

Sistemas Digitales: Introducción a los microcontroladores

Ingeniería Electrónica
UPC 2018

Por Kalun Lau



- Microcontroladores →
- Dispositivos programable "todo en uno"
- para desarrollar aplicaciones en electrónica,
- Es "la solución en un chip" muy específicas

Ej:

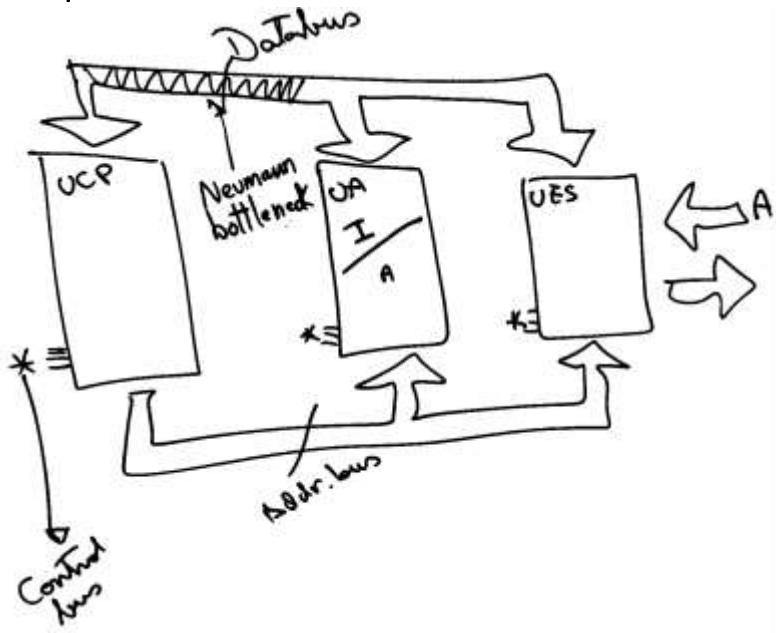


- Tiene un montón de periféricos:
 - Mouse - Soundcard
 - Keyboard - Video card
 - HDD - DVD
 - Ethernet - USB

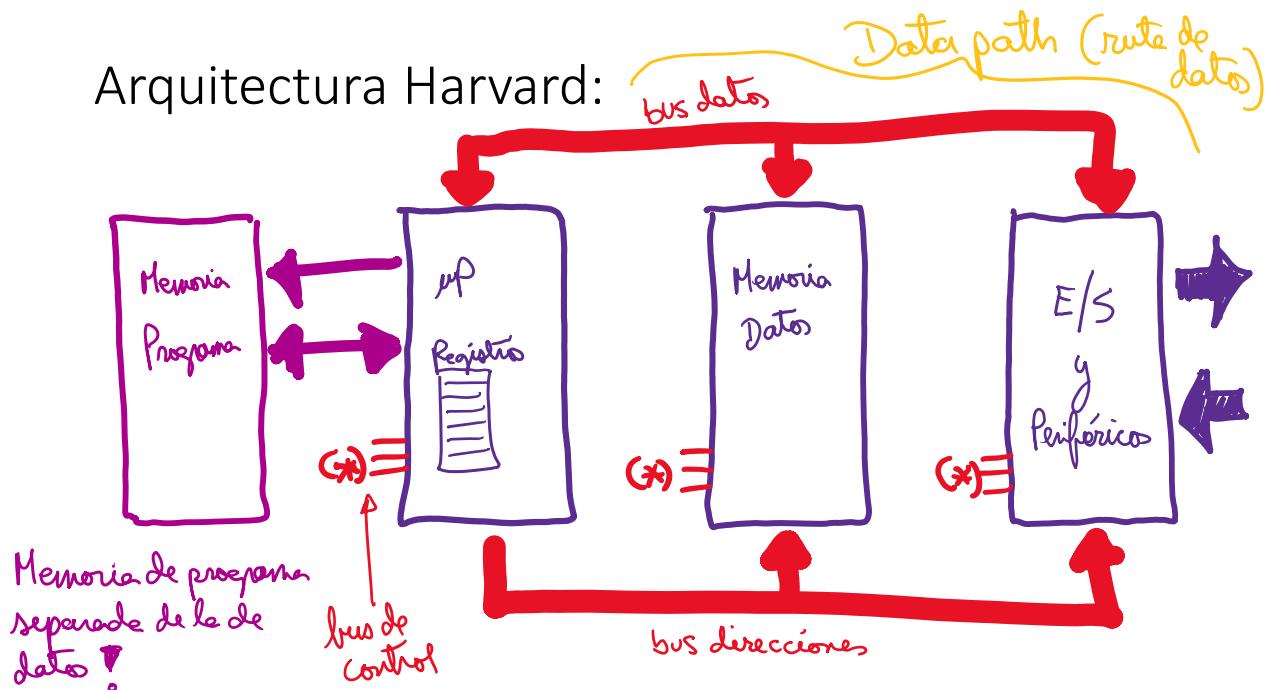
Fabricantes: (Microchip, Renesas, NXP, ST, etc)



Arquitectura Von Neumann



Arquitectura Harvard:



Nosotros usaremos el PIC18F4550!

40-Pin PDIP



cuarenta pines!

Tecnología
CMOS

$$\begin{aligned} VDD &= 5V \\ VSS &= 0V \end{aligned}$$

$$\begin{aligned} VDD_{\minimo} &: \\ &3.0V \text{ (F)} \\ &2.0V \text{ (LF)} \end{aligned}$$

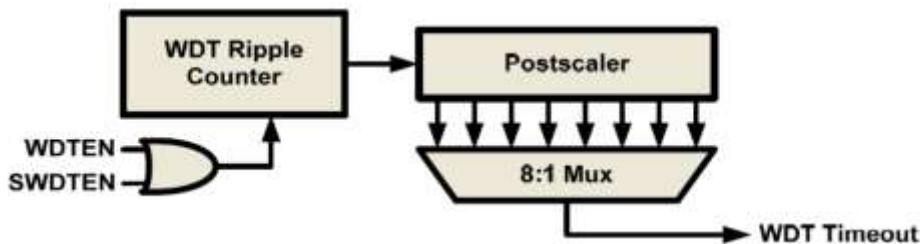
$$\begin{aligned} VDD_{\maximo} &: \\ &5.5V \text{ (F y LF)} \end{aligned}$$

Características Especiales PIC18F

- Amplio Voltaje de operación: 2.0V a 5.5V
- Memoria de Programa Flash Mejorada con 100,000 ciclos de borrado/escritura
- Memoria de Datos EEPROM con 1,000,000 de ciclos de borrado/escritura
- Retención de Datos en Memoria EEPROM Flash/Data : 100 años típico

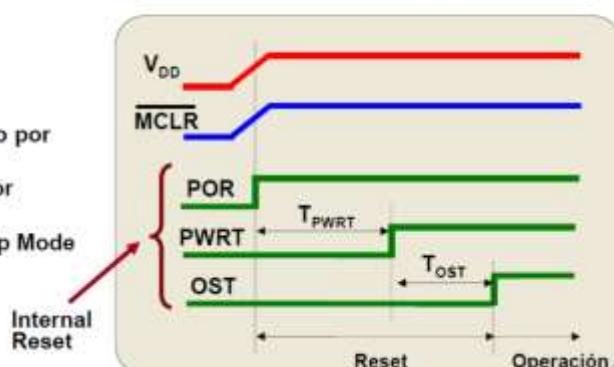
Watchdog Timer

- Ayuda al software a recuperarse de un mal funcionamiento
- Usa un oscilador libre RC en el chip
- WDT es borrado por la instrucción CLRWDT
- WDT Habilitable (WDTEN) no puede ser borrado por soft
- el overflow (desborde) del WDT reactiva al chip
- Período del timeout programable : 18ms a 3.0s típico
- Opera en modo SLEEP; sobre el time out, despierta la CPU.



Generador de Reset Interno

- **POR: Power On Reset**
 - con MCLR atado a V_{DD} , se genera un pulso de RESET cuando el flanco de subida a V_{DD} es detectado
- **PWRT: Power Up Timer**
 - 72 ms (nominal)
 - Desacoplado del BOR
- **OST: Oscillator Startup Timer**
 - Mantiene en RESET al dispositivo por 1024 cyclos de maquina (TCYs)
 - Le permite al cristal o al resonador estabilizarse
 - Bypassado en: Two Speed Startup Mode
 - INTOSC usa un clock para el procesador no estable



BOR –Brown Out Reset

- Cuando el voltaje cae por debajo de un umbral particular, el dispositivo se pone en RESET
- Impide el funcionamiento irregular o inesperado
- Elimina la necesidad de un circuito externo BOR

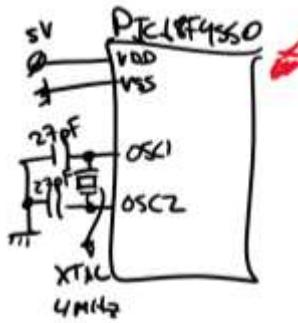
PBOR – Programmable Brown Out Reset

- Opción de configuración (fijado en tiempo de programa)
 - No puede ser activado/desactivado por software
- Cuatro puntos de disparo BVDD seleccionables
 - 2.5V – Mini V_{DD} para OTP MCUs PICmicro®
 - 2.7V
 - 4.2V
 - 4.5V
- Para otros umbrales use un supervisor externo (MCP1xx, MCP8xx/TCM8xx, or TC12xx)

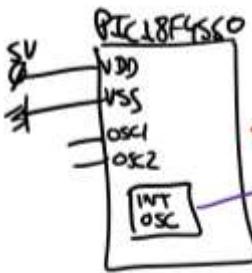
Fuentes de reloj:

- Cristal externo:

4 MHz, 8 MHz, 10 MHz
20 MHz hasta 48 MHz



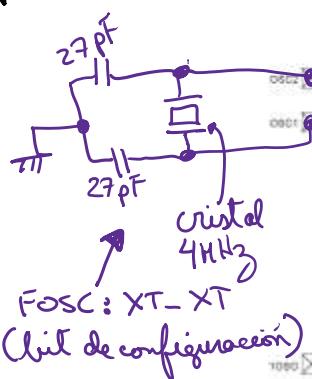
- Oscilador interno: 8 MHz configurable



oscilador RC

varía su precisión
con la temperatura

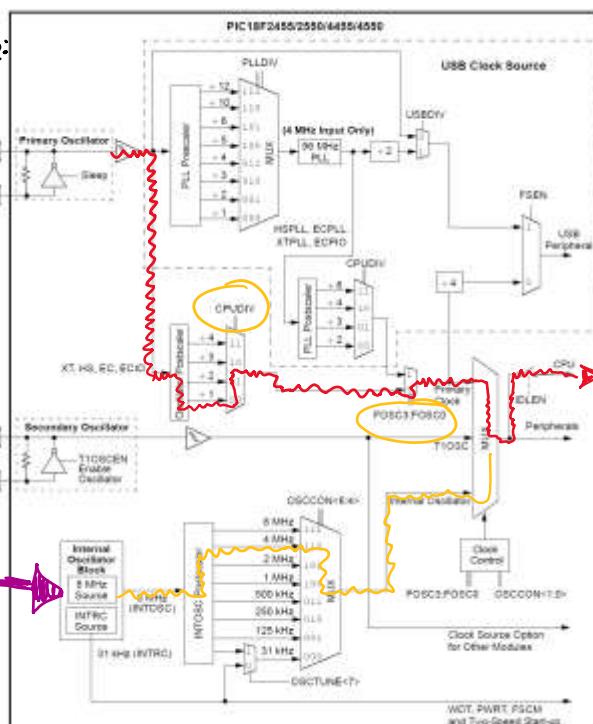
Configuración inicial a detalle:



FOSC: XT-XT
(bit de configuración)

Opción para usar el
oscilador interno:

FOSC: INTOSC - EC
(bit de configuración)



PLL: Sistema
que permite
multiplicar la
frecuencia del
cristal externo.

Ojo: Solo se puede
usar una fuente
de reloj

¿Cómo se programa?

Lenguaje ensamblador (MPASM)

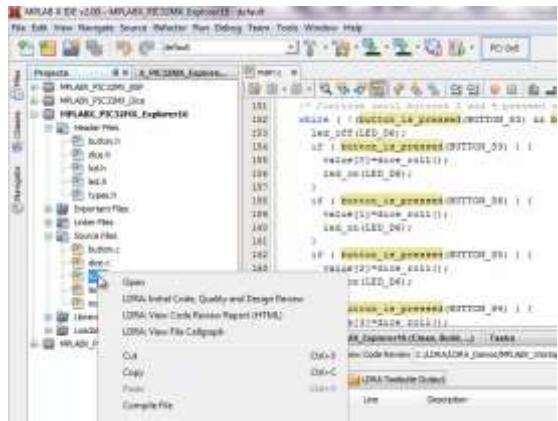
→ PBasic compiler (melabs.com, USD 249.99)

→ C → XC8 (microchip.com)

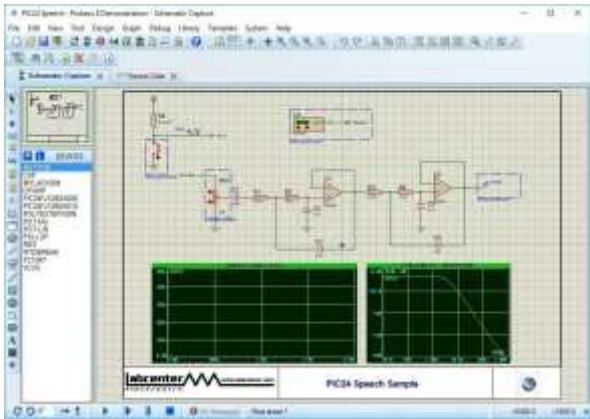
→ CSCSC Compiler

→ mikroC compiler (mikroe.com)

Herramientas de desarrollo para el microcontrolador PIC18F4550



Microchip:
MPLAB X (última versión: 5.05)
(www.microchip.com/mplab-x-ide)

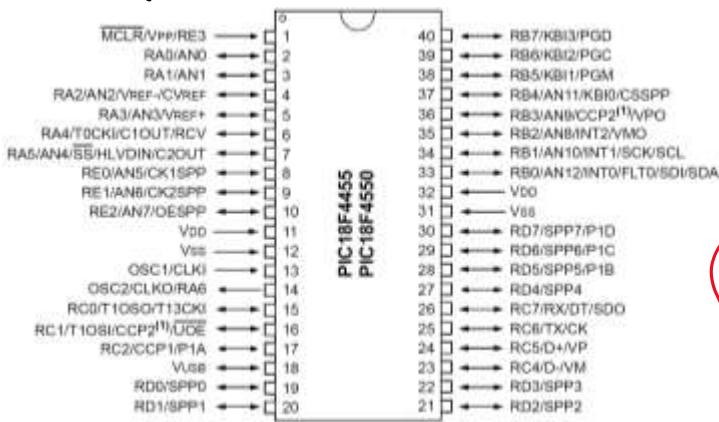


Labcenter: ProTens VSM (virtual system modelling)

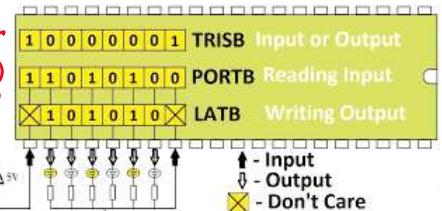
Se desarrollará el código

Se Marán
las simbolaciones

Manejo de puertos de E/S en el microcontrolador PIC18F4550:



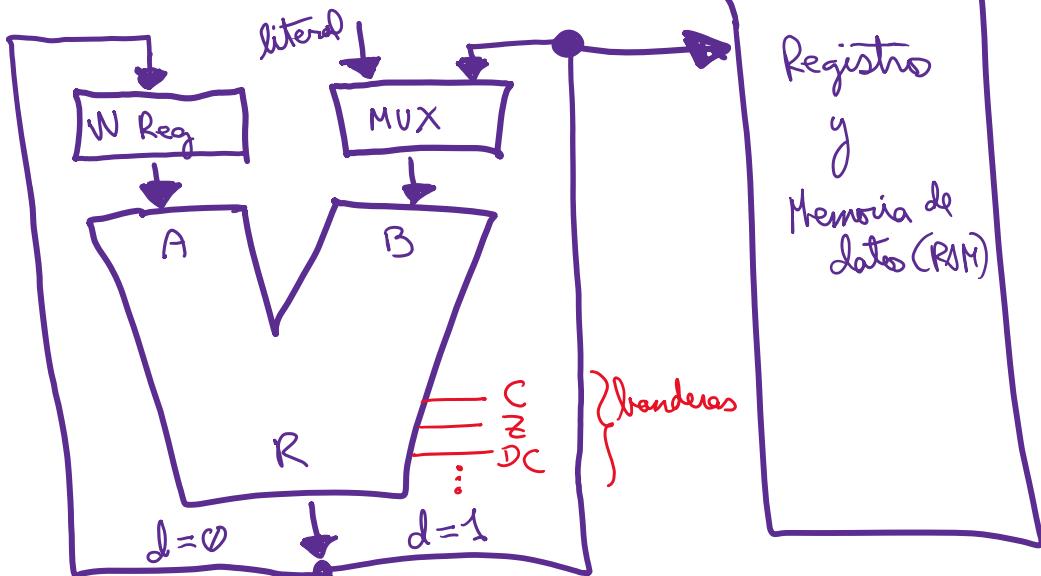
TIP: Declarar si es E o S antes de usar el puerto.



- En modo digital E/S: R_XY_Z
- Algunos puertos son o solo "E" o solo "S".

↑
letra del puerto ↑
 número del puerto

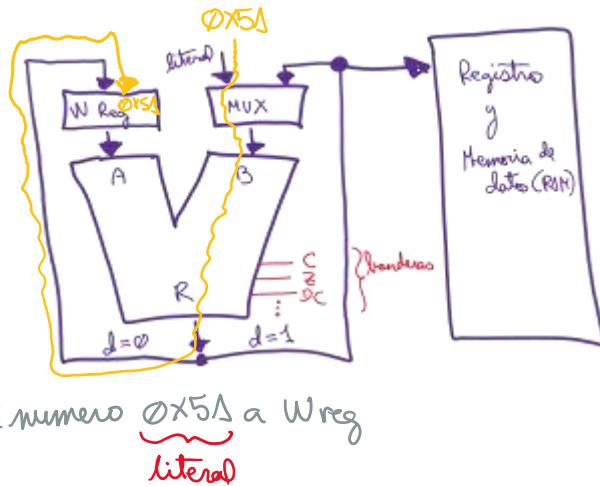
Ruta de datos del PIC18F4550



MPASM: Instrucciones de movimiento de datos

MOVLW Move literal to W	
Syntax:	[label] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → W
Status Affected:	None
Encoding:	0000 1110 KKKK KKKK
Description:	The eight-bit literal 'k' is loaded into W.
Words:	1
Cycles:	1
Q Cycle Activity:	
	Q1 Q2 Q3 Q4
	Decode Read Literal k Process Data Write to W

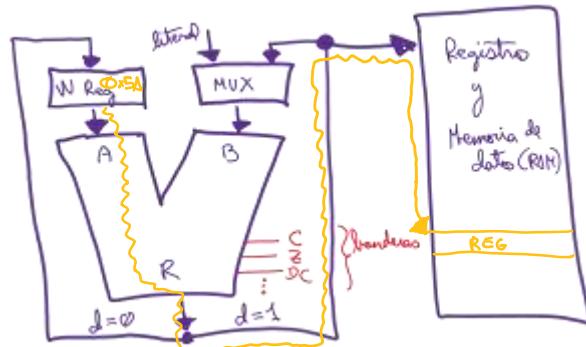
Example: MOVLW 0x5A
 After Instruction
 W = 0x5A



MPASM: Instrucciones de movimiento de datos

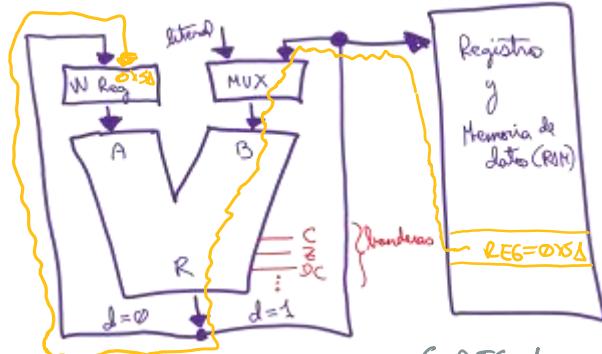
MOVWF Move W to F	
Syntax:	[label] MOVWF f.[a]
Operands:	0 ≤ f ≤ 255 a = [0,1]
Operation:	(W) → f
Status Affected:	None
Encoding:	0110 1110 ffff ffff
Description:	Move data from W to register T. Location 'T' can be anywhere in the 256 byte bank. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).
Words:	1
Cycles:	1
Q Cycle Activity:	
	Q1 Q2 Q3 Q4
	Decode Read register T Process Data Write register T

Example: MOVWF 000, 0
 Before Instruction
 W = 0x4F
 REG = 0xFF
 After Instruction
 W = 0x4F
 REG = 0x4F



MPASM: Instrucciones de movimiento de datos

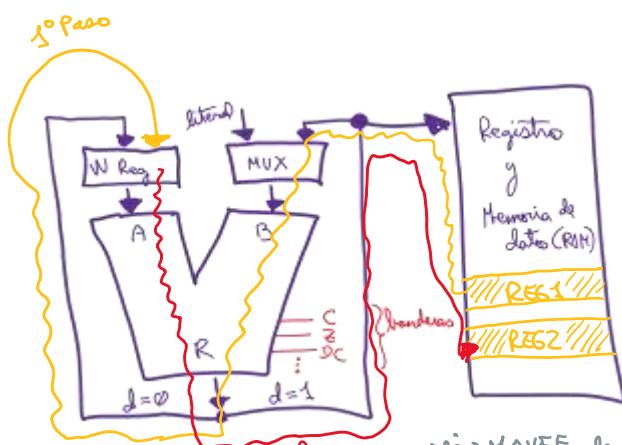
MOVF		Move F
Syntax:	(label) MOVF T,d(a)	
Operands:	0 ≤ d ≤ 255 0 ≤ T ≤ 1 a ∈ {0,1}	
Operator:	T ← dest	
Status Affected:	N, Z	
Encoding:	0001 0000 CFFF FFFF	The contents of register T are moved to a destination dependent upon the status of Y. If Y is 0, the result is placed back in register T (labeled). Location T can be anywhere in the 256 byte bank. If a is 0, the A/B/C Bank will be selected, overriding the BSR value. If a = 1, the bank will be selected as per the BSR value (labeled).
Words:	1	
Cycles:	1	
O Cycle Activity:	C1 Decode C2 Read register T C3 Process Data C4 Write W	
Example:	MOVF R0, 1, 0	
Before Instruction	REG W = 0x22	
After Instruction:	REG W = 0x22	



Mueve el contenido de REG1 hacia W ($d=0$) / si movf REG,1 : moverá el contenido hacia si mismo!
Actualizará el estado de los banderas!

MPASM: Instrucciones de movimiento de datos

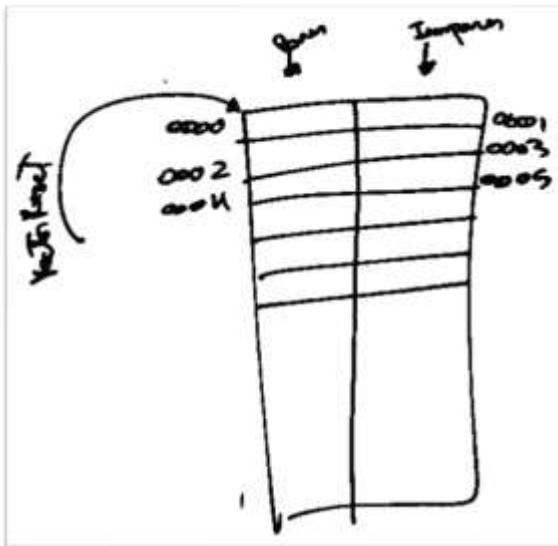
MOVFF		Move F to F
Syntax:	(label) MOVFF T ₁ ,T ₂	
Operands:	0 ≤ T ₁ ≤ 4095 0 ≤ T ₂ ≤ 4095	
Operator:	T ₂ → T ₁	
Status Affected:	N	
Encoding:	0000 0000 0000 0000	The contents of source register T ₁ are received to destination register T ₂ . Location of source T ₁ can be anywhere in the 256 byte data space (0000 to FFFF), and location of destination T ₂ can also be anywhere from 0000 to FFFF.
Description:	This instruction is useful for moving data between memory locations, or to a peripheral register (such as the timer buffer or an I/O port). The MOVFF instruction cannot use the PCL, TBLW, TBHF or TBLB as the destination register.	This instruction is useful for moving data between memory locations, or to a peripheral register (such as the timer buffer or an I/O port). The MOVFF instruction cannot use the PCL, TBLW, TBHF or TBLB as the destination register.
Note:	The MOVFF instruction should not be used to move data between memory locations if any interrupt is enabled. See section 8.6 for more information.	
Words:	2	
Cycles:	2 (D)	
O Cycle Activity:	C1 Decode C2 Read register T ₁ C3 No operation C4 Write register T ₂	
Example:	MOVFF R0, R1, H00	
Before Instruction:	R0 = 0x00 R1 = 0x00	
After Instruction:	R0 = 0x00 R1 = 0x00	



Mueve el contenido de REG1 hacia REG2

ojo: MOVFF demora dos ciclos en ejecutarse

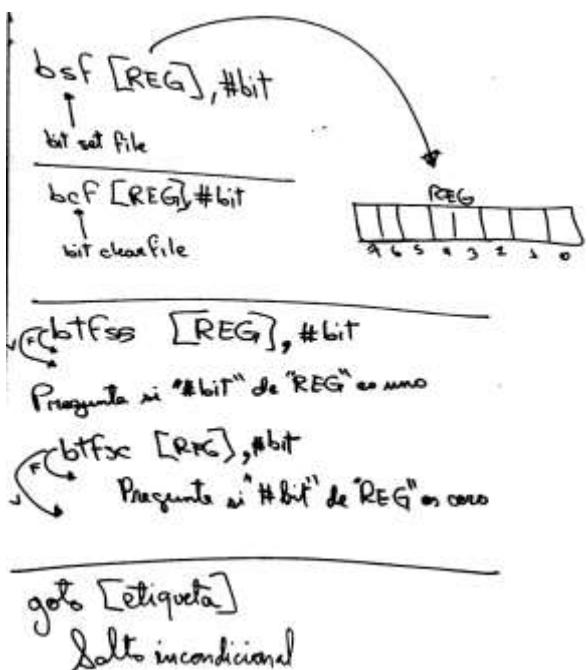
Estructura de la memoria de programa



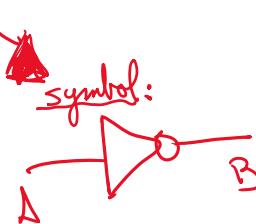
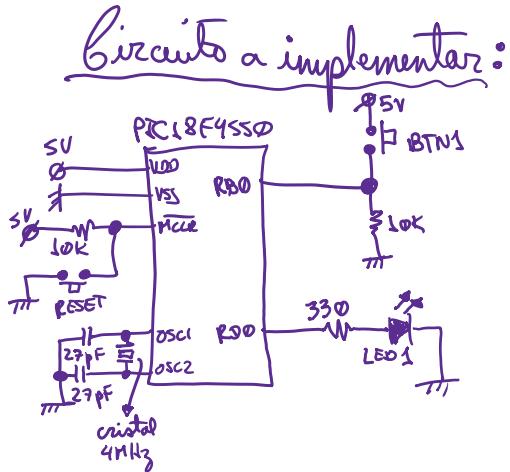
- Memoria de programa de 8 bits de ancho, 32K de capacidad.
- Las instrucciones ocupan 2 bytes (cada una)
- El contador de programa (PC) se incrementa de 2 en 2.
- Se puede usar la memoria de programa para almacenar datos (8 bits)

MPASM - Instrucciones

- Manipulación de bit en un registro (bsf, bcf)
- Pregunta de bit en un registro (btfs, btfsc)
- Salto incondicional (goto)



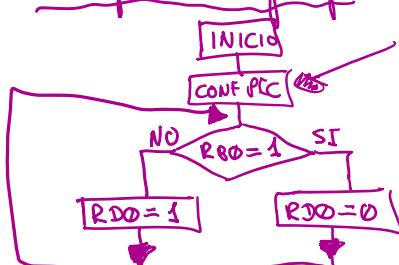
Ejemplo: Negador lógico



truth table:

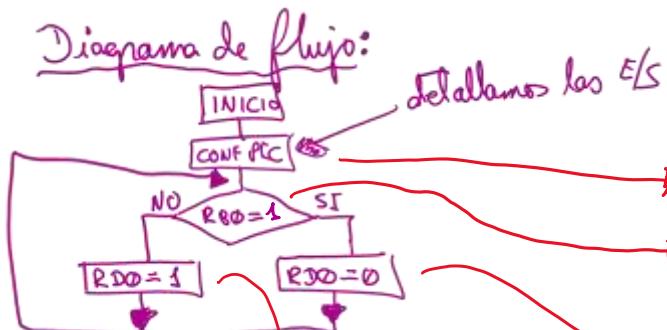
A	B
0	1
1	0

Diagrama de flujo:



detallamos los E/S

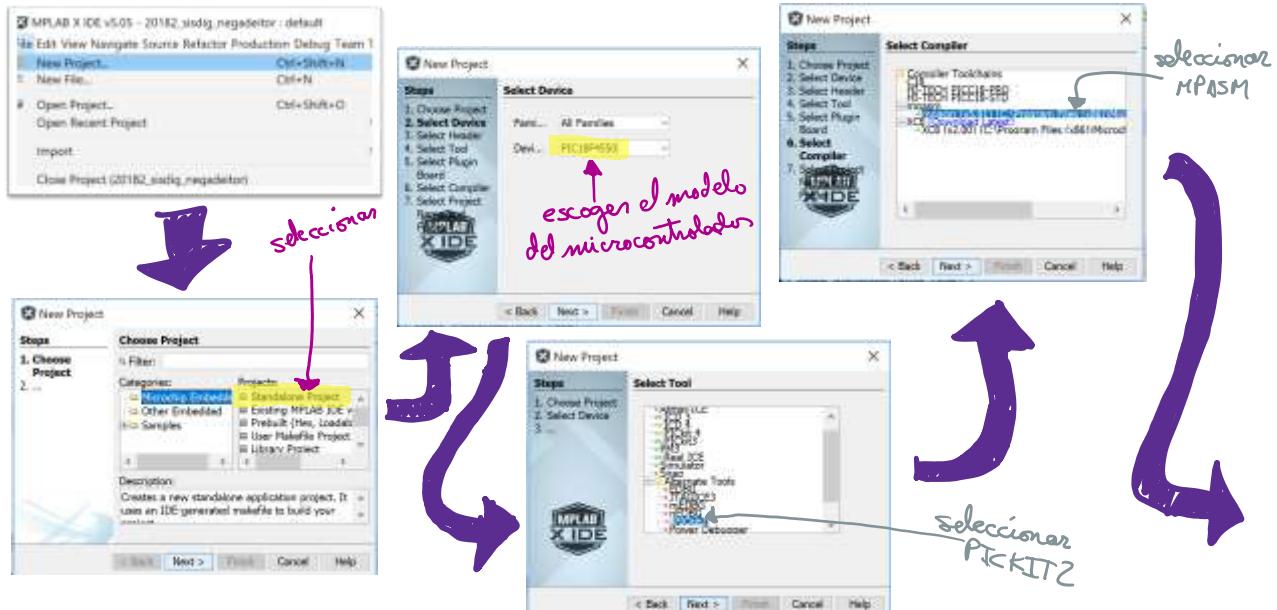
Ejemplo: Negador lógico



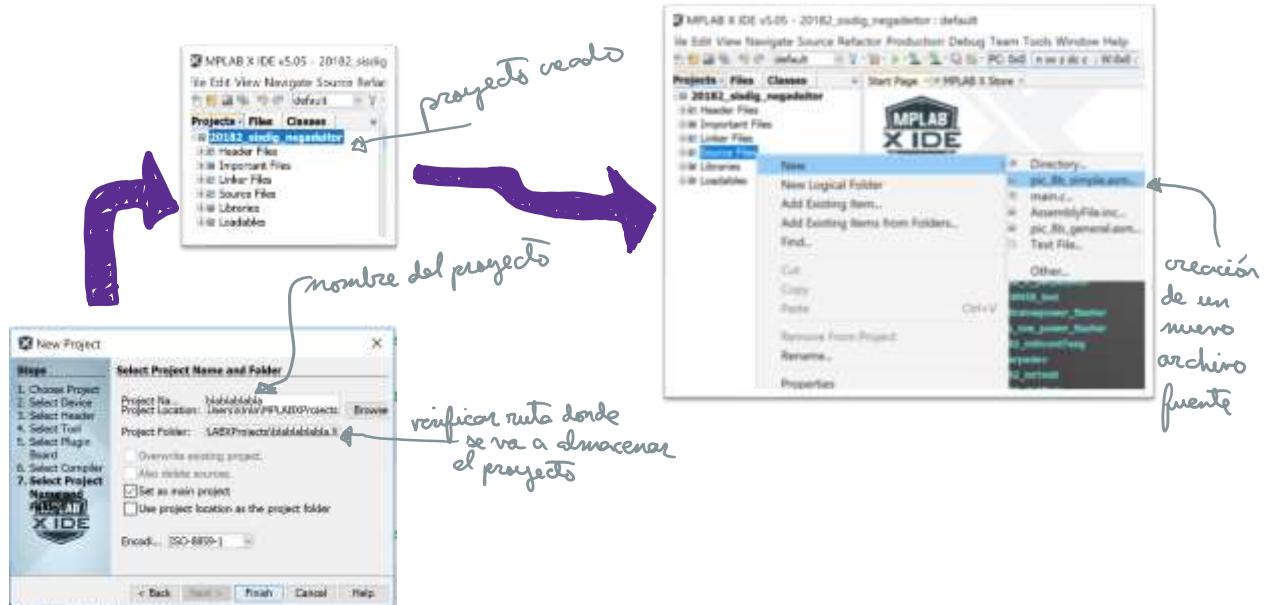
Código en MPASM:

- configuro: bcf TRISD, 0
- inicio: btfss PORTB, 0
goto falso
- verdadero: bcf LATD, 0
goto inicio
- falso: bsf LATD, 0
goto inicio

Creación del proyecto en el MPLAB X



Creación del proyecto en el MPLAB X

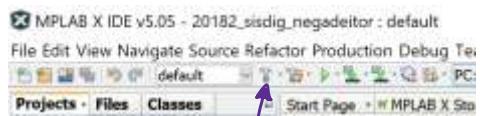


Plantilla y código del programa

```

1      list p=18F4550          ;Modelo del microcontrolador
2      #include <pic18f4550.h>    ;librería de nombres.
3
4      /*Zona de los bits de configuración del microcontrolador
5      CONFIG_FOSC = XT_XT;           ; Oscillator Selection bits (XT oscillator (XT))
6      CONFIG_PMRST = ON;            ; Power-up Timer Enable bit (PMEN enabled)
7      CONFIG_BOREN = OFF;           ; Brown-out Reset Enable bits (Brown-out Reset disabled in hardware and software)
8      CONFIG_WDT = OFF;             ; Watchdog Timer Enable bit (WDT disabled (control is placed on the SWDTEN bit))
9      CONFIG_PFBADEN = OFF;         ; PORTB A/D Enable bit (PORTB4:0 pins are configured as digital I/O on Reset)
10     CONFIG_LVPEE = OFF;           ; Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
11
12     org 0x0000                  ;Vector de reset
13     goto configura
14
15     org 0x0020                  ;Zona de programa de usuario
16 configura:
17     bcf TRISD, 0                ;Puerto D0 como salida
18     bcf TRISE, 0                ;Puerto D0 como entrada
19 inicio:
20     btfss PORTB, 0
21     goto falson
22 verdaderon:
23     bcf LATD, 0
24     goto inicio
25 falson:
26     bcf LATD, 0
27     goto inicio
28
29 end

```



Ventana de salida (output)

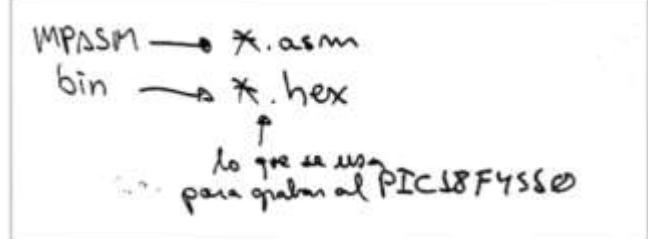
```

PICDEM100 - L74700E
Atmel AVRStudio Version 1.44
Copyright © 1999-2011 Atmel Corporation. All rights reserved.
Version 1.44

HEX2LDR - C974 to HEX File Converter
Copyright © 1999-2011 Atmel Corporation. All rights reserved.
Version 1.0

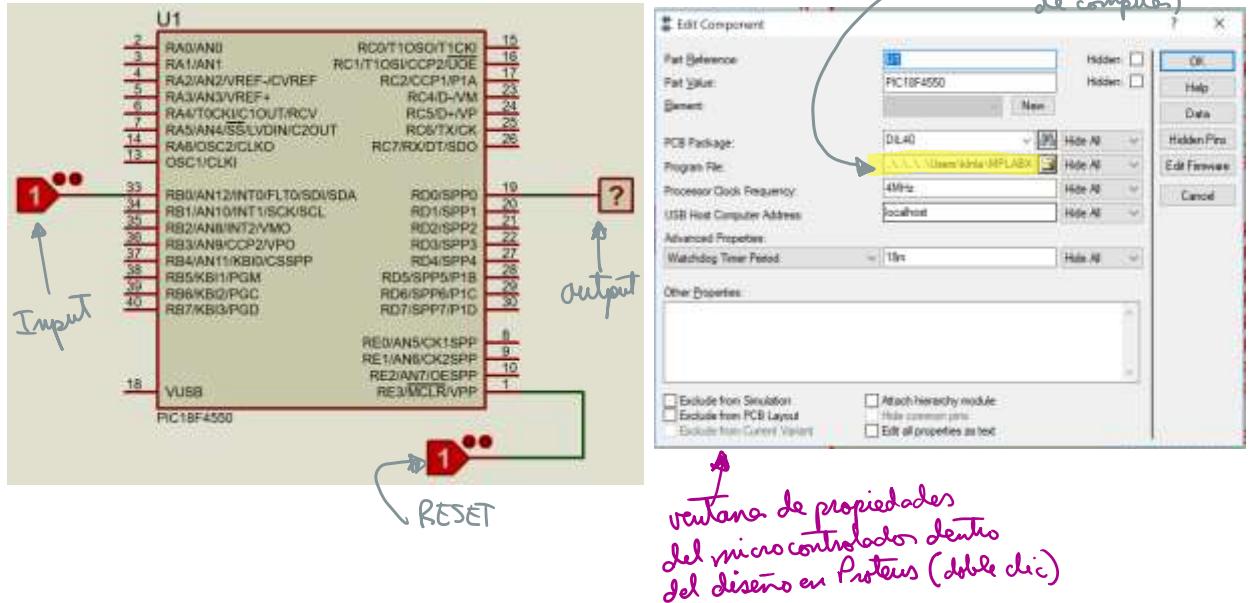
atmel(l1) :> listing directory "C:\Users\Kikia\My Documents\Atmel\AVRStudio\Projects\20182_sisdig_negadeitor\20182_sisdig_negadeitor"
atmel(l1) :> listing directory "C:\Users\Kikia\My Documents\Atmel\AVRStudio\Projects\20182_sisdig_negadeitor\20182_sisdig_negadeitor.xproj\product\bin\20182_sisdig_negadeitor\20182_sisdig_negadeitor.hex"
listing completed

```

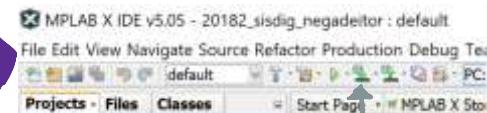
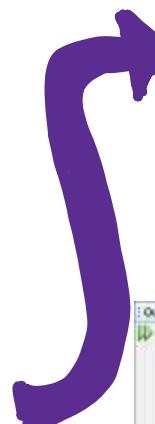
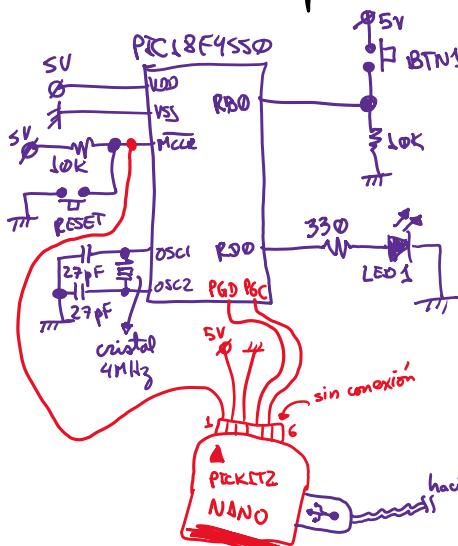


si "build successful" \Rightarrow
se creará el archivo
*.hex

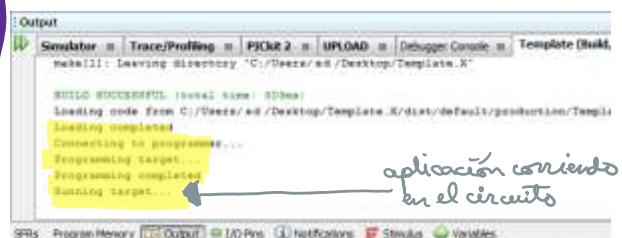
Simulación en Proteus VSM



Conexión del PICKIT2 con el circuito de la aplicación:



botón para grabar el .hex en el microcontrolador usando el PICKIT2



aplicación corriendo en el circuito

revisar que el PICKIT2 se encuentre reconocido por el S.O. Windows