the configuration of the remote terminal, and couplers/modems. Reload the RML PDF from the mini-disk and re-activate the RML. If connection is not made, place the remote terminal into monitor mode by using the following procedure.

1. Enter CTRL, SHIFT (must be held down) R W M O D E.
2. Enter CTRL, SHIFT (must be held down) R H (release SHIFT key) 0 0 7 F.
3. Enter F0 at address 7F as below.

   007F F0

4. Enter CTRL, SHIFT (must be held down) R C X X. A check mark will appear in the lower left corner of the ODT screen.
5. To view message coming in, depress the LOCAL key. Page advance to page 2 by entering CTRL (right arrow). All data transmitted and received will be displayed on the ODT screen.
6. Reactivate the link.

During a good connection the sequence of events is as follows.

1. Carrier light on coupler flashes on and off.
2. LTAI light on keyboard illuminates.

3. Data as received and transmitted is displayed on page 2 of remote terminal. The data should be:

   ENQ ACK | ENQ ACK | ESC SEQUENCES RL:ACTIVE ACK |

If the terminal monitor fails to display any data received, check that data is actually being transmitted from the system. This can be done by picking up the phone receiver and having the system operator reactivate the link. If data is being transmitted, audible tones are present. If the tones are heard, suspect the problem to be in either the modem or terminal.

If no tones are heard, the system operator should check the coupler. This can be done by removing the handset and activating the link. Audible tones should be present. If no tones are present, suspect a problem in either the coupler or UC.

If garbled data is displayed on the monitor, change equalization switches (switches 1,2 and 3) on the system coupler. If this does not correct the problem, suspect the problem to be in either the coupler or modem.

ICMD Exerciser

The console resident diagnostics provide the capability to exercise the ICMD through the flexible disk drive Maintenance Test Routine (MTR). However, there are no means to create descriptors and issue operations to the disk through the ICMD Peripheral Device Firmware (PDF) if difficulties occur while attempting to load a diskette.

The following instructions describe the operations required to create a short program that can call on sections of the ICMD PDF which can operate on the disk. Sufficient RAM space must be allotted to handle both the program and assigned stack areas.

PROGRAM REQUIREMENTS

The program, using a minimal amount of 8080 code, can be created by taking advantage of certain console MTR features. These features provide the ability to:

- Execute direct from RAM.
- Use the ICMD PDF.
- Use the console lamps.

PROGRAM SPACE

Four areas must be assigned for use by the program:

- PDF descriptor area (six fields where each field is 16 bits long).
- Stack area (for call returns).
- Program area (variable, depending on the length of the program).
- Storage area (for data).

PDF DESCRIPTOR AREA

The PDF descriptor area is 12 bytes in length and is divided into the following subareas (assuming that the address for the descriptor begins at location 0200 Hex):

Table 2-12. PDF Descriptor Subareas

Field	Location	Definition
Status	0200	MSB of status word
	0201	LSB of status word
OP	0202	LSB of OP-code
	0203	MSB of OP-code
A	0204	LSB of memory address for data transfer
	0205	MSB of memory address for data transfer
B	0206	LSB of length in bytes of data transfer
	0207	MSB of length in bytes of data transfer
C	0208	Sector address (Valid sector numbers are 1-26)
	0209	Track address (Valid track numbers are 0-77)
D	020A	MSB of extended status
	020B	LSB of extended status

The meanings of the status fields are found on pages 10 and 11 of the ICMD PDF listing.

The values for the OP-codes are found on pages 5 through 9 of the ICMD PDF listing.

The program, descriptor, and stack areas can be moved to any location in RAM. Address modification has to be performed in order to accomodate the relocation. If the program is relocated, a jump instruction must be written at addresses 0001-0003 (Hex) since a reset causes the microprocessor to start executing at address 0001 (Hex).

PDF ENTRY POINTS

Two routines within the ICMD PDF are used to access the disk. One of the routines initializes the ICMD Interface and the other routine issues descriptors to the ICMD PDF for execution.

The two routines, descriptor, stack, and address of the Maintenance Panel LED are defined in the following list.

Label	8080 Instruction	Comment
Stack	Equate 00090	Assign Stack Area
Lights	Equate 0EF04	Address of LEDS
Descriptor	Equate 00200	Descriptor Begin Address
ED-FD-Initial	Equate 0FA10	ICMD PDF Entry Point Address
ED-FD-Drive	Equate 0FA41	ICMD PDF Entry Point Address

This program assumes that the ICMD PDF entry points remain as currently defined. If the ICMD PDF is ever reassembled and new PROM are issued, the entry point addresses may have to change.

NOTE

PDF automatically overwrites certain descriptor fields. Therefore, shortening of the machine language program is inadvisable.

Assembler Program

The 8080 assembler program is entered through the Console Maintenance Panel to exercise the ICMD drive.

Address	Code	8080 Instruction	Comment
0001	C3	JMP START	Push START to jump
0002	00		to start of program
0003	01		
0100	START = F3	DI	Disable interrupts
0101	31	LXI SP, STACK	Initialize stack pointer
0102			LSB of Stack Address
0103			MSB of Stack Address
0104	CD	CALL ED-FD-INITIAL	Initialize ICMD Interface
0105	10		LSB of PDF Entry Point
0106	FA		MSB of PDF Entry Point
0107	LOOP = 01	LXI B, DESCRIPTOR	Put Descriptor Addr in B,C
0108	00	ADDRESS	LSB of Descriptor Address
0109	02		MSB of Descriptor Address
010A	3E	MVI A, 1	Load Sector Number
010B	01		
010C	32	STA	Store Sector Number
010D	08		LSB of Descriptor Address
010E	02		MSB of Descriptor Address
010F	3E	MVI A, 3	Load Data Buffer Addr MSB
0110	03		
0111	32	STA	Store Buffer Address MSB

Address	Code	8080 Instruction	Comment
0112	05		LSB of Descriptor Address
0113	02		MSB of Descriptor Address
0114	3E	MVI A, 0	Load Data Buffer Addr LSB
0115	00		
0116	32	STA	Store Buffer Address LSB
0117	04		LSB of Descriptor Address
0118	02		MSB of Descriptor Address
0119	3E	MVI A, 0	Load Track Number
011A	00		
011B	32	STA	Store Track Number
011C	09		LSB of Descriptor Address
011D	02		MSB of Descriptor Address
011E	CD	CALL ED-FD-DRIVE	Issue descriptor to PDF
011F	41		LSB of PDF Entry Point
0120	FA		MSB of PDF Entry Point
0121	3E	MVI A, 55	Load hold code of 55
0122	55		into accumulator
0123	32	STA LIGHTS	Put 55 in LEDS
0124	04		LSB of panel LEDS
0125	EF		MSB of panel LEDS
0126	D3	HOLD	Stop the processor to
0127	01		look at result status
0128	C3	JMP LOOP	Push START to send out
0129	07		descriptor again
012A	01		
0202	00		LSB od OP-Code
0203	74		MSB of OP-Code
0206	80		LSB of bytes to transfer
0207	00		MSB of bytes to transfer

Operating Instructions

1. Set single cycle switch ON.
2. Press RESET.
3. Enter machine language program via maintenance panel.

NOTE
The RAM must be enabled first by entering 06 in MCR (address EF02). This also enables DMA for ICMD operations.

4. Reset SINGLE CYCLE switch.
5. Press START.
6. If the program runs correctly, a descriptor is issued to the ICMD PDF and the program stops with 55 indicated by the LED display (set the DATA BUS/MTR REG switch to MTR REG).
7. To repeat operation, press START again.

The data field can be examined by reading 0300-0380 (HEX).

The result status field can be examined by reading 0200-0201 (HEX).

Options

Four options are possible as listed below.

SELECT TRACK

To seek to a track other than 00, enter track number in hex at address 011A (hex).

E.G. Track 38 = 26 (hex), Track 76 = 48 (hex)

CONTINUOUS READ

To do a continuous read, enter 00 (hex) at addresses 0126 and 0127.

OP CODE

To select OP code other than 74 (read OP using DMA), enter required OP code at address 0203 (hex).

SECTOR NUMBER

To select sector other than 01, enter required sector number in hex at address 010B (hex).

Quick Entry

Loading program sequentially by addresses is time consuming. To speed up entry of a program, write all the addresses with identical data at the same time.

For example: Load Data Register with 02 (hex). This can then be written into addresses 0109, 010E, 0113, 0118, and 011D without reloading the data register.

CODE	ADDR	DATA	ADDR	DATA	ADDR
06	EF02	07	0129	55	0122
00	0002	08	010D	74	0203
	0103				
	0108				
	0115				
	011A				
	0202	09	011C	80	0206
	0207				
01	0003	10	0105	90	0102
	0107				
	010B				
	0127			C3	0001
	012A	31	0101		0128
02	0109	32	010C	CD	0104
	010E		0111		011E
	0113		0116		
	0118		011B		
	011D		0123	D3	0126
03	0110	3E	010A	EF	0125
			010F		
			0114		
04	0117		0119	F3	0100
	0124		0121		
05	0112	41	011F	FA	0106
					0120

1. Insert SMV non-resident diagnostic diskette.
2. Enter TEST FD (wait until idle).
3. Insert scratch diskette.
4. Enter SEC 6;RUN (initializes diskette).
5. Enter SEC 0;OP WRIT;DATA @63;END TRK 0;RUN (writes @63 track 00).
6. Enter DATA @FF;BEG TRK 38;END TRK 38;RUN (writes @FF track 38).
7. Enter DATA @00;BEG TRK 75;END TRK 75;RUN (writes @00 track 75).

NOTE

Initialization data pattern is @E5.

ICMD Seek Verification Using Resident MTR

The last eight columns represent display lights which are read in hex.

Table 2-13. Console Maintenance Panel Display Lights

Operation	Procedure	7	6	5	4	3	2	1	0
SELECT MTR	Set SWITCH 1 and press RESET				X	X	X	X	X
SEEK TRACK 00	Set SWITCHes 5 & 6 and press START				X				X
LOAD HEAD	Set SWITCH 2 and press START				X		X		X
SEEK TRACK 38	Set SWITCH 5 and press START								X
LOAD HEAD	Set SWITCH 2 and press START						X		X
RESTORE TO 00	Set SWITCHes 5 & 6 and press START				X				X
LOAD HEAD	Set SWITCH 2 and press START				X		X		X
SEEK TRACK 75	Set SWITCH 6 and press START								X
LOAD HEAD	Set SWITCH 2 and press START						X		X
LOAD HEAD	Set SWITCH 2 and press START				X		X		X

Power Verification

Power verification is accomplished as follows.

BLOWERS

Verify that all cabinet blowers are operating.

VOLTAGE VERIFICATION

Check to see that the following voltages are available.

NOTE

If voltage adjustments are required, refer to B 2900 Power Subsystem FETM #1115706.

185 VDC

WARNING

A battery operated instrument must be used for measurement of HVDC.

Measure the HVDC between pins 1 and 3 of connector J1 on the back of the 5KW Power Converter. Ensure that HVDC is 185 VDC (+ or − 5V) before proceeding.

24 VDC and 12 VDC

Verify the voltages on the Supervisory Module. Measure voltages with a DVM at the test points marked on the CPS/24V Regulator Board which is located directly behind the Supervisory Module Display Panel. Remove

the air flow grill, located directly above the module, to access the test points adjustment potentiometers. The grill is not secured and is lifted straight up without the use of any tools.

12 V CPS/24 V Voltage Adjustment

Test Point	Adjusted Setting
24 VDC	24.1 VDC
12 V CPS *	12.1 VDC

* Control Power Supply

+5 VDC

Verify +5 VDC power module(s) output by checking voltage at the output bus bar. The voltage should be 5.1 VDC. If the voltage is over 5.2 VDC or below 5.0 VDC, an adjustment is required.

−2 VDC

Verify −2 VDC power module output by checking voltage at the output bus bar. The voltage should be −2.1 VDC. If the voltage is over −2.2 VDC or below −2.0 VDC, an adjustment is required.

SECTION 3

DLP DIAGNOSTICS

This section provides the operating instructions for DL-2 DLP Diagnostics. The instructions are given for DIAGNO (On-Line) and for the Test Bus Control Program (Off-Line). The DIAGNO program and mini-disk image files are available on Test Routine release TN815O. The Test Bus Control program and associated mini-disks are shipped with the T&F documentation for the B 2900 system.

On-Line DLP Diagnostics Using DIAGNO

DIAGNO is an On-Line normal state program used to run diagnostics on DL-2 DLP, Maintenance Cards, and Distribution Cards. All commands are entered through the ODT (Operator Display Terminal). Any incorrect operator input results in the ODT displaying an error message and a self-explanatory corrective action required.

The diagnostic test cases are loaded from mini-disk images located on system disk. The file names and part numbers of all mini-disk images are listed in the TNDOC (located on the TN815O tape).

NOTE: The installation to or removal of a DLP from an active DLP base, will corrupt data within the base and cause a system failure.

Executing DIAGNO/

1. DIAGNO and all desired diagnostic mini-disk images must be loaded to system disk. A syntax example follows.

   LOAD TN815O DIAGNO CRDDN/
   (CRDDN/ = Card reader diagnostic mini-disk images)

2. The Console DLP and Test Bus Interface must be dedicated to the system at Coldstart/Warmstart time or by using the syntax:

   DLP CC (CC = Console DLP channel number)
   UNIT CC/1 NST (CC = Console DLP channel number)

3. The channel number of the unit to be tested must be inhibited by using the syntax:

   XC+TT (TT = Tested unit channel number)

4. Execute the program by using the syntax:

   EX DIAGNO

5. The program begins (BOJ DIAGNO) and requests the Console DLP channel number. Enter the channel number with the syntax:

   <mix no.> AX CC (CC = Console DLP channel number)

6. The program responds with CC/1 NST SAVED, BEGIN OF UNIT TEST, and ENTER FILE ID. Enter the desired diagnostic mini-disk image, such as CRDDN1, by using the syntax:

   <mix no.> AX [file name]

7. On the ODT, the program displays the 8-digit part number and title of the mini-disk image. Ensure that the mini-disk image is correct by referencing the Diagnostic Index in the T&F documentation for the unit under test.

8. The program displays an AWAITING DIRECTIVES message. Operator input is required before the program continues. The recommended order of the commands follows.

   a. <mix no.> AX MC COMMAND
   b. <mix no.> AX UNIT COMMAND
   c. <mix no.> AX [any desired options]

   All commands are listed in the User Commands section. Some commands are valid only for DIAGNO and are so specified.

9. After all commands are entered, begin diagnostic test cases by using the syntax:

   <mix no.> AX GO

   NOTE

   Once the test cases have been initiated by the GO command, user commands have no effect. To use the commands, the program must be interrupted by using the syntax:

   <mix no.> SW8 1

10. The program informs the operator of next test case to be executed. If an error is found in the connection process or in the unit under test, a message stating the error is displayed. For interpretation of the message and corrective action, refer to the Error Display and Fault Correction sections.

If no error is found, completion of the test is indicated by the message NEXT TEST CASE = 1, AWAITING DIRECTIVES. To terminate the program use the syntax:

<mix no.> AX QUIT

To continue to the next mini-disk image, use the syntax:

<mix no.> AX END

The program then returns to step 6, and a new mini-disk image file-id can be entered.

