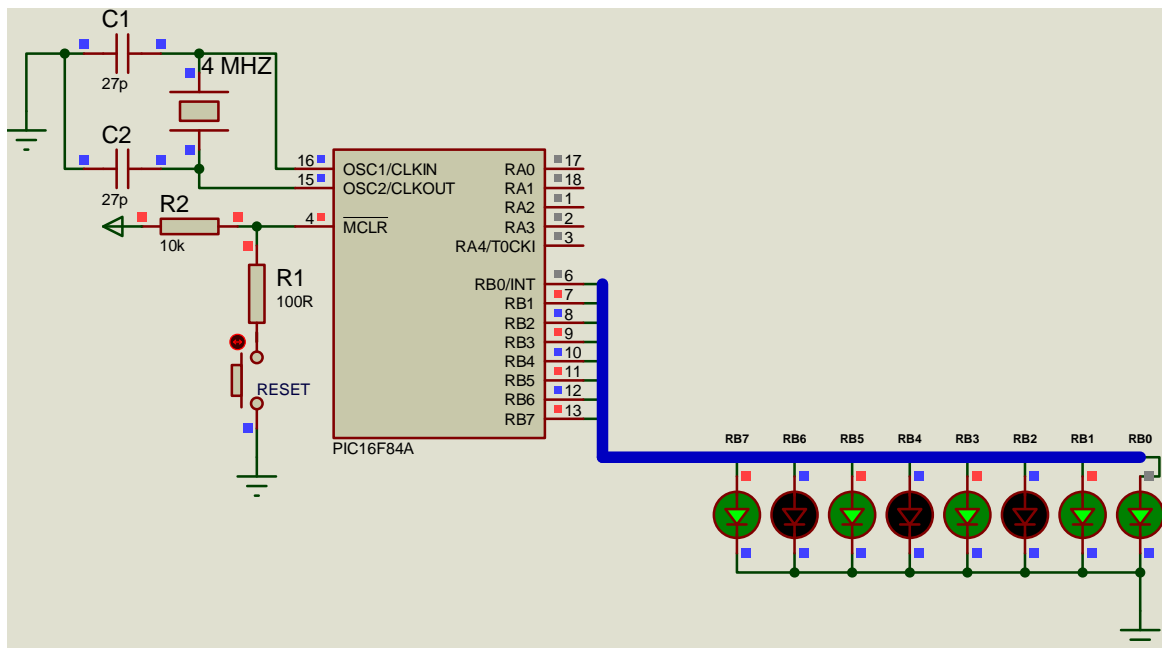


1. Proponer un circuito que tenga el puerto b como salida conectando 8 leds, realizando un programa que me encienda los bits 0, 1, 3, 5, 7.

*a. Diagrama de flujo*



*b. Circuito*



*c. Código .asm*

```
LIST      P=16F84A      ;comando que indica el pic usado.
RADIX     HEX          ;los valores en hexadecimal

STATUS   EQU          0x03      ;direcciona al registro de STATUS
PTOB     EQU          0x06      ;direcciona al puerto B

          ORG          0x00
          BSF          STATUS,5  ;seleccionar el banco 1
          MOVLW       0x00      ;carga w con el valor 00h
          MOVWF       PTOB      ;para habilitar el puerto B como salida
          BCF          STATUS,5  ;seleccionar el banco 0

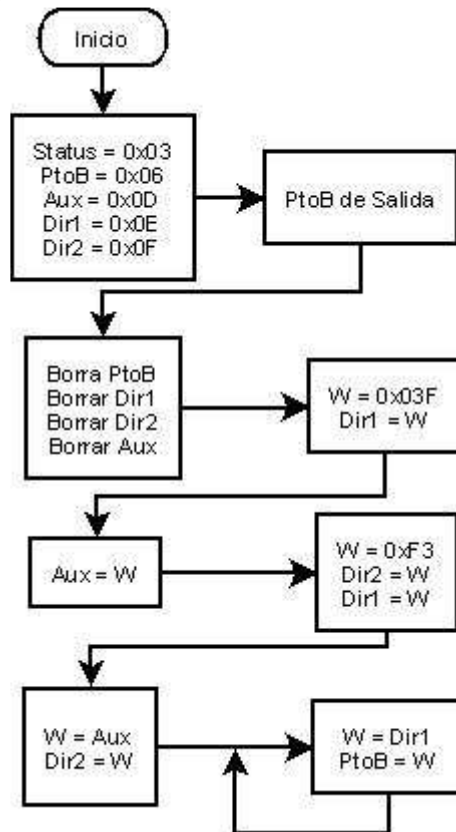
CICLO    CLRF         PTOB      ;limpia el puerto B
          MOVLW       0xAB      ;cargamos registro w con el numero ABh
          MOVWF       PTOB      ;enviamos el registro w al puerto B
          GOTO        CICLO     ;ir a ciclo
          END          ;fin del programa
```



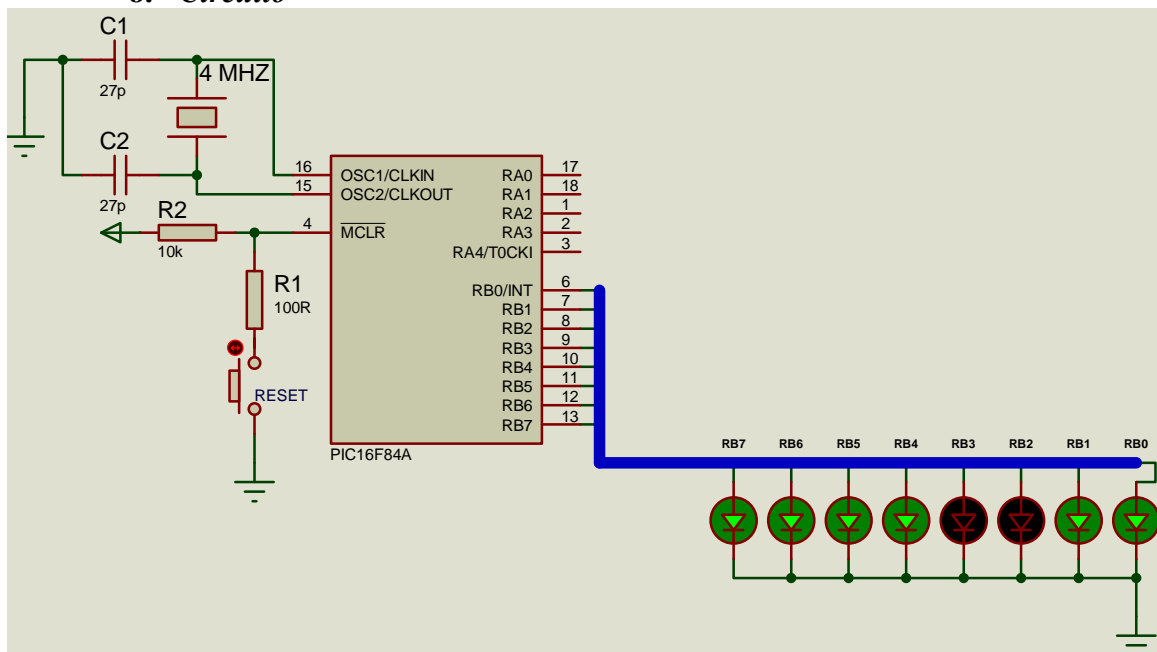
```
MOVWF    DIR1    ;almacenar el reg w en la dir 0Eh
MOVWF    AUX     ;almacenar el reg w en aux 0Dh
MOVLW    0xF3    ;cargar reg w con el numero F3h
MOVWF    DIR2    ;almacenar el reg w en la dir 0Fh
MOVWF    DIR1    ;almacenar el reg w en la dir 0Eh
MOVFW    AUX     ;cargar reg w con dato de dir 0Dh
MOVWF    DIR2    ;almacenar reg w en la dir 0Fh
END        ;fin del programa
```

3. Del ejercicio anterior, sabiendo los datos existentes en las localidades de memoria 0Eh y 0Fh, modifique el programa que transfiera el dato de las localidades de memoria 0Eh al puerto B.

**a. Diagrama de flujo**



**b. Circuito**



c. *Código .asm*

```

LIST      P=16F84A      ;comando que indica el pic usado.
RADIX     HEX          ;los valores en hexadecimal

STATUS   EQU           0x03      ;direcciona al registro de STATUS
PTOB     EQU           0x06      ;direcciona al puerto B
AUX      EQU           0x0D      ; direccion de mamoria 0DH
DIR1     EQU           0x0E      ;direccion de mamoria 0EH
DIR2     EQU           0x0F      ;direccion de mamoria 0FH

ORG      0x00

BSF      STATUS,5      ;seleccionar el banco 1
MOVLW   0x00          ;carga w con el valor 00h
MOVWF   PTOB          ;para habilitar el puerto B salida
BCF     STATUS,5      ;seleccionar el banco 0

ORG      0x00

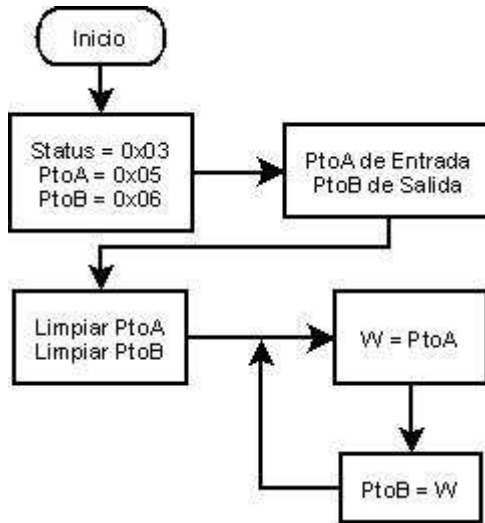
CLRF    DIR1          ;limpia la direccion 0EH
CLRF    DIR2          ;limpia la direccion 0FH
CLRF    AUX           ;limpia la direccion 0DH

MOVLW   0X3f          ;cargar el reg w con el num 3Fh
MOVWF   DIR1          ;almacenar el reg w en la dir 0Eh
MOVWF   AUX           ;almacenar el reg w en aux 0Dh
MOVLW   0xF3          ;cargar reg w con el numero F3h
MOVWF   DIR2          ;almacenar el reg w en la dir 0Fh
MOVWF   DIR1          ;almacenar el reg w en la dir 0Eh
MOVWF   AUX           ;cargar reg w con dato de dir 0Dh
MOVWF   DIR2          ;almacenar reg w en la dir 0Fh
CICLO   MOVFW         DIR1      ;cargar reg w con dato de dir 0Eh
MOVWF   PTOB         ;enviar el reg w al pto B
GOTO    CICLO        ;ir a ciclo
END     ;fin del programa

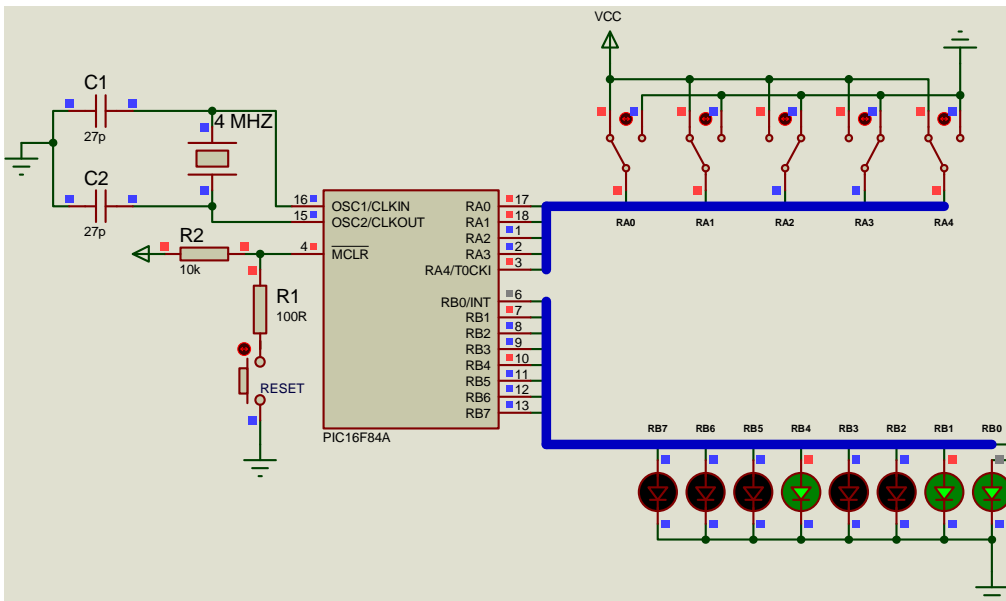
```

