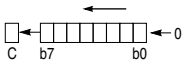
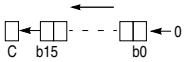
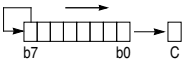
Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 1 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
ABA	Add Accumulators	A + B → A	INH	1B		1	2	2-1	- - ↕ - ↕ ↕ ↕ ↕
ABX	Add B to X	IX + 00:B → IX	INH	3A		1	3	2-2	- - - - - - - -
ABY	Add B to Y	IY + 00:B → IY	INH	18 3A		2	4	2-4	- - - - - - - -
ADCA (opr)	Add with Carry to A	A + M + C → A	A IMM	89	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			A DIR	99	dd	2	3	4-1	
			A EXT	B9	hh ll	3	4	5-2	
			A IND,X	A9	ff	2	4	6-2	
			A IND,Y	18 A9	ff	3	5	7-2	
ADCB (opr)	Add with Carry to B	B + M + C → B	B IMM	C9	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			B DIR	D9	dd	2	3	4-1	
			B EXT	F9	hh ll	3	4	5-2	
			B IND,X	E9	ff	2	4	6-2	
			B IND,Y	18 E9	ff	3	5	7-2	
ADDA (opr)	Add Memory to A	A + M → A	A IMM	8B	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			A DIR	9B	dd	2	3	4-1	
			A EXT	BB	hh ll	3	4	5-2	
			A IND,X	AB	ff	2	4	6-2	
			A IND,Y	18 AB	ff	3	5	7-2	
ADDB (opr)	Add Memory to B	B + M → B	B IMM	CB	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			B DIR	DB	dd	2	3	4-1	
			B EXT	FB	hh ll	3	4	5-2	
			B IND,X	EB	ff	2	4	6-2	
			B IND,Y	18 EB	ff	3	5	7-2	
ADDD (opr)	Add 16-Bit to D	D + M:M + 1 → D	IMM	C3	jj kk	3	4	3-3	- - - - ↕ ↕ ↕ ↕
			DIR	D3	dd	2	5	4-7	
			EXT	F3	hh ll	3	6	5-10	
			IND,X	E3	ff	2	6	6-10	
			IND,Y	18 E3	ff	3	7	7-8	
ANDA (opr)	AND A with Memory	A•M → A	A IMM	84	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	94	dd	2	3	4-1	
			A EXT	B4	hh ll	3	4	5-2	
			A IND,X	A4	ff	2	4	6-2	
			A IND,Y	18 A4	ff	3	5	7-2	
ANDB (opr)	AND B with Memory	B•M → B	B IMM	C4	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D4	dd	2	3	4-1	
			B EXT	F4	hh ll	3	4	5-2	
			B IND,X	E4	ff	2	4	6-2	
			B IND,Y	18 E4	ff	3	5	7-2	
ASL (opr)	Arithmetic Shift Left		EXT	78	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	68	ff	2	6	6-3	
ASLA			IND,Y	18 68	ff	3	7	7-3	
			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	77	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	67	ff	2	6	6-3	
ASRA			IND,Y	18 67	ff	3	7	7-3	
			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•($\overline{mm}$) → M	DIR	15	dd mm	3	6	4-10	- - - - ↕ ↕ 0 -
			IND,X	1D	ff mm	3	7	6-13	
			IND,Y	18 1D	ff mm	4	8	7-10	
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 ⊕ V = 0	REL	2C	rr	2	3	8-1	- - - - - - - -
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	2	3	8-1	- - - - - - - -
BHI (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.

