

Motorola Directives and Constants

Two types of constants

- Run-time
- Assembly-Time

Run-Time Constants (occupies storage area in user's programs)

DC Directive (generates code)

[symbol] dc.e exp[,exp]

e: represents size and the alignment and may be

- b (byte) - no added data alignment occurs
- w (word) - word boundary alignment
- l (longword) - word boundary alignment

Examples:

	Code		
0003	16	dc.b 22	
0004	0016	dc.w 22	
0006	00000016	dc.l 22	
000A	22	dc.b \$22	
000C	0022	dc.w \$22	; aligned at word boundary
000E	00000022	dc.l \$22	
0012	414243	dc.b 'ABC'	; each character converted to byte
0016	4243	dc.w 'ABC'	; aligned at word boundary
			; A is truncated
0018	00004142	dc.l 'AB'	
001C	48656C6C6F00	dc.b 'Hello', 0	
0022			

DS Directive

- does not generate code, only allocates storage
- sometimes used to perform alignment without allocating storage
 - o example: ds.w 0

Examples:

0004	ds.b 3
0007	ds.w 0
0008	ds.b 1
000A	ds.w 2
000E	

Assembly-Time Constants (no storage is allocated)

EQU, * directives

EQU (equivalence directive)

- to define the value of a symbol

```
example      symbol equ exp
              setit equ  $1000
```

Assembler every occurrence of *setit* to its equivalent value \$1000

- no storage allocated
- value cannot be changed
- declaration should precede before referencing it

*** directive**

- to extract the current contents of location counter (LC)

```
$001000      setit equ *
```

Then if current LC = \$001000

Then setit = \$001000

```
string dc.b 'coen 311'
length equ 8
```

OR

```
string dc.b 'coen 311'
length equ * - string
```

ORG and END directives

```
[symbol] org exp
```

Every module should end with directive *end*