68HC11 Instruction Set

Each addressing mode shows the opcode, #cyles and # bytes. – means the addressing mode is not valid for that instruction.

instruct.	Imm / rel	direct	ind,x	ind,y	extend	inherent	explanation	h	i	n	z	v	c
aba	-- - -	-- - -	-- - -	-- - -	-- - -	1b 2 1	A=(A)+(B)	x	-	x	x	x	x
abx	-- - -	-- - -	-- - -	-- - -	-- - -	3a 3 1	IX=(IX)+(B)	-	-	-	-	-	-
aby	-- - -	-- - -	-- - -	-- - -	-- - -	183a 4 1	IY=(IY)+(B)	-	-	-	-	-	-
adca	89 2 2	99 3 2	a9 4 2	18a9 5 3	b9 4 3	-- - -	A=(A)+(M)+(C)	x	-	x	x	x	x
adcb	c9 2 2	d9 3 2	e9 4 2	18e9 5 3	f9 4 3	-- - -	B=(B)+(M)+(C)	x	-	x	x	x	x
adda	8b 2 2	9b 3 2	ab 4 2	18ab 4 2	bb 4 3	-- - -	A=(A)+(M)	x	-	x	x	x	x
addb	cb 2 2	db 3 2	eb 4 2	18eb 5 3	fb 4 3	-- - -	B=(B)+(M)	x	-	x	x	x	x
addd	c3 3 4	d3 5 2	e3 6 2	18e3 7 3	f3 6 3	-- - -	D=(D)+(M;M+1)	-	-	x	x	x	x
anda	84 2 2	94 3 2	a4 4 2	18a4 5 3	b4 4 3	-- - -	A=(A)&(M)	-	-	x	x	0	-
andb	c4 2 2	d4 3 2	e4 4 2	18e4 5 3	f4 4 3	-- - -	B=(B)&(M)	-	-	x	x	0	-
asl	-- - -	-- - -	68 6 2	1868 7 3	78 6 3	-- - -	see notes	-	-	x	x	x	x
asla	-- - -	-- - -	-- - -	-- - -	-- - -	48 2 1	see notes	-	-	x	x	x	x
aslb	-- - -	-- - -	-- - -	-- - -	-- - -	58 2 1	see notes	-	-	x	x	x	x
asld	-- - -	-- - -	-- - -	-- - -	-- - -	05 3 1	see notes	-	-	x	x	x	x
asr	-- - -	-- - -	67 6 2	1867 7 3	77 6 3	-- - -	see notes	-	-	x	x	x	x
asra	-- - -	-- - -	-- - -	-- - -	-- - -	47 2 1	see notes	-	-	x	x	x	x
asrb	-- - -	-- - -	-- - -	-- - -	-- - -	57 2 1	see notes	-	-	x	x	x	x
bcc	24 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if C==0						
bclr	-- - -	15 6 3	1d 7 3	181d 8 4	-- - -	-- - -	bit clear	-	-	x	x	0	-
bcs	25 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if C==1						
beq	27 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if Z==1						
bge	2c 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if greater or equal						
bgt	2e 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if greater than						
bhi	22 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if higher						
bhs	24 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if high or same						
bita	85 2 2	95 3 2	a5 4 2	18a5 5 3	b5 4 3	-- - -	bit test A	-	-	x	x	0	-
bitb	c5 2 2	d5 3 2	e5 4 2	18e5 5 3	f5 4 3	-- - -	bit test B	-	-	x	x	0	-
ble	2f 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if less than or equal						