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 2 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
BITA (opr)	Bit(s) Test A with Memory	A•M	A IMM	85	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	95	dd	2	3	4-1	
			A EXT	B5	hh ll	3	4	5-2	
			A IND,X	A5	ff	2	4	6-2	
			A IND,Y	18 A5	ff	3	5	7-2	
BITB (opr)	Bit(s) Test B with Memory	B•M	B IMM	C5	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D5	dd	2	3	4-1	
			B EXT	F5	hh ll	3	4	5-2	
			B IND,X	E5	ff	2	4	6-2	
			B IND,Y	18 E5	ff	3	5	7-2	
BLE (rel)	Branch if ≤ Zero	? Z + (N ⊕ 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 ⊕ 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	13	dd mm rr	4	6	4-11	- - - - - - - -
			IND,X	1F	ff mm rr	4	7	6-14	
			IND,Y	18 1F	ff mm rr	5	8	7-11	
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	2	3	8-1	- - - - - - - -
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? ($\overline{M}$) • mm = 0	DIR	12	dd mm rr	4	6	4-11	- - - - - - - -
			IND,X	1E	ff mm rr	4	7	6-14 7-	
			IND,Y	18 1E	ff mm rr	5	8	11	
BSET(opr) (msk)	Set Bit(s)	M + mm → M	DIR	14	dd mm	3	6	4-10	- - - - ↕ ↕ 0 -
			IND,X	1C	ff mm	3	7	6-13	
			IND,Y	18 1C	ff mm	4	8	7-10	
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	- - - - ↕ ↕ ↕ ↕
CLC	Clear Carry Bit	0 → C	INH	0C		1	2	2-1	- - - - - - - 0
CLI	Clear Interrupt Mask	0 → I	INH	0E		1	2	2-1	- - - 0 - - - -
CLR (opr)	Clear Memory Byte	0 → M	EXT	7F	hh ll	3	6	5-8	- - - - 0 1 0 0
			IND,X	6F	ff	2	6	6-3	
			IND,Y	18 6F	ff	3	7	7-3	
CLRA	Clear Accumulator A	0 → A	A INH	4F		1	2	2-1	- - - - 0 1 0 0
CLRB	Clear Accumulator B	0 → B	B INH	5F		1	2	2-1	- - - - 0 1 0 0
CLV	Clear Overflow Flag	0 → V	INH	0A		1	2	2-1	- - - - - - 0 -
CMPA (opr)	Compare A to Memory	A – M	A IMM	81	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			A DIR	91	dd	2	3	4-1	
			A EXT	B1	hh ll	3	4	5-2	
			A IND,X	A1	ff	2	4	6-2	
			A IND,Y	18 A1	ff	3	5	7-2	
CMPB (opr)	Compare B to Memory	B – M	B IMM	C1	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			B DIR	D1	dd	2	3	4-1	
			B EXT	F1	hh ll	3	4	5-2	
			B IND,X	E1	ff	2	4	6-2	
			B IND,Y	18 E1	ff	3	5	7-2	
COM (opr)	1's Complement Memory Byte	$FF – M → M	EXT	73	hh ll	3	6	5-8	- - - - ↕ ↕ 0 1
			IND,X	63	ff	2	6	6-3	
			IND,Y	18 63	ff	3	7	7-3	
COMA	1's Complement A	$FF – A → A	A INH	43		1	2	2-1	- - - - ↕ ↕ 0 1
COMB	1's Complement B	$FF – B → B	B INH	53		1	2	2-1	- - - - ↕ ↕ 0 1
CPD (opr)	Compare D to Memory 16-Bit	D – M:M + 1	IMM	1A 83	jj kk	4	5	3-5	- - - - ↕ ↕ ↕ ↕
			DIR	1A 93	dd	3	6	4-9	
			EXT	1A B3	hh ll	4	7	5-11	
			IND,X	1A A3	ff	3	7	6-11	
			IND,Y	CD A3	ff	3	7	7-8	

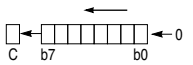
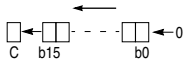
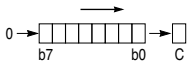
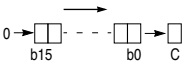
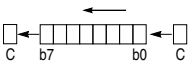
*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 3 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
CPX (opr)	Compare X to Memory 16-Bit	IX – M:M + 1	IMM	8C	jj kk	3	4	3-3	- - - - ↕ ↕ ↕ ↕
			DIR	9C	dd	2	5	4-7	
			EXT	BC	hh ll	3	6	5-10	
			IND,X	AC	ff	2	6	6-10	
			IND,Y	CD AC	ff	3	7	7-8	
CPY (opr)	Compare Y to Memory 16-Bit	IY – M:M + 1	IMM	18 8C	jj kk	4	5	3-5	- - - - ↕ ↕ ↕ ↕
			DIR	18 9C	dd	3	6	4-9	
			EXT	18 BC	hh ll	4	7	5-11	
			IND,X	1A AC	ff	3	7	6-11	
			IND,Y	18 AC	ff	3	7	7-8	
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19		1	2	2-1	- - - - ↕ ↕ ↕ ↕
DEC (opr)	Decrement Memory Byte	M – 1 → M	EXT	7A	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ -
			IND,X	6A	ff	2	6	6-3	
			IND,Y	18 6A	ff	3	7	7-3	
DECA	Decrement Accumulator A	A – 1 → A	A INH	4A		1	2	2-1	- - - - ↕ ↕ ↕ -
DECB	Decrement Accumulator B	B – 1 → B	B INH	5A		1	2	2-1	- - - - ↕ ↕ ↕ -
DES	Decrement Stack Pointer	SP – 1 → SP	INH	34		1	3	2-3	- - - - - - - -
DEX	Decrement Index Register X	IX – 1 → IX	INH	09		1	3	2-2	- - - - - ↕ - -
DEY	Decrement Index Register Y	IY – 1 → IY	INH	18 09		2	4	2-4	- - - - - ↕ - -
EORA (opr)	Exclusive OR A with Memory	A ⊕ M → A	A IMM	88	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	98	dd	2	3	4-1	
			A EXT	88	hh ll	3	4	5-2	
			A IND,X	A8	ff	2	4	6-2	
			A IND,Y	18 A8	ff	3	5	7-2	
EORB (opr)	Exclusive OR B with Memory	B ⊕ M → B	B IMM	C8	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D8	dd	2	3	4-1	
			B EXT	F8	hh ll	3	4	5-2	
			B IND,X	E8	ff	2	4	6-2	
			B IND,Y	18 E8	ff	3	5	7-2	
FDIV	Fractional Divide 16 by 16	D/IX → IX; r → D	INH	03		1	41	2-17	- - - - - ↕ ↕ ↕
IDIV	Integer Divide 16 by 16	D/IX → IX; r → D	INH	02		1	41	2-17	- - - - - ↕ 0 ↕
INC (opr)	Increment Memory Byte	M + 1 → M	EXT	7C	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ -
			IND,X	6C	ff	2	6	6-3	
			IND,Y	18 6C	ff	3	7	7-3	
INCA	Increment Accumulator A	A + 1 → A	A INH	4C		1	2	2-1	- - - - ↕ ↕ ↕ -
INCB	Increment Accumulator B	B + 1 → B	B INH	5C		1	2	2-1	- - - - ↕ ↕ ↕ -
INS	Increment Stack Pointer	SP + 1 → SP	INH	31		1	3	2-3	- - - - - - - -
INX	Increment Index Register X	IX + 1 → IX	INH	08		1	3	2-2	- - - - - ↕ - -
INY	Increment Index Register Y	IY + 1 → IY	INH	18 08		2	4	2-4	- - - - - ↕ - -
JMP (opr)	Jump	See Special Ops	EXT	7E	hh ll	3	3	5-1	- - - - - - - -
			IND,X	6E	ff	2	3	6-1	
			IND,Y	18 6E	ff	3	4	7-1	
JSR (opr)	Jump to Subroutine	See Special Ops	DIR	9D	dd	2	5	4-8	- - - - - - - -
			EXT	BD	hh ll	3	6	5-12	
			IND,X	AD	ff	2	6	6-12	
			IND,Y	18 AD	ff	3	7	7-9	
LDAA (opr)	Load Accumulator A	M → A	A IMM	86	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	96	dd	2	3	4-1	
			A EXT	B6	hh ll	3	4	5-2	
			A IND,X	A6	ff	2	4	6-2	
			A IND,Y	18 A6	ff	3	5	7-2	
LDAB (opr)	Load Accumulator B	M → B	B IMM	C6	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D6	dd	2	3	4-1	
			B EXT	F6	hh ll	3	4	5-2	
			B IND,X	E6	ff	2	4	6-2	
			B IND,Y	18 E6	ff	3	5	7-2	
LDD (opr)	Load Double Accumulator D	M → A,M + 1 → B	IMM	CC	jj kk	3	3	3-2	- - - - ↕ ↕ 0 -
			DIR	DC	dd	2	4	4-3	
			EXT	FC	hh ll	3	5	5-4	
			IND,X	EC	ff	2	5	6-6	
			IND,Y	18 EC	ff	3	6	7-6	

