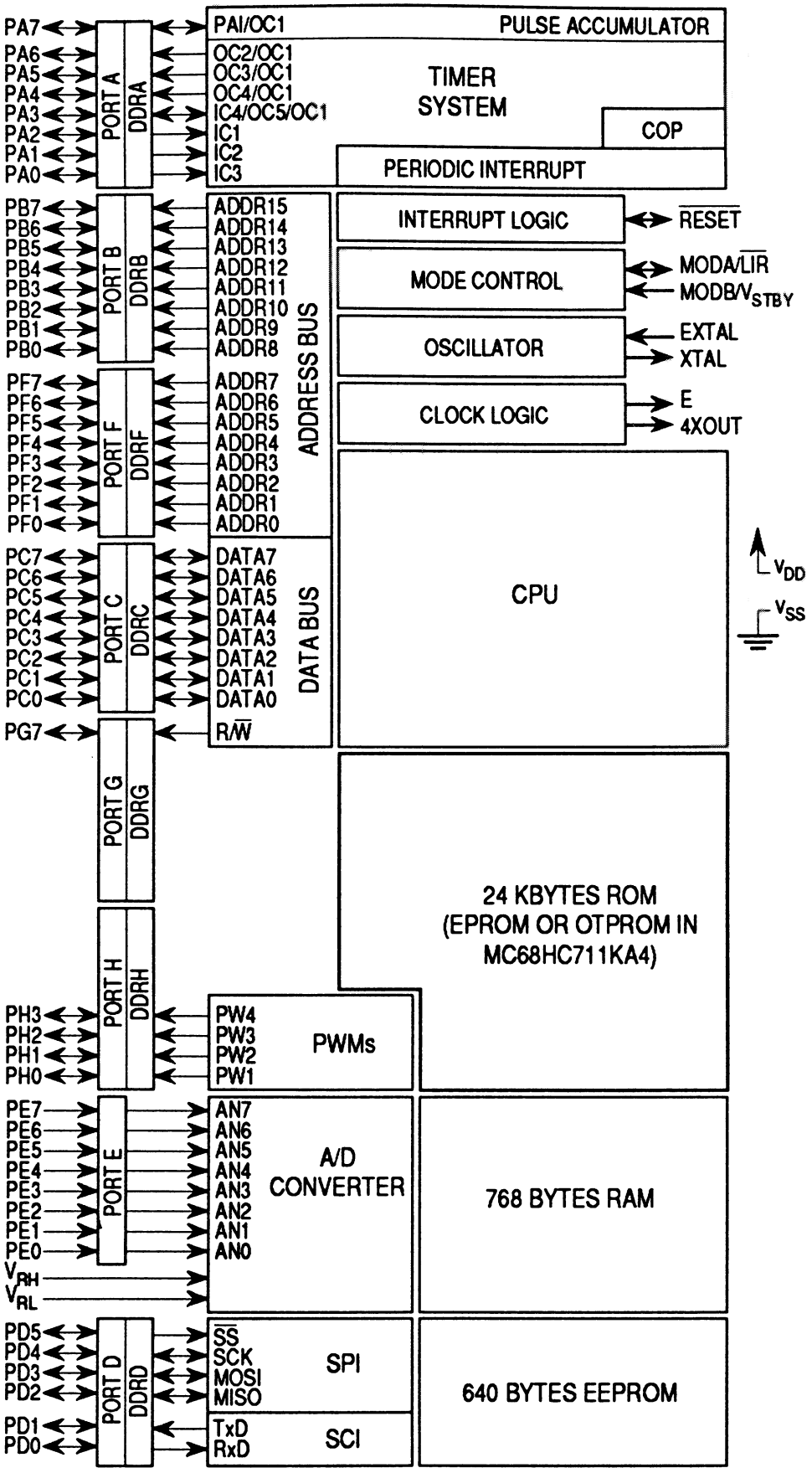
HC11

MC68HC11KA4
MC68HC711KA4

PROGRAMMING REFERENCE GUIDE

MOTOROLA

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

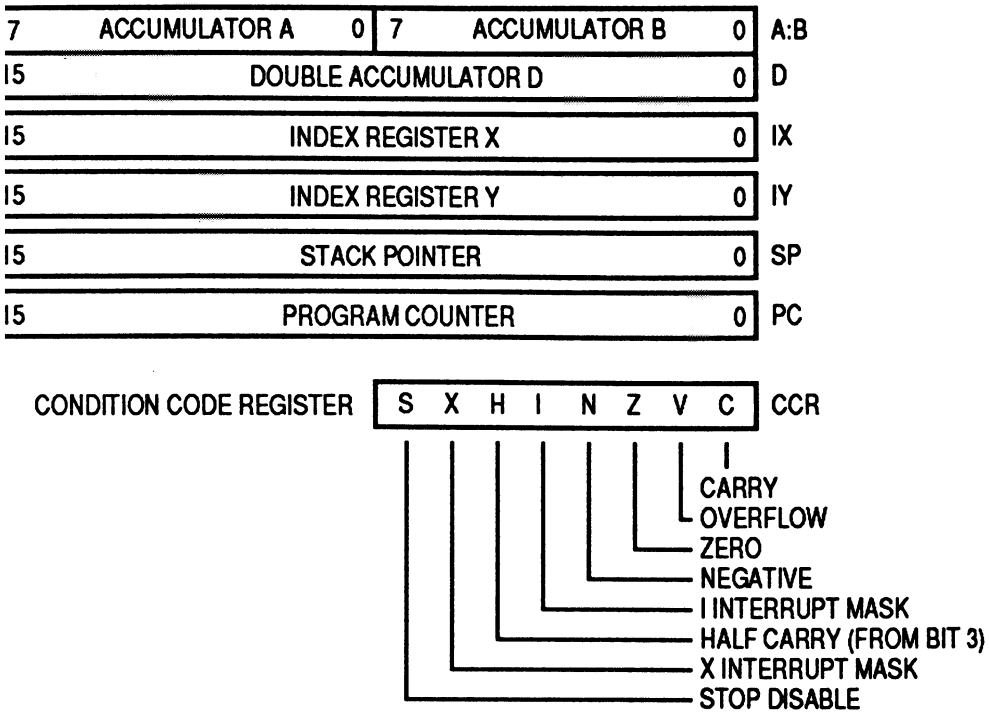
PROGRAMMING MODEL
CRYSTAL DEPENDENT TIMING
INTERRUPTS

MEMORY MAP
OPCODE MAPS

INSTRUCTIONS
ADDRESSING MODES
EXECUTION TIMES
SPECIAL OPERATIONS

REGISTER AND
CONTROL BIT
ASSIGNMENTS

MECHANICAL DATA
HEX/DEC CONVERSION
ASCII CHART



SCI Baud Rate Control Values

Target Baud Rate	Crystal Frequency					
	8 MHz		12 MHz		16 MHz	
	Dec Value	Hex Value	Dec Value	Hex Value	Dec Value	Hex Value
110	2272	$08E0	3409	$0D51	4545	$11C1
150	1666	$0682	2500	$09C4	3333	$0D05
300	833	$0341	1250	$04E2	1666	$0682
600	416	$01A0	625	$0271	833	$0341
1200	208	$00D0	312	$0138	416	$01A0
2400	104	$0068	156	$009C	208	$00D0
4800	52	$0034	78	$004E	104	$0068
9600	26	$001A	39	$0027	52	$0034
19.2 K	13	$000D	20	$0014	26	$001A
38.4 K	—	—	—	—	13	$000D

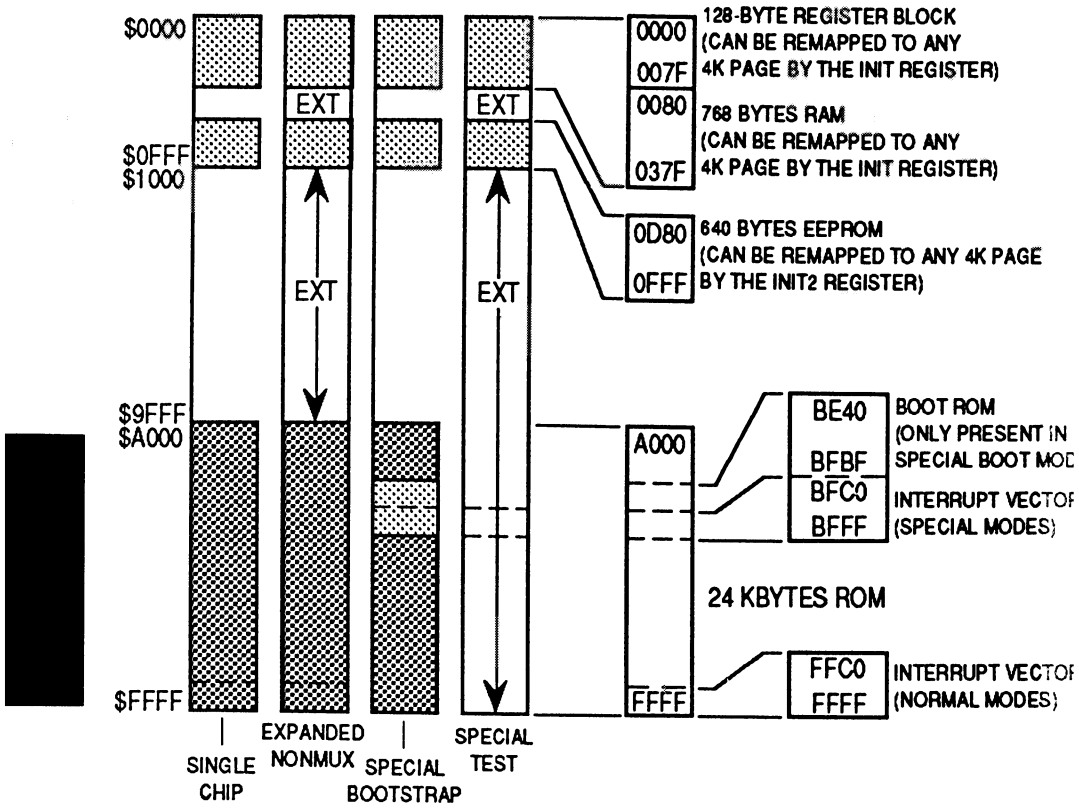
Crystal Dependent Timer Summary

	Selected Crystal	Common XTAL Frequencies		
		8.0 MHz	12.0 MHz	16.0 MHz
CPU Clock	(E)	2.0 MHz	3.0 MHz	4.0 MHz
Cycle Time	(1/E)	500 ns	333 ns	250 ns
	Pulse Accumulator (in Gated Mode)			
($E/2^6$) ($E/2^{14}$)	1 count — overflow —	32.0 µs 8.192 ms	21.330 µs 5.491 ms	16.0 µs 4.096 ms
	PR[1:0]	Main Timer Count Rates		
(E/1) ($E/2^{16}$)	0 0 1 count — overflow —	500 ns 32.768 ms	333 ns 21.845 ms	250 ns 16.384 ms
(E/4) ($E/2^{18}$)	0 1 1 count — overflow —	2.0 µs 131.07 ms	1.333 µs 87.381 ms	1.0 µs 65.536 ms
(E/8) ($E/2^{19}$)	1 0 1 count — overflow —	4.0 µs 262.14 ms	2.667 µs 174.76 ms	2.0 µs 131.07 ms
(E/16) ($E/2^{20}$)	1 1 1 count — overflow —	8.0 µs 524.29 ms	5.333 µs 349.52 ms	4.0 µs 262.14 ms
	RTR[1:0]	Periodic (RTI) Interrupt Rates		
($E/2^{13}$)	0 0	4.096 ms	2.731 ms	2.048 ms
($E/2^{14}$)	0 1	8.192 ms	5.491 ms	4.096 ms
($E/2^{15}$)	1 0	16.384 ms	10.923 ms	8.192 ms
($E/2^{16}$)	1 1	32.768 ms	21.845 ms	16.384 ms
	CR[1:0]	COP Watchdog Timeout Rates		
($E/2^{15}$)	0 0	16.384 ms	10.923 ms	8.192 ms
($E/2^{17}$)	0 1	65.536 ms	43.691 ms	32.768 ms
($E/2^{19}$)	1 0	262.14 ms	174.76 ms	131.07 ms
($E/2^{21}$)	1 1	1.049 s	699.05 ms	5024.28 ms
($E/2^{15}$)	Timeout Tolerance (– 0 ms/+...)	16.4 ms	10.9 ms	8.192 ms

