

Sistemas de Tiempo Real

Microcontroladores

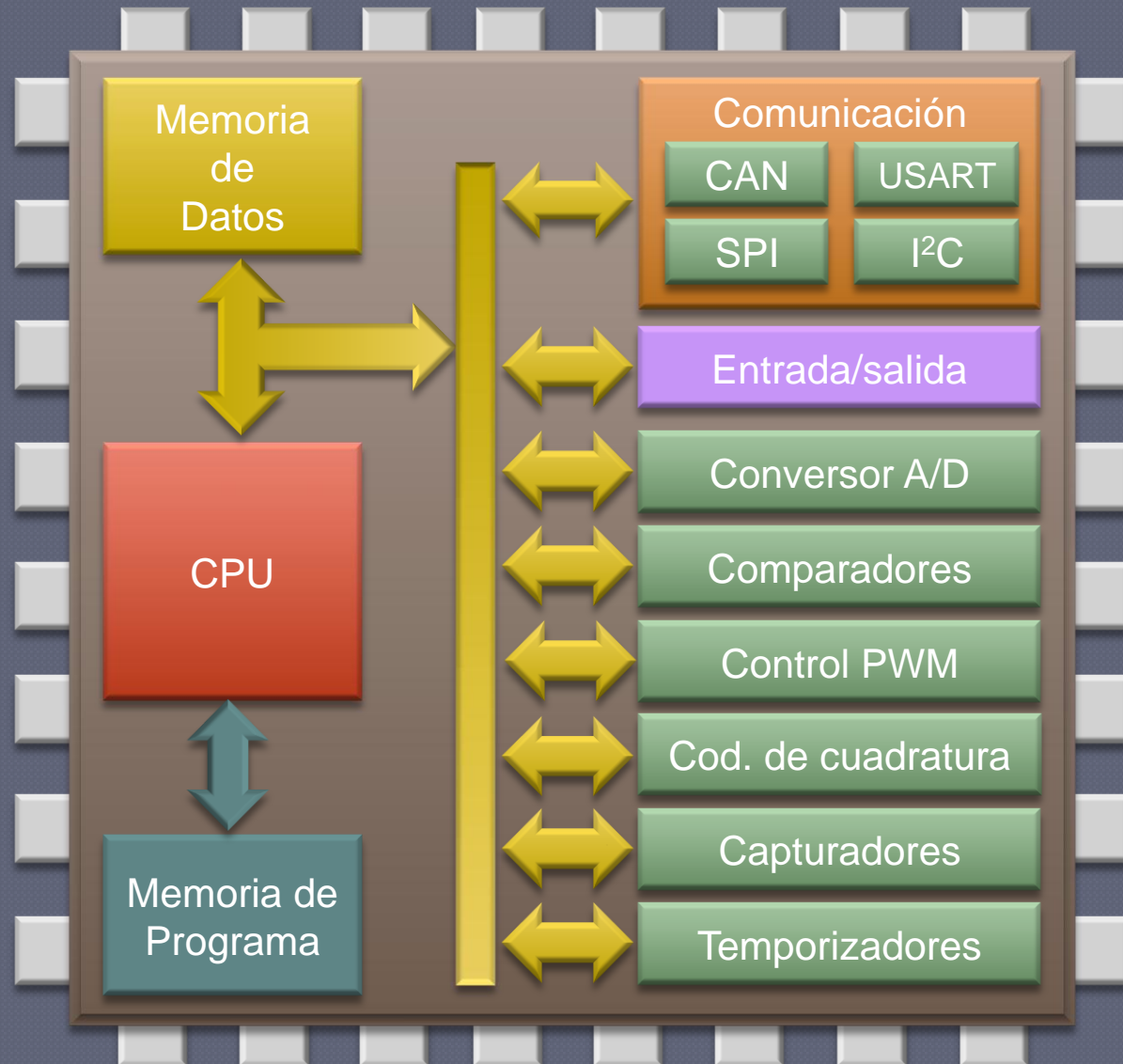
Microcontroladores – Introducción

- ◉ Un microcontrolador es un microprocesador especializado en el control de dispositivos o equipos electrónicos.
- ◉ Integra memoria de datos y programa, y módulos de hardware que le permiten “sensar,” controlar y comunicarse con otros dispositivos.
- ◉ Existe en el mercado una gran cantidad de microcontroladores de muchas empresas: Atmel, Freescale (Motorola), Hitachi, Intel, National Semiconductor, Microchip, NEC, Parallax, ST Microelectronics, Texas Instruments

Microprocesador vs. Microcontrolador

Microprocesador	Microcontrolador
Alto nivel de generalización	Alto nivel de especialización
No integra periféricos dentro del mismo chip	Integra periféricos dentro del chip
Alto Costo	Bajo Costo
Consumo de energía medio/alto	Consumo de energía bajo/medio
Velocidades de reloj altas	Velocidades de reloj medianas/bajas
Generalmente son CISC	Generalmente son RISC
Se utiliza para procesar información	Se utiliza para controlar dispositivos electrónicos
Necesita de gran cantidad de componentes externos para funcionar	Necesita de pocos componentes externos para funcionar
Memorias de datos y programa externas	Memorias de datos y programa internas
Tiempo de ejecución variable entre instrucciones	Tiempo de ejecución constante entre instrucciones

Microprocesador vs. Microcontrolador



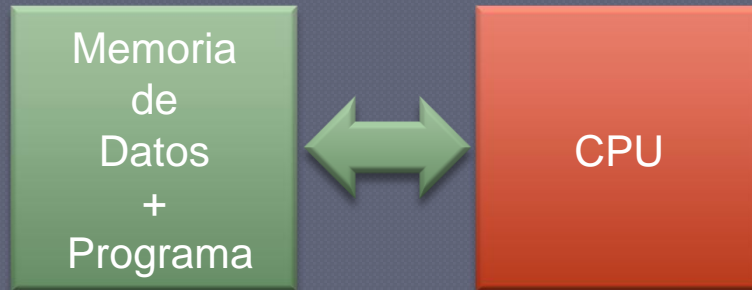
Microcontroladores - Aplicaciones

- ◉ Control de motores: continuos, PWM, etc.
- ◉ Biometría: Reconocimiento huellas digitales y VOZ
- ◉ Telefonía: detección de DTMF, identificadores de llamadas
- ◉ Aparatos electrónicos: impresoras, scanners, módems, microondas, etc.
- ◉ Mediciones en general: temperatura, humedad, señales biológicas.
- ◉ Procesamiento de señales: reducción de ruidos.
- ◉ Automotriz: airbag, frenos ABS.
- ◉ Juguetes y Robótica.

