Motorola

6800 MICROPROCESSOR Instruction Set Summary

Mnem.	Op	HINZVC	IEXD#R	~	Description	Notes
ABA	1B	*-****	X	2	Add accumulators	A=A+B
ADCA s	B9	*-****	XXXX	4	Add with Carry	A=A+s+C
ADCB s	F9	*-****	XXXX	4	Add with Carry	B=B+s+C
ADDA s	BB	*-****	XXXX	4	Add	A=A+s
ADDB s	FB	*-****	XXXX	4	Add	B=B+s
ANDA s	B4	--**0-	XXXX	4	Logical AND	A=A&s
ANDB s	F4	--**0-	XXXX	4	Logical AND	B=B&s
ASL d	78	--****	XX	6	Arithmetic Shift Left	d=d*2
ASLA	48	--****	X	2	Arithmetic Shift Left	A=A*2
ASLB	58	--****	X	2	Arithmetic Shift Left	B=B*2
ASR d	77	--****	XX	6	Arithmetic Shift Right	d=d/2
ASRA	47	--****	X	2	Arithmetic Shift Right	A=A/2
ASRB	57	--****	X	2	Arithmetic Shift Right	B=B/2
BCC a	24	------	X	4	Branch if Carry Clear	If C=0
BCS a	25	------	X	4	Branch if Carry Set	If C=1
BEQ a	27	------	X	4	Branch if Equal	If Z=1
BGE a	2C	------	X	4	Branch if Greater or Eq	If NxV=0
BGT a	2E	------	X	4	Branch if Greater Than	If Zv{NxV}=0
BHI a	22	------	X	4	Branch if Higher	If CvZ=0
BITA s	B5	--**0-	XXXX	4	Bit Test	A&s
BITB s	F5	--**0-	XXXX	4	Bit Test	B&s
BLE a	2F	------	X	4	Branch if Less or Equal	If Zv{NxV}=0
BLS a	23	------	X	4	Branch if Lower or Same	If CvZ=1
BLT a	2D	------	X	4	Branch if Less Than	If NxV=1
BMI a	2B	------	X	4	Branch if Minus	If N=1
BNE a	26	------	X	4	Branch if Not Equal	If Z=0
BPL a	2A	------	X	4	Branch if Plus	If N=0
BRA a	20	------	X	4	Branch Always	PC=a
BSR a	8D	------	X	8	Branch to Subroutine	-[S]=PC,PC,a
BVC a	28	------	X	4	Branch if Overflow Clr	If V=0
BVS a	29	------	X	4	Branch if Overflow Set	If V=1

CBA	11	--****	X	2	Compare accumulators	A-B
CLC	0C	-----0	X	2	Clear Carry	C=0
CLI	0E	-0----	X	2	Clear Interrupt Mask	I=0
CLR d	7F	--0100	XX	6	Clear	d=0
CLRA	4F	--0100	X	2	Clear accumulator	A=0
CLRB	5F	--0100	X	2	Clear accumulator	B=0
CLV	0A	----0-	X	2	Clear Overflow	V=0
CMPA s	B1	--****	XXXX	4	Compare	A-s
CMPB s	F1	--****	XXXX	4	Compare	B-s
COM d	63	--**01	XX	7	Complement	d=~d
COMA	43	--**01	X	2	Complement accumulator	A=~A
COMB	53	--**01	X	2	Complement accumulator	B=~B
CPX s	BC	--****	XXX*	5	Compare Index Register	X-s
DAA	19	--****	X	2	Decimal Adjust Acc.	A=BCD format
DEC d	7A	--**?-	XX	6	Decrement	d=d-1
DECA	4A	--**?-	X	2	Decrement accumulator	A=A-1
DECB	5A	--**?-	X	2	Decrement accumulator	B=B-1
DES	34	------	X	4	Decrement Stack Pointer	S=S-1
DEX	09	---*--	X	4	Decrement Index Reg.	X=X-1
EORA s	B8	--**0-	XXXX	4	Logical Exclusive OR	A=Axs
EORB s	F8	--**0-	XXXX	4	Logical Exclusive OR	B=Bxs
INC d	7C	--**?-	XX	6	Increment	d=d+1
INCA	4C	--**?-	X	2	Increment accumulator	A=A+1
INCB	5C	--**?-	X	2	Increment accumulator	B=B+1
INS	31	------	X	4	Increment Stack Pointer	S=S+1
INX	08	---*--	X	4	Increment Index Reg.	X=X+1
JMP d	7E	------	XX	3	Jump	PC=d
JSR d	BD	------	XX	9	Jump to Subroutine	-[S]=PC,PC=d
LDAA s	B6	--**0-	XXXX	4	Load Accumulator	A=s
LDAB s	F6	--**0-	XXXX	4	Load Accumulator	B=s
LDS s	BE	--**0-	XXX*	5	Load Stack Pointer	S=s
LDX s	FE	--**0-	XXX*	5	Load Index Register	X=s
LSR d	74	--0***	XX	6	Logical Shift Right	d=->{0,d,C}
LSRA	44	--0***	X	2	Logical Shift Right	A=->{0,A,C}
LSRB	54	--0***	X	2	Logical Shift Right	B=->{0,B,C}
NEG d	70	--****	XX	6	Negate	d=-d

NEGA	40	--****	X	2	Negate accumulator	A=-A
NEGB	50	--****	X	2	Negate accumulator	B=-B
NOP	01	------	X	2	No Operation	
ORAA s	BA	--**0-	XXXX	4	Logical inclusive OR	A=Avs
ORAB s	FA	--**0-	XXXX	4	Logical inclusive OR	B=Bvs
PSHA	36	------	X	4	Push	-[S]=A
PSHB	37	------	X	4	Push	-[S]=B
PULA	32	------	X	4	Pull	A=[S]+
PULB	33	------	X	4	Pull	B=[S]+

Mnem.	Op	HINZVC	IEXD#R	~	Description	Notes
ROL d	79	--**?*	XX	6	Rotate Left	d={C,d}<-
ROLA	49	--**?*	X	2	Rotate Left accumulator	A={C,A}<-
ROLB	59	--**?*	X	2	Rotate Left accumulator	B={C,B}<-
ROR d	76	--**?*	XX	6	Rotate Right	d=->{C,d}
RORA	46	--**?*	X	2	Rotate Right acc.	A=->{C,A}
RORB	56	--**?*	X	2	Rotate Right acc.	B=->{C,B}
RTI	3B	??????	X	A	Return from Interrupt	{regs}=[S]+
RTS	39	------	X	5	Return from Subroutine	PC=[S]+
SBA	10	--****	X	2	Subtract accumulators	A=A-B
SBCA s	B2	--****	XXXX	4	Subtract with Carry	A=A-s-C
SBCB s	F2	--****	XXXX	4	Subtract with Carry	B=B-s-C
SEC	0D	-----1	X	2	Set Carry	C=1
SEI	0F	-1----	X	2	Set Interrupt Mask	I=1
SEV	0B	----1-	X	2	Set Overflow	V=1
STAA d	B7	--**0-	XXX	5	Store Accumulator	d=A
STAB d	F7	--**0-	XXX	5	Store Accumulator	d=B
STS d	BF	--**0-	XXX	6	Store Stack Pointer	d=S
STX d	FF	--**0-	XXX	6	Store Index Register	d=X
SUBA s	B0	--****	XXXX	4	Subtract	A=A-s
SUBB s	F0	--****	XXXX	4	Subtract	B=B-s
SWI	3F	-1----	X	C	Software Interrupt	-[S]={regs}
TAB	17	--**0-	X	2	Transfer accumulators	B=A
TAP	06	******	X	2	Transfer to CCR	P=A

```
TBA      17   --**0-   X        2   Transfer accumulators     A=B
TPA      07   ------   X        2   Transfer from CCR         A=P
TST  s   7D   --**00    XX       6   Test                      s
TSTA     4D   --**00   X        2   Test accumulator          A
TSTB     5D   --**00   X        2   Test accumulator          B
TSX      30   ------   X        4   Transfer Stack Pointer    X=S
TXS      35   ------   X        4   Transfer Index Register   S=X
WAI      3E   -*----   X        9   Wait for Interrupt        -[S]={regs}
--------+----+--------+--------+---+-------------------------+------------
 CCR        -*01?                   Unaffect/affected/reset/set/unknown
 H          H                       Half carry (Bit 5)
 I           I                      Interrupt mask (Bit 4)
 N            N                     Negative (Bit 3)
 Z             Z                    Zero (Bit 2)
 V              V                   Overflow (Bit 1)
 C               C                  Carry (Bit 0)
-----------------+------+-+-----------------------------------
                   I            Inherent
 nn,E               E           Extended (Op=E, ~s=e)
 nn,X                X          Index (Op=E-10H, ~s=e+1, JSR ~s=e-1)
 n,D                  D         Direct (Op=E-20H, ~s=e-1)
 #n                   #         Immediate (8-bit, Op=E-30H, ~s=e-2)
 #nn                  *         Immediate (16-bit, Op=E-30H, ~s=e-2)
 a                     R        Relative (PC=PC+2+offset)
--------------------------+-----------------------------------
DIRECT                       Direct addressing mode
EXTEND                       Extended addressing mode
FCB      n                   Form Constant Byte
FCC      'string'            Form Constant Characters
FDB      nn                  Form Double Byte
RMB      nn                  Reserve Memory Bytes
--------------------------+-----------------------------------
 A  B                        Accumulators (8-bit)
 P                           Condition Code Register (CCR, 8-bit)
 PC                          Program Counter (16-bit)
 S                           Stack Pointer (16-bit)
```

X	Index Register (16-bit)
a	Relative address (-125 to +129)
d s	Destination/source
n nn	8/16-bit expression (0 to 255/65535)
+ - * /	Add/subtract/multiply/divide
& ~ v x	AND/NOT/inclusive OR/exclusive OR
<- ->	Rotate left/right
[ ]	Indirect addressing
[ ]+ -[ ]	Indirect auto-increment/decrement
{ }	Combination of operands
{regs}	All registers {PC,X,A,B,P}
FFF8H to FFF9H	Hardware interrupt vector
FFFAH to FFFBH	SWI instruction interrupt vector
FFFCH to FFFDH	Non-maskable interrupt vector
FFFEH to FFFFH	Reset vector