Toshiba 8X MCU

Instruction Set Table
Instruction OpCode Matrix
Registers and Control Bit Assignments

based on M68HC11 reference datasheet manual
based on tdis.pl by Greg Kunyavsky

by David Sobon <dsobon+supramania.gmail@com>

v0.01 (30-Apr-2011)

Index

Instructions

D8X Name	***D8X OpCo***	***6811 Name***	***6811 OpCo***	***Desc***	***Addr Mode***	***Example***	***H***	***I***	***N***	***Z***	***V***	***C***
add	**0x08**	ABA	0x1B	**A** ← A + **B**	INH	add a, b	⧫	-	⧫	⧫	⧫	⧫
add	**0x0E**	ABX	0x3A	**X** ← X + (00:**B**)	INH	add x, b	-	-	-	-	-	-
add	**0x0F**	ABY	0x3A	**Y** ← Y + (00:**B**)	INH	add y, b	-	-	-	-	-	-
add	**0xC0**	ADDA	0x8B	**A** ← A + #**xx**	IMM	add a, #xx	⧫	-	⧫	⧫	⧫	⧫
add	**0xD0**	ADDA	0x9B	**A** ← A + $**xx**	DIR	add a, $xx	⧫	-	⧫	⧫	⧫	⧫
add	**0xF0**	ADDA	0xBB	**A** ← A + $**xxxx**	EXT	add a, $xxxx	⧫	-	⧫	⧫	⧫	⧫
add	**0xE0**	ADDA	0xAB	**A** ← A + $**&**(**X**+**O**)	IND,X	add a, x + 0xXX	⧫	-	⧫	⧫	⧫	⧫
add	**0xE0**	ADDA	0xAB	**A** ← A + $**&**(**Y**+**O**)	IND,Y	add a, y + 0xXX	⧫	-	⧫	⧫	⧫	⧫
add	**0xC1**	ADDB	0xCB	**B** ← B + #**xx**	IMM	add b, #xx	⧫	-	⧫	⧫	⧫	⧫
add	**0xD1**	ADDB	0xDB	**B** ← B + $**xx**	DIR	add b, $xx	⧫	-	⧫	⧫	⧫	⧫
add	**0xF1**	ADDB	0xFB	**B** ← B + $**xxxx**	EXT	add b, $xxxx	⧫	-	⧫	⧫	⧫	⧫
add	**0xE1**	ADDB	0xEB	**B** ← B + $**&**(**X**+**O**)	IND,X	add b, x + 0xXX	⧫	-	⧫	⧫	⧫	⧫
add	**0xE1**	ADDB	0xEB	**B** ← B + $**&**(**Y**+**O**)	IND,Y	add b, y + 0xXX	⧫	-	⧫	⧫	⧫	⧫
add	**0x87**	ADDD	0xC3	**D** ← D + #**xx**	IMM	add d, #xx	-	-	⧫	⧫	⧫	⧫
add	**0x97**	ADDD	0xD3	**D** ← D + $(**xx:xx+1**)	DIR	add d, $xx	-	-	⧫	⧫	⧫	⧫
add	**0xB7**	ADDD	0xF3	**D** ← D + $(**xxxx:xxxx+1**)	EXT	add d, $xxxx	-	-	⧫	⧫	⧫	⧫
add	**0xA7**	ADDD	0xE3	**D** ← D + $**&**(**X**+**O**:**X**+**O**+**1**)	IND,X	add d, x + 0xXX	-	-	⧫	⧫	⧫	⧫
add	**0xA7**	ADDD	0xE3	**D** ← D + $**&**(**Y**+**O**:**Y**+**O**+**1**)	IND,Y	add d, y + 0xXX	-	-	⧫	⧫	⧫	⧫
addc	**0x80**	ADCA	0x89	**A** ← A + #**xx** + C	IMM	addc a, #xx	⧫	-	⧫	⧫	⧫	⧫
addc	**0x90**	ADCA	0x99	**A** ← A + $**xx** + C	DIR	addc a, $xx	⧫	-	⧫	⧫	⧫	⧫
addc	**0xB0**	ADCA	0xB9	**A** ← A + $**xxxx** + C	EXT	addc a, $xxxx	⧫	-	⧫	⧫	⧫	⧫
addc	**0xA0**	ADCA	0xA9	**A** ← A + $&(**X**+**O**) + C	IND,X	addc a, x + 0xXX	⧫	-	⧫	⧫	⧫	⧫
addc	**0xA0**	ADCA	0xA9	**A** ← A + $&(**Y**+**O**) + C	IND,Y	addc a, y + 0xXX	⧫	-	⧫	⧫	⧫	⧫
---	**---**	ADCB	0xC9	---	IMM	---	⧫	-	⧫	⧫	⧫	⧫
---	**---**	ADCB	0xD9	---	DIR	---	⧫	-	⧫	⧫	⧫	⧫
---	**---**	ADCB	0xF9	---	EXT	---	⧫	-	⧫	⧫	⧫	⧫
---	**---**	ADCB	0xE9	---	IND,X	---	⧫	-	⧫	⧫	⧫	⧫
---	**---**	ADCB	0xE9	---	IND,Y	---	⧫	-	⧫	⧫	⧫	⧫