blo	25 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if lower						
bls	23 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if lower or same						
blt	2d 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if less than						
bmi	2b 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if minus						
bne	26 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if not equal						
bpl	2a 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if plus						
bra	20 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra always						
brclr	-- - -	13 6 4	1f 7 4	181f 8 5	-- - -	-- - -	bra bits clr	-	-	x	x	0	-
brset	-- - -	12 6 4	1e 7 4	181e 8 5	-- - -	-- - -	bra bits set	-	-	x	x	0	-
bset	-- - -	14 6 3	1c 7 3	181c 8 4	-- - -	-- - -	bit set	-	-	x	x	0	-
bsr	8d 6 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra to subroutine						
bvc	28 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if V==0						
bvs	29 3 2	-- - -	-- - -	-- - -	-- - -	-- - -	bra if V==1						
cba	-- - -	-- - -	-- - -	-- - -	-- - -	11 2 1	(A) – (B)	-	-	x	x	x	x
clc	-- - -	-- - -	-- - -	-- - -	-- - -	0c 2 1	C=0	-	-	-	-	-	0
cli	-- - -	-- - -	-- - -	-- - -	-- - -	0e 2 1	I=0	-	0	-	-	-	-
clr	-- - -	-- - -	6f 6 2	186f 7 3	7f 6 3	-- - -	M=0	-	-	x	x	0	0
clra	-- - -	-- - -	-- - -	-- - -	-- - -	4f 2 1	A=0	-	-	x	x	0	0
clrb	-- - -	-- - -	-- - -	-- - -	-- - -	5f 2 1	B=0	-	-	x	x	0	0
clv	-- - -	-- - -	-- - -	-- - -	-- - -	0a 2 1	V=0	-	-	-	-	0	-
cmpa	81 2 2	91 3 2	a1 4 2	18a1 5 3	b1 4 3	-- - -	(A)-(M)	-	-	x	x	x	x
cmpb	c1 2 2	d1 3 2	e1 4 2	18e1 5 3	f1 4 3	-- - -	(B)-(M)	-	-	x	x	x	x
com	-- - -	-- - -	63 6 2	1863 7 3	73 6 3	-- - -	M=$FF-(M)	-	-	x	x	0	1
coma	-- - -	-- - -	-- - -	-- - -	-- - -	43 2 1	A=$FF-(A)	-	-	x	x	0	1
comb	-- - -	-- - -	-- - -	-- - -	-- - -	53 2 1	B=$FF-(B)	-	-	x	x	0	1
cpd	1a83 5 4	1a93 6 3	1ab3 7 4	cdb3 7 4	1aa3 7 3	-- - -	(D)-(M:M+!)	-	-	x	x	x	x
cpx	8c 4 3	9c 5 2	ac 6 2	18ac 7 3	bc 6 3	-- - -	(IX)-(M:M+!)	-	-	x	x	x	x
cpy	188c 5 4	189c 6 3	18ac 7 3	1aac 7 3	18bc 7 4	-- - -	(IY)-(M:M+!)	-	-	x	x	x	x
daa	-- - -	-- - -	-- - -	-- - -	-- - -	19 2 1	BCD adjust A	-	-	x	x	x	3
dec	-- - -	-- - -	6a 6 2	186a 7 3	7a 6 3	-- - -	M=(M)-1	-	-	x	x	x	-
deca	-- - -	-- - -	-- - -	-- - -	-- - -	4a 2 1	A=(A)-1	-	-	x	x	x	-
decb	-- - -	-- - -	-- - -	-- - -	-- - -	5a 2 1	B=(B)-1	-	-	x	x	x	-
des	-- - -	-- - -	-- - -	-- - -	-- - -	34 3 1	SP=(SP)-1	-	-	-	-	-	-

instruct.	immed	direct	ind,x	ind,y	extend	inherent	explanation	h	i	n	z	v	c
dex	-- - -	-- - -	-- - -	-- - -	-- - -	09 3 1	IX=(IX)-1	-	-	-	x	-	-
dey	-- - -	-- - -	-- - -	-- - -	-- - -	1809 4 2	IY=(IY)-1	-	-	-	x	-	-
eora	88 2 2	98 3 2	a8 4 2	18a8 5 3	b8 4 3	-- - -	A=(A)⊕(M)	-	-	x	x	0	-
eorb	c8 2 2	d8 3 2	e8 4 2	18e8 5 3	f8 4 3	-- - -	B=(B)⊕(M)	-	-	x	x	0	–
fdiv	-- - -	-- - -	-- - -	-- - -	-- - -	0341 1	see notes	-	-	-	x	x	x
idiv	-- - -	-- - -	-- - -	-- - -	-- - -	0241 1	see notes	-	-	-	x	0	x
inc	-- - -	-- - -	6c 6 2	186c 7 3	7c 6 3	-- - -	M=(M)+1	-	-	x	x	x	-
inca	-- - -	-- - -	-- - -	-- - -	-- - -	4c 2 1	A=(A)+1	-	-	x	x	x	-
incb	-- - -	-- - -	-- - -	-- - -	-- - -	5c 2 1	B=(B)+1	-	-	x	x	x	-
ins	-- - -	-- - -	-- - -	-- - -	-- - -	31 3 1	SP=(SP)+1	-	-	-	-	-	-
inx	-- - -	-- - -	-- - -	-- - -	-- - -	08 3 1	IX=(IX)+1	-	-	-	x	-	-
iny	-- - -	-- - -	-- - -	-- - -	-- - -	1808 4 2	IY=(IY)+1	-	-	-	x	-	-
jmp	-- - -	-- - -	6e 3 2	186e 4 3	7e 3 3	-- - -	jump	-	-	-	-	-	-
jsr	-- - -	9d 5 2	ad 6 2	18ad 7 3	bd 6 3	-- - -	jump to sub.	-	-	-	-	-	-
ldaa	86 2 2	96 3 2	a6 4 2	18a6 5 3	b6 4 3	-- - -	A=(M)	-	-	x	x	0	-
ldab	c6 2 2	d6 3 2	e6 4 2	18e6 5 3	f6 4 3	-- - -	B=(M)	-	-	x	x	0	-
ldd	cc 3 3	dc 4 2	ec 5 2	18ec 6 3	fc 5 3	-- - -	D=(M:M+1)	-	-	x	x	0	-
lds	8e 3 3	9e 4 2	ae 5 2	18ae 6 3	be 5 3	-- - -	SP=(M:M+1)	-	-	x	x	0	-
ldx	ce 3 3	de 4 2	ee 5 2	cdee 5 2	fe 5 3	-- - -	IX=(M:M+1))	-	-	x	x	0	-
ldy	18ce 4 4	18de 5 3	1aee 6 3	18ee 6 3	18fe 6 4	-- - -	IY=(M:M+1))	-	-	x	x	0	-
lsl	-- - -	-- - -	68 6 2	1868 7 3	78 6 3	-- - -	see notes	-	-	x	x	x	x
lsla	-- - -	-- - -	-- - -	-- - -	-- - -	48 2 1	see notes	-	-	x	x	x	x
lslb	-- - -	-- - -	-- - -	-- - -	-- - -	58 2 1	see notes	-	-	x	x	x	x
lsld	-- - -	-- - -	-- - -	-- - -	-- - -	04 3 1	see notes	-	-	x	x	x	x
lsr	-- - -	-- - -	64 6 2	1864 7 3	74 6 3	-- - -	see notes	-	-	x	x	x	x
lsra	-- - -	-- - -	-- - -	-- - -	-- - -	44 2 1	see notes	-	-	x	x	x	x
lsrb	-- - -	-- - -	-- - -	-- - -	-- - -	54 2 1	see notes	-	-	x	x	x	x
lsrd	-- - -	-- - -	-- - -	-- - -	-- - -	04 3 1	see notes	-	-	x	x	x	x
mul	-- - -	-- - -	-- - -	-- - -	-- - -	3d 7 1	see notes	-	-	-	-	-	4
neg	-- - -	-- - -	60 6 2	1860 7 3	70 6 3	-- - -	M=$00-(M)	-	-	x	x	x	x
nega	-- - -	-- - -	-- - -	-- - -	-- - -	40 2 1	A=$00-(A)	-	-	x	x	x	x
negb	-- - -	-- - -	-- - -	-- - -	-- - -	50 2 1	b=$00-(B)	-	-	x	x	x	x
nop	-- - -	-- - -	-- - -	-- - -	-- - -	01 2 1	no operation	-	-	-	-	-	-
oraa	8a 2 2	9a 3 2	aa 4 2	18aa 5 3	ba 4 3	-- - -	A=(A)|(M)	-	-	x	x	0	-
orab	ca 2 2	da 3 2	ea 4 2	18ea 5 3	fa 4 3	-- - -	B=(B)|(M)	-	-	x	x	0	-
psha	-- - -	-- - -	-- - -	-- - -	-- - -	36 3 1	push A;decr SP	-	-	-	-	-	-
pshb	-- - -	-- - -	-- - -	-- - -	-- - -	37 3 1	push B;decr SP	-	-	-	-	-	-
pshx	-- - -	-- - -	-- - -	-- - -	-- - -	3c 4 1	psh IX;decr SP	-	-	-	-	-	-
pshy	-- - -	-- - -	-- - -	-- - -	-- - -	183c 5 2	psh IY;decr SP	-	-	-	-	-	-
pula	-- - -	-- - -	-- - -	-- - -	-- - -	32 3 1	incr SP;pull A	-	-	-	-	-	-
pulb	-- - -	-- - -	-- - -	-- - -	-- - -	33 3 1	incr SP;pull B	-	-	-	-	-	-
pulx	-- - -	-- - -	-- - -	-- - -	-- - -	38 4 1	incr SP;pul IX	-	-	-	-	-	-
puly	-- - -	-- - -	-- - -	-- - -	-- - -	1838 5 2	incr SP;pul IY	-	-	-	-	-	-
rol	-- - -	-- - -	69 6 2	1869 7 3	79 6 3	-- - -	see notes	-	-	x	x	x	x
rola	-- - -	-- - -	-- - -	-- - -	-- - -	49 2 1	see notes	-	-	x	x	x	x
rolb	-- - -	-- - -	-- - -	-- - -	-- - -	59 2 1	see notes	-	-	x	x	x	x
ror	-- - -	-- - -	66 6 2	1866 7 2	76 6 3	-- - -	see notes	-	-	x	x	x	x
rora	-- - -	-- - -	-- - -	-- - -	-- - -	46 2 1	see notes	-	-	x	x	x	x
rorb	-- - -	-- - -	-- - -	-- - -	-- - -	56 2 1	see notes	-	-	x	x	x	x
rti	-- - -	-- - -	-- - -	-- - -	-- - -	3b12 1	rtn from int.	x	x	x	x	x	x
rts	-- - -	-- - -	-- - -	-- - -	-- - -	39 5 1	rtn from sub	-	-	-	-	-	-
sba	-- - -	-- - -	-- - -	-- - -	-- - -	10 2 1	A=(A)-(B)	-	-	x	x	x	x
sbca	82 2 2	92 3 2	a2 4 2	18a2 5 3	b2 4 3	-- - -	A=(A)-(M)-(C)	-	-	x	x	x	x
sbcb	c2 2 2	d2 3 2	e2 4 2	18e2 5 3	f2 4 3	-- - -	B=(B)-(M)-(C)	-	-	x	x	x	x
sec	-- - -	-- - -	-- - -	-- - -	-- - -	0d 2 1	C=1	-	-	-	-	-	1
sei	-- - -	-- - -	-- - -	-- - -	-- - -	0f 2 1	I=1	-	1	-	-	-	-
sev	-- - -	-- - -	-- - -	-- - -	-- - -	0b 2 1	V=1	-	-	-	-	1	-
staa	-- - -	97 3 2	a7 4 2	18a7 4 2	b7 4 3	-- - -	M=(A)	-	-	x	x	0	-
stab	-- - -	d7 3 2	e7 4 2	18e7 4 2	f7 4 3	-- - -	M=(B)	-	-	x	x	0	-
std	-- - -	dd 4 2	ed 5 2	18ed 6 3	fd 5 3	-- - -	(M:M+1)=D	-	-	x	x	0	-
sts	-- - -	9f 4 2	af 5 2	18af 6 3	bf 5 3	-- - -	(M:M+1)=S	-	-	x	x	0	-
stx	-- - -	df 4 2	ef 5 2	18ef 6 3	ff 5 3	-- - -	(M:M+1)=IX	-	-	x	x	0	-
sty	-- - -	18df 5 3	1aef 6 3	18ef 6 3	18ff 6 4	-- - -	(M:M+1)=IY	-	-	x	x	0	-
suba	80 2 2	90 3 2	a0 4 2	18a0 5 3	b0 4 3	-- - -	A=(A)-(M)	-	-	x	x	x	x
subb	c0 2 2	d0 3 2	e0 4 2	18e0 5 3	f0 4 3	-- - -	B=(B)-(M)	-	-	x	x	x	x
subd	83 4 3	93 5 2	a3 6 2	18a3 7 3	b3 6 3	-- - -	D=(D)-(M:M+1)	-	-	x	x	x	x
swi	-- - -	-- - -	-- - -	-- - -	-- - -	3f14 1	software int.	-	1	-	-	-	-

instruct.	immed	direct	ind,x	ind,y	extend	inherent	explanation	h	i	n	z	v	c
tab	-- - -	-- - -	-- - -	-- - -	-- - -	16 2 1	B=(A)	-	-	x	x	0	-
tap	-- - -	-- - -	-- - -	-- - -	-- - -	06 2 1	CCR=(A)	x	x	x	x	x	x
tba	-- - -	-- - -	-- - -	-- - -	-- - -	17 2 1	A=(B)	-	-	x	x	0	-
tpa	-- - -	-- - -	-- - -	-- - -	-- - -	07 2 1	A=(CCR)	-	-	-	-	-	-
tst	-- - -	-- - -	6d 6 2	186d 7 3	7d 6 3	-- - -	(M)-$00	-	-	x	x	0	0
tsta	-- - -	-- - -	-- - -	-- - -	-- - -	4d 2 1	(A)-$00	-	-	x	x	0	0
tstb	-- - -	-- - -	-- - -	-- - -	-- - -	5d 2 1	(B)-$00	-	-	x	x	0	0
tsx	-- - -	-- - -	-- - -	-- - -	-- - -	30 3 1	IX=(SP)+1	-	-	-	-	-	-
tsy	-- - -	-- - -	-- - -	-- - -	-- - -	1830 4 2	IY=(SP)+1	-	-	-	-	-	-
txs	-- - -	-- - -	-- - -	-- - -	-- - -	35 1 1	SP=(IX)-1	-	-	-	-	-	-
tys	-- - -	-- - -	-- - -	-- - -	-- - -	1835 4 2	SP=(IY)-1	-	-	-	-	-	-
wai	-- - -	-- - -	-- - -	-- - -	-- - -	3e - 1	wait	-	5	-	-	-	-
xgdx	-- - -	-- - -	-- - -	-- - -	-- - -	8f 3 1	exch. D & IX	-	-	-	-	-	-
xgdy	-- - -	-- - -	-- - -	-- - -	-- - -	188f 4 2	exch. D & IY	-	-	-	-	-	-

NOTES:

Condition flag setting: - not changed, x updated according to data 3 c|=(msn>9), 4 Most significant bit of b, 5 Set when interrupt occurs.

Branch instructions: (except for BRCLR and BRSET) do not affect the condition code flags.

Arithmetic shifts: arithmetic shift left is the same as a logical shift left; arithmetic shift right preserves the sign bit and shifts the least significant bit into the C flag.

Logical shifts: logical shift left shifts 0 into the least significant bit and shifts the most significant bit into the C flag; logical shift right shifts 0 into the most significant bit and shifts the least significant bit into the C flag.

Rotates: a ROL shifts the bits one position left – the C flag goes into the least significant bit while the most significant bit goes into the C flag; a ROR shifts the bits one position right – the C flag goes into the most significant bit while the least significant bit goes into the C flag.

FDIV: computes an unsigned fractional divide of the value in D by the value in IX; the quotient is placed in IX and the remainder is placed in D.

IDIV: computes an unsigned integer divide of the value in D by the value in IX; the quotient is placed in IX and the remainder is placed in D.

MUL: computes the unsigned multiplication of the value in A by the value in B placing the unsigned product in D.

6811 Timer Section

Input Captures

The 6811 timer section has three input capture pins which can be used to capture an external event. The event can be programmed to be a rising edge, a falling edge or either edge. When an input event occurs the value of the free-running timer TCNT is copied into a time of capture(TIC) register.

The following table shows
   * the port bit associated with each of the three input captures
   * the interrupt vector location for each pin
   * the default value of each interrupt vector(the jump table location)
   * the location of the TIC register for each input capture pin

Input Capture	Port Location	Int. Vector	Jump Table LOcation	Time of Input Capture (TIC)
1	Port A, Bit 2	$FFEE-$FFEF	$00E8	$1010-$1011
2	Port A, Bit 1	$FFEC-$FFED	$00E5	$1012-$1013
3	Port A, Bit 0	$FFEA-$FFEB	$00E2	$1014-$1015

The type of event, the input capture flag and the interrupt enable are contained in three registers

Name	Addr.	B7	B6	B5	B4	B3	B2	B1	B0
TCTL2	$1021	0	0	E1B	E1A	E2B	E2A	E3B	E3A
TMSK1	$1022						IC1I	IC2I	IC3I
TFLG1	$1023						IC1F	IC2F	IC3F

ICnI is the interrupt enable flag for input capture n and ICnF is the event flag for input capture n.

The edge sensitivity is programmed as follows:

EnB	EnA	
0	0	Capture disabled
0	1	Capture on rising edge
1	0	Capture on falling edge
1	1	Capture on either edge

Output Compares

The 6811 has 5 output compares (OC), but you should avoid using OC1 as it is a multi-function pin with several unique features. The program loads TOC register with a value and when TCNT reaches the specified value, the output compare event happens. The particular event is programmable and an interrupt can be generated.

The following table is similar to the first table above:

Output Compare	Port Location	Int. Vector	Jump Table Location	Time of Output Compare (TOC)
1	Port A, Bit 7	$FFE8-$FFE9	$00DF	$1016-$1017
2	Port A, Pin 6	$FFE6-$FFE7	$00DC	$1018-$1019
3	Port A, Pin 5	$FFE4-$FFE5	$00D9	$101A-$101B
4	Port A, Pin 4	$FFE2-$FFE3	$00D6	$101C-$101D
5	Port A, Pin 3	$FFE0-$FFE1	$00D3	$101E-$101F

The type of event, the input capture flag and the interrupt enable are contained in three registers

Name	Addr.	B7	B6	B5	B4	B3	B2	B1	B0
TCTL1	$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
TMSK1	$1022	OC1I	OC2I	OC3I	OC4I	OC5I			
TFLG1	$1023	OC1F	OC2F	OC3F	OC4F	OC5F			

The output event is programmed as follows:

OMn	OLn	
0	0	Pin is not affected (OC1 may be)
0	1	Toggle OCn
1	0	Clear OCn to 0
1	1	Set OCn to 1

PORT Addresses: A $1000; B $1004; C $1003; D $1008; E $100A

Other Addresses: DDRC $1007; DDRD $1009; TCNT $100E,$100F