E3ABP30534M 000
E3AZP30554 000

Technical Training

JSS DATA PROCESSOR AND DISPLAY MAINTENANCE

HMP-1116 PROGRAMMER GUIDE

MAY 1986

USAF TECHNICAL TRAINING SCHOOL
3300th Technical Training WING
Keesler Air Force Base, Mississippi

SYMBOLS AND ABBREVIATIONS

A2		The 16-bit halfword address which is a part of the RX instructions.
CC		Condition Code of four bits contained in the PSW.
	C	Carry Bit contained in the Condition Code (Bit 12 of PSW).
	V	Overflow Bit contained in the Condition Code (Bit 13 of PSW).
	G	Greater Than Bit contained in the Condition Code (Bit 14 of PSW).
	L	Less Than Bit contained in the Condition Code (Bit 15 of PSW).
D		The four-bit displacement value used with Short Format Branch instructions.
I2		The immediate value which is used as the second operand.
N		The four-bit second operand used with Short Format Immediate insturctions.
PSW		Program Status Word of 32 bits containing the Status Condition Code, and current instruction address.
R1		The address of a General Register the content of which is the first operand.
R2		The address of a General Register the content of which is the second operand of RR instruction.
M1		Mask of four-bits specifying Branch on Condition testing.
X2		The address of a General Register the content of which is used as an index value.
( ), [ ]		Parentheses or Brackets. Read as "the content of . . .".
{ }		Quantity
←		Arrow. Read as "is replaced by . . ." or "replaces . . .".
+		Add.
−		Subtract.
*		Multiply.
/		Divide.
:		Logical comparison, when used alone (e.g., R1:R2).
(0:7) (8:15) (16:31)		A bit grouping within a byte, a halfword, or a fullword. Read as "0 thru 7 inclusive," "Bits 8 through 15 inclusive," etc.
(R1,R1+1)		Fullword contained in a pair of registers.
[A2+(X2), A2+(X2)+2]		Fullword located in memory.
(R1,R1+1,R1+2,R1+3)		Doubleword contained in four consecutive registers.

INSTRUCTION REPERTOIRE

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
ACKNOWLEDGE INTERRUPT (RX)	DF	AI	[R1 (8:15)] ← DEVICE ADDRESS [R1 (0:7)] ← ZERO [A2 + (X2)] ← DEVICE STATUS BYTE [PSW (12:15)] ← DEVICE STATUS BYTE (4:7)
ACKNOWLEDGE INTERRUPT (RR)	9F	AIR	[R1 (8:15)] ← DEVICE ADDRESS [R1 (0:7)] ← ZERO [R2 (8:15)] ← DEVICE STATUS BYTE [R2 (0:7)] ← ZERO [PSW (12:15) ← DEVICE STATUS BYTE (4:7)
ADD FULLWORD (RX)	5A	ADP	(R1, R1 + 1) ← (R1, R1 + 1) + [A2 + (X2), A2 + (X2) + 2]
ADD FULL WORD (RR)	1A	ADPR	(R1, R1 + 1) ← (R1, R1 + 1) + (R2, R2 + 1)
ADD HALFWORD (RX)	4A	AH	(R1) ← (R1) + [A2 + (X2)]
ADD HALFWORD IMMEDIATE (RI)	CA	AHI	(R1) ← (R1) + {I2 + (X2)}
ADD HALFWORD (RR)	0A	AHR	(R1) ← (R1) + (R2)
ADD HALFWORD TO MEMORY (RX)	61	AHM	[A2 + (X2)] ← (R1) + [A2 + (X2)]
ADD IMMEDIATE SHORT (SF)	26	AIS	(R1) ← (R1) + N
ADD TO BOTTOM OF LIST (RX)	65	ABL	R1, A2 (X2)
ADD TO TOP OF LIST (RX)	64	ATL	R1, A2 (X2)
ADD WITH CARRY HALFWORD (RX)	4E	ACH	(R1) ← (R1) + [A2 + (X2)] + C
ADD WITH CARRY HALFWORD (RR)	0E	ACHR	(R1) ← (R1) + (R2) + C
AND HALFWORD (RX)	44	NH	(R1) ← (R1) AND [A2 + (X2)]
AND HALFWORD IMMEDIATE (RI)	C4	NHI	(R1) ← (R1) AND {I2 + (X2)}
AND HALFWORD (RR)	04	NHR	(R1) ← (R1) AND (R2)
ARC SINE (RX)	F4	ASN	(R1, R1 + 1) ← ARCSIN (Y/Z) { Z = [A2 + (X2), A2 + (X2) + 2] { Y = [A2 + (X2) + 4, A2 + (X2) + 6]

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
ARC SINE (RR)	B4	ASNR	(R1, R1 + 1) ← ARCSIN (Y/Z) Z = (R2, R2 + 1) Y = (R2 + 2, R2 + 3)
ARC TAN, $\sqrt{X^2 + Y^2}$ (RX)	F2	ATN	(R1, R1 + 1) ← ARCTAN (Y/X) (R1 + 2, R1 + 3) ← $\sqrt{X^2 + Y^2}$ X = [A2 + (X2), A2 + (X2) + 2] Y = [A2 + (X2) + 4, A2 + (X2) + 6]
ARC TAN, $\sqrt{X^2 + Y^2}$ (RR)	B2	ATNR	(R1, R1 + 1) ← ARCTAN (Y/X) (R1 + 2, R1 + 3) ← $\sqrt{X^2 + Y^2}$ X = (R2, R2 + 1) Y = (R2 + 2, R2 + 3)
AUTOLOAD (RX)	DS	AL	1. n ← 0 2. (X'80' + n) ← BYTE 3. n ← n + 1 4. IF A2 + (X2) < X'80' + n, INSTRUCTION IS FINISHED, OTHERWISE RETURN TO STEP 2
BRANCH AND LINK (RX)	41	BAL	(R1) ← [PSW (16:31)] + 4 [PSW (16:31)] ← A2 + (X2)
BRANCH AND LINK (RR)	01	BALR	(R1) ← [PSW (16:31)] + 2 [PSW (16:31)] ← (R2)
BRANCH ON FALSE CONDITION (RX)	43	BFC	FALSE: [PSW (16:31)] ← A2 + (X2) TRUE: [PSW (16:31)] ← [PSW (16:31)] + 4
BRANCH ON FALSE CONDITION (RR)	03	BFCR	FALSE: [PSW (16:31)] ← (R2) TRUE: [PSW (16:31)] ← [PSW (16:31)] + 2
BRANCH ON TRUE CONDITION (RX)	42	BTC	TRUE: [PSW (16:31)] ← A2 + (X2) FALSE: [PSW (16:31)] ← [PSW (16:31)] + 4

INSTRUCTION REPERTOIRE

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
BRANCH ON TRUE CONDITION (RR)	02	BTCR	TRUE: [PSW (16:31)] ← (R2) FALSE: [PSW (16:31)] ← [PSW (16:31)] + 2
BRANCH ON FALSE BACKWARD SHORT (SF)	22	BFBS	FALSE: [PSW (16:31)] ← [PSW (16:31)] -2D TRUE: [PSW (16:31)] ← [PSW (16:31)] +2
BRANCH ON FALSE FORWARD SHORT (SF)	23	BFFS	FALSE: [PSW (16:31)] ← [PSW (16:31)] +2D TRUE: [PSW (16:31)] ← [PSW (16:31)] +2
BRANCH ON TRUE BACKWARD SHORT (SF)	20	BTBS	TRUE: [PSW (16:31)] ← [PSW (16:31)] -2D FALSE: [PSW (16:31)] ← [PSW (16:31)] +2
BRANCH ON TRUE FORWARD SHORT (SF)	21	BTFS	TRUE: [PSW (16:31)] ← [PSW (16:31)] +2D FALSE: [PSW (16:31)] ← [PSW (16:31)] +2
BRANCH ON INDEX HIGH (RX)	C0	BXH	(R1) ← (R1) + (R1 + 1) (R1) : (R1 + 2) IF (R1) > (R1 + 2), THEN [PSW (16:31)] ← A2 + (X2)
BRANCH ON INDEX LOW OR EQUAL (RX)	C1	BXLE	IF (R1) ≤ (R1 + 2), THEN [PSW (16:31)] ← [PSW (16:31)] +4 (R1) ← (R1) + (R1 + 1) (R1) : (R1 + 2) IF (R1) ≤ (R1 + 2), THEN [PSW (16:31)] ← A2 + (X2) (R1) > (R1 + 2), THEN [PSW (16:31)] ← [PSW (16:31)] +4
COMPARE FULLWORD (RX)	59	CDP	(R1, R1 + 1) : [A2 + (X2), A2 + (X2) + 2]
COMPARE FULLWORD (RR)	19	CDPR	(R1, R1 + 1) : (R2, R2 + 1)
COMPARE HALFWORD (RX)	49	CH	(R1) : [A2 + (X2)]
COMPARE HALFWORD IMMEDIATE (RI)	C9	CHI	(R1) : [I2 + (X2)]
COMPARE HALFWORD (RR)	09	CHR	(R1) : (R2)

INSTRUCTION (FORMAT)	CODE	MNEMONIC	DESCRIPTION
COMPARE LOGICAL BYTE (RX)	D4	CLB	(R1 (8:15)) : (A2 + (X2))
COMPARE LOGICAL FULLWORD (RX)	55	CLDP	(R1, R1 + 1) : (A2 + (X2), A2 + (X2) + 2)
COMPARE LOGICAL FULLWORD (RR)	25	CLDPR	(R1, R1 + 1) : (R2, R2 + 1)
COMPARE LOGICAL HALFWORD (RX)	45	CLH	(R1) : (A2 + (X2))
COMPARE LOGICAL HALFWORD IMMEDIATE (RI)	C5	CLHI	(R1) : {I2 + (X2)}
COMPARE LOGICAL HALFWORD (RR)	05	CLHR	(R1) : (R2)
COSINE/SINE (RX)	F1	COS	(R1, R1 + 1) ← COS ((A2 + (X2)), (A2 + (X2) + 2)) (R1 + 2, R1 + 3) ← SIN ((A2 + (X2)), (A2 + (X2) +2))
COSINE/SINE (RR)	B1	COSR	(R1, R1 + 1) ← COS (R2, R2 + 1) (R1 + 2, R1 + 3) ← SIN (R2, R2 + 1)
DIVIDE FULLWORD (RX)	5D	DDP	(R1 + 2, R1 + 3) ← (R1, R1 + 1, R1 + 2, R1 + 3)/ (A2 + (X2), A2 + (X2) + 2) (R1, R1 + 1) ← REMAINDER
DIVIDE FULLWORD (RR)	1D	DDPR	(R1 + 2, R1 + 3) ← (R1, R1 +1, R1 + 2, R1 + 3)/ (R2, R2 + 1) (R1, R1 + 1) ← REMAINDER
DIVIDE HALFWORD (RX)	4D	DH	(R1 + 1) ← (R1, R1 + 1)/(A2 + (X2)) (R1) ← REMAINDER
DIVIDE HALFWORD (RR)	0D	DHR	(R1 + 1) ← (R1, R1 + 1)/(R2) (R1) ← REMAINDER
EXCHANGE BYTE (RR)	94	EXBR	(R1 (10:7)) ← (R2 (8:15)) (R1 (8:15)) ← (R2 (0:7))
EXCHANGE CORDIC PRECISION REGISTER (RR)	B0	ECPR	TEMP ← (CPR) (CPR) ← X'0010' OR (X'001C' AND (R2)) (R1) ← TEMP
EXCHANGE OPERAND BANK ADDRESS (RR)	2E	EPOR	(PSW (10:11)) ← (R1 (10:11)) (R1 (0:15)) ← FORMER (PSW (0:15))

INSTRUCTION REPERTOIRE

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
EXCHANGE PROGRAM ADDRESS (RR)	2F	EPPR	[PSW (8:11)] ← [R1 (8:11)] [PSW (16:31)] ← (R1 + 1) (R1) ← FORMER [PSW (0:15)] (R1 + 1) ← FORMER [PSW (16:31)]
EXCHANGE PROGRAM STATUS (RR)	95	EPSR	[PSW (0:15)] → (R1) [PSW (0:15)] ← (R2)
EXCLUSIVE OR HALFWORD (RX)	47	XH	(R1) ← (R1) XOR [A2 + (X2)]
EXCLUSIVE OR HALFWORD (RI)	C7	XHI	(R1) ← (R1) XOR [I2 + (X2)]
EXCLUSIVE OR HALFWORD (RR)	07	XHR	(R1) ← (R1) XOR (R2)
FLOATING POINT (48-BIT) ADD (RX)	7A	AF	(R1) ← (R1) + [A2 + (X2)]
FLOATING POINT (48-BIT) ADD (RR)	3A	AFR	(R1) ← (R1) + (R2)
FLOATING POINT (48-BIT) COMPARE (RX)	79	CF	(R1) : [A2 + (X2)]
FLOATING POINT (48-BIT) COMPARE (RR)	39	CFR	(R1) : (R2)
FLOATING POINT (48-BIT) DIVIDE (RX)	7D	DF	(R1) ← (R1)/[A2 + (X2)]
FLOATING POINT (48-BIT) DIVIDE (RR)	3D	DFR	(R1) ← (R1)/(R2)
FLOATING POINT (48-BIT) LOAD (RX)	78	LF	(R1) ← [A2 + (X2)]
FLOATING POINT (48-BIT) LOAD (RR)	38	LFR	(R1) ← (R2)
FLOATING POINT (48-BIT) MULTIPLY (RX)	7C	MF	(R1) * [A2 + (X2)]
FLOATING POINT (48-BIT) MULTIPLY (RR)	3C	MFR	(R1) ← (R1) * (R2)
FLOATING POINT (48-BIT) STORE (RX)	70	STF	[A2 + (X2)] ← (R1)
FLOATING POINT (48-BIT) SUBTRACT (RX)	7B	SF	(R1) ← (R1) - [A2 + (X2)]
FLOATING POINT (48-BIT) SUBTRACT (RR)	3B	SFR	(R1) ← (R1) - (R2)
FLOATING — POINT (32-BIT) ADD (RX)	6A	AE	(R1) ← (R1) + [AE + (X2)]
FLOATING — POINT (32-BIT) ADD (RR)	2A	AER	(R1) ← (R1) + (R2)
FLOATING — POINT (32-BIT) COMPARE (RX)	69	CE	(R1) : [A2 + (X2)]
FLOATING — POINT (32-BIT) COMPARE (RR)	29	CER	(R1) : (R2)

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
FLOATING — POINT (32-BIT) DIVIDE (RX)	6D	DE	(R1) ← (R1)/[A2 + (X2)]
FLOATING — POINT (32-BIT) DIVIDE (RR)	2D	DER	(R1) ← (R1)/(R2)
FLOATING — POINT (32-BIT) LOAD (RX)	68	LE	(R1) ← [A2 + (X2)]
FLOATING — POINT (32-BIT) LOAD (RR)	28	LER	(R1) ← (R2)
FLOATING — POINT (32-BIT) MULTIPLY (RX)	6C	ME	(R1) ← (R1) * [A2 + (X2)]
FLOATING — POINT (32-BIT) MULTIPLY (RR)	2C	MER	(R1) ← (R1) * (R2)
FLOATING — POINT (32-BIT) STORE (RX)	60	STE	[A2 + (X2)] ← (R1)
FLOATING — POINT (32-BIT) SUBTRACT (RX)	6B	SE	(R1) ← (R1) − [A2 + (X2)]
FLOATING — POINT (32-BIT) SUBTRACT (RR)	2B	SER	(R1) ← (R1) − (R2)
LOAD ABSOLUTE VALUE HALFWORD (RX)	52	LAV	IF [A2 + (X2)] > 0 THEN: (R1) ← [A2 + (X2)] IF [A2 + (X2)] < 0 THEN: (R1) ← − [A2 + (X2)]