Off-Line DLP Diagnostics Using Test Bus Control Program

To test DL2 DLP, Maintenance Cards, and Distribution Cards from the Off-Line mode it is necessary to load the Test Bus Control program into the Universal Console. This program controls the Test Bus Interface and all communication to and from the unit being tested. Test cases are loaded from mini-disk under program control.

Executing Test Bus Control Program

If the system is running, it must be halted before loading the control program. Place the system in Console mode by depressing the MODE pushbutton. Depressing the SPCFY key then halts the system.

1. To load the control program, insert mini-disk labeled TBC43 into the mini-disk drive. Place the system in the Console mode by depressing the MODE pushbutton. Set the Console Maintenance Panel switches to the following positions:

   Input Switches 0-15 in the down position.
   Trace/Normal switch to Normal.
   Data Bus/MTR Reg switch to MTR Reg.
   Clear Proc Only/Entire Console switch to Entire Console.
   Single Cycle/Run switch to Run.
   Panel Enable/Disable switch to Enable.

2. Depress the RESET pushbutton. The Universal Console executes the resident diagnostics before loading the mini-disk. If the FAULT lamp on the keyboard fails to extinguish after 30 seconds, an error condition exists in the Universal Console (refer to Universal Console T&F Vol. 1). If the resident diagnostics run successfully

(test runs approximately 20 seconds), the mini-disk loads, and the ODT displays:

```
BEGIN OF UNIT TEST
    > AWAITING DIRECTIVES
```

3. The message above indicates that the program is waiting for input from the operator before continuing. All commands are entered through the keyboard. The format for all commands are listed in the User Commands section. Some commands are valid only for DIAGNO and are so specified.

   At this time the following are selected:

   a. Maintenance card address (use MC COMMAND)
   b. Tested DLP unit number (use UNIT COMMAND)
   c. Any option commands listed in the User Commands section.

The GO command allows the program to continue.

4. The program displays LOAD DISK, ENTER GO. Place the desired mini-disk, such as Printer Diagnostic DISK01, into the mini-disk drive. Enter GO to continue.
5. The title and part number of the mini-disk are now displayed on the ODT. Ensure that the mini-disk is correct for the unit by referencing the Diagnostic Index in the T&F documentation for the unit under test.
6. The program displays RIGHT DISK? ENTER GO OR NO.

   a. If NO is entered the program returns to step 4.
   b. If GO is entered the program connects with the unit through the Maintenance Card, and begins executing test cases from the mini-disk. If a connection is made to the unit, the red lamp on the Maintenance Card plug-on is illuminated. If connection is not made, the reason is displayed on the ODT.

7. When all test cases from the mini-disk have completed with no errors, the ODT displays END OF TEST. The program then returns to step 4. If an error is encountered, the ODT displays a message describing the error. For problem analysis and corrective action, refer to Error Display and Fault Correction sections. Command options,such as loop instead of terminate, may affect the completion of the test cases.

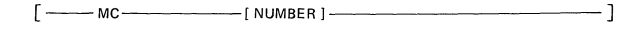
User Commands

The following section details all available commands used by DIAGNO and the Test Bus Control Program. Each command is given with a brief description.

AVAILABLE COMMANDS

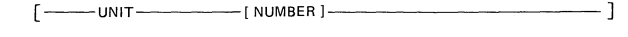
MC
UNIT
DISPLAY
END
GO
HALT
LOOP
NO
PAGE
RECONNECT
STATUS
STEP
TEST
QUIT
RUN
PRINT
TRACE

MC COMMAND



MC selects the Maintenance Card address. The address is strapped on the Maintenance Card plug-on (refer to Maintenance Card T&F documentation). The number must be between 0 and 63.

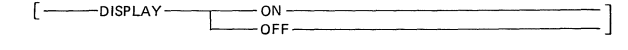
UNIT COMMAND



UNIT selects unit number as strapped on the DLP. The number must be between 0 and 31.

NOTE
The Distribution Card address (driven by PROM) is 16.

DISPLAY COMMAND



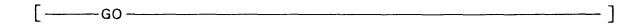
DISPLAY sets or resets the display option. If the option is set, all error messages are displayed. If the option is reset, only syntax error messages are displayed.

END COMMAND

For the Test Bus Control program, END directs the program to clear the unit, disconnect from the Maintenance Card and then restart.

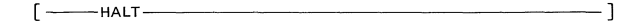
For DIAGNO, END directs the program to request a new mini-disk image file.

GO COMMAND



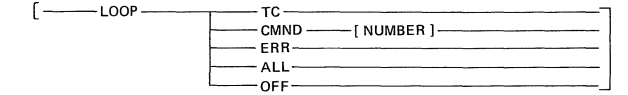
GO directs the program to continue.

HALT COMMAND



HALT causes the test case in progress to pause, allowing the operator to change options, restart, etc.

LOOP COMMAND



LOOP determines the looping method of the program.

1. LOOP:TC causes the program to continuously loop on the same test case.
2. LOOP:CMND causes the program to continuously loop on the same test case, starting with command 1 and ending with the command specified by [NUMBER].
3. LOOP:ERR causes the program to loop on the same test if an error is encountered. The loop continues until the test case runs without an error.
4. LOOP:ALL causes the program to continuously loop on the entire data base starting with test case 1 and ending with the last test case.
5. LOOP:OFF resets all of the above loop functions.

NOTE

LOOP:ERR can be set concurrently with any one of the other loop functions.

NO COMMAND

[——— NO ———————————————————————————]

NO is the negative response to the GO or NO message.

PAGE COMMAND

[———————— PAGE ———————— [NUMBER] ————————————]

PAGE directs the program to display current status of the registers in any of one of 16 pages.

RECONNECT COMMAND

[———————— RECONNECT ———————————————————————]

RECONNECT directs the program to reconnect to the Maintenance Card as specified in the previous MC command.

STATUS COMMAND

[———————— STATUS ———————————————————————]

STATUS displays the status of the control program. The following is a display of the status:

MC XX UNIT XX DSEL XX LOOP XXXX LOOP ERR XXX DISPLAY XXX STEP XXXXX TEST XXXXX CMND XXXXX FE BUFFER XX

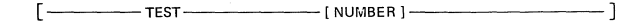
MC XX Current Maintenance Card address.
UNIT XX Current unit number address.
DSEL XX Current page number.
LOOP XXXX Current loop option (TC, CMND, ALL, OFF).
LOOP ERR XXX Loop error option ON or OFF.
DISPLAY XXX Display option ON or OFF.
STEP XXXXX Current value of step option (ON or OFF).
TEST XXXXX Current test case number.
CMND XXXXX Number of last maintenance command executed.
FE BUFFER XX Number of command in the FE Buffer.

STEP COMMAND

[———————— STEP ———————— [NUMBER] ————————————]

STEP sends the given number of commands to the Maintenance Card and then waits until directed to continue. If the step option is set, GO repeats the same action as stated above.

TEST COMMAND



TEST directs the program to start with this test case or discontinue present test case and begin this test case immediately. The number is the test case to be run.

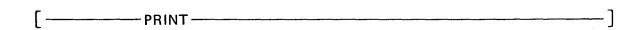
QUIT COMMAND

For DIAGNO use only. QUIT terminates the program.

RUN COMMAND

For DIAGNO use only. RUN combines the actions of the TEST and GO commands.

PRINT COMMAND



For DIAGNO use only. PRINT causes DIAGNO to place all program messages into a print file. The print file also contains status information and trace data (if trace option set).

TRACE COMMAND

For DIAGNO use only. TRACE causes DIAGNO to record all I-O activity for the Test Bus.

Error Display

If DIAGNO or the Test Bus Control program detects an error, the operator is notified. The error can be an interface error or an error detected in the unit under test.

An interface error displays a message describing the error on the ODT. The operator may try again, but persistent errors indicate a bad connection or hardware fault in the Universal Console or Maintenance Card. The Universal Console can be tested using non-resident diagnostics (refer to Universal Console T&F). The Maintenance Card can be tested using the Test Bus Control Program (or DIAGNO) and the proper mini-disks (images).

An error in the unit under test results in the Maintenance Card receiving data that was not expected. The program generates a message in the following format.

Line 1 contains:

1. Command number relative to test case.
2. Test case number.
3. Page number (group number).
4. Byte command in error.

Lines 2 through 9 contains flip/flop names with current bit settings and the expected bit settings. An asterisk denotes all bit settings not read as expected. NOTE: Line 2 is the least significant bit and line 9 the most significant bit.

Fault Correction

After an error is displayed, corrective action is obtained by referencing the T&F documentation for the unit under test. The Diagnostic Description section describes each of the test cases. The Fault Dictionary section details corrective action for the particular faults.

By using the test case number and the command number, a brief description of the test can be found in the Diagnostic Description section. Testing of individual components is accomplished by grouping several commands together. Consequently, if the exact command number is not listed, the next lowest command number is used.

For example, the ODT displays that in test case 1, command 47 is error. The following table indicates GPRIF logic is being tested.

```
01:00011 DISPLAY FAILURE
01:00024 MLI AND DLI ENABLED
01:00045 GPRIF LOGIC
01:00049 MLI AND DLI SIMULATED
```

By referring to the Fault Dictionary, the test case and command number specify a corrective action. Since testing is done by command groups, the exact command number may not be listed. If the exact command number is not listed, the next lowest command number is used.

For example, the ODT displays test case 1, command 47 is in error. From the table below the corrective action would be to replace chips N5 and G4 on the Common Front End board.

```
01:00011 REPLACE PROMS I2, N2. (CFE)
01:00024 REPLACE CHIP B2. (CFE)
01:00045 REPLACE CHIPS N5, G4. (CFE)
01:00049 REPLACE PROM A1. (CFE)
```

SECTION 4

UNIVERSAL I/O DLP

IOT

Table 4-1. IOT I/O Descriptors

Operation	Sys OP	Variants		Addresses
Read Extended R/D	70	00	00	A & B Addr Required
Conditional Cancel	71	00	CC	
Unconditional Cancel	72	00	CC	

B4900 Discontinue 73

NOTES

1. CC = Channel number to be canceled.
2. OP 70 is executed to the channel number specified by BF in the IIO instruction.

Result Descriptors

To determine the memory location of the Result Descriptor use the following formula.

(Channel number X 20) + 100

The memory location contains a 12 digit R/D in the following format.

IOT R/D	Descriptor Link	DLP R/D
C000	0200	8000

NOTE
The above R/D values are examples only.

	A	B	C	D
8	1 OP COMPLETE	5 DLP NOT PRESENT	9 EXTENDED R/D PRESENT	13 LPW ERROR
4	2 EXCEPTION	6 MEMORY PARITY ERROR (DOUBLE BIT)	10 ADDRESS PARITY	14 R/D PARITY
2	3 DLP TIMEOUT	7 BUFFER FULL	11 UNABLE TO CANCEL	15 ADDRESS MOD ERROR
1	4 DLP IN ILLEGAL STATUS	8 PORT BUSY	12 VERTICAL PARITY ERROR	16 C ADDRESS ERROR

NOTES

A8 = This bit is set unconditionally in every R/D.

A4 = This bit is set if any condition bits are set in the remainder of the IOT R/D or in the DLP R/D.

A2 = Indicates that while the IOT was connected to a DLP, the IOT failed to receive a response from the DLP after 16 IOT clock periods.

A1 = The connected DLP was found to be at an illegal status count as defined by the DLP common flow.

B8 = Upon attempting to connect to a DLP, the IOT detected a status count of 0 indicating the DLP is not installed or is in an off-line mode.

B4 = A memory parity error (double bit) was detected during data transfer to or from system memory.

B2 = There was no room available in the IIO buffer to accept the Initiate I/O command.

B1 = IOT connection to a DLP could not be made because the DLP base was in use by another system.

Figure 4-1. IOT Result Descriptors

C8 = More than one word (4 digits) of non-zero R/D was received by the IOT from the DLP. The additional words are stored in the IOT Scratchpad. To obtain the extended R/D, channel number locations 4 and 5 must be read. Enter the syntax:

```
ISPD *xx4 2
    where xx = 000B BDDD
        BB = DLP Base number.
        DDD = DLP channel number.
```

C4 = The IOT was unable to connect to a DLP because the DLP base detected incorrect (even) parity on the address transmission (DLP number).

C2 = This bit can only be reported at address 260 (IOT R/D location). It indicates that the IOT was unsuccessful in completing a DLP cancel descriptor.

C1 = Data received by the IOT from the DLP contained incorrect (even) vertical parity.

D8 = The longitudinal check word (LPW) received by the IOT from the DLP at the end of a data transfer did not agree with the LPW generated by the IOT.

D4 = The R/D received by the IOT from the DLP contained incorrect parity as indicated by the C1 or D8 bit.

D2 = A DLP which can only transfer an even number of bytes to/from the IOT was directed by the I/O descriptor to transfer an odd number of bytes. On data transfers to memory, the last byte of data is not stored. On data transfers from memory, some undetermined character is sent to the DLP.

D1 = A DLP requested a C address from the IOT, but the IOT was not provided with a C address in the I/O descriptor.

Figure 4-1. IOT Result Descriptors (Cont)

DATA LINK PROCESSORS (DLP)

GENERAL

Address jumpers and clock signal are common to all DLP with one exception; the SSP DLP jumpers differ from the rest of the DLP. The SSP address jumpers are found under the SSP heading.

Address Jumpers

Each DLP has address jumper pins for system initiated connections (LCPADn), DLP request jumper pins for DLP-initiated connections (LCPRQn) and maintenance address jumper pins for maintenance operations (MADn-m).

The three sets of address jumper pins are located on the Common Front End Card. The three addresses must have the same value.

Table 4-2. DLP Address Jumpers

Address Value	Logic Name	Jumper	
		From	To
DLP 0	LCPRQ 0	VP 51	VQ 55
	LCPAD 0	M5 I5	VQ 66
	MAD 0-8	VQ 67	VQ 70
DLP 1	LCPRQ 1	VP 51	VP 55
	LCPAD 1	M5 I5	VP 66
	MAD 1-9	VQ 67	VP 72
DLP 2	LCPRQ 2	VP 51	K5 I5
	LCPAD 2	M5 I5	VQ 65
	MAD 2-10	VQ 67	VQ 69
DLP 3	LCPRQ 3	VP 51	VQ 54
	LCPAD 3	M5 I5	VP 65
	MAD 3-11	VQ 67	VQ 68
DLP 4	LCPRQ 4	VP 51	VQ 53
	LCPAD 4	M5 I5	VQ 64
	MAD 4-12	VQ 67	VQ 76
DLP 5	LCPRQ 5	VP 51	VP 52
	LCPAD 5	M5 I5	VP 63
	MAD 5-13	VQ 67	VQ 73
DLP 6	LCPRQ 6	VP 51	VQ 52
	LCPAD 6	M5 I5	VQ 63
	MAD 6-14	VQ 67	VQ 75
DLP 7	LCPRQ 7	VP 51	VQ 51
	LCPAD 7	M5 I5	VP 62
	MAD 7-15	VQ 67	VQ 74

NOTES

1. For SSP DLP Address Jumpers, see under SSP heading.
2. An * in digit positions C and D indicates that these jumpers normally are not used for B 2900/B 3900 Systems.