Microcontroladores - Características

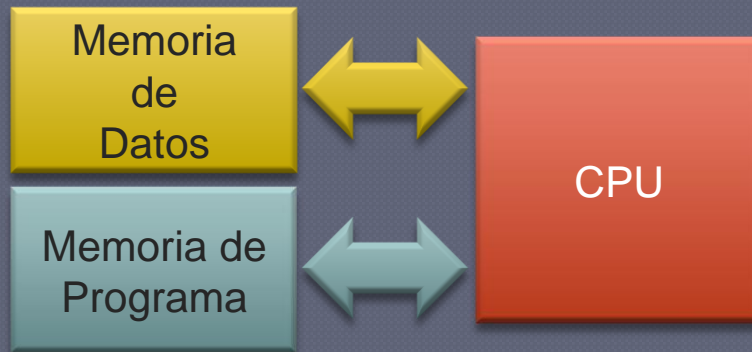
- ◉ Arquitectura Harvard.
- ◉ Memoria de programa reducida:
 - Unos pocos Kbytes.
 - suelen ser ROM (una grabación) o EEPROM (muchas grabaciones).
- ◉ Memoria de datos reducida:
 - Desde unos bytes a un par de Kbytes.
- ◉ Conjunto Reducido de Instrucciones (RISC).
- ◉ Salvo líneas de altas prestaciones y DSP, no tienen soporte para multiplicaciones y divisiones.
- ◉ Integra temporizadores, conversores A/D, dispositivos de comunicación.

Microcontroladores - Arquitectura



Arquitectura Von Neumann

- Flexibilidad para almacenar datos y programas.



Arquitectura Harvard

- Independencia de tamaños de buses de memoria y datos.
- Simplificación de la electrónica de control.
- Acceso simultáneo a memoria de datos y programa.

Microcontroladores - Dispositivos

- ◉ Entrada / salida digital.
- ◉ Comunicación :
 - SPI, I²C, UART, CAN, USB, SPP.
- ◉ Temporizadores.
- ◉ Comparadores de salida.
- ◉ Capturadores de entrada.
- ◉ Conversores analógico/digital.
- ◉ Control por PWM (Modulación de ancho de pulso)
- ◉ Codificadores de cuadratura.

Microcontroladores - Comunicación

SPI, Serial Peripheral Interface:

● Líneas de comunicación :

- SDI: entrada de datos serie (serial data input).
- SDO: salida de datos serie (serial data output).
- SCK: señal reloj para E/S (serial clock).
- SS: selección de esclavo (slave select).

● Características:

- Comunicación sincrónica full-duplex.
- Transmisión y recepción de flujos de 8/16 bits, 1 ciclo/bit.
- Maestro/esclavo, el maestro genera señal SCK y activa señal SS.
- Requiere de una línea SS por dispositivo esclavo.
- Capacidad de interrumpir al transmitir/recibir.
- Velocidades de transferencia entre 1Mhz y 70Mhz.

Microcontroladores – Comunicación

I²C, Inter-Integrated Circuit :

◉ Líneas de comunicación :

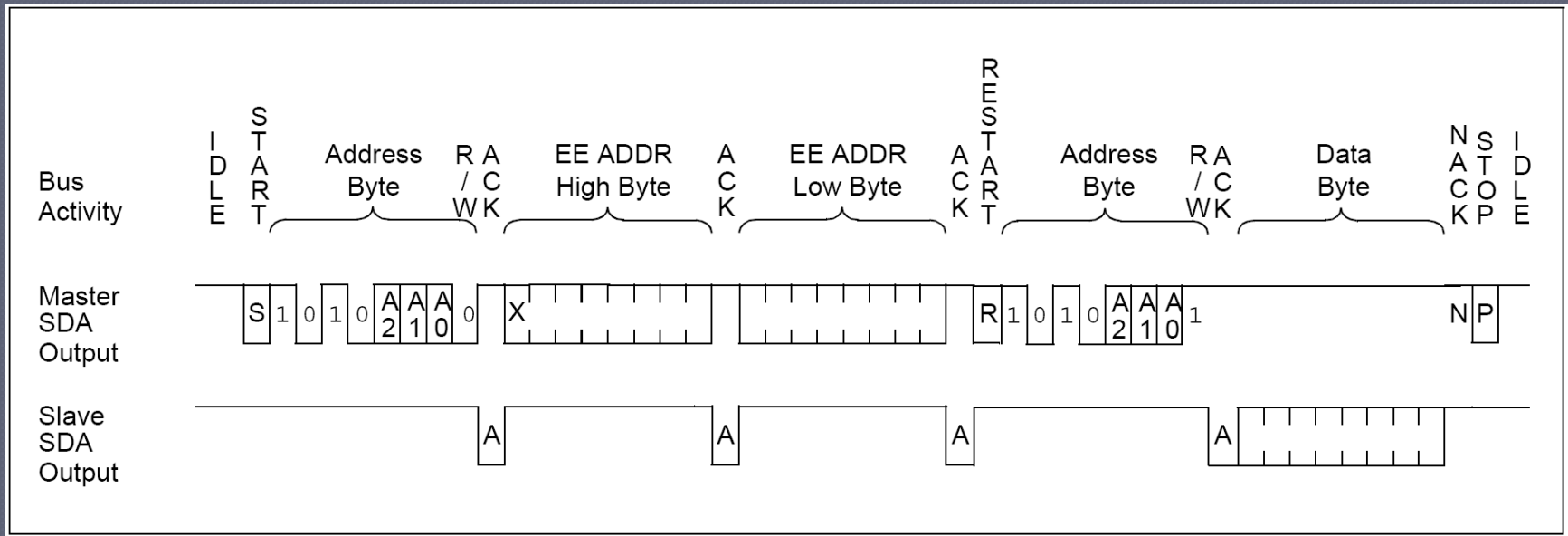
- SDA: salida de datos serie (serial data).
- SCL : señal reloj para E/S (serial clock).

◉ Características:

- Comunicación sincrónica half-duplex.
- Modos Maestro, esclavo, o multi-maestro con detección de colisiones y arbitraje.
- El maestro transmite número de dispositivo e información de la operación a realizar . El dispositivo esclavo responde .
- Comunicación con tantos dispositivos como direcciones pueda codificar.
- Capacidad de interrumpir al transmitir/recibir.
- Velocidades de transferencia de 100Khz y 400Khz.

Microcontroladores – Comunicación

I²C, Inter-Integrated Circuit:



Lectura de memoria EEPROM

Microcontroladores – Comunicación

USART, Universal Synchronous/Asynchronous Receiver/Transmitter:

○ Características:

- Comunicación sincronica/asincrónica full-duplex de 8 o 9 bits.
- Opciones de paridad par, impar o ninguna. 1 o 2 Bits de parada.
- Detección de errores de paridad y desbordamiento de buffers.
- Capacidad de interrumpir al transmitir/recibir.
- Velocidades de transferencia de 29 bps hasta 1,875 Mbps.

Microcontroladores – Comunicación

CAN, Control Area Network:

- ◉ Sistema de comunicación orientado al mensaje. El mensaje corre por la red y cada dispositivo filtra los de su interés.
- ◉ Diseñado para entornos ruidosos.
- ◉ Longitud de datos entre 0 y 8 bytes.
- ◉ Tipos de Tramas:
 - Estandar: transmisión de datos.
 - Extendida: extender transmisión normal.
 - Remota: para que algún nodo responda a un pedido.
 - Error: transmitida por un nodo cuando detecta un error.
- ◉ Capacidad de interrumpir al transmitir/recibir.
- ◉ Velocidades de transferencia de 1 Mbps.

Microcontroladores - Dispositivos

Temporizadores (Timers)/Contadores:

- Registros para contar (“tiempo” o eventos)
- Resolución: 8bits, 16bits, 32 bits.
- Para contar tiempos “largos” se combinan con:
 - Otros contadores: 2 de 8 bits o 2 de 16 bits.
 - Divisores de frecuencia:
 - Prescaler: cuando esta como entrada del contador.
 - Postscaler: cuando esta como salida del contador.
- Fuentes de entrada, controla cuando contar:
 - interna: reloj interno, prescaler, contador.
 - externa: señal externa que entrada desde un pin del micro.
- Generan eventos por desborde o por comparación con registros programables:
 - Interrupciones.
 - Disparan conversiones A/D.
 - Cambio en un pin de salida.

Microcontroladores - Dispositivos

Comparadores de salida:

- ◉ Permite comparar dos valores analógicos a través de 2 pines de entrada.
- ◉ Puede realizar una acción sobre un pin de salida:
 - Cambiar su estado.
 - Generar un pulso
 - Generar un tren de pulsos.
- ◉ Tienen capacidad de interrumpir.

Microcontroladores – Dispositivos

Capturadores de entrada:

- ◉ Permite detectar un cambio en un pin de entrada.
- ◉ Cuando registra el evento almacena el valor de un temporizador en un registro
- ◉ Útil para medir frecuencias.
- ◉ Tienen capacidad de interrumpir.

Microcontroladores – Dispositivos

Conversores Analógico/Digital:

- Permiten convertir una entrada analógica en un valor digital.
- Resolución comunes de 8, 10, 12 y 16 bits.
- Normalmente tienen un conversor A/D y un multiplexor para seleccionar canales.
- Los micros de alto rendimiento normalmente pueden programarse para muestreo secuencia de canales.
- Tienen capacidad de interrumpir al finalizar la conversión.
- Tipo de conversiones:
 - Manual: cuando se lo indica el programador.
 - Automática: dispara una nueva conversión al terminar la actual.
 - Automática por evento: se dispara la conversión al producirse un evento asociado a un dispositivo (ej: timer).

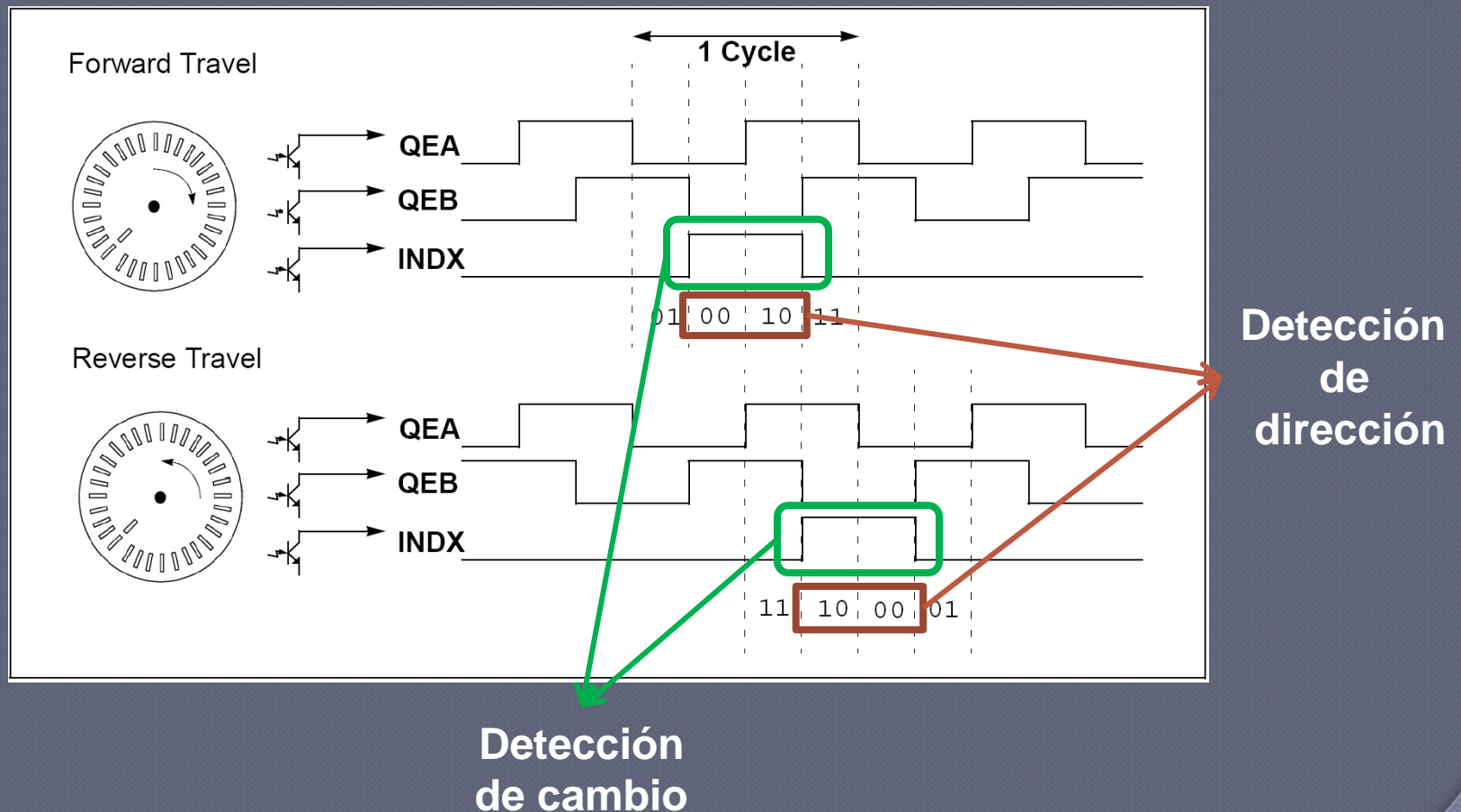
Microcontroladores – Dispositivos

Codificadores de Cuadratura:

- son codificadores incrementales para detectar la posición y velocidad de rotación de movimiento
- Tiene 3 señales: Fase A (QEA), Fase B(QEB) e índice (INDX).
- Con estas señales e incrementa o decrementa un contador según la dirección de giro.
- Este dispositivo cuenta con:
 - Contador de posición (ej: 16 bits).
 - Mantiene cuenta y estado de dirección.
 - Resolución de cuenta por dos (x2) y por cuatro (x4), para manejar la sensibilidad.
 - Interrupciones generadas por INDX o por eventos del contador.

Microcontroladores– Dispositivos

○ Codificadores de Cuadratura



Microcontroladores – Microchip PIC

○ Son populares:

- Hay muchas herramientas para desarrollo.
- Es posible conseguir muchos de los modelos con facilidad para.
- Es posible adquirir dispositivos para grabar los programas (“programadores”) fácilmente y a bajo costo.

○ Muchas “familias” de microcontroladores:

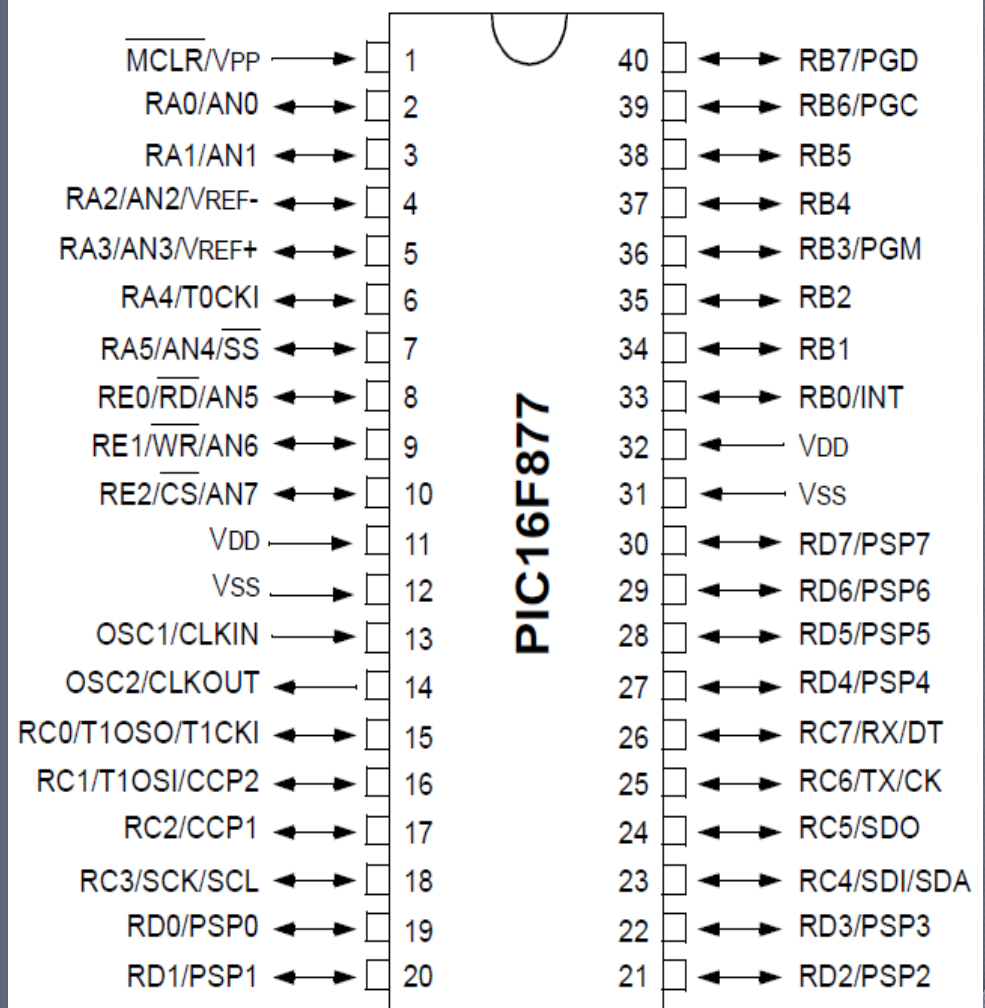
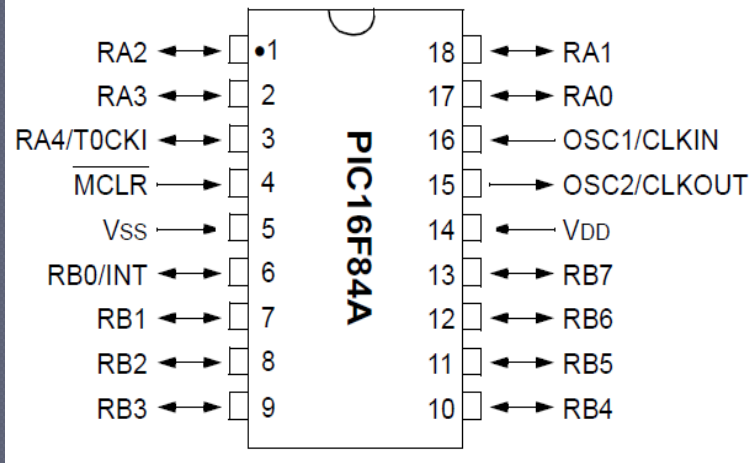
- Datos de 8 bits
- Datos de 16 bits
- Datos de 32 bits
- Procesamiento Digital de Señales (DSP).

Microcontroladores – Microchip PIC

En la práctica usaremos:

- Microcontroladores PIC 16F84A (básico) y PIC 16F877 (completo).
- Entorno Isis – Proteus (versión demo) que permite simular circuitos electrónicos , inclusive microcontroladores.
- Compilador C, PICC (versión “lite”).

Microcontroladores – Microchip PIC



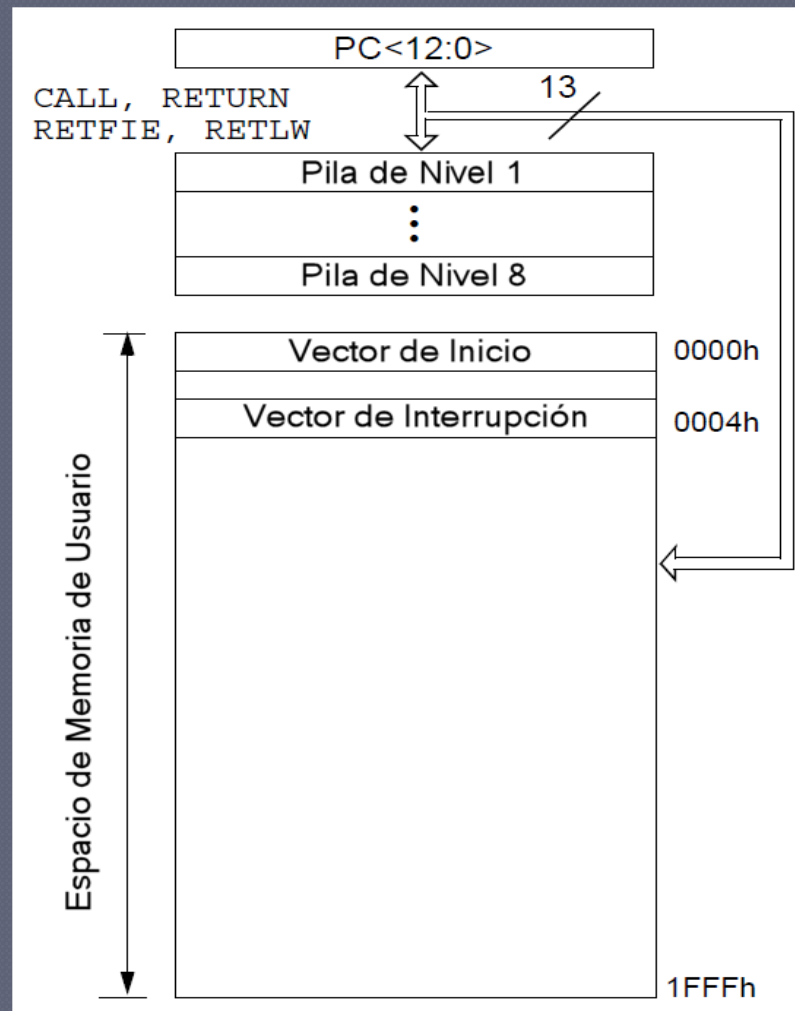
Microchip PIC - Arquitectura

- ◉ Arquitectura Harvard.
- ◉ Núcleo RISC.
- ◉ Cauce (Pipeline) de 4 etapas: *fetch*, *decode*, *execute*, *write-back*.
- ◉ 35 instrucciones de 14 bits.
- ◉ Ejecución de instrucciones en 1 ciclo de instrucción (4 de reloj). 2 ciclos para saltos.
- ◉ Velocidades de reloj de hasta 20 MHz (5 MIPS).
- ◉ Una única interrupción para varias fuentes.
- ◉ Memoria EEPROM que permite mantener datos persistentes al estar apagado.

PIC – Memoria de Programa

- Hasta unos Kbytes de palabras.
- Vector de inicio en la posición 0: instrucción de salto a la primera línea del programa.
- Vector de interrupción en posición 4: instrucción de salto a la primera línea de rutina de interrupción única.
- Programa a partir de la posición 5, la capacidad depende del modelo del microcontrolador.
- Pila para subrutinas de 8 niveles.

PIC – Memoria de Programa



PIC – Memoria de Datos

- Memoria paginada (bancos), basada en banco de registros: posición de memoria = registro.
- Siempre hay una pagina activa, para acceder a posiciones fuera de la actual hay que cambiar de página (si usamos C esto lo hace el compilador).
- Por razones de rendimiento algunos registros que se usan frecuentemente (ej: para direccionamiento o estado de ALU) se mapean en todas las páginas o bancos.

PIC – Memoria de Datos - Registros

- Propósito especial:

- Asociados al estado y control de dispositivos.
- reciben nombres según su función, incluso su bits.
- En general se ubican en las posiciones iniciales.

- Propósito general:

- RAM normal, se ubican después de los de propósito especial.

BANCO DE REGISTROS 0			BANCO DE REGISTROS 1		
Dir	Nombre	Descripción	Dir	Nombre	Descripción
00h	INDF*	Contenido apuntado por el registro FSR (no es un registro físico)	80h	INDF*	Contenido apuntado por el registro FSR (no es un registro físico)
01h	TMR0	Contador de 8 bit	81h	OPTION	Configuración asociada a TMR0
02h	PCL*	8 bit bajos del Contador de Programa	82h	PCL*	8 bit bajos del Contador de Programa
03h	STATUS*	Estado de operaciones aritméticas	83h	STATUS*	Estado de operaciones aritméticas
04h	FSR*	Puntero indirecto de direccionamiento de datos	84h	FSR*	Puntero indirecto de direccionamiento de datos
05h	PORTA	Puerto de entrada/salida A	85h	TRISA	Configuración del puerto A
06h	PORTB	Puerto de entrada/salida B	86h	TRISB	Configuración del puerto B
07h	--	No implementada, se lee como "00"	87h	--	No implementada, se lee como "00"
08h	EEDATA	Registro de datos EEPROM	88h	EECON1	Registro de control de EEPROM
09h	EEADR	Registro de direcciones EEPROM	89h	EECON2	Registro de control de EEPROM
0Ah	PCLATH*	Bits más significativos del PC	8Ah	PCLATH*	Bits más significativos del PC
0Bh	INTCON*	Registro de control de interrupciones	8Bh	INTCON*	Registro de control de interrupciones
0Ch a 4Fh	--	68 registros de propósito general (SDRAM)*	8Ch	--*	Mapeados al Banco 0, es decir que la lectura/escritura de estos registros se hace sobre los registros del Banco 0
50h A 7Fh	--	No implementados	D0h A FFh	--	No implementados

Registros de PIC 16F84A. * registros de bancos diferentes mapeados físicamente a un mismo registro.

PIC – Memoria de EEPROM

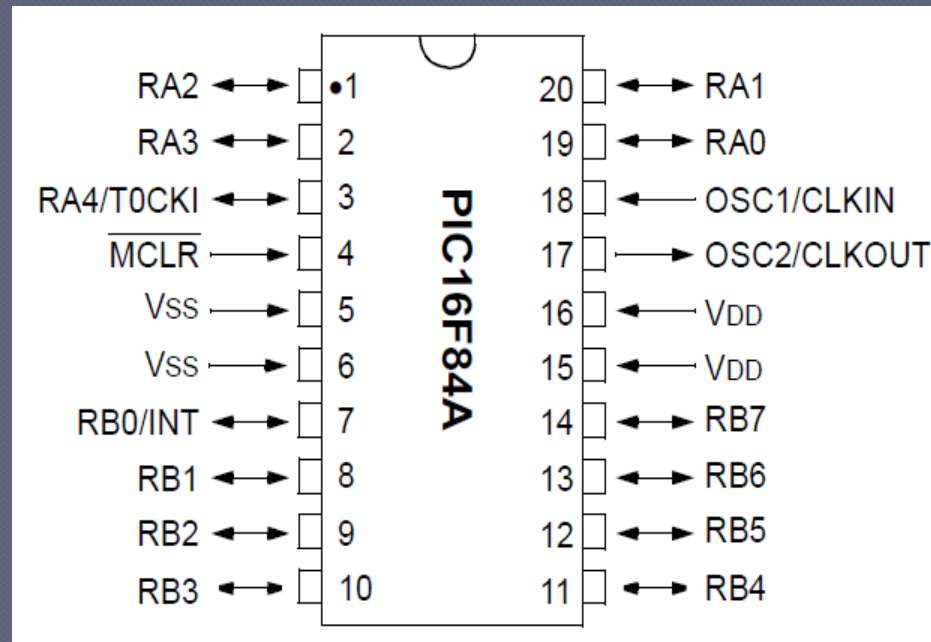
- Memoria de tipo flash, normalmente de hasta 256 bytes, algunos micros almacenan algunos kbytes. De necesitar más se apoyan en dispositivos externos con comunicación I2C o SPI.
- No es accesible directamente, se utilizan registros de funciones especiales:
 - Dirección: posición que se quiere acceder.
 - Datos: registro donde se lee/escribe el valor.
- Generan interrupción al finalizar la operación.

PIC – Entrada/Salida

- ◉ Permiten a través de un registro (puerto) enviar/recibir una señal digital al exterior
- ◉ La cantidad de puertos depende del modelo del microcontrolador.
- ◉ Se nombra al registro que permite el acceso como $\text{PORT}\{X\}$ donde X es una letra, “A” para el 1º, “B” para el 2º, etc.
- ◉ Un registro $\text{PORT}\{X\}$ tiene asociado otro de control $\text{TRIS}\{X\}$ que configura el hardware para controlar el sentido de cada bit:
 - ◉ “1” en bit de $\text{TRIS}\{X\}$, bit de $\text{PORT}\{X\}$ es entrada.
 - ◉ “0” en bit de $\text{TRIS}\{X\}$, bit de $\text{PORT}\{X\}$ es salida.

PIC – Entrada/Salida

- Cada bit del registro $\text{PORT}\{X\}$ también recibe un nombre $R\{X\}\{Y\}$ donde X es el nombre del puerto que pertenecen e Y es el número de bit

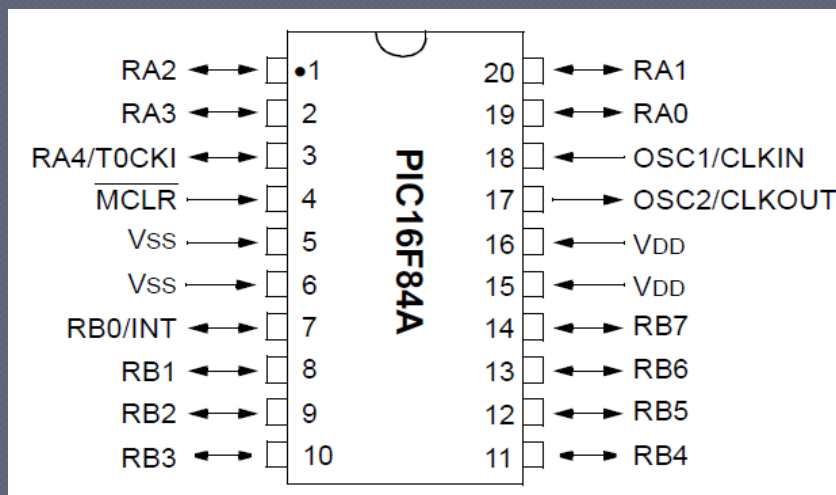


PIC – Entrada/Salida

- La programación del 2º puerto (PORTB) puede realizarse:

```
3 // configuración del puerto B
4 // como 7 entradas y 1 salida
5 TRISB = 0xFE; // hexadecimal
6 TRISB = 0b1111110 // binario
7 TRISB = 254 // decimal
```

```
3 // Puerto B, 7 entradas y 1 salida
4 // lee 8, ult. bit no importa
5 mi_var = PORTB;
6 // activa salida de puerto B
7 RB0 = 1;
```



PIC – Temporizador/contador

- ◉ Un temporizador/contador es un registro que permite contar eventos:
 - ◉ Si el evento es asincrónico lo denominamos “contador”.
 - ◉ Si el evento es sincrónico lo denominamos “temporizador”.
- ◉ Pueden combinarse con prescaler y postcaler.
- ◉ Usamos el módulo temporización “Timer0” del PIC 16F84A:
 - ◉ Resolución de 8 bits.
 - ◉ Puede usar un prescaler con 8 valores para dividir frecuencia: 2,4,8,16,32,64,128,256.
 - ◉ Puede interrumpir al desbordar (transición 255→0).

PIC – Temporizador/contador

- Registros asociados al módulo “Timer0”:
 - TMR0: contador de eventos.
 - OPTION_REG: controla la programación del prescaler y la fuente que sincroniza la cuenta de TMR0. Utiliza los bits 5 a 0.
 - INTCON: contiene el estado y el controla la habilitación de las interrupciones. Bits 7, 5 y 2 asociados al Timer.

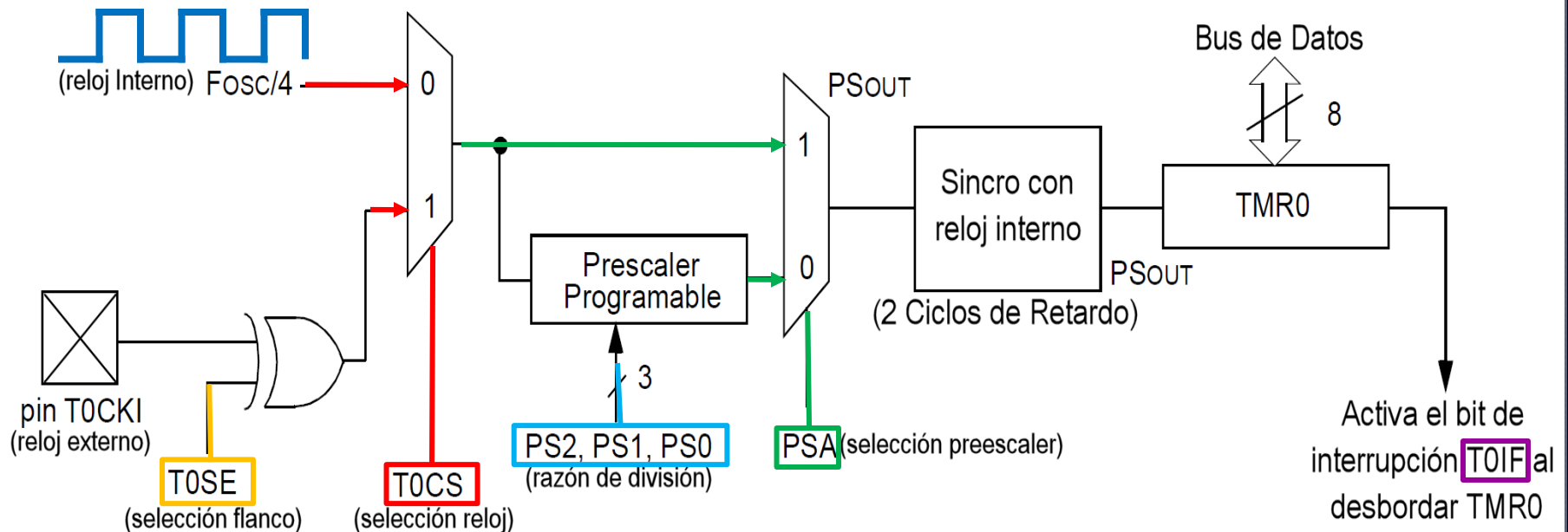
PIC – Registro OPTION_REG

Bit	Nombre	Descripción
5	TOCS	Bit de selección de reloj para incremento (TMROClockSourceSelect): <ul style="list-style-type: none"> • 1 = reloj externo conectado a pin TOCKI • 0 = reloj interno asociado al ciclo de instrucción.
4	TOSE	Bit de selección de flanco de transición (TMROSourceEdgeSelect) , solo tiene sentido cuando la fuente es reloj externo (TOCS=1): <ul style="list-style-type: none"> • 1 = Incrementa en transición alto-bajo del pin TOCKI. • 0 = Incrementa en transición bajo-alto del pin TOCKI.
3	PSA	Bit de asignación de prescaler (PrescalerAssignment): <ul style="list-style-type: none"> • 1 = Prescaler asignado a Temporizador Watchdog. • 0 = Prescaler asignado a módulo del Timer0.
2, 1, 0	PS2, PS2, PS0	3 bits (valor de 0 a 7) que indican los ciclos que cuenta el prescaler para generar un ciclo para el temporizador. La división que realiza el prescaler es $1: 2^{(n+1)}$. Un valor $n=1:1:1=7 \rightarrow 1:256$ (1 ciclo temporizador cada 256 del prescaler).

PIC – Registro INTCON

Bit	Nombre	Descripción
7	GIE	Habilitación General de Interrupciones (G lobal I nterrupt E nable): <ul style="list-style-type: none">• 1 = Habilitación general de interrupciones.• 0 = Deshabilita todas las interrupciones.
5	TOIE	Habilitación de interrupción por desborde del Timer0 (TMR0 Overflow I nterrupt E nable): <ul style="list-style-type: none">• 1 = Habilita interrupción del Timer0.• 0 = Deshabilita interrupción del Timer0.
2	TOIF	Flag de estado de interrupción por desborde del Timer0 (TMR0 Overflow I nterrupt F lag): <ul style="list-style-type: none">• 1 = El registro TMR0 ha desbordado (requiere limpiado por soft).• 0 = El registro TMR0 no ha desbordado.

PIC – Módulo Timer0



OPTION_REG

-	-	T0C S	T0S E	PSA	PS2	PS1	PS0
---	---	----------	----------	-----	-----	-----	-----

INTCON

GIE	-	T0IE	-	-	T0IF	-	-
-----	---	------	---	---	------	---	---

PIC – Módulo Timer0 - Programación

- ◉ Establecer la fuente de reloj del registro OPTION_REG:
 - Reloj interno (incrementa con ciclo de instrucción): T0CS=0
 - Reloj externo (pin T0CLKI): T0CS=1 y T0SE con flanco de transición
- ◉ Establecer el uso prescaler del registro OPTION_REG:
 - Usa prescaler: PSA=1 y PS2:PS0 con valor para prescaler.
 - No usa prescaler: PSA=0.
- ◉ Establecer los bits la interrupción del registro INTCON:
 - Usa interrupción: T0IE=1 y GIE=1
 - No usa interrupción: T0IE=0 y GIE depende si hay mas dispositivos programados para interrumpir.
- ◉ Inicialización del registro TMR0:
 - El valor depende de la necesidad. Recordar que el desborde genera interrupción, por ejemplo para interrumpir en 100 ciclos el valor a guardar es 156 (256-100).

PIC – Cálculo de tiempo

- Todo cálculo de tiempo depende de la frecuencia de un reloj:
 - Interno: se toma el ciclo de instrucción (4 de reloj). Si el ciclo de reloj es de 1Mhz, el de instrucción es de 250khz (4 microsegundos por instrucción).
 - Externo: se toma cada ciclo de reloj y depende de su frecuencia.
- Es posible utilizar el prescaler como divisor de frecuencia:
 - Divide por 2,4,8,16,32,64,128 o 256.
- Para tiempos “cortos” se utiliza solo el temporizador.
- Para tiempos “largos” se utiliza el temporizador combinado con el prescaler.

PIC – Cálculo de tiempo

- La siguiente ecuación permite calcular los valores del prescaler y/o temporizador:

$$\textbf{\textit{Tiempo}} = \textbf{\textit{Tpolnstr}} * \textbf{\textit{CiclosPrescaler}} * \textbf{\textit{CiclosTimer}}$$

- Para tiempos cortos donde no se utiliza el prescaler, este valor se reemplaza por 1:

$$\textbf{\textit{Tiempo}} = \textbf{\textit{Tpolnstr}} * \textbf{\textit{CiclosTimer}}$$

PIC – Cálculo de tiempo - Ejemplo

- Supongamos que tenemos un microcontrolador con un reloj de 1Mhz y necesitamos un retardo de alrededor de 100 milisegundos (ms).
- Calculamos el tiempo de una instrucción:
 - $1\text{Mhz} / 4 = 250\text{Khz}$, me da la frecuencia de instrucción.
 - $1 / 250\text{khz} = 0,000004 \text{ segundos} = 4 \text{ microsegundos}$.
- Despejando los ciclos del Timer, asumiendo valores de 64, 128 y 256 para los ciclos del prescaler y reemplazando el tiempo que buscamos, tenemos:
 - $\text{CiclosTimer} = 100000 \mu\text{s} / (4\mu\text{s} * 256) = 97,65625$**
 - $\text{CiclosTimer} = 100000 \mu\text{s} / (4\mu\text{s} * 128) = 195,3125$**
 - $\text{CiclosTimer} = 100000 \mu\text{s} / (4\mu\text{s} * 64) = 390,625$**

PIC – Cálculo de tiempo - Ejemplo

- Obtenidos los valores, re-calculamos los tiempos con los valores incrementados del timer y seleccionamos la mejor aproximación.
- Si la diferencia de μs nos importa, ajustamos por soft.

Prescaler		Timer		Tiempo en μs	
Ciclos	Valor de Programación (3 bits PS)	Ciclos	Valor de Programación (256-ciclos)	Valor más cercano a 100000	Difer.
256	8	97	159	99328	662
256	8	98	158	100352	352
128	7	195	61	99840	160
128	7	196	60	100352	352

PIC – Conversión A/D

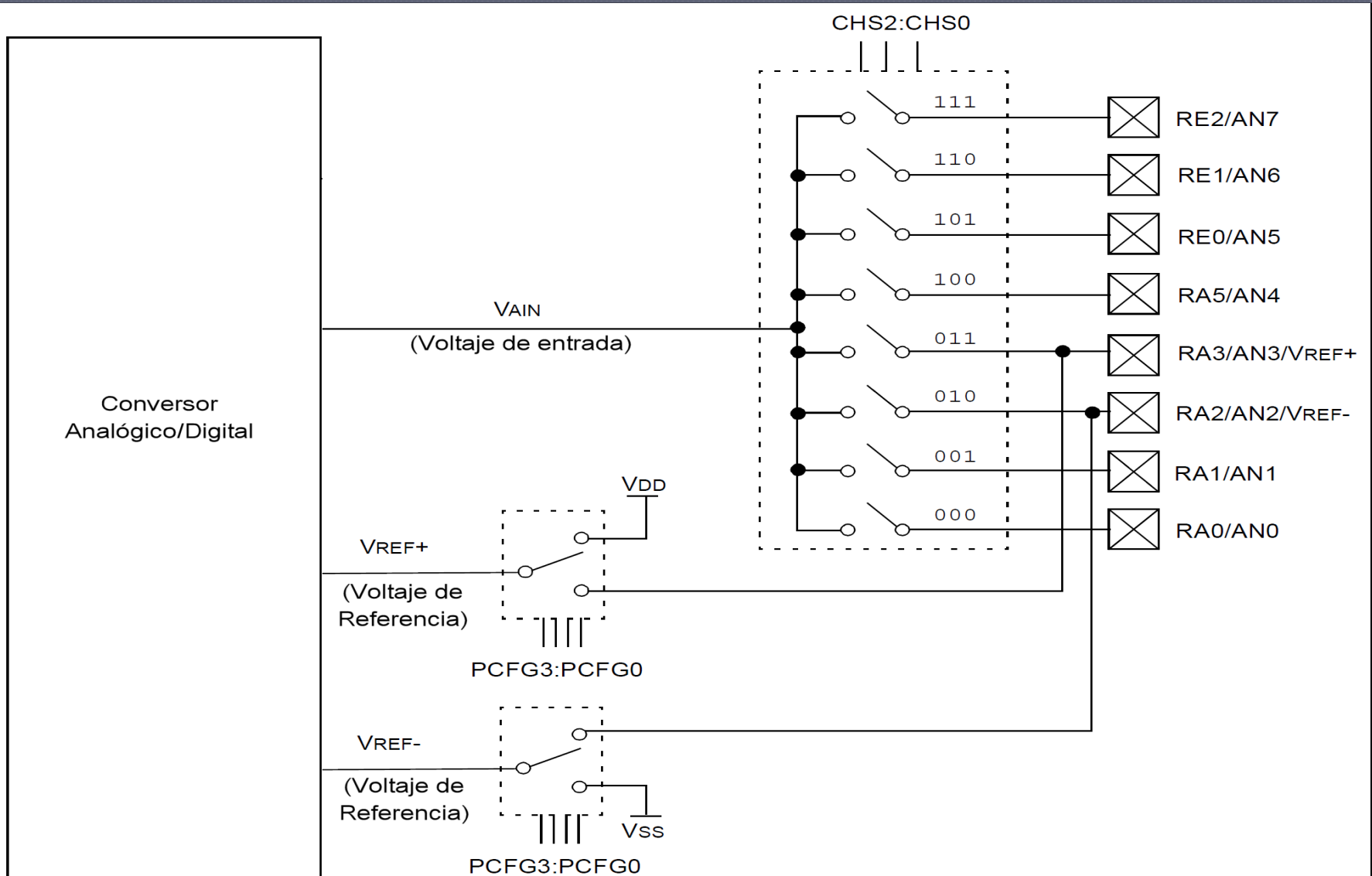
- Un conversor Analógico / Digital nos permite tomar señales externas al microcontrolador y convertirlas a un valor digital.
- Usamos el módulo de conversión A/D del PIC 16F877:
 - Resolución de 10 bits.
 - Hasta 8 canales multiplexados.
 - Permite establecer la tensión (normalmente 5 volts) de referencia que se utiliza para “escalar” los valores.
 - Puede interrumpir al finalizar la conversión.