*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.

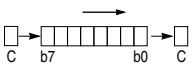
Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 4 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
LDS (opr)	Load Stack Pointer	M:M + 1 → SP	IMM	8E	jj kk	3	3	3-2	- - - - ↕ ↕ 0 -
			DIR	9E	dd	2	4	4-3	
			EXT	BE	hh ll	3	5	5-4	
			IND,X	AE	ff	2	5	6-6	
			IND,Y	18 AE	ff	3	6	7-6	
LDX (opr)	Load Index Register X	M:M + 1 → IX	IMM	CE	jj kk	3	3	3-2	- - - - ↕ ↕ 0 -
			DIR	DE	dd	2	4	4-3	
			EXT	FE	hh ll	3	5	5-4	
			IND,X	EE	ff	2	5	6-6	
			IND,Y	CD EE	ff	3	6	7-6	
LDY (opr)	Load Index Register Y	M:M + 1 → IY	IMM	18 CE	jj kk	4	4	3-4	- - - - ↕ ↕ 0 -
			DIR	18 DE	dd	3	5	4-5	
			EXT	18 FE	hh ll	4	6	5-6	
			IND,X	1A EE	ff	3	6	6-7	
			IND,Y	18 EE	ff	3	6	7-6	
LSL (opr) LSLA LSLB	Logical Shift Left		EXT	78	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	68	ff	2	6	6-3	
			IND,Y	18 68	ff	3	7	3-7	
			A INH	48		1	2	2-1	
			B INH	58		1	2	2-1	
LSLD	Logical Shift Left Double		INH	05		1	3	2-2	- - - - ↕ ↕ ↕ ↕
LSR (opr) LSRA LSRB	Logical Shift Right		EXT	74	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	64	ff	2	6	6-3	
			IND,Y	18 64	ff	3	7	7-3	
			A INH	44		1	2	2-1	
			B INH	54		1	2	2-1	
LSRD	Logical Shift Right Double		INH	04		1	3	2-2	- - - - 0 ↕ ↕ ↕
MUL	Multiply 8 by 8	AxB → D	INH	3D		1	10	2-13	- - - - - - - ↕
NEG (opr)	2's Complement Memory Byte	0 – M → M	EXT	70	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	60	ff	2	6	6-3	
			IND,Y	18 60	ff	3	7	7-3	
NEGA	2's Complement A	0 – A → A	A INH	40		1	2	2-1	- - - - ↕ ↕ ↕ ↕
NEGB	2's Complement B	0 – B → B	B INH	50		1	2	2-1	- - - - ↕ ↕ ↕ ↕
NOP	No Operation	No Operation	INH	01		1	2	2-1	- - - - - - - -
ORAA (opr)	OR Accumulator A (Inclusive)	A + M → A	A IMM	8A	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	9A	dd	2	3	4-1	
			A EXT	BA	hh ll	3	4	5-2	
			A IND,X	AA	ff	2	4	6-2	
			A IND,Y	18 AA	ff	3	5	7-2	
ORAB (opr)	OR Accumulator B (Inclusive)	B + M → B	B IMM	CA	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	DA	dd	2	3	4-1	
			B EXT	FA	hh ll	3	4	5-2	
			B IND,X	EA	ff	2	4	6-2	
			B IND,Y	18 EA	ff	3	5	7-2	
PSHA	Push A onto Stack	A → Stk, SP = SP–1	A INH	36		1	3	2-6	- - - - - - - -
PSHB	Push B onto Stack	B → Stk, SP = SP–1	B INH	37		1	3	2-6	- - - - - - - -
PSHX	Push X onto Stack (Lo First)	IX → Stk, SP = SP–2	INH	3C		1	4	2-7	- - - - - - - -
PSHY	Push Y onto Stack (Lo First)	IY → Stk, SP = SP–2	INH	18 3C		2	5	2-8	- - - - - - - -
PULA	Pull A from Stack	SP = SP + 1, A←Stk	A INH	32		1	4	2-9	- - - - - - - -
PULB	Pull B from Stack	SP = SP + 1, B←Stk	B INH	33		1	4	2-9	- - - - - - - -
PULX	Pull X from Stack (Hi First)	SP = SP + 2, IX←Stk	INH	38		1	5	2-10	- - - - - - - -
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY←Stk	INH	18 38		2	6	2-11	- - - - - - - -
ROL (opr) ROLA ROLB	Rotate Left		EXT	79	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	69	ff	2	6	6-3	
			IND,Y	18 69	ff	3	7	7-3	
			A INH	49		1	2	2-1	
			B INH	59		1	2	2-1	

