

Microcontroladores

Semestre: 2022-1

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

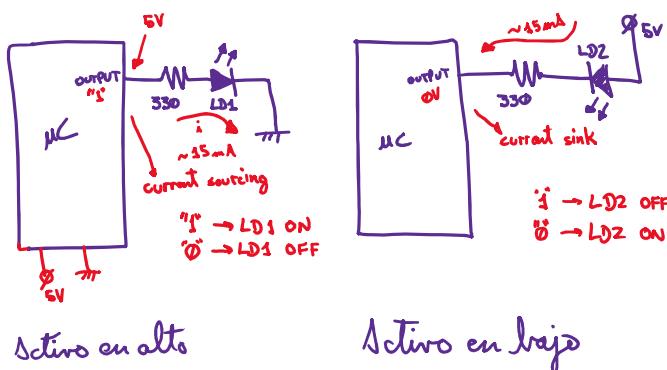
2

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 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 los va a atender durante las sesiones con el compromiso de que el alumno practique.

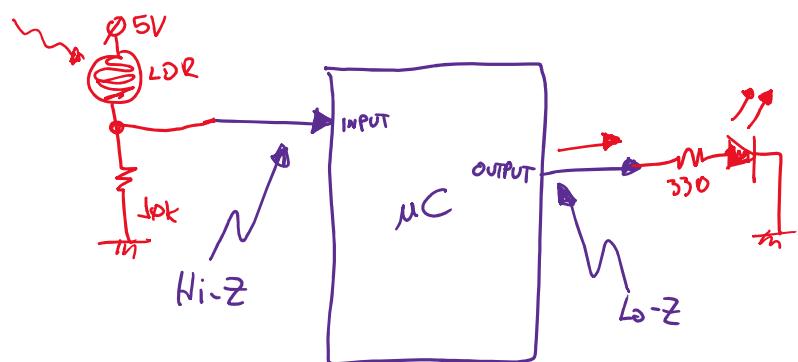
3

¿Current source? ¿Current sink?



4

Input: Hi-z / Output: Lo-z



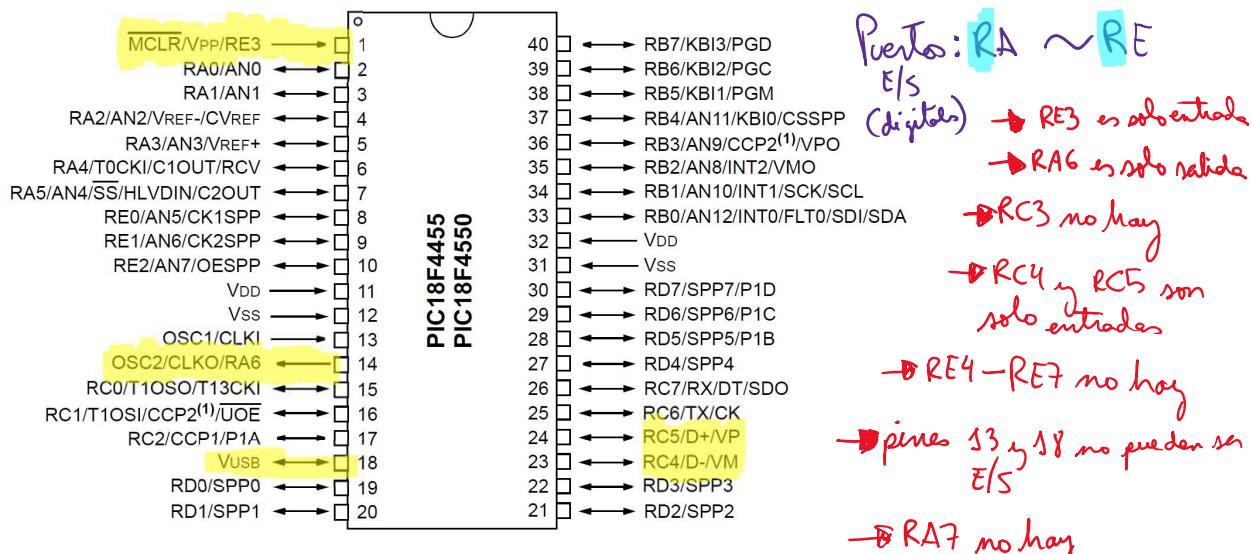
5

Recordatorio:

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

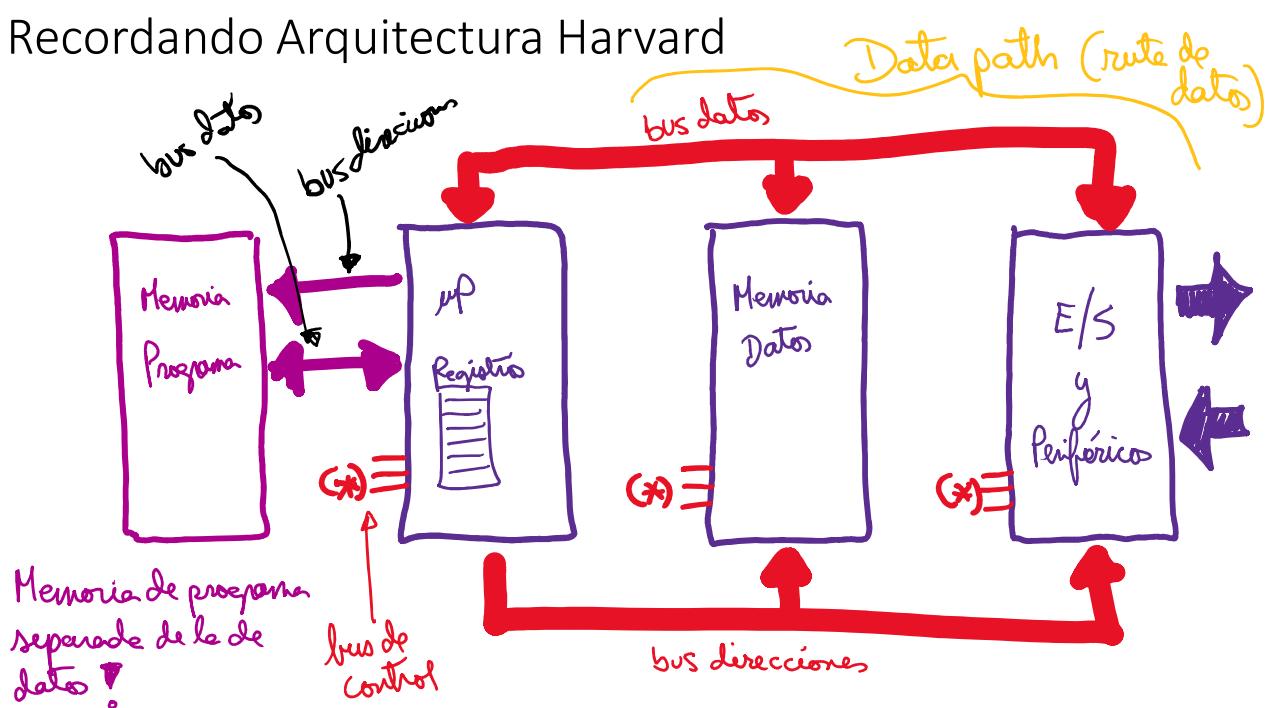
6

Diagrama de pines del PIC18F4550

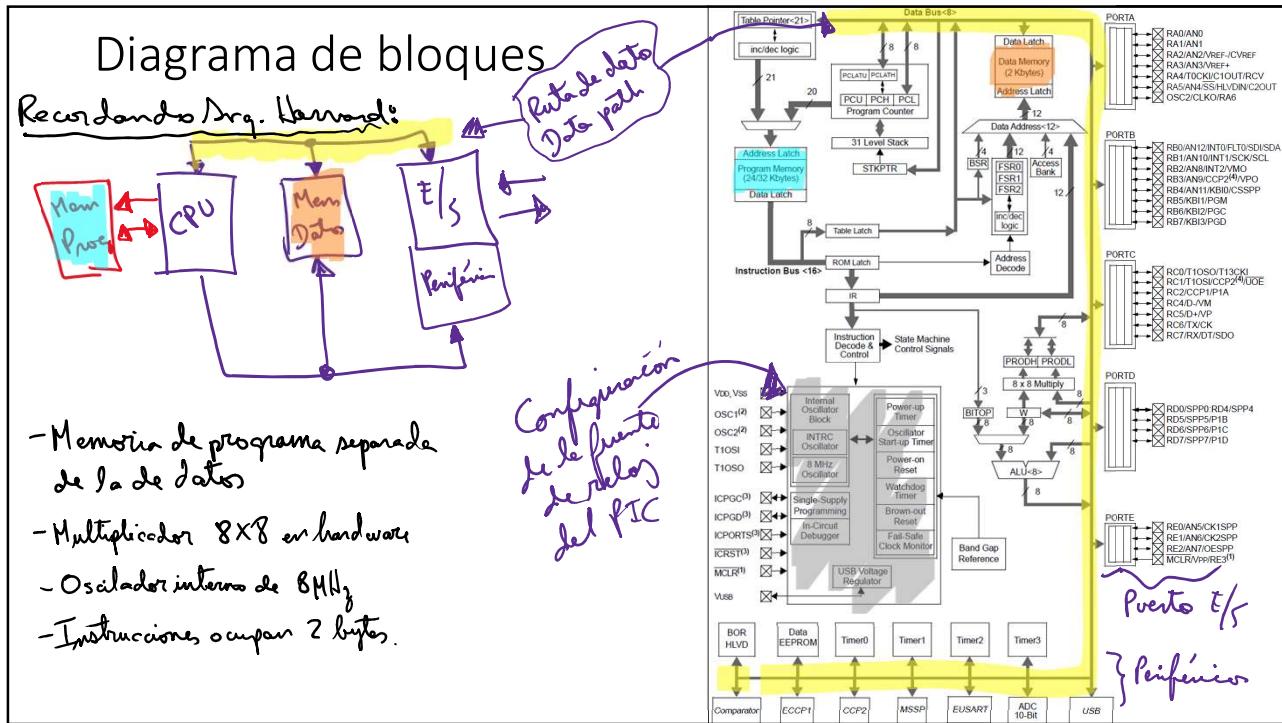


7

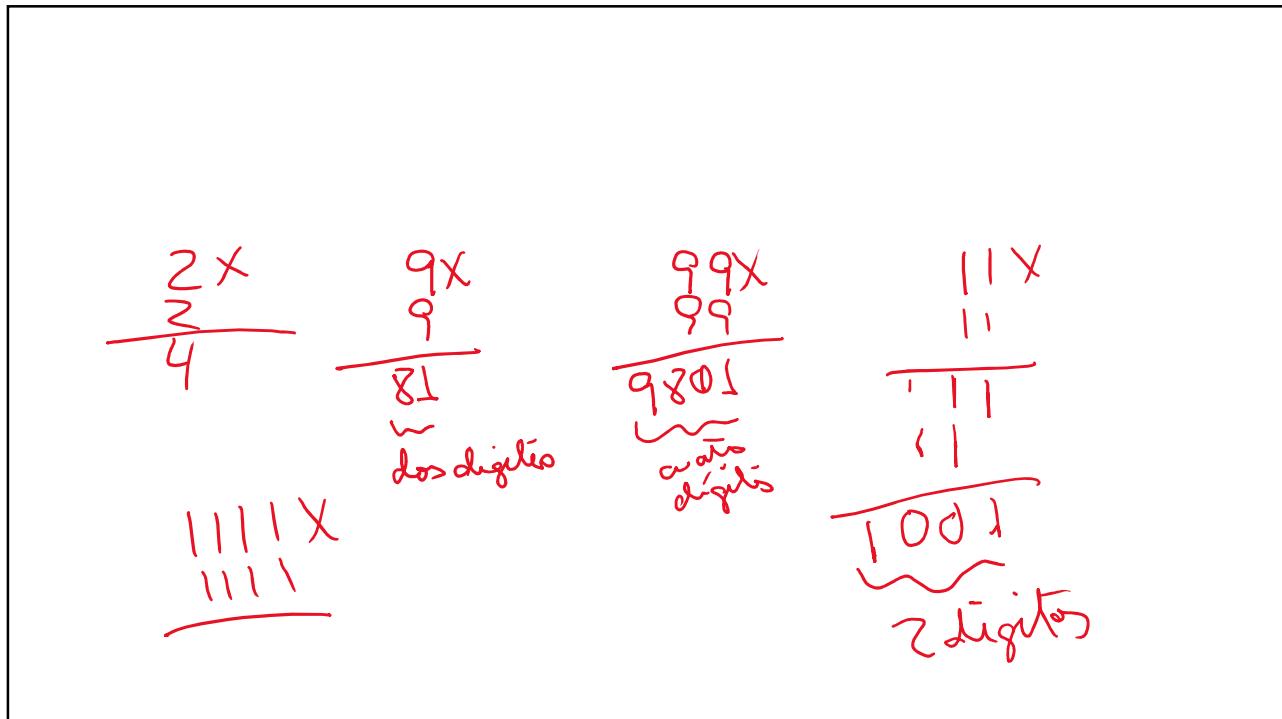
Recordando Arquitectura Harvard



8



9

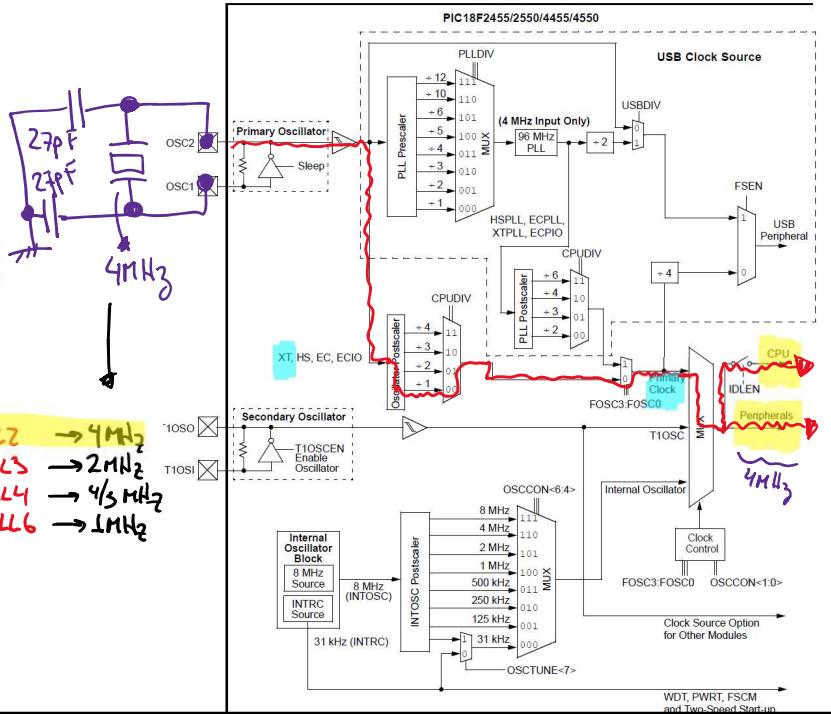


10

Configuración de la fuente de reloj del PIC18F4550

*Configuración inicial:
CONFIG FOSC XT-XI*

CONFIG CPUDIV
 OSC1 -PLL2 → 4MHz
 OSC2 -PLL3 → 2MHz
 OSC3 -PLL4 → 4/3MHz
 OSC4 -PLL6 → 1MHz



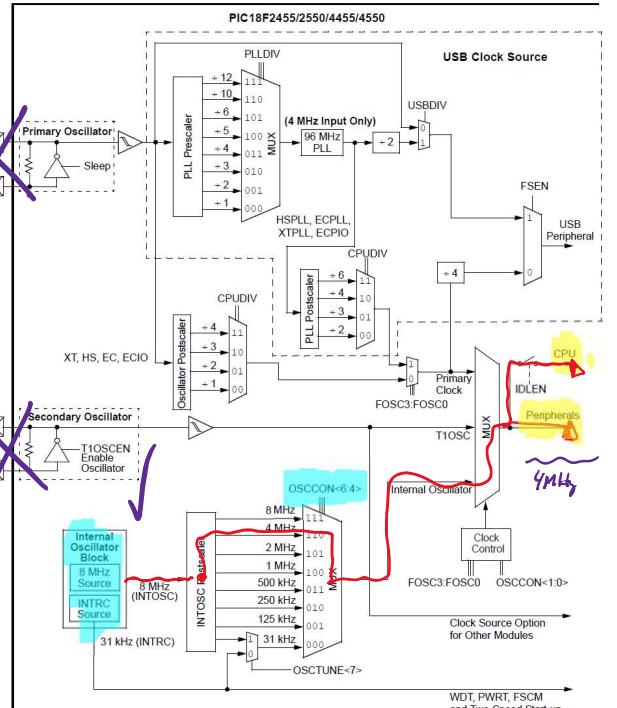
11

Configuración de la fuente de reloj del PIC18F4550

*Configuración inicial:
CONFIG FOSC INTOSC IO-EC
oscilador interno y RS6 habilitado*

*En el código:
CONFIGURE: bcf OSCCON, 6 } oscilador interno de
 btf OSCCON, 5 } 8MHz, funcionando
 btf OSCCON, 4 } a 4MHz*

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

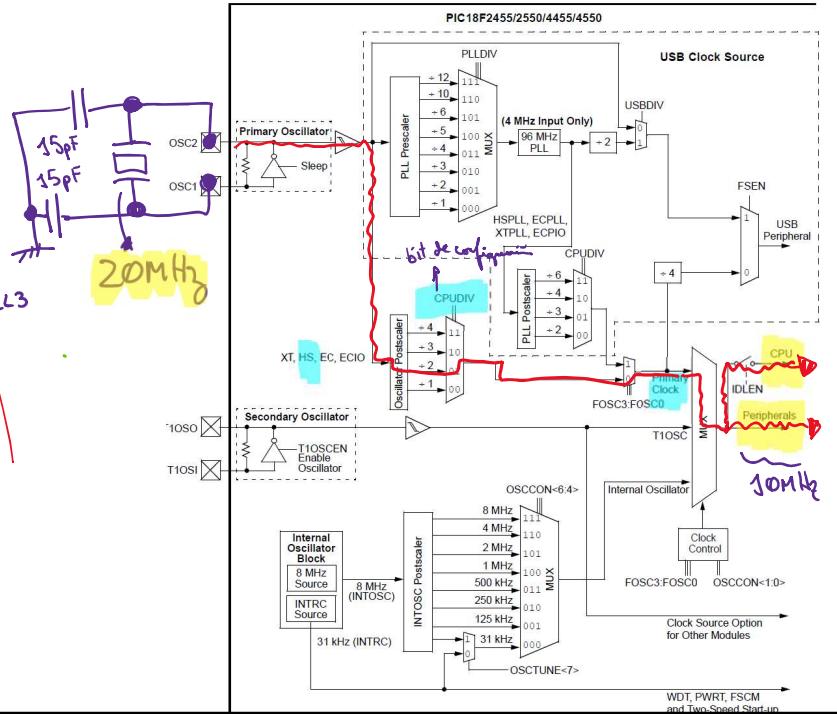


12

Configuración de la fuente de reloj del PIC18F4550

Configuración inicial:

