

Cognoms, Nom _____ DNI _____

Tota resposta sense justificar es considerarà nul·la !

P1. (1,5 punts) Indica si les següents afirmacions sobre el PIC18F són certes (C) o falses (F). (Encert +0.3, Error -0.2)

- ☐ El byte menys significatiu d'una instrucció, s'emmagatzema sempre en una adreça parell de la memòria de programa
- ☐ Treballar usant 'Access bank' a la RAM és més ràpid que treballar amb la RAM en mode banked i disposem de més memòria
- ☐ Usant una Fosc de 4MHz, el temps d'execució d'una instrucció (no de salt) 'single-word' serà d'1µs
- ☐ La instrucció 'ADDWF INDF1, W, A' accedeix al banc de memòria indicat a FSR1H
- ☐ El bus d'adreces de la memòria de dades és de 12 bits, això permet tenir fins a 4096Bytes de dades sense haver de modificar l'arquitectura del PIC18F

P2. (1,5 punts) Una persona d'una altra facultat ha decidit fer ús de chatGPT per escriure una funció en ASM per llegir el valor del Program Counter d'un PIC18F45k22. Aquesta és la conversa amb les respostes reals de la IA. Comenta els errors i proposa una solució millor:

Usuari: Hola!

ChatGPT: Hola! Com et puc ajudar avui?

Usuari: Pots donar-me un exemple de com llegir el PC d'un PIC18F?

ChatGPT: Sí, es possible llegir el valor del Program Counter (PC) d'un PIC18F a través del llenguatge d'assemblea de baix nivell.

Aquí tens un exemple de codi en assemblea per llegir el valor del PC d'un PIC18F:

```
; Inici del programa
org 0x000 ; Adreça d'inici del programa

main:
    ; Llegir el valor del PC
    movf PCL, W ; Mova el contingut del registre PCL a la memòria temporal W
    ; Bucle infinit
    goto main
```

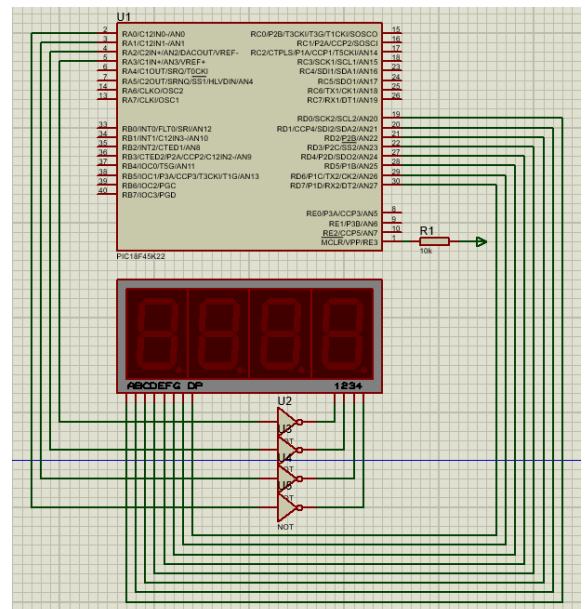
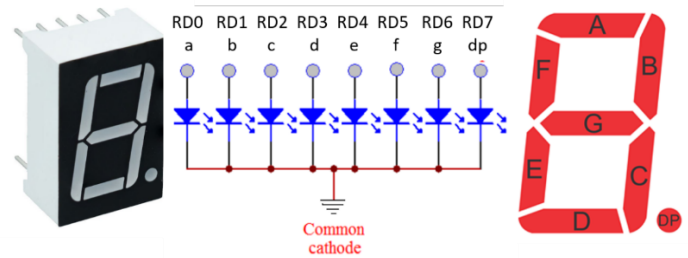
Usuari: Moltes gràcies!

ChatGPT: De res, sempre estic aquí per ajudar-te. Si tens alguna altra pregunta o necessites ajuda, no dubtis en demanar-m'ho!

P3. (2 punts) Tenim 4 displays 7-segments connectats a un PIC18F45K22 seguint el mateix esquema que a la placa de desenvolupament feta servir al laboratori:

La gestió de la selecció dels displays és: al treure un "1" al pin corresponent del PORTA, es selecciona el càtode comú d'un display i aquest s'encén. RA0 selecciona el display de més a la dreta, RA3 el de més a l'esquerra.

3.1 (0,5p) Quin valor tindran els registres ANSELA, ANSELD, TRISA i TRISD?



3.2 (1,5p) Dibuixa un cronograma en el que indiquis els valors que has d'escriure als registres LATA i LATD (el PORTA i el PORTD) per tal de veure el número '3.142' als displays (atenció al punt)

Cognoms, Nom _____ DNI _____

Tota resposta sense justificar es considerarà nul·la !

P4. (3 punts) El proper diumenge 7 de maig Barcelona acull la cursa popular de 10Km d’El Tall Anglès. Els organitzadors de la carrera ens demanen que implementem un cronòmetre més precís que el que usaven fins ara. Abans mesuraven el temps de pas dels corredors en segons. Ens demanen que ara mesurem el temps en centèsimes de segon. Farem servir el micro PIC18F45K22, amb una Fosc=12MHz. Cal utilitzar el **Timer2**, configurat per tal que cridem **la seva RSI exactament cada centèsima**, dins de la qual s’incrementarà la variable global *csCount*.

4.1 (1,5 punts) Configura els SFR necessaris per tal que el Timer2 generi un Interrupt Flag cada centèsima.

4.2 (1 punt) Configura els SFR necessaris del sistema d’interrupcions per tal que es cridi a la **RSI d’alta prioritat** amb cada Flag de Timer2.

4.3 (0,5 punts) El reglament de la cursa diu que tancarà l’arribada a meta al cap d’1 hora i 45 minuts de la sortida. Amb quin tipus de dades en C declararies la variable *csCount* per tal que pugui emmagatzemar valors fins el temps màxim de la prova, usant el mínim espai de memòria possible? (en la taula de la dreta pots veure quins tipus de dades admet XC8)

XC8 INTEGER DATA TYPES	
Type	Size (bits)
unsigned char	8
unsigned int	16
unsigned short long	24
unsigned long	32

P5. (2 punts) La pregunta anterior proporciona un comptador de centèsimes des de l'inici d'una prova esportiva. Considereu que cada cop que un corredor/a creua la meta, es genera un flanc ascendent en el pin INT1, i llegint el PORT C podem obtenir un codi de 8bits que identifica el corredor que acaba de passar. La RSI també fa una còpia del valor actual de centèsimes per guardar el temps d'aquest corredor/a.

Suposeu que ja està tot configurat per tal que es cridi la RSI d'alta prioritat amb cada flanc ascendent al pin INT1 i amb cada Flag del Timer2.

A continuació teniu el codi font assembler amb la RSI:

```
csCount    equ 0x000

            org 0x08

checkTMR2:  BTFSS PIR1,1,A
            BRA checkINT1
            BTFSS PIE1,1,A
            BRA checkINT1
            BCF PIR1,1,A
            CALL 0x200 ; increments csCount
checkINT1:  BTFSS INTCON3,0,A
            BRA finishRSI
            BTFSS INTCON3,3,A
            BRA finishRSI
            BCF INTCON3,0,A
copyId:     MOVF PORTC,W,A
            MOVWF POSTINC0,A
            LFSR FSR1,csCount
            MOVLW 3
copyTime:   MOVFF POSTINC1,POSTINC0
            CPFSEQ FSR1L,A
            BRA copyTime
finishRSI:  RETFIE FAST
```

5.1 (1 punt) Suposeu que hem entrat a la RSI perquè **ha vingut un Flag de INT1 però no de TMR2 (INT1IF=1, TMR2IF=0)**. Sabem que $F_{osc}=12\text{MHz}$. Calculeu el temps d'execució de la RSI.

5.2 (1 punt) Si a continuació del codi de dalt afegíssim una RSI de baixa prioritat, escriuríem una directiva `org 0x18`, i tot seguit el codi de la RSI_low. ¿Per a què afegim les directives `org 0x08` i `org 0x18`?

Amb el codi de l'enunciat, tindríem algun problema en relació amb les directives *org*? Si fos el cas, com ho solucionariéu?