

**Nom i Cognoms:** \_\_\_\_\_ **Una possible solució** \_\_\_\_\_

1) Justifiqueu quina de les dues últimes línies del següent codi és innecessària, si ho són les dues, o bé si calen les dues.

```

                org      0x00

start          movf     0x20,W,A      ; WREG ← [0x20]
               addwf    0x30,W,A      ; WREG ← [0x20] + [0x30]
               movwf    0x50,A        ; 0x50 ← sum (in WREG)

// dues últimes línies de codi
forever        goto forever
               end

```

Les dues són necessàries.

Un programa no pot acabar la seva execució. Cal el bucle infinit.

La directiva end és necessària per a indicar a l'ensamblador on acaba el codi.

2) Quines unitats de memòria diferents identifiqueu en l'esquema del PIC18F4550? Enumera-les, indicant el la seva amplada en bits.

Memòria	Amplada
De programa	16
De dades	8
Pila	20
EEPROM	8

3) Un enginyer de telecomunicacions presenta la següent rutina per a calcular el factorial d'un cert nombre  $n$ .

Us sembla raonable implementar-lo en el PIC18F4550? Justifiqueu la resposta.

```

int factorial(int n)
{
    int temp;

    if(n <= 1) return 1;
    temp = n * factorial(n - 1);
    return temp;
}

```

No és gens raonable implementar rutines recursives amb una pila de tan poca capacitat.

4) En que es diferencia el resultat d'aquestes dues instruccions?

a) PORTA = valor;    b) LATA = valor;

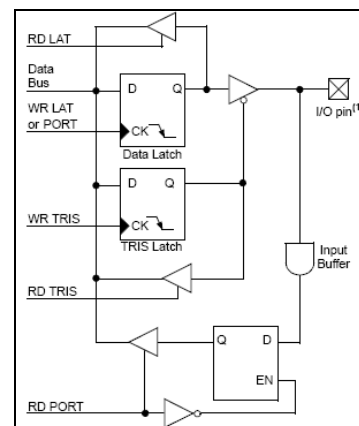
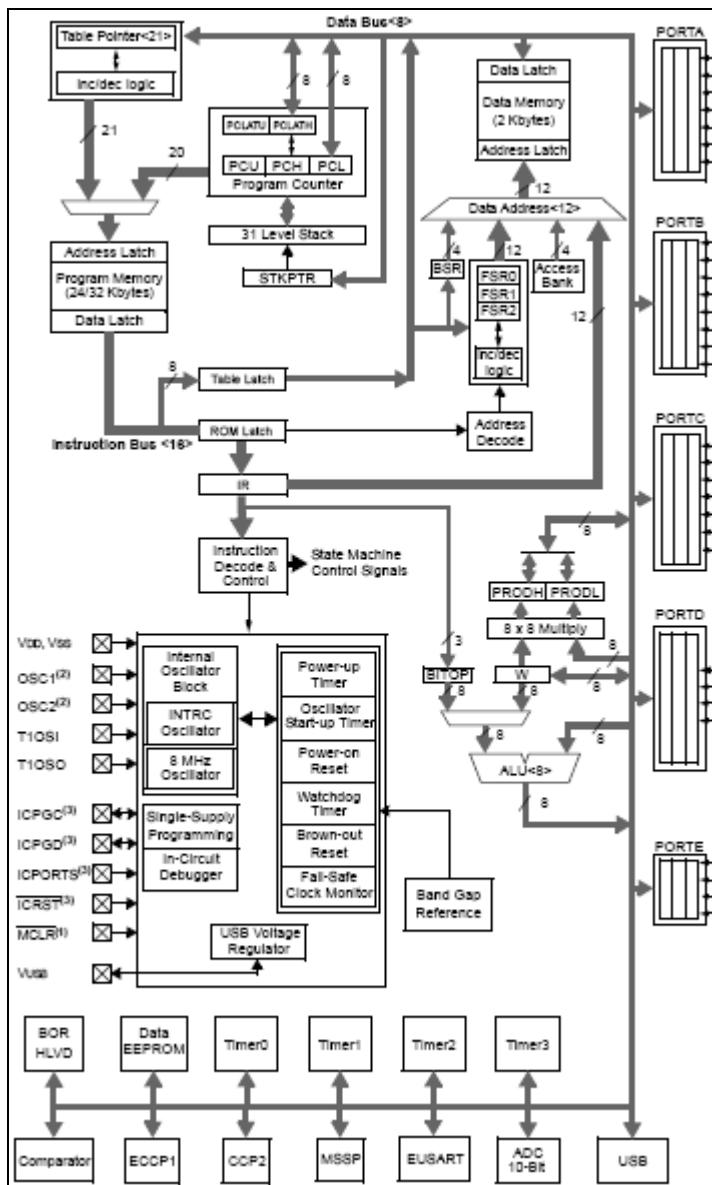
Són equivalents

