Table 3-2 Instruction Set (Sheet 1 of 6)

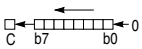
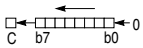
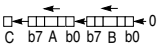
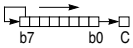
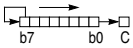
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ABA	Add Accumulators	A + B ⇒ A	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	IX + (00 : B) ⇒ IX	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	IY + (00 : B) ⇒ IY	INH	18 3A	—	4	—	—	—	—	—	—	—	—
ADCA (opr)	Add with Carry to A	A + M + C ⇒ A	A IMM	89	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			A DIR	99	dd	3								
			A EXT	B9	hh ll	4								
			A IND,X	A9	ff	4								
			A IND,Y	18 A9	ff	5								
ADCB (opr)	Add with Carry to B	B + M + C ⇒ B	B IMM	C9	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			B DIR	D9	dd	3								
			B EXT	F9	hh ll	4								
			B IND,X	E9	ff	4								
			B IND,Y	18 E9	ff	5								
ADDA (opr)	Add Memory to A	A + M ⇒ A	A IMM	8B	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			A DIR	9B	dd	3								
			A EXT	BB	hh ll	4								
			A IND,X	AB	ff	4								
			A IND,Y	18 AB	ff	5								
ADDB (opr)	Add Memory to B	B + M ⇒ B	B IMM	CB	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			B DIR	DB	dd	3								
			B EXT	FB	hh ll	4								
			B IND,X	EB	ff	4								
			B IND,Y	18 EB	ff	5								
ADDD (opr)	Add 16-Bit to D	D + (M : M + 1) ⇒ D	IMM	C3	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			DIR	D3	dd	5								
			EXT	F3	hh ll	6								
			IND,X	E3	ff	6								
			IND,Y	18 E3	ff	7								
ANDA (opr)	AND A with Memory	A • M ⇒ A	A IMM	84	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	94	dd	3								
			A EXT	B4	hh ll	4								
			A IND,X	A4	ff	4								
			A IND,Y	18 A4	ff	5								
ANDB (opr)	AND B with Memory	B • M ⇒ B	B IMM	C4	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D4	dd	3								
			B EXT	F4	hh ll	4								
			B IND,X	E4	ff	4								
			B IND,Y	18 E4	ff	5								
ASL (opr)	Arithmetic Shift Left		EXT	78	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	68	ff	6								
			IND,Y	18 68	ff	7								
ASLA	Arithmetic Shift Left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic Shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic Shift Left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic Shift Right		EXT	77	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	67	ff	6								
			IND,Y	18 67	ff	7								
ASRA	Arithmetic Shift Right A		A INH	47	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic Shift Right B		B INH	57	—	2	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	3	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear Bit(s)	M • (mm) ⇒ M	DIR	15	dd mm	6	—	—	—	—	Δ	Δ	0	—
			IND,X	1D	ff mm	7								
			IND,Y	18 1D	ff mm	8								
BCS (rel)	Branch if Carry Set	? C = 1	REL	25	rr	3	—	—	—	—	—	—	—	—
BEQ (rel)	Branch if = Zero	? Z = 1	REL	27	rr	3	—	—	—	—	—	—	—	—
BGE (rel)	Branch if Δ Zero	? N ⊕ V = 0	REL	2C	rr	3	—	—	—	—	—	—	—	—

Table 3-2 Instruction Set (Sheet 2 of 6)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	3	—	—	—	—	—	—	—	—
BHI (rel)	Branch if Higher	? C + Z = 0	REL	22	rr	3	—	—	—	—	—	—	—	—
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	rr	3	—	—	—	—	—	—	—	—
BITA (opr)	Bit(s) Test A with Memory	A • M	A IMM	85	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	95	dd	3								
			A EXT	B5	hh ll	4								
			A IND,X	A5	ff	4								
			A IND,Y	18 A5	ff	5								
BITB (opr)	Bit(s) Test B with Memory	B • M	B IMM	C5	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D5	dd	3								
			B EXT	F5	hh ll	4								
			B IND,X	E5	ff	4								
			B IND,Y	18 E5	ff	5								
BLE (rel)	Branch if Δ Zero	? Z + (N ⊕ V) = 1	REL	2F	rr	3	—	—	—	—	—	—	—	—
BLO (rel)	Branch if Lower	? C = 1	REL	25	rr	3	—	—	—	—	—	—	—	—
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23	rr	3	—	—	—	—	—	—	—	—
BLT (rel)	Branch if < Zero	? N ⊕ V = 1	REL	2D	rr	3	—	—	—	—	—	—	—	—
BMI (rel)	Branch if Minus	? N = 1	REL	2B	rr	3	—	—	—	—	—	—	—	—
BNE (rel)	Branch if not = Zero	? Z = 0	REL	26	rr	3	—	—	—	—	—	—	—	—
BPL (rel)	Branch if Plus	? N = 0	REL	2A	rr	3	—	—	—	—	—	—	—	—
BRA (rel)	Branch Always	? 1 = 1	REL	20	rr	3	—	—	—	—	—	—	—	—
BRCLR(opr) (msk) (rel)	Branch if Bit(s) Clear	? M • mm = 0	DIR	13	dd mm rr	6	—	—	—	—	—	—	—	—
			IND,X	1F	ff mm rr	7								
			IND,Y	18 1F	ff mm rr	8								
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	3	—	—	—	—	—	—	—	—
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? ($\overline{M}$) • mm = 0	DIR	12	dd mm rr	6	—	—	—	—	—	—	—	—
			IND,X	1E	ff mm rr	7								
			IND,Y	18 1E	ff mm rr	8								
BSET (opr) (msk)	Set Bit(s)	M + mm ⇒ M	DIR	14	dd mm	6	—	—	—	—	Δ	Δ	0	—
			IND,X	1C	ff mm	7								
			IND,Y	18 1C	ff mm	8								
BSR (rel)	Branch to Subroutine	See Figure 3–2	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A to B	A – B	INH	11	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLC	Clear Carry Bit	0 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear Interrupt Mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear Memory Byte	0 ⇒ M	EXT	7F	hh ll	6	—	—	—	—	0	1	0	0
			IND,X	6F	ff	6								
			IND,Y	18 6F	ff	7								
CLRA	Clear Accumulator A	0 ⇒ A	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear Accumulator B	0 ⇒ B	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear Overflow Flag	0 ⇒ V	INH	0A	—	2	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A to Memory	A – M	A IMM	81	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	91	dd	3								
			A EXT	B1	hh ll	4								
			A IND,X	A1	ff	4								
			A IND,Y	18 A1	ff	5								
CMPB (opr)	Compare B to Memory	B – M	B IMM	C1	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			B DIR	D1	dd	3								
			B EXT	F1	hh ll	4								
			B IND,X	E1	ff	4								
			B IND,Y	18 E1	ff	5								

Table 3-2 Instruction Set (Sheet 3 of 6)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
COM (opr)	Ones Complement Memory Byte	$FF – M ⇒ M	EXT IND,X IND,Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	1
COMA	Ones Complement A	$FF – A ⇒ A	A INH	43	—	2	—	—	—	—	Δ	Δ	0	1
COMB	Ones Complement B	$FF – B ⇒ B	B INH	53	—	2	—	—	—	—	Δ	Δ	0	1
CPD (opr)	Compare D to Memory 16-Bit	D – M : M + 1	IMM DIR EXT IND,X IND,Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
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	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
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	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19	—	2	—	—	—	—	Δ	Δ	Δ	Δ
DEC (opr)	Decrement Memory Byte	M – 1 ⇒ M	EXT IND,X IND,Y	7A 6A 18 6A	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
DECA	Decrement Accumulator A	A – 1 ⇒ A	A INH	4A	—	2	—	—	—	—	Δ	Δ	Δ	—
DECB	Decrement Accumulator B	B – 1 ⇒ B	B INH	5A	—	2	—	—	—	—	Δ	Δ	Δ	—
DES	Decrement Stack Pointer	SP – 1 ⇒ SP	INH	34	—	3	—	—	—	—	—	—	—	—
DEX	Decrement Index Register X	IX – 1 ⇒ IX	INH	09	—	3	—	—	—	—	—	Δ	—	—
DEY	Decrement Index Register Y	IY – 1 ⇒ IY	INH	18 09	—	4	—	—	—	—	—	Δ	—	—
EORA (opr)	Exclusive OR A with Memory	A ⊕ M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	88 98 B8 A8 18 A8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
EORB (opr)	Exclusive OR B with Memory	B ⊕ M ⇒ 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 3 4 4 5	—	—	—	—	Δ	Δ	0	—
FDIV	Fractional Divide 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	03	—	41	—	—	—	—	—	Δ	Δ	Δ
IDIV	Integer Divide 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	02	—	41	—	—	—	—	—	Δ	0	Δ
INC (opr)	Increment Memory Byte	M + 1 ⇒ M	EXT IND,X IND,Y	7C 6C 18 6C	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
INCA	Increment Accumulator A	A + 1 ⇒ A	A INH	4C	—	2	—	—	—	—	Δ	Δ	Δ	—
INCB	Increment Accumulator B	B + 1 ⇒ B	B INH	5C	—	2	—	—	—	—	Δ	Δ	Δ	—
INS	Increment Stack Pointer	SP + 1 ⇒ SP	INH	31	—	3	—	—	—	—	—	—	—	—
INX	Increment Index Register X	IX + 1 ⇒ IX	INH	08	—	3	—	—	—	—	—	Δ	—	—

