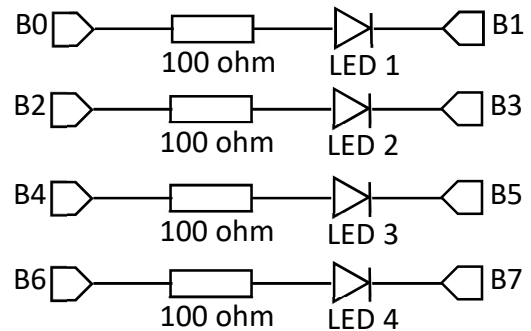


Nom i Cognoms: _____

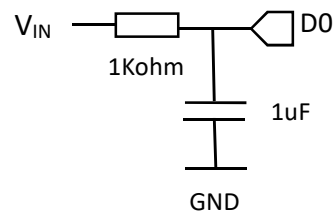
1. Tenim el següent muntatge als Pins del PORTB, i executem el tros de codi de sota. Indica quin o quins LED s'encendran (Els registres ANSELB, TRISB i PORTB es troben al Bank F de memòria). Justifica la resposta. **(1,5 PUNTS)**



```
MOVLB    0Fh           // al bank F hi ha els tres registres involucrats.
CLRF     ANSELB, B
CLRF     TRISB, B
MOVLW    7Ah
MOVWF    PORTB, B
```

2. Tenim connectat al PIN D0 un circuit resistència-condensador, que es carregarà segons l'equació de càrrega del condensador vista a classe **(2,5 PUNTS)**:

$$V_c = V_{in} (1 - e^{-t/RC})$$



Si $V_{IN} = 5V$, $V_{DD} = 5V$, el condensador està inicialment descarregat i el Pin D0 està configurat com a entrada digital:

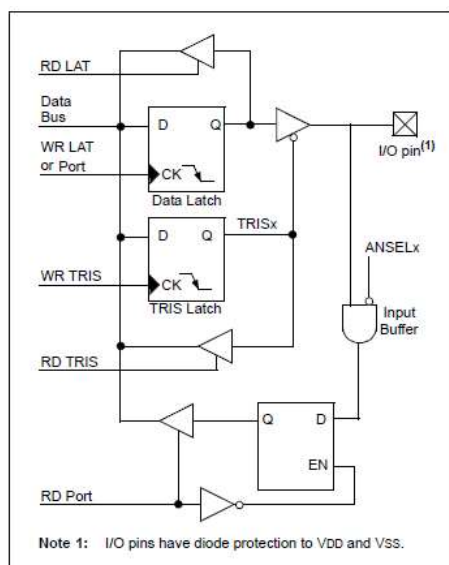
- Calcula durant quant temps l'entrada està amb seguretat a "0" lògic.

- Calcula a partir de quin instant de temps l'entrada a D0 serà amb seguretat un "1" lògic.

- Què ha passat entremig amb el valor lògic de l'entrada digital?

27.8 DC Characteristics: Input/Output Characteristics, PIC18(L)F2X/4XK22

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$				
Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
D140 D140A D141 D142 D142A	V _{IL}	Input Low Voltage					
		I/O PORT:					
		with TTL buffer	—	—	0.8	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
			—	—	$0.15 V_{DD}$	V	$1.8\text{V} \leq V_{DD} \leq 4.5\text{V}$
		with Schmitt Trigger buffer	—	—	$0.2 V_{DD}$	V	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$
		with I ² C levels	—	—	$0.3 V_{DD}$	V	
		with SMBus levels	—	—	0.8	V	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$
		MCLR, OSC1 (RC mode) ⁽¹⁾	—	—	$0.2 V_{DD}$	V	
D142A		OSC1 (HS mode)	—	—	$0.3 V_{DD}$	V	
D147 D147A D148	V _{IH}	Input High Voltage					
		I/O ports:		—	—		
		with TTL buffer	2.0	—	—	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
			$0.25 V_{DD} + 0.8$	—	—	V	$1.8\text{V} \leq V_{DD} \leq 4.5\text{V}$
		with Schmitt Trigger buffer	$0.8 V_{DD}$	—	—	V	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$
DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$				
Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
D159	V _{OL}	Output Low Voltage I/O ports	—	—	0.6	V	I _{OL} = 8 mA, V _{DD} = 5V I _{OL} = 6 mA, V _{DD} = 3.3V I _{OL} = 1.8 mA, V _{DD} = 1.8V
D161	V _{OH}	Output High Voltage ⁽³⁾ I/O ports	V _{DD} - 0.7	—	—	V	I _{OH} = 3.5 mA, V _{DD} = 5V I _{OH} = 3 mA, V _{DD} = 3.3V I _{OH} = 1 mA, V _{DD} = 1.8V



REGISTER 10-8: TRISx: PORTx TRI-STATE REGISTER⁽¹⁾

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx7	TRISx6	TRISx5	TRISx4	TRISx3	TRISx2	TRISx1	TRISx0
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 7-0 **TRISx<7:0>**: PORTx Tri-State Control bit
 1 = PORTx pin configured as an input (tri-stated)
 0 = PORTx pin configured as an output

Note 1: Register description for TRISA, TRISB, TRISC and TRISD.

Nom i Cognoms: _____

3. Omple la taula amb els valors resultants dels registres després d'executar aquest codi: **(1 PUNT)**

Dada1 equ 05h
Dada2 equ 019h
CLRF 00, A
MOVLW Dada1
MOVWF 01, A
BTFSC 01, 0, A
INCF 00, F, A
BTFSC 00, 1, A
INCF 00, F, A
BTFSC 00, 2, A
INCF 00, F, A
BTFSC 00, 3, A
INCF 00, F, A

	Valor inicial	Valor final
WREG	AAh	
BSR	02h	
000h	0Eh	
001h	01h	
002h	45h	
018h	89h	
019h	4Ah	
020h	32h	

4. Quants bytes ocuparà el programa a la memòria de programa? **(0,5 PUNTS)**
5. Calcula el temps que triga a executar-se la següent funció, des de que entra fins que retorna al codi que la crida (recordeu: RETURN 2 cicles, single WORD; BRA 2 CICLES, single WORD) amb un Fosc=20 MHz **(2 PUNTS)**

CopiaBank: LFSR 000h, FSR0
LFSR 100h, FSR0
Etq: MOVF POSTINC0, W
MOVWF POSTINC1
BTFSC FSR0H, 0
RETURN
BRA Etq

Si ens adonem que el segon LFSR posa al punter 0 el valor 100h (256), veiem que mai s'executa el bucle ja que el BTFSC no fa skip, per tant el codi triga:

TABLE 25-2: PIC18(L)F2X/4XK22 INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes	
			MSb		LSb				
BYTE-ORIENTED OPERATIONS									
ADDWF	f, d, a	Add WREG and f	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	1, 2
ADDWFC	f, d, a	Add WREG and CARRY bit to f	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	AND WREG with f	1	0001	01da	ffff	ffff	Z, N	1, 2
CLRF	f, a	Clear f	1	0110	101a	ffff	ffff	Z	2
COMF	f, d, a	Complement f	1	0001	11da	ffff	ffff	Z, N	1, 2
CPFSEQ	f, a	Compare f with WREG, skip =	1 (2 or 3)	0110	001a	ffff	ffff	None	4
CPFSGT	f, a	Compare f with WREG, skip >	1 (2 or 3)	0110	010a	ffff	ffff	None	4
CPFSLT	f, a	Compare f with WREG, skip <	1 (2 or 3)	0110	000a	ffff	ffff	None	1, 2
DECf	f, d, a	Decrement f	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010	11da	ffff	ffff	None	1, 2, 3, 4
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100	11da	ffff	ffff	None	1, 2
INCF	f, d, a	Increment f	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011	11da	ffff	ffff	None	4
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100	10da	ffff	ffff	None	1, 2
IORWF	f, d, a	Inclusive OR WREG with f	1	0001	00da	ffff	ffff	Z, N	1, 2
MOVF	f, d, a	Move f	1	0101	00da	ffff	ffff	Z, N	1
MOVFF	f _s , f _d	Move f _s (source) to f _d (destination)	2	1100	ffff	ffff	ffff	None	
MOVWF	f, a	Move WREG to f	1	0110	111a	ffff	ffff	None	
MULWF	f, a	Multiply WREG with f	1	0000	001a	ffff	ffff	None	1, 2
NEGF	f, a	Negate f	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	
RLCF	f, d, a	Rotate Left f through Carry	1	0011	01da	ffff	ffff	C, Z, N	1, 2
RLNCF	f, d, a	Rotate Left f (No Carry)	1	0100	01da	ffff	ffff	Z, N	
RRCF	f, d, a	Rotate Right f through Carry	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	0100	00da	ffff	ffff	Z, N	
SETF	f, a	Set f	1	0110	100a	ffff	ffff	None	1, 2
SUBFWB	f, d, a	Subtract f from WREG with borrow	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	
SUBWF	f, d, a	Subtract WREG from f	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	1, 2
SUBWFB	f, d, a	Subtract WREG from f with borrow	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	
SWAPF	f, d, a	Swap nibbles in f	1	0011	10da	ffff	ffff	None	4
TSTFSZ	f, a	Test f, skip if 0	1 (2 or 3)	0110	011a	ffff	ffff	None	1, 2
XORWF	f, d, a	Exclusive OR WREG with f	1	0001	10da	ffff	ffff	Z, N	
LITERAL OPERATIONS									
ADDLW	k	Add literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW	k	AND literal with WREG	1	0000	1011	kkkk	kkkk	Z, N	
IORLW	k	Inclusive OR literal with WREG	1	0000	1001	kkkk	kkkk	Z, N	
LFSR	f, k	Move literal (12-bit) 2nd word to FSR(f) 1st word	2	1110	1110	00ff	kkkk	None	
MOVLB	k	Move literal to BSR<3:0>	1	0000	0001	0000	kkkk	None	
MOVLW	k	Move literal to WREG	1	0000	1110	kkkk	kkkk	None	
MULLW	k	Multiply literal with WREG	1	0000	1101	kkkk	kkkk	None	
RETLW	k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
SUBLW	k	Subtract WREG from literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW	k	Exclusive OR literal with WREG	1	0000	1010	kkkk	kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS									
TBLRD*		Table Read	2	0000	0000	0000	1000	None	
TBLRD*+		Table Read with post-increment		0000	0000	0000	1001	None	
TBLRD*-		Table Read with post-decrement		0000	0000	0000	1010	None	
TBLRD*+		Table Read with pre-increment		0000	0000	0000	1011	None	
TBLWT*		Table Write	2	0000	0000	0000	1100	None	
TBLWT*+		Table Write with post-increment		0000	0000	0000	1101	None	
TBLWT*-		Table Write with post-decrement		0000	0000	0000	1110	None	
TBLWT*+		Table Write with pre-increment		0000	0000	0000	1111	None	
BIT-ORIENTED OPERATIONS									
BCF	f, b, a	Bit Clear f	1	1001	bbba	ffff	ffff	None	1, 2
BSF	f, b, a	Bit Set f	1	1000	bbba	ffff	ffff	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	3, 4
BTG	f, b, a	Bit Toggle f	1	0111	bbba	ffff	ffff	None	1, 2

d = 0 for result destination to be WREG register
d = 1 for result destination to be file register (f)

Nom i Cognoms: _____

6. Completa el codi per garantir que si hi ha un canvi al PORTB es saltarà a una interrupció de baixa prioritat (RSI_LOW). **(1,5 PUNTS)**

Org 000h

GOTO Codi

Org 008h

GOTO RSI_HIGH

Org 018h

GOTO RSI_LOW

Codi:

```
MOVLB    0Fh    // Triem el Bank F de registres (és on
CLRFB    ANSELB,B // hi ha ANSELB i TRISB)
SETFB    TRISB,B // Posem PORTB com a entrada digital
```

Final:

```
BRA      End
End
```

Final

// El programa acaba aquí

7. Hi ha algun error en el codi de la interrupció RSI_LOW que tenim aquí sota?, explica'l: **(1 punt)**

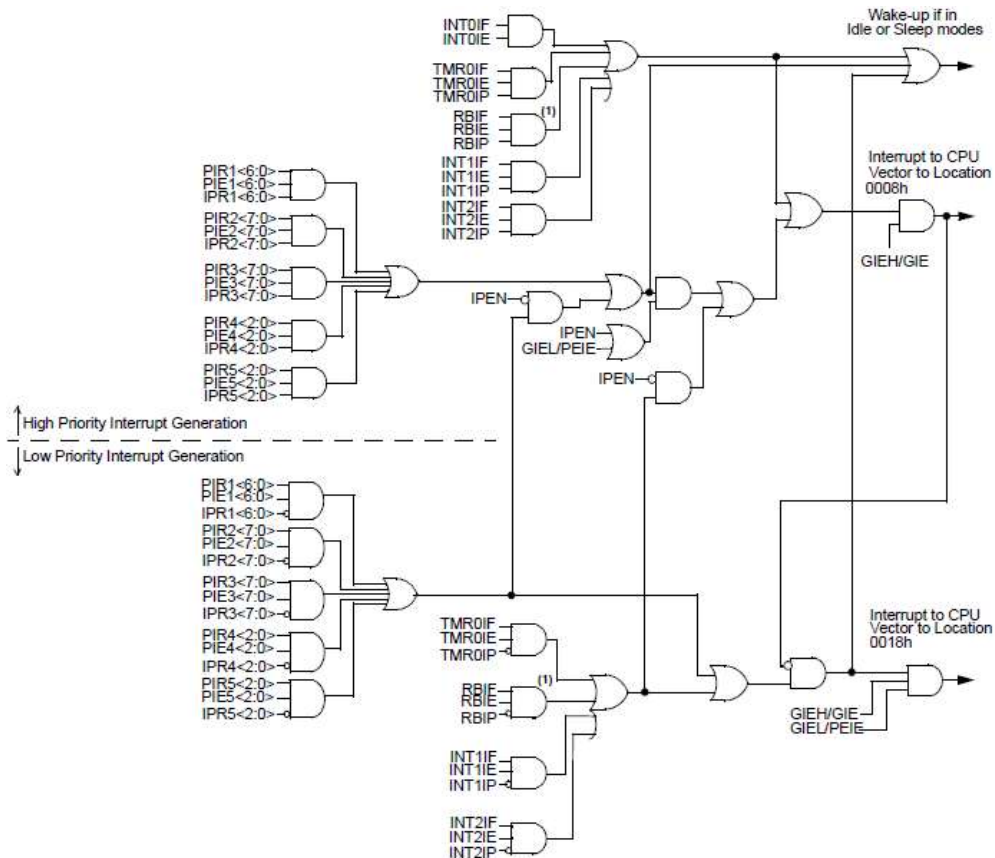
RSI_LOW:

```
BTFSC    INTCON, 0, A // Mirem si el flag és del PORTB
RETFIE FAST
INCF     Counter, F, A // Comptem la interrupció
BCF      INTCON, 0, A // Borrem el flag perquè no reentri
RETFIE FAST           // Sortim de la interrupció
```


TABLE 9-1: REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
INTCON2	RBP _U	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP
IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP
IPR5	—	—	—	—	—	TMR6IP	TMR5IP	TMR4IP
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE
PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE
PIE5	—	—	—	—	—	TMR6IE	TMR5IE	TMR4IE
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF
PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF
PIR5	—	—	—	—	—	TMR6IF	TMR5IF	TMR4IF
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RCON	IPEN	SBOREN	—	RI	TO	PD	POR	BOR

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for Interrupts.

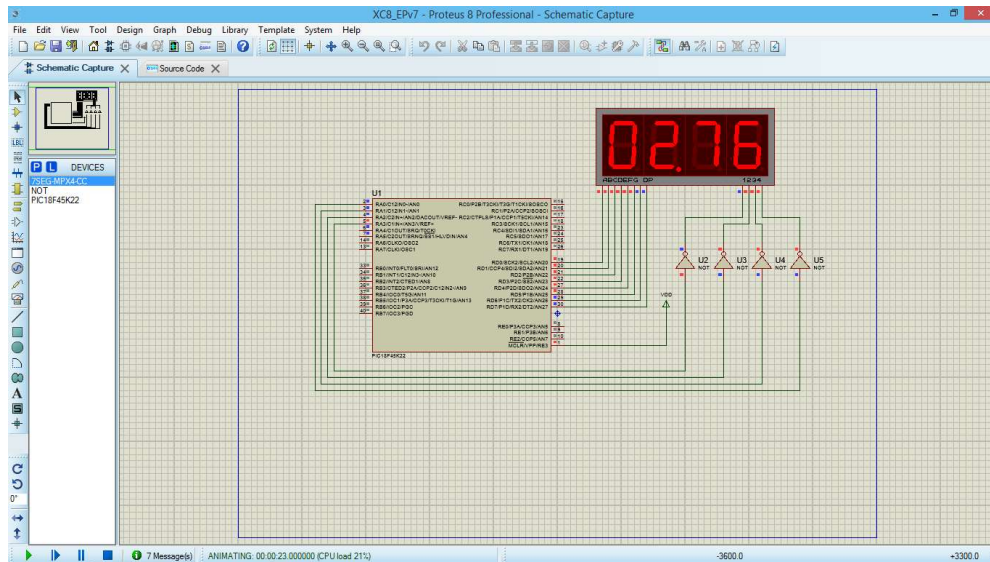


Nom i Cognoms: _____

PREGUNTA CONTROL DE LABORATORI

(els repetidors amb LAB convalidat no l'heu de fer)

A la pràctica de laboratori del display 7 segments (veure figura), heu fet una funció per pintar un nombre amb decimals a la pantalla. Indica si són certes (C) o falses (F) les afirmacions següents: **(+1,25 encert, -0,75 fallada)**



- ☐ El punt decimal s'ha de posar sempre i en cas que el nombre sigui enter s'ha de posar després de la última xifra.
- ☐ Es pot fer (canviant el codi) que uns dígit brillin més que altres.
- ☐ Aquesta pràctica va ser la última feta en Assembler del PIC18F45K22.
- ☐ Si volem fer sortir un dígit pels quatre displays alhora, podem treure els seus bits pel PORTD i activar 4 uns al PORTA.
- ☐ Si estem pintant el nombre '2019' i parem el programa per debug o aquest acaba amb un "while (1);", quedarà pintat el nombre a la pantalla.
- ☐ Si un cop tenim el codi acabat canviem l'oscil·lador del xip de 8MHz a 16 MHz utilitzarem la meitat de dígit per representar el mateix nombre.
- ☐ A la placa de laboratori (circuit real) també hi ha portes NOT al control de l'activació dels 7-segments.
- ☐ Es podria dedicar el primer 7-segments (per software) a representar el signe del nombre i així pintar negatius també.