

Microcontroladores

Semestre: 2021-2

Profesor: Kalun José Lau Gan

Semana 2: El Microcontrolador PIC18F4550

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
- Conjunto de instrucciones.
- 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

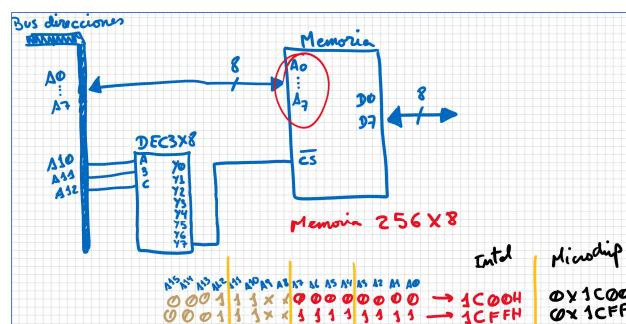
Preguntas previas:

- ¿Debo de tener los materiales para esta semana?
 - Así es, esta semana empezamos con la implementación de circuitos basados en el microcontrolador PIC18F4550 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 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 DD y TF
 - El lenguaje Assembler para el microcontrolador lo debemos de saber previamente o lo vamos a ver en las clases?
 - Se los va a atender durante las sesiones con el compromiso de que el alumno practique.

3

Preguntas previas:

- ¿Puede solucionarnos la EE?



٦

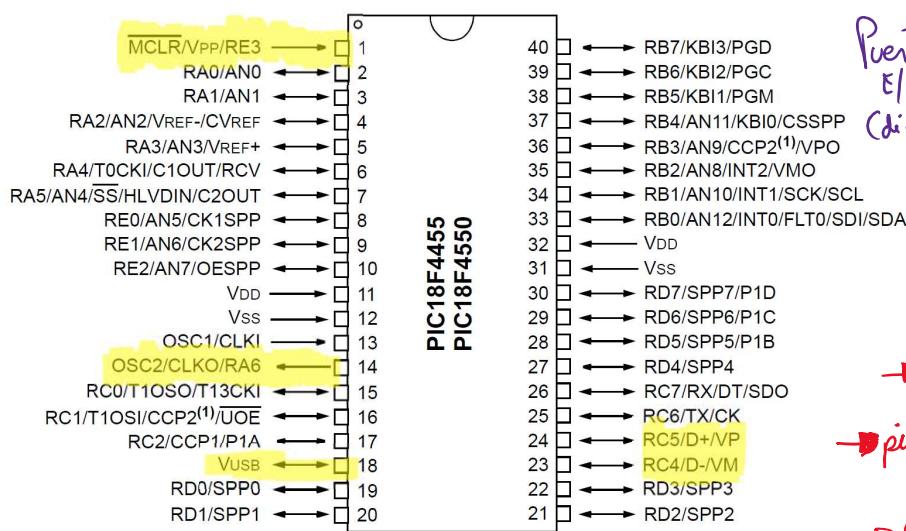
4

Recordatorio:

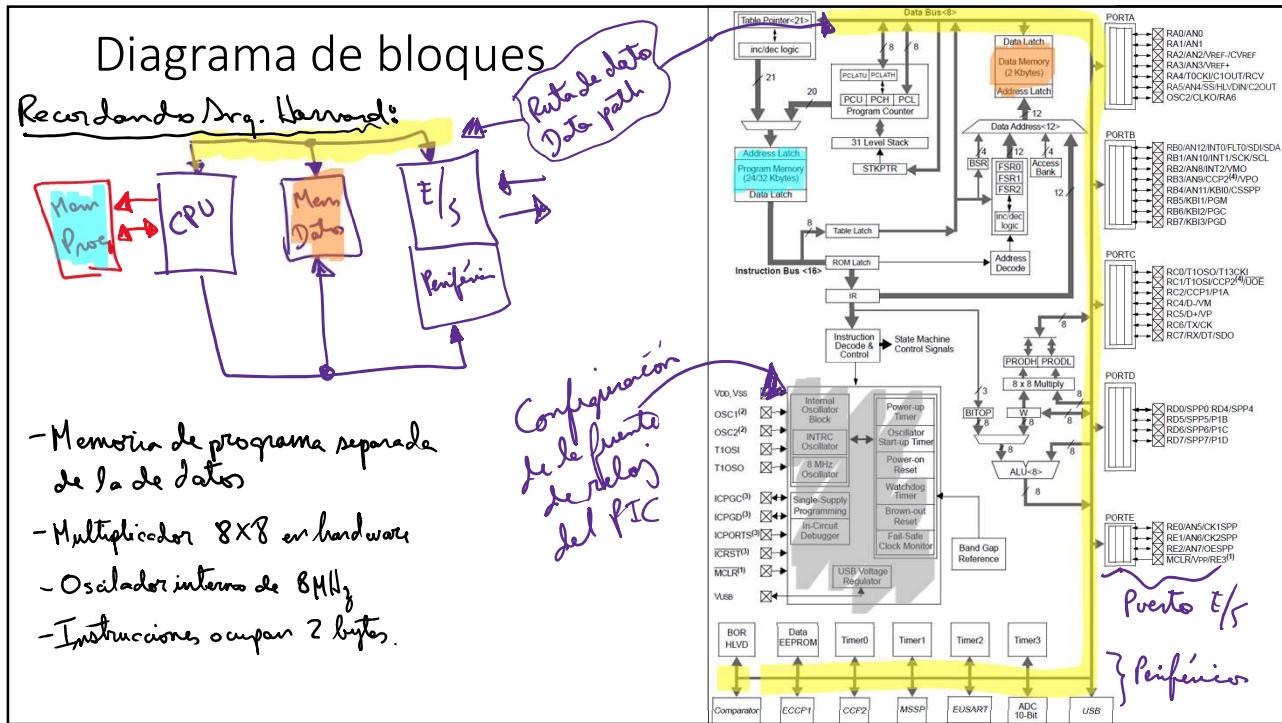
- La información en las siguientes vistas provienen de la hoja técnica del microcontrolador PIC18F4550

5

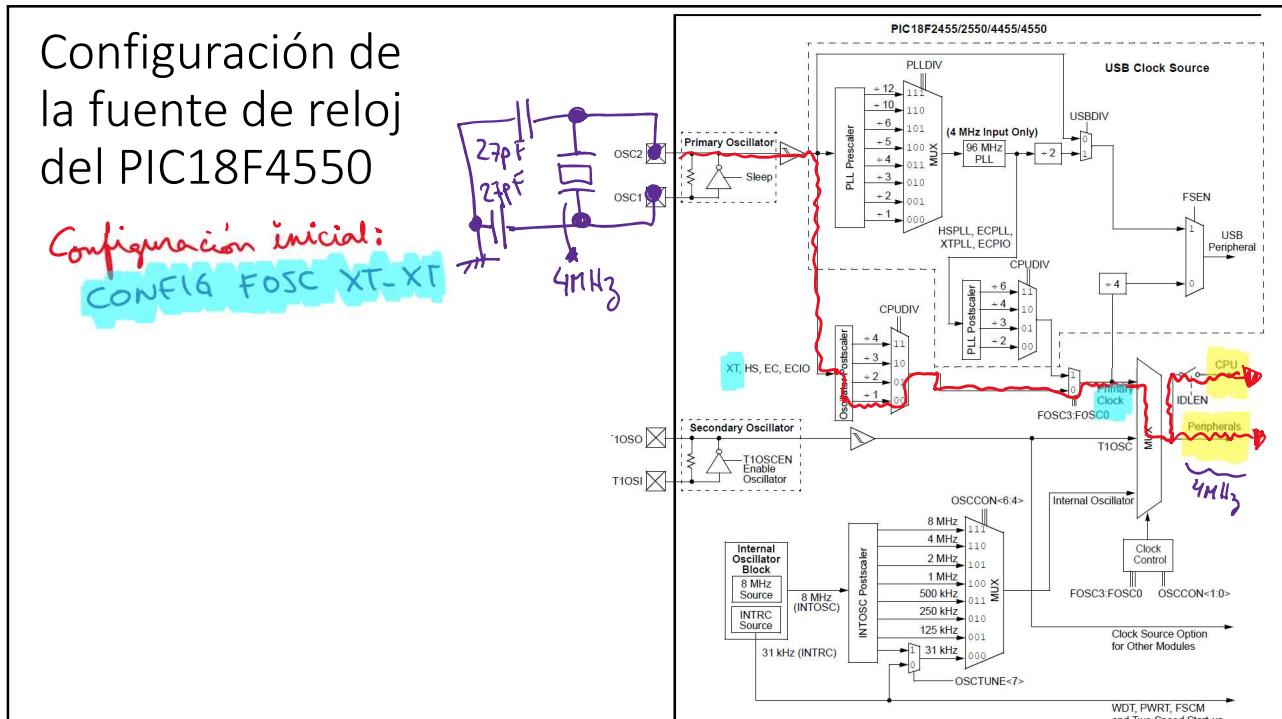
Diagrama de pines del PIC18F4550



6



7



8

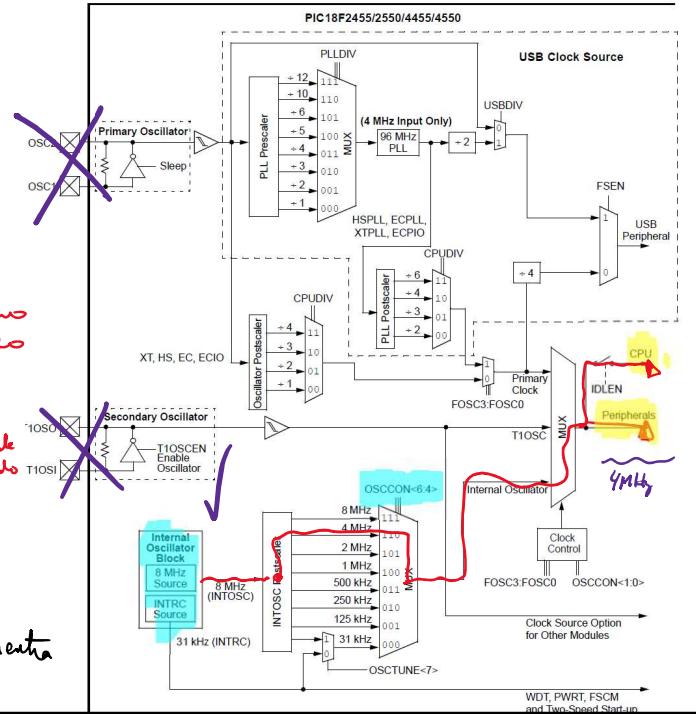
Configuración de la fuente de reloj del PIC18F4550

Configuración inicial:
CONFIG FOSC INTOSC IO_EC
 oscilador interno y RA6 habilitado

En el código:

CONFIG: bcf OSCCON, 6 } oscilador interno de
 bcf OSCCON, 5 } 8MHz funcionando
 bcf OSCCON, 4 } a 4MHz

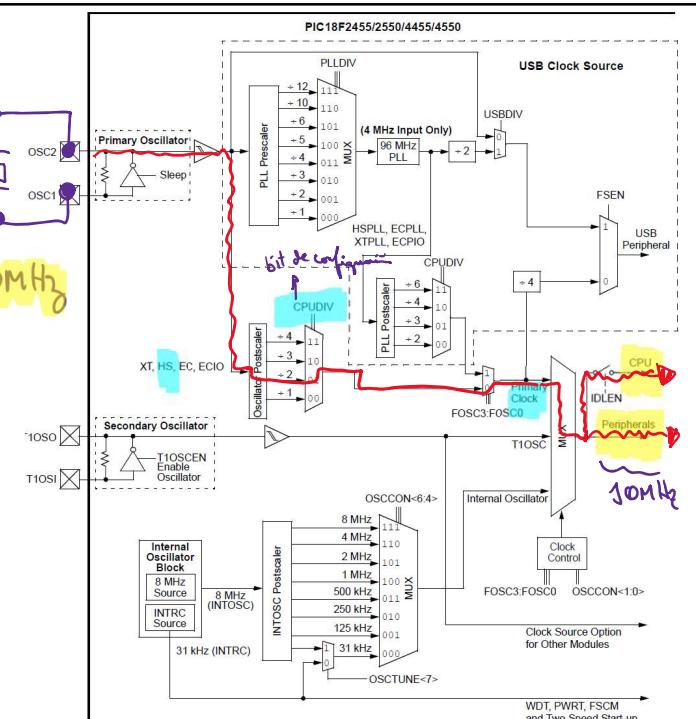
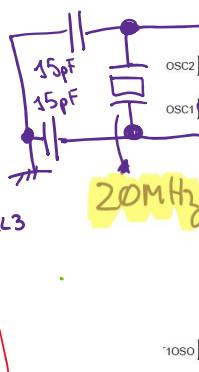
Nota:
 Por defecto el MUX del INTOSC se encuentra en 1MHz