Table 3-2 Instruction Set (Sheet 4 of 6)

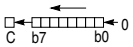
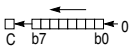
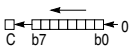
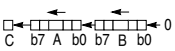
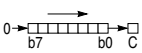
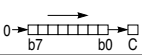
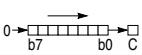
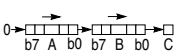
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
INY	Increment Index Register Y	IY + 1 ⇒ IY	INH	18 08	—	4	—	—	—	—	—	Δ	—	—
JMP (opr)	Jump	See Figure 3–2	EXT	7E	hh ll	3	—	—	—	—	—	—	—	—
			IND,X	6E	ff	3								
			IND,Y	18 6E	ff	4								
JSR (opr)	Jump to Subroutine	See Figure 3–2	DIR	9D	dd	5	—	—	—	—	—	—	—	—
			EXT	BD	hh ll	6								
			IND,X	AD	ff	6								
			IND,Y	18 AD	ff	7								
LDAA (opr)	Load Accumulator A	M ⇒ A	A IMM	86	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	96	dd	3								
			A EXT	B6	hh ll	4								
			A IND,X	A6	ff	4								
			A IND,Y	18 A6	ff	5								
LDAB (opr)	Load Accumulator B	M ⇒ B	B IMM	C6	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D6	dd	3								
			B EXT	F6	hh ll	4								
			B IND,X	E6	ff	4								
			B IND,Y	18 E6	ff	5								
LDD (opr)	Load Double Accumulator D	M ⇒ A,M + 1 ⇒ B	IMM	CC	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	DC	dd	4								
			EXT	FC	hh ll	5								
			IND,X	EC	ff	5								
			IND,Y	18 EC	ff	6								
LDS (opr)	Load Stack Pointer	M : M + 1 ⇒ SP	IMM	8E	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	9E	dd	4								
			EXT	BE	hh ll	5								
			IND,X	AE	ff	5								
			IND,Y	18 AE	ff	6								
LDX (opr)	Load Index Register X	M : M + 1 ⇒ IX	IMM	CE	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	DE	dd	4								
			EXT	FE	hh ll	5								
			IND,X	EE	ff	5								
			IND,Y	CD EE	ff	6								
LDY (opr)	Load Index Register Y	M : M + 1 ⇒ IY	IMM	18 CE	jj kk	4	—	—	—	—	Δ	Δ	0	—
			DIR	18 DE	dd	5								
			EXT	18 FE	hh ll	6								
			IND,X	1A EE	ff	6								
			IND,Y	18 EE	ff	6								
LSL (opr)	Logical Shift Left		EXT	78	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	68	ff	6								
			IND,Y	18 68	ff	7								
LSLA	Logical Shift Left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLB	Logical Shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLD	Logical Shift Left Double		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
LSR (opr)	Logical Shift Right		EXT	74	hh ll	6	—	—	—	—	0	Δ	Δ	Δ
			IND,X	64	ff	6								
			IND,Y	18 64	ff	7								
LSRA	Logical Shift Right A		A INH	44	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical Shift Right B		B INH	54	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical Shift Right Double		INH	04	—	3	—	—	—	—	0	Δ	Δ	Δ
MUL	Multiply 8 by 8	A * B ⇒ D	INH	3D	—	10	—	—	—	—	—	—	—	Δ
NEG (opr)	Two's Complement Memory Byte	0 – M ⇒ M	EXT	70	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	60	ff	6								
			IND,Y	18 60	ff	7								
NEGA	Two's Complement A	0 – A ⇒ A	A INH	40	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Two's Complement B	0 – B ⇒ B	B INH	50	—	2	—	—	—	—	Δ	Δ	Δ	Δ

Table 3-2 Instruction Set (Sheet 5 of 6)

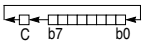
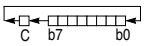
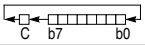
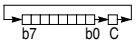
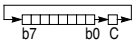
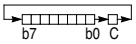
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
NOP	No operation	No Operation	INH	01	—	2	—	—	—	—	—	—	—	—
ORAA (opr)	OR Accumulator A (Inclusive)	A + M ⇒ A	A IMM	8A	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	9A	dd	3								
			A EXT	BA	hh ll	4								
			A IND,X	AA	ff	4								
			A IND,Y	18 AA	ff	5								
ORAB (opr)	OR Accumulator B (Inclusive)	B + M ⇒ B	B IMM	CA	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	DA	dd	3								
			B EXT	FA	hh ll	4								
			B IND,X	EA	ff	4								
			B IND,Y	18 EA	ff	5								
PSHA	Push A onto Stack	A ⇒ Stk,SP = SP – 1	A INH	36	—	3	—	—	—	—	—	—	—	—
PSHB	Push B onto Stack	B ⇒ Stk,SP = SP – 1	B INH	37	—	3	—	—	—	—	—	—	—	—
PSHX	Push X onto Stack (Lo First)	IX ⇒ Stk,SP = SP – 2	INH	3C	—	4	—	—	—	—	—	—	—	—
PSHY	Push Y onto Stack (Lo First)	IY ⇒ Stk,SP = SP – 2	INH	18 3C	—	5	—	—	—	—	—	—	—	—
PULA	Pull A from Stack	SP = SP + 1, A ⇐ Stk	A INH	32	—	4	—	—	—	—	—	—	—	—
PULB	Pull B from Stack	SP = SP + 1, B ⇐ Stk	B INH	33	—	4	—	—	—	—	—	—	—	—
PULX	Pull X From Stack (Hi First)	SP = SP + 2, IX ⇐ Stk	INH	38	—	5	—	—	—	—	—	—	—	—
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY ⇐ Stk	INH	18 38	—	6	—	—	—	—	—	—	—	—
ROL (opr)	Rotate Left		EXT	79	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	69	ff	6								
			IND,Y	18 69	ff	7								
ROLA	Rotate Left A		A INH	49	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate Left B		B INH	59	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROR (opr)	Rotate Right		EXT	76	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	66	ff	6								
			IND,Y	18 66	ff	7								
RORA	Rotate Right A		A INH	46	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate Right B		B INH	56	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RTI	Return from Interrupt	See Figure 3–2	INH	3B	—	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
RTS	Return from Subroutine	See Figure 3–2	INH	39	—	5	—	—	—	—	—	—	—	—
SBA	Subtract B from A	A – B ⇒ A	INH	10	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with Carry from A	A – M – C ⇒ A	A IMM	82	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	92	dd	3								
			A EXT	B2	hh ll	4								
			A IND,X	A2	ff	4								
			A IND,Y	18 A2	ff	5								
SBCB (opr)	Subtract with Carry from B	B – M – C ⇒ B	B IMM	C2	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			B DIR	D2	dd	3								
			B EXT	F2	hh ll	4								
			B IND,X	E2	ff	4								
			B IND,Y	18 E2	ff	5								
SEC	Set Carry	1 ⇒ C	INH	0D	—	2	—	—	—	—	—	—	—	1
SEI	Set Interrupt Mask	1 ⇒ I	INH	0F	—	2	—	—	—	1	—	—	—	—
SEV	Set Overflow Flag	1 ⇒ V	INH	0B	—	2	—	—	—	—	—	—	1	—
STAA (opr)	Store Accumulator A	A ⇒ M	A DIR	97	dd	3	—	—	—	—	Δ	Δ	0	—
			A EXT	B7	hh ll	4								
			A IND,X	A7	ff	4								
			A IND,Y	18 A7	ff	5								

Table 3-2 Instruction Set (Sheet 6 of 6)

Mnemonic	Operation	Description	Addressing Mode	Instruction Opcode	Operand	Cycles	Condition Codes S	X	H	I	N	Z	V	C
STAB (opr)	Store Accumulator B	B ⇒ M	B DIR	D7	dd	3	—	—	—	—	Δ	Δ	0	—
			B EXT	F7	hh ll	4								
			B IND,X	E7	ff	4								
			B IND,Y	18 E7	ff	5								
STD (opr)	Store Accumulator D	A ⇒ M, B ⇒ M + 1	DIR	DD	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	FD	hh ll	5								
			IND,X	ED	ff	5								
			IND,Y	18 ED	ff	6								
STOP	Stop Internal Clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—
STS (opr)	Store Stack Pointer	SP ⇒ M : M + 1	DIR	9F	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	BF	hh ll	5								
			IND,X	AF	ff	5								
			IND,Y	18 AF	ff	6								
STX (opr)	Store Index Register X	IX ⇒ M : M + 1	DIR	DF	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	FF	hh ll	5								
			IND,X	EF	ff	5								
			IND,Y	CD EF	ff	6								
STY (opr)	Store Index Register Y	IY ⇒ M : M + 1	DIR	18 DF	dd	5	—	—	—	—	Δ	Δ	0	—
			EXT	18 FF	hh ll	6								
			IND,X	1A EF	ff	6								
			IND,Y	18 EF	ff	6								
SUBA (opr)	Subtract Memory from A	A – M ⇒ A	A IMM	80	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	90	dd	3								
			A EXT	B0	hh ll	4								
			A IND,X	A0	ff	4								
			A IND,Y	18 A0	ff	5								
SUBB (opr)	Subtract Memory from B	B – M ⇒ B	A IMM	C0	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	D0	dd	3								
			A EXT	F0	hh ll	4								
			A IND,X	E0	ff	4								
			A IND,Y	18 E0	ff	5								
SUBD (opr)	Subtract Memory from D	D – M : M + 1 ⇒ D	IMM	83	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			DIR	93	dd	5								
			EXT	B3	hh ll	6								
			IND,X	A3	ff	6								
			IND,Y	18 A3	ff	7								
SWI	Software Interrupt	See Figure 3–2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A ⇒ B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC Register	A ⇒ CCR	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	B ⇒ A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00	—	*	—	—	—	—	—	—	—	—
TPA	Transfer CC Register to A	CCR ⇒ A	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opr)	Test for Zero or Minus	M – 0	EXT	7D	hh ll	6	—	—	—	—	Δ	Δ	0	0
			IND,X	6D	ff	6								
			IND,Y	18 6D	ff	7								
TSTA	Test A for Zero or Minus	A – 0	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for Zero or Minus	B – 0	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer Stack Pointer to X	SP + 1 ⇒ IX	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer Stack Pointer to Y	SP + 1 ⇒ IY	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to Stack Pointer	IX – 1 ⇒ SP	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to Stack Pointer	IY – 1 ⇒ SP	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E	—	**	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	IX ⇒ D, D ⇒ IX	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	IY ⇒ D, D ⇒ IY	INH	18 8F	—	4	—	—	—	—	—	—	—	—

Cycle

*	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).

Operands

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 Mask (Set Bits to be Affected)
rr	= Signed Relative Offset $80 (–128) to $7F (+127) (Offset Relative to Address Following Machine Code Offset Byte))

Operators

( )	Contents of register shown inside parentheses
⇐	Is transferred to
⇑	Is pulled from stack
⇓	Is pushed onto stack
•	Boolean AND
+	Arithmetic Addition Symbol except where used as Inclusive-OR symbol in Boolean Formula
⊕	Exclusive-OR
*	Multiply
:	Concatenation
–	Arithmetic subtraction symbol or Negation symbol (Two's Complement)

Condition Codes

—	Bit not changed
0	Bit always cleared
1	Bit always set
Δ	Bit cleared or set, depending on operation
↓	Bit can be cleared, cannot become set

SECTION 4 OPERATING MODES AND ON-CHIP MEMORY

This section contains information about the operating modes and the on-chip memory for M68HC11 E-series MCUs. Except for a few minor differences, operation is identical for all devices in the E series. Differences are noted where necessary.

4.1 Operating Modes

The values of the mode select inputs MODB and MODA during reset determine the operating mode. Single-chip and expanded multiplexed are the normal modes. In single-chip mode only on-chip memory is available. Expanded mode, however, allows access to external memory. Each of the two normal modes is paired with a special mode. Bootstrap, a variation of the single-chip mode, is a special mode that executes a bootloader program in an internal bootstrap ROM. Test is a special mode that allows privileged access to internal resources.

4.1.1 Single-Chip Mode

In single-chip mode, ports B and C and strobe pins A (STRA) and B (STRB) are available for general-purpose parallel I/O. In this mode, all software needed to control the MCU is contained in internal resources. ROM/EPROM (if present) will always be enabled out of reset, ensuring that the reset and interrupt vectors will be available at locations $FFC0–$FFFF. For the MC68HC811E2, the vector locations are the same, however, they are contained in the 2048-byte EEPROM array.

4.1.2 Expanded Mode

In expanded operating mode, the MCU can access the full 64-Kbyte address space. The space includes the same on-chip memory addresses used for single-chip mode as well as addresses for external peripherals and memory devices. The expansion bus is made up of ports B and C, and control signals AS and R/$\overline{W}$. The R/$\overline{W}$ (read/write) and AS (address strobe) allow the low-order address and the 8-bit data bus to be multiplexed on the same pins. During the first half of each bus cycle address information is present. During the second half of each bus cycle the pins become the bidirectional data bus. AS is an active-high latch enable signal for an external address latch. Address information is allowed through the transparent latch while AS is high and is latched when AS drives low.

The address, R/$\overline{W}$, and AS signals are active and valid for all bus cycles, including accesses to internal memory locations. The E clock is used to enable external devices to drive data onto the internal data bus during the second half of a read bus cycle (E clock high). R/$\overline{W}$ controls the direction of data transfers. R/$\overline{W}$ drives low when data is being written to the internal data bus. R/$\overline{W}$ will remain low during consecutive data bus write cycles, such as when a double-byte store occurs. Notice that the write enable signal for an external memory is the NAND of the E clock and the inverted R/$\overline{W}$ signal. Refer to the example diagram of address and data demultiplexing.

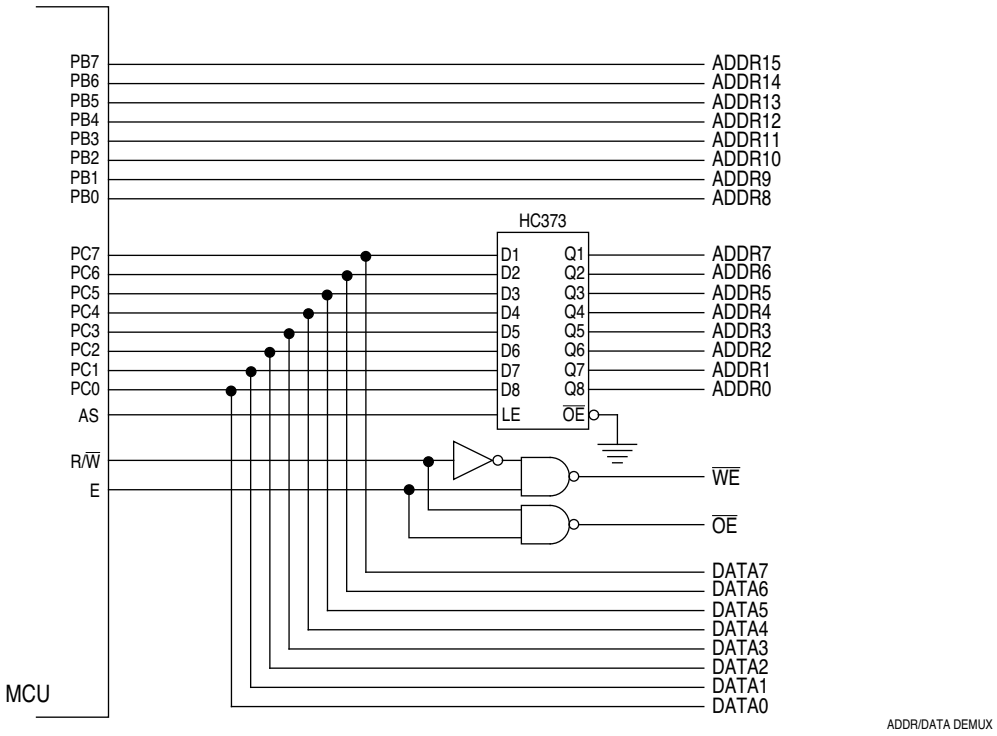


Figure 4-1 Address/Data Demultiplexing

4.1.3 Test Mode

Test mode, a variation of the expanded mode, is primarily used during Motorola's internal production testing; however, it is accessible for programming the CONFIG register, programming calibration data into EEPROM, and supporting emulation and debugging during development.

4.1.4 Bootstrap Mode

When the MCU is reset in special bootstrap mode, a small on-chip ROM is enabled at address $BF00–$BFFF. The ROM contains a bootloader program and a special set of interrupt and reset vectors. The MCU fetches the reset vector, then executes the bootloader. Bootstrap mode is a special variation of the single-chip mode. Bootstrap mode allows special-purpose programs to be entered into internal RAM. When boot mode is selected at reset, a small bootstrap ROM becomes present in the memory map. Reset and interrupt vectors are located in this ROM at $BFC0–$BFFF. The bootstrap ROM contains a small program which initializes the SCI and allows the user to download a program into on-chip RAM. The size of the downloaded program can be as large as the size of the on-chip RAM. After a four-character delay, or after receiving the character for the highest address in RAM, control passes to the loaded program at $0000. Refer to the memory map diagram.

Use of an external pull-up resistor is required when using the SCI transmitter pin because port D pins are configured for wired-OR operation by the bootloader. In boot-

strap mode, the interrupt vectors are directed to RAM. This allows the use of interrupts through a jump table. Refer to Motorola application note AN1060, M68HC11 Bootstrap Mode.

4.2 Memory Map

The operating mode determines memory mapping and whether external addresses can be accessed. Refer to **Figure 4-2**, **Figure 4-3**, and **Figure 4-4**, which illustrate the memory maps for each of the three families comprising the M68HC11 E series of MCUs. Memory locations for on-chip resources are the same for both expanded and single-chip modes. Control bits in the CONFIG register allow EPROM and EEPROM (if present) to be disabled from the memory map. The RAM is mapped to $0000 after reset. It can be placed at any 4-Kbyte boundary ($x000) by writing an appropriate value to the INIT register. The 64-byte register block is mapped to $1000 after reset and can also be placed at any 4-Kbyte boundary ($x000) by writing an appropriate value to the INIT register. If RAM and registers are mapped to the same boundary, the first 64 bytes of RAM will be inaccessible. Refer to **Table 4-1**, which details the MCU register and control bit assignments.

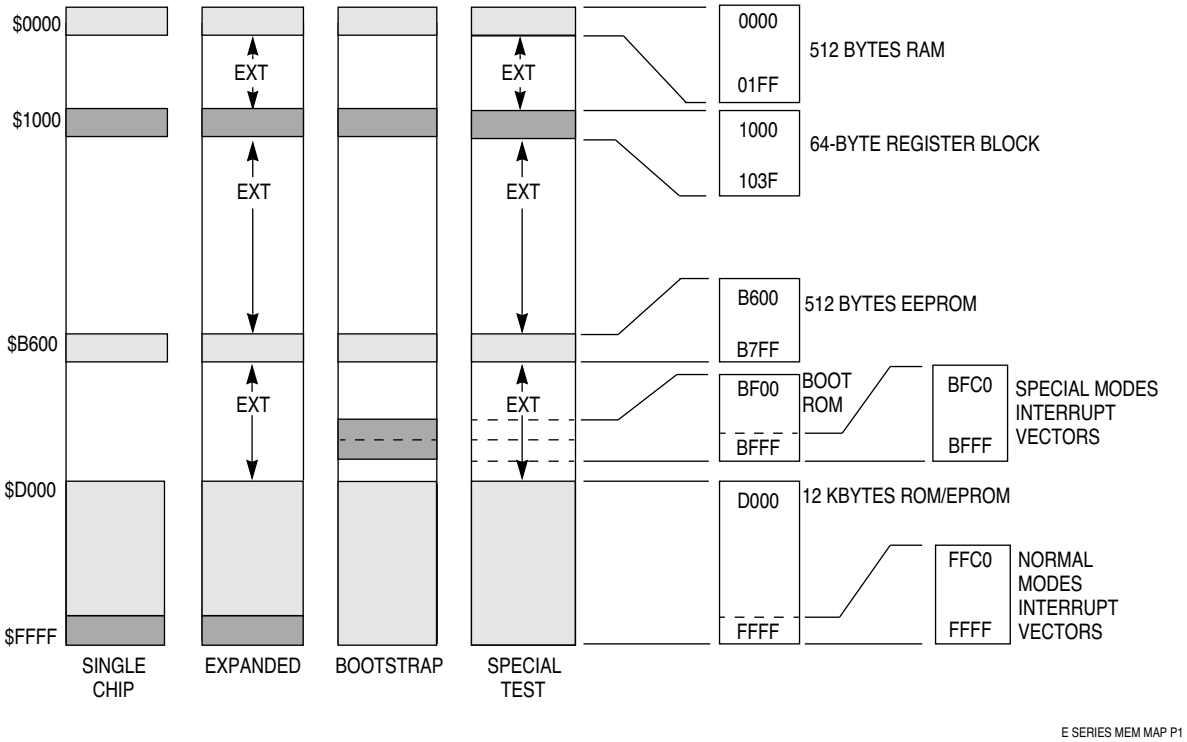


Figure 4-2 Memory Map for MC68HC11E0, MC68HC11E1, MC68HC11E8, and MC68HC(7)11E9

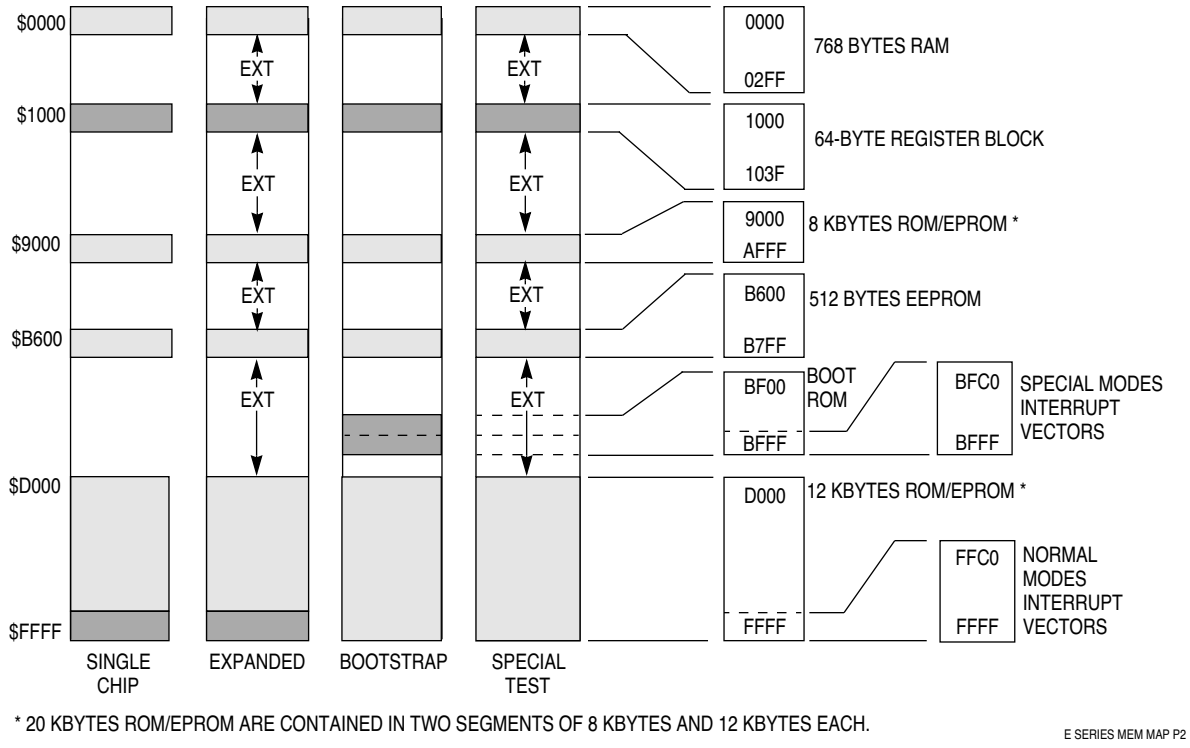


Figure 4-3 Memory Map for MC68HC(7)11E20