70-SERIES FAMILY INSTRUCTION SET

INSTRUCTION	MNE-MONIC	FIRST OPERAND	SECOND OPERAND	ADDRESSING												FLAGS AFFECTED								OPERATION PERFORMED
				PC RELATIVE	POINTER RELATIVE			IMMEDIATE	DIRECT	AUTO-INDEXED		IMPLIED	INDEXED		ABSOLUTE or INDIRECT									
					SP	P2	P3			P2	P3		P2	P3										
				OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	CY	OV	SB	SA	F3	F2	F1	IE	
LOAD	LD	A		C0 7 2	C1 7 2	C2 7 2	C3 7 2	C4 5 2	C5 7 2	C6 8 2	C7 8 2													(A)←(addr)
		EA		80 10 2	81 10 2	82 10 2	83 10 2	84 8 3	85 10 2	86 11 2	87 11 2													(EA)←(addr + 1, addr)
		T		A0 10 2	A1 10 2	A2 10 2	A3 10 2	A4 8 3	A5 10 2	A6 11 2	A7 11 2													(T)←(addr + 1, addr)
		SP						25 8 3																(SP)←(PC + 2, PC + 1)
		P2						26 8 3																(P2)←(PC + 2, PC + 1)
		P3						27 8 3																(P3)←(PC + 2, PC + 1)
		A	E									40 4 1												(A)←(E)
		E	A									48 4 1												(E)←(A)
		A	S									06 3 1												(A)←(S)
		S	A									07 3 1												(S)←(A)
		EA	PC									30 4 1												(EA)←(PC)
		EA	SP									31 4 1												(EA)←(SP)
		EA	P2									32 4 1												(EA)←(P2)
		EA	P3									33 4 1												(EA)←(P3)
		SP	EA									45 5 1												(SP)←(EA)
		P2	EA									46 5 1												(P2)←(EA)
		P3	EA									47 5 1												(P3)←(EA)
		T	EA									09 4 1												(T)←(EA)
		EA	T									0B 4 1												(EA)←(T)
STORE	ST	A		C8 7 2	C9 7 2	CA 7 2	CB 7 2		CD 7 2	CE 8 2	CF 8 2													(A)→(addr)
		EA		88 10 2	89 10 2	8A 10 2	8B 10 2		8D 10 2	8E 11 2	8F 11 2													(AE)→(addr + 1, addr)
ADD	ADD	A		F0 7 2	F1 7 2	F2 7 2	F3 7 2	F4 7 2	F5 7 2	F6 8 2	F7 8 2					•	•							(A)←(A) + (addr)
		A	E									70 4 1				•	•							(A)←(A) + (E)
		EA		B0 10 2	B1 10 2	B2 10 2	83 10 2	B4 10 2	B5 10 2	B6 11 2	B7 11 2					•	•							(EA)←(EA) + (addr + 1, addr)
SUBTRACT	SUB	A		F8 7 2	F9 7 2	FA 7 2	FB 7 2	FC 7 2	FD 7 2	FE 8 2	FF 8 2					•	•							(A)←(A) - (addr)
		A	E									78 4 1				•	•							(A)←(A) - (E)
		EA		B8 10 2	B9 10 2	BA 10 2	BB 10 2	BC 10 2	BD 10 2	BE 11 2	BF 11 2					•	•							(EA)←(EA) - (addr + 1, addr)
MULTIPLY	MPY	EA	T									2C 37 1				•	•							(EA)←[(EA) * (T)] 31:16; (T)←[(EA) * (T)] 15:0
DIVIDE	DIV	EA	T									0D 41 1				•	•							Quotient: (EA)←[(EA / (T)] 15:0 Remainder: (T)←[(EA) / (T)]
AND	AND	A		D0 7 2	D1 7 2	D2 7 2	D3 7 2	D4 7 2	D5 7 2	D6 8 2	D7 8 2													(A)←(A) /\ (addr)
		A	E									50 4 1												(A)←(A) /\ (E)
		S						39 5 2								•	•			•	•	•	•	(S)←(S) /\ data
OR	OR	A		D8 7 2	D9 7 2	DA 7 2	DB 7 2	DC 7 2	DD 7 2	DE 8 2	DF 8 2													(A)←(A) \/ (addr)
		A	E									58 4 1												(A)←(A) \/ (E)
		S						3B 5 2								•	•			•	•	•	•	(S)←(S) \/ data
EXCLUSIVE-OR	XOR	A		Eb 7 2	E1 7 2	E2 7 2	E3 7 2	E4 7 2	E5 7 2	E6 8 2	E7 9 2													(A)←(A) ⊻ (addr)
		A	E									60 4 1												(A)←(A) ⊻ (E)

INSTRUCTION	MNE-MONIC	FIRST OPERAND	SECOND OPERAND	ADDRESSING												FLAGS AFFECTED								OPERATION PERFORMED
				PC RELATIVE	POINTER RELATIVE			IMMEDIATE	DIRECT	AUTO-INDEXED		IMPLIED	INDEXED		ABSOLUTE or INDIRECT									
					SP	P2	P3			P2	P3		P2	P3										
			•	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	CY	OV	SB	SA	F3	F2	F1	IE	
EXCHANGE REGISTERS	XCH	A	E									01 5 1												(A)←→(E)
		EA	SP									4D 7 1												(EA)←→(SP)
		EA	P2									4E 7 1												(EA)←→(P2)
		EA	P3									4F 7 1												(EA)←→(P3)
SHIFT RIGHT	SR	A										3C 3 1												(Ai)→(Ai-1); i = 7, 1; 0→A7
		EA										0C 4 1												(EAi)→(EAi-1); i = 15, 1; 0→EA15
SHIFT RIGHT WITH LINK	SRL	A										3D 3 1				•								(Ai)→(Ai-1); i = 7, 1; CY→A7
ROTATE RIGHT	RR	A										3E 3 1												(Ai)→(Ai-1); i = 7, 1; A0→A7
ROTATE RIGHT WITH LINK	RRL	A										3F 3 1				•								(Ai)→(Ai-1); i = 7, 1; A0→CY→A7
SHIFT LEFT	SL	A										0E 3 1												(Ai+1)←(Ai); i = 6, 0; 0→A0
		EA										0F 4 1												(EAi+1)←(EAi); i = 14, 0; 0→EA0
SEARCH AND SKIP IF CHARACTER MATCHED	SSM												2E 1	2F 1										See Text
BRANCH IF NOT DIGIT	BND			2D 2																				If (A) ≠ ASCII: (PC)← addr If not digit: 7 or 9, If digit: 7
PUSH	PUSH	A										0A 5 1												(SP)←(SP) - 1; ((SP))←(A)
		EA										08 8 1												((SP)) - 1←(E); ((SP)) - 2←(A); (SP)←(SP) - 2
		PC										54 8 1												((SP)) - 1←(PCH); ((SP)) - 2←(PCL); (SP)←(SP) -2
		P2										56 8 1												((SP)) - 1←(P2H); ((SP)) - 2←(P2L); (SP)←(SP) - 2
		P3										57 8 1												((SP)) - 1←(P3H); ((SP)) - 2←(P3L); (SP)←(SP) - 2
PUSH AND LOAD IMMEDIATE	PLI	P2						22 15 3																((SP)) - 1←(P2H); ((SP)) - 2←(P2L); (SP)←(SP) - 2; (P2H)←byte 3; (P2L)←byte 2
		P3						23 15 3																((SP)) - 1←(P3H); ((SP)) - 2←(P3L); (SP)←(SP) - 2; (P3H)←byte 3; (P3L)←byte 2
POP	POP	A										38 6 1												(A)←((SP)); (SP)←(SP) + 1
		EA										3A 9 1												(A)←((SP)); (E)←((SP)) + 1; (SP)←(SP) + 2
		P2										5E 10 1												(P2L)←((SP)); (P2H)←((SP)) + 1; (SP)←(SP) + 2
		P3										5F 10 1												(P3L)←((SP)); (P3H)←((SP)) + 1; (SP)←(SP) + 2
BRANCH UNCONDITIONAL	BRA			74 5 2		76 5 2	77 5 2																	(PC)←(PC) + disp
BRANCH POSITIVE	BP			64 5 2		66 5 2	67 5 2																	If (A) > 0: (PC)←(PC) + disp
BRANCH ZERO	BZ			6C 5 2		6E 5 2	6F 5 2																	If (A) = 0: (PC)←(PC) + disp
BRANCH NOT ZERO	BNZ			7C 5 2		7E 5 2	7F 5 2																	If (A) ≠ 0: (PC)←(PC) + disp
JUMP UNCONDITIONAL	JMP														24 8 3									(PCH)←byte 3; (PCL)←byte 2
JUMP TO SUBROUTINE	JSR														20 15 3									((SP)) - 1←(PCH)←byte 3 ((SP)) - 2←(PCL)←byte 2, (SP)←(SP) - 2
CALL	CALL	0 - 15													10-1F 17 1									((SP)) - 1←(PCH)←(021 + 2*N); ((SP)) - 2←(PCL)←(020 + 2*N); (SP)←(SP) - 2
RETURN	RET											5C 10 1												(PCL)←((SP)); (PCH)←((SP)) + 1; (SP)←(SP) + 2
LOAD PC	LD	PC	EA									44 5 1												(PC)←(EA)
EXCHANGE PC	XCH	PC	EA									4C 7 1												(PC)←→(EA)
INCREMENT AND LOAD	ILD	A		90 8 2	91 8 2	92 8 2	93 8 2		95 8 2	96 9 2	97 9 2													(A), (addr)←(addr) + 1
DECREMENT AND LOAD	DLD	A		98 8 2	99 8 2	9A 8 2	9B 8 2		9D 8 2	9E 9 2	9F 9 2													(A), (addr)←(addr) - 1
NO OPERATION	NOP											00 3 1												(PC)←(PC) + 1