Empleo del calculador integrado a MPASM

La expresión:

movlw (1 << GIE) | (1 << T0IE)

Carga “W” con un byte, posiblemente destinado al registro INTCON, el cual tiene a los bits GIE (7) y T0IF (5) puestos a “1”. En este caso el calculador se emplea para construir el valor a cargar en “W”. Debe conocerse que el recorrido de un bit por un cierto número de veces, convierte al número de bit en un valor constante que pone a “1” ese bit en particular.

To reset specific bits, the same trick can be used, but the bits have to be reset, which is done by a bitwise inversion of the bits and then ANDing the result with the current value. XORing the set bit value with 0x0FF accomplishes the bitwise inversion. For example, to clear bits 4, 2, and 1 in the w register, the following instruction could be used:

andlw 0x0FF ^ ((1 << 4) | (1 << 2) | (1 << 1))

If you were to do this manually, you would have to follow these steps:

**1** Calculate the values for bits 4, 2, and 1 being set:

(1 << 4) = 16

(1 << 2) = 4

(1 << 1) = 2

which translates to

(1 << 4) | (1 << 2) | (1 << 1) = 16 | 4 |2

= 22

= 0x016

**2** Calculate the inverse (XOR with 0x0FF):

0x0FF ^ 0x016 = 0x0E9

**3** Put the value into the andlw instruction:

andlw 0x0E9

También se puede usar:

andlw ~((1 << 4) | (1 << 2) | (1 << 1))

Para los mismos fines.

For 16-bit values, you can use the “low” and “high” assembler directives. For

example, if you wanted to jump to a specific address in another page, you could use

the code:

movlw HIGH Label ; ”Label“ is the

movwf PCLATH ; Destination

movlw LOW Label

movwf PCL

which is the same as:

movlw ((label && 0x0FF00) >> 8)

movwf PCLATH

movlw LABEL && 0x0FF

movwf PCL

In this example, the function of the first four instructions (which use HIGH and LOW) is much clearer to somebody reading the code than the second four instructions, which require the reader to evaluate what the arithmetic operations are doing.

Para comentar selectivamente partes del código se usan las directivas **ifdef**, **ifndef**, **endif**.

Por ejemplo:

Ifdef Prueba ; 🡪 INICIAN PRUEBAS

. ; Zona de Pruebas, añade rastreo o monitoreo

. ; Uso de Puertos, canales de comunicación, ..., etc.

. ;🡨 CONCLUYEN PRUEBAS

Endif

For example, tests against addresses could be performed for interpage jumping in midrange

PIC MCUs:

if ((($ & 0x01800) ^ (Label & 0x01800)) != 0)

movlw HIGH Label ; Different Pages - Update PCLATH

movwf PCLATH

endif

goto Label & 0x07FF ; Jump to Label

In this example, if the destination is in a different page from the current location (which

is returned by the $ directive in MPLAB), then PCLATH is updated before the goto

statement.

Macro para calcular la dirección de la etiqueta hacia donde se salta:

lgoto Macro Label

if (((($ + 2) & 0x01000) ^ (Label & 0x01000)) != 0)

if ((($ + 2) & 0x01000) == 0)

bsf PCLATH, 5 ; Label in Pages 2 or 3

else

bcf PCLATH, 5 ; Label in Pages 0 or 1

endif

else

nop ; No Difference in High Pages

endif

if (((($ + 2) & 0x00800) ^ (Label & 0x00800)) != 0)

if ((($ + 2) & 0x00800) == 0)

bsf PCLATH, 4 ; Label in Pages 1 or 3

else

bcf PCLATH, 4 ; Label in Pages 0 or 2

endif

else

nop ; No Difference in Low Pages

endif

goto (Label & 0x07FF) | ($ & 0x01800); Jump to Label

endm