4. Proponer un circuito que tenga el puerto B como salida conectando 8 leds y el puerto A como entradas, realizar un programa que transfiera todas las entradas en el puerto A hacia el puerto B.

*a. Diagrama de flujo*



*b. Circuito*



c. *Código .asm*

```
LIST      P=16F84A      ;comando que indica el pic usado.
RADIX     HEX           ;los valores en hexadecimal

STATUS   EQU           0x03      ;direcciona al registro de STATUS
PTOA     EQU           0x05      ;direcciona al puerto A
PTOB     EQU           0X06      ;direcciona al puerto B

        ORG            0x00
        BSF            STATUS,5   ;seleccionar el banco 1
        MOVLW          0x1F       ;carga w con el valor 01h para habilitar
        MOVWF          PTOA       ; el puerto A como entrada
        MOVLW          0x00       ;carga w con el valor 00h para habilitar
        MOVWF          PTOB       ; el puerto B como salida
        BCF            STATUS,5   ;seleccionar el banco 0

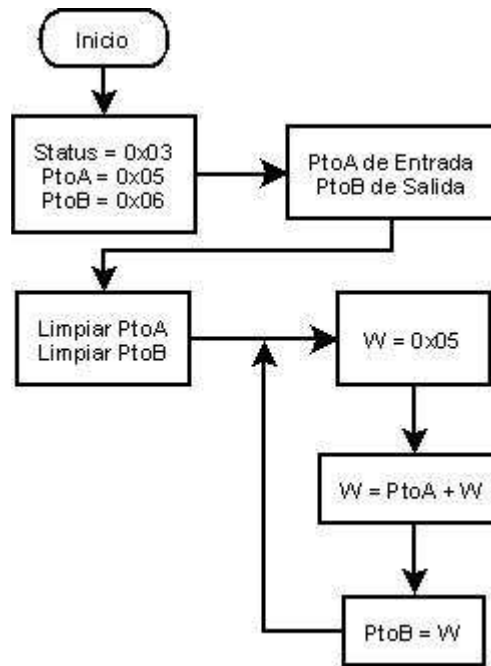
        CLRF          PTOA        ;limpia el puerto A
        CLRF          PTOB        ;limpia el puerto B

CICLO    MOVFW         PTOA        ;cargamos w con el valor del puerto A
        MOVWF          PTOB        ;enviamos el registro w al puerto B
        GOTO          CICLO       ;ir a la subrutina CICLO
        END            ;fin del programa
```

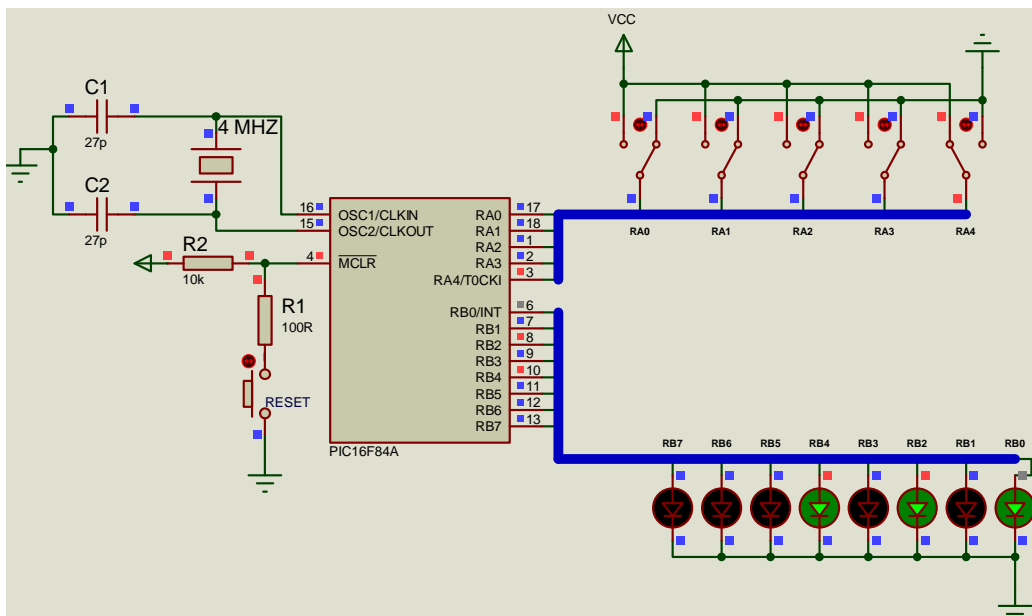


5. Proponer un circuito que tenga el puerto B como salida conectando 8 leds y el puerto A como entradas realizar un programa que permita hacer la operación de suma del puerto A (entrada) con el dato 05h y el resultado mostrarlo en el puerto B (salida).  
 $porta = porta + 05h$ .

*a. Diagrama de flujo*



*b. Circuito*



c. *Código .asm*

```
LIST      P=16F84A           ;comando que indica el pic usado.
RADIX     HEX                ;los valores en hexadecimal
STATUS   EQU      0x03      ;direcciona al registro de STATUS
PTOA     EQU      0x05      ;direcciona al puerto A
PTOB     EQU      0x06      ;direcciona al puerto B

ORG       0x00
BSF       STATUS,5          ;seleccionar el banco 1
MOVLW    0x1F              ;carga w con 1Fh
MOVWF    PTOA              ;habilitar puerto A como entrada
MOVLW    0x00              ;carga w con 00h
MOVWF    PTOB              ;habilitar puerto B como salida
BCF      STATUS,5          ;seleccionar el banco 0

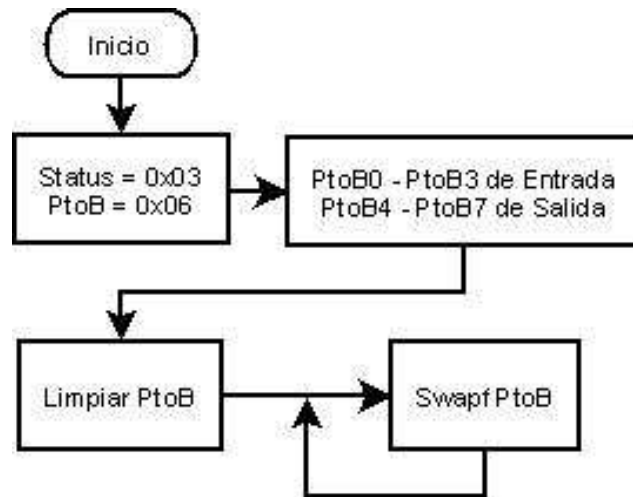
CLRF     PTOA              ;limpia el puerto A
CLRF     PTOB              ;limpia el puerto B

CICLO    MOVLW    0x05      ;cargamos w con el numero 05h
          ADDWF   PTOA,0    ;suma, W = PuertoA + W
          MOVWF   PTOB      ;enviamos el reg W al puerto B
          GOTO   CICLO      ;ir a CICLO

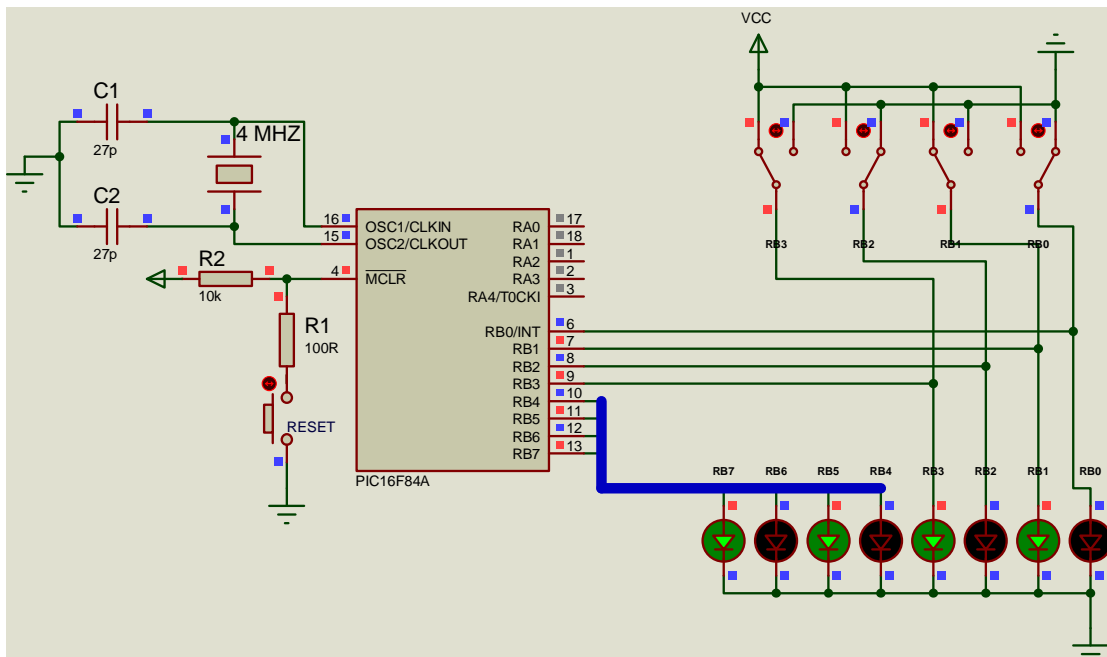
END      ;fin del programa
```

6. Realizar un programa que permita hacer la función espejo en el puerto B, el nibble menos significativo del puerto B será de entrada y el nibble más significativo será de salida, permitiendo de esta manera monitorear el estado del nibble de entrada en el nibble de salida.

