MC68HC11A8RG/AD

MOTOROLA

MC68HC11A8

PROGRAMMING REFERENCE GUIDE

HCMOS Single-Chip Microcomputer

BLOCK DIAGRAM

PROGRAMMING MODEL
CRYSTAL DEPENDENT TIMING
INTERRUPTS

MEMORY MAP
OPCODE MAPS

INSTRUCTIONS
ADDRESSING MODES
EXECUTION TIMES
SPECIAL OPERATIONS

REGISTER AND
CONTROL BIT
ASSIGNMENTS

MECHANICAL DATA
HEX/DEC CONVERSION
ASCII CHART

PROGRAMMING MODEL
CRYSTAL DEPENDENT TIMING
INTERRUPTS

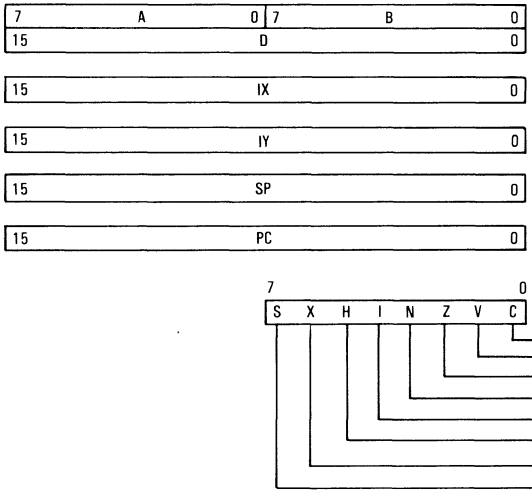
MEMORY MAP
OPCODE MAPS

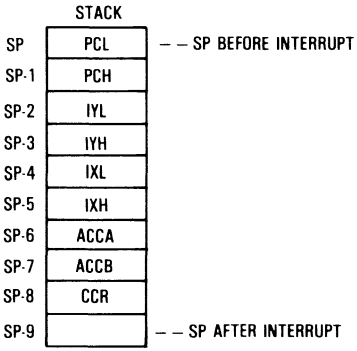
INSTRUCTIONS
ADDRESSING MODES
EXECUTION TIMES
SPECIAL OPERATIONS

REGISTER AND
CONTROL BIT
ASSIGNMENTS

MECHANICAL DATA
HEX/DEC CONVERSION
ASCII CHART

PROGRAMMING MODEL





CRYSTAL DEPENDENT TIMING SUMMARY

		Common Xtal Frequencies (others could be used)		
	Selected Crystal	2^{23} Hz 8.389 MHz	8.0 MHz	4.0 MHz
CPU Clock	**(E)**	**2.1 MHz**	**2.0 MHz**	**1.0 MHz**
Cycle Time	**(1/E)**	**477 ns**	**500 ns**	**1000 ns**
Periodic (RTI) Interrupt Rates	RTR__ 1 0			
($E/2^{13}$)	0 0	3.91 ms	4.10 ms	8.19 ms
($E/2^{14}$)	0 1	7.81 ms	8.19 ms	16.38 ms
($E/2^{15}$)	1 0	15.62 ms	16.38 ms	32.77 ms
($E/2^{16}$)	1 1	31.25 ms	32.77 ms	65.54 ms
SPI Bit Rates (baud)	SPR__ 1 0			
(E/2)	0 0	1.05 MHz	1.0 MHz	500 K
(E/4)	0 1	524 K	500 K	250 K
(E/16)	1 0	131 K	125 K	62.5 K
(E/32)	1 1	65.5 K	62.5 K	31.25 K

SCI Baud Rates	Control Bits SCP__ 1 0	Control Bits SCR__ 2 1 0	Only the max and min baud rates are shown in this table (SCR2-0 = 000 or 111). Additional rates are simple multiples of 2 from min to max.		
(E/1)	0 0		custom	-----	-----
(pre out/16)		0 0 0	131.07 K	124.80 K	62.40 K
(pre out/2048)		1 1 1	1024	976.6	488.3
(E/3)	0 1		-----	-----	-----
(pre out/16)		0 0 0	43.691 K	41.6 K	20.8 K
(pre out/2048)		1 1 1	341.3	325.5	162.8
(E/4)	1 0		custom	-----	-----
(pre out/16)		0 0 0	32.768 K	31.2 K	15.6 K
(pre out/2048)		1 1 1	256	244.1	122.1
(E/13)	1 1		-----	common	common
(pre out/16)		0 0 0	10.082 K	9600	4800
(pre out/2048)		1 1 1	78.77	75	37.56

CRYSTAL DEPENDENT TIMING SUMMARY

	Selected Crystal	Common Xtal Frequencies (others could be used)		
		2^{23} Hz 8.389 MHz	8.0 MHz	4.0 MHz
CPU Clock	**(E)**	**2.1 MHz**	**2.0 MHz**	**1.0 MHz**
Cycle Time	**(1/E)**	**477 ns**	**500 ns**	**1000 ns**
COP Watchdog Timeout Rates	CR__ 1 0			
(E/2^{15})	0 0	15.625 ms	16.384 ms	32.768 ms
(E/2^{17})	0 1	62.5 ms	65.536 ms	131.07 ms
(E/2^{19})	1 0	250 ms	262.14 ms	524.29 ms
(E/2^{21})	1 1	1 s	1.049 s	2.1 s
(E/2^{15})	Timeout Tolerance (−0 ms/+...)	15.6 ms	16.4 ms	32.8 ms
Pulse Accumulator (in gated mode)				
(E/2^{6})	1 count -	30.52 μs	32 μs	64 μs
(E/2^{14})	overflow-	7.81 ms	8.19 ms	16.38 ms

Main Timer Count Rates	PR__ 1 0			
	0 0			
(E/1)	1 count -	477 ns	500 ns	1.0 μs
(E/2^{16})	overflow-	31.25 ms	32.77 ms	65.54 ms
	0 1			
(E/4)	1 count -	1.91 μs	2.0 μs	4.0 μs
(E/2^{18})	overflow-	125 ms	131.1 ms	262.1 ms
	1 0			
(E/8)	1 count -	3.81 μs	4.0 μs	8.0 μs
(E/2^{19})	overflow-	250 ms	262.1 ms	524.3 ms
	1 1			
(E/16)	1 count -	7.63 μs	8.0 μs	16.0 μs
(E/2^{20})	overflow-	0.5 s	524.3 ms	1.049 s

Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
•	•		
•	•		
FFD4, D5	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	See Table
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Output Compare 5	I Bit	OC5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	IC3I
FFEC, ED,	Timer Input Capture 2	I Bit	IC2I
FFEE, EF	Timer Input Capture 1	I Bit	IC1I
FFF0, F1	Real Time Interrupt	I Bit	RTII
FFF2, F3	$\overline{IRQ}$ (External Pin or Parallel I/O)	I Bit	See Table
	SEE HPRIO REGISTER FOR HIGHEST PRIORITY I-BIT SOURCE		
FFF4, F5	$\overline{XIRQ}$ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	$\overline{RESET}$	None	None

LOWEST

RELATIVE PRIORITY

HIGHEST

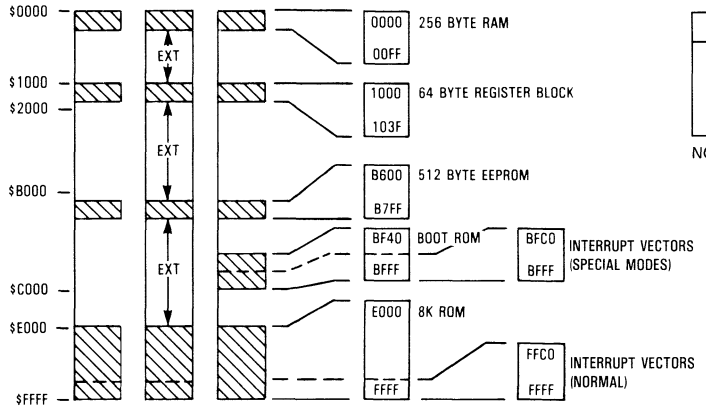
SCI Serial System Interrupts

Interrupt Cause	Local Mask
Receive Data Register Full	RIE
Receiver Overrun	RIE
Idle Line Detect	ILIE
Transmit Data Register Empty	TIE
Transmit Complete	TCIE

IRQ Vector Interrupts

Interrupt Cause	Local Mask
External Pin	None
Parallel I/O Handshake	STAI

MEMORY MAPS



MODB	MODA	Mode Selected
1	0	Single-Chip (Mode 0)
1	1	Expanded Multiplexed (Mode 1)
0	0	Special Bootstrap
0	1	Special Test

NOTE:
1 = Logic High
0 = Logic Low

NOTES:
1. Either or both the internal RAM and registers can be remapped to any 4K boundary by software.
2. Either or both the ROM and EEPROM can be disabled using a control register (CONFIG) which is implemented with EEPROM cells.

OPCODE MAP PAGE 1

DIR

		INH	INH	REL	INH	ACCA	ACCB	IND,X	EXT	A C C A IMM	A C C A DIR	A C C A IND,X	A C C A EXT	A C C B IMM	A C C B DIR	A C C B IND,X	A C C B EXT	
MSB →		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LSB ↓		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	TEST*	SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS					CMP								1
0010	2	IDIV	BRSET	BHI	PULA					SBC								2
0011	3	FDIV	BRCLR	BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD	BSET	BCC	DES	LSR				AND								4
0101	5	ASLD	BCLR	BCS	TXS					BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	PAGE 2	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	PAGE 3	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI					ADD								B
1100	C	CLC	BSET	BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC	BCLR	BLT	MUL	TST				BSR	JSR			PAGE 4	STD			D
1110	E	CLI	BRSET	BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI	BRCLR	BLE	SWI	CLR				XGDX	STS			STOP	STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

IND,X

OPCODE MAP PAGE 2 (18xx)

		INH			INH			IND,Y		A C C A IMM	A C C A DIR	A C C A IND,Y	A C C A EXT	A C C B IMM	A C C B DIR	A C C B IND,Y	A C C B EXT		
MSB → / LSB ↓		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				TSY			NEG				SUB				SUB		0	0000
0001	1											CMP				CMP		1	0001
0010	2											SBC				SBC		2	0010
0011	3							COM				SUBD				ADDD		3	0011
0100	4							LSR				AND				AND		4	0100
0101	5				TYS							BIT				BIT		5	0101
0110	6							ROR				LDA				LDA		6	0110
0111	7							ASR				STA				STA		7	0111
1000	8	INY			PULY			ASL				EOR				EOR		8	1000
1001	9	DEY						ROL				ADC				ADC		9	1001
1010	A				ABY			DEC				ORA				ORA		A	1010
1011	B											ADD				ADD		B	1011
1100	C		BSET		PSHY			INC		CPY						LDD		C	1100
1101	D		BCLR					TST				JSR				STD		D	1101
1110	E		BRSET					JMP				LDS		LDY				E	1110
1111	F		BRCLR					CLR		XGDY		STS			STY			F	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		

IND,Y

OPCODE MAP PAGE 3 (1Axx)

									A C C A				A C C B					
									IMM	DIR	IND,X	EXT			IND,X			
MSB →		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LSB ↓		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0																	0
0001	1																	1
0010	2																	2
0011	3									CPD								3
0100	4																	4
0101	5																	5
0110	6																	6
0111	7																	7
1000	8																	8
1001	9																	9
1010	A																	A
1011	B																	B
1100	C											CPY						C
1101	D																	D
1110	E															LDY		E
1111	F															STY		F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

OPCODE MAP PAGE 4 (CDxx)

											ACCA				ACCB			
													IND,Y				IND,Y	
MSB → / LSB ↓		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																	0
0001	1																	1
0010	2																	2
0011	3											CPD						3
0100	4																	4
0101	5																	5
0110	6																	6
0111	7																	7
1000	8																	8
1001	9																	9
1010	A																	A
1011	B																	B
1100	C											CPX						C
1101	D																	D
1110	E															LDX		E
1111	F															STX		F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

SIMPLE BRANCHES

Mnemonic	Opcode	Cycles
BRA	20	3
BRN	21	3
BSR	8D	7

SIMPLE CONDITIONAL BRANCHES

Test	True	Opcode	False	Opcode
N = 1	BMI	2B	BPL	2A
Z = 1	BEQ	27	BNE	26
V = 1	BVS	29	BVC	28
C = 1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES

Test	True	Opcode	False	Opcode
r > m	BGT	2E	BLE	2F
r ≥ m	BGE	2C	BLT	2D
r = m	BEQ	27	BNE	26
r ≤ m	BLE	2F	BGT	2E
r < m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES

Test	True	Opcode	False	Opcode
r > m	BHI	22	BLS	23
r ≥ m	BHS/BCC	24	BLO/BCS	25
r = m	BEQ	27	BNE	26
r ≤ m	BLS	23	BHI	22
r < m	BLO/BCS	25	BHS/BCC	24

BIT MANIPULATION BRANCHES

BRCLR — Branch if all selected bits are clear
(opcode) (operand addr) (mask) (rel offset)
M•mm = 0? M = operand in memory; mm = mask

BRSET — Branch if all selected bits are set
(opcode) (operand addr) (rel offset)
($\overline{M}$)•mm = 0? M = operand in memory; mm = mask

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
00		TEST	INH	—
01		NOP	INH	2
02		IDIV	INH	41
03		FDIV	INH	41
04		LSRD	INH	3
05		ASLD/LSLD	INH	3
06		TAP	INH	2
07		TPA	INH	2
08		INX	INH	3
09		DEX	INH	3
0A		CLV	INH	2
0B		SEV	INH	2
0C		CLC	INH	2
0D		SEC	INH	2
0E		CLI	INH	2
0F		SEI	INH	2
10		SBA	INH	2
11		CBA	INH	2
12	dd mm rr	BRSET (opr) (msk) (rel)	DIR	6
13	dd mm rr	BRCLR (opr) (msk) (rel)	DIR	6
14	dd mm	BSET (opr) (msk)	DIR	6
15	dd mm	BCLR (opr) (msk)	DIR	6
16		TAB	INH	2
17		TBA	INH	2
18		(Page 2 Switch)		
19		DAA	INH	2
1A		(Page 3 Switch)		
1B		ABA	INH	2
1C	ff mm	BSET (opr) (msk)	IND,X	7
1D	ff mm	BCLR (opr) (msk)	IND,X	7
1E	ff mm rr	BRSET (opr) (msk) (rel)	IND,X	7
1F	ff mm rr	BRCLR (opr) (msk) (rel)	IND,X	7

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
20	rr	BRA (rel)	REL	3
21	rr	BRN (rel)	REL	3
22	rr	BHI (rel)	REL	3
23	rr	BLS (rel)	REL	3
24	rr	BCC/BHS (rel)	REL	3
25	rr	BCS/BLO (rel)	REL	3
26	rr	BNE (rel)	REL	3
27	rr	BEQ (rel)	REL	3
28	rr	BVC (rel)	REL	3
29	rr	BVS (rel)	REL	3
2A	rr	BPL (rel)	REL	3
2B	rr	BMI (rel)	REL	3
2C	rr	BGE (rel)	REL	3
2D	rr	BLT (rel)	REL	3
2E	rr	BGT (rel)	REL	3
2F	rr	BLE (rel)	REL	3
30		TSX	INH	3
31		INS	INH	3
32		PULA	INH	4
33		PULB	INH	4
34		DES	INH	3
35		TXS	INH	3
36		PSHA	INH	3
37		PSHB	INH	3
38		PULX	INH	5
39		RTS	INH	5
3A		ABX	INH	3
3B		RTI	INH	12
3C		PSHX	INH	4
3D		MUL	INH	10
3E		WAI	INH	14
3F		SWI	INH	14
40		NEGA	INH	2
43		COMA	INH	2
44		LSRA	INH	2
46		RORA	INH	2
47		ASRA	INH	2
48		ASLA/LSLA	INH	2
49		ROLA	INH	2
4A		DECA	INH	2
4C		INCA	INH	2
4D		TSTA	INH	2
4F		CLRA	INH	2
50		NEGB	INH	2

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
53		COMB	INH	2
54		LSRB	INH	2
56		RORB	INH	2
57		ASRB/ASLB	INH	2
58		LSLB	INH	2
59		ROLB	INH	2
5A		DECB	INH	2
5C		INCB	INH	2
5D		TSTB	INH	2
5F		CLRB	INH	2
60	ff	NEG (opr)	IND,X	6
63	ff	COM (opr)	IND,X	6
64	ff	LSR (opr)	IND,X	6
66	ff	ROR (opr)	IND,X	6
67	ff	ASR (opr)	IND,X	6
68	ff	ASL/LSL (opr)	IND,X	6
69	ff	ROL (opr)	IND,X	6
6A	ff	DEC (opr)	IND,X	6
6C	ff	INC (opr)	IND,X	6
6D	ff	TST (opr)	IND,X	6
6E	ff	JMP (opr)	IND,X	3
6F	ff	CLR (opr)	IND,X	6
70	hh ll	NEG (opr)	EXT	6
73	hh ll	COM (opr)	EXT	6
74	hh ll	LSR (opr)	EXT	6
76	hh ll	ROR (opr)	EXT	6
77	hh ll	ASR (opr)	EXT	6
78	hh ll	ASL/LSL (opr)	EXT	6
79	hh ll	ROL (opr)	EXT	6
7A	hh ll	DEC (opr)	EXT	6
7C	hh ll	INC (opr)	EXT	6
7D	hh ll	TST (opr)	EXT	6
7E	hh ll	JMP (opr)	EXT	3
7F	hh ll	CLR (opr)	EXT	6
80	ii	SUBA (opr)	IMM	2
81	ii	CMPA (opr)	IMM	2
82	ii	SBCA (opr)	IMM	2
83	jj kk	SUBD (opr)	IMM	4
84	ii	ANDA (opr)	IMM	2
85	ii	BITA (opr)	IMM	2
86	ii	LDAA (opr)	IMM	2
88	ii	EORA (opr)	IMM	2
89	ii	ADCA (opr)	IMM	2
8A	ii	ORAA (opr)	IMM	2

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
8B	ii	ADDA (opr)	IMM	2
8C	jj kk	CPX (opr)	IMM	4
8D	rr	BSR (rel)	REL	6
8E	jj kk	LDS (opr)	IMM	3
8F		XGDX	INH	3
90	dd	SUBA (opr)	DIR	3
91	dd	CMPA (opr)	DIR	3
92	dd	SBCA (opr)	DIR	3
93	dd	SUBD (opr)	DIR	5
94	dd	ANDA (opr)	DIR	3
95	dd	BITA (opr)	DIR	3
96	dd	LDAA (opr)	DIR	3
97	dd	STAA (opr)	DIR	3
98	dd	EORA (opr)	DIR	3
99	dd	ADCA (opr)	DIR	3
9A	dd	ORAA (opr)	DIR	3
9B	dd	ADDA (opr)	DIR	3
9C	dd	CPX (opr)	DIR	5
9D	dd	JSR (opr)	DIR	5
9E	dd	LDS (opr)	DIR	4
9F	dd	STS (opr)	DIR	4
A0	ff	SUBA (opr)	IND,X	4
A1	ff	CMPA (opr)	IND,X	4
A2	ff	SBCA (opr)	IND,X	4
A3	ff	SUBD (opr)	IND,X	6
A4	ff	ANDA (opr)	IND,X	4
A5	ff	BITA (opr)	IND,X	4
A6	ff	LDAA (opr)	IND,X	4
A7	ff	STAA (opr)	IND,X	4
A8	ff	EORA (opr)	IND,X	4
A9	ff	ADCA (opr)	IND,X	4
AA	ff	ORAA (opr)	IND,X	4
AB	ff	ADDA (opr)	IND,X	4
AC	ff	CPX (opr)	IND,X	6
AD	ff	JSR (opr)	IND,X	6
AE	ff	LDS (opr)	IND,X	5
AF	ff	STS (opr)	IND,X	5
B0	hh ll	SUBA (opr)	EXT	4
B1	hh ll	CMPA (opr)	EXT	4
B2	hh ll	SBCA (opr)	EXT	4
B3	hh ll	SUBD (opr)	EXT	6
B4	hh ll	ANDA (opr)	EXT	4
B5	hh ll	BITA (opr)	EXT	4
B6	hh ll	LDAA (opr)	EXT	4

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
B7	hh ll	STAA (opr)	EXT	4
B8	hh ll	EORA (opr)	EXT	4
B9	hh ll	ADCA (opr)	EXT	4
BA	hh ll	ORAA (opr)	EXT	4
BB	hh ll	ADDA (opr)	EXT	4
BC	hh ll	CPX (opr)	EXT	6
BD	hh ll	JSR (opr)	EXT	6
BE	hh ll	LDS (opr)	EXT	5
BF	hh ll	STS (opr)	EXT	5
C0	ii	SUBB (opr)	IMM	2
C1	ii	CMPB (opr)	IMM	2
C2	ii	SBCB (opr)	IMM	2
C3	jj kk	ADDD (opr)	IMM	4
C4	ii	ANDB (opr)	IMM	2
C5	ii	BITB (opr)	IMM	2
C6	ii	LDAB (opr)	IMM	2
C8	ii	EORB (opr)	IMM	2
C9	ii	ADCB (opr)	IMM	2
CA	ii	ORAB (opr)	IMM	2
CB	ii	ADDB (opr)	IMM	2
CC	jj kk	LDD (opr)	IMM	3
CD		(Page 4 Switch)		
CE	jj kk	LDX (opr)	IMM	3
CF		STOP	INH	2
D0	dd	SUBB (opr)	DIR	3
D1	dd	CMPB (opr)	DIR	3
D2	dd	SBCB (opr)	DIR	3
D3	dd	ADDD (opr)	DIR	5
D4	dd	ANDB (opr)	DIR	3
D5	dd	BITB (opr)	DIR	3
D6	dd	LDAB (opr)	DIR	3
D7	dd	STAB (opr)	DIR	3
D8	dd	EORB (opr)	DIR	3
D9	dd	ADCB (opr)	DIR	3
DA	dd	ORAB (opr)	DIR	3
DB	dd	ADDB (opr)	DIR	3
DC	dd	LDD (opr)	DIR	4
DD	dd	STD (opr)	DIR	4
DE	dd	LDX (opr)	DIR	4
DF	dd	STX (opr)	DIR	4
E0	ff	SUBB (opr)	IND,X	4
E1	ff	CMPB (opr)	IND,X	4
E2	ff	SBCB (opr)	IND,X	4
E3	ff	ADDD (opr)	IND,X	6

