Assembler Directives

An assembler directive line looks just like an instruction line in an assembly language program.

A directive tells the assembler to do something other than create the machine code for an instruction.

The available assembler directives vary from one assembler to another even for the same processor.

Assembly programmer should refer to the user’s manual of the specific assembler for details.

Assembler Directives

Defining Constants (1/3)

db	**(define byte)**
dc.b	**(define constant byte)**
fcb	**(form constant byte)**

Three equivalent directives to define the value of a byte (or a number of bytes) in memory.

```
Ex:         space  db     $20
```

Initializes one byte in memory to the value $20.

```
Ex:         array    dc.b  $11,$22,$33,$44
```

Initializes four consecutive bytes in memory to:

```
            $11
            $22
            $33
            $44
```

The address of the byte with the value $33 will be array+2.

Assembler Directives

Defining Constants (2/3)

dw	**(define word)**
dc.w	**(define constant word)**
fcw	**(form constant word)**

Three equivalent directives to define the value of a word (or a number of words) in memory.

Ex:

```
vec_tab  dw     $1234,$5678
```

Initializes two words in memory to the values:
$1234
$5678

The address of the word with the value $1234 will be the current location counter.
The address of the word with the value $5678 will be the current location counter + 2.

Assembler Directives

Defining Constants (3/3)

fcc (form constant character)

Used to define a string of characters (a message) in memory.
Similar to form constant byte **fcb** but the characters are encoded in ASCII.
The string of characters should be enclosed between delimiters, usually quotations.

```
Ex:        alpha  fcc     “EF”
           generates the following values in memory:
           $45
           $46
while
            alpha  fcb     $EF
generates one byte with the value $EF in memory.
```

fill (fill memory)

Used to fill a number of memory locations with a given value.

```
Ex:        SpaceLine     fill     $20,40
fills 40 bytes in memory with the value $20 starting from the current location counter.
```

Assembler Directives

Defining Storage

ds (define sorage)
ds.b (define storage byte)
rmb (reserve memory byte)

Three equivalent directives to reserve a number of bytes in RAM.

Ex: buffer ds 100

reserves one hundred bytes in memory without initializing their values. When reading this line, the assembler assigns the value of the location counter to the label buffer, and increments the location counter by 100.

ds.w (define storage word)
rmw (reserve memory word)

Ex: buff ds.w 20

reserves fourty consecutive bytes in memory. The assembler assigns the value of the location counter to the label buff, and increments the location counter by 40. The address of the first word will be buff, and the address of the last word will be buff + 19x2.

Assembler Directives

org   xxxx

Forces the location counter used by the assembler to the value   xxxx.
Used by the programmer to place segments of data and segments to instructions at certain locations in memory.

```
Ex:             org   $1000
      array     db     $11,$12,$13,$14
```

initializes the contents of memory locations $1000-$1003   to:

$1000	$11
$1001	$12
$1002	$13
$1003	$14

and assigns the value $1000 to the label “array”.

```
Ex:             org    $8000
                lds     $0800
```

places the instruction lds $0800   in location $8000 in memory.

Assembler Directives

equu

assigns a value to a label. (makes the program more readable)

```
Ex:    loop_cnt    equ   50
       :
       :
       ldaa   #loop_cnt
```

Whenever the assembler encounters the symbol loop_cnt in the program, it replaces it by the value of 50.

end

Should be placed at the end of the assembly program to tell the assembler to stop reading more lines.