Assembler Directives

As mentioned earlier, the Motorola DSP56300 Assembler converts your ASCII text file into an object file that the DSP56302 can execute. Assembly programs can contain *assembler directives* which specify auxiliary actions to be performed by the assembler. These directives are different from the DSP56300 instruction set and can greatly aid in the development process. The assembler directives are defined in Assembler Reference Manual. A few useful assembler directives and examples are given here.

Equate (EQU)

```
<label>     EQU     [{X: | Y: | L: | P: | E:}]<expression>
```

The EQU directive assigns the value and memory space attribute of <expression> to the symbol <label>. The label cannot be redefined anywhere else in the program.

Example:

```
N     equ     16
```

Originate (ORG)

```
     ORG     <rms>[<rlc>][<rmp>]:[<expr1>][,<lms>[<llc>][<lmp>]:[<exp2>]]
     ORG     <rms>[<rmp>][(<rce>)]:[<expr1>][,<lms>[<lmp>][(<lce>)]:[<exp2>]]
```

The ORG directive is used to specify addresses and to indicate memory space and mapping changes. Two of the parameters used with the ORG directive are:

<rms>---which memory space (X, Y, L, P, or E) will be used as the runtime memory space. If the memory space is L, any allocated datum with a value greater than the target word size will be extended to two words; otherwise it is truncated. If the memory space is E, then depending on the memory space qualifier, any generated words will be split into bytes, one byte per word, or a 16/8-bit combination.

<exp1>---initial value assigned to the runtime counter used as the <rlc>. If <exp1> is an absolute expression the assembler uses the absolute location counter. If <exp1> is not specified, then the last value and mode that the counter had will be used.

Example:

```
ORG     x:$00000a     ;lay things out in x-memory starting at $000a
```

Define Storage (DS)

```
<label>     DS     <expression>
```

The DS directive reserves a block of memory the length of which in words is equal to the value of <expression>. This directive causes the runtime location counter to be advanced by the value of the absolute integer expression in the operand field. The block of memory reserved is not initialized to any value.

<label>, if present, will be assigned the value of the runtime location counter at the start of the directive processing.

Example:

```
ARRAY     ds     12     ;12 words of memory allocated for array
```

Define Constant (DC)

```
<label>     DC     <arg>[,<arg>,...,<arg>]
```


The DC directive allocates and initializes a word of memory for each argument, <arg>. Arguments may be a numeric constant, a single or multiple character string constant, a symbol, and/or an expression. The DC directive may have one or more arguments separated by commas. Multiple arguments are stored in successive address locations.

<label>, if present, will be assigned the value of the runtime location counter at the start of the directive processing. Integer arguments are stored "as is"; floating point numbers are converted to binary values.

Example:
```
COEFS     dc     0.1,0.2,0.3,0.4,0.5     ;coefficients stored in memory
```

Define Storage Modulo (DSM)
```
<label>     DSM     <expression>
```

The DSM directive reserves a block of memory the length of which in words is equal to the value in <expression>. If the runtime location counter is not zero, this directive first advances the runtime location counter to a base address that is a multiple of 2^k, where 2^k ≥ <expression>. Next the runtime location counter is advanced by the value of the integer expression in the operand field. <expression> can have any memory space attribute. The block of memory reserved is not initialized to any given value.

<label>, if present, will be assigned the value of the runtime location counter at the start of the directive processing.

Example:
```
CIRCULAR_BUFFER     dsm     16     ;16 words allocated for circ. buffer
```

Another Example
In programming FIR filters, we will need to allocate modulo storage for input samples and allocate memory for filter coefficients. While the latter does not have to be modulo, doing so will simplify pointer reset to the first coefficient. The following assembler directives will prepare memory for the FIR filter.

```
N equ 5                      ;filter length
 ORG x:$000000               ;set RLC to x:$000000
input dsm N                  ;allocate modulo storage for samples
 ORG y:$000000               ;set RLC to y:$000000
coefs dsm N                  ;allocate modulo storage for coefficients
 ORG y:coefs                 ;reset RLC to beginning of coefficient storage
 dc     0.2,0.2,0.2,0.2,0.2  ;initialize coefficient memory
```

We note the DSM directive allocating memory for coefs advances the runtime location counter by N. Therefore we must reset the location counter back to y:coefs and then begin laying down the coefficients in memory. The DC directive will also advance the location counter by as many coefficients as we put in memory. If the number of coefficients we lay down is less than or equal to N, the runtime location counter will be set properly for the next memory allocation directive. Otherwise we will overwrite our coefficients.

Addressing Modes (cont)

Motorola Training Notes 2-21 (Modulo Addressing Exercise)

