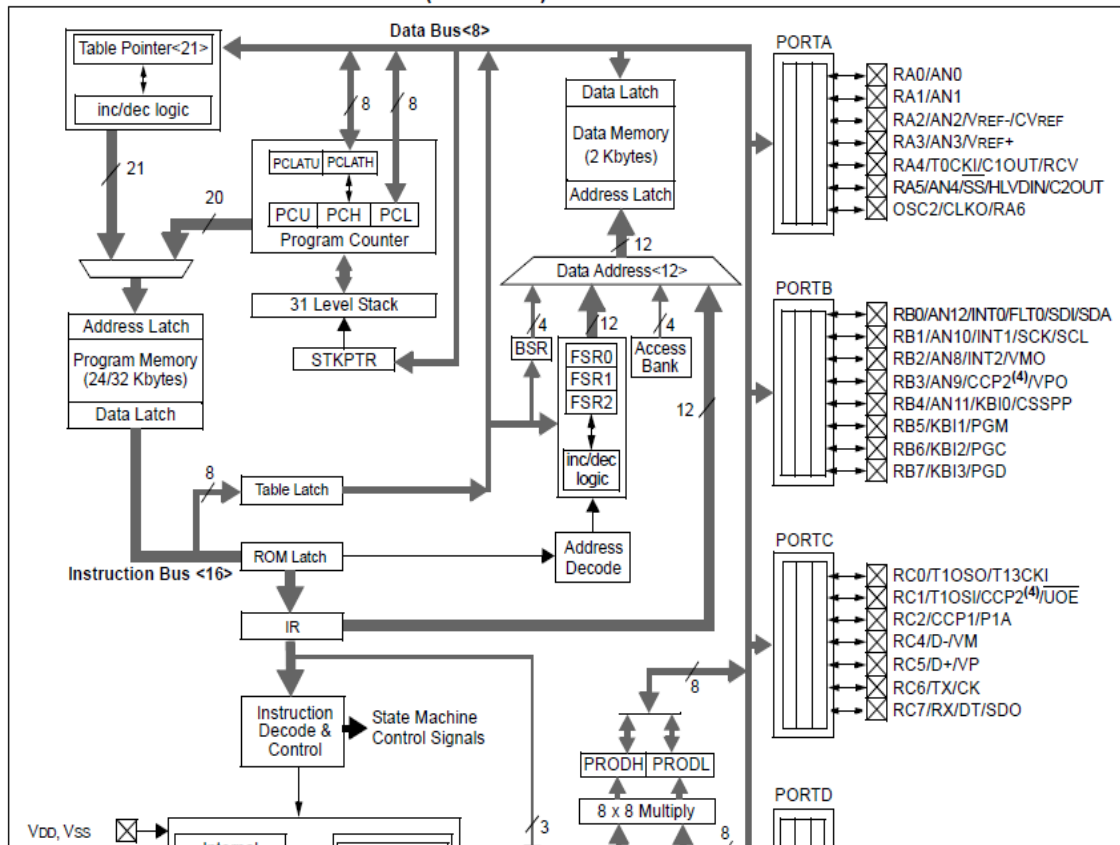


Cognoms i Nom: _____

Doc. Identitat: _____

Totes les respostes han d'estar degudament justificades

1) Per poder adreçar la totalitat de la memòria de programa en el PIC18F4550 són necessaris 21 bit. En canvi, segons el diagrama de blocs de la CPU del PIC18F4550 de la figura, el bus per transportar les adreces que van del Program Counter cap al multiplexor que marcarà la posició de memòria de programa que volem adreçar té 20 bits. Explica aquesta circumstància. (1 punts)

FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM

La raó d'aquesta circumstància es que les instruccions sempre esta adreçades en posicions de memòria parells i per tant el bit menys significatiu és sempre 0. Per optimitzar el número de connexions internes aquest bit, de valor sempre fixe, no es transporta pel bus.

2) Quin és el factor més important que heu de considerar a l'hora de decidir si en el vostre programa fareu servir el mode d'adreçament Access bank ó Banked memory, per a accedir a la memòria de dades? (1,5 punts)

El factor més important a considerar serà el nombre de variables que utilitzarem. Si la quantitat total de les variables (GPR) és inferior o igual a 96 bytes llavors farem servir mode accés bank simplificant el mode d'accés de memòria de dades ja que no ens caldrà manegar el registre BSR. En cas contrari haurem de fer servir mode Banked memory.

