Language of the M6800 Microprocessor

Instruction Set

ABA	Add Accumulators
ADC	Add with Carry
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch if Carry Clear
BCS	Branch if Carry Set
BEQ	Branch if Equal to Zero
BGE	Branch if Greater or Equal Zero
BGT	Branch if Greater than Zero
BHI	Branch if Higher
BIT	Bit Test
BLE	Branch if Less or Equal
BLS	Branch if Lower or Same
BLT	Branch if Less than Zero
BMI	Branch if Minus
BNE	Branch if Not Equal to Zero
BPL	Branch if Plus
BRA	Branch Always
BSR	Branch to Subroutine
BVC	Branch if Overflow Clear
BVS	Branch if Overflow Set
CBA	Compare Accumulators
CLC	Clear Carry
CLI	Clear Interrupt Mask
CLR	Clear
CLV	Clear Overflow
CMP	Compare
COM	Complement
CPX	Compare Index Register
DAA	Decimal Adjust
DEC	Decrement
DES	Decrement Stack Pointer
DEX	Decrement Index Register
EOR	Exclusive OR
INC	Increment
INS	Increment Stack Pointer
INX	Increment Index Register
JMP	Jump
JSR	Jump to Subroutine
LDA	Load Accumulator
LDS	Load Stack Pointer
LDX	Load Index Register
LSR	Logical Shift Right
NEG	Negate
NOP	No Operation
ORA	Inclusive OR Accumulator
PSH	Push Data
PUL	Pull Data
ROL	Rotate Left
ROR	Rotate Right
RTI	Return from Interrupt
RST	Return from Subroutine
SBA	Subtract Accumulators
SEC	Subtract with Carry
SEC	Set Carry
SEI	Set Interrupt Mask
SEV	Set Overflow
STA	Store Accumulator
STS	Store Stack Register
STX	Store Index Register
SUB	Subtract
SWI	Software Interrupt
TAB	Transfer Accumulators
TAP	Transfer Accumulators to Condition Code Reg.
TEA	Transfer Accumulators
TPA	Transfer Condition Code Reg. to Accumulator
TST	Test
TSX	Transfer Stack Pointer to Index Register
TXS	Transfer Index Register to Stack Pointer
WAI	Wait for Interrupt

Instruction Execution Time

(in microseconds assuming a 1 MHz clock)

	Dual operand	ACCX	Immediate	Direct	Extended	Index	Implied	Implied
ABA		•	•	•	•	•	2	•
ADC	x	•	2	3	4	5	•	•
ADD	x	•	2	3	4	5	•	•
AND	x	•	2	3	4	5	•	•
ASL		2	•	•	6	7	•	•
ASR		2	•	•	6	7	•	•
BCC		•	•	•	•	•	•	4
BCS		•	•	•	•	•	•	4
BEQ		•	•	•	•	•	•	4
BGE		•	•	•	•	•	•	4
BGT		•	•	•	•	•	•	4
BHI		•	•	•	•	•	•	4
BIT	x	•	2	3	4	5	•	•
BLE		•	•	•	•	•	•	4
BLS		•	•	•	•	•	•	4
BLT		•	•	•	•	•	•	4
BMI		•	•	•	•	•	•	4
BNE		•	•	•	•	•	•	4
BPL		•	•	•	•	•	•	4
BRA		•	•	•	•	•	•	4
BSR		•	•	•	•	•	•	8
BVC		•	•	•	•	•	•	4
BVS		•	•	•	•	•	•	4
CBA		•	•	•	•	•	2	•
CLC		•	•	•	•	•	2	•
CLI		•	•	•	•	•	2	•
CLR		2	•	•	6	7	•	•
CLV		•	•	•	•	•	2	•
CMP	x	•	2	3	4	5	•	•
COM		2	•	•	6	7	•	•
CPX		•	3	4	5	6	•	•
DAA		•	•	•	•	•	2	•
DEC		2	•	•	6	7	•	•
DES		•	•	•	•	•	4	•
DEX		•	•	•	•	•	4	•
EOR	x	•	2	3	4	5	•	•
INC		2	•	•	6	7	•	•
INS		•	•	•	•	•	4	•
INX		•	•	•	•	•	4	•
JMP		•	•	•	3	4	•	•
JSR		•	•	•	9	9	•	•
LDA	x	•	2	3	4		•	•
LDS		•	3	4	5	6	•	•
LDX		•	3	4	5	6	•	•
LSR		2	•	•	6	7	•	•
NEG		2	•	•	6	7	•	•
NOP		•	•	•	•	•	2	•
ORA	x	•	2	3	4	5	•	•
PSH		•	•	•	•	•	4	•
PUL		•	•	•	•	•	4	•
ROL		2	•	•	6	7	•	•
POP		2	•	•	6	7	•	•
RTI		•	•	•	•	•	10	•
RTS		•	•	•	•	•	5	•
SBA		•	•	•	•	•	2	•
SBC	x	•	2	3	4	5	•	•
SEC		•	•	•	•	•	2	•
SEI		•	•	•	•	•	2	•
SEV		•	•	•	•	•	2	•
STA	x	•	•	4	5	6	•	•
STS		•	•	5	6	7	•	•
STX		•	•	5	6	7	•	•
SUB	x	•	2	3	4	5	•	•
SWI		•	•	•	•	•	12	•
TAB		•	•	•	•	•	2	•
TAP		•	•	•	•	•	2	•
TBA		•	•	•	•	•	2	•
TPA		•	•	•	•	•	2	•
TST		2	•	•	6	7	•	•
TSX		•	•	•	•	•	4	•
TXS		•	•	•	•	•	4	•
WAI		•	•	•	•	•	9	•

Instruction Addressing Modes

ACCX (accumulator only) **Addressing**
In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

Immediate Addressing
In immediate addressing, the operand is contained in the second byte of the instruction. No further addressing of memory is required. The MPU addresses this location when it fetches the immediate instruction for execution. These are two/three-byte instructions.

Direct Addressing
In direct addressing, the address of the operand is, contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine; i.e., locations zero through 255. That part of the memory should be used for temporary data storage and intermediate results. In most configurations, it should be a random access memory. These are two-byte instructions.

Extended Addressing
In extended addressing, the value contained in the second byte of the instruction is used as the higher eight-bits of the address of the operand. The third byte of the instruction is used as the lower eight-bits of the address of the operand. This gives one a 16-bit address for the operand. This is an absolute address in memory. There are three-byte instructions.

Indexed Addressing
In indexed addressing, the value contained in the second byte of the instruction is added to the index register lower eight-bits in the MPU. The carry is then added to the higher older eight-bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are two-byte instructions.

Implied Addressing
In the implied addressing mode the instruction gives the address (i.e. stack pointer, index register, etc.). These are one-byte instructions.

Relative Addressing
In relative addressing, the value contained in the second byte of the instruction is added to the program counters lowest eight-bits plus two. The carry or borrow is then added to the high eight-bits. This allows the user to address data within a range of -125 to +129 bytes of the present instruction. These are two byte instructions.