

APPENDIX B RASM05 MACRO ASSEMBLER SYNTAX AND DIRECTIVES

B.1 ASSEMBLY LANGUAGE SYNTAX AND ASSEMBLER DIRECTIVES

This appendix provides information concerning the assembly language syntax and assembler directive for the M6805 HMOS/M146805 CMOS Family. This information is more thoroughly discussed in *Macro Assemblers Reference Manual M68MASR(D2)* for M6800, 6801, 6805, and 6809; Motorola Literature Distribution Center, Phoenix, Az.

M6805 Family assembly language source statements follow the same format as M6800 source statements. See *Macro Assembler Reference Manual M68MASR(D2)* for detailed MC6805 HMOS/M146805 CMOS Family syntax. Highlights of syntax and assembler directives are discussed in the following paragraphs.

B.2 OPERATION FIELD SYNTAX

All instruction mnemonics for the M6805 HMOS/M146805 CMOS Family are three, four, or five characters long. Examples are:

- LDA
- JSR
- INC
- BHCC
- BRSET

If the accumulator or index register is used as the operand of read/modify/write instructions, then the register is appended to the operation field. For example:

- NEGA
- RORX
- INCX
- DECA
- TSTA

B.3 OPERAND FIELD SYNTAX

B.3.1 Inherent

Inherent instructions are the only type which do not include information in the operand field. All information necessary is incorporated in the operation field. Some examples are listed below. Note that an "A" or an "X" is added to the opcode for the register reference inherent instructions.

```
RTS
CLC
INCA
RORA
INCX
RORX
```

B.3.2 Immediate

The immediate value appears in the operand field preceded by a "#". Example:

```
LDA    #30
LDX    #$49
CPX    #$FF
LDA    #ADDR
```

B.3.3 Direct Addressing

The direct address appears in the operand field. If, on any pass through the source program, the assembler finds an unresolved (undefined) forward reference, the longer extended addressing mode is chosen instead of the direct addressing mode even if the address is subsequently found to be on page zero. To ensure direct addressing for direct variables, always define the variable before using it. In read/modify/write instructions all addresses are assumed to be direct since extended addressing is illegal with this mode. Examples:

```
LDA    CAT
STA    $30
CPX    DOG
ROL    $01
```

Where CAT and DOG have addresses <\$100.

B.3.4 Extended Addressing

The extended address appears in the operand field. This mode is only legal when executing register/memory instructions. Examples:

```
LDA    BIG
LDA    $325
STA    COW
```

Where BIG and COW have addresses >\$100.

B.3.5 Indexed — No Offset

The characters comma and X appear in the operand field. For example:

```
LDA    ,X
COM    ,X
STA    ,X
INC    ,X
TST    ,X
```

B.3.6 Indexed — One Byte Offset

The offset appears followed by a comma and "X". The offset must have a value <\$100. Examples:

```
LDA    3, X
LDA    TABLE, X
INC    50, X
```

Where TABLE <\$100.

B.3.7 Indexed — Two Byte Offset

The offset appears followed by a comma and "X". The offset would normally have a value >\$100. Examples:

```
LDA    300, X
LDA    ZOT, X
COM    500, X
```

Where ZOT >\$100.

B.3.8 Bit Set/Clear

The bit set and clear instructions contain the bit number followed by a comma and the address. Examples:

```
BSET 3, CAT
BCLR 4, $30
BCLR 5, DOG
```

Where CAT and DOG are <\$100.

B.3.9 Bit Test and Branch

The bit test and branch instructions contain the bit number, a comma, the address to be tested, a comma, and the location to branch to if the test was successful. Examples:

```
PIG BRSET 3, CAT, DOG
DOG BRCLR 4, CAT, PIG
```

Where CAT <\$100, DOG and PIG are relative addresses similar to those explained in the next paragraph.

B.3.10 Relative Addressing

The operand field contains the label of the address to be loaded into the program counter if the branch is taken. The branch address must be in the range -126 to +129. Examples:

```
BEQ CAT
BNE DOG
BRA PIG
```

B.4 ASSEMBLER DIRECTIVE SUMMARY

The assembler directives are instructions to the assembler rather than instructions which are directly translated into object code. Detailed descriptions are provided in the M68MASR(D2) reference manual.

B.4.1 Assembly Control Directives

END	Program end
FAIL	Programmer generated errors
NAM	Assign program name
ORG	Origin program counter

B.4.6. Listing Control Directives

OPT ABS	Select absolute MDOS-loadable object output
OPT CL	Print conditional assembly directives
OPT NOCL	Don't print conditional assembly directives
OPT CMO	Allow CMOS instructions STOP and WAIT (M146805 CMOS only)
OPT NOCMO	Don't allow CMOS instructions STOP and WAIT (M146805 CMOS only)
OPT CRE	Print cross reference table
OPT G	Print generated lines of FCB, FCC, and FDB directives
OPT NOG	Don't print generated lines of FCB, FCC, and FDB directives
OPT L	Print source listing from this point
OPT NOL	Inhibit printing of source listing from this point
OPT LLE = n	Change line length
OPT LOAD	Select absolute EXORciser-loadable object output
OPT M	Creat object output in memory
OPT MC	Print macro calls
OPT NOMC	Don't print macro calls
OPT MD	Print macro definitions
OPT NOMD	Don't print macro definitions
OPT MEX	Print macro expansions
OPT NOMEX	Don't print macro expansions
OPT O	Create object output file
OPT NOO	Do not create object output file
OPT P = n	Change page length
OPT NOP	Inhibit paging and printing of headings
OPT REL	Select relocatable object output
OPT S	Print symbol table
OPT SE	Print user-supplied sequence numbers
OPT U	Print unassembled code from conditional directives
OPT NOU	Don't print unassembled code from conditional directives
PAGE	Print subsequent statements on top of next page
SPC	Skip lines
TTL	Initialize heading for source listing

APPENDIX C

INSTRUCTION SET

DETAILED DEFINITION

C.1 INTRODUCTION

In the pages that follow this section, the various accumulator and memory operations, together with the respective mnemonic, provides a heading for each of the executable instructions. The STOP and WAIT instructions apply only to the M146805 CMOS Family. The pages are arranged in alphabetical order of the mnemonic. A brief description of the operation is provided along with other applicable pertinent information, including: condition code status, Boolean formula, source forms, usable addressing modes, number of execution cycles (for both HMOS and CMOS), number of bytes required, and the opcode for each usable addressing mode. Paragraph C.2 contains a listing of the various nomenclature (abbreviations and signs) used in the operations.

C.2 NOMENCLATURE

The following nomenclature is used in the executable instructions which follow this paragraph.

(a) Operators:

- () indirection, i.e., (SP) means the value pointed to by SP
- is loaded with (read: "gets")
- boolean AND
- v boolean (inclusive) OR
- ⊕ boolean EXCLUSIVE OR
- ~ boolean NOT
- negation (twos complement)

(b) Registers in the MPU:

- ACCA Accumulator (shown as A in Boolean formula for condition codes and source forms)
- CC Condition Code Register
- X Index Register
- PC Program Counter
- PCH Program Counter High Byte
- PCL Program Counter Low Byte
- SP Stack Pointer

(c) Memory and Addressing:

- M Contents of any memory location (one byte)
- Rel Relative address (i.e., the two's complement number stored in the second byte of machine code in a branch instruction)

(d) Bits in the Condition Code Register:

- C Carry/Borrow, Bit 0
- Z Zero Indicator, Bit 1
- N Negative Indicator, Bit 2
- I Interrupt Mask, Bit 3
- H Half Carry Indicator, Bit 4

(e) Status of Individual Bits BEFORE Execution of an Instruction

- An Bit n of ACCA (n = 7, 6, 5, 4, 3, 2, 1, 0)
- Xn Bit n of X (n = 7, 6, 5, 4, 3, 2, 1, 0)
- Mn Bit n of M (n = 7, 6, 5, 4, 3, 2, 1, 0). In read/modify/write instructions, Mn is used to represent bit n of M, A or X.

(f) Status of Individual Bits AFTER Execution of an Instruction:

- Rn Bit n of the result (n = 7, 6, 5, 4, 3, 2, 1, 0)

(g) Source Forms:

- P Operands with IMMEDIATE, DIRECT, EXTENDED and INDEXED (0, 1, 2 byte offset) addressing modes
- Q Operands with DIRECT, INDEXED (0 and 1 byte offset) addressing modes
- dd Relative operands
- DR Operands with DIRECT addressing mode only.

(h) iff abbreviation for if-and-only-if.

ADC

Add with Carry

ADC

Operation: ACCA – ACCA + M + C

Description: Adds the contents of the C bit to the sum of the contents of ACCA and M, and places the result in ACCA.

Condition

- Codes:**
- H: Set if there was a carry from bit 3; cleared otherwise.
 - I: Not affected.
 - N: Set if the most significant bit of the result is set; cleared otherwise.
 - Z: Set if all bits of the result are cleared; cleared otherwise.
 - C: Set if there was a carry from the most significant bit of the result; cleared otherwise.

Boolean Formulae for Condition Codes:

$$H = A3 \cdot M3 \vee M3 \cdot R3 \vee R3 \cdot A3$$

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = A7 \cdot M7 \vee M7 \cdot \overline{R7} \vee \overline{R7} \cdot A7$$

Source

Form(s): ADC P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A9
Direct	4	3	2	B9
Extended	5	4	3	C9
Indexed 0 Offset	4	3	1	F9
Indexed 1-Byte	5	4	2	E9
Indexed 2-Byte	6	5	3	D9

ADD

Add

ADD

Operation: ACCA – ACCA + M

Description: Adds the contents of ACCA and the contents of M and places the result in ACCA.

Condition

Codes:

- H: Set if there was a carry from bit 3; cleared otherwise.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Set if there was a carry from the most significant bit of the result; cleared otherwise.

Boolean Formulae for Condition Codes:

$$H = A3 \cdot M3 \vee M3 \cdot R3 \vee R3 \cdot A3$$

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = A7 \cdot M7 \vee M7 \cdot \overline{R7} \vee \overline{R7} \cdot A7$$

Source

Form(s): ADD P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	AB
Direct	4	3	2	BB
Extended	5	4	3	CB
Indexed 0 Offset	4	3	1	FB
Indexed 1-Byte	5	4	2	EB
Indexed 2-Byte	6	5	3	DB

AND

Logical AND

AND

Operation: ACCA ← ACCA • M

Description: Performs logical AND between the contents of ACCA and the contents of M and places the result in ACCA. Each bit of ACCA after the operation will be the logical AND result of the corresponding bits of M and of ACCA before the operation.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): AND P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A4
Direct	4	3	2	B4
Extended	5	4	3	C4
Indexed 0 Offset	4	3	1	F4
Indexed 1-Byte	5	4	2	E4
Indexed 2-Byte	6	5	3	D4

ASL

Arithmetic Shift Left

ASL



Description: Shifts all bits of ACCA, X, or M one place to the left. Bit 0 is loaded with a zero. The C bit is loaded from the most significant bit of ACCA, X or M.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Set if, before the operation, the most significant bit of ACCA, X or M was set; cleared otherwise.

Boolean Formulae for Condition Codes:

$N = R_7$
 $Z = \overline{R_7} \cdot \overline{R_6} \cdot \overline{R_5} \cdot \overline{R_4} \cdot \overline{R_3} \cdot \overline{R_2} \cdot \overline{R_1} \cdot \overline{R_0}$
 $C = b_7 \text{ (before operation)}$

Comments: Same opcode as LSL

Source

Form(s): ASL Q, ASLA, ASLX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	48
Index Register	4	3	1	58
Immediate				
Direct	6	5	2	38
Extended				
Indexed 0 Offset	6	5	1	78
Indexed 1-Byte	7	6	2	68
Indexed 2-Byte				

ASR

ASR

Arithmetic Shift Right



Description: Shifts all bits of ACCA, X or M one place to the right. Bit 7 is held constant. Bit 0 is loaded into the C bit.

Condition

- Codes:**
- H: Not affected.
 - I: Not affected.
 - N: Set if the most significant bit of the result is set; cleared otherwise.
 - Z: Set if all bits of the result are cleared; cleared otherwise.
 - C: Set if, before the operation, the least significant bit of ACCA, X or M was set; cleared otherwise.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = b0 \text{ (before operation)}$$

Source

Form(s): ASR Q, ASRA, ASRX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	47
Index Register	4	3	1	57
Immediate				
Direct	6	5	2	37
Extended				
Indexed 0 Offset	6	5	1	77
Indexed 1-Byte	7	6	2	67
Indexed 2-Byte				

BCC

Branch If Carry Clear

BCC

Operation: $PC \leftarrow PC + 0002 + Rel$ iff $C = 0$

Description: Tests the state of the C bit and causes a branch iff C is clear. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: Same opcode as BHS

Source

Form(s): BCC dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	24
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BCLR n

Clear Bit In Memory

BCLR n

Operation: Mn - 0

Description: Clear bit n ($n = 0, 7$) in location M. All other bits in M are unaffected.

Condition

Codes: Not affected.

Source

Form(s): BCLR n, DR

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate				
Direct	7	5	2	$11 + 2 \cdot n$
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BCS

Branch if Carry Set

BCS

Operation: $PC - PC + 0002 + Rel$ iff $C = 1$

Description: Tests the state of the C bit and causes a branch iff C is set. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: Same opcode as BLO

Source

Form(s): BCS dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	25
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BEQ

Branch if Equal

BEQ

Operation: $PC \leftarrow PC + 0002 + Rel$ iff $Z = 1$

Description: Tests the state of the Z bit and causes a branch iff Z is set. Following a compare or subtract instruction BEQ will cause a branch if the arguments were equal. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s): BEQ dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	27
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BHCC

Branch If Half Carry Clear

BHCC

Operation: $PC \leftarrow PC + 0002 + \text{Rel}$ iff H = 0

Description: Tests the state of the H bit and causes a branch iff H is clear. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s): BHCC dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	28
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BHCS

Branch if Half Carry Set

BHCS

Operation: $PC \leftarrow PC + 0002 + \text{Rel}$ iff $H = 1$

Description: Tests the state of the H bit and causes a branch iff H is set. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s): BHCS dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	29
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BHI

Branch If Higher

BHI

Operation: $PC \leftarrow PC + 0002 + \text{Rel}$ iff $(C \vee Z) = 0$
i.e., if $\text{ACCA} > M$ (unsigned binary numbers)

Description: Causes a branch iff both C and Z are zero. If the BHI instruction is executed immediately after execution of either of the CMP or SUB instructions, the branch will occur if and only if the unsigned binary number represented by the minuend (i.e., ACCA) was greater than the unsigned binary number represented by the subtrahend (i.e., M). See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s): BHI dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	22
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BHS

Branch If Higher or Same

BHS

Operation: $PC - PC + 0002 + Rel$ iff $C = 0$

Description: Following an unsigned compare or subtract, BHS will cause a branch iff the register was higher than or the same as the location in memory. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: Same opcode as BCC

Source

Form(s): BHS dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	24
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BIH

Branch if Interrupt Line is High

BIH

Operation: $PC \leftarrow PC + 0002 + Rel$ iff $\overline{INT} = 1$

Description: Tests the state of the external interrupt pin and branches iff it is high. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: In systems not using interrupts, this instruction and BIL can be used to create an extra I/O input bit. This instruction does NOT test the state of the interrupt mask bit nor does it indicate whether an interrupt is pending. All it does is indicate whether the \overline{INT} line is high.

Source

Form(s): BIH dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	2F
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BIL

Branch if Interrupt Line is Low

BIL

Operation: $PC - PC + 0002 + Rel$ iff $\overline{INT} = 0$

Description: Tests the state of the external interrupt pin and branches iff it is low. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: In systems not using interrupts, this instruction and BIH can be used to create an extra I/O input bit. This instruction does NOT test the state of the interrupt mask bit nor does it indicate whether an interrupt is pending. All it does is indicate whether the \overline{INT} line is Low.

Source

Form(s): BIL dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	2E
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BIT

Bit Test Memory with Accumulator

BIT

Operation: ACCA • M

Description: Performs the logical AND comparison of the contents of ACCA and the contents of M and modifies the condition codes accordingly. The contents of ACCA and M are unchanged.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result of the AND is set; cleared otherwise.
- Z: Set if all bits of the result of the AND are cleared; cleared otherwise.
- C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): BIT P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A5
Direct	4	3	2	B5
Extended	5	4	3	C5
Indexed 0 Offset	4	3	1	F5
Indexed 1-Byte	5	4	2	E5
Indexed 2-Byte	6	5	3	D5

BLO

Branch if Lower

BLO

Operation: $PC - PC + 0002 + Rel$ iff $C = 1$

Description: Following a compare, BLO will branch iff the register was lower than the memory location. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: Same opcode as BCS

Source

Form(s): BLO dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	25
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BLS

Branch if Lower or Same

BLS

Operation: $PC \leftarrow PC + 0002 + \text{Rel}$ iff $(C \vee Z) = 1$
i.e., if $ACCA \leq M$ (unsigned binary numbers)

Description: Causes a branch if (C is set) OR (Z is set). If the BLS instruction is executed immediately after execution of either of the instructions CMP or SUB, the branch will occur if and only if the unsigned binary number represented by the minuend (i.e., ACCA) was less than or equal to the unsigned binary number represented by the subtrahend (i.e., M). See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s): BLS dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	23
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BMC

Branch if Interrupt Mask is Clear

BMC

Operation: $PC \leftarrow PC + 0002 + \text{Rel}$ iff $I = 0$

Description: Tests the state of the I bit and causes a branch iff I is clear. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: This instruction does NOT branch on the condition of the external interrupt line. The test is performed only on the interrupt mask bit.

Source

Form(s): BMC dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	2C
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BMI

Branch if Minus

BMI

Operation: $PC \leftarrow PC + 0002 + \text{Rel}$ iff $N = 1$

Description: Tests the state of the N bit and causes a branch iff N is set. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s) BMI dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	2B
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BMS

Branch if Interrupt Mask Bit is Set

BMS

Operation: $PC \leftarrow PC + 0002 + Rel$ iff $I = 1$

Description: Tests the state of the I bit and causes a branch iff I is set. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Comments: This instruction does NOT branch on the condition of the external interrupt line. The test is performed only on the interrupt mask bit.

Source

Form(s): BMS dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	2D
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BNE

Branch if Not Equal

BNE

Operation: $PC \leftarrow PC + 0002 + Rel$ iff $Z = 0$

Description: Tests the state of the Z bit and causes a branch iff Z is clear. Following a compare or subtract instruction BNE will cause a branch if the arguments were different. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s): BNE dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	26
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BPL

Branch If Plus

BPL

Operation: $PC \leftarrow PC + 0002 + Rel$ iff $N = 0$

Description: Tests the state of the N bit and causes a branch iff N is clear. See BRA instruction for further details of the execution of the branch.

Condition

Codes: Not affected.

Source

Form(s): BPL dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	2A
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BRA

Branch Always

BRA

Operation: $PC \leftarrow PC + 0002 + Rel$

Description: Unconditional branch to the address given by the foregoing formula, in which Rel is the relative address stored as a twos complement number in the second byte of machine code corresponding to the branch instruction.

NOTE: The source program specifies the destination of any branch instruction by its absolute address, either as a numerical value or as a symbol or expression which can be evaluated by the assembler. The assembler obtains the relative address Rel from the absolute address and the current value of the program counter.

Condition

Codes: Not affected.

Source

Form(s): BRA dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	20
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BRCLR n Branch if Bit n is Clear BRCLR n

Operation: $PC - PC + 0003 + Rel$ iff bit n of M is zero

Description: Tests bit n ($n = 0, 7$) of location M and branches iff the bit is clear.

Condition

Codes:
 H: Not affected.
 I: Not affected.
 N: Not affected.
 Z: Not affected.
 C: Set if $M_n = 1$; cleared otherwise.

Boolean Formulae for Condition Codes:

$$C = M_n$$

Comments: The C bit is set to the state of the bit tested. Used with an appropriate rotate instruction, this instruction is an easy way to do serial to parallel conversions.

Source

Form(s): BRCLR n, DR, dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	10	5	3	$01 + 2 \cdot n$
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BRN

Branch Never

BRN

Description: Never branches. Branch never is a 2 byte 4 cycle NOP.

Condition

Codes: Not affected.

Comments: BRN is included here to demonstrate the nature of branches on the M6805 HMOS/M146805 CMOS Family. Each branch is matched with an inverse that varies only in the least significant bit of the opcode. BRN is the inverse of BRA. This instruction may have some use during program debugging.

Source

Form(s): BRN dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	4	3	2	21
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BRSET n

Branch if Bit n is Set

BRSET n

Operation: $PC \leftarrow PC + 0003 + Rel$ iff Bit n of M is not zero

Description: Tests bit n ($n = 0, 7$) of location M and branches iff the bit is set.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Not affected.
- Z: Not affected.
- C: Set if $M_n = 1$; cleared otherwise.

Boolean Formulae for Condition Codes:

$$C = M_n$$

Comments: The C bit is set to the state of the bit tested. Used with an appropriate rotate instruction, this instruction is an easy way to provide serial to parallel conversions.

Source

Form(s): BRSET n, DR, dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	10	5	3	2•n
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BSET n

Set Bit in Memory

BSET n

Operation: $M_n \leftarrow 1$

Description: Set bit n ($n = 0, 7$) in location M . All other bits in M are unaffected.

Condition

Codes: Not affected.

Source

Form(s): BSET n, DR

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate				
Direct	7	5	2	$10 + 2 \cdot n$
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

BSR

Branch to Subroutine

BSR

Operation: PC ← PC + 0002
(SP) ← PCL; SP ← SP - 0001
(SP) ← PCH; SP ← SP - 0001
PC ← PC + Rel

Description: The program counter is incremented by 2. The least (low) significant byte of the program counter contents is pushed onto the stack. The stack pointer is then decremented (by one). The most (high) significant byte of the program counter contents is then pushed onto the stack. Unused bits in the program counter high byte are stored as 1s on the stack. The stack pointer is again decremented (by one). A branch then occurs to the location specified by the relative offset. See the BRA instruction for details of the branch execution.

Condition

Codes: Not affected.

Source

Form(s): BSR dd

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative	8	6	2	AD
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

CLC

Clear Carry Bit

CLC

Operation: C bit ← 0

Description: Clears the carry bit in the processor condition code register.

Condition

Codes: H: Not affected.
I: Not affected.
N: Not affected.
Z: Not affected.
C: Cleared.

Boolean Formulae for Condition Codes:

$$C = 0$$

Source

Form(s): CLC

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	98
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

CLI

Clear Interrupt Mask Bit

CLI

Operation: I bit ← 0

Description: Clears the interrupt mask bit in the processor condition code register. This enables the microprocessor to service interrupts. Interrupts that were pending while the I bit was set will now begin to have effect.

Condition

Codes: H: Not affected.
I: Cleared
N: Not affected.
Z: Not affected.
C: Not affected.

Boolean Formulae for Condition Codes:

$$I = 0$$

Source

Form(s): CLI

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	9A
Relative				
Accumulator				
Index Registers				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

CLR

Clear

CLR

Operation: X – 00 or,
ACCA – 00 or,
M – 00

Description: The contents of ACCA, X, or M are replaced with zeroes.

Condition

Codes: H: Not affected.
I: Not affected.
N: Cleared.
Z: Set.
C: Not affected.

Boolean Formulae for Condition Codes:

N = 0
Z = 1

Source

Form(s): CLR Q, CLRA, CLRX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	4F
Index Register	4	3	1	5F
Immediate				
Direct	6	5	2	3F
Extended				
Indexed 0 Offset	6	5	1	7F
Indexed 1-Byte	7	6	2	6F
Indexed 2-Byte				

CMP

Compare Accumulator with Memory

CMP

Operation: ACCA – M

Description: Compares the contents of ACCA and the contents of M and sets the condition codes, which may then be used for controlling the conditional branches. Both operands are unaffected.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result of the subtraction is set; cleared otherwise.
- Z: Set if all bits of the result of the subtraction are cleared; cleared otherwise.
- C: Set if the absolute value of the contents of memory is larger than the absolute value of the accumulator; cleared otherwise.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = \overline{A7} \cdot M7 \vee M7 \cdot R7 \vee R7 \cdot \overline{A7}$$

Source

Form(s): CMP P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A1
Direct	4	3	2	B1
Extended	5	4	3	C1
Indexed 0 Offset	4	3	1	F1
Indexed 1-Byte	5	4	2	E1
Indexed 2-Byte	6	5	3	D1

COM

Complement

COM

Operation: $X - \sim X = \$FF - X$ or,
 $ACCA - \sim ACCA = \$FF - ACCA$ or,
 $M - \sim M = \$FF - M$

Description: Replaces the contents of ACCA, X, or M with the ones complement. Each bit of the operand is replaced with the complement of that bit.

Condition

Codes: H: Not affected.
 I: Not affected.
 N: Set if the most significant bit of the result is set; cleared otherwise.
 Z: Set if all bits of the result are cleared; cleared otherwise.
 C: Set.

Boolean Formulae for Condition Codes:

$N = R7$
 $Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
 $C = 1$

Source

Form(s): COM Q, COMA, COMX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	43
Index Register	4	3	1	53
Immediate				
Direct	6	5	2	33
Extended				
Indexed 0 Offset	6	5	1	73
Indexed 1-Byte	7	6	2	63
Indexed 2-Byte				

CPX

Compare Index Register with Memory

CPX

Operation: X - M

Description: Compares the contents of X to the contents of M and sets the condition codes, which may then be used for controlling the conditional branches. Both operands are unaffected.

Condition

- Codes:**
- H: Not affected.
 - I: Not affected.
 - N: Set if the most significant bit of the result of the subtraction is set; cleared otherwise.
 - Z: Set if all bits of the result of the subtraction are cleared; cleared otherwise.
 - C: Set if the absolute value of the contents of memory is larger than the absolute value of the index register; cleared otherwise.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = \overline{X7} \cdot M7 \vee M7 \cdot R7 \vee R7 \cdot \overline{X7}$$

Source

Form(s): CPX P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A3
Direct	4	3	2	B3
Extended	5	4	3	C3
Indexed 0 Offset	4	3	1	F3
Indexed 1-Byte	5	4	2	E3
Indexed 2-Byte	6	5	3	D3

DEC

Decrement

DEC

Operation: X – X – 01 or,
ACCA – ACCA – 01 or,
M – M – 01

Description: Subtract one from the contents of ACCA, X, or M. The N and Z bits are set or reset according to the result of this operation. The C bit is not affected by this operation.

Condition

Codes: H: Not affected.
I: Not affected.
N: Set if the most significant bit of the result is set; cleared otherwise.
Z: Set if all bits of the result are cleared; cleared otherwise.
C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$
$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): DEC Q, DECA, DECX, (DEX is recognized by the Assembler as DECX)

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	4A
Index Register	4	3	1	5A
Immediate				
Direct	6	5	2	3A
Extended				
Indexed 0 Offset	6	5	1	7A
Indexed 1-Byte	7	6	2	6A
Indexed 2-Byte				

EOR

Exclusive Or Memory with Accumulator

EOR

Operation: $ACCA \leftarrow ACCA \oplus M$

Description: Performs the logical EXCLUSIVE OR between the contents of ACCA and the contents of M, and places the result in ACCA. Each bit of ACCA after the operation will be the logical EXCLUSIVE OR of the corresponding bit of M and ACCA before the operation.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): EOR P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A8
Direct	4	3	2	B8
Extended	5	4	3	C8
Indexed 0 Offset	4	3	1	F8
Indexed 1-Byte	5	4	2	E8
Indexed 2-Byte	6	5	3	D8

INC

Increment

INC

Operation: $X \leftarrow X + 01$ or,
 $ACCA \leftarrow ACCA + 01$ or,
 $M \leftarrow M + 01$

Description: Add one to the contents of ACCA, X, or M. The N and Z bits are set or reset according to the result of this operation. The C bit is not affected by this operation.

Condition

Codes: H: Not affected.
I: Not affected.
N: Set if the most significant bit of the result is set; cleared otherwise.
Z: Set if all bits of the result are cleared; cleared otherwise.
C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$
$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): INC Q, INCA, INCX (INX is recognized by the Assembler as INCX)

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	4C
Index Register	4	3	1	5C
Immediate				
Direct	6	5	2	3C
Extended				
Indexed 0 Offset	6	5	1	7C
Indexed 1-Byte	7	6	2	6C
Indexed 2-Byte				

JMP

Jump

JMP

Operation: PC – effective address

Description: A jump occurs to the instruction stored at the effective address. The effective address is obtained according to the rules for EXTended, DIRect or IN-Dexed addressing.

Condition

Codes: Not affected.

Source

Form(s): JMP P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate				
Direct	3	2	2	BC
Extended	4	3	3	CC
Indexed 0 Offset	3	2	1	FC
Indexed 1-Byte	4	3	2	EC
Indexed 2-Byte	5	4	3	DC

JSR

Jump to Subroutine

JSR

Operation: PC ← PC + N
(SP) ← PCL; SP ← SP - 0001
(SP) ← PCH ; SP ← SP - 0001
PC ← effective address

Description: The program counter is incremented by N (N = 1, 2, or 3 depending on the addressing mode), and is then pushed onto the stack (least significant byte first). Unused bits in the program counter high byte are stored as 1s on the stack. The stack pointer points to the next empty location on the stack. A jump occurs to the instruction stored at the effective address. The effective address is obtained according to the rules for EXTended, DIRect, or INdEXed addressing.

Condition

Codes: Not affected.

Source

Form(s): JSR P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate				
Direct	7	5	2	BD
Extended	8	6	3	CD
Indexed 0 Offset	7	5	1	FD
Indexed 1-Byte	8	6	2	ED
Indexed 2-Byte	9	7	3	DD

LDA

Load Accumulator from Memory

LDA

Operation: ACCA ← M

Description: Loads the contents of memory into the accumulator. The condition codes are set according to the data.

Condition

- Codes:**
- H: Not affected.
 - I: Not affected.
 - N: Set if the most significant bit of the accumulator is set; cleared otherwise.
 - Z: Set if all bits of the accumulator are cleared; cleared otherwise.
 - C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): LDA P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A6
Direct	4	3	2	B6
Extended	5	4	3	C6
Indexed 0 Offset	4	3	1	F6
Indexed 1-Byte	5	4	2	E6
Indexed 2-Byte	6	5	3	D6

LDX

Load Index Register from Memory

LDX

Operation: X ← M

Description: Loads the contents of memory into the index register. The condition codes are set according to the data.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the index register is set; cleared otherwise.
- Z: Set if all bits of the index register are cleared; cleared otherwise.
- C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): LDX P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	AE
Direct	4	3	2	BE
Extended	5	4	3	CE
Indexed 0 Offset	4	3	1	FE
Indexed 1-Byte	5	4	2	EE
Indexed 2-Byte	6	5	3	DE

LSL

LSL

Logical Shift Left



Description: Shifts all bits of the ACCA, X or M one place to the left. Bit 0 is loaded with a zero. The C bit is loaded from the most significant bit of ACCA, X or M.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Set if, before the operation, the most significant bit of ACCA, X or M was set; cleared otherwise.

Boolean Formulae for Condition Codes:

$N = R7$
 $Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
 $C = b7$ (before operation)

Comments: Same as ASL

Source

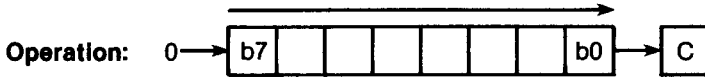
Form(s): LSL Q, LSLA, LSLX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	48
Index Register	4	3	1	58
Immediate				
Direct	6	5	2	38
Extended				
Indexed 0 Offset	6	5	1	78
Indexed 1-Byte	7	6	2	68
Indexed 2-Byte				

LSR

LSR

Logical Shift Right



Description: Shifts all bits of ACCA, X or M one place to the right. Bit 7 is loaded with a zero. Bit 0 is loaded into the C bit.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Cleared.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Set if, before the operation, the least significant bit of ACCA, X or M was set; cleared otherwise.

Boolean Formulae for Condition Codes:

$N = 0$
 $Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
 $C = b0$ (before operation)

Source

Form(s): LSR Q, LSRA, LSRX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	44
Index Register	4	3	1	54
Immediate				
Direct	6	5	2	34
Extended				
Indexed 0 Offset	6	5	1	74
Indexed 1-Byte	7	6	2	64
Indexed 2-Byte				

NEG

Negate

NEG

Operation: $X \leftarrow -X$ (i.e., $00 - X$) or,
 $ACCA \leftarrow -ACCA$ (i.e., $00 - ACCA$) or,
 $M \leftarrow -M$ (i.e., $00 - M$)

Description: Replaces the contents of ACCA, X or M with its twos complement. Note that \$80 is left unchanged.

Condition

Codes: H: Not affected.
I: Not affected.
N: Set if the most significant bit of the result is set; cleared otherwise.
Z: Set if all bits of the result are cleared; cleared otherwise.
C: Set if there would be a borrow in the implied subtraction from zero; the C bit will be set in all cases except when the contents of ACCA, X or M before the NEG is 00.

Boolean Formulae for Condition Codes:

$N = R7$
 $Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
 $C = R7vR6vR5vR4vR3vR2vR1vR0$

Source

Form(s): NEG Q, NEGA, NEGX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	40
Index Register	4	3	1	50
Immediate				
Direct	6	5	2	30
Extended				
Indexed 0 Offset	6	5	1	70
Indexed 1-Byte	7	6	2	60
Indexed 2-Byte				

NOP

No Operation

NOP

Description: This is a single-byte instruction which causes only the program counter to be incremented. No other registers are changed.

Condition

Codes: Not affected.

Source

Form(s): NOP

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	9D
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

ORA

Inclusive OR

ORA

Operation: ACCA – ACCA v M

Description: Performs logical OR between the contents of ACCA and the contents of M and places the result in ACCA. Each bit of ACCA after the operation will be the logical (inclusive) OR result of the corresponding bits of M and ACCA before the operation.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Not affected.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

Source

Form(s): ORA P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	AA
Direct	4	3	2	BA
Extended	5	4	3	CA
Indexed 0 Offset	4	3	1	FA
Indexed 1-Byte	5	4	2	EA
Indexed 2-Byte	6	5	3	DA

ROL

ROL

Rotate Left thru Carry



Description: Shifts all bits of the ACCA, X, or M one place to the left. Bit 0 is loaded from the C bit. The C bit is loaded from the most significant bit of ACCA, X, or M.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Set if, before the operation, the most significant bit of ACCA, X or M was set; cleared otherwise.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = b7 \text{ (before operation)}$$

Source

Form(s):

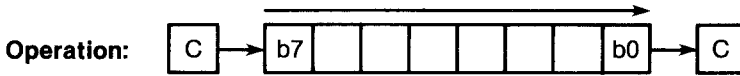
ROL Q, ROLA, ROLX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	49
Index Register	4	3	1	59
Immediate				
Direct	6	5	2	39
Extended				
Indexed 0 Offset	6	5	1	79
Indexed 1-Byte	7	6	2	69
Indexed 2-Byte				

ROR

ROR

Rotate Right Thru Carry



Description: Shifts all bits of ACCA, X, or M one place to the right. Bit 7 is loaded from the C bit. Bit 0 is loaded into the C bit.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Set if, before the operation, the least significant bit of ACCA, X or M was set; cleared otherwise.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = b0 \text{ (before operation)}$$

Source

Form(s):

ROR Q, RORA, RORX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	46
Index Register	4	3	1	56
Immediate				
Direct	6	5	2	36
Extended				
Indexed 0 Offset	6	5	1	76
Indexed 1-Byte	7	6	2	66
Indexed 2-Byte				

RSP

Reset Stack Pointer

RSP

Operation: SP – \$7F

Description: Resets the stack pointer to the top of the stack.

Condition

Codes: Not affected.

Source

Form(s): RSP

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	9C
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

RTI

Return from Interrupt

RTI

Operation: SP ← SP + 0001 ; CC ← (SP)
SP ← SP + 0001 ; ACCA ← (SP)
SP ← SP + 0001 ; X ← (SP)
SP ← SP + 0001 ; PCH ← (SP)
SP ← SP + 0001 ; PCL ← (SP)

Description: The condition codes, accumulator, index register, and the program counter are restored according to the state previously saved on the stack. Note that the interrupt mask bit (I bit) will be reset if and only if the corresponding bit stored on the stack is zero.

Condition

Codes: Set or cleared according to the first byte pulled from the stack.

Source

Form(s): RTI

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	9	9	1	80
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

RTS

Return from Subroutine

RTS

Operation: $SP \leftarrow SP + 0001$; $PCH \leftarrow (SP)$
 $SP \leftarrow SP + 0001$; $PCL \leftarrow (SP)$

Description: The stack pointer is incremented (by one). The contents of the byte of memory, pointed to by the stack pointer, are loaded into the high byte of the program counter. The stack pointer is again incremented (by one). The byte pointed to by the stack pointer is loaded into the low byte of the program counter.

Condition

Codes: Not affected.

Source

Form(s): RTS

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	6	6	1	81
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

SBC

Subtract with Carry

SBC

Operation: ACCA – ACCA – M – C

Description: Subtracts the contents of M and C from the contents of ACCA, and places the result in ACCA.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- C: Set if the absolute value of the contents of memory plus the previous carry is larger than the absolute value of the accumulator; cleared otherwise.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = \overline{A7} \cdot M7 \vee M7 \cdot R7 \vee R7 \cdot A7$$

Source

Form(s): SBC P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A2
Direct	4	3	2	B2
Extended	5	4	3	C2
Indexed 0 Offset	4	3	1	F2
Indexed 1-Byte	5	4	2	E2
Indexed 2-Byte	6	5	3	D2

SEC

Set Carry Bit

SEC

Operation: C bit ← 1

Description: Sets the carry bit in the processor condition code register.

Condition

Codes: H: Not affected.
I: Not affected.
N: Not affected.
Z: Not affected.
C: Set.

Boolean Formulae for Condition Codes:

$$C = 1$$

Source

Form(s): SEC

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	99
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

SEI

Set Interrupt Mask Bit

SEI

Operation: I bit ← 1

Description: Sets the interrupt mask bit in the processor condition code register. The microprocessor is inhibited from servicing interrupts, and will continue with execution of the instructions of the program until the interrupt mask bit is cleared.

Condition

Codes: H: Not affected.
I: Set
N: Not affected.
Z: Not affected.
C: Not affected.

Boolean Formulae for Condition Codes:

I = 1

Source

Form(s): SEI

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	9B
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

STA

Store Accumulator in Memory

STA

Operation: M ← ACCA

Description: Stores the contents of ACCA in memory. The contents of ACCA remain the same.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the accumulator is set; cleared otherwise.
- Z: Set if all bits of the accumulator are clear; cleared otherwise.
- C: Not affected.

Boolean Formulae for Condition Codes:

$$N = A_7$$
$$Z = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$$

Source

Form(s): STA P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate				
Direct	5	4	2	B7
Extended	6	5	3	C7
Indexed 0 Offset	5	4	1	F7
Indexed 1-Byte	6	5	2	E7
Indexed 2-Byte	7	6	3	D7

STOP

Enable \overline{IRQ} , Stop Oscillator

STOP

Description: Reduces power consumption by eliminating all dynamic power dissipation. Results in: (1) timer prescaler to clear, (2) disabling of timer interrupts, (3) timer interrupt flag bit to clear, (4) external interrupt request enabling, and (5) inhibiting of oscillator.

When \overline{RESET} or \overline{IRQ} input goes low: (1) oscillator is enabled, (2) a delay of 1920 instruction cycles allows oscillator to stabilize, (3) the interrupt request vector is fetched, and (4) service routine is executed.

External interrupts are enabled following the RTI command.

Condition

Codes:
H: Not affected.
I: Cleared.
N: Not affected.
Z: Not affected.
C: Not affected.

Source

Form(s): STOP

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	—	2	1	8E
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

STX

Store Index Register in Memory

STX

Operation: M – X

Description: Stores the contents of X in memory. The contents of X remain the same.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the index register is set; cleared otherwise.
- Z: Set if all bits of the index register are clear; cleared otherwise.
- C: Not affected.

Boolean Formulae for Condition Codes:

$$N = X_7$$

$$Z = \overline{X_7} \cdot \overline{X_6} \cdot \overline{X_5} \cdot \overline{X_4} \cdot \overline{X_3} \cdot \overline{X_2} \cdot \overline{X_1} \cdot \overline{X_0}$$

Source

Form(s): STX P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate				
Direct	5	4	2	BF
Extended	6	5	3	CF
Indexed 0 Offset	5	4	1	FF
Indexed 1-Byte	6	5	2	EF
Indexed 2-Byte	7	6	3	DF

SUB

Subtract

SUB

Operation: ACCA – ACCA – M

Description: Subtracts the contents of M from the contents of ACCA and places the result in ACCA.

Condition

Codes:

- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the results are cleared; cleared otherwise.
- C: Set if the absolute value of the contents of memory are larger than the absolute value of the accumulator; cleared otherwise.

Boolean Formulae for Condition Codes:

$$N = R7$$

$$Z = \overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$$

$$C = \overline{A7} \cdot M7vR7 \cdot R7vR7 \cdot \overline{A7}$$

Source

Form(s): SUB P

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator				
Index Register				
Immediate	2	2	2	A0
Direct	4	3	2	B0
Extended	5	4	3	C0
Indexed 0 Offset	4	3	1	F0
Indexed 1-Byte	5	4	2	E0
Indexed 2-Byte	6	5	3	D0

Operation: PC ← PC + 0001
(SP) ← PCL ; SP ← SP - 0001
(SP) ← PCH ; SP ← SP - 0001
(SP) ← X ; SP ← SP - 0001
(SP) ← ACCA ; SP ← SP - 0001
(SP) ← CC ; SP ← SP - 0001
I bit ← 1
PCH ← n - 0003
PCL ← n - 0002

Description: The program counter is incremented (by one). The program counter, index register and accumulator are pushed onto the stack. The condition code register bits are then pushed onto the stack with bits H, I, N, Z, and C going into bit positions 4 through 0 with the top three bits (7, 6 and 5) containing ones. The stack pointer is decremented by one after each byte is stored on the stack.

The interrupt mask bit is then set. The program counter is then loaded with the address stored in the software interrupt vector located at memory locations n - 0002 and n - 0003, where n is the address corresponding to a high state on all lines of the address bus.

Condition

Codes: H: Not affected.
I: Set.
N: Not affected.
Z: Not affected.
C: Not affected.

Boolean Formulae for Condition Codes:

I = 1

Caution: This instruction is used by Motorola in some of its software products and may be unavailable for general use.

Source

Form(s): SWI

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	11	10	1	83
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

TAX

Transfer Accumulator to Index Register

TAX

Operation: X ← ACCA

Description: Loads the index register with the contents of the accumulator. The contents of the accumulator are unchanged.

Condition

Codes: Not affected.

Source

Form(s): TAX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	97
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

TST

Test for Negative or Zero

TST

Operation: X – 00 or,
ACCA – 00 or,
M – 0

Description: Sets the condition codes N and Z according to the contents of ACCA, X, or M.

Condition

Codes:
H: Not affected.
I: Not affected.
N: Set if the most significant bit of the contents of ACCA, X, or M is set; cleared otherwise.
Z: Set if all bits of ACCA, X, or M are clear; cleared otherwise.
C: Not affected.

Boolean Formulae for Condition Codes:

$$N = M7$$
$$Z = \overline{M7} \cdot \overline{M6} \cdot \overline{M5} \cdot \overline{M4} \cdot \overline{M3} \cdot \overline{M2} \cdot \overline{M1} \cdot \overline{M0}$$

Source

Form(s): TST Q, TSTA, TSTX

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent				
Relative				
Accumulator	4	3	1	4D
Index Register	4	3	1	5D
Immediate				
Direct	6	4	2	3D
Extended				
Indexed 0 Offset	6	4	1	7D
Indexed 1-Byte	7	5	2	6D
Indexed 2-Byte				

TXA

Transfer Index Register to Accumulator

TXA

Operation: ACCA ← X

Description: Loads the accumulator with the contents of the index register. The contents of the index register are unchanged.

Condition

Codes: Not affected.

Source

Form(s): TXA

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	2	2	1	9F
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				

WAIT

Enable Interrupt, Stop Processor

WAIT

Description: Reduces power consumption by eliminating dynamic power dissipation in all circuits except the timer and timer prescaler. Causes enabling of external interrupts and stops clocking or processor circuits.

Timer interrupts may be enabled or disabled by programmer prior to execution of WAIT.

When $\overline{\text{RESET}}$ or $\overline{\text{IRQ}}$ input goes low, or timer counter reaches zero with counter interrupt enabled: (1) processor clocks are enabled, and (2) interrupt request, or reset, or timer interrupt vector is fetched.

Interrupts are enabled following the RTI command.

Condition

Codes:
H: Not affected.
I: Cleared.
N: Not affected.
Z: Not affected.
C: Not affected.

Source

Form(s): WAIT

Addressing Mode	Cycles		Bytes	Opcode
	HMOS	CMOS		
Inherent	—	2	1	8F
Relative				
Accumulator				
Index Register				
Immediate				
Direct				
Extended				
Indexed 0 Offset				
Indexed 1-Byte				
Indexed 2-Byte				