3) En un microcontrolador de les característiques del PIC18F es considera ampliar la memòria RAM fins a 64KBytes, organitzada en bancs de 1KBytes. Tenint en compte aquesta modificació, **responeu de forma justificada** les següents preguntes: (2 punts)

a) És possible tenir memòria de programa i memòria de dades separades ?

SI. Aquesta característica té a veure amb que l'arquitectura és tipus Harvard, i és independent de com estigui organitzada la memòria RAM.

b) És possible mantenir el mecanisme de l'*access bank* ?

SI. El fet que el fabricant proposi un mecanisme d' access bank és independent de la dimensió de la memòria i de la mida de cada banc.

c) Quin seria el nombre mínim de bits necessaris pel LSFR (adreçament indirecte) ?

Els registres FSR0, FSR1 i FSR2 adrecen la memòria de dades de forma continua (sense divisions en bancs), i per tant el seu contingut indica de forma única la posició de RAM a la que s'ha d'accedir. Per accedir a una posició d'una memòria de 64KB, calen com a mínim 16 bits.

d) Quin seria el nombre mínim de bits necessaris pel BSR ?

Una memòria de 64KB dividida en bancs de 1KB, té 64 bancs. Per tant necessitem com a mínim 6 bits.

e) Quin seria el nombre mínim de bits necessaris, a incloure en una instrucció, per a indicar la posició a accedir dintre de l'access bank ?

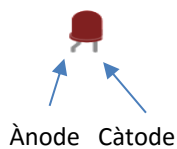
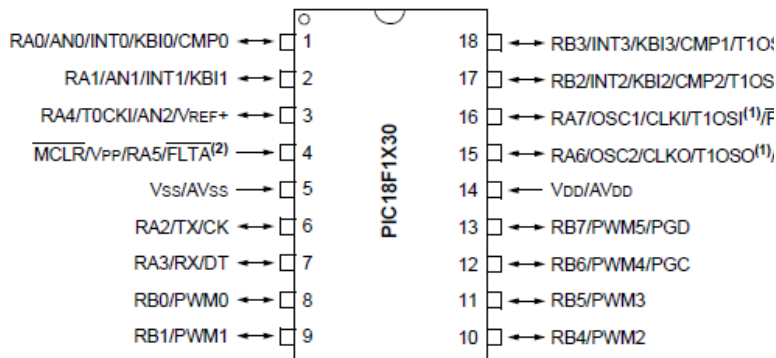
Si els bancs són de 1KB, per indicar 1 posició dintre de les 1024 que formen el banc necessitem com a mínim 10 bits.

Cognoms i Nom: _____

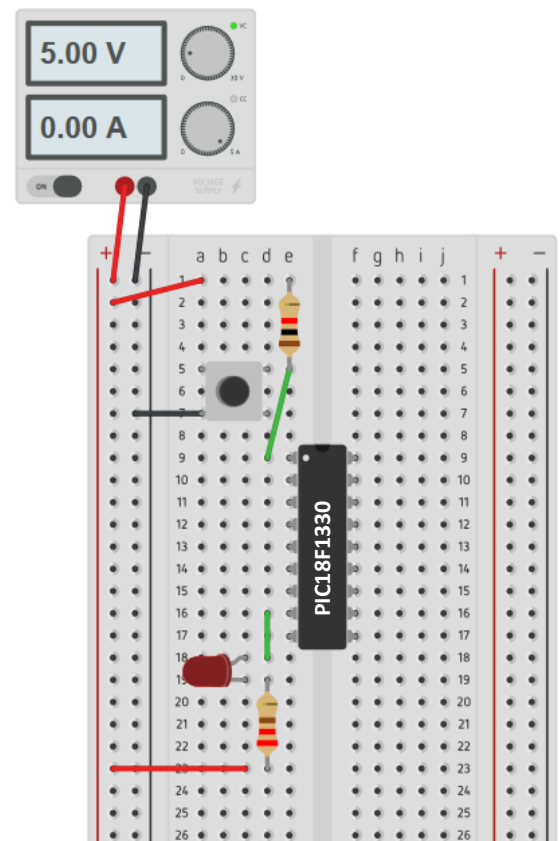
Doc. Identitat: _____

Totes les respostes han d'estar degudament justificades

4) Identifica els possibles errors que no permeten el funcionament correcte del circuit muntat sobre protoboard de la figura (senyal d'entrada binari generat amb un polsador, i visualització del senyal de sortida amb un LED), i per cada cas, quina seria la forma de solucionar el problema. (2 punts)



