

Sistemas Digitales:

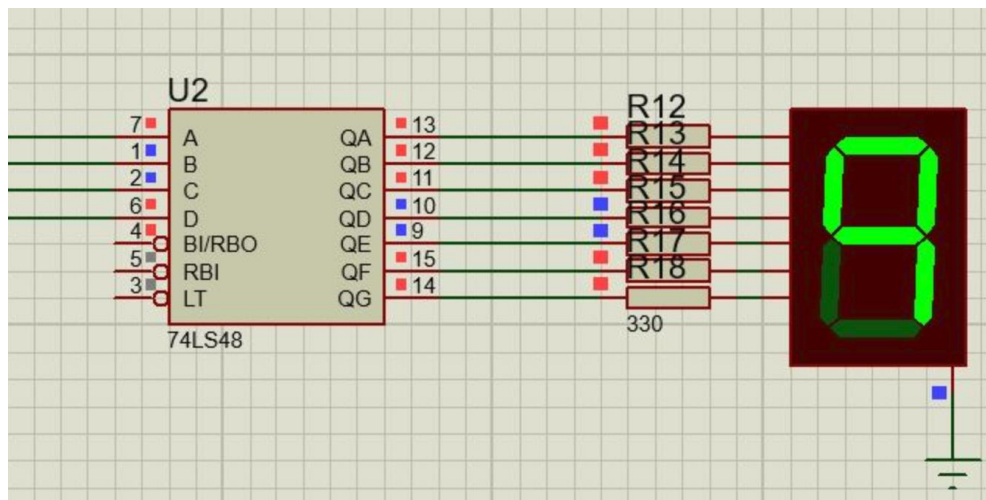
Introducción a los microcontroladores

Parte 4

Ingeniería Electrónica
UPC 2018

Por Kalun Lau

Ejercicio: Modelo de decodificador BCD-7Segmentos



Ejercicio: Modelo de decodificador BCD-7Segmentos

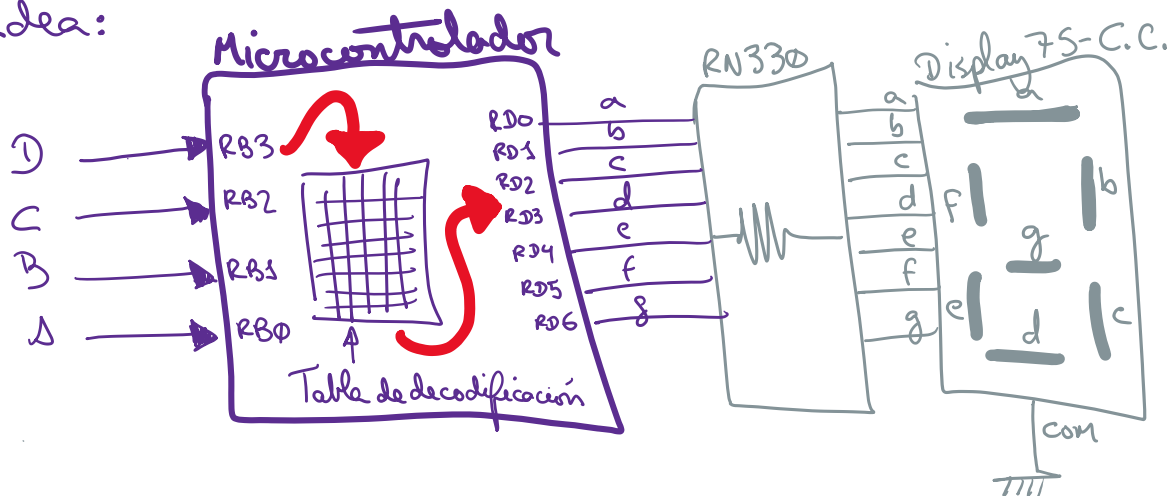
Tabla de decodificación (cátodo común)

s	BCD	G	F	E	D	C	B	A
0	0000	0	1	1	1	1	1	1
1	0001	0	0	0	0	1	1	0
2	0010	1	0	1	1	0	1	1
3	0011	1	0	0	1	1	1	1
4	0100	1	1	0	0	1	1	0
5	0101	1	1	0	1	1	0	1
6	0110	1	1	1	1	1	0	1
7	0111	0	0	0	0	1	1	1
8	1000	1	1	1	1	1	1	1
9	1001	1	1	0	1	1	1	1

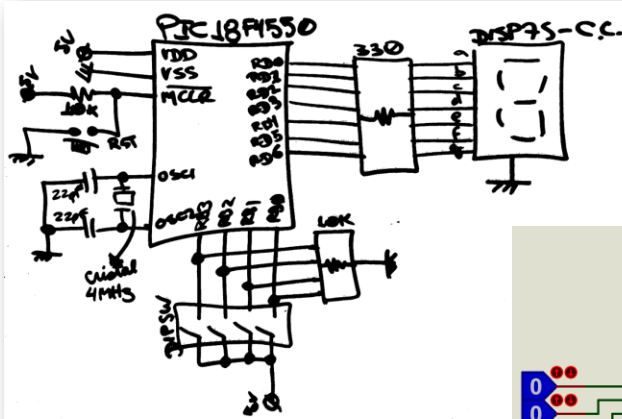
Ejercicio: Modelo de decodificador BCD-7Segmentos

Manejo de tablas en MPASM para PIC18F4550

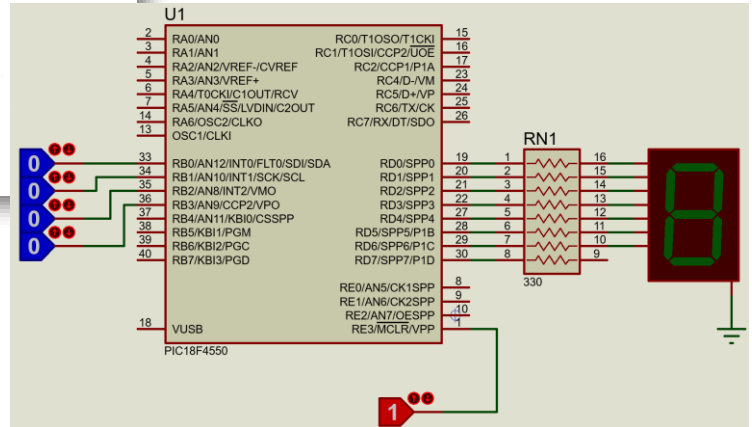
Idea:



Ejercicio: Modelo de decodificador BCD-7Segmentos



Simulación en Proteus

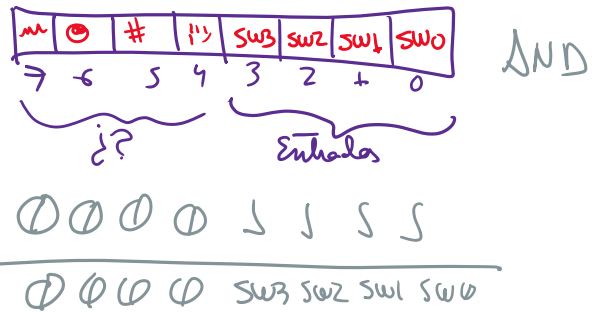
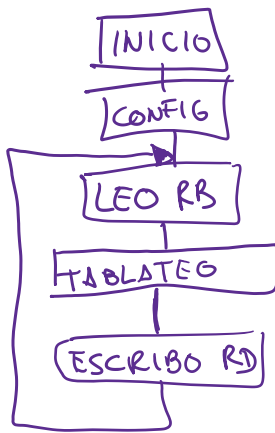


Implementación

Ejercicio: Modelo de decodificador BCD-7Segmentos

Opción 1: Empleando el PC

Diagrama de Flujo: El puerto B:

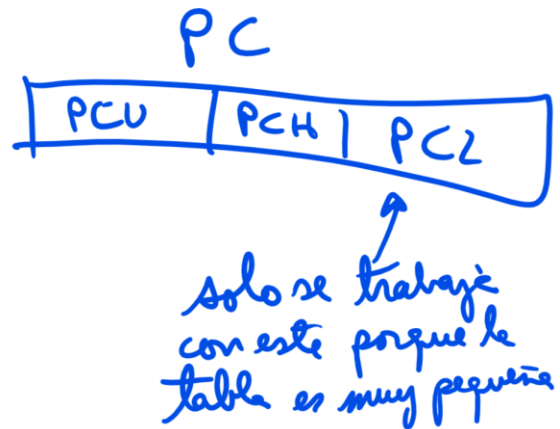


Ejercicio: Modelo de decodificador BCD-7Segmentos

```

28  inicio:
29      movf PORTB, W           ;Leemos el RB y lo mandamos a W
30      andlw 0x0F              ;Enmascaramiento
31      call tablaton
32
33  tablaton:
34      addwf PCL, f
35      retlw 0x3F ;0
36      retlw 0x06 ;1
37      retlw 0x5B ;2
38      retlw 0x4F ;3
39      retlw 0x66 ;4
40      retlw 0x6D ;5
41      retlw 0x7D ;6
42      retlw 0x07 ;7
43      retlw 0x7F ;8
44      retlw 0x67 ;9
45      retlw 0x79 ;E

```



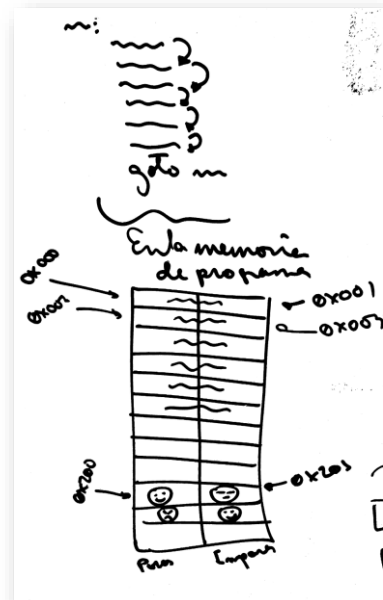
Ejercicio: Modelo de decodificador BCD-7Segmentos

Diagram illustrating the jump mechanism for the `tablaton` table:

The table contains 16 entries, each with a 7-segment display pattern (hex value) and a digit (decimal value). The entries are:

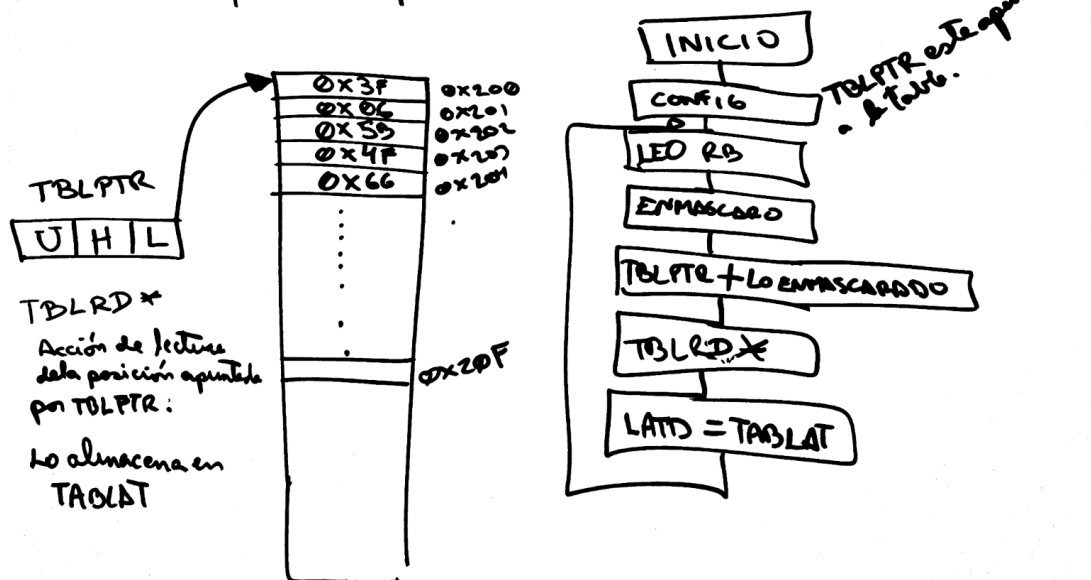
Instruction	7-segment pattern (hex)	Digit (decimal)
<code>addwf PCL, f</code>		
<code>retlw 0x3F</code>	0x3F	0
<code>retlw 0x06</code>	0x06	1
<code>retlw 0x5B</code>	0x5B	2
<code>retlw 0x4F</code>	0x4F	3
<code>retlw 0x66</code>	0x66	4
<code>retlw 0x6D</code>	0x6D	5
<code>retlw 0x7D</code>	0x7D	6
<code>retlw 0x07</code>	0x07	7
<code>retlw 0x7F</code>	0x7F	8
<code>retlw 0x67</code>	0x67	9
<code>retlw 0x79</code>	0x79	E
<code>retlw 0x79</code>	0x79	E
<code>retlw 0x79</code>	0x79	E
<code>retlw 0x79</code>	0x79	E
<code>retlw 0x79</code>	0x79	E

Handwritten annotations show the jump mechanism: a curved arrow from the `addwf PCL, f` instruction to the `retlw` instructions indicates the jump target. The label `PCL+6` is written near the jump, and `PCL+3` is written near the `retlw 0x7D ;6` instruction.



Ejercicio: Modelo de decodificador BCD-7Segmentos

Opción 2 Empleando el puntero de tabla



Ejercicio: Modelo de decodificador BCD-7Segmentos

```

org 0x0200  ← posición de la memoria de programa
tabla7s db 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0X7F, 0x67, 0x79, 0x79, 0x79, 0x79, 0x79, 0x79
;Disp7seg: 0 1 2 3 4 5 6 7 8 9 E E E E E E

org 0x0000 ;Vector de reset
goto configura

org 0x0020  ← zona de programa de usuario
configura:
    clrf TRISD ;Todo el puertoD como salida
    movlw UPPER tabla7s ;Para apuntar puntero de tabla TBLPTR hacia "tabla7s"
    movwf TBLPTRU ;(TBLPTRU:TBLPTRH:TBLPTL) -> 21 bits
    movlw HIGH tabla7s
    movwf TBLPTRH
    movlw LOW tabla7s
    movwf TBLPTL

inicio:
    clrf TBLPTL ;Colocamos el TBLPTR en la primera posición de "tabla7s"
    movf PORTB, W ;Leemos el RB y lo almacenamos en W
    andlw 0x0F ;Enmascaramos los cuatro primeros bits
    addwf TBLPTL ;Sumamos el contenido leído y enmascarado hacia el TBLPTR
    TBLRD* ;Acción de lectura del puntero (lo leído lo almacena en TABLAT)
    movff TABLAT, LATD ;El contenido de TABLAT lo enviamos al puerto RD
    goto inicio
end

```

Handwritten notes and arrows:

- An arrow points from "posición de la memoria de programa" to `org 0x0200`.
- An arrow points from "zona de programa de usuario" to `org 0x0020`.
- An arrow points from "Tabla de valores del decodificador" to the `tabla7s` data definition.

Ejercicio: Contador 0-9 autoincremental

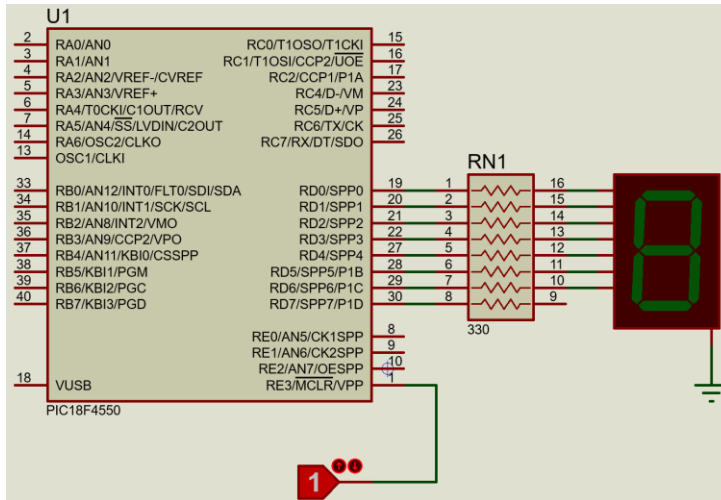
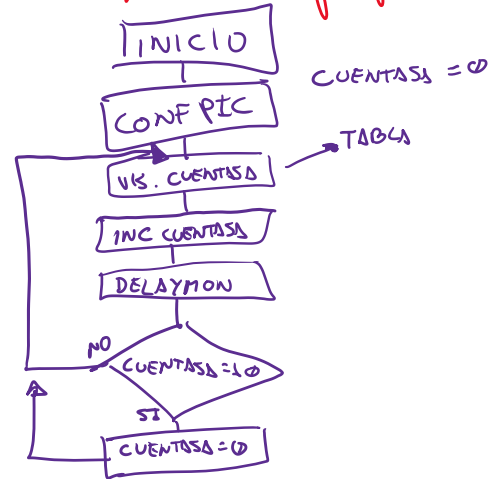


Diagrama de flujo:



Ejercicio: Contador 0-9 autoincremental

```

cblock 0x0020 ;Zona de declaración de etiquetas a los
cta_a ;registros GPR (variables)
cta_b
cta_c
cuentasa ;Registro de la cuenta del contador
endc

org 0x0200
tabla7s db 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67, 0x79, 0x79, 0x79, 0x79, 0x79, 0x79
;Disp7seg: 0 1 2 3 4 5 6 7 8 9 E E E E E E

org 0x0000 ;Vector de reset
goto configura

org 0x0020
configura:
clrf TRISD ;Todo el puertoD como salida
movlw UPPER tabla7s ;Para apuntar puntero de tabla TBLPTR hacia "tabla7s"
movwf TBLPTRU ;(TBLPTRU:TBLPTRH:TBLPTRL) -> 21 bits
movlw HIGH tabla7s
movwf TBLPTRH
movlw LOW tabla7s
movwf TBLPTRL
clrf cuentasa ;La cuenta inicia en cero
  
```

Ejercicio: Contador 0-9 autoincremental

```

inicio:
    call displayada
    incf cuentaasa, f
    call delaymon
    movlw .10
    cpfseq cuentaasa
    goto inicio
    clrf cuentaasa
    goto inicio

displayada:
    clrf TBLPTRL
    movf cuentaasa, W           ;Leemos el RB y lo almacenamos en W
    addwf TBLPTRL              ;Sumamos el contenido leído y enmascarado hacia el TBLPTR
    TBLRD+                     ;Acción de lectura del puntero (lo leído lo almacena en TABLAT)
    movff TABLAT, LATD         ;El contenido de TABLAT lo enviamos al puerto RD
    return
  
```

Nota: La declaración de las librerías, bits de configuración y la rutina de retardo se han obviado por ser las mismas que en los ejercicios anteriores

Asignación: Desarrollar un contador 00-79 autoincremental

