MC6802 Instructions

Two clock cycles per machine cycles. 2 MHz lock gives 1 MHz machine cycles, C.
3.58 MHz clock gives 4.56 µs per machine cycle.

Registers
Accumulator A, 8 bits
Accumulator B, 8 bits
Index Register, 16 bits
Stack Pointer, 16 bits
Program counter, 16 bits
Status Register, 8 bite (only 6 used)

		B	C (T, µs with 3.58 MHz osc))	
ABA	1B	1	2 (9 µs)	Add B to A
ADC A	89 ##	2	2 (9 µs)	Add number ## to A, with carry
ADC A	99 ##	2	3 (14 µs)	Add memory (##) to A, with carry
ADC A	A9 ##	2	5 (23 µs)	Add memory (Index Register + ##) to A, with carry
ADC A	B9 HH LL	3	4 (19 µs)	Add memory (HHLL) to A, with carry
ADC B	C9 ##	2	2 (9 µs)	Add number ## to B, with carry
ADC B	D9 ##	2	3 (14 µs)	Add memory (##) to B, with carry
ADC B	E9 ##	2	5 (23 µs)	Add memory (Index Register + ##) to B, with carry
ADC B	F9 HH LL	3	4 (19 µs)	Add memory (HHLL) to B, with carry
ADD A	8B ##	2	2 (9 µs)	Add number ## to A
ADD A	9B ##	2	3 (14 µs)	Add memory (##) to A
ADD A	AB ##	2	5 (23 µs)	Add data ## plus Index Register to A
ADD A	BB HH LL	3	4 (19 µs)	Add memory (HHLL) to A
ADD B	CB ##	2	2 (9 µs)	Add number ## to B
ADD B	DB ##	2	3 (14 µs)	Add memory (##) to B
ADD B	EB ##	2	5 (23 µs)	Add data ## plus Index Register to B
ADD B	FB HH LL	3	4 (19 µs)	Add memory (HHLL) to B
AND A	84 ##	2	2 (9 µs)	Logical AND data ## to A
AND A	94 ##	2	3 (14 µs)	Logical AND memory (##) to A
AND A	A4 ##	2	5 (23 µs)	Logical AND data in mem (Index Reg + ##) to A
AND A	B4 HH LL	3	4 (19 µs)	Logical AND memory (HHLL) to A
AND B	C4 ##	2	2 (9 µs)	Logical AND data ## to B
AND B	D4 ##	2	3 (14 µs)	Logical AND memory (##) to B
AND B	E4 ##	2	5 (23 µs)	Logical AND memory (Index Register + ##) to B
ANDB	F4 HH LL	3	4 (19 µs)	Logical AND memory (HHLL) to B
ASL A	48	1	2 (9 µs)	Arithmetic shift left A
ASL	68 ##	2	7 (32 µs)	Arithmetic shift left memory (Index Register + ##)
ASL	78 HH LL	3	6 (27 µs)	Arithmetic shift left memory (HHLL)

ASL B	58	1	2 (9 µs)	Arithmetic shift left B
ASR	67 ##	2	7 (32 µs)	Arithmetic shift right memory (Index Reg + ##)
ASR	77 HHLL	3	6 (27 µs)	Arithmetic shift right memory (HHLL)
ASR A	47	1	2 (9 µs)	Arithmetic shift right A
ASR B	57	1	2 (9 µs)	Arithmetic shift right B
BRA	20 ##	2	4 (18 µs)	Branch always, relative
BHI	22 ##	2	4 (18 µs)	Branch if Higher, relative (carry + zero = zero)
BLS	23 ##	2	4 (18 µs)	Branch if the Result is lower or the same, relative
BCC	24 ##	2	4 (18 µs)	Branch if Carry is clear, relative
BCS	25 ##	2	4 (18 µs)	Branch if Carry is set, relative
BNE	26 ##	2	4 (18 µs)	Branch if not equal, relative
BE0	27 ##	2	4 (18 µs)	Branch if Result is zero, relative
BVC	28 ##	2	4 (18 µs)	Branch if Overflow is clear, relative
BVS	29 ##	2	4 (18 µs)	Branch if Overflow is set, relative
BPL	2A ##	2	4 (18 µs)	Branch if positive (plus), relative
BMI	2B ##	2	4 (18 µs)	Branch if the Result is Minus (Negative), relative
BGE	2C ##	2	4 (18 µs)	Branch if Result is greater than or equal to zero
BLT	2D ##	2	4 (18 µs)	Branch if the result is less than zero, relative
BLE	2F ##	2	4 (18 µs)	Branch if the Result is less than or equal to zero
BSR	8D ##	2	8 (36 µs)	Branch to subroutine, relative
BIT A	85 ##	2	2 (9 µs)	Bit test data ## to A.
BIT A	95 ##	2	3 (14 µs)	Bit test data from memory (##) with A
BIT A	A5 ##	2	5 (23 µs)	Bit test data from mem (Index Reg + ##) with A
BIT A	B5 HH LL	3	4 (19 µs)	Test bits of A using data from memory (HHLL)
BIT B	C5 ##	2	2 (9 µs)	Bit test data ## to B
BIT B	D5 ##	2	3 (14 µs)	Bit test data from memory (##) with B
BIT B `	E5 ##	2	5 (23 µs)	Bit test data from mem (Index Reg + ##) with B
BIT B	F5 HH LL	3	4 (19 µs)	Test bits of B using data from memory (HHLL)
CBA	11	1	2 (9 µs)	Compare A and B
CLC	0C	1	2 (9 µs)	Clear Carry bit
CLR	6F ##	ecClear memory location Index Register + ##)		
CLR	7F HH LL	3	6 (28 µs)	Clear memory location (HHLL)
CLR A	4F	1	2 (9 µs)	Clear A
CLR B	5F	1	2 (9 µs)	Clear B
CLV	0A	1	2 (9 µs)	Clear Overflow
CMP A	81 ##	2	2 (9 µs)	Compare data ## with A
CMP A	91 ##	2	3 (13 µs)	Compare data in memory (##) with A
CMP A	A1 ##	2	5 (23 µs)	Compare memory (Index Register + ##) with A
CMP A	B1 HH LL	3	4 (19 µs)	Compare memory (HHLL) with A
CMP B	C1 ##	2	2 (9 µs)	Compare data ## with B

CMP B	D1 ##	2	3 (13 µs)	Compare data in memory (##) with B
CMP B	E1 ##	2	5 (23 µs)	Compare memory (Index Register + ##) with B
CMP B	F1 HH LL	3	4 (19 µs)	Compare memory (HHLL) with B
COM	73 HH LL	2	7 (32 µs)	Complement memory (HHLL)
COM	63 ##	3	6 (28 µs)	Complement memory (Index Register + ##)
COM A	43	2	2 (9 µs)	Complement A
COM B	53	2	2 (9 µs)	Complement B
CPX A	8C ##	3	3 (13 µs)	Compare (Index register) with data ##
CPX	9C ##	2	4 (19 µs)	Compare (Index register) with data in memory (##)
CPX	AC ##	2	6 (28 µs)	ind ???
CPX	BC HH LL	3	5 (23 µs)	Compare (Ind Reg plus 1) with memory (HHLL)
DAA	19	1	2 (9 µs)	Decimal Adjust A
DEC	6A ##	2	7 (32 µs)	Decrement memory (Index Register + ##)
DEC	7A HH LL	3	6 (28 µs)	Decrement memory (HHLL)
DEC A	4A	1	2 (9 µs)	Decrement A
DEC B	5A	1	2 (9 µs)	Decrement B
DES	34	1	4 (18 µs)	Decrement Stack Pointer
DEX	09	1	4 (18 µs)	Decrement Index Register
EOR A	88 ##	2	2 (9 µs)	Logic Exclusive-OR data ## to A
EOR A	98 ##	2	3 (13 µs)	Logic Exclusive-OR data in memory (##) to A
EOR A	A8 ##	2	5 (23 µs)	Logic Ex-OR data in mem (Ind Register + ##) to A
EOR A	B8 HH LL	3	4 (19 µs)	Logical Exclusive-OR memory (HHLL) to A
EOR B	C8 ##	2	2 (9 µs)	Logic Exclusive-OR data ## to B
EOR B	D8 ##	2	3 (13 µs)	Logic Exclusive-OR data in memory (##) to B
EOR B	E8 ##	2	5 (23 µs)	Logic Ex-OR data in mem (Ind Register + ##) to B
EOR B	F8 HH LL	3	4 (19 µs)	Logical Exclusive-OR memory (HHLL) to B
INC	6C ##	2	7 (32 µs)	Increment memory (Index Register + ##)
INC	7C HH LL	3	6 (28 µs)	Increment memory (HHLL)
INC A	4C	1	2 (9 µs)	Increment A
INC B	5C	1	2 (9 µs)	Increment B
INS	31	1	4 (18 µs)	Increment Stack Pointer
INX	08	1	4 (18 µs)	Increment Index Register
JMP	6E ##	2	4 (19 µs)	Jump to address (Index Register + ##)
JMP	7E HH LL	3	3 (13 µs)	Jump to address (HHLL)
JSR	AD ##	2	8 (36 µs)	Jump to subroutine (Index Register + ##)
JSR	BD HHLL	3	9 (40 µs)	Jump to subroutine (HHLL)

LDA A	86 ##	2	2 (9 µs)	Load A with data ##
LDA A	96 ##	2	3 (13 µs)	Load A with data in memory (##)
LDA A	A6 ##	2	5 (23 µs)	Load A with data in memory (Index Register + ##)
LDA A	B6 HH LL	3	4 (19 µs)	Load A from memory (HHLL)
LDA B	C6 ##	2	2 (9 µs)	Load B with data ##
LDA B	D6 ##	2	3 (13 µs)	Load B with data in memory (##)
LDA B	E6 ##	2	5 (23 µs)	Load B with data in memory (Index Register + ##)
LDA B	F6 HH LL	3	4 (19 µs)	Load B from memory (HHLL)
LDS	8E ##	3	3 (13 µs)	Load the Stack Pointer with data ##
LDS	9E ##	3	3 (13 µs)	Load the Stack Pointer with data from memory (##)
LDS	AE ## `	2	5 (23 µs)	Load Stack Pointer from mem (Index Register + ##)
LDS	BE HHLL	3	6 (26 µs)	Load Stack Pointer from memory (HHLL)
LDX	CE ##	3	3 (13 µs)	Load Index Register with data ##
LDX	DE ##	3	3 (13 µs)	Load Ind Reg with data from mem (##) and (## + 1)
LDX	EE ##	2	5 (23 µs)	Load Ind Reg with data from mem (Ind Reg + ##)
1)LDX	FE HH LL	3	6 (26 µs)	Load Index Register from memory (HHLL)
LSR A	44	1	2 (9 µs)	Logical shift right register A
LSR B	54	1	2 (9 µs)	Logical shift right register B
LSR	64 ##	2	7 (31 µs)	Logical shift right memory (Index Register + ##)
LSR	74 HH LL	3	6 (26 µs)	Logical shift right memory (HHLL)
NEG A	40	1	2 (9 µs)	Negate A, 2’s complement
NEG B	50	1	2 (9 µs)	Negate B, 2’s complement
NEG	60 ##	2	7 (31 µs)	Neg mem (Ind Reg + ##), 2’s complement
NEG	70 HH LL	3	6 (26 µs)	Negate memory (HHLL), 2’s complement
NOP	01	1	2 (9 µs)	No operation
ORA A	8A ##	2	2 (9 µs)	Logical OR data ## to A
ORA A	9A ##	2	2 (9 µs)	Logical OR data in memory (##) to A
ORA A	AA ##	2	5 (23 µs)	Logical OR data in memory (Ind Reg + ##) to A
ORA A	BA HH LL	3	4 (19 µs)	Logical OR memory (HHLL) with A
ORA B	CA ##	2	2 (9 µs)	Logical OR data ## to B
ORA B	DA ##	2	2 (9 µs)	Logical OR data in memory (##) to B
ORA B	EA ##	2	5 (23 µs)	Logical OR data in mem (Index Register + ##) to B
ORA B	FA HH LL	3	4 (19 µs)	Logical OR memory (HHLL) with B
PSH A	36	1	4 (19 µs)	Push A onto the Stack
PSH B	37	1	4 (19 µs)	Push B onto the Stack
PUL A	32	1	4 (19 µs)	Pull data from the Stack to A
PUL B	33	1	4 (19 µs)	Pull data from the Stack to B

ROL A	49	1	2 (9 µs)	Rotate A left
ROL B	59	1	2 (9 µs)	Rotate B left
ROL	69 ##	2	7 (31 µs)	Rotate memory (Index Register + ##) left
ROL	79 HH LL	3	6 (26 µs)	Rotate memory (HHLL) left
ROR A	46	1	2 (9 µs)	Rotate A right
ROR B	56	1	2 (9 µs)	Rotate B right
ROR	66 ##	2	7 (31 µs)	Rotate memory (Index Register + ##) right
ROR	76 HHLL	3	6 (26 µs)	Rotate memory (HHLL) right
RTI	3B	1	10 (45 µs)	Return from interrupt
RTS	39	1	5 (23 µs)	Return from subroutine
SBA	10	1	2 (9 µs)	Subtract A from B
SBC A	82 ##	2	2 (9 µs)	Subtract immediate data ## from A. with carry
SBC A	92 ##	2	3 (13 µs)	Subtract data in memory (##) from A
SBC A	A2 ##	2	5 (23 µs)	Subtract data in memory (Ind Reg + ##) from A
SBC A	B2 HH LL	3	4 (19 µs)	Subtract memory (HHLL) from A, with carry
SBC B	C2 ##	2	2 (9 µs)	Subtract immediate data ## from B. with carry
SBC B	D2 ##	2	3 (13 µs)	Subtract data in memory (##) from B
SBC B	E2 ##	2	5 (23 µs)	Subtract data in memory (Ind Reg + ##) from B
SBC B	F2 HH LL	3	4 (19 µs)	Subtract memory (HHLL) from B, with carry
SEV	0B	1	2 (9 µs)	Set Overflow bit
SEC	0D	1	2 (9 µs)	Set Carry bit
SEI	0F	1	2 (9 µs)	Set Interrupt Enable
STA A	97 ##	2	4 (19 µs)	Store A in memory (##)
STA A	A7 ##	2	6 (28 µs)	Store A in memory (Index Register + ##)
STA A	B7 HH LL	3	5 (23 µs)	Store A in memory (HHLL)
STS	9F ##	2	5 (23 µs)	Store (Stack Pointer) to A
STS	AF ##	2	7 (32 µs)	Store Stack Pointer to mem (Index Register + ##)
STS	BF HHLL	3	6 (28 µs)	Store Stack Pointer to memory (HHLL)
STA B	D7 ##	2	4 (19 µs)	Store B in memory (##)
STA B	E7 ##	2	6 (28 µs)	Store B in memory (Index Register + ##)
STA B	F7 HH LL	3	5 (23 µs)	Store B in memory (HHLL)
STX	DF ##	2	5 (23 µs)	Store Index Register to (## and ## + !)
STX `	EF ##	2	7 (32 µs)	Store (Index Register) to (## and ## + 1)
STX	FF HHLL	3	6 (28 µs)	Store Index Register to memory (HHLL)
SUB A	80 ##	2	2 (9 µs)	Subtract immediate data ## from A
SUB A	90 ##	2	3 (13 µs)	Subtract data in memory (##) from A
SUB A	A0 ##	2	5 (23 µs)	Subtract memory (Stack Pointer + ##) from A
SUB A	B0 HH LL	3	4 (19 µs)	Subtract memory (HHLL) from A
SUB B	C0 ##	2	2 (9 µs)	Subtract immediate data ## from B

SUB B	D0 ##	2	3 (13 µs)	Subtract data in memory (##) from B
SUB B	E0 ##	2	5 (23 µs)	Subtract memory (Stack Pointer + ##) from B
SUB B	F0 HH LL	3	4 (19 µs)	Subtract memory (HHLL) from B
SWI	3F	1	12 (54 µs)	Software Interrupt
TAB	16	1	2 (9 µs)	Transfer A to B
TBA	17	1	2 (9 µs)	Transfer B to A
TAP	06	1	2 (9 µs)	Transfer A to the Condition Code Register
TPA	07	1	2 (9 µs)	Transfer the Condition Code Register to A
TSX	30	1	4 (19 µs)	Transfer (Stack Pointer minus 1) to Index Register
TXS	35	1	4 (19 µs)	Transfer (Index Register) to Stack Pointer
TST A	4D	1	2 (9 µs)	Test A
TST B	5D	1	2 (9 µs)	Test B
TST	6D ##	2	7 (32 µs)	Test memory location (Index Register + #3)
TST	7D HH LL	3	6 (28 µs)	Test memory location (HHLL)
WAI	3E	1	9 (40 µs)	Wait for Interrupt