



# MICROCHIP

PIC18 con CCS

(PIC18F14K50)

Autor: *Andrés Raúl Bruno Saravia*  
*Microchip RTC Argentina*

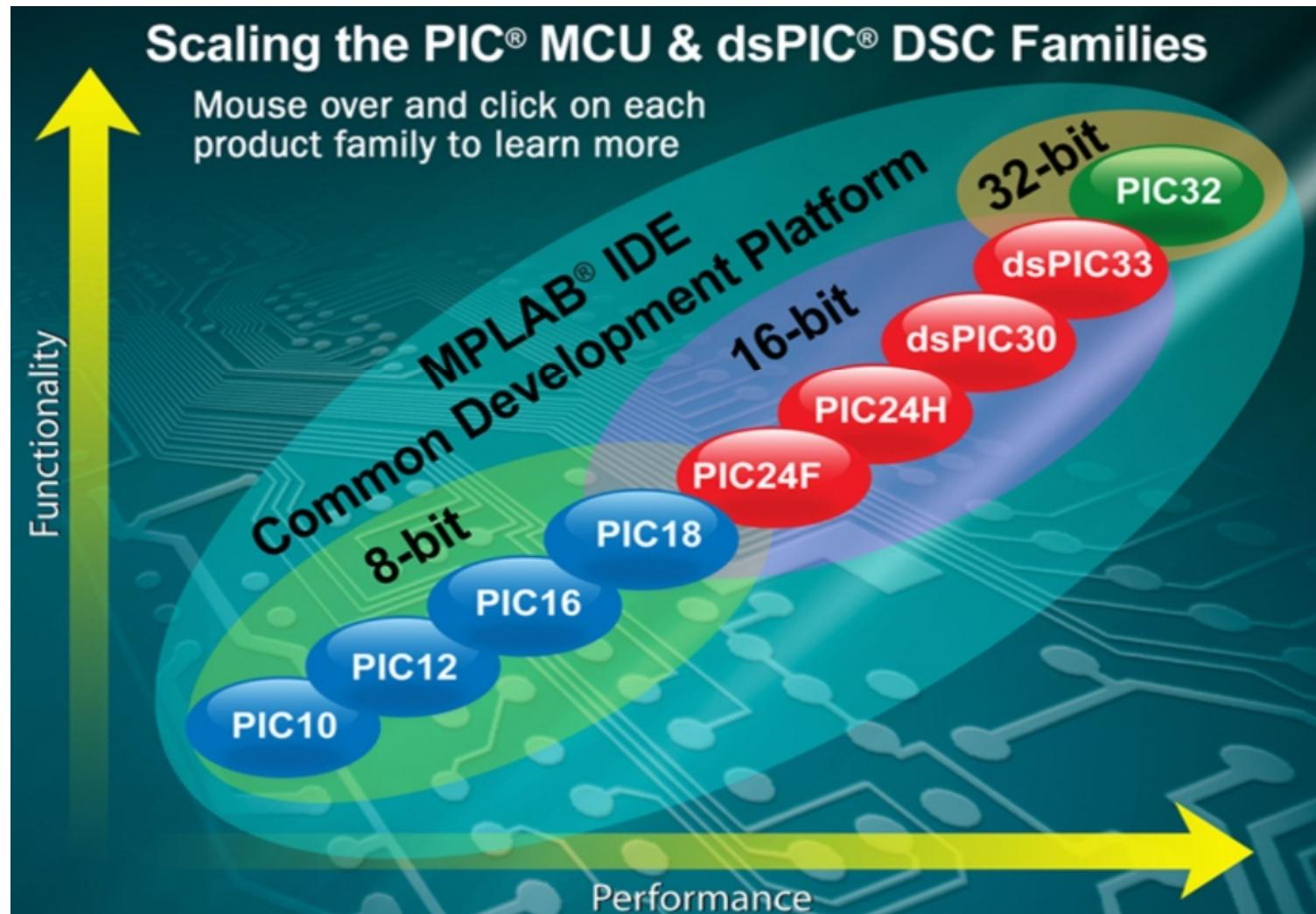


MICROCHIP

# PIC18 Arquitectura

Revisión

# Microchip Familias de MCUs



# Familia PIC18

PIC18 Tradicional

PIC18 serie - J

PIC18 serie - K

4KB

32KB

128KB

## Program Flash

*Tipicamente los productos con mucha memoria tambien tienen muchos pines y perifericos avanzados integrados*

### Tradiciona PIC18

40 MHz, 10 MIPS, 5V  
Flash endurance 100k  
EEPROM  
Características Premium

### PIC18 serie J

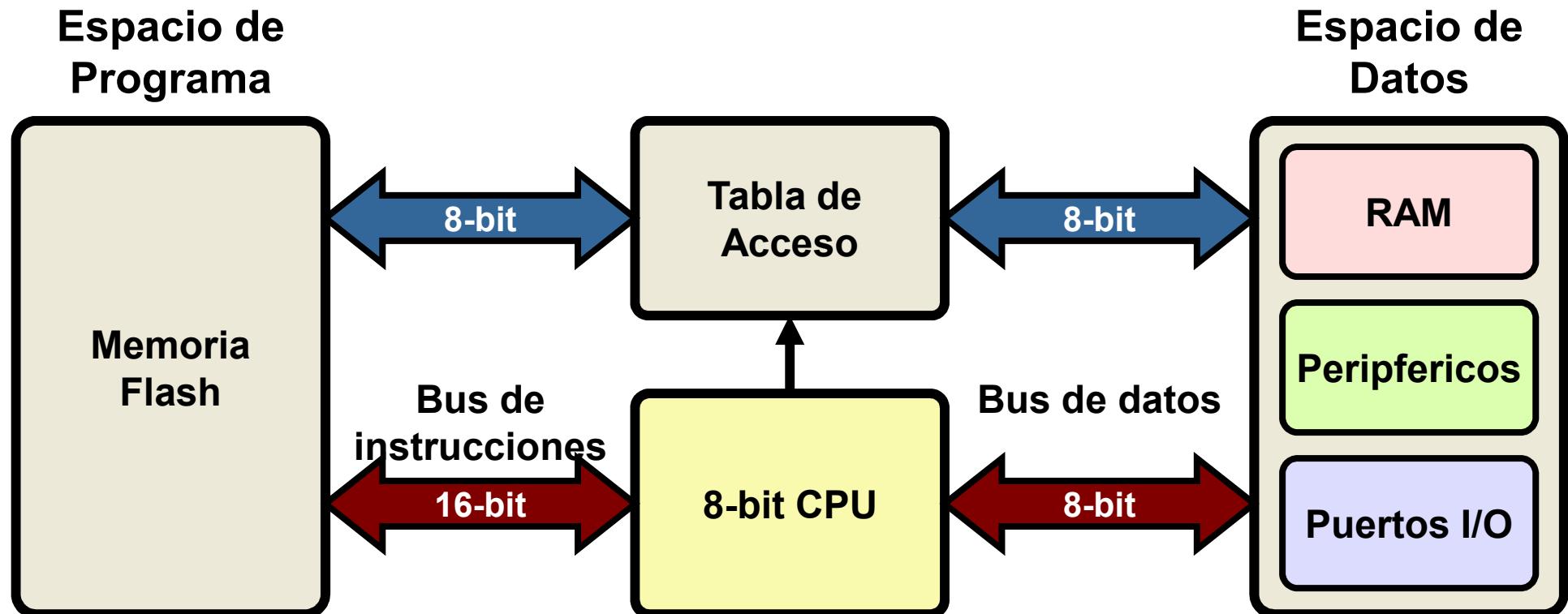
40-48 MHz, 10-12 MIPS, 3V  
Flash endurance 1k – 10k  
Emulate EEPROM  
Menor costo >32KB Flash

### PIC18 serie K

64MHz, 16 MIPS, 3V  
Flash endurance 10k  
EEPROM  
Menor costo <32KB Flash

# Diagrama en Bloques Simplificado

## Arquitectura PIC® 8-bit

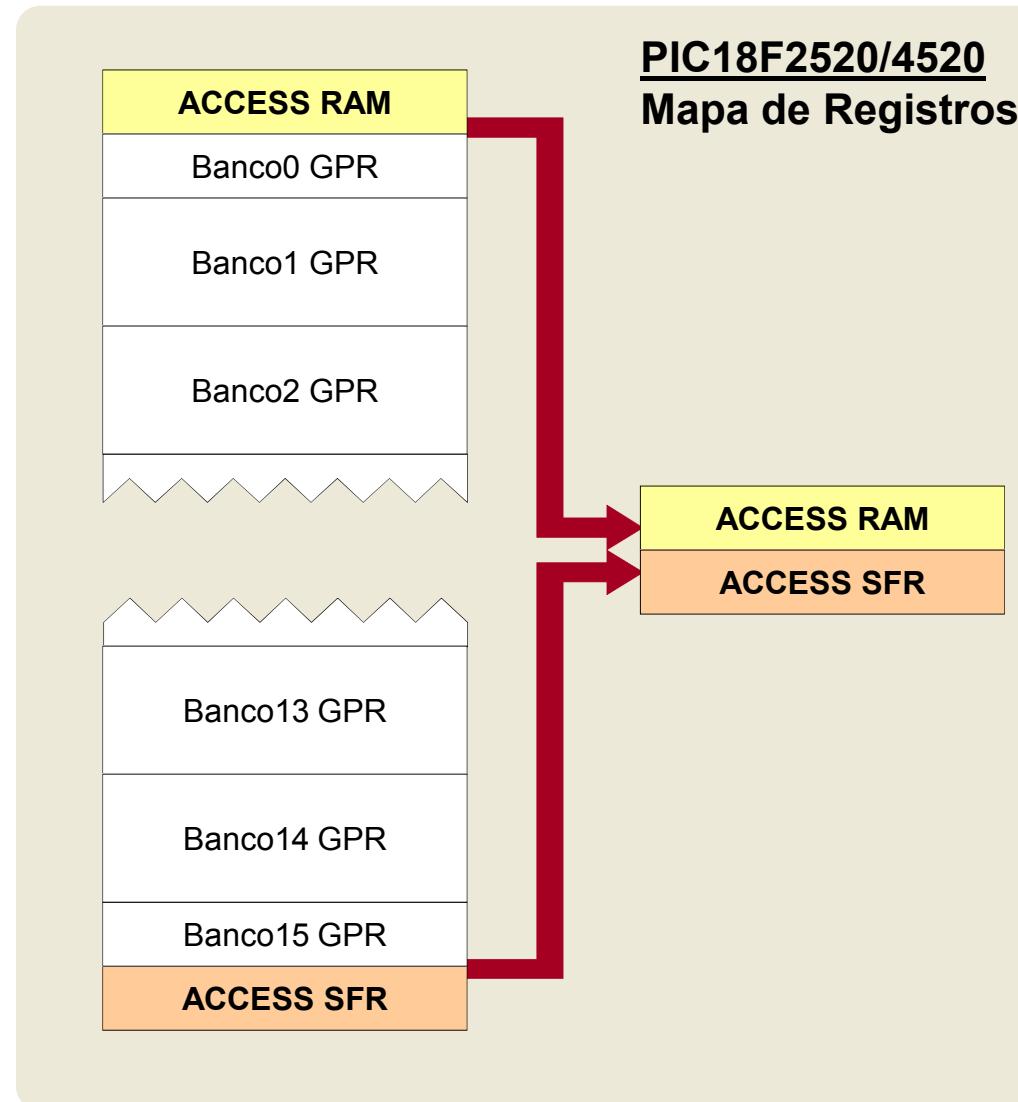


# Mapa de Memoria de Datos

## Arquitectura PIC® 8-bit

- **Memoria de datos hasta 4k bytes**
- **Dividida en bancos de 256 byte**
- **Medio Banco 0 y medio banco 15 son usados para crear un banco virtual de acceso directo**

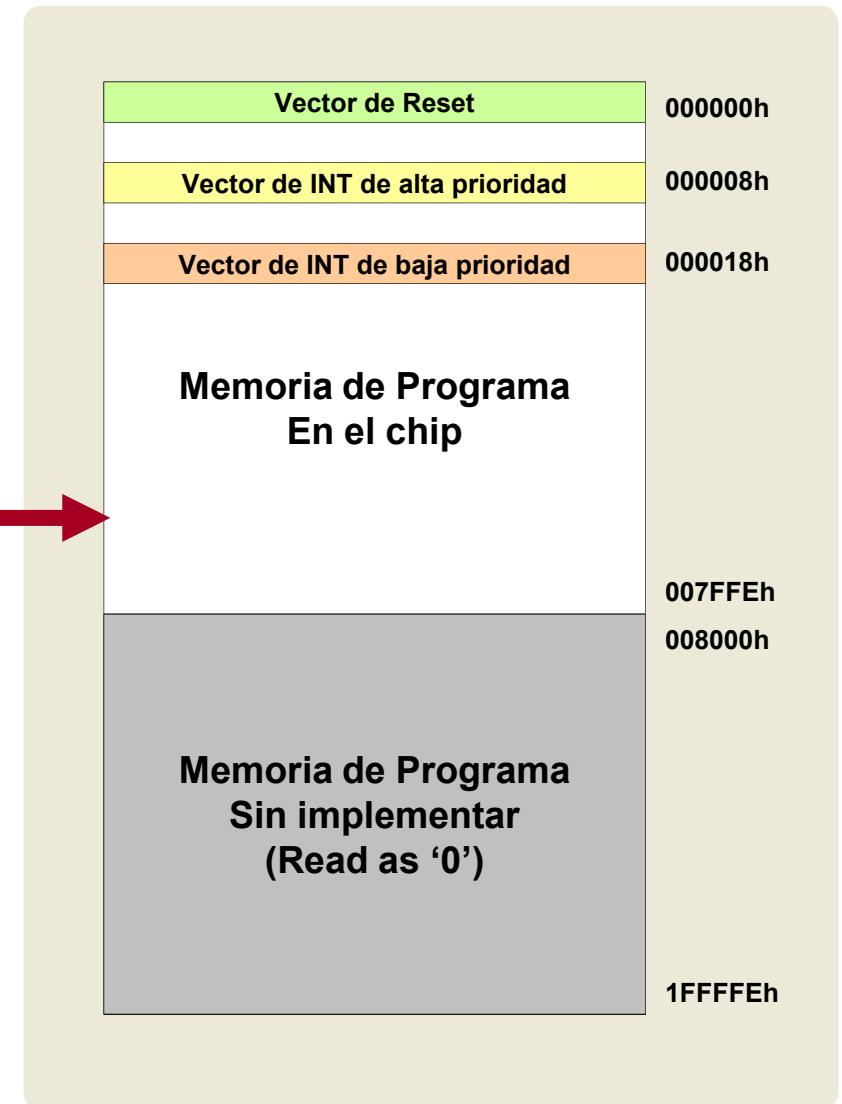
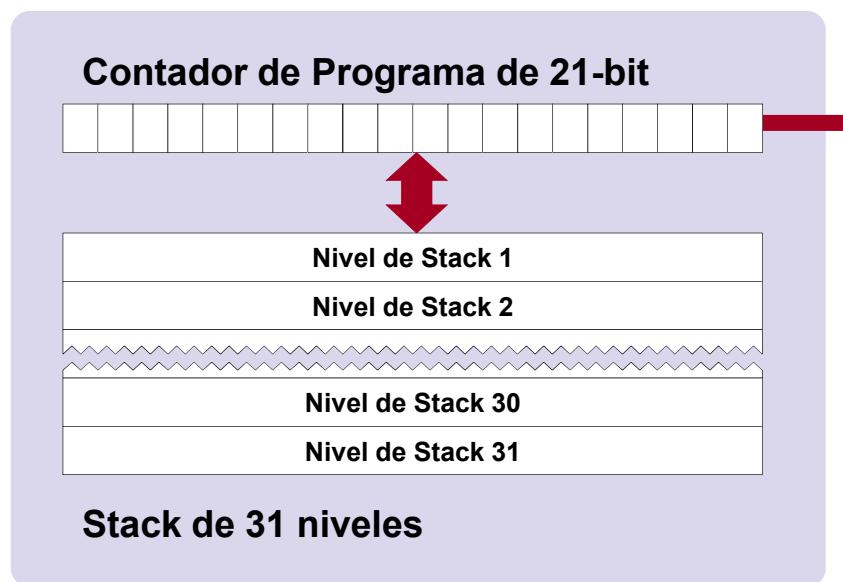
\* La división 50/50 no se mantiene en los dispositivos con muchos registros SFR (Special Function Registers)



# Mapa de memoria de Programa

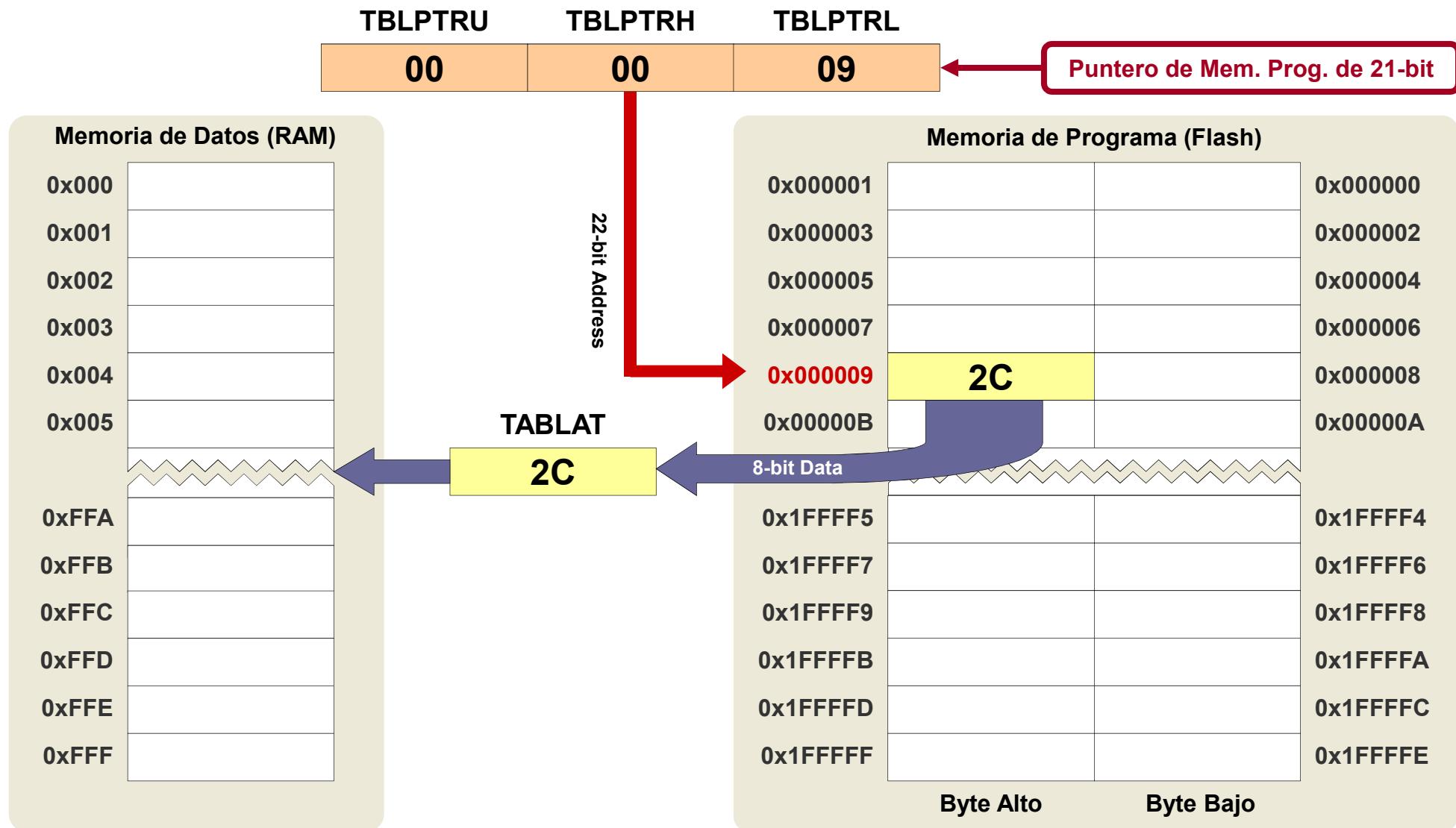
## Arquitectura PIC® 8-bit

- Hasta 2MB (1M Words) continuos totalmente lineal



# Tabla de Lectura

## Arquitectura PIC® 8-bit





MICROCHIP

# Características especiales

Como setear los bits de configuración del PIC®



MICROCHIP

# Osciladores

# PIC18F: Sistema de Clock

## ■ Fuentes de Clock

### ■ Primaria

- Selección Fija
- LP, XT, HS, RC, EC, Int RC Osc

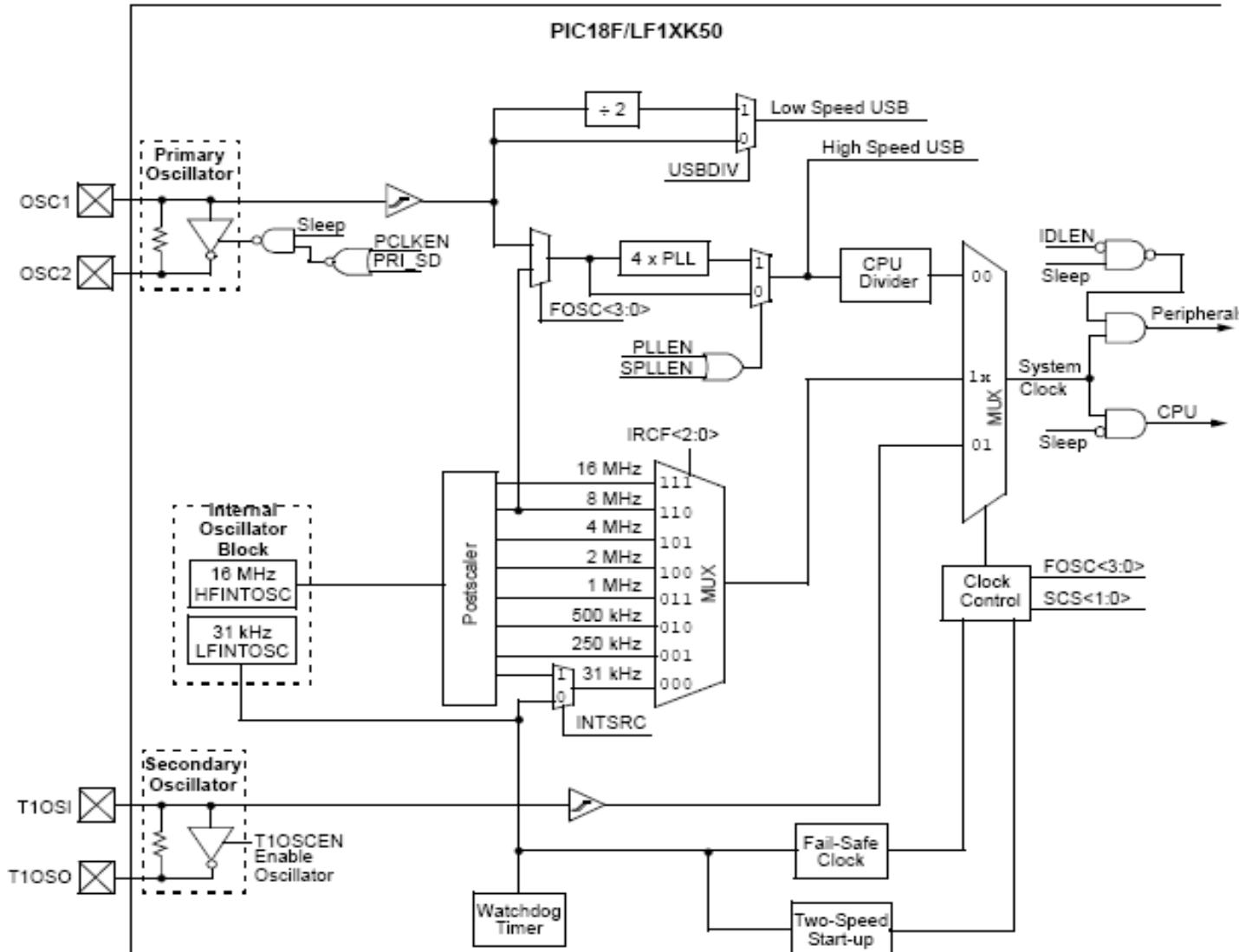
### ■ Secundaria

- Oscilador Timer1 – Frecuencia fija
- Necesario para base de tiempo de RTR

### ■ Oscilador Interno RC

- INTOSC (8 MHz por default)
  - 4, 2, 1 MHz, 500, 250, 125 y 31 kHz seleccionables
- INTRC (31 kHz)

# Sistema de Clock PIC18F1XK50



# PIC18F1XK50 Opciones del Oscilador para el USB

Clock Mode	Clock Frequency	USBDIV	4x PLL Enabled	CPUDIV<1:0>	System Clock Frequency (MHz)
EC High/HS	12 MHz	1	Yes	00	48
				01	24
				10	16
				11	12
			No	00	12
				01	6
				10	4
				11	3
	6 MHz	0	Yes	00	24
				01	12
				10	8
				11	6
			No	00	6
				01	3
				10	2
				11	1.5

LOW SPEED USB CLOCK

Clock Mode	Clock Frequency	4x PLL Enabled	CPUDIV<1:0>	System Clock Frequency (MHz)
EC High	48 MHz	No	00	48
			01	24
			10	16
			11	12
EC High/HS	12 MHz	Yes	00	48
			01	24
			10	16
			11	12

FULL-SPEED USB CLOCK



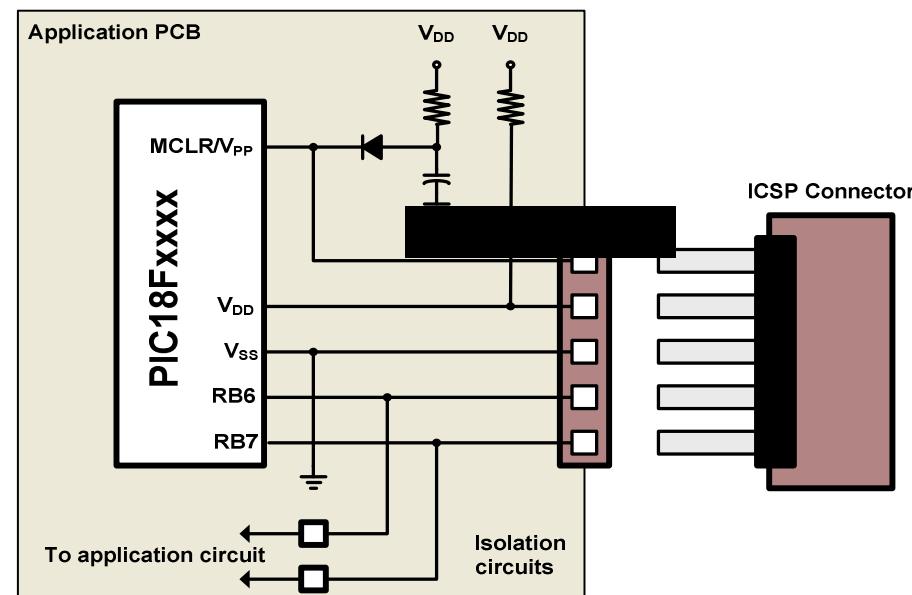
MICROCHIP

# PIC18F Características Especiales

# In-Circuit Serial Programming™

- Solo necesita 2 pines para programar al dispositivo
- Convenient para programar en sistemas
  - Calibración de datos
  - Serialización de datos
- Soportado por debuggers y programadores en MPLAB®

Pin	Function
V <sub>PP</sub>	Programming Voltage
V <sub>DD</sub>	Supply Voltage
V <sub>SS</sub>	Ground
RB6	Clock Input
RB7	Data I/O & Command Input

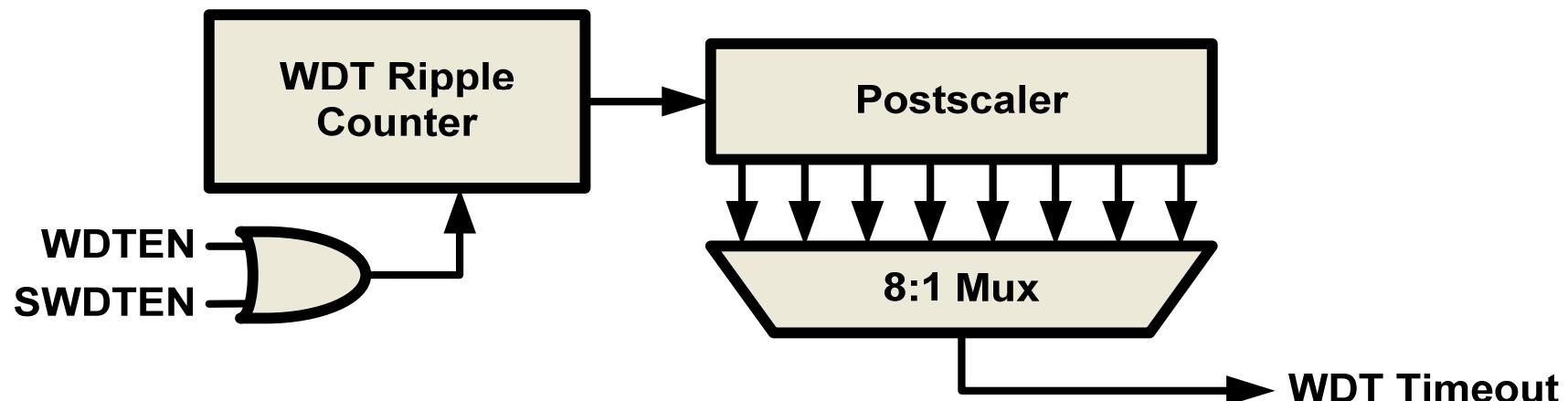


# Características Especiales PIC18F

- **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 reeeea 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

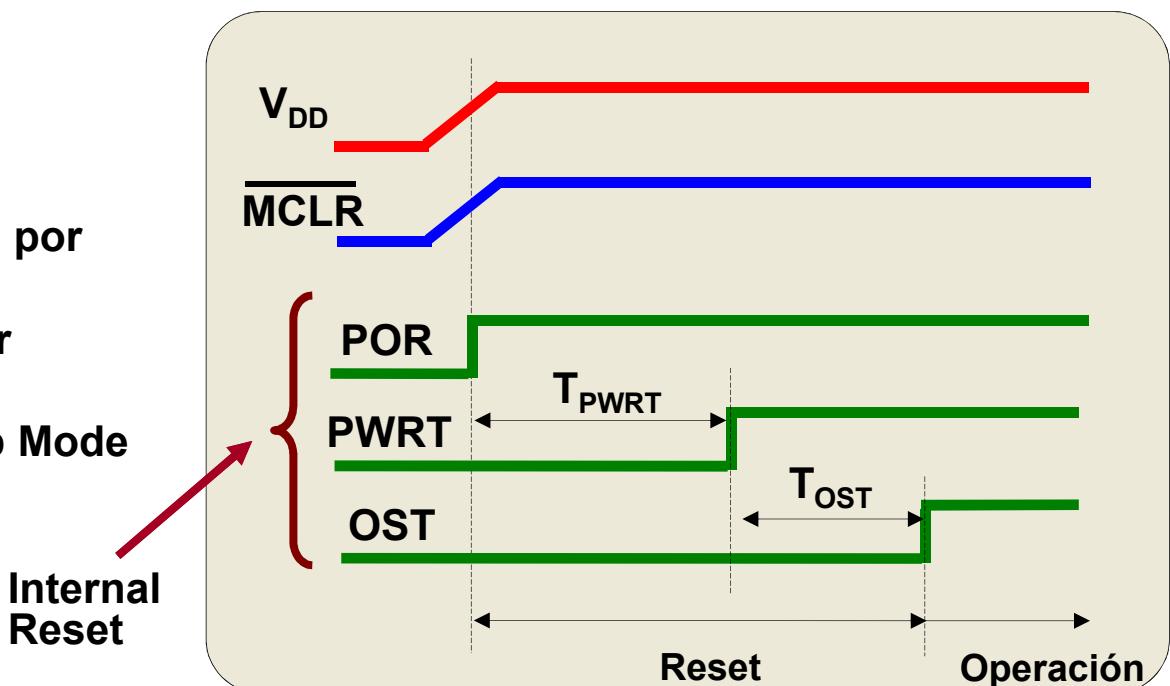
- con MCLR atado a  $V_{DD}$ , es generado un pulso de RESET cuando el flanco de subida a  $V_{DD}$  es detectado

## ■ PWRT: Power Up Timer

- 72 ms (nominal)
- Desacoplado del BOR

## ■ OST: Oscillator Startup Timer

- Mantiene en RESET al dispositivo por 1024 ciclos de maquina (TCYs)
- Le permite al cristal o al resonador estabilizarse
- Bypaseado en: Two Speed Startup Mode
  - INTOSC usa un clock para el procesador no estable



# BOR –Brown Out Reset

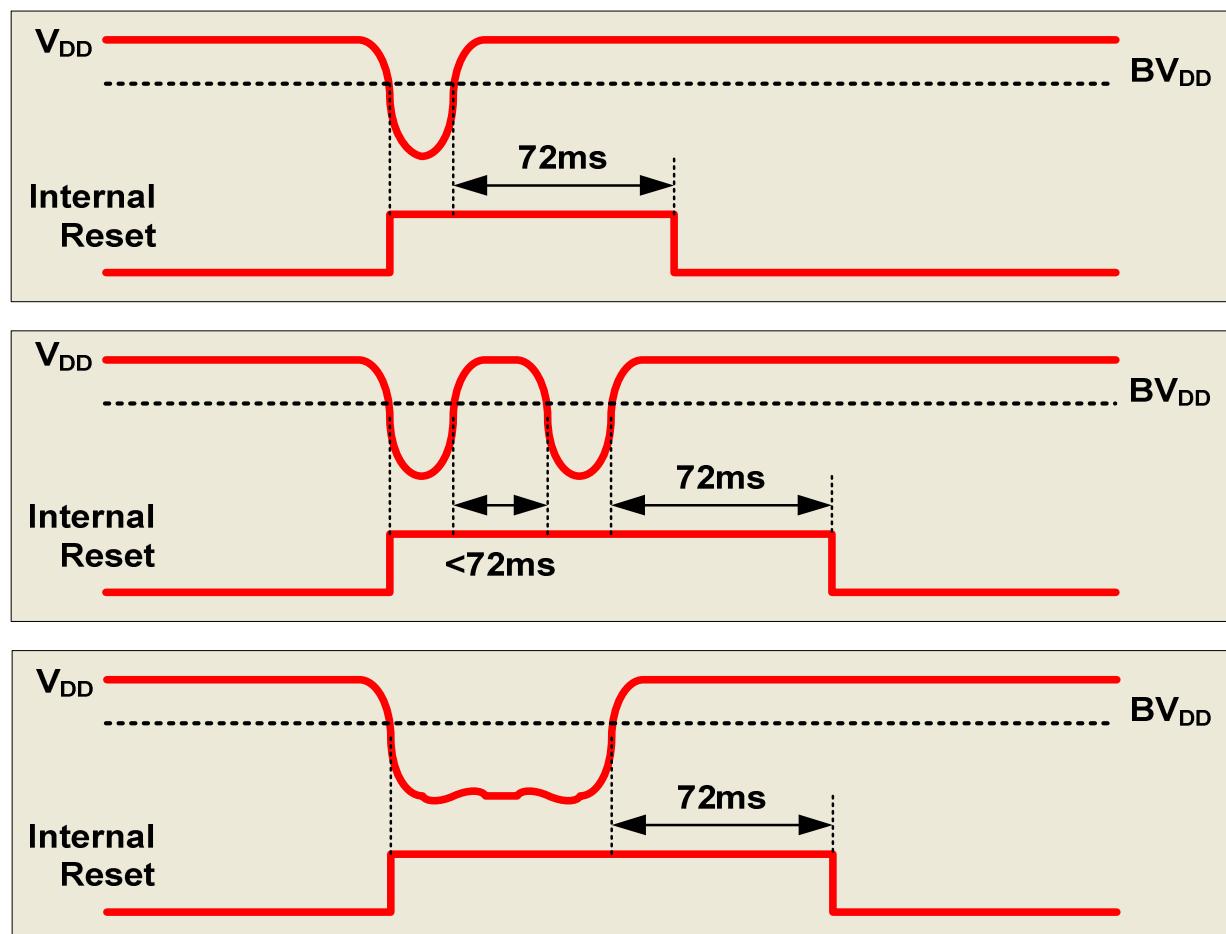
- Cuando el voltaje cae por debajo de un umbral particular, el dispositivo se pone en RESET
- Impide el funcionamiento irregular o inesperado
- Elimina la necesidad de un circuito externo BOR

# 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<sub>DD</sub> para OTP MCUs PICmicro®
  - 2.7V
  - 4.2V
  - 4.5V
- Para otros umbrales use un supervisor externo (MCP1xx, MCP8xx/TCM8xx, or TC12xx)

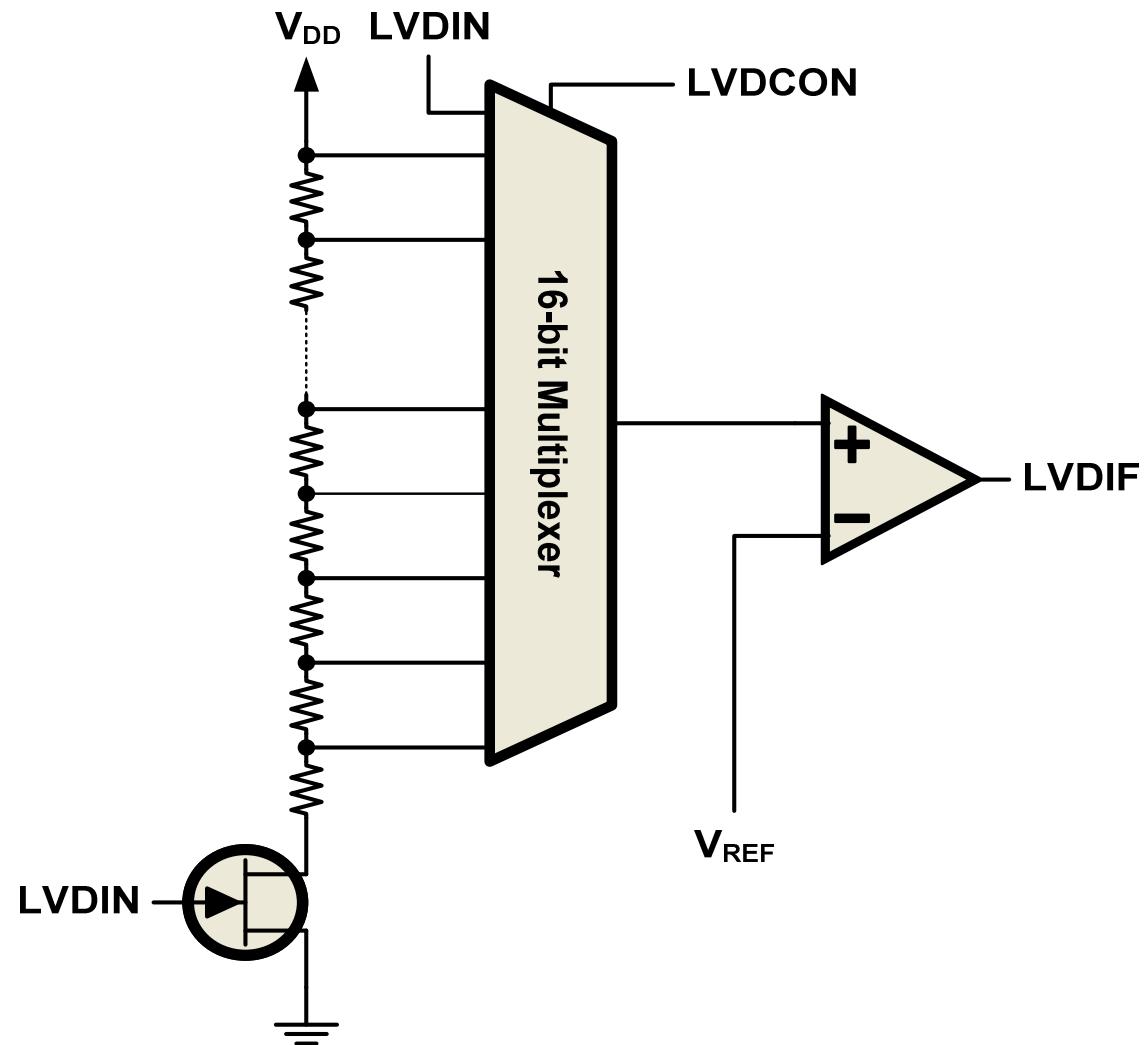
# PBOR – Programmable Brown Out Reset

- Mantiene al MCU PICmicro® MCU en RESET durante ~72ms despues que  $V_{DD}$  supero el nivel de umbral



# PLVD – Detector de Bajo Voltaje Programable

- Alerta antes que el brown out
- 16 Puntos de disparo seleccionables
  - 1.8V hasta 4.5V en pasos de 0.1 a 0.2V
  - Entrada analógica externa
- $V_{REF}$  interno



# PIC18F Características especiales

## Resets

- **PIC18 RESETS**
  - Power-on Reset (POR)
  - MCLR Reset durante la operación normal
  - Programmable Brown-out Reset (BOR)
  - Watchdog Timer (WDT) Reset (durante la ejecución )
  - Instrucción RESET
  - Reset por Stack Full
  - Reset por Stack Underflow
- Para todos los resets el vector del PC es la dirección 0
- Despues del RESET el PC tiene 0x000000

# PIC18F Características especiales

## RESET Registers

### ■ Los siguientes bits son afectados por el RESET

#### ■ RCON Register

- POR = '0': *Power On RESET*
- BOR = '0' & POR = '1': *BOR RESET*
- TO = '0': *WDT RESET*
- RI = '0': *RESET Instrucción*

#### ■ STKPTR Register

- STKFUL = '1': *Stack over flow RESET*
- STKUNF = '1': *Stack under flow RESET*

#### ■ MCLR RESET ies indicado por:

- POR, BOR, TO & RI = '1' y STKFUL & STKUNF = '0'



MICROCHIP

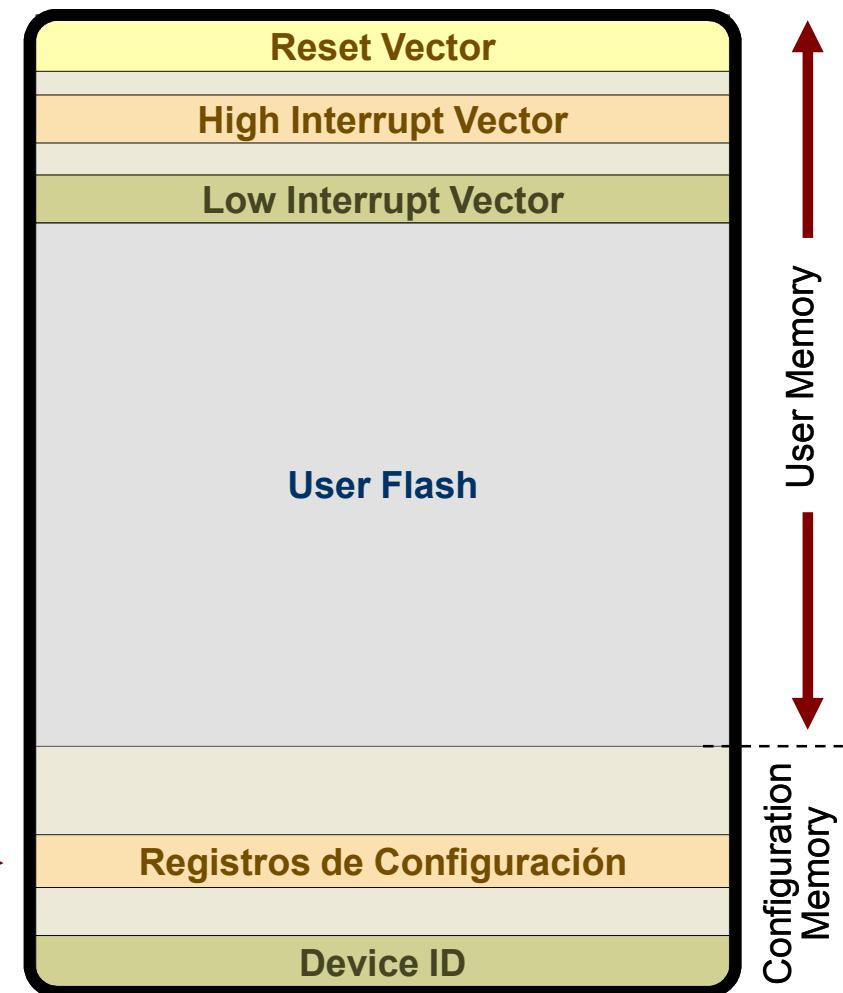
# PIC18 Bits de Configuración

# Que son los bits de Configuración?

- Son usados para activar o no las características especiales:
  - Protección del Código
  - Watchdog Timer
  - Tipo de Oscilador
  - Opción de Debug
  - Otras...

Registro CONFIG localizado en la memoria de programa, fuera del área de ejecución del código (inicia en @ 0x300000)

16-bit Program Memory



# Bits de Configuración

## PIC18F4520 (4 of 11)

### CONFIG1H Register

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| IESO  | FCMEN | —     | —     | FOSC3 | FOSC2 | FOSC1 | FOSC0 |
| bit 7 |       |       |       |       |       |       | bit 0 |

### CONFIG2L Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	BORV1	BORV0	BOREN1	BOREN0	PWRTE
bit 7							bit 0

### CONFIG2H Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN
bit 7							bit 0

### CONFIG3H Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MCLRE	—	—	—	—	LPT1OSC	PBADEN	CCP2MX
bit 7							bit 0

# Configurations Bits con CCS

- Usar directiva **#FUSES**

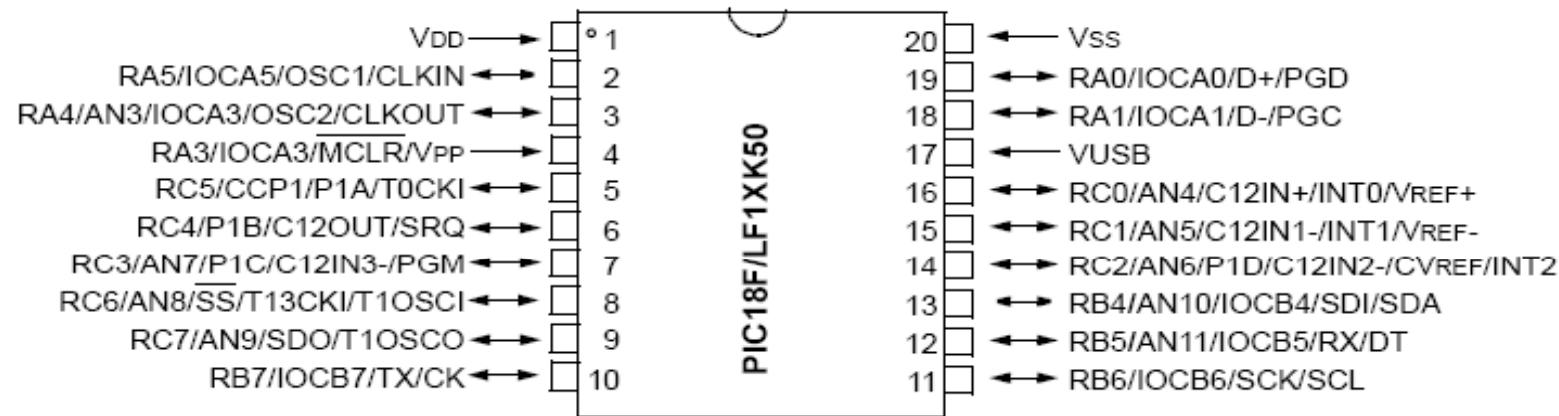
```
#fuses HS,NOWDT
```

- *options* puede variar segun el dispositivo. Un listado de las opciones válidas esta en el tope de cada archivo de cabecera de cada dispositivo.

# PIC18F1x50

Device	Program Memory		Data Memory		I/O <sup>(1)</sup>	10-bit A/D (ch) <sup>(2)</sup>	ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-bit	USB
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I <sup>2</sup> C <sup>TM</sup>				
PIC18F13K50/ PIC18LF13K50	8K	4096	512 <sup>(3)</sup>	256	15	11	1	Y	Y	1	2	1/3	Y
PIC18F14K50/ PIC18LF14K50	16K	8192	768 <sup>(3)</sup>	256	15	11	1	Y	Y	1	2	1/3	Y

# PIC18F14K50 Pin Out





MICROCHIP

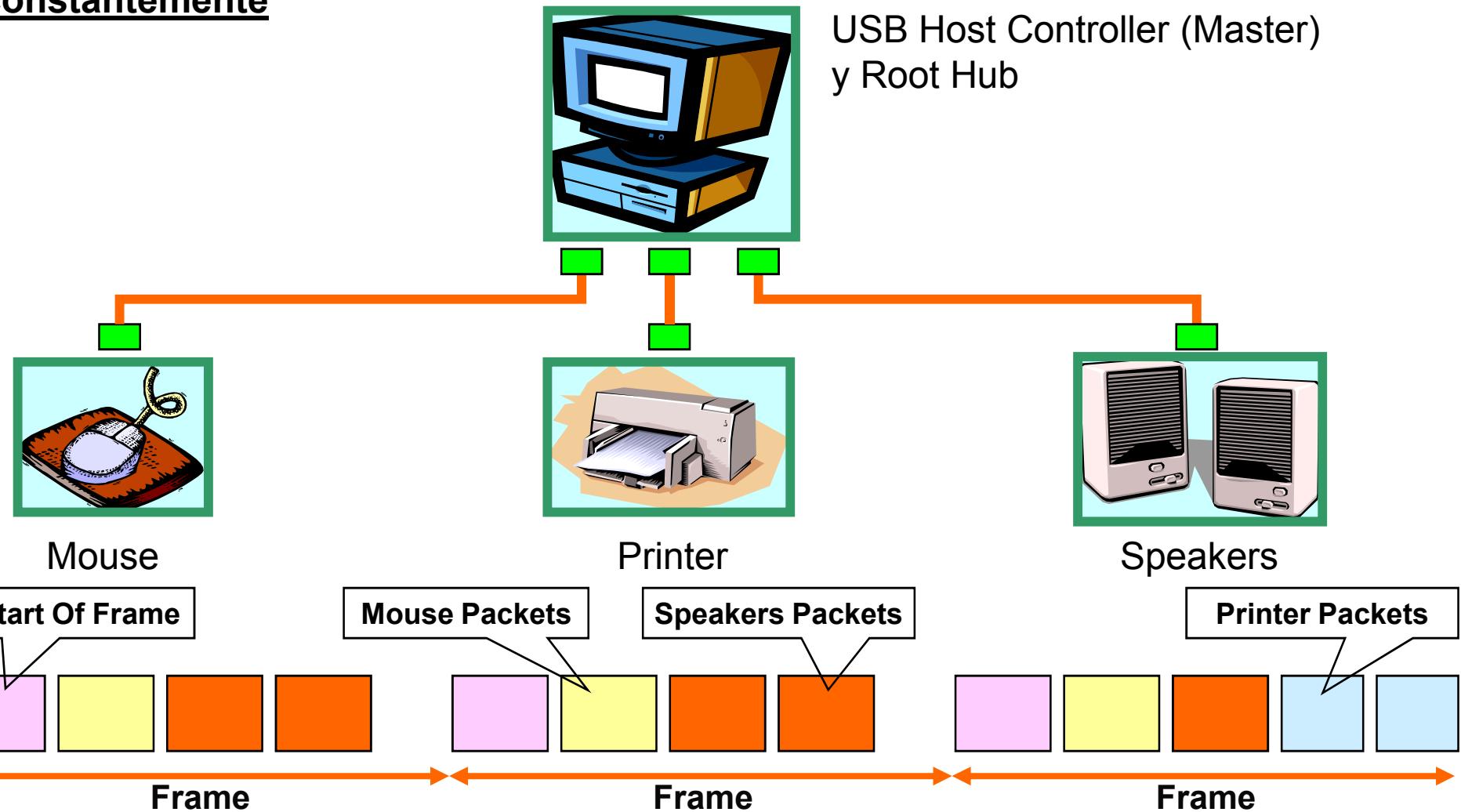
# USB Fundamentos

# Un poco de historia...

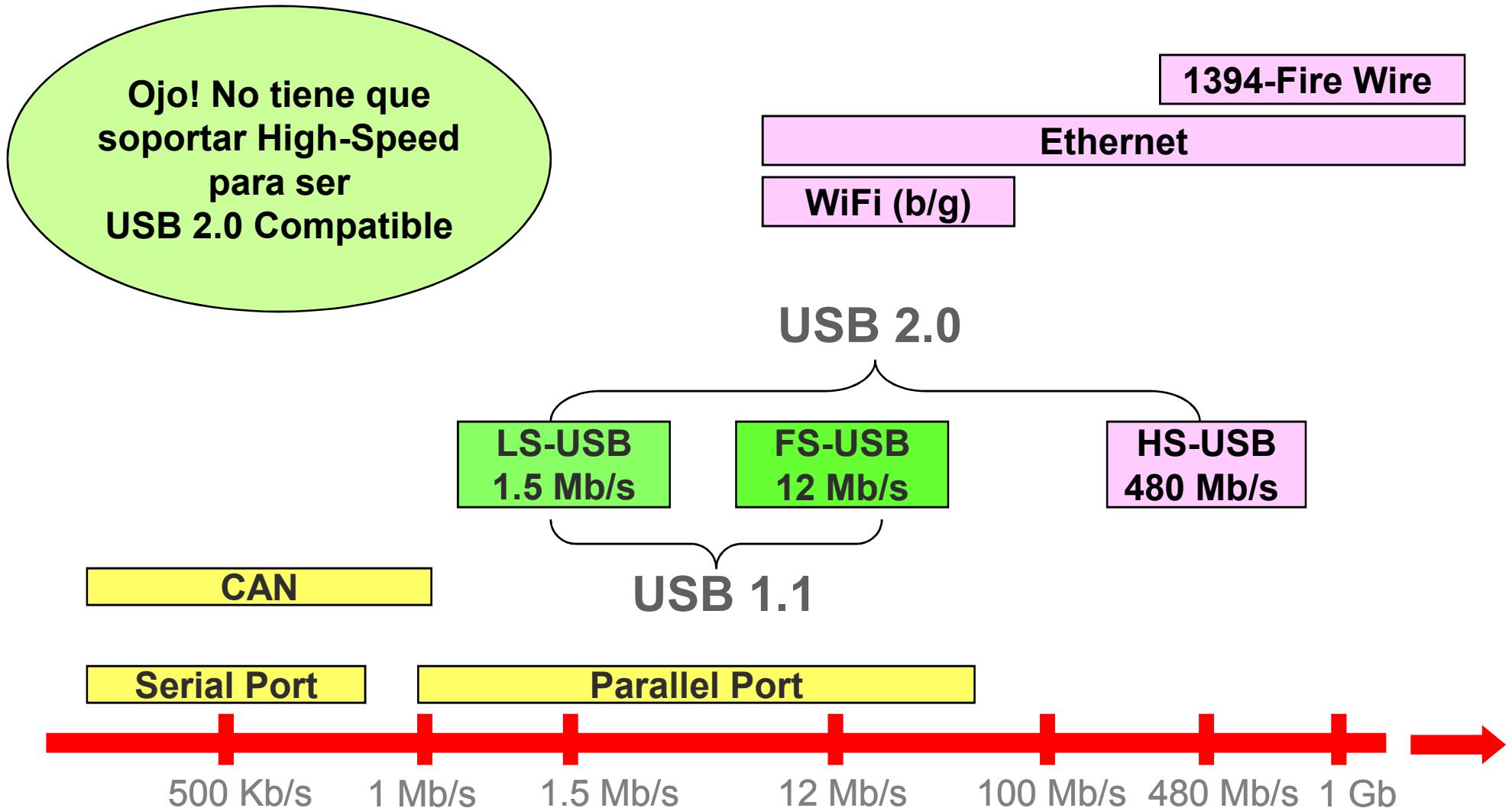
- **USB fue co-desarrollado por un grupo de compañías....**
  - Compaq
  - Intel
  - Microsoft
  - NEC
  - ...quería hacer mucho mas facil el adicionar/remover perifericos de la PCs
- **1998 – USB 1.1**
- **2000 – USB 2.0**
- **2003 – On-the-Go complementa a USB 2.0 (v1.0a)**

# USB Basico

- USB un maestro “Single Master + Multiple Slaves” bus encuestado constantemente



# Comparación de Buses



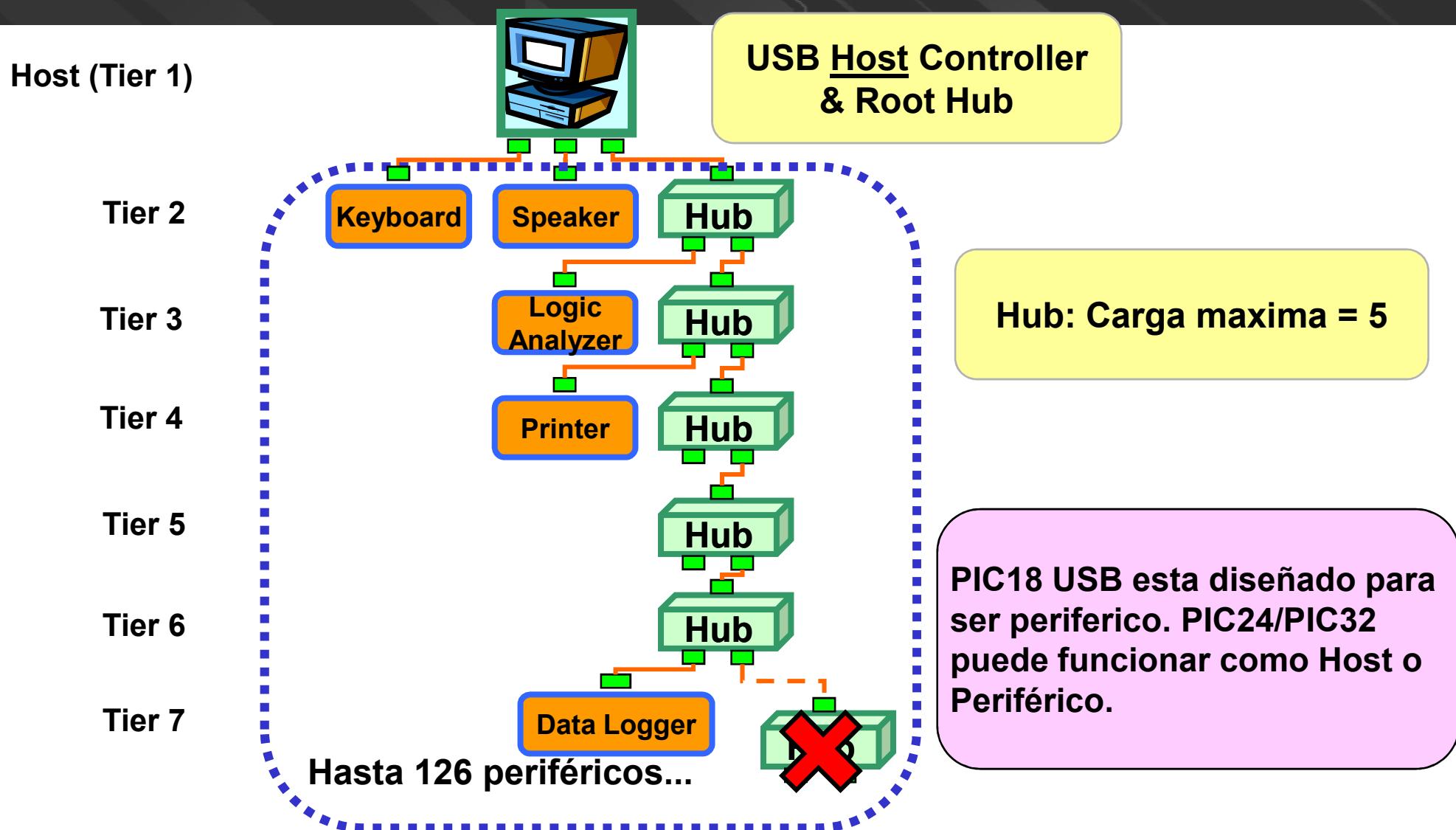
# Grandes Mitos

- **Mito:** un Periférico USB Low-Speed puede transferir datos hasta la velocidad de 187.5 KB/s (1.5 Mbps)
- **Real:** Imposible, por las restricciones de las especificaciones del USB
  - **8 byte de datos transferidos cada 10 ms**
  - **= 800 Bytes/segundo solamente**

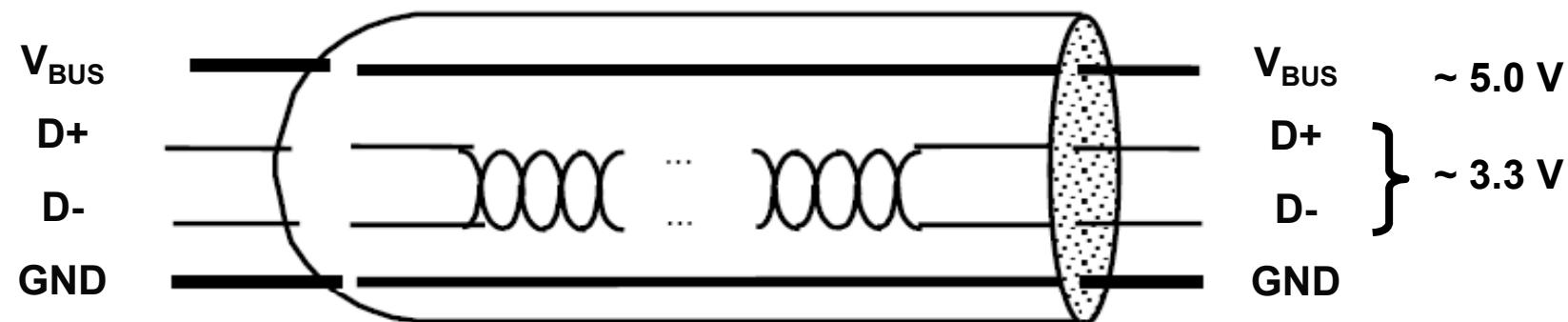
# Siguiente Gran Mito!

- **Mito:** Un periférico USB Full-Speed puede transferir datos hasta 1.5 MB/s (12 Mb/s)
- **Real:** Imposible, 1.5 MB/s es el total del ancho de banda del BUS
  - El ancho de banda se reparte entre todos los periféricos USB
  - El Over head del Protocolo
  - Las restricciones del Protocolo
  - La realidad de la transferencia de datos de un único periférico puede llegar a ~1.0 MB/s en el mejor de los casos
  - Solo 64 KB/s en la mayoría de los casos

# Topología del BUS USB Físico



# Interfaz Física



- Half Duplex con codificación de datos NRZI
- Bus Alimentado para cada dispositivo:
  - 4.40-5.25V
  - Garantizado 100 mA
  - 500 mA max por medio de negociación

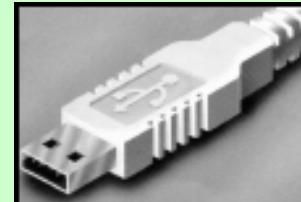
Debe usar alimentación externa si se necesita mas de uno

# Conectores Standard

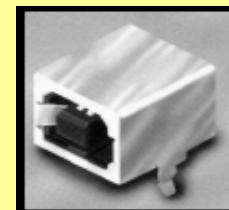
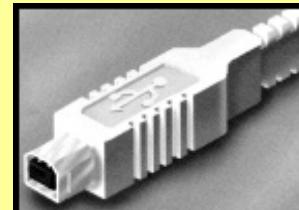
## - USB 2.0 -

“A”

USB Host



“B”  
FS, HS  
Device



“mini-B”  
FS, HS  
Device



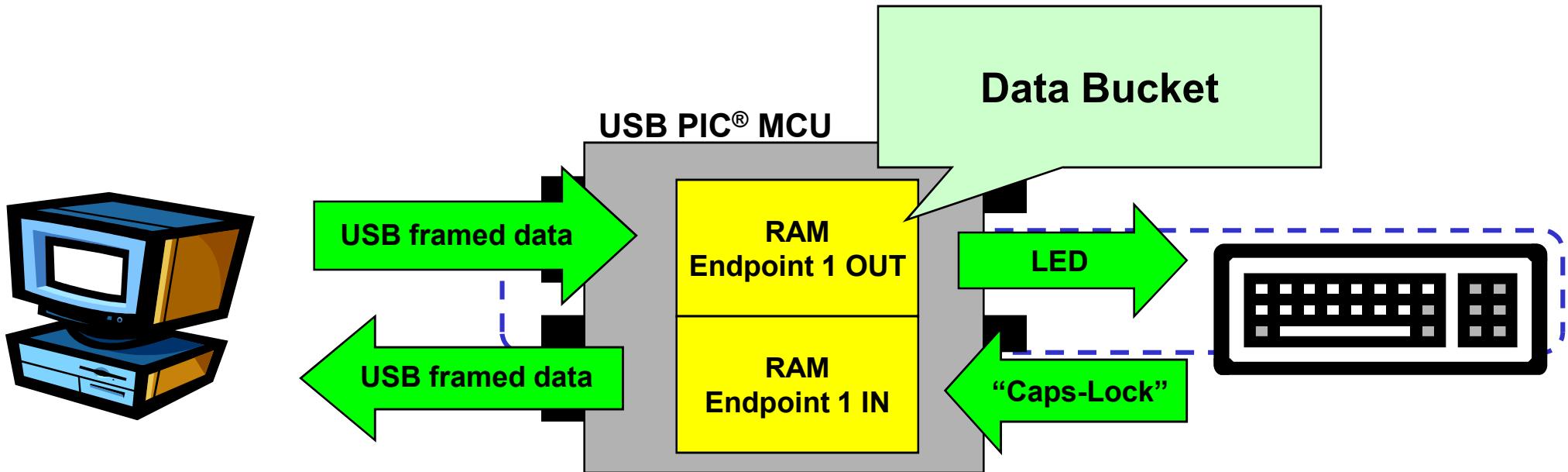
# Conectores Standard

## - Micro-USB -

- **Objetivo:** Define un tamaño más pequeño, conector compatible con USB 2.0 para dispositivos muy pequeños (es decir, teléfonos celulares)
- **Plugs y Receptaculos**
  - **Micro-B plug y receptaculo**
    - Permitido en productos OTG y on OTG
  - **Micro-A/B receptaculo**
    - Permitido en productos OTG
  - **Micro-A plug**
    - Indicacual es escencialmente el host (OTG)

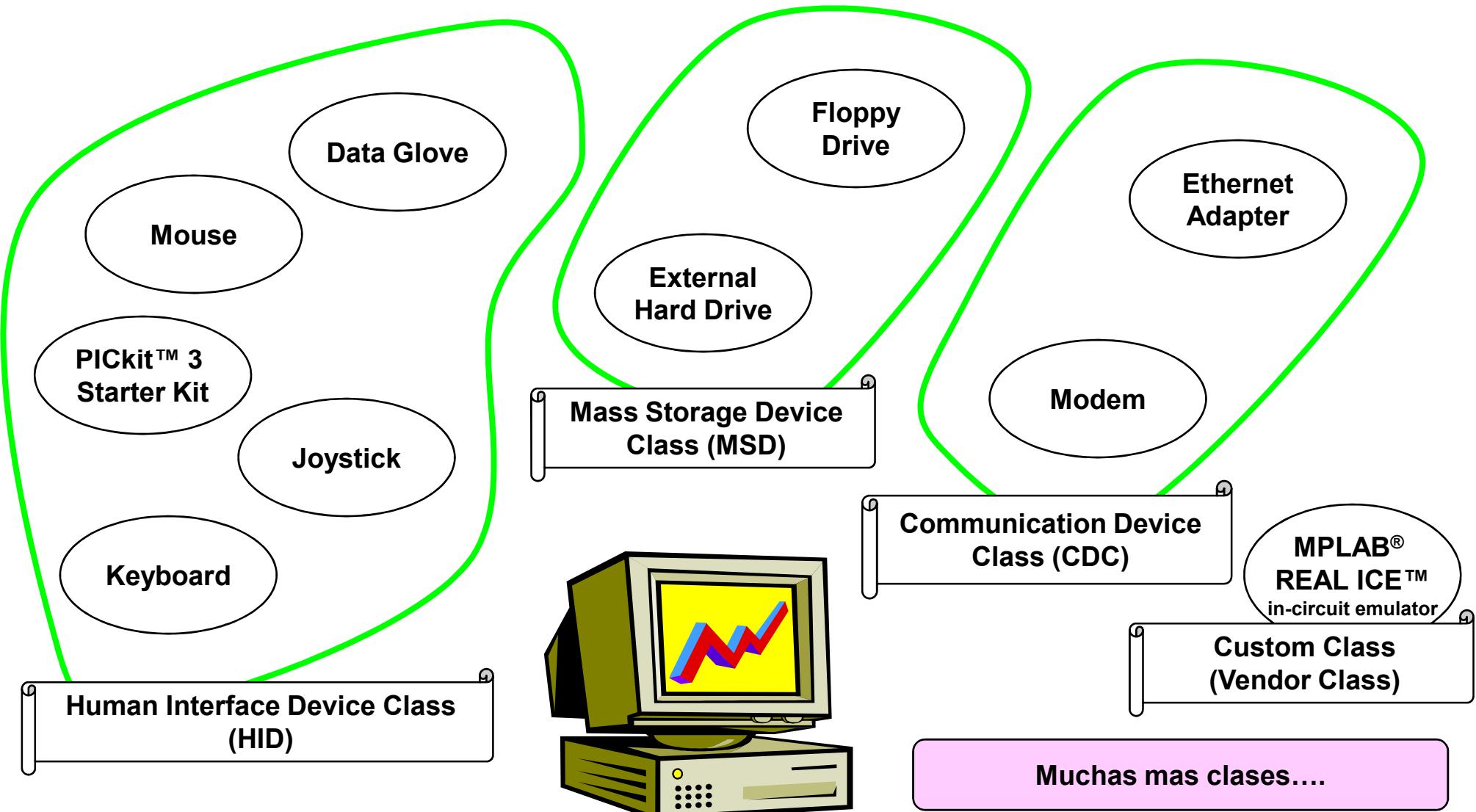


# Endpoints: Source/Destino de datos USB en un periférico

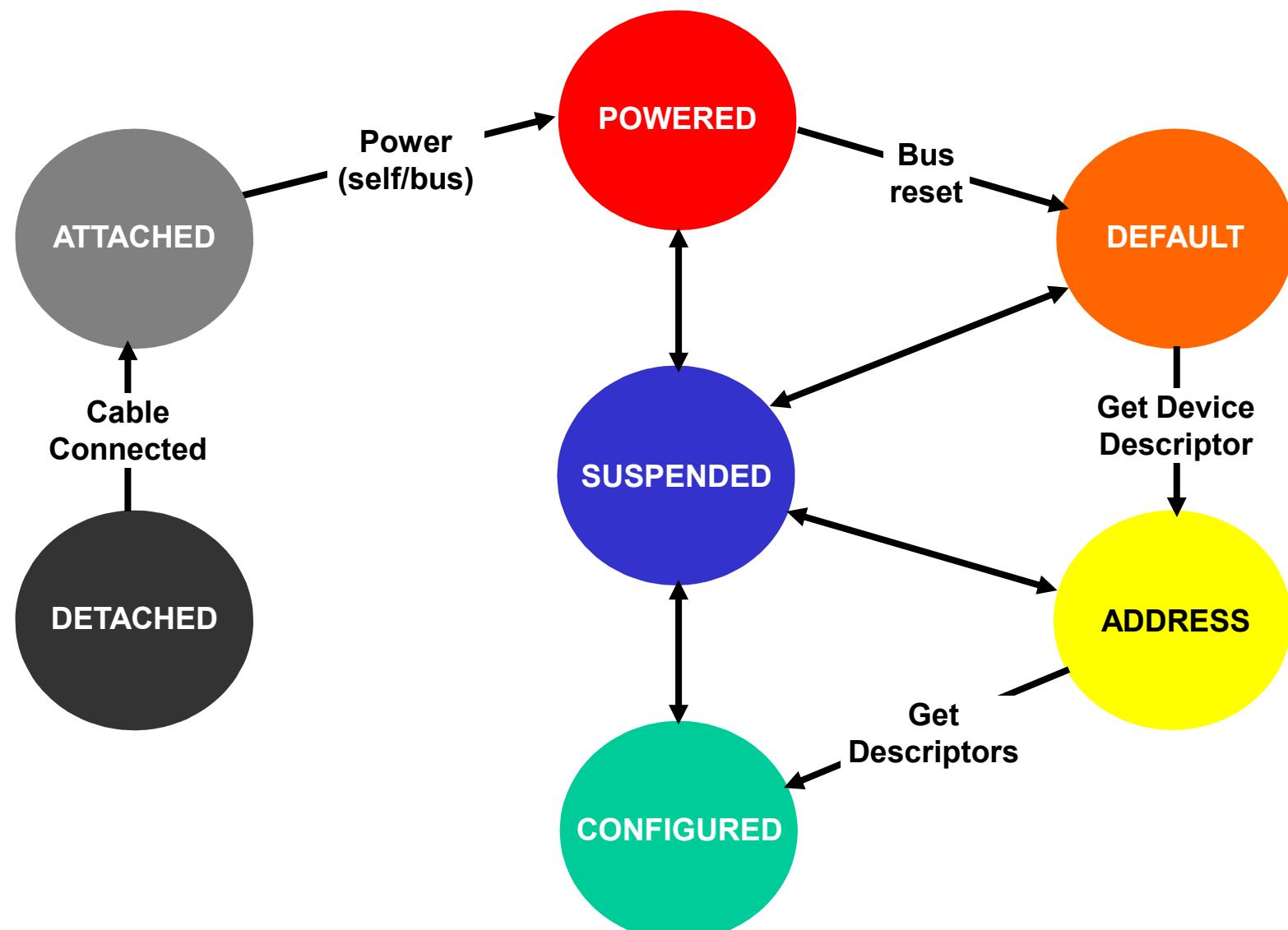


- Número máximo de endpoints por dispositivo especificados por el USB:
  - 16 OUT endpoints + 16 IN endpoints = 32 endpoints
  - PIC18F87J50, PIC18F4550, PIC24F, PIC32MX soporta hasta 32 endpoints
  - ~~PIC18F14K50 soporta hasta 16 endpoints~~
- EP0 = es el “caño” de comunicación por Default

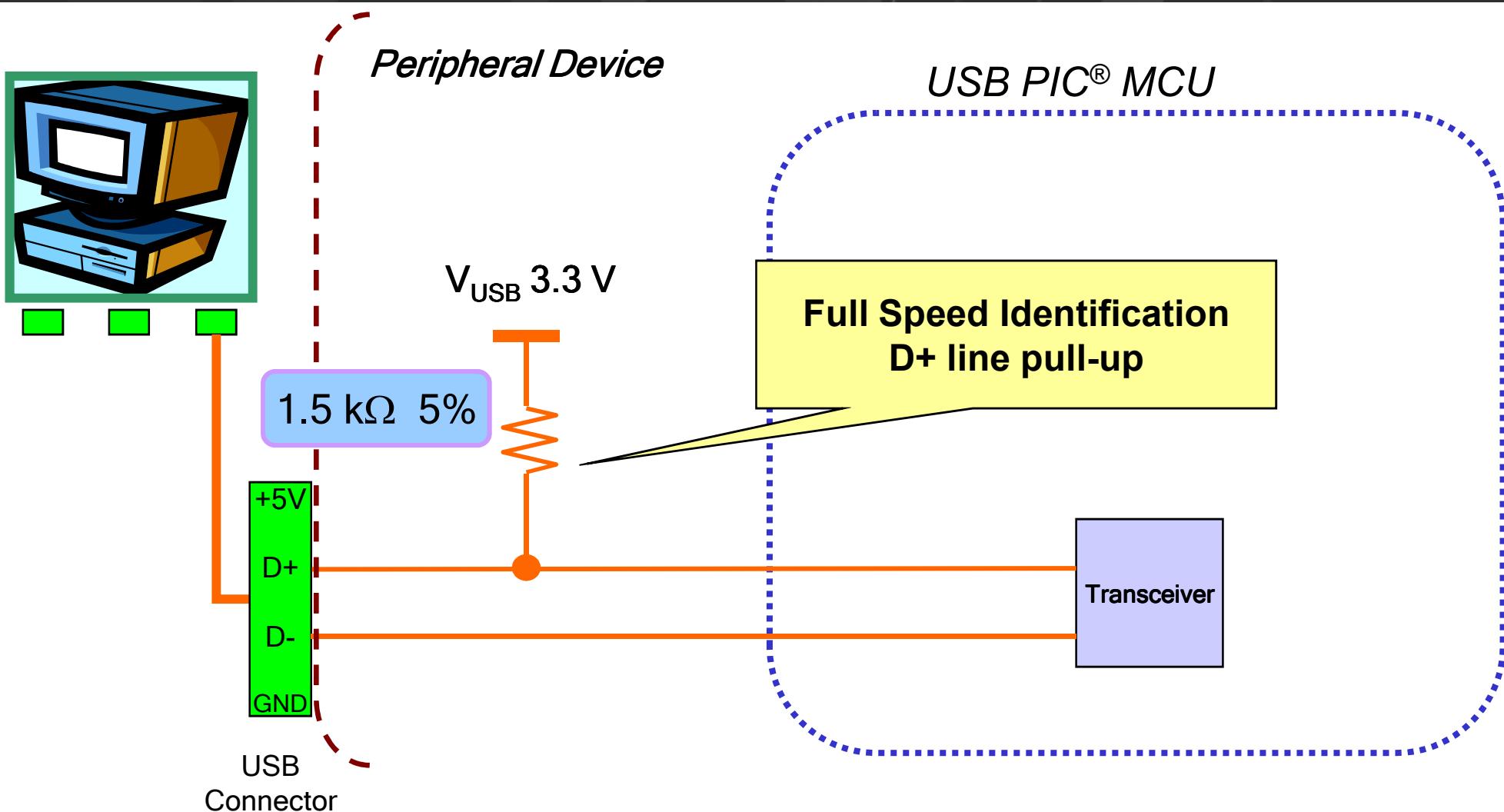
# Clases de Dispositivos USB



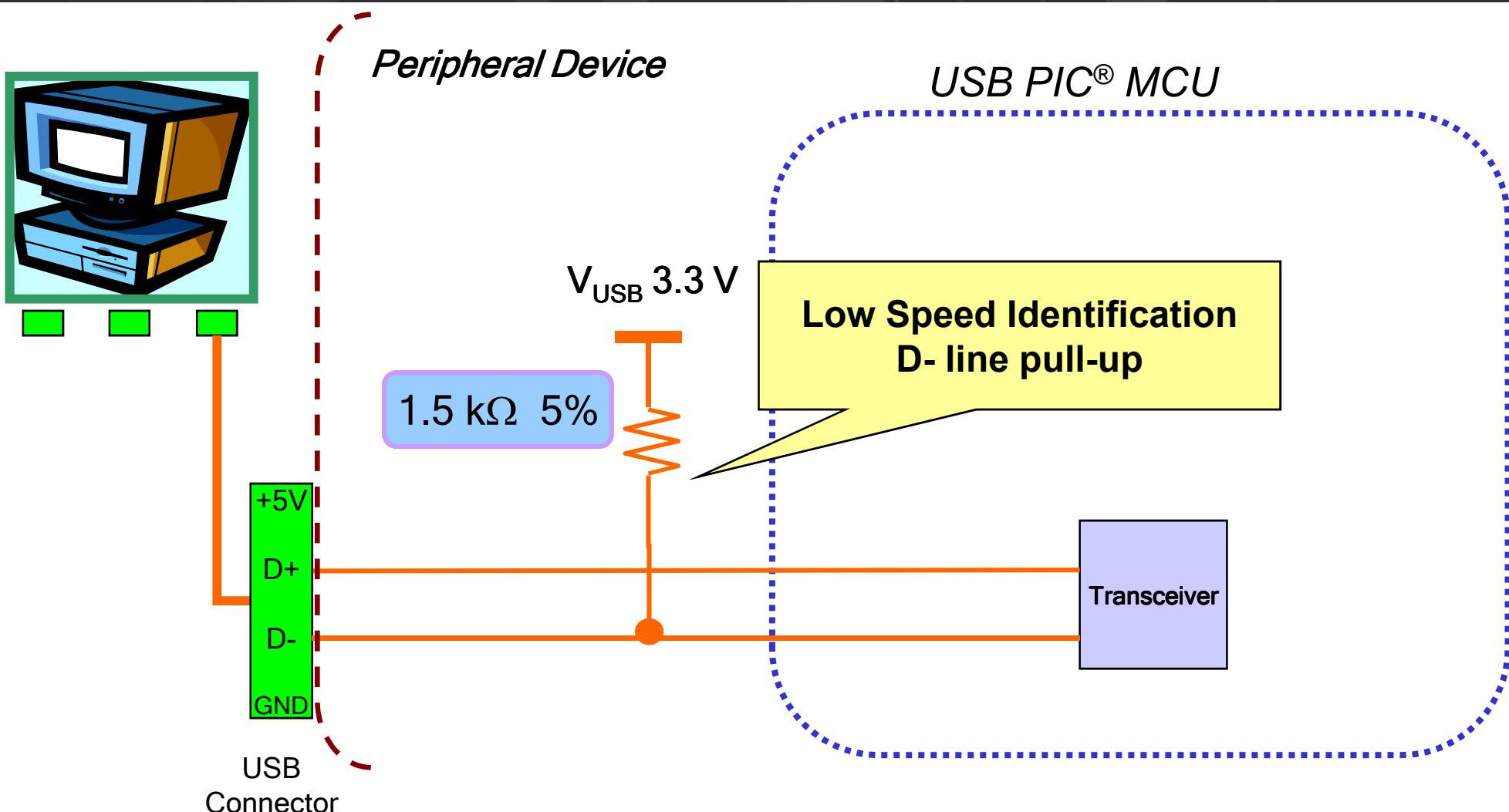
# El Proceso de Enumeración



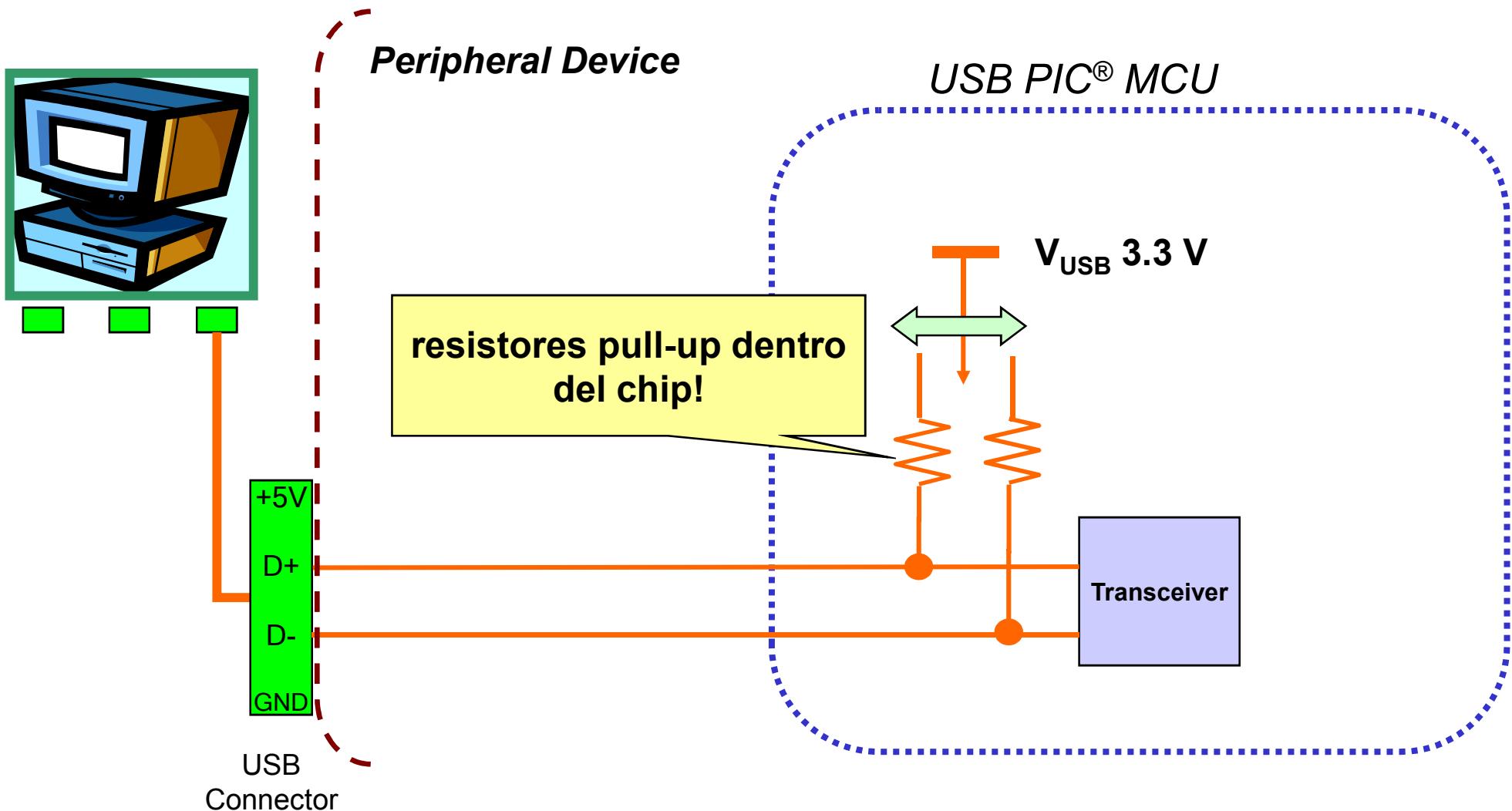
# Auto-Detección: Full-Speed



# Auto-Detección: Low-Speed



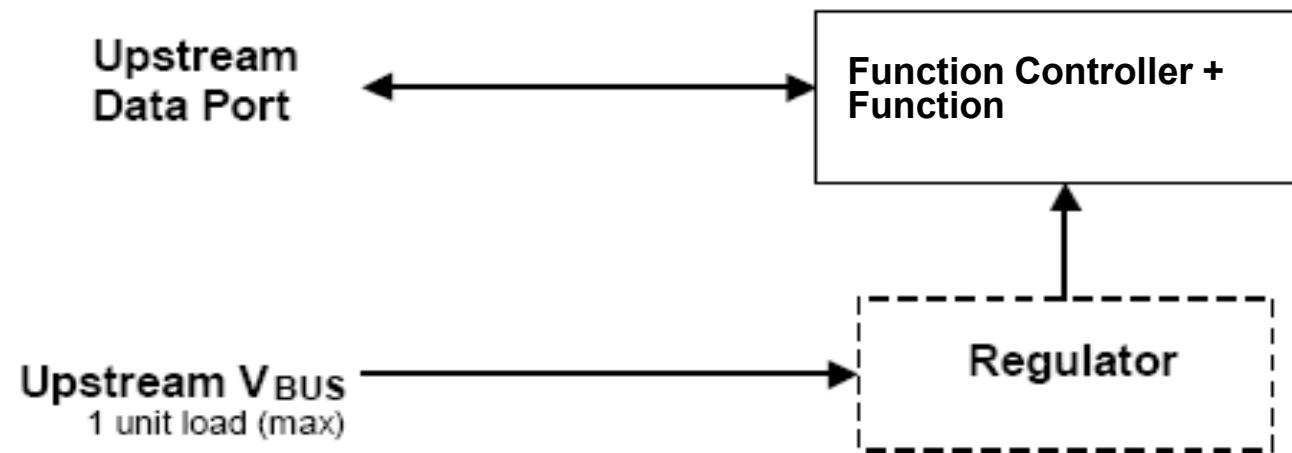
# Resistores Pull-up dentro del chip



# Planeando la Alimentación

## - Arquitectura -

- “Low Power” alimentado del bus
  - Hasta 100 mA (1 ‘carga única’) desde el BUS

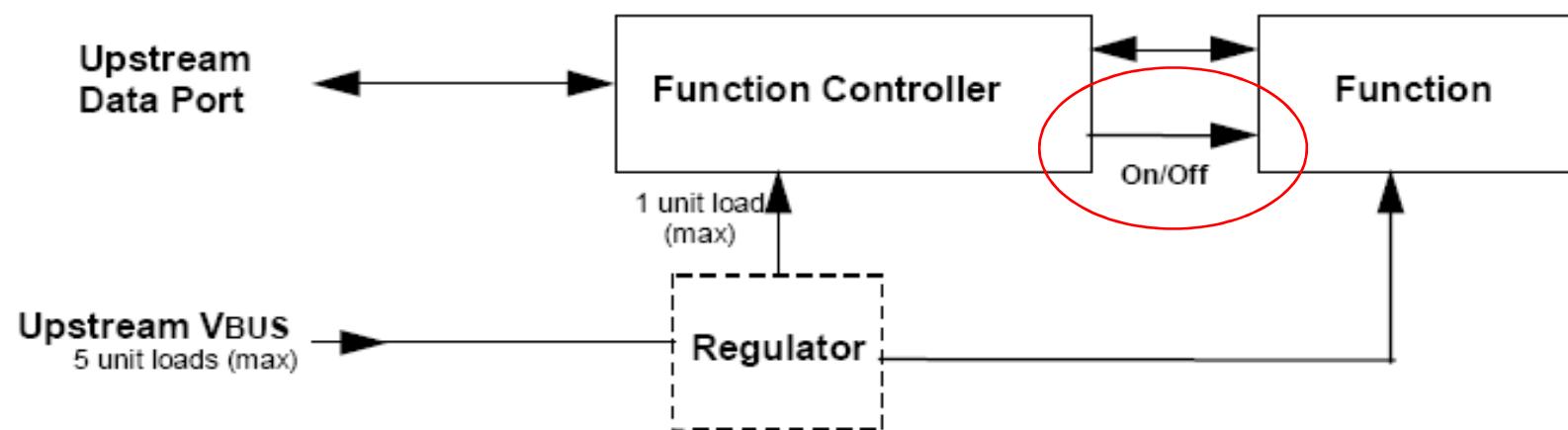


# Planeando la Alimentación

## - Arquitectura -

### ■ “High Power”

- 100-500 mA desde el bus
- Debe ser enumerado a low power (100 mA)
  - Device pide el **bMaxPower**
  - Host habilita la configuración deseada solicitando el **Set\_Configuration**

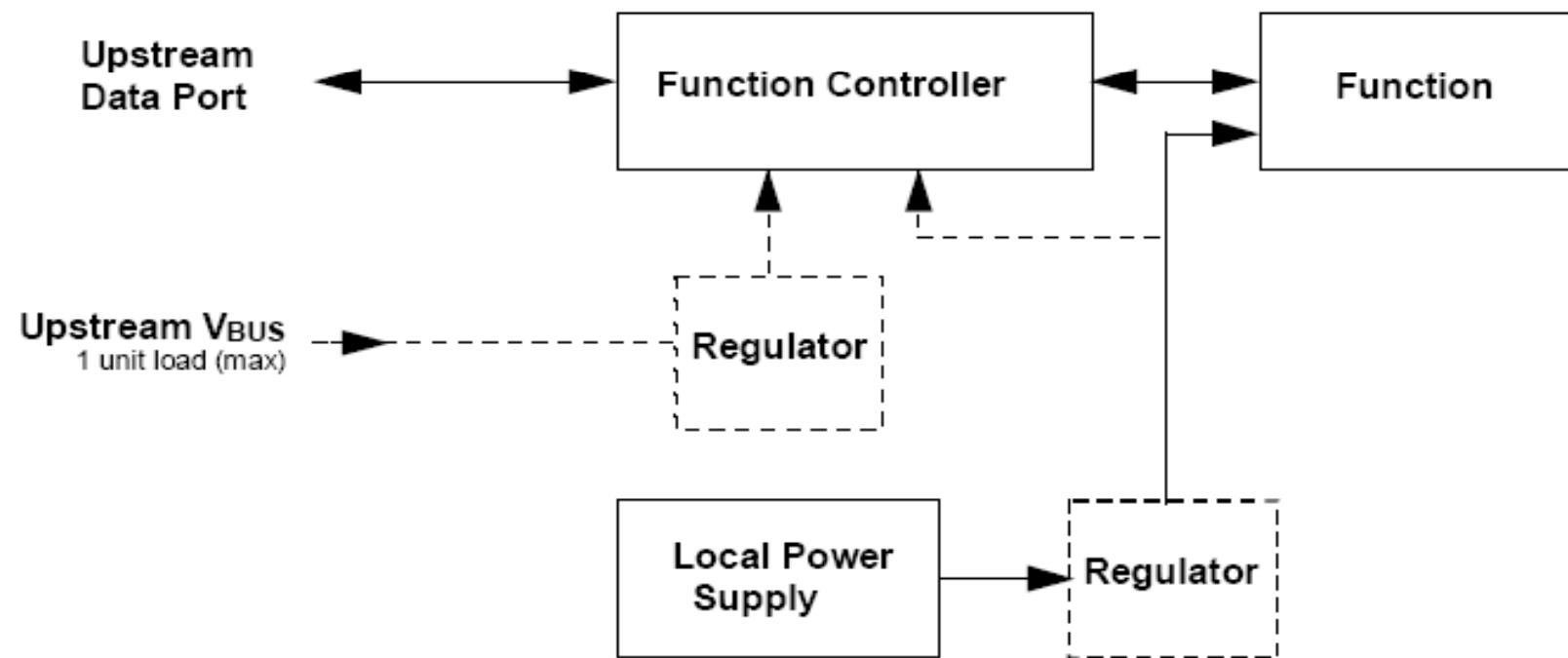


# Planeando la Alimentación

## - Arquitectura -

### ■ “Self Powered” device

- Opcionalmente se puede consumir hasta 100 mA del bus (si es reactivado) + tanto como se encuentra disponible en su propia fuente



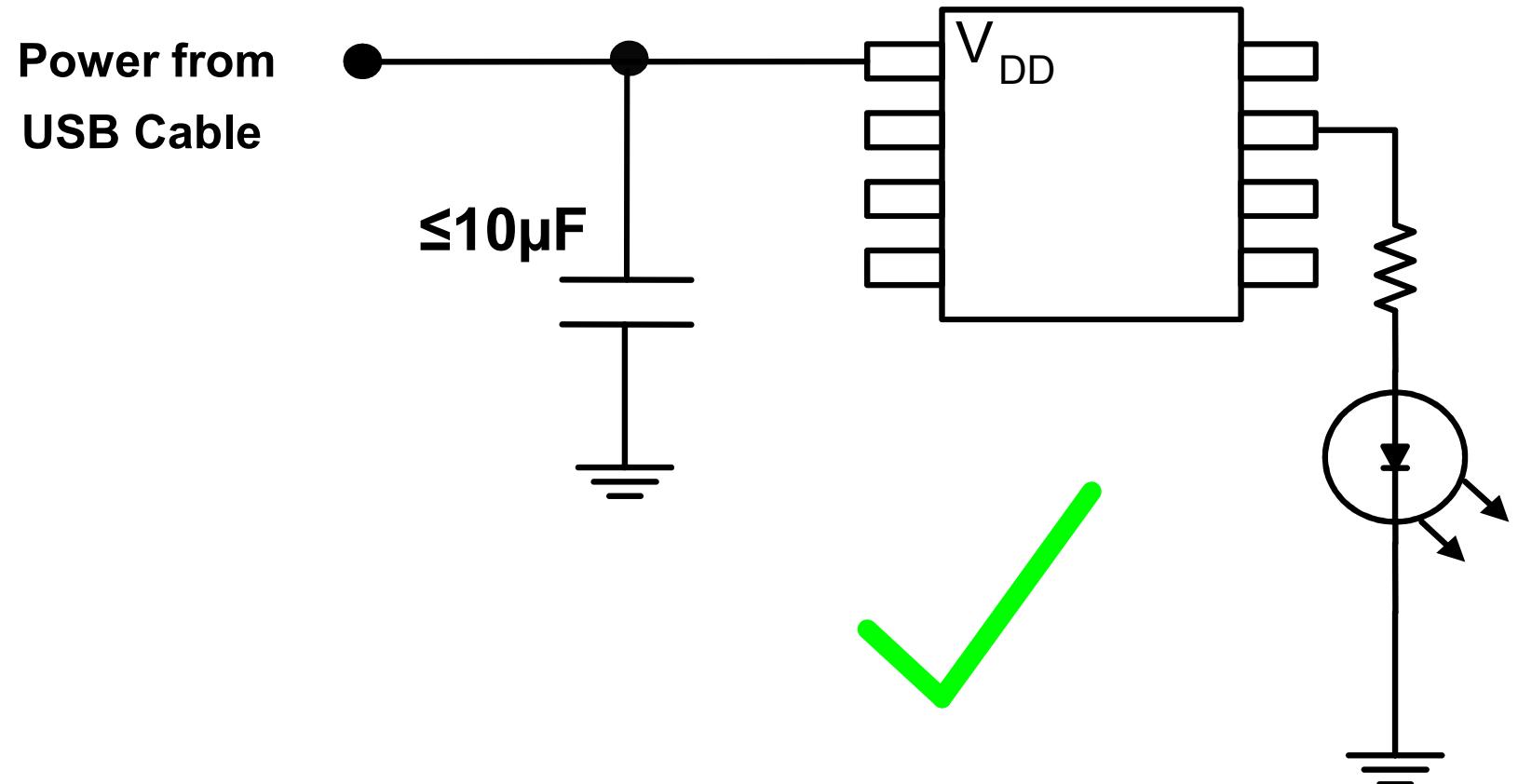
# Planeando la Alimentación

- Necesito Autoalimentación? -

- Deberá alimentarse al dispositivo externamente si:

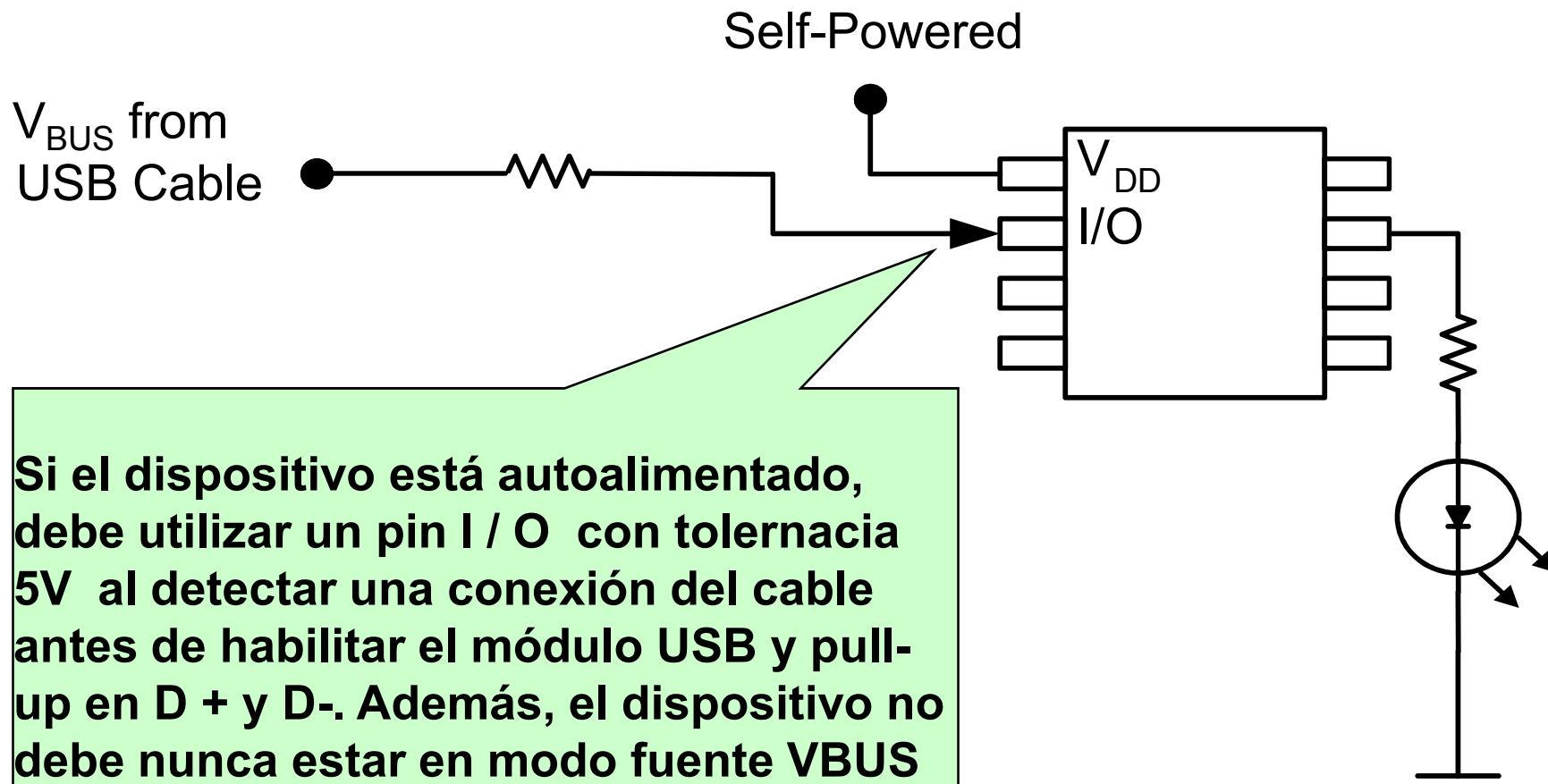
- este necesita funcionar sin estar conectado al BUS
- este necesita mas de 500 mA
- Este va a estar conectado a una PC con baterías, o hubs

# Planeando la Alimentación

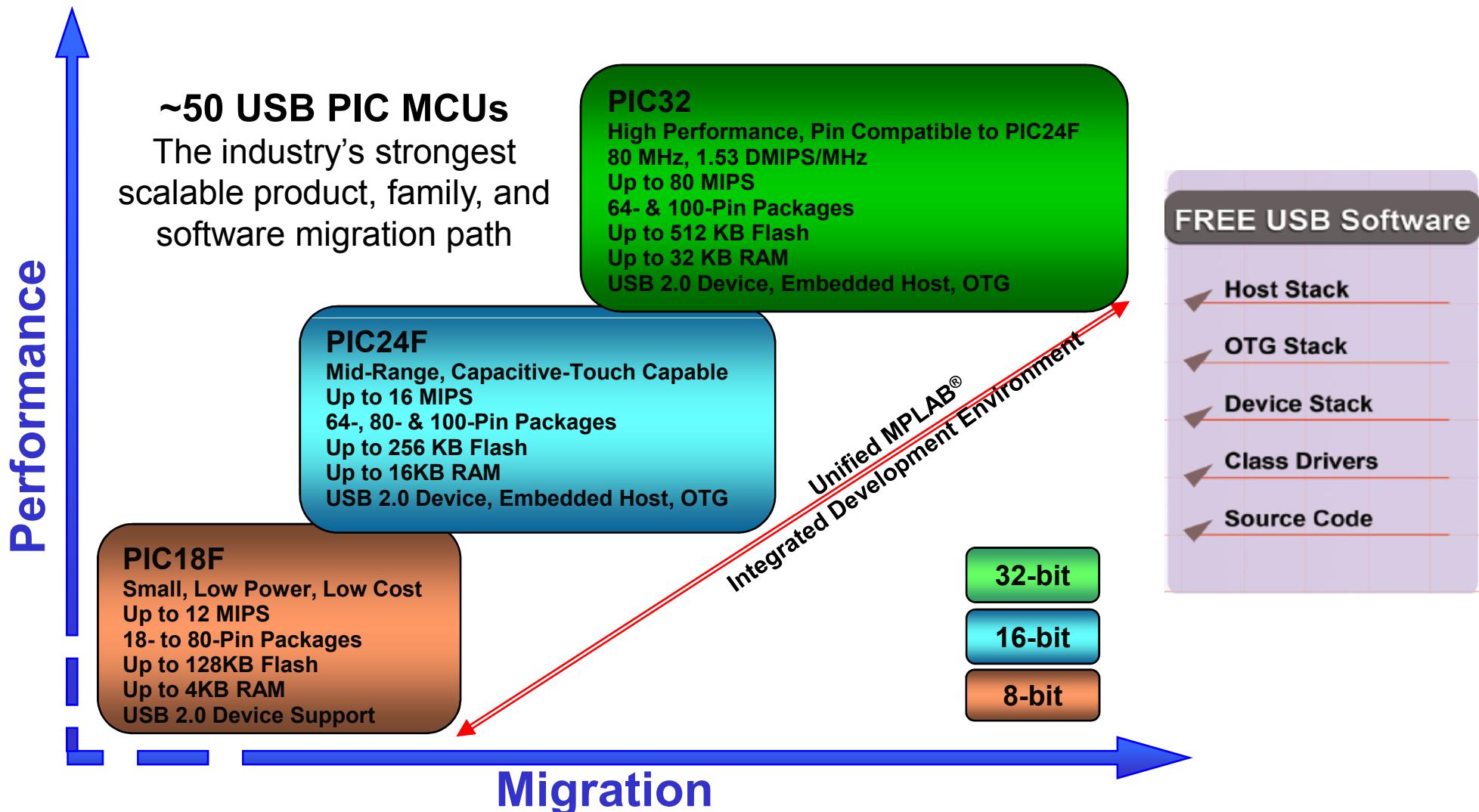


# Dispositivos alimentado externamente

## - Detectando la conexión al USB -



# Scalable USB PIC® MCU Portfolio



# Control USB with CCS compiler

- `usb_init()`
- `usb_task()`
- `usb_enumerated()`
- `usb_cdc_kbhit()`
- `usb_cdc_getc()`
- `usb_cdc_putc(c)`
- `usb_cdc_puts(*str)`
- Include files:
  - `<usb_desc_cdc.h>` & `<usb_cdc.h>`

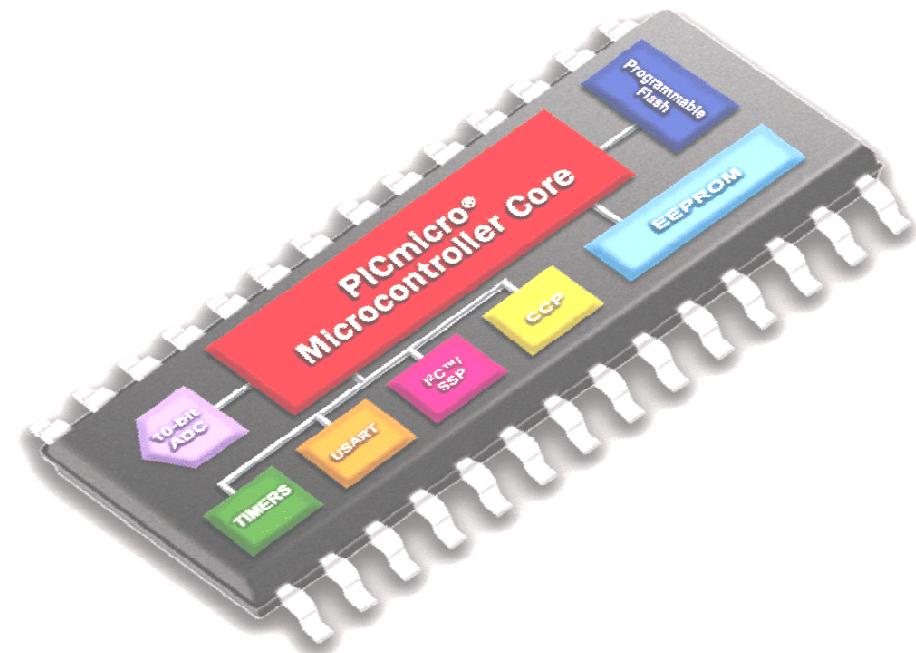


MICROCHIP

# PICPeriféricos Comunes

# PIC18 Common Peripherals

- Puertos I/O
- Analog Comparator
- Converso Analógico Digital
- Timers (0,1,2,3)
- CCP como PWM
- Addressable USART (AUSART)





MICROCHIP

# Puertos I/O

# Puertos I/O

- **Hasta 70 pines bidireccionales**
  - Los mismos estan multiplexados con Periféricos
- **Alta capacidad de Corriente**
  - 25mA en modo fuente/sumidero
- **Manipulación directa de cada bit en un solo ciclo**
- **Diodos de Protección ESD 4kV**
  - Basado sobre el modelo del cuerpo Humano
- **Despues del reset:**
  - Todos los I/O son entradas (Hi-Z)
  - Todos los pines con capacidades analógicas activadas por default

# Configurando los pines como digitales

- El método depende del dispositivo específico

- Puede escontrarse en el registro ADCON 1

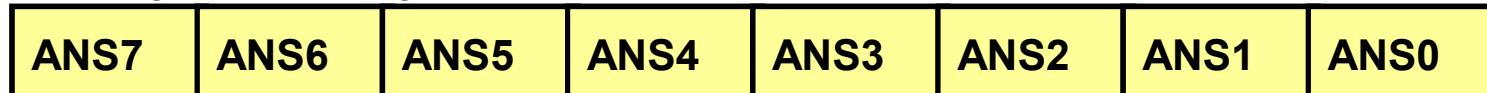
Registro de control (ADCON1)



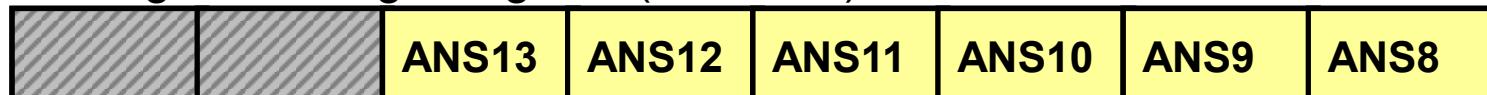
- Puede encontrarse en el registro ANSEL

- 1 = Analog; 0 = Digital

Analog Select Register (ANSEL)

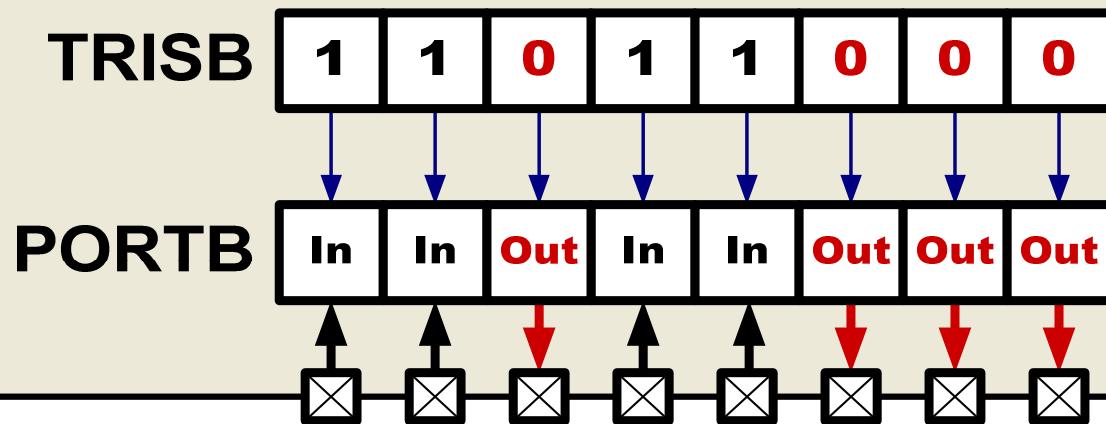


Analog Select High Register (ANSELH)



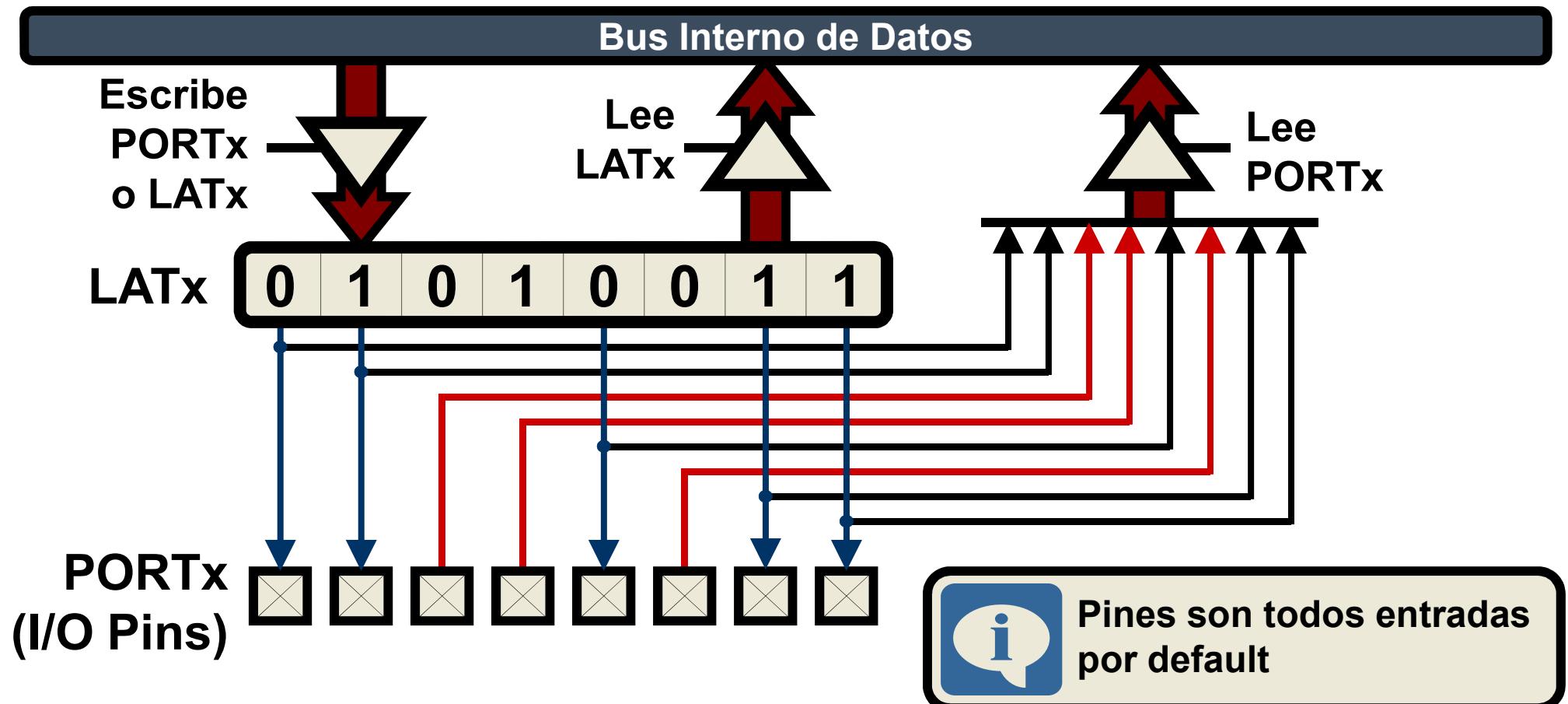
# Control de la dirección I/O

## I/O Pin Direction Control



- Bit n en el registro de control TRISx controla la dirección en el Bit n en PORTx
- 1 = Entrada, 0 = Salida

# Entrada Slida de Datos Registros TRIS, PORT y LAT



# PORTB Opciones

- Todos los pines del PORTB tienen resistores de PULL UP.
  - Un bit controla el pull up de todos los pines

## INTCON2 Register

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	-	TMR0IP	-	RBIP
bit 7							

bit 7      **RBPU:** PORTB Pull-up Enable bit  
1 = All PORTB Pull-ups are disabled  
0 = PORTB Pull-ups are enabled by individual port latch values

bit 0      **RBIP:** RB Port Change Interrupt Priority bit  
1 = High Priority  
0 = Low Priority

# Control I/O con CCS

- **output\_a (*value*)**
- **output\_bit (*pin, value*)**
- **output\_high (*pin*)**
- **output\_low (*pin*)**
- **output\_toggle(*pin*)**
- **variable = input (*pin*)**
- **variable = input\_a()**

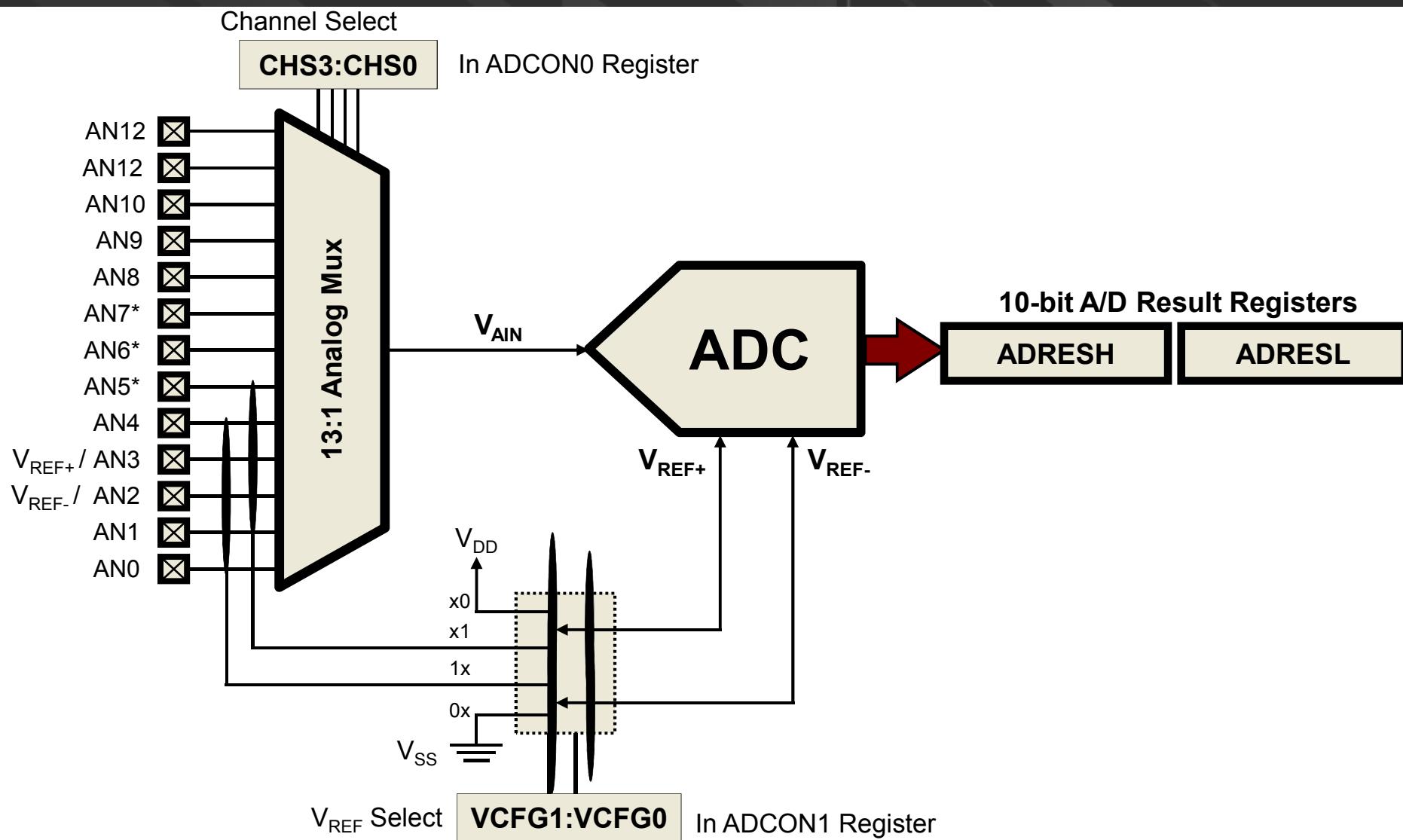


MICROCHIP

# Conversor A/D:

A/D de 10-bit

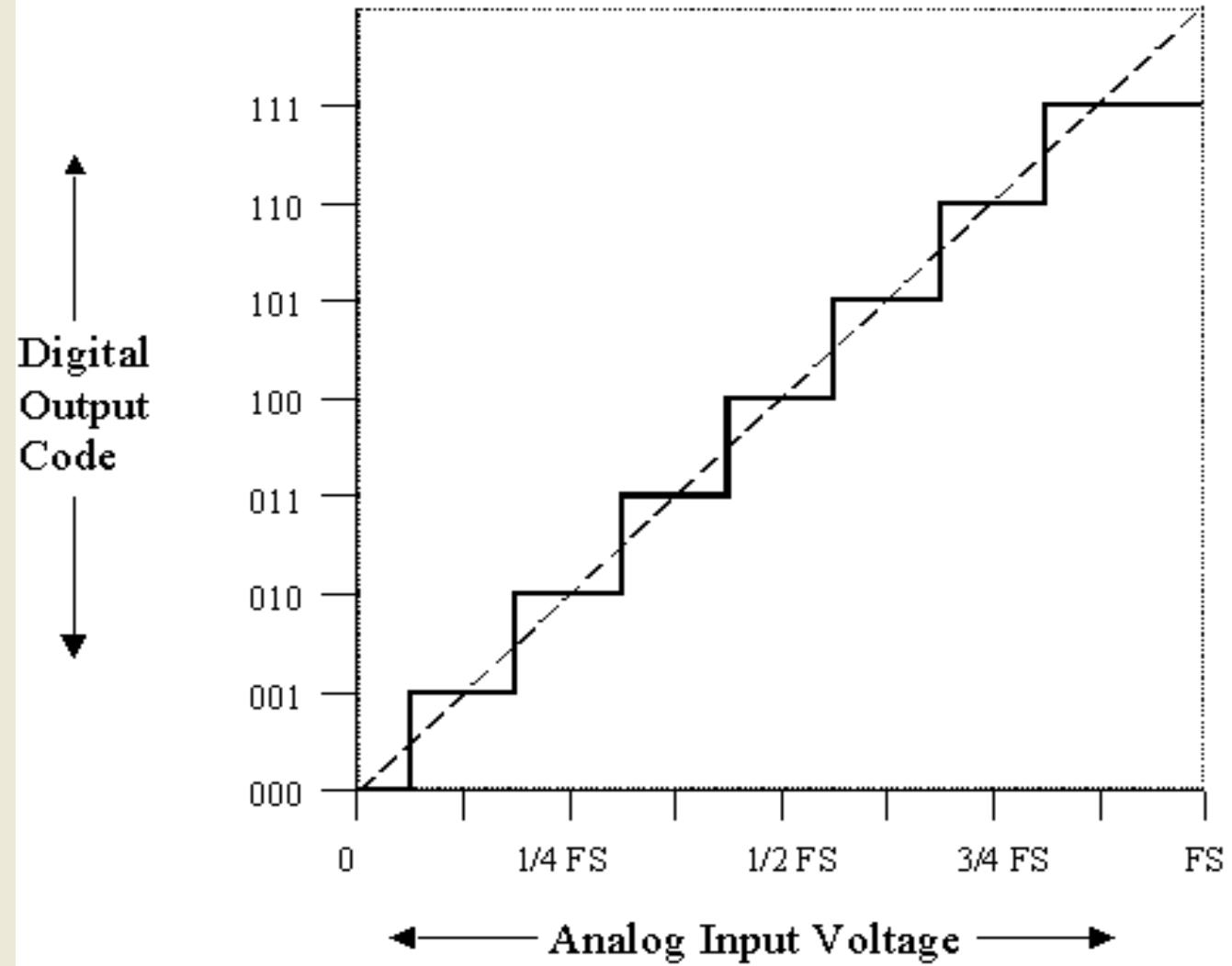
# Conensor A/D de 10-bit



\*AN5-AN7 not available on 28-pin devices

# ADC 10-bit Escalando resultados

- $FS$  = Fondo de escala
- $N$ -bits,  $2^N$  Códigos posibles
- “Tamaño del Bit” o  $Paso = V_{FS}/2^N$



# ADC 10-bit Escalando resultados

## ■ Ejemplo:

- 10-bits ADC =>  $2^{10} = 1024$  codigos
- Escala de valores completa 0 – 5.12 V
- Cual es el tamaño de un bit?

## ■ Respuesta:

- Tamaño del Bit =  $5.12 / 1024 = 5 \text{ mV}$

# ADC Interpretando los resultados

## ■ Ejemplo:

- 8-bits ADC =>  $2^8$
- Escala completa 0 – 5.0 V
  - 1 Bit = 19.53 mV
- Calculo del valor si ADRESH = 0xCA

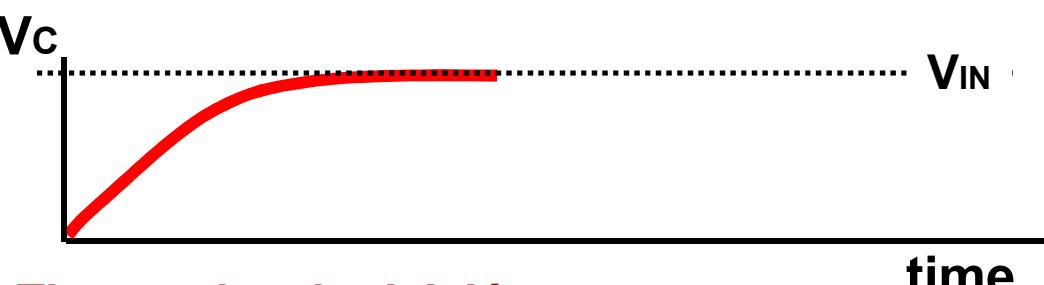
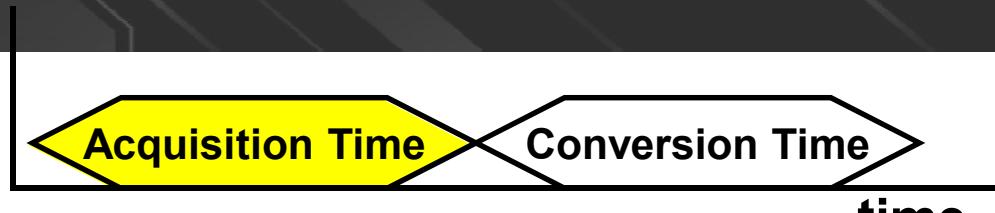
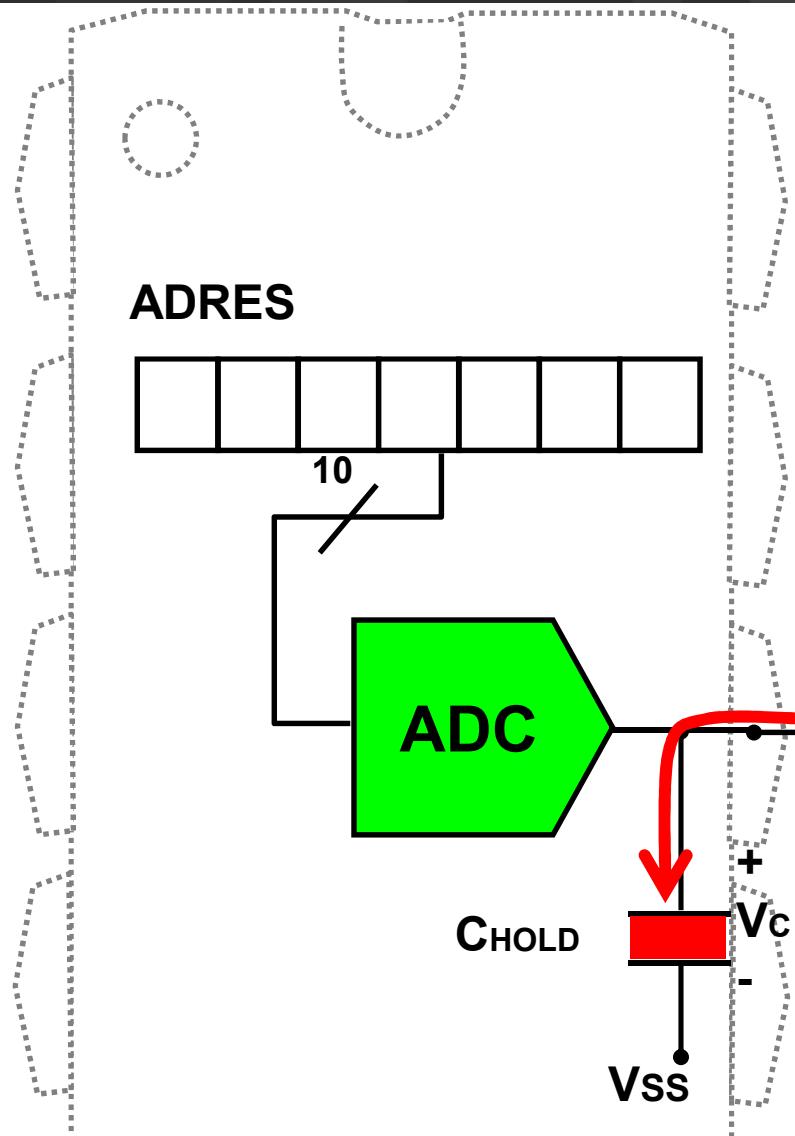
**Result =  $V_{IN}$  / Tamaño del Bit**

**$V_{IN} = Result * Tamaño del Bit$**

$$V_{IN} = 202 * 19.53 \text{ mV}$$

$$V_{IN} = 3.9456 \text{ V}$$

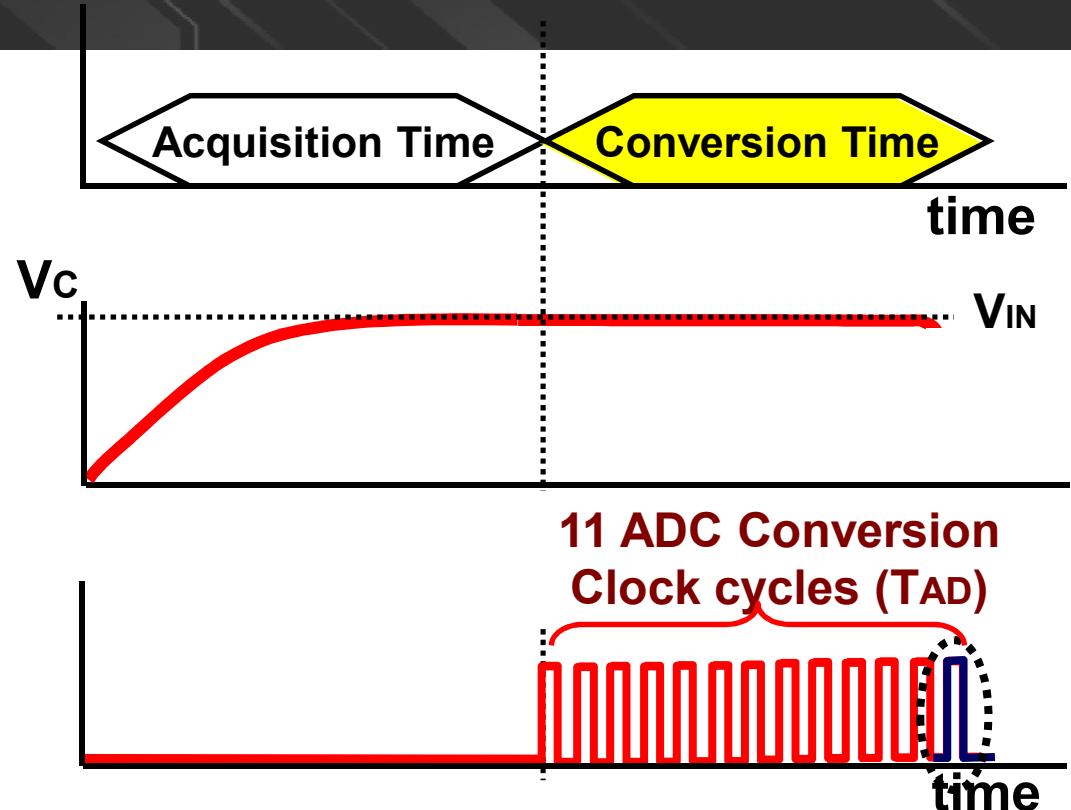
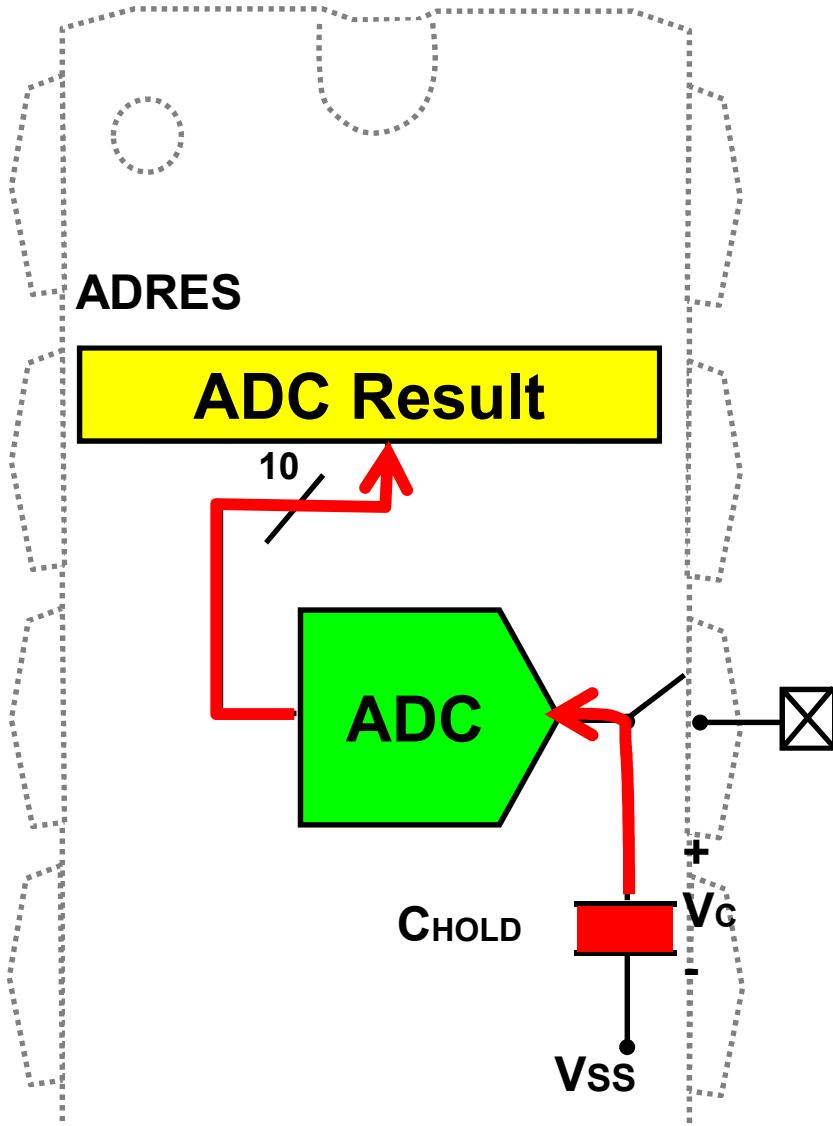
# Adquisición de la señal



Tiempo de adquisición  
determinado por la  
capacidad  $C_{HOLD}$  y la  
Impedancia de fuente

Este tiempo le permite cargarse  
completamente a  $C_{HOLD}$

# ADC Conversion



# Conversor A/D de 10-bit

## Configuración

### ADCON0 Register

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

bit 7    bit 0

#### bit 5-2 CHS3:CHS0

Analog Channel Select Bits

- 0000 = Channel 0 (AN0)
- 0001 = Channel 1 (AN1)
- 0010 = Channel 2 (AN2)
- 0011 = Channel 3 (AN3)
- 0100 = Channel 4 (AN4)
- 0101 = Channel 5 (AN5)
- 0110 = Channel 6 (AN6)
- 0111 = Channel 7 (AN7)
- 1000 = Channel 8 (AN8)
- 1001 = Channel 9 (AN9)
- 1010 = Channel 10 (AN10)
- 1011 = Channel 11 (AN11)
- 1100 = Channel 12 (AN12)

#### bit 1

#### GO/DONE: A/D Conversion Status bit

- 1 = A/D Conversion in progress
- 0 = A/D Idle

#### bit 0

#### ADON: A/D On bit

- 1 = A/D Converter module is enabled
- 0 = A/D Converter module is disabled

# Conversor A/D de 10-bit

## Configuración

### ADCON1 Register

U-0	U-0	R/W-0	R/W-0	R/W-0*	R/W*	R/W*	R/W*
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

bit 7    bit 0

bit 5   **VCFG1:** Voltage Reference ( $V_{REF-}$ ) Configuration bit

1 =  $V_{REF-}$  (AN2)

0 =  $V_{SS}$

bit 4   **VCFG0:** Voltage Reference ( $V_{REF+}$ ) Configuration bit

1 =  $V_{REF+}$  (AN3)

0 =  $V_{DD}$

# Conversor A/D de 10-bit

## Configuración

### ADCON1 Register

U-0	U-0	R/W-0	R/W-0	R/W-0*	R/W*	R/W*	R/W*
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

#### bit 3-0 PCFG3:PCFG0

A/D Port Configuration Control bits

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog Input  
D = Digital I/O

# Conversor A/D de 10-bit

## Configuración

### ADCON2 Register

U-0	U-0	R/W-0	R/W-0	R/W-0*	R/W*	R/W*	R/W*
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7				bit 0			

**NOTE: This parameter determines the value of  $T_{AD}$ : the conversion time per bit.**

bit 2-0 **ADCS2:ADCS0**: A/D Conversion Clock Selection bits

111 =  $F_{RC}$  (Clock derived from A/D RC oscillator)

110 =  $F_{OSC}/64$

101 =  $F_{OSC}/16$

100 =  $F_{OSC}/4$

011 =  $F_{RC}$  (Clock derived from A/D RC oscillator)

010 =  $F_{OSC}/32$

001 =  $F_{OSC}/8$

000 =  $F_{OSC}/2$

# Conversor A/D de 10-bit

## Configuración

### ■ Seleccionando el clock del A/D

A/D Clock Source ( $T_{AD}$ )		Maximum Device Frequency
Operation	ADCS2:ADCS0	PIC18F4520
$2 T_{OSC}$	000	2.86 MHz
$4 T_{OSC}$	100	5.71 MHz
$8 T_{OSC}$	001	11.43 MHz
$16 T_{OSC}$	101	22.86 MHz
$32 T_{OSC}$	010	40.0 MHz
$64 T_{OSC}$	110	40.0 MHz
RC	x11	*

At  $F_{OSC} = 4$  MHz:

$$T_{OSC} = 1/F_{OSC} \\ = .25 \mu s$$

$$T_{AD} = 4 \cdot T_{OSC} \\ = 1 \mu s$$

\* The Internal RC has a typical  $T_{AD}$  time of  $1.2 \mu s$

**Minimum  $T_{AD} = 0.7 \mu s$**

See data sheet for official specification.

# Conversor A/D de 10-bit

## Configuración

### ■ Calculando el tiempo de adquisición

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$

$T_{AMP}$  = Amplifier Settling Time

=  $T_{amp\ internal}$  +  $T_{amp\ external}$

$T_C$  = Charging Time

=  $-(C_{HOLD}) \cdot (R_{IC} + R_{SS} + R_S) \cdot \ln(1/2047) \mu s$

$T_{COFF}$  = Temperature Coefficient

= ( $\text{Temp} - 25\text{ C}$ )  $\cdot (0.02 \mu s / \text{C})$  for  $\text{Temp} > 25\text{ C}$

= 0 for  $\text{Temp} < 25\text{ C}$

# Conversor A/D de 10-bit

## Configuración

### ■ Calculando el tiempo del amplificados

$$T_{AMP} = T_{AMP\ INT} + T_{AMP\ EXT}$$

- $T_{AMP\ INT}$ : Tiempo interno del amplificador. Chequear el data sheet para el dispositivo específico. Para el PIC18F4520,  $T_{AMP\ INT}$  es 0.2  $\mu$ s.
- $T_{AMP\ EXT}$ : Es el tiempo del amplificador externo. Este es muy importante cuando el circuito externo acondiciona la señal. Para la placa de microrochip PICDEM2 Plus es un POTENCIOMETRO, y este número es 0  $\mu$ s.

$$T_{AMP} = 0.2 \mu s$$

# Conversor A/D de 10-bit

## Configuración

### ■ Calculando el Tiempo de Carga

$$T_C = -(C_{HOLD}) \cdot (R_{IC} + R_{SS} + R_S) \cdot \ln(1/2047) \mu s$$

$C_{HOLD}$  = 25pF

$R_{IC}$  = 1kΩ

$R_{SS}$  = 2kΩ

$R_S$  = 2.5kΩ MAX

Asumiendo que la impedancia máxima es de  
2.5kΩ :

$$T_C = 1.05 \mu s$$

# Conversor A/D de 10-bit

## Configuración

### ■ Calculando el coeficiente de temperatura

$$T_{COFF} = (\text{Temp} - 25 \text{ C}) \cdot (0.02\mu\text{s}/ \text{C})$$

**Temp** = Temperatura de Operación

Valido para Temp. > 25 C.

$T_{COFF} = 0$  para Temp. < 25 C

Usando la máxima temperatura de operación de 85 C para un dispositivo industrial:

$$T_{COFF} = 1.2\mu\text{s}$$

# Conversor A/D de 10-bit

## Configuración

### ■ Calculando el Tiempo de adquisición

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$

Para  $R_s = 2.5k \Omega$

Temp. = 85 °C

$T_{AMP} = 0.2 \mu s$

$$T_{ACQ} = 0.2 + 1 + 1.2 = 2.4 \mu s$$

# Conversor A/D de 10-bit

## Configuración

### ADCON2 Register

U-0	U-0	R/W-0	R/W-0	R/W-0*	R/W*	R/W*	R/W*
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7				bit 0			

bit 5-3 ACQT2:ACQT0: A/D Acquisition Time Selection bits

$$111 = 20 T_{AD}$$

$$110 = 16 T_{AD}$$

$$101 = 12 T_{AD}$$

$$100 = 8 T_{AD}$$

$$011 = 6 T_{AD}$$

$$010 = 4 T_{AD}$$

$$001 = 2 T_{AD}$$

$$000 = 0 T_{AD}$$

Given:

$$T_{ACQ} = 2.4 \mu s$$

$$T_{AD} = 1 \mu s$$

Then:

$$ACQT2:0 = 010 = 4.0 \mu s$$

# Conversor A/D de 10-bit

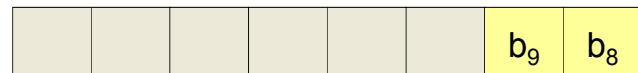
## Configuración

### ADCON2 Register

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7				bit 0			

bit 7    **ADFM:** Selecciona el resultado del formato del A/D  
1 = Justificado a la derecha

**ADRESH**

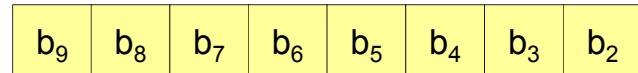


**ADRESL**

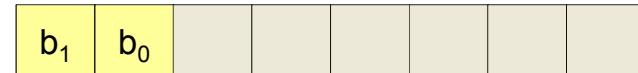


0 = Justificado a la izquierda

**ADRESH**



**ADRESL**



# Control ADC con CCS

- `setup_adc(mode)`
- `setup_adc_ports(value)`
- `set_adc_channel(channel)`
- `read_adc(mode)`
- `adc_done()`
- `#device adc=xx (xx=8,10,12 o16 bits)`



MICROCHIP

# Timers

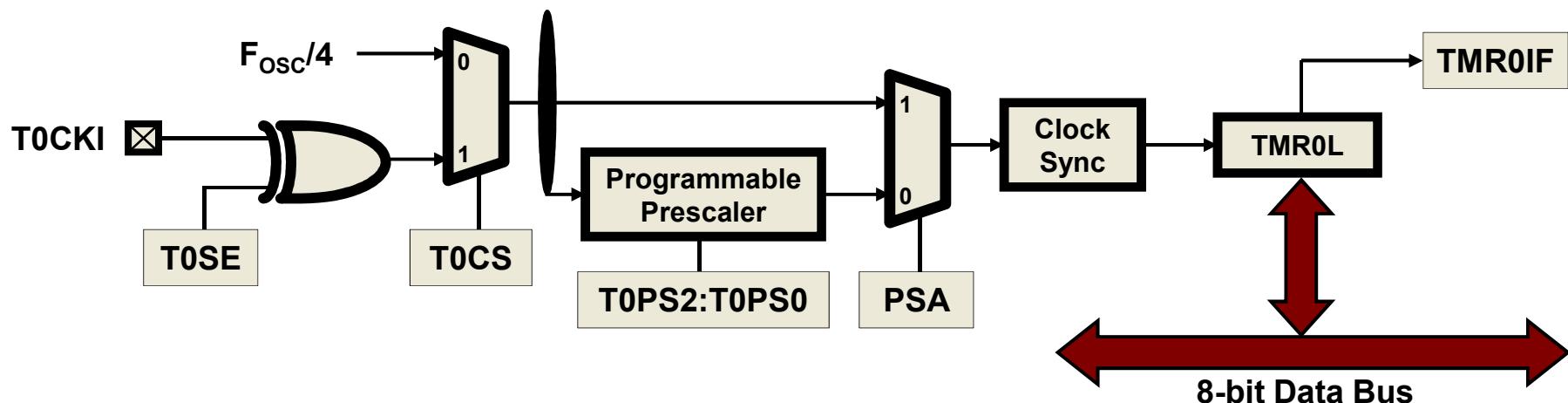
# Timer Comparativa

	TIMER 0	TIMERS 1 & 3	TIMERS 2 & 4
SIZE OF REGISTER	8-bits or 16-bits	16-bits	8-bits
CLOCK SOURCE (Internal)	Fosc/4	Fosc/4	Fosc/4
CLOCK SOURCE (External )	T0CKI pin	T13CKI pin or Timer 1 oscillator (T1OSC)	None
CLOCK SCALING AVAILABLE (Resolution)	Prescaler 8-bits (1:2→1:256)	Prescaler 2-bits (1:1, 1:2, 1:4, 1:8)	Prescaler (1:1,1:4,1:16) Postscaler (1:1→1:16)
INTERRUPT EVENT	On overflow FFh→00h	On overflow FFFFh→0000h	TMR REG matches PR2
CAN WAKE PIC FROM SLEEP?	NO	YES	NO

# Timer0

## Modo compatible con PIC16

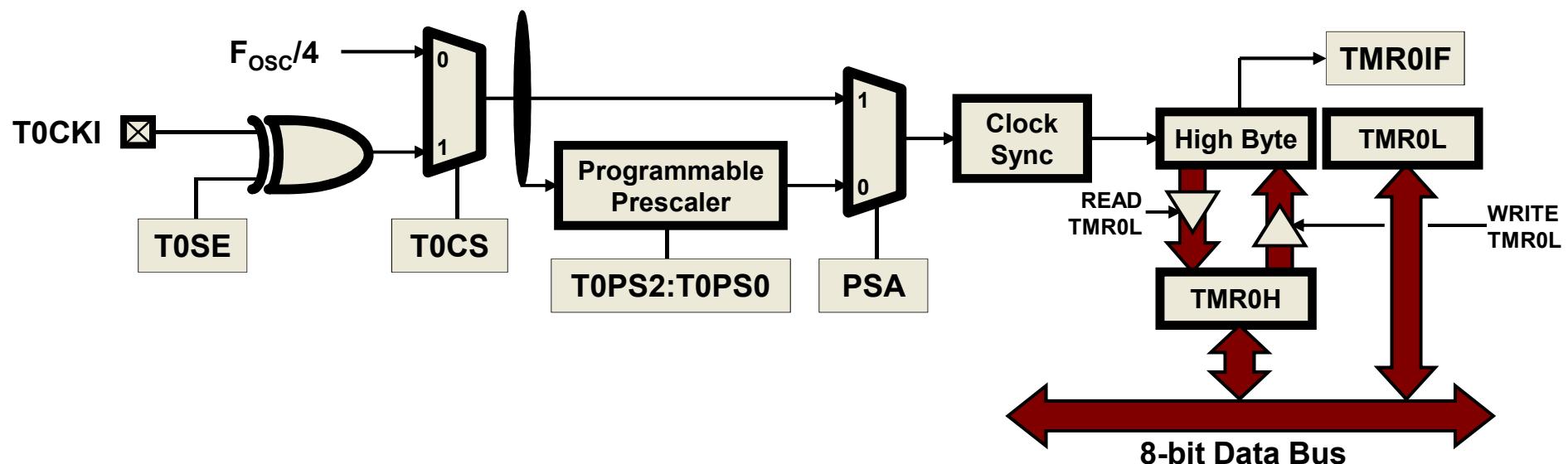
- 8-bit Timer/Counter
- Prescaler Programable de 8 bits
- Fuente de clock externa o interna
- Interrupción sobre el overflow de FF a 00



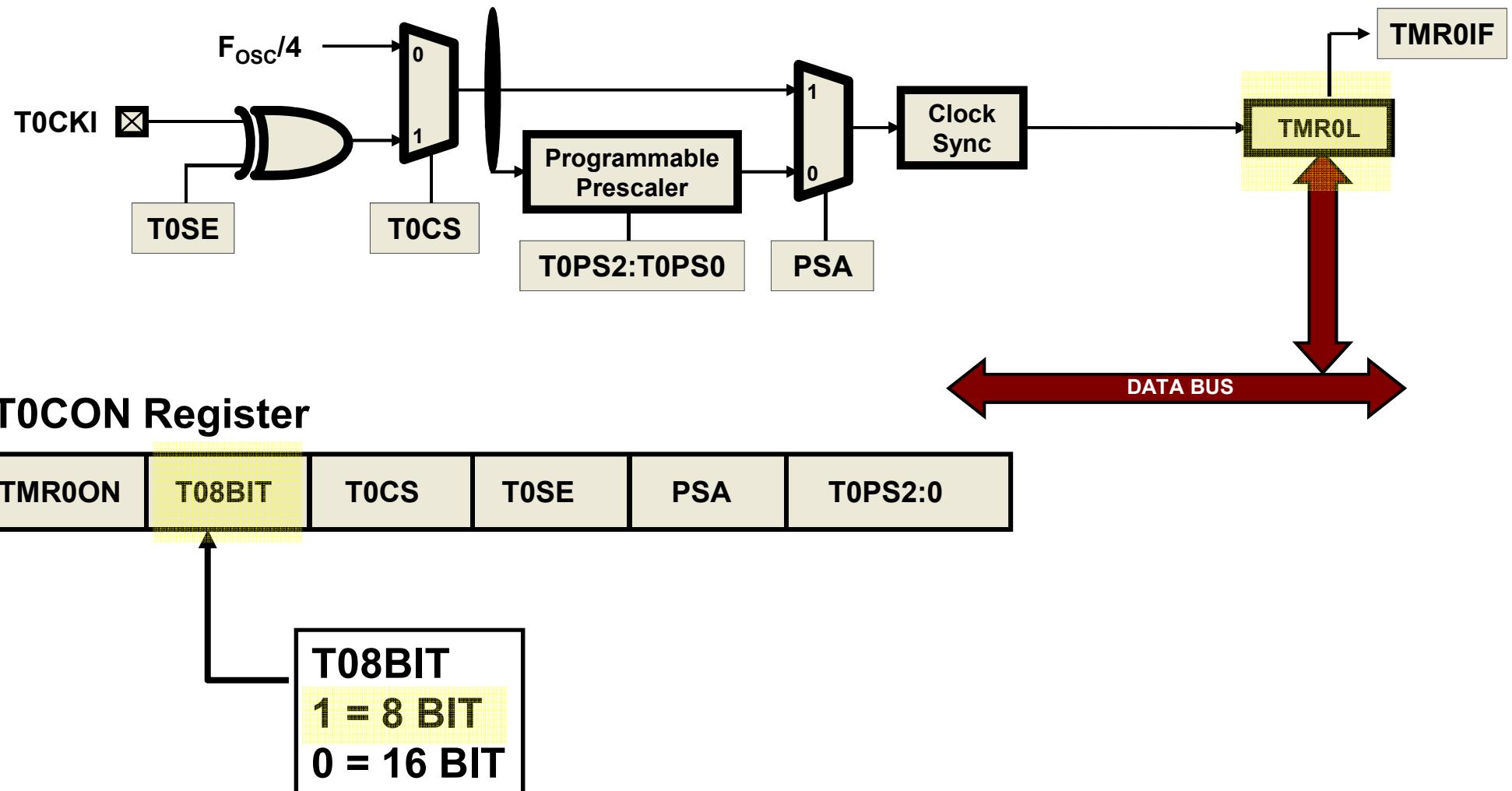
# Timer0

## Modo 16-bit

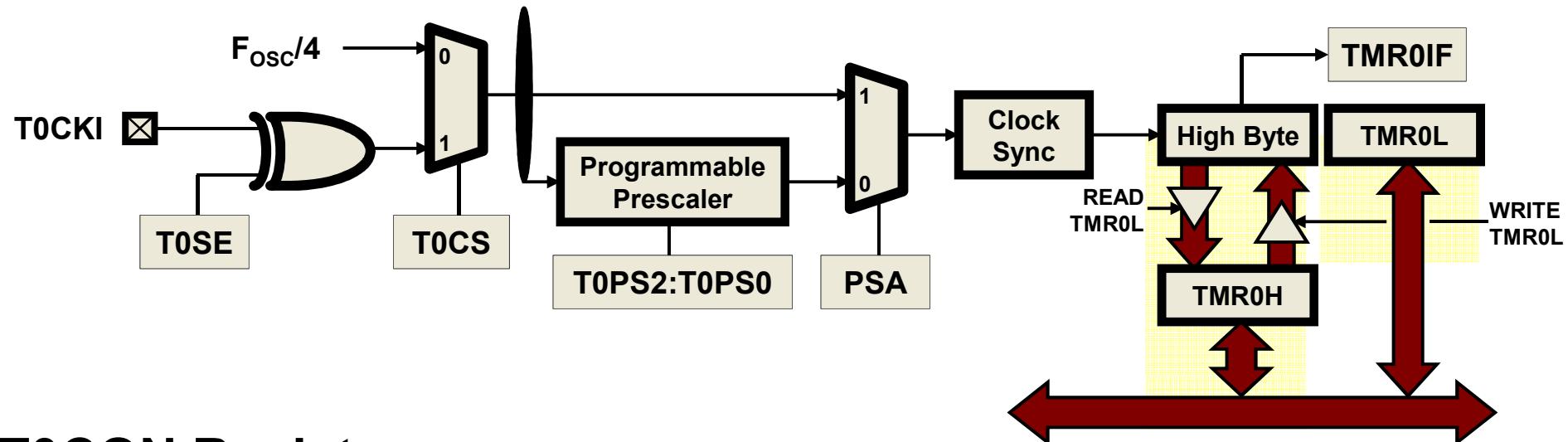
- 16-bit Timer / Counter
- Modo Lectura/Escritura en 16-bit
- Interrupción sobre overflow de FFFF a 0000
- Same basic features as compatibility mode



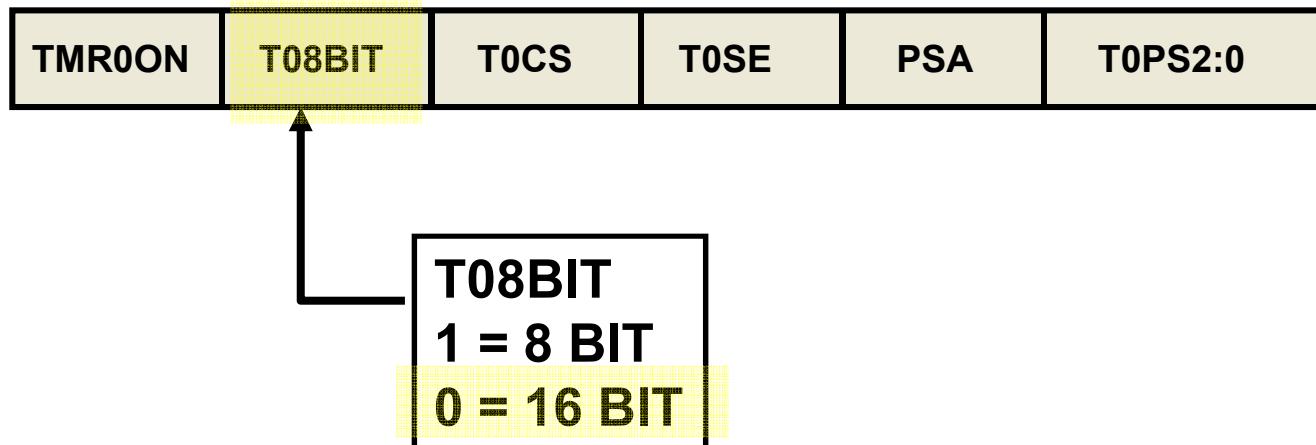
# Timer0 Operation



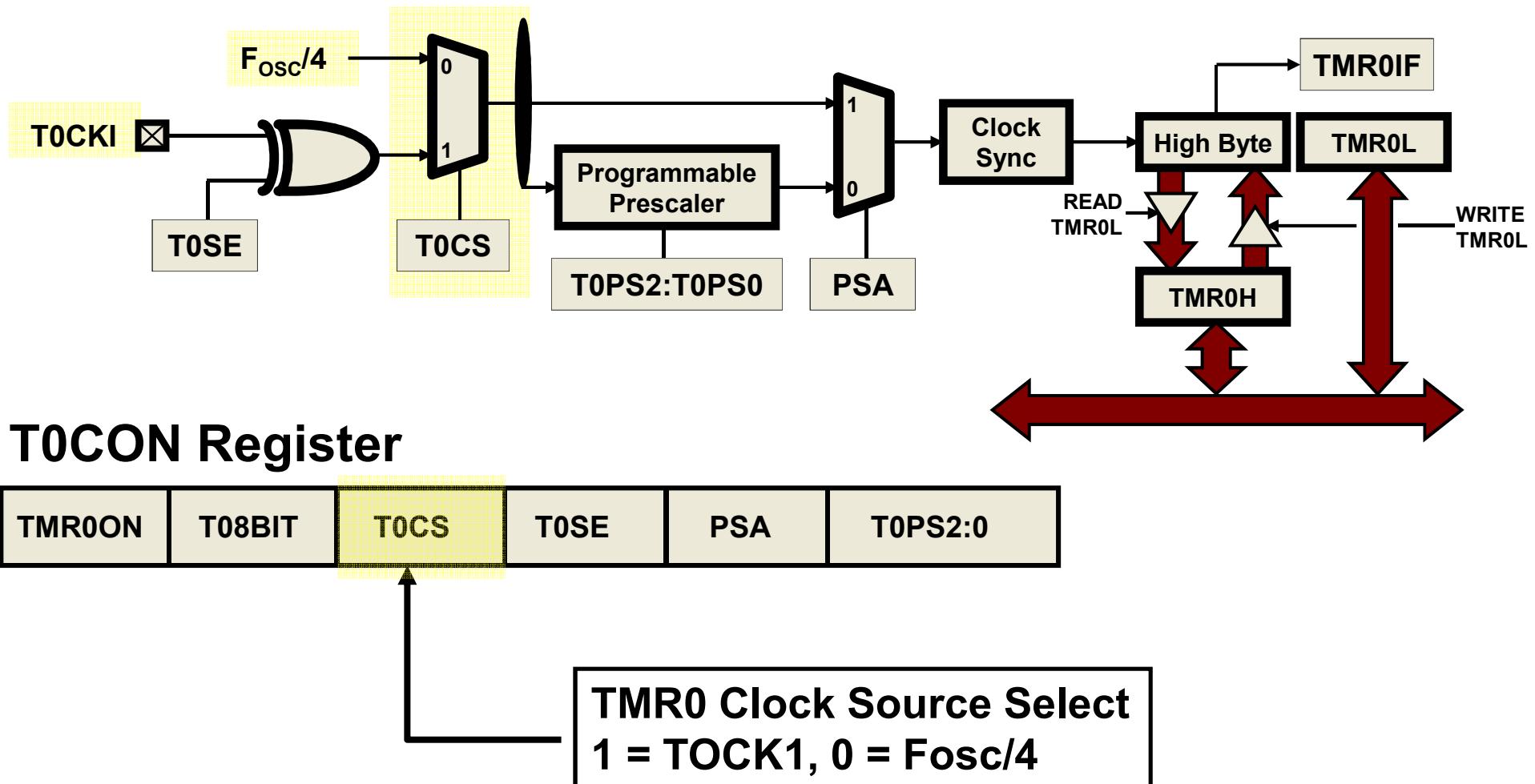
# Timer0 Operation



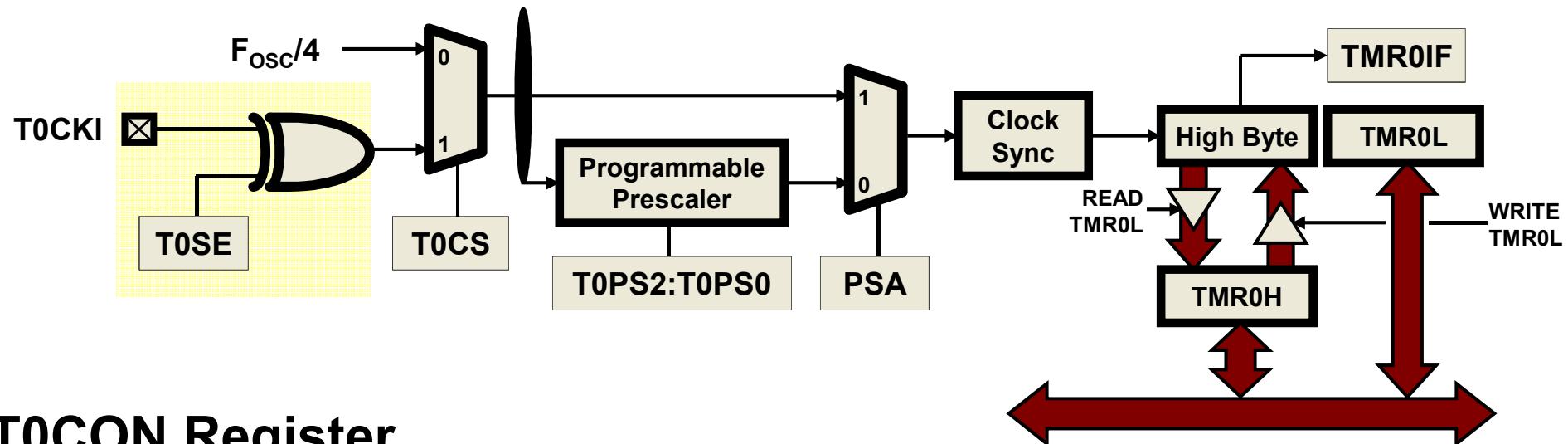
## T0CON Register



# Timer0 Operation



# Timer0 Operation

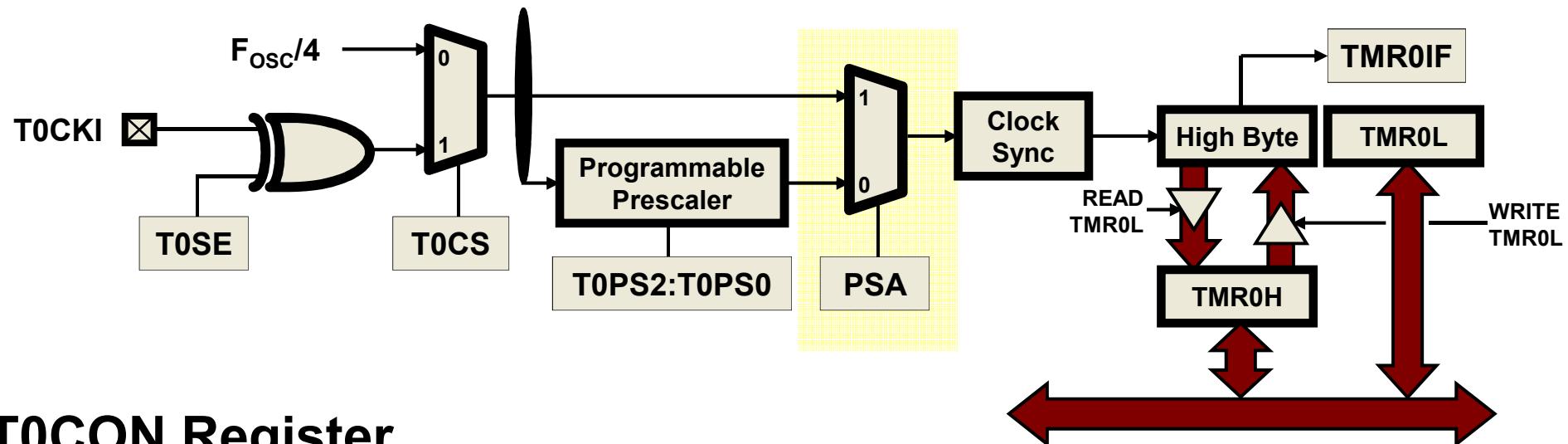


T0CON Register

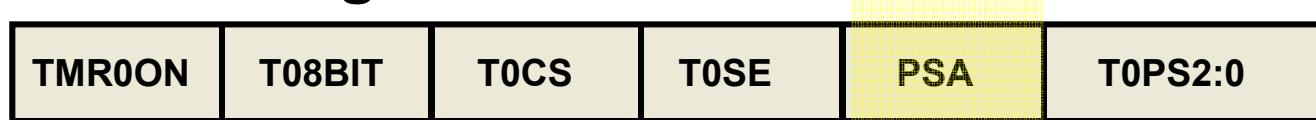
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2:0
--------	--------	------	------	-----	---------

Source Edge Select  
1 = increment TMR0 on rising edge  
0 = increment TMR0 on falling edge

# Timer0 Operation

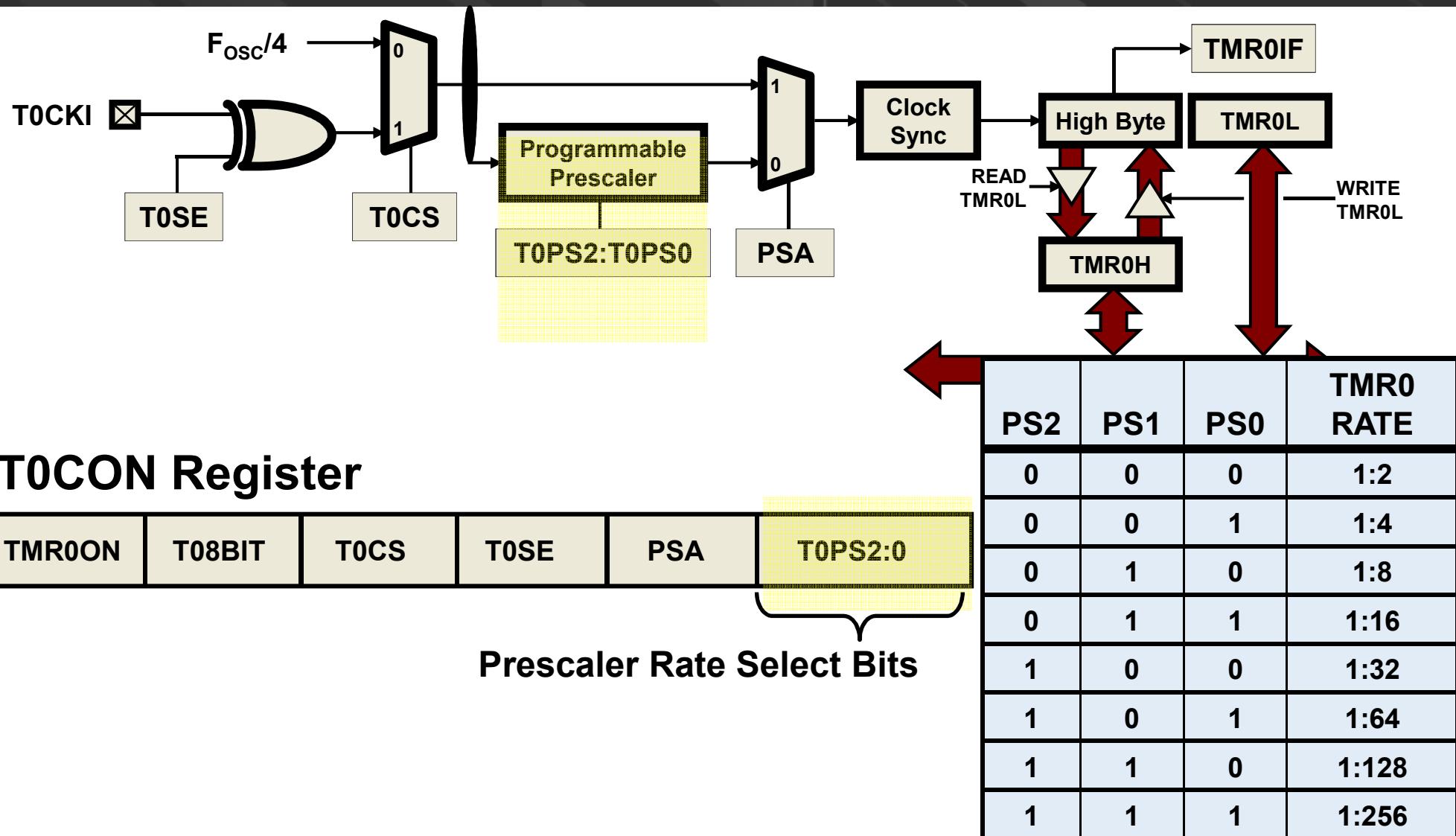


## T0CON Register

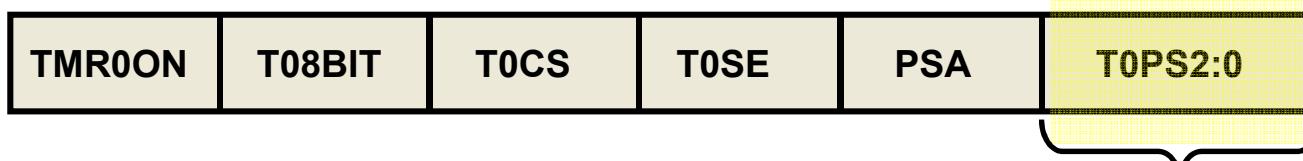


**Prescaler Assignment**  
1= prescaler NOT assigned  
0= prescaler IS assigned

# Timer0 Operation

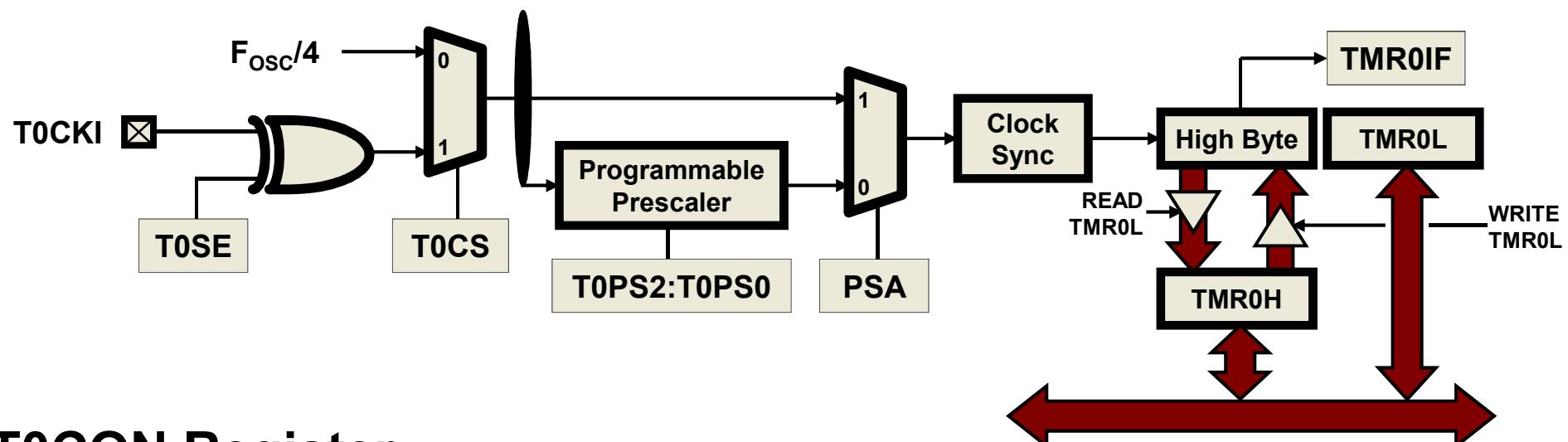


## T0CON Register

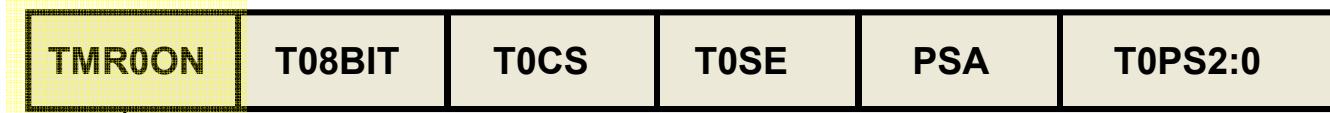


Prescaler Rate Select Bits

# Timer0 Operation



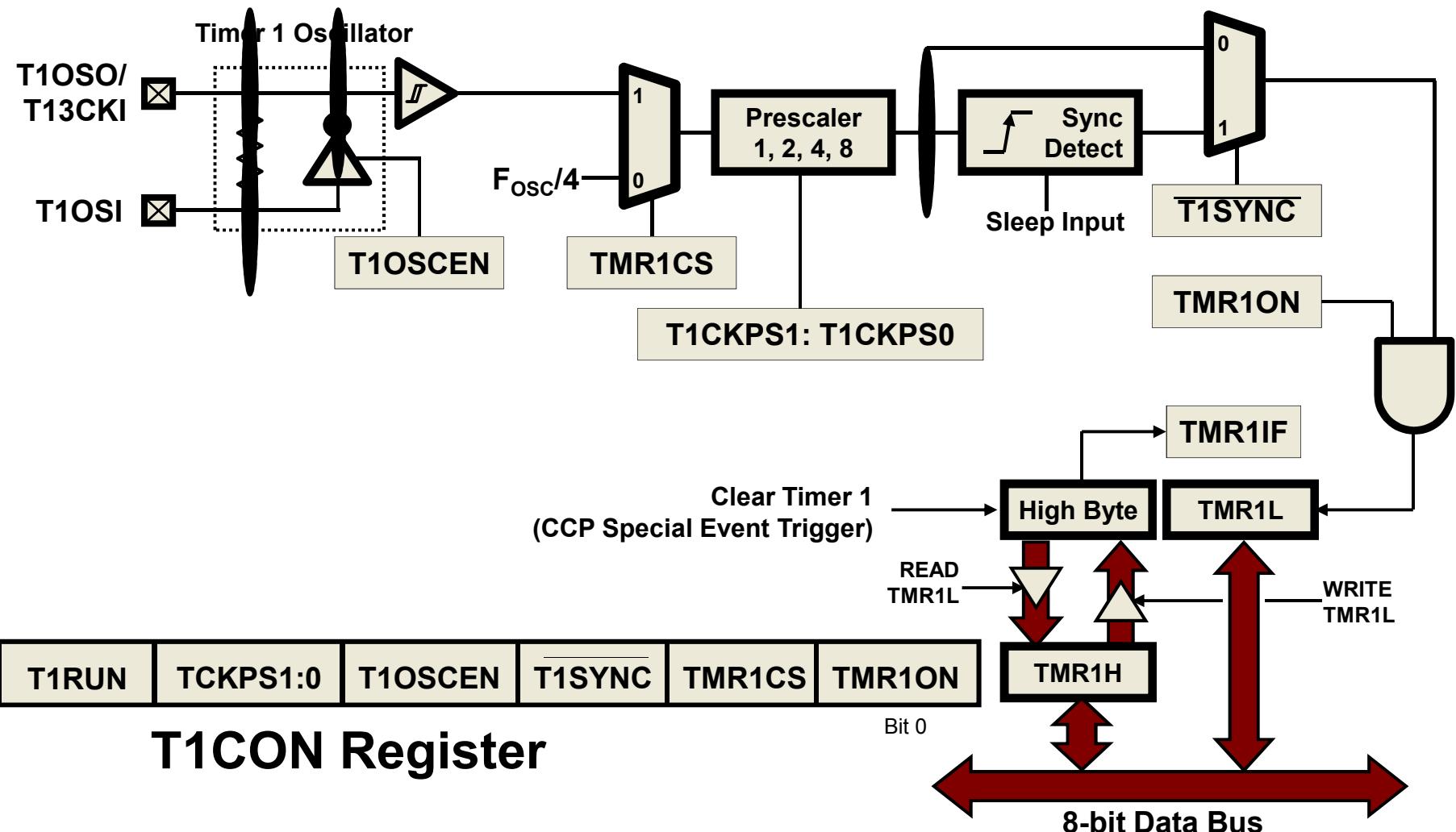
## T0CON Register



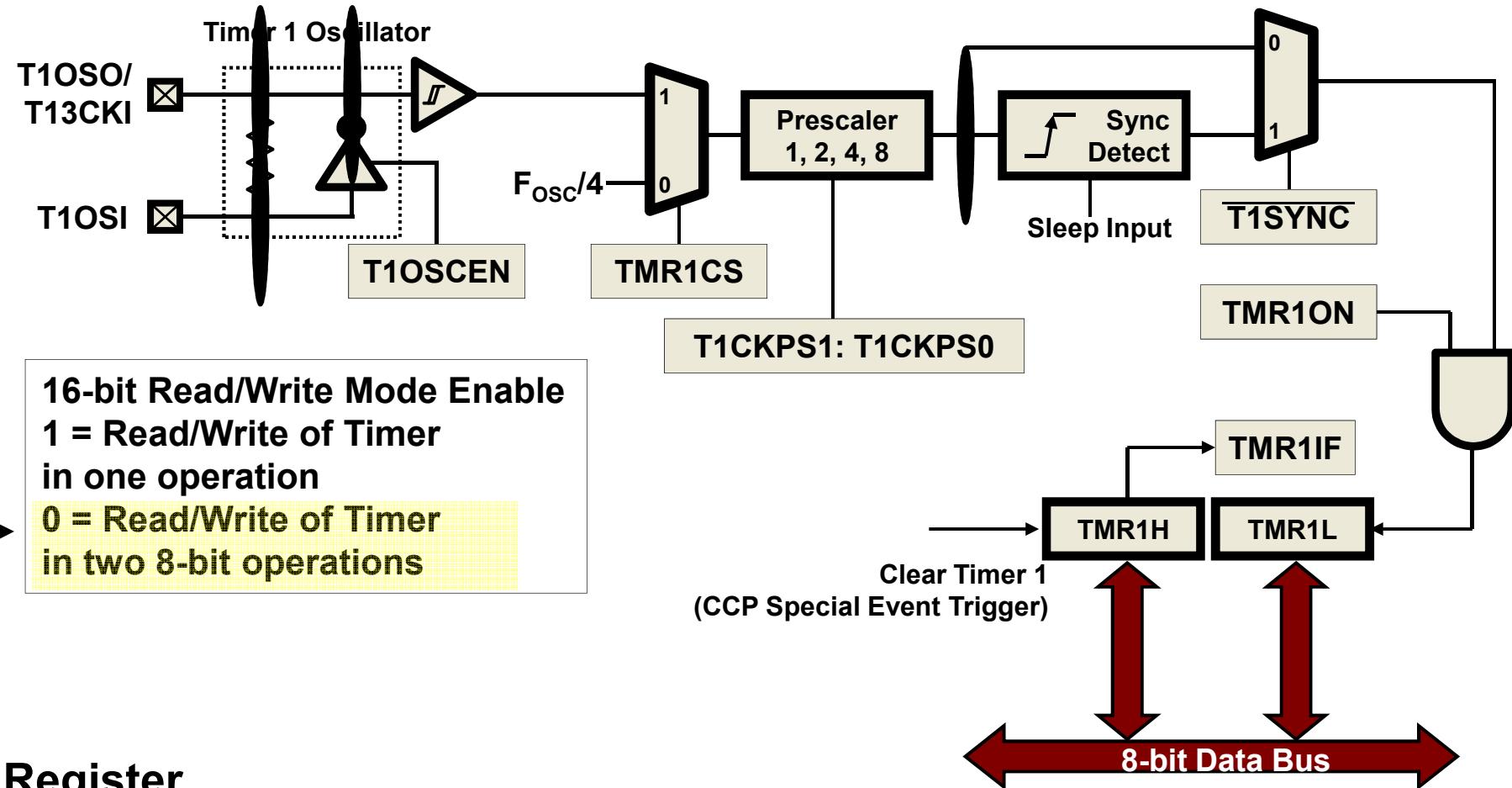
# Timer1 and Timer3

- **16-bit Timer / Counter**
- **Timer, Synchronous Counter or Asynchronous Counter**
- **Can operate from separate external crystal**
- **Two read/write 8-bit registers**
- **$\div 1$ ,  $\div 2$ ,  $\div 4$ , or  $\div 8$  Prescaler**
- **Interrupt on overflow from FFFFh to 0000h**

# Timer1 and Timer3



# Timer1 and Timer3



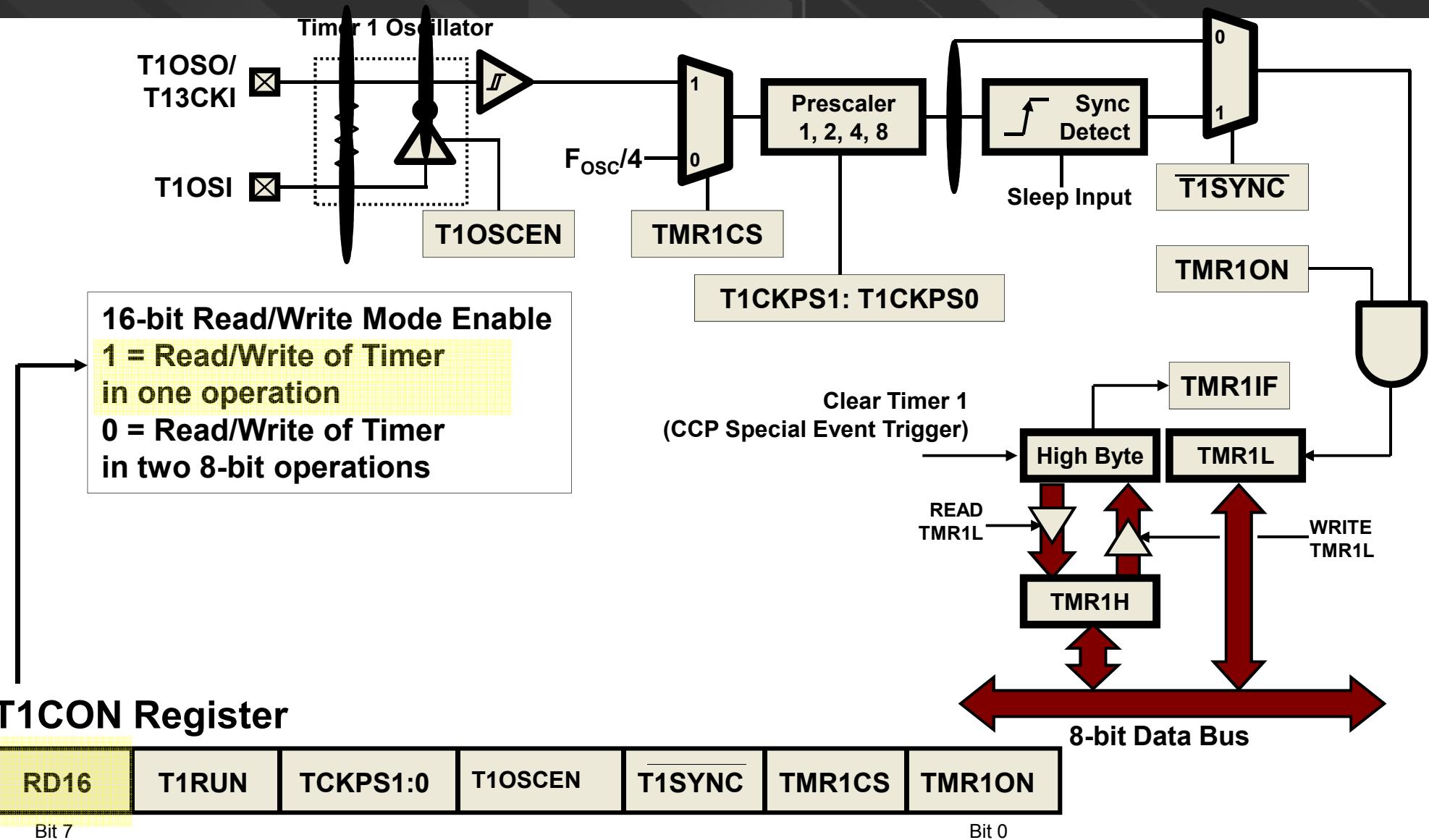
**T1CON Register**

RD16	T1RUN	TCKPS1:0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
------	-------	----------	---------	--------	--------	--------

Bit 7

Bit 0

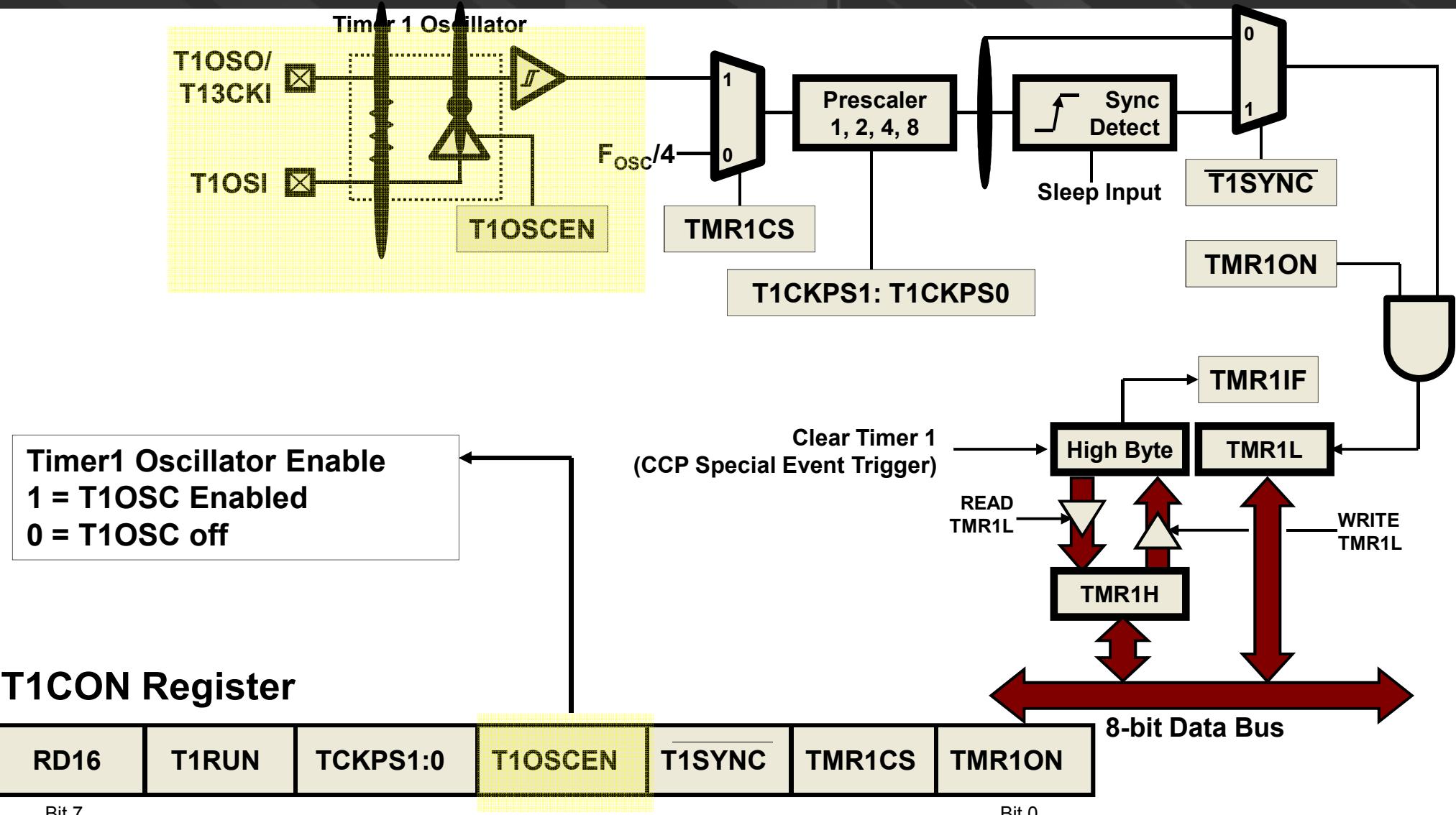
# Timer1 and Timer3



**T1CON Register**

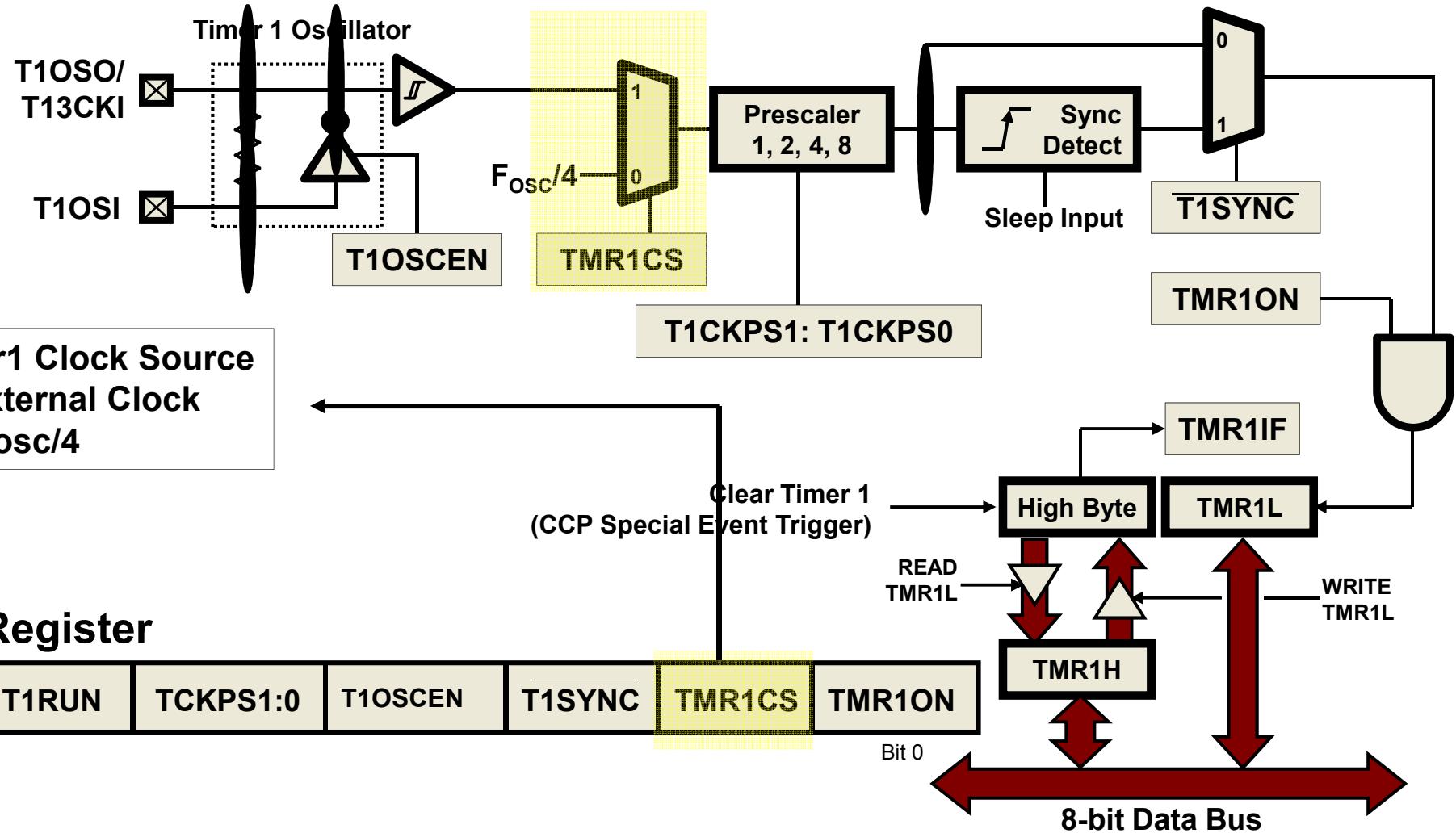
RD16	T1RUN	TCKPS1:0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
Bit 7					Bit 0	