Clock Signal

Verify that the following clock pulse is present and within the proper specifications.

Clock Signal	Test Point	Specification
CLOCK..0	048	8 MHz with 50V duty cycle

CARD READER DLP

Table 4-3. Card Reader DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Read DLP Buffer	40	*	8	4	0	0	A and B
Binary Unpacked		*	8	4	1	0	Addresses
Binary Packed		*	8	4	2	0	Required
Standard		*	8	4	4	0	
EBCDIC		*	8	4	8	0	
READ							
Card Standard		*	8	D	0	0	
Cards Standard		*	8	F	0	0	
Card Binary Unpacked		*	8	8	0	0	
Cards Binary Unpacked		*	8	A	0	0	
Card Binary Packed		*	8	8	2	0	
Cards Binary Packed		*	8	A	2	0	
Card EBCDIC		*	8	C	0	0	
Cards EBCDIC		*	8	E	0	0	
ECHO	48						
Standard		*	1	D	0	0	
Binary Unpacked		*	1	8	0	0	
Binary Packed		*	1	8	2	0	
EBCDIC		*	1	C	0	0	
TEST	44	*	2	9	0	0	
Ignore Data to EOF (Standard)		*	2	B	0	0	
Ignore Data to EOF (EBCDIC)		*	2	A	0	0	
Wait For Ready		*	2	0	0	0	
Wait for Not Ready		*	2	4	0	0	
Conditional Cancel		*	2	8	0	0	
ID		*	2	C	0	0	

NOTES

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:
  - 8-bit = Inhibit data transfer to memory
  - 4-bit = Reserved
  - 2-bit = ASCII translation
  - 1-bit = Reserved

	A	B	C	D
8	1 NOT READY	5 VALIDITY CHECK	9 RAM PARITY ERROR	13 ZERO
4	2 DESCRIPTOR ERROR	6 CONTROL CHARACTER DETECTED	10 CON-DITIONAL CANCEL COMPLETE	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 READ CHECK	11 INCOMPLETE CARD READ	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 ZERO	12 ZERO	16 ZERO

Figure 4-2. Card Reader DLP Result Descriptor

A				B				C				D			
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-3. Test/ID Result Word

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-4. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

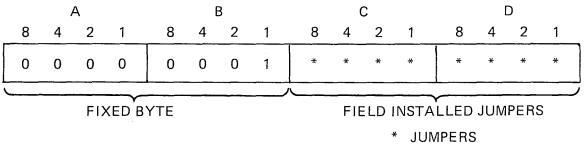


Figure 4-4. Card Reader ID Word

The Test/ID jumpers are located on the Peripheral Dependent Card.

Table 4-4. Test/ID Jumpers

Digit Bit	Bit Value	Jumper From	Jumper To
C8	128	VG 03	VG 04
C4	64	VH 03	VH 04
C2	32	VG 08	VG 09
C1	16	VH 08	VH 09
D8	8	VG 13	VG 14
D4	4	VH 13	VH 14
D2	2	VG 18	VG 19
D1	1	VH 18	VH 19

Translation Jumper

The Card Reader DLP supports EBCDIC and one of the three standard card codes, BCL, ICT, and BULL. If BULL is specified, it must be jumpered as follows:

VE 23 to VE 24

If ICT or BCL PROMS are inserted, this jumper must not be installed.

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-5.

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-6.

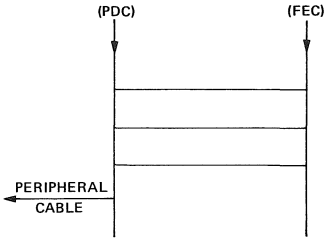


Figure 4-5. Card Reader DLP Frontplane Connectors

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND

Figure 4-6. Card Reader DLP Voltage Test Points

CARD PUNCH DLP

Table 4-5. Card Punch DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
WRITE	42						A and B
Card, in buffer		*	4	0	N	0	Addresses
Cards, binary unpacked		*	4	2	N	0	Required
Cards, binary unpacked		*	4	3	N	0	
Card, standard		*	4	4	N	0	
Cards, standard		*	4	5	N	0	
Card, binary packed		*	4	6	N	0	
Cards, binary packed		*	4	7	N	0	
Card, EBCDIC		*	4	8	N	0	
Cards, EBCDIC		*	4	9	N	0	
Test	44	*	2	0	0	0	
TEST							
Wait Ready		*	2	2	0	0	
Wait Not Ready		*	2	4	0	0	
Conditional Cancel		*	2	8	0	0	
ID		*	2	C	0	0	
ECHO	48	*	1	8	0	0	

NOTES

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

  8-bit = Inhibit data transfer to memory

  4-bit = Reserved

  2-bit = ASCII translation

  1-bit = Reserved

N designates bits used for encoded information. See following for permissable formatting operations.

Table 4-6. Card Punch DLP Encoded Information

N	Comments
8	Use auxiliary Stacker.
4	The character in column 1 is replaced by a control character. This variant results in a 1-2-3 punch in column 1 of the current card, or the first card of a multiple card operation.
2	Indicates that one retry is attempted, if a punch error occurs. The card in error is directed to the error stacker. If the retry is successful, then the successful-punch-retry bit is set in the result descriptor. If the retry is not successful, the punch-check-retry-successful bit is set in the R/D, and both cards are directed to the error stacker. If bit 2=0 and a punch error occurs, the punch check-no-retry bit will be set in the R/D and the card will be directed to the error stacker.
1	Indicates that the occurrence of a delimiter character is an error. If it occurred in the data stream, the operation halts without punching the card.

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

	A	B	C	D
8	1 NOT READY	5 SUCCESSFUL PUNCH RETRY	9 ZERO	13 ZERO
4	2 DESCRIPTOR ERROR	6 PUNCH CHECK NO RETRY	10 CON-DITIONAL CANCEL	14 ZERO
2	3 VERTICAL PARITY ERROR	7 PUNCH CHECK RETRY UN-SUCCESSFUL	11 INVALID CHARACTER	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR	8 ZERO	VERTICAL[12] PAR ERR (DLP RAM BUFFER) OUTPUT NOTE	16 ZERO

NOTE

EVEN vertical parity was detected during transmission to the punch or Even parity was returned to the host system during an ECHO operation.

Figure 4-7. Card Punch DLP Result Descriptor

Test/ID Jumpers

Refer to Figure 4-8. Digits C and D require unique jumpers to identify the ID number (0-255) of this particular DLP within the system.

A				B				C				D			
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
0	0	0	0	0	1	0	1	*	*	*	*	*	*	*	*

FIXED BYTE

FIELD INSTALLED JUMPERS
* JUMPERS

Figure 4-8. Card Punch ID Word

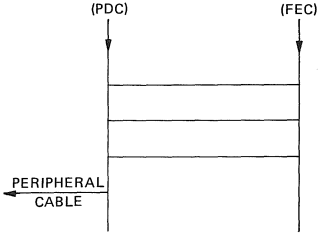
The Test/ID jumpers are located on the Peripheral Dependent Card. Table 4-7 shows the jumper positions.

Table 4-7. Card Punch DLP Test/ID Jumpers

Digit Bit	Bit Value	Jumper	
		From	To
C8	128	VC 43	VB 43
C4	64	VC 44	VB 44
C2	32	VC 45	VB 45
C1	16	VC 46	VB 46
D8	8	VC 47	VB 47
D4	4	VC 48	VB 48
D2	2	VC 49	VB 49
D1	1	VC 50	VB 50

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-9.



PDC = PERIPHERAL DEPENDENT CARD
FEC = FRONT END CARD

Figure 4-9. Card Punch DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-10.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND
-12V	103	-11.5V TO -12.5V
+12V	003	11.5V TO 12.5V

Figure 4-10. Card Punch DLP Voltage Test Points

UNIVERSAL CONSOLE DLP

Table 4-8. Universal Console DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
READ	50	*	8	0	U	0	A,B,C Addresses
Buffer		*	8	F	U	0	Required
Extended Status	50	*	8	D	U	0	A & B Addr. Rqd.
WRITE	52	*	4	0	U	0	A,B,C Addresses
Buffer	52	*	4	F	U	0	Required
Restore (Disk)	54	*	2	9	U	0	
TEST UNIT	54	*	2	0	U	0	
TEST							
Wait For Ready		*	2	1	U	0	
Wait For Not Ready		*	2	2	U	0	
Wait For Transmit		*	2	5	U	0	
Discontinue		*	2	A	U	0	
ID		*	2	C	U	0	
Conditional Cancel	**	*	2	8	U	0	

NOTES
** is an IOT OP Code

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

  8-bit = Inhibit data transfer to memory

  4-bit = Reserved

  2-bit = ASCII translation

  1-bit = Reserved

	A	B	C	D
8	1 NOT READY	5 SECTOR SEEK ERROR	9 INCORRECT STATE WORD (NOTE 2)	13 BUFFER PARITY ERROR
4	2 UNABLE TO INITIATE (UI) (NOTE 1)	6 TRACK SEEK ERROR	10 DIS-CONTINUED/ CANCELLED	14 ADDRESS CRC ERROR
2	3 VERTICAL PARITY ERROR	7 UNABLE TO COMPLETE	11 TIMEOUT	15 DATA CRC ERROR
1	4 LONG-ITUDINAL PARITY ERROR	8 INVALID I/O DESCRIPTOR ID	12 UNIT NOT PRESENT	16 EXCEPTION

NOTES

1. The UC-DLP was unable to initiate the operation and at least one other exception condition is set.

2. UC-DLP is unable to accept the I/O Descriptor because the permitted number of outstanding Descriptors has been exceeded or an invalid Cancel/Discontinue operation was received.

Figure 4-11. ICMD Word 1 Result Descriptor

	A	B	C	D
8	1 FIRST ACTION	5 END OF TRACK ZERO	9 ZERO	13 ZERO
4	2 WRITE FAULT	6 END OF DISKETTE	10 ZERO	14 ZERO
2	3 DELETED FAULTY SECTOR	7 ERROR ADDRESS AVAILABLE	11 ZERO	15 ZERO
1	4 RETRY SUCCESSFUL	8 ZERO	12 ZERO	16 ZERO

Figure 4-12. ICMD Word 2 Result Descriptor

Table 4-9. DLP-Test Bus I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Read – Data	50	*	8	0	U	0	A,B,C Addresses Required
Write – Command	52	*	4	0	U	0	
TEST UNIT	54	*	2	0	U	0	
TEST ID	54	*	C	0	U	0	

NOTES

U designates Unit Number. (U=1 for DLP-TEST Bus).

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

	A	B	C	D
8	1 ZERO	5 ZERO	9 INCORRECT STATE	13 ZERO
4	2 UNABLE TO INITIATE	6 ZERO	10 ZERO	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 UNABLE TO COMPLETE	11 ZERO	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID I/O DESCRIPTOR	12 UNIT NOT PRESENT	16 EXCEPTION

Figure 4-13. DLP-Test Bus (TEST UNIT Operation) Word 1 Result Descriptor

	A	B	C	D
8	1 ZERO	5 ZERO	9 INCORRECT STATE	13 ZERO
4	2 UNABLE TO INITIATE	6 ZERO	10 ZERO	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 UNABLE TO COMPLETE	11 ZERO	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID I/O DESCRIPTOR	12 ZERO	16 ZERO

Figure 4-14. DLP-Test Bus (Write Operation) Word 1 Result Descriptor

	A	B	C	D
8	1 ZERO	5 COMMAND INDEX	9 ZERO	13 FRAMING ERROR
4	2 ZERO	6 COMMAND INDEX	10 DATA LOST	14 NAK BACK FROM MAINTENANCE CARD
2	3 ZERO	7 COMMAND INDEX	11 MAINTENANCE TIMEOUT	15 TRANSMISSION ERROR
1	4 COMMAND INDEX	8 COMMAND INDEX	12 READ AND COMPARE ERROR	16 ZERO

Figure 4-15. DLP-TEST Bus (Write Operation) Word 2 Result Descriptor

	A	B	C	D
8	1 ZERO	5 EARLY TERMINATION	9 INCORRECT STATE	13 ZERO
4	2 UNABLE TO INITIATE	6 ZERO	10 ZERO	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 UNABLE TO COMPLETE	11 ZERO	15 ZERO
1	4 LONGITUDINAL PARITY ERROR (MLI)	8 INVALID I/O DESCRIPTOR	12 ZERO	16 EXCEPTION

NOTE
The Result Descriptor for DLP-Test Bus, Read Operation, Word 2 is all ZEROs.

Figure 4-16. DLP-Test Bus (Read Operation) Word 1 Result Descriptor

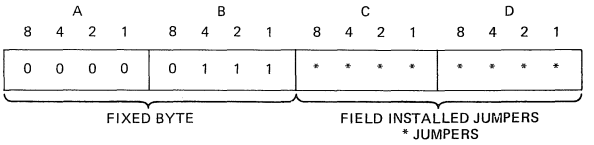


Figure 4-17. Test/ID Result Word

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-17. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

The Test/ID jumpers are located on the Peripheral Dependent Card. Table 4-10 shows the jumper positions.

Table 4-10. Universal Console Test/ID Jumpers

Digit Bit	Bit Value	Jumper	
		From	To
C8	128	VA 66	K1E1
C4	64	VA 67	VB 51
C2	32	VA 68	VB 53
C1	16	VA 69	VE 52
D8	8	VA 70	VC 55
D4	4	VA 71	VE 58
D2	2	VA 72	VF 53
D1	1	VA 73	VF 54

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-18.

