

SOLUCIÓ PARCIAL 2 DE CI. QUATRIMESTRE DE PRIMAVERA, 2012

1. (1 Punt) En una rutina d'interrupció d'alta prioritat, per error, ens trobem el següent codi:

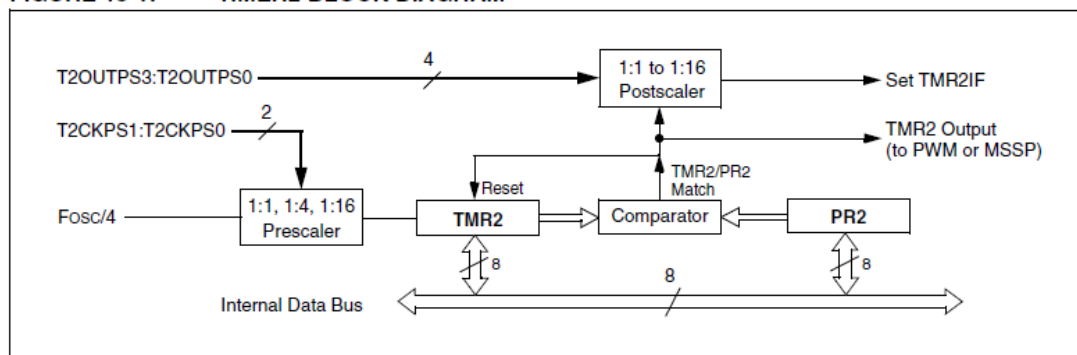
```
MOVWF    W_TEMP                ; W_TEMP is in virtual bank
MOVFF    STATUS, STATUS_TEMP    ; STATUS_TEMP located anywhere
MOVFF    BSR, BSR_TEMP          ; BSR_TEMP located anywhere
;
; USER ISR CODE
;
MOVFF    BSR_TEMP, BSR          ; Restore BSR
MOVF     W_TEMP, W              ; Restore WREG
MOVFF    STATUS_TEMP, STATUS    ; Restore STATUS
```

De quina manera podria afectar aquest codi a una rutina de baixa prioritat que conté les mateixes instruccions per guardar i recuperar el context?

Si en mig de l'execució de la rutina de baixa prioritat salta una interrupció d'alta prioritat, el context quedarà modificat. Llavors, al retornar de la rutina de baixa prioritat, el context que es recuperarà serà el context del moment de la interrupció de alta prioritat.

2. (1 Punt) Amb una freqüència del sistema de 8MHz, calcular la freqüència màxima i mínima del senyal TMR2IF que es pot generar amb el timer 2, si el valor del registre PR2 és de 100.

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



Freqüència màxima (amb prescaler = 1 i postscaler = 1):

$$F_{\text{TMR2IF}} = \text{FOSC} / (4 * \text{PRESCALER} * \text{POSTSCALER} * 100) = 20\text{KHz}$$

Freqüència màxima (amb prescaler = 16 i postscaler = 16):

$$F_{\text{TMR2IF}} = \text{FOSC} / (4 * \text{PRESCALER} * \text{POSTSCALER} * 100) = 78,125 \text{ Hz}$$

3. (3 Punts) Es vol saber el temps existent entre dues interrupcions (INT0) mitjançant la consulta directa del TMR0 configurat en mode de 16 bits, amb un rellotge del sistema de 1MHz i un prescaler de 1.

- a) Quin és el temps mínim i màxim que es pot arribar a mesurar?
- b) Indiqueu breuement el codi de la funció d'interrupció que realitzaria la mesura entre dues interrupcions INT0, sense considerar possibles overflows i underflows del contador TMR0.
- c) Milloreu el codi de l'apartat b) per introduir la detecció d'overflows i underflows del contador TMR0.

- a) La latència d'una subrutina oscil·la entre 3 o 4 cicles que sumada al temps d'execució de la subrutina de servei a la interrupció fins a baixar el flag podrien ser uns 10 cicles. aproximadament. Llavors els intervals de temps inferiors a $10 * T_{\text{TMR0}} = 10 * 4 / \text{FOSC} = 40 \mu\text{seg}$ no serien mesurables. El temps màxim (sense utilitzar altres comptadors) seria de $65536 * T_{\text{TMR0}} = 262 \text{ mseg}$.

b)

```
Void ISR_INT0()
{
if (INT0IF) { // consultem el motiu de la interrupció
INT0IF = 0; // baixem immediatament el flag
DT = TMR0L; // lectura del timer0 (lectura en aquest ordre)
DT = ((uint) TMR0H >> 8) | DT; // DT variable global uint on es deixa el resultat
TMR0H = 0; TMR0L = 0; // preparem el timer per a la següent interrupció (escriptura en aquest ordre)
}
... // la resta del codi
}
```

c) Si s'ha produït un desbordament del TMRO el flag TMROIF estarà a 1.

```
If (TMROIF) {error = OVERFLOW; DT = -1; return}
```

Si la lectura del comptador és molt baixa el resultat no és fiable...

```
If (DT < 10) {error = UNDERFLOW; DT = -1; return}
```

4. (1 Punt) Com podríem aprofitar el hardware del PIC per fer una funció igual a la de la pregunta 3?