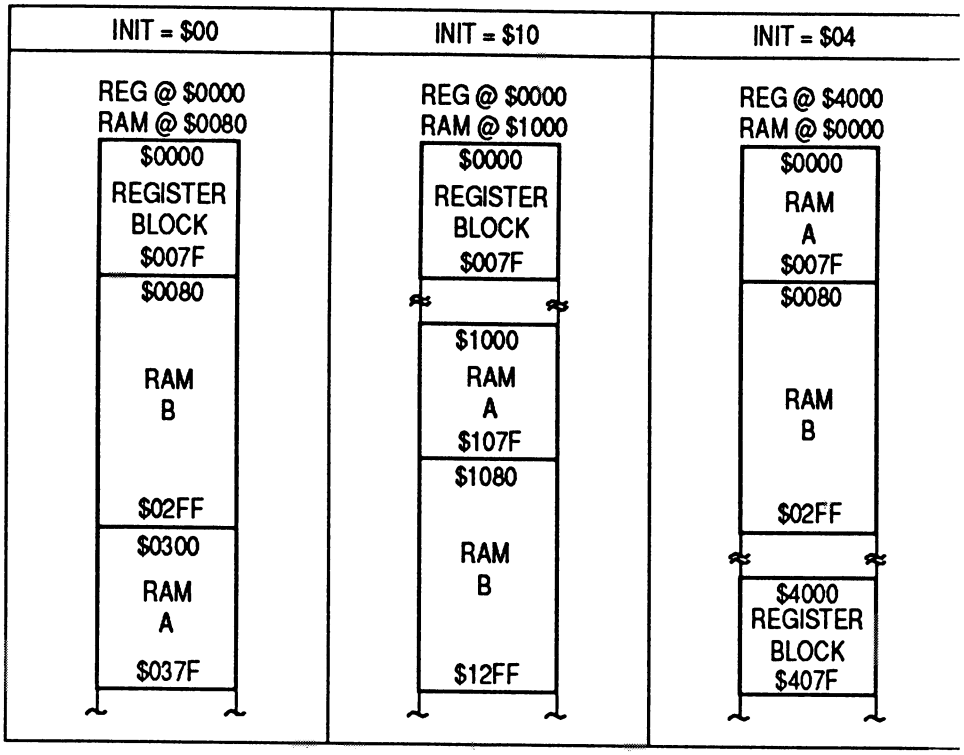
Interrupt Vector Assignments

Vector Address	Interrupt Source	CCR Mask Bit	Local Mask
FFC0, C1 – FFD4, D5	Reserved	—	—
FFD6, D7	SCI Serial System*	I	
	• SCI Receive Data Register Full		RIE
	• SCI Receiver Overrun		RIE
	• SCI Transmit Data Register Empty		TIE
	• SCI Transmit Complete		TCIE
	• SCI Idle Line Detect		ILIE
FFD8, D9	SPI Serial Transfer Complete	I	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I	PAII
FFDC, DD	Pulse Accumulator Overflow	I	PAOVI
FFDE, DF	Timer Overflow	I	TOI
FFE0, E1	Timer Input Capture 4/ Output Compare 5	I	I4/O5I
FFE2, E3	Timer Output Compare 4	I	OC4I
FFE4, E5	Timer Output Compare 3	I	OC3I
FFE6, E7	Timer Output Compare 2	I	OC2I
FFE8, E9	Timer Output Compare 1	I	OC1I
FFEA, EB	Timer Input Capture 3	I	IC3I
FFEC, ED	Timer Input Capture 2	I	IC2I
FFEE, EF	Timer Input Capture 1	I	IC1I
FFF0, F1	Real-Time Interrupt	I	RTII
FFF2, F3	$\overline{\text{IRQ}}$ (External Pin)	I	None
FFF4, F5	$\overline{\text{XIRQ}}$ Pin	X	None
FFF6, F7	Software Interrupt	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure	None	NOCOP
FFFC, FD	Clock Monitor Fail	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

*Interrupts generated by SCI; read SCSR to determine source



Register and RAM Mapping, Examples



OPCODE MAP PAGE 1

		INH	INH	REL	INH	ACCA	ACCB	IND,X	EXT	ACCA IMM	ACCA DIR	ACCA IND,X	ACCA EXT	ACCB IMM	ACCB DIR	ACCB IND,X	ACCB 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	EDIV	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

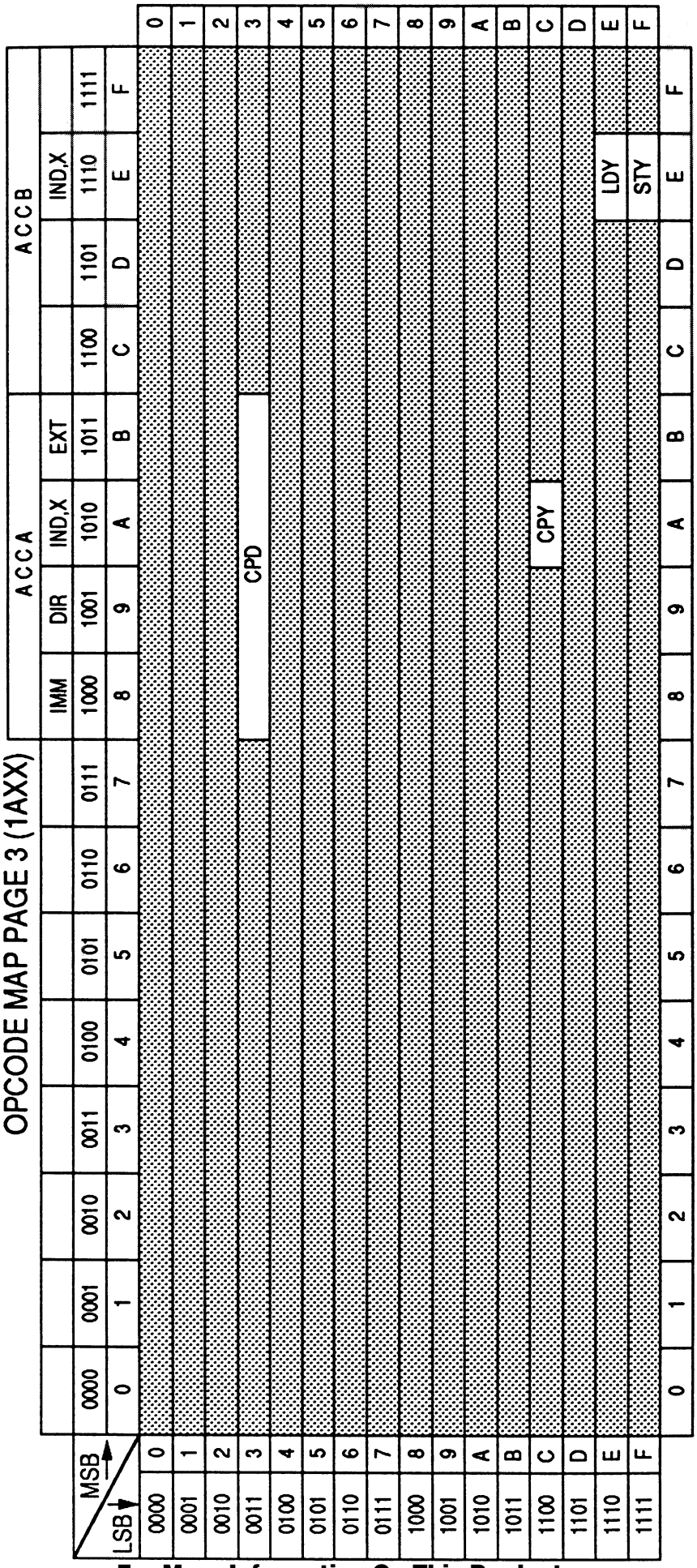
[redacted]

[redacted]

OPCODE MAP PAGE 2 (18XX)

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

IND Y



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																
0001	1																
0010	2																
0011	3											CPD					
0100	4																
0101	5																
0110	6																
0111	7																
1000	8																
1001	9																
1010	A																
1011	B																
1100	C											CPX					
1101	D																
1110	E															LDX	
1111	F															STX	
		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				
	True		False	
Test	Instruction	Opcode	Instruction	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				
	True		False	
Test	Instruction	Opcode	Instruction	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				
	True		False	
Test	Instruction	Opcode	Instruction	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	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	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.

[redacted]

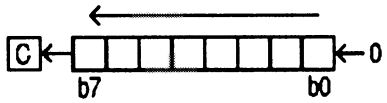
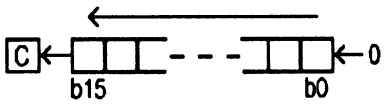
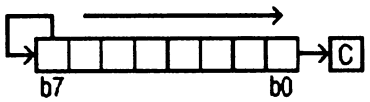
INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes S	X	H	I	N	Z	V	C
ABA	Add Accumulators	A + B → A	INH	1B		1	2	—	—	Δ	—	Δ	Δ	Δ	Δ
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	—	—	Δ	—	Δ	Δ	Δ	Δ
			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	—	—	Δ	—	Δ	Δ	Δ	Δ
			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								

ADDA (opr)	Add Memory to A	A + M → A	A	IMM	8B	ii		2	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			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	—	—	Δ	—	Δ	Δ	Δ	Δ
			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	—	—	—	—	Δ	Δ	Δ	Δ
				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	—	—	—	—	Δ	Δ	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								

[redacted]

[redacted]

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes S	X	H	I	N	Z	V	C
ANDB (opr)	AND B with Memory	B • M → B	B IMM	C4	ii	2	2	—	—	—	—	Δ	Δ	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 (opr)	Arithmetic Shift Left		EXT	78	hh ll	3	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	68	ff	2	6								
			IND,Y	18 68	ff	3	7								
			A INH	48		1	2								
			B INH	58		1	2								
ASLD	Arithmetic Shift Left Double		INH	05		1	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR (opr)	Arithmetic Shift Right		EXT	77	hh ll	3	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	67	ff	2	6								
			IND,Y	18 67	ff	3	7								
			A INH	47		1	2								
			B INH	57		1	2								
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	2	3	—	—	—	—	—	—	—	—

BCLR	(opr) (msk)	Clear Bit(s)	M • (mm) → M		DIR IND,X IND,Y	15 1D 18 1D	dd ff ff	mm mm mm		3 3 4	6 7 8	—	—	—	—	Δ	Δ	0	—
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	—	—	—	—	—	—	—	—
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 A A A A	IMM DIR EXT IND,X IND,Y	85 95 B5 A5 18 A5	ii dd hh ff ff		‖	2 2 3 2 3	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BITB	(opr)	Bit(s) Test B with Memory	B • M	B B B B B	IMM DIR EXT IND,X IND,Y	C5 D5 F5 E5 18 E5	ii dd hh ff ff		‖	2 2 3 2 3	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—