CONFIG FOSC HS
CONFIG CPUDIV OSC2_PLL3



13

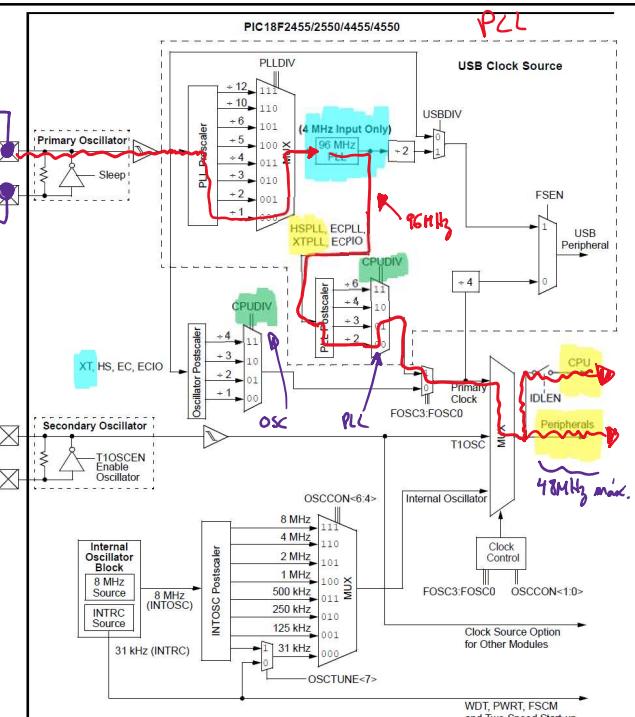
Configuración de la fuente de reloj del PIC18F4550

Configuración inicial:

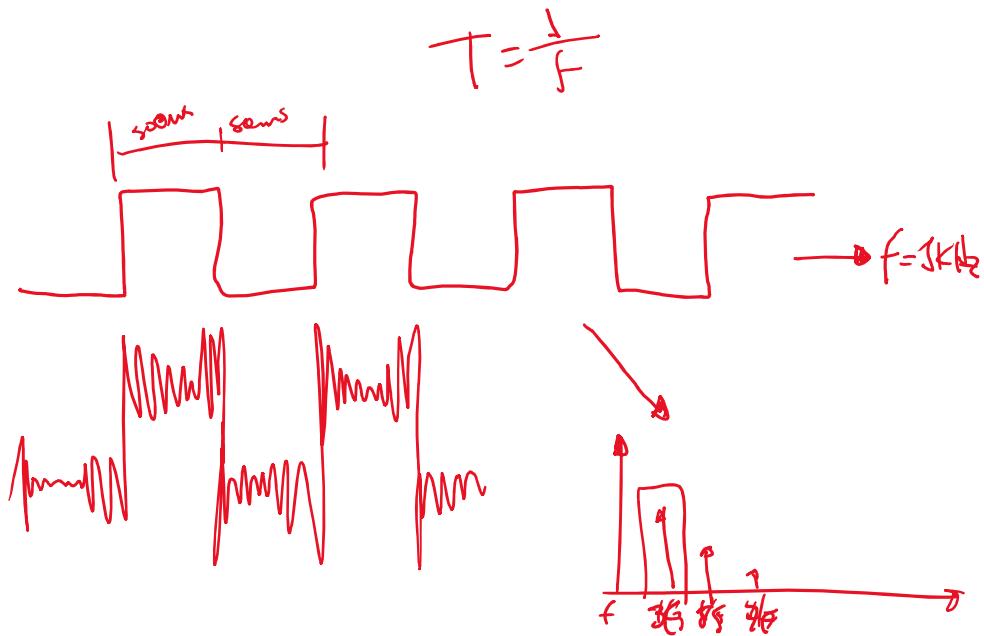
CONFIG FOSC XTPLL-XT
CONFIG CPUDIV OSC1_PLL2

OSC2_PLL3
OSC3_PLL4
OSC4_PLL6
✓ simPLL
✓ conPLL

Nota:
PLL funciona solo con 4MHz de entrada



14



15

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, comunicaciones.

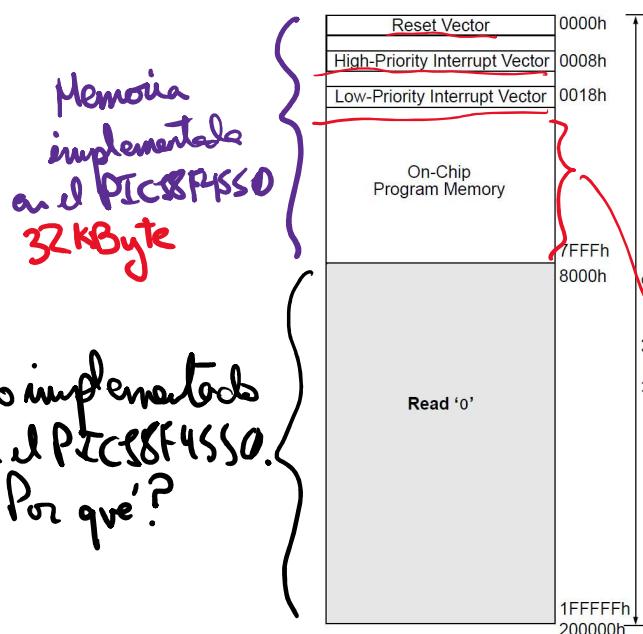
16

Formato numérico en XC8 PIC Assembler:

- Tenemos el número 126 en decimal	126
- En MPASM : Decimal	d'126'
	.126
Binario	b'01111110'
Hexadecimal	0x7E
	7EH
	→ 0FFH
	0A5H
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	

17

Memoria de Programa



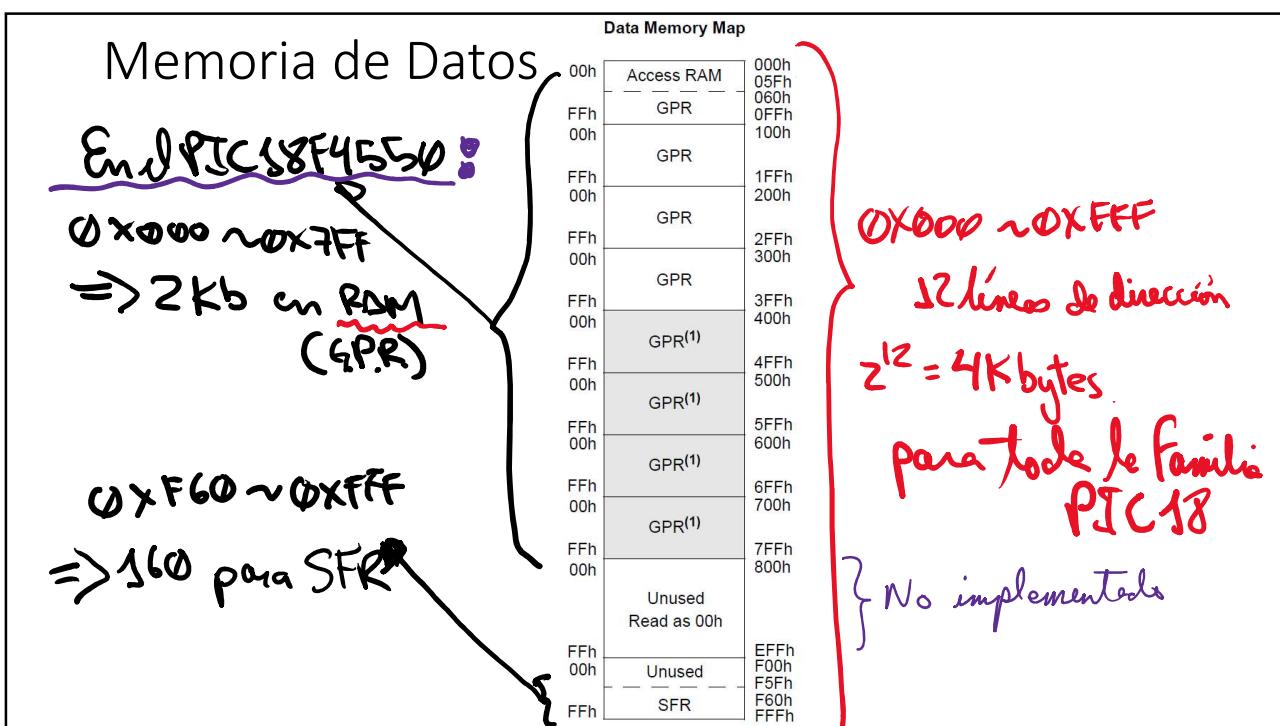
No implementada
en el PIC32F4550.
¿Por qué?

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

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

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

18



19

Detalle de los registros S.F.R.

#include<pic18f4550.h>

$0xFFFF$

PC

TBLPTR

Son 160 registros

Ejemplo:
movwf LATD

lwreg → **0xF8C**

WREG

LATD

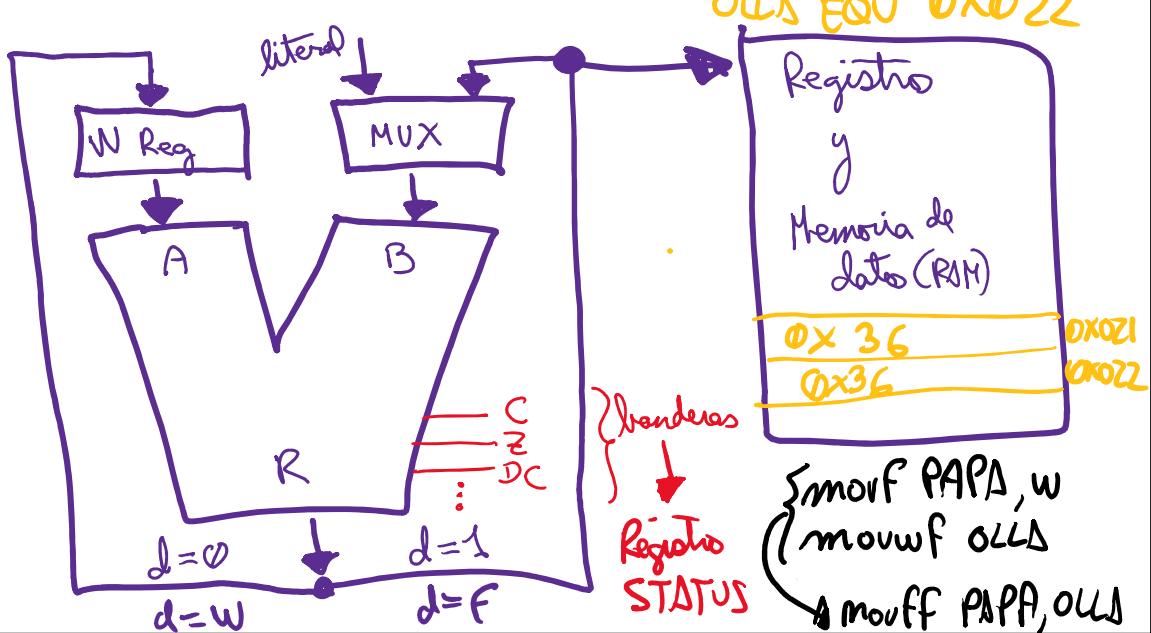
movwf 0xF8C

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR2H	F9Ch	—(2)	F7Ch	UEP12
FFFh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	UEP11
FAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	—(2)	F7Ah	UEP10
FBh	PCL	FDAh	FSR2L	FB9h	—(2)	F99h	—(2)	F79h	UEP9
FF8h	TBLPTR ₀	FDBh	STATUS	F80h	BAUDCON	F98h	—(2)	F78h	UEP8
FF7h	TBLPTRH	FD7h	TMROH	FB7h	ECCP1DEL	F97h	—(2)	F77h	UEP7
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE ⁽³⁾	F76h	UEP6
FF5h	TADLAT	FD5h	TOCON	FB5h	CVRCON	F95h	TRISD ⁽³⁾	F75h	UEP5
FF4h	PRODH	FD4h	—(2)	FB4h	CMCON	F94h	TRISC	F74h	UEP4
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	UEP3
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA	F72h	UEP2
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—(2)	F71h	UEP1
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	—(2)	F70h	UEP0
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—(2)	F6Fh	UCFG
FECh	POSTINC0 ⁽¹⁾	FECh	TMR1L	FAEh	RCREG	F8Eh	—(2)	F6Eh	UADDR
FE Dh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG	F8Dh	—(2)	F6Dh	UCON
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD ⁽³⁾	F6Ch	USTAT
FE Bh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FEAh	FSRH0	FCAh	T2CON	FAAh	—(2)	F8Ah	LATB	F6Ah	UEIR
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA	F69h	UIE
FE8h	SSPADD	FC8h	SSPADD	FA8h	EEDATA	F88h	—(2)	F68h	UIR
FE7h	INDF1 ⁽¹⁾	FC7h	SSPSTAT	FA7h	EECON2 ⁽¹⁾	F87h	—(2)	F67h	UFRMH
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSPCON1	FA6h	EECON1	F86h	—(2)	F66h	UFRML
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSPCON2	FA5h	—(2)	F85h	—(2)	F65h	SPPCON ⁽³⁾
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	—(2)	F84h	PORTE	F64h	SPPEPS ⁽³⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	—(2)	F83h	PORTD ⁽³⁾	F63h	SPPCFG ⁽³⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SPPDATA ⁽³⁾
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	—(2)
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	—(2)

0xF60

20

Flujo de datos en el CPU



21

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

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, ADDIW, 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, ADDIW, 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

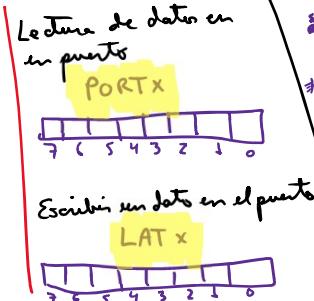
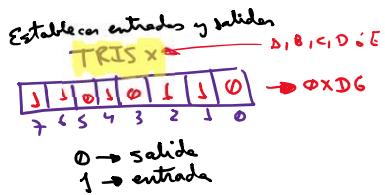
- Note 1: For Borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.
- 2: For Borrow, the polarity is reversed. A subtraction is executed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

22

Manejo de los puertos de E/S en el PIC18F4550

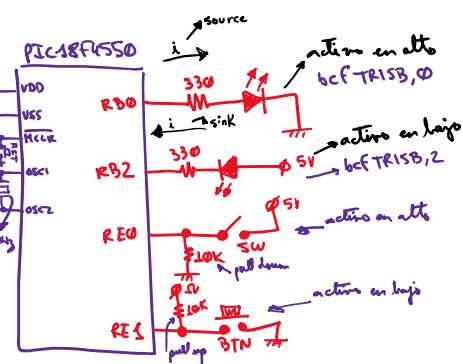
Puerto A: RA0 – RA6 - 7
 Puerto B: RB0 – RB7 - 8
 Puerto C: RC0, RC1, RC2, RC4, RC5, RC6, RC7 - 7
 Puerto D: RD0 – RD7 - 8
 Puerto E: RE0 – RE3 - 4

Necesitamos tres registros:



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

Conexión de LED's y botones:



23

R

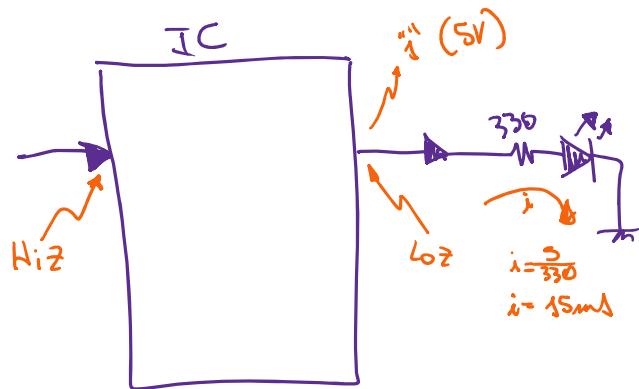
Consideraciones adicionales en el manejo de puertos de E/S:

- Los RB0:RB4 para que sean digitales se debe establecer en el bit de configuración PBADEN = OFF
- Para 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 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

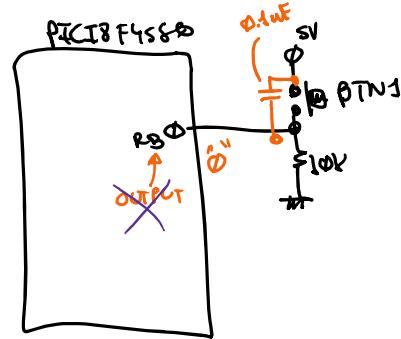
24

Casos:

Alta impedancia en entradas y baja impedancia en salidas:

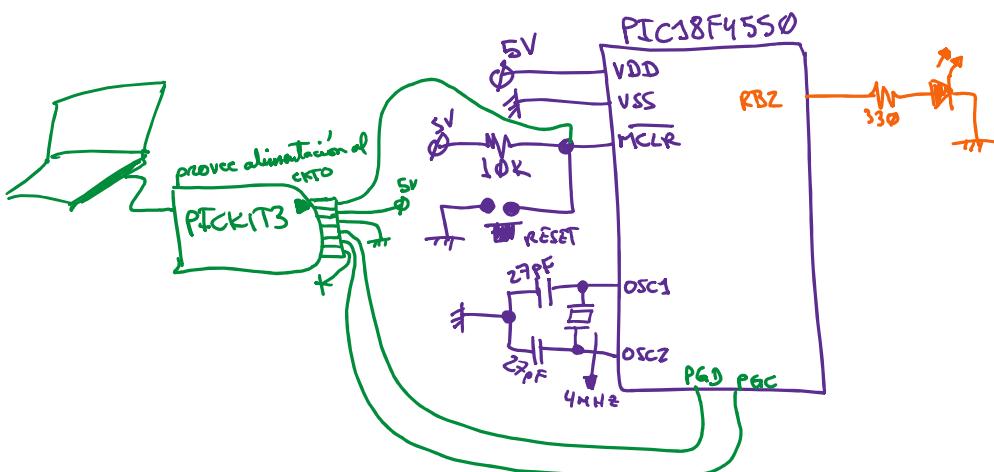


Filtro para rebote de botones empleando condensador:



25

Circuito mínimo de operación del PIC18F4550



26

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?

27

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”?

28