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
E4	ff	ANDB (opr)	IND,X	4
E5	ff	BITB (opr)	IND,X	4
E6	ff	LDAB (opr)	IND,X	4
E7	ff	STAB (opr)	IND,X	4
E8	ff	EORB (opr)	IND,X	4
E9	ff	ADCB (opr)	IND,X	4
EA	ff	ORAB (opr)	IND,X	4
EB	ff	ADDB (opr)	IND,X	4
EC	ff	LDD (opr)	IND,X	5
ED	ff	STD (opr)	IND,X	5
EE	ff	LDX (opr)	IND,X	5
EF	ff	STX (opr)	IND,X	5
F0	hh ll	SUBB (opr)	EXT	4
F1	hh ll	CMPB (opr)	EXT	4
F2	hh ll	SBCB (opr)	EXT	4
F3	hh ll	ADDD (opr)	EXT	6
F4	hh ll	ANDB (opr)	EXT	4
F5	hh ll	BITB (opr)	EXT	4
F6	hh ll	LDAB (opr)	EXT	4
F7	hh ll	STAB (opr)	EXT	4
F8	hh ll	EORB (opr)	EXT	4
F9	hh ll	ADCB (opr)	EXT	4
FA	hh ll	ORAB (opr)	EXT	4
FB	hh ll	ADDB (opr)	EXT	4
FC	hh ll	LDD (opr)	EXT	5
FD	hh ll	STD (opr)	EXT	5
FE	hh ll	LDX (opr)	EXT	5
FF	hh ll	STX (opr)	EXT	5
18 08		INY	INH	4
18 09		DEY	INH	4
18 1C	ff mm	BSET (opr) (msk)	IND,Y	8
18 1D	ff mm	BCLR (opr) (msk)	IND,Y	8
18 1E	ff mm rr	BRSET (opr) (msk) (rel)	IND,Y	8
18 1F	ff mm rr	BRCLR (opr) (msk) (rel)	IND,Y	8
18 30		TSY	INH	4
18 35		TYS	INH	4
18 38		PULY	INH	6
18 3A		ABY	INH	4

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
18 3C		PSHY	INH	5
18 60	ff	NEG (opr)	IND,Y	7
18 63	ff	COM (opr)	IND,Y	7
18 64	ff	LSR (opr)	IND,Y	7
18 66	ff	ROR (opr)	IND,Y	7
18 67	ff	ASR (opr)	IND,Y	7
18 68	ff	ASL/LSL (opr)	IND,Y	7
18 69	ff	ROL (opr)	IND,Y	7
18 6A	ff	DEC (opr)	IND,Y	7
18 6C	ff	INC (opr)	IND,Y	7
18 6D	ff	TST (opr)	IND,Y	7
18 6E	ff	JMP (opr)	IND,Y	4
18 6F	ff	CLR (opr)	IND,Y	7
18 8C	jj kk	CPY (opr)	IMM	5
18 8F		XGDY	INH	4
18 9C	dd	CPY (opr)	DIR	6
18 A0	ff	SUBA (opr)	IND,Y	5
18 A1	ff	CMPA (opr)	IND,Y	5
18 A2	ff	SBCA (opr)	IND,Y	5
18 A3	ff	SUBD (opr)	IND,Y	7
18 A4	ff	ANDA (opr)	IND,Y	5
18 A5	ff	BITA (opr)	IND,Y	5
18 A6	ff	LDAA (opr)	IND,Y	5
18 A7	ff	STAA (opr)	IND,Y	5
18 A8	ff	EORA (opr)	IND,Y	5
18 A9	ff	ADCA (opr)	IND,Y	5
18 AA	ff	ORAA (opr)	IND,Y	5
18 AB	ff	ADDA (opr)	IND,Y	5
18 AC	ff	CPY (opr)	IND,Y	7
18 AD	ff	JSR (opr)	IND,Y	7
18 AE	ff	LDS (opr)	IND,Y	6
18 AF	ff	STS (opr)	IND,Y	6
18 BC	hh ll	CPY (opr)	EXT	7
18 CE	jj kk	LDY (opr)	IMM	4
18 DE	dd	LDY (opr)	DIR	5
18 DF	dd	STY (opr)	DIR	5
18 E0	ff	SUBB (opr)	IND,Y	5
18 E1	ff	CMPB (opr)	IND,Y	5
18 E2	ff	SBCB (opr)	IND,Y	5
18 E3	ff	ADDD (opr)	IND,Y	5
18 E4	ff	ANDB (opr)	IND,Y	5
18 E5	ff	BITB (opr)	IND,Y	5
18 E6	ff	LDAB (opr)	IND,Y	5
18 E7	ff	STAB (opr)	IND,Y	5

Opcode vs Instruction Cross Reference

Opcode	Operands	Instruction	ADDR Mode	Cycle
18 E8	ff	EORB (opr)	IND,Y	5
18 E9	ff	ADCB (opr)	IND,Y	5
18 EA	ff	ORAB (opr)	IND,Y	5
18 EB	ff	ADDB (opr)	IND,Y	5
18 EC	ff	LDD (opr)	IND,Y	6
18 ED	ff	STD (opr)	IND,Y	6
18 EE	ff	LDY (opr)	IND,Y	6
18 EF	ff	STY (opr)	IND,Y	6
18 FE	hh ll	LDY (opr)	EXT	6
18 FF	hh ll	STY (opr)	EXT	6
1A 83	jj kk	CPD (opr)	IMM	5
1A 93	dd	CPD (opr)	DIR	6
1A A3	ff	CPD (opr)	IND,X	7
1A AC	ff	CPY (opr)	IND,X	7
1A B3	hh ll	CPD (opr)	EXT	7
1A EE	ff	LDY (opr)	IND,X	6
1A EF	ff	STY (opr)	IND,X	6
CD A3	ff	CPD (opr)	IND,Y	7
CD AC	ff	CPX (opr)	IND,Y	7
CD EE	ff	LDX (opr)	IND,Y	6
CD EF	ff	STX (opr)	IND,Y	6

NOTES:

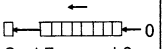
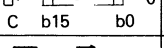
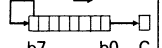
Operands:

- dd = 8-bit direct address $0000-$00FF. (High byte assumed to be $00.)
- ff = 8-bit positive offset $00 (0) to $FF (255) 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 the address following the machine code offset byte.

INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes							
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
ABA	Add Accumulators	A + B → A	INH	1B		1	2	-	-	↕	-	↕	↕	↕	↕
ABX	Add B to X	IX + 00:B → IX	INH	3A		1	3	-	-	-	-	-	-	-	-
ABY	Add B to Y	IY + 00:B → IY	INH	18 3A		2	4	-	-	-	-	-	-	-	-
ADCA (opr)	Add with Carry to A	A + M + C → A	A IMM	89	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			A DIR	99	dd	2	3								
			A EXT	B9	hh ll	3	4								
			A IND,X	A9	ff	2	4								
			A IND,Y	18 A9	ff	3	5								
ADCB (opr)	Add with Carry to B	B + M + C → B	B IMM	C9	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			B DIR	D9	dd	2	3								
			B EXT	F9	hh ll	3	4								
			B IND,X	E9	ff	2	4								
			B IND,Y	18 E9	ff	3	5								

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes							
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
ADDA (opr)	Add Memory to A	A + M → A	A IMM	8B	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			A DIR	9B	dd	2	3								
			A EXT	BB	hh ll	3	4								
			A IND,X	AB	ff	2	4								
			A IND,Y	18 AB	ff	3	5								
ADDB (opr)	Add Memory to B	B + M → B	B IMM	CB	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			B DIR	DB	dd	2	3								
			B EXT	FB	hh ll	3	4								
			B IND,X	EB	ff	2	4								
			B IND,Y	18 EB	ff	3	5								
ADDD (opr)	Add 16-Bit to D	D + M:M + 1 → D	IMM	C3	jj kk	3	4	-	-	-	-	↕	↕	↕	↕
			DIR	D3	dd	2	5								
			EXT	F3	hh ll	3	6								
			IND,X	E3	ff	2	6								
			IND, Y	18 E3	ff	3	7								
ANDA (opr)	AND A with Memory	A•M → A	A IMM	84	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	94	dd	2	3								
			A EXT	B4	hh ll	3	4								
			A IND,X	A4	ff	2	4								
			A IND,Y	18 A4	ff	3	5								

ANDB (oprx)	AND B with Memory	B•M → B	B IMM	C4	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D4	dd	2	3								
			B EXT	F4	hh ll	3	4								
			B IND,X	E4	ff	2	4								
			B IND,Y	18 E4	ff	3	5								
ASL (oprx)	Arithmetic Shift Left		EXT	78	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	68	ff	2	6								
		C b7 b0	IND,Y	18 68	ff	3	7								
ASLA			A INH	48		1	2								
ASLB			B INH	58		1	2								
ASLD	Arithmetic Shift Left Double		INH	05		1	3	-	-	-	-	↕	↕	↕	↕
		C b15 b0													
ASR (oprx)	Arithmetic Shift Right		EXT	77	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	67	ff	2	6								
		b7 b0 C	IND,Y	18 67	ff	3	7								
ASRA			A INH	47		1	2								
ASRB			B INH	57		1	2								
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	2	3	-	-	-	-	-	-	-	-
BCLR (oprx) (msk)	Clear Bit(s)	M•(mm) → M	DIR	15	dd mm	3	6	-	-	-	-	↕	↕	0	-
			IND,X	1D	ff mm	3	7								
			IND,Y	18 1D	ff mm	4	8								
BCS (rel)	Branch if Carry Set	? C = 1	REL	25	rr	2	3	-	-	-	-	-	-	-	-
BEQ (rel)	Branch if = Zero	? Z = 1	REL	27	rr	2	3	-	-	-	-	-	-	-	-

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes							
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
BGE (rel)	Branch if ≥ = Zero	? N ⊕ V = 0	REL	2C	rr	2	3	-	-	-	-	-	-	-	-
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	2	3	-	-	-	-	-	-	-	-
BHI (rel)	Branch if Higher	? C + Z = 0	REL	22	rr	2	3	-	-	-	-	-	-	-	-
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	rr	2	3	-	-	-	-	-	-	-	-
BITA (opr)	Bit(s) Test A with Memory	A•M	A IMM	85	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	95	dd	2	3								
			A EXT	B5	hh ll	3	4								
			A IND,X	A5	ff	2	4								
			A IND,Y	18 A5	ff	3	5								
BITB (opr)	Bit(s) Test B with Memory	B•M	B IMM	C5	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D5	dd	2	3								
			B EXT	F5	hh ll	3	4								
			B IND,X	E5	ff	2	4								
			B IND,Y	18 E5	ff	3	5								
BLE (rel)	Branch if ≤ Zero	? Z + (N ⊕ V) = 1	REL	2F	rr	2	3	-	-	-	-	-	-	-	-
BLO (rel)	Branch if Lower	? C = 1	REL	25	rr	2	3	-	-	-	-	-	-	-	-
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23	rr	2	3	-	-	-	-	-	-	-	-

BLT (rel)	Branch If < Zero	? N ⊕ V = 1	REL	2D	rr	2	3	-	-	-	-	-	-	-	-
BMI (rel)	Branch if Minus	? N = 1	REL	2B	rr	2	3	-	-	-	-	-	-	-	-
BNE (rel)	Branch if Not = Zero	? Z = 0	REL	26	rr	2	3	-	-	-	-	-	-	-	-
BPL (rel)	Branch if Plus	? N = 0	REL	2A	rr	2	3	-	-	-	-	-	-	-	-
BRA (rel)	Branch Always	? 1 = 1	REL	20	rr	2	3	-	-	-	-	-	-	-	-
BRCLR(opr)	Branch if Bit(s) Clear	? M• mm = 0	DIR	13	dd mm rr	4	6	-	-	-	-	-	-	-	-
(msk)			IND,X	1F	ff mm rr	4	7								
(rel)			IND,Y	18 1F	ff mm rr	5	8								
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	2	3	-	-	-	-	-	-	-	-
BRSET(opr)	Branch if Bit(s) Set	? (M)•mm = 0	DIR	12	dd mm rr	4	6	-	-	-	-	-	-	-	-
(msk)			IND,X	1E	ff mm rr	4	7								
(rel)			IND,Y	18 1E	ff mm rr	5	8								
BSET(opr)	Set Bit(s)	M + mm → M	DIR	14	dd mm	3	6	-	-	-	-	↕	↕	0	-
(msk)			IND,X	1C	ff mm	3	7								
			IND,Y	18 1C	ff mm	4	8								
BSR (rel)	Branch to Subroutine	See Special Ops	REL	8D	rr	2	6	-	-	-	-	-	-	-	-
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	rr	2	3	-	-	-	-	-	-	-	-
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	rr	2	3	-	-	-	-	-	-	-	-
CBA	Compare A to B	A − B	INH	11		1	2	-	-	-	-	↕	↕	↕	↕
CLC	Clear Carry Bit	0 → C	INH	0C		1	2	-	-	-	-	-	-	-	0
CLI	Clear Interrupt Mask	0 → I	INH	0E		1	2	-	-	-	0	-	-	-	-

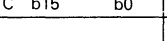
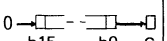
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes							
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
CLR (opr)	Clear Memory Byte	0 → M	EXT	7F	hh ll	3	6	-	-	-	-	0	1	0	0
			IND,X	6F	ff	2	6								
			IND,Y	18 6F	ff	3	7								
CLRA	Clear Accumulator A	0 → A	A INH	4F		1	2	-	-	-	-	0	1	0	0
CLRB	Clear Accumulator B	0 → B	B INH	5F		1	2	-	-	-	-	0	1	0	0
CLV	Clear Overflow Flag	0 → V	INH	0A		1	2	-	-	-	-	-	-	0	-
CMPA (opr)	Compare A to Memory	A − M	A IMM	81	ii	2	2	-	-	-	-	↕	↕	↕	↕
			A DIR	91	dd	2	3								
			A EXT	B1	hh ll	3	4								
			A IND,X	A1	ff	2	4								
			A IND,Y	18 A1	ff	3	5								
CMPB (opr)	Compare B to Memory	B − M	B IMM	C1	ii	2	2	-	-	-	-	↕	↕	↕	↕
			B DIR	D1	dd	2	3								
			B EXT	F1	hh ll	3	4								
			B IND,X	E1	ff	2	4								
			B IND,Y	18 E1	ff	3	5								
COM (opr)	1's Complement Memory Byte	$FF − M → M	EXT	73	hh ll	3	6	-	-	-	-	↕	↕	0	1
			IND,X	63	ff	2	6								
			IND,Y	18 63	ff	3	7								

COMA	1's Complement A	$FF – A → A	A INH	43			1	2	-	-	-	-	↕	↕	0	1
COMB	1's Complement B	$FF – B → B	B INH	53			1	2	-	-	-	-	↕	↕	0	1
CPD (opr)	Compare D to Memory 16-Bit	D – M:M + 1	IMM	1A 83	jj	kk	4	5	-	-	-	-	↕	↕	↕	↕
			DIR	1A 93	dd		3	6								
			EXT	1A B3	hh	ll	4	7								
			IND,X	1A A3	ff		3	7								
			IND,Y	CD A3	ff		3	7								
CPX (opr)	Compare X to Memory 16-Bit	IX – M:M + 1	IMM	8C	jj	kk	3	4	-	-	-	-	↕	↕	↕	↕
			DIR	9C	dd		2	5								
			EXT	BC	hh	ll	3	6								
			IND,X	AC	ff		2	6								
			IND,Y	CD AC	ff		3	7								
CPY (opr)	Compare Y to Memory 16-Bit	IY – M:M + 1	IMM	18 8C	jj	kk	4	5	-	-	-	-	↕	↕	↕	↕
			DIR	18 9C	dd		3	6								
			EXT	18 BC	hh	ll	4	7								
			IND,X	1A AC	ff		3	7								
			IND,Y	18 AC	ff		3	7								
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19			1	2	-	-	-	-	↕	↕	↕	↕
DEC (opr)	Decrement Memory Byte	M – 1 → M	EXT	7A	hh	ll	3	6	-	-	-	-	↕	↕	↕	-
			IND,X	6A	ff		2	6								
			IND,Y	18 6A	ff		3	7								
DECA	Decrement Accumulator A	A – 1 → A	A INH	4A			1	2	-	-	-	-	↕	↕	↕	-
DECB	Decrement Accumulator B	B – 1 → B	B INH	5A			1	2	-	-	-	-	↕	↕	↕	-
DES	Decrement Stack Pointer	SP – 1 → SP	INH	34			1	3	-	-	-	-	-	-	-	-

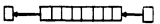
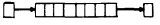
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes							
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
DEX	Decrement Index Register X	IX − 1 → IX	INH	09		1	3	-	-	-	-	-	↕	-	-
DEY	Decrement Index Register Y	IY − 1 → IY	INH	18 09		2	4	-	-	-	-	-	↕	-	-
EORA (opr)	Exclusive OR A with Memory	A ⊕ M → A	A IMM	88	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	98	dd	2	3								
			A EXT	B8	hh ll	3	4								
			A IND,X	A8	ff	2	4								
			A IND,Y	18 A8	ff	3	5								
EORB (opr)	Exclusive OR B with Memory	B ⊕ M → B	B IMM	C8	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D8	dd	2	3								
			B EXT	F8	hh ll	3	4								
			B IND,X	E8	ff	2	4								
			B IND,Y	18 E8	ff	3	5								
FDIV	Fractional Divide 16 by 16	D/IX → IX; r → D	INH	03		1	41	-	-	-	-	-	↕	↕	↕
IDIV	Integer Divide 16 by 16	D/IX → IX; r → D	INH	02		1	41	-	-	-	-	-	↕	0	↕
INC (opr)	Increment Memory Byte	M + 1 → M	EXT	7C	hh ll	3	6	-	-	-	-	↕	↕	↕	-
			IND,X	6C	ff	2	6								
			IND,Y	18 6C	ff	3	7								
INCA	Increment Accumulator A	A + 1 → A	A INH	4C		1	2	-	-	-	-	↕	↕	↕	-
INCB	Increment Accumulator B	B + 1 → B	B INH	5C		1	2	-	-	-	-	↕	↕	↕	-

INS	Increment Stack Pointer	SP + 1 → SP	INH	31		1	3	-	-	-	-	-	-	-	-
INX	Increment Index Register X	IX + 1 → IX	INH	08		1	3	-	-	-	-	-	↕	-	-
INY	Increment Index Register Y	IY + 1 → IY	INH	18 08		2	4	-	-	-	-	-	↕	-	-
JMP (opr)	Jump	See Special Ops	EXT	7E	hh ll	3	3	-	-	-	-	-	-	-	-
			IND,X	6E	ff	2	3								
			IND,Y	18 6E	ff	3	4								
JSR (opr)	Jump to Subroutine	See Special Ops	DIR	9D	dd	2	5	-	-	-	-	-	-	-	-
			EXT	BD	hh ll	3	6								
			IND,X	AD	ff	2	6								
			IND,Y	18 AD	ff	3	7								
LDAA (opr)	Load Accumulator A	M → A	A IMM	86	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	96	dd	2	3								
			A EXT	B6	hh ll	3	4								
			A IND,X	A6	ff	2	4								
			A IND,Y	18 A6	ff	3	5								
LDAB (opr)	Load Accumulator B	M → B	B IMM	C6	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D6	dd	2	3								
			B EXT	F6	hh ll	3	4								
			B IND,X	E6	ff	2	4								
			B IND,Y	18 E6	ff	3	5								

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycle	Condition Codes S	X	H	I	N	Z	V	C
LDD (opr)	Load Double Accumulator D	M → A, M + 1 → B	IMM	CC	jj kk	3	3	-	-	-	-	↕	↕	0	-
			DIR	DC	dd	2	4								
			EXT	FC	hh ll	3	5								
			IND,X	EC	ff	2	5								
			IND,Y	18 EC	ff	3	6								
LDS (opr)	Load Stack Pointer	M:M + 1 → SP	IMM	8E	jj kk	3	3	-	-	-	-	↕	↕	0	-
			DIR	9E	dd	2	4								
			EXT	BE	hh ll	3	5								
			IND,X	AE	ff	2	5								
			IND,Y	18 AE	ff	3	6								
LDX (opr)	Load Index Register X	M:M + 1 → IX	IMM	CE	jj kk	3	3	-	-	-	-	↕	↕	0	-
			DIR	DE	dd	2	4								
			EXT	FE	hh ll	3	5								
			IND,X	EE	ff	2	5								
			IND,Y	CD EE	ff	3	6								
LDY (opr)	Load Index Register Y	M:M + 1 → IY	IMM	18 CE	jj kk	4	4	-	-	-	-	↕	↕	0	-
			DIR	18 DE	dd	3	5								
			EXT	18 FE	hh ll	4	6								
			IND,X	1A EE	ff	3	6								
			IND,Y	18 EE	ff	3	6								

LSL (opr)	Logical Shift Left		EXT	78 hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	68 ff	2	6								
			IND,Y	18 68 ff	3	7								
LSLA			A INH	48	1	2								
LSLB			B INH	58	1	2								
LSLD	Logical Shift Left Double		INH	05	1	3	-	-	-	-	↕	↕	↕	↕
LSR (opr)	Logical Shift Right		EXT	74 hh ll	3	6	-	-	-	-	0	↕	↕	↕
			IND,X	64 ff	2	6								
			IND,Y	18 64 ff	3	7								
LSRA			A INH	44	1	2								
LSRB			B INH	54	1	2								
LSRD	Logical Shift Right Double		INH	04	1	3	-	-	-	-	0	↕	↕	↕
MUL	Multiply 8 by 8	AxB → D	INH	3D	1	10	-	-	-	-	-	-	-	↕
NEG (opr)	2's Complement Memory Byte	0 − M → M	EXT	70 hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	60 ff	2	6								
			IND,Y	18 60 ff	3	7								
NEGA	2's Complement A	0 − A → A	A INH	40	1	2	-	-	-	-	↕	↕	↕	↕
NEGB	2's Complement B	0 − B → B	B INH	50	1	2	-	-	-	-	↕	↕	↕	↕
NOP	No Operation	No Operation	INH	01	1	2	-	-	-	-	-	-	-	-

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes							
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
ORAA (opr)	OR Accumulator A (Inclusive)	A + M → A	A IMM	8A	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	9A	dd	2	3								
			A EXT	BA	hh ll	3	4								
			A IND,X	AA	ff	2	4								
			A IND,Y	18 AA	ff	3	5								
ORAB (opr)	OR Accumulator B (Inclusive)	B + M → B	B IMM	CA	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	DA	dd	2	3								
			B EXT	FA	hh ll	3	4								
			B IND,X	EA	ff	2	4								
			B IND,Y	18 EA	ff	3	5								
PSHA	Push A onto Stack	A → Stk, SP = SP − 1	A INH	36		1	3	-	-	-	-	-	-	-	-
PSHB	Push B onto Stack	B → Stk, SP = SP − 1	B INH	37		1	3	-	-	-	-	-	-	-	-
PSHX	Push X onto Stack (Lo First)	IX → Stk, SP = SP − 2	INH	3C		1	4	-	-	-	-	-	-	-	-
PSHY	Push Y onto Stack (Lo First)	IY → Stk, SP = SP − 2	INH	18 3C		2	5	-	-	-	-	-	-	-	-
PULA	Pull A from Stack	SP = SP + 1, A ← Stk	A INH	32		1	4	-	-	-	-	-	-	-	-
PULB	Pull B from Stack	SP = SP + 1, B ← Stk	B INH	33		1	4	-	-	-	-	-	-	-	-
PULX	Pull X from Stack (Hi First)	SP = SP + 2, IX ← Stk	INH	38		1	5	-	-	-	-	-	-	-	-
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY ← Stk	INH	18 38		2	6	-	-	-	-	-	-	-	-