[redacted]

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes S X H I N Z V C
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) (msk) (rel)	Branch if Bit(s) Clear	?M • mm = 0	DIR IND,X IND,Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	4 4 5	6 7 8	— — — — — — — —
BRN (rel)	Branch Never	?1 = 0	REL	21	rr	2	3	— — — — — — — —
BRSET (opr) (msk) (rel)	Branch if Bit(s) Set	?(M) • mm = 0	DIR IND,X IND,Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	4 4 5	6 7 8	— — — — — — — —

BSET	(opr) (msk)	Set Bit(s)	M + mm → M	DIR IND,X IND,Y	14 1C 18 1C	dd mm ff mm ff mm	3 3 4	6 7 8	—	—	—	—	Δ	Δ	0	—
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	—	—	—	—	Δ	Δ	Δ	Δ
CLC		Clear Carry Bit	0 → C	INH	0C		1	2	—	—	—	—	—	—	—	0
CLI		Clear Interrupt Mask	0 → I	INH	0E		1	2	—	—	—	0	—	—	—	—
CLR	(opr)	Clear Memory Byte	0 → M	EXT IND,X IND,Y	7F 6F 18 6F	hh ll ff ff	3 2 3	6 6 7	—	—	—	—	0	1	0	0
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	—

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes S	X	H	I	N	Z	V	C
CMPA (opr)	Compare A to Memory	A – M	A IMM	81	ii	2	2	—	—	—	—	Δ	Δ	Δ	Δ
			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	—	—	—	—	Δ	Δ	Δ	Δ
			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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	0	1
COMB	1's Complement B	$FF – B → B	B INH	53		1	2	—	—	—	—	Δ	Δ	0	1
CPD (opr)	Compare D to Memory 16-Bit	D – M:M + 1	IMM	1A 83	jj kk	4	5	—	—	—	—	Δ	Δ	Δ	Δ
			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 DIR EXT IND,X IND,Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff		3 2 3 2 3	4 5 6 6 7	—	—	—	—	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		4 3 4 3 3	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
DAA		Decimal Adjust A	Adjust Sum to BCD		INH	19			1	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		3 2 3	6 6 7	—	—	—	—	Δ	Δ	Δ	—
DECA		Decrement Accumulator A	A – 1 → A	A	INH	4A			1	2	—	—	—	—	Δ	Δ	Δ	—
DECB		Decrement Accumulator B	B – 1 → B	B	INH	5A			1	2	—	—	—	—	Δ	Δ	Δ	—
DES		Decrement Stack Pointer	SP – 1 → SP		INH	34			1	3	—	—	—	—	—	—	—	—
DEX		Decrement Index Register X	IX – 1 → IX		INH	09			1	3	—	—	—	—	—	Δ	—	—

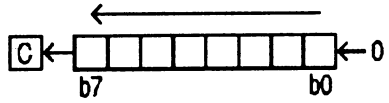
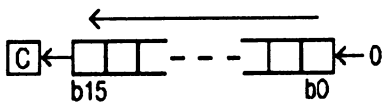
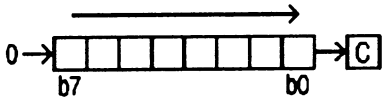
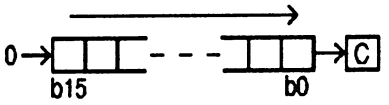
[redacted]

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes S	X	H	I	N	Z	V	C
DEY	Decrement Index Register Y	IY − 1 → IY	INH	18 09		2	4	—	—	—	—	—	Δ	—	—
EORA (opr)	Exclusive OR A with Memory	A ⊕ M → A	A IMM	88	ii	2	2	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	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	—	—	—	—	—	Δ	Δ	Δ
IDIV	Integer Divide 16 by 16	D/IX → IX; r → D	INH	02		1	41	—	—	—	—	—	Δ	0	Δ
INC (opr)	Increment Memory Byte	M + 1 → M	EXT	7C	hh ll	3	6	—	—	—	—	Δ	Δ	Δ	—
			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	—	—	—	—	Δ	Δ	Δ	—
INCB	Increment Accumulator B	B + 1 → B	B INH	5C		1	2	—	—	—	—	Δ	Δ	Δ	—

INS		Increment Stack Pointer	SP + 1 → SP		INH		31			1	3	—	—	—	—	—	—	—	—
INX		Increment Index Register X	IX + 1 → IX		INH		08			1	3	—	—	—	—	—	Δ	—	—
INY		Increment Index Register Y	IY + 1 → IY		INH		18 08			2	4	—	—	—	—	—	Δ	—	—
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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	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								

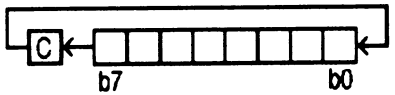
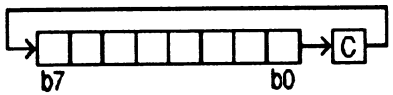
[redacted]

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	Δ	Δ
					IND,X	68	ff		2	6								
					IND,Y	18 68	ff		3	7								
				A	INH	48			1	2								
				B	INH	58			1	2								
LSLD		Logical Shift Left Double			INH	05			1	3	—	—	—	—	Δ	Δ	Δ	Δ
LSR	(opr)	Logical Shift Right			EXT	74	hh	ll	3	6	—	—	—	—	0	Δ	Δ	Δ
					IND,X	64	ff		2	6								
					IND,Y	18 64	ff		3	7								
				A	INH	44			1	2								
				B	INH	54			1	2								
LSRD		Logical Shift Right Double			INH	04			1	3	—	—	—	—	0	Δ	Δ	Δ
MUL		Multiply 8 by 8	A × B → D		INH	3D			1	10	—	—	—	—	—	—	—	Δ
NEG	(opr)	2's Complement Memory Byte	0 – M → M		EXT	70	hh	ll	3	6	—	—	—	—	Δ	Δ	Δ	Δ
					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	—	—	—	—	Δ	Δ	Δ	Δ
NEGB		2's Complement B	0 – B → B	B	INH	50			1	2	—	—	—	—	Δ	Δ	Δ	Δ
NOP		No Operation	No Operation		INH	01			1	2	—	—	—	—	—	—	—	—

[redacted]

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes S	X	H	I	N	Z	V	C
ORAA (opr)	OR Accumulator A (Inclusive)	A + M → A	A IMM	8A	ii	2	2	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	Δ	Δ
					IND,X	69	ff		2	6								
					IND,Y	18 69	ff		3	7								
				A	INH	49			1	2								
				B	INH	59			1	2								
ROR	(opr)	Rotate Right			EXT	76	hh	ll	3	6	—	—	—	—	Δ	Δ	Δ	Δ
					IND,X	66	ff		2	6								
					IND,Y	18 66	ff		3	7								
				A	INH	46			1	2								
				B	INH	56			1	2								
RTI		Return from Interrupt	See Special Ops		INH	3B			1	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
RTS		Return from Subroutine	See Special Ops		INH	39			1	5	—	—	—	—	—	—	—	—
SBA		Subtract B from A	A – B → A		INH	10			1	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA	(opr)	Subtract with Carry from A	A – M – C → A	A	IMM	82	ii		2	2	—	—	—	—	Δ	Δ	Δ	Δ
				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	—	—	—	—	Δ	Δ	Δ	Δ
				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								

[redacted]

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes 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 A EXT A IND,X A IND,Y	97 B7 A7 18 A7	dd hh ‖ ff ff	2 3 2 3	3 4 4 5	— — — — Δ Δ 0 —
STAB (opr)	Store Accumulator B	B → M	B DIR B EXT B IND,X B IND,Y	D7 F7 E7 18 E7	dd hh ‖ ff ff	2 3 2 3	3 4 4 5	— — — — Δ Δ 0 —
STD (opr)	Store Accumulator D	A → M, B → M + 1	DIR EXT IND,X IND,Y	DD FD ED 18 ED	dd hh ‖ ff ff	2 3 2 3	4 5 5 6	— — — — Δ Δ 0 —
STOP	Stop Internal Clocks		INH	CF		1	2	— — — — — — — —
STS (opr)	Store Stack Pointer	SP → M:M + 1	DIR EXT IND,X IND,Y	9F BF AF 18 AF	dd hh ‖ ff ff	2 3 2 3	4 5 5 6	— — — — Δ Δ 0 —

STX	(opr)	Store Index Register X	IX → M:M + 1		DIR	DF	dd		2	4	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	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	—	—	—	—	Δ	Δ	Δ	Δ
				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	—	—	—	—	Δ	Δ	Δ	Δ
				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	—	—	—	—	Δ	Δ	Δ	Δ
					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	—	—	—	—
TAB		Transfer A to B	A → B		INH	16			1	2	—	—	—	—	Δ	Δ	0	—
TAP		Transfer A to CCR	A → CCR		INH	06			1	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ

[redacted]

Source Forms	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycles	Condition Codes S	X	H	I	N	Z	V	C
TBA	Transfer B to A	B → A	INH	17		1	2	—	—	—	—	Δ	Δ	0	—
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00		1	*	—	—	—	—	—	—	—	—
TPA	Transfer CCR to A	CCR → A	INH	07		1	2	—	—	—	—	—	—	—	—
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	—	—	—	—	Δ	Δ	0	0
TSTA		A – 0	A INH	4D		1	2	—	—	—	—	Δ	Δ	0	0
TSTB		B – 0	B INH	5D		1	2	—	—	—	—	Δ	Δ	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 and WAIT	INH	3E		1	* *	—	—	—	—	—	—	—	—
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).