*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.

10

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 5 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
ROR (opr)	Rotate Right		EXT	76	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	66	ff	2	6	6-3	
RORA			IND,Y	18 66	ff	3	7	7-3	
RORB			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	82	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			A DIR	92	dd	2	3	4-1	
			A EXT	B2	hh ll	3	4	5-2	
			A IND,X	A2	ff	2	4	6-2	
			A IND,Y	18 A2	ff	3	5	7-2	
SBCB (opr)	Subtract with Carry from B	B – M – C → B	B IMM	C2	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			B DIR	D2	dd	2	3	4-1	
			B EXT	F2	hh ll	3	4	5-2	
			B IND,X	E2	ff	2	4	6-2	
			B IND,Y	18 E2	ff	3	5	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	97	dd	2	3	4-2	- - - - ↕ ↕ 0 -
			A EXT	B7	hh ll	3	4	5-3	
			A IND,X	A7	ff	2	4	6-5	
			A IND,Y	18 A7	ff	3	5	7-5	
STAB (opr)	Store Accumulator B	B → M	B DIR	D7	dd	2	3	4-2	- - - - ↕ ↕ 0 -
			B EXT	F7	hh ll	3	4	5-3	
			B IND,X	E7	ff	2	4	6-5	
			B IND,Y	18 E7	ff	3	5	7-5	
STD (opr)	Store Accumulator D	A → M, B → M + 1	DIR	DD	dd	2	4	4-4	- - - - ↕ ↕ 0 -
			EXT	FD	hh ll	3	5	5-5	
			IND,X	ED	ff	2	5	6-8	
			IND,Y	18 ED	ff	3	6	7-7	
STOP	Stop Internal Clocks		INH	CF		1	2	2-1	- - - - - - - -
STS (opr)	Store Stack Pointer	SP → M:M + 1	DIR	9F	dd	2	4	4-4	- - - - ↕ ↕ 0 -
			EXT	BF	hh ll	3	5	5-5	
			IND,X	AF	ff	2	5	6-8	
			IND,Y	18 AF	ff	3	6	7-7	
STX (opr)	Store Index Register X	IX → M:M + 1	DIR	DF	dd	2	4	4-4	- - - - ↕ ↕ 0 -
			EXT	FF	hh ll	3	5	5-5	
			IND,X	EF	ff	2	5	6-8	
			IND,Y	CD EF	ff	3	6	7-7	
STY (opr)	Store Index Register Y	IY → M:M + 1	DIR	18 DF	dd	3	5	4-6	- - - - ↕ ↕ 0 -
			EXT	18 FF	hh ll	4	6	5-7	
			IND,X	1A EF	ff	3	6	6-9	
			IND,Y	18 EF	ff	3	6	7-7	
SUBA (opr)	Subtract Memory from A	A – M → A	A IMM	80	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			A DIR	90	dd	2	3	4-1	
			A EXT	B0	hh ll	3	4	5-2	
			A IND,X	A0	ff	2	4	6-2	
			A IND,Y	18 A0	ff	3	5	7-2	
SUBB (opr)	Subtract Memory from B	B – M → B	B IMM	C0	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			B DIR	D0	dd	2	3	4-1	
			B EXT	F0	hh ll	3	4	5-2	
			B IND,X	E0	ff	2	4	6-2	
			B IND,Y	18 E0	ff	3	5	7-2	
SUBD (opr)	Subtract Memory from D	D – M:M + 1→ D	IMM	83	jj kk	3	4	3-3	- - - - ↕ ↕ ↕ ↕
			DIR	93	dd	2	5	4-7	
			EXT	B3	hh ll	3	6	5-10	
			IND,X	A3	ff	2	6	6-10	
			IND,Y	18 A3	ff	3	7	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.

