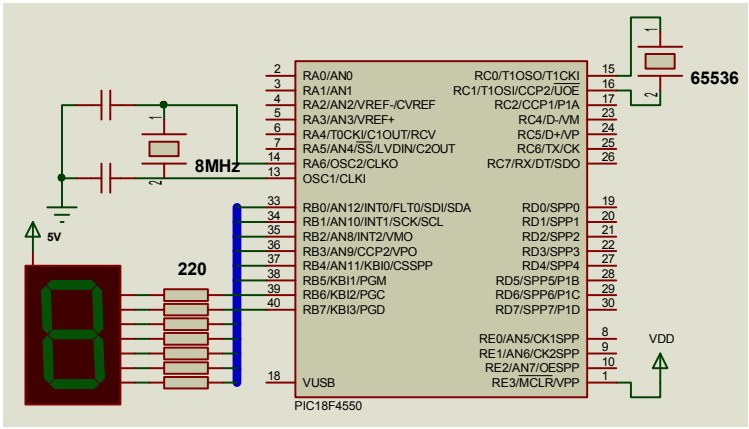
Nom i Cognoms:_____

1) El circuit de la figura, presenta la connexió d'un display de 7 segments al PORTB del PIC18F4550



a) (1p.) Quin valor hem de carregar als registres TRISB i PORTB per a encendre tots leds del display? Per què?

TRISB = _____

PORTB = _____

b) (1p.) Calculeu el corrent I_{RB0} quan el led corresponent està encès. Considereu V_γ=1'1V, V_{dd} = 5V, Vol = 0'6V, Voh = 4'3V. JUSTIFICA el resultat

I_{RB0} =_____

c) (2p.) Volem mostrar pel display el comptatge de segons (cíclic del 0 al 9). Per a tal fi, usarem el timer1 connectat a un oscil.lador de quartz extern de 2^{16} Hz.

```
main ( )                                    }
{
        ....//inicialitzacions    vàries    i    Void interrupt RSI_timer1 (void)
correctes                                   {
        for ( ; ; )                                 segs ++;
        {                                           segs = segs %10 ;
                PORTB = PatroLeds[segs];            TMR1IF = 0;
                Sleep ( ) ;                 }
        }
```

A partir del codi presentat, indiqueu el valor (en binari) del registre T1CON per a que el comptatge de segons sigui correcte. JUSTIFICA la resposta.

T1CON = _____

d) (1p.) Durant quin percentatge de temps aproximat el micro estarà dormint? (Podeu considerar que el compilador optimitza el codi).

Nom i Cognoms:_____

	7	6	5	4	3	2	1	0
	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON
value after reset	0	0	0	0	0	0	0	0