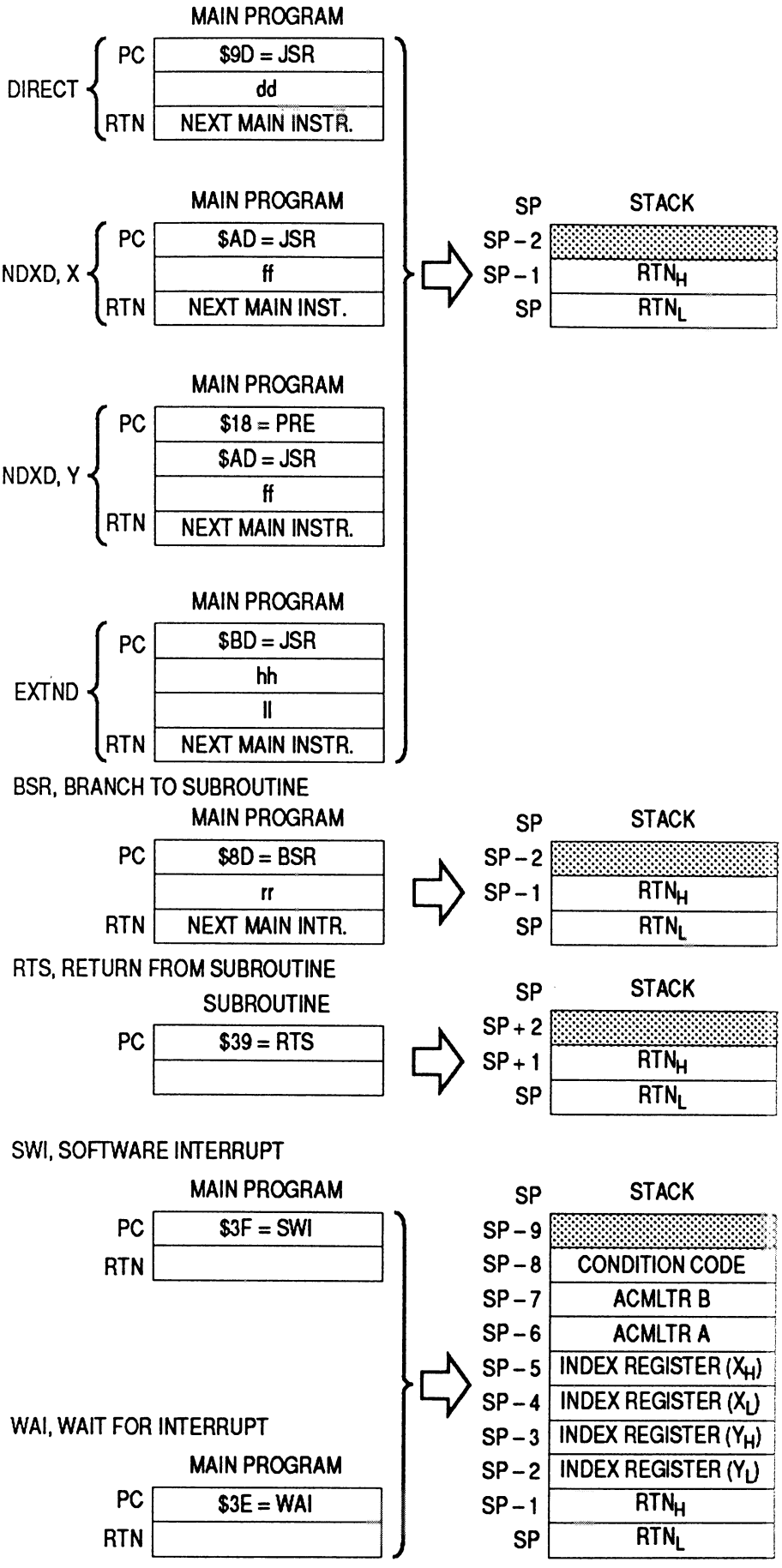
Operands:

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.

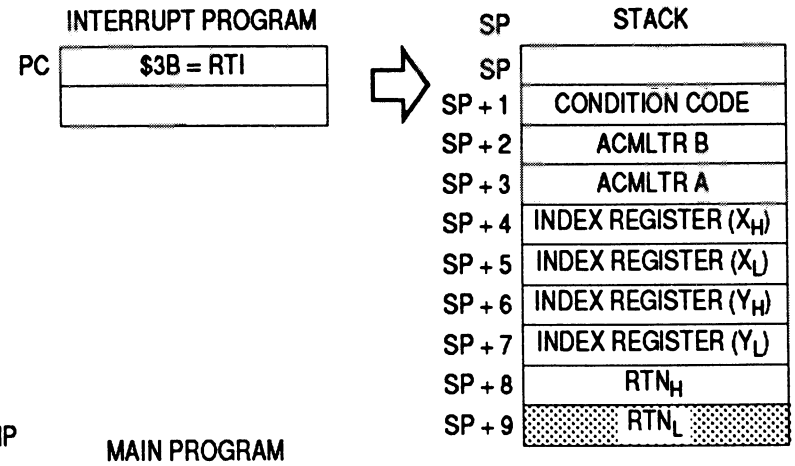
Condition Codes:

—	=	Bit not changed
0	=	Always cleared (logic 0).
1	=	Always set (logic 1).
Δ	=	Bit cleared or set depending on operation.
↓	=	Bit may be cleared, cannot become set.

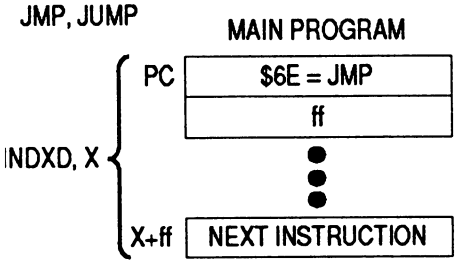
[redacted]

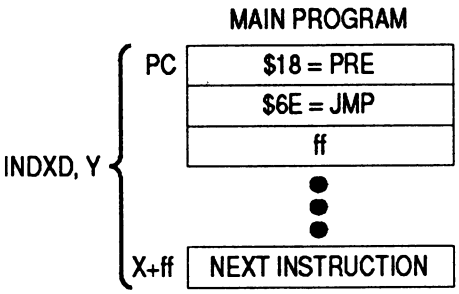


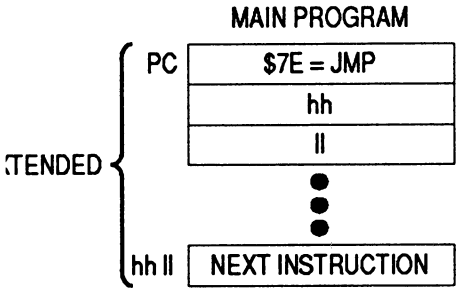
RTI, RETURN FROM INTERRUPT



JMP, JUMP







EGEND

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 location 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)

The 128-byte register block can be remapped to any 4K boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	DDRF
$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
$0008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
$0009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
$000E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (Hi
$000F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Lo
$0010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (Hi)
$0011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Lo)
$0012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (Hi)
$0013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Lo)
$0014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (Hi)
$0015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Lo)
$0016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1 (Hi)
$0017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Lo)
$0018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (Hi)
$0019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Lo)
$001A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (Hi)
$001B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Lo)
$001C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (Hi)
$001D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Lo)
$001E	Bit 15	14	13	12	11	10	9	Bit 8	TI4/O5 (H
$001F	Bit 7	6	5	4	3	2	1	Bit 0	TI4/O5 (L
$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2

Freescale MAP Registers (2 of 3)

	Bit 7	6	5	4	3	2	1	Bit 0	
$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	TMSK1
$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	TFLG1
$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	PACTL
$0027	Bit 7	6	5	4	3	2	1	Bit 0	PACNT
$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$0029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
$002A	Bit 7	6	5	4	3	2	1	Bit 0	SPDR
$002B	MBE	0	ELAT	EXCOL	EXROW	0	0	EPGM	EPROG
$002C	0	0	0	0	HPPUE	GPPUE	FPPUE	BPPUE	PPAR
$002D									Reserved
$002E									Reserved
$002F									Reserved
$0030	CCF	0	SCAN	MULT	CD	CC	CB	CA	ADCTL
$0031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
$0032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
$0033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
$0034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4
$0035	BULKP	LVPEN	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT
$0036									Reserved
$0037	EE3	EE2	EE1	EE0	0	0	0	0	INIT2
$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	XDV1	XDV0	OPT2
$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	OPTION
$003A	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
$003B	ODD	EVEN	LVPI	BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
$003E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	0	TEST1
$003F	ROMAD	1	CLKX	PAREN	1	NOCOP	ROMON	EEON	CONFIG
$0040 to $005F									Reserved

	Bit 7	6	5	4	3	2	1	Bit 0	
$0060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	PWCLK
$0061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	PWPOL
$0062	Bit 7	6	5	4	3	2	1	Bit 0	PWSCAL
$0063	TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1	PWEN
$0064	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT1
$0065	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT2
$0066	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT3
$0067	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT4
$0068	Bit 7	6	5	4	3	2	1	Bit 0	PWPER1
$0069	Bit 7	6	5	4	3	2	1	Bit 0	PWPER2
$006A	Bit 7	6	5	4	3	2	1	Bit 0	PWPER3
$006B	Bit 7	6	5	4	3	2	1	Bit 0	PWPER4
$006C	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY1
$006D	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY2
$006E	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY3
$006F	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY4
$0070	BTST	BSPL	0	SBR12	SBR11	SBR10	SBR9	SBR8	SCBDH
$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	SCBDL
$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	SCCR1
$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	SCSR1
$0075	0	0	0	0	0	0	0	RAF	SCSR2
$0076	R8	T8	0	0	0	0	0	0	SCDRH
$0077	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDRL
$0078									Reserved
to									
$007B									Reserved
$007C	0	0	0	0	PH3	PH2	PH1	PH0	PORTH
$007D	0	0	0	0	DDH3	DDH2	DDH1	DDH0	DDRH
$007E	PG7	0	0	0	0	0	0	0	PORTG
$007F	DDG7	0	0	0	0	0	0	0	DDRG

A/D Control/Status

	Bit 7	6	5	4	3	2	1	Bit 0
$0030	CCF	0	SCAN	MULT	CD	CC	CB	CA
RESET:	U	0	U	U	U	U	U	U

CCF — Conversions Complete Flag
This bit is set after an A/D conversion cycle and cleared when ADCTL is written.
Bit 6 — Not implemented; always reads zero
SCAN — Continuous Scan Control
0 = Do four conversions and stop
1 = Convert four channels in selected group continuously
MULT — Multiple-Channel/Single-Channel Control
0 = Convert single channel selected
1 = Convert four channels in selected group
CD–CA — Channel Select D through A

Channel Select Control Bits				Channel Signal	Result in ADRx if MULT = 1
CD	CC	CB	CA		
0	0	0	0	AN0	ADR1
0	0	0	1	AN1	ADR2
0	0	1	0	AN2	ADR3
0	0	1	1	AN3	ADR4
0	1	0	0	AN4	ADR1
0	1	0	1	AN5	ADR2
0	1	1	0	AN6	ADR3
0	1	1	1	AN7	ADR4
1	0	X	X	Reserved	ADR1–ADR4
1	1	0	0	V_{RH}*	ADR1
1	1	0	1	V_{RL}*	ADR2
1	1	1	0	$(V_{RH})/2$*	ADR3
1	1	1	1	Reserved*	ADR4

*Used for factory testing

A/D Results

$0031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
$0032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
$0033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
$0034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4

Analog Input to 8-Bit Result Translation Table

	Bit 7	6	5	4	3	2	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

(1) % of V_{RH}–V_{RL} (2) Volts for V_{RL} = 0; V_{RH} = 5.0 V

BPROT

Block Protect

	Bit 7	6	5	4	3	2	1	Bit 0
$0035	BULKP	LVPEN	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0
RESET:	1	1	1	1	1	1	1	1

BULKP — Bulk Erase of EEPROM Protect
- 0 = EEPROM can be bulk erased normally.
- 1 = EEPROM cannot be bulk or row erased.

LVPEN — Low Voltage Protect Enable
- 0 = Low voltage EEPROM protect disabled
- 1 = Low voltage EEPROM protect enabled

BPRT[4:0] — Block Protect Bits for EEPROM

Block protect register bits can be written to zero (protection disabled) only once within 64 cycles of a reset in normal modes, or at any time in special modes. Block protect register bits can be written to one (protection enabled) at any time.
- 0 = Protection disabled
- 1 = Protection enabled

Bit Name	Block Protected	Block Size
BPRT4	$xF80–$xFFF	128 Bytes
BPRT3	$xE60–$xF7F	288 Bytes
BPRT2	$xDE0–$xE5F	128 Bytes
BPRT1	$xDA0–$xDDF	64 Bytes
BPRT0	$xD80–$xD9F	32 Bytes

PTCON — Protect for CONFIG
- 0 = CONFIG register can be programmed or erased normally.
- 1 = CONFIG register cannot be programmed or erased.

Timer Compare Force

	Bit 7	6	5	4	3	2	1	Bit 0
$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
RESET:	0	0	0	0	0	0	0	0

FOC[5:1] — Write ones to Force Compare(s)
0 = Not affected
1 = Output x action occurs.
Bits [2:0] — Not implemented; always read 0

CONFIG

COP, ROM Mapping, EEPROM Enables

	Bit 7	6	5	4	3	2	1	Bit 0
$003F	ROMAD	1	CLKX	PAREN	1	NOCOP	ROMON	EEON
RESET:	—	1	—	—	1	—	—	—

The CONFIG bits can be read at any time. The value read is the value latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence. Unused bits always read as ones.

If SMOD = 1, these bits are writable any time. If SMOD = 0 these bits can only be written using the EEPROM programming sequence, and are neither readable nor active until latched via the next reset.

ROMAD — ROM Mapping Control
In single-chip mode, ROMAD is forced to one out of reset
0 = ROM addressed from $2000 to $7FFF
1 = ROM addressed from $A000 to $FFFF
Bit 6 — Not implemented; always reads one
CLKX — X Clock Enable
0 = XOUT pin disabled
1 = X clock driven out on XOUT pin
PAREN — Pull-Up Assignment Register Enable
0 = Disable PPAR register
1 = Enable PPAR register
Bit 3 — Not implemented; always reads one
NOCOP — COP System Disable
Resets to programmed value
0 = COP enabled (forces reset on timeout)
1 = COP disabled (does not force reset on timeout)
ROMON — ROM Enable
In single-chip mode ROMON is forced to one out of reset. In special test mode, ROMON is forced to zero out of reset.
0 = ROM removed from the memory map
1 = ROM present in the memory map

EEON — EEPROM Enable
0 = 640 bytes of EEPROM removed from the memory map
1 = 640 bytes of EEPROM present in the memory map with location depending on value specified in EE[3:0] in INIT2

COPRST

Arm/Reset COP Timer Circuitry

	Bit 7	6	5	4	3	2	1	Bit 0
$003A	7	6	5	4	3	2	1	0
RESET:	0	0	0	0	0	0	0	0

Write $55 to COPRST to arm COP watchdog clearing mechanism. Write $AA to COPRST to reset COP watchdog.

DDRA

Data Direction Register for Port A

	Bit 7	6	5	4	3	2	1	Bit 0
$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
RESET:	0	0	0	0	0	0	0	0

DDA[7:0] — Data Direction for Port A
0 = Input
1 = Output

DDRB

Data Direction Register for Port B

	Bit 7	6	5	4	3	2	1	Bit 0
$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
RESET:	0	0	0	0	0	0	0	0

DDB[7:0] — Data Direction for Port B
0 = Input
1 = Output

DDRC

Data Direction Register for Port C

	Bit 7	6	5	4	3	2	1	Bit 0
$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
RESET:	0	0	0	0	0	0	0	0

DDC[7:0] — Data Direction for Port C
0 = Input
1 = Output

DDRD

Data Direction Register for Port D

	Bit 7	6	5	4	3	2	1	Bit 0
$0009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
RESET:	0	0	0	0	0	0	0	0

Bits [7:6] — Not implemented; always read zero
DDD[5:0] — Data Direction for Port D
0 = Input
1 = Output

DDRF

Data Direction Register for Port F

	Bit 7	6	5	4	3	2	1	Bit 0
$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
RESET:	0	0	0	0	0	0	0	0

DDF[7:0] — Data Direction for Port F
0 = Input
1 = Output

DDRG

Data Direction Register for Port G

	Bit 7	6	5	4	3	2	1	Bit 0
$007F	DDG7	0	0	0	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

DDG7 — Data Direction for Port G
0 = Input
1 = Output
Bits [6:0] — Not implemented; always read zero

Data Direction Register for Port H

	Bit 7	6	5	4	3	2	1	Bit 0
$007D	0	0	0	0	DDH3	DDH2	DDH1	DDH0
RESET:	0	0	0	0	0	0	0	0

Bits [7:4] — Not implemented; always read zero
DDH[3:0] — Data Direction for Port H
0 = Bits cleared to zero to configure corresponding I/O pin for input only
1 = Bits set to one to configure corresponding I/O pin for output

NOTE

PWM circuitry forces the I/O state to be an output for each port H line associated with an enabled pulse-width modulator in any mode. In this case, data direction bits are not changed and have no effect on these lines. DDRH reverts to controlling the I/O state of a pin when the associated function is disabled.

HPRIO

Highest Priority I-Bit Interrupt and Miscellaneous

	Bit 7	6	5	4	3	2	1	Bit 0
$003C	RBOOT*	SMOD*	MDA*	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0
RESETS:								
Single-Chip Mode	0	0	0	0	0	1	1	0
Exp'd Nonmux'd	0	0	1	0	0	1	1	0
Bootstrap	1	1	0	0	0	1	1	0
Special Test	0	1	1	0	0	1	1	0

*RBOOT, SMOD, and MDA resets depend on mode selected at power-up.

RBOOT — Read Bootstrap ROM
Valid only when SMOD is set to one (special bootstrap or special test mode). Can only be written in special modes.
0 = Bootloader ROM disabled and not in map
1 = Bootloader ROM enabled and in map at $BE40–$BFFF

SMOD and MDA — Special Mode Select and Mode Select A
These two bits can be read at any time. SMOD can only be written in special modes. MDA can be written at any time in special modes, but only once in normal mode.

Inputs		Mode	Latched at Reset	
MODB	MODA		SMOD	MDA
1	0	Single Chip	0	0
1	1	Expanded Nonmultiplexed	0	1
0	0	Special Bootstrap	1	0
0	1	Special Test	1	1

PSEL[4:0] — Priority Select Bit 4 through Bit 0
Writable only while bit I in the CCR is set (interrupts disabled). These bits select one interrupt source to be elevated above all other I-bit related sources.

PSELx					
4	3	2	1	0	Interrupt Source Promoted
0	0	0	X	X	Reserved (Default to $\overline{IRQ}$)
0	0	1	0	0	Reserved (Default to $\overline{IRQ}$)
0	0	1	0	1	Reserved (Default to $\overline{IRQ}$)
0	0	1	1	0	$\overline{IRQ}$ (External Pin)
0	0	1	1	1	Real-Time Interrupt
0	1	0	0	0	Timer Input Capture 1
0	1	0	0	1	Timer Input Capture 2
0	1	0	1	0	Timer Input Capture 3
0	1	0	1	1	Timer Output Compare 1
0	1	1	0	0	Timer Output Compare 2
0	1	1	0	1	Timer Output Compare 3
0	1	1	1	0	Timer Output Compare 4
0	1	1	1	1	Timer Output Compare 5/Input Capture 4
1	0	0	0	0	Timer Overflow
1	0	0	0	1	Pulse Accumulator Overflow
1	0	0	1	0	Pulse Accumulator Input Edge
1	0	0	1	1	SPI Serial Transfer Complete
1	0	1	0	0	SCI Serial System
1	0	1	0	1	Reserved (Default to $\overline{IRQ}$)
1	0	1	1	0	Reserved (Default to $\overline{IRQ}$)
1	0	1	1	1	Reserved (Default to $\overline{IRQ}$)
1	1	X	X	X	Reserved (Default to $\overline{IRQ}$)

RAM and I/O Mapping

	Bit 7	6	5	4	3	2	1	Bit 0
$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG4	REG1	REG0
RESET:	0	0	0	0	0	0	0	0

RAM[3:0] — Internal RAM Map Position
REG[3:0] — 128-Byte Register Block Map Position

NOTE

Can be written only once in first 64 cycles out of reset in normal modes or at any time in special modes

INIT2

EEPROM Mapping

	Bit 7	6	5	4	3	2	1	Bit 0
$0037	EE3	EE2	EE1	EE0	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

EE[3:0] — EEPROM Map Position
EEPROM is at $xD80–$xFFF, where x is the hexadecimal digit represented by EE[3:0] bits.
Bits [3:0] — Not implemented; always read zero

OC1D

Output Compare 1 Data

	Bit 7	6	5	4	3	2	1	Bit 0
$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
RESET:	0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit x on successful OC1 compares.
Bits [2:0] — Not implemented; always read 0

OC1M

Output Compare 1 Mask

	Bit 7	6	5	4	3	2	1	Bit 0
$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
RESET:	0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding port A pin(s).
Bits [2:0] — Not implemented; always read 0

System Configuration Options 2

	Bit 7	6	5	4	3	2	1	Bit 0
$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	XDV1	XDV0
RESET:	0	0	0	—	0	0	0	0

LIRDV — Load Instruction Register Driven
- 0 = LIR not driven out of reset
- 1 = LIR driven high one quarter cycle

CWOM — Port C Wired-OR Mode
- 0 = Port C operates normally
- 1 = Port C outputs are open drain

STRCH — Stretch
- 0 = Normal operation
- 1 = Off-chip accesses to $0000–$7FFF extended by one E-clock cycle

IRVNE — Internal Read Visibility/Not E (IRVNE can be written once in any mode.)

In expanded modes, IRVNE determines whether IRV is on or off. In special test mode, IRVNE is reset to 1. In all other modes, IRVNE is reset to 0.
- 0 = No internal read visibility on external bus
- 1 = Data from internal reads is driven out of the external data bus.

In single-chip modes, this bit determines whether the E clock drives out from the chip.
- 0 = E is driven out from the chip.
- 1 = E pin is driven low.

Mode	IRVNE Out of Reset	E Clock Out of Reset	IRV Out of Reset	IRVNE Affects Only	IRVNE Can Be Written
Single Chip	0	On	Off	E	Once
Expanded	0	On	Off	IRV	Once
Boot	0	On	Off	E	Once
Special Test	1	On	On	IRV	Once

LSBF — SPI LSB First Enable
- 0 = SPI data transferred MSB first
- 1 = SPI data transferred LSB first

SPR2 — SPI Clock (SCK) Rate Select

Adds a divide by 4 to SPI clock chain.

XDV1–XDV0 — XOUT Clock Divide Select

These bits control the frequency of the clock driven out of the XOUT pin.

XDV [1:0]	XOUT = EXTAL Divided By	Frequency at EXTAL = 8 MHz	Frequency at EXTAL = 12 MHz	Frequency at EXTAL = 16 MHz
0 0	1	8 MHz	12 MHz	16 MHz
0 1	4	2 MHz	3 MHz	4 MHz
1 0	6	1.3 MHz	2 MHz	2.7 MHz
1 1	8	1 MHz	1.5 MHz	2 MHz

System Configuration Options

	Bit 7	6	5	4	3	2	1	Bit 0
$0039	ADPU	CSEL	IRQE*	DLY*	CME	FCME*	CR1*	CR0*
RESET:	0	0	0	1	0	0	0	0

*Can be written only once in first 64 cycles out of reset in normal mode, or at any time in special modes.

ADPU — A/D Power-Up
- 0 = A/D powered down
- 1 = A/D powered up

CSEL — Clock Select
- 0 = A/D and EEPROM use system E clock
- 1 = A/D and EEPROM use internal RC clock

IRQE — $\overline{\text{IRQ}}$ Select Edge-Sensitive Only
- 0 = Low level recognition
- 1 = Falling edge recognition

DLY — Enable Oscillator Start-Up Delay on Exit from STOP
- 0 = No stabilization delay on exit from STOP
- 1 = Stabilization delay enabled on exit from STOP

CME — Clock Monitor Enable
- 0 = Clock monitor disabled; slow clocks can be used
- 1 = Slow or stopped clocks cause clock failure reset

FCME — Force Clock Monitor Enable
- 0 = Clock monitor follows the state of the CME bit
- 1 = Clock monitor circuit is enabled until next reset

CR[1:0] — COP Timer Rate Select

CR [1:0]	Divide E/2^{15} By	XTAL = 8.0 MHz Timeout –0/+16.4 ms	XTAL = 12.0 MHz Timeout –0/+10.9 ms	XTAL = 16.0 MHz Timeout –0/+8.2 ms
0 0	1	16.384 ms	10.923 ms	8.192 ms
0 1	4	65.536 ms	43.691 ms	32.768 ms
1 0	16	262.14 ms	174.76 ms	131.07 ms
1 1	64	1.049 sec	699.05 ms	524.29 ms
	E =	2.0 MHz	3.0 MHz	4.0 MHz

PACNT

Pulse Accumulator Counter

	Bit 7	6	5	4	3	2	1	Bit 0
$0027	Bit 7	6	5	4	3	2	1	Bit 0

Readable and writable.

PACTL

Pulse Accumulator Control

	Bit 7	6	5	4	3	2	1	Bit 0
$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

Bits 7 and 3 — Not implemented; always read zero
PAEN — Pulse Accumulator System Enable
0 = Pulse Accumulator disabled
1 = Pulse Accumulator 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
I4/O5 — Input Capture 4/Output Compare 5
Configure TI4/O5 for input capture or output compare
0 = OC5 enabled
1 = IC4 enabled
RTR[1:0] — Real-Time Interrupt (RTI) Rate

RTR [1:0]	Divide E By	XTAL = 8.0 MHz	XTAL = 12.0 MHz	XTAL = 16.0 MHz
0 0	2^{13}	4.096 ms	2.731 ms	2.048 ms
0 1	2^{14}	8.192 ms	5.461 ms	4.096 ms
1 0	2^{15}	16.384 ms	10.923 ms	8.192 ms
1 1	2^{16}	32.768 ms	21.845 ms	16.383 ms
	E =	2.0 MHz	3.0 MHz	4.0 MHz

PORTA

Port A Data

	Bit 7	6	5	4	3	2	1	Bit 0
$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
RESET:	HiZ	0	0	0	HiZ	HiZ	HiZ	HiZ
Alt. Pin Func.:	PAI	OC2	OC3	OC4	OC5/IC4	IC1	IC2	IC3
And/or:	OC1	OC1	OC1	OC1	OC1	—	—	—

Port B Data

	Bit 7	6	5	4	3	2	1	Bit 0
$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
S. Chip or Boot:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET:	0	0	0	0	0	0	0	0
Expan. or Test:	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	ADDR 10	ADDR 9	ADDR 8

PORTC

Port C Data

	Bit 7	6	5	4	3	2	1	Bit 0
$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
S. Chip or Boot:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:	0	0	0	0	0	0	0	0
Expan. or Test:	DATA 7	DATA 6	DATA 5	DATA 4	DATA 3	DATA 2	DATA 1	DATA 0

PORTD

Port D Data

	Bit 7	6	5	4	3	2	1	Bit 0
$0008	0	0	PD5	PD4	PD3	PD2	PD1	PD0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Func.:	—	—	$\overline{SS}$	SCK	MOSI	MISO	TxD	RxD

PORTE

Port E Data

	Bit 7	6	5	4	3	2	1	Bit 0
$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:	U	U	U	U	U	U	U	U
Alt. Pin Func.:	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

Port F Data

	Bit 7	6	5	4	3	2	1	Bit 0
$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
S. Chip or Boot:	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Func.:	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

PORTG

Port G Data

	Bit 7	6	5	4	3	2	1	Bit 0
$007E	PG7	0	0	0	0	0	0	0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Func.:	R/$\overline{W}$	—	—	—	—	—	—	—

PG7 performs R/$\overline{W}$ in expanded and special test modes.

PORTH

Port H Data

	Bit 7	6	5	4	3	2	1	Bit 0
$007C	0	0	0	0	PH3	PH2	PH1	PH0
RESET:	U	U	U	U	U	U	U	U
Alt. Pin Func.:	—	—	—	—	PW4	PW3	PW2	PW1

PPAR

Port Pull-Up Assignment

	Bit 7	6	5	4	3	2	1	Bit 0
$002C	0	0	0	0	HPPUE	GPPUE	FPPUE	BPPUE
RESET:	0	0	0	0	1	1	1	1

Bits [7:4] — Not implemented; always read zero
xPPUE — Port x Pin Pull-Up Enable
0 = Port x pin on-chip pull-up devices disabled
1 = Port x pin on-chip pull-up devices enabled
FPPUE and BPPUE do not apply in expanded modes because ports F and B are address outputs.

EEPROM Programming Control

	Bit 7	6	5	4	3	2	1	Bit 0
$003B	ODD	EVEN	LVPI	BYTE	ROW	ERASE	EELAT	EEPGM
RESET:	0	0	0	0	0	0	0	0

ODD — Program Odd Rows in Half of EEPROM (TEST)
EVEN — Program Even Rows in Half of EEPROM (TEST)
LVPI — Low Voltage Programming Inhibit (Read Only). See also LVPEN bit in BPROT register.

LVPI reflects the condition of an on-chip voltage detector.

0 = EEPROM programming allowed
1 = EEPROM programming inhibited

BYTE — Byte/Other EEPROM Erase Mode

0 = Row or bulk erase mode used
1 = Erase only one byte of EEPROM

ROW — Row/All EEPROM Erase Mode (only valid when BYTE = 0)

0 = All 640 bytes of EEPROM erased
1 = Erase only one 16-byte row of EEPROM

BYTE	ROW	Action
0	0	Bulk Erase (All 640 Bytes)
0	1	Row Erase (16 Bytes)
1	0	Byte Erase
1	1	Byte Erase

ERASE — Erase/Normal Control for EEPROM

0 = Normal read or program mode
1 = Erase mode

EELAT — EEPROM Latch Control

0 = EEPROM address and data bus configured for normal reads
1 = EEPROM address and data bus configured for programming or erasing

EEPGM — EEPROM Program Command

0 = Program or erase voltage switched off to EEPROM array
1 = Program or erase voltage switched on to EEPROM array

Pulse-Width Modulation Timer Clock Select

	Bit 7	6	5	4	3	2	1	Bit 0
$0060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1
RESET:	0	0	0	0	0	0	0	0

CON34 — Concatenate Channels 3 and 4
Channel 3 is high-order byte, and channel 4 (port H, bit 3) is output.
0 = Channels 3 and 4 are separate 8-bit PWMs.
1 = Channels 3 and 4 are concatenated to create one 16-bit PWM channel.

CON12 — Concatenate Channels 1 and 2
Channel 1 is high-order byte, and channel 2 (port H, bit 1) is output.
0 = Channels 1 and 2 are separate 8-bit PWMs.
1 = Channels 1 and 2 are concatenated to create one 16-bit PWM channel.

PCKA[2:1] — Prescaler for Clock A (See also PWSCAL register)
Determines the rate of clock A

PCKA[2:1]	Value of Clock A
0 0	E
0 1	E/2
1 0	E/4
1 1	E/8

Bit 3 — Not implemented; always reads zero

PCKB[3:1] — Prescaler for Clock B
Determines the rate of clock B

PCKB[3:1]	Value of Clock B
0 0 0	E
0 0 1	E/2
0 1 0	E/4
0 1 1	E/8
1 0 0	E/16
1 0 1	E/32
1 1 0	E/64
1 1 1	E/128

Pulse-Width Modulation Timer Counter 1 to 4

	Bit 7	6	5	4	3	2	1	Bit 0	
$0064	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT1
$0065	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT2
$0066	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT3
$0067	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT4
RESET:	0	0	0	0	0	0	0	0	

PWCNT fields contain the counter modulus value for the associated PWM channel.

PWDTY1–4

Pulse-Width Modulation Timer Duty Cycle 1 to 4

	Bit 7	6	5	4	3	2	1	Bit 0	
$006C	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY1
$006D	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY2
$006E	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY3
$006F	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY4
RESET:	1	1	1	1	1	1	1	1	

PWDTY fields determine the duty cycle of the associated PWM channel.

PWEN

Pulse-Width Modulation Timer Enable

	Bit 7	6	5	4	3	2	1	Bit 0
$0063	TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1
RESET:	0	0	0	0	0	0	0	0

TPWSL — PWM Scaled Clock Test Bit (TEST)
DISCP — Disable Compare Scaled E Clock (TEST)
Bits [5:4] — Not implemented; always read zero
PWEN[4:1] — Pulse-Width Channel [4:1] Enable
0 = Channel disabled
1 = Channel enabled

Pulse-Width Modulation Timer Period 1 to 4

	Bit 7	6	5	4	3	2	1	Bit 0	
$0068	Bit 7	6	5	4	3	2	1	Bit 0	PWPER1
$0069	Bit 7	6	5	4	3	2	1	Bit 0	PWPER2
$006A	Bit 7	6	5	4	3	2	1	Bit 0	PWPER3
$006B	Bit 7	6	5	4	3	2	1	Bit 0	PWPER4
RESET:	1	1	1	1	1	1	1	1	

PWPER fields determine the period of the associated PWM channel.

PWPOL

Pulse-Width Modulation Timer Polarity

	Bit 7	6	5	4	3	2	1	Bit 0
$0061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1
RESET:	0	0	0	0	0	0	0	0

PCLK4 — Pulse-Width Channel 4 Clock Select
0 = Clock B is source
1 = Clock S is source
PCLK3 — Pulse-Width Channel 3 Clock Select
0 = Clock B is source
1 = Clock S is source
PCLK2 — Pulse-Width Channel 2 Clock Select
0 = Clock A is source
1 = Clock S is source
PCLK1 — Pulse-Width Channel 1 Clock Select
0 = Clock A is source
1 = Clock S is source
PPOL[4:1] — Pulse-Width Channel x Polarity
0 = PWM channel x output is low at the beginning of the clock cycle and goes high when duty count is reached
1 = PWM channel x output is high at the beginning of the clock cycle and goes low when duty count is reached

PWSCAL

Pulse-Width Modulation Timer Prescaler

	Bit 7	6	5	4	3	2	1	Bit 0
$0062	7	6	5	4	3	2	1	0
RESET:	0	0	0	0	0	0	0	0

Scaled clock S is generated by dividing clock A by the value in PWSCAL, then dividing the result by 2. If PWSCAL = $00, divide clock A by 256, then divide the result by 2.

SCI Baud Rate Control High, Low

	Bit 7	6	5	4	3	2	1	Bit 0	
$0070	BTST	BSPL	0	SBR12	SBR11	SBR10	SBR9	SBR8	High
$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	Low
RESET:	0	0	0	0	0	0	0	0	

BTST — Baud Register Test (TEST)
BSPL — Baud Rate Counter Split (TEST)
Bit 5 — Not implemented; always reads zero
SBR[12:0] — SCI Baud Rate Selects

To calculate special rates, use the following formula. Example rates are located in the table of baud rate control values. SCI baud rate = CLKX ÷ [16 • (2 • BR)] where BR is the contents of SCBDH, L (BR = 1, 2, 3,..., 8191). BR = 0 disables the baud rate generator.