LOAD ABSOLUTE VALUE HALFWORD (RR)	11	LAVR	IF (R2) > 0
			THEN: (R1) ← (R2)
			IF (R2) < 0
			THEN: (R1) ← −(R2)

LOAD AND CHANGE NUMBER BASE HALFWORD (RX)	51	LCNH	IF [A2 + (X2)] > 0 THEN: (R1) ← [A2 + (X2)] IF [A2 + (X2)] < 0 THEN: (R1) ← −([A2 + (X2)])

LOAD AND CHANGE NUMBER BASE HALFWORD (RR)	10	LCNHR	IF (R2) > 0
			THEN: (R1) ← (R2)
			IF (R2) < 0
			THEN: (R1) ← − (R2)

INSTRUCTION REPERTOIRE

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
LOAD BYTE (RX)	D3	LB	(R1 (8:15)) ← (A2 + (X2)) (R1 (0:7)) ← ZERO
LOAD BYTE (RR)	93	LBR	(R1 (8:15)) ← (R2 (8:15)) (R1 (0:7)) ← ZERO
LOAD COMPLEMENT HALFWORD (RX)	53	LCH	(R1) ← -(A2 + (X2))
LOAD COMPLEMENT HALFWORD (RR)	12	LCHR	(R1) ← -(R2)
LOAD COMPLEMENT SHORT (SF)	25	LCS	(R1) ← -N
LOAD FROM BANK 0 (RX)	74	LH0	(R1) ← (A2 + (X2) + 0)
LOAD FROM BANK 1 (RX)	75	LH1	(R1) ← (A2 + (X2) + 65,536)
LOAD FROM BANK 2 (RX)	76	LH2	(R1) ← (A2 + (X2) + 131,072)
LOAD FROM BANK 3 (RX)	77	LH3	(R1) ← (A2 + (X2) + 196,608)
LOAD FULLWORD (RX)	58	LDP	(R1, R1 + 1) ← (A2 + (X2), A2 + (X2) +2)
LOAD FULLWORD (RR)	18	LDPR	(R1, R1 + 1) ← (R2, R2 + 1)
LOAD HALFWORD (RX)	48	LH	(R1) ← (A2 + (X2))
LOAD HALFWORD IMMEDIATE (RI)	C8	LHI	(R1) ← (I2 + (X2))
LOAD HALFWORD (RR)	08	LHR	(R1) ← (R2)
LOAD IMMEDIATE SHORT (SF)	24	LIS	(R1) ← N
LOAD MULTIPLE (RX)	D1	LM	1. (R1) ← (A2 + (X2)) 2. R1: X'F' IF R1 = X'F', THE INSTRUCTION IS FINISHED IF R1 ≠ X'F', THEN: 3. R1 ← R1 + 1 4. A2 ← A2 + 2, RETURN TO STEP 1
LOAD PROGRAM STATUS WORD (RX)	C2	LPSW	(PSW (0:31)) ← (A2 + (X2))
MULTIPLY FULLWORD (RX)	5C	MDP	(R1, R1 + 1, R1 + 2, R1 + 3) ← (R1 + 2, R1 + 3) * (A2 + (X2), A2 + (X2) + 2)
MULTIPLY FULLWORD (RR)	1C	MDPR	(R1, R1 + 1, R 1 + 2, R 1 + 3) ← (R1 + 2, R1 + 3) * (R2, R2 + 1)

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
MULTIPLY HALFWORD (RX)	4C	MH	(R1, R1 + 1) ← (R1 + 1) * [A2 + (X2)]
MULTIPLY HALFWORD (RR)	0C	MHR	(R1, R1 + 1) ← (R1 + 1) * (R2)
MULTIPLY HALFWORD UNSIGNED (RX)	DC	MHU	(R1, R1 + 1) ← (R1 + 1) * [A2 + (X2)]
MULTIPLY HALFWORD UNSIGNED (RR)	9C	MHUR	(R1, R1 + 1) ← (R1 + 1) * (R2)
OR HALFWORD (RX)	46	OH	(R1) ← (R1) OR [A2 + (X2)]
OR HALFWORD IMMEDIATE (RI)	C6	OHI	(R1) ← (R1) OR [I2 + (X2)]
OR HALFWORD (RR)	06	OHR	(R1) ← (R1) OR (R2)
OUTPUT COMMAND (RX)	DE	OC	DEVICE ← [A2 + (X2)]
OUTPUT COMMAND (RR)	9E	OCR	DEVICE ← [R2 (8:15)]
READ BLOCK (RX)	D7	RB	1. N = [A2 + (X2)] 2. IF N > [A2 + (X2) + 2] THEN TERMINATE WITH A CONDITION CODE = 0000 ELSE: 3. (N) ← DATA BYTE 4. N ← N + 1 5. RETURN TO STEP 2
READ BLOCK (RR)	97	RBR	1. N = (R2) 2. IF N > (R2 + 1), THEN: TERMINATE WITH A CONDITION CODE = 0000 ELSE: 3. (N) ← DATA BYTE 4. N = N + 1 5. RETURN TO STEP 2
READ DATA (RX)	DB	RD	[A2 + (X2)] ← DATA BYTE

INSTRUCTION REPERTOIRE

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
READ DATA (RR)	9B	RDR	[R2 (8:15)] ← DATA BYTE [R2 (0:7)] ← ZERO
READ HALFWORD (RX)	D9	RH	[A2 + (X2)] ← FIRST DATA BYTE [A2 + (X2) + 1] ← SECOND DATA BYTE [A2 + (X2)] ← HALFWORD OF DATA
READ HALFWORD (RR)	99	RHR	[R2 (0:7)] ← FIRST DATA BYTE [R2 (8:15)] ← SECOND DATA BYTE [R2 (0:15)] ← HALFWORD OF DATA
RESET STATUS BITS (RI)	E3	RESB	[PSW (0:7)] ← [PSW (0:7)] AND [I2 (0:7)]
ROTATE LEFT LOGICAL (RI)	EB	RLL	R1, I2 (X2)
ROTATE RIGHT LOGICAL (RI)	EA	RRL	R1, I2 (X2)
REMOVE FROM BOTTOM OF LIST (RX)	67	RBL	R1, A2 (X2)
REMOVE FROM TOP OF LIST (RX)	66	RTL	R1, A2 (X2)
SENSE STATUS (RX)	OD	SS	[A2 + (X2)] ← DEVICE STATUS BYTE [PSW (12:15)] ← DEVICE STATUS BYTE (4:7)
SENSE STATUS (RR)	9D	SSR	[R2 (8:15)] ← DEVICE STATUS BYTE [R2 (0:7)] ← ZERO [PSW (12:15)] ← DEVICE STATUS BYTE (4:7)