9

Configuración de la fuente de reloj del PIC18F4550

Configuración inicial:
CONFIG FOSC HS
CONFIG CPUDIV OSC2 PLL3



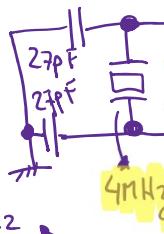
10

Configuración de la fuente de reloj del PIC18F4550

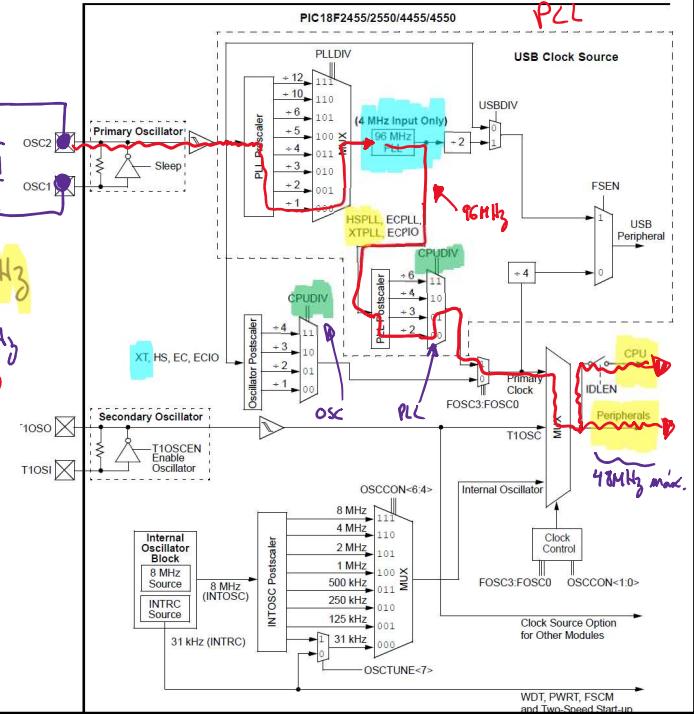
Configuración inicial:

```
CONFIG FOSC XTPLL-XT
CONFIG CPUDIV OSC1-PLL2
OSC2-PLL3
OSC3-PLL4
OSC4-PLL5
✓ SimPLL
✓ ConPLL
```

Note:
PLL funciona solo con 4MHz de entrada



4MHz



11

Consideraciones con respecto a la fuente de reloj del PIC18F4550

- Se puede emplear cristal con frecuencia máxima de 20MHz (los tradicionales de 2 pines). Configuración HS
- Se emplea cristal con señal cuadrada de salida de hasta 48MHz (los de cuatro pines). Configuración EC
- A mayor frecuencia de trabajo mayor consumo de energía por parte del CPU.
- Oscilador interno es menos preciso frente a un cristal externo. No se recomienda el uso del INTOSC para aplicaciones de RTC.

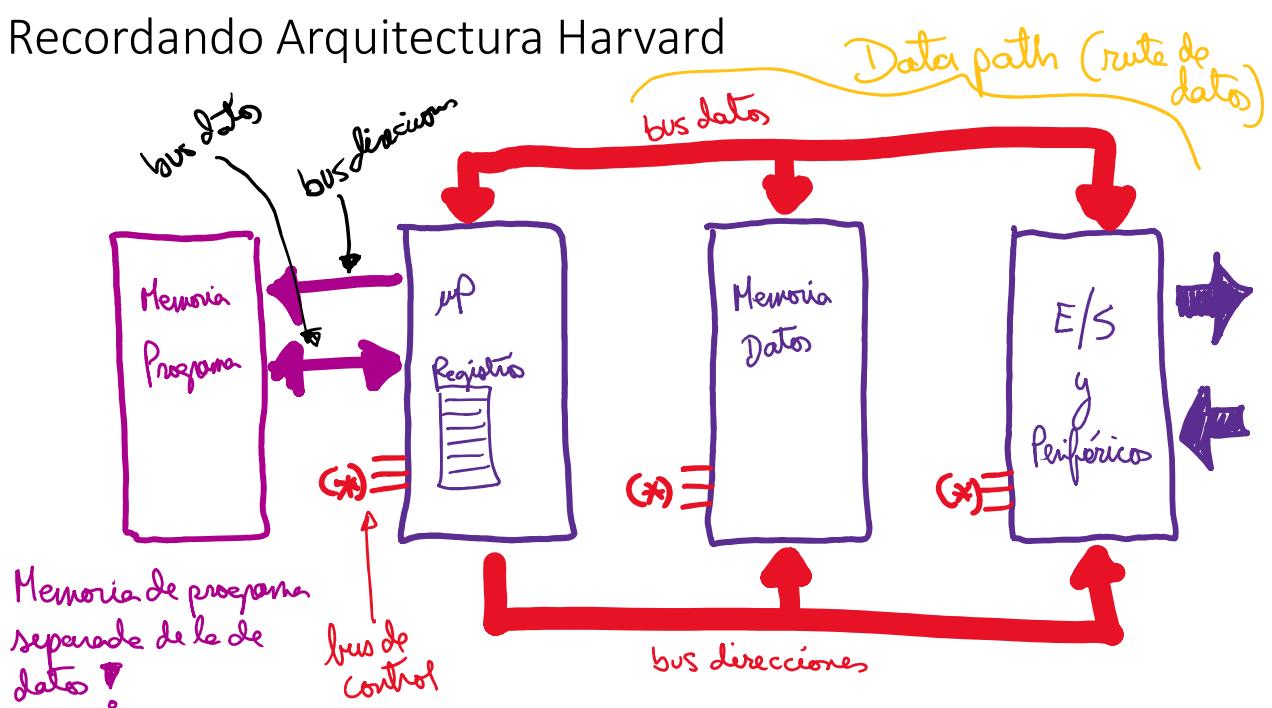
12

Formato numérico en XC8 PIC Assembler:

- Tenemos el número 126 en decimal
 - En MPASM : Decimal $d'126'$
126
 - Binario $b'01111110'$
 - Hexadecimal $0x7E$
- | | |
|-----------|------|
| 126 | 126D |
| 01111110B | |
| 7EH | |

13

Recordando Arquitectura Harvard

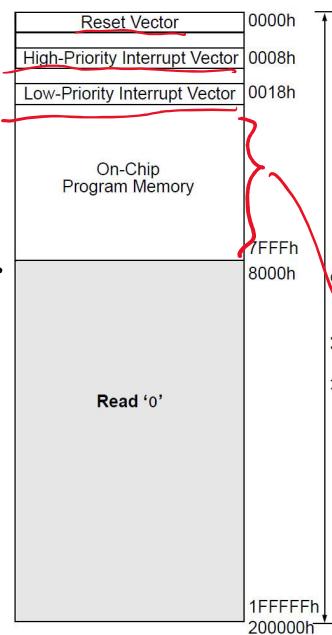


14

Memoria de Programa

Memoria implementada en el PIC18F4550
32 KByte

No implementada en el PIC18F4550.
¿Por qué?



¿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

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

15

Memoria de Datos

En el PIC18F4550:

0x000 ~ 0x7FF
 \Rightarrow 2 kb en RAM (GPR)

0xF60 ~ 0xFFFF
 \Rightarrow 160 para SFR

Data Memory Map

00h	Access RAM	00h
FFh	GPR	05h
00h	GPR	06h
FFh	GPR	0Fh
00h	GPR	100h
FFh	GPR(1)	1FFh
00h	GPR(1)	200h
FFh	GPR(1)	2FFh
00h	GPR(1)	300h
FFh	GPR(1)	3FFh
00h	GPR(1)	400h
FFh	GPR(1)	4FFh
00h	GPR(1)	500h
FFh	GPR(1)	5FFh
00h	GPR(1)	600h
FFh	GPR(1)	6FFh
00h	GPR(1)	700h
FFh	GPR(1)	7FFh
00h	Unused Read as 00h	800h
FFh	Unused	EFFh
00h	SFR	F00h
FFh		F5Fh
00h		F60h
FFh		FFFh

0x000 ~ 0xFFFF

12 líneas de dirección

$$2^{12} = 4K \text{ bytes}$$

para todo la familia PIC18

} No implementada

16

Detalle de los registros S.F.R.

#include<P19F4520.h>

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFn	TOSU	FDFn	INDF2 ⁽¹⁾	FBFn	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEn	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCPICON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR2H	F9Ch	— ⁽²⁾	F7Ch	UEP12
FFFh	PCLATH	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCP2CON	F9Bh	OSCTUNE	F7Bh	UEP11
FFAh	PCLATU	FDAh	FSR2H	FBAh	— ⁽²⁾	F9Ah	— ⁽²⁾	F7Ah	UEP10
FFBh	PCL	FDFh	FSR2L	F9Bh	— ⁽²⁾	F9Bh	— ⁽²⁾	F7Bh	UEP9
FF8h	TBLPTR ⁽¹⁾	FDBh	STATUS	F80h	ADCON1	F98h	— ⁽⁴⁾	F78h	UEP8
FF7h	TBLPTRH	FDBh	ECCP1DEL	F97h	— ⁽²⁾	F97h	UEP7	F77h	UEP6
FF6h	TBLPTRL	FDBh	ECCP1IAS	F96h	TRISE ⁽³⁾	F96h	TRISD ⁽³⁾	F76h	UEP5
FF5h	TABLAT	FDBh	CVRCON	F94h	TRISC	F93h	TRISB	F73h	UEP4
FF4h	PRODH	FDBh	CMCON	F92h	TRISA	F92h	UEP3	F72h	UEP2
FF3h	PRODL	FDBh	TMR3H	F91h	— ⁽²⁾	F91h	— ⁽²⁾	F71h	UEP1
FF2h	INTCON	FDBh	TMR3L	F80h	SPBRGH	F90h	— ⁽²⁾	F70h	UEP0
FF1h	INTCON2	FDBh	TMR0H	F8Fh	SPBRG	F8Fh	— ⁽²⁾	F6Fh	UCFG
FF0h	INTCON3	FDBh	TMR0L	F8Eh	RCREG	F8Eh	— ⁽²⁾	F6Eh	UADDR
FEFh	INDF0 ⁽¹⁾	FCEh	TMR1H	FADh	TXREG	F80h	LATE ⁽³⁾	F6Dh	UCON
FE Eh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FACH	TXSTA	F8Ch	LATD ⁽³⁾	F6Ch	USTAT
FE Dh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FE Ch	PREINC0 ⁽¹⁾	FCHh	TMR2	FAAh	T2CON	F8Ah	LATB	F6Ah	UEIR
FE Bh	PLUSW0 ⁽¹⁾	FCBh	PR2	FCAh	SSPBUF	F89h	EEADR	F69h	UIE
FE Ah	FSR0H	FCBh	PR0	FCBh	SSPADD	F88h	EEDATA	F68h	UIR
FE 9h	FSR0L	FCBh	PR0L	FCBh	SSPSTAT	F87h	EECON2 ⁽¹⁾	F67h	UFRMH
FE 8h	WREG	FCBh	PR1	FCBh	SSPCON1	F86h	EECON1	F66h	UFRML
FE 7h	INDF1 ⁽¹⁾	FC7h	ADRESH	FA5h	SSPCON2	F85h	— ⁽²⁾	F65h	SPPCON ⁽³⁾
FE 6h	POSTINC1 ⁽¹⁾	FC7h	ADRESL	FA4h	— ⁽²⁾	F84h	PORTE	F64h	SPPEPS ⁽³⁾
FE 5h	POSTDEC1 ⁽¹⁾	FC7h	ADRESL	FA3h	— ⁽²⁾	F83h	PORTD ⁽³⁾	F63h	SPPCFG ⁽³⁾
FE 4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA2h	ADCON0	F82h	PORTC	F62h	SPPDATA ⁽³⁾
FE 3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA1h	ADCON1	F81h	PORTB	F61h	— ⁽²⁾
FE 2h	FSR1H	FC2h	ADCON0	FA0h	ADCON2	F80h	PORTA	F60h	— ⁽²⁾
FE 1h	FSR1L	FC1h	ADCON1						
FE 0h	BSR	FO0h	ADCON2						

QFFF → **PC** → **STATUS**

Son 160 registros

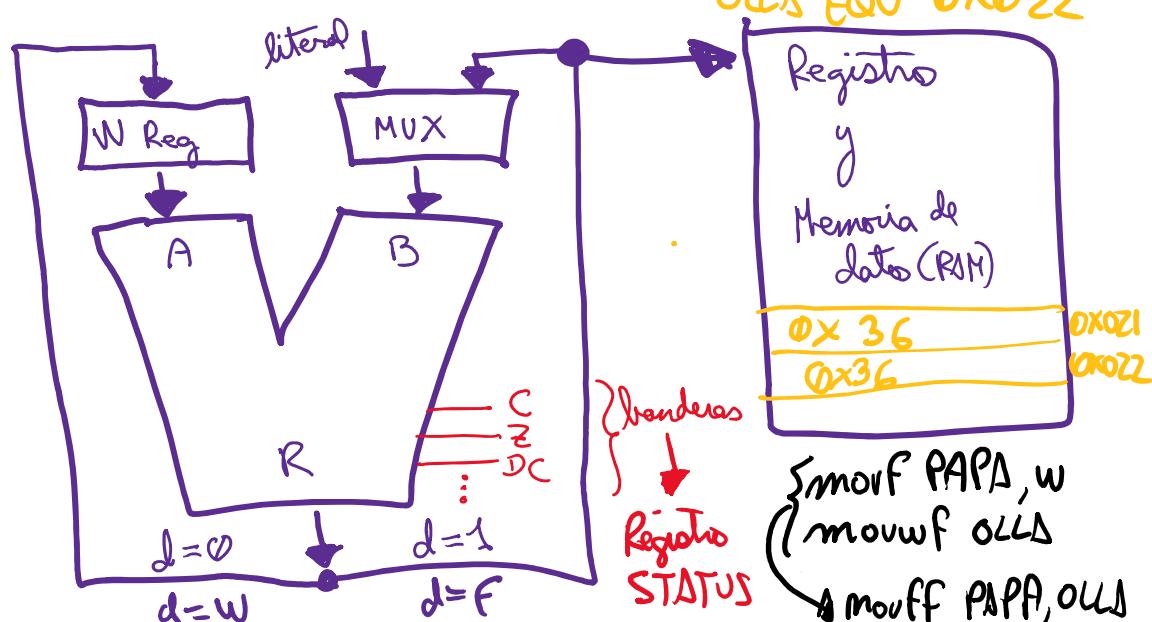
Ejemplo:
movwf LATD

[wreg] → **0x F8C** → LATD
movwf 0xF8C

0XF60

17

Flujo de datos en el CPU



18

Registro STATUS:

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

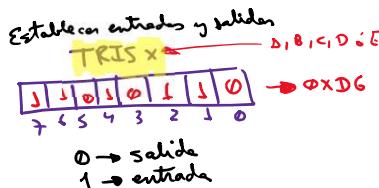
- Note 1:** For Borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRE, RLF) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.
- 2:** For Borrow, the polarity is reversed. A subtraction is executed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

19

Manejo de los puertos de E/S en el PIC18F4550

Puerto A: RA0 – RA6 -7
 Puerto B: RB0 – RB7 -8
 Puerto C: RC0, RC1, RC2, RC4, RC5, RC6, RC7 -7
 Puerto D: RD0 – RD7 -8
 Puerto E: RE0 – RE3 -4

Necesitamos tres registros:



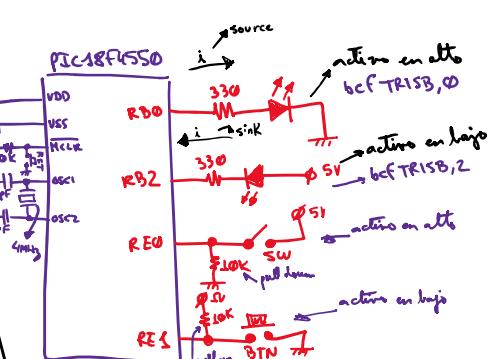
Lectura de datos en el puerto
 PORTX



Escribir un dato en el puerto
 LAT X

Nota: Los puertos son entrada en un reset o al energizarse

Conexión de LED's y botones:



R

20

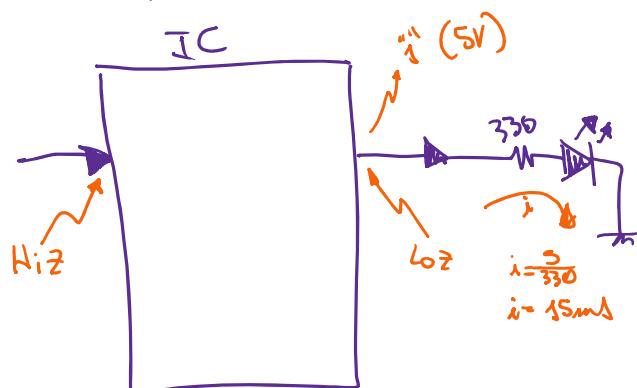
Consideraciones adicionales en el manejo de puertos de E/S:

- Los RBO:RB4 para que sean digitales se debe establecer en el bit de configuración PBADEN = OFF
- Para que RE sean digitales hay que escribir en el registro ADCON1 el valor de 0x0F:
 $\text{movlw } 0x0F$
 movwf ADCON1
- Para usar RA6 debes de establecer el bit de configuración FOSC con el valor de INTOSCIO_EC
- Para usar RE3 se debe de deshabilitar la función de MCLR con el bit de configuración MCLRE = OFF
- Recordar que RE3 es solo entrada y RA6 es solo salida
- Recordar que RC4 y RC5 son solo entradas
- RC3 no hay en este modelo de microcontrolador

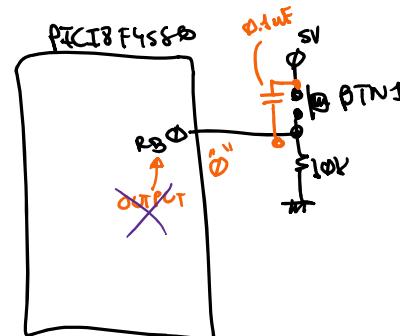
21

Casos:

Alta impedancia en entradas y baja
impedancia en salidas:

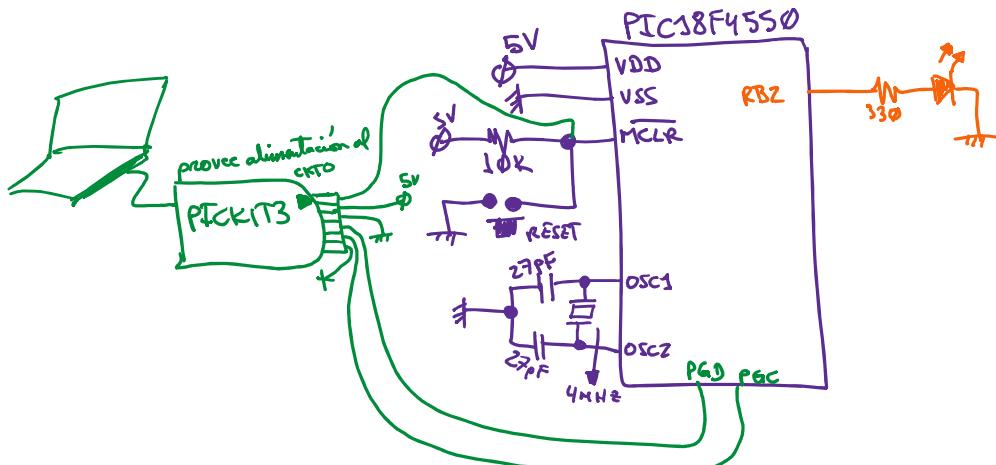


Filtro para rebote de botones
empleando condensador:



22

Circuito mínimo de operación del PIC18F4550



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 PIC18F4550? ¿Y del PIC18LF4550?
- Explicar el funcionamiento del Watchdog en el PIC18F4550
- ¿Qué procedimiento hay que hacer para que los pines del puerto B sean 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 PIC18F4550 con el programador PICKIT3?
- Cuánta corriente puede como máximo entregar un pin del microcontrolador PIC18F4550 configurado como salida?
- Si cambiamos el cristal de 4MHz por uno de 20MHz. ¿Qué debemos de configurar en el microcontrolador para que funcione correctamente? ¿Cuánto demorará en ejecutarse una instrucción “nop”?