i el d'aquestes dues?

a) vble=PORTA;    b) vble = LATA;

La primera llegeix l'estat dels pins d'entrada, la segona llegeix l'últim valor escrit al PORT

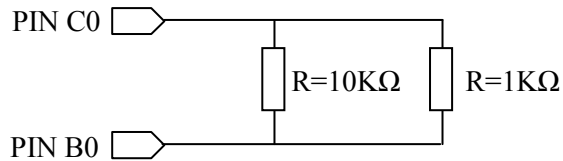
5) La següent imatge mostra un fragment del protoboard.  
 Uniu amb ratlles tots aquells punts elèctrics que estan físicament curtcircuitats en el protoboard real.



Nom i Cognoms: \_\_\_\_\_ **Una possible solució** \_\_\_\_\_

6) Quin valor lògic llegirem a l'entrada del PIN B0 després d'executar aquest codi?

```
TRISC=0x00;
TRISB=0xFF
LATC=0xFF;
var=PORTB;
```



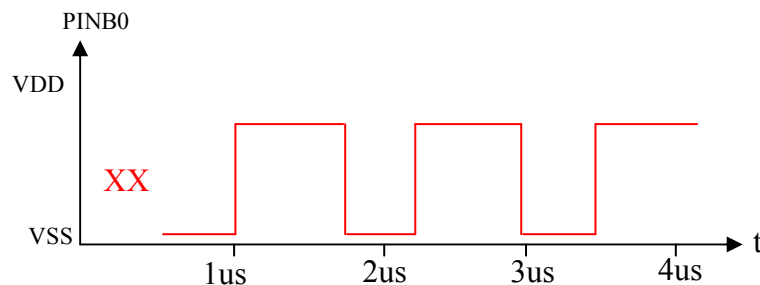
(nota: VDD=4'8V, VOLmax=0'5V, VOHmin=4V, VILmax=1V, VIHmin=3V)

Pel bit 0 del port C sortirà un "1" lògic, amb tensió entre 4V i 4'8V. Com que a les resistències no caurà pràcticament tensió, a l'entrada tindrem un 1 lògic també ja que la tensió superarà els 3V. De fet la tensió màxima que pot caure a les resistències seria 909Ω (equivalent del paral·lel) per la corrent de fuga (que és de l'ordre de 200nA), per tant inferior al mV.

7) Si posem un oscil·lador de 16 MHz al nostre PIC, quin senyal tindrem a la sortida del PIN B0 a l'executar el següent codi?

; temps t=0

```
et:   clrf TRISB,0
      clrf PORTB,0
      nop
      setf PORTB,0
      bra et
```



Com que la freqüència és de 16MHz, el TOSC és de 62,5ns i per tant el cicle d'instrucció és de  $4 \times 62,5\text{ns} = 250\text{ns}$ . Durant el temps de fetch i execució del `clrf TRISB` la sortida és indeterminada (dos cicles) a partir d'aquí s'executa el bucle: es posa a 0 durant dos cicles (`clrf, nop`), i després a 1 durant tres (`setf, bra`) i així per sempre.

8) Amb quin valor queda el WREG després d'executar el següent codi?

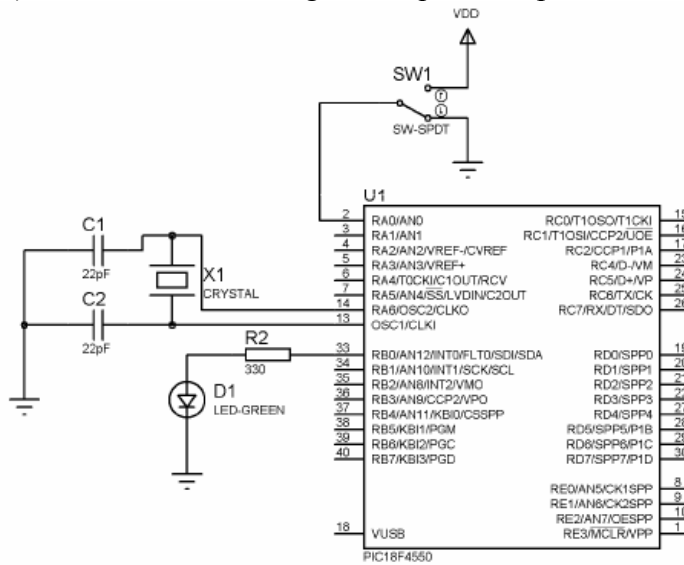
```
movlw    0x12
setf     0x12,0
addwf    0x12,0,0
```

primer WREG=0x12, després la posició 0x12 de l'access bank es posa a FF, finalment es suma 0xFF amb el WREG=0x12, i el resultat queda al WREG, per tant al final: WREG=0x11.

9) Per quin motiu en el PIC18F4550 el registre BSR només implementa 4 bits?

Com que aquest model només té 4KB de RAM, calen 12 bits per adreçar-la. 8 els tenim en el paràmetre de la instrucció, els altres 4 s'obtenen del Bank Select Register.

10) Quina errada en el següent esquema impediria una simulació correcta en el Proteus?



El pin 1 del microcontrolador, MCLR, es troba a l'aire, per tant aniria fent resets contínuament! Es pot arreglar posant-hi una tensió de VDD amb una resistència de Pull-up.

BRA	Unconditional Branch
Syntax:	BRA n
Operands:	$-1024 \leq n \leq 1023$
Operation:	$(PC) + 2 + 2n \rightarrow PC$
Status Affected:	None
Encoding:	1101 0nnn nnnn nnnn
Description:	Add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$ . This instruction is a two-cycle instruction.

ADDWF	ADD W to f
Syntax:	ADDWF f{,d{,a}}
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	$(W) + (f) \rightarrow \text{dest}$
Status Affected:	N, OV, C, DC, Z
Encoding:	0010 01da ffff ffff
Description:	Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

SETF	Set f
Syntax:	SETF f{,a}
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$
Operation:	$FFh \rightarrow f$
Status Affected:	None
Encoding:	0110 100a ffff ffff
Description:	The contents of the specified register are set to FFh. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the

NOP	No Operation
Syntax:	NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Encoding:	0000 1111 0000 0000
Description:	No operation.
Words:	1
Cycles:	1

MOVLW	Move Literal to W
Syntax:	MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	$k \rightarrow W$
Status Affected:	None
Encoding:	0000 1110 kkkk kkkk
Description:	The eight-bit literal 'k' is loaded into W.
Words:	1
Cycles:	1

CLRF	Clear f
Syntax:	CLRF f{,a}
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$
Operation:	$000h \rightarrow f$ , $1 \rightarrow Z$
Status Affected:	Z
Encoding:	0110 101a ffff ffff
Description:	Clears the contents of the specified register. If 'a' is '0', the Access Bank is selected.