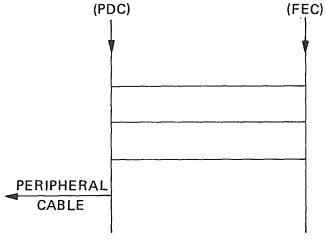


Figure 4-18. Universal Console DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-19.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND

Figure 4-19. Universal Console Voltage Test Points

5N DISK FILE DLP

Table 4-11. 5N Disk File DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
READ	50						A, B, and C
Data/Unconditional		*	8	0	U	0	Addresses
Transfer/Unconditional		*	8	2	U	0	Required
Maint/Unconditional		*	8	8	U	0	
Maint/Transfer/Uncond		*	8	A	U	0	
Extended Status	50	*	8	D	U	0	A & B Addr. Req.
Unit Status	50	*	8	E	U	0	A, B, and C
Buffer		*	8	F	U	0	Addresses Required
WRITE	52						
Data/Unconditional		*	4	0	U	0	
Maint/Unconditional		*	4	8	0	0	
Buffer		*	4	F	U	0	
TEST	44	*	2	0	U	0	A, B, and C
Wait Available	44	*	2	1	U	0	Addresses
Wait Not Available	44	*	2	2	U	0	Required for
Verify	54	*	2	4	U	0	Verify
Conditional Cancel	44	*	2	8	U	0	
ID	44	*	2	C	0	0	

NOTES

U designates Unit Number (0-F HEX)

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

  8-bit = Inhibit data transfer to memory

  4-bit = Reserved

  2-bit = ASCII translation

  1-bit = Reserved

	A	B	C	D
8	1 DEVICE NOT READY	5 EARLY TERM-INATION	9 INCORRECT STATE	13 BUFFER PARITY ERROR
4	2 DESCRIPTOR ERROR	6 READ EXTENDED STATUS	10 DIS-CONTINUED/ CANCELLED	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 ZERO	11 DATA ERROR	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID IN-FORMATION	12 ZERO	16 EXCEPTION

Figure 4-20. 5N DF DLP Word 1 Result Descriptor

	A	B	C	D
8	1 DISK STORAGE UNIT NOT READY	5 DATA TRANS-MISSION ERROR	9 RSB PARITY ERROR (NOTE 2)	13 ZERO
4	2 WRITE LOCKOUT (NOTE1)	6 WARNING	10 EXCHANGE ERROR (NOTE 2)	14 DATA CORRECTED (FROM ECC)
2	3 ADDRESS ERROR	7 ZERO	11 TIMEOUT (NOTE 2)	15 DATA READ ERROR (FROM ECC)
1	4 COMMAND TRANS-MISSION ERROR (NOTE 2)	8 ZERO	12 ZERO	16 EXTRA DISK REV-OLUTION

Figure 4-21. 5N DF DLP Word 2 Result Descriptor

NOTE

1. Indicates that the Write Lockout switch corresponding to the addressed disk is in the ON position.

2. If this bit is set in conjunction with the Exchange Error bit, the DLP has detected an error from the exchange.

Figure 4-21. 5N DF DLP Word 2 Result Descriptor (Cont)

	A	B	C	D
8	1 DEVICE NOT READY	5 ZERO	9 ZERO	13 ZERO
4	2 DESCRIPTOR ERROR	6 ZERO	10 ZERO	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 ZERO	11 ZERO	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID IN-FORMATION	12 ZERO	16 EXCEPTION

Figure 4-22. 5N DF DLP Test/ID Word 1 Result Descriptor

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-23. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

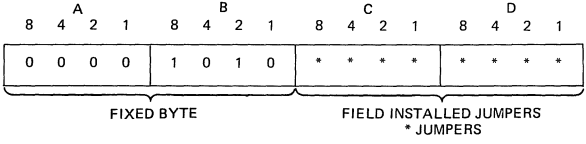


Figure 4-23. 5N DF DLP Test/ID Word 2 Result Descriptor

The Test/ID jumpers are located on the Peripheral Dependent Card, (CD, 5N/1). Table 4-12 shows the jumper positions.

Table 4-12. Test/ID Jumpers

Digit Bit	Bit Value	Jumper	
		From	To
C8	128	VB 53	VC 53
C4	64	VB 54	VC 54
C2	32	VB 55	VC 55
C1	16	VB 56	VC 56
D8	8	VB 57	VC 57
D4	4	VB 58	VC 58
D2	2	VB 59	VC 59
D1	1	VB 60	VC 60

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-24.

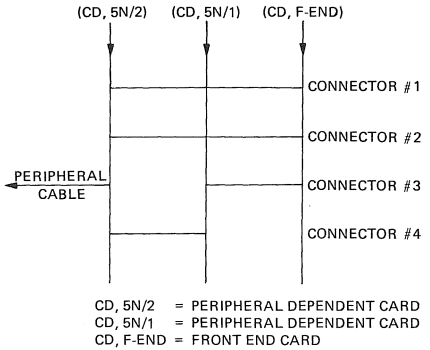


Figure 4-24. 5N DF DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-25.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND

Figure 4-25. 5N Disk File DLP Voltage Test Points

HOST TRANSFER DLP

Table 4-13. Host Transfer DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
ECHO	48	*	8	0	0	0	A and B
READ	50						A, B, and C
Data		*	8	0	0	U	Addresses
Data (No Timeout)		*	8	8	0	U	
WRITE	52						
Data		*	4	0	0	U	
Host Load			4	0	0	U	
Test	54	*	2	0	0	U	

NOTES

U designates HEX Unit Number.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

Table 4-14. Host Transfer DLP B x384/B x385 Command Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
READ	50						A, B, and C
(EPC Enable)		*		0	0	U	Addresses
(EPC Disable)		*		0	1	U	Required
Unit ID		*		0	2	U	
Memory		*		0	4	U	
Absolute		*		0	8	U	
Subsystem Poll		*		0	E	U	
WRITE	52	*		0	0	U	
Initialize		*		0	8	U	
Initialize (Data Field Only)		*		0	A	U	
TEST	54	*		0	0	U	
Power Down Unit		*		0	2	U	
Contr. Lock Enable		*		0	5	U	

Table 4-14. Host Transfer DLP B x384/B x385 Command Descriptors (Cont)

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Contr. Lock Disable		*		0	4	U	
Verify (EPC Only)		*		0	6	U	
Verify (Data and EPC)		*		0	7	U	
Relocate		*		0	E	U	

NOTES

U is Unit Designate (0 thru 15)

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory

4-bit = Reserved

2-bit = ASCII translation

1-bit = Reserved

Table 4-15. Host Transfer DLP B 9387 Command Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
READ	50						A, B, and C
(EPC Enable)		*		0	0	U	Addresses
(EPC Disable)		*		0	1	U	Required
Unit ID		*		0	2	U	
Memory		*		0	4	U	
Absolute		*		0	8	U	
Subsystem Poll		*		0	E	0	
WRITE	52	*		0	0	U	
Initialize		*		0	8	U	
Initialize Data Field Only		*		0	A	U	
TEST	54	*		0	0	U	
Power Unit Down		*		0	2	U	
Power Unit UP		*		0	3	U	
Contr. Lock Enable		*		0	5	U	
Contr. Lock Disable		*		0	4	U	
Verify (EPC Only)		*		0	6	U	
Verify (Data and EPC)		*		0	7	U	
Take Unit out of Maintenance Mode		*		0	8	U	
Place Unit into Maintenance Mode		*		0	9	U	
Relocate		*		0	E	U	

NOTES

U is Unit Designate (0 thru 15)

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

Table 4-16. Host Transfer DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
ECHO	48	*	1	0	0	0	
READ		*	8	0	D	D	A, B, and C
S Memory	50	*	8	1	0	0	Addresses
No Timeout		*	8	8	D	D	Required
Interrogate Line Status		*	8	2	0	0	
WRITE		*	4	0	F	F	
Host Load	52	*	4	8	0	0	
Last S Load		*	4	0	0	0	
S Memory Table		*	4	0	F	F	
TEST		*	2	0	0	0	
ID	54	*	2	C	0	0	
Conditional Cancel	**	*	2	8	0	0	

NOTES

** is an IOT OP Code.

F designates Function Number.

D designates Deallocate Pointer.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

	A	B	C	D
8	1 NOT READY	5 TIMEOUT	9 RECOVERY	13 LOCKED
4	2 DESCRIPTOR ERROR	6 HD-DDP VERTICAL PARITY ERROR (NOTE 1)	10 WRITE LOCKOUT	14 CANCELLED
2	3 VERTICAL PARITY ERROR (MLI)	7 HD-DDP LONG-ITUDINAL PARITY ERROR (NOTE 2)	11 SEEK INITIATE	15 RESERVED (ZERO)
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 DRIVE BUSY	12 RESERVED (ZERO)	16 ROM COMMAND ERROR

NOTES

1. Vertical Parity Error detected by the DLP on a word transfer from the DPDC.

2. Longitudinal Parity Error detected by the DLP on a data block or a Result Descriptor from the DPDC.

Figure 4-26. HT DLP/DPDC Word 1 Result Descriptor (Bx9384/Bx385/ Bx387)

Table 4-17. Host Transfer DLP Word 2 and Word 3 R/D (DPDC)

	Result Descriptor Bits							
	Word 2				Word 3			
	A	B	C	D	A	B	C	D
Description	8421	8421	8421	8421	8421	8421	8421	8421
Data Error	1111	0000	Sector Address					
Seek Error	1111	0001	Sector Address					
Address Error	1111	0010	Sector Address					
Write Retry	1111	0011	Sector Address					
Read Retry	1111	0100	Sector Address					
Data Error Correction	1111	1000	Sector Address					
Command Error (B)	1111	1111	Sector Address					
Command Error (A)	0000	R/D Tag			Command			
Disk Error	0001	R/D Tag			Disk DDP Status			
Disk Error	0100	R/D Tag			Disk Unit Status			
Disk Error	0101	R/D Tag			Drive Msg. Word 1			
Host Error	0110	R/D Tag			Host DDP Status			
Processor Error	1001	R/D Tag			Memory Status			
Controller Failure	1010	R/D Tag			Not Used			
Exchange Path	1011	R/D Tag			As Defined Below			
Exchange 0 Failed					0000			
Exchange 1 Failed					0001			

NOTES

1-Word Result Descriptors are not lockable.

3-Word Result Descriptors are lockable.

Consult individual controlware listing for more detail of the R/D tag supplied by the DPDC.

	A	B	C	D
8	1 CON-TROLLER NOT READY	5 TIMEOUT	9 SUCCESSFUL RECOVERY	13 CON-TROLLER LOCKED
4	2 DESCRIPTOR ERROR	6 HD-DDP VERTICAL PARITY (NOTE 1)	10 WRITE LOCKOUT	14 CANCELLED
2	3 VERTICAL PARITY ERROR (MLI)	7 HD-DDP LONG-ITUDINAL PARITY ERROR (NOTE 2)	11 SEEK INITIATE	15 RESERVED (ZERO)
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 DRIVE BUSY	12 RESERVED (ZERO)	16 ROM COMMAND ERROR

NOTES

1. Vertical Parity Error detected by the DLP on a word transfer from the DPDC.

2. Longitudinal Parity Error detected by the DLP on a data block or a Result Descriptor from the DPDC.

Figure 4-27. HT DLP/DPDC Word 1 Result Descriptor

Table 4-18. Host Transfer DLP/DCP R/D

R/D	Description
0 0 0 0	OP Complete
8 0 0 0	DDP Not Ready
4 0 0 0	Descriptor Error
2 0 0 0	System Vert. Parity Err.
1 0 0 0	Sys. Long. Parity Error
0 0 1 0	Invalid Message Pointer
0 0 2 0	Invalid Stat. (PSN) No.
0 0 3 0	Invalid Function
0 0 4 0	Suspended Host Output
0 0 5 0	Cancel Complete
0 0 6 0	Cancel Invalid
0 0 7 0	DCP Short Block
0 0 0 2	S Parity Error

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Table 4-19. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

Table 4-19. Host Transfer DLP Test/ID Word 2 Result Descriptor

Digit	A	B	C	D
Bit	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1
Unit ID Word	0 0 0 0	1 0 0 0	* * * *	* * * *
Unit ID Word-DCP	0 0 0 0	1 1 1 1	* * * *	* * * *
Unit ID Word-DPDC	0 0 0 1	0 0 1 0	* * * *	* * * *

* The standard configuration ID that is field strappable.

The Test/ID jumpers are located on the Peripheral Dependent Card. Table 4-20 shows the jumper positions.

Table 4-20. Host Transfer DLP Test/ID Jumpers

Digit Bit	Bit Value	Jumper From	Jumper To
C8	128	VA 37	VB 37
C4	64	VA 36	VB 36
C2	32	VA 35	VB 35
C1	16	VA 34	VB 34
D8	8	VA 33	VB 33
D4	4	VA 32	VB 32
D2	2	VA 31	VB 31
D1	1	VA 30	VB 30

Data Communication Processor (DCP) Jumper

The HT-DLP operates both the DCP and the Disk Pack Drive Controller (DPDC).

If the DLP is connected to a DCP, install the jumper. If the DLP is connected to a DPDC, remove the jumper.

Jumper location is VA 50 to VB 50.

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-28.

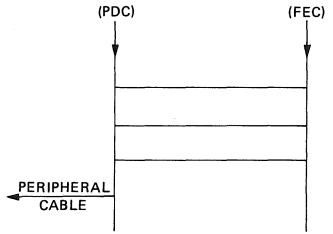


Figure 4-28. Host Transfer DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-29.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND

Figure 4-29. Host Transfer DLP Voltage Test Points

PE MAGNETIC TAPE DLP

Table 4-21. PE Magnetic Tape DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
READ Forward Backward	 40 41	 * *	 8 8	 2 3	 U U	 M M	A and B Addresses Required
WRITE Tape Marks ERASE	42	* * *	4 4 4	6 C 4	U U U	M 0 M	
REWIND REWIND UNLOAD	44	* *	2 2	1 B	U U	0 0	
SPACE FORWARD SPACE BACKWARD	54	* *	2 2	8 9	U U	M M	**C Address Required
TEST Wait Ready Wait Not Ready Cancel ID	44	* * * * *	2 2 2 2 2	F 3 2 4 C	U U U U 0	0 0 0 0 0	
ECHO	48	*	1	0	0	0	A & B Adr Reqd

NOTES

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory

4-bit = Reserved

2-bit = ASCII translation

1-bit = Reserved

U designates Unit Number (U = 0 designates Tape Unit 16).

** C address designates Space Count. Two MSD indicate 1 to 100 records, if space count is decimal. Two MSD indicates 1 to 256 records, if space count is binary.

M designates Mask Bits.

Mask Bits Descriptions: SKLN

8-bit Used by Space Forward/Backward to indicate that the space count Variant Digits contain a decimal (8-bit = TRUE) or a binary (8-bit = FALSE) value.

Table 4-21. PE Magnetic Tape DLP I/O Descriptors (Cont)

4-bit Used by Read Forward/Backward and ECHO to inhibit the reporting of the Record Longer Than Memory Buffer Exception condition.

2-bit Used by Read Forward/Backward, WRITE, ERASE, and ECHO to inhibit the reporting of the Record Shorter Than Memory Buffer Exception condition.

1-bit Used by Space Forward/Backward to modify the space operation. The MEC is requested to space one record at a time (1-bit = FALSE) or to space continuously (1-bit = TRUE).

	A	B	C	D
8	1 MEC NOT READY	5 VERTICAL PARITY ERROR (NOTE 2)	9 BOT	13 RECORD LONGER THAN HOST BUFFER
4	2 DESCRIPTOR ERROR	6 VERTICAL PARITY ERROR (NOTE 3)	10 EOT	14 RECORD SHORTER THAN HOST BUFFER
2	3 VERTICAL PARITY ERROR (NOTE 1)	7 VERTICAL PARITY ERROR (NOTE 4)	11 TAPE MARK	15 TAPE UNIT BUSY
1	4 LONG-ITUDINAL PARITY ERROR	8 TAPE UNIT NOT READY	12 WRITE LOCKOUT	16 REWINDING

NOTES

1. The Vertical Parity detected on the Host System interface was EVEN.

2. The MEC detected incorrect Vertical Parity from the DLP on the DLP Interface to MEC.

Figure 4-30. PEMT DLP Word 1 Result Descriptor

3. The DLP detected incorrect Vertical Parity from the MEC on the MEC Interface to DLP.

4. Incorrect Vertical Parity was detected from the tape media during a Read or Space operation.

Figure 4-30. PEMT DLP Word 1 Result Descriptor (Cont)

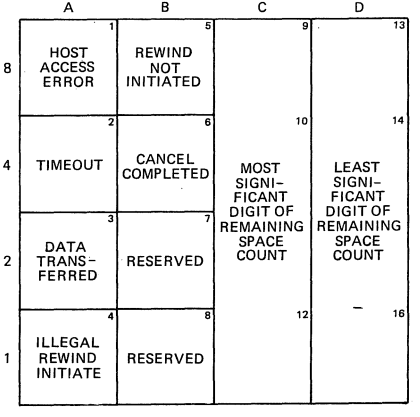


Figure 4-31. PEMT DLP Word 2 Result Descriptor

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-32 Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

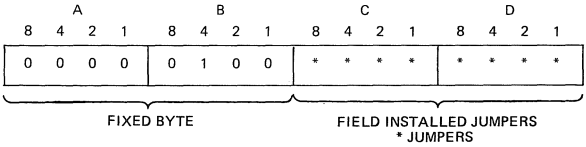


Figure 4-32. Magnetic Tape ID Word

The Test/ID jumpers are located on the Peripheral Dependent Card. Table 4-22 shows the jumper positions.

Table 4-22. PE Magnetic Tape Test/ID Jumpers

Digit Bit	Bit Value	Jumper	
		From	To
C8	128	VA 37	VB 37
C4	64	VA 36	VB 36
C2	32	VA 35	VB 35
C1	16	VA 34	VB 34
D8	8	VA 33	VB 33
D4	4	VA 32	VB 32
D2	2	VA 31	VB 31
D1	1	VA 30	VB 30

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-33.

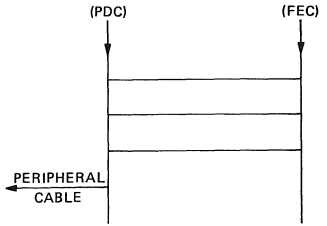


Figure 4-33. PE Magnetic Tape DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-34.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND
-12V	103	-11.5V TO -12.5V
+12V	003	11.5V TO 12.5V

Figure 4-34. PE Magnetic Tape DLP Voltage Test Points

READER SORTER DLP

Table 4-23. Reader Sorter DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Read Buffer	40	*	8	F	0	0	A & B Addresses Required
Pocket Select-Read	42	*	4	0	0	0	
Start Flow-Read		*	4	2	0	V	
Demand Feed-Read		*	4	1	0	V	
Write Buffer		*	4	F	0	0	
Test Status	44	*	2	0	X	X	
Test/Wait Available		*	2	1	X	X	
Test/Wait Not Available		*	2	2	X	X	
Test/ID		*	2	C	X	X	
Image Count Mark		*	2	3	X	N	
Pocket Light		*	2	4	N	N	
Slew Microfilm		*	2	5	X	X	
Cancel		*	2	8	X	X	

Table 4-23. Reader Sorter DLP I/O Descriptors (Cont)

NOTES

N designates Decimal digit.

V designates Variant.

8-bit = The last 9 bits of band 1 is header information.

4-bit = Not used.

2-bit = Set concurrently with 1-bit; read from stations A and B.

1-bit = Read from station A.

X specifies that either a 1 or 0 will be accepted.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory

4-bit = Reserved

2-bit = ASCII translation

1-bit = Reserved

	A	B	C	D
8	1 DEVICE NOT READY	5 FILM OPER. NOT COMPL. PROP. OR PKT. LIGHT NOT COMPL.	9 BAD INTERFACE INFOR-MATION	13 READER SORTER POWER FAILURE
4	2 DESCRIPTOR ERROR	6 NON-PRESENT OPTION OR CAMERA NOT PRESENT	10 TIMEOUT (500 MS)	14 MEM. OVRFL. OR COND. CANCEL OR INT. PAR. ERROR
2	3 VERTICAL PARITY ERROR (MLI)	7 RESERVED	11 CAMERA NOT READY	15 DLP ERROR
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 POCKET-SELECT ERROR	12 PARITY ERROR (SORTER TO DLP)	16 RESULT STATUS ATTENTION

Figure 4-35. Reader Sorter DLP Result Descriptor

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card (C,F-END). It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-36. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

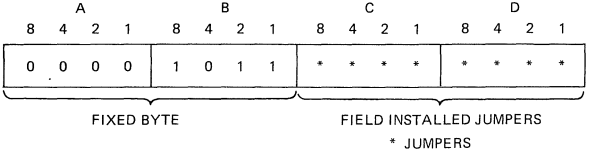


Figure 4-36. Reader Sorter DLP Test/ID Word 2 Result Descriptor

	A	B	C	D
8	1 DEVICE NOT READY	5 ENDORSER BAND 1 PRESENT	9 ENDORSER NOT READY	13 READER SORTER POWER FAILURE
4	2 DESCRIPTOR ERROR	6 ENDORSER BAND 2 PRESENT	10 CAMERA PRESENT	14 RESERVED
2	3 VERTICAL PARITY ERROR (MLI)	7 ENDORSER BAND 3 PRESENT	11 CAMERA NOT READY	15 DLP STRAPPED FOR 9138 READER SORTER
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 ENSORSER BAND 4 PRESENT	12 FILM SLEWING	16 DLP STRAPPED NON-STANDARD (MEDIUM SYSTEMS)

Figure 4-37. Reader Sorter DLP Test/Status Result Descriptor

The Test/ID jumpers are located on the RS1 Peripheral Dependent Card, (CD, RS1). Table 4-24 shows the jumper positions.

Table 4-24. Reader Sorter DLP Test/ID Jumpers

Digit Bit	Bit Value	Jumper	
		From	To
C8	128	VA 73	VB 73
C4	64	VA 74	VB 74
C2	32	VA 75	VB 75
C1	16	VA 76	VB 76
D8	8	VA 77	VB 77
D4	4	VA 78	VB 78
D2	2	VA 79	VB 79
D1	1	VA 80	VB 80

Standard/Non-Standard Jumper

The DLP must be configured to indicate whether Standard or Non-Standard I/O Descriptors are accepted.

To accept Non-Standard (Medium System) I/O Descriptors, the Jumper VB 35 to VB 37 is needed, otherwise no jumper is needed.

B 9137/B 9138 Jumper

The DLP must be configured to indicate whether it communicates with a B 9137 or a B 9138 Reader Sorter.

To communicate with the B 9137 Reader Sorter, the Jumper VA 53 to VB 53 is needed. No Jumper is needed for the DLP to communicate with the B 9138 Reader Sorter.

The B 9137/B 9138 Jumper is located on the RS2 Peripheral Dependent Card.

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-38.

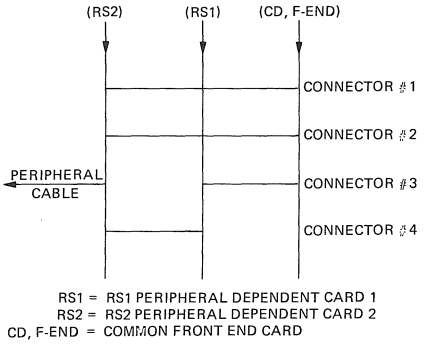


Figure 4-38. R-S DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-39.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND

Figure 4-39. Reader Sorter DLP Voltage Test Points

Buffer Memory

Buffer Memory is contained on the CFE card, but the associated address and data registers are on the PDB. Figure 4-40 is a memory map for the Buffer Memory.

NOTES

Band Load Buffer – receives the band load data from the system.

Send Buffer – receives the Read Data from the Tank Read Stations through the translator.

Tank Read Station – two Tanks for each Read Station.

ENDAD – contains the end address of the Send Buffer.

RDAD – holds the R/D generated by the firmware for ICM, Slew and Pocket Light descriptors.

MSGDAAD – contains the address of the next word of band load data when transferring the data to the sorter.

PRONTO – used to flag the System of an error condition from the sorter that requires immediate attention.

ST1INFO – stores Strobe 1 data when OPTI is set.

ST2INFO – stores Strobe 2 data when OPTI was set during the previous Strobe 1 time.

MSGTO – contains data that the BEM firmware is sending to the FEM firmware.

MSGFR – contains data that the FEM firmware is sending to the BEM firmware.

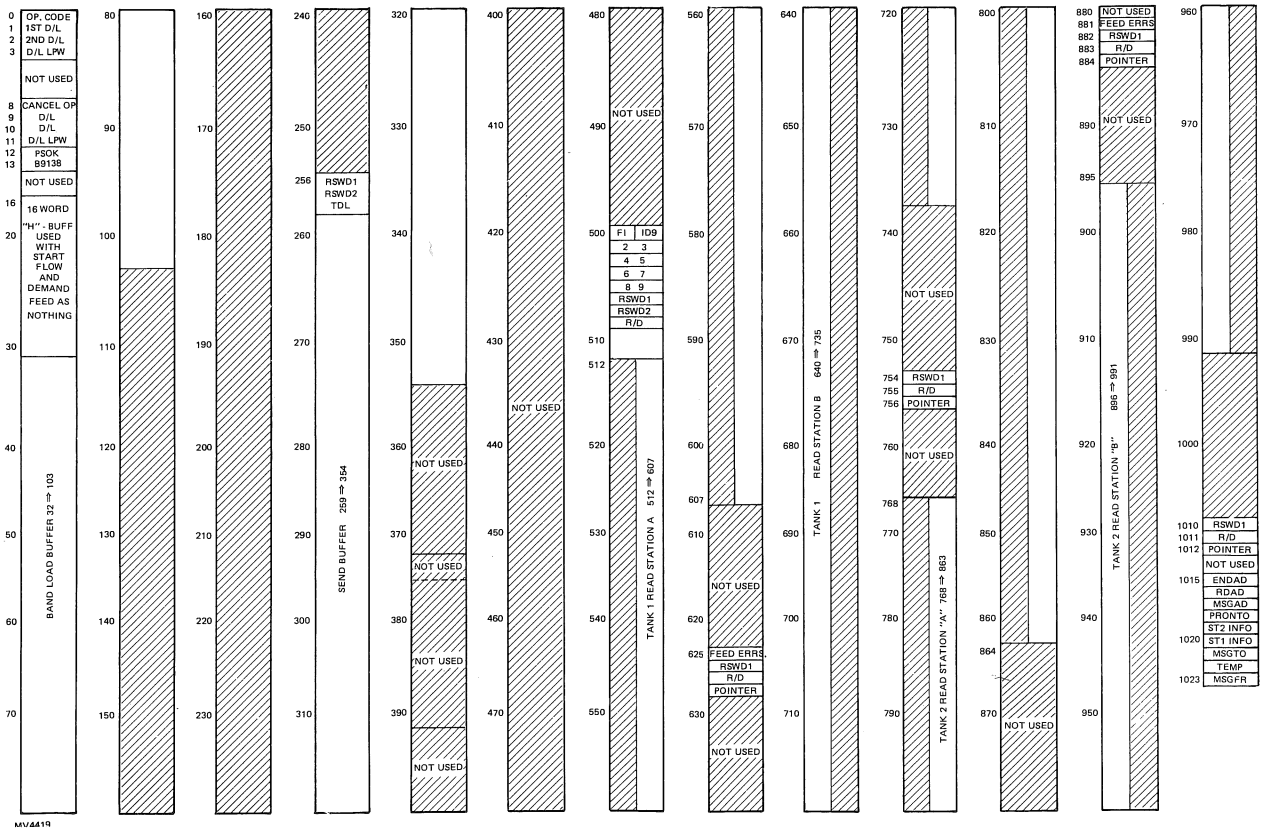


Figure 4-40. RS-DLP Buffer Memory Map

TRAIN PRINTER DLP

Table 4-25. Train Printer DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Write	42	*	4	0	0	0	A & B Addresses
Write/Halt		*	4	1	0	0	Required
Write Line		*	4	4	0	N	
Write Line/Halt		*	4	5	0	N	
Write/Load TIB		*	4	8	F	F	
Move/Write Line		*	4	6	0	N	
Move/Write Line/Halt		*	4	7	0	N	
TEST	44	*	2	0	0	0	
Wait Ready		*	2	1	0	0	
Wait Not Ready		*	2	2	0	0	
Skip		*	2	4	0	N	
Conditional Cancel		*	2	8	0	0	
ID		*	2	C	0	0	
ECHO	48	*	1	0	0	0	

NOTES

Variant L2 uses Bits 8&4 for J, and Bits 2&1 for F.

N designates the bits used to encode the paper motion format. See Table 4-26.

F designates the printer speed and train identifier as follows:

The 8-bit and 4-bit of L2 indicate printer speed.

00 is 750 LPM
01 is 1100 LPM
10 is 1500 LPM
11 is Reserved

The 2-bit and 1-bit of L2 and all bits of L3 are combined to encode the train identifier.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

Table 4-26. Train Printer DLP Paper Motion Format Codes

Code	Description	Print Format Tape
0 0 0 0	No paper Motion	2 and 12 Channel Tape
0 0 0 1	Advance to Heading	2 and 12 Channel Tape
0 0 1 0	Advance to Channel 2	2 and 12 Channel Tape
0 0 1 1	Advance to Channel 3	2 and 12 Channel Tape
0 1 0 0	Advance to Channel 4	2 and 12 Channel Tape
0 1 0 1	Advance to Channel 5	2 and 12 Channel Tape
0 1 1 0	Advance to Channel 6	2 and 12 Channel Tape
0 1 1 1	Advance to Channel 7	2 and 12 Channel Tape
1 0 0 0	Advance to Channel 8	2 and 12 Channel Tape
1 0 0 1	Advance to Channel 9	2 and 12 Channel Tape
1 0 1 0	Advance to Channel 10	2 and 12 Channel Tape
1 0 1 1	Advance to Channel 11	2 and 12 channel Tape
1 1 0 0	Advance to End Of Page	2 and 12 Channel Tape
1 1 0 1	Advance to EOP on Next Channel	2 Channel Tape Only
1 1 1 0	Single Space	2 and 12 Channel Tape
1 1 1 1	Double Space	2 and 12 Channel Tape

	A	B	C	D
8	1 NOT READY	5 TRAIN IMAGE BUFFER NOT LOADED	9 RESERVED	13 PRINT CHECK/ ERROR LINE PRINTED
4	2 DESCRIPTOR ERROR	6 INCORRECT TRAIN	10 RESERVED	14 PRINT CHECK/RAM PARITY
2	3 VERTICAL PARITY ERROR (MLI)	7 END OF PAGE	11 PRINT CHECK/SYNC ERROR	15 PRINT CHECK/ HUNG
1	4 LONG- ITUDINAL PARITY ERROR (MLI)	8 RESERVED	12 PRINT CHECK/ INVALID	16 PRINT CHECK/ FORMAT

Figure 4-41. Train Printer DLP Write Result Descriptor

	A	B	C	D
8	1 NOT READY	5 TIB NOT LOADED	9 RESERVED	13 TRAIN ID
4	2 DESCRIPTOR ERROR (NOTE 1)	6 INCORRECT TRAIN	10 RESERVED	14 TRAIN ID
2	3 VERTICAL PARITY ERROR (MLI)	7 RESERVED	11 TRAIN ID	15 TRAIN ID
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 CANCEL COMPLETE	12 TRAIN ID	16 TRAIN ID

Figure 4-42. Test/Waits + Test/Cond. Cancel Result Descriptor

	A	B	C	D
8	1 NOT READY	5 TIB NOT LOADED	9 RESERVED	13 RESERVED
4	2 DESCRIPTOR ERROR	6 INCORRECT TRAIN	10 RESERVED	14 RESERVED
2	3 VERTICAL PARITY ERROR (MLI)	7 END OF PAGE	11 RESERVED	15 RESERVED
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 RESERVED	12 RESERVED	16 RESERVED

Figure 4-43. Test/Skip Result Descriptor

	A	B	C	D
8	1 NOT READY	5 TIB NOT LOADED	9 LT B2 (NOTE 2)	13 TRAIN ID
4	2 DESCRIPTOR ERROR (NOTE 1)	6 INCORRECT TRAIN	10 LT B1 (NOTE 2)	14 TRAIN ID
2	3 VERTICAL PARITY ERROR (MLI)	7 CW B2 (NOTE 1)	11 TRAIN ID	15 TRAIN ID
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 CW B1 (NOTE 1)	12 TRAIN ID	16 TRAIN ID

NOTES

1. Train Printer Column Width.

   CW B2/ * CW B1/ = 132
   CW B2/ * CW B1 = 120
   CW B2 * CW B1/ = 80

2. Train Printer Speed.

   LT B2/ * LT B1 = 1100 LPM
   LT B2 * LT B1/ = 1500 LPM

Figure 4-44. Test/ID, Test+Echo Result Descriptor

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-45. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

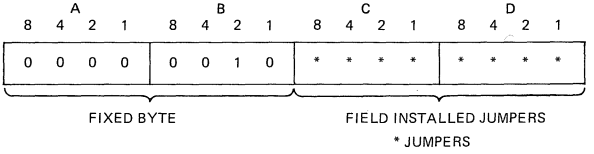


Figure 4-45. Train Printer DLP Test/ID Word 2 Result Descriptor

The Test/ID jumpers are located on the Peripheral Dependent Card. Table 4-27 shows the jumper positions.