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 6 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
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 IND,X IND,Y	7D 6D 18 6D	hh ll ff ff	3 2 3	6 6 7	5-9 6-4 7-4	- - - - ↕ ↕ 0 0
TSTA		A – 0	A INH	4D		1	2	2-1	- - - - ↕ ↕ 0 0
TSTB		B – 0	B INH	5D		1	2	2-1	- - - - ↕ ↕ 0 0
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	- - - - - - - -
XGDX	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	- - - - - - - -

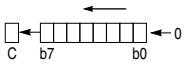
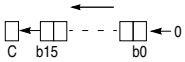
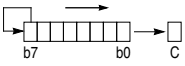
*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

**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
ii = 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)

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 1 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
ABA	Add Accumulators	A + B → A	INH	1B		1	2	2-1	- - ↕ - ↕ ↕ ↕ ↕
ABX	Add B to X	IX + 00:B → IX	INH	3A		1	3	2-2	- - - - - - - -
ABY	Add B to Y	IY + 00:B → IY	INH	18 3A		2	4	2-4	- - - - - - - -
ADCA (opr)	Add with Carry to A	A + M + C → A	A IMM	89	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			A DIR	99	dd	2	3	4-1	
			A EXT	B9	hh ll	3	4	5-2	
			A IND,X	A9	ff	2	4	6-2	
			A IND,Y	18 A9	ff	3	5	7-2	
ADCB (opr)	Add with Carry to B	B + M + C → B	B IMM	C9	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			B DIR	D9	dd	2	3	4-1	
			B EXT	F9	hh ll	3	4	5-2	
			B IND,X	E9	ff	2	4	6-2	
			B IND,Y	18 E9	ff	3	5	7-2	
ADDA (opr)	Add Memory to A	A + M → A	A IMM	8B	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			A DIR	9B	dd	2	3	4-1	
			A EXT	BB	hh ll	3	4	5-2	
			A IND,X	AB	ff	2	4	6-2	
			A IND,Y	18 AB	ff	3	5	7-2	
ADDB (opr)	Add Memory to B	B + M → B	B IMM	CB	ii	2	2	3-1	- - ↕ - ↕ ↕ ↕ ↕
			B DIR	DB	dd	2	3	4-1	
			B EXT	FB	hh ll	3	4	5-2	
			B IND,X	EB	ff	2	4	6-2	
			B IND,Y	18 EB	ff	3	5	7-2	
ADDD (opr)	Add 16-Bit to D	D + M:M + 1 → D	IMM	C3	jj kk	3	4	3-3	- - - - ↕ ↕ ↕ ↕
			DIR	D3	dd	2	5	4-7	
			EXT	F3	hh ll	3	6	5-10	
			IND,X	E3	ff	2	6	6-10	
			IND,Y	18 E3	ff	3	7	7-8	
ANDA (opr)	AND A with Memory	A•M → A	A IMM	84	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	94	dd	2	3	4-1	
			A EXT	B4	hh ll	3	4	5-2	
			A IND,X	A4	ff	2	4	6-2	
			A IND,Y	18 A4	ff	3	5	7-2	
ANDB (opr)	AND B with Memory	B•M → B	B IMM	C4	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D4	dd	2	3	4-1	
			B EXT	F4	hh ll	3	4	5-2	
			B IND,X	E4	ff	2	4	6-2	
			B IND,Y	18 E4	ff	3	5	7-2	
ASL (opr)	Arithmetic Shift Left		EXT	78	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	68	ff	2	6	6-3	
ASLA			IND,Y	18 68	ff	3	7	7-3	
			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	77	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	67	ff	2	6	6-3	
ASRA			IND,Y	18 67	ff	3	7	7-3	
			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•($\overline{mm}$) → M	DIR	15	dd mm	3	6	4-10	- - - - ↕ ↕ 0 -
			IND,X	1D	ff mm	3	7	6-13	
			IND,Y	18 1D	ff mm	4	8	7-10	
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 ⊕ V = 0	REL	2C	rr	2	3	8-1	- - - - - - - -
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	2	3	8-1	- - - - - - - -
BHI (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.

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 2 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
BITA (opr)	Bit(s) Test A with Memory	A•M	A IMM	85	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	95	dd	2	3	4-1	
			A EXT	B5	hh ll	3	4	5-2	
			A IND,X	A5	ff	2	4	6-2	
			A IND,Y	18 A5	ff	3	5	7-2	
BITB (opr)	Bit(s) Test B with Memory	B•M	B IMM	C5	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D5	dd	2	3	4-1	
			B EXT	F5	hh ll	3	4	5-2	
			B IND,X	E5	ff	2	4	6-2	
			B IND,Y	18 E5	ff	3	5	7-2	
BLE (rel)	Branch if ≤ Zero	? Z + (N ⊕ 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 ⊕ 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	13	dd mm rr	4	6	4-11	- - - - - - - -
			IND,X	1F	ff mm rr	4	7	6-14	
			IND,Y	18 1F	ff mm rr	5	8	7-11	
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	2	3	8-1	- - - - - - - -
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? ($\overline{M}$) • mm = 0	DIR	12	dd mm rr	4	6	4-11	- - - - - - - -
			IND,X	1E	ff mm rr	4	7	6-14 7-	
			IND,Y	18 1E	ff mm rr	5	8	11	
BSET(opr) (msk)	Set Bit(s)	M + mm → M	DIR	14	dd mm	3	6	4-10	- - - - ↕ ↕ 0 -
			IND,X	1C	ff mm	3	7	6-13	
			IND,Y	18 1C	ff mm	4	8	7-10	
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	- - - - ↕ ↕ ↕ ↕
CLC	Clear Carry Bit	0 → C	INH	0C		1	2	2-1	- - - - - - - 0
CLI	Clear Interrupt Mask	0 → I	INH	0E		1	2	2-1	- - - 0 - - - -
CLR (opr)	Clear Memory Byte	0 → M	EXT	7F	hh ll	3	6	5-8	- - - - 0 1 0 0
			IND,X	6F	ff	2	6	6-3	
			IND,Y	18 6F	ff	3	7	7-3	
CLRA	Clear Accumulator A	0 → A	A INH	4F		1	2	2-1	- - - - 0 1 0 0
CLRB	Clear Accumulator B	0 → B	B INH	5F		1	2	2-1	- - - - 0 1 0 0
CLV	Clear Overflow Flag	0 → V	INH	0A		1	2	2-1	- - - - - - 0 -
CMPA (opr)	Compare A to Memory	A – M	A IMM	81	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			A DIR	91	dd	2	3	4-1	
			A EXT	B1	hh ll	3	4	5-2	
			A IND,X	A1	ff	2	4	6-2	
			A IND,Y	18 A1	ff	3	5	7-2	
CMPB (opr)	Compare B to Memory	B – M	B IMM	C1	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			B DIR	D1	dd	2	3	4-1	
			B EXT	F1	hh ll	3	4	5-2	
			B IND,X	E1	ff	2	4	6-2	
			B IND,Y	18 E1	ff	3	5	7-2	
COM (opr)	1's Complement Memory Byte	$FF – M → M	EXT	73	hh ll	3	6	5-8	- - - - ↕ ↕ 0 1
			IND,X	63	ff	2	6	6-3	
			IND,Y	18 63	ff	3	7	7-3	
COMA	1's Complement A	$FF – A → A	A INH	43		1	2	2-1	- - - - ↕ ↕ 0 1
COMB	1's Complement B	$FF – B → B	B INH	53		1	2	2-1	- - - - ↕ ↕ 0 1
CPD (opr)	Compare D to Memory 16-Bit	D – M:M + 1	IMM	1A 83	jj kk	4	5	3-5	- - - - ↕ ↕ ↕ ↕
			DIR	1A 93	dd	3	6	4-9	
			EXT	1A B3	hh ll	4	7	5-11	
			IND,X	1A A3	ff	3	7	6-11	
			IND,Y	CD A3	ff	3	7	7-8	