8-BIT ORIENTED DEVICE CONTROLLER

16-BIT ORIENTED DEVICE CONTROLLER

8-BIT ORIENTED DEVICE CONTROLLER

16-BIT ORIENTED DEVICE CONTROLLER

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
SET STATUS BITS	E4	SESB	[PSW (0:7)] ← [PSW (0:7)] OR [I2 (0:7)]
SHIFT LEFT DOUBLEWORD ARITHMETIC (RI)	E9	SLQA	R1, I2 (X2)
SHIFT LEFT DOUBLEWORD LOGICAL (RI)	E7	SLQL	R1, I2 (X2)
SHIFT LEFT FULLWORD ARITHMETIC (RI)	EF	SLA	R1, I2 (X2)
SHIFT LEFT FULLWORD LOGICAL (RI)	ED	SLL	R1, I2 (X2)
SHIFT LEFT HALFWORD ARITHMETIC (RI)	CF	SLHA	R1, I2 (X2)
SHIFT LEFT HALFWORD LOGICAL (RI)	CD	SLHL	R1, I2 (X2)
SHIFT LEFT LOGICAL SHORT (SF)	91	SLLS	R1, N
SHIFT RIGHT DOUBLEWORD ARITHMETIC (RI)	E8	SRQA	R1, I2 (X2)
SHIFT RIGHT DOUBLEWORD LOGICAL (RI)	E6	SRQL	R1, I2 (X2)
SHIFT RIGHT FULLWORD ARITHMETIC (RI)	EE	SRA	R1, I2 (X2)
SHIFT RIGHT FULLWORD LOGICAL (RI)	EC	SRL	R1, I2 (X2)
SHIFT RIGHT HALFWORD ARITHMETIC (RI)	CE	SRHA	R1, I2 (X2)
SHIFT RIGHT HALFWORD LOGICAL (RI)	CC	SRHL	R1, I2 (X2)
SHIFT RIGHT LOGICAL SHORT (SF)	90	SRLS	R1, N
SIMULATE INTERRUPT (RI)	E2	SINT	I2 (X2)
SQUARE ROOT OF THE DIFFERENCE $\sqrt{x^2 - y^2}$ (RX)	F3	SQD	(R1, R1 + 1) ← $\sqrt{x^2 - y^2}$ where X = [A2 + (X2), A2 + (X2) + 2], Y = [A2 + (X2) + 4, A2 + (X2) + 6]
SQUARE ROOT OF THE DIFFERENCE $\sqrt{x^2 - y^2}$ (RR)	B3	SQDR	(R1, R1 + 1) ← $\sqrt{x^2 - y^2}$ where X = (R2, R2 + 1), Y = (R2 + 2, R2 + 3)
STORE BYTE (RX)	D2	STB	[R1 (8:15)] → [A2 + (X2)]
STORE BYTE (RR)	92	STBR	[R1 (8:15)] → [R2 (8:15)]
STORE FULLWORD (RX)	50	STDP	(R1, R1 + 1) → [A2 + (X2), A2 + (X2) + 2]

INSTRUCTION REPERTOIRE

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
STORE HALFWORD (RX)	40	STH	(R1) → [A2 + (X2)]
STORE MULTIPLE (RX)	D0	STM	1. (R1) → [A2 + (X2)] 2. R1: X'F' IF R1 ≠ X'F', THE INSTRUCTION IS FINISHED IF R1 ≠ X'F', THEN: 3. R1 ← R1 + 1 4. A2 ← A2 + 2, RETURN TO STEP 1
STORE IN BANK 0 (RX)	F8	STH0	(R1) → [A2 + (X2) + 0]
STORE IN BANK 1 (RX)	F9	STH1	(R1) → [A2 + (X2) + 65,536]
STORE IN BANK 2 (RX)	FA	STH2	(R1) → [A2 + (X2) + 131,072]
STORE IN BANK 3 (RX)	FB	STH3	(R1) → [A2 + (X2) + 196,608]
SUBTRACT FULLWORD (RX)	5B	SDP	(R1, R1 + 1) ← (R1, R1 + 1) - [A2 + (X2), A2 + (X2) + 2]
SUBTRACT FULLWORD (RR)	1B	SDPR	(R1, R1 + 1) ← (R1, R1 + 1) - (R2, R2 + 1)
SUBTRACT HALFWORD (RX)	4B	SH	(R1) ← (R1) - [A2 + (X2)]
SUBTRACT HALFWORD IMMEDIATE (RI)	CB	SHI	(R1) ← (R1) - {I2 + (X2)}
SUBTRACT HALFWORD (RR)	0B	SHR	(R1) ← (R1) - (R2)
SUBTRACT IMMEDIATE SHORT (SF)	27	SIS	(R1) ← (R1) - N
SUBTRACT WITH CARRY HALFWORD (RX)	4F	SCH	(R1) ← (R1) - [A2 + (X2)] - C
SUBTRACT WITH CARRY HALFWORD (RR)	0F	SCHR	(R1) ← (R1) - (R2) - C
SUPERVISOR CALL (RX)	E1	SVC	(X'0094') ← A2 + (X2) (X'0096') ← [PSW (0:15)] (X'0098') ← [PSW (16:31)] + 4 (X'009A') → [PSW (0:15)] (X'009C' + 2 * R1) → [PSW (16:31)]
TEST HALFWORD IMMEDIATE (RI)	C3	THI	(R1) AND {I2 + (X2)}

INSTRUCTION (FORMAT)	OP-CODE	MNEMONIC	DESCRIPTION
WRITE BLOCK (RX)	D6	WB	1. N = [A2 + (X2)] 2. IF N > [A2 + (X2) + 2] THEN: TERMINATE WITH A CONDITION CODE = 0000 ELSE: 3. (N) → DEVICE 4. N ← N + 1 5. RETURN TO STEP 2
WRITE BLOCK (RR)	96	WBR	1. N ← (R2) 2. IF N > (R2 + 1) THEN: TERMINATE WITH A CONDITION CODE = 0000 ELSE: 3. (N) → DEVICE 4. N ← N + 1 5. RETURN TO STEP 2
WRITE DATA (RX)	DA	WD	[A2 + (X2)] → DEVICE
WRITE DATA (RR)	9A	WDR	[R2 (8:15)] → DEVICE
WRITE HALFWORD (RX)	D8	WH	[A2 + (X2)] → DEVICE — 8 BIT ORIENTED DEVICE CONTROLLER [A2 + (X2) + 1] → DEVICE [A2 + (X2)] → DEVICE — 16 BIT ORIENTED DEVICE CONTROLLER
WRITE HALFWORD (RR)	98	WHR	[R2 (0:7)] → DEVICE — 8 BIT ORIENTED DEVICE CONTROLLER [R2 (8:15)] → DEVICE [R2 (0:15)] → DEVICE — 16 BIT ORIENTED DEVICE CONTROLLER

OP CODE MAP

MSD ↓ / LSD →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		BALR	BTCR	BFCR	NHR	CLHR	OHR	XHR	LHR	CHR	AHR	SHR	MHR	DHR	ACHR	SCHR	RR
1	LCNHR	LAVR	LCHR			CLDPR			LDPR	CDPR	ADPR	SDPR	MDPR	DDPR			RR
2	BTBS	BTFS	BFBS	BFFS	LIS	LCS	AIS	SIS	LER	CER	AER	SER	MER	DER	EPOR	EPPR	SF/RR
3									LFR	CFR	AFR	SFR	MFR	DFR			RR
4	STH	BAL	BTC	BFC	NH	CLH	OH	XH	LH	CH	AH	SH	MH	DH	ACH	SCH	RX
5	STDP	LCNH	LAV	LCH		CLDP			LDP	CDP	ADP	SDP	MDP	DDP			RX
6	STE	AHM			ATL	ABL	RTL	RBL	LE	CE	AE	SE	ME	DE			RX
7	STF				LH0	LH1	LH2	LH3	LF	CF	AF	SF	MF	DF			RX
9	SRLS	SLLS	STBR	LBR	EXBR	EPSR*	WBR*	RBR*	WHR*	RHR*	WDR*	RDR*	MHUR	SSR*	OCR*	AIR*	SF/RR
B	ECPR	COSR	ATNR	SQDR	ASNR												RR
C	BXH	BXLE	LPSW*	THI	NHI	CLHI	OHI	XHI	LHI	CHI	AHI	SHI	SRHL	SLHL	SRHA	SLHA	RX/RI
D	STM	LM	STB	LB	CLB	AL*	WB*	RB*	WH*	RH*	WD*	RD*	MHU	SS*	OC*	AI*	RX
E		SVC	SINT*	RESB*	SESB*		SRQL	SLQL	SRQA	SLQA	RRL	RLL	SRL	SLL	SRA	SLA	RX/RI
F		COS	ATN	SQD	ASN				STH0	STH1	STH2	STH3					RX

* = PRIVILEGED INSTRUCTION

CONDITION CODE MAP

MSD ↓ / LSD →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		X	X	X	C	E	C	C	A	F	B	B	X	X	B	B
1	T	U	S			F			A	F	B	B	X	X		
2	X	X	X	X	P	Q	B	B	L	M	N	N	N	N	X	X
3									L	M	N	N	N	N		
4	X	X	X	X	C	E	C	C	A	F	B	B	X	X	B	B
5	X	T	U	S		F			A	F	B	B	X	X		
6	X	B			J	J	K	K	L	M	N	N	N	N		
7	X				A	A	A	A	L	M	N	N	N	N		
9	H	H	X	X	X	Y	R	R	Z	Z	Z	Z	X	R	Z	R
B	X	X	W	II	V											
C	X	X	Y	D	C	E	C	C	A	F	B	B	H	H	I	I
D	X	X	X	X	G	R	R	R	Z	Z	Z	Z	X	R	Z	R
E		Y	X	X	X		I	I	I	I	C	C	H	H	I	I
F		X	W	II	V				X	X	X	X				

TO FIND WHAT EFFECT AN INSTRUCTION HAS ON THE CONDITION CODE, USE THE TWO HEX DIGITS OF THE OP CODE TO FIND THE APPROPRIATE ENTRY IN THE MAP. THE LETTER FOUND IN THE MAP CORRESPONDS TO ONE OF THE DESCRIPTIONS ON THE FOLLOWING PAGES.

A.

C	V	G	L	
X	X	0	0	Operand is zero
X	X	0	1	Operand is less than zero
X	X	1	0	Operand is greater than zero

B.

C	V	G	L	
X	X	0	0	Result is zero
X	X	0	1	Result is less than zero
X	X	1	0	Result is greater than zero
X	1	X	X	Arithmetic overflow
1	X	X	X	Carry/borrow

C.

C	V	G	L	
X	X	0	0	Logical result is zero
X	X	0	1	Logical result is not zero
X	X	1	0	

D.

C	V	G	L	
X	X	0	0	None of the bits of the result set
X	X	0	1	Bit 0 of the result set
X	X	1	0	One or more of Bits 1 thru 15 of the result set

E.

C	V	G	L	
X	X	0	0	First operand equal to second operand
X	X	0	1	First operand not equal to second operand
X	X	1	0	
1	X	X	X	First operand less than second operand
0	X	X	X	First operand equal to or greater th second operand

F.

C	V	G	L	
X	X	0	0	First operand equal to second operand
X	X	0	1	First operand less than second operand
X	X	1	0	First operand greater than second operand
1	X	X	X	First operand less than second operand
0	X	X	X	First operand equal to or greater than second operand

G.

C	V	G	L	
0	X	0	0	First operand equals second operand
1	X	0	1	First operand less than second operand
1	X	1	0	
0	X	0	1	First operand greater than second operand
0	X	1	0	

H.

C	V	G	L	
X	0	0	0	Result is zero
X	0	0	1	Result is not zero
X	0	1	0	Result is not zero
0	X	X	X	Last bit that was shifted out was a zero
1	X	X	X	Last bit that was shifted out was a one

I.

C	V	G	L	
X	0	0	0	Result is zero
X	0	0	1	Result is less than zero
X	0	1	0	Result is greater than zero
0	X	X	X	Last bit that was shifted out was a zero
1	X	X	X	Last bit that was shifted out was a one

J.

C	V	G	L	
0	1	0	0	List overflow
0	0	0	0	Element added successfully

K.

C	V	G	L	
0	1	0	0	List was already empty
0	0	0	0	List is now empty
0	0	1	0	List is not yet empty

L.

C	V	G	L	
X	0	0	0	Zero
X	0	0	1	Less than zero
X	0	1	0	Greater than zero
X	1	0	0	Exponent underflow

M.

C	V	G	L	
0	X	0	0	First operand equal to second operand
1	X	0	1	First operand less than second operand
0	X	1	0	First operand greater than second operand

N.

C	V	G	L	
0	0	0	0	Result is zero
0	0	0	1	Result is less than zero
0	0	1	0	Result is greater than zero
0	1	0	1	Overflow (Negative)
0	1	1	0	Overflow (Positive)
0	1	0	0	Underflow
1	1	X	X	* DIVISOR Equal to zero

* = DIVISION ONLY

P.

C	V	G	L	
X	X	0	0	Operand is zero
X	X	1	0	Operand is greater than zero

Q.

C	V	G	L	
X	X	0	0	Operand is zero
X	X	0	1	Operand is less than zero

R.

C	V	G	L	
0	0	0	0	Block data transfer completed correctly
1	X	X	X	Device busy (BSY)
X	1	X	X	Examine status (EX) or time-out
X	X	1	X	End of Medium (EOM)
X	X	X	1	Device unavailable (DU)

S.

C	V	G	L	
0	0	0	0	Result is zero
1	0	0	1	Result is less than zero (operand ≠ X'8000')
1	0	1	0	Result is greater than zero
1	1	0	1	Arithmetic overflow (operand = X'8000')

T.

C	V	G	L	
0	0	0	0	Result is zero (operand = zero)
0	0	0	1	Result is less than zero (operand < zero, operand ≠ X'8000')
0	0	1	0	Result is greater than zero
1	0	0	0	Result is zero (operand = X'8000')

U.

C	V	G	L	
0	0	0	0	Result is zero
0	0	1	0	Result is greater than zero (operand > zero)
1	0	1	0	Result is greater than zero (operand < zero, operand ≠ X'8000')
1	1	0	1	Arithmetic overflow (operand = X'8000')

V. V-FLAG indicates Z = Y = 0 or |Z| < |Y|

W. V-FLAG indicates result is greater than X'7FFF FFFF'

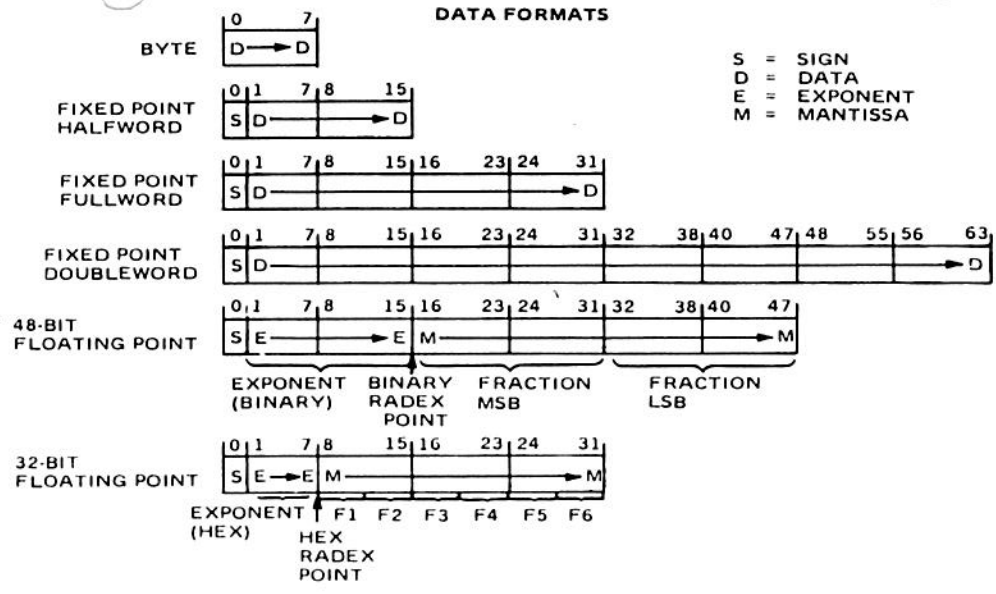
X. FLAGS unchanged

Y. New FLAGS determined by PSW loaded

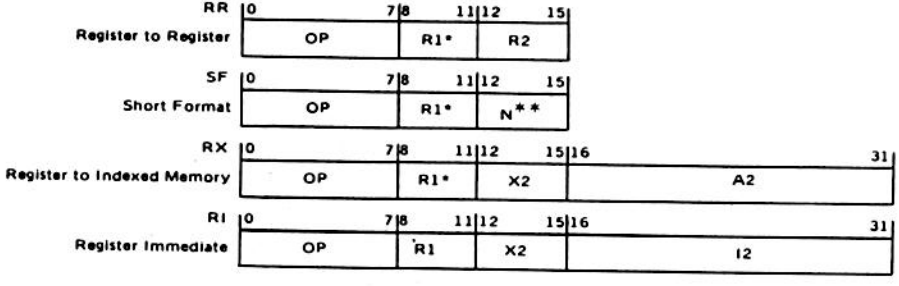
Z. V-FLAG indicates timeout (I/O instruction)

II. V-FLAG indicates |X| < |Y|

DATA FORMATS

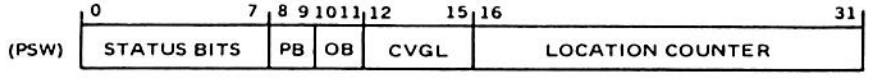


INSTRUCTION FORMATS



A2 = MEMORY ADDRESS IN RX FORMAT
D = FOUR BIT DISPLACEMENT VALUE USED WITH SF BRANCH INSTRUCTIONS
I2 = DATA TO BE USED AS AN IMMEDIATE OPERAND IN THE RI FORMAT
M1 = FOUR BIT MASK SPECIFYING BRANCH ON CONDITION TESTING
N ** = FOUR BIT DATA TO BE USED IN FIXED POINT ARITHMETIC SF INSTRUCTIONS
R1 * = ADDRESS OF 1ST OPERAND AND DEVICE ADDRESS FOR I/O INSTRUCTIONS
R2 = ADDRESS OF 2ND OPERAND
X2 = ADDRESS OF GENERAL REGISTER (1 THROUGH 15) WHOSE CONTENTS ARE USED AS AN INDEX VALUE (X2 = 0 SPECIFIES NO INDEXING)
* = REPLACED BY M1 FOR BRANCH INSTRUCTIONS
** = REPLACED BY D FOR SHORT FORMAT BRANCH INSTRUCTIONS

PROGRAM STATUS WORD



PSW BITS	MEANING
0	WAIT STATE
1	EXTERNAL INTERRUPT ENABLE
2	MACHINE MALFUNCTION INTERRUPT ENABLE
3	FIXED POINT DIVIDE FAULT INTERRUPT ENABLE
4	AUTOMATIC I/O SERVICE ENABLE
5	FLOATING POINT ARITHMETIC FAULT INTERRUPT ENABLE
6	CHANNEL TERMINATION INTERRUPT ENABLE
7	PROTECT MODE ENABLE
8,9	PROGRAM BANK ADDRESS
10,11	OPERAND BANK ADDRESS
12	CARRY/BORROW (C)
13	OVERFLOW (V)
14	GREATER THAN ZERO (G)
15	LESS THAN ZERO (L)
16-31	ADDRESS OF NEXT INSTRUCTION

MEMORY ALLOCATION (48-BIT)

FUNCTION	HEXADECIMAL MEMORY ADDRESS	ASSIGNMENT
48-BIT FLOATING-POINT REGISTERS M = MOST SIGNIFICANT HALF OF FRACTION L = LEAST SIGNIFICANT HALF OF FRACTION SE = SIGN & EXPONENT	00-01	FLOATING-POINT REGISTER, R0 (SE)
	02-03	FLOATING-POINT REGISTER, R2 (SE)
	04-05	FLOATING-POINT REGISTER, R4 (SE)
	06-07	FLOATING-POINT REGISTER, R6 (SE)
	08-09	FLOATING-POINT REGISTER, R8 (SE)
	0A-0B	FLOATING-POINT REGISTER, RA (SE)
	0C-0D	FLOATING-POINT REGISTER, RC (SE)
	0E-0F	FLOATING-POINT REGISTER, RE (SE)
	10-11	FLOATING-POINT REGISTER, R0 (M)
	12-13	FLOATING-POINT REGISTER, R2 (M)
	14-15	FLOATING-POINT REGISTER, R4 (M)
	16-17	FLOATING-POINT REGISTER, R6 (M)
	18-19	FLOATING-POINT REGISTER, R8 (M)
	1A-1B	FLOATING-POINT REGISTER, RA (M)
	1C-1D	FLOATING-POINT REGISTER, RC (M)
	1E-1F	FLOATING-POINT REGISTER, RE (M)
	C0-C1	FLOATING-POINT REGISTER, R0 (L)
	C2-C3	FLOATING-POINT REGISTER, R2 (L)
	C4-C5	FLOATING-POINT REGISTER, R4 (L)
	C6-C7	FLOATING-POINT REGISTER, R6 (L)
	C8-C9	FLOATING-POINT REGISTER, R8 (L)
	CA-CB	FLOATING-POINT REGISTER, RA (L)
	CC-CD	FLOATING-POINT REGISTER, RC (L)
	CE-CF	FLOATING-POINT REGISTER, RE (L)

MEMORY ALLOCATION (32-BIT)

FUNCTION	HEXADECIMAL MEMORY ADDRESS	ASSIGNMENT
32-BIT FLOATING-POINT REGISTERS	00–03	FLOATING–POINT REGISTER, R0
	04–07	FLOATING–POINT REGISTER, R2
	08–0B	FLOATING-POINT REGISTER, R4
	0C–0F	FLOATING–POINT REGISTER, R6
	10–13	FLOATING–POINT REGISTER, R8
	14–17	FLOATING–POINT REGISTER, RA
	18–1B	FLOATING–POINT REGISTER, RC
	1C–1F	FLOATING–POINT REGISTER, RE

MEMORY ALLOCATION (COMMON)

POWER–FAIL LOCATIONS	20–21	RESERVED FOR CORDIC REGISTER
	22–23	REGISTER SAVE POINTER
	24–27	CURRENT PSW SAVE AREA
INTERRUPT PSWs	28–2B	OLD PSW FLPT ARITHMETIC FAULT INTERRUPT
	2C–2F	NEW PSW FLPT ARITHMETIC FAULT INTERRUPT
	30–33	OLD PSW ILLEGAL INSTRUCTION INTERRUPT
	34–37	NEW PSW ILLEGAL INSTRUCTION INTERRUPT
	38–3B	OLD PSW MACHINE MALFUNCTION INTERRUPT
	3C–3F	NEW PSW MACHINE MALFUNCTION INTERRUPT
	40–43	OLD PSW EXTERNAL INTERRUPT
	44–47	NEW PSW EXTERNAL INTERRUPT
	48–4B	OLD PSW FIXED–POINT DIVIDE FAULT INTERRUPT
	4C–4F	NEW PSW FIXED–POINT DIVIDE FAULT INTERRUPT
BOOTSTRAP LOADER	50–7F	50 SEQUENCE (SEE PAGE 27)

MEMORY ALLOCATION (COMMON)

FUNCTION	HEXADECIMAL MEMORY ADDRESS	ASSIGNMENT
CHANNEL I/O TERMINATION PARAMETERS	80–81	TERMINATION QUEUE POINTER
	82–85	OLD PSW CHANNEL I/O TERMINATION INTERRUPT
	86–89	NEW PSW CHANNEL I/O TERMINATION INTERRUPT
	8A–8B	OVERFLOW TERMINATION POINTER
	8C–8F	OLD PSW TERMINATION QUEUE OVERFLOW INTERRUPT
	90–93	NEW PSW TERMINATION QUEUE OVERFLOW INTERRUPT
SUPERVISOR CALL PARAMETERS	94–95	SUPERVISOR CALL ARGUMENT POINTER
	96–99	OLD PSW SUPERVISOR CALL
	9A–9B	NEW PSW (STATUS AND COND. CODE) SUPERVISOR CALL
	9C–9D	NEW PSW (LOC. CNTR) SUPERVISOR CALL 0
	9E–9F	NEW PSW (LOC. CNTR) SUPERVISOR CALL 1
	A0–A1	NEW PSW (LOC. CNTR) SUPERVISOR CALL 2
	A2–A3	NEW PSW (LOC. CNTR) SUPERVISOR CALL 3
	A4–A5	NEW PSW (LOC. CNTR) SUPERVISOR CALL 4
	A6–A7	NEW PSW (LOC. CNTR) SUPERVISOR CALL 5
	A8–A9	NEW PSW (LOC. CNTR) SUPERVISOR CALL 6
	AA–AB	NEW PSW (LOC. CNTR) SUPERVISOR CALL 7
	AC–AD	NEW PSW (LOC. CNTR) SUPERVISOR CALL 8
	AE–AF	NEW PSW (LOC. CNTR) SUPERVISOR CALL 9
	B0–B1	NEW PSW (LOC. CNTR) SUPERVISOR CALL 10
	B2–B3	NEW PSW (LOC. CNTR) SUPERVISOR CALL 11
	B4–B5	NEW PSW (LOC. CNTR) SUPERVISOR CALL 12
	B6–B7	NEW PSW (LOC. CNTR) SUPERVISOR CALL 13
	B8–B9	NEW PSW (LOC. CNTR) SUPERVISOR CALL 14
	BA–BB	NEW PSW (LOC. CNTR) SUPERVISOR CALL 15
	BC–CF	RESERVED
INTERRUPT POINTER TABLE	D0–D1	SERVICE POINTER, DEVICE 0
	D2–D3	SERVICE POINTER, DEVICE 1
	⋮	⋮
	2CE-2CF	SERVICE POINTER, DEVICE 255

50-SEQUENCE LOADER

HEX. MEM. ADDR.	OP CODE/DATA	MNEMONIC	DESCRIPTION
22	0058	SAVPTR	X'0058'
50	0500	AL	AUTO LOAD
52	00CF		X'00CF'
54	4300	BFC	BRANCH IF FALSE
56	0080		TO X'0080'
78	DATA SWITCHES		X'xxyy'

xx = DEVICE ADDRESS
yy = OUTPUT COMMAND BYTE FOR THE DEVICE

MEMORY TEST

TO INITIATE THE MEMORY TEST

1. SET THE MEMORY TEST SWITCH TO ON.
2. SET THE PROGRAM LOAD, BREAK POINT, SINGLE AND RUN SWITCHES TO OFF.
3. DEPRESS THE MASTER CLEAR SWITCH.
4. SET THE FUNCTION SWITCH TO ONE OF THE FOLLOWING POSITIONS:

POSITION	TEST
R2-3	GALPAT TEST
R6-7	ADDRESS TEST — EACH WORD GETS ITS OWN ADDRESS
R8-9	ADDRESS-BAR TEST — EACH WORD GETS THE ONE'S COMPLEMENT OF ITS OWN ADDRESS
RA-B	DATA TEST — EACH WORD GETS THE SAME DATA (USING THE CURRENT VALUE OF D)

(D1 SHALL NOW CONTAIN LAST ADDR. + 1
D2 SHALL NOW CONTAIN 01234)

5. DEPRESS THE EXECUTE SWITCH AND ALLOW THE TEST TO RUN FOR A FEW SECONDS BEFORE PERFORMING STEP 6.
6. SET THE RUN SWITCH TO ON.

(THE TEST WILL NOW STOP IF AN ERROR IS ENCOUNTERED AND WILL LOOP CONTINUOUSLY IF NO ERROR IS ENCOUNTERED.)

DEVICE ADDRESSES

LSD → / MSD ↓	0	1	2	3	4	5	6	7
0		PMP	TTY	PAPER TAPE RDR ONLY	CARD READER	LOADER STORAGE UNIT	HAC CARD READER	DMD
1	← PASLA →			PUNCH ONLY READER/ PUNCH CCM		← μPIC →		
2				← 8 LINE INTERRUPT MCD →				
3	← CCIU NO. 1 →				← CCIU NO. 2 →			
4	PCC	RMM	RMM			FIRST CASSETE SYS.		
5	← UICC NO. 1 →					← UICC NO. 2 →		
6			LINE PRINTER	HAC LINE PRINTER		SECOND CASSETE SYS.		
7							DMAB NO. 2 (DIAG)	DMAB NO. 1 (DIAG)
8					FIRST MAG TAPE	FIRST MAG TAPE	MAGNETIC DRUM	
9	CMUX NO. 1	BIC	ICDD NO. 1	ICDD NO. 1	SECOND MAG TAPE	SECOND MAG TAPE		
A	CMUX NO. 2		ICDD NO. 2	ICDD NO. 2	THIRD MAG TAPE	THIRD MAG TAPE		
B	CMUX NO. 3				FOURTH MAG TAPE	FOURTH MAG TAPE	REMOVABLE CARTRIDGE DISC CONT	HFDD
C	SELCH NO. 1	SELCH NO. 2	SELCH NO. 3	SELCH NO. 4			DISC 0	
D	SELCH NO. 5	SELCH NO. 6	SELCH NO. 7	SELCH NO. 8		-	DISC 1	
E	PADS FIRST ADDRESS						DISC 2	
F	[illegible]						DISC 3	

* NOT USED IN SAME SYSTEM

** THIS BLOCK RESERVED FOR PASLA's THE EVEN ADDRESS IS USED FOR INPUT TO THE CONTROLLER, THE ODD ADDRESS IS USED FOR OUTPUT FROM THE CONTROLL

DEVICE ADDRESSES (CONTINUED)

MSD ↓ / →	8	9	A	B	C	D	E	F
0	AP-1A CH-D	AP-1A CH-C	AP-1A CH-A	AP-1A CH-B	DATA BUS		201/301 DATA SET 1/2 DUPLEX	201/301 DATA SET FULL DUPLEX
1	AP-1A EI-0	AP-1A EI-1	AP-1A EI-2	AP-1A EI-3				
2	← SECOND 8 LINE INTERRUPT MODULE →							
3	← CCIU NO. 3 →				← CCIU NO. 4 →			
4				DIGITAL MUX				
5							103/202 DATA SET 1/2 DUPLEX	103/202 DATA SET FULL DUPLEX
6					UNIVERSAL CLOCK VARIABLE 60 Hz			
7	← RS-232 RAYMOND NO. 1 TGPSG NO. 1 →		← RS-232 RAYMOND NO. 2 TGPSG NO. 2 →		← RS-232 RAYMOND NO. 3 TGPSG NO. 3 →		← RS-232 RAYMOND NO. 4 TGPSG NO. 4 →	
8	DMAB NO. 1 (NORM)	DMAB NO. 2 (NORM)		UNIVERSAL I/O MOD				
9		FIXED HEAD DISC 1	FIXED HEAD DISC 2			FIXED HEAD DISC 3	FIXED HEAD DISC 4	
A	← ASR733 PASLA →						MEMORY PROTECT CONTROL	
B	← MIED PASLA →							
C					← HP2100 CH M →			
D	**				← HP2100 CH X →			
E					← HP2100 CH I →			
F		MPC			IGS/ TCG			

PERIPHERAL DEVICE STATUS AND COMMAND BYTES

DEVICE	ADDRESS	8 / 0	9 / 1	10 / 2	11 / 3	12 / 4	13 / 5	14 / 6	15 / 7	
DISPLAY PANEL		← MODE →				← 12-POSITION FUNCTION SWITCH CODE →				Status Bits / Command Bits
	X'01'	NORM	INC							Status
										Command
TELETYPE	X'02'			BRK		BSY	EX		DU	Status
		DIS	ENB	UNB	BLK	WR	RD	PON	POFF	Command
CARD READER	X'04'	EOV	TBL	HE	NMTN	BSY	EX	EOM	DU	Status
		DIS	ENB	FEED						Command
AP-1A	X'08 – X'0B OR X'18' – X1B					RWC	IOA	UI	PE	Status
						OLC	OLC	CC	CC	Command
HIGH SPEED RDR/PNCH	X'13'	OV			NMTN	HSY	EX		DU	Status
		DIS	ENB	STOP	RUN	INCR	SLEW	WR	RD	Command
PCC	X'40'							ETCT	RTCO	Status
		ARTC	DRTC	LRTC	RRTC	AETC	DETC	LETC		Command
RMM	X'41'								BSY	Status
	X'42'	OP CODE							MSB	Command
CASSET[illegible]	X'45'	ERR		EOT	NMTN	BSY	EX	EOR	DU	S[illegible]
	X'55'	DIS	ENB				← OP CODE →			[illegible]nd

