

))

CHAPTER 4

THE M68HC12 INSTRUCTION SET

))Q

OBJECTIVES

This chapter describes and gives examples showing how to use the instructions in the M68HC12 instruction set. The instructions are grouped into functional categories. When you learn these categories, you will be able to find a particular instruction to give you the function you need in your program.

- 4.1 Introduction**
- 4.2 M68HC12 Instruction Set Categories**
- 4.3 M68HC12 Instruction and Operand Syntax**
- 4.4 Load and Store Register Instructions**
 - Eight-bit Load and Store Instructions**
 - Sixteen-bit Load and Store Instructions**
 - Stack Instructions**
 - Load Effective Address Instructions**
- 4.5 Transfer Register Instructions**
- 4.6 Move Register Instructions**
- 4.7 Decrement and Increment Instructions**
- 4.8 Clear and Set Instructions**
- 4.9 Shift and Rotate Instructions**
- 4.10 Arithmetic Instructions**

Add and Subtract
Decimal Arithmetic
Negating and Sign Extension Instructions
Multiplication
Fractional Number Arithmetic
Division

4.11 Logic Instructions

4.12 Data Test Instructions

4.13 Conditional Branch Instructions **Signed and Unsigned Conditional Branches**

4.14 Loop Primitive Instructions

4.15 Unconditional Jump and Branch Instructions **Branches to Subroutines**

4.16 Condition Code Register Instructions

4.17 Interrupt Instructions

4.18 Fuzzy Logic Instructions

4.19 Miscellaneous Instructions

4.20 Chapter Summary Points

- ! Learning the instruction set is easier if you first learn the categories of instructions available.
- ! Table 4-1 is useful for quickly finding an operation in the category you want.
- ! After finding the operation, you must specify the operand and its addressing mode.
- ! Load and store instructions modify the condition code register.
- ! Sixteen-bit load and store instructions require two memory locations for the data.
- ! Push, pull and other stack operations must be balanced.
- ! `BSET` and `BCLR` instructions set and clear one or more bits in an operand.
- ! `LEAX` and `LEAY` instructions are useful for calculating effective addresses.
- ! The `DAA` instruction produces the correct result when adding packed BCD numbers.
- ! `MUL`, `FDIV`, `IDIV`, `EMUL` and `EDIV` can be used only with unsigned numbers.
- ! `IDIVS`, `EMULS` and `EDIVS` can be used with signed (two's-complement) numbers.
- ! Logical instructions perform a bit-wise logical operation between two operands.
- ! Data test instructions set the condition code register bits used in conditional branching.
- ! A compare instruction subtracts one operand from another but does not change either.
- ! When comparing signed numbers, the conditional branch instructions with the words "greater" and "less" are to be used.

- ! When comparing unsigned numbers, the conditional branch instructions with the words "higher" and "lower" are to be used.
- ! The branch instructions all use relative addressing.
- ! Always use the `JSR`, `BSR` or `CALL` instruction to transfer to a subroutine.
- ! Always use the `RTS` or `RTC` instructions to transfer back to the calling program.
- ! Condition code register bits may be used to transfer binary information between parts of a program.

4.21 Bibliography and Further Reading

There are two Motorola references that will help with your programming. The *M68HC12xxx Programming Reference Guide* is a small, quick reference guide with all instructions and their symbolic operations, addressing modes, hexadecimal machine codes, the number of bytes, the number of clock cycles and the effect on the condition code register. The *68HC12 CPU12 Reference Manual* contains details for programming all I/O in the M68HC12 and a complete listing of the instruction set.