*a. Diagrama de flujo*



*b. Circuito*



c. *Código .asm*

```

LIST      P=16F84A           ;comando que indica el pic usado.
RADIX     HEX                ;los valores en hexadecimal

STATUS   EQU      0x03       ;direcciona al registro de STATUS
PTOA     EQU      0x05       ;direcciona al puerto A
PTOB     EQU      0x06       ;direcciona al puerto B

                ORG      0x00
                BSF      STATUS,5 ;seleccionar el banco 1
                MOVLW   0x0F     ;carga w con el valor 0fh
                MOVWF   PTOB     ;habilitar puerto B en nibble
SALIDA y nibble ENTRADA
                BCF      STATUS,5 ;seleccionar el banco 0

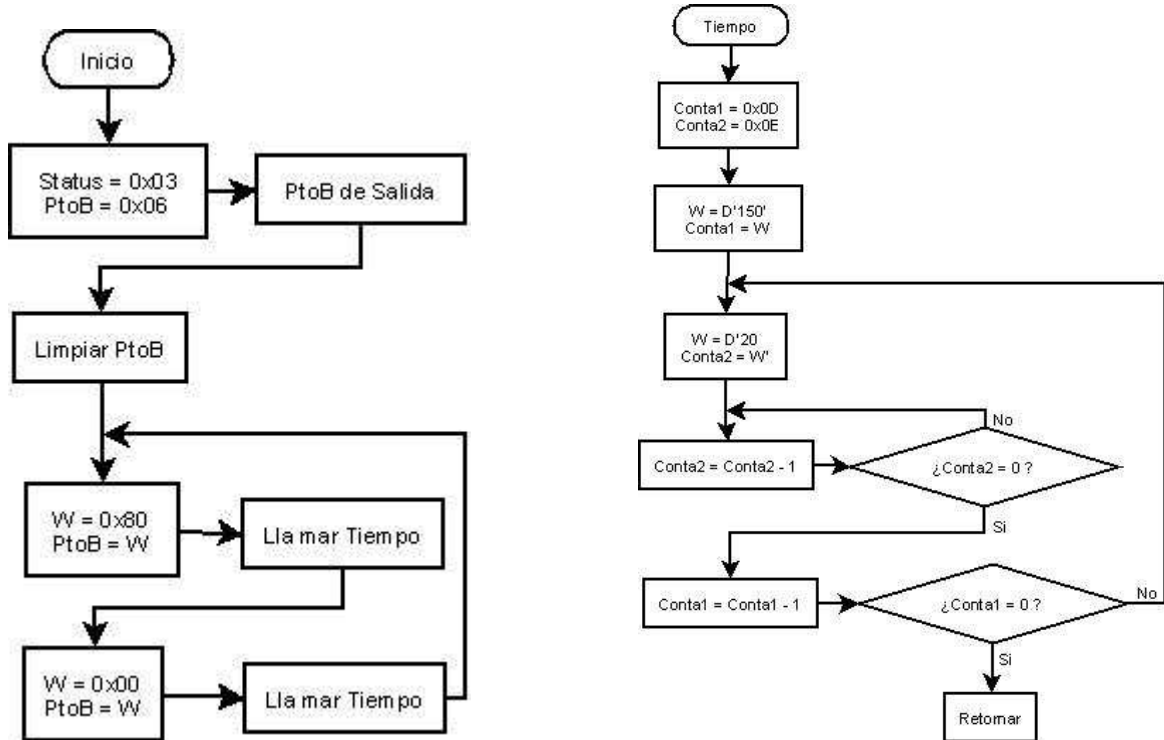
                CLRF    PTOB     ;limpia el puerto B
CICLO    SWAPF    PTOB,1       ;intercambia nibbles del puerto B
                GOTO    CICLO   ;ir a la subrutina CICLO

                END            ;fin del programa

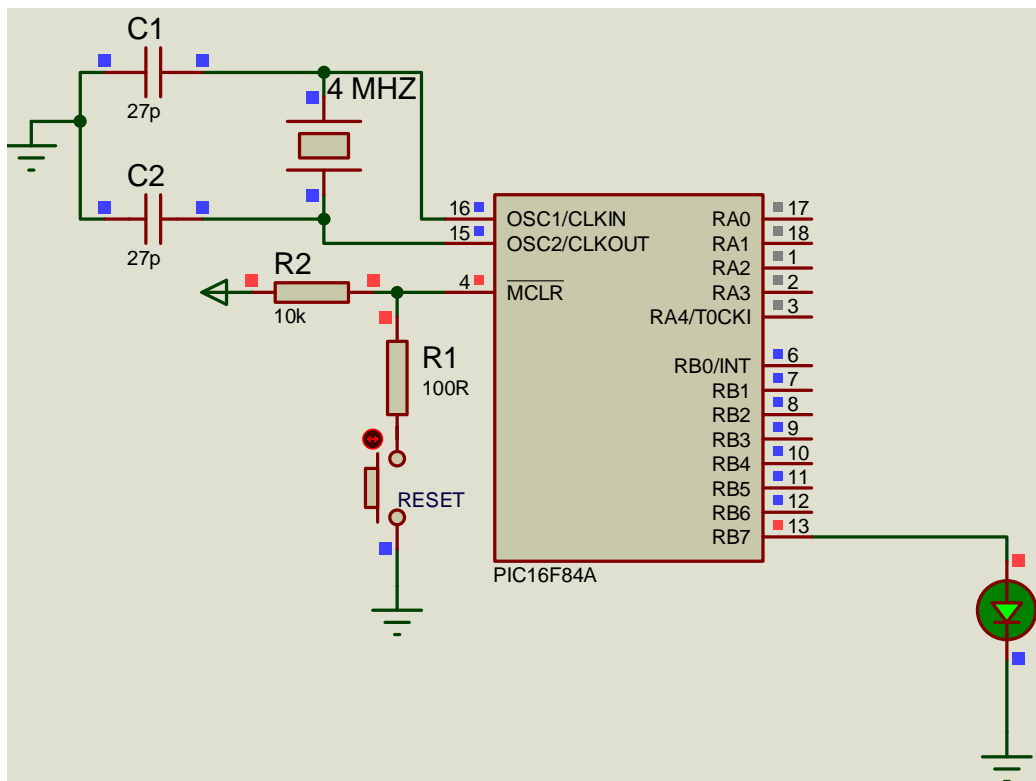
```

7. Realizar un programa que permita tener una señal cuadrada simétrica de 50 hz en el bit 7 del puerto B.

a. Diagrama de flujo



b. Circuito



*c. Código .asm*

```

LIST      P=16F84A      ;comando que indica el pic usado.
RADIX     HEX          ;los valores se representan en hexadecimal

STATUS   EQU          0x03      ;direcciona al registro de STATUS
PTOB     EQU          0x06      ;direcciona al puerto B
CONTA1   EQU          0x0D      ;hace refencia a la direccion 0Dh
CONTA2   EQU          0x0E      ;hace refencia a la direccion 0Eh

        ORG          0x00
        BSF          STATUS,5    ;seleccionar el banco 1
        MOVLW       0x00      ;carga w con 00h para habilitar puerto B salida
        MOVWF      PTOB      ;transfiere contenido de w en 0x05 del banco 1
        BCF          STATUS,5    ;seleccionar el banco 0

CICLO    CLRF        PTOB      ;limpia el puerto B
        MOVLW       0x80      ;cargamos w con el numero 80h
        MOVWF      PTOB      ;enviamos el registro w al puerto B
        CALL       TIEMPO     ;llamamos a la subrutina de tiempo TIEMPO
        MOVLW       0x00      ;cargamos w con el numero 80h
        MOVWF      PTOB      ;enviamos el registro w al puerto B
        CALL       TIEMPO     ;llamamos a la subrutina de tiempo TIEMPO
        GOTO      CICLO      ;ir a la subrutina CICLO

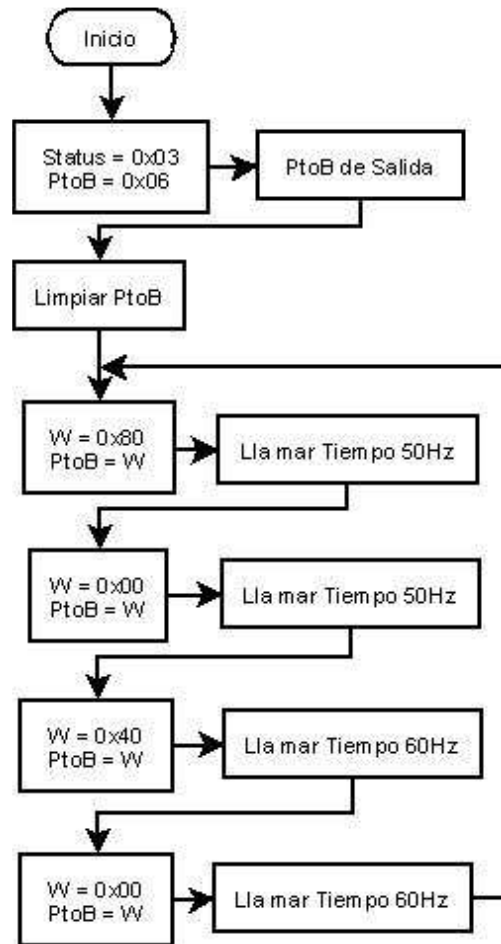
TIEMPO   MOVLW       D'150'    ;cargamos w con el numero decimal 5
        MOVWF      CONTA1     ;almacenamos w en la direccion 0Dh
LAZO2    MOVLW       D'20'     ;cargamos w con el numero decimal 5
        MOVWF      CONTA2     ;almacenamos w en la direccion 0Eh
LAZO1    DECFSZ     CONTA2,1    ;decrementa CONTA2 -1 y si el resultado es
igual a 0, salta 1 instruccion
        GOTO      LAZO1      ;ir a la subrutina LAZO1
        DECFSZ     CONTA1,1    ;decrementa CONTA1 -1 y si el resultado es
igual a 0, salta 1 instruccion
        GOTO      LAZO2      ;ir a la subrutina LAZO2
        RETURN      ;retorno de la instruccion CALL

        END                ;fin del programa

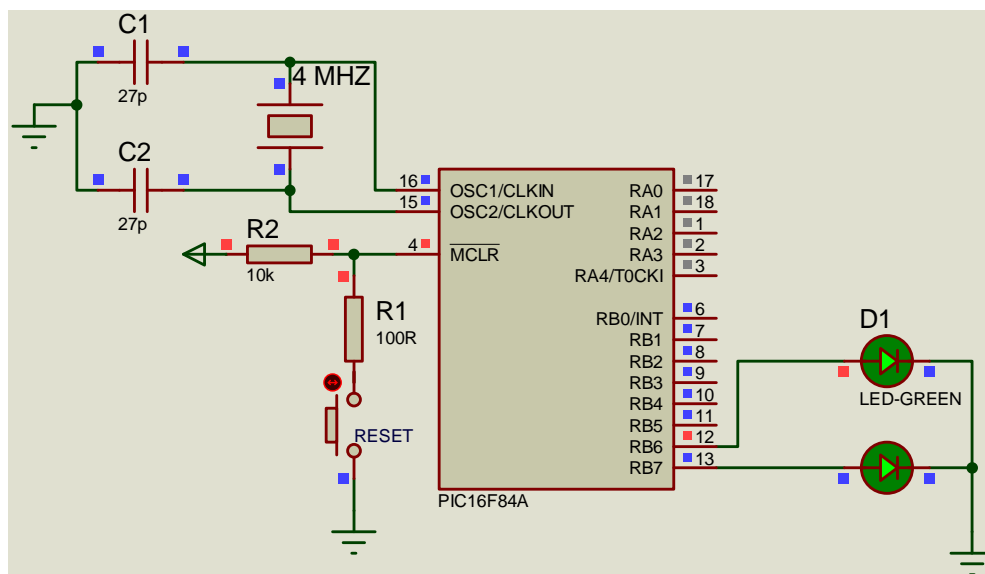
```

8. Realizar un programa que permita tener una señal cuadrada simétrica de 50 hz en el bit 7, una señal cuadrada simétrica de 60 hz en el bit 6 del puerto B.

*a. Diagrama de flujo*



*b. Circuito*



*c. Código .asm*

```
LIST      P=16F84
RADIX     HEX

W         EQU      0x00
F         EQU      0x01
EDO       EQU      0x03
PTAB      EQU      0x06
CONTA4    EQU      0x0C
CONTA1    EQU      0x0D
CONTA2    EQU      0x0E
CONTA3    EQU      0x0F

          ORG      0x00

INICIO    BSF      EDO,5
          MOVLW    b'00000000'
          MOVWF    PTAB
          BCF      EDO,5
```

-----INICIA PROGRAMA-----

```
BUCLE     MOLVW    0x80
          MOVWF    PTAB
          CALL     TEMPO
          MOVLW    0x00
          MOVWF    PTAB
          CALL     TEMPO

          MOVLW    0x40
          MOVWF    PTAB
          CALL     TEMPO1
          MOVLW    0x00
          MOVWF    PTAB
          CALL     TEMPO1
          GOTO     BUCLE
```

-----RUTINA DE TIEMPO-----

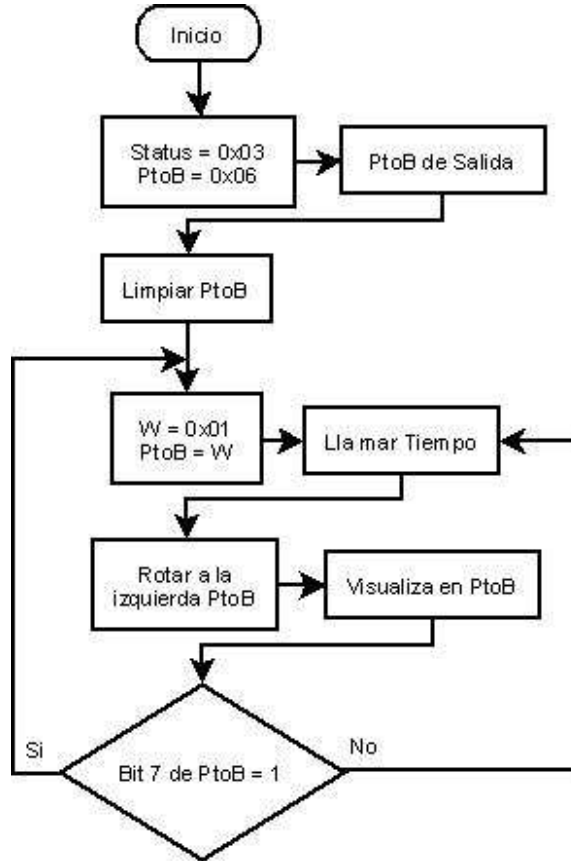
```
TEMPO     MOVLW    D'5'
          MOVWF    CONTA1
LAZO2     MOVLW    D'5'
          MOVWF    CONTA2
LAZO1     DECFSZ   CONTA2,1
          GOTO     LAZO1
          DECFSZ   CONTA1,1
          GOTO     LAZO2
          RETURN
```



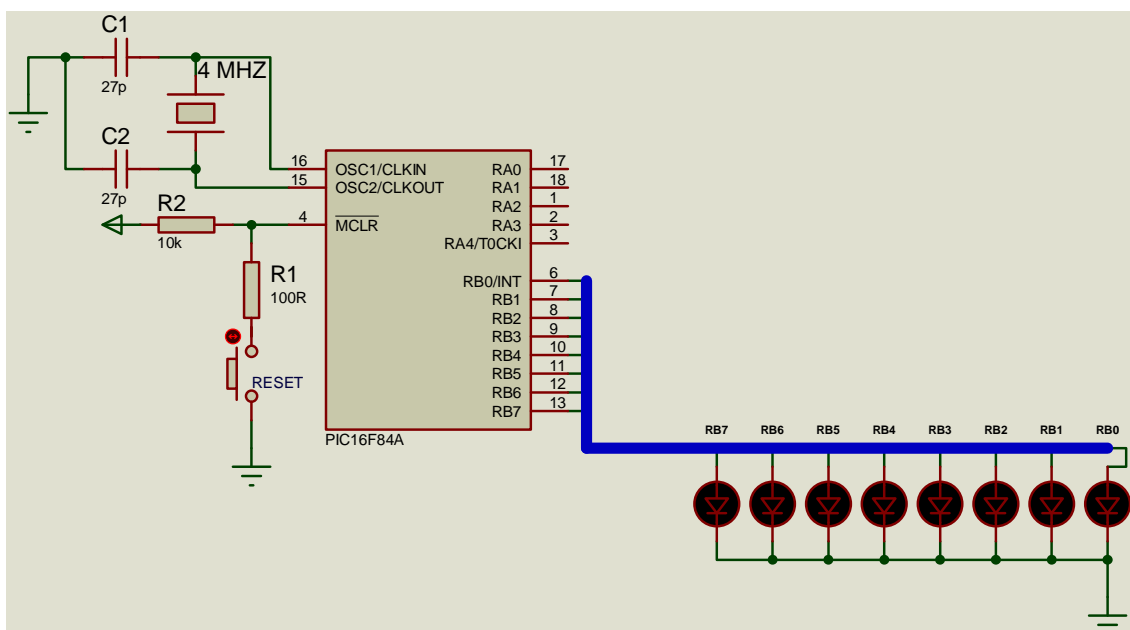
```
TEMPO1    MOVLW    D'6'  
          MOVWF    CONTA3  
LAZO4     MOVLW    D'6'  
          MOVWF    CONTA4  
LAZO3     DECFSZ   CONTA4,1  
          GOTO     LAZO3  
          DECFSZ   CONTA3,1  
          GOTO     LAZO4  
          RETURN  
  
          END
```

9. Proponer un circuito que tenga el puerto B como salida conectando 8 leds, realizar un programa que permita rotar el dato 01h en el puerto B de manera infinita. Controle la rotación para que el bit se rote de la posición del bit 0 al bit 7.

**a. Diagrama de flujo**



**b. Circuito**

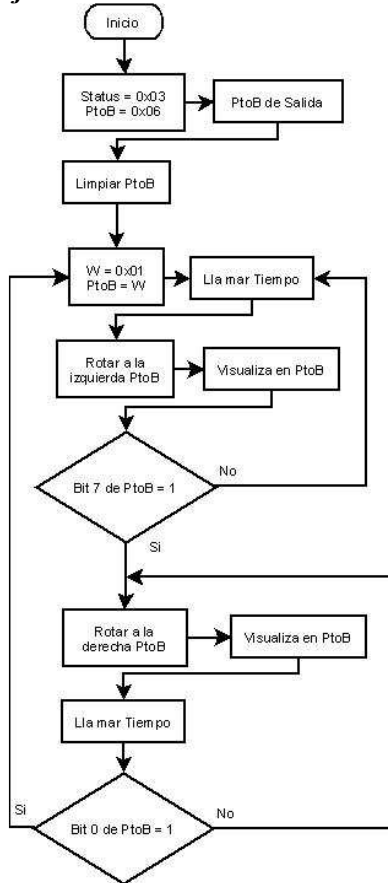


c. *Código .asm*

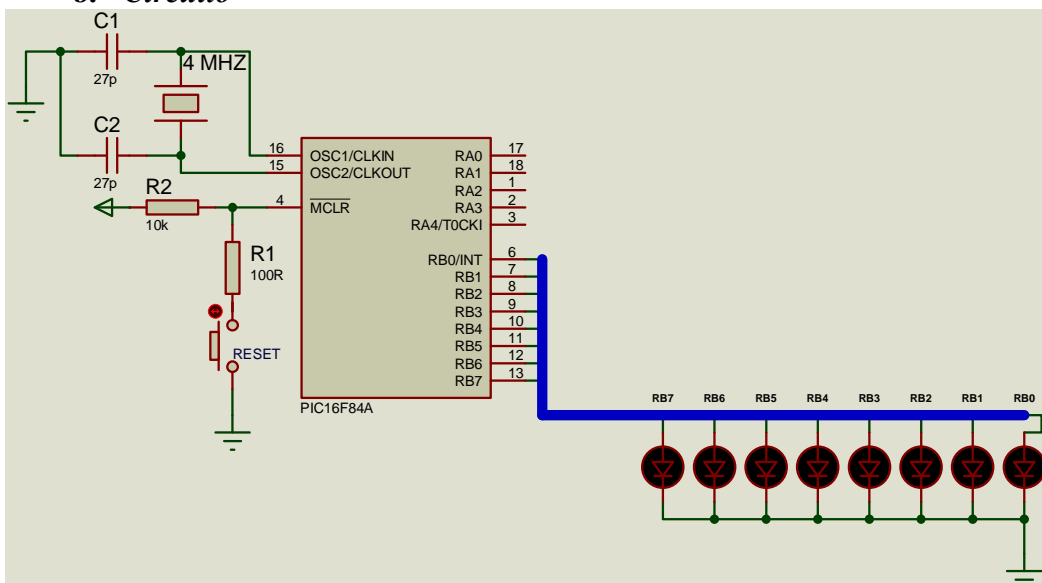
LIST	P=16F84A		;comando que indica el pic usado.
RADIX	HEX		;los valores se representan en hexadecimal
STATUS	EQU	0x03	;direcciona al registro de STATUS
PTOB	EQU	0x06	;direcciona al puerto B
CONT1	EQU	0x0D	;hace refencia a la direccion 0Dh
CONT2	EQU	0x0E	;hace refencia a la direccion 0Eh
	ORG	0x00	
	BSF	STATUS,5	;seleccionar el banco 1
	MOVLW	0x00	;carga w con 00h
	MOVWF	PTOB	;transfiere contenido de w en 0x05 del banco 1
	BCF	STATUS,5	;seleccionar el banco 0
	CLRF	PTOB	;limpia el puerto B
	BCF	STATUS,0	;PONEMOS ACARREO A CERO
INICIO	MOVLW	0x01	;cargar W con 0x01
	MOVWF	PTOB	;mandar w en puerto B
IZQ	CALL	TIEMPO	;llamamos subrutina de tiempo
	RLF	PTOB,1	;rotamos un 1 a la izquierda en el puerto B
	BTFSZ	PTOB,7	;compara si b7 del puerto B es igual a 0, salta si
es 1	GOTO	IZQ	;no b7=0 entonces ir a IZQ
	GOTO	INICIO	;si b7=1 entonces ir a INICIO
			;rutina de tiempo
TIEMPO	MOVLW	D'150'	;carga w con un valor decimal
	MOVWF	CONT1	;almacena w en CONT1
LAZO2	MOVLW	D'20'	;carga w con un valor decimal
	MOVWF	CONT2	;almacena w en CONT2
LAZO1	DECFSZ	CONT2,1	;decrementa 1 a CONT2, salta si CONT2 es
igual a 0	GOTO	LAZO1	;si CONT2 es diferente de 0. Ir a LAZO1
	DECFSZ	CONT1,1	;si CONT2 es igual a 0. Decrementa 1 a
CONT1, salta si CONT1 es igual a 0	GOTO	LAZO2	;si CONT1 es diferente de 0. Ir a LAZO2
	RETURN		;retorno de subrutina
	END		;fin del programa

10. Proponer un circuito que tenga el puerto B como salida conectando 8 leds, realizar un programa que permita rotar el dato 01h en el puerto B a la izquierda y cuando llegue al bit 7 se realice la rotación a la derecha, de la posición de bit 7 a la posición de bit 0 de forma infinita.

**a. Diagrama de flujo**



**b. Circuito**



### c. Código .asm

```

;+-----+
;|                                     |
;+-----+
LIST    p = 16f84                      ;usar el PIC16F84
radix   hex                             ;y la numeración hexadecimal
;+-----+
;|                                     |
;+-----+
estado  equ    0x03                      ;Le asignamos nombres a los registro
portb   equ    0x06                      ;encuentran
Aux1    equ    0x0C                      ;Direcciones de ocupados para
Aux2    equ    0x0D                      ;la subrutina de retardo
Aux3    equ    0x0E
;+-----+
;|   CONFIGURACIÓN DEL PUERTO B COMO SALIDA   |
;+-----+
ORG     0x00                            ;Origen del programa, siempre empieza aqui
;cuando se establece reset
CBF     estado,5                          ;Nos pasamos al banco 1
CLRF    portb
;de configuracion del puerto B, será de salida
BCF     estado,5                          ;Nos regresamos al banco 0
;+-----+
;|   AQUI ENCIENDE Y APAGA EL LED   |
;+-----+
LED)    MOVLW   0x01                      ;Pone un 1 en el pin INT/RB0 (enciende el
LED)
MOVWF   portb
BCF     estado,0
CALL    Retardo                          ;Llamar a la subrutina de retardo
rotari  RLF     portb,1                    ;rotar a la izquierda portb
CALL    Retardo                          ;Llamar a la subrutina de retardo
rotari  BTFSS   portb,7                    ;el b 7 de portb = 1?
rotard  GOTO    rotari                     ;No, ir a rotari
rotard  RRF     portb,1                    ;Si, rotar a la derecha portb
CALL    Retardo                          ;llamar a la subrutina de retardo
rotard  BTFSS   portb,0                    ;El b 0 de portb = 1?
rotard  GOTO    rotard                     ;No, ir a ratard
rotard  GOTO    rotari                     ;Si, ir a rotari
;+-----+
;|                                     |
;+-----+
RUTINA DE RETARDO
;+-----+
Retardo MOVLW   0x0F                      ;239
MOVWF   Aux1
Uno     MOVLW   0x0E                      ;232
MOVWF   Aux2

```

Dos	CLRWDT		;Limpiar el reloj del Perro guardian
Tres	GOTO	Cuatro	
Cuatro	GOTO	Cinco	
Cinco	CLRWDT		
	DECFSZ	Aux2,f	;Decrementar Aux3 -> Aux3-1
	GOTO	Dos	
	DECFSZ	Aux1,f	;Decrementar Aux2 -> Aux2-1
	GOTO	Uno	
Seis	GOTO	Siete	
Siete	GOTO	Ocho	
Ocho	GOTO	Nueve	
Nueve	CLRWDT		
	RETURN		
	END		;Fin del programa



```

BCF          ESTADO,5 ;Volver al banco 0
;+-----+
;|          PROGRAMA PRINCIPAL
;+-----+
Ciclo1      CLRf          cuenta      ;Inicializar el contador
Ciclo2      MOVf          cuenta,w     ;Pasar a W el contenido de contador
           CALL          Tabla        ;Llamar a la tabla dependiendo de W
           MOVWF         PUERTOB      ;Mandar al Puerto B el valor obtenido
           CALL          Retardo       ;Llamar la subrutina de retardo
           INCf          cuenta,f     ;Incrementar al contador
           MOVLW         0x10         ;Mover b'00010000' a W 16 decimal
           XORWF         cuenta,w     ;Hacer ope XOR cuenta con W
           BTFS         ESTADO,2     ;¿El contador es igual a 16?
           GOTO          Ciclo2      ;No, seguir con el cciclo
           GOTO          Ciclo1      ;Si, ir a inicilizar el ciclo
;+-----+
;|          RUTINA DE RETARDO
;+-----+
Retardo     MOVLW         0x0D         ;14
           MOVWF         Aux1
Uno         MOVLW         0x48         ;72
           MOVWF         Aux2
Dos        MOVLW         0x7A         ;0xF7 ;247
           MOVWF         Aux3
Tres       CLRWDT         ;Limpiar el reloj del Perro guardian
           DECFSZ        Aux3,f       ;Decrementar Aux3 -> Aux3-1
           GOTO          Tres
           DECFSZ        Aux2,f       ;Decrementar Aux2 -> Aux2-1
           GOTO          Dos
           DECFSZ        Aux1,f       ;Decrementar Aux1 -> Aux1-1
           GOTO          Uno
           GOTO          Sig
Sig        CLRWDT         ;Limpiar el reloj del Perro guardian
           RETURN
;+-----+
;|          TABLA DE LA SECUENCIA
;+-----+
Tabla      ADDWF         PCL,f
           RETLW         B'10000000' ; 0x80
           RETLW         B'11000000' ; 0xC0
           RETLW         B'11100000' ; 0xE0
           RETLW         B'11110000' ; 0xF0
           RETLW         B'11111000' ; 0xF8
           RETLW         B'11111100' ; 0xFC
           RETLW         B'11111110' ; 0xFE
           RETLW         B'11111111' ; 0xFF
           RETLW         B'01111111' ; 0x7F
           RETLW         B'00111111' ; 0x3F

```

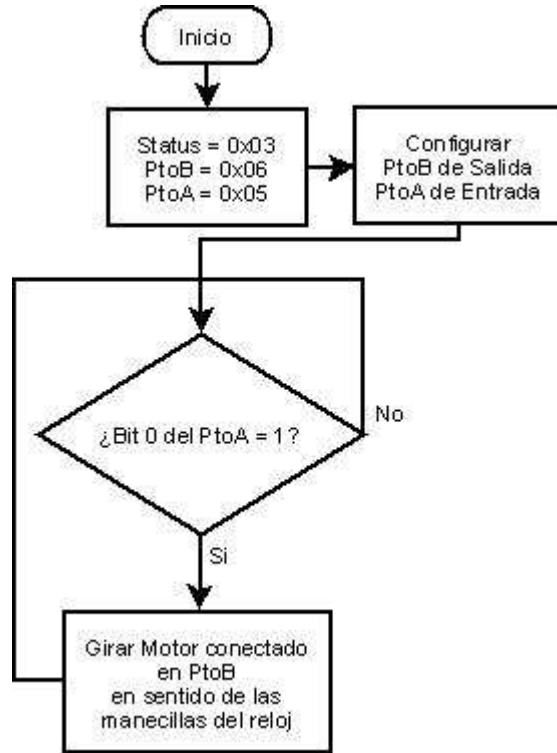


```
RETLW    B'00011111' ; 0x1F
RETLW    B'00001111' ; 0x0F
RETLW    B'00000111' ; 0x07
RETLW    B'00000011' ; 0x03
RETLW    B'00000001' ; 0x01
RETLW    B'00000000' ; 0x00

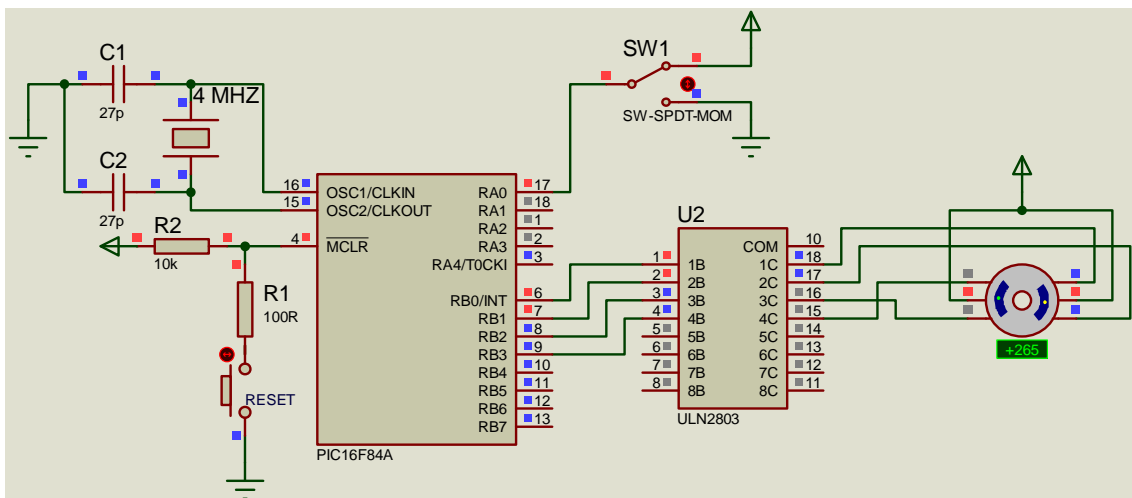
END                                ;Fin del programa
```

12. Proponer un circuito que tenga el puerto B como salida conectando un motor a pasos en sentido del reloj. Que inicie la rotación cuando se presione el bit 0 del puerto A y si se presiona cuando esta rotando este deberá parar, es decir el bit será de arranque y paro.

*a. Diagrama de flujo*



*b. Circuito*



*c. Código .asm*

```

;+-----+
;|          ENCABEZADO          |
;+-----+
LIST      p = 16F84              ;Usar el PIC16F84A-04/P
RADIX     HEX                    ;Todo en hexadecimal
;+-----+
;|          VARIABLES          |
;+-----+
w         equ      0              ;Cuando d=0 el destino es w
f         equ      1              ;Cuando d=1 el destino es f
PUERTOA   equ      0x05           ;Dirección del Puerto A
PUERTOB   equ      0x06           ;Dirección del Puerto B
ESTADO    equ      0x03           ;Dirección del Estado
PCL       equ      0x02           ;Dirección de PCL
Aux1      equ      0x0D
Aux2      equ      0x0E
npaso     equ      0x20           ;Dirección del reg que lleva el conteo de
los pasos
;+-----+
;|  CONFIGURACIÓN DEL PUERTO B COMO SALIDA  |
;|  Y EL PUERTO A COMO DE SALIDA          |
;+-----+
          ORG      0x00           ;Dirección de inicio
          BSF      ESTADO,5       ;Pasarse al Banco 1
          MOVLW   0x00
          MOVWF   PUERTOB         ;Establecer el Puerto B como de salida
          MOVLW   0x0F
          MOVWF   PUERTOA         ;Puerto A como de entrada RA0-RA3
          BCF     ESTADO,5       ;Volver al banco 0
;+-----+
;|          PROGRAMA PRINCIPAL          |
;+-----+
;inicio
Inicio    CLRf      npaso         ;Borrar contenido de npaso
Test      BTFSF    PUERTOA,0     ;El bit 0 del puerto A es 1?
          GOTO     Inicio         ;No, ir a inicio
          GOTO     Giro          ;Si, ir a Giro
;fin
;+-----+
;|          GIRA MOTOR HACIA LA DERECHA  |
;+-----+
;inicio
Giro      MOVF     npaso,w        ;Pasarse a W el contenido de contador
          CALL    TablaD         ;Llamar la tabla de pasos
          MOVWF   PUERTOB       ;Mandar al Puerto B el valor obtenido
          CALL    Retardo        ;Llamar subrutina de retardo
          INCF    npaso,f        ;Incrementar, npaso = npaso + 1
          MOVLW   0x04          ;Mover b'00000100' a W

```

```

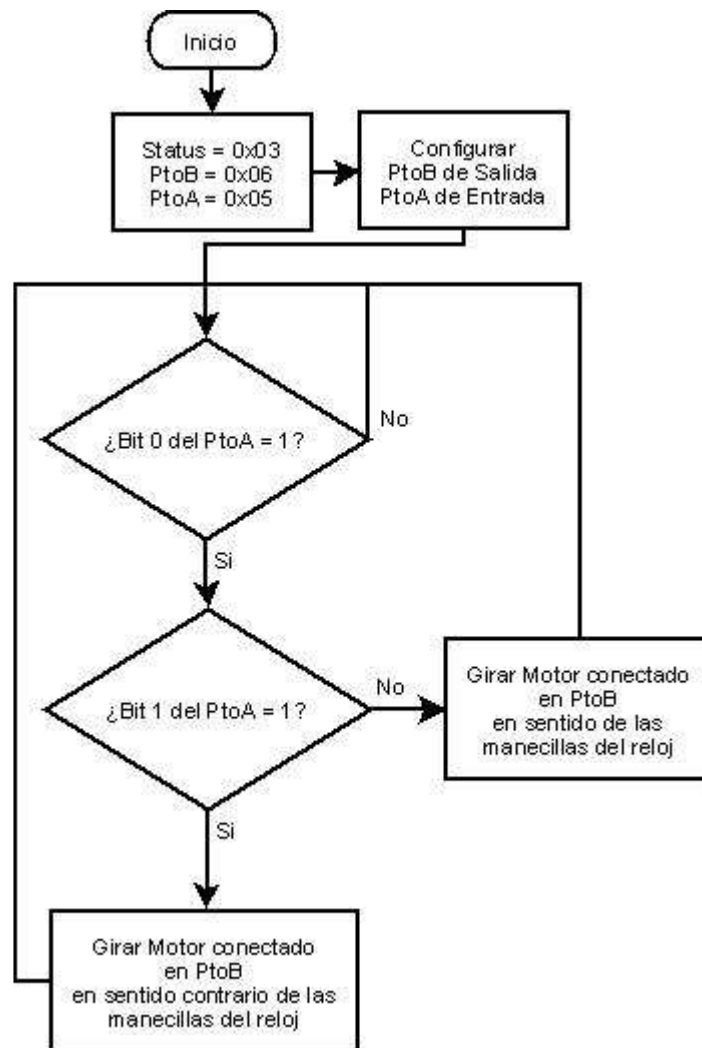
        XORWF    npaso,w           ;Hacer ope XOR cuenta con W
        BTFSS   ESTADO,2         ;¿El contador es igual a 4?
        GOTO    Test             ;No,va a ciclo
        GOTO    Inicio          ;Si, regresa al testeo
;fin
;+-----+
;|   RETARDO DE 60 mlSeg       |
;+-----+
Retardo   MOVLW   .33           ; 1 set number of repetitions (B)
          MOVWF   Aux1         ; 1 |
PLoop1   MOVLW   .60           ; 1 set number of repetitions (A)
          MOVWF   Aux2         ; 1 |
PLoop2   CLRWDT                ; 1 clear watchdog
          CLRWDT                ; 1 cycle delay
          DECFSZ   Aux2,1      ; 1 + (1) is the time over? (A)
          GOTO    PLoop2      ; 2 no, loop
          DECFSZ   Aux1,1      ; 1 + (1) is the time over? (B)
          GOTO    PLoop1      ; 2 no, loop
          CLRWDT                ; 1 cycle delay
          RETURN                ; 2+2 Done
;fin
;+-----+
;|   TABLA MOTOR GIRA A LA DERECHA |
;+-----+
TablaD   ADDWF   PCL,f
          RETLW   B'00001001'   ;09
          RETLW   B'00000011'   ;03
          RETLW   B'00000110'   ;06
          RETLW   B'00001100'   ;0C

          END                   ;Fin del programa

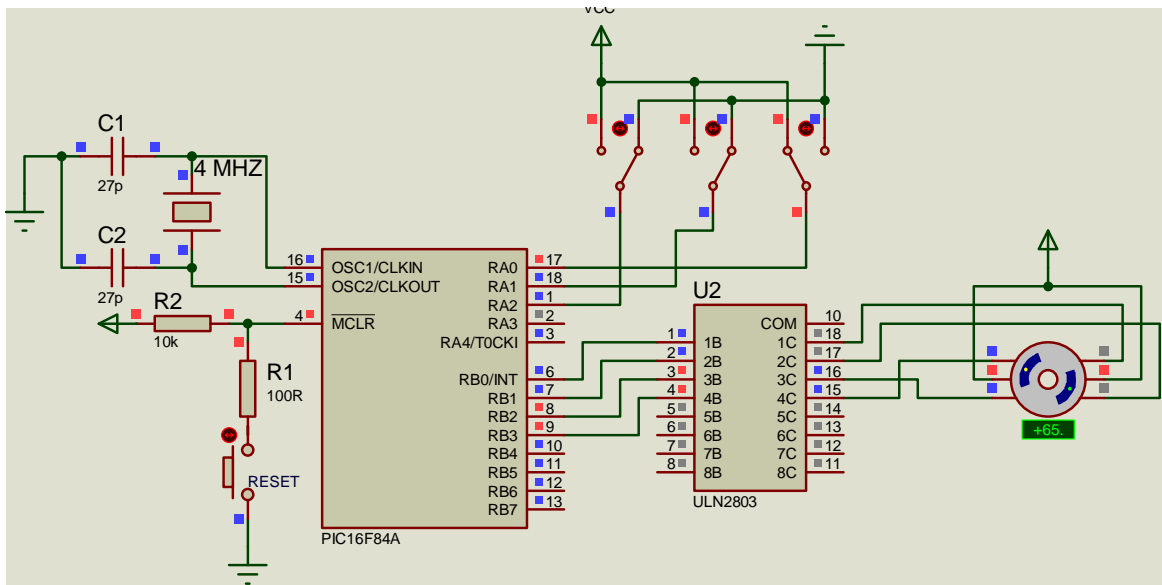
```

13. Proponer un circuito que tenga el puerto B como salida conectando un motor a pasos en sentido contrario del reloj. El bit 0 del puerto A será de arranque y paro, el bit 1 controlara el sentido de giro en dirección del reloj, el bit 1 controlara el sentido de giro en dirección contraria del reloj.