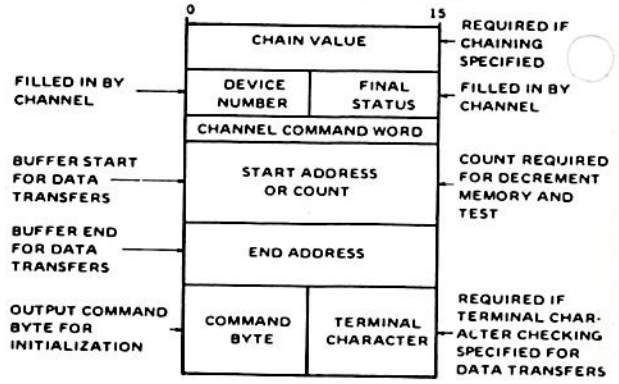
LINE PRINTER	X'62'			PAPER OUT		INTER-LOCK	BUSY	EX		DU	Status
		DIS		ENB							Command
MAG TAPE	X'85' X'95'	ERR		EOF	EOT	NMTN	BSY	EX	EOM	DU	Status
	X'A5' X'B5'	DIS		ENB	FWD	BKSP			WR	RD	Command
SELECTOR CHANNEL	X'F0'						BSY				Status
					READ	GO	STOP				Command
MPC	X'F9'								DMAV	PV	Status
					EPF	DPF		LMPV	DMA		Command
PASLA	X'10' X'11'		OV	PF OR CL25	FR ERR	RCR	BSY	EX*	CARR OFF	RING	Status
		RCV	DIS	EN	DTR	ECHO-PLEX	RCT OR DTB	TRANS LB	WRT OR RD	1	Command 1
		SND	DIS	EN							
				CLK	BIT SEL		STOP BIT	PARITY		0	Command 2

*EX = OV + PF + $\overline{\text{DATA SET READY}}$ + FR ERR

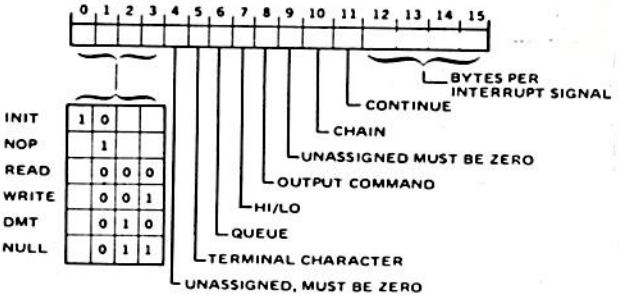
PLA	X'78/79' TO X'7E/7F'	0	CTS	0	0	TBSY	DSRY	0	0	Transmit Status
		OR	PE	FE	0	RBSY	EX*	CARR	0	Receive Status
		DIS	EN	DTR	EPLEX	0	TLB	WRT	1	Command 1
		0	0	BIT SEL		STOP BIT	PARITY		0	Command 2

*EX = OR + PE + FE + $\overline{\text{DSRY}}$

CHANNEL CONTROL BLOCK



CHANNEL COMMAND WORD



HEXADECIMAL TO DECIMAL CONVERSION TABLE

BYTE 0				BYTE 1				BYTE 2			
F1		F2		F3		F4		F5		F6	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15

POWERS OF 2 AND 16

POWERS OF 16		POWERS OF 2	
16^n	n	2^n	n
1	0	512	9
16	1	1 024	10
256	2	2 048	11
4 096	3	4 096	12
65 536	4	8 192	13
1 048 576	5	16 384	14
16 777 216	6	32 768	15
268 435 456	7	65 536	16
4 294 967 296	8	131 072	17
68 719 476 736	9	262 144	18
1 099 511 627 776	10	524 288	19
17 592 186 044 416	11	1 048 576	20
281 474 976 710 656	12	2 097 152	21
4 403 599 627 370 496	13	4 194 304	22
72 057 594 037 927 936	14	8 388 608	23
1 152 921 504 606 846 976	15	16 777 216	24

NOTES

NOTES

MEMORY ALLOCATION (SBI)

FUNCTION	HEXADECIMAL MEMORY ADDRESS
48-BIT FLOATING-POINT REGISTERS	00-1F
POWER-FAIL LOCATIONS	20-27
INTERRUPT PSWs	28-4F
BOOTSTRAP LOADER (50 SEQUENCE)	50-7F
CHANNEL I/O TERMINATION PARAMETERS	80-93
SUPERVISOR CALL PARAMETERS	94-BB
RESERVED	BC-BF
48-BIT FLOATING-POINT REGISTERS	C0-CF
INTERRUPT POINTER TABLE	D0-2CF
UNUSED	2D0-2FF
READ-MOSTLY MEMORY LOAD	300-6FB
UNUSED	6FC-9FF
INPUT TERMINATION QUEUE	A00-BFF
OUTPUT TERMINATION QUEUE	C00-DFF
LOST MESSAGE QUEUE	E00-FFF
INPUT FUNCTION CODE ACTIVATION TABLE	1000-2FFF*

* HIGHEST LOCATION DEPENDS ON THE NUMBER OF ACTIVE FUNCTION CODES. LOCATION 2FFF CORRESPONDS TO ALL 4,096 FUNCTION CODES BEING USED

EXTENDED BRANCH MNEMONICS
COMMON ASSEMBLY LANGUAGE (CAL)

INSTRUCTION (FORMAT)	OP-CODE AND M1 MASK	MNEMONIC
BRANCH ON CARRY RX	428	BC
BRANCH ON CARRY RR	028	BCR
BRANCH ON CARRY SHORT BKWD/FWD	208/218	BCS
BRANCH ON NO CARRY RX	438	BNC
BRANCH ON NO CARRY RR	038	BNCR
BRANCH ON NO CARRY SHORT BKWD/FWD	228/238	BNCS
BRANCH ON EQUAL RX	433	BE
BRANCH ON EQUAL RR	033	BER
BRANCH ON EQUAL SHORT BKWD/FWD	223/233	BES
BRANCH ON NOT EQUAL RX	423	BNE
BRANCH ON NOT EQUAL RR	023	BNER
BRANCH ON NOT EQUAL SHORT BKWD/FWD	203/213	BNES
BRANCH ON LOW RX	428	BL
BRANCH ON LOW RR	028	BLR
BRANCH ON LOW SHORT BKWD/FWD	208/218	BLS
BRANCH ON NOT LOW RX	438	BNL
BRANCH ON NOT LOW RR	038	BNLR
BRANCH ON NOT LOW SHORT BKWD/FWD	228/238	BNLS
BRANCH ON MINUS RX	421	BM
BRANCH ON MINUS RR	021	BMR
BRANCH ON MINUS SHORT BKWD/FWD	201/211	BMS
BRANCH ON NOT MINUS RX	431	BNM
BRANCH ON NOT MINUS RR	031	BNMR
BRANCH ON NOT MINUS SHORT BKWD/FWD	221/231	BNMS
BRANCH ON PLUS RX	422	BP
BRANCH ON PLUS RR	022	BPR
BRANCH ON PLUS SHORT BKWD/FWD	202/212	BPS
BRANCH ON NOT PLUS RX	432	BNP
BRANCH ON NOT PLUS RR	032	BNPR
BRANCH ON NOT PLUS SHORT BKWD/FWD	222/232	BNPS
BRANCH ON OVERFLOW RX	424	BO
BRANCH ON OVERFLOW RR	024	BOR
BRANCH ON OVERFLOW SHORT BKWD/FWD	204/214	BOS
BRANCH ON NO OVERFLOW RX	434	BNO
BRANCH ON NO OVERFLOW RR	034	BNOR
BRANCH ON NO OVERFLOW SHORT BKWD/FWD	224/234	BNOS
BRANCH UNCONDITIONAL RX	430	B
BRANCH UNCONDITIONAL RR	030	BR
BRANCH UNCONDITIONAL SHORT BKWD/FWD	220/230	BS
BRANCH ON ZERO RX	433	BZ
BRANCH ON ZERO RR	033	BZR
BRANCH ON ZERO SHORT BKWD/FWD	223/233	BZS
BRANCH ON NOT ZERO RX	423	BNZ
BRANCH ON NOT ZERO RR	023	BNZR
BRANCH ON NOT ZERO SHORT BKWD/FWD	203/213	BNZS
NO OPERATION RX	420	NOP
NO OPERATION RR	020	NOPR