Table 4-27. Train Printer Test/ID Jumpers

Digit Bit	Bit Value	Jumper	
		From	To
C8	128	VB 53	VC 53
C4	64	VB 54	VC 54
C2	32	VB 55	VC 55
C1	16	VB 56	VC 56
D8	8	VB 57	VC 57
D4	4	VB 58	VC 58
D2	2	VB 59	VC 59
D1	1	VB 60	VC 60

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-46.

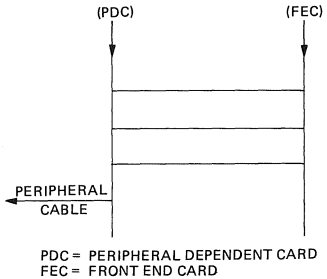


Figure 4-46. Train Printer DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-47.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND
-12V	103	-11.5V TO -12.5V
+12V	003	11.5V TO 12.5V

Figure 4-47. Train Printer DLP Voltage Test Points

UNILINE DLP

Table 4-28. Uniline DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Read To Control	40	*	8	0	0	0	A and B Addresses Required
Inhibit Timeout		*	8	1	0	0	
Write To Control	42	*	4	0	0	0	
WC/RC		*	4	8	0	0	
WC/RC Inhibit Timeout		*	4	9	0	0	
Firmware Load		*	4	2	0	0	
Test	44	*	2	0	0	0	
Test/ID		*	2	C	0	0	
Initiate MPU		*	2	1	0	0	
Enable		*	2	4	0	0	
Inhibit Timeout		*	2	5	0	0	
Echo	48	*	1	0	0	0	A and B Addresses Required
Dump RAM Memory		*	1	0	8	1	
Dump MLI Buffer		*	1	A	0	0	
Dump RAM Memory		*	1	B	0	0	
Load/Dump Firmware		*	1	C	0	0	

Table 4-28. Uniline DLP I/O Descriptors (Cont)

NOTES

WC/RC = Write to Control/Read to Control.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

	A	B	C	D
8	1 NOT READY	5 CANCEL COMPLETE	9 TERM-INATED WRITE	13 FIRMWARE NOT LOADED AND TIMEOUT VARIANT
4	2 DESCRIPTOR ERROR	6 TIMEOUT	10 TERMINAL VERTICAL PARITY (NOTE)	14 MPU MEMORY PARITY ERROR
2	3 VERTICAL PARITY ERROR	7 BCC ERROR	11 ACCESS ERROR	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR	8 ZERO	12 TERMINATE READ	16 ZERO

NOTE

Vertical Parity was received from the remote device.

Figure 4-48. Uniline DLP Result Descriptor

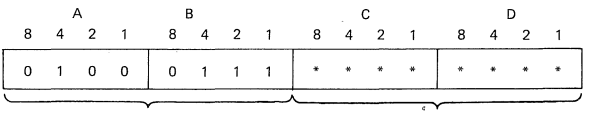


Figure 4-49. Uniline DLP Test/ID Result Descriptor

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on Card 1 which is the Message Level Controller Card. It is reserved for expansion to 16 DLP/Base.

Jumper V077 to V071

Test/ID Jumpers

A unique unit identifier (00 thru FF) to be returned as part of the R/D by a 'Test ID' descriptor may be installed.

The Test ID jumpers are located on Card 2, which is the Micro Processor Unit Card (MPU). Follow the instructions in Table 4-29 to properly install the Test ID Jumpers.

Table 4-29. Uniline DLP Test/ID Jumpers

Digit Bit	Jumper	
	From	To
C8/	I 109	I 108
C4/	I 110	I 107
C2/	I 111	I 106
C1/	I 112	I 105
D8/	I 113	I 104
D4/	I 114	I 103
D2/	I 115	I 102
D1/	I 116	I 101

Optional Jumpers

Depending upon use and configuration of the Uniline DLP the following jumpers are options.

Burroughs Direct Interface (BDI)

The following are located on card 3 which is the Data Transfer Adapter Card (DTM) and must installed as follows:

C5 E1 to B5 E5
C3 F1 to C3 H1
A5 E5 to B5 E1

Two-Wire Direct Interface (TDI)

Jumpers for this option are located on Card 3 (DTM). For the TDI Interface, the following jumpers must be removed or installed:

C5 E1 to B5 E5 (remove)
A5 E5 to B5 E1 (remove)
C3 H1 to C3 G1 (install)

Maintenance Card Interface

Jumpers for this option are located on Card 3 (DTM) and must be installed as follows:

A4 E5 to A4 F5
B4 E1 to B4 F1

Code Recognition

The UL-2 must be strapped to detect ASCII delimiter including EOT.

The ASCII delimiter and EOT options must be jumpered as follows on Card 1 (MLC).

B3 G1 to B3 H1
F3 G1 to F3 H1

If EBCDIC delimiter codes are used, install the following jumpers:

B3 F1 to B3 H1
F3 F1 to F3 H1

If ASCII-EOT/ delimiter codes are used, install the following jumpers:

B3 E1 to B3 H1
F3 E1 to F3 H1

Jumpers for the Baud Rate are located on Card 3 (DTM).

Baud Rate for the DTM will be 9600 if no jumpers are installed, otherwise the following options are available.

Table 4-30. Baud Rates

	Jumpers	
Baud Rate	**From**	**To**
4800	I1 I1	I1 I3
2400	VG 26	VG 28
1200	VG 26 and I1 I1	VG 28 and I1 I3

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-50.

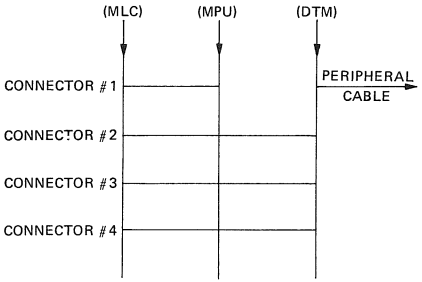


Figure 4-50. Uniline DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-51.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND
-12V	103	-11.5V TO -12.5V
+12V	003	11.5V TO 12.5V

Figure 4-51. Uniline DLP Voltage Test Points

GCR TAPE DLP

Table 4-31. GCR Tape DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
READ							A and B
Forward	40	*	8	2	U	0	Addresses
Backward	41	*	8	3	U	0	Required
Extended Status	40	*	8	D	U	0	
Unit Status	40	*	8	E	U	0	
Buffer	40	*	8	F	U	0	
WRITE	42						A and B
Data		*	4	0	U	d	Addresses
Buffer		*	4	F	U	0	Required
REWIND	44	*	2	4	U	0	
REWIND/UNLOAD	44	*	2	5	U	0	
SPACE FORWARD	54	*	2	6	U	x	**C Address
SPACE BACKWARD	54	*	2	7	U	x	Required
ERASE	54	*	2	E	U	x	
TEST	44						
Unit		*	2	0	U	0	
Wait Available		*	2	1	U	0	
Wait Not Available		*	2	2	U	0	
Cancel		*	2	8	U	0	
ID		*	2	C	0	0	
WRITE TAPE MARK	44	*	2	D	U	d	
DATA SECURITY ERASE	54	*	2	F	U	0	

NOTES

U designates Unit Number.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

d = Density bit. Bit 1 set to 1, operation is in GCR mode (6250 bpi). Bit 1 set to 0, operation is in PE mode (1600 bpi).

x defines density and space count. Bit 1 = density (1 = GCR, 0 = PE). Bit 2 = Space count (1 = decimal, 0 = binary).

Table 4-31. GCR Tape DLP I/O Descriptors (Cont)

NOTES

** The middle byte of the C Address contains space count for space and erase operations.

0 = Space to tape mark.

	A	B	C	D
8	1 DEVICE NOT READY	5 EARLY TERM-INATION	9 INCORRECT STATE	13 UNIT NOT READY
4	2 DESCRIPTOR ERROR	6 READ EXTENDED STATUS	10 DIS-CONTINUED/ CANCELLED	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 TCU TIMEOUT	11 DATA ERROR	15 TRANS-MISSION ERROR
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID INFOR-MATION	12 HOST ACCESS ERROR	16 EXCEPTION

Figure 4-52. GCR Tape DLP Word 1 Result Descriptor

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

	A	B	C	D
8	1 INVALID OP RECEIVED	5 WRITE LOCKOUT	9 BOT	13 UNIT CHECK
4	2 MTU NOT READY	6 TAPE MARK	10 EOT	14 DATA NOT TRANS-FERRED
2	3 TCU TRANS-MISSION ERROR	7 DATA SECURITY ERASE/ REWIND NOT INITIATED	11 REWINDING	15 ERROR CORRECTION APPLIED
1	4 DATA CHECK (PARITY ERROR)	8 DEVICE TIMEOUT	12 MTU BUSY	16 SHORT BLOCK

Figure 4-53. GCR Tape DLP Word 2 Result Descriptor

Test/ID Jumpers

Refer to Figure 4-54. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

A				B				C				D			
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
0	0	0	0	1	1	0	1	*	*	*	*	*	*	*	*

FIXED BYTE

FIELD INSTALLED JUMPERS
* JUMPERS

Figure 4-54. GCR Tape DLP Test/ID Word 2 Result Descriptor

The Test/ID jumpers are located on the Peripheral Dependent Card, (CD, GCR/1). Table 4-32 shows the jumper positions.

Table 4-32. GCR Tape DLP Test/ID Jumpers

Digit Bit	Bit Value	Jumper	
		From	To
C8	128	VB 53	VC 53
C4	64	VB 54	VC 54
C2	32	VB 55	VC 55
C1	16	VB 56	VC 56
D8	8	VB 57	VC 57
D4	4	VB 58	VC 58
D2	2	VB 59	VC 59
D1	1	VB 60	VC 60

TCU/Formatter Strap

The TCU/Formatter strap on the Common Peripheral Card is not used for Medium Systems. The strap (VA 34 to VB 34) is not to be installed.

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-55.

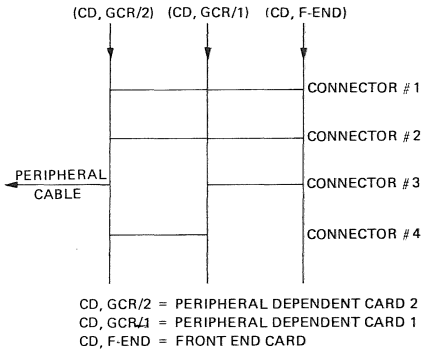


Figure 4-55. GCR Tape DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-56.

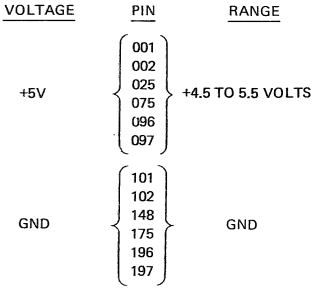


Figure 4-56. GCR Tape DLP Voltage Test Points

NRZ TAPE DLP

Table 4-33. NRZ Tape DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
READ	40						A and B
Forward	40	*	8	2	U	0	Addresses
Backward	41	*	8	3	U	0	Required
Unit Status	40	*	8	E	U	0	
Buffer	40	*	8	F	U	0	
WRITE	42						A & B Addresses
Data		*	4	0	U	0	Required
Buffer		*	4	F	U	0	
REWIND	44	*	2	4	U	0	
REWIND/UNLOAD	44	*	2	5	U	0	
SPACE FORWARD	54	*	2	6	U	c	**C Address
SPACE BACKWARD	54	*	2	7	U	c	Required
ERASE	54	*	2	E	U	c	
TEST	44						
Unit		*	2	0	U	0	
Wait Available		*	2	1	U	0	
Wait Not Available		*	2	2	U	0	
Cancel		*	2	8	U	0	
ID		*	2	C	0	0	
WRITE TAPE MARK	44	*	2	D	U	0	

Table 4-33. NRZ Tape DLP I/O Descriptors (Cont)

NOTES

U designates Unit Number.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory

4-bit = Reserved

2-bit = ASCII translation

1-bit = Reserved

c defines space count. Bit 1 = 1 indicates decimal spacing from 1 to 100 records. Bit 1 = 0 indicates binary spacing from 1 to 256 records.

** C Address middle two digits contain space count for space and erase operations.

0 = Space to tape mark.

	A	B	C	D
8	1 ZERO	5 EARLY TERM-INATION	9 INCORRECT STATE	13 UNIT NOT READY
4	2 DESCRIPTOR ERROR	6 ZERO	10 DIS-CONTINUED/ CANCELLED	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 DLP TIMEOUT	11 DATA ERROR	15 TRANS-MISSION ERROR
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID INFOR-MATION	12 HOST ACCESS ERROR	16 EXCEPTION

Figure 4-57. NRZ Tape DLP Word 1 Result Descriptor

	A	B	C	D
8	1 ZERO	5 WRITE LOCKOUT	9 BOT	13 ZERO
4	2 MTU NOT READY	6 TAPE MARK	10 EOT	14 DATA NOT TRANS-FERRED
2	3 ZERO	7 ZERO	11 REWINDING	15 ZERO
1	4 ZERO	8 DEVICE TIMEOUT	12 ZERO	16 SHORT BLOCK

Figure 4-58. NRZ Tape DLP Word 2 Result Descriptor

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-59. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

A				B				C				D			
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
0	0	0	0	1	1	1	0	*	*	*	*	*	*	*	*

FIXED BYTE

FIELD INSTALLED JUMPERS
* JUMPERS

Figure 4-59. NRZ Tape DLP Test/ID Word 2 Result Descriptor

The Test/ID jumpers are located on the Peripheral Dependent Card, (CD, NRZ/1). The following list the jumper positions.

Table 4-34. NRZ Tape DLP Test/ID Jumpers

Digit Bit	Bit Value	Jumper From	Jumper To
C8	128	VB 53	VC 53
C4	64	VB 54	VC 54
C2	32	VB 55	VC 55
C1	16	VB 56	VC 56
D8	8	VB 57	VC 57
D4	4	VB 58	VC 58
D2	2	VB 59	VC 59
D1	1	VB 60	VC 60

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-60.

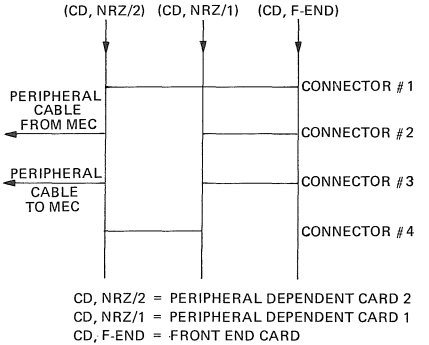


Figure 4-60. NRZ Tape DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-61.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND

Figure 4-61. NRZ Tape DLP Voltage Test Points

BUFFERED PRINTER DLP

Table 4-35. Buffered Printer DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Read Unit Status	40	*	8	E	0	0	A & B Addresses
Read Buffer		*	8	F	0	0	Required
Write Line	42	*	4	0	0	N	A & B Addresses
Write Line/Translate		*	4	1	0	N	Required
Write Buffer		*	4	E	0	0	
Load Translate Table		*	4	2	0	0	
TEST	44	*	2	0	0	0	
Wait Ready		*	2	1	0	0	
Wait Not Ready		*	2	2	0	0	
Conditional Cancel		*	2	8	0	0	
ID		*	2	C	0	0	
Move Paper	44	*	2	3	0	N	

NOTES

N designates the bits used to encode the paper motion format. See Table 4-36.

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

Table 4-36. Buffered Printer DLP Paper Motion Format Codes

Code	Description	Print Format Tape
0 0 0 0	No paper Motion	12 Channel Tape
0 0 0 1	Advance to Heading	12 Channel Tape
0 0 1 0	Advance to Channel 2	12 Channel Tape
0 0 1 1	Advance to Channel 3	12 Channel Tape
0 1 0 0	Advance to Channel 4	12 Channel Tape
0 1 0 1	Advance to Channel 5	12 Channel Tape
0 1 1 0	Advance to Channel 6	12 Channel Tape
0 1 1 1	Advance to Channel 7	12 Channel Tape
1 0 0 0	Advance to Channel 8	12 Channel Tape
1 0 0 1	Advance to Channel 9	12 Channel Tape
1 0 1 0	Advance to Channel 10	12 Channel Tape
1 0 1 1	Advance to Channel 11	12 Channel Tape
1 1 0 0	Advance to End Of Page	12 Channel Tape
1 1 1 0	Single Space	12 Channel Tape
1 1 1 1	Double Space	12 Channel Tape

	A	B	C	D
8	1 NOT READY	5 ZERO	9 INCORRECT STATE	13 ZERO
4	2 DESCRIPTOR ERROR	6 ZERO	10 ZERO	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 ZERO	11 DATA ERROR	15 TRANS-MISSION ERROR
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID INFOR-MATION	12 ZERO	16 EXCEPTION

Figure 4-62. Buffered Printer DLP Result Descriptor Word 1

	A	B	C	D
8	1 MESSAGE ERROR	5 END OF PAGE	9 PRINTER LPC ERROR	13 ZERO
4	2 MESSAGE PARITY ERROR	6 INVALID PRINT CHARACTER	10 ZERO	14 DATA NOT TRANS-FERRED
2	3 MESSAGE LPC ERROR	7 PRINTER MESSAGE ERROR	11 TRANSLATE TABLE NOT LOADED	15 ZERO
1	4 PAPER EMPTY	8 PRINTER PARITY ERROR	12 ZERO	16 ZERO

Figure 4-63. Buffered Printer Result Descriptor Word 2

Address Jumpers

See table 4-2.

Local Address Expander Jumper

This jumper is located on the Common Front End Card. It is reserved for expansion to 16 DLP/Base.

Jumper V071 to V072

Test/ID Jumpers

Refer to Figure 4-64. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

A				B				C				D			
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
0	0	0	1	0	0	1	1	*	*	*	*	*	*	*	*

FIXED BYTE

FIELD INSTALLED JUMPERS
* JUMPERS

Figure 4-64. Buffered Printer DLP Test/ID Word 2 Result Descriptor

The Test/ID jumpers are located on the Peripheral Dependent Card. Table 4-37 lists the jumper positions.

Table 4-37. Buffered Printer DLP Test/ID Jumpers

Digit Bit	Bit Value	Jumper From	Jumper To
C8	128	VB 53	VC 53
C4	64	VB 54	VC 54
C2	32	VB 55	VC 55
C1	16	VB 56	VC 56
D8	8	VB 57	VC 57
D4	4	VB 58	VC 58
D2	2	VB 59	VC 59
D1	1	VB 60	VC 60

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-65.

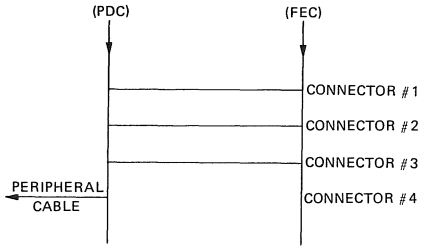


Figure 4-65. Buffered Printer DLP Frontplane Connectors

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-66.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND
-12V	103	-11.5V TO -12.5V
+12V	003	11.5V TO 12.5V

Figure 4-66. Buffer Printer DLP Voltage Test Points

SSP DLP

Additional Shared System Processor DLP information appears in section 5.

Table 4-38. SSP DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Read Status	40	*	8	1	0	0	A and B
Memory Dump		*	8	2	v	0	Addresses
Report Long		*	8	3	0	0	Required
Report Short		*	8	4	0	0	
Report Quick		*	8	5	0	0	
Read Buffer		*	8	F	U	0	
Load	42	*	4	1	v	0	A and B
Set Time		*	4	2	0	0	Addresses
Check		*	4	3	0	0	Required
Lock		*	4	4	0	0	
Unlock Single		*	4	5	0	m	
Clear Single		*	4	6	0	m	
Write Buffer		*	4	F	0	0	
Unlock All	44	*	2	1	0	m	
Clear All		*	2	2	0	m	
Test ID		*	2	C	0	0	

Table 4-38. SSP DLP I/O Descriptors (Cont)

NOTES

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows:

8-bit = Inhibit data transfer to memory
4-bit = Reserved
2-bit = ASCII translation
1-bit = Reserved

v defines operation type. Bit = 1 indicates a first operation. Bit 2 = 1 indicates a continuation operation. If an first operation is in progress, and another first operation is received, the current operation is aborted. The new operation is then initiated. If a continuation operation is received while no first operation is in progress, the continuation is rejected and the SSP returns an incorrect state RD.

m defines host mask field.

m = 8 Host number 3
m = 4 Host number 2
m = 2 Host number 1
m = 1 Host number 0

	A	B	C	D
8	1 ZERO	5 EARLY TERM-INATION	9 INCORRECT STATE	13 INVALID HOST NUMBER
4	2 DESCRIPTOR ERROR	6 ZERO	10 ZERO	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 ZERO	11 DATA ERROR	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID INFOR-MATION	12 ZERO	16 EXCEPTION

Figure 4-67. SSP Word 1 Result Descriptor

	A	B	C	D
8	1 LOAD MARK MISMATCH	5 ZERO	9 NO ENTRY	13 ZERO
4	2 LOAD CODE FILE TOO LARGE	6 ZERO	10 SSP FULL	14 ZERO
2	3 INVALID CHECK SUM	7 ZERO	11 LOCKED BEFORE OPERATION	15 ZERO
1	4 SEQUENCE COMPLETE	8 DIFFERENT HOST	12 ZERO	16 ZERO

Figure 4-68. SSP Word 2 Result Descriptor

Address Jumpers

Each DLP has address jumper pins for system initiated connections (LCPADn), DLP request jumper pins for DLP-initiated connections (LCPRQn) and maintenance address jumper pins for maintenance operations (MADn-m).

The three sets of address jumper pins are located on the DLI/MLI Card (CD,DLI/MLI). The three addresses must have the same value. Tables 4-39 and 4-40 show the required jumpers for a given address.

Table 4-39. SSP DLP Address Jumpers

Address Value	Logic Name	Jumper	
		From	To
DLP 0	LCPRQ 0	K6 E1	K6 I5
	LCPAD 0	N6 B1	M6 I5
DLP 1	LCPRQ 1	K6 E1	K6 J5
	LCPAD 1	N6 B1	M6 J5
DLP 2	LCPRQ 2	K6 E1	L6 I1
	LCPAD 2	N6 B1	M6 G5
DLP 3	LCPRQ 3	K6 E1	L6 J1
	LCPAD 3	N6 B1	M6 H5

Table 4-39. SSP DLP Address Jumpers (Cont)

Address Value	Logic Name	Jumper	
		From	To
DLP 4	LCPRQ 4	K6 E1	K6 I1
	LCPAD 4	N6 B1	M6 E5
DLP 5	LCPRQ 5	K6 E1	K6 J1
	LCPAD 5	N6 B1	M6 E5
DLP 6	LCPRQ 6	K6 E1	K6 H1
	LCPAD 6	N6 B1	M6 D5
DLP 7	LCPRQ 7	K6 E1	K6 G1
	LCPAD 7	N6 B1	M6 D5

Table 4-40. SSP DLP Maintenance Address Straps

Address	1	2	3
0	VJ72 to VJ71	N6G1 to P6E1	M6B5 to M5H5
1	VJ72 to MJ71	N6G1 to P6D1	M6B5 to M5H5
2	VJ72 to VJ71	N6G1 to P6C1	M6B5 to M5H5
3	VJ72 to VJ71	N6G1 to P6B1	M6B5 to M5H5
4	VJ72 to VJ71	N6G1 to P6A1	M6B5 to M5H5
5	VJ72 to VJ71	N6G1 to P6F1	M6B5 to M5H5
6	VJ72 to VJ71	N6G1 to P6G1	M6B5 to M5H5
7	VJ72 to VJ71	N6G1 to P6H1	M6B5 to M5H5

Test/ID Jumpers

Refer to Figure 4-69. Digits C and D require unique jumpers to identify the ID number (0-255), of this particular DLP within the system.

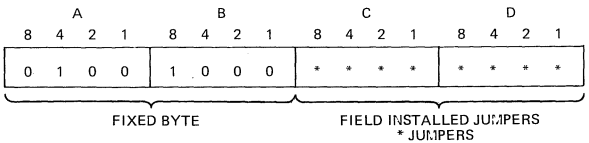


Figure 4-69. SSP Test/ID Word 2 Result Descriptor

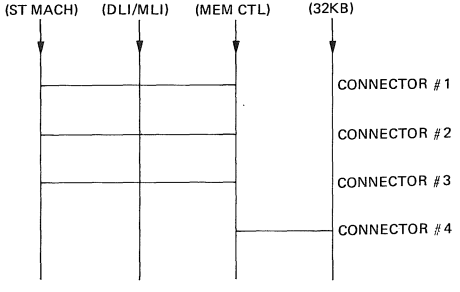
The Test/ID jumpers are located on the DLI/MLI Card, (CD, DLI/MLI). Table 4-41 shows the jumper positions.

Table 4-41. SSP DLP Test/ID Jumpers

Digit Bit	Bit Value	Jumper From	Jumper To
C8	128	A3 B5	B3 B1
C4	64	A3 C5	B3 C1
C2	32	A3 F5	B3 D1
C1	16	A3 H5	B3 E1
D8	8	A3 J5	B3 F1
D4	4	A3 I5	B3 G1
D2	2	A3 G5	B3 H1
D1	1	A3 D5	B3 I1

Frontplane Connectors

Frontplane connectors are installed as shown in figure 4-70.

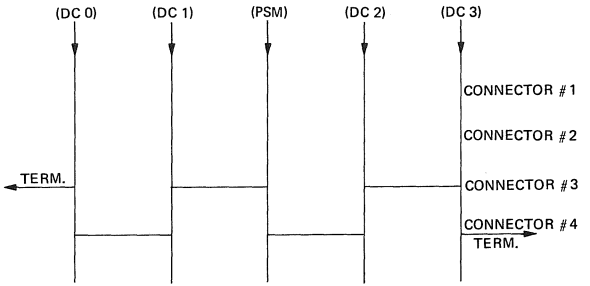


ST MACH = STATE MACHINE CARD
MLI/MLI = DLI/MLI INTERFACE CARD
MEM CTL = SSP MEMORY CONTROL
32KB = 32KB RAM WITH TERMINATION

Figure 4-70. SSP Frontplane Connectors

Distribution Card and Path Selection Module Connections

Each Host system requires a Distribution Card. One Path Selection Module (per SSP) is required. Figure 4-71 shows the recommended configuration for a four system SSP.



DC = DISTRIBUTION CARD
PSM = PATH SELECTION MODULE
TERM. = TERMINATOR

NOTE

DC 0 should be located in the left-most backplane location of the Base Module.

Figure 4-71. Distribution Card and Path Selection Connections

Voltage Test Points

Verify that the correct voltage appears at the pins shown in figure 4-72.

VOLTAGE	PIN	RANGE
+5V	001 002 025 075 096 097	+4.5 TO 5.5 VOLTS
GND	101 102 148 175 196 197	GND

Figure 4-72. SSP DLP Voltage Test Points

SECTION 5

SHARED SYSTEM PROCESSOR

Introduction

The Shared System Processor (SSP) is a programmable DLP that allows up to four Medium Systems (B x800 and B x900) to share the same disk or disk pack resources. The 4-card SSP is installed in an IODC Base Module and therefore must reside in a B x900 Medium System.

In situations where several programs access the same data file, some means of preventing simultaneous updates (locking) must be provided. If locking is not provided updates can be lost.

When locking is not provided, the following can occur.

1. Program 1 reads a record and proceeds to modify the data.
2. Program 2 reads the same record, modifies it, and then writes it back to the data file.
3. Program 1 finishes its modification and writes the record back in the data file; destroying the modification performed by Program 2.

One solution to prevent lost updates is not to allow more than one program to access the same data file. This solution is not acceptable in shared system applications where numerous updates are done on one master data file.

The SSP prevents access to a particular record by a program while another program is updating that record. This method of access prevention is often referred to as locking. By using the situation in the previous example, the following sequence of events will occur.