D8X Name	***D8X OpCo***	***6811 Name***	***6811 OpCo***	***Desc***	***Addr Mode***	***Example***	***H***	***I***	***N***	***Z***	***V***	***C***
bra	**0x40**	BRA	0x20	if (**1 == 1**) then BRANCH	REL	bra 0xXX	-	-	-	-	-	-
brn	**0x41**	BRN	0x21	if (**1 == 0**) then BRANCH	REL	brn 0xXX	-	-	-	-	-	-
bgt	**0x42**	BHI	0x22	if (**C + Z == 0**) then BRANCH	REL	bgt 0xXX	-	-	-	-	-	-
ble	**0x43**	BLS	0x23	if (**C + Z == 1**) then BRANCH	REL	ble 0xXX	-	-	-	-	-	-
bcc	**0x44**	BCC	0x24	if (**C == 0**) then BRANCH	REL	bcc 0xXX	-	-	-	-	-	-
bcc	**0x44**	BHS	0x24	if (**C == 0**) then BRANCH	REL	bcc 0xXX	-	-	-	-	-	-
bcs	**0x45**	BLO	0x25	if (**C == 1**) then BRANCH	REL	bcs 0xXX	-	-	-	-	-	-
bcs	**0x45**	BCS	0x25	if (**C == 1**) then BRANCH	REL	bcs 0xXX	-	-	-	-	-	-
bne	**0x46**	BNE	0x26	if (**Z == 0**) then BRANCH	REL	bne 0xXX	-	-	-	-	-	-
beq	**0x47**	BEQ	0x27	if (**Z == 1**) then BRANCH	REL	beq 0xXX	-	-	-	-	-	-
bvc	**0x48**	BVC	0x28	if (**V == 0**) then BRANCH	REL	bvc 0xXX	-	-	-	-	-	-
bvs	**0x49**	BVS	0x29	if (**V == 1**) then BRANCH	REL	bvs 0xXX	-	-	-	-	-	-
bpz	**0x4A**	BPL	0x2A	if (**N == 0**) then BRANCH	REL	bpz 0xXX	-	-	-	-	-	-
bmi	**0x4B**	BMI	0x2B	if (**N == 1**) then BRANCH	REL	bmi 0xXX	-	-	-	-	-	-
bge	**0x4C**	BGE	0x2C	if (**N^V == 0**) then BRANCH	REL	bge 0xXX	-	-	-	-	-	-
blta	**0x4D**	BLT	0x2D	if (**N^V == 1**) then BRANCH	REL	blta 0xXX	-	-	-	-	-	-
bgta	**0x4E**	BGT	0x2E	if ((**Z + N^V**) == **0**) then BR	REL	bgta 0xXX	-	-	-	-	-	-
blea	**0x4F**	BLE	0x2F	if ((**Z + N^V**) == **1**) then BR	REL	blea 0xXX	-	-	-	-	-	-
tbbc	**0x37**	BRCLR	0x13	if (**$xx.bit# == 0**) then BRANCH	DIR	tbbc bit.#, $xx, 0x..	-	-	-	-	-	-
---	**---**	BRCLR	0x1F	---	IND,X	---	-	-	-	-	-	-
---	**---**	BRCLR	0x1F	---	IND,Y	---	-	-	-	-	-	-
tbbs	**0x35**	BRSET	0x12	if (**$xx.bit# == 1**) then BRANCH	DIR	tbbs bit.#, $xx, 0x..	-	-	-	-	-	-
---	**---**	BRSET	0x1E	---	IND,X	---	-	-	-	-	-	-
---	**---**	BRSET	0x1E	---	IND,Y	---	-	-	-	-	-	-
bsr	**0x61**	BSR	0x8D	**JUMP** to **+OFFSET** and **RET**	REL	bsr 0xXX	-	-	-	-	-	-
jmp	**0x03**	JMP	0x7E	**JUMP &xxxx**	EXT	jmp 0xXXXX	-	-	-	-	-	-
jmp	**0x23**	JMP	0x6E	**JUMP** &$(**X+OFFSET**)	IND,X	jmp x + 0xXX	-	-	-	-	-	-
jmp	**0x23**	JMP	0x6E	**JUMP** &$(**Y+OFFSET**)	IND,Y	jmp y + 0xXX	-	-	-	-	-	-
jsr	**0x31**	JSR	0x9D	**JUMP** &(**xx:xx+1**) then **RET**	DIR	jmp $xx	-	-	-	-	-	-
jsr	**0x01**	JSR	0xBD	**JUMP &xxxx** then **RET**	EXT	jmp 0xXXXX	-	-	-	-	-	-
jsr	**0x21**	JSR	0xAD	**JUMP** &$(**X+O**) then **RET**	IND,X	jmp x + 0xXX	-	-	-	-	-	-
jsr	**0x21**	JSR	0xAD	**JUMP** &$(**Y+O**) then **RET**	IND,Y	jmp y + 0xXX	-	-	-	-	-	-
nop	**0x00**	NOP	0x01	**NO OP**	INH	nop	-	-	-	-	-	-
reti	**0x73**	RTI	0x3B		INH	reti	♦	♦	♦	♦	♦	♦
ret	**0x63**	RTS	0x39		INH	ret	-	-	-	-	-	-
---	**---**	STOP	0xCF	sleep	INH	---	-	-	-	-	-	-
wait	**0x83**	WAI	0x3E	stack regs & WAIT	INH	wait	-	-	-	-	-	-
---	**---**	SWI	0x3F	---	INH	---	-	1	-	-	-	-
---	**---**	TEST	0x00	address bus counts	INH	---	-	-	-	-	-	-

D8X Name	***D8X OpCo***	***6811 Name***	***6811 OpCo***	***Desc***	***Addr Mode***	***Example***	***H***	***I***	***N***	***Z***	***V***	***C***
cmp	**0x0B**	CBA	0x11	TEST(**A - B**)	INH	cmp a, b	-	-	♦	♦	♦	♦
cmpz	**0x58**	TSTA	0x4D	TEST(**A - 0**)	INH	cmpz a	-	-	♦	♦	0	0
cmpz	**0x59**	TSTB	0x5D	TEST(**B - 0**)	INH	cmpz b	-	-	♦	♦	0	0
cmp	**0xCC**	CMPA	0x81	TEST(**A - #xx**)	IMM	cmp a, #xx	-	-	♦	♦	♦	♦
cmp	**0xDC**	CMPA	0x91	TEST(**A - $xx**)	DIR	cmp a, $xx	-	-	♦	♦	♦	♦
cmp	**0xFC**	CMPA	0xB1	TEST(**A - $xxxx**)	EXT	cmp a, $xxxx	-	-	♦	♦	♦	♦
cmp	**0xEC**	CMPA	0xA1	TEST(**A - $&(X+O)**)	IND,X	cmp a, x + 0xXX	-	-	♦	♦	♦	♦
cmp	**0xEC**	CMPA	0xA1	TEST(**A - $&(Y+O)**)	IND,Y	cmp a, y + 0xXX	-	-	♦	♦	♦	♦
cmp	**0xCD**	CMPB	0xC1	TEST(**B - #xx**)	IMM	cmp b, #xx	-	-	♦	♦	♦	♦
cmp	**0xDD**	CMPB	0xD1	TEST(**B - $xx**)	DIR	cmp b, $xx	-	-	♦	♦	♦	♦
cmp	**0xFD**	CMPB	0xF1	TEST(**B - $xxxx**)	EXT	cmp b, $xxxx	-	-	♦	♦	♦	♦
cmp	**0xED**	CMPB	0xE1	TEST(**B - $&(X+O)**)	IND,X	cmp b, x + 0xXX	-	-	♦	♦	♦	♦
cmp	**0xED**	CMPB	0xE1	TEST(**B - $&(Y+O)**)	IND,Y	cmp b, y + 0xXX	-	-	♦	♦	♦	♦
cmp	**0x8C**	CPX	0x8C	TEST(**X - #xxxx**)	IMM	cmp x, #xxxx	-	-	♦	♦	♦	♦
cmp	**0x9C**	CPX	0x9C	TEST(**X - $(xx:xx+1)**)	DIR	cmp x, $xx	-	-	♦	♦	♦	♦
cmp	**0xBC**	CPX	0xBC	TEST(**X - $(xxxx:xxxx+1)**)	EXT	cmp x, $xxxx	-	-	♦	♦	♦	♦
cmp	**0xAC**	CPX	0xAC	TEST(**X - $&(X+OFFSET)**)	IND,X	cmp x, x + 0xXX	-	-	♦	♦	♦	♦
cmp	**0xAC**	CPX	0xAC	TEST(**X - $&(Y+OFFSET)**)	IND,Y	cmp x, y + 0xXX	-	-	♦	♦	♦	♦
cmp	**0x8D**	CPY	0x8C	TEST(**Y - #xxxx**)	IMM	cmp y, #xxxx	-	-	♦	♦	♦	♦
cmp	**0x9D**	CPY	0x9C	TEST(**Y - $(xx:xx+1)**)	DIR	cmp y, $xx	-	-	♦	♦	♦	♦
cmp	**0xBD**	CPY	0xBC	TEST(**Y - $(xxxx:xxxx+1)**)	EXT	cmp y, $xxxx	-	-	♦	♦	♦	♦
cmp	**0xAD**	CPY	0xAC	TEST(**Y - $&(X+OFFSET)**)	IND,X	cmp y, x + 0xXX	-	-	♦	♦	♦	♦
cmp	**0xAD**	CPY	0xAC	TEST(**Y - $&(Y+OFFSET)**)	IND,Y	cmp y, y + 0xXX	-	-	♦	♦	♦	♦
cmp	**0x89**	CPD	0x83	TEST(**D - #xxxx**)	IMM	cmp d, #xxxx	-	-	♦	♦	♦	♦
cmp	**0x99**	CPD	0x93	TEST(**D - $(xx:xx+1)**)	DIR	cmp d, $xx	-	-	♦	♦	♦	♦
cmp	**0xB9**	CPD	0xB3	TEST(**D - $(xxxx:xxxx+1)**)	EXT	cmp d, $xxxx	-	-	♦	♦	♦	♦
cmp	**0xA9**	CPD	0xA3	TEST(**D - $&(X+OFFSET)**)	IND,X	cmp d, x + 0xXX	-	-	♦	♦	♦	♦
cmp	**0xA9**	CPD	0xA3	TEST(**D - $&(Y+OFFSET)**)	IND,Y	cmp d, y + 0xXX	-	-	♦	♦	♦	♦
cmpb	**0xCD**	BITA	0x85	TEST(**A & #xx**)	IMM	cmpb a, #xx	-	-	♦	♦	0	-
cmpb	**0xDD**	BITA	0x95	TEST(**A & $xx**)	DIR	cmpb a, $xx	-	-	♦	♦	0	-
cmpb	**0xFD**	BITA	0xB5	TEST(**A & $xxxx**)	EXT	cmpb a, $xxxx	-	-	♦	♦	0	-
cmpb	**0xED**	BITA	0xA5	TEST(**A & $&(X+O)**)	IND,X	cmpb a, x + 0xXX	-	-	♦	♦	0	-
cmpb	**0xED**	BITA	0xA5	TEST(**A & $&(Y+O)**)	IND,Y	cmpb a, y + 0xXX	-	-	♦	♦	0	-
cmpb	**0xCF**	BITB	0xC5	TEST(**B & #xx**)	IMM	cmpb b, #xx	-	-	♦	♦	0	-
cmpb	**0xDF**	BITB	0xD5	TEST(**B & $xx**)	DIR	cmpb b, $xx	-	-	♦	♦	0	-
cmpb	**0xFF**	BITB	0xF5	TEST(**B & $xxxx**)	EXT	cmpb b, $xxxx	-	-	♦	♦	0	-
cmpb	**0xEF**	BITB	0xE5	TEST(**B & $&(X+O)**)	IND,X	cmpb b, x + 0xXX	-	-	♦	♦	0	-
cmpb	**0xEF**	BITB	0xE5	TEST(**B & $&(Y+O)**)	IND,Y	cmpb b, y + 0xXX	-	-	♦	♦	0	-

D8X Name	***D8X OpCo***	***6811 Name***	***6811 OpCo***	***Desc***	***Addr Mode***	***Example***	***H***	***I***	***N***	***Z***	***V***	***C***
ld	**0x1A**	---	---	**A** ← $&(**Y**) ; **Y++**	INH	ld a, [y]						
ld	**0x1B**	---	---	**D** ← $&(**Y:Y+1**) ; **Y++; Y++**	INH	ld d, [y]						
ld	**0xCA**	LDAA	0x86	**A** ← #**xx**	IMM	ld a, #xx	-	-	♦	♦	0	-
ld	**0xDA**	LDAA	0x96	**A** ← $**xx**	DIR	ld a, $xx	-	-	♦	♦	0	-
ld	**0xFA**	LDAA	0xB6	**A** ← $**xxxx**	EXT	ld a, 0xXXXX	-	-	♦	♦	0	-
ld	**0xEA**	LDAA	0xA6	**A** ← $**&(X+OFFSET**)	IND,X	ld a, x + 0xXX	-	-	♦	♦	0	-
ld	**0xEA**	LDAA	0xA6	**A** ← $**&(Y+OFFSET**)	IND,Y	ld a, y + 0xXX	-	-	♦	♦	0	-
ld	**0xCB**	LDAB	0xC6	**B** ← #**xx**	IMM	ld b, #xx	-	-	♦	♦	0	-
ld	**0xDB**	LDAB	0xD6	**B** ← $**xx**	DIR	ld b, $xx	-	-	♦	♦	0	-
ld	**0xFB**	LDAB	0xF6	**B** ← $**xxxx**	EXT	ld b, 0xXXXX	-	-	♦	♦	0	-
ld	**0xEB**	LDAB	0xE6	**B** ← $**&(X+OFFSET**)	IND,X	ld b, x + 0xXX	-	-	♦	♦	0	-
ld	**0xEB**	LDAB	0xE6	**B** ← $**&(Y+OFFSET**)	IND,Y	ld b, y + 0xXX	-	-	♦	♦	0	-
ld	**0x86**	LDD	0xCC	**D** ← #**xxxx**	IMM	ld d, #xxxx	-	-	♦	♦	0	-
ld	**0x96**	LDD	0xDC	**D** ← $(**xx:xx+1**)	DIR	ld d, $xx	-	-	♦	♦	0	-
ld	**0xB6**	LDD	0xFC	**D** ← $(**xxxx:xxxx+1**)	EXT	ld d, 0xXXXX	-	-	♦	♦	0	-
ld	**0xA6**	LDD	0xEC	**D** ← $**&(X+O:X+O+1**)	IND,X	ld d, x + 0xXX	-	-	♦	♦	0	-
ld	**0xA6**	LDD	0xEC	**D** ← $**&(Y+O:Y+O+1**)	IND,Y	ld d, y + 0xXX	-	-	♦	♦	0	-
ld	**0x8B**	LDS	0x8E	**SP** ← #**xxxx**	IMM	ld s, #xxxx	-	-	♦	♦	0	-
ld	**0x9B**	LDS	0x9E	**SP** ← $(**xx:xx+1**)	DIR	ld s, $xx	-	-	♦	♦	0	-
ld	**0xBB**	LDS	0xBE	**SP** ← $(**xxxx:xxxx+1**)	EXT	ld s, 0xXXXX	-	-	♦	♦	0	-
ld	**0xAB**	LDS	0xAE	**SP** ← $**&(X+O:X+O+1**)	IND,X	ld s, x + 0xXX	-	-	♦	♦	0	-
ld	**0xAB**	LDS	0xAE	**SP** ← $**&(Y+O:Y+O+1**)	IND,Y	ld s, y + 0xXX	-	-	♦	♦	0	-
ld	**0x8E**	LDX	0xCE	**X** ← #**xxxx**	IMM	ld x, #xxxx	-	-	♦	♦	0	-
ld	**0x9E**	LDX	0xDE	**X** ← $(**xx:xx+1**)	DIR	ld x, $xx	-	-	♦	♦	0	-
ld	**0xBE**	LDX	0xFE	**X** ← $(**xxxx:xxxx+1**)	EXT	ld x, 0xXXXX	-	-	♦	♦	0	-
ld	**0xAE**	LDX	0xEE	**X** ← $**&(X+O:X+O+1**)	IND,X	ld x, x + 0xXX	-	-	♦	♦	0	-
ld	**0xAE**	LDX	0xEE	**X** ← $**&(Y+O:Y+O+1**)	IND,Y	ld x, y + 0xXX	-	-	♦	♦	0	-
ld	**0x8F**	LDY	0xCE	**Y** ← #**xxxx**	IMM	ld y, #xxxx	-	-	♦	♦	0	-
ld	**0x9F**	LDY	0xDE	**Y** ← $(**xx:xx+1**)	DIR	ld y, $xx	-	-	♦	♦	0	-
ld	**0xBF**	LDY	0xFE	**Y** ← $(**xxxx:xxxx+1**)	EXT	ld y, 0xXXXX	-	-	♦	♦	0	-
ld	**0xAF**	LDY	0xEE	**Y** ← $**&(X+O:X+O+1**)	IND,X	ld y, x + 0xXX	-	-	♦	♦	0	-
ld	**0xAF**	LDY	0xEE	**Y** ← $**&(Y+O:Y+O+1**)	IND,Y	ld y, y + 0xXX	-	-	♦	♦	0	-

D8X Name	***D8X OpCo***	***6811 Name***	***6811 OpCo***	***Desc***	***Addr Mode***	***Example***	***H***	***I***	***N***	***Z***	***V***	***C***
and	**0xC2**	ANDA	0x84	**A** ← A & #**xx**	IMM	and a, #xx	-	-	⧫	⧫	0	-
and	**0xD2**	ANDA	0x94	**A** ← A & $**xx**	DIR	and a, $xx	-	-	⧫	⧫	0	-
and	**0xF2**	ANDA	0xB4	**A** ← A & $**xxxx**	EXT	and a, $xxxx	-	-	⧫	⧫	0	-
and	**0xE2**	ANDA	0xA4	**A** ← A & $**&(X+OFFSET**)	IND,X	and a, x + 0xXX	-	-	⧫	⧫	0	-
and	**0xE2**	ANDA	0xA4	**A** ← A & $**&(Y+OFFSET**)	IND,Y	and a, y + 0xXX	-	-	⧫	⧫	0	-
and	**0xC3**	ANDB	0xC4	**B** ← B & #**xx**	IMM	and b, #xx	-	-	⧫	⧫	0	-
and	**0xD3**	ANDB	0xD4	**B** ← B & $**xx**	DIR	and b, $xx	-	-	⧫	⧫	0	-
and	**0xF3**	ANDB	0xF4	**B** ← B & $**xxxx**	EXT	and b, $xxxx	-	-	⧫	⧫	0	-
and	**0xE3**	ANDB	0xE4	**B** ← B & $**&(X+OFFSET**)	IND,X	and b, x + 0xXX	-	-	⧫	⧫	0	-
and	**0xE3**	ANDB	0xE4	**B** ← B & $**&(Y+OFFSET**)	IND,Y	and b, y + 0xXX	-	-	⧫	⧫	0	-
or	**0xC6**	ORAA	0x8A	**A** ← A | #**xx**	IMM	or a, #xx	-	-	⧫	⧫	0	-
or	**0xD6**	ORAA	0x9A	**A** ← A | $**xx**	DIR	or a, $xx	-	-	⧫	⧫	0	-
or	**0xF6**	ORAA	0xBA	**A** ← A | $**xxxx**	EXT	or a, 0xXXXX	-	-	⧫	⧫	0	-
or	**0xE6**	ORAA	0xAA	**A** ← A | $**&(X+OFFSET**)	IND,X	or a, x + 0xXX	-	-	⧫	⧫	0	-
or	**0xE6**	ORAA	0xAA	**A** ← A | $**&(Y+OFFSET**)	IND,Y	or a, y + 0xXX	-	-	⧫	⧫	0	-
or	**0xC7**	ORAB	0xCA	**B** ← B | #**xx**	IMM	or b, #xx	-	-	⧫	⧫	0	-
or	**0xD7**	ORAB	0xDA	**B** ← B | $**xx**	DIR	or b, $xx	-	-	⧫	⧫	0	-
or	**0xF7**	ORAB	0xFA	**B** ← B | $**xxxx**	EXT	or b, 0xXXXX	-	-	⧫	⧫	0	-
or	**0xE7**	ORAB	0xEA	**B** ← B | $**&(X+OFFSET**)	IND,X	or b, x + 0xXX	-	-	⧫	⧫	0	-
or	**0xE7**	ORAB	0xEA	**B** ← B | $**&(Y+OFFSET**)	IND,Y	or b, y + 0xXX	-	-	⧫	⧫	0	-
xor	**0xC8**	EORA	0x88	**A** ← A ^ #**xx**	IMM	xor a, #xx	-	-	⧫	⧫	0	-
xor	**0xD8**	EORA	0x98	**A** ← A ^ $**xx**	DIR	xor a, $xx	-	-	⧫	⧫	0	-
xor	**0xF8**	EORA	0xB8	**A** ← A ^ $**xxxx**	EXT	xor a, $xxxx	-	-	⧫	⧫	0	-
xor	**0xE8**	EORA	0xA8	**A** ← A ^ $**&(X+OFFSET)**	IND,X	xor a, x + 0xXX	-	-	⧫	⧫	0	-
xor	**0xE8**	EORA	0xA8	**A** ← A ^ $**&(Y+OFFSET)**	IND,Y	xor a, y + 0xXX	-	-	⧫	⧫	0	-
xor	**0xC9**	EORB	0xC8	**B** ← B ^ #**xx**	IMM	xor b, #xx	-	-	⧫	⧫	0	-
xor	**0xD9**	EORB	0xD8	**B** ← B ^ $**xx**	DIR	xor b, $xx	-	-	⧫	⧫	0	-
xor	**0xF9**	EORB	0xF8	**B** ← B ^ $**xxxx**	EXT	xor b, $xxxx	-	-	⧫	⧫	0	-
xor	**0xE9**	EORB	0xE8	**B** ← B ^ $**&(X+OFFSET)**	IND,X	xor b, x + 0xXX	-	-	⧫	⧫	0	-
xor	**0xE9**	EORB	0xE8	**B** ← B ^ $**&(Y+OFFSET)**	IND,Y	xor b, y + 0xXX	-	-	⧫	⧫	0	-
---	---	COM	0x73	---	EXT	---	-	-	⧫	⧫	0	1
---	---	COM	0x63	---	IND,X	---	-	-	⧫	⧫	0	1
---	---	COM	0x63	---	IND,Y	---	-	-	⧫	⧫	0	1
---	---	COMA	0x43	---	INH	---	-	-	⧫	⧫	0	1
---	---	COMB	0x53	---	INH	---	-	-	⧫	⧫	0	1