ROL (opr)	Rotate Left		EXT	79	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	69	ff	2	6								
			IND,Y	18 69	ff	3	7								
ROLA			A INH	49		1	2								
ROLB			B INH	59		1	2								
ROR (opr)	Rotate Right		EXT	76	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	66	ff	2	6								
			IND,Y	18 66	ff	3	7								
RORA			A INH	46		1	2								
RORB			B INH	56		1	2								
RTI	Return from Interrupt	See Special Ops	INH	3B		1	12	↕	↓	↕	↕	↕	↕	↕	↕
RTS	Return from Subroutine	See Special Ops	INH	39		1	5	-	-	-	-	-	-	-	-
SBA	Subtract B from A	A – B → A	INH	10		1	2	-	-	-	-	↕	↕	↕	↕
SBCA (opr)	Subtract with Carry from A	A – M – C → A	A IMM	82	ii	2	2	-	-	-	-	↕	↕	↕	↕
			A DIR	92	dd	2	3								
			A EXT	B2	hh ll	3	4								
			A IND,X	A2	ff	2	4								
			A IND,Y	18 A2	ff	3	5								
SBCB (opr)	Subtract with Carry from B	B – M – C → B	B IMM	C2	ii	2	2	-	-	-	-	↕	↕	↕	↕
			B DIR	D2	dd	2	3								
			B EXT	F2	hh ll	3	4								
			B IND,X	E2	ff	2	4								
			B IND,Y	18 E2	ff	3	5								

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes							
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
SEC	Set Carry	1 → C	INH	0D		1	2	-	-	-	-	-	-	-	1
SEI	Set Interrupt Mask	1 → I	INH	0F		1	2	-	-	-	1	-	-	-	-
SEV	Set Overflow Flag	1 → V	INH	0B		1	2	-	-	-	-	-	-	1	-
STAA (opr)	Store Accumulator A	A → M	A DIR	97	dd	2	3	-	-	-	-	↕	↕	0	-
			A EXT	B7	hh ll	3	4								
			A IND,X	A7	ff	2	4								
			A IND,Y	18 A7	ff	3	5								
STAB (opr)	Store Accumulator B	B → M	B DIR	D7	dd	2	3	-	-	-	-	↕	↕	0	-
			B EXT	F7	hh ll	3	4								
			B IND,X	E7	ff	2	4								
			B IND,Y	18 E7	ff	3	5								
STD (opr)	Store Accumulator D	A → M, B → M + 1	DIR	DD	dd	2	4	-	-	-	-	↕	↕	0	-
			EXT	FD	hh ll	3	5								
			IND,X	ED	ff	2	5								
			IND,Y	18 ED	ff	3	6								
STOP	Stop Internal Clocks		INH	CF		1	2	-	-	-	-	-	-	-	-
STS (opr)	Store Stack Pointer	SP → M:M + 1	DIR	9F	dd	2	4	-	-	-	-	↕	↕	0	-
			EXT	BF	hh ll	3	5								
			IND,X	AF	ff	2	5								
			IND,Y	18 AF	ff	3	6								

STX (opr)	Store Index Register X	IX → M:M + 1	DIR	DF	dd	2	4	-	-	-	-	↕	↕	0	-
			EXT	FF	hh ll	3	5								
			IND,X	EF	ff	2	5								
			IND,Y	CD EF	ff	3	6								
STY (opr)	Store Index Register Y	IY → M:M + 1	DIR	18 DF	dd	3	5	-	-	-	-	↕	↕	0	-
			EXT	18 FF	hh ll	4	6								
			IND,X	1A EF	ff	3	6								
			IND,Y	18 EF	ff	3	6								
SUBA (opr)	Subtract Memory from A	A − M → A	A IMM	80	ii	2	2	-	-	-	-	↕	↕	↕	↕
			A DIR	90	dd	2	3								
			A EXT	B0	hh ll	3	4								
			A IND,X	A0	ff	2	4								
			A IND,Y	18 A0	ff	3	5								
SUBB (opr)	Subtract Memory from B	B − M → B	B IMM	C0	ii	2	2	-	-	-	-	↕	↕	↕	↕
			B DIR	D0	dd	2	3								
			B EXT	F0	hh ll	3	4								
			B IND,X	E0	ff	2	4								
			B IND,Y	18 E0	ff	3	5								
SUBD (opr)	Subtract Memory from D	D − M:M + 1 → D	IMM	83	jj kk	3	4	-	-	-	-	↕	↕	↕	↕
			DIR	93	dd	2	5								
			EXT	B3	hh ll	3	6								
			IND,X	A3	ff	2	6								
			IND,Y	18 A3	ff	3	7								
SWI	Software Interrupt	See Special Ops	INH	3F		1	14	-	-	-	1	-	-	-	-

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Condition Codes						
				Opcode	Operand(s)			S	X	H	I	N	Z	V	C
TAB	Transfer A to B	A → B	INH	16		1	2	-	-	-	-	↕	↕	0	-
TAP	Transfer A to CC Register	A → CCR	INH	06		1	2	↕	↕	↕	↕	↕	↕	↕	↕
TBA	Transfer B to A	B → A	INH	17		1	2	-	-	-	-	↕	↕	0	-
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00		1	*	-	-	-	-	-	-	-	-
TPA	Transfer CC Register to A	CCR → A	INH	07		1	2	-	-	-	-	-	-	-	-
TST (opr)	Test for Zero or Minus	M – 0	EXT	7D	hh ll	3	6	-	-	-	-	↕	↕	0	0
			IND,X	6D	ff	2	6								
			IND,Y	18 6D	ff	3	7								
TSTA		A – 0	A INH	4D		1	2	-	-	-	-	↕	↕	0	0
TSTB		B – 0	B INH	5D		1	2	-	-	-	-	↕	↕	0	0
TSX	Transfer Stack Pointer to X	SP + 1 → IX	INH	30		1	3	-	-	-	-	-	-	-	-
TSY	Transfer Stack Pointer to Y	SP + 1 → IY	INH	18 30		2	4	-	-	-	-	-	-	-	-
TXS	Transfer X to Stack Pointer	IX – 1 → SP	INH	35		1	3	-	-	-	-	-	-	-	-
TYS	Transfer Y to Stack Pointer	IY – 1 → SP	INH	18 35		2	4	-	-	-	-	-	-	-	-
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E		2	**	-	-	-	-	-	-	-	-
XGDX	Exchange D with X	IX → D, D → IX	INH	8F		1	3	-	-	-	-	-	-	-	-
XGDY	Exchange D with Y	IY → D, D → IY	INH	18 8F		2	4	-	-	-	-	-	-	-	-

NOTES:

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 cycle (n) until an interrupt is recognized. Finally, two additional cycles are used to fetch the appropriate interrupt vector (total = 14 + n).

Operands:

dd = 8-bit direct address $0000-$00FF. (High byte assumed to be $00.)

ff = 8-bit positive offset $00 (0) to $FF (255) 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 the address following the machine code offset byte.

Condition Codes:

— Bit not changed

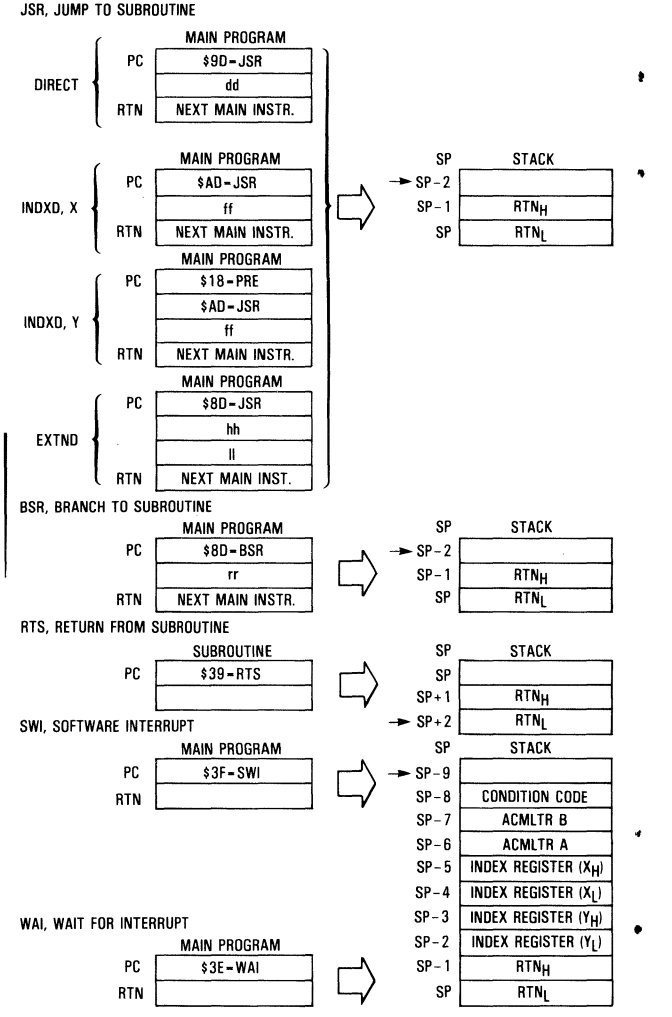
0 Always cleared (logic 0).

1 Always set (logic 1).

↕ Bit cleared or set depending on operation.

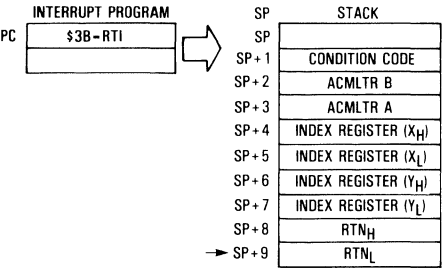
↓ Bit may be cleared, cannot become set.

SPECIAL OPERATIONS

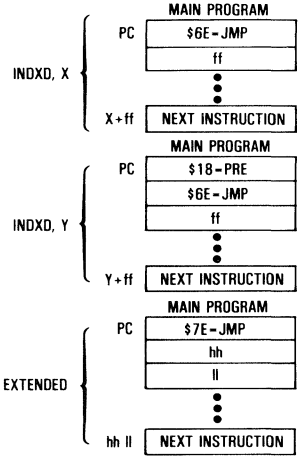


SPECIAL OPERATIONS

RTI, RETURN FROM INTERRUPT



JMP, JUMP



LEGEND

RTN = Address of next instruction in Main Program to be executed upon return from subroutine

RTN_H = Most significant byte of Return Address

RTN_L = Least significant byte of Return Address

→ = Stack Pointer After Execution

dd = 8-Bit Direct Address ($0000-$00FF) (High Byte Assumed to be $00)

ff = 8-Bit Positive Offset $00 (0) to $FF (256) (Is Added to Index)

hh = High Order Byte of 16-Bit Extended Address

ll = Low Order Byte of 16-Bit Extended Address

rr = Signed Relative Offset $80 (− 128) to $7F (+ 127)
(Offset Relative to the Address Following the Machine Code Offset Byte)

