

## Terms, Literals, and Expressions

- The number of the statement that defines the symbol. A symbol is defined when it appears in the name entry of a statement.
- The number of all the statements in which the symbol is used as an operand.

**Symbol Table:** When the assembler processes your source statements for the first time, it assigns an absolute or relocatable value to every symbol that appears in the name field of an instruction. The assembler enters this value, which normally reflects the setting of the location counter, into the symbol table. It also enters the attributes associated with the data represented by the symbol. The values of the symbol and its attributes are available later when the assembler finds this symbol or attribute reference used as a term in an operand or expression. See “Symbol Length Attribute Reference” and “Self-Defining Terms” in this chapter for more details. The three types of symbols recognized by the assembler are:

- Ordinary symbols
- Variable symbols
- Sequence symbols

**Ordinary Symbols:** Ordinary symbols can be used in the name and operand fields of machine and assembler instruction statements. Code them to conform to these rules:

- The symbol must not consist of more than 63 alphanumeric characters. The first character must be an alphabetic character. An *alphabetic character* is a letter from A through Z, or from a through z, or \$, \_, #, or @. The other characters in the symbol may be alphabetic characters, digits, or a combination of the two.
  - The assembler makes no distinction between upper-case and lower-case letters used in symbols.
  - If the GOFF option is not specified, external symbols may not consist of more than 8 characters.
  - No other special characters may be included in an ordinary symbol.
  - No spaces are allowed in an ordinary symbol.
  - No double-byte data is allowed in an ordinary symbol.

In the following sections, the term *symbol* refers to the ordinary symbol.

The following examples are valid ordinary symbols:

```
ORDSYM#435A      HERE          $OPEN
K4               #0123         X
B49467LITTLENAIL @33          _TOTAL_SAVED
```

**Variable Symbols:** Variable symbols must begin with an & followed by an alphabetic character and, optionally, up to 61 alphanumeric characters. Variable symbols can be used in macro processing and conditional assembly instructions, and to provide substitution in machine and assembler instructions. They allow different values to be assigned to one symbol. A complete discussion of variable symbols appears in Chapter 7, “How to Specify Macro Definitions” on page 243.

The following examples are valid variable symbols: