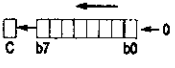
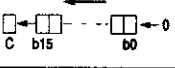
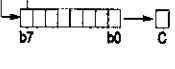


Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				
ABA	Add Accumulators	$A + B \rightarrow A$	INH	1B		1	2	2-1	--↑↑↑↑
ABX	Add B to X	$IX + 00:B \rightarrow IX$	INH	3A		1	3	2-2	-----
ABY	Add B to Y	$IY + 00:B \rightarrow IY$	INH	18 3A		2	4	2-4	-----
ADCA (opr)	Add with Carry to A	$A + M + C \rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	89 ii 99 dd B9 hh II A9 ff 18 A9 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	--↑↑↑↑
ADCB (opr)	Add with Carry to B	$B + M + C \rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C9 ii D9 dd F9 hh II E9 ff 18 E9 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	--↑↑↑↑
ADDA (opr)	Add Memory to A	$A + M \rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	8B ii 9B dd BB hh II AB ff 18 AB ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	--↑↑↑↑
ADDB (opr)	Add Memory to B	$B + M \rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	CB ii DB dd FB hh II EB ff 18 EB ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	--↑↑↑↑
ADDD (opr)	Add 16-Bit to D	$D + M:M + 1 \rightarrow D$	IMM DIR EXT IND,X IND,Y	C3 jj kk D3 dd F3 hh II E3 ff 18 E3 ff		3 2 3 2 3	4 5 6 6 7	3-3 4-7 5-10 6-10 7-8	----↑↑↑↑
ANDA (opr)	AND A with Memory	$A \cdot M \rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	84 ii 94 dd B4 hh II A4 ff 18 A4 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑0-
ANDB (opr)	AND B with Memory	$B \cdot M \rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C4 ii D4 dd F4 hh II E4 ff 18 E4 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑0-
ASL (opr)	Arithmetic Shift Left		EXT IND,X IND,Y	78 hh II 68 ff 18 68 ff		3 2 3	6 6 7	5-8 6-3 7-3	----↑↑↑↑
ASLA			A INH	48		1	2	2-1	
ASLB			B INH	58		1	2	2-1	
ASLD	Arithmetic Shift Left Double		INH	05		1	3	2-2	----↑↑↑↑
ASR (opr)	Arithmetic Shift Right		EXT IND,X IND,Y	77 hh II 67 ff 18 67 ff		3 2 3	6 6 7	5-8 6-3 7-3	----↑↑↑↑
ASRA			A INH	47		1	2	2-1	
ASRB			B INH	57		1	2	2-1	
BCC (rel)	Branch if Carry Clear	$? C = 0$	REL	24 rr		2	3	8-1	-----
BCLR (opr) (msk)	Clear Bit(s)	$M \cdot (mm) \rightarrow M$	DIR IND,X IND,Y	15 dd mm 1D ff mm 18 1D ff mm		3 3 4	6 7 8	4-10 6-13 7-10	----↑↑0-
BCS (rel)	Branch if Carry Set	$? C = 1$	REL	25 rr		2	3	8-1	-----
BEQ (rel)	Branch if = Zero	$? Z = 1$	REL	27 rr		2	3	8-1	-----
BGE (rel)	Branch if ≥ Zero	$? N \oplus V = 0$	REL	2C rr		2	3	8-1	-----
BGT (rel)	Branch if > Zero	$? Z + (N \oplus V) = 0$	REL	2E rr		2	3	8-1	-----
BH! (rel)	Branch if Higher	$? C + Z = 0$	REL	22 rr		2	3	8-1	-----
BHS (rel)	Branch if Higher or Same	$? C = 0$	REL	24 rr		2	3	8-1	-----

*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

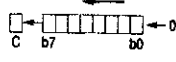
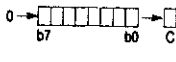
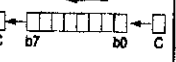
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Byte	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				
BITA (opr)	Bit(s) Test A with Memory	A-M	A IMM A DIR A EXT A IND,X A IND,Y	85 ll 95 dd B5 hh ll A5 ff 18 A5 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	---- \uparrow 10-
BITB (opr)	Bit(s) Test B with Memory	B-M	B IMM B DIR B EXT B IND,X B IND,Y	C5 ll D5 dd F5 hh ll E5 ff 18 E5 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	---- \uparrow 10-
BLE (rel)	Branch if \leq Zero	? Z + (N \oplus V) = 1	REL	2F rr		2	3	8-1	-----
BLO (rel)	Branch if Lower	? C = 1	REL	25 rr		2	3	8-1	-----
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23 rr		2	3	8-1	-----
BLT (rel)	Branch if < Zero	? N \oplus V = 1	REL	2D rr		2	3	8-1	-----
BMI (rel)	Branch if Minus	? N = 1	REL	2B rr		2	3	8-1	-----
BNE (rel)	Branch if Not = Zero	? Z = 0	REL	26 rr		2	3	8-1	-----
BPL (rel)	Branch if Plus	? N = 0	REL	2A rr		2	3	8-1	-----
BRA (rel)	Branch Always	? 1 = 1	REL	20 rr		2	3	8-1	-----
BRCLR(opr) (msk) (rel)	Branch if Bit(s) Clear	? M • mm = 0	DIR IND,X IND,Y	13 dd mm rr 1F ff mm rr 18 1F ff mm rr		4 4 5	6 7 8	4-11 6-14 7-11	-----
BRN (rel)	Branch Never	? 1 = 0	REL	21 rr		2	3	8-1	-----
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? (M) • mm = 0	DIR IND,X IND,Y	12 dd mm rr 1E ff mm rr 18 1E ff mm rr		4 4 5	6 7 8	4-11 6-14 7-11	-----
BSET(opr) (msk)	Set Bit(s)	M + mm \rightarrow M	DIR IND,X IND,Y	14 dd mm 1C ff mm 18 1C ff mm		3 3 4	6 7 8	4-10 6-13 7-10	---- \uparrow 10-
BSR (rel)	Branch to Subroutine	See Special Ops	REL	8D rr		2	6	8-2	-----
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28 rr		2	3	8-1	-----
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29 rr		2	3	8-1	-----
CBA	Compare A to B	A - B	INH	11		1	2	2-1	---- \uparrow \uparrow \uparrow \uparrow
CLC	Clear Carry Bit	0 \rightarrow C	INH	0C		1	2	2-1	-----0
CLI	Clear Interrupt Mask	0 \rightarrow I	INH	0E		1	2	2-1	---0---
CLR (opr)	Clear Memory Byte	0 \rightarrow M	EXT IND,X IND,Y	7F hh ll 6F ff 18 6F ff		3 2 3	6 6 7	5-8 6-3 7-3	----0100
CLRA	Clear Accumulator A	0 \rightarrow A	A INH	4F		1	2	2-1	----0100
CLRB	Clear Accumulator B	0 \rightarrow B	B INH	5F		1	2	2-1	----0100
CLV	Clear Overflow Flag	0 \rightarrow V	INH	0A		1	2	2-1	-----0-
CPMA (opr)	Compare A to Memory	A - M	A IMM A DIR A EXT A IND,X A IND,Y	81 ll 91 dd B1 hh ll A1 ff 18 A1 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	---- \uparrow \uparrow \uparrow
CMPB (opr)	Compare B to Memory	B - M	B IMM B DIR B EXT B IND,X B IND,Y	C1 ll D1 dd F1 hh ll E1 ff 18 E1 ff		2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	---- \uparrow \uparrow \uparrow
COM (opr)	1's Complement Memory Byte	\$FF - M \rightarrow M	EXT IND,X IND,Y	73 hh ll 63 ff 18 63 ff		3 2 3	6 6 7	5-8 6-3 7-3	---- \uparrow 101
COMA	1's Complement A	\$FF - A \rightarrow A	A INH	43		1	2	2-1	---- \uparrow 101
COMB	1's Complement B	\$FF - B \rightarrow B	B INH	53		1	2	2-1	---- \uparrow 101
CPD (opr)	Compare D to Memory 16-Bit	D - M: M + 1	IMM DIR EXT IND,X IND,Y	1A 83 ll kk 1A 93 dd 1A B3 hh ll 1A A3 ff CD A3 ff		4 3 4 3 3	5 6 7 7 7	3-5 4-9 5-11 6-11 7-8	---- \uparrow \uparrow \uparrow

*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

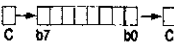
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				
CPX (opr)	Compare X to Memory 16-Bit	$IX - M:M + 1$	IMM DIR EXT IND,X IND,Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff	3 2 3 2 3	4 5 6 6 7	3-3 4-7 5-10 6-10 7-8	----↑↑↑↑
CPY (opr)	Compare Y to Memory 16-Bit	$IY - M:M + 1$	IMM DIR EXT IND,X IND,Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd hh ll ff ff	4 3 4 3 3	5 6 7 7 7	3-5 4-9 5-11 6-11 7-8	----↑↑↑↑
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19		1	2	2-1	----↑↑↑↑
DEC (opr)	Decrement Memory Byte	$M - 1 \rightarrow M$	EXT IND,X IND,Y	7A 6A 18 6A	hh ll ff ff	3 2 3	6 6 7	5-8 6-3 7-3	----↑↑↑-
DECA	Decrement Accumulator A	$A - 1 \rightarrow A$	A INH	4A		1	2	2-1	----↑↑↑-
DECB	Decrement Accumulator B	$B - 1 \rightarrow B$	B INH	5A		1	2	2-1	----↑↑↑-
DES	Decrement Stack Pointer	$SP - 1 \rightarrow SP$	INH	3A		1	3	2-3	-----
DEX	Decrement Index Register X	$IX - 1 \rightarrow IX$	INH	09		1	3	2-2	-----↑--
DEY	Decrement Index Register Y	$IY - 1 \rightarrow IY$	INH	18 09		2	4	2-4	-----↑--
EORA (opr)	Exclusive OR A with Memory	$A \oplus M \rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	88 98 88 A8 18 A8	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑0-
EORB (opr)	Exclusive OR B with Memory	$B \oplus M \rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C8 D8 F8 E8 18 E8	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑0-
FDIV	Fractional Divide 16 by 16	$D/IX \rightarrow IX; r \rightarrow D$	INH	03		1	41	2-17	-----↑↑↑
IDIV	Integer Divide 16 by 16	$D/IX \rightarrow IX; r \rightarrow D$	INH	02		1	41	2-17	-----↑0↑
INC (opr)	Increment Memory Byte	$M + 1 \rightarrow M$	EXT IND,X IND,Y	7C 6C 18 6C	hh ll ff ff	3 2 3	6 6 7	5-8 6-3 7-3	----↑↑↑-
INCA	Increment Accumulator A	$A + 1 \rightarrow A$	A INH	4C		1	2	2-1	----↑↑↑-
INCB	Increment Accumulator B	$B + 1 \rightarrow B$	B INH	5C		1	2	2-1	----↑↑↑-
INS	Increment Stack Pointer	$SP + 1 \rightarrow SP$	INH	31		1	3	2-3	-----
INX	Increment Index Register X	$IX + 1 \rightarrow IX$	INH	08		1	3	2-2	-----↑--
INY	Increment Index Register Y	$IY + 1 \rightarrow IY$	INH	18 08		2	4	2-4	-----↑--
JMP (opr)	Jump	See Special Ops	EXT IND,X IND,Y	7E 6E 18 6E	hh ll ff ff	3 2 3	3 3 4	5-1 6-1 7-1	-----
JSR (opr)	Jump to Subroutine	See Special Ops	DIR EXT IND,X IND,Y	9D BD AD 18 AD	dd hh ll ff ff	2 3 2 3	5 6 6 7	4-8 5-12 6-12 7-9	-----
LDAA (opr)	Load Accumulator A	$M \rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑0-
LDAB (opr)	Load Accumulator B	$M \rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C6 D6 F6 E6 18 E6	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑0-
LDD (opr)	Load Double Accumulator D	$M \rightarrow A, M + 1 \rightarrow B$	IMM DIR EXT IND,X IND,Y	CC DC FC EC 18 EC	jj kk dd hh ll ff ff	3 2 3 2 3	3 4 5 5 6	3-2 4-3 5-4 6-6 7-6	----↑↑0-

