

Microcontroladores

Semestre: 2023-1

Profesor: Kalun José Lau Gan

Semana 2: El Microcontrolador PIC18F45K50

1

Agenda:

- El CPU del PIC18Fxxxx, la memoria de programa y memoria de datos, la pila (stack).
- Configuración de la fuente de reloj del microcontrolador
- Repertorio de instrucciones en XC8 PIC Assembler
- El contador de programa, el puntero de tabla, modos de direccionamiento.
- Esquematización de un algoritmo en diagrama de flujo.
- Estructuras condicionales y de repetición en XC8 PIC ASM

2

Preguntas previas:

- ¿Debo de tener los materiales para esta semana?
 - Así es, esta semana empezamos con la implementación de circuitos basados en el microcontrolador PIC18F45K50 en la sesión de laboratorio.
- Cuando creo el proyecto en el MPLAB X no me aparece el pic-as como opción para escoger.
 - Se tiene que instalar tanto el MPLAB X como el XC8 (últimas versiones) en la ruta por defecto para evitar que el primer programa no encuentre al segundo.
- ¿Las evaluaciones son grupales o individuales?
 - Revisa el sílabo en la sección evaluaciones, todas las evaluaciones son individuales a excepción de LB, DD (asignación aleatoria de miembros) y TF (asignación de miembros desde semana 9)
- El lenguaje Assembler para el microcontrolador lo debemos de saber previamente o lo vamos a ver en las clases?
 - Se lo va a atender durante las sesiones con el compromiso de que el alumno practique.

3

Recordatorio:

- La información en las siguientes vistas provienen de la hoja técnica del microcontrolador PIC18F45K50

4

Diagrama de pines del PIC18F45K50 (presentación 40-PIN PDIP)

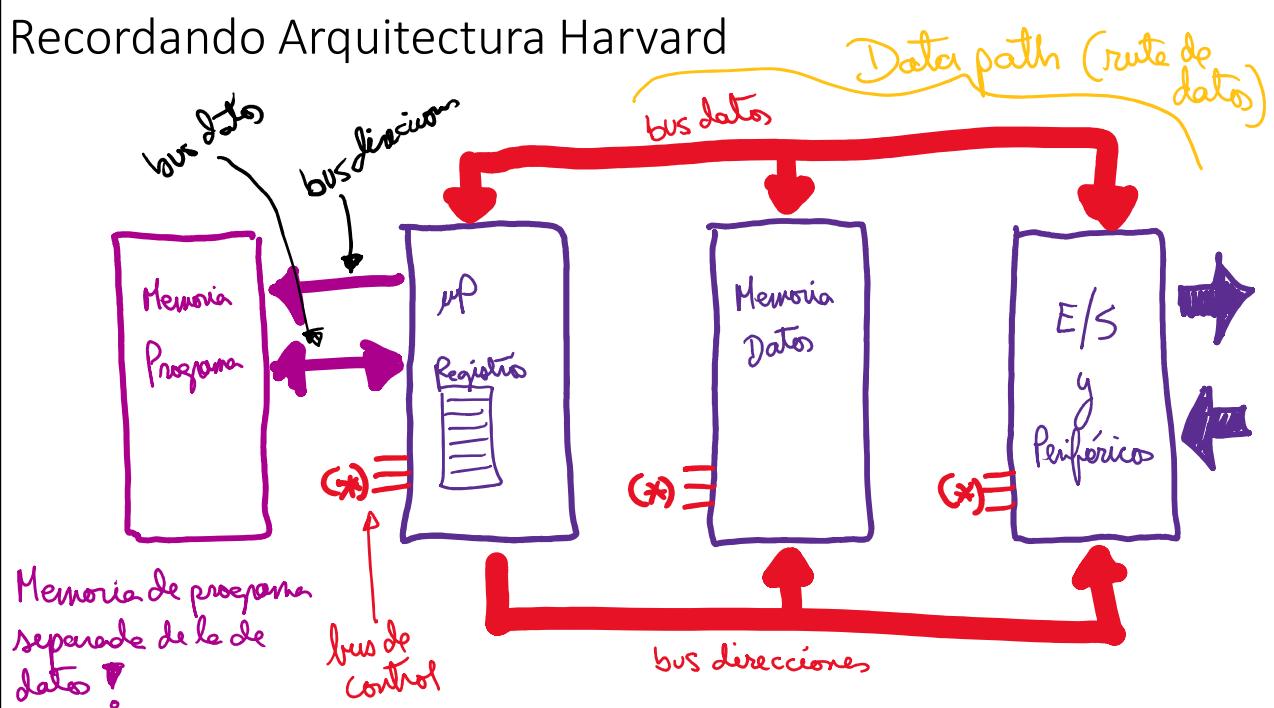
MCLR/VPP/RE3	1	RB7
RA0	2	RB6
RA1	3	RB5
RA2	4	RB4
RA3	5	RB3
RA4	6	RB2
RA5	7	RB1
RE0	8	RB0
RE1	9	VDD
RE2	10	Vss
VDD	11	RD7
Vss	12	RD6
RA7	13	RD5
RA6	14	RD4
RC0	15	RD3
RC1	16	RD2
RC2	17	D+
VUSB3V3	18	D-
RD0	19	RD1
RD1	20	RD0

PIC18(L)F45K50

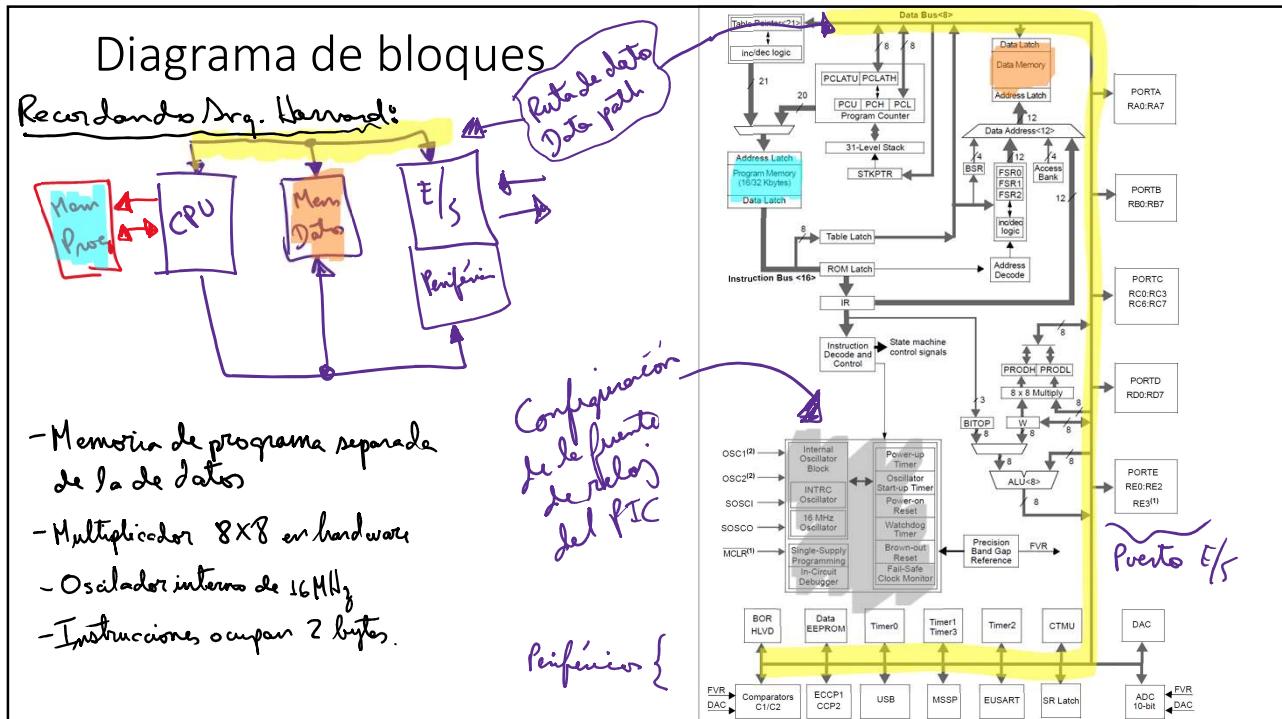
- Prestar especial atención a los pines de E/S en la hoja técnica (cap. 11)

5

Recordando Arquitectura Harvard



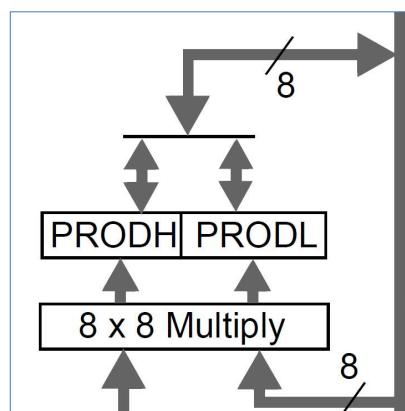
6



7

En el multiplicador 8x8:

¿Por qué el resultado se aloja en dos registros?



8

Operaciones aritméticas de multiplicación

* En decimal:

$$\begin{array}{r} 2 \times \\ 2 \\ \hline 4 \end{array}$$

$$\begin{array}{r} 99 \times \\ 99 \\ \hline 9801 \end{array}$$

cuatro dígitos

$$\begin{array}{r} 9 \times \\ 9 \\ \hline 81 \end{array}$$

dos dígitos

* En binario:

$$\begin{array}{r} 11 \times \\ 11 \\ \hline 1100 \end{array}$$

2 dígitos

$$\begin{array}{r} 1111 \times \\ 1111 \\ \hline ? \end{array}$$

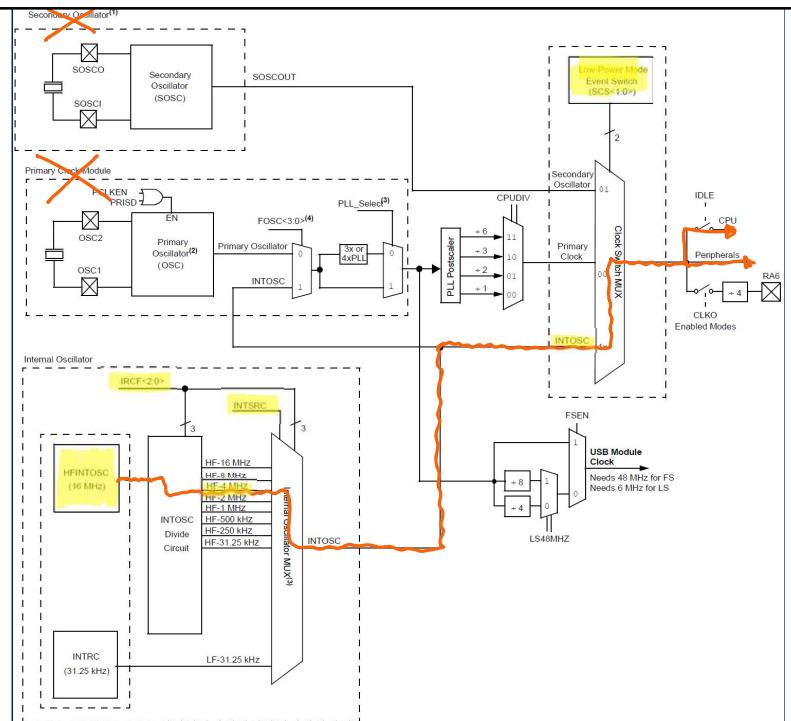
8 dígitos !

9

Configuración de la fuente de reloj del PIC18F45K50

Se empleará el HFINTOSC para la mayoría de ejemplos

Se presenta el caso para obtener 4MHz al CPU y los periféricos



10

Consideraciones con respecto a la fuente de reloj del PIC18F45K50

- La frecuencia máxima de trabajo es de 48MHz.
- Se puede emplear cristal con frecuencia máxima de 20MHz en el oscilador primario (los tradicionales de 2 pines o resonadores de 3 pines). Configuración HS
- A mayor frecuencia de trabajo mayor consumo de energía por parte del CPU.
- Hay una opción para que el HFINTOSC se conecte al PLL para así generar 48MHz al CPU y periféricos.
- El INTRC es la fuente de reloj de menor consumo energético.
- Siempre el uso de cristal externo garantiza una mejor precisión y estabilidad de frecuencia frente a un oscilador interno.

11

Formato numérico en XC8 PIC Assembler:

- Tenemos el número 126 en decimal	126
- En MPASM : Decimal $d'126'$	126D
.126	0111110B
Binario $b'0111110'$	7EH
Hexadecimal $0x7E$	



 0,1,2,3,4,5,6,7,8,9,AB,C,D,E,F

12

Formato numérico en XC8 PIC Assembler:

3.1

Constants and Radices

The default radix for constants in the PIC Assembler is different to that used by MPASM and the radix specifiers used by each assembler are different.

Without any radix specifier or directive, numeric constants in the PIC Assembler are interpreted as decimal values. Such values would be interpreted as hexadecimal in MPASM. Use the RADIX hex directive in migrated assembly code to ensure that the default radix assumed by the PIC Assembler matches that used by MPASM.

The MPASM radix specifiers and the equivalent specifiers used by the PIC Assembler are tabulated below.

Table 3-1. Equivalent Constants Radix Specifiers

MPASM Constant Forms	Radix	PIC Assembler Equivalent
B'binary_digits'	Binary	binary_digitsB
O'octal_digits'	Octal	octal_digits[0 o Q q]
D'decimal_digits' or .decimal_digits	Decimal	decimal_digits[D d nothing]
H'hexadecimal_digits' or 0xhexadecimal_digits	Hexadecimal	0hexadecimal_digits[H h] or 0xhexadecimal_digits
A'character' or 'character'	ASCII	'character'

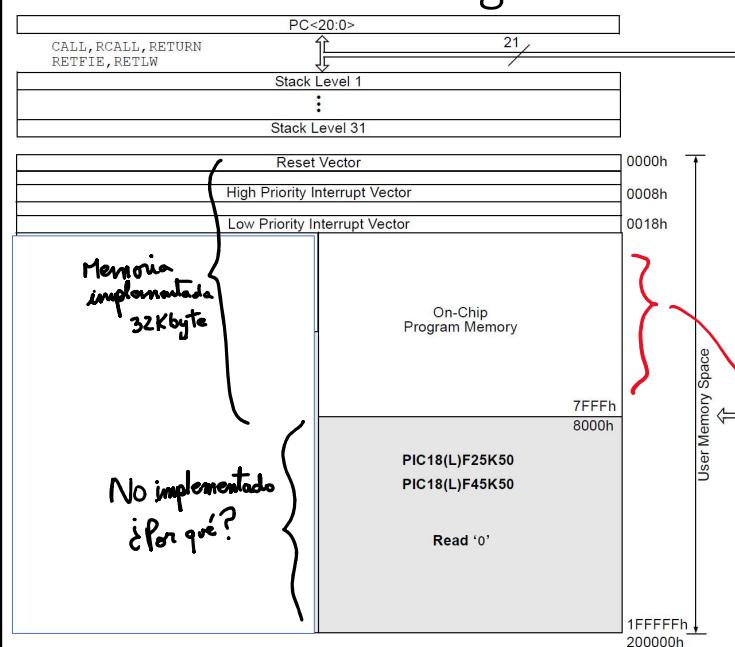
Note that the binary digits suffix (B) used by the PIC Assembler must be in upper case. Hexadecimal values must always begin with a zero digit, 0.

The following example shows the PIC Assembler's radix specifiers in use.

```
movlw 10110011B ;binary value
movlw 72q ;octal value
movlw 34 ;decimal value
movlw 04Fh ;hexadecimal value
movlw 'b' ;ASCII value
```

13

Memoria de Programa



¿Cuál es el tamaño total de la memoria de programa sabiendo que el rango de direcciones va desde 0x000000 hasta 0x1FFFFF?

$$2^{21} = 21 \text{ bits}$$

Programa de usuario

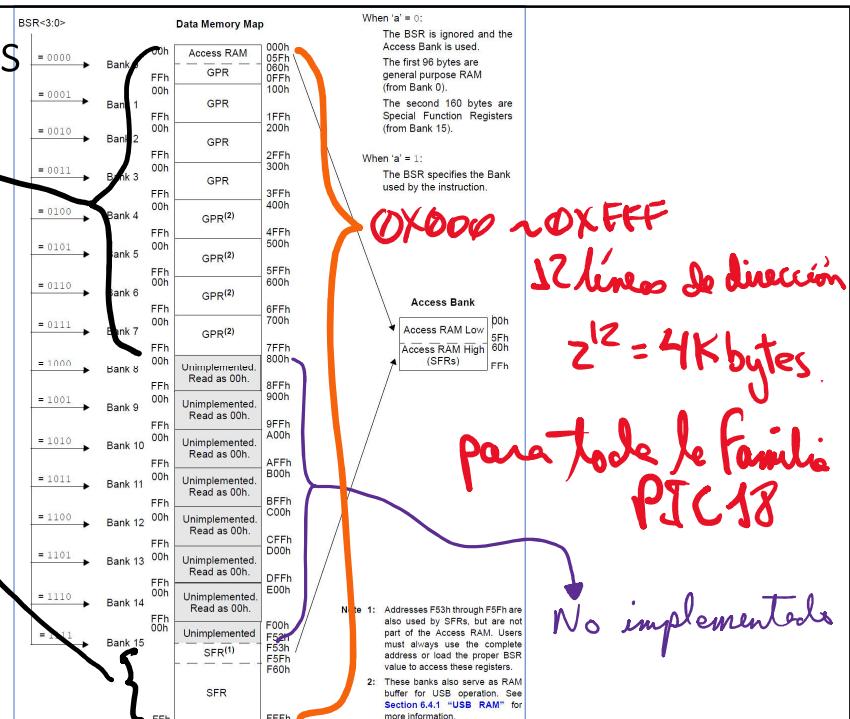
$$2^{21} = 2M \text{ byte}$$

14

Memoria de Datos

$\$X000 \sim \$X7FF$
 $\Rightarrow 2\text{Kb en RAM}$
 (GPR)

$\$XF53 \sim \FFF
 $\Rightarrow \text{Registros SFR}$



15

Detalle de los registros S.F.R.

#include<(P18F4550.h)>

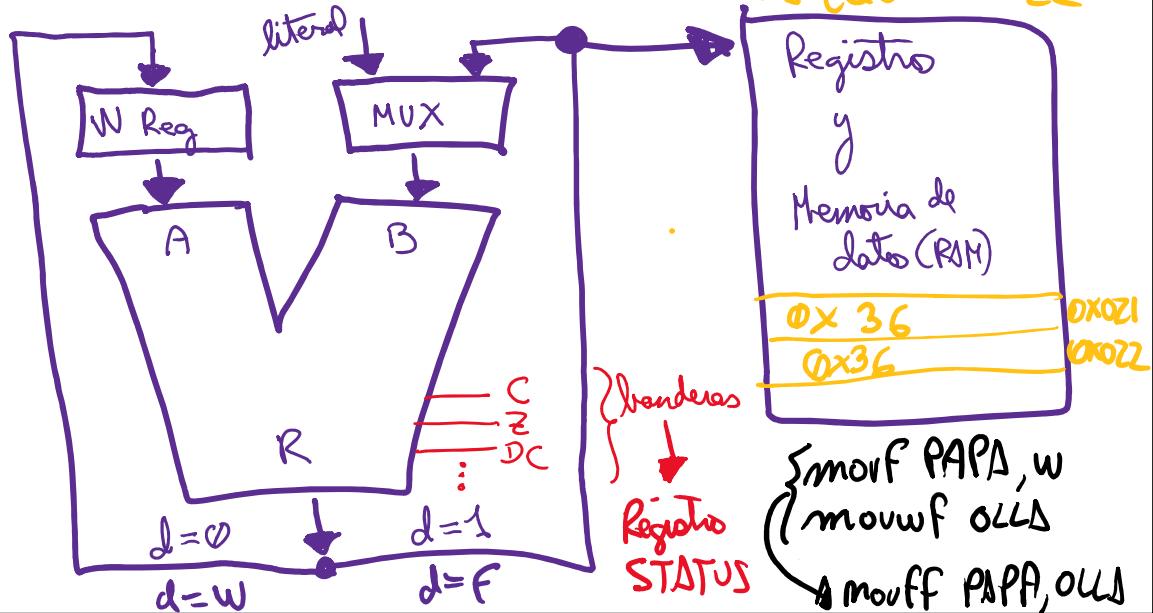
Address	Name	Address	Name	Address	Name	Address	Name
FFCh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F7Fh	UEP15
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Eh	UEP14
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR2H	F9Dh	UEP13
FFEh	PCLATH	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Ch	UEP12
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Bh	OSCTUNE
FFBh	PCL	FDDh	FSR2L	F99h	—(2)	F9Ah	UEP11
FFCh	TBLPTR ₀	FD7h	STATUS	F98h	BAUDCON	F9Ah	UEP10
FFEh	TBLPTRH	FD7h	TMR0H	F97h	—(2)	F78h	UEP8
FF6h	TBLPTRL	FD6h	TMR0L	F96h	ECCP1AS	F77h	UEP7
FF5h	TADLAT	FD5h	TOCON	F95h	TRISE ⁽³⁾	F76h	UEP6
FF4h	PRODH	FD4h	—(2)	F94h	CMCON	F75h	UEP5
FF3h	PRODL	FD3h	OSCCON	F93h	TMR3H	F74h	UEP4
FF2h	INTCON	FD2h	HLVDCON	F92h	TMR3L	F73h	UEP3
FF1h	INTCON2	FD1h	WDTCON	F91h	T3CON	F72h	UEP2
FF0h	INTCON3	F00h	RCON	F90h	—(2)	F70h	UEP0
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	F8Fh	—(2)	F6Fh	UCFG
FECh	POSTINC0 ⁽¹⁾	FECh	TMR1L	F8Eh	RCREG	F6Eh	UADDR
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	F8Dh	TXREG	F6Dh	UCON
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACh	TXSTA	F6Ch	USTAT
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA	F6Bh	UEIE
FEAh	FSR0H	FCAh	T2CON	FAAh	—(2)	F6Ah	UEIR
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F69h	UIE
FE8h	WREG	FC8h	SSPADD	F8Bh	EEDATA	F68h	UIR
FE7h	INDF1 ⁽¹⁾	FC7h	SSPSTAT	F87h	EECON2 ⁽¹⁾	F67h	UFRMH
FE6h	POSTINC1 ⁽¹⁾	F06h	SSPCON1	FA6h	EECON1	F66h	UFRML
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSPCON2	FA5h	—(2)	F65h	SPPCON ⁽³⁾
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	—(2)	F64h	SPPEPS ⁽³⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	—(2)	F63h	SPPCFG ⁽³⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F62h	SPPDATA ⁽³⁾
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F61h	—(2)
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F60h	—(2)

• Son 160 registros
 • Ejemplo: movwf LATD
 • wreg → \$XF8C
 • LATD
 • movwf \$XF8C
 • \$XF60

16

Flujo de datos en el CPU

PAPA EQU 0X021
OLLA EQU 0X022



17

Registro STATUS:

- Acá se encuentran las banderas del CPU que se actualizan cada vez que se ejecuta alguna operación numérica

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

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.

18

Manejo de los puertos de E/S en el PIC18F45K50

Necesitamos cuatro registros:

Establecer entradas y salidas
TRIS X → D_{1..8}, C_{1..D_{0..E}}

1	0	1	0	1	1	0	0
7	6	5	4	3	2	1	0

0 → salida
1 → entrada

Establecer analógicos o digitales
ANSEL X

0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0

0 → digital
1 → analógico

Lectura de datos en el puerto
PORTX

1	0	1	0	1	1	1	0
7	6	5	4	3	2	1	0

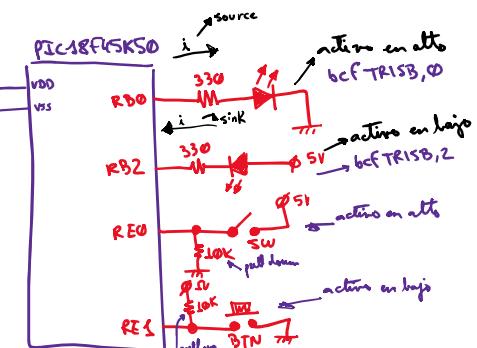
ΦAEN → D_{6H}

Escribir un dato en el puerto
LAT X

1	0	0	1	0	1	1	0
7	6	5	4	3	2	1	0

ΦW → 96H

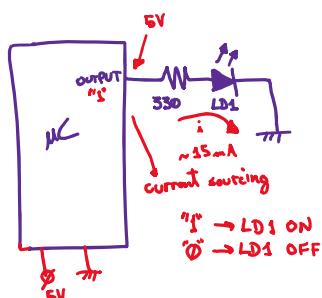
Conexión de LED's y botones:



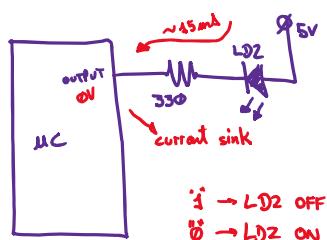
Nota: Los puertos son entradas digitales en un reset o al energizarse

19

¿Current source? ¿Current sink?



Activo en alto



Activo en bajo

20

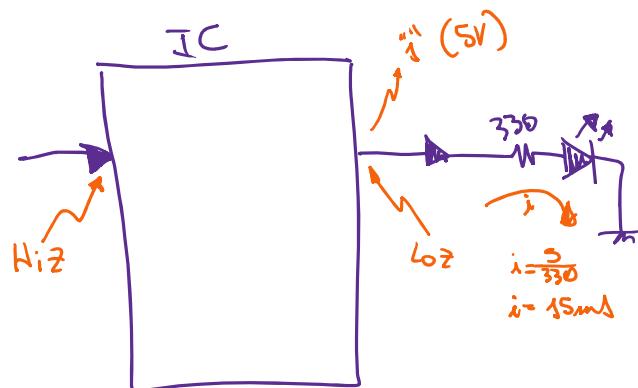
Consideraciones adicionales en el manejo de puertos de E/S:

- Los RB0:RB4 para que sean digitales se debe establecer en el bit de configuración PBADEN = OFF
- Para usar RA6 debes establecer el bit de configuración FOSC con el valor de INTOSCIO
- Para usar RE3 se debe deshabilitar la función de MCLR con el bit de configuración MCLRE = OFF

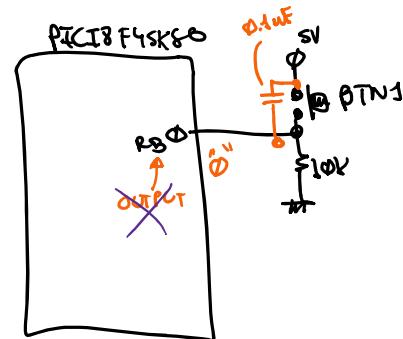
21

Casos:

Alta impedancia en entradas y baja impedancia en salidas:

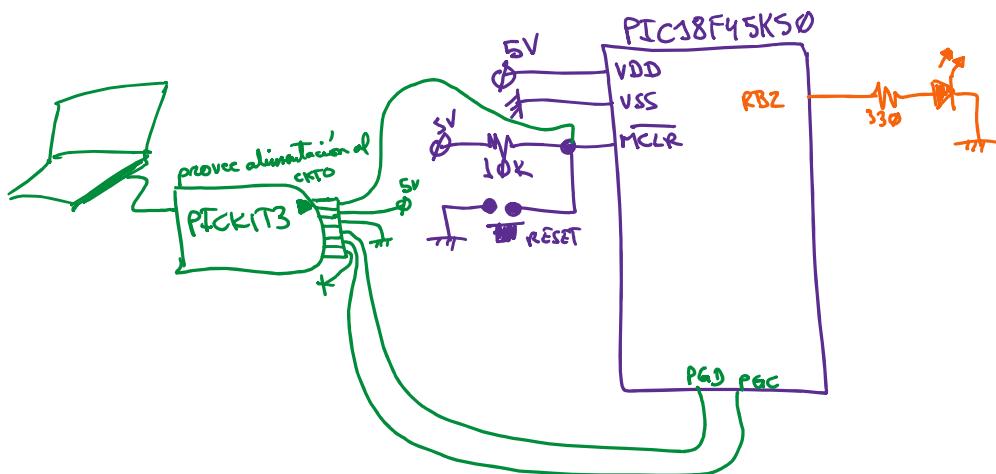


Filtro para rebote de botones empleando condensador:



22

Circuito mínimo de operación del PIC18F45K50



23

Cuestionario:

- Explicar el cómo funciona la estructura pipeline de los microcontroladores de la familia PIC18
- ¿Qué puertos son los que tienen la funcionalidad de pull-up interno? ¿Cómo se activan dichas pull-ups?
- Según la hoja técnica. ¿Cuál es el rango de voltaje de operación del PIC18F45K50? ¿Y del PIC18LF45K50?
- Explicar el funcionamiento del Watchdog en el PIC18F45K50
- ¿Qué procedimiento hay que hacer para que los pines del puerto B sean entradas digitales?
- ¿Cómo desactivo el MCLR para poder usar dicho pin como entrada digital?

24

Cuestionario:

- ¿Qué nombre tienen los pines necesarios para hacer la programación del microcontrolador PIC18F45K50 con el programador PICKIT3?
- Cuánta corriente puede como máximo entregar un pin del microcontrolador PIC18F45K50 configurado como salida?
- ¿Cómo configuramos el PIC18F45K50 para que funcione con un cristal externo de 4MHz? ¿Qué debemos de configurar en el microcontrolador para que funcione correctamente? ¿Cuánto demorará en ejecutarse una instrucción “nop”?
- ¿Cómo configuro el PIC18F45K50 para que funcione a 48MHz empleando el HFINTOSC y el PLL?