PIC – Conversión A/D

Registros asociados al módulo A/D:

- ◉ ADRESH(**A/D**Result**H**igh): parte alta de conversión (8 o 2 bits más significativos).
- ◉ ADRESL(A/DResultLow): parte baja de conversión (2 u 8 bits menos significativos).
- ◉ ADCON0 (**A/D Control 0**): configura el módulo A/D.
- ◉ ADCON1 (**A/D Control 1**): configura los pines del puerto.
- ◉ TRISA: E/S de puerto A. debe coincidir con ADCON1.
- ◉ PIE1 (**P**eripheral**I**nterrupt**E**nable **1**): controla la habilitación de interrupciones asociadas a los periféricos.
- ◉ PIR1 (**P**eripheral**I**nterrupt**R**egister **1**): contiene bits de estado de interrupción asociados con los periféricos.
- ◉ INTCON (**I**nterrupt **C**ontrol): controla la habilitación general de interrupciones.

PIC – Conversión A/D - Esquema



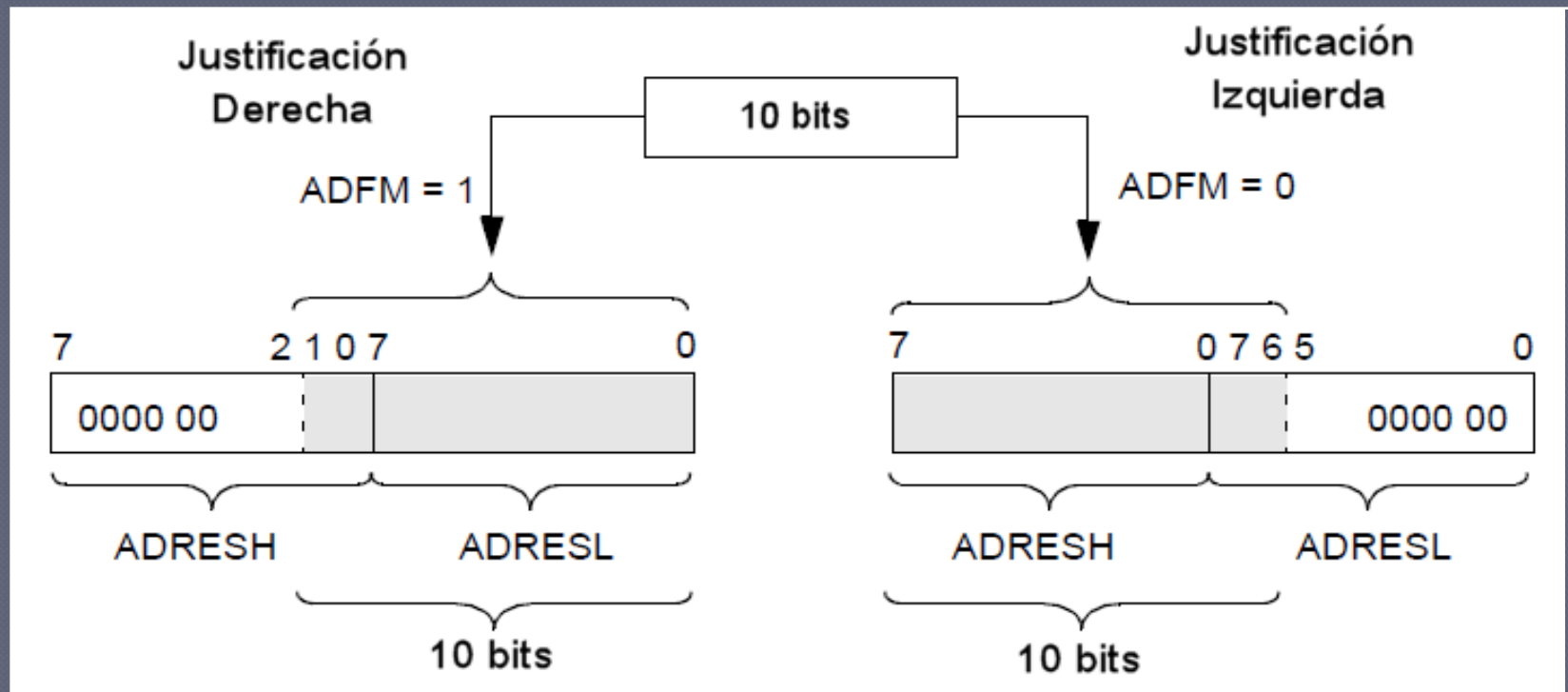
PIC – Conversión A/D – ADCON0

Bit	Nombre	Descripción
7, 6	ADCS1, ADCS0	Bits de selección del reloj para realizar la conversión (A/D Clock Selection): <ul style="list-style-type: none"> • 0:0 = 0 → FOSC/2 (1 ciclo del reloj cada 2 del reloj del sistema). • 0:1 = 1 → FOSC/8 (1 ciclo del reloj cada 8 del reloj del sistema). • 1:0 = 2 → FOSC/32 (1 ciclo del reloj cada 32 del sistema). • 1:1 = 3 → FRC : reloj derivado del oscilador RC interno del modulo A/D
5, 4, 3	CHS2, CHS1, CHS0	Selector del canal A/D para convertir (ChannelSelection). Valores de 0 a 7
2	GO/DONE	Bit de estado y control de la conversión: <ul style="list-style-type: none"> • 1 → conversión A/D en progreso. En “1” para comenzar la conversión. • 0 → no hay conversión A/D . En “0” por hardware al finalizar conversión.
1	-	No implementado. Al leerlo siempre hay un “0”.
0	ADON	Bit de control para activar el módulo de conversión A/D (A/D ON): <ul style="list-style-type: none"> • 1 → El módulo A/D esta activado. • 0 → El módulo A/D esta desactivado. No consume corriente.

PIC – Conversión A/D – ADCON1

Bit	Nombre	Descripción
7	ADFM	Bit de selección para establecer la forma en que se almacenan los 10 bits de la conversión A/D en los registros ADRESH:ADRESL : 1 → Justificar a derecha: 2 en ADRESH y 8 en ADRESL. 0 → Justificar a izquierda: 8 en ADRESH y 2 en ADRESL
6, 5, 4	-	No se utilizan
3, 2, 1, 0	PCFG3, PCFG2, PCFG1, PCFG0	Combinación de bits para configurar el puerto de control A/D. La combinación de estos bits establece la cantidad de entradas analógicas, entradas digitales y entradas que se utilizarán como tensión de referencia (ver siguiente tabla)

PIC – Conversión A/D – Bit ASFM



PIC – Conