

))

## CHAPTER 3

### AN ASSEMBLER PROGRAM

))

#### OBJECTIVES

This chapter discusses the operation of a typical assembler.<sup>1</sup> We will learn the assembler syntax now to be able to more easily understand examples showing the instruction set in the next chapter.

#### **3.1**    Assembly Language Example

#### **3.2**    M68HC12 CASM12 Assembler

#### **3.3**    Assembler Source Code Fields

Label Field

Opcode or Operation Field

Operand Field

Comment Field

#### **3.4**    Assembler Pseudo-operations and Directives

Assembler Pseudo-operations

Code location

Defining symbols

Reserving memory locations

Defining constants in memory

---

<sup>1</sup>    The assembler described in this chapter is CASM12, a component of an integrated assembly language development system including an editor, cross-assembler and communications environment. CASM12 is provided by Motorola with their evaluation boards (EVBs). IASM12 and other products for the M68HC12 can be purchased from P&E Microcomputer Systems <<http://www.pemicro.com/>>.

**Assembler Directives****Source control****Conditional assembly****Listing control****Macros****Macro parameters****Labels in macros****Macros and subroutines****3.5 Assembler Output Files****3.6 Assembler Invocation****3.7 Assembler Errors****3.8 Chapter Summary Points**

- ! CASM12 is an absolute assembler.
- ! There must be a white space between each of the four fields in a source code line.
- ! Labels are not case sensitive; ABCD is the same as AbCd.
- ! You must not have duplication of labels.
- ! Labels may optionally end with a colon (:).
- ! The opcode field contains operation mnemonics, assembler pseudo-operations or directives, or macro names.
- ! The opcodes, pseudo-operations and directives are not case sensitive.
- ! The operand field may have symbols, constants or expressions.
- ! The default base for constants is hexadecimal.
- ! Decimal constants are signified by a !.
- ! Hexadecimal constants are signified by a \$.
- ! Binary constants are signified by a %.
- ! ASCII constants are signified by enclosing the character in single ( ' ') or double ( " ") quotes.
- ! Assembler pseudo-operations and directives allow you to direct the assembler how to do its job.
- ! Program code and other constant data should be located in ROM memory using the ORG pseudo-operation.
- ! Data variables should be located in RAM memory using the ORG pseudo-operation.
- ! Memory space for data variables should be allocated using the DS or RMB pseudo-operation.
- ! An EQU directive allows us to define symbols and constants.
- ! Byte constants to be in ROM memory may be defined using the DB.
- ! ASCII string constants in ROM memory may be defined using the DB or FCB pseudo-operation.
- ! Sixteen-bit constants in ROM memory are defined using the DW or FDB pseudo-operation.
- ! The assembler may produce a listing with a symbol table and object files.
- ! A current listing file should be printed to help with debugging.