D8X Name	*D8X OpCo*	*6811 Name*	*6811 OpCo*	*Desc*	*Addr Mode*	*Example*	*H*	*I*	*N*	*Z*	*V*	*C*
inc	**0x76**	INC	0x7C	**&xx** ← **$xx** + 1	EXT	inc $xx	-	-	♦	♦	♦	-
inc	**0x66**	INC	0x6C	**$&(X+OFFSET)**++	IND,X	inc x + 0xXX	-	-	♦	♦	♦	-
inc	**0x66**	INC	0x6C	**$&(Y+OFFSET)**++	IND,Y	inc y + 0xXX	-	-	♦	♦	♦	-
inc	**0x56**	INCA	0x4C	**A** ← A + 1	INH	inc a	-	-	♦	♦	♦	-
inc	**0x57**	INCB	0x5C	**B** ← B + 1	INH	inc b	-	-	♦	♦	♦	-
inc	**0x1C**	INX	0x08	**X** ← X + 1	INH	inc x	-	-	-	♦	-	-
inc	**0x1D**	INY	0x08	**Y** ← Y + 1	INH	inc y	-	-	-	♦	-	-
inc	**0x2D**	INS	0x31	**SP** ← SP + 1	INH	inc s	-	-	-	-	-	-
dec	**0x70**	DEC	0x7A	**&xx** ← **$xx** - 1	EXT	dec $xx	-	-	♦	♦	♦	-
dec	**0x60**	DEC	0x6A	**$&(X+OFFSET)--**	IND,X	dec x + 0xXX	-	-	♦	♦	♦	-
dec	**0x60**	DEC	0x6A	**$&(Y+OFFSET)--**	IND,Y	dec y + 0xXX	-	-	♦	♦	♦	-
dec	**0x50**	DECA	0x4A	**A** ← A - 1	INH	dec a	-	-	♦	♦	♦	-
dec	**0x51**	DECB	0x5A	**B** ← B - 1	INH	dec b	-	-	♦	♦	♦	-
dec	**0x1E**	DEX	0x09	**X** ← X - 1	INH	dec x	-	-	-	♦	-	-
dec	**0x1F**	DEY	0x09	**Y** ← Y - 1	INH	dec y	-	-	-	♦	-	-
dec	**0x2F**	DES	0x34	**SP** ← SP - 1	INH	dec s	-	-	-	-	-	-
clr	**0x72**	CLR	0x7F	**&xx** ← 0	EXT	clr $xx	-	-	0	1	0	0
clr	**0x62**	CLR	0x6F	**&(X+O)** ← 0	IND,X	clr x + 0xXX	-	-	0	1	0	0
clr	**0x62**	CLR	0x6F	**&(Y+O)** ← 0	IND,Y	clr y + 0xXX	-	-	0	1	0	0
clr	**0x52**	CLRA	0x4F	**A** ← 0	INH	clr a	-	-	0	1	0	0
clr	**0x53**	CLRB	0x5F	**B** ← 0	INH	clr b	-	-	0	1	0	0
clrb	**0x75**	BCLR	0x15	**&xx.bit#** ← 0	DIR	clrb bit#, $xx	-	-	♦	♦	0	-
---	**---**	BCLR	0x1D	---	IND,X	---	-	-	♦	♦	0	-
---	**---**	BCLR	0x1D	---	IND,Y	---	-	-	♦	♦	0	-
setb	**0x77**	BSET	0x14	**&xx.bit#** ← 1	DIR	setb bit#, $xx	-	-	♦	♦	0	-
---	**---**	BSET	0x1C	---	IND,X	---	-	-	♦	♦	0	-
---	**---**	BSET	0x1C	---	IND,Y	---	-	-	♦	♦	0	-
clrc	**0x65**	CLC	0x0C	**C** ← 0	INH	clrc	-	-	-	-	-	0
setc	**0x67**	SEC	0x0D	**C** ← 1	INH	setc	-	-	-	-	-	1
clrv	**0x25**	CLV	0x0A	**V** ← 0	INH	clrv	-	-	-	-	0	-
setv	**0x27**	SEV	0x0B	**V** ← 1	INH	setv	-	-	-	-	1	-
di	**0x05**	CLI	0x0E	**I** ← 0	INH	di	-	0	-	-	-	-
ei	**0x07**	SEI	0x0F	**I** ← 1	INH	ei	-	1	-	-	-	-

D8X Name	***D8X OpCo***	***6811 Name***	***6811 OpCo***	***Desc***	***Addr Mode***	***Example***	***H***	***I***	***N***	***Z***	***V***	***C***
sub	**0x09**	SBA	0x10	**A** ← A **-** B	INH	sub a, b	-	-	⧫	⧫	⧫	⧫
sub	**0xC4**	SUBA	0x80	**A** ← A **- #xx**	IMM	sub a, #xx	-	-	⧫	⧫	⧫	⧫
sub	**0xD4**	SUBA	0x90	**A** ← A **- $xx**	DIR	sub a, $xx	-	-	⧫	⧫	⧫	⧫
sub	**0xF4**	SUBA	0xB0	**A** ← A **- $xxxx**	EXT	sub a, $xxxx	-	-	⧫	⧫	⧫	⧫
sub	**0xE4**	SUBA	0xA0	**A** ← A **- $&(X + O)**	IND,X	sub a, x + 0xXX	-	-	⧫	⧫	⧫	⧫
sub	**0xE4**	SUBA	0xA0	**A** ← A **- $&(Y + O)**	IND,Y	sub a, y + 0xXX	-	-	⧫	⧫	⧫	⧫
sub	**0xC5**	SUBB	0xC0	**B** ← B **- #xx**	IMM	sub b, #xx	-	-	⧫	⧫	⧫	⧫
sub	**0xD5**	SUBB	0xD0	**B** ← B **- $xx**	DIR	sub b, $xx	-	-	⧫	⧫	⧫	⧫
sub	**0xF5**	SUBB	0xF0	**B** ← B **- $xxxx**	EXT	sub b, $xxxx	-	-	⧫	⧫	⧫	⧫
sub	**0xE5**	SUBB	0xE0	**B** ← B **- $&(X + O)**	IND,X	sub b, x + 0xXX	-	-	⧫	⧫	⧫	⧫
sub	**0xE5**	SUBB	0xE0	**B** ← B **- $&(Y + O)**	IND,Y	sub b, y + 0xXX	-	-	⧫	⧫	⧫	⧫
sub	**0x88**	SUBD	0x83	**D** ← D **- #xxxx**	IMM	sub d, #xx	-	-	⧫	⧫	⧫	⧫
sub	**0x98**	SUBD	0x93	**D** ← D **-** $**(xx:xx+1)**	DIR	sub d, $xx	-	-	⧫	⧫	⧫	⧫
sub	**0xB8**	SUBD	0xB3	**D** ← D **-** $**(xxxx:xxxx+1)**	EXT	sub d, $xxxx	-	-	⧫	⧫	⧫	⧫
sub	**0xA8**	SUBD	0xA3	**D** ← D **- $&(X+O:X+O+1)**	IND,X	sub d, x + 0xXX	-	-	⧫	⧫	⧫	⧫
sub	**0xA8**	SUBD	0xA3	**D** ← D **- $&(Y+O:Y+O+1)**	IND,Y	sub d, y + 0xXX	-	-	⧫	⧫	⧫	⧫
subc	**0x84**	SBCA	0x82	**A** ← A **- #xx** - C	IMM	subc a, #xx	-	-	⧫	⧫	⧫	⧫
subc	**0x94**	SBCA	0x92	**A** ← A **- $xx** - C	DIR	subc a, $xx	-	-	⧫	⧫	⧫	⧫
subc	**0xB4**	SBCA	0xB2	**A** ← A **- $xxxx** - C	EXT	subc a, $xxxx	-	-	⧫	⧫	⧫	⧫
subc	**0xA4**	SBCA	0xA2	**A** ← A **- $&(X+O)** - C	IND,X	subc a, x + 0xXX	-	-	⧫	⧫	⧫	⧫
subc	**0xA4**	SBCA	0xA2	**A** ← A **- $&(Y+O)** - C	IND,Y	subc a, y + 0xXX	-	-	⧫	⧫	⧫	⧫
---	**---**	SBCB	0xC2	**B** ← B **- #xx** - C	IMM	---	-	-	⧫	⧫	⧫	⧫
---	**---**	SBCB	0xD2	**B** ← B **- $xx** - C	DIR	---	-	-	⧫	⧫	⧫	⧫
---	**---**	SBCB	0xF2	**B** ← B **- $xxxx** - C	EXT	---	-	-	⧫	⧫	⧫	⧫
---	**---**	SBCB	0xE2	**B** ← B **- $&(X+O)** - C	IND,X	---	-	-	⧫	⧫	⧫	⧫
---	**---**	SBCB	0xE2	**B** ← B **- $&(Y+O)** - C	IND,Y	---	-	-	⧫	⧫	⧫	⧫
neg	**0x74**	NEG	0x70	**&xx** ← 0 - $**xx**	EXT	neg $xx	-	-	⧫	⧫	⧫	⧫
neg	**0x64**	NEG	0x60	**&(X+O)** ← 0 - $**&(X+O)**	IND,X	neg x + 0xXX	-	-	⧫	⧫	⧫	⧫
neg	**0x64**	NEG	0x60	**&(Y+O)** ← 0 - $**&(Y+O)**	IND,Y	neg y + 0xXX	-	-	⧫	⧫	⧫	⧫
neg	**0x54**	NEGA	0x40	**A** ← 0 - A	INH	neg a	-	-	⧫	⧫	⧫	⧫
neg	**0x55**	NEGB	0x50	**B** ← 0 - B	INH	neg b	-	-	⧫	⧫	⧫	⧫
adj	**0x5E**	DAA	0x19	Adjust sum to BCD	INH	adj a	-	-	⧫	⧫	⧫	⧫
---	**---**	TST	0x7D	TEST(**$xxxx - 0**)	EXT	---	-	-	⧫	⧫	0	0
---	**---**	TST	0x6D	TEST(**($&(X+O:X+O+1)) - 0**)	IND,X	---	-	-	⧫	⧫	0	0
---	**---**	TST	0x6D	TEST(**($&(Y+O:Y+O+1)) - 0**)	IND,Y	---	-	-	⧫	⧫	0	0

D8X Name	*D8X OpCo*	*6811 Name*	*6811 OpCo*	*Desc*	*Addr Mode*	*Example*	*H*	*I*	*N*	*Z*	*V*	*C*
st	**0x82**	---	---	**&(Y)** ← A ; **Y++**	INH	st a, [y]	-	-	⧫	⧫	0	-
st	**0x8A**	---	---	**&(Y:Y+1)** ← D ; **Y++; Y++**	INH	st d, [y]	-	-	⧫	⧫	0	-
st	**0x92**	STAA	0x97	**&xx** ← A	DIR	st a, $xx	-	-	⧫	⧫	0	-
st	**0xB2**	STAA	0xB7	**&xxxx** ← A	EXT	st a, $xxxx	-	-	⧫	⧫	0	-
st	**0xA2**	STAA	0xA7	**&(X+O)** ← A	IND,X	st a, x + 0xXX	-	-	⧫	⧫	0	-
st	**0xA2**	STAA	0xA7	**&(Y+O)** ← A	IND,Y	st a, y + 0xXX	-	-	⧫	⧫	0	-
st	**0x93**	STAB	0xD7	**&xx** ← B	DIR	st b, $xx	-	-	⧫	⧫	0	-
st	**0xB3**	STAB	0xF7	**&xxxx** ← B	EXT	st b, $xxxx	-	-	⧫	⧫	0	-
st	**0xA3**	STAB	0xE7	**&(X+O)** ← B	IND,X	st b, x + 0xXX	-	-	⧫	⧫	0	-
st	**0xA3**	STAB	0xE7	**&(Y+O)** ← B	IND,Y	st b, y + 0xXX	-	-	⧫	⧫	0	-
st	**0x9A**	STD	0xDD	**&(xx:xx+1)** ← D	DIR	st d, $xx	-	-	⧫	⧫	0	-
st	**0xBA**	STD	0xFD	**&(xxxx:xxxx+1)** ← D	EXT	st d, $xxxx	-	-	⧫	⧫	0	-
st	**0xAA**	STD	0xED	**&(X+O:X+O+1)** ← D	IND,X	st d, x + 0xXX	-	-	⧫	⧫	0	-
st	**0xAA**	STD	0xED	**&(Y+O:Y+O+1)** ← D	IND,Y	st d, y + 0xXX	-	-	⧫	⧫	0	-
st	**0x39**	STS	0x9F	**&(xx:xx+1)** ← SP	DIR	st s, $xx	-	-	⧫	⧫	0	-
---	**---**	STS	0xBF	**&(xxxx:xxxx+1)** ← SP	EXT	---	-	-	⧫	⧫	0	-
st	**0x29**	STS	0xAF	**&(X+O:X+O+1)** ← SP	IND,X	st s, x + 0xXX	-	-	⧫	⧫	0	-
st	**0x29**	STS	0xAF	**&(Y+O:Y+O+1)** ← SP	IND,Y	st s, y + 0xXX	-	-	⧫	⧫	0	-
st	**0x0A**	STX	0xDF	**&(xx:xx+1)** ← X	DIR	st x, $xx	-	-	⧫	⧫	0	-
st	**0x3A**	STX	0xFF	**&(xxxx:xxxx+1)** ← X	EXT	st x, $xxxx	-	-	⧫	⧫	0	-
st	**0x2A**	STX	0xEF	**&(X+O:X+O+1)** ← X	IND,X	st x, x + 0xXX	-	-	⧫	⧫	0	-
st	**0x2A**	STX	0xEF	**&(Y+O:Y+O+1)** ← X	IND,Y	st x, y + 0xXX	-	-	⧫	⧫	0	-
st	**0x3B**	STY	0xDF	**&(xx:xx+1)** ← Y	DIR	st y, $xx	-	-	⧫	⧫	0	-
---	**---**	STY	0xFF	**&(xxxx:xxxx+1)** ← Y	EXT	---	-	-	⧫	⧫	0	-
st	**0x2B**	STY	0xEF	**&(X+O:X+O+1)** ← Y	IND,X	st y, x + 0xXX	-	-	⧫	⧫	0	-
st	**0x2B**	STY	0xEF	**&(Y+O:Y+O+1)** ← Y	IND,Y	st y, y + 0xXX	-	-	⧫	⧫	0	-
push	**0x6C**	PSHA	0x36	**STACK** ← A; **SP** ← SP - 1	INH	push a	-	-	-	-	-	-
push	**0x6D**	PSHB	0x37	**STACK** ← B; **SP** ← SP - 1	INH	push b	-	-	-	-	-	-
push	**0x6E**	PSHX	0x3C	**STACK** ← X; **SP** ← SP - 2	INH	push x	-	-	-	-	-	-
push	**0x6F**	PSHY	0x3C	**STACK** ← Y; **SP** ← SP - 2	INH	push y	-	-	-	-	-	-
pull	**0x7C**	PULA	0x32	**SP** ← SP + 1; **A** ← STACK	INH	pull a	-	-	-	-	-	-
pull	**0x7D**	PULB	0x33	**SP** ← SP + 1; **B** ← STACK	INH	pull b	-	-	-	-	-	-
pull	**0x7E**	PULX	0x38	**SP** ← SP + 2; **X** ← STACK	INH	pull x	-	-	-	-	-	-
pull	**0x7F**	PULY	0x38	**SP** ← SP + 2; **Y** ← STACK	INH	pull y	-	-	-	-	-	-

D8X Name	*D8X OpCo*	*6811 Name*	*6811 OpCo*	*Desc*	*Addr Mode*	*Example*	*H*	*I*	*N*	*Z*	*V*	*C*
mov	**0x5A**	TBA	0x17	**A** ← B	INH	mov b, a	-	-	♦	♦	0	-
mov	**0x5B**	TAB	0x16	**B** ← A	INH	mov a, b	-	-	♦	♦	0	-
mov	**0x3C**	---	---	**D** ← X	INH	mov x, d						
mov	**0x3D**	---	---	**D** ← Y	INH	mov y, d						
mov	**0x3E**	---	---	**X** ← D	INH	mov d, x	-	-	-	-	-	-
mov	**0x3F**	---	---	**Y** ← D	INH	mov d, y	-	-	-	-	-	-
mov	**0x2E**	TSX	0x30	**X** ← SP + 1	INH	mov s, x	-	-	-	-	-	-
---	**---**	TSY	0x30	**Y** ← SP + 1	INH	---	-	-	-	-	-	-
mov	**0x2C**	TXS	0x35	**SP** ← X - 1	INH	mov x, s	-	-	-	-	-	-
---	**---**	TYS	0x35	**SP** ← Y - 1	INH	---	-	-	-	-	-	-
mov	**0x5D**	TAP	0x06	**CCR** ← A	INH	mov a, ocr	♦	♦	♦	♦	♦	♦
mov	**0x5C**	TPA	0x07	**A** ← CCR	INH	mov ocr, a	-	-	-	-	-	-
xch	**0x02**	---	---	**A** ← B; **B** ← A	INH	xch a, b	-	-	-	-	-	-
xch	**0x69**	---	---	**X** ← Y; **Y** ← X	INH	xch x, y	-	-	-	-	-	-
xch	**0x7A**	---	---	**A** ← $xx; **&xx** ← A	DIR	xch a, $xx	-	-	-	-	-	-
xch	**0x6A**	---	---	**A** ← $**&(X+O)**; **&(X+O)** ← A	IND,X	xch a, x + 0xXX	-	-	-	-	-	-
xch	**0x6A**	---	---	**A** ← $**&(Y+O)**; **&(Y+O)** ← A	IND,Y	xch a, y + 0xXX	-	-	-	-	-	-
xch	**0x7B**	---	---	**B** ← $xx; **&xx** ← B	DIR	xch b, $xx	-	-	-	-	-	-
xch	**0x6B**	---	---	**B** ← $**&(X+O)**; **&(X+O)** ← B	IND,X	xch b, x + 0xXX	-	-	-	-	-	-
xch	**0x6B**	---	---	**B** ← $**&(Y+O)**; **&(Y+O)** ← B	IND,Y	xch b, y + 0xXX	-	-	-	-	-	-
---	**---**	XGDX	0x8F	**D** ← X; **X** ← D	INH	---	-	-	-	-	-	-
---	**---**	XGDY	0x8F	**D** ← Y; **Y** ← D	INH	---	-	-	-	-	-	-
mul	**0x81**	---	---	**D** ← A * **#xx**	IMM	mul a, #xx	-	-	-	-	-	♦
mul	**0x91**	---	---	**D** ← A * **$xx**	DIR	mul a, $xx	-	-	-	-	-	♦
mul	**0xB1**	---	---	**D** ← A * **$xxxx**	EXT	mul a, $xxxx	-	-	-	-	-	♦
mul	**0xA1**	---	---	**D** ← A * **$&(X+O)**	IND,X	mul a, x + 0xXX	-	-	-	-	-	♦
mul	**0xA1**	---	---	**D** ← A * **$&(Y+O)**	IND,Y	mul a, y + 0xXX	-	-	-	-	-	♦
div	**0x85**	---	---	**B** ← D / **#xx** **; A** ← D % **#xx**	IMM	div d, #xx	-	-				
div	**0x95**	---	---	**B** ← D / **$xx** **; A** ← D % **$xx**	DIR	div d, $xx	-	-				
div	**0xB5**	---	---	**B** ← D / **$xxxx ; A**← D % **$xxxx**	EXT	div d, $xxxx	-	-				
div	**0xA5**	---	---	**B** ← D / **$&(X+O) ; A** ← D % ...	IND,X	div d, x + 0xXX	-	-				
div	**0xA5**	---	---	**B** ← D / **$&(Y+O) ; A** ← D % ...	IND,Y	div d, y + 0xXX	-	-				
---	**---**	FDIV	0x03	**X** ← D / X ; **D** ← r	INH	---	-	-	-	♦	♦	♦
---	**---**	IDIV	0x02	**X** ← D / X ; **D** ← r	INH	---	-	-	-	♦	0	♦
---	**---**	MUL	0x3D	**D** ← A * B	INH	---	-	-	-	-	-	♦

D8X Name	***D8X OpCo***	***6811 Name***	***6811 OpCo***	***Desc***	***Addr Mode***	***Example***	***H***	***I***	***N***	***Z***	***V***	***C***
rolc	**0x36**	ROL	0x79	**&xx** ← **$xx** << 1 (←C←)	DIR	rolc $xx	-	-	♦	♦	♦	♦
rolc	**0x26**	ROL	0x69	**&(X+O)** ← **$&(X+O)** << 1	IND,X	rolc x + 0xXX	-	-	♦	♦	♦	♦
rolc	**0x26**	ROL	0x69	**&(X+O)** ← **$&(Y+O)** << 1	IND,Y	rolc y + 0xXX	-	-	♦	♦	♦	♦
rolc	**0x16**	ROLA	0x49	**A** ← A << 1 (←C←)	INH	rolc a	-	-	♦	♦	♦	♦
rolc	**0x17**	ROLB	0x59	**B** ← B << 1 (←C←)	INH	rolc b	-	-	♦	♦	♦	♦
rorc	**0x34**	ROR	0x76	**&xx** ← **$xx** >> 1 (→C→)	DIR	rorc $xx	-	-	♦	♦	♦	♦
rorc	**0x24**	ROR	0x66	**&(X+O)** ← **$&(X+O)** >> 1	IND,X	rorc x + 0xXX	-	-	♦	♦	♦	♦
rorc	**0x24**	ROR	0x66	**&(Y+O)** ← **$&(Y+O)** >> 1	IND,Y	rorc y + 0xXX	-	-	♦	♦	♦	♦
rorc	**0x14**	RORA	0x46	**A** ← A >> 1 (→C→)	INH	rorc a	-	-	♦	♦	♦	♦
rorc	**0x15**	RORB	0x56	**B** ← B >> 1 (→C→)	INH	rorc b	-	-	♦	♦	♦	♦
---	---	ASL	0x78	---	EXT	---	-	-	♦	♦	♦	♦
---	---	ASL	0x68	---	IND,X	---	-	-	♦	♦	♦	♦
---	---	ASL	0x68	---	IND,Y	---	-	-	♦	♦	♦	♦
---	---	ASLA	0x48	**A** ← A << 1	INH	---	-	-	♦	♦	♦	♦
---	---	ASLB	0x58	**B** ← B << 1	INH	---	-	-	♦	♦	♦	♦
---	---	ASLD	0x05	**D** ← D << 1	INH	---	-	-	♦	♦	♦	♦
shra	**0x38**	ASR	0x77	**&xx** ← $xx >> 1	EXT	shra $xx	-	-	♦	♦	♦	♦
shra	**0x28**	ASR	0x67	**&(X+O)** ← $**&(X+O)** >> 1	IND,X	shra x + 0xXX	-	-	♦	♦	♦	♦
shra	**0x28**	ASR	0x67	**&(Y+O)** ← $**&(Y+O)** >> 1	IND,Y	shra y + 0xXX	-	-	♦	♦	♦	♦
shra	**0x18**	ASRA	0x47	**A** ← A >> 1	INH	shra a	-	-	♦	♦	♦	♦
shra	**0x19**	ASRB	0x57	**B** ← B >> 1	INH	shra b	-	-	♦	♦	♦	♦
shl	**0x32**	---	---	**&xx** ← $xx << 1	DIR	shl $xx	-	-	♦	♦	♦	♦
---	---	LSL	0x78	**&xxxx** ← $**xxxx** << 1	EXT	---	-	-	♦	♦	♦	♦
---	---	LSL	0x68	**&(X+O)** ← $**&(X+O)** << 1	IND,X	---	-	-	♦	♦	♦	♦
---	---	LSL	0x68	**&(Y+O)** ← $**&(Y+O)** << 1	IND,Y	---	-	-	♦	♦	♦	♦
shl	**0x12**	LSLA	0x48	**A** ← A << 1	INH	shl a	-	-	♦	♦	♦	♦
shl	**0x13**	LSLB	0x58	**B** ← B << 1	INH	shl b	-	-	♦	♦	♦	♦
shl	**0x06**	LSLD	0x05	**D** ← D << 1	INH	shl d	-	-	♦	♦	♦	♦
shl	**0x22**	---	---	**X** ← X << 1	INH	shl x	-	-	♦	♦	♦	♦
shr	**0x30**	---	---	**&xx** ← $xx << 1	DIR	shl $xx	-	-	0	♦	♦	♦
---	---	LSR	0x74	**&xxxx** ← $**xxxx** >> 1	EXT	---	-	-	0	♦	♦	♦
shr	**0x28**	LSR	0x64	**&(X+O)** ← $**&(X+O)** >> 1	IND,X	shr x + 0xXX	-	-	0	♦	♦	♦
shr	**0x28**	LSR	0x64	**&(Y+O)** ← $**&(Y+O)** >> 1	IND,Y	shr y + 0xXX	-	-	0	♦	♦	♦
shr	**0x10**	LSRA	0x44	**A** ← A >> 1	INH	shr a	-	-	0	♦	♦	♦
shr	**0x11**	LSRB	0x54	**B** ← B >> 1	INH	shr b	-	-	0	♦	♦	♦
shr	**0x04**	LSRD	0x04	**D** ← D >> 1	INH	shr d	-	-	0	♦	♦	♦
shr	**0x20**	---	---	**X** ← X >> 1	INH	shr x	-	-	0	♦	♦	♦

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	nop	jsr **&**	xch **ab**	jmp	shr **d**	d**i**	shl **d**	ei	add **ba**	sub **ba**	st x,$	cmp **ba**	add **ax**	add **ay**	add **bx**	add **by**
1x	shr **a**	shr b	shl **a**	shl **b**	rorc **a**	rorc **b**	rolc **a**	rolc **b**	shra a	shra b	ld a,[y]	ld d,[y]	inc **x**	inc **y**	dec **x**	dec **y**
2x	shr **x**	jsr **&+**	shl **x**	jmp &+	rorc&+	clr**v**	rolc &+	set**v**	shra&+	st s,&+	st x,&+	st y,&+	mov **xs**	inc **s**	mov **sx**	dec **s**
3x	shr **$**	jsr **$**	shl $	ld #$	rorc $	tbbs	rolc $	tbbc	shra $	st s,$	st x,$	st y,$	mov **xd**	mov **yd**	mov **dx**	mov **dy**
4x	bra	brn	bgt	ble	bcc	bcs	bne	beq	bvc	bvs	bpz	bmi	bge	blta	bgta	blea
5x	dec **a**	dec **b**	clr **a**	clr **b**	neg **a**	neg **b**	inc **a**	inc **b**	cmpz **a**	cmpz **b**	mov **ba**	mov **ab**	mov Ca	mov aC	adj **a**	nmi
6x	dec **&+**	bsr	clr **&+**	ret	neg **&+**	clr c	inc **&+**	set c	push **d**	xch **xy**	xch a	xch b	push **a**	push **b**	push **x**	push **y**
7x	dec $	tbs	clr $	reti	neg **$**	clrb **$**	inc **$**	setb **$**	pull **d**	cmp #$	xch a,$	xch b,$	pull **a**	pull **b**	pull **x**	pull **y**
8x	addc**a**	mul **a**	st a,[y]	wait	subc **a**	div **d**	ld **d**	add **d**	sub **d**	cmp **d**	st d,[y]	ld **s**	cmp **x**	cmp **y**	ld **x**	ld **y**
9x	addc**a**	mul **a**	st **a**	st **b**	subc **a**	div **d**	ld **d**	add **d**	sub **d**	cmp **d**	st **d**	ld **s**	cmp **x**	cmp **y**	ld **x**	ld **y**
Ax	addc**a**	mul **a**	st **a**	st **b**	subc **a**	div **d**	ld **d**	add **d**	sub **d**	cmp **d**	st **d**	ld **s**	cmp **x**	cmp **y**	ld **x**	ld **y**
Bx	addc**a**	mul **a**	st **a**	st **b**	subc **a**	div **d**	ld **d**	add **d**	sub **d**	cmp **d**	st **d**	ld **s**	cmp **x**	cmp **y**	ld **x**	ld **y**
Cx	add **a**	add **b**	and **a**	and **b**	sub **a**	sub **b**	or **a**	or **b**	xor **a**	xor **b**	ld **a**	ld **b**	cmp **a**	cmp **b**	cmpb **a**	cmpb **b**
Dx	add **a**	add **b**	and **a**	and **b**	sub **a**	sub **b**	or **a**	or **b**	xor **a**	xor **b**	ld **a**	ld **b**	cmp **a**	cmp **b**	cmpb **a**	cmpb **b**
Ex	add **a**	add **b**	and **a**	and **b**	sub **a**	sub **b**	or **a**	or **b**	xor **a**	xor **b**	ld **a**	ld **b**	cmp **a**	cmp **b**	cmpb **a**	cmpb **b**
Fx	add **a**	add **b**	and **a**	and **b**	sub **a**	sub **b**	or **a**	or **b**	xor **a**	xor **b**	ld **a**	ld **b**	cmp **a**	cmp **b**	cmpb **a**	cmpb **b**

Table: Instruction Set OpCode Matrix

ADDR	***bit 7***	***bit 6***	***bit 5***	***bit 4***	***bit 3***	***bit 2***	***bit 1***	***bit 0***	***Name***
$0000									**DDRA**
$0001									**DDRB**
$0002									**WDC**
$0003									**TIMER3**
$0004									**TIMER**
$0005									**TIMERL**
$0006									**SIDR_SODR**
$0007									**SMRC_SIR**
$0008									**CPR0**
$0009									**CPR0L**
$000A									**CPR1**
$000B									**CPR1L**
$000C									**CPR2**
$000D									**CPR2L**
$000E									**CPR3**
$000F									**CPR3l**
$0010									**ASR0P**
$0011									**ASR0PL**
$0012									**ASR0N**
$0013									**ASR0NL**
$0014									**ASR1P**
$0015									**ASR1PL**
$0016									**ASR1N**
$0017									**ASR1NL**
$0018									**ASR2**
$0019									**ASR2L**
$001A									**ASR3**
$001B									**ASR3L**
$001C									
$001D									
$001E									
$001F									**OMODE**

Table: Registers and Control Bit Assignments: $0000-$001F

ADDR	*bit 7*	*bit 6*	*bit 5*	*bit 4*	*bit 3*	*bit 2*	*bit 1*	*bit 0*	*Name*
$0020									**PORTA**
$0021									**PORTAL**
$0022									**PORTB**
$0023									**PBCS**
$0024									**TAIT**
$0025									**LDOUT**
$0026									**DOUT**
$0027									**DOM**
$0028									**PORTC**
$0029									**PORTD_ASRIN**
$002A									**RAMST**
$002B									**SSD**
$002C									**IRQL**
$002D									**IRQLL**
$002E									**IMASK**
$002F									**IMASKL**
$0030									
$0031									
$0032									
$0033									
$0034									
$0035									
$0036									
$0037									
$0038									**CPR4**
$0039									**CPR4L**
$003A									**CPR5**
$003B									**CPR5L**
$003C									**CPR6**
$003D									**CPR6L**
$003E									**CPR7**
$003F									**CPR7L**

Table: Registers and Control Bit Assignments: $0020-$003F