

Microcontroladores

Laboratorio Semana 1

Semestre: 2021-0

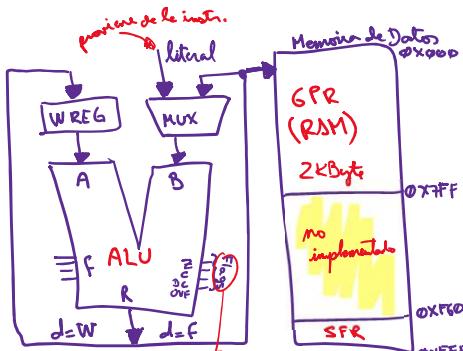
Profesor: Kalun José Lau Gan

1

Agenda

- El flujo de datos en el microcontrolador PIC18F4550
- Instrucciones básicas en MPASM
- Ejemplos básicos con el microcontrolador PIC18F4550 en lenguaje Assembler

El flujo de datos en el microcontrolador PIC18F4550



Nota: La configuración y uso de los puertos y periféricos se hace a través de registros SFR.

GPR: Registros de Propósito General
SFR: Registros de Funciones Especiales

TRISB, PORTB, LATB, ...

Access RAM	000h	05Fh	060h	07Fh	080h
GPR	000h	001h	002h	003h	004h
GPR	1FFh	200h			
GPR	2FFh	300h			
GPR	3FFh	400h			
GPR ⁽¹⁾	4FFh	500h			
GPR ⁽¹⁾	5FFh	600h			
GPR ⁽¹⁾	6FFh	700h			
GPR ⁽¹⁾	7FFh	800h			

F0Fh	TOSU	F0Dh	INFO2 ⁽¹⁾	F0Bh	COPR1H	F06h	IPR1	F0Fh	UEP15
F0Eh	TOSH	F0Dh	POSTINC2 ⁽¹⁾	F0Bh	COPR1L	F06h	PIR1	F0Eh	UEP14
F0Dh	TOSL	F0Dh	PREDEC2 ⁽¹⁾	F0Dh	COP1ICON	F06h	PIE1	F0Dh	UEP13
F0Cf	TOSL	F0Dh	PREINC2 ⁽¹⁾	F0Dh	COP2ICON	F06h	PIE2	F0Cf	UEP12
F0Bh	PCALTU	F0Dh	PREINC1 ⁽¹⁾	F0Bh	COP2ICON	F0Bh	OC2TUNE	F0Bh	UEP11
F0Ah	PCALTH	F0Ah	FSR2H	F0Bh	OC2TUNE	F0Ah	__00	F0Ah	UEP10
F0Bh	PCL	F0Bh	FSR2L	F0Bh	__00	F0Bh	__00	F0Bh	UEP9
F0Bh	TBLPTRU	F0Bh	STATUS	F0Bh	BAUDCON	F0Bh	__00	F0Bh	UEP8
F0Th	TBLPTRH	F0Th	TMR0H	F0Th	ECCP1DEL	F0Th	__00	F0Th	UEP7
F0Fh	TBLPTRL	F0Fh	TMR0L	F0Fh	ECCP1IAS	F0Fh	TRISB ⁽¹⁾	F0Fh	UEP6
F0Fh	TMOD	F0Fh	TCON	F0Fh	ECCP2DEL	F0Fh	TRISD ⁽¹⁾	F0Fh	UEP5
F0Ah	PRODH	F0Ah	ICON	F0Ah	ECCP2IAS	F0Ah	TRISC ⁽¹⁾	F0Ah	UEP4
F0Bh	PROOL	F0Bh	OCN0	F0Bh	RCSTA	F0Bh	TRISB	F0Bh	UEP3
F0Dh	PROOL	F0Dh	QSCCON	F0Dh	TMR3H	F0Dh	TRISB	F0Dh	UEP2
F0Dh	INTCON	F0Dh	HVDCON	F0Dh	TMR3L	F0Dh	TRISA	F0Dh	UEP1
F0Fh	INTCON	F0Dh	WDTCON	F0Bh	T3CON	F0Bh	__00	F0Bh	UEP0
F0Fh	INTCON3	F0Dh	RCON	F0Bh	SPBRGH	F0Bh	__00	F0Bh	UEP0
F0Fh	INTCON4	F0Dh	ADRESH	F0Bh	SPBRGL	F0Bh	__00	F0Bh	UEP0
F0Dh	POSTINC1 ⁽¹⁾	F0Dh	TMR1L	F0Bh	RCREG	F0Bh	__00	F0Bh	UAODR
F0Dh	POSTDEC1 ⁽¹⁾	F0Dh	T1CON	F0Ah	TXREG	F0Ah	LATE ⁽¹⁾	F0Ah	UCON
F0Cf	PREINC1 ⁽¹⁾	F0Cf	TMR2	F0Ah	TXSTA	F0Ah	LATD ⁽¹⁾	F0Ah	USTAT
F0Bh	PLSWW ⁽¹⁾	F0Bh	PR2	F0Bh	RCSTA	F0Bh	LATC	F0Bh	UEIE
F0Ah	FSR0H	F0Ah	T2CON	F0Ah	RCSTA	F0Ah	LATE	F0Ah	UEIR
F0Bh	FSR0L	F0Bh	SSP1ADD	F0Ah	EEDATA	F0Ah	LATE	F0Ah	UFBR
F0Bh	FSR1H	F0Bh	SSP1ADD0	F0Ah	EEDATA	F0Ah	UFBR	F0Ah	UFBR
F0Bh	FSR1L	F0Bh	SSP1STAT	F0Ah	EEDATA	F0Ah	UFBR	F0Ah	UFRLH
F0Bh	POSTINC1 ⁽¹⁾	F0Bh	SSP2CON1	F0Ah	EEC01 ⁽¹⁾	F0Ah	UFRLM	F0Ah	UFRLM
F0Bh	POSTDEC1 ⁽¹⁾	F0Bh	SSP2CON0	F0Ah	EEC01 ⁽¹⁾	F0Ah	UFRLM	F0Ah	UFRLM
F0Cf	PREINC1 ⁽¹⁾	F0Cf	SSP2STAT	F0Ah	UFRLM	F0Ah	UFRLM	F0Ah	UFRLM
F0Ah	ADRESH	F0Ah	UFRLM	F0Ah	UFRLM	F0Ah	UFRLM	F0Ah	UFRLM
F0Bh	ADRESL	F0Ah	UFRLM	F0Ah	UFRLM	F0Ah	UFRLM	F0Ah	UFRLM
F0Ah	FSR1H	F0Ah	ADRESH	F0Ah	UFRLM	F0Ah	PORTD ⁽¹⁾	F0Ah	SPPEPS ⁽¹⁾
F0Ah	FSR1L	F0Ah	ADCON1	F0Ah	UFRLM	F0Ah	PORTD ⁽¹⁾	F0Ah	SPPEPS ⁽¹⁾
F0Bh	BSR	F0Bh	ADCON2	F0Ah	UFRLM	F0Ah	PORTA	F0Ah	SPPDATAR ⁽¹⁾
F0Bh	ADC0N2	F0Ah	UFRLM	F0Ah	UFRLM	F0Ah	PORTA	F0Ah	UFRLM

Todos los registros son de 8bits

Partes de un programa en MPASM

```

list p=18f4550
#include<p18f4550.inc>

; aqui declaramos los b:
CONFIG FOSC = XT_XT
CONFIG PWRT = ON
CONFIG BOR = OFF
CONFIG WDT = OFF
CONFIG PBADEN = OFF
CONFIG LVP = OFF

.org 0x0000
goto configuro

.org 0x0020
configuro:
    bsf TRISB, 0
    bcf TRISD, 0

principal:
    btfss PORTB, 0
    goto principal
    btg LATD, 0

otro:
    btfsc PORTB, 0
    goto otro
    goto principal

end

```

lado

Configuraciones

Programa

Ejemplo: Titilar un LED conectado en RD2

Hardware:

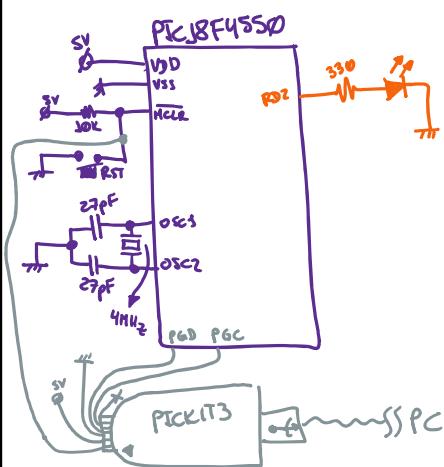
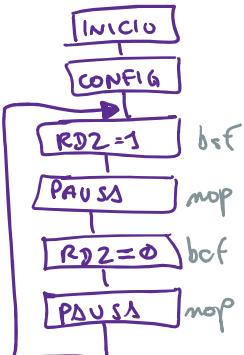


Diagrama de Flujos:



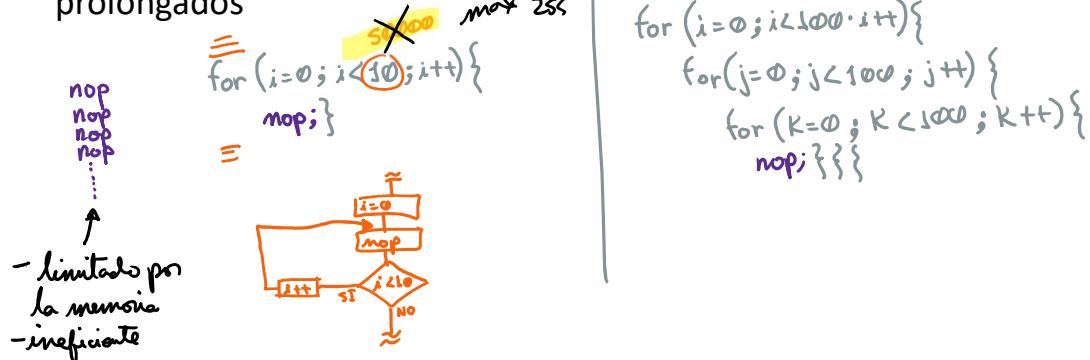
Note:
 $T_{ejec\ instr.} = \left(\frac{F_{osc}}{4}\right)^{-1}$
 $F_{osc} = 4\text{MHz}$
 $\Rightarrow T_{ejec\ instr.} = 1\text{ }\mu\text{s}$

Tenemos que alargar el tiempo
¿Puedo colocar 50000 'nop'?

29

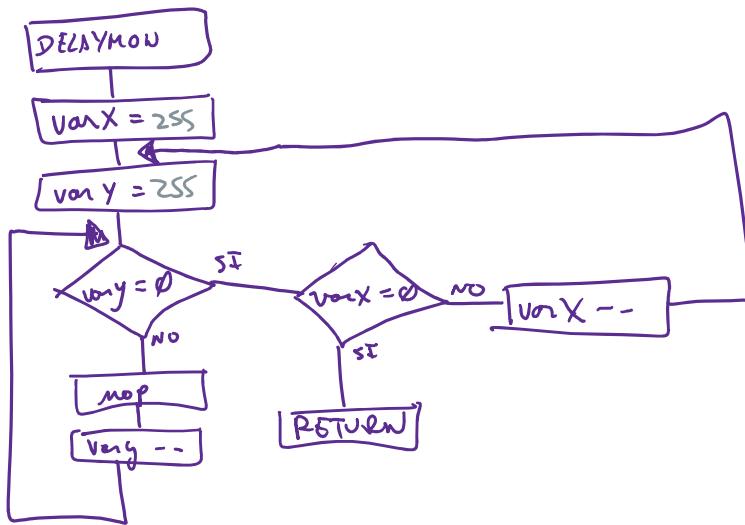
Ejemplo: Titilar un LED conectado en RD2

- Debemos de crear un bucle de repetición para generar retardos más prolongados



30

Propuesta de algoritmo con dos anillos de repetición



31

Código del titilador de LED en MPASM:

<pre> 1 list p=18f4550 2 #include <p18f4550.inc> ;libreria de nombre de la 3 4 CONFIG FOSC = XT_XT 5 CONFIG CPUDIV = OSC1_PLL2 6 CONFIG PWRT = ON ; Power-up Timer Enabled 7 CONFIG BOR = OFF ; Brown-out Reset Enabled 8 CONFIG WDT = OFF ; Watchdog Timer Enabled 9 CONFIG PBDEN = OFF ; PORTB A/D Enable bit 10 CONFIG LVP = OFF ; Single-Supply ICSP Enabled 11 12 cblock 0x000 13 var_i 14 var_j 15 var_k 16 endc 17 18 org 0x0000 ;Vector de reset 19 goto configuro 20 21 org 0x0020 ;Zona de programa de usuario 22 configuro: 23 bcf TRISD, 2 ;RD0 como salida 24 25 inicio: 26 bsf LATD, 2 ;prendo el led 27 call delaymon 28 bcf LATD, 2 ;apago el led 29 call delaymon 30 goto inicio </pre>	<pre> 32 delaymon: 33 movlw .50 34 movwf var_i 35 otro1: 36 call bucle1 ;Salto a subrutina 37 decfsz var_i,f 38 goto otro1 39 return 40 41 bucle1: 42 movlw .55 43 movwf var_j 44 otro2: 45 nop 46 nop 47 call bucle2 ;Salto a subrutina 48 decfsz var_j,f 49 goto otro2 50 return 51 52 bucle2: 53 movlw .20 54 movwf var_k 55 otro3: 56 nop 57 decfsz var_k,f 58 goto otro3 59 return 60 end </pre>
--	--

32

Prototipo físico del circuito:



33

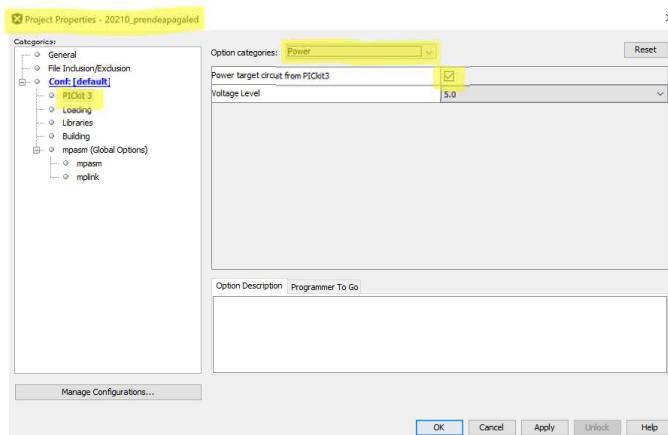
Recomendaciones al momento de implementar el circuito en físico:

- Verificar continuidad en los cables jumper utilizados en el circuito
- Verificar que la resistencia en pin MCLR hacia 5V sea de 10K
- Verificar que la PC haya detectado correctamente el PICKIT3
- Revisar siempre los mensajes en la ventana de “output” por posibles fallos en el evento de compilación y evento de programación.
- Tener a la mano un multímetro para verificar voltajes y continuidad en el prototipo.

34

Recomendaciones al momento de implementar el circuito en físico:

- Si se está empleando el PICKIT3 como programador y deseas que éste administre el voltaje de alimentación al circuito de prueba:



35

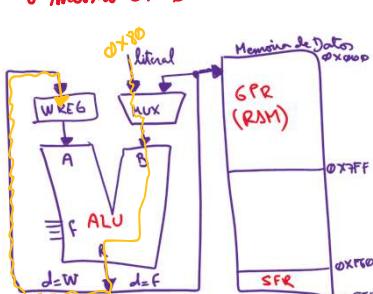
Instrucciones básicas en MPASM

- Instrucciones de movimiento de datos

movlw [literal]

- mover un literal hacia WREG

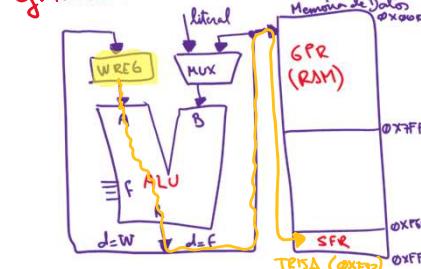
Ej: *movlw 0x80*



movwf [registro]

- mover el contenido de WREG hacia [registro]

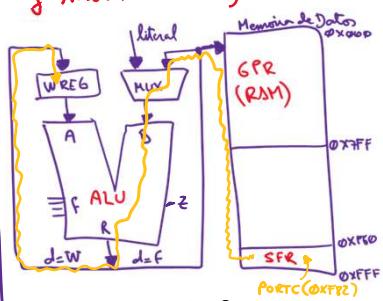
Ej: *movwf TRISA*



movf [registro], d

mover el contenido de [registro] y lo muere según "d".

Ej: *movf PORTC,W*

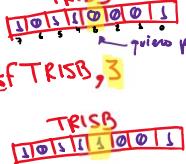


Note: *movf [registro], f* sirve para act. flag

36

Instrucciones básicas en MPASM

- Instrucciones de manipulación de bits en un registro

bsf [registro], #bit *posición*
coloca a "1" el #bit de [registro]
Ej.: 

bcf [registro], #bit
coloca a "0" el #bit de [registro]
Ej.: 

btg [registro], #bit
aplica complemento al #bit de [registro]
Ej.: btg TRISB, 6 

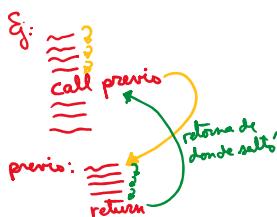
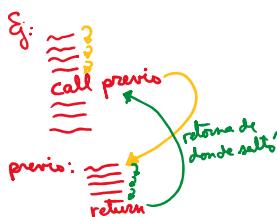
37

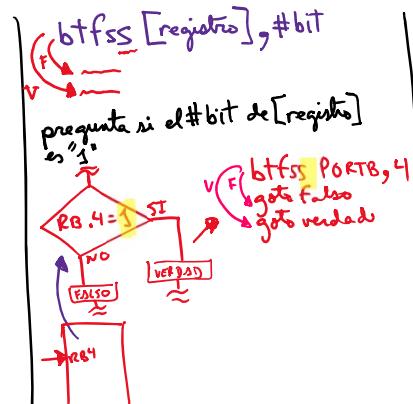
Instrucciones básicas en MPASM

goto [etiqueta]
salto incondicional

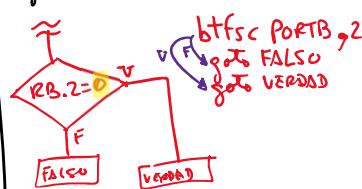
Ej.: bucle: bsf LATD, 6
goto bucle

call [etiqueta]
salto a subrutina

Ej.: 
previo: 



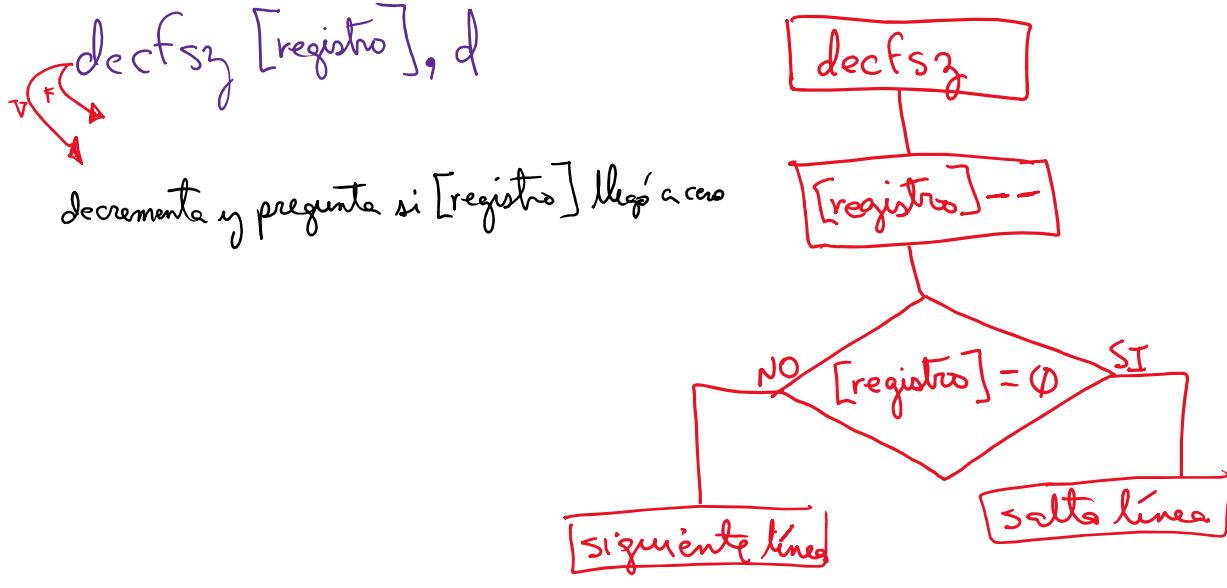
btfsc [registro], #bit
pregunta si el #bit de [registro] es igual a "0"



mop
instrucción de no operación
lo podemos considerar como un micro retraso

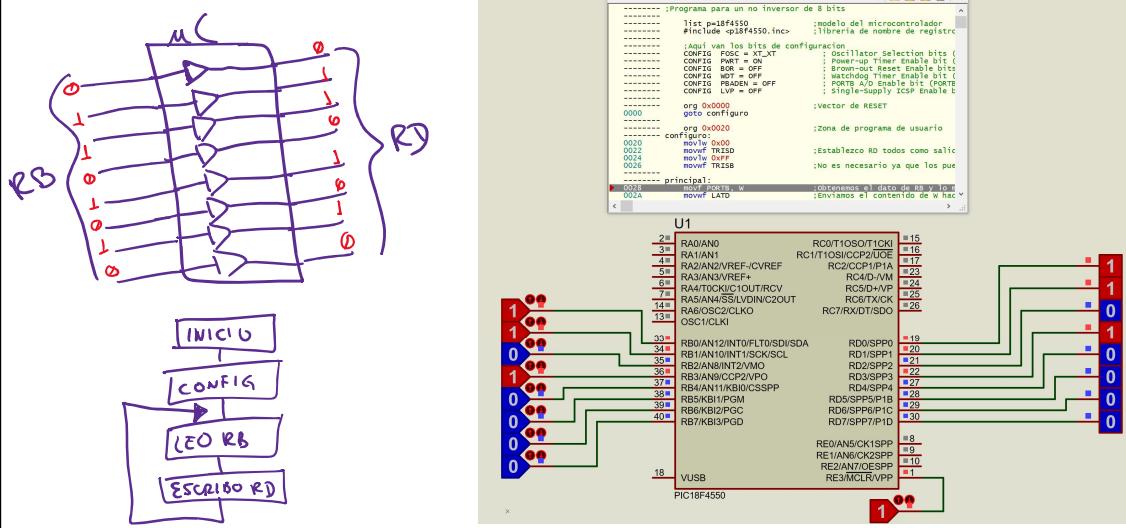
38

Instrucciones básicas en MPASM



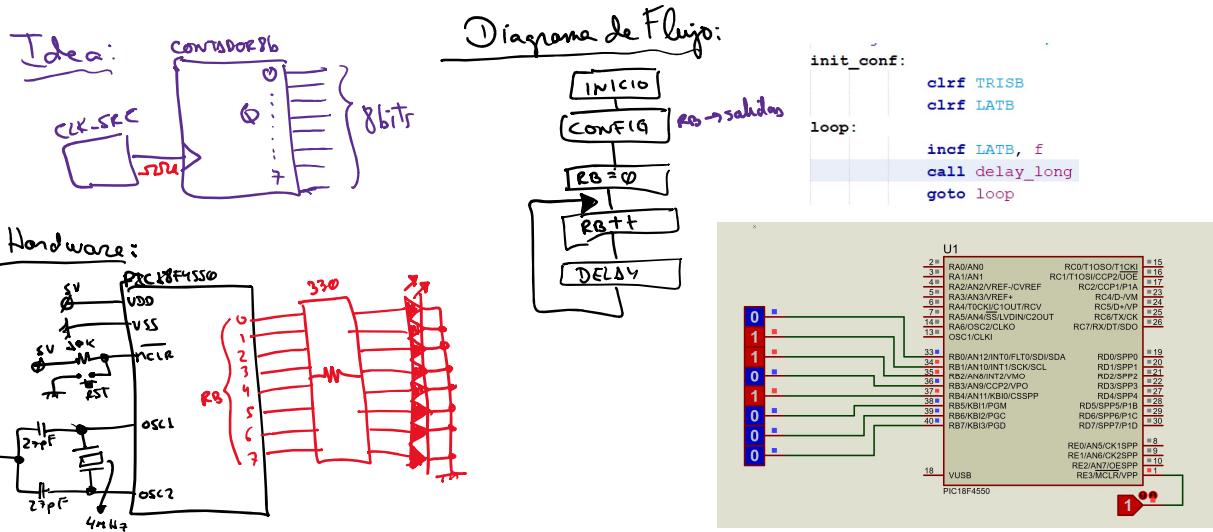
39

Ejemplo: no-inversor de 8 bits



40

Modificar el titileo de LED para que se visualice un contador autoincremental de 8 bits



41

Ejercicios adicionales:

Tener en cuenta que se debe de seguir el procedimiento visto en clase (idea, desarrollo del circuito, diagrama de flujo, código en MPASM, simulación, prototipo físico)

- Desarrollar un titilador de un LED conectado en RE0 en el cual su periodo de parpadeo dependerá del estado de un switch conectado en RB0, si RB0=1 el periodo será de 500ms, si RB0=0 el periodo será de 100ms.
- Desarrollar una señal de cruce de tren con entrada de activación.
- Desarrollar una “vela electrónica” con entrada de activación, dicha entrada tendrá como sensor de luz a un L.D.R.

42

Fin de la sesión!