

Microcontroladores Laboratorio

Semana 3

Semestre 2021-1

Profesor: Kalun José Lau Gan

Empezamos en breve

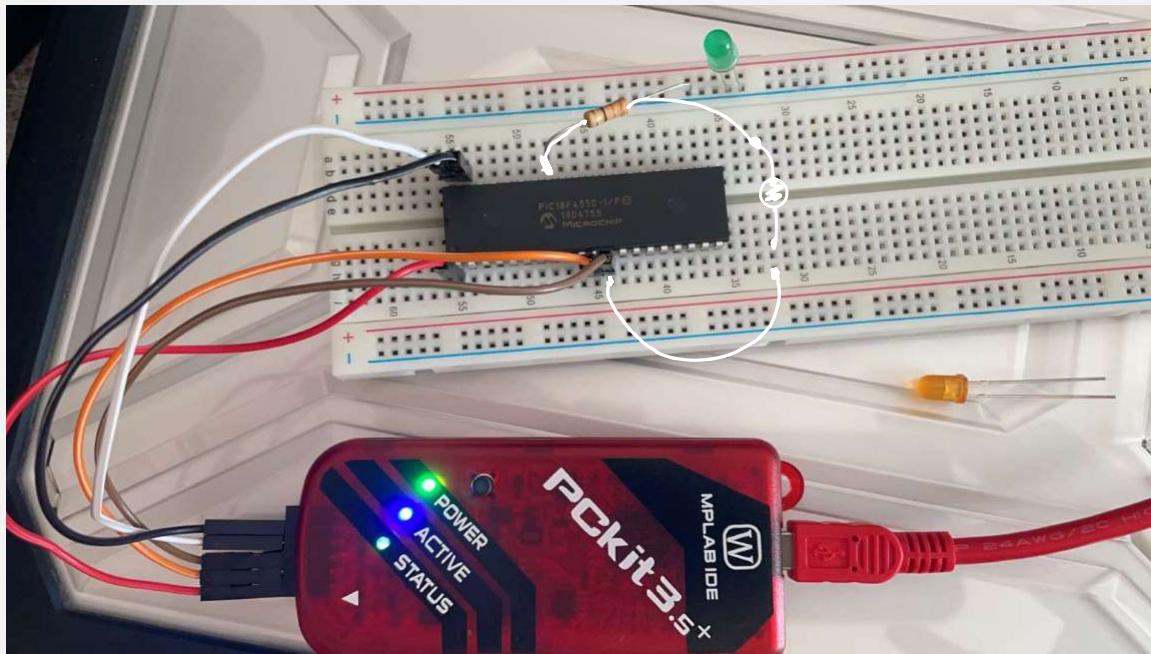
1

Preguntas previas:

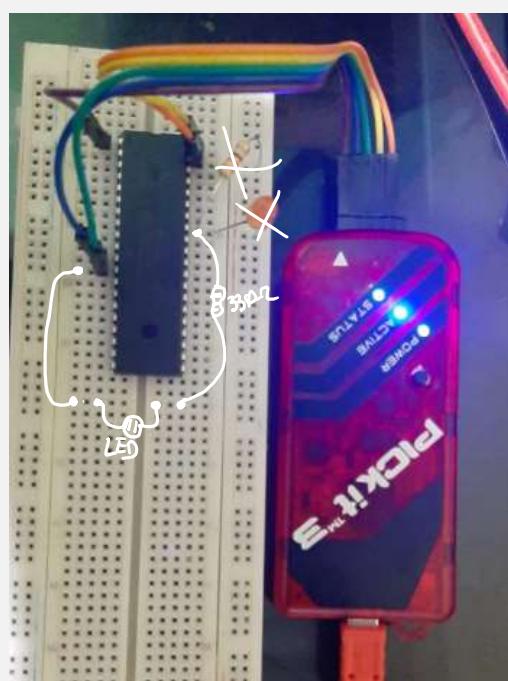
- ¿Cuáles eran los punteros?
 - Para la memoria de datos: FSR0, FSR1, FSR2
 - Para la memoria de programa: TBLPTR
- El ejemplo de ayer se queda mostrando 0. ¿Qué mas debe de hacer?
 - Se tiene que interactuar las cuatro entradas (BCD) del RB0 al RB3 para que funcione el decodificador.
- Ayer se mencionó que faltaba una configuración para que funcionaran las salidas en RE. ¿Qué era lo que faltaba?
 - El puerto E (RE2:RE0) por defecto son entradas analógicas por lo que hay que configurarlas como digital haciendo lo siguiente:

movlw 0x0F
movwf ADCON1

2



3



4

Agenda:

- Instrucciones CPFSEQ, CPFSLT, CPFSGT en MPASM
- Interface a display de siete segmentos
- Multiplexación en displays de siete segmentos

5

Registro STATUS

- Ref. página 73 de la hoja técnica del microcontrolador PIC18F4550
- Contiene las banderas del CPU, esas banderas serán actualizadas luego de operar

REGISTER 5-2: STATUS REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC ⁽¹⁾	C ⁽²⁾
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

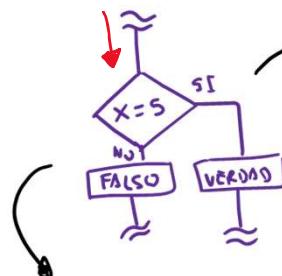
bit 7-5	Unimplemented: Read as '0'
bit 4	N: Negative bit This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1). 1 = Result was negative 0 = Result was positive
bit 3	OV: Overflow bit This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7 of the result) to change state. 1 = Overflow occurred for signed arithmetic (in this arithmetic operation) 0 = No overflow occurred
bit 2	Z: Zero bit 1 = The result of an arithmetic or logic operation is zero 0 = The result of an arithmetic or logic operation is not zero
bit 1	DC: Digit Carry/Borrow bit ⁽¹⁾ For ADDWF, ADDLW, SUBLW and SUBWF instructions: 1 = A carry-out from the 4th low-order bit of the result occurred 0 = No carry-out from the 4th low-order bit of the result
bit 0	C: Carry/Borrow bit ⁽²⁾ For ADDWF, ADDLW, SUBLW and SUBWF instructions: 1 = A carry-out from the Most Significant bit of the result occurred 0 = No carry-out from the Most Significant bit of the result occurred

- Note 1:** For Borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.
- 2:** For Borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low-order bit of the source register.

6

Instrucciones de comparación numérica (CPFSEQ, CPFSLT, CPFSGT)

$\text{CPFSEQ} \rightarrow f = W_{\text{reg}}$
 $\text{CPFS LT} \rightarrow f < W_{\text{reg}}$
 $\text{CPFS GT} \rightarrow f > W_{\text{reg}}$



En MPASM:

Usaremos CPFSEQ:

Pregunta
si reg = W

CPFSEQ [reg]

En lenguaje de alto nivel:

```
if (x = 5) {
    [VERDAD]
} else {
    [Falso]
}
```

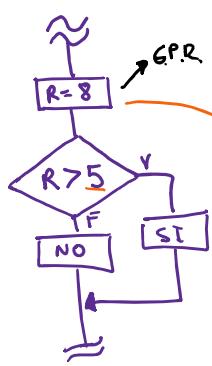
$(f) = W$

```
movlw .5
cpfseq x-reg
    goto Falso
    goto Verdad
```

$x = 5$

7

Ejemplo: Pasar a MPASM el siguiente diagrama de flujo



R-reg EQU 0X0000

:
 movlw d'8'
 movwf R-reg
 movlw d'5'
 cpfsat R-reg
 Comparando si:
 $R-reg > 5$

Nota: En las instrucciones cpfs-- el segundo parámetro de la comparación debe de estar en Wreg

8

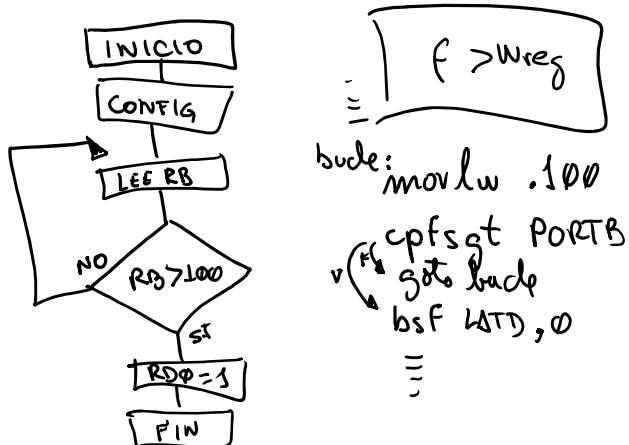
Instrucciones con toma de decisión:



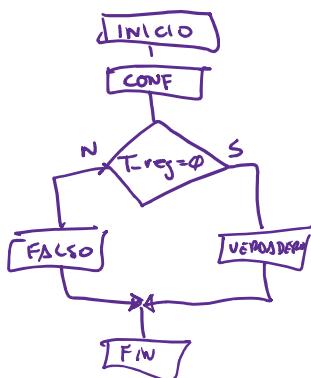
- btfss, btfsc
 - Prueba el #bit de [registro] si es 0 ó 1
 - decfsz, incfsz
 - Decrementa o incrementa [registro] y pregunta si es cero.
 - dcfnsz, icfnsz
 - Decrementa o incrementa [registro] y pregunta si no es cero
 - cpfgsgt, cpfseq, cpfslt
 - Comparaciones numéricas entre [registro] y W
 - tstfsz
 - Prueba [registro] y salta si es cero

9

Ejemplo: Leer el valor de RB y colocar a uno RDO únicamente cuando el valor sea mayor de 100.



Ejemplo: Pregunta si T_reg es igual a cero, si es cierto va a etiqueta verdadero, si no es cierto va a etiqueta falso



Opción 1:

inicio: movlw .0
cpfseq T-reg
vF goto FALSO
goto VERDADERO

Opción 2:

inicio: tstfsz T-reg
vF goto FALSO
goto VERDADERO

Opción 3:

inicio: movf T-reg, W
sublw .0
btfsz STATUS, Z
vF goto FALSO
goto VERDADERO

Opción 4:

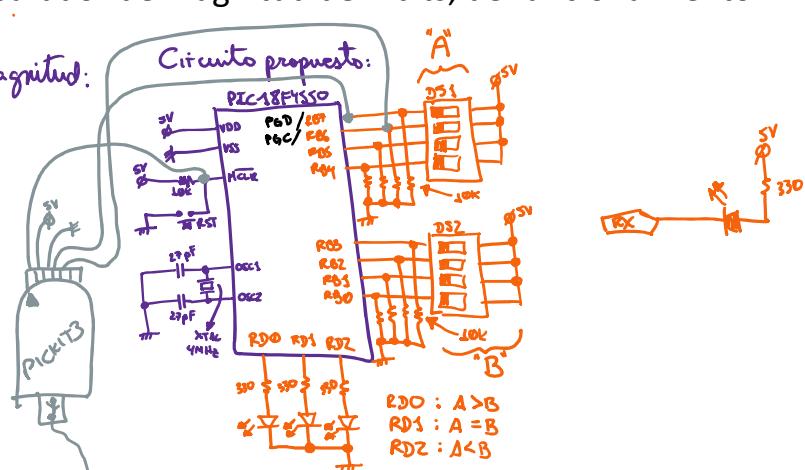
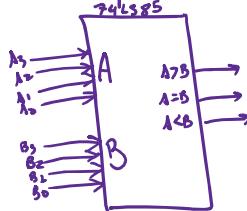
inicio: movf T-reg, W
sublw .0
b3 VERDADERO
FALSO: =
VERDADERO: =

11

Ejemplo

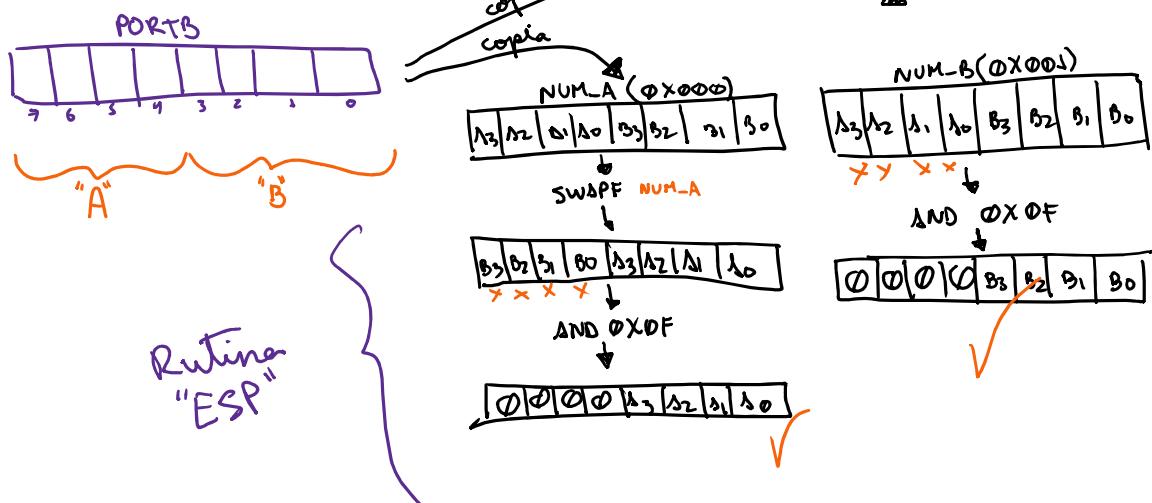
- Desarrollar un comparador de magnitud de 4 bits, de funcionamiento similar al 74LS85

Un comparador de magnitud:



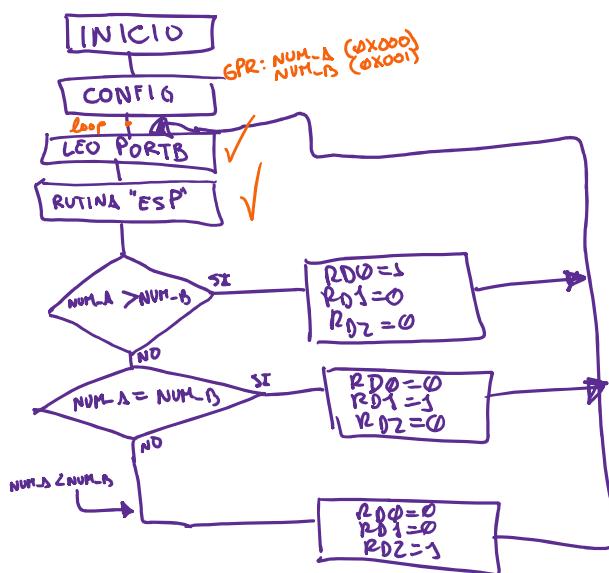
12

Algoritmo:



13

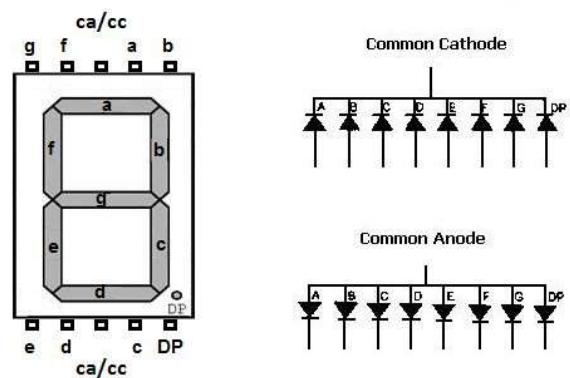
Algoritmo (cont.)



14

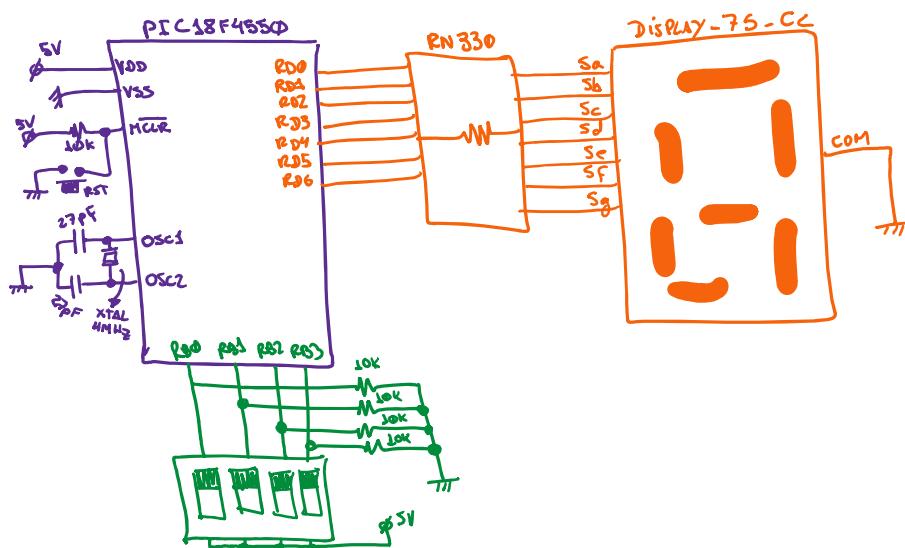
El display de siete segmentos

- Dos tipos: ánodo común y cátodo común
- Tablas de decodificación diferentes entre ellos.



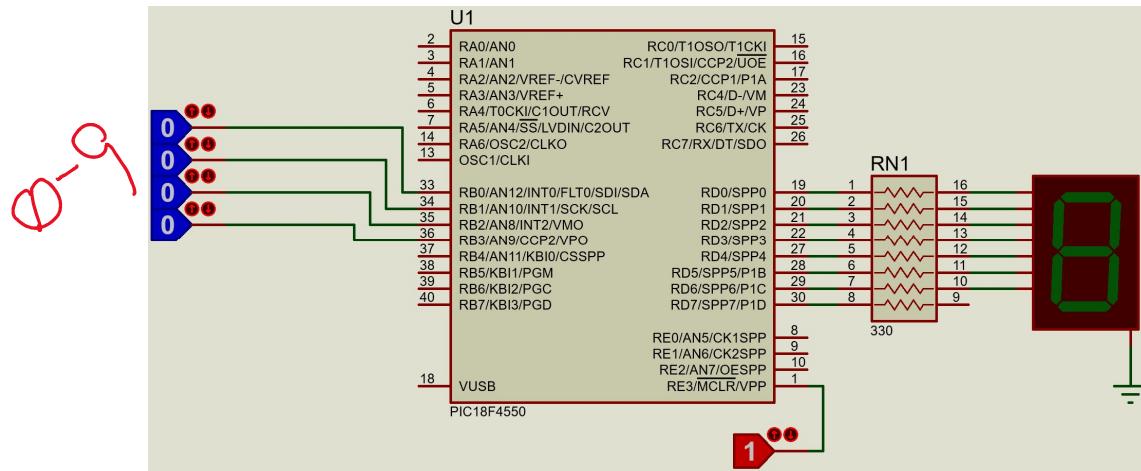
15

Ejercicio: Decodificador BCD a siete segmentos cátodo común



16

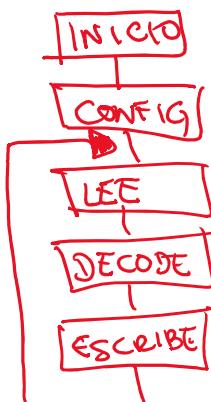
Ejemplo: Decodificador BCD-7Segmentos empleando tablas con PC



17

Código en MPASM

Diagrama de flujo:



1	;Este es un comentario, se le ant	25	loop:
2	list p=18f4550 ;Modelo d	26	movf PORTB, W
3	#include <p18f4550.inc> ;	27	andlw 0x0F ;En
4		28	mullw .2 ;Mu
5	;Directivas de preprocessador	29	movf PRODL, W
6	CONFIG PLLDIV = 1	30	call tabla_PC
7	CONFIG CPUDIV = OSC1_PLL2	31	movwf LATD
8	CONFIG FOSC = XT_XT	32	goto loop
9	CONFIG PWRT = ON	33	
10	CONFIG BOR = OFF	34	tabla_PC:
11	CONFIG WDT = OFF	35	addwf PCL, f
12	CONFIG CCP2MX = ON	36	retlw 0xBF ;0
13	CONFIG PBADEN = OFF	37	retlw 0x86 ;1
14	CONFIG MCLRE = ON	38	retlw 0xDB ;2
15	CONFIG LVP = OFF	39	retlw 0xCF ;3
16		40	retlw 0xE6 ;4
17	temp EQU 0x000 ;posición	41	retlw 0xED ;5
18	org 0x0000	42	retlw 0xFD ;6
19	goto init_conf	43	retlw 0x87 ;7
20		44	retlw 0xFF ;8
21	org 0x0020	45	retlw 0xE7 ;9
22	init_conf:	46	
23	clrf TRISD	47	end
24			

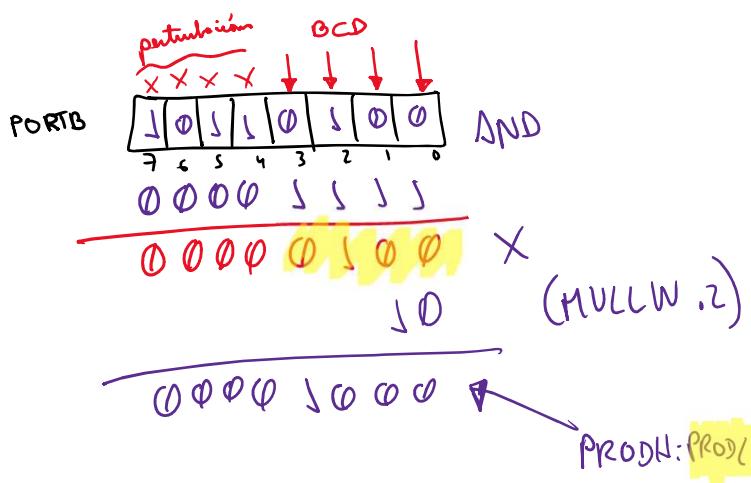
18

Cuestionario referente al ejemplo anterior

- ¿Cuál es la función de PC en este ejemplo?
- ¿Cuál es la diferencia entre RETURN y RETLW?
- ¿Por qué se tuvo que multiplicar x2 el valor recibido en el PORTB para que funcione correctamente la función del decodificador?

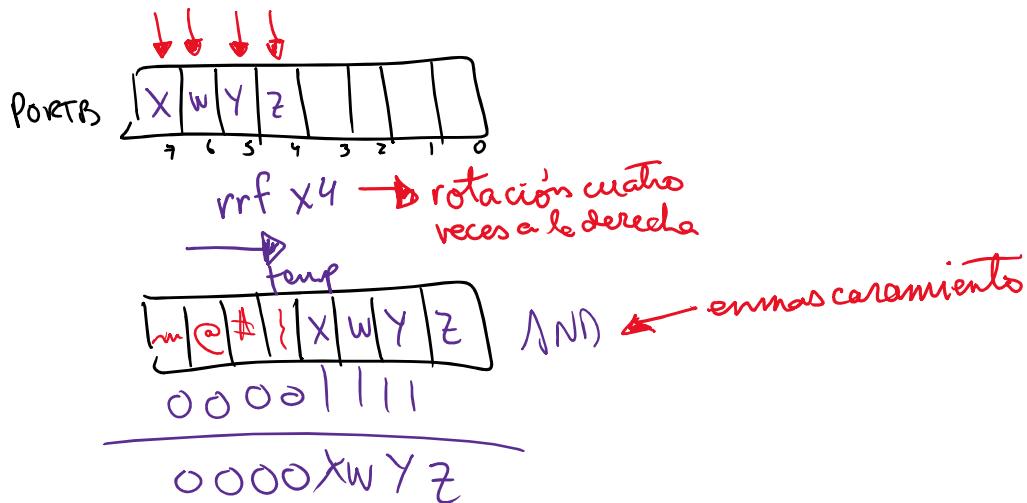
19

Operación de enmascaramiento de bits?



20

¿Qué pasa si uso RB7:RB4 como entradas al decodificador BCD-7S?



21

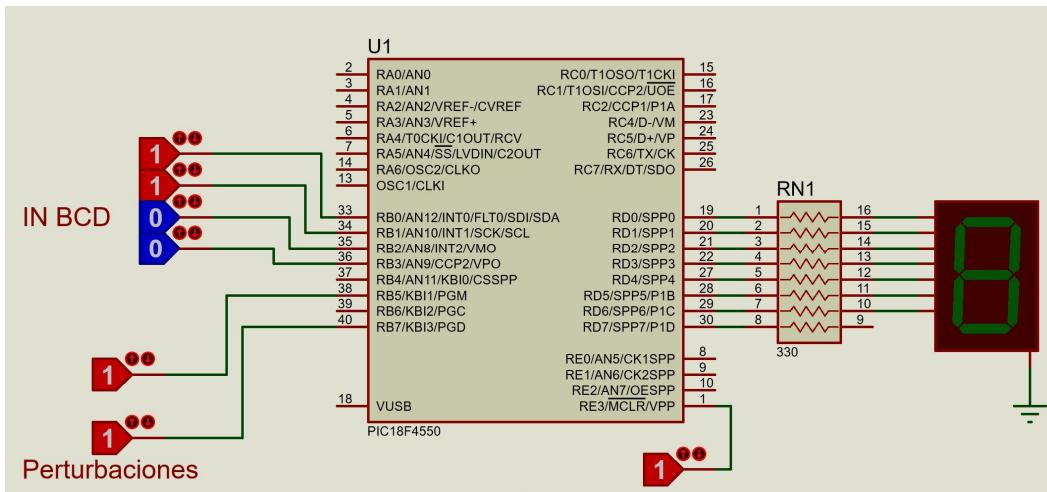
Cómo se están grabando las instrucciones en la memoria de programa:

Memoria de Programa		
35	addwf PCL, f	0XF0
36	retlw 0xBF ;0	0XF2
37	retlw 0x86 ;1	0XF4
38	retlw 0xDB ;2	0XF6
39	retlw 0xCF ;3	0XF8
40	retlw 0xE6 ;4	0XFA
41	retlw 0xED ;5	
42	retlw 0xFD ;6	
43	retlw 0x87 ;7	
44	retlw 0xFF ;8	
45	retlw 0xE7 ;9	

Nota: Hay que recordar que las instrucciones ocupan dos bytes en la memoria de programa

22

Ejemplo: Decodificador BCD-7Segmentos empleando tablas con TBLPTR



23

Código en MPASM

```

1 ;Este es un comentario, se le antecede un punto y
2     list p=18f4550      ;Modelo del microcontrolador
3     #include <p18f4550.inc>    ;Llamada a la librería
4
5 ;Directivas de preprocesador o bits de configuración
6 CONFIG  PLLDIV = 1           ; PLL Prescaler
7 CONFIG  CPUDIV = OSC1_PLL2   ; System Clock Prescaler
8 CONFIG  FOSC = XT_XT        ; Oscillator Selection
9 CONFIG  PWRT = ON           ; Power-up Timer
10 CONFIG  BOR = OFF           ; Brown-out Reset
11 CONFIG  WDT = OFF           ; Watchdog Timer
12 CONFIG  CCP2MX = ON          ; CCP2 MUX bit (On)
13 CONFIG  PBADEN = OFF         ; PORTB A/D Enable
14 CONFIG  MCILRE = ON          ; MCLR Pin Enable
15 CONFIG  LVP = OFF            ; Single-Supply ICSP Enable bit (Single-supply)
16
17 org 0x0000
18 goto init_conf
19
20 org 0x0300
21 tabla_7s db 0xBF, 0x86, 0xDB, 0xCF, 0xE6, 0xED, 0xFD, 0x87, 0xFF, 0xE7
22
23     org 0x0020
24     init_conf:
25         clrf TRISD
26         movlw HIGH tabla_7s
27         movwf TBLPTRH
28         movlw LOW tabla_7s
29         movwf TBLPTRL
30
31     loop:
32         movf PORTB, W
33         andlw 0x0F           ;Enmascaras los cuatro primeros bits
34         movwf TBLPTRL
35         TBLRD*
36         movff TABLAT, LATD
37         goto loop
38
39     end

```

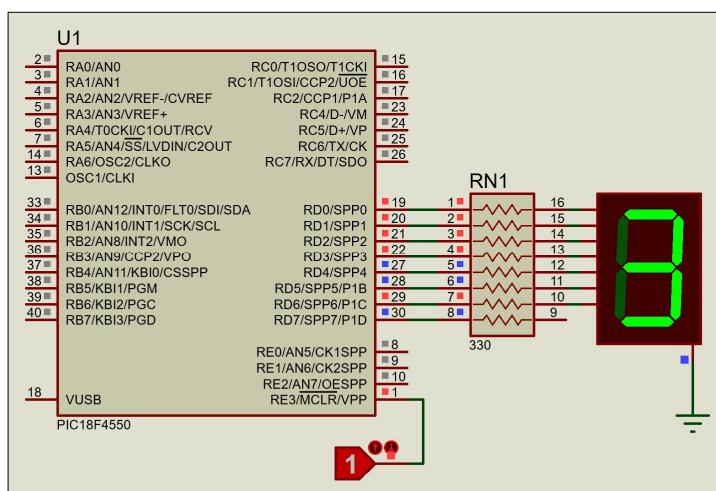
24

Cuestionario referente al ejemplo anterior:

- ¿Cuál es la función de la instrucción andlw 0x0F?
- ¿Qué es lo que hace TBLPTR en este ejemplo?
- ¿De qué trata la instrucción movff y cuántos ciclos demora?
- ¿Qué pasaría si se coloca un número superior al formato BCD?
- Modificar el decodificador para que de entrada sea binario en lugar de BCD y se muestre en el display de siete segmentos hasta el valor F
- Si en lugar de usar display de cátodo común se usa del tipo ánodo común. ¿Qué es lo que se modificaría en el código y hardware?

25

Ejemplo: Contador autoincremental 0-9



26

Ejemplo: Contador autoincremental 0-9

```

1 ;Este es un comentario, se le antecede un punto y coma
2 list p=18f4550 ;Modelo del microcontrolador
3 #include <p18f4550.inc> ;Llamada a la librería de nombre de los registros
4
5 ;Directivas de preprocesador o bits de configuración
6 CONFIG_PLLDIV = 1 ; PLL Prescaler Selection bits (No prescale (4 MHz oscillator input divide)
7 CONFIG_CFDUDIV = OSC1_PLL2 ; System Clock Postsclaler Selection bits ([Primary Oscillator input divide])
8 CONFIG_FOSC = XT_XT ; Oscillator Selection bits (XT oscillator (XT))
9 CONFIG_FWRT = ON ; Power-up Timer Enable bit (FWRT enabled)
10 CONFIG_BOR = OFF ; Brown-out Reset Enable bit (BOR disabled)
11 CONFIG_WDT = OFF ; Watchdog Timer Enable bit (WDT disabled (control is placed on the user software))
12 CONFIG_CCP2MX = ON ; CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
13 CONFIG_PBALEN = OFF ; PORIB A/D Enable bit (PORIB<4:0> pins are configured as analog inputs)
14 CONFIG_MCLRE = ON ; MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)
15 CONFIG_LVP = OFF ; Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
16
17 cblock 0x000
18 var_i
19 var_j
20 var_k
21 endc
22
23 org 0x0000
24 goto init_conf
25
26 ;Lo de líneas se almacenara de manera secuencial desde la 0x0300 en la mem prog
27 org 0x0500
28 tabla_7s db 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67, 0x79, 0x79, 0x79, 0x79, 0x79
29
30 org 0x0020
31 init_conf:
32     movlw 0x80
33     movwf TRISD ;RD6-RD0 como salidas
34     movlw HIGH tabla_7s
35     movwf TBLPTRH
36     movlw LOW tabla_7s
37     movwf TBLPTRL ;TBLPTR apunta a 0x0300 de la mem de prog.

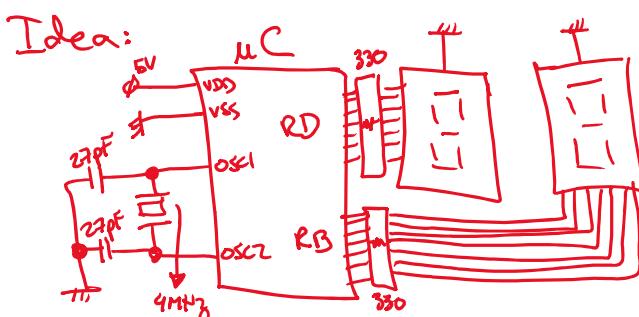
```

<pre> 39 loop: 40 TBLRD+ 41 movff TABLAT, LATD 42 movlw .9 43 cpfseq TBLPTRL 44 goto falso 45 clrf TBLPTRL 46 goto otro 47 falso: 48 incf TBLPTRL, f 49 otro: 50 call delay_long 51 goto loop </pre>	<pre> 53 ;subrutina de retardo 54 delay_long: 55 movlw .50 56 movwf var_i 57 otrol: 58 call bucle1 59 decfsz var_i,f 60 goto otrol 61 return 62 bucle1: 63 movlw .55 64 movwf var_j 65 otro2: 66 nop 67 nop 68 call bucle2 69 decfsz var_j,f 70 goto otro2 71 return 72 bucle2: 73 movlw .20 74 movwf var_k 75 otro3: 76 nop 77 decfsz var_k,f 78 goto otro3 79 return 80 end </pre>
---	--

27

Ejercicio:

- Realizar un contador 00-99 autoincremental con periodo de incremento de 500ms aproximadamente, el dígito unidad será emitido por el puerto RB y el dígito decena será emitido por el puerto RD



28

Ejercicios: