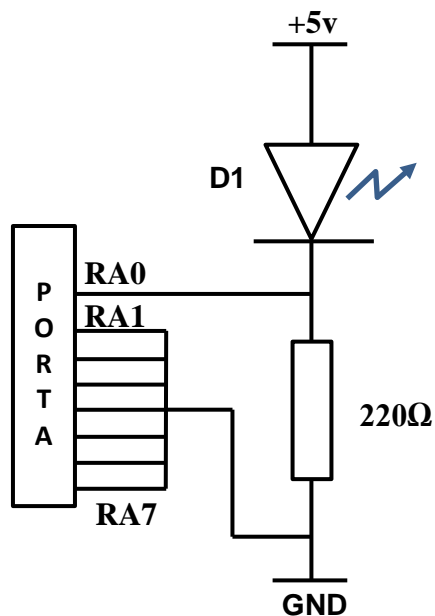


Nom i Cognoms: _____

Contesteu en aquest mateix full en l'espai reservat de cada pregunta. Raoneu totes les respostes. Les respostes que no vagin acompanyades d'una mínima explicació no s'admetran.

1.) En relació al circuit esquemàtic de la figura següent, es demana:

- Quins serien els valors a escriure en els registres PORTA i TRISA que apagarien el LED D1? (1 Punt)
- Quins serien els valors MÉS ADEQUATS a escriure en els registres PORTA i TRISA per encendre el LED D1? (1 Punt)
- Considerant les limitacions electròniques dels pins de sortida, proveu si realment es podrà apagar el LED D1. Els paràmetres dels pins de sortida són : $V_{OH,min} = 4.7\text{ V}$, $V_{OL,max} = 0.7\text{ V}$, $I_{LEAKAGE} = 0.2\text{ }\mu\text{A}$, $I_{OH,max} = 25\text{ mA}$, $I_{OL,max} = 25\text{ mA}$ i la tensió llindar del LED és de 2,3 V. (1.5 Punts)



Comentaris previs: Els pins RA1 a RA7 estan connectats directament a zero volts. Aquests pins no haurien de ser configurats de sortida doncs qualsevol intent de 'escriure' un voltatge diferent a exactament zero volts (ni que sigui per poc) provocarà un corrent innecessari. L'únic pin que es podria configurar d'entrada o de sortida és el pin RA0.

- Si el voltatge en RA0 és superior a 2,7v (5v-2,3v) el LED s'apagarà, doncs no hi haurà prou diferència de tensió en els borns del díode. Llavors cal configurar TRISA = 1111110b = FEh i escriure PORTA = xxxxxx1b = 01h.
- La manera més adient d'encendre el led és configurant RA0 d'entrada (el pin associat queda en alta impedància, desconnectat). En aquestes condicions la sola branca de la resistència és suficient per encendre el led. L'opció de posar RA0 = 0 no és correcta degut a que no hi hauria una resistència que limités a valors adequats el corrent I_{OL} .
- Amb RA0 = 1, tenim que el voltatge en el pin RA0 apagaria el led però el corrent I_{OH} hauria de ser suficient per alimentar la resistència. Per alimentar a la resistència i

mantenir un voltatge mínim V_{OHMIN} caldria un corrent de sortida de $I_{OH} = V_{OHMIN}/220 = 4,7/220 = 21,3$ mA. Segons les especificacions, aquest corrent és inferior al màxim corrent de sortida que el pin pot proporcionar (que és de 25 mA), en conseqüència el led s'apagaria.

- 2.) Justifiqueu el motiu per el qual la instrucció **movff** (*move file to file*) és una *double-word instruction*. (0.5 Punt)

La instrucció movff mou un valor d'un registre (file) a un altre registre utilitzant adreces absolutes. En l'arquitectura del PIC18, per adreçar un registre de forma absoluta són necessaris 12bits, llavors com a mínim són necessaris 24b per codificar les adreces (sense comptar el codi d'operació). Per codificar més de 24b no és suficient amb un word (16b), cal utilitzar un dword (32b).

- 3.) En relació als següent codi. Quantes vegades s'executarà el bucle Loop1? Quants bytes en total ocupa el segment de codi? Feu les anotacions del que ocupa cada instrucció al costat esquerra de la mateixa línia i indiqueu la suma total (1 Punt).

```
0 RST code 0x0 ; La directiva code fa la mateixa funció que abans ORG
2 goto Start
0 PGM code
0 Start
1 movlw 0x0F
1 movwf count
1 clrf maxval
2 lfsr FSR0, 0x20
0 Loop1
1 movf maxval,TOWREG,ACCESSBANK
1 cpfsht INDF0
1 bra next
2 movff INDF0,maxval
0 next
1 incf FSR0L,TOFILE;
1 decf count,TOFILE,ACCESSBANK
1 bz Loop1
```

Total ROM [bytes]: 30 B

El Loop1 només s'executa un sol cop, doncs al decrementar count (que val inicialment 15) no aixeca el flag de zero i no executa el salt condicional del branc (bz).

Nom i Cognoms: _____

Contesteu en aquest mateix full en l'espai reservat de cada pregunta. Raoneu totes les respostes. Les respostes que no vagin acompanyades d'una mínima explicació no s'admetran.

- 4.) Si el cristall de quars del PIC18F4550 és de 12 MHz, quant de temps passarà com a mínim des de que el bit 0 del port B es posa a 1, fins que el port D es posi a 0xFF ? (1,5 Punts)

...		
bsf	TRISB,0,0	
...		//suposem que aquí el bit B0 = 0 encara
loop: btfss	PORTB,0,0	
bra	loop	
nop		
clrf	TRISD,0	
setf	PORTD,0	

La instrucció bsf TRISB,0,0 fa que el pin 0 del port B (B0) sigui una entrada. Abans del loop sabem que el bit B0 estarà a 0, per tant en algun moment dins del bucle ens posaran el pin a 1. Mentre això no passi, el bucle anirà executant:

...							
Fetch btfss	Exe btfss						
	Fetch bra	Exe bra					
		Fetch nop	----				
			Fetch btfss	Exe btfss			
				Fetch bra	Exe bra		
					Fetch nop	----	
						Fetch btfss	Exe btfss
							...

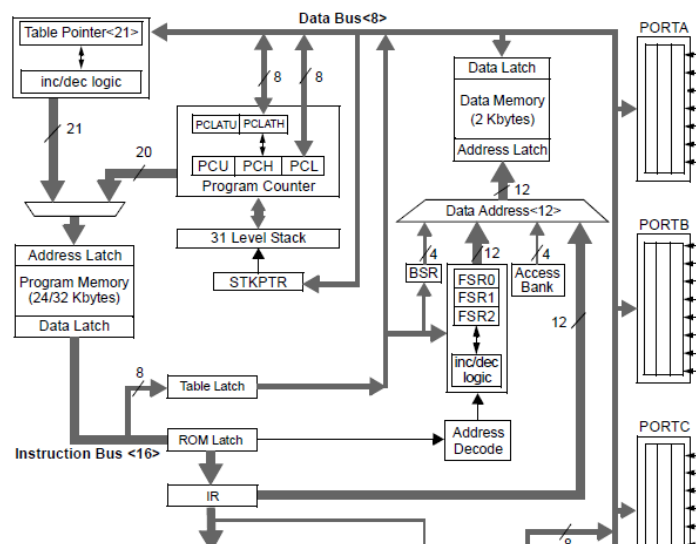
Com que ens demanen el temps mínim que passarà, hem de buscar el millor cas, que és que el pin B0 es posi a 1 just abans del Exe del btfss. En aquest cas s'executarà el següent (mireu que es fa skip del bra i que el Port D es posa a 0xFF a l'executar el setf) :

PIN B0=1					
Cicle Instr 0	Cicle Instr 1	Cicle Instr 2	Cicle Instr 3	Cicle Instr 4	Cicle Instr 5
Fetch btfss	Exe btfss				
	Fetch bra	-----			
		Fetch nop	Exe nop		
			Fetch clrf	Exe clrf	
				Fetch setf	Exe setf
					PORTD=0xFF

Ens calen doncs 5 cicles d'instrucció. Com que cada cicle d'instrucció necessita 4 tics del rellotge seran 20 tics de rellotge. Si cada tic de rellotge dura $1/12\text{MHz} = 83,3 \text{ ns}$, el temps total serà $20 \times 83,3 \text{ ns} = 1666,6 \text{ ns} = 1,6 \text{ us}$. (1,6 microsegons)

5.) Fixeu-vos en la següent figura en el bloc que genera les adreces per la memòria de dades (Data Address).

- Per quin motiu arriben fins a 12 bits de l'instruction register? (1 Punt)
Per les instruccions que contenen adreces completes de 12 bits, com ara el movff.
- Per quin motiu arriben 12 bits del bloc dels FSR ? (0,75 Punts)
Perquè els File Select Register (que fan les funcions de punter o mode indirecte) també contenen adreces completes de 12 bits.
- Quan es fan servir els 4 bits de l'access o del BSR, d'on vénen els bits que falten? (0,75 punts)
Els 8 bits restants venen de l'Instruction Register, per les instruccions més comunes, com el ADD, SUB, etc.



6.) Indiqueu els valors finals del WREG i la posició 0 i 1 del bank 0 després d'executar el següent codi (1 punt).

R0 equ 0		// la etiqueta R0 val 0
R1 equ 1		// la etiqueta R1 val 1
movlw	0x12	// posem el valor 12 hexa al WREG
movwf	R0, 0	// carreguem el 12h a la posició 0 de memòria
		// usant l'access bank (per tant bank 0)
movlw	0x19	// posem el valor 19 hexa al WREG
movwf	R1, 0	// carreguem el 19h a la posició 1 de memòria
		// i el WREG segueix amb el 19h
addwf	R0, 1, 0	// sumem el contingut del WREG (19h) amb el
		// de la posició 0 (12h) i ho guardem a la
		// posició 0 (perquè el bit D=1). Per tant a la
		// posició 0 i queda un 2B en hexa
addwf	R1, 1, 0	// El mateix per la posició 1 on hi queda 32 hexa

Per tant queda el WREG a 19h, la posició 0 a 2Bh i la posició 1 a 32h.