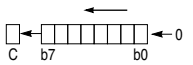
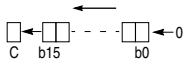
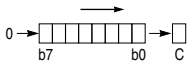
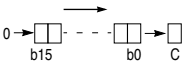
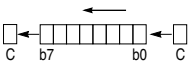
*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation.
Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 3 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
CPX (opr)	Compare X to Memory 16-Bit	IX – M:M + 1	IMM	8C	jj kk	3	4	3-3	- - - - ↕ ↕ ↕ ↕
			DIR	9C	dd	2	5	4-7	
			EXT	BC	hh ll	3	6	5-10	
			IND,X	AC	ff	2	6	6-10	
			IND,Y	CD AC	ff	3	7	7-8	
CPY (opr)	Compare Y to Memory 16-Bit	IY – M:M + 1	IMM	18 8C	jj kk	4	5	3-5	- - - - ↕ ↕ ↕ ↕
			DIR	18 9C	dd	3	6	4-9	
			EXT	18 BC	hh ll	4	7	5-11	
			IND,X	1A AC	ff	3	7	6-11	
			IND,Y	18 AC	ff	3	7	7-8	
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19		1	2	2-1	- - - - ↕ ↕ ↕ ↕
DEC (opr)	Decrement Memory Byte	M – 1 → M	EXT	7A	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ -
			IND,X	6A	ff	2	6	6-3	
			IND,Y	18 6A	ff	3	7	7-3	
DECA	Decrement Accumulator A	A – 1 → A	A INH	4A		1	2	2-1	- - - - ↕ ↕ ↕ -
DECB	Decrement Accumulator B	B – 1 → B	B INH	5A		1	2	2-1	- - - - ↕ ↕ ↕ -
DES	Decrement Stack Pointer	SP – 1 → SP	INH	34		1	3	2-3	- - - - - - - -
DEX	Decrement Index Register X	IX – 1 → IX	INH	09		1	3	2-2	- - - - - ↕ - -
DEY	Decrement Index Register Y	IY – 1 → IY	INH	18 09		2	4	2-4	- - - - - ↕ - -
EORA (opr)	Exclusive OR A with Memory	A ⊕ M → A	A IMM	88	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	98	dd	2	3	4-1	
			A EXT	88	hh ll	3	4	5-2	
			A IND,X	A8	ff	2	4	6-2	
			A IND,Y	18 A8	ff	3	5	7-2	
EORB (opr)	Exclusive OR B with Memory	B ⊕ M → B	B IMM	C8	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D8	dd	2	3	4-1	
			B EXT	F8	hh ll	3	4	5-2	
			B IND,X	E8	ff	2	4	6-2	
			B IND,Y	18 E8	ff	3	5	7-2	
FDIV	Fractional Divide 16 by 16	D/IX → IX; r → D	INH	03		1	41	2-17	- - - - - ↕ ↕ ↕
IDIV	Integer Divide 16 by 16	D/IX → IX; r → D	INH	02		1	41	2-17	- - - - - ↕ 0 ↕
INC (opr)	Increment Memory Byte	M + 1 → M	EXT	7C	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ -
			IND,X	6C	ff	2	6	6-3	
			IND,Y	18 6C	ff	3	7	7-3	
INCA	Increment Accumulator A	A + 1 → A	A INH	4C		1	2	2-1	- - - - ↕ ↕ ↕ -
INCB	Increment Accumulator B	B + 1 → B	B INH	5C		1	2	2-1	- - - - ↕ ↕ ↕ -
INS	Increment Stack Pointer	SP + 1 → SP	INH	31		1	3	2-3	- - - - - - - -
INX	Increment Index Register X	IX + 1 → IX	INH	08		1	3	2-2	- - - - - ↕ - -
INY	Increment Index Register Y	IY + 1 → IY	INH	18 08		2	4	2-4	- - - - - ↕ - -
JMP (opr)	Jump	See Special Ops	EXT	7E	hh ll	3	3	5-1	- - - - - - - -
			IND,X	6E	ff	2	3	6-1	
			IND,Y	18 6E	ff	3	4	7-1	
JSR (opr)	Jump to Subroutine	See Special Ops	DIR	9D	dd	2	5	4-8	- - - - - - - -
			EXT	BD	hh ll	3	6	5-12	
			IND,X	AD	ff	2	6	6-12	
			IND,Y	18 AD	ff	3	7	7-9	
LDAA (opr)	Load Accumulator A	M → A	A IMM	86	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	96	dd	2	3	4-1	
			A EXT	B6	hh ll	3	4	5-2	
			A IND,X	A6	ff	2	4	6-2	
			A IND,Y	18 A6	ff	3	5	7-2	
LDAB (opr)	Load Accumulator B	M → B	B IMM	C6	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	D6	dd	2	3	4-1	
			B EXT	F6	hh ll	3	4	5-2	
			B IND,X	E6	ff	2	4	6-2	
			B IND,Y	18 E6	ff	3	5	7-2	
LDD (opr)	Load Double Accumulator D	M → A,M + 1 → B	IMM	CC	jj kk	3	3	3-2	- - - - ↕ ↕ 0 -
			DIR	DC	dd	2	4	4-3	
			EXT	FC	hh ll	3	5	5-4	
			IND,X	EC	ff	2	5	6-6	
			IND,Y	18 EC	ff	3	6	7-6	