*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

(continued)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycles	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				
LDS (opr)	Load Stack Pointer	$M:M + 1 \rightarrow SP$	IMM DIR EXT IND,X IND,Y	8E 9E BE AE 18 AE	jj kk dd hh ll ff ff	3 2 3 2 3	3 4 5 5 6	3-2 4-3 5-4 6-6 7-6	--- $\uparrow\downarrow 0$ -
LDX (opr)	Load Index Register X	$M:M + 1 \rightarrow IX$	IMM DIR EXT IND,X IND,Y	CE DE FE EE CD EE	jj kk dd hh ll ff ff	3 2 3 2 3	3 4 5 5 6	3-2 4-3 5-4 6-6 7-6	--- $\uparrow\downarrow 0$ -
LDY (opr)	Load Index Register Y	$M:M + 1 \rightarrow IY$	IMM DIR EXT IND,X IND,Y	18 CE 18 DE 18 FE 1A EE 18 EE	jj kk dd hh ll ff ff	4 3 4 3 3	4 5 6 6 6	3-4 4-5 5-6 6-7 7-6	--- $\uparrow\downarrow 0$ -
LSL (opr)	Logical Shift Left		EXT IND,X IND,Y A INH B INH	78 68 18 68 48 58	hh ll ff ff ff ff	3 2 3 1 1	6 6 7 2 2	5-8 6-3 3-7 2-1 2-1	--- $\uparrow\uparrow\uparrow\uparrow$
LSLA									
LSLB									
LSLD									
LSR (opr)	Logical Shift Right		EXT IND,X IND,Y A INH B INH	74 64 18 64 44 54	hh ll ff ff ff ff	3 2 3 1 1	6 6 7 2 2	5-8 6-3 7-3 2-1 2-1	--- $\uparrow\uparrow\uparrow\uparrow$
LSRA									
LSRB									
LSRD									
MUL	Multiply 8 by 8	$A \times B \rightarrow D$	INH	3D		1	10	2-13	----- \uparrow
NEG (opr)	2's Complement Memory Byte	$0 - M \rightarrow M$	EXT IND,X IND,Y	70 60 18 60	hh ll ff ff	3 2 3	6 6 7	5-8 6-3 7-3	--- $\uparrow\uparrow\uparrow\uparrow$
NEGA	2's Complement A	$0 - A \rightarrow A$	A INH	40		1	2	2-1	--- $\uparrow\uparrow\uparrow\uparrow$
NEGB	2's Complement B	$0 - B \rightarrow B$	B INH	50		1	2	2-1	--- $\uparrow\uparrow\uparrow\uparrow$
NOP	No Operation	No Operation	INH	01		1	2	2-1	-----
ORAA (opr)	OR Accumulator A (Inclusive)	$A + M \rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	8A 9A BA AA 18 AA	jj dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	--- $\uparrow\downarrow 0$ -
ORAB (opr)	OR Accumulator B (Inclusive)	$B + M \rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	CA DA FA EA 18 EA	jj dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	--- $\uparrow\downarrow 0$ -
PSHA	Push A onto Stack	$A \rightarrow Stk, SP = SP - 1$	A INH	36		1	3	2-6	-----
PSHB	Push B onto Stack	$B \rightarrow Stk, SP = SP - 1$	B INH	37		1	3	2-6	-----
PSHX	Push X onto Stack (Lo First)	$IX \rightarrow Stk, SP = SP - 2$	INH	3C		1	4	2-7	-----
PSHY	Push Y onto Stack (Lo First)	$IY \rightarrow Stk, SP = SP - 2$	INH	18 3C		2	5	2-8	-----
PULA	Pull A from Stack	$SP = SP + 1, A \leftarrow Stk$	A INH	32		1	4	2-9	-----
PULB	Pull B from Stack	$SP = SP + 1, B \leftarrow Stk$	B INH	33		1	4	2-9	-----
PULX	Pull X from Stack (Hi First)	$SP = SP + 2, IX \leftarrow Stk$	INH	38		1	5	2-10	-----
PULY	Pull Y from Stack (Hi First)	$SP = SP + 2, IY \leftarrow Stk$	INH	18 38		2	6	2-11	-----
ROL (opr)	Rotate Left		EXT IND,X IND,Y A INH B INH	79 69 18 69 49 59	hh ll ff ff ff ff	3 2 3 1 1	6 6 7 2 2	5-8 6-3 7-3 2-1 2-1	--- $\uparrow\uparrow\uparrow\uparrow$
ROLA									
ROLB									
ROLB									

*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				
ROR (opr)	Rotate Right		EXT	76 hh ll	3	6	5-8	----↑↑↑↑	
RORA			IND,X	66 ff	2	6	6-3		
RORB			IND,Y	18 66 ff	3	7	7-3		
			A INH	46	1	2	2-1		
		B INH	56	1	2	2-1			
RTI	Return from Interrupt	See Special Ops	INH	3B		1	12	2-14	↑↓↑↑↑↑↑↑
RTS	Return from Subroutine	See Special Ops	INH	39		1	5	2-12	-----
SBA	Subtract B from A	A - B → A	INH	10		1	2	2-1	----↑↑↑↑
SBCA (opr)	Subtract with Carry from A	A - M - C → A	A IMM A DIR A EXT A IND,X A IND,Y	82 ii 92 dd B2 hh ll A2 ff 18 A2 ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑↑↑	
SBCB (opr)	Subtract with Carry from B	B - M - C → B	B IMM B DIR B EXT B IND,X B IND,Y	C2 ii D2 dd F2 hh ll E2 ff 18 E2 ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑↑↑	
SEC	Set Carry	1 → C	INH	OD		1	2	2-1	-----1
SEI	Set Interrupt Mask	1 → I	INH	OF		1	2	2-1	---1----
SEV	Set Overflow Flag	1 → V	INH	OB		1	2	2-1	-----1-
STAA (opr)	Store Accumulator A	A → M	A DIR A EXT A IND,X A IND,Y	97 dd B7 hh ll A7 ff 18 A7 ff	2 3 2 3	3 4 4 5	4-2 5-3 6-5 7-5	----↑↑0-	
STAB (opr)	Store Accumulator B	B → M	B DIR B EXT B IND,X B IND,Y	D7 dd F7 hh ll E7 ff 18 E7 ff	2 3 2 3	3 4 4 5	4-2 5-3 6-5 7-5	----↑↑0-	
STD (opr)	Store Accumulator D	A → M, B → M + 1	DIR EXT IND,X IND,Y	DD dd FD hh ll ED ff 18 ED ff	2 3 2 3	4 5 5 6	4-4 5-5 6-8 7-7	----↑↑0-	
STOP	Stop Internal Clocks		INH	CF		1	2	2-1	-----
STS (opr)	Store Stack Pointer	SP → M:M + 1	DIR EXT IND,X IND,Y	9F dd BF hh ll AF ff 18 AF ff	2 3 2 3	4 5 5 6	4-4 5-5 6-8 7-7	----↑↑0-	
STX (opr)	Store Index Register X	IX → M:M + 1	DIR EXT IND,X IND,Y	DF dd FF hh ll EF ff CD EF ff	2 3 2 3	4 5 5 6	4-4 5-5 6-8 7-7	----↑↑0-	
STY (opr)	Store Index Register Y	IY → M:M + 1	DIR EXT IND,X IND,Y	18 DF dd 18 FF hh ll 1A EF ff 18 EF ff	3 4 3 3	5 6 6 6	4-6 5-7 6-9 7-7	----↑↑0-	
SUBA (opr)	Subtract Memory from A	A - M → A	A IMM A DIR A EXT A IND,X A IND,Y	80 ii 90 dd B0 hh ll A0 ff 18 A0 ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑↑↑	
SUBB (opr)	Subtract Memory from B	B - M → B	B IMM B DIR B EXT B IND,X B IND,Y	C0 ii D0 dd F0 hh ll E0 ff 18 E0 ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	----↑↑↑↑	
SUBD (opr)	Subtract Memory from D	D - M:M + 1 → D	IMM DIR EXT IND,X IND,Y	83 ll kk 93 dd B3 hh ll A3 ff 18 A3 ff	3 2 3 2 3	4 5 6 6 7	3-3 4-7 5-10 6-10 7-8	----↑↑↑↑	
SWI	Software Interrupt	See Special Ops	INH	3F		1	14	2-15	---1---
TAB	Transfer A to B	A → B	INH	16		1	2	2-1	----↑↑0-
TAP	Transfer A to CC Register	A → CCR	INH	06		1	2	2-1	↑↓↑↑↑↑↑↑
TBA	Transfer B to A	B → A	INH	17		1	2	2-1	----↑↑0-

*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes S X H I N Z V C
				Opcode	Operand(s)				
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00		1	**	2-20	-----
TPA	Transfer CC Register to A	CCR → A	INH	07		1	2	2-1	-----
TST (opr)	Test for Zero or Minus	M = 0	EXT	7D	hh ll	3	6	5-9	----↑↑00
			IND,X	6D	ff	2	6	6-4	
			IND,Y	18 6D	ff	3	7	7-4	
TSTA		A = 0	A INH	4D		1	2	2-1	----↑↑00
TSTB		B = 0	B INH	5D		1	2	2-1	----↑↑00
TSX	Transfer Stack Pointer to X	SP + 1 → IX	INH	30		1	3	2-3	-----
TSY	Transfer Stack Pointer to Y	SP + 1 → IY	INH	18 30		2	4	2-5	-----
TXS	Transfer X to Stack Pointer	IX - 1 → SP	INH	35		1	3	2-2	-----
TYS	Transfer Y to Stack Pointer	IY - 1 → SP	INH	18 35		2	4	2-4	-----
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E		1	***	2-16	-----
XGDY	Exchange D with X	IX → D, D → IX	INH	8F		1	3	2-2	-----
XGDY	Exchange D with Y	IY → D, D → IY	INH	18 8F		2	4	2-4	-----

*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

**Infinity or Until Reset Occurs

***12 Cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-clock cycles (n) until an interrupt is recognized. Finally, two additional cycles are used to fetch the appropriate interrupt vector (14 + n total).

dd = 8-Bit Direct Address (\$0000 - \$00FF) (High Byte Assumed to be \$00)
ff = 8-Bit Positive Offset \$00 (0) to \$FF (255) (Is Added to Index)
hh = High Order Byte of 16-Bit Extended Address
ll = One Byte of Immediate Data
jj = High Order Byte of 16-Bit Immediate Data
kk = Low Order Byte of 16-Bit Immediate Data
ll = Low Order Byte of 16-Bit Extended Address
mm = 8-Bit Bit Mask (Set Bits to be Affected)
rr = Signed Relative Offset \$80 (- 128) to \$7F (+ 127)
(Offset Relative to the Address Following the Machine Code Offset Byte)