a. *Diagrama de flujo*



### b. Circuito



### c. Código .asm

```

;-----+
;|                                     |
;|                                     |
;|                                     |
;-----+
LIST          p = 16F84                ;Usar el PIC16F84A-04/P
RADIX         HEX                      ;Todo en hexadecimal
;-----+
;|                                     |
;|                                     |
;|                                     |
;-----+
w             equ      0                ;Cuando d=0 el destino es w
f             equ      1                ;Cuando d=1 el destino es f
PUERTOA      equ      0x05             ;Dirección del Puerto A
PUERTOB      equ      0x06             ;Dirección del Puerto B
ESTADO       equ      0x03             ;Dirección del Estado
PCL          equ      0x02             ;Dirección de PCL
Aux1         equ      0x0D
Aux2         equ      0x0E
n Paso       equ      0x20             ;Dirección del reg que lleva el conteo de
los pasos
;-----+
;|                                     |
;| CONFIGURACIÓN DEL PUERTO B COMO SALIDA |
;| Y EL PUERTO A COMO DE SALIDA          |
;|                                     |
;-----+
ORG          0x00                      ;Dirección de inicio
BSF         ESTADO,5                   ;Pasarse al Banco 1
MOVLW      0x00
MOVWF      PUERTOB                     ;Establecer el Puerto B como de salida
MOVLW      0x0F

```

```

MOVWF PUERTOA ;Puerto A como de entrada RA0-RA3
BCF ESTADO,5 ;Volver al banco 0
+-----+
;| PROGRAMA PRINCIPAL |
+-----+
;inicio
Inicio CLRf npaso ;Borrar contenido de npaso
Test BTFSS PUERTOA,0 ;El bit 0 del puerto A es 1?
GOTO Inicio ;No, ir a inicio
BTFSS PUERTOA,1 ;Si,El bit 1 del puerto A es 1?
GOTO GiroD ;No, ir a gira hacia la derecha
GOTO GiroI ;Si, ir a gira hacia la izquierda

;fin
+-----+
;| GIRA MOTOR HACIA LA DERECHA |
+-----+
;inicio
GiroD MOVF npaso,w ;Pasar a W el contenido de contador
CALL TablaD ;Llamar la tabla de pasos
MOVWF PUERTOB ;Mandar al Puerto B el valor obtenido
CALL Retardo ;Llamar retardo
INCF npaso,f ;Incrementar npaso = npaso + 1
MOVLW 0x04 ;Mover b'00000100' a W
XORWF npaso,w ;Hacer ope XOR cuenta con W
BTFSS ESTADO,2 ;¿El contador es igual a 4?
GOTO Test ;No,va a ciclo
GOTO Inicio ;Si, regresa al testeo

;fin
+-----+
;| GIRA MOTOR HACIA LA IZQUIERDA |
+-----+
;inicio
GiroI MOVF npaso,w ;Pasar a W el contenido de contador
CALL TablaI ;Llamar la tabla de pasos
MOVWF PUERTOB ;Mandar al Puerto B el valor obtenido
CALL Retardo ;Llamar retardo
INCF npaso,f ;Incrementar npaso = npaso + 1
MOVLW 0x04 ;Mover b'00000100' a W
XORWF npaso,w ;Hacer ope XOR cuenta con W
BTFSS ESTADO,2 ;¿El contador es igual a 4?
GOTO Test ;No,va a ciclo
GOTO Inicio ;Si, regresa al testeo

;fin
+-----+
;| RETARDO DE 60 mlSeg |
+-----+
Retardo MOVLW .33 ; 1 set number of repetitions (B)
MOVWF Aux1 ; 1 |
PLoop1 MOVLW .60 ; 1 set number of repetitions (A)

```

```

PLoop2    MOVWF    Aux2            ; 1 |
          CLRWDT                   ; 1 clear watchdog
          CLRWDT                   ; 1 cycle delay
          DECFSZ   Aux2,1          ; 1 + (1) is the time over? (A)
          GOTO     PLoop2         ; 2 no, loop
          DECFSZ   Aux1,1          ; 1 + (1) is the time over? (B)
          GOTO     PLoop1         ; 2 no, loop
          CLRWDT                   ; 1 cycle delay
          RETURN                   ; 2+2 Done

```

```

;fin

```

```

;+-----+
;|          TABLA MOTOR GIRA A LA DERECHA  |
;+-----+

```

```

TablaD    ADDWF    PCL,f
          RETLW   B'00001001'      ;09
          RETLW   B'00000011'      ;03
          RETLW   B'00000110'      ;06
          RETLW   B'00001100'      ;0C

```

```

;+-----+
;|          TABLA MOTOR GIRA A LA IZQUIERDA |
;+-----+

```

```

TablaI    ADDWF    PCL,f
          RETLW   B'00001100'      ;0C
          RETLW   B'00000110'      ;06
          RETLW   B'00000011'      ;03
          RETLW   B'00001001'      ;09

```

```

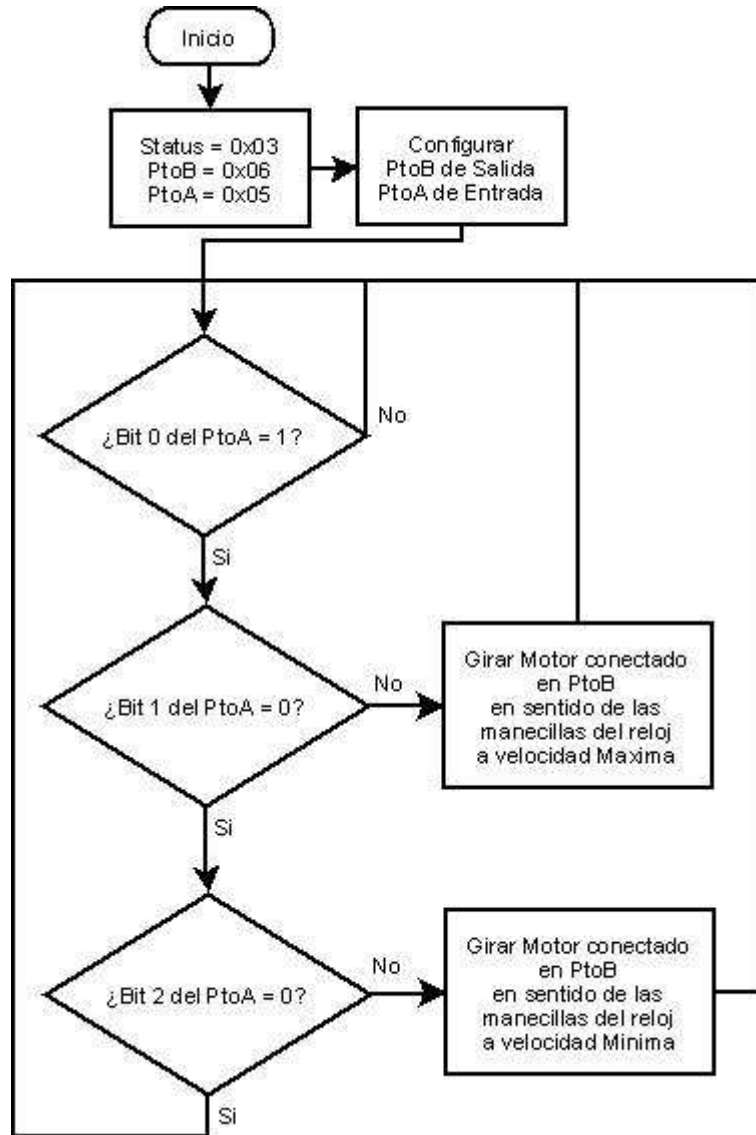
END                                ;Fin del programa

```



14. Proponer un circuito que tenga el puerto B como salida conectando un motor a pasos, conectado al puerto B, y en el puerto A como entrada: el bit 0 incrementa la velocidad, el bit 1 decrementa la velocidad.

a. Diagrama de flujo





```

                                BCF          ESTADO,5          ;Volver al banco 0
;+-----+
;|          PROGRAMA PRINCIPAL          |
;+-----+
;inicio
Inicio      CLRFB          npaso          ;Borrar contenido de npaso
Test        BTFSB          PUERTOA,0      ;El bit 0 del puerto A es 1?
           GOTO          Inicio          ;No, ir a inicio
           BTFSB          PUERTOA,1      ;Si,El bit 1 del puerto A es 0?
           GOTO          VelMax         ;No, gira a vel max
           BTFSB          PUERTOA,2      ;Si, el bit 2 del puerto A es 0?
           GOTO          VelMin         ;No, gira a vel min
           GOTO          Inicio         ;Si, vuelve al testeo

;fin
;+-----+
;|          GIRA MOTOR A VEL MAX          |
;+-----+
;inicio
VelMax      MOVFB          npaso,w        ;Pasar a W el contenido de contador
           CALL          TablaD          ;Llamar la tabla de pasos
           MOVWF          PUERTOB        ;Mandar al Puerto B el valor obtenido
           CALL          Retardo1        ;llamar la subrutina de retardo
           INCF          npaso,f         ;Incrementar npaso = npaso + 1
           MOVLW          0x04          ;Mover b'00000100' a W
           XORWF          npaso,w        ;Hacer ope XOR cuenta con W
           BTFSB          ESTADO,2      ;¿El contador es igual a 4?
           GOTO          Test           ;No,va a ciclo
           GOTO          Inicio         ;Si, regresa al testeo

;fin
;+-----+
;|          GIRA MOTOR A VEL MIN          |
;+-----+
;inicio
VelMin      MOVFB          npaso,w        ;Pasar a W el contenido de contador
           CALL          TablaD          ;Llamar la tabla de pasos
           MOVWF          PUERTOB        ;Mandar al Puerto B el valor obtenido
           CALL          Retardo         ;Llamar la subrutina de retardo1
           INCF          npaso,f         ;Incrementar npaso = npaso + 1
           MOVLW          0x04          ;Mover b'00000100' a W
           XORWF          npaso,w        ;Hacer ope XOR cuenta con W
           BTFSB          ESTADO,2      ;¿El contador es igual a 4?
           GOTO          Test           ;No,va a ciclo
           GOTO          Inicio         ;Si, regresa al testeo

;fin
;+-----+
;|          RETARDO DE 60 mlSeg          |
;+-----+
Retardo     MOVLW          .33           ; 1 set number of repetitions (B)
           MOVWF          Aux1          ; 1 |

```

```

PLoop1    MOVLW    .60                ; 1 set number of repetitions (A)
          MOVWF    Aux2                ; 1 |
PLoop2    CLRWDT   ; 1 clear watchdog
          CLRWDT   ; 1 cycle delay
          DECFSZ   Aux2,1              ; 1 + (1) is the time over? (A)
          GOTO     PLoop2              ; 2 no, loop
          DECFSZ   Aux1,1              ; 1 + (1) is the time over? (B)
          GOTO     PLoop1              ; 2 no, loop
          CLRWDT   ; 1 cycle delay
          RETURN   ; 2+2 Done

```

```

;fin

```

```

;+-----+

```

```

;|   RETARDO DE 30 mlSeg   |

```

```

;+-----+

```

```

Retardo1  MOVLW    .15                ; 1 set number of repetitions (B)
          MOVWF    Aux1                ; 1 |
PLoop1    MOVLW    .30                ; 1 set number of repetitions (A)
          MOVWF    Aux2                ; 1 |
PLoop2    CLRWDT   ; 1 clear watchdog
          CLRWDT   ; 1 cycle delay
          DECFSZ   Aux2,1              ; 1 + (1) is the time over? (A)
          GOTO     PLoop2              ; 2 no, loop
          DECFSZ   Aux1,1              ; 1 + (1) is the time over? (B)
          GOTO     PLoop1              ; 2 no, loop
          CLRWDT   ; 1 cycle delay
          RETURN   ; 2+2 Done

```

```

;fin

```

```

;+-----+

```

```

;|   TABLA MOTOR GIRA A LA DERECHA   |

```

```

;+-----+

```

```

TablaD    ADDWF    PCL,f
          RETLW    B'00001001'        ;09
          RETLW    B'00000011'        ;03
          RETLW    B'00000110'        ;06
          RETLW    B'00001100'        ;0C

```

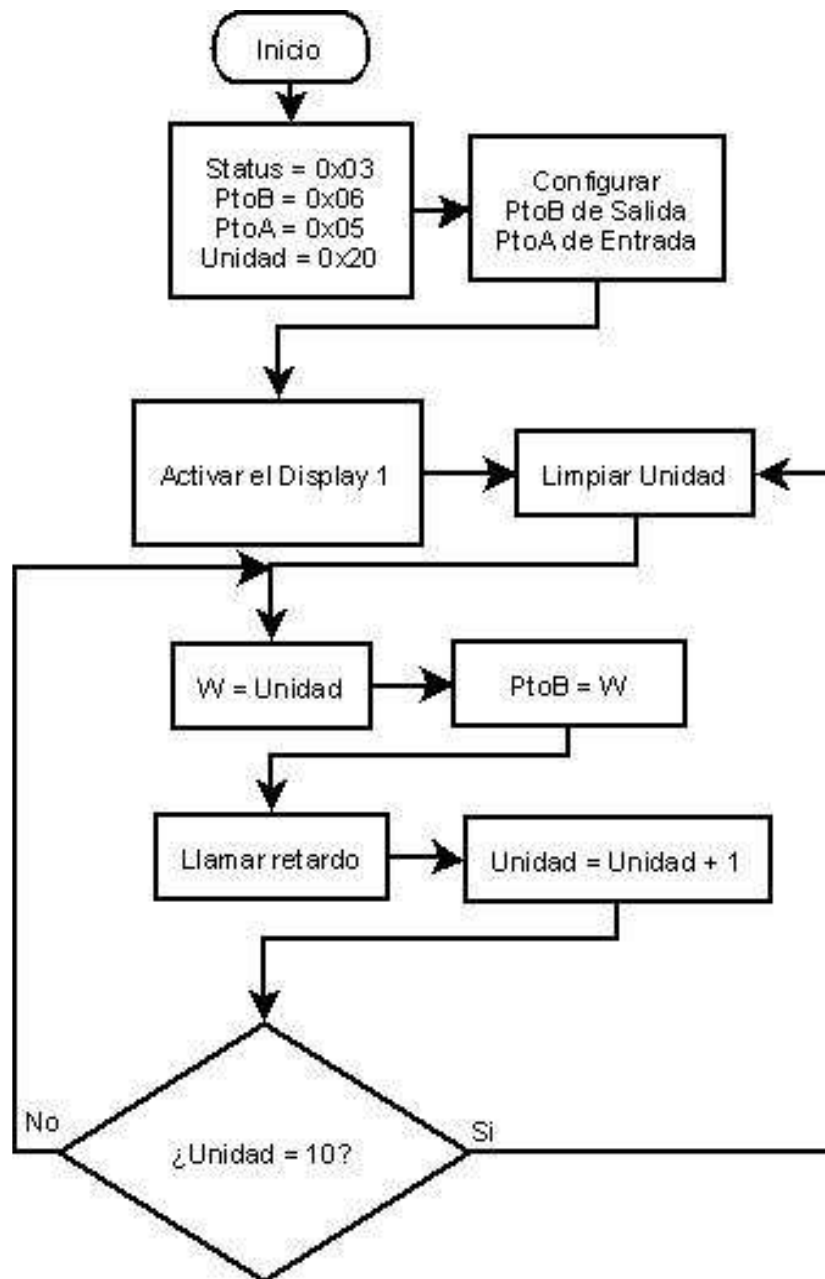
```

          END                          ;Fin del programa

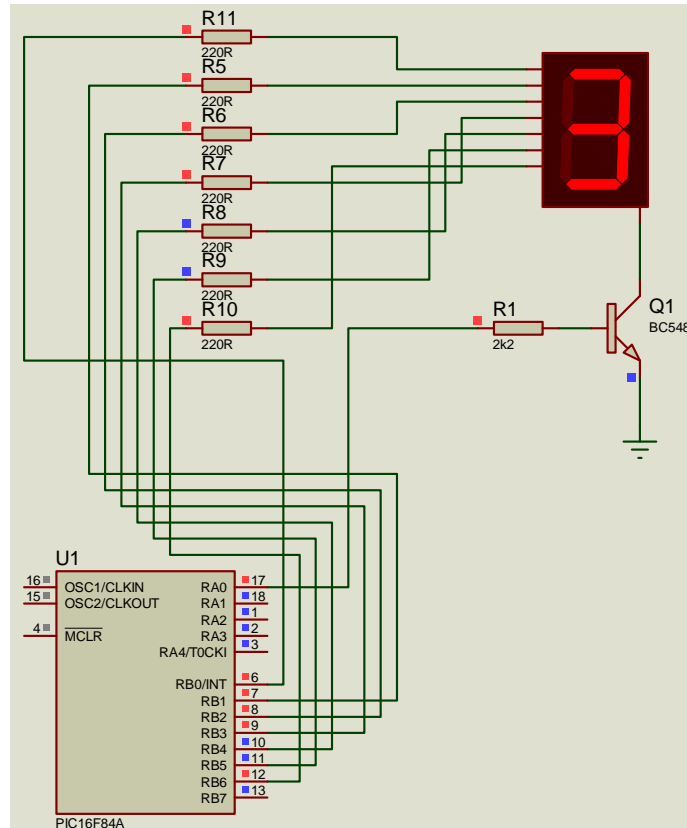
```

15. Proponer un circuito que tenga el puerto B como salida conectando un display de 7 segmentos conectando al puerto B de tal forma que podamos visualizar en el display un contador de 0 al 9.

*a. Diagrama de flujo*



### b. Circuito



### c. Código .asm

```

;-----+
;|                                     |
;|                                     |
;-----+
LIST      p = 16F84                    ;Usar el PIC16F84A-04/P
RADIX    HEX                          ;Todo en hexadecimal
;-----+
;|                                     |
;|                                     |
;-----+
w        equ      0                    ;Cuando d=0 el destino es w
f        equ      1                    ;Cuando d=1 el destino es f
PUERTOB  equ      0x06                 ;Dirección del Puerto B
PUERTOA  equ      0x05
ESTADO   equ      0x03                 ;Dirección del Estado
PCL      equ      0x02                 ;Dirección de PCL
Aux1     equ      0x0C                 ;Direcciones de ocupados para
Aux2     equ      0x0D                 ;la subrutina de retardo
Aux3     equ      0x0E
cuenta   equ      0x20                 ;Dirección del registro que lleva el conteo
;-----+
;| CONFIGURACIÓN DEL PUERTO B COMO SALIDA |
;-----+
ORG      0x00                          ;Dirección de inicio

```

```

BSF      ESTADO,5 ;Pasarse al Banco 1
CLRF     PUERTOB  ;Establecer el Puerto B como de salida
CLRF     PUERTOA
BCF      ESTADO,5 ;Volver al banco 0
;+-----+
;|          PROGRAMA PRINCIPAL          |
;+-----+
BSF      PUERTOA,0 ;Activar el display 1

Ciclo1   CLRF      cuenta      ;Inicializar el contador
Ciclo2   MOVF     cuenta,w     ;Pasar a W el contenido de contador
CALL     Tabla      ;Llamar a la tabla dependiendo de W
MOVWF   PUERTOB    ;Mandar al Puerto B el valor obtenido
CALL     Retardo    ;Llamar la subrutina de retardo
INCF    cuenta,f   ;Incrementar al contador
MOVLW   0x0A       ;Mover b'1010' a W
XORWF   cuenta,w   ;Hacer ope XOR cuenta con W
BTFS    ESTADO,2  ;¿El contador es igual a 10?
GOTO    Ciclo2    ;No, seguir con el conteo
GOTO    Ciclo1    ;Si, ir a inicializar el contador
;+-----+
;|          RUTINA DE RETARDO          |
;+-----+
Retardo  MOVLW    0x0D       ;14
        MOVWF    Aux1
Uno      MOVLW    0x48       ;72
        MOVWF    Aux2
Dos      MOVLW    0x7A       ;0xF7      ;247
        MOVWF    Aux3
Tres     CLRWDT                    ;Limpiar el reloj del Perro guardian
        DECFSZ   Aux3,f         ;Decrementar Aux3 -> Aux3-1
        GOTO     Tres
        DECFSZ   Aux2,f         ;Decrementar Aux2 -> Aux2-1
        GOTO     Dos
        DECFSZ   Aux1,f         ;Decrementar Aux1 -> Aux1-1
        GOTO     Uno
        GOTO     Sig
Sig      CLRWDT                    ;Limpiar el reloj del Perro guardian
        RETURN
;+-----+
;|          TABLA DE LOS DÍGITOS (0-9)          |
;+-----+
Tabla    ;B'gfedcba'
        ADDWF   PCL,f
        RETLW   B'00111111' ; 0
        RETLW   B'00000110' ; 1
        RETLW   B'01011011' ; 2
        RETLW   B'01001111' ; 3

```

```
RETLW    B'01100110' ; 4
RETLW    B'01101101' ; 5
RETLW    B'01111100' ; 6
RETLW    B'00000111' ; 7
RETLW    B'01111111' ; 8
RETLW    B'01100111' ; 9

END                                ;Fin del programa
```