

Nom i Cognoms: _____ **Una possible solució** _____

1) Volem comptar el nombre d'interrupcions que esdevenen en INT0 i anar-ho visualitzant en uns leds que hem connectat a la sortida del PORTD. Un enginyer de telecomunicacions presenta el següent codi d'atenció a la RSI externa:

```
ISR_hi      btfss  INTCON,INT0IF    ; check interrupt source
             retfie fast             ; not caused by INT0, return
             decf   count,F
             movff  count,PORTD      ; output count to LEDs
             retfie fast
```

Raoneu si la RSI us sembla correcta, i en cas contrari esmenteu quin és l'error.

No és correcta. Falta baixar el flag INT0IF un cop atesa la interrupció.

bcf INTCON,INT0IF

2) Quan usem el pragma:

```
#pragma interruptlow rsi
void rsi (void)
{
    ...
}
```

Quina particularitat té la rutina *rsi* que no tindria sense el pragma?

El compilador acabarà la rutina amb retfie.

3) És correcte acabar una rsi de baixa prioritat amb la instrucció *retfie FAST*? Que passaria si ho féssim?

No és correcte. Si entrés una interrupció d'alta prioritat mentre s'està executant la de baixa, es matxacaria el contingut dels shadow registers.

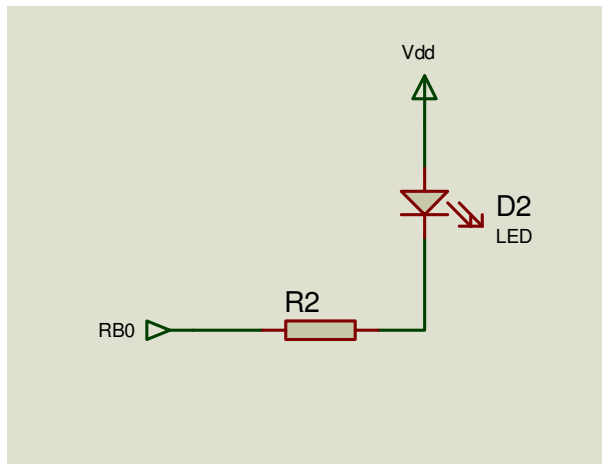
4) La latència d'una interrupció del PIC18 està entre 3 i 4 cicles d'instrucció. Expliqueu el perquè d'aquesta indeterminació

Perquè hi ha interrupcions assíncrones que poden arribar en mig del cicle d'instrucció. Cal que acabi el cicle i a continuació esperar 3 cicles d'instrucció més.

5) Hi ha algun cas en que aquesta indeterminació no es produeixi i es pugui afirmar que la latència és de 3 cicles?

Si, en el cas d'interrupcions síncrones. La latència és llavors de 3 cicles exactament.

6) Trobeu el valor de R2 per a que pel LED passi un corrent permanent de 10mA quan l'activem mitjançant el pin RB0 del micro.



$V_{\gamma} = 1.5V$
 $V_{IH} = 2V$
 $V_{IL} = 0.8V$
 $V_{OH} = 5V$
 $V_{OL} = 0.6V$
 $V_{dd} = 5V$

$$R2 = (V_{dd} - V_{\gamma} - V_{OL}) / 10mA = 290\Omega$$

7) Uns estudiants presenten en la seva pràctica sobre la placa EASYPIC6 una espera activa per a comprovar si el LCD està disponible abans d'enviar-hi una nova dada. L'espera es fa amb la següent crida a la funció XLCDIsBusy :

```
while (XLCDIsBusy) ;
```

Aquesta funció llegeix el bit BUSY de la LCD.
Raoneu si us sembla correcta aquesta solució.

No és correcta. La LCD de la placa EASYPIC no es pot llegir

8) Per a canviar l'estat d'un led penjat al pin RA0, se'ns presenta el següent codi:

```

TRISAbits.TRISA0 = 1; // RA0 for input
if (PORTAbits.RA0)
{
    TRISAbits.TRISA0 = 0; // RA0 for output
    PORTAbits.RA0 = 0;    // Apaguem
}
else
{
    TRISAbits.TRISA0 = 0; // RA0 for output
    PORTAbits.RA0 = 1;    // Encenem
}

```

Per quin motiu aquest codi no es correcte ?

Perquè el que llegim del PORTA no és necessàriament el que hem escrit. Cal llegir el registre LATA.