

Nom i Cognoms: \_\_\_\_\_ POSSIBLE SOLUCIÓ \_\_\_\_\_

1. El PIC18F45K22 disposa de una memòria EEPROM de dades de 256 bytes. Al ser una memòria no volàtil les dades emmagatzemades no es perden en absència d'alimentació. Per accedir a una dada de la memòria EEPROM hem de configurar tres SFR anomenats EECON1 (especificat a sota), EEDATA i EEADR.

**REGISTER 6-1: EECON1: DATA EEPROM CONTROL 1 REGISTER**

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

El procés per llegir una dada consisteix en posar a 0 el bit CFGS i el bit EEGPD del registre EECON1 i escriure l'adreça de la dada que es vol llegir al registre EEADR. A continuació hem de posar a 1 el bit RD del registre EECON1 per tal d'iniciar una lectura. En el següent cicle d'instrucció la dada estarà disponible en el registre EEDATA. El bit RD es posa automàticament a 0 quan acaba el procés de lectura.

Volem recuperar una dada guardada a l'adreça 2 de la EEPROM. Escriu el codi d'una subrutina en assembler que recuperi aquesta dada i la retorni mitjançant el registre W. (2 PUNTS)

```
Sub: BCF EECON1, CFGS
      BCF EECON1, EEGPD
      MOVLW 2
      MOVWF EEADR
      BSF EECON1, RD
      MOVF EEDATA, W
      RETURN
```

2. Marca les afirmacions que siguin certes. (2 PUNTS)

- ☐ Totes les instruccions que modifiquen el PC triguen 2 cicles d'instrucció en executar-se.  
**Fals les instruccions d'skip poden trigar tres cicles.**
- ☐ El PC (Program Counter) del PIC esta format per dos registres de 8 bits PCH i PCL.  
**Fals, té 20 bits i hi ha el PCU també.**
- ☐ La memòria de programa només es pot adreçar amb adreces parelles .  
**Fals, amb les instruccions de TBL tenim un camí de 21 bits per accedir Byte a Byte a la memòria.**

Nom i Cognoms: \_\_\_\_\_ POSSIBLE SOLUCIÓ \_\_\_\_\_

- ☐ Hi ha un únic llindar de voltatge a partir del qual el PIC interpreta que el senyal d'entrada és un '0' lògic o un '1' lògic.  
**Fals, hi ha un llindar de 0, un llindar de 1, i enmig queda una regió on no se'ns pot garantir si llegirem un 0 o un 1 lògic.**
- ☐ En la configuració ànode comú d'un LCD 7 segments, hem d'escriure un '1' en un pin per tal que el LED associat s'encengui.  
**Fals, en ànode comú s'activen els LEDs a '0'.**

3. La instrucció RCALL, de la mateixa manera que la instrucció CALL, serveix per fer una crida a una subrutina. En el cas de RCALL, però, el paràmetre de la instrucció no especifica una adreça de salt, sinó un desplaçament relatiu, codificat en complement a 2, respecte el valor actual del PC en el moment de cridar-se. Aquí tens l'especificació (1 PUNT).

RCALL	Relative Call				
Syntax:	RCALL n				
Operands:	$-1024 \leq n \leq 1023$				
Operation:	$(PC) + 2 \rightarrow TOS,$ $(PC) + 2 + 2n \rightarrow PC$				
Status Affected:	None				
Encoding:	<table><tr><td>1101</td><td>1nnn</td><td>nnnn</td><td>nnnn</td></tr></table>	1101	1nnn	nnnn	nnnn
1101	1nnn	nnnn	nnnn		
Description:	Subroutine call with a jump up to 1K from the current location. First, return address $(PC + 2)$ is pushed onto the stack. Then, 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 2-cycle instruction.				

Segons aquesta especificació, quin és el rang d'adreces d'instrucció a les que podem saltar?

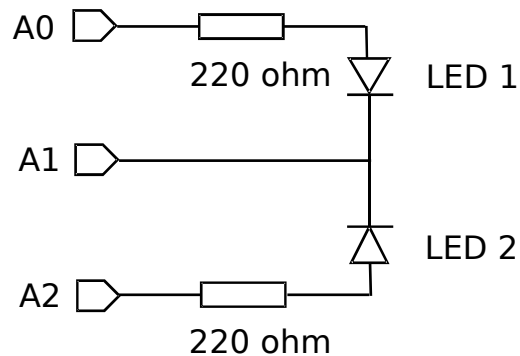
**$2048 \leq \text{salt} \leq 2046$**

Amb quants bits especifiquem la distància del salt relatiu?

**S'especifica amb 11 bits. Tot i que hi ha un bit amagat pel fet de que les instruccions estan sempre alineades a adreces parelles. D'aquí que l'adreça de salt en la formula  $(PC)+2+ 2n \rightarrow p$  la "n" es multipliqui per 2.**

Nom i Cognoms: \_\_\_\_\_ POSSIBLE SOLUCIÓ \_\_\_\_\_

4. Escriu el codi necessari perquè s'encenguin els dos LEDs de la figura:



Suposeu que el micro acaba d'arrancar i no hi ha res configurat (1,5 PUNTS).

```

MOVLB    0Fh          ; triem bank 15, on hi ha els registres SFR
BCF       ANSELA, 0, 1
BCF       ANSELA, 1, 1
BCF       ANSELA, 2, 1; posem els tres pins en mode digital
BCF       TRISA, 0, 1
BCF       TRISA, 1, 1
BCF       TRISA, 2, 1 ; configurem els tres com a sortida
BSF       LATA, 0, 1   ; pin A0 a '1' = 5V (podríem usar PORTA)
BCF       LATA, 1, 1   ; pin A1 a '0' = 0V (podríem usar PORTA)
BSF       LATA, 2, 1   ; pin A2 a '1' = 5V (podríem usar PORTA)
  
```

5. Calcula durant quant temps estarà a '1' el pin 0 del port B. El clock del micro és de 8MHz i el pin ja està configurat com a digital i de sortida (1,5 PUNTS).

```

NOP
SETF     PORTB, A      ; aquí s'encén
MOVLW    18d
BCF      PORTB, 2, A
NOP
Loop MOVWF X, A         ; es torna a carregar X amb 18 (W)
DECFSZ   X, F, A       ; mai surt del bucle!!!!!! X=17
BRA      Loop
NOP
BCF      PORTB, 0, A    ; aquí s'apagaria
  
```

No surt mai del bucle, per tant estarà sempre a '1'. (Si comptéssim que el bucle és "correcte" serien 3 instruccions d'un cycle + 17 vegades quatre cycles el bucle + una vegada el bucle 3 cycles + 2 cycles, uns 76 cycles, a 500 ns = 38 us.)

Nom i Cognoms: \_\_\_\_\_ POSSIBLE SOLUCIÓ \_\_\_\_\_

6. Omple la taula amb els valors resultants dels registres després d'executar aquest codi (2 PUNTS):

```
CLRF      01, B
DECF      01, F, B
DECF      01, W, A
DECF      00, F, B
MOVFF     001h ,000h
MOVLW     13d
MOVLB     0
ADDWF     01, F, B
```

	Valor inicial del registre	Valor final del registre
WREG	12h	13d
BSR	01h	00
000h	00h	01
001h	01h	14
100h	00h	FFh (-1d)
101h	23h	FFh (-1d)