

Nom i Cognoms: _____

1) (3p) Suposant que el clock del sistema és de 8 MHz, calculeu:

- a) el temps d'execució del codi en negreta indicat a continuació.
- b) amb quins valors caldria inicialitzar COUNT1 i COUNT2 per a que el codi del *Loop* trigui aproximadament 4'5 mil·lisegons.

COUNT1 equ 0A ;

COUNT2 equ 0B ;

F equ 1

W equ 0

ACCESS equ 0

...

movlw 02h

movwf COUNT1, ACCESS

movlw 03h

movwf COUNT2, ACCESS

Loop: **decfsz COUNT1, F, ACCESS**

goto Loop

movlw 02h

movwf COUNT1, ACCESS

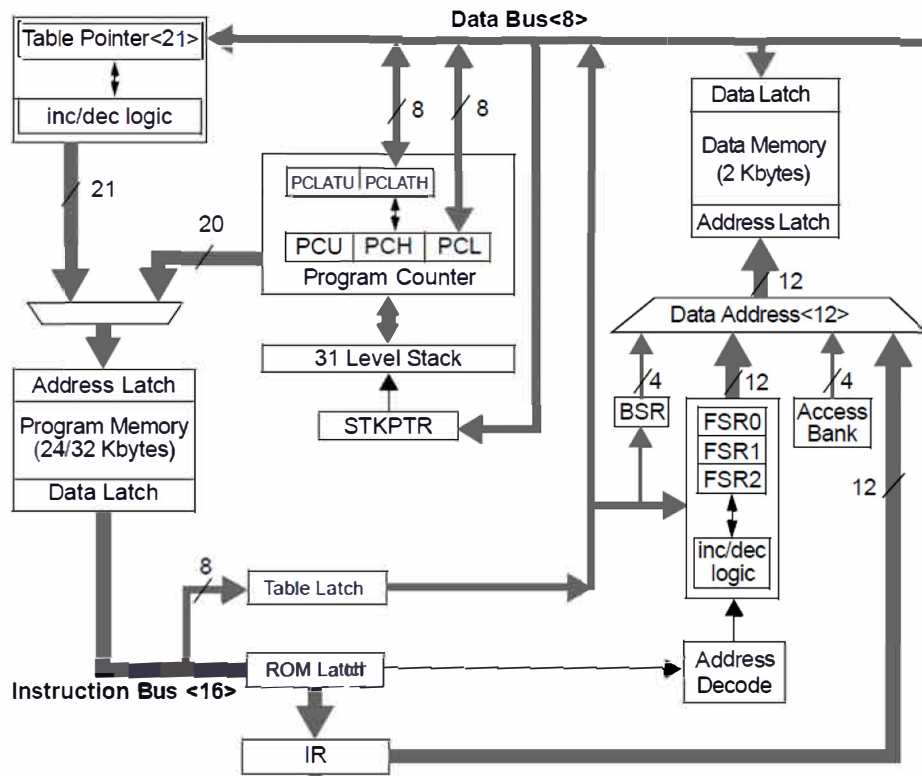
decfsz COUNT2, F, ACCESS

goto Loop

...

...

2) (1p) En el següent esquema simplificat del PIC18F, a la dreta podem observar un bloc amb la etiqueta **inc/dec logic**. Quin és el significat de la referida etiqueta i quina relació guarda amb l'adreçament a la RAM?



3) (1p) Si es volgués 'crear' una nova instrucció en llenguatge *assembler* per fer el intercanvi (swap) dels valors que hi han a adreça absoluta F i la adreça F+1:

a) quants bytes ocuparia com a mínim aquesta instrucció?

b) en quants cicles s'executaria?

Nom i Cognoms: _____

4) (1'5p.) Com queden els registres de la taula després d'executar el següent codi? Justifiqueu la resposta afegint un comentari després de cada instrucció.

```
movlb 1
movf 0,0,1
movwf 1,0
movf 0,0,0
movwf 0,1
movf 1,0,0
movwf 0,0
```

Bank	Address	Valor previ	Valor final
0	0	1	
0	1	2	
0	2	3	
1	0	4	
1	1	5	
1	2	6	

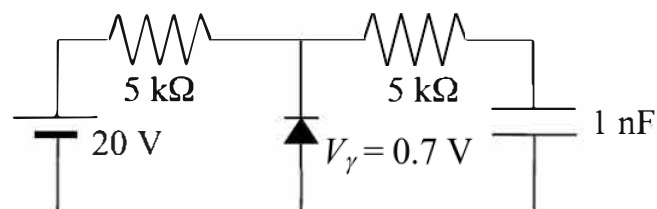
5) (1'5p.) Detalla tres exemples on la instrucció TSTFSZ provoqui un retard d'un cicle d'instrucció, dos cicles d'instrucció, o tres cicles d'instrucció.

1 CICLE:

2 CICLES:

3 CICLES:

6) (2p.) Del circuit de la figura



a) Trobeu la tensió en bornes del condensador un cop assoli el règim estacionari.

b) i si invertim el diode?

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	

d = 0 for result destination to be WREG register
 d = 1 for result destination to be file register (f)
 a = 0 to force Access Bank
 a = 1 for BSR to select bank
 f = 8-bit file register address