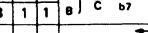
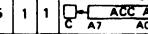
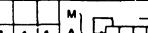
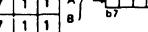
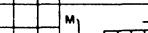
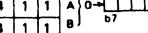
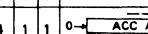
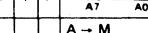
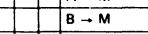


Table 1 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes										Boolean/ Arithmetic Operation	Condition Code Register														
		IMMED		DIRECT		INDEX		EXTEND		IMPLIED			5	4	3	2	1	0									
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	H	I	N	Z	V	C								
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3	A + M → A	1	•	1	1	1								
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3	B + M → B	1	•	1	1	1								
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3	A : B + M : M + 1 → A : B	0	•	1	1	1								
Add Accumulators	ABA													1B	1	1	A + B → A	1	•	1	1	1					
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3	A + M + C → A	1	•	1	1	1								
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3	B + M + C → B	1	•	1	1	1								
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3	A · M → A	0	•	1	1	R •								
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3	B · M → B	0	•	1	1	R •								
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3	A · M	0	•	1	1	R •								
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3	B · M	0	•	1	1	R •								
Clear	CLR							6F	5	2	7F	5	3	00 → M	0	•	R	S	R R								
	CLRA													4F	1	1	00 → A	0	•	R	S	R R					
	CLRB													5F	1	1	00 → B	0	•	R	S	R R					
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3	A - M	0	•	1	1	1								
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3	B - M	0	•	1	1	1								
Compare Accumulators	CBA													11	1	1	A - B	0	•	1	1	1					
Complement, 1's	COM							63	6	2	73	6	3	M → M	0	•	1	1	R S								
	COMA													43	1	1	A → A	0	•	1	1	R S					
	COMB													53	1	1	B → B	0	•	1	1	R S					
Complement, 2's (Negate)	NEG							60	6	2	70	6	3	00 - M → M	0	•	1	1	① (②)								
	NEGA													40	1	1	00 - A → A	0	•	1	1	① (②)					
	NEGB													50	1	1	00 - B → B	0	•	1	1	① (②)					
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format				•	•	1	1	1	③	
Decrement	DEC							6A	6	2	7A	6	3	M - 1 → M	0	•	1	1	④	•							
	DECA													4A	1	1	A - 1 → A	0	•	1	1	④	•				
	DECB													5A	1	1	B - 1 → B	0	•	1	1	④	•				
Exclusive OR	EORA	88	2	2	98	3	2	AB	4	2	BB	4	3	A ⊕ M → A	0	•	1	1	R •								
	EORB	C8	2	2	D8	3	2	EB	4	2	FB	4	3	B ⊕ M → B	0	•	1	1	R •								
Increment	INC							6C	6	2	7C	6	3	M + 1 → M	0	•	1	1	⑤	•							
	INCA													4C	1	1	A + 1 → A	0	•	1	1	⑤	•				
	INCB													5C	1	1	B + 1 → B	0	•	1	1	⑤	•				
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3	M → A	0	•	1	1	R •								
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3	M → B	0	•	1	1	R •								
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3	M + 1 → B, M → A	0	•	1	1	R •								
Multiply Unsigned	MUL													3D	7	1	A × B → A : B				•	•	•	•	•	⑪	
	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3	A + M → A	0	•	1	1	R •								
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3	B + M → B	0	•	1	1	R •								
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP				•	•	•	•	•		
	PSHB													37	4	1	B → Msp, SP - 1 → SP				•	•	•	•	•		
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A				•	•	•	•	•		
	PULB													33	3	1	SP + 1 → SP, Msp → B				•	•	•	•	•		
Rotate Left	ROL							69	6	2	79	6	3		M	A							•	•	1	⑥	1
	ROLA													49	1	1							•	•	1	⑥	1
	ROLB													59	1	1							•	•	1	⑥	1
Rotate Right	ROR							66	6	2	76	6	3		M	A							•	•	1	⑥	1
	RORA													46	1	1							•	•	1	⑥	1
	RORB													56	1	1							•	•	1	⑥	1

Operations	Mnemonic	Addressing Modes										Boolean/ Arithmetic Operation	Condition Code Register						
		IMMED		DIRECT		INDEX		EXTEND		IMPLIED									
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	M	A	C	b7	b0	
Shift Left Arithmetic	ASL				68	6	2	78	6	3				48	1	1			
	ASLA												58	1	1				
	ASLB																		
Double Shift Left, Arithmetic	ASLD										05	1	1	0	ACC	A/ACC B	0		
Shift Right Arithmetic	ASR				67	6	2	77	6	3				47	1	1			
	ASRA										57	1	1						
	ASRB																		
Shift Right Logical	LSR				64	6	2	74	6	3				44	1	1			
	LSRA										54	1	1						
	LSRB																		
Double Shift Right Logical	LSRD										04	1	1	0	ACC	A/ACC B	0		
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3			A → M			
	STAB				D7	3	2	E7	4	2	F7	4	3			B → M			
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3			A → M			
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3			B → M + 1			
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3			A → M → A			
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3			B → M → B			
Subtract Accumulators	SBA													10	1	1	A → B → A		
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3			A → M → C → A			
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3			B → M → C → B			
Transfer Accumulators	TAB													16	1	1	A → B		
	TBA													17	1	1	B → A		
Test Zero or Minus	TST							6D	4	2	7D	4	3			M → 00			
	TSTA													4D	1	1	A → 00		
TSTB														5D	1	1	B → 00		
And Immediate	AIM				71	6	3	61	7	3						M → IMM → M			
OR Immediate	OIM				72	6	3	62	7	3						M + IMM → M			
EOR Immediate	EIM				75	6	3	65	7	3						M ⊖ IMM → M			
Test Immediate	TIM				7B	4	3	6B	5	3						M → IMM			

OP CODE					ACC A	ACC B	IND	EXT DIR	ACCA or SP				ACCB or X									
									IMM	DIR	IND	EXT	IMM	DIR	IND	EXT						
HI	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111						
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F						
0000	0	SBA	BRA	TSX	NEG				SUB				0									
0001	I	NOP	CBA	BRN	INS	AIM				CMP				1								
0010	2	BHI				PULA	OIM				SBC				2							
0011	3	BLS				PULB	COM				SUBD				ADDD							
0100	4	LSRD	BCC				DES	LSR				AND				3						
0101	5	ASLD	BCS				TXS	EIM				BIT				4						
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA				5				6				
0111	7	TPA	TBA	BEQ	PSHB	ASR				STA				STA				7				
1000	8	INX	XGDX	BVC	PULX	ASL				EOR				8				9				
1001	9	DEX	DAA	BVS	RTS	ROL				ADC				9				A				
1010	A	CLV	SLP	BPL	ABX	DEC				ORA				A				B				
1011	B	SEV	ABA	BMI	RTI	TIM				ADD				B				C				
1100	C	CLC	BGE				PSHX	INC				CPX				C						
1101	D	SEC	BLT				MUL	TST				BSR	JSR				D					
1110	E	CLI	BGT				WAI	JMP				LDS	LDX				E					
1111	F	SEI	BLE				SWI	CLR				STS	STX				F					
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F						

Table 2 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes					Boolean/ Arithmetic Operation	Condition Code Register
		IMMED.	DIRECT	INDEX	EXTEND	IMPLIED		
		OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #		
Compare Index Reg	CPX	BC	3 3	9C	4 2	AC	5 2	X - M + 1
Decrement Index Reg	DEX						09	X - 1 → X
Decrement Stack Pntr	DES						34	SP - 1 → SP
Increment Index Reg	INX						08	X + 1 → X
Increment Stack Pntr	INS						31	SP + 1 → SP
Load Index Reg	LDX	CE	3 3	DE	4 2	EE	5 2	M → X _H , (M + 1) → X _L
Load Stack Pntr	LDS	BE	3 3	9E	4 2	AE	5 2	M → SP _H , (M + 1) → SP _L
Store Index Reg	STX			DF	4 2	EF	5 2	X _H → M, X _L → (M + 1)
Store Stack Pntr	STS			9F	4 2	AF	5 2	SP _H → M, SP _L → (M + 1)
Index Reg → Stack Pntr	TXS						35	X - 1 → SP
Stack Pntr → Index Reg	TSX						30	SP + 1 → X
Add	ABX						3A	B + X → X
Push Data	PSHX						3C	X _L → M _{sp} , SP - 1 → SP
Pull Data	PULX						38	X _H → M _{sp} , SP - 1 → SP
Exchange	XGDX						18	SP + 1 → SP, M _{sp} → X _H
								ACCD ↔ IX

Table 3 Jump, Branch Instructions

Operations	Mnemonic	Addressing Modes					Branch Test	Condition Code Register
		RELATIVE	DIRECT	INDEX	EXTEND	IMPLIED		
OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #	OP ~ #
Branch Always	BRA	20	3 2				None	• • • • •
Branch Never	BRN	21	3 2				None	• • • • •
Branch If Carry Clear	BCC	24	3 2				C = 0	• • • • •
Branch If Carry Set	BCS	25	3 2				C = 1	• • • • •
Branch If = Zero	BEQ	27	3 2				Z = 1	• • • • •
Branch If > Zero	BGE	2C	3 2				N ⊕ V = 0	• • • • •
Branch If > Zero	BGT	2E	3 2				Z + (N ⊕ V) = 0	• • • • •
Branch If Higher	BHI	22	3 2				C + Z = 0	• • • • •
Branch If < Zero	BLE	2F	3 2				Z + (N ⊕ V) = 1	• • • • •
Branch If Lower Or Same	BLS	23	3 2				C + Z = 1	• • • • •
Branch If < Zero	BLT	2D	3 2				N ⊕ V = 1	• • • • •
Branch If Minus	BMI	2B	3 2				N = 1	• • • • •
Branch If Not Equal Zero	BNE	26	3 2				Z = 0	• • • • •
Branch If Overflow Clear	BVC	28	3 2				V = 0	• • • • •
Branch If Overflow Set	BVS	29	3 2				V = 1	• • • • •
Branch If Plus	BPL	2A	3 2				N = 0	• • • • •
Branch To Subroutine	BSR	8D	5 2					• • • • •
Jump	JMP			6E	3 2	7E	3 3	
Jump To Subroutine	JSR		9D	5 2	AD	5 2	BD	6 3
No Operation	NOP					01	1 1	Advances Prog. Cntr. Only
Return From Interrupt	RTI					3B	10 1	⑧
Return From Subroutine	RTS					39	5 1	
Software Interrupt	SWI					3F	12 1	
Wait for Interrupt*	WAI					3E	9 1	⑨
Sleep	SLP					1A	4 1	

Table 4 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes					Condition Code Register	Condition Code Register
		IMPLIED						
		OP ~ #	~	#			OP ~ #	OP ~ #
Clear Carry	CLC	0C	1	1			0 → C	• • • • •
Clear Interrupt Mask	CLI	0E	1	1			0 → I	• R • • • •
Clear Overflow	CLV	0A	1	1			0 → V	• • • • • R •
Set Carry	SEC	0D	1	1			1 → C	• • • • • S •
Set Interrupt Mask	SEI	0F	1	1			1 → I	• S • • • •
Set Overflow	SEV	0B	1	1			1 → V	• • • • • S •
Accumulator A → CCR	TAP	06	1	1			A → CCR	⑩
CCR → Accumulator A	TPA	07	1	1			CCR → A	• • • • • •

Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

- ① (Bit V) Test: Result = 10000000?
- ② (Bit C) Test: Result ≠ 00000000?
- ③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
- ④ (Bit V) Test: Operand = 10000000 prior to execution?
- ⑤ (Bit V) Test: Operand = 01111111 prior to execution?
- ⑥ (Bit V) Test: Set equal to N⊕C=1 after the execution of instructions
- ⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
- ⑧ (All) Load Condition Code Register from Stack
- ⑨ (Bit I) Set when interrupt occurs. If previously set, a NMI is required to exit the wait
- ⑩ (All Bit) Set according to the contents of Accumulator A.
- ⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