Target Baud Rate	Crystal Frequency					
	8 MHz		12 MHz		16 MHz	
	Dec Value	Hex Value	Dec Value	Hex Value	Dec Value	Hex Value
110	2272	$08E0	3409	$0D51	4545	$11C1
150	1666	$0682	2500	$09C4	3333	$0D05
300	833	$0341	1250	$04E2	1666	$0682
600	416	$01A0	625	$0271	833	$0341
1200	208	$00D0	312	$0138	416	$01A0
2400	104	$0068	156	$009C	208	$00D0
4800	52	$0034	78	$004E	104	$0068
9600	26	$001A	39	$0027	52	$0034
19.2 K	13	$000D	20	$0014	26	$001A
38.4 K	—	—	—	—	13	$000D

SCI Control 1

	Bit 7	6	5	4	3	2	1	Bit 0
$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT
RESET:	0	0	0	0	0	0	0	0

LOOPS — SCI LOOP Mode Enable
- 0 = SCI transmit and receive operate normally.
- 1 = SCI transmit and receive are disconnected from TxD and RxD pins, and transmitter output is fed back into the receiver input.

WOMS — Wired-Or Mode for SCI Pins (PD1, PD0; see also DWOM bit in SPCR.)
- 0 = TxD and RxD operate normally.
- 1 = TxD and RxD are open drains if operating as an output.

Bit 5 — Not implemented; always reads 0

M — Mode (Select Character Format)
- 0 = Start bit, 8 data bits, 1 stop bit
- 1 = Start bit, 9 data bits, 1 stop bit

WAKE — Wakeup by Address Mark/Idle
- 0 = Wakeup by IDLE line recognition
- 1 = Wakeup by address mark (most significant data bit set)

ILT — Idle Line Type
- 0 = Short (SCI counts consecutive ones after start bit.)
- 1 = Long (SCI counts ones only after stop bit.)

PE — Parity Enable
- 0 = Parity disabled
- 1 = Parity enabled

PT — Parity Type
- 0 = Parity even (even number of ones causes parity bit to be zero, odd number of ones causes parity bit to be one.)
- 1 = Parity odd (odd number of ones causes parity bit to be zero, even number of ones causes parity bit to be one.)

SCI Control 2

	Bit 7	6	5	4	3	2	1	Bit 0
$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
- 0 = TDRE interrupts disabled
- 1 = SCI interrupt requested when TDRE status flag is set

TCIE — Transmit Complete Interrupt Enable
- 0 = TC interrupts disabled
- 1 = SCI interrupt requested when TC status flag is set

RIE — Receiver Interrupt Enable
- 0 = RDRF and OR interrupts disabled
- 1 = SCI interrupt requested when RDRF flag or the OR status flag is set

ILIE — Idle Line Interrupt Enable
- 0 = IDLE interrupts disabled
- 1 = SCI interrupt requested when IDLE status flag is set

TE — Transmitter Enable
- 0 = Transmitter disabled
- 1 = Transmitter enabled

RE — Receiver Enable
- 0 = Receiver disabled
- 1 = Receiver enabled

RWU — Receiver Wakeup Control
- 0 = Normal SCI receiver
- 1 = Wakeup enabled and receiver interrupts inhibited

SBK — Send Break
- 0 = Break generator off
- 1 = Break codes generated as long as SBK = 1

SCDRH, SCDRL

SCI Data High, Low

	Bit 7	6	5	4	3	2	1	Bit 0	
$0076	R8	T8	0	0	0	0	0	0	SCDRH (Hi)
$0077	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDRL (Lo)

R8 — Receiver Bit 8
Ninth serial data bit received when SCI is configured for nine data bit operation.

T8 — Transmitter Bit 8
Ninth serial data bit transmitted when SCI is configured for nine data bit operation.

Bits [5:0] — Not implemented; always read zero

R[7:0]/T[7:0] — Receiver/Transmitter Data Bits [7:0]
SCI data is double buffered in both directions.

SCI Status Register 1

	Bit 7	6	5	4	3	2	1	Bit 0
$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
RESET:	1	1	0	0	0	0	0	0

TDRE — Transmit Data Register Empty Flag
This flag is set when SCDR is empty. Clear the TDRE flag by reading SCSR1 with TDRE set and then writing to SCDR.
0 = SCDR busy
1 = SCDR empty

TC — Transmit Complete Flag
This flag is set when the transmitter is idle (no data, preamble, or break transmission in progress). Clear the TC flag by reading SCSR1 with TC set and then writing to SCDR.
0 = Transmitter busy
1 = Transmitter idle

RDRF — Receive Data Register Full Flag
This flag is set if a received character is ready to be read from SCDR. Clear the RDRF flag by reading SCSR1 with RDRF set and then reading SCDR.
0 = SCDR empty
1 = SCDR full

IDLE — Idle Line Detected Flag
This flag is set if the RxD line is idle. Once cleared, IDLE is not set again until the RxD line has been active and becomes idle again. The IDLE flag is inhibited when RWU = 1. Clear IDLE by reading SCSR1 with IDLE set and then reading SCDR.
0 = RxD line is active
1 = RxD line is idle

OR — Overrun Error Flag
OR is set if a new character is received before a previously received character is read from SCDR. Clear the OR flag by reading SCSR1 with OR set and then reading SCDR.
0 = No overrun
1 = Overrun detected

NF — Noise Error Flag
NF is set if majority sample logic detects anything other than a unanimous decision. Clear NF by reading SCSR1 with NF set and then reading SCDR.
0 = Unanimous decision
1 = Noise detected

FE — Framing Error
FE is set when a zero is detected where a stop bit was expected. Clear the FE flag by reading SCSR1 with FE set and then reading SCDR.
0 = Stop bit detected
1 = Zero detected

PF — Parity Error Flag
PF is set if received data has incorrect parity. This bit is cleared by a read of SCSR1 with PE set and then reading SCDR.
0 = Parity correct
1 = Received data has incorrect parity

SCI Status Register 2

	Bit 7	6	5	4	3	2	1	Bit 0
$0075	0	0	0	0	0	0	0	RAF
RESET:	0	0	0	0	0	0	0	0

Bits [7:1] — Not implemented; always read zero
RAF — Receiver Active Flag (Read Only)
0 = A character is not being received.
1 = A character is being received.

SPCR

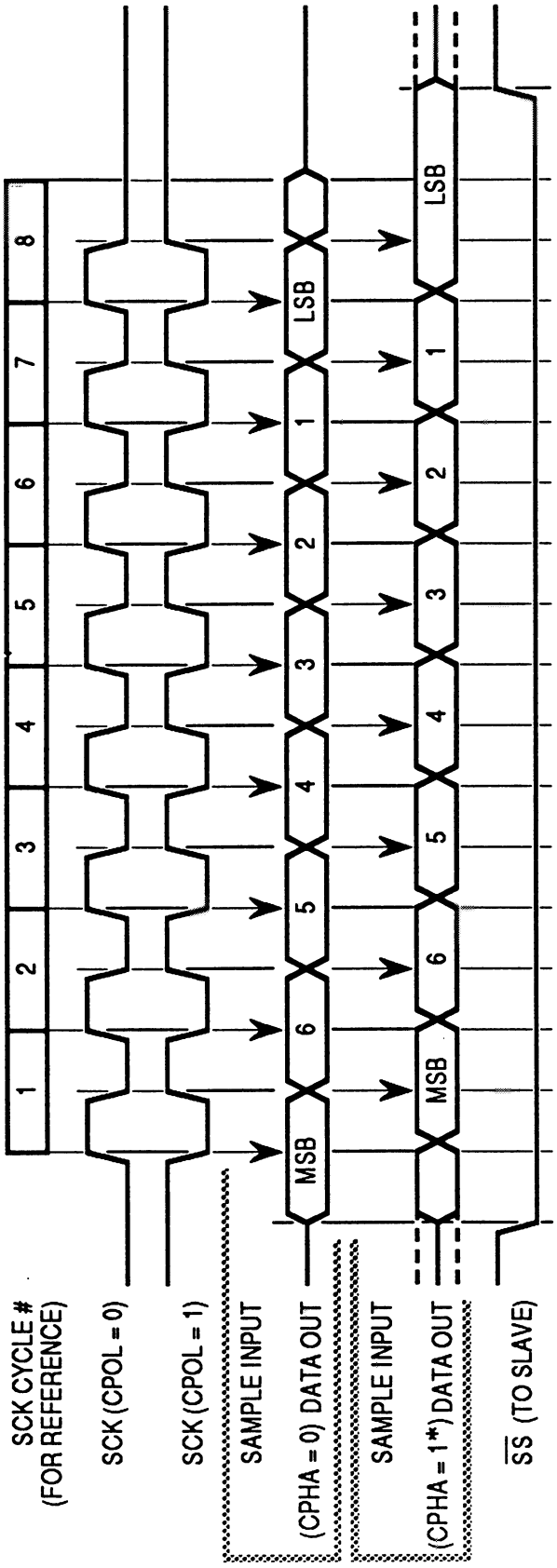
Serial Peripheral Control

	Bit 7	6	5	4	3	2	1	Bit 0
$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET:	0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable
0 = SPI interrupts disabled
1 = SPI interrupts enabled
SPE — Serial Peripheral System Enable
0 = SPI off
1 = SPI on
DWOM — Port D Wired-OR Mode Option for SPI Pins PD5–PD2
(See also WOMS bit in SCCR2.)
0 = Normal CMOS outputs
1 = Open-drain outputs
MSTR — Master Mode Select
0 = Slave mode
1 = Master mode
CPOL, CPHA — Clock Polarity, Clock Phase
(Refer to SPI Transfer Format.)
SPR[2:0] — SPI Clock Rate Selects (SPR2 is located in OPT2 register)

SPR[2:0]	Divide E Clock By	Frequency at E = 2 MHz (Baud)
0 0 0	2	1.0 MHz
0 0 1	4	500 kHz
0 1 0	16	125 kHz
0 1 1	32	62.5 kHz
1 0 0	8	250 kHz
1 0 1	16	125 kHz
1 1 0	64	31.3 kHz
1 1 1	128	15.6 kHz

SPI Transfer Format



SPI Data

	Bit 7	6	5	4	3	2	1	Bit 0
$002A	Bit 7	6	5	4	3	2	1	Bit 0

SPI is double buffered in, single buffered out.

SPSR

Serial Peripheral Status

	Bit 7	6	5	4	3	2	1	Bit 0
$0029	SPIF	WCOL	0	MODF	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

SPIF — SPI Transfer Complete Flag
SPIF is set when an SPI transfer is complete. This bit is cleared by reading SPSR with SPIF set, followed by an SPDR access.
0 = Incomplete SPI transfer
1 = SPI transfer complete

WCOL — Write Collision
WCOL is set when SPDR is written while a transfer is in progress. It is cleared by reading SPSR with WCOL set, followed by an SPDR access.
0 = No write collision
1 = Write collision detected

Bits 5 and [3:0] — Not implemented; always read zero

MODF — Mode Fault (Mode fault terminates SPI operation)
MODF is set when $\overline{SS}$ is pulled low while MSTR = 1. This bit is cleared by reading SPCR with MODF set, followed by a write to SPCR.
0 = No mode fault detected
1 = Mode fault detected

TCNT

Timer Count

$000E	Bit 15	14	13	12	11	10	9	Bit 8	High	**TCNT**
$000F	Bit 7	6	5	4	3	2	1	Bit 0	Low	

TCNT resets to $0000.
In normal modes, TCNT is read-only.