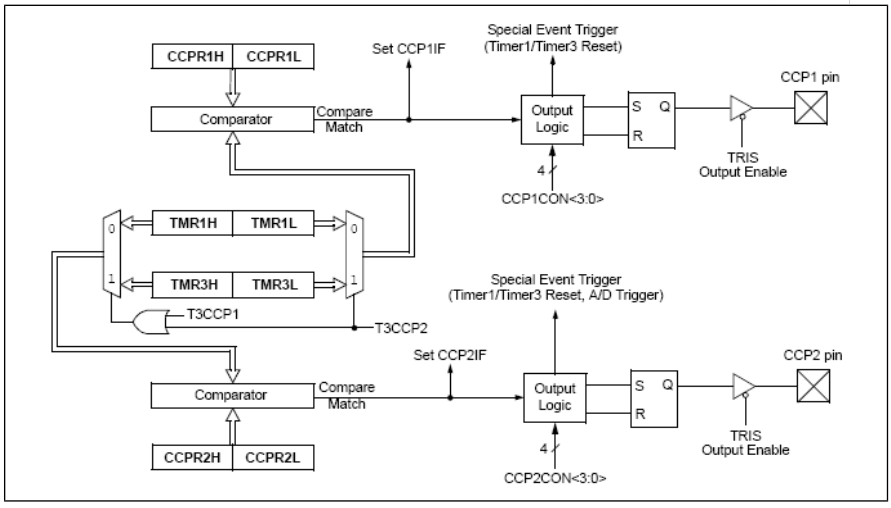
RD16: 16-bit read/write mode enable bit
0 = Enables read/write of Timer3 in two 8-bit operations
1 = Enables read/write of Timer3 in 16-bit operation
T3CCP2:T3CCP1: Timer3 and Timer1 to CCPx enable bits
00 = Timer1 and Timer2 are the clock sources for CCP1 through CCP5
01 = Timer3 and Timer4 are the clock sources for CCP2 through CCP5;
Timer1 and Timer2 are the clock sources for CCP1
10 = Timer3 and Timer4 are the clock sources for CCP3 through CCP5;
Timer1 and Timer2 are the clock sources for CCP1 and CCP2
11 = Timer3 and Timer4 are the clock sources for CCP1 through CCP5
T3CKPS1:T3CKPS0: Timer3 input clock prescale select bits
00 = 1:1 prescale value
01 = 1:2 prescale value
10 = 1:4 prescale value
11 = 1:8 prescale value
T3SYNC: Timer3 external clock input synchronization select bit
When TMR3CS = 1
0 = Synchronizes external clock input
1 = Do not synchronize external clock input
When TMR3CS = 0
This bit is ignored.
TMR3CS: Timer3 clock source select bit
0 = Instruction cycle clock (FOSC/4)
1 = External clock from pin RC0/T1OSO/T13CKI
TMR3ON: Timer3 on bit
0 = Stops Timer3
1 = Enables Timer3

Figure 8.8. T3CON contents (redraw with permission of Microchip)

	7	6	5	4	3	2	1	0
	--	--	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
value after reset	0	0	0	0	0	0	0	0

DCxB1:DCxB0: PWM duty cycle bit 1 and bit 0 for CCP module x
capture mode:
unused
compare mode:
unused
PWM mode:
These two bits are the lsbs (bit 1 and bit 0) of the 10-bit PWM duty cycle.
CCPxM3:CCPxM0: CCP module x mode select bits
0000 = capture/ compare/ PWM disabled (resets CCPx module)
0001 = reserved
0010 = compare mode, toggle output on match (CCPxIF bit is set)
0100 = capture mode, every falling edge
0101 = capture mode, every rising edge
0110 = capture mode, every 4th rising edge
0111 = capture mode, every 16th rising edge
1000 = compare mode, initialize CCP pin low, on compare match force CCP pin high (CCPxIF bit is set)
1001 = compare mode, initialize CCP pin high, on compare match force CCP pin low (CCPxIF bit is set)
1010 = compare mode, generate software interrupt on compare match (CCP pin unaffected, CCPxIF bit is set).
1011 = compare mode, trigger special event (CCPxIF bit is set)
For CCP1 and CCP2: Timer1 or Timer3 is reset on event
For all other modules: CCPx pin is unaffected and is configured as an I/O port.
11xx = PWM mode

Figure 8.10 CCPxCON register (x = 1,..,5) (redraw with permission of Microchip)



Nom i Cognoms:_____

2) Volem utilitzar la unitat CCP2 per fer una funció Delay_us(int n) on rebem un paràmentre n que és el nombre de microsegons que volem esperar. Internament, farem servir el Timer3 ja que el Timer1 el tenim ocupat a l'exercici 1. L'oscil·lador connectat al nostre processador (Fosc) és de 8 MHz. Observeu el codi de la funció:

```
void Delay_us(int n)
{
    TIMER3 = 0 ;
    CCPR2 = n;
    while(!CCP2IF);
    CCP2IF=0;
}
```

a) Per que la funció Delay_us(int n) sigui efectiva i actuï correctament, amb quins valors haurem hagut d'inicialitzar, els registres CCP2CON i T3CON? Justifica la resposta!! (1p)

CCP2CON =

T3CON =

b) Quin és el nombre màxim que podem acceptar com a paràmetre? Per què? (0,5p)

c) Té sentit cridar la funció amb un n petit, com per exemple: Delay_us(3) Justifica la resposta. (1p)

Nom i Cognoms:_____

	7	6	5	4	3	2	1	0
	--	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
value after reset	0	0	0	0	0	0	0	0

TOUTPS3:TOUTPS0: Timer2 output postscale select bits
0000 = 1:1 postscale
0001 = 1:2 postscale
.
.
.
1111 = 1:16 postscale
TMR2ON: Timer2 on bit
0 = Timer2 is off
1 = Timer2 is on
T2CKPS1: T2CKPS0: Timer2 clock prescale select bits
00 = prescaler is 1
01 = prescaler is 4
1x = prescaler is 16

Figure 8.6. T2CON control register (redraw with permission of Microchip)

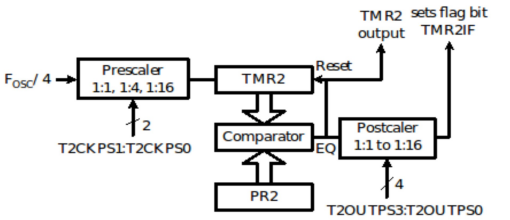


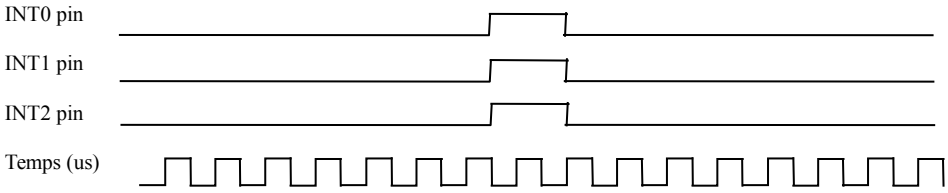
Figure 8.5 Timer2 block diagram (redraw with permission of Microchip)

3) Observeu el següent codi. El rellotge del PIC és de **8MHz**. (suposeu que, inicialment, INT0IE=0; INT1IE=1; INT2IE=0; GIEH=1; INT0IP=1; INT1IP=1; INT2IP=1, INT0IF=0; INT1IF=0; INT2IF=0);

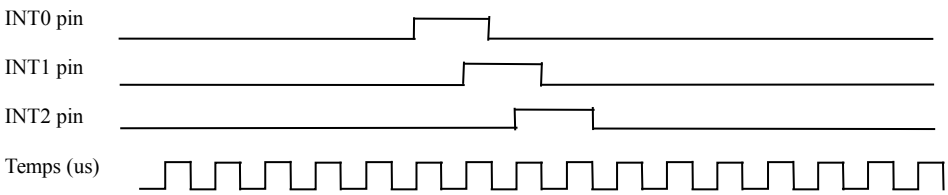
```
void interrupt InterruptHigh( )
{
        if ((INT0IF)&&(INT0IE))
        {
                INT0IF=0; INT1IF=0;INT2IF=0;
                INT2IE=1;
        }
        if ((INT1IF)&&(INT1IE))
        {
                INT0IF=0; INT1IF=0;INT2IF=0;
                INT0IE=1;
        }
        if ((INT2IF)&&(INT2IE))
        {
                INT0IF=0; INT1IF=0;INT2IF=0;
                EncénLED();
        }
}
```

Ara observeu les següent seqüències d'events, amb referència temporal, als pins d'interrupció, que estan ben configurats:

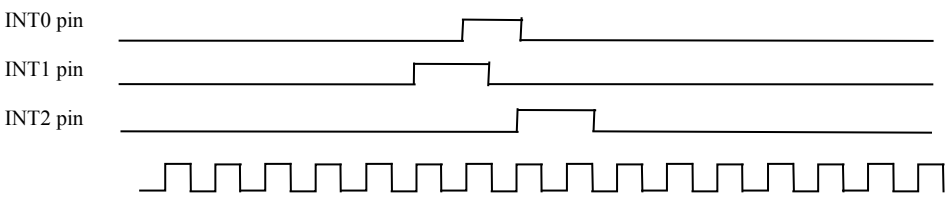
Seqüència 1:



Seqüència 2:



Seqüència 3:



Nom i Cognoms:_____

Alguna o algunes d'aquestes seqüències provocaran l'encesa del LED? Justifica la resposta (1p) ?

4) Programeu tot el necessari per generar al PIN CCP1 un senyal PWM de freqüència 1KHz i DC=18%. Suposeu Fosc=8MHz. (1,5p)

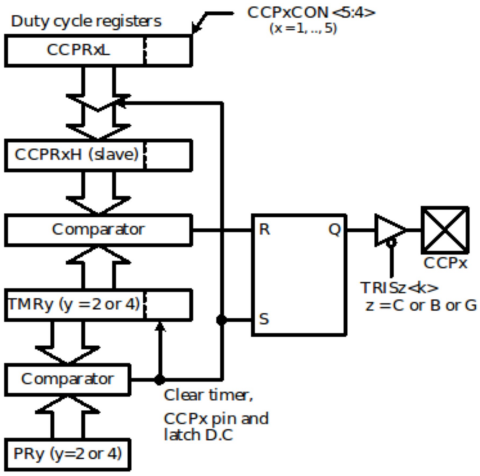


Figure 8.24 Simplified PWM block diagram (redraw with permission of Microchip)

PWM period = [(PRy) + 1] × 4 × TOSC × (TMRy prescale factor)
PWM duty cycle = (CCPRxL:CCPxCON<5:4>) × TOSC × (TMRy prescale factor)

Nom i Cognoms:_____

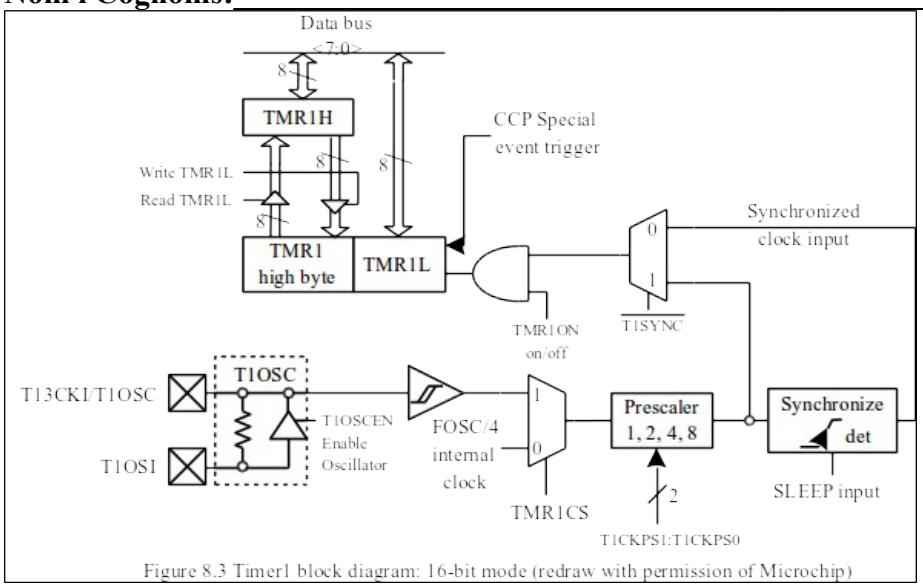


Figure 8.3 Timer1 block diagram: 16-bit mode (redraw with permission of Microchip)

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
1 = Device clock is derived from Timer1 oscillator
0 = Device clock is derived from another source

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from RC0/T1OSO/T13CKI pin (on the rising edge)
0 = Internal clock (Fosc/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1