Resistència superior= 10KΩ
Resistència inferior= 220Ω
Tensió llindà LED 1,9 V



Error 1. El xip microcontrolador no està alimentat. **SOLUCIÓ:** Cal connectar els borns + i – de la font d'alimentació als pins Vcc (14) i Vss (5) respectivament.

Error 2. El microcontrolador no té connectat el /MCLR. **Solució:** Connectar una residència de pull-up en el pin 4 (un valor adequat seria 10KΩ).

Error 3. El LED està mal connectat. **SOLUCIÓ:** Cal connectar l'ànode al potencial més alt, i el càtode al potencial més baix (és a dir, girar el LED). El LED 's'encén quan el pin de sortida està a "0". **SOLUCIÓ ALTERNATIVA:** deixar l'ànode connectat al pin 8, i el càtode a la resistència, però l'altre extrem de la resistència la connectem a GND enlloc de Vcc. En aquest cas, el LED s'encén quan el pin de sortida està a "1".

5) Segons l'especificació de la instrucció de salt relatiu BRA, entre quins valors pot estar el literal que indiquem com a operand de la instrucció. (1,5 punts)

BRA		Unconditional Branch						
Syntax:	[<i>label</i>] BRA <i>n</i>							
Operation:	$(PC) + 2 + 2n \rightarrow PC$							
Status Affected:	None							
Encoding:	<table border="1"><tr><td>1101</td><td>0nnn</td><td>nnnn</td><td>nnnn</td></tr></table>				1101	0nnn	nnnn	nnnn
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.							
Words:	1							
Cycles:	2							

Donat que el literal de la instrucció pot tenir 11 bits codificats en complement a 2 el rang que podem especificar és de: -1024 ≤ n ≤ 1023

6) Indica el valor final de totes les posicions de la memòria de dades i registres de la CPU que modifica l'execució del següent conjunt d'instruccions. A la taula adjunta s'indica el valor de la memòria RAM abans l'execució (descripció parcial). (2 punts)

```
.....
i EQU 0x0B                ; definim etiqueta i amb valor 0x0B
    movlb 0x04              ; BSR <- 0x04
    lfsr 0, 0x000           ; FSR0<- 0x000
    lfsr 1, 0x100           ; FSR1<- 0x100
    lfsr 2, 0x200           ; FSR2<- 0x200
    movlw 0x03              ; WREG<- 0x03
    movwf i,0               ; i<- WREG;
Loop
    movf POSTINC1, W, 0      ; WREG<- [FSR1] i FSR1++ (postincrement de FSR1)
    addwf POSTDEC2, W, 0     ; WREG<- WREG + [FSR2] i FSR2-- (postdecrement de FSR2)
    movwf POSTINC0, 0        ; [FSR0]<- WREG i FSR0++ (postincrement de FSR0)
    decfsz i,F,A             ; decrementa i (i<-i-1), skip si i=0
    goto Loop                ; salt incondicional a etiqueta Loop
fi nop                       ; bucle final
    goto fi
END
```

@ RAM	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00_	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
0x0F_	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0x10_	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
0x1F_	30	2F	2E	2D	2C	2B	2A	29	28	27	26	25	24	23	22	21
0x20_	20	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F

Solució:

Nota: Els comentaris del codi en color vermell són part de la solució

Posicions i registres modificats per l'execució del codi indicats en vermell.

@ RAM	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00_	30	32	34	F3	F4	F5	F6	F7	F8	F9	FA	00	FC	FD	FE	FF
0x0F_	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0x10_	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
0x1F_	30	2F	2E	2D	2C	2B	2A	29	28	27	26	25	24	23	22	21
0x20_	20	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F

BSR= 0x04 FSR0= 0x0003 FSR1= 0x0103 FSR2= 0x01FD WREG= 0x34