*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.

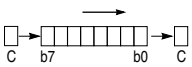
Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 4 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
LDS (opr)	Load Stack Pointer	M:M + 1 → SP	IMM	8E	jj kk	3	3	3-2	- - - - ↕ ↕ 0 -
			DIR	9E	dd	2	4	4-3	
			EXT	BE	hh ll	3	5	5-4	
			IND,X	AE	ff	2	5	6-6	
			IND,Y	18 AE	ff	3	6	7-6	
LDX (opr)	Load Index Register X	M:M + 1 → IX	IMM	CE	jj kk	3	3	3-2	- - - - ↕ ↕ 0 -
			DIR	DE	dd	2	4	4-3	
			EXT	FE	hh ll	3	5	5-4	
			IND,X	EE	ff	2	5	6-6	
			IND,Y	CD EE	ff	3	6	7-6	
LDY (opr)	Load Index Register Y	M:M + 1 → IY	IMM	18 CE	jj kk	4	4	3-4	- - - - ↕ ↕ 0 -
			DIR	18 DE	dd	3	5	4-5	
			EXT	18 FE	hh ll	4	6	5-6	
			IND,X	1A EE	ff	3	6	6-7	
			IND,Y	18 EE	ff	3	6	7-6	
LSL (opr) LSLA LSLB	Logical Shift Left		EXT	78	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	68	ff	2	6	6-3	
			IND,Y	18 68	ff	3	7	3-7	
			A INH	48		1	2	2-1	
			B INH	58		1	2	2-1	
LSLD	Logical Shift Left Double		INH	05		1	3	2-2	- - - - ↕ ↕ ↕ ↕
LSR (opr) LSRA LSRB	Logical Shift Right		EXT	74	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	64	ff	2	6	6-3	
			IND,Y	18 64	ff	3	7	7-3	
			A INH	44		1	2	2-1	
			B INH	54		1	2	2-1	
LSRD	Logical Shift Right Double		INH	04		1	3	2-2	- - - - 0 ↕ ↕ ↕
MUL	Multiply 8 by 8	AxB → D	INH	3D		1	10	2-13	- - - - - - - ↕
NEG (opr)	2's Complement Memory Byte	0 – M → M	EXT	70	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	60	ff	2	6	6-3	
			IND,Y	18 60	ff	3	7	7-3	
NEGA	2's Complement A	0 – A → A	A INH	40		1	2	2-1	- - - - ↕ ↕ ↕ ↕
NEGB	2's Complement B	0 – B → B	B INH	50		1	2	2-1	- - - - ↕ ↕ ↕ ↕
NOP	No Operation	No Operation	INH	01		1	2	2-1	- - - - - - - -
ORAA (opr)	OR Accumulator A (Inclusive)	A + M → A	A IMM	8A	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			A DIR	9A	dd	2	3	4-1	
			A EXT	BA	hh ll	3	4	5-2	
			A IND,X	AA	ff	2	4	6-2	
			A IND,Y	18 AA	ff	3	5	7-2	
ORAB (opr)	OR Accumulator B (Inclusive)	B + M → B	B IMM	CA	ii	2	2	3-1	- - - - ↕ ↕ 0 -
			B DIR	DA	dd	2	3	4-1	
			B EXT	FA	hh ll	3	4	5-2	
			B IND,X	EA	ff	2	4	6-2	
			B IND,Y	18 EA	ff	3	5	7-2	
PSHA	Push A onto Stack	A → Stk, SP = SP–1	A INH	36		1	3	2-6	- - - - - - - -
PSHB	Push B onto Stack	B → Stk, SP = SP–1	B INH	37		1	3	2-6	- - - - - - - -
PSHX	Push X onto Stack (Lo First)	IX → Stk, SP = SP–2	INH	3C		1	4	2-7	- - - - - - - -
PSHY	Push Y onto Stack (Lo First)	IY → Stk, SP = SP–2	INH	18 3C		2	5	2-8	- - - - - - - -
PULA	Pull A from Stack	SP = SP + 1, A←Stk	A INH	32		1	4	2-9	- - - - - - - -
PULB	Pull B from Stack	SP = SP + 1, B←Stk	B INH	33		1	4	2-9	- - - - - - - -
PULX	Pull X from Stack (Hi First)	SP = SP + 2, IX←Stk	INH	38		1	5	2-10	- - - - - - - -
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY←Stk	INH	18 38		2	6	2-11	- - - - - - - -
ROL (opr) ROLA ROLB	Rotate Left		EXT	79	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	69	ff	2	6	6-3	
			IND,Y	18 69	ff	3	7	7-3	
			A INH	49		1	2	2-1	
			B INH	59		1	2	2-1	

*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.

10

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 5 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
ROR (opr)	Rotate Right		EXT	76	hh ll	3	6	5-8	- - - - ↕ ↕ ↕ ↕
			IND,X	66	ff	2	6	6-3	
RORA			IND,Y	18 66	ff	3	7	7-3	
RORB			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	82	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			A DIR	92	dd	2	3	4-1	
			A EXT	B2	hh ll	3	4	5-2	
			A IND,X	A2	ff	2	4	6-2	
			A IND,Y	18 A2	ff	3	5	7-2	
SBCB (opr)	Subtract with Carry from B	B – M – C → B	B IMM	C2	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			B DIR	D2	dd	2	3	4-1	
			B EXT	F2	hh ll	3	4	5-2	
			B IND,X	E2	ff	2	4	6-2	
			B IND,Y	18 E2	ff	3	5	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	97	dd	2	3	4-2	- - - - ↕ ↕ 0 -
			A EXT	B7	hh ll	3	4	5-3	
			A IND,X	A7	ff	2	4	6-5	
			A IND,Y	18 A7	ff	3	5	7-5	
STAB (opr)	Store Accumulator B	B → M	B DIR	D7	dd	2	3	4-2	- - - - ↕ ↕ 0 -
			B EXT	F7	hh ll	3	4	5-3	
			B IND,X	E7	ff	2	4	6-5	
			B IND,Y	18 E7	ff	3	5	7-5	
STD (opr)	Store Accumulator D	A → M, B → M + 1	DIR	DD	dd	2	4	4-4	- - - - ↕ ↕ 0 -
			EXT	FD	hh ll	3	5	5-5	
			IND,X	ED	ff	2	5	6-8	
			IND,Y	18 ED	ff	3	6	7-7	
STOP	Stop Internal Clocks		INH	CF		1	2	2-1	- - - - - - - -
STS (opr)	Store Stack Pointer	SP → M:M + 1	DIR	9F	dd	2	4	4-4	- - - - ↕ ↕ 0 -
			EXT	BF	hh ll	3	5	5-5	
			IND,X	AF	ff	2	5	6-8	
			IND,Y	18 AF	ff	3	6	7-7	
STX (opr)	Store Index Register X	IX → M:M + 1	DIR	DF	dd	2	4	4-4	- - - - ↕ ↕ 0 -
			EXT	FF	hh ll	3	5	5-5	
			IND,X	EF	ff	2	5	6-8	
			IND,Y	CD EF	ff	3	6	7-7	
STY (opr)	Store Index Register Y	IY → M:M + 1	DIR	18 DF	dd	3	5	4-6	- - - - ↕ ↕ 0 -
			EXT	18 FF	hh ll	4	6	5-7	
			IND,X	1A EF	ff	3	6	6-9	
			IND,Y	18 EF	ff	3	6	7-7	
SUBA (opr)	Subtract Memory from A	A – M → A	A IMM	80	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			A DIR	90	dd	2	3	4-1	
			A EXT	B0	hh ll	3	4	5-2	
			A IND,X	A0	ff	2	4	6-2	
			A IND,Y	18 A0	ff	3	5	7-2	
SUBB (opr)	Subtract Memory from B	B – M → B	B IMM	C0	ii	2	2	3-1	- - - - ↕ ↕ ↕ ↕
			B DIR	D0	dd	2	3	4-1	
			B EXT	F0	hh ll	3	4	5-2	
			B IND,X	E0	ff	2	4	6-2	
			B IND,Y	18 E0	ff	3	5	7-2	
SUBD (opr)	Subtract Memory from D	D – M:M + 1→ D	IMM	83	jj kk	3	4	3-3	- - - - ↕ ↕ ↕ ↕
			DIR	93	dd	2	5	4-7	
			EXT	B3	hh ll	3	6	5-10	
			IND,X	A3	ff	2	6	6-10	
			IND,Y	18 A3	ff	3	7	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.

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 6 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
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 IND,X IND,Y	7D 6D 18 6D	hh ll ff ff	3 2 3	6 6 7	5-9 6-4 7-4	- - - - ↕ ↕ 0 0
TSTA		A – 0	A INH	4D		1	2	2-1	- - - - ↕ ↕ 0 0
TSTB		B – 0	B INH	5D		1	2	2-1	- - - - ↕ ↕ 0 0
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	- - - - - - - -
XGDX	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
ii = 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)