REGISTER AND CONTROL BIT SUMMARY

	Bit 7	6	5	4	3	2	1	Bit 0	
$i000	Bit 7	–	–	–	–	–	–	Bit 0	PORTA
$i001									Reserved
$i002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC
$i003	Bit 7	–	–	–	–	–	–	Bit 0	PORTC
$i004	Bit 7	–	–	–	–	–	–	Bit 0	PORTB
$i005	Bit 7	–	–	–	–	–	–	Bit 0	PORTCL
$i006									Reserved
$i007	Bit 7	–	–	–	–	–	–	Bit 0	DDRC
$i008			Bit 5	–	–	–	–	Bit 0	PORTD
$i009			Bit 5	–	–	–	–	Bit 0	DDRD
$i00A	Bit 7	–	–	–	–	–	–	Bit 0	PORTE
$i00B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC
$i00C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M
$i00D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D
$i00E	Bit 15	–	–	–	–	–	–	Bit 8	TCNT
$i00F	Bit 7	–	–	–	–	–	–	Bit 0	
$i010	Bit 15	–	–	–	–	–	–	Bit 8	TIC1
$i011	Bit 7	–	–	–	–	–	–	Bit 0	
$i012	Bit 15	–	–	–	–	–	–	Bit 8	TIC2
$i013	Bit 7	–	–	–	–	–	–	Bit 0	
$i014	Bit 15	–	–	–	–	–	–	Bit 8	TIC3
$i015	Bit 7	–	–	–	–	–	–	Bit 0	
$i016	Bit 15	–	–	–	–	–	–	Bit 8	TOC1
$i017	Bit 7	–	–	–	–	–	–	Bit 0	
$i018	Bit 15	–	–	–	–	–	–	Bit 8	TOC2
$i019	Bit 7	–	–	–	–	–	–	Bit 0	
$i01A	Bit 15	–	–	–	–	–	–	Bit 8	TOC3
$i01B	Bit 7	–	–	–	–	–	–	Bit 0	
$i01C	Bit 15	–	–	–	–	–	–	Bit 8	TOC4
$i01D	Bit 7	–	–	–	–	–	–	Bit 0	
$i01E	Bit 15	–	–	–	–	–	–	Bit 8	TOC5
$i01F	Bit 7	–	–	–	–	–	–	Bit 0	
	Bit 7	6	5	4	3	2	1	Bit 0	

REGISTER AND CONTROL BIT ASSIGNMENTS

	Bit 7	6	5	4	3	2	1	Bit 0	
$i020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
$i021			EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
$i022	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I	TMSK1
$i023	OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F	TFLG1
$i024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2
$i025	TOF	RTIF	PAOVF	PAIF					TFLG2
$i026	DDRA7	PAEN	PAMOD	PEDGE			RTR1	RTR0	PACTL
$i027	Bit 7	—	—	—	—	—	—	Bit 0	PACNT
$i028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$i029	SPIF	WCOL		MODF					SPSR
$i02A	Bit 7	—	—	—	—	—	—	Bit 0	SPDR
$i02B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
$i02C	R8	T8		M	WAKE				SCCR1
$i02D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
$i02E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR
$i02F	Bit 7	—	—	—	—	—	—	Bit 0	SCDR
$i030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL
$i031	Bit 7	—	—	—	—	—	—	Bit 0	ADR1
$i032	Bit 7	—	—	—	—	—	—	Bit 0	ADR2
$i033	Bit 7	—	—	—	—	—	—	Bit 0	ADR3
$i034	Bit 7	—	—	—	—	—	—	Bit 0	ADR4
$i035									Reserved
$i036									Reserved
$i037									Reserved
$i038									Reserved
$i039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION
$i03A	Bit 7	—	—	—	—	—	—	Bit 0	COPRST
$i03B	ODD	EVEN		BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
$i03C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$i03D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
$i03E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1
$i03F					NOSEC	NOCOP	ROMON	EEON	CONFIG
	Bit 7	6	5	4	3	2	1	Bit 0	

A/D Control/Status Register

	B7							B0
$_030	CCF		SCAN	MULT	CD	CC	CB	CA

CCF = Conversions Complete Flag
(Sets after fourth conversion)
SCAN = Continuous Scan Control
0 = 4 conversions and stop
1 = Convert continuously
MULT = Multiple Channel/Single Channel Control
0 = Convert single channel
1 = Convert four channel group
CD – CA = Channel Select D thru A

CD	CC	CB	CA	Channel Signal	Result in ADRx if MULT = 1
0	0	0	0	AD0 port E bit-0	ADR1
0	0	0	1	AD1 port E bit-1	ADR2
0	0	1	0	AD2 port E bit-2	ADR3
0	0	1	1	AD3 port E bit-3	ADR4
0	1	0	0	AD4 port E bit-4	ADR1
0	1	0	1	AD5 port E bit-5	ADR2
0	1	1	0	AD6 port E bit-6	ADR3
0	1	1	1	AD7 port E bit-7	ADR4
1	0	0	0	Reserved	ADR1
1	0	0	1	Reserved	ADR2
1	0	1	0	Reserved	ADR3
1	0	1	1	Reserved	ADR4
1	1	0	0	Vref hi	ADR1
1	1	0	1	Vref low	ADR2
1	1	1	0	Vref hi / 2	ADR3
1	1	1	1	Test/reserved *	ADR4

* denotes used in product testing

A/D Result Registers

	B7			ADR1				B0
$_031	AD17	AD16	AD15	AD14	AD13	AD12	AD11	AD10

	B7			ADR2				B0
$_032	AD27	AD26	AD25	AD24	AD23	AD22	AD21	AD20

	B7			ADR3				B0
$_033	AD37	AD36	AD35	AD34	AD33	AD32	AD31	AD30

	B7			ADR4				B0
$_034	AD47	AD46	AD45	AD44	AD43	AD42	AD41	AD40

Analog Input to 8-Bit Result Translation Table

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
% (1)	50%	25%	12.5%	6.25%	3.12%	1.56%	0.78%	0.39%
Volts (2)	2.500	1.250	0.625	0.3125	0.1562	0.0781	0.0391	0.0195

NOTES:
(1) % of $V_{RH} - V_{RL}$
(2) Volts for $V_{RL} = 0$; $V_{RH} = 5.0$ V

SCI Baud Rate Control Register

	B7							B0
$_02B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET=	0	0	0	0	0	U	U	U

TCLR = Clear Baud Counter Chain (Test only)
RCKB = SCI Baud Rate Clock Test (Test only)

SCP1, SCP0 = Serial Prescaler Selects

S S C C P P 1 0	Divide E by	Highest Baud Rate Xtal = 2^{23}	Highest Baud Rate Xtal = 8.0 MHz	Highest Baud Rate Xtal = 4.0 MHz
0 0	1	131.07 K	—	—
0 1	3	—	—	—
1 0	4	32.768 K	—	—
1 1	13	—	9600	4800
	E =	2.1 MHz	2.0 MHz	1.0 MHz

SCR2 – SCR0 = SCI Rate Select bit 2 thru bit 0

S S S C C C R R R 2 1 0	Prescaler Output Divide-by Factor	Highest Baud Rate 32.768 K	Highest Baud Rate 9600	Highest Baud Rate 4800
0 0 0	1	32.768 K	9600	4800
0 0 1	2	—	4800	2400
0 1 0	4	8.192 K	2400	1200
0 1 1	8	—	1200	600
1 0 0	16	2.048 K	600	300
1 0 1	32	—	300	—
1 1 0	64	512	—	—
1 1 1	128	—	—	—

Timer Compare Force Register

	B7							B0
$_00B	FOC1	FOC2	FOC3	FOC4	FOC5			
RESET=	0	0	0	0	0	0	0	0

Write 1's to force compare(s).

Configuration Control Register

	B7							B0
$_03F					NOSEC	NOCOP	ROMON	EEON

NOSEC = Security Mode Disable (available by MASK option)
- 0 = Security mode (only if MASK option selected)
- 1 = System operates normally (no security)

NOCOP = COP System Disable
- 0 = COP System enabled (forces reset on timeout)
- 1 = COP System disabled

ROMON = ROM Enable
- 0 = ROM is not in the memory map
- 1 = ROM on at $E000 to $FFFF

EEON = EEPROM Enable
- 0 = EEPROM is not in the memory map
- 1 = EEPROM is present at $B600 to $B7FF

NOTE

The bits of this register are implemented with EEPROM cells. Programming and erasure follow normal EEPROM procedures. The erased state of this location is $0F. A new value programmed into this register is not readable until after a subsequent reset sequence.

COPRST

Arm/Reset COP Timer Circuitry

	B7							B0
$_03A	–	–	–	–	–	–	–	–

Write $55 and $AA to reset COP watchdog timer.

DDRC

Data Direction Register for Port C

	B7							B0
$_007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
RESET=	0	0	0	0	0	0	0	0

0 = Inputs
1 = Outputs

DDRD

Data Direction Register for Port D

	B7							B0
$_009			DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
RESET=	0	0	0	0	0	0	0	0

0 = Inputs
1 = Outputs

DDRD

Highest Priority Interrupt and Misc.

	B7							B0
$_03C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0
RESET=	–	–	–	–	0	1	0	1

(Reset condition of RBOOT, SMOD, MDA, and IRV depend on power-up initialization mode.)

RBOOT = Read Bootstrap ROM (Only writable if SMOD = 1)
0 = Boot ROM not in map (normal)
1 = Boot ROM enabled

SMOD = Special Mode Select

MDA = Mode Select

Inputs		Mode Description	Latched at Reset	
MODB	MODA		SMOD	MDA
1	0	(0) Single chip	0	0
1	1	(1) Expanded multiplexed	0	1
0	0	Special bootstrap	1	0
0	1	Special test	1	1

IRV = Internal Read Visibility
(Reset: in test or boot mode — 1; in normal modes - 0)
0 — No visibility of internal reads on external bus.
1 — Data from internal reads is driven out data bus.

PSEL3 – PSEL0 = Priority Select bit 3 thru bit 0
(May only be written if I bit in CC Register is 1.)

PSEL3	PSEL2	PSEL1	PSEL0	Interrupt Source Promoted
0	0	0	0	Timer overflow
0	0	0	1	Pulse accum. overflow
0	0	1	0	Pulse accum. input edge
0	0	1	1	SPI serial xfer complete
0	1	0	0	SCI serial system
0	1	0	1	Reserved (default to IRQ)
0	1	1	0	IRQ (ext. pin or parallel I/O)
0	1	1	1	Real time interrupt
1	0	0	0	Timer input capture 1
1	0	0	1	Timer input capture 2
1	0	1	0	Timer input capture 3
1	0	1	1	Timer output compare 1
1	1	0	0	Timer output compare 2
1	1	0	1	Timer output compare 3
1	1	1	0	Timer output compare 4
1	1	1	1	Timer output compare 5

INIT

RAM and I/O Mapping Register

	B7							B0
$_03D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0
RESET=	0	0	0	0	0	0	0	1

(Time protected)

RAM3 – RAM0 = RAM map position
REG3 – REG0 = 64-byte register block map position

OC1D

Output Compare 1 Data Register

	B7							B0
$_00D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3			

If OC1Mx is set, data in OC1Dx is output to port A bit-x on successful OC1 compares.

OC1M

Output Compare 1 Mask Register

	B7							B0
$_00C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3			
RESET=	0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

OC1M

System Configuration Options

	B7							B0
$_039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0
RESET=	0	0	0	1	0	0	0	0

APDU = A to D Power Up
- 0 = A/D system powered down
- 1 = A/D system powered up

CSEL = Clock Select
(Should be set to one if E clock less than 1 MHz)
- 0 = A/D and EE use system E clock
- 1 = A/D and EE use an internal R-C clock

IRQE = IRQ Select Edge Sensitive Only (Time protected)
- 0 = IRQ configured for low LEVEL
- 1 = IRQ configured for falling EDGEs

DLY = Enable Oscillator Start-Up Delay (On exit from stop) (Time protected)
- 0 = No delay
- 1 = A delay is imposed

CME = Clock Monitor Enable
- 0 = Disabled
- 1 = Slow or stopped clocks cause reset

CR1, CR0 = COP Timer Rate Select Bits (Time protected)

C C R R 1 0	E/2^{15} Divided by	Xtal = 2^{23} Timeout −0/+15.6 ms	Xtal = 8.0 MHz Timeout −0/+16.4 ms	Xtal = 4.0 MHz Timeout −0/+32.8 ms
0 0	1	15.625 ms	16.384 ms	32.768 ms
0 1	4	62.5 ms	65.536 ms	131.07 ms
1 0	16	250 ms	262.14 ms	524.29 ms
1 1	64	1 s	1.049 s	2.1 s
	E =	2.1 MHz	2.0 MHz	1.0 MHz

PACNT

Pulse Accumulator Count Register

	B7							B0
$_027	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0

(Readable and writable)

PACTL

Pulse Accumulator Control Register

	B7							B0
$_026	DDRA7	PAEN	PAMOD	PEDGE			RTR1	RTR0
RESET=	0	0	0	0	0	0	0	0

DDRA7 = Data Direction for Port A Bit 7
0 = Input
1 = Output

PAEN = Pulse Accumulator System Enable
0 = Disabled
1 = Enabled

PAMOD = Pulse Accumulator Mode
0 = Event counter
1 = Gated time accumulation

PEDGE = Pulse Accumulator Edge Control
0 = Falling edges, high level enables accumulation
1 = Rising edges, low level enables accumulation

RTR1, RTR0 = RTI Interrupt Rate

RTR 10	Divide E by	Xtal = 2^{23}	Xtal = 8.0 MHz	Xtal = 4.0 MHz
0 0	2^{13}	3.91 ms	4.10 ms	8.19 ms
0 1	2^{14}	7.81 ms	8.19 ms	16.38 ms
1 0	2^{15}	15.62 ms	16.38 ms	32.77 ms
1 1	2^{16}	31.25 ms	32.77 ms	65.54 ms
	E =	2.1 MHz	2.0 MHz	1.0 MHz

Parallel I/O Control Register

	B7							B0
$_002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB
RESET=	0	0	0	0	0	U	1	1

STAF = Strobe A Flag
0 = Inactive
1 = Set at active edge of STRA pin

STAI = Strobe A Interrupt Enable
0 = No hardware interrupt generated
1 = Hardware interrupt requested when STAF = 1

CWOM = Port C Wire-OR Mode
0 = Port C outputs normal
1 = Open-drain

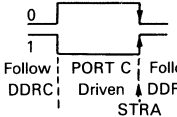
HNDS = Handshake/Simple Strobe Mode Select
0 = Simple strobe mode
1 = Full handshake modes

OIN = Output/Input Handshake Select
0 = Input
1 = Output

PLS = Pulse Mode Select for STRB Output
0 = STRB level active
1 = STRB pulses

EGA = Active Edge Select for STRA
0 = Hi to Lo (falling)
1 = Lo to Hi (rising)

INVB = Invert STRB Output
0 = STRB active low
1 = STRB active high

	Set by active edge on STRA						
	STAF Clearing Seq.	**HNDS**	**OIN**	**PLS**	**EGA**	**PORT C**	**PORT B**
Simple Strobe Mode	Read PIOC with STAF-1 then read PORTCL	0	X	X		Inputs latched into PORTCL on any active edge on STRA	STRB Pulses on writes to PORT B
Full Input Handshake	Read PIOC with STAF-1 then read PORTCL	1	0	0-STRB active LEVEL 1-STRB active PULSE		Inputs latched into PORTCL on any active edge on STRA	Normal output port. Unaffected by handshake modes.
Full Output Handshake	Read PIOC with STAF-1 then read PORTCL	1	1	0-STRB active LEVEL 1-STRB active PULSE		Driven as outputs if STRA at active level. Follows DDRC if STRA not at active level.	Normal output port. Unaffected by handshake modes.

PORTA

Port A Data Register

	B7							B0
$_000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	PA7/ PAI/ OC1	PA6/ OC2/ OC1	PA5/ OC3/ OC1	PA4/ OC4/ OC1	PA3/ OC5/ OC1	PA2/ IC1	PA1/ IC2	PA0/ IC3
RESET=	HiZ	0	0	0	0	HiZ	HiZ	HiZ

PORTB

Port B Data Register

	B7							B0
$_004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
MODE 0 or BOOT	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	(RESET TO LOGIC ZEROS)							
MODE 1 or TEST	A15	A14	A13	A12	A11	A10	A9	A8

PORTC

Port C Data Register

	B7							B0
$_003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
MODE 0 or BOOT	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
	(RESET TO HiZ INPUTS)							
MODE 1 or TEST	A7/ D7	A6/ D6	A5/ D5	A4/ D4	A3/ D3	A2/ D2	A1/ D1	A0/ D0

PORTC

PORTCL

Port C Latched Data Register

	B7							B0
$_005	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0

(Writes affect port C pins, PORTCL address used in handshake clearing mechanism.)

PORTD

Port D Data Register

	B7							B0
$_008			PD5	PD4	PD3	PD2	PD1	PD0
MODE 0 or BOOT	STRB	STRA	PD5/ SS*	PD4/ SCK	PD3/ MOSI	PD2/ MISO	PD1/ TXD	PD0/ RXD
RESET	0 1	1 Pulse	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ
MODE 1 or TEST	R/W	AS	PD5/ SS*	PD4/ SCK	PD3/ MOSI	PD2/ MISO	PD1/ TXD	PD0/ RXD

PORTE

Port E Data Register

	B7							B0
$_00A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
	PE7/ AN7	PE6/ AN6	PE5/ AN5	PE4/ AN4	PE3/ AN3	PE2/ AN2	PE1/ AN1	PE0/ AN0

(Inputs only always HiZ)

PORTE

EEPROM Programming Register

	B7							B0
$_03B	ODD	EVEN		BYTE	ROW	ERASE	EELAT	EEPGM
RESET=	0	0	0	0	0	0	0	0

ODD = Program odd rows in half of EEPROM (Test only)
EVEN = Program even rows in half of EEPROM (Test only)
BYTE = Byte/Other EEPROM Erase Mode
0 = Row or all erase mode will be used
1 = Erase only one byte of EEPROM
ROW = Row/All EEPROM Erase Mode
0 = All 512 bytes of EEPROM to be erased
1 = Erase only one 16 byte row of EEPROM
ERASE = Erase/Normal Control of EEPROM
0 = Normal read or program mode
1 = Erase mode
EELAT = EEPROM Latch Control
0 = EEPROM addr and data bus configured for reads
1 = EEPROM busses configured for program or erase
(EEPROM cannot be read while EELAT = 1)
EEPGM = EEPROM Program Command
(Writable only if EELAT = 1)
0 = Program (or erase) power switched off
1 = Program (or erase) power switched on

NOTE

To program EEPROM, set EELAT, then write data to desired address and then set EEPGM for the required programming time. To erase EEPROM select ROW = 0 or 1, select BYTE = 0 or 1, set ERASE = EELAT = 1, write to an EEPROM address to be erased, and set EEPGM for the required erase time period.

SCCR1

SCI Control Register 1

	B7							B0
$_02C	R8	T8		M	WAKE			
RESET=	U	U	0	0	0	0	0	0

R8 = Receive bit 8
T8 = Transmit bit 8
M = Mode (select character format)
0 = 1 start, 8 data, 1 stop bit
1 = 1 start, 8 data, ninth data, 1 stop bit
WAKE = Wake up (by address mark/idle)
0 = Wake up by idle line
1 = Wake up by address mark

SCCR2

SCI Control Register 2

	B7							B0
$_02D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET=	0	0	0	0	0	0	0	0

TIE = Transmit Interrupt Enable
TCIE = Transmit Complete Interrupt Enable
RIE = Receiver Interrupt Enable
ILIE = Idle Line Interrupt Enable
0 = Inhibit interrupts
1 = Enable interrupts
TE = Transmitter Enable
(Toggle to queue idle character)
RE = Receiver Enable
0 = Off
1 = On
RWU = Receiver Wake-Up Control
0 = Normal
1 = Receiver asleep
SBK = Send Break

SCCR2

SCI Data Register

	B7							B0
$_02F	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0

(Receive and transmit double buffered)

SCI Status Register

	B7							B0
$_02E	TDRE	TC	RDRF	IDLE	OR	NF	FE	
RESET=	1	1	0	0	0	0	0	0

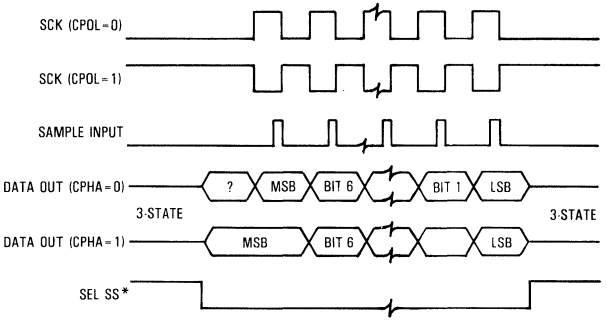
TDRE = Transmit Data Register Empty Flag
TC = Transmit Complete Flag
RDRF = Receive Data Register Full Flag
IDLE = Idle Line Detected Flag
OR = Over-Run Error Flag
NF = Noise Error Flag
FE = Framing Error Flag

SPCR

SPI Control Register

	B7							B0
$_028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET=	0	0	0	0	0	1	U	U

SPIE = SPI Interrupt Enable
SPE = SPI System Enable
DWOM = Port D Wire-Or Mode
0 = Port D outputs normal
1 = Open-drain
MSTR = Master/Slave Mode Select
0 = Slave mode
1 = Master mode
CPOL = Clock Polarity
CPHA = Clock Phase



SPR1, SPR0 = SPI Clock (SCK) Rate Select Bits

SPR1	SPR0	E-Clock Divided-by
0	0	2
0	1	4
1	0	16
1	1	32

SPDR

SPI Data Register

	B7							B0
$_02A	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0

(Double buffered in, single buffered out)

SPSR

SPI Status Register

	B7							B0
$_029	SPIF	WCOL		MODF				
RESET=	0	0	0	0	0	0	0	0

SPIF = SPI Interrupt Request
WCOL = Write Collision Status Flag
MODF = SPI Mode Error Interrupt Status Flag

TCNT

Timer Counter Register

	B7	TCNT (HIGH)						B0
$_00E	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8

	B7	TCNT (LOW)						B0
$_00F	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0

RESET: $0000 (Readable not writable)

TCTL1

Timer Control Register 1

	B7							B0
$_020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET=	0	0	0	0	0	0	0	0

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

TCTL2

Timer Control Register 2

	B7							B0
$_021			EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET=	0	0	0	0	0	0	0	0

Timer input capture edge specifications:

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

Factory Test Register

	B7							B0
$_03E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON
RESET=	0	0	0	0	–	0	0	0

(Restricted test modes only)

TILOP = Test Illegal Opcode
OCCR = Output Condition Code Register status to timer port
CBYP = Timer Divider Chain Bypass
DISR = Disable Resets from COP and Clock Monitor
FCM = Force Clock Monitor failure
FCOP = Force COP Watchdog failure
TCON = Test Configuration

TFLG1

Main Timer Interrupt Flag Reg. 1

	B7							B0
$_023	OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F
RESET=	0	0	0	0	0	0	0	0

OC1F-OC5F = Output Compare "x" Flag
IC1F-IC3F = Input Capture "x" Flag

(Write with bit(s) set to clear corresponding flag(s).)

TFLG2

Misc. Timer Interrupt Flag Reg. 2

	B7							B0
$_025	TOF	RTIF	PAOVF	PAIF				
RESET=	0	0	0	0	0	0	0	0

TOF = Timer Overflow Flag
RTIF = Real Time (periodic) Interrupt Flag
PAOVF = Pulse Accumulator Overflow Flag
PAIF = Pulse Accumulator Input edge Flag

(Write with bit(s) set to clear corresponding flag(s).)

Timer Input Capture Registers

	B7			TIC1 (HIGH)				B0
$_010	IC115	IC114	IC113	IC112	IC111	IC110	IC19	IC18

	B7			TIC1 (LOW)				B0
$_011	IC17	IC16	IC15	IC14	IC13	IC12	IC11	IC10

	B7			TIC2 (HIGH)				B0
$_012	IC215	IC214	IC213	IC212	IC211	IC210	IC29	IC28

	B7			TIC2 (LOW)				B0
$_013	IC27	IC26	IC25	IC24	IC23	IC22	IC21	IC20

	B7			TIC3 (HIGH)				B0
$_014	IC315	IC314	IC313	IC312	IC311	IC310	IC39	IC38

	B7			TIC3 (LOW)				B0
$_015	IC37	IC36	IC35	IC34	IC33	IC32	IC31	IC30

Main Timer Interrupt Mask Reg. 1

	B7							B0
$_022	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I
RESET=	0	0	0	0	0	0	0	0

OC1I – OC5I = Output Compare "x" Interrupt enable
IC1I – IC3I = Input Capture "x" Interrupt enable
0 = Interrupt inhibited
1 = Interrupt requested if flag set

TMSK2

Misc. Timer Interrupt Mask Reg. 2

	B7							B0
$_024	TOI	RTII	PAOVI	PAII			PR1	PR0
RESET=	0	0	0	0	0	0	0	0

TOI = Timer Overflow Interrupt enable

RTII = RTI Interrupt enable

PAOVI = Pulse Accumulator Overflow Interrupt enable

PAII = Pulse Accumulator Input Interrupt enable

0 = Interrupt inhibited

1 = Interrupt requested if flag set

PR1, PR0 = Timer Prescaler select (Time protected)

PR1	PR0	Prescale Factor
0	0	1
0	1	4
1	0	8
1	1	16

TOC1-TOC5

Timer Output Compare Registers

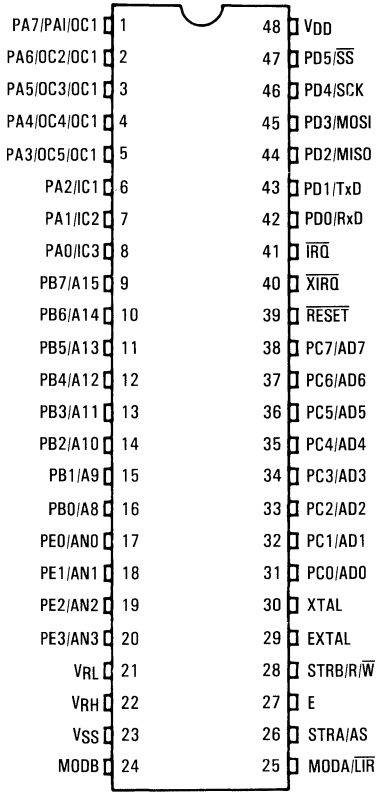
	B7			TOC1 (HIGH)				B0
$_016	OC115	OC114	OC113	OC112	OC111	OC110	OC19	OC18
	B7			TOC1 (LOW)				B0
$_017	OC17	OC16	OC15	OC14	OC13	OC12	OC11	OC10
	B7			TOC2 (HIGH)				B0
$_018	OC215	OC214	OC213	OC212	OC211	OC210	OC29	OC28
	B7			TOC2 (LOW)				B0
$_019	OC27	OC26	OC25	OC24	OC23	OC22	OC21	OC20
	B7			TOC3 (HIGH)				B0
$_01A	OC315	OC314	OC313	OC312	OC311	OC310	OC39	OC38
	B7			TOC3 (LOW)				B0
$_01B	OC37	OC36	OC35	OC34	OC33	OC32	OC31	OC30
	B7			TOC4 (HIGH)				B0
$_01C	OC415	OC414	OC413	OC412	OC411	OC410	OC49	OC48
	B7			TOC4 (LOW)				B0
$_01D	OC47	OC46	OC45	OC44	OC43	OC42	OC41	OC40
	B7			TOC5 (HIGH)				B0
$_01E	OC515	OC514	OC513	OC512	OC511	OC510	OC59	OC58
	B7			TOC5 (LOW)				B0
$_01F	OC57	OC56	OC55	OC54	OC53	OC52	OC51	OC50

(RESET: $FFFF all TOCx registers)

TOC1-TOC5

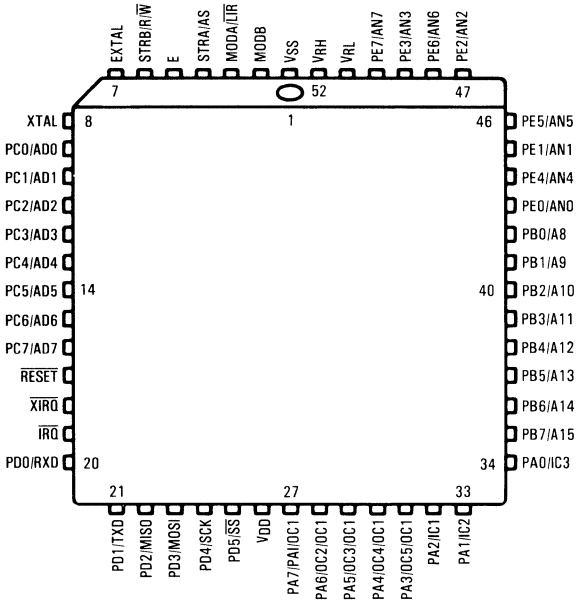
MC68HC11A8
MICROCOMPUTER

PIN ASSIGNMENT
(DUAL-IN-LINE)



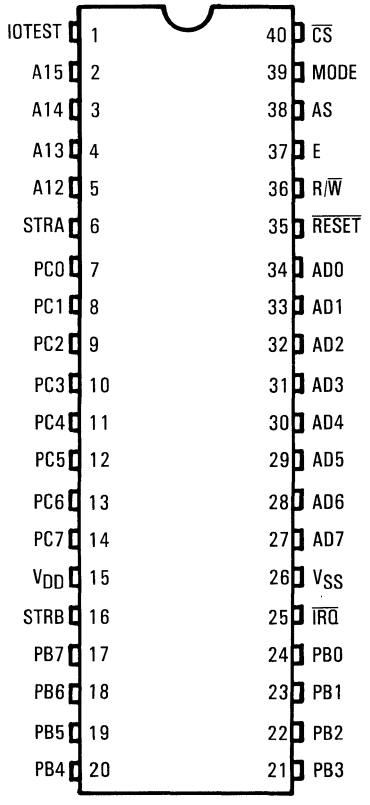
MC68HC11A8
MICROCOMPUTER

PIN ASSIGNMENT
(QUAD)



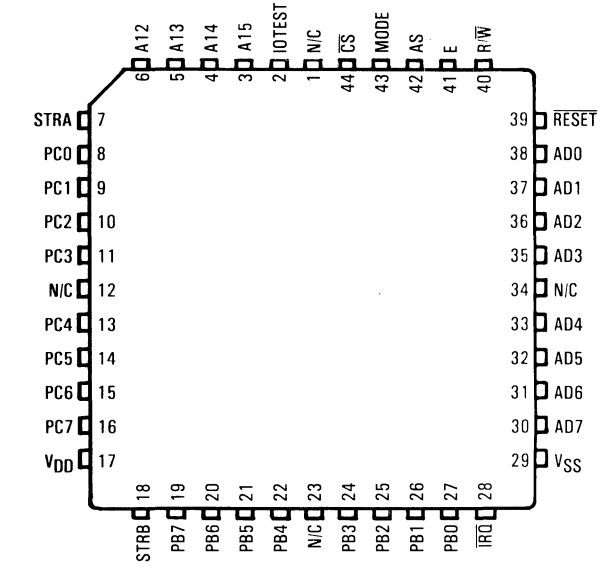
MC68HC24
PORT REPLACEMENT UNIT (PRU)

PIN ASSIGNMENT
(DUAL-IN-LINE)



MC68HC24
PORT REPLACEMENT UNIT (PRU)

PIN ASSIGNMENT
(QUAD)



HEXADECIMAL AND DECIMAL CONVERSION

How to use:

Conversion to Decimal: Find the decimal weights for corresponding hexadecimal characters beginning with the least significant character. The sum of the decimal weights is the decimal value of the hexadecimal number.

Conversion to Hexadecimal: Find the highest decimal value in the table which is lower than or equal to the decimal number to be converted. The corresponding hexadecimal character is the most significant. Subtract the decimal value found from the decimal number to be converted. With the difference repeat the process to find subsequent hexadecimal characters.

15		Byte			8	7		Byte			0
15	Char	12	11	Char	8	7	Char	4	3	Char	0
Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec
0		0	0		0	0		0	0		0
1		4,096	1		256	1		16	1		1
2		8,192	2		512	3		32	2		2
3		12,288	3		768	3		48	3		3
4		16,384	4		1,024	4		64	4		4
5		20,480	5		1,280	5		80	5		5
6		24,576	6		1,536	6		96	6		6
7		28,672	7		1,792	7		112	7		7
8		32,768	8		2,048	8		128	8		8
9		36,864	9		2,304	9		144	9		9
A		40,960	A		2,560	A		160	A		10
B		45,056	B		2,816	B		176	B		11
C		49,152	C		3,072	C		192	C		12
D		53,248	D		3,328	D		208	D		13
E		57,344	E		3,584	E		224	E		14
F		61,440	F		3,840	F		240	F		15

ASCII CHART

ASCII CHARACTER SET (7-Bit Code)								
MS Dig. / LS Dig.	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

PROGRAMMING MODEL
CRYSTAL DEPENDENT TIMING
INTERRUPTS

MEMORY MAP
OPCODE MAPS

INSTRUCTIONS
ADDRESSING MODES
EXECUTION TIMES
SPECIAL OPERATIONS

REGISTER AND
CONTROL BIT
ASSIGNMENTS

MECHANICAL DATA
HEX/DEC CONVERSION
ASCII CHART

PROGRAMMING MODEL
CRYSTAL DEPENDENT TIMING
INTERRUPTS

MEMORY MAP
OPCODE MAPS

INSTRUCTIONS
ADDRESSING MODES
EXECUTION TIMES
SPECIAL OPERATIONS

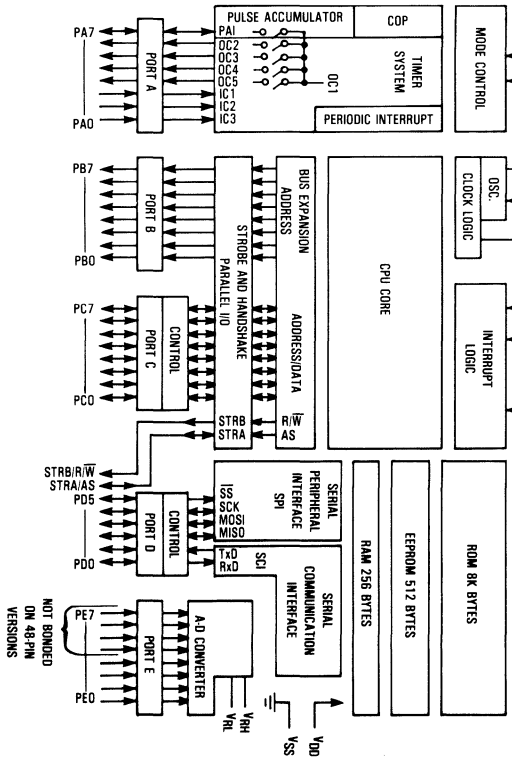
REGISTER AND
CONTROL BIT
ASSIGNMENTS

MECHANICAL DATA
HEX/DEC CONVERSION
ASCII CHART

NOTES

NOTES

BLOCK DIAGRAM



A18695 PRINTED IN USA 9-85 IMPERIAL LITHO C33808 50,000

MC68HC11A8

HCMOS Single-Chip Microcomputer

USA: Motorola Literature Distribution
P.O. Box 20912; Phoenix, Arizona 85036
EUROPE: Motorola Ltd.; European Literature Center
88 Tanners Drive, Blakelands Milton Keynes, MK145BP, England
HONG KONG: Motorola Inc.; International Semiconductor Group
P.O. Box 80300; Cheung Sha Wan Post Office; Kowloon Hong Kong

MOTOROLA