# Timer1 and Timer3

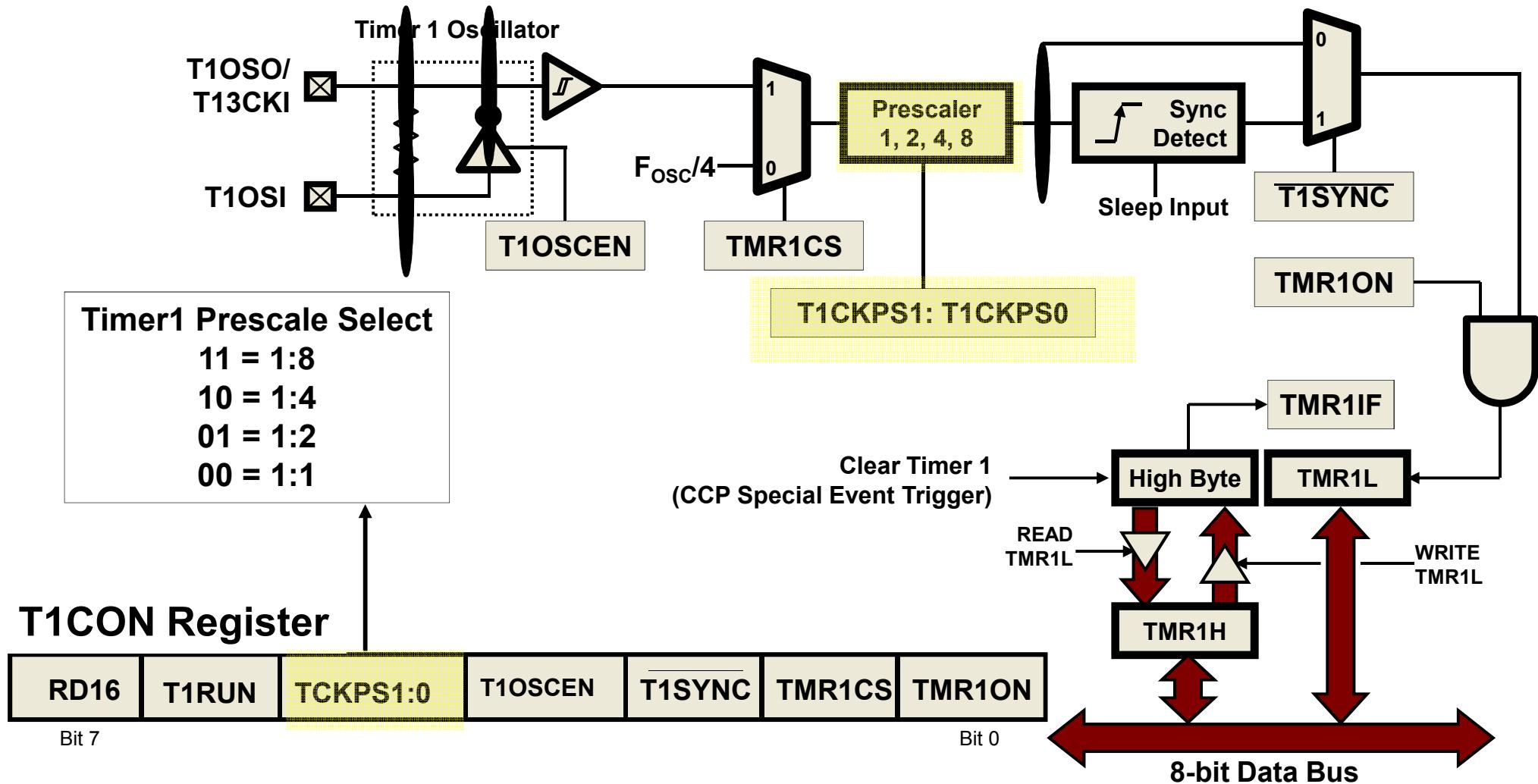


**T1CON Register**

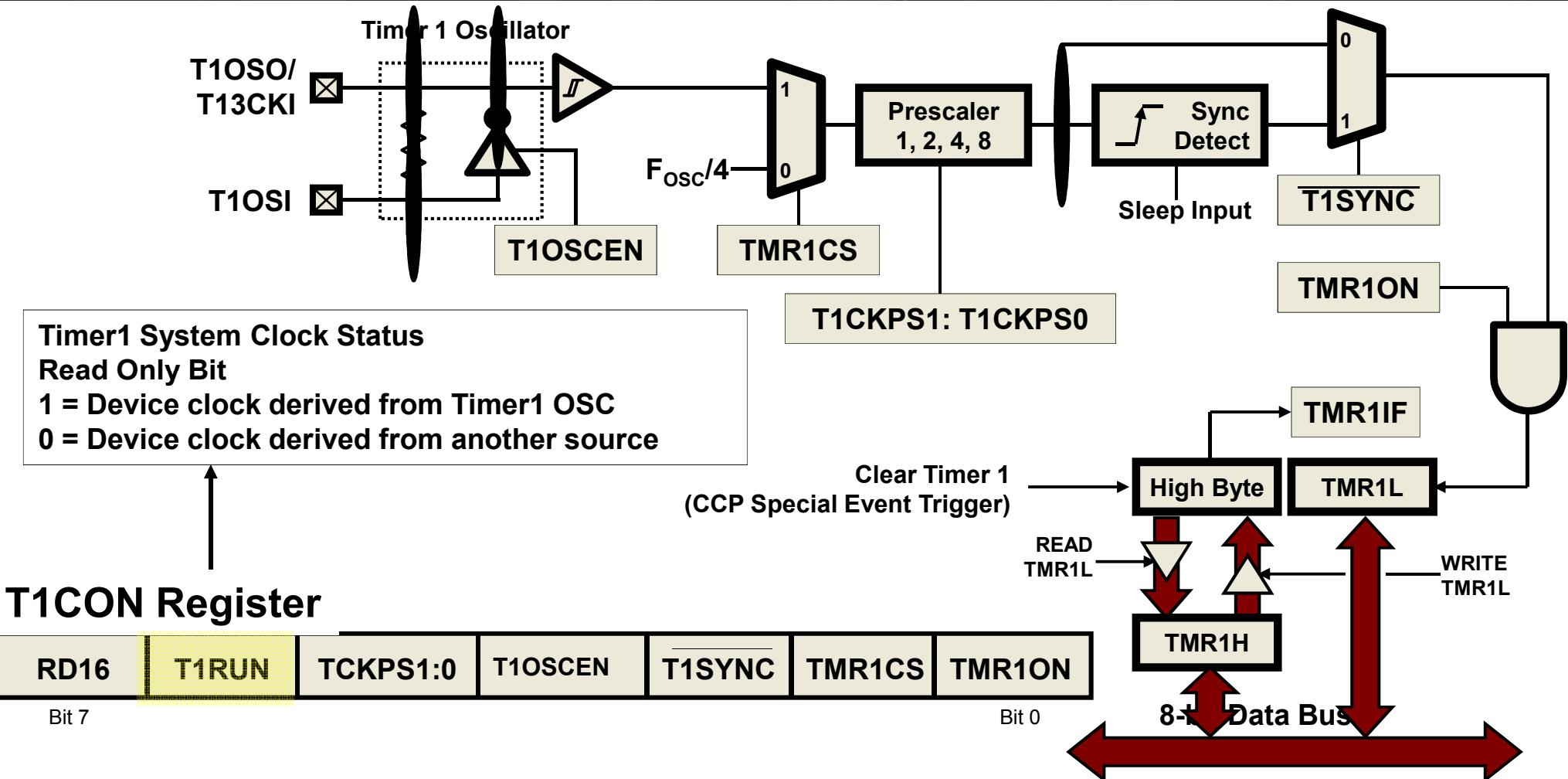
# Timer1 and Timer3



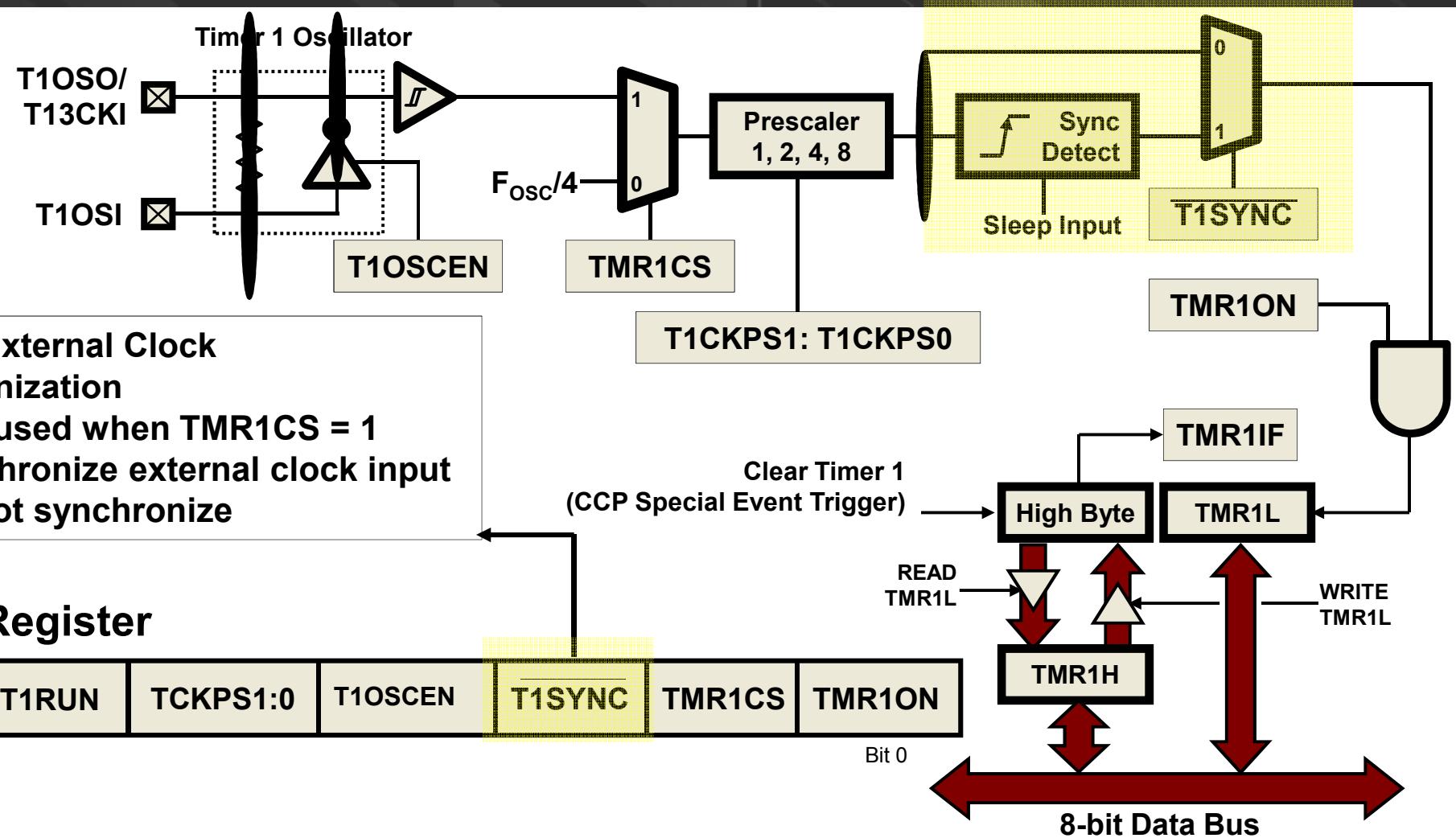
# Timer1 and Timer3



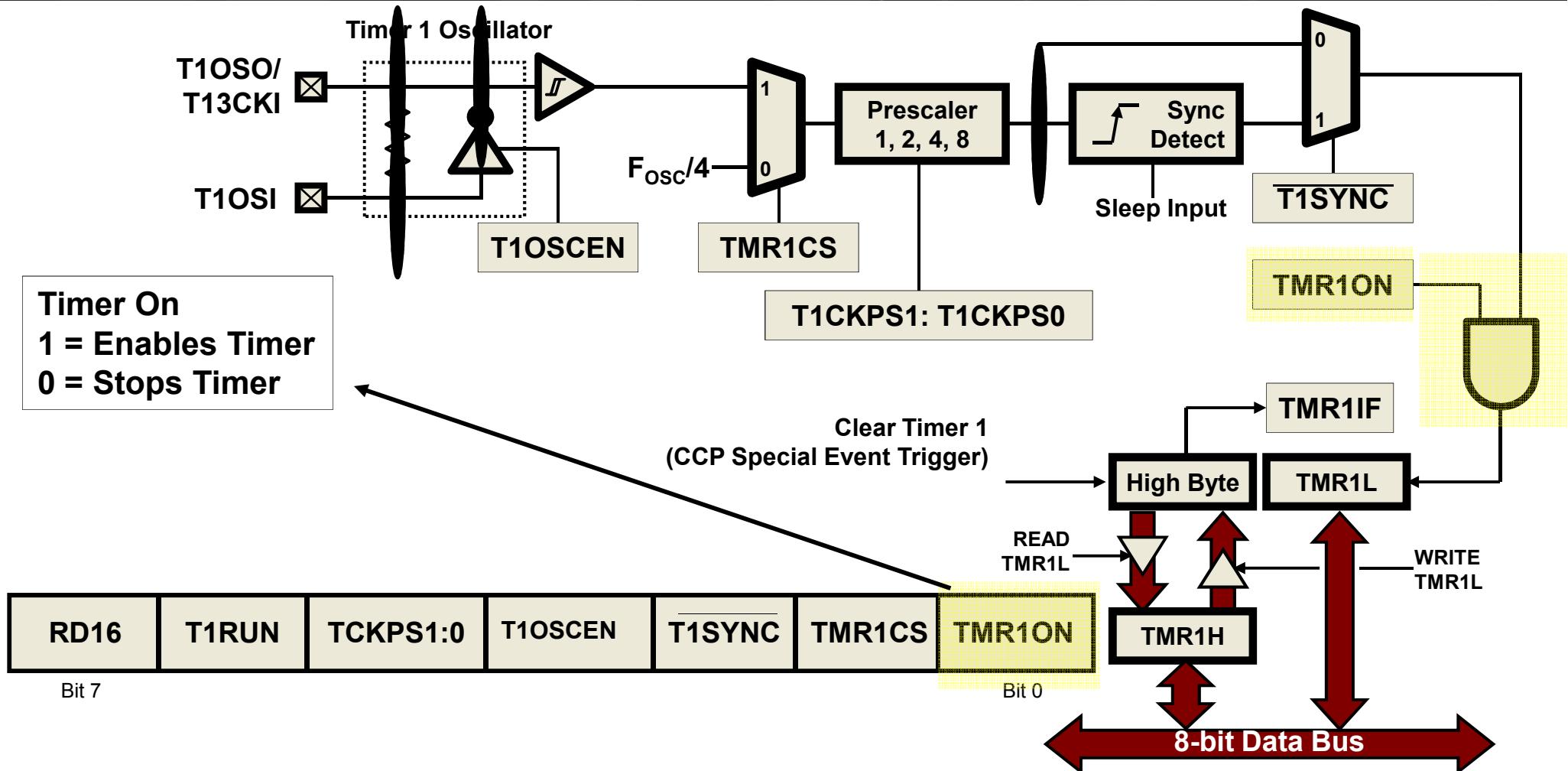
# Timer1 and Timer3



# Timer1 and Timer3



# Timer1 and Timer3



# Timer2

- 8-bit Timer with prescaler and postscaler
- Used as PWM time base
- TMR2 is readable & writable
- TMR2 resets on match with PR2
- Match with PR2 generates interrupt
- Used as baud clock for MSSP (SPI™)

# Timer2

## T2CON Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0

bit 7 bit 0

00 = 1:1  
 01 = 1:4  
 1x = 1:16

**TMR2 Postscale Value Selection Bits**

**T2OUTPS3:T2OUTPS0**

0000 = 1:1  
 0001 = 1:2  
 ...  
 1111 = 1:16

**TMR2 Prescale Value Selection Bits**

**T2CKPS1:T2CKPS0**

$F_{osc}/4$  → **1:1, 1:4, 1:16 Prescaler**

**1:1 to 1:16 Postscaler**

**TMR2IF**

**Timer 2 Output (To PWM or MSSP)**

**TMR2/PR2 Match**

**PR2**

**8-bit Data Bus**

# Control Timers con CCS

- `setup_timer_x(mode)`
- `set_timerx(value).`
- `value=get_timerx.`



MICROCHIP

# PIC18 Interrupts

# Interrupción

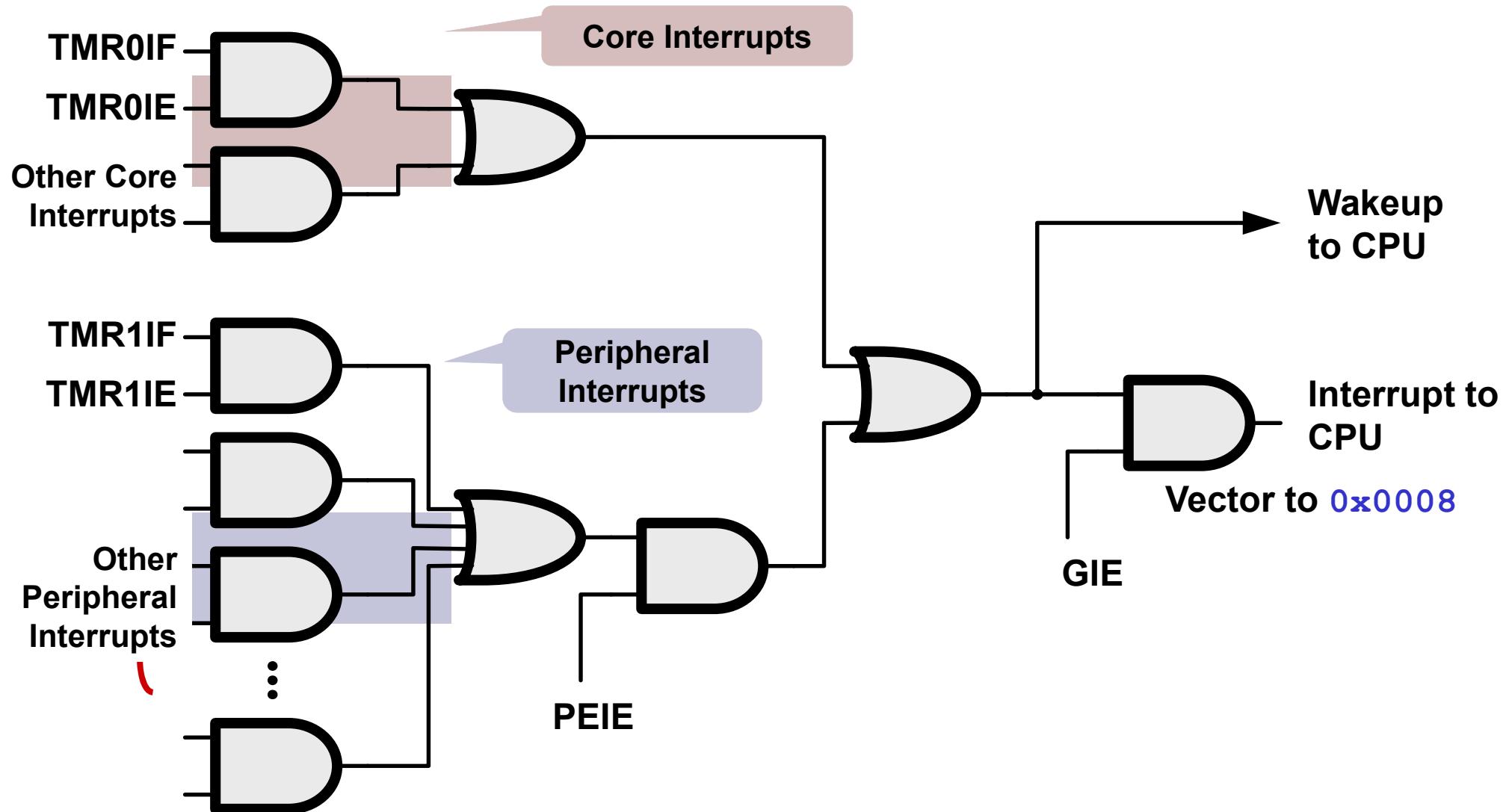
## Definición

**Interrupción** son llamadas a funciones disparadas por eventos del Hardware, a dichas funciones no se les puede pasar parámetros ni tampoco pueden devolver parámetros

- PIC18 Tiene 2 vectores: Alta y Baja Prioridad
- Las funciones de interrupciones deben ser tratadas de forma especial:
  - ISRs no reciben ni devuelven parámetros
  - ISRs no son llamadas por el código principal
  - ISRs no son llamadas por otras funciones

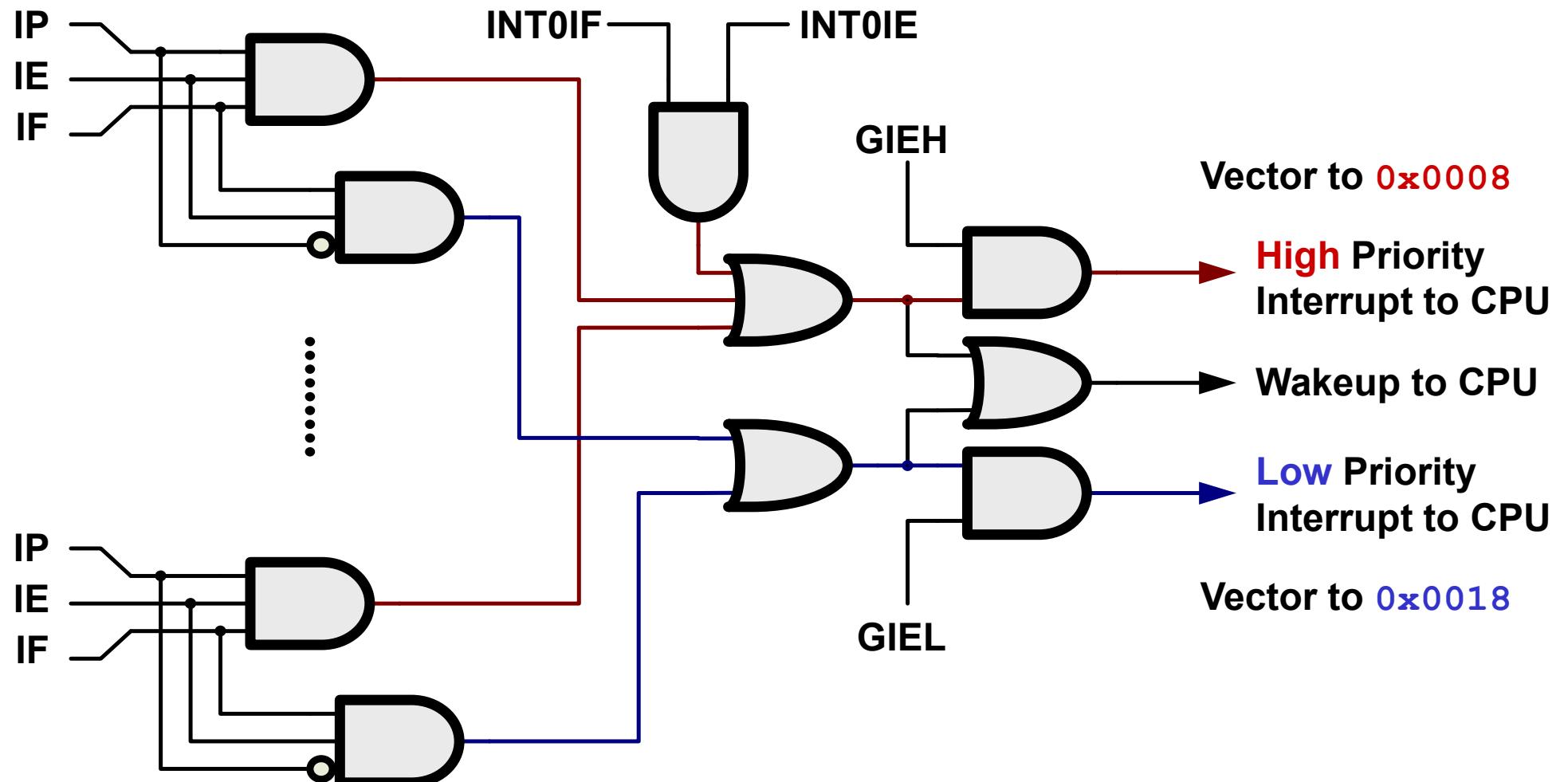
# Lógica de Interrupción

## Modo Legal o compatible con PIC16



# Lógica de Interrupción

## Modo Prioridad



# PIC18 Fuentes de Interrupción

## ■ Fuentes de Interrupción

- 3 o 4 Int. Externas (INT0-INT3)
  - Edge Triggered
  - Rising or Falling selected in INTCON2 register
- PORTB Interrupción por cambio (RB4-RB7)
- Timer Rollover/Overflow
- Cambio en la salida del comparador
- Final de Conversión A/D
- Eventos en el canal de comunicaciones
- Eventos en otros periféricos...

# Habilitando las Interrupciones

## (1 de 9)

- Select Legacy or Priority mode interrupt operation
  - Default is no priority levels
    - PIC16 Compatibility mode

RCON Register

IPEN	SBOREN	---	RI	TO	PD	POR	BOR
------	--------	-----	----	----	----	-----	-----

**IPEN:** Interrupt Priority Enable  
1 = Enable priority levels on interrupts  
0 = Disable priority levels on interrupt

# Habilitando las Interrupciones

## (2 de 9)

### ■ Set Peripheral Interrupt Priority

- Only needed if using Priority Mode
- 1 = High Priority, 0 = Low Priority

IPR1 Register

PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
-------	------	------	------	-------	--------	--------	--------

PSPIP: Parallel Slave Interrupt Priority  
ADIP: A/D Converter Interrupt Priority  
RCIP: EUSART Rcv Interrupt Priority  
TXIP: EUSART Tx Interrupt Priority

SSPIP: MSSP Interrupt Priority  
CCPIP: CCP1 Interrupt Priority  
TMR2IP: Timer2 Interrupt Priority  
TMR1IP: Timer1 Interrupt Priority

IPR2 Register

OSCIP	CMIP	---	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP
-------	------	-----	------	-------	--------	--------	--------

OSCIP: Oscillator Fail Interrupt Priority  
CMIP: Comparator Interrupt Priority  
---: Unimplemented Bit  
EEIP: Data EEPROM/Flash Write Operation Interrupt Priority

BCLIP: Bus Collision Interrupt Priority  
HLVDIP: High/Low Voltage Detect Interrupt Priority  
TMR3IP: Timer3 Interrupt Priority  
CCP2IP: CCP2 Interrupt Priority

# Habilitando las Interrupciones

## (3 de 9)

### ■ Ensure Peripheral Interrupt Flags are clear

- If set, an interrupt is pending or being processed
- 1 = Set, 0 = Clear

PIR1 Register

PSPIF	ADIF	RCIF	TXIF	SSPIF	CCPIF	TMR2IF	TMR1IF
-------	------	------	------	-------	-------	--------	--------

PSPIF: Parallel Slave Interrupt Flag  
ADIF: A/D Converter Interrupt Flag  
RCIF: EUSART Rcv Interrupt Flag  
TXIF: EUSART Tx Interrupt Flag

SSPIF: MSSP Interrupt Flag  
CCPIF: CCP1 Interrupt Flag  
TMR2IF: Timer2 Interrupt Flag  
TMR1IF: Timer1 Interrupt Flag

PIR2 Register

OSCIF	CMIF	---	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF
-------	------	-----	------	-------	--------	--------	--------

OSCIF: Oscillator Fail Interrupt Flag  
CMIF: Comparator Interrupt Flag  
----: Unimplemented Bit  
EEIF: Data EEPROM/Flash Write Operation Interrupt Flag

BCLIF: Bus Collision Interrupt Flag  
HLVDIF: High/Low Voltage Detect Interrupt Flag  
TMR3IF: Timer3 Interrupt Flag  
CCP2IF: CCP2 Interrupt Flag

# Habilitando las Interrupciones

## (4 de 9)

### ■ Set Peripheral Interrupt Enables

- 1 = Enabled, 0 = Disabled

#### PIE1 Register

PSPIE	ADIE	RCIE	TXIE	SSPIE	CCPIE	TMR2IE	TMR1IE
-------	------	------	------	-------	-------	--------	--------

PSPIE: Parallel Slave Interrupt Enable  
ADIE: A/D Converter Interrupt Enable  
RCIE: EUSART Rcv Interrupt Enable  
TXIE: EUSART Tx Interrupt Enable

SSPIE: MSSP Interrupt Enable  
CCPIE: CCP1 Interrupt Enable  
TMR2IE: Timer2 Interrupt Enable  
TMR1IE: Timer1 Interrupt Enable

#### PIE2 Register

OSCIE	CMIE	---	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE
-------	------	-----	------	-------	--------	--------	--------

OSCIE: Oscillator Fail Interrupt Enable  
CMIE: Comparator Interrupt Enable  
---: Unimplemented Bit  
EEIE: Data EEPROM/Flash Write Operation Interrupt Enable

BCLIE: Bus Collision Interrupt Enable  
HLVDIE: High/Low Voltage Detect Interrupt Enable  
TMR3IE: Timer3 Interrupt Enable  
CCP2IE: CCP2 Interrupt Enable

# Habilitando las Interrupciones (5 de 9)

## ■ Set Core Interrupt Priority

- Only needed if using Priority Mode
- 1 = High Priority, 0 = Low Priority

INT0 does not have an IP bit – it is always a high priority interrupt

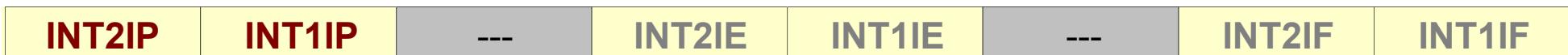


INTCON2 Register



TMR0IP: TMR0 Overflow Interrupt Priority  
RBIP: RB Port Change Interrupt Priority

INTCON3 Register



INT2IP: INT2 External Interrupt Priority  
INT1IP: INT1 External Interrupt Priority

# Habilitando las Interrupciones

## (6 de 9)

- Ensure Core Interrupt Flags are Clear
  - If set, an interrupt is pending or being processed
  - 1 = Set, 0 = Clear

### INTCON Register

GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	<b>TMR0IF</b>	<b>INT0IF</b>	<b>RBIF</b>
----------	-----------	--------	--------	------	---------------	---------------	-------------

TMR0IF: TMR0 Overflow Interrupt Flag

INT0IF: INT0 External Interrupt Flag

RBIF: RB Port Change Interrupt Flag

### INTCON3 Register

INT2IP	INT1IP	---	INT2IE	INT1IE	---	<b>INT2IF</b>	<b>INT1IF</b>
--------	--------	-----	--------	--------	-----	---------------	---------------

INT2IF: INT2 External Interrupt Flag

INT1IF: INT1 External Interrupt Flag

# Habilitando las Interrupciones

## (6 de 9)

### ■ Set Core Interrupt Enables

- 1 = Enabled, 0 = Disabled

#### INTCON Register

GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
----------	-----------	--------	--------	------	--------	--------	------

TMR0IE: TMR0 Overflow Interrupt Enable

INT0IE: INT0 External Interrupt Enable

RBIE: RB Port Change Interrupt Enable

#### INTCON3 Register

INT2IP	INT1IP	---	INT2IE	INT1IE	---	INT2IF	INT1IF
--------	--------	-----	--------	--------	-----	--------	--------

INT2IE: INT2 External Interrupt Enable

INT1IE: INT1 External Interrupt Enable

# Habilitando las Interrupciones

## (8 de 9)

- **Enable Low Priority or Peripheral Interrupts**
  - Still need to enable Global Interrupts when in PIC16 compatibility mode

### INTCON Register

GIE/GIEH	<b>PEIE/GIEL</b>	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
----------	------------------	--------	--------	------	--------	--------	------

#### PEIE/GIEL :

**Peripheral Interrupt Enable (PIC16 Compatibility Mode) /  
Global Interrupt Enable Low (Priority Mode)**

#### In PIC16 Compatibility Mode (RCON<IPEN> = 0):

1 = Enables all unmasked peripheral interrupts  
0 = Disables all peripheral interrupts

#### In Priority Mode (RCON<IPEN> = 1):

1 = Enables all low priority peripheral interrupts  
0 = Disables all low priority peripheral interrupts

# Habilitando las Interrupciones

## (9 de 9)

### ■ Enable High Priority or Global Interrupts

#### INTCON Register

<b>GIE/GIEH</b>	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
-----------------	-----------	--------	--------	------	--------	--------	------

#### GIE/GIEH :

**Global Interrupt Enable (PIC16 Compatibility Mode) /  
Global Interrupt Enable High (Priority Mode)**

#### In PIC16 Compatibility Mode (RCON<IPEN> = 0):

1 = Enables all unmasked interrupts  
0 = Disables all interrupts

#### In Priority Mode (RCON<IPEN> = 1):

1 = Enables all high priority interrupts  
0 = Disables all high priority interrupts

# Control de interrupciones con CCS

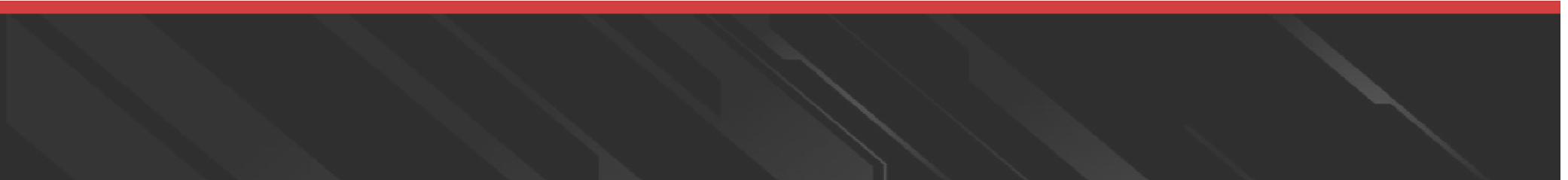
- Se usa la directiva #INT\_XXX para indicar la fuente de la interrupcion.
  - XXX=depende del periférico interruptor
- La rutina de interrupción no puede recibir ni pasar parámetros, debe ser **void**
- Ejemplo:

```
#int_rda  
void isr_recepcion (void)  
{  
    sentencias  
}
```



MICROCHIP

*Thank You!*



## Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, Mindi, MiWi, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.