16-BIT INSTRUCTION REPERTOIRE

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO SEC	C 9	C 8
				LOAD INSTRUCTIONS								
				LOAD								
01	0			Load	LR a, m	RR	(Rm) → Ra	0	0	.84	X	X
01	1			Load	LI a, m	RI	(Y*) → Ra	0	0	1.56	X	X
01	2			Load	LK a, y, m	RK	Y → Ra	0	0	1.66	X	X
01	3			Load	L a, y, m	RX	(Y) → Ra	0	0	2.35	X	X
63	0			Load	LL a, m	RL	m → Ra	0	0	1.10	0	X
				LOAD DOUBLE								
02	1			Load Double	LDI a, m	RI	(Y*) → Ra, (Y* + 1) → Ra + 1	0	0	2.35	X	X
02	3			Load Double	LD a, y, m	RX	(Y, Y + 1) → Ra, Ra + 1	0	0	3.05	X	X
				LOAD MULTIPLE								
03	3			Load Multiple	LM a, y, m	RX	(Y...Y + m - a) → Ra...Rm	NC	NC	2.35 + .85(n)	NC	NC
				LOAD SPECIAL								
03	0		4	Load P Register	LPR a	RR	(Ra) → P	NC	NC	1.30	NC	NC
03	0		5	Load Status Register 1	LSOR a	RR	(Ra) → SR1	NC	NC	1.60	NC	NC
03	0		6	Load Status Register 2	LSTR a	RR	(Ra) → SR2	NC	NC	1.00	NC	NC
03	0		7	Load Real-Time Clock	LCR a	RR	(Ra) → R/T Clock Register	NC	NC	1.00	NC	NC
03	0		12	Load and Enable Monitor Clock	LEM a	RR	(Ra) → Monitor Clock	NC	NC	1.46	NC	NC
03	0		14	Load Real-Time Clock Double	LCRD a	RR	(Ra, Ra + 1) → R/T Clock Register	NC	NC	1.60	NC	NC
05	1			Load and Index By 1	LXI a, m	RI	(Y*) → Ra; (Rm) + 1 → Rm	0	0	1.60	X	X
05	3			Load and Index By 1	LX a, y, m	RX	(Y) → Ra; (Rm) + 1 → Rm	0	0	2.45	X	X
06	1			Load Double and Index By 2	LDXI a, m	RI	(Y*, Y* + 1) → Ra, Ra + 1; (Rm) + 2 → Rm	0	0	2.55	X	X
06	3			Load Double and Index By 2	LDX a, y, m	RX	(Y, Y+1) → Ra, Ra + 1 (Rm) + 2 → Rm	0	0	3.4	X	X
07	1	0		Load PSW	LPI m	RI	(Y*, Y* + 1, Y* + 2) → P, SR1, SR2	NC	NC	3.20	NC	NC
07	3	0		Load PSW	LP y, m	RX	(Y, Y + 1, Y + 2) → P, SR1, SR2	NC	NC	4.05	NC	NC
54	0			Load Address Register	LARR a, m	RR	(Rm) → AR	NC	NC	1.46	NC	NC
54	1			Load Address Register	LARI a, m	RI	(Y*) → ARr	NC	NC	1.65	NC	NC
54	3			Load Address Register Multiple	LARM a, y, m	RX	(Y,..., Y + u) → ARr...AR1 + u; r = $(Ra)_{5-0}$; u = $(Ra)_{13-8}$ r = Word Designator, u = Count	NC	NC	3.4 + 1.1(n)	NC	NC
				LOAD-BYTE								
00	3			Byte Load	BL a, y, m	RX	(Y) byte → Ra	0	0	2.35	0	X
04	3			Byte Load and Index	BLX a, y, m	RX	(Y) byte → Ra; (Rm) + 1 → Rm,	0	0	2.45	0	X
				STORE INSTRUCTIONS								
				STORE								
11	1			Store	SI a, m	RI	(Ra) → Y*	NC	NC	.95	NC	NC
11	3			Store	S a, y, m	RX	(Ra) → Y	NC	NC	2.45	NC	NC
				STORE DOUBLE								
12	1			Store Double	SDI a, m	RI	(Ra, Ra + 1) → Y*, Y* + 1	NC	NC	2.35	NC	NC
12	3			Store Double	SD a, y, m	RX	(Ra, Ra + 1) → Y, Y + 1	NC	NC	3.15	NC	NC
				STORE MULTIPLE								
13	3			Store Multiple	SM a, y, m	RX	(Ra,...Rm) → Y,...,Y + m - a	NC	NC*	2.4 + .95(n)	NC	NC
				SPECIAL STORE								
03	0		1	Store Status Register 1	SSOR a	RR	(SR1) → Ra	0	0	1.00	X	X
03	0		2	Store Status Register 2	SSTR a	RR	(SR2) → Ra	0	0	1.00	X	X
03	0		3	Store Real-Time Clock	SCR a	RR	(R/T Clock Register) → Ra	0	0	1.00	X	X
03	0		15	Store Real-Time Clock Double	SCRD a	RR	(R/T Clock Register) → Ra, Ra + 1	NC	NC	1.60	NC	NC

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO-SEC	C 9	C 8
15	1			Store and Index By 1	SXI a, m	RI	(Ra) → Y*; (Rm) + 1 → Rm	NC	NC	1.64	NC	NC
15	3			Store and Index By 1	SX a, y, m	RX	(Ra) → Y; (Rm) + 1 → Rm	NC	NC	2.35	NC	NC
16	1			Store Double and Index By 2	SDXI a, m	RI	(Ra, Ra + 1) → Y; Y* + 1; (Rm) + 2 → Rm	NC	NC	2.35	NC	NC
16	3			Store Double and Index By 2	SDX a, y, m	RX	(Ra, Ra + 1) → Y, Y + 1; (Rm) + 2 → Rm	NC	NC	3.15	NC	NC
17	1	0		Store Zeros	SZI m	RI	Ø → Y*	NC	NC	1.60	NC	NC
17	3	0		Store Zeros	SZ y, m	RX	Ø → Y	NC	NC	2.20	NC	NC
55	0			Store Address Register	SARR a, m	RR	(ARr) → Rm	NC	NC	1.79	NC	NC
55	1			Store Address Register	SARI a, m	RI	(ARr) → Y	NC	NC	2.50	NC	NC
55	3			Store Address Register Multiple	SARM a, y, m	RX	(ARr, ..., ARr + (u-1) → Y, ..., Y + (u-1) (ARr, ... AR1 + u) → Y, ... Y + u; r = $(Ra)_{5-0}$; u = $(Ra)_{13-8}$ r = Word Designator, u = Count	NC	NC	3.7 + 1.5(n)	NC	NC
				BYTE-STORE								
10	3			Byte-Store	BS a, y, m	RX	(Ra) bits 0–7 → Y byte	NC	NC	2.40	NC	NC
14	3			Byte-Store and Index By 1	BSX a, y, m	RX	(Ra) bits 0–7 → Y byte; (Rm) + 1 → Rm	NC	NC	2.25	NC	NC
				UNCONDITIONAL JUMPS								
				JUMP								
40	0	10		Jump	JR m	RR	(Rm) → P	NC	NC	1.14	NC	NC
40	2	10		Jump	J y, m	RK	Y → P	NC	NC	1.95	NC	NC
40	3	10		Jump	J *y, m	RX	(Y) → P	NC	NC	2.40	NC	NC
				JUMP AFTER STOP								
40	0	11		Jump After Stop	JSR m	RR	Stop; Upon Restart, (Rm) → P	NC	NC	1.1	NC	NC
40	2	11		Jump After Stop	JS y, m	RK	Stop; Upon Restart, Y → P	NC	NC	1.7	NC	NC
40	3	11		Jump After Stop	JS *y, m	RX	Stop; Upon Restart, (Y) → P	NC	NC	2.4	NC	NC
				JUMP, LINK REGISTER								
42	0			Jump, Link Register	JLRR a, m	RR	(P) + 1 → Ra; (Rm) → P	NC	NC	1.28	NC	NC
42	2			Jump, Link Register	JLR a, y, m	RK	(P) + 2 → Ra; Y → P	NC	NC	2.15	NC	NC
42	3			Jump, Link Register	JLR a,*y, m	RX	(P) + 2 → Ra; (Y) → P	NC	NC	2.35	NC	NC
				JUMP, LINK MEMORY								
43	2	0		Jump, Link Memory	JLM y, m	RK	(P) + 2 → Y; Y + 1 → P	NC	NC	2.40	NC	NC
43	3	0		Jump, Link Memory	JLM *y, m	RX	(P) + 2 → (Y); (Y) + 1 → P	NC	NC	3.05	NC	NC
				CONDITIONAL JUMPS								
				COMPARE CONDITIONS								
40	0	0		Jump Equal	JER m	RR	If (CC) is =, (Rm) → P	NC	NC	1.14	NC	NC
40	2	0		Jump Equal	JE y, m	RK	If (CC) is =, Y → P	NC	NC	1.95	NC	NC
40	3	0		Jump Equal	JE *y, m	RX	If (CC) is =, (Y) → P	NC	NC	2.40	NC	NC
40	0	1		Jump Not Equal	JNER m	RR	If (CC) is ≠, (Rm) → P	NC	NC	1.14	NC	NC
40	2	1		Jump Not Equal	JNE y, m	RK	If (CC) is ≠, Y → P	NC	NC	1.95	NC	NC
40	3	1		Jump Not Equal	JNE *y, m	RX	If (CC) is ≠, (Y) → P	NC	NC	2.40	NC	NC
40	0	3		Jump Less	JLSR m	RR	If (CC) is <, (Rm) → P	NC	NC	1.14	NC	NC
40	2	3		Jump Less	JLS y, m	RK	If (CC) is <, Y → P	NC	NC	1.95	NC	NC
40	3	3		Jump Less	JLS *y, m	RX	If (CC) is <, (Y) → P	NC	NC	2.40	NC	NC
40	0	2		Jump Greater or Equal	JGER m	RR	If (CC) is ≥, (Rm) → P	NC	NC	1.14	NC	NC
40	2	2		Jump Greater or Equal	JGE y, m	RK	If (CC) is ≥, Y → P	NC	NC	1.95	NC	NC
40	3	2		Jump Greater or Equal	JGE *y, m	RX	If (CC) is ≥, (Y) → P	NC	NC	2.40	NC	NC
				MANUAL JUMPS								
40	0	12		Jump Key Set After Stop	JKSR I, m	RR	If Key 1Set, Stop; (Rm) → P	NC	NC	1.1	NC	NC
40	2	12		Jump Key Set After Stop	JKS I, y, m	RK	If Key 1 Set, Stop; Y → P	NC	NC	1.7	NC	NC
40	3	12		Jump Key Set After Stop	JKS I, *y, m	RX	If Key 1 Set, Stop; (Y) → P	NC	NC	2.4	NC	NC
40	0	13		Jump Key Set After Stop	JKSR 2, m	RR	If Key 2 Set, Stop; (Rm) → P	NC	NC	1.1	NC	NC
40	2	13		Jump Key Set After Stop	JKS 2, y, m	RK	If Key 2 Set, Stop; Y → P	NC	NC	1.7	NC	NC

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO-SEC	C 9	C 8
40	3	13		Jump Key Set After Stop	JKS 2, *y, m	RX	If Key 2 Set, Stop; (Y) → P	NC	NC	2.4	NC	NC
40	0	7		Jump Bootstrap 2 Selected	JBR m	RR	If Bootstrap 2 Selected, (Rm) → P	NC	NC	1.14	NC	NC
40	2	7		Jump Bootstrap 2 Selected	JB y, m	RK	If Bootstrap 2 Selected, Y → P	NC	NC	1.95	NC	NC
40	3	7		Jump Bootstrap 2 Selected	JB *y, m	RX	If Bootstrap 2 Selected, (Y) → P	NC	NC	2.40	NC	NC
				REGISTER CONDITIONS								
44	0			Jump Zero	JZR a, m	RR	If (Ra) = 0, (Rm) → P	NC	NC	1.48	NC	NC
44	2			Jump Zero	JZ a, y, m	RK	If (Ra) = 0, Y → P	NC	NC	2.15	NC	NC
44	3			Jump Zero	JZ a, *y, m	RX	If (Ra) = 0, (Y) → P	NC	NC	2.35	NC	NC
45	0			Jump Not Zero	JNZR a, m	RR	If (Ra) ≠ 0, (Rm) → P	NC	NC	1.48	NC	NC
45	2			Jump Not Zero	JNZ a, y, m	RK	If (Ra) ≠ 0, Y → P	NC	NC	2.15	NC	NC
45	3			Jump Not Zero	JNZ a, *y, m	RX	If (Ra) ≠ 0, (Y) → P	NC	NC	2.35	NC	NC
46	0			Jump Positive	JPR a, m	RR	If (Ra) ≥ 0, (Rm) → P	NC	NC	1.48	NC	NC
46	2			Jump Positive	JP a, y, m	RK	If (Ra) ≥ 0, Y → P	NC	NC	2.15	NC	NC
46	3			Jump Positive	JP a, *y, m	RX	If (Ra) ≥ 0, (Y) → P	NC	NC	2.35	NC	NC
47	0			Jump Negative	JNR a, m	RR	If (Ra) < 0, (Rm) → P	NC	NC	1.48	NC	NC
47	2			Jump Negative	JN a, y, m	RK	If (Ra) < 0, Y → P	NC	NC	2.15	NC	NC
47	3			Jump Negative	JN a, *y, m	RX	If (Ra) < 0, (Y) → P	NC	NC	2.35	NC	NC
				INDEX JUMP								
41	0			Index Jump	XJR a, m	RR	If (Ra) ≠ 0, (Ra) - 1 → Ra, (Rm) → P	NC	NC	1.46	NC	NC
41	2			Index Jump	XJ a, y, m	RK	If (Ra) ≠ 0, (Ra) - 1 → Ra, Y → P	NC	NC	2.15	NC	NC
41	3			Index Jump	XJ a, *y, m	RX	If (Ra) ≠ 0, (Ra) - 1 → Ra, (Y) → P	NC	NC	2.50	NC	NC
				SPECIAL CONDITIONS								
40	0	4		Jump Overflow	JOR m	RR	If Overflow Set, (Rm) → P	NC	NC	1.14	NC	NC
40	2	4		Jump Overflow	JO y, m	RK	If Overflow Set, Y → P	NC	NC	1.95	NC	NC
40	3	4		Jump Overflow	JO *y, m	RX	If Overflow Set, (Y) → P	NC	NC	2.40	NC	NC
40	0	5		Jump Carry	JCR m	RR	If Carry Set, (Rm) → P	NC	NC	1.14	NC	NC
40	2	5		Jump Carry	JC y, m	RK	If Carry Set, Y → P	NC	NC	1.95	NC	NC
40	3	5		Jump Carry	JC *y, m	RX	If Carry Set, (Y) → P	NC	NC	2.40	NC	NC
40	0	6		Jump Power Out of Tolerance	JPTR m	RR	If Power Out of Tolerance (Rm) → P	NC	NC	1.14	NC	NC
40	2	6		Jump Power Out of Tolerance	JPT y, m	RK	If Power Out of Tolerance Y → P	NC	NC	1.95	NC	NC
40	3	6		Jump Power Out of Tolerance	JPT *y, m	RX	If Power Out of Tolerance (Y) → P	NC	NC	2.40	NC	NC
				LOCAL JUMPS								
41	1			Local Jump Indirect	LJI xD	#RI	((P) + xD) → P	NC	NC	2.20	NC	NC
40	1			Local Jump	LJ xD	#RI	(P) + xD → P	NC	NC	1.32	NC	NC
43	1			Local Jump, Link Memory	LJLM xD	#RI	(P) + 1 → P + xD; (P) + xD + 1 → P	NC	NC	2.15	NC	NC
44	1			Local Jump Equal	LJE xD	#RI	If (CC) is =, (P) + xD → P	NC	NC	1.32	NC	NC
45	1			Local Jump Not Equal	LJNE xD	#RI	If (CC) is ≠, (P) + xD → P	NC	NC	1.32	NC	NC
47	1			Local Jump Less	LJLS xD	#RI	If (CC) is <, (P) + xD → P	NC	NC	1.32	NC	NC
46	1			Local Jump Greater or Equal	LJGE xD	#RI	If (CC) is ≥, (P) + xD → P	NC	NC	1.32	NC	NC
				ARITHMETIC INSTRUCTIONS								
				ARITHMETIC SINGLE LENGTH								
20	0			Subtract	SUR a, m	RR	(Ra) - (Rm) → Ra	X	X	.84	X	X
20	1			Subtract	SUI a, m	RI	(Ra) - (Y*) → Ra	X	X	1.60	X	X
20	2			Subtract	SUK a, y, m	RK	(Ra) - Y → Ra	X	X	1.68	X	X
20	3			Subtract	SU a, y, m	RX	(Ra) - (Y) → Ra	X	X	2.35	X	X
62	0			Subtract	LSU a, m	RL	(Ra) - m → Ra	X	X	1.63	X	X
22	0			Add	AR a, m	RR	(Ra) + (Rm) → Ra	X	X	.84	X	X
22	1			Add	AI a, m	RI	(Ra) + (Y*) → Ra	X	X	1.60	X	X
22	2			Add	AK a, y, m	RK	(Ra) + Y → Ra	X	X	1.68	X	X
22	3			Add	A a, y, m	RX	(Ra) + (Y) → Ra	X	X	2.30	X	X
62	2			Add	LA a, m	RL	(Ra) + m → Ra	X	X	1.40	X	X

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO SEC	C 9	C 8
26	0			Multiply	MR a, m	RR	(Ra + 1)* (Rm) → Ra, Ra + 1	0	0	3.55	X	X
26	1			Multiply	MI a, m	RI	(Ra + 1)* (Y*) → Ra, Ra + 1	0	0	4.30	X	X
26	2			Multiply	MK a, y, m	RK	(Ra + 1)* Y → Ra, Ra + 1	0	0	4.45	X	X
26	3			Multiply	M a, y, m	RX	(Ra + 1)* (Y) → Ra, Ra + 1	0	0	5.10	X	X
63	2			Multiply	LMUL a, m	RL	(Ra + 1)* m → Ra, Ra + 1	0	0	4.20	X	X
27	0			Divide	DR a, m	RR	(Ra, Ra + 1)/(Rm) → Ra + 1; Remainder → Ra	0	X	6.80	X	X
27	1			Divide	DI a, m	RI	(Ra, Ra + 1)/(Y*) → Ra + 1; Remainder → Ra	0	X	7.20	X	X
27	2			Divide	DK a, y, m	RK	(Ra, Ra + 1)/ Y → Ra + 1; Remainder → Ra	0	X	7.40	X	X
27	3			Divide	D a, y, m	RX	(Ra, Ra + 1)/(Y) → Ra + 1; Remainder → Ra	0	X	7.95	X	X
63	3			Divide	LDIV a, m	RL	(Ra, Ra + 1)/m → Ra + 1; Remainder → Ra	0	X	7.40	X	X
				ARITHMETIC DOUBLE								
21	0			Subtract Double	SUDR a, m	RR	(Ra, Ra + 1) - (Rm, Rm + 1) → Ra, Ra + 1	X	X	1.74	X	X
21	1			Subtract Double	SUDI a, m	RI	(Ra, Ra + 1) - (Y*, Y* + 1) → Ra, Ra + 1	X	X	2.45	X	X
21	3			Subtract Double	SUD a, y, m	RX	(Ra, Ra + 1) - (Y, Y + 1) → Ra, Ra + 1	X	X	3.20	X	X
62	1			Subtract Double	LSUD a, m	RL	(Ra, Ra + 1) - m → Ra, Ra + 1	X	X	2.15	X	X
23	0			Add Double	ADR a, m	RR	(Ra, Ra + 1) + (Rm, Rm + 1) → Ra, Ra + 1	X	X	1.58	X	X
23	1			Add Double	ADI a, m	RI	(Ra, Ra + 1) + (Y*, Y* + 1) → Ra, Ra + 1	X	X	2.45	X	X
23	3			Add Double	AD a, y, m	RX	(Ra, Ra + 1) + (Y, Y + 1); → Ra, Ra + 1	X	X	3.15	X	X
62	3			Add Double	LAD a, m	RL	(Ra, Ra + 1) + m → Ra, Ra + 1	X	X	2.15	X	X
				BYTE-ARITHMETIC								
64	3			Byte Subtract	BSU a, y, m	RX	(Ra) - (Y)Byte → Ra	X	X	2.35	X	X
65	3			Byte Add	BA a, y, m	RX	(Ra) + (Y) Byte → Ra	X	X	2.35	X	X
				COUNTERS								
02	0		10	Increment Ra by 1	IROR a	RR	(Ra) + 1 → Ra	X	X	.84	X	X
02	0		11	Decrement Ra by 1	DROR a	RR	(Ra) - 1 → Ra	X	X	.84	X	X
02	0		12	Increment Ra by 2	IRTR a	RR	(Ra) + 2 → Ra	X	X	1.10	X	X
02	0		13	Decrement Ra by 2	DRTR a	RR	(Ra) - 2 → Ra	X	X	1.10	X	X
				SPECIAL ARITHMETIC								
02	0		2	Round Ra	RR a	RR	If (Ra) ≥ 0, (Ra) + (Ra + 1) 15 → Ra: If (Ra) < 0, (Ra) - (Ra + 1) 15 → Ra	X	X	1.56	X	X
04	0		1	Reverse Register	RVR a	RR	Reverse Order of Bits in Ra	0	0	6.50	K	X
04	0		2	Count Ones	CNT a	RR	Number of Binary 1's in Ra → Ra + 1	NC	NC	7.0	NC	NC
04	0		3	Scale Factor	SFR a	RR	Shift (Ra, Ra + 1) Left Until (Ra) 15 ≠ (Ra) 14; Shift Count → Ra + 2	NC	NC	3.20	NC	NC

LOGICAL INSTRUCTIONS

				AND								
30	0			AND	ANDR a, m	RR	(Ra) ⊙ (Rm) → Ra	0	0	.84	X	X
30	1			AND	ANDI a, m	RI	(Ra) ⊙ (Y*) → Ra	0	0	1.58	X	X
30	2			AND	ANDK a, y, m	RK	(Ra) ⊙ Y → Ra	0	0	1.68	X	X
30	3			AND	AND a, y, m	RX	(Ra) ⊙ (Y) → Ra	0	0	2.35	X	X
				OR								
31	0			OR	ORR a, m	RR	(Ra) ⊛ (Rm)→ Ra	0	0	.84	X	X
31	1			OR	ORI a, m	RI	(Ra) ⊛ (Y*) → Ra	0	0	1.58	X	X
31	2			OR	ORK a, y, m	RK	(Ra) ⊛ Y → Ra	0	0	1.83	X	X
31	3			OR	OR a, y, m	RX	(Ra) ⊛ (Y) → Ra	0	0	2.25	X	X

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO SEC	C 9	C 8
				EXCLUSIVE OR								
32	0			Exclusive OR	XORR a, m	RR	(Ra) ⊕ (Rm) → Ra	0	0	.84	X	X
32	1			Exclusive OR	XORI a, m	RI	(Ra) ⊕ (Y*) → Ra	0	0	1.58	X	X
32	2			Exclusive OR	XORK a, y, m	RK	(Ra) ⊕ Y → Ra	0	0	1.68	X	X
32	3			Exclusive OR	XOR a, y, m	RX	(Ra) ⊕ (Y) → Ra	0	0	2.35	X	X
				MASKED SUBSTITUTE								
33	0			Masked Substitute	MSR a, m	RR	(Rm) bits 0-15 → Ra bits 0-15 for (Ra + 1) bits 0-15 = 1	0	0	1.42	X	X
33	1			Masked Substitute	MSI a, m	RI	(Y*) bits 0-15 → Ra bits 0-15 for (Ra + 1) bits 0-15 = 1	0	0	1.60	X	X
33	2			Masked Substitute	MSK a, y, m	RK	Y bits 0-15 → Ra bits 0-15 for (Ra + 1) bits 0-15 = 1	0	0	2.20	X	X
33	3			Masked Substitute	MS a, y, m	RX	(Y)bits 0-15 → Ra bits 0-15 for (Ra + 1) bits 0-15 = 1	0	0	2.45	X	X
				COMPARE MASKED								
34	0			Compare Masked	CMR a, m	RR	|(Ra) ⊙ (Ra + 1)|: |(Rm) ⊙ (Ra + 1)|	0	0	1.56	X	X
34	1			Compare Masked	CMI a, m	RI	|(Ra) ⊙ (Ra + 1)|: |(Y*) ⊙ (Ra + 1)|	0	0	1.78	X	X
34	2			Compare Masked	CMK a, y, m	RK	|(Ra) ⊙ (Ra + 1)|: |Y⊙ (Ra + 1)|	0	0	2.40	X	X
34	3			Compare Masked	CM a, y, m	RX	|(Ra) ⊙ (Ra + 1)|: |(Y) ⊙ (Ra + 1)|	0	0	2.60	X	X
				BYTE-OPERATIONS								
66	3			Byte-Compare	BC a, y, m	RX	(Ra): (Y) Byte	X	X	2.40	X	X
67	3			Byte-Compare and Index by 1	BCX a, y, m	RX	(Ra): (Y) Byte; (Rm) + 1 → Rm	X	X	2.60	X	X
				COMPARE INSTRUCTIONS								
24	0			Compare	CR a, m	RR	(Ra):(Rm)	X	X	.94	X	X
24	1			Compare	CI a, m	RI	(Ra):(Y*)	X	X	1.66	X	X
24	2			Compare	CK a, y, m	RK	(Ra):Y	X	X	1.75	X	X
24	3			Compare	C a, y, m	RX	(Ra):(Y)	X	X	2.35	X	X
63	1			Compare	LC a, m	RL	(Ra):m	X	X	1.56	X	X
25	0			Compare Double	CDR a, m	RR	(Ra, Ra + 1):(Rm, Rm + 1)	X	X	1.70	X	X
25	1			Compare Double	CDI a, m	RI	(Ra, Ra + 1):(Y*, Y* + 1)	X	X	2.45	X	X
25	3			Compare Double	CD a, y, m	RX	(Ra, Ra + 1):(Y, Y + 1)	X	X	3.20	X	X
				MISCELLANEOUS LOGICAL								
02	0		0	Make Positive	PR a	RR	If (Ra) < 0, (Ra)' → Ra; If (Ra) ≥ 0, (Ra) Unchanged	X	X	1.24	X	X
02	0		1	Make Negative	NR a	RR	If (Ra) > 0, (Ra)→ Ra; If (Ra) ≤ 0, (Ra) Unchanged	X	0	1.42	X	X
02	0		4	Two's Complement	TCR a	RR	(Ra)' → Ra	X	X	.84	X	X
02	0		5	Two's Complement Double	TCDR a	RR	(Ra, Ra + 1)' → Ra, Ra + 1	X	X	1.74	X	X
02	0		6	One's Complement	OCR a	RR	Bit-by-Bit Complement of (Ra) → Ra	0	0	.84	X	X

SHIFT INSTRUCTIONS

				LOGICAL SINGLE								
10	0			Logical Right Single Shift	LRSR a, m	RR	Shift (Ra) Right (Rm) 0-5 Places, Zero Fill	0	0	.94	X	X
10	2			Logical Right Single Shift	LRS a, y, m	RK	Shift (Ra) Right Y 0-5 Places, Zero Fill	0	0	1.76	X	X
60	0			Logical Right Single Shift	LLRS a, m	RL	Right Shift (Ra)n Places, Zero Fill	0	0	1.10	X	X
				ALGEBRAIC-SINGLE								
11	0			Algebraic Right Single Shift	ARSR a, m	RR	Shift (Ra) Right (Rm) 0-5 Places, Sign Fill	0	0	.95	X	X
11	2			Algebraic Right Single Shift	ARS a, y, m	RK	Shift (Ra) Right Y 0-5 Places, Sign Fill	0	0	1.80	X	X
60	1			Algebraic Right Single Shift	LARS a, m	RL	Right Shift (Ra)n Places, Sign Fill	0	0	1.10	X	X

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO SEC	C 9	C 8
14	0			Algebraic Left Single Shift	ALSR a, m	RR	Shift (Ra) Left (Rm) 0-5 Places, Zero Fill	0	X	2.30	X	X
14	2			Algebraic Left Single Shift	ALS a, y, m	RK	Shift (Ra) Left Y 0-5 Places, Zero Fill	0	X	3.15	X	X
61	0			Algebraic Left Single Shift	LALS a, m	RL	Left Shift (Ra)mPlaces Zero Fill	0	X	3.00	X	X
				CIRCULAR-SINGLE								
15	0			Circular Left Single Shift	CLSR a, m	RR	Shift (Ra) Left Circularly (Rm) 0-5 Places	0	0	.94	X	X
15	2			Circular Left Single Shift	CLS a, y, m	RK	Shift (Ra) Left Circularly Y 0-5 Places	0	0	1.75	X	X
61	1			Circular Left Shift (Ra)n Places; Set CC	LCLS a, m	RL	Circular-Left Shift (Ra)m Places	0	0	1.10	X	X
				LOGICAL DOUBLE								
12	0			Logical Right Double Shift	LRDR a, m	RR	Shift (Ra, Ra + 1) Right (Rm) 0-5 Places, Zero Fill	0	0	2.00	X	X
12	2			Logical Right Double Shift	LRD a, y, m	RK	Shift (Ra, Ra + 1) Right Y 0-5 Places, Zero Fill	0	0	2.80	X	X
60	2			Logical Right Double Shift	LLRD a, m	RL	Right Shift (Ra, Ra + 1)m Places, Zero Fill	0	0	2.15	X	X
				ALGEBRAIC-DOUBLE								
13	0			Algebraic Right Double Shift	ARDR a, m	RR	Shift (Ra, Ra + 1) Right (Rm) 0-5 Places, Sign Fill	0	0	2.00	X	X
13	2			Algebraic Right Double Shift	ARD a, y, m	RK	Shift (Ra, Ra + 1) Right Y 0-5 Places, Sign Fill	0	0	2.85	X	X
60	3			Algebraic Right Double Shift	LARD a, m	RL	Right Shift (Ra, Ra + 1)m Places, Sign Fill	0	0	2.15	X	X
16	0			Algebraic Left Double Shift	ALDR a, m	RR	Shift (Ra, Ra + 1) Left (Rm) 0-5 Places, Zero Fill	0	X	4.10	X	X
16	2			Algebraic Left Double Shift	ALD a, y, m	RK	Shift (Ra, Ra + 1) Left Y 0-5 Places, Zero Fill	0	X	4.95	X	X
61	2			Algebraic Left Double Shift	LALD a, m	RL	Left Shift (Ra, Ra + 1)m Places, Zero Fill	0	X	4.80	X	X
				CIRCULAR-DOUBLE								
17	0			Circular Left Double Shift	CLDR a, m	RR	Shift (Ra, Ra + 1) Left Circularly (Rm) 0-5 Places	0	0	2.0	X	X
17	2			Circular Left Double Shift	CLD a, y, m	RK	Shift (Ra, Ra + 1) Left Circularly 0-5 Places	0	0	2.85	X	X
61	3			Circular Left Double Shift	LCLD a, m	RL	Circular-Left Shift (Ra, Ra + 1)mPlaces	0	0	2.15	X	X
				MISCELLANEOUS INSTRUCTIONS								
				DISABLE/ENABLE CLOCKS								
03	0	0	10	Enable Real-Time Clock	ECR	RR	Enables R/T Clock Register	NC	NC	1.30	NC	NC
03	0	0	11	Disable Real-Time Clock	DCR	RR	Disables R/T Clock Register	NC	NC	1.30	NC	NC
03	0	0	16	Enable Real-Time Clock Overflow Interrupt	ECIR	RR	Enables RTC Overflow Interrupt	NC	NC	1.00	NC	NC
03	0	0	17	Disable RTC Overflow Interrupt	DCIR	RR	Disable RTC Overflow Interrupt	NC	NC	1.00	NC	NC
03	0	0	13	Disable Monitor Clock	DM	RR	Disable Monitor Clock and Monitor Clock Interrupt	NC	NC	1.00	NC	NC
				SPECIAL INSTRUCTIONS								
				Indirect Word	IW j, y, x		Causes a Two-Word Indirect Word (IW) to be Generated					
30	0	0		No Operation	NOP	RR	(RØ) ⊙ RØ → RØ	0	0	.99	X	X
30	0	0	0	No Operation Double	NOPD	RR	(RØ) ⊙ RØ → RØ twice	0	0	1.98	X	X
03	0	0	0	Executive Return	ER a	RR	Generate Interrupt; (P) + 1 → Ra	0	0	10.00	X	X
05	0			Set Bit	SBR a, m	RR	1 → Bit Position m of (Ra)	0	0	1.21	X	X
06	0			Zero Bit	ZBR a, m	RR	0-Bit Position m of (Ra)	0	0	1.23	X	X
07	0			Compare Bit	CBR a, m	RR	Bit Position m of (Ra)	0	0	1.58	X	X

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO SEC	C 9	C 8
35	0	0	0	Input/Output Command	IOCR	RR	Execute I/O Command Instruction	NC	NC	1.0 + I/O INST.	NC	NC
35	1	0		Biased Fetch	BFI m	RI	(Y*) Checked → CC; 1 → Y* Bits 15 and 14	0	0	2.45	X	X
35	2	0		Remote Execute	REX y, m	RK	Execute (Y); (P) + 2 → P	NC	NC	1.68 + INST.	NC	NC
35	3	0		Biased Fetch	BF y, m	RX	(Y) Checked → CC; 1 → Y Bits 15 and 14	0	0	3.0	X	X
67	0			User Macro	UM1 a, m	RR	Reserved for Arbitrary User Macro Instructions					
67	0			User Macro	UM2 a, m	RR	Reserved for Arbitrary User Macro Instructions					
67	1			User Macro	UMI a, m	RI	Reserved for Arbitrary User Macro Instructions					
67	2			User Macro	UMK a, y, m	RK	Reserved for Arbitrary User Macro Instructions					
				I/O INSTRUCTIONS □								
				ACTIVITY CONTROL								
70	0	0	0	Activity Control	ACR m	RR	Master Clear All Channels	u		18.9		
70	0	0	4	Activity Control	ACR m	RR	Enable All External Interrupts	u		10.5		
70	0	0	5	Activity Control	ACR m	RR	Disable All External Interrupts	u		10.5		
70	0	0	6	Activity Control	ACR m	RR	Enable All External Monitors	u		10.5		
70	0	0	7	Activity Control	ACR m	RR	Disable All External Monitors	u		10.5		
				CHANNEL CONTROL								
70	0		10	Channel a Control Operation	CCR a, m	RR	Master Clear Chan. a	z, u		3.25		
70	0		14	Channel a Control Operation	CCR a, m	RR	Enable Chan. a External Interrupts	z, u		2.75		
70	0		15	Channel a Control Operation	CCR a, m	RR	Disable Chan. a External Interrupts	z, u		2.75		
70	0		16	Channel a Control Operation	CCR a, m	RR	Enable Chan. a External Interrupts Monitor	z, u		2.75		
70	0		17	Channel a Control Operation	CCR a, m	RR	Disable Chan. a External Interrupts Monitor	z, u		2.75		
				ADDITIONAL I/O INSTRUCTIONS								
71	2		2	Initiate Input Chain	ICK a, y	RK	Initiate Input Chain	u		2.85		
71	2		6	Initiate Output Chain	OCK a, y	RK	Initiate Output Chain	u		2.85		
71	3			Write Control Memory	WIM a, y, m	RX	(Y) → Control Memory for Chan. a	u		3.30		
71	3			Write Control Memory	WCM am, y	RX	(Y) → Control Memory for Chan. a	u		3.30		
72	3			Read Control Memory	RIM a, y, m	RX	(Control Memory for Chan. a) → Y	u		3.35		
72	3			Read Control Memory	RCM am, y	RX	(Control Memory for Chan. a) → Y	u		3.35		
70	3		0	Initiate Transfer	IO a, y	RX	Initiate I/O	z		4.30		
71	2	0		Load Control Memory	LCMK m, y	RK	Y → Control Memory Specified by M	z		2.90		
71	3	0		Load Control Memory	LCM m, y	RX	(Y) → Control Memory Specified by M	z		3.30		
72	3	0		Store Control Memory	SCM m, y	RX	Control Memory Specified by M → Y	z		3.35		

16-BIT INSTRUCTION REPERTOIRE (CONT)

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	EXEC TIME MICRO SEC	C 9	C 8
73	0	0	0	Halt Chain	HCR	RR	This Instruction Halts the Chaining Action	z		2.3		
73	0	1	0	Interrupt Processor	IPR	RR	Generate Chain Interrupt	z		2.3		
73	3	0	0	Zero Flag	ZF y	RX	0 → Y 15, 14	z		4.0		
73	3	1	0	Set Flag	SF y	RX	1 → Y 15, 14	z		4.0		
74	2		0	Serial Jump on Condition	SJMC a, y	RK	If a-designator Condition is Met, Y - P; If Not, Execute Next Inst.	z		2.6		
75	0	0		Search For Synch	SFSC m	RR	Set Monitor/Suppress Flag and Enable Next Inst. of Chain	z		3.85		
76	0			Serial Interface Control	SICR a, m	RR	Sets or Clears Channel Discretes	u		2.5		
76	0	0		Serial Interface Control	CSIR m	RR	Sets or Clears Serial Channel Discretes	z		2.5		
76	3			Store Status	SST a, y	RX	Serial Status → Y	u		3.15		
76	3	0		Store Status	CSST y	RX	Serial Status Data → Y	z		3.15		

Footnotes:

NC:	No Change in the Designator
0:	End Result is 0
X:	Contingent Upon the Designator Function for that Instruction
NA:	Not Applicable
Y:	Y = y + (Rm), For All But RØ Y = y, For RØ Only Y = (P) + xd, Sign Extended to 16 Bits
#:	RI Type 1
u:	Command
z:	Chaining
□:	See User's Handbook for Further Information
/:	Command/Chaining
Y*:	The Effective Operand Address Contained in Rm

OPTIONAL MATH-PAC INSTRUCTIONS

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	(1) EXEC TIME MICRO SEC	C 9	C 8
04	0			Squareroot	SQRa	RR	$\sqrt{}$(Ra,Ra + 1) → Ra + 1; rem → Ra	0	X	9.65	X	X
37	0			See Pages 9 and 10								
50	0			Floating Point Subtract (Register)	FSUR a, m	RR	(Ra, Ra + 1) - (Rm, Rm + 1) → Ra	0	X	7.7–17.4 19.0	X	X
50	1			Floating Point Subtract (Indirect)	FSUI a, m	RI	(Ra, Ra + 1) - (Y*, Y* + 1) → Ra, Ra + 1; res → Ra + 2, Ra + 3	0	X	7.7–17.4 18.85	X	X
50	3			Floating Point Subtract	FSU a, y, m	RX	(Ra, Ra + 1) - (Y, Y + 1) → Ra, Ra + 1; res → Ra + 2, Ra + 3	0	X	19.7	X	X
51	0			Floating Point Add (Register)	FAR a, m	RR	(Ra, Ra + 1) + (Rm, Rm + 1) → Ra, Ra + 1; res → Ra + 2, Ra + 3	0	X	18.7	X	X
51	1			Floating Point Add (Indirect)	FAI a, m	RI	(Ra, Ra + 1) + (Y*, Y* + 1) → Ra, Ra + 1, res → Ra + 2, Ra + 3	0	X	18.7	X	X

(1) VARIATION DEPENDENT ON DATA
(2) IF OVERFLOW/UNDERFLOW AND CLASS II INTERRUPTS ARE DISABLED, THEN 11.3 μSEC.
(3) IF UNDERFLOW AND CLASS II INTERRUPTS ARE DISABLED, THEN 13.8 μSEC.
(4) NORMALIZED TIMES; IF UNNORMALIZED OPERANDS ARE USED, THEN 45.9 μSEC.
(5) NORMALIZED TIMES; IF UNNORMALIZED OPERANDS ARE USED, THEN 46.6 μSEC.

OPTIONAL MATH-PAC INSTRUCTIONS

o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	(1) EXEC TIME MICRO SEC	C 9	C 8
51	3			Floating Point Add	FA a, y, m	RX	(Ra, Ra + 1) + (Y, Y + 1) → Ra, Ra + 1; res → Ra + 2, Ra + 3	0	X	19.5	X	X
52	0			Floating Point Multiply (Register)	FMR a, m	RR	(Ra, Ra + 1)* (Rm, Rm + 1) → Ra, Ra + 1, Ra + 2, Ra + 3	0	X	20.0	X	X
52	1			Floating Point Multiply (Indirect)	FMI a, m	RI	(Ra, Ra + 1)* (Y*, Y* + 1) → Ra, Ra + 1, Ra + 2, Ra + 3	0	X	20.3	X	X
52	3			Floating Point Multiply	FM a, y, m	RX	(Ra, Ra + 1)* (Y, Y + 1) → Ra, Ra + 1, Ra + 2, Ra + 3	0	X	21.2	X	X
53	0			Floating Point Divide (Register)	FDR a, m	RR	(Ra, Ra + 1)/(Rm, Rm + 1) → Ra, Ra + 1; rem → Ra + 2, Ra + 3	0	X	25.7 (4)	X	X
53	1			Floating Point Divide (Indirect)	FDI a, m	RI	(Ra, Ra + 1)/(Y*, Y* + 1) → Ra, Ra + 1; rem → Ra + 2, Ra + 3	0	X	25.7 (4)	X	X
53	3			Floating Point Divide	RD a, y, m	RX	(Ra, Ra + 1)/(Y, Y + 1) → Ra, Ra + 1; rem → Ra + 2, Ra + 3	0	X	26.4 (5)	X	X
56	0			Multiply Double (Register)	MDR a, m	RR	(Ra, Ra + 1)* (Rm, Rm + 1) → Ra, Ra + 1, Ra + 2, Ra + 3	0	0	9.35	X	X
56	1			Multiply Double (Indirect)	MDI a, m	RI	(Ra, Ra + 1)* (Y*, Y* + 1) → Ra, Ra + 1, Ra + 2, Ra + 3	0	0	9.60	X	X
56	3			Multiply Double	MD a, y, m	RX	(Ra, Ra + 1)* (Y, Y + 1) → Ra, Ra + 1, Ra + 2, Ra + 3	0	0	10.6	X	X
57	0			Divide Double (Register)	DDR a, m	RR	(Ra, Ra + 1, Ra + 2, Ra + 3) /(Rm, Rm + 1) → Ra + 2, Ra + 3; rem → Ra, Ra + 1	0	X	19.8	X	X
57	1			Divide Double (Indirect)	DDI a, m	RI	(Ra, Ra + 1, Ra + 2, Ra + 3)/(Y*, Y* + 1) → Ra + 2, Ra + 3; rem → Ra, Ra + 1	0	X	21.0	X	X
57	3			Divide Double	DD a, y, m	RX	(Ra, Ra + 1, Ra + 2, Ra + 3)/(Y, Y + 1) → Ra + 2, Ra + 3; rem → Ra, Ra + 1	0	X	21.0	X	X

OPTIONAL MATH-PAC II INSTRUCTIONS

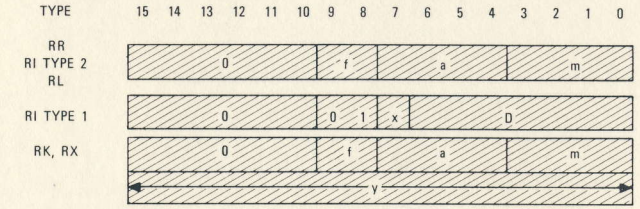
o	f	a	m	NAME OF INSTRUCTION	MNEMONIC a, y, m	FOR-MAT	OPERATION	CARRY DESIG	OVER FLOW	(1) EXEC TIME MICRO SEC	C 9	C 8
37	0		10	Floating Point Compare	FC a, y	RR	(Ra, Ra + 1); (Y, Y + 1)	0	0	4.70	X	X
37	0		11	Fixed to Floating Point Conversion	FXC a	RR	(Ra) → EXP.; (Ra + 1) → MAN	X	X	8.4 (2)	X	X
37	0		12	Floating Point to Fixed Single Conversion	FLC a	RR	CONVERT (Ra, Ra + 1); EXP. → Ra; MAN. → Ra + 1	0	0	5.45	X	X
37	0		13	Floating Point Normalize	NF a	RR	NORMALIZE (Ra, Ra + 1)	X	X	23→ (3) 10.5	X	X
37	0		16	Algebraic Left Quadruple Shift	QAL a, y	RR	SHIFT (Ra, Ra + 1, Ra + 2, Ra + 3) left Y_{5-0} places, Zero Fill	0	X	10.0	X	X
37	0		17	Algebraic Right Quadruple Shift	QAR a, y	RR	SHIFT (Ra, Ra + 1, Ra + 2, Ra + 3) right Y_{5-0} places, Sign Fill	0	0	8.3	X	X

(1) VARIATION DEPENDENT ON DATA
(2) IF OVERFLOW/UNDERFLOW AND CLASS II INTERRUPTS ARE DISABLED, THEN 11.3 μSEC.
(3) IF UNDERFLOW AND CLASS II INTERRUPTS ARE DISABLED, THEN 13.8 μSEC.
(4) NORMALIZED TIMES; IF UNNORMALIZED OPERANDS ARE USED, THEN 45.9 μSEC.
(5) NORMALIZED TIMES; IF UNNORMALIZED OPERANDS ARE USED, THEN 46.6 μSEC.