Fent us del mòdul capture de la unitat CCP. Connectarem el senyal extern a l'entrada CCPx. El primer flanc provocarà una còpia del valor del Timer (1 o 3) al registre CCPRx i una interrupció per avisar que ha arribat. El segon cop que arribi es farà altre cop una còpia del valor del Timer i una altra interrupció. En aquest moment podem calcular la diferència de temps.

5. (1 Punt) Què li passa a la sortida CCP2 de la unitat PWM si el valor de CCPR2L > PR2 ?

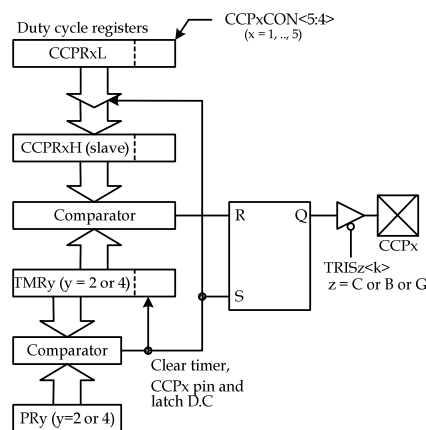


Figure 8.24 Simplified PWM block diagram (redraw with permission of Microchip)

Si CCPR2L és major que PR2, es farà el SET del biestable i un reset al valor del Timer abans de que el seu valor arribi al de CCPR2L (tingueu en compte que aquest valor es

copia a CCPR2H a cada període). Per tant el comparador de dalt mai donarà cert i no es farà mai un RESET al biestable. En conseqüència la sortida sempre estarà a “1” ja que només fem SETs al biestable.

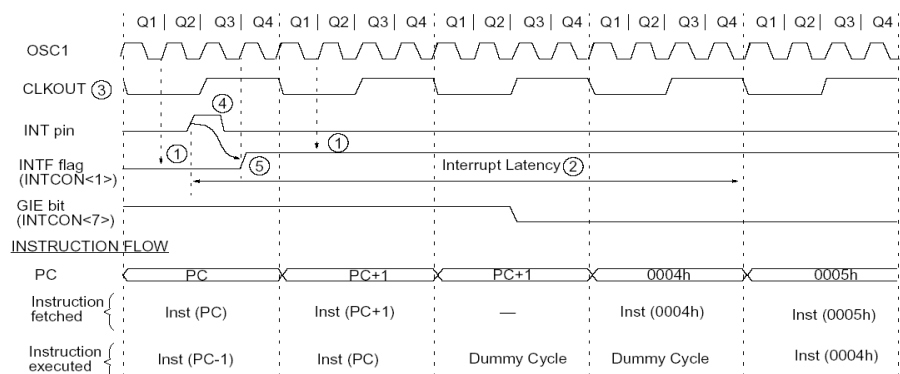
6. (3 Punts) Si la freqüència de l'oscil·lador és de 8MHz, quin és el temps mínim que ha de passar entre dos motius d'interrupció per garantir que es s'atendran els dos (és a dir que la RSI es dispararà dues vegades), en els següents casos:

- Les dues són del generades per la INT2 externa.

Com es veu a la figura, des del flanc d'interrupció al pin INT fins que s'executa la primera instrucció de la RSI passaran 3 o 4 cicles d'instrucció. Si aquesta primera instrucció baixa el INTF (el posa a 0 amb un clear), ja estarem a punt per enregistrar un altre motiu d'interrupció. Si suposem doncs 4 cicles d'instrucció, són 16 cicles d'oscil·lador, a 125ns cada un (1/8Mhz) tenim un temps de 2us.

- Una és de l'INT2 i l'altra del Timer0.

Com que els motius d'interrupció són diferents, encara que arribin simultaneament les podem enregistrar (i s'atendran quan es pugui). Per tant no hi ha temps mínim entre els dos motius d'interrupció.



Note 1: INTF flag is sampled here (every Q1).
 2: Interrupt latency = 3-4 Tcy where Tcy = instruction cycle time.
 Latency is the same whether Instruction (PC) is a single cycle or a 2-cycle instruction.
 3: CLKOUT is available only in RC oscillator mode.
 4: For minimum width of INT pulse, refer to AC specs.
 5: INTF is enabled to be set anytime during the Q4-Q1 cycles.