TCTL1

Timer Control 1

	Bit 7	6	5	4	3	2	1	Bit 0
$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET:	0	0	0	0	0	0	0	0

OM2–OM5 — Output Mode
OL2–OL5 — Output Level

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

TCTL2

Timer Control 2

	Bit 7	6	5	4	3	2	1	Bit 0
$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET:	0	0	0	0	0	0	0	0

Timer Control Configuration

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

TEST1

Factory Test

	Bit 7	6	5	4	3	2	1	Bit 0
$003E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	0
RESET:	0	0	0	0	—	0	0	0

TILOP —Test Illegal Opcode (Test modes only)
Bits 6 and 0 — Not implemented; always read zero
OCCR — Output Condition Code Register to Timer Port (Test modes only)
CBYP — Timer Divider Chain Bypass (Test modes only)
DISR — Disable Resets from COP and Clock Monitor (Test modes only)
FCM — Force Clock Monitor Failure (Test modes only)
FCOP — Force COP Watchdog Failure (Test modes only)

Timer Interrupt Flag 1

	Bit 7	6	5	4	3	2	1	Bit 0
$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
RESET:	0	0	0	0	0	0	0	0

Clear flags by writing a one to the corresponding bit position(s).

OC1F–OC4F — Output Compare x Flag
Set each time the counter matches output compare x value.

I4/O5F — Input Capture 4/Output Compare 5 Flag
Set by IC4 or OC5, depending on which function was enabled by I4/O5 of PACTL.

IC1F–IC3F — Input Capture x Flag
Set each time a selected active edge is detected on the ICx input line.

TFLG2

Timer Interrupt Flag 2

	Bit 7	6	5	4	3	2	1	Bit 0
$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

Clear flags by writing a one to the corresponding bit position(s).

TOF — Timer Overflow Flag
Set when TCNT changes from $FFFF to $0000

RTIF — Real-Time (Periodic) Interrupt Flag
Set periodically. Refer to RTR[1:0] in PACTL register.

PAOVF — Pulse Accumulator Overflow Flag
Set when PACNT changes from $FF to $00.

PAIF — Pulse Accumulator Input Edge Flag
Set each time a selected active edge is detected on the PAI input line.

Bits [3:0] — Not implemented; always read zero

TI4/O5

Timer Input Capture 4/Output Compare 5

$001E	Bit 15	14	13	12	11	10	9	Bit 8	High	TI4/O5
$001F	Bit 7	6	5	4	3	2	1	Bit 0	Low	

All TI4/O5 register pairs reset to ones ($FFFF).

Timer Input Capture

$0010	Bit 15	14	13	12	11	10	9	Bit 8	High	TIC1
$0011	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$0012	Bit 15	14	13	12	11	10	9	Bit 8	High	TIC2
$0013	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$0014	Bit 15	14	13	12	11	10	9	Bit 8	High	TIC3
$0015	Bit 7	6	5	4	3	2	1	Bit 0	Low	

TICx not affected by reset

TMSK1

Timer Interrupt Mask 1

	Bit 7	6	5	4	3	2	1	Bit 0
$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I
RESET:	0	0	0	0	0	0	0	0

OC1I–OC4I — Output Compare x Interrupt Enable
I4/O5I — Input Capture 4 or Output Compare 5 Interrupt Enable
IC1I–IC3I — Input Capture x Interrupt Enable

TMSK2

Timer Interrupt Mask 2

	Bit 7	6	5	4	3	2	1	Bit 0
$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
RESET:	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
0 = TOF interrupts enabled
1 = Interrupt requested when TOF is set to one
RTII — Real-Time Interrupt Enable
0 = RTIF interrupts disabled
1 = Interrupt requested when RTIF is set to one.
PAOVI — Pulse Accumulator Overflow Interrupt Enable
0 = Interrupt inhibited
1 = Interrupt requested if PAOVF is set
PAII — Pulse Accumulator Input Edge Interrupt Enable
0 = Interrupt inhibited
1 = Interrupt requested if PAIF is set
Bits [3:2] — Not implemented; always read zero

PR[1:0] — Timer Prescaler Select

In normal modes, PR1 and PR0 can only be written once, and the write must occur within 64 cycles after reset.

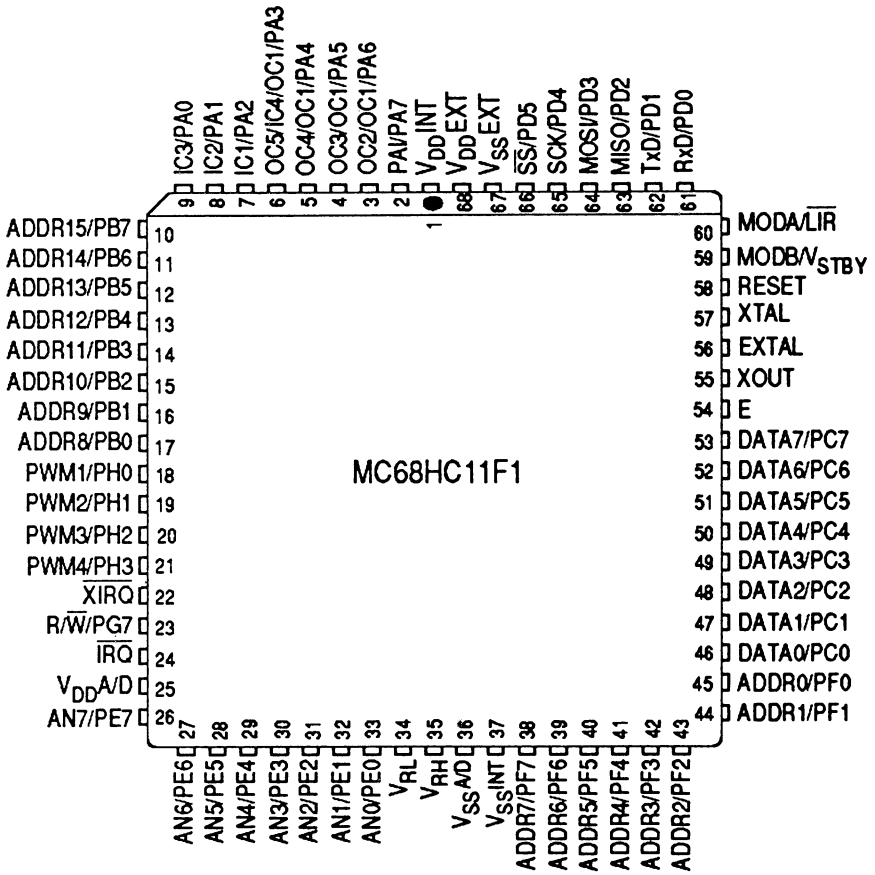
PR[1:0]	Prescaler
0 0	1
0 1	4
1 0	8
1 1	16

TOC1–TOC4

Timer Output Compare

$0016	Bit 15	14	13	12	11	10	9	Bit 8	High	TOC1
$0017	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$0018	Bit 15	14	13	12	11	10	9	Bit 8	High	TOC2
$0019	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$001A	Bit 15	14	13	12	11	10	9	Bit 8	High	TOC3
$001B	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$001C	Bit 15	14	13	12	11	10	9	Bit 8	High	TOC4
$001D	Bit 7	6	5	4	3	2	1	Bit 0	Low	

All TOCx register pairs reset to ones ($FFFF).



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	2	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 CHARACTER SET (7-Bit Code)								
MS Digit / LS Digit	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

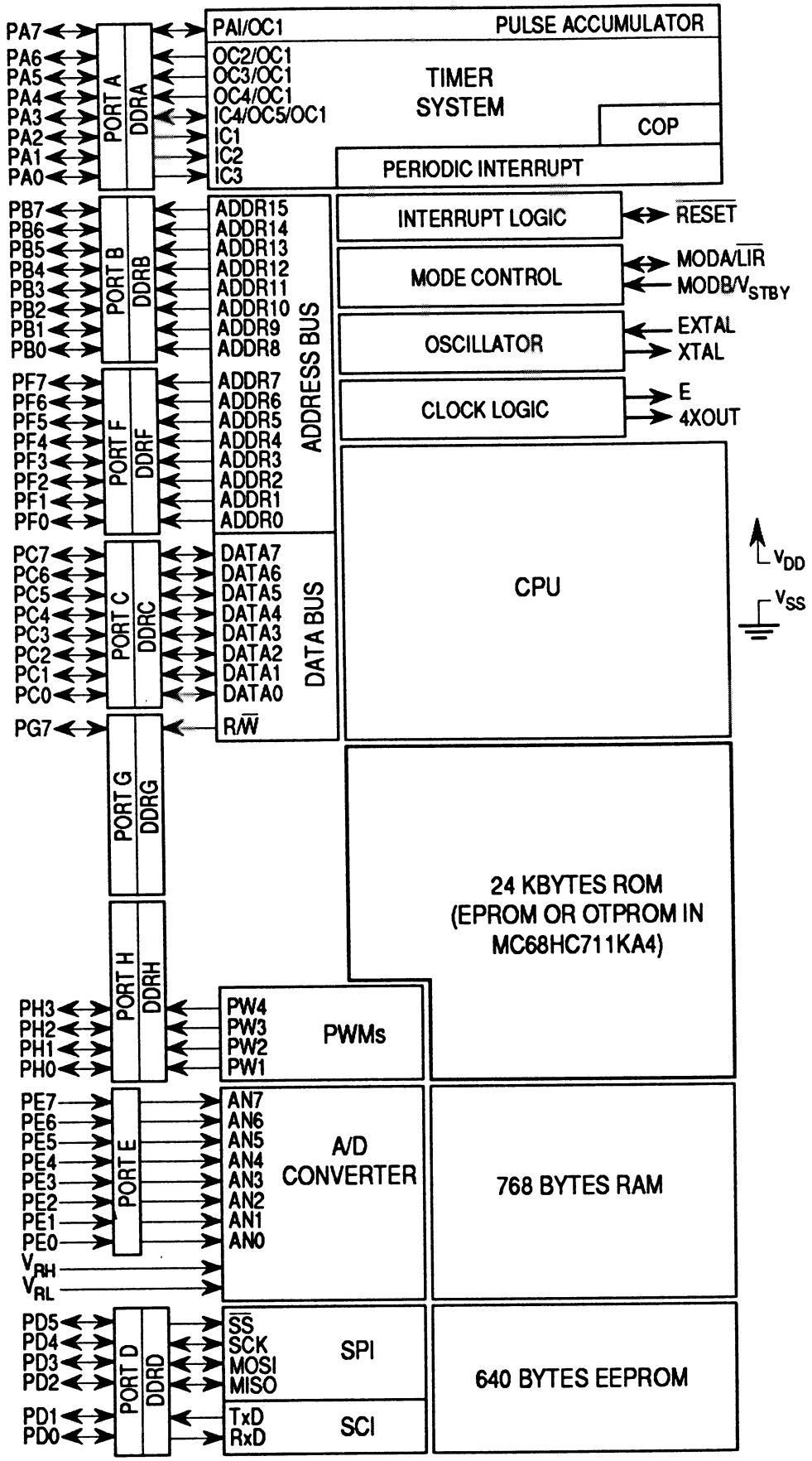
PROGRAMMING MODEL
CRYSTAL DEPENDENT TIMING
INTERRUPTS

MEMORY MAP
OPCODE MAPS

INSTRUCTIONS
ADDRESSING MODES
EXECUTION TIMES
SPECIAL OPERATIONS

REGISTER AND
CONTROL BIT
ASSIGNMENTS

MECHANICAL DATA
HEX/DEC CONVERSION
ASCII CHART



1ATX31058-0 PRINTED IN USA 4/92 IMPERIAL LITHO 85012 18,000 MCU YGACAA

Literature Distribution Centers:

USA: Motorola Literature Distribution
P.O. Box 20912; Phoenix, Arizona 85036.

EUROPE: Motorola Ltd.; European Literature Centre
88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.

JAPAN: Nippon Motorola Ltd.
4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.

ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.
Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong.

MOTOROLA

MC68HC11KA4RG/AD