o	f	a	m	CODING FORMAT	FUNCTION	INPUT PARAMETERS R_a	INPUT PARAMETERS R_{a+1}	INPUT PARAMETERS R_{a+2}	OUTPUT RESULTS $Y \rightarrow R_a$	OUTPUT RESULTS $X \rightarrow R_{a+1}$	OUTPUT RESULTS $W \rightarrow R_{a+2}$	EXEC TIME MICRO SEC.
37	0	a	00	VF a	Trigonometric vector	y	x	0	0	$X = \frac{R}{K} = \sqrt{\frac{x^2 + y^2}{K}}$	$W = \theta = \tan^{-1} \frac{y}{x}$	12.6
37	0	a	01	RF a	Trigonometric rotate	y	x	θ	$Y = \frac{y \cos\theta + x \sin\theta}{K}$	$X = \frac{x \cos\theta - y \sin\theta}{K}$	0	12.3
37	0	a	02	VFP a	Trig. vector with prescale	y	x	0	0	$X = R = \sqrt{x^2 + y^2}$	$W = \theta = \tan^{-1} \frac{y}{x}$	15.9
37	0	a	03	RFP a	Trig. rotate with prescale	y	x	θ	$Y = y \cos\theta + x \sin\theta$	$X = x \cos\theta - y \sin\theta$	0	15.6
37	0	a	04	VH a	Hyperbolic vector	y	x	0	0	$X = \sqrt{x^2 - y^2}{K_1}$	$W = v = \tanh^{-1} \frac{y}{x}$	12.8
37	0	a	05	RH a	Hyperbolic rotate	y	x	v	$Y = \frac{y \cosh v + x \sinh v}{K_1}$	$X = \frac{x \cosh v + y \sinh v}{K_1}$	0	12.8
37	0	a	06	VHP a	Hyp. vector with postscale	y	x	0	0	$X = \sqrt{x^2 - y^2}$	$W = v = \tanh^{-1} \frac{y}{x}$	16.0
37	0	a	07	RHP a	Hyp. rotate with postscale	y	x	v	$Y = y \cosh v + x \sinh v$	$X = x \cosh v + y \sinh v$	0	16.0
37	0	a	01	RF a	Sin θ, COS θ without prescale	0	0.466728_8	θ	$Y = \sin\theta$	$X = \cos\theta$	0	12.3
37	0	a	06	VHP a	$\text{Log}_e x$	x-1	x+1	0	0	2x	$W = 1/2 \log_e \frac{x+1}{x-1}$ $= \tanh^{-1} \frac{x+1}{x-1}$	16.0
37	0	a	07	RHP a	Exponential	1	1	v positive	$Y = e^v = \sinh v + \cosh v$	$X = e^v = \sinh v + \cosh v$	0	16.0
37	0	a	01	RF a	Polar to Cartesian without prescale	0	R	θ	$Y = \frac{R \sin\theta}{K}$	$X = \frac{R \cos\theta}{K}$	0	12.3
37	0	a	03	RFP a	Polar to Cartesian with prescale	0	R	θ	$Y = R \sin\theta$	$X = R \cos\theta$	0	15.6
37	0	a	01	RF a	Sin θ; cos θ	0	1	θ	$Y = \frac{\sin\theta}{K}$	$X = \frac{\cos\theta}{K}$	0	12.3

TRIGONOMETRIC AND HYPERBOLIC FUNCTIONS
(Operation Code 37)

x&y Cartesian coordinates. Radix point assumed between bits 15 and 14
θ Angle of rotation Trignometric mode (BAMS) Bit 14 = 90°
v Angle of rotation Hyperbolic mode Radix point assumed between bits 15 and 14
K 0.466728_8
K_1 1.152178_8



DEFINITION OF FIELDS

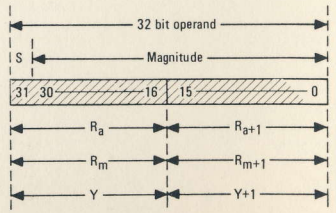
- 0 Operation (Function) Code
- f Format Designator
  - 00 ⇒ Format RR, Register to Register or RL-1 Format
  - 01 ⇒ Format RI, Register Indirect Memory or RL-2 Format
  - 10 ⇒ Format RK, Register-Literal Constant or RL-3 Format
  - 11 ⇒ Format RX, Register-Indexed Address, Constant or RL-4 Format
- a General Register or Subfunction Designator
- m General Register or Subfunction Designator
  - 4-bit Unsigned Literal Constant in RL Format
- xD Signed Deviation Value (Two's Complement)
- y Address or Arithmetic Constant

Figure 1. Instruction Word Format

LEGEND

- B Byte pointer, 0 → Upper, 1 → Lower
- C Carry
- CC Condition Code
- OV Overflow
- IW Indirect Word
- j Designator Field in IW
- x General Register Designator in IW1
- y Contents of Second Instruction Word or IW2
- Y Effective Operand Address or Constant
- Y* Effective Operand Address in Rm
- TM I/O Transfer Mode
  - 00 → Abort Transfer
  - 01 → 8-bit Byte Transfer
  - 10 → 16-bit Word Transfer
  - 11 → 32-bit Dual Word Transfer
- BWC Buffer Word Count
- BAP Buffer Address Pointer
- CM Control Memory Word
- CAP Chain Address Pointer
- RTC Real-Time Clock
- ( ) Contents of register or address
- r $(R_a)_{5\text{-}0}$
- u $(R_a)_{13\text{-}8}$

FORMAT	OPERAND FORMATION
RR	Operand = (R_m)
RI-1	Local Jump Address Y = (P) + xD
RI-2	Operand at Y* = (R_m)
RK	Operand Y = y + (R_m) if m ≠ 0 Operand Y = y if m = 0
RX Word	Operand at Y = y if m = 0 Operand at Y = y + (R_m) if m ≠ 0 Operand at indirect address if m = 10,12,14,16
RX Byte	Operand at Y upper if m = 0 Operand at Y = $(R_m)/2$ + y if m ≠ 0,10,12,14,16; B = $(R_m)_0$ Operand at indirect address if m = 10,12,14,16
RL	Operand = m (an absolute literal)



Double Length Operands

OR	0	1
0	0	1
1	1	1

XOR	0	1
0	0	1
1	1	0

AND	0	1
0	0	0
1	0	1

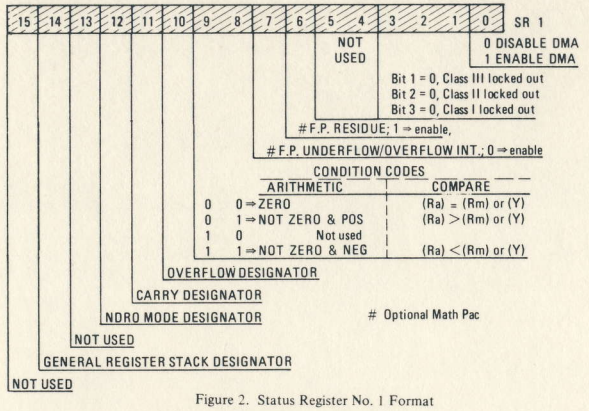
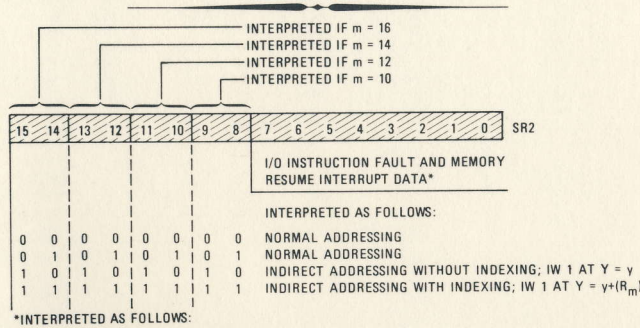
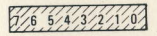


Figure 2. Status Register No. 1 Format



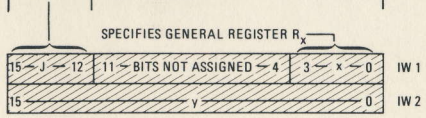
*INTERPRETED AS FOLLOWS:



C C C C 0 X 1 0 CHAIN INSTRUCTION CCCC = CHAN #; X = 0 ⇒ INPUT; X = 1 ⇒ OUTPUT
0 0 0 0 0 0 0 1 COMMAND INSTRUCTION
1 M M M 0 0 1 0 MEMORY RESUME INTERRUPT; MMM = 8K MODULE NO.

Figure 3. Status Register No. 2 Format

OCTAL J-VALUE	OPERAND/IW1, LOCATION
0	WORD AT Y = (IW2) BYTE AT UPPER HALF OF Y = (IW2)
1	WORD AT Y = (IW2) + (R_x) BYTE AT Y = (IW2) + (R_x) */2
2	WORD AT Y = (IW2) + (R_m) BYTE AT Y = (IW2) + (R_m)*/2
3	WORD AT Y = (IW2) + (R_{m+1}) BYTE AT Y = (IW2) + (R_{m+1}) */2
4	NEXT IW 1 AT ADDRESS Y = (IW2)
5	NEXT IW 1 AT ADDRESS Y = (IW2) + (R_x)
6	NEXT IW 1 AT ADDRESS Y = (IW2) + (R_m)
7	NEXT IW 1 AT ADDRESS Y = (IW2) + (R_{m+1})
10-17	NOT ASSIGNED



* B = LSB of register

Figure 4. Indirect Addressing

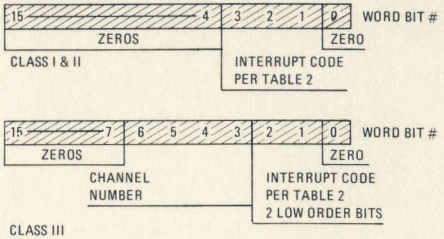


Figure 5. Interrupt Entrance Address Index

TABLE 1. ASSIGNED MEMORY ADDRESS

Function	Address Assignment to Class III	Class II	Class I
Store P addresses	110	120	130
Store SR # 1 addresses	111	121	131
Store SR # 2 addresses	112	122	132
Store RTC lower addresses	113	123	133
P Reload addresses	114	124	134
SR # 1 Reload addresses	115	125	135
SR # 2 Reload addresses	116	126	136
Store RTC upper addresses	117	127	137
I/O Command cells	140-141		
Auto start entrance	177		
External interrupt word storage	200-217		
NDRO	00-77, 300-477		

TABLE 2. INTERRUPT PRIORITY

Class	Priority Within Class	Interrupt	Binary Interrupt Code Generated
Class I, Hardware Errors	1	Power Fault	000
	2	Memory Resume	001
Class II, Software Interrupts	1	CP Instruction Fault	000
	2	I/O Instruction Fault	001
	3	#F.P. Overflow/Underflow Interrupt	010
	4	Executive Return Instruction	011
	5	RTC Overflow	100
	6	Monitor Clock	101
Class III, IOC Interrupts	1	Intercomputer Time-Out	11
	2	External Interrupt or Discrete Interrupt ①	00
	3	Output Chain Interrupt	10
	4	Input Chain Interrupt	01

① Serial MIL-STD-188C vacales, or EIA-STD-RS 232C Channels # Optional Math Pac function

a-Value	m-Value	CONTROL MEMORY Register Selected 15 14 13 12 11 0
	0	TM 0 B BWC (IN)
	1	BAP (IN)
	2	CAP (IN)
	3	Not used
	4	TM 0 B BWC (OUT)
	5	BAP (OUT)
	6	CAP (OUT)
	7	Not used
	10	Monitor register (Serial)
	11	Suppress register (Serial)
	12	Serial mode information*
	13-17	Not used
0-17	Channel designator	

*MIL-STD-188C or RS-232

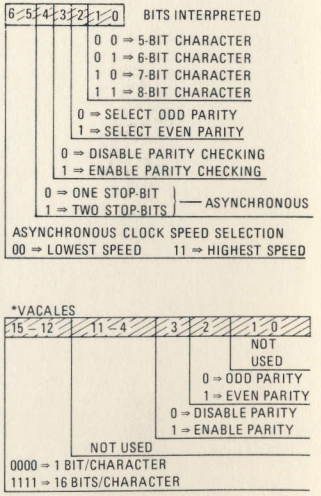


Figure 6. I/O Control Memory

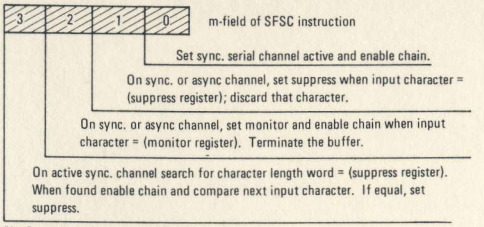


Figure 7. SFSC Operations

BITS	MIL-STD-188	RS-232	VACALES
0 - 7	ALWAYS ONES	ALWAYS ONES	ALWAYS ONES
8	1 ⇒ B DISCRETE TURNED ON	1 ⇒ RING INDICATOR ON	1 ⇒ B DISCRETE TURNED ON
9	1 ⇒ C DISCRETE TURNED OFF	1 ⇒ RECEIVED LINE SIGNAL DETECTOR OFF	1 ⇒ CARRIER DETECT TURNED OFF
10	1 ⇒ I DISCRETE TURNED ON	ALWAYS ONE	1 ⇒ ALARM INDICATE TURNED ON
11	ALWAYS ONE	ALWAYS ONE	1 ⇒ SYNC ERROR TURNED ON
12	ALWAYS ONE	ALWAYS ONE	1 ⇒ TRANSMIT FULL ON TURNED OFF
13 - 15	ALWAYS ONES	ALWAYS ONES	ALWAYS ONES

FIGURE 8. SERIAL CHANNEL INTERRUPT WORD FORMAT

TABLE 3. SERIAL I/O DISCRETE FUNCTIONS

Octal m-Value	Function	MIL-STD-188C/VACALES Discrete	Line Designator (188C)	Line Designator (Vacales)	EIA-STD-RS232 Discrete	Line Designator
0	Set	Loop test (internal)	–	–	Loop test (internal)	–
1	Clear	Loop test (internal)	–	–	Loop test (internal)	–
2	Clear	Not used	–	–	Spare	–
3	Set	Not used	–	–	Spare	–
4	Clear	Control Line 6	J	J1	Spare	–
5	Set	Control Line 6	J	J1	Spare	–
6	Clear	Control Line 5	H	TRAN. PREP	Enable Ring Indicator	CE
7	Set	Control Line 5	H	TRAN. PREP	Enable Ring Indicator	CE
10	Clear	Control Line 4	G	G1	Request to Send	CA
11	Set	Control Line 4	G	G1	Request to Send	CA
12	Clear	Control Line 3	F	F1	New Sync	CH
13	Set	Control Line 3	F	F1	New Sync	CH
14	Clear	Control Line 2	D	D1	Data Terminal Ready	CD
15	Set	Control Line 2	D	D1	Data Terminal Ready	CD
16	Clear	Control Line 1	A	LOOP BACK	Loop Test (external)	
17	Set	Control Line 1	A	LOOP BACK	Loop Test (external)	

TABLE 4. SERIAL I/O STATUS INTERPRETATION

Word Bit #	MIL-STD-188 Function	EIA-STD-RS232 Function	VACALES FUNCTION
2^0	Parity Error	Parity Error	–
2^1	Overrun	Overrun	Overrun
2^2	Break	Break	Parity Error
2^3	E Active	Clear to Send	Sync Error