1. Program 1 reads the record by:

   a. Locking the record by means of an I/O descriptor to the SSP.
   b. Reading the record by means of an I/O descriptor to the Disk (or pack) DLP.

2. Program 2 attempts to read the record by:

   a. Locking the record by means of an I/O descriptor to the SSP.

   However, a result is returned by the SSP stating that the record is locked. The MCP recognizes that the record is locked and will not perform the disk read. The MCP queues the request, and retries the lock process at some later point.

3. Program 1 finishes its update and then:

   a. Writes the record back to disk by means of an I/O descriptor to the disk DLP.

b. Unlocks the record by means of an I/O descriptor to the SSP.

4. Program 2 now attempts to lock the record. The lock is allowed and the program can proceed with the update.

In a maximum shared configuration, either program could have been executed on any one of four host systems.

The block diagram of a typical shared system is illustrated in figure 5-1. Note that the SSP has no direct access to the disk or disk pack subsystem.

The Host System Input/Output Translator (IOT) connects to all base modules by means of a distribution card. The base module in which the SSP is installed will have one distribution card for each host system.

For additional SSP information, see SSP DLP in section 4.

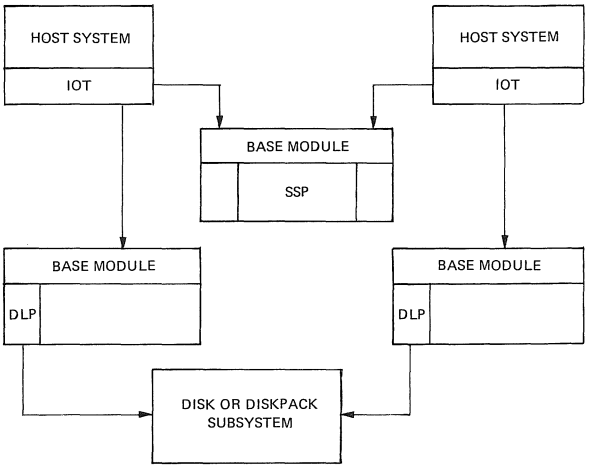


Figure 5-1. Typical Shared System Block Diagram

Overview of SSP Operation

All operations (except Identification and Status) manipulate or search the SSP Table. A microprogram executed by the SSP performs these operations.

The microprogram is located in both PROM and RAM. The portion in RAM must match the PROM version and is loaded (see Loading SSP Firmware) by PKLODR.

NOTE

The RAM portion of the SSP microprogram is referred to as SSP Firmware. The SSP is not operational until this firmware is loaded. The SSP Firmware can only be loaded by PKLODR.

The SSP consists of the following four cards.

State Machine

The State Machine functions as a microprocessor and controls all the operations of the SSP. The State Machine is interruptable and contains logic to save pertinent data before servicing the interrupt.

The State Machine has 8K of PROM present on the card.

DLI/MLI

The DLI/MLI card performs all communication between the DLI (Base Module backplane) and the State Machine. The DLI/MLI is the only card that can interrupt the State Machine.

When the DLI/MLI receives a descriptor (from the DLI) the State Machine is interrupted. This forces the State Machine to an interrupt handling routine, which will read pertinent information from a buffer in the DLI/MLI card. From this information, the State Machine will decode the descriptor and will select the proper routine to perform the operation.

Memory Control

The Memory Control card has logic used to address all RAM memory. Present on the card is 8K of 'fast' RAM, which the State Machine can access in one clock.

The card also contains logic to address up to an additional 131K of RAM located on 32KB RAM cards.

32KB RAM

The 32KB RAM card provides an additional 16K words of RAM with termination.

The SSP Table

The SSP maintains information concerning the status of records in an internal table. This table is referred to as the SSP Table, and stores up to 256 entries. Each entry is 40 bits wide and contains the following information.

Bit	Significance
1	Address is locked
2	Stored as 0 in report
3-4	Host System for which address is locked = 0 System 0 = 1 System 1 = 2 System 2 = 3 System 3
5	System 3 contending for address
6	System 2 contending for address
7	System 1 contending for address
8	System 0 contending for address
9-40	Disk address (8 digits)

NOTE

The disk address specifies the record. The two most significant digits are the logical EU number (as declared at Coldstart) of the disk or pack. The other six digits represents the starting sector address of the record block.

SSP DLP I/O Descriptors

Table 5-1 is a list of I/O Descriptors used by the SSP. Following the list is a brief explanation of each descriptor and the effect it has on the SSP Table.

Table 5-1. SSP DLP I/O Descriptors

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Read Status	40	*	8	1	0	0	A and B
Memory Dump		*	8	2	v	0	Addresses
Report Long		*	8	3	0	0	Required
Report Short		*	8	4	0	0	
Report Quick		*	8	5	0	0	
Read Buffer		*	8	F	U	0	
Load	42	*	4	1	v	0	A and B
Set Time		*	4	2	0	0	Addresses
Check		*	4	3	0	0	Required
Lock		*	4	4	0	0	
Unlock Single		*	4	5	0	m	

Table 5-1. SSP DLP I/O Descriptors (Cont)

Operation	SYS OP	S	MLI OP	L1	L2	L3	Addresses
Clear Single		*	4	6	0	m	
Write Buffer		*	4	F	0	0	
Unlock All	44	*	2	1	0	m	
Clear All		*	2	2	0	m	
Test ID		*	2	C	0	0	

NOTES

MLI OP defines the actual OP Code sent to the DLP by the IOT. This code is not in system memory.

* S Digit specifies IOT operation as follows.

8-bit = Inhibit data transfer to memory

4-bit = Reserved

2-bit = ASCII translation

1-bit = Reserved

v defines operation type. Bit 1 = 1 indicates a first operation. Bit 2 = 1 indicates a continuation operation. If an first operation is in progress, and another first operation is received, the current operation is aborted. The new operation is then initiated. If a continuation operation is received while no first operation is in progress, the continuation is rejected and the SSP returns an incorrect state RD.

m defines host mask field.

m = 8 Host number 3

m = 4 Host number 2

m = 2 Host number 1

m = 1 Host number 0

Read Status, interrogates the SSP maintained time-of-day clock as well as other SSP status information.

Memory Dump, initiates an SSP memory dump.

Report Long, retreives information on all addresses in the SSP Table that are locked or contended for.

Report Short, retrieves information on the first address in the SSP Table that is unlocked and contended for by a requesting host.

Report Quick, retrieves information on the first address in the SSP Table that is unlocked and contended for by the requesting host.

Read Buffer, used with Write Buffer, echoes data to the SSP buffers.

Load, loads the SSP firmware.

Set Time, sets the SSP maintained time-of-day clock.

Check, checks the lock status of an address in the SSP Table.

Lock, conditionally locks an address in the SSP Table.

Unlock Single, unlocks the address that was locked by the specified host.

Clear Single, removes the contended for bit for the address specified.

Write Buffer, used with Read Buffer echoes data to the SSP buffers.

Unlock All, unlocks all address locked by the specified host.

Clear All, resets the contended for bit (for specified host) in all addresses in the SSP Table.

Test ID, determines DLP type and Configuration ID of the SSP.

SSP DLP Result Descriptors

	A	B	C	D
8	1 ZERO	5 EARLY TERM-INATION	9 INCORRECT STATE	13 INVALID HOST NUMBER
4	2 DESCRIPTOR ERROR	6 ZERO	10 ZERO	14 ZERO
2	3 VERTICAL PARITY ERROR (MLI)	7 ZERO	11 DATA ERROR	15 ZERO
1	4 LONG-ITUDINAL PARITY ERROR (MLI)	8 INVALID INFOR-MATION	12 ZERO	16 EXCEPTION

Figure 5-2. SSP Word 1 Result Descriptor

Descriptor Error, an error was detected in the I/O Descriptor or Descriptor Link. At least one other exception bit is set to indicate the type of error.

Vertical Parity Error, the SSP detected a vertical parity error on an I/O descriptor, descriptor link, or on the data sent from the host. At least one other exception bit is set to indicate the type of error.

Longitudinal Parity Error, the SSP detected a longitudinal parity error on an I/O descriptor, descriptor link, or on data sent from the host. At least one other exception bit is set to indicate the type of error.

Early Termination, the host terminated a data transfer before the number of bytes required by the I/O descriptor were transferred. Data Error and Exception Bit is also be set.

Invalid Information, the SSP detected invalid information in the data of the I/O descriptor fields.

Incorrect State, the SSP state is incompatible with the specified operation. One of two conditions exists. Either the SSP firmware is not loaded or a Memory Dump or Load continuation descriptor was received when there was no sequence in operation.

Data Error, an error was detected in the data transferred between the host and the SSP. At least one other bit is set to indicate the type of error.

Invalid Host Number, one of two conditions exists. Either the Host Return Field in the descriptor link was a number other than 0 to 3, or a continuation (Memory Dump or Load) descriptor was received and the host number did not match the host number which initiated the sequence.

Exception, an unexpected event is reported in the result descriptor. At least one other bit is set to indicate the cause of the exception.

	A	B	C	D
8	1 LOAD MARK MISMATCH	5 ZERO	9 NO ENTRY	13 ZERO
4	2 LOAD CODE FILE TOO LARGE	6 ZERO	10 SSP FULL	14 ZERO
2	3 INVALID CHECK SUM	7 ZERO	11 LOCKED BEFORE OPERATION	15 ZERO
1	4 SEQUENCE COMPLETE	8 DIFFERENT HOST	12 ZERO	16 ZERO

Figure 5-3. SSP Word 2 Result Descriptor

Mark Mismatch, the firmware file being loaded does not match the PROM firmware present. The firmware will not be loaded.

Code File Too Large, the length of the firmware code file exceeds the amount of available memory. The firmware will not be loaded.

Invalid Checksum, one of the firmware code file internal checksums is invalid for a Load operation. The firmware can not be loaded.

Sequence Complete, the sequence for a Load or Memory Dump is complete.

Different Host, the requested address was locked by a different host. The exception bit is not set as a result of this bit.

No Entry, no entry containing the specified address was found for an operation that involved altering or searching of the SSP Table for a single entry.

SSP Full, no more memory locations in the FPM Table are available for use in a Lock operation. The specified address cannot be stored at this time.

Locked Before Operation, at least one SSP Table address was found to be locked. This bit can only occur in operations that alter an entry in the SSP Table or that search the table.

Figure 5-4 details R/D word 2 returned by the SSP on a TEST/ID operation. R/D word 1 is the same as specified in Figure 5-2.

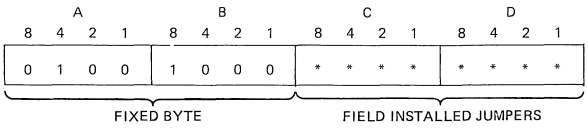


Figure 5-4. SSP Test/ID Word 2 Result Descriptor

SSP Installation

The SSP is installed in an IODC Base Module, and therefore must be installed in a B 2900 or B3955 system. An IODC Base Module with an SSP installed is illustrated in Figure 5-5.

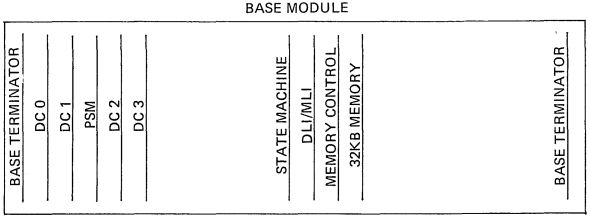


Figure 5-5. SSP Installed In An IODC Base Module

It is recommended that the SSP be installed in a B 2900 or B3955 Extension Cabinet. The extension cabinet provides a power source independent of the processor. Thus, if the processor loses power it is possible that the SSP remains operational to the other sharing processors.

Frontplane connectors for the SSP are installed as shown in Figure 5-6.

Each Host system requires a Distribution Card. One Path Selection Module (per SSP) is required. Figure 5-7 shows the recommended configuration for a four system SSP.

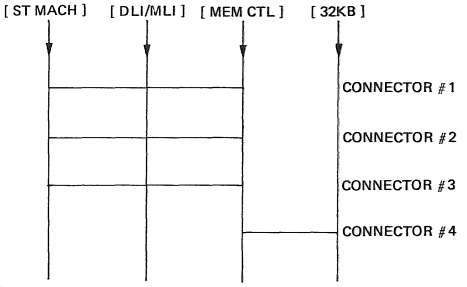
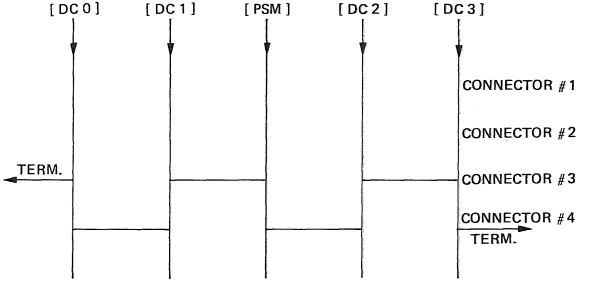


Figure 5-6. SSP Frontplane Connectors



NOTE

Host system connections are made to J1 of the appropriate Distribution Card. For example, the host system strapped as 1, would be connected to J1 of DC 1.

Figure 5-7. Distribution Card and Path Selection Connections

The Distribution Cards, Path Select Module and SSP cards require certain options be strapped. Refer to the B 2900/B3955 Installation Manual and Special Instructions for the above specified units.

Loading SSP Firmware

The SSP requires firmware to be loaded into RAM. This firmware is available on the current MCP System tape. Current file name of the SSP firmware is SSP11A.

To load the firmware use PKLODR. PKLODR is a Control State program and is available on the current MCP System Tape or on the Cardless System Mini-disk. Instructions for PKLODR are located in the Software Operators Guide.

NOTE

The SSP must be cleared before being loaded with firmware. To accomplish clearing, press the BASE CLEAR pushbutton on the plug-on mounted on J2 of the Maintenance Card.

After the SSP is loaded care should be taken to avoid using the CLEAR syntax of the B 2900. For example: the syntax CLEAR, LOAD ALT generates a Master Clear and will clear the SSP firmware. A firmware load will then be required.

Maintenance

The SSP requires no preventive maintenance or adjustments.

For problem definition, SSP maintenance should be performed in the following manner:

1. Run the Test Routine SSPTBO (either normal or control state) to localize the problem. The program SSPTBO is available on the latest B 2900 series test routine release tape. The Current release is TN81AO. The instruction file (SSPTBI) is also available on the tape.
2. If the malfunction appears to be within the SSP subsystem, run the SSP Maintenance Test. The test (SSPMND) is run under control of PTDMNO. These tests and their instruction files are also available on the test routine tape.
3. If the fault appears to be internal to the SSP, run the diagnostics provided in the T & F package for the SSP.

NOTE

The diagnostics run tests on all four SSP cards. However, component repair is required only on the Memory Control and 32KB Memory cards. Faults detected on the State Machine or DLI/MLI Cards are corrected by replacing the cards. A spare State Machine and DLI/MLI card is located in the SSP Maintenance Kit. The SSP Maintenance Kit is not automatically shipped with an SSP order. If the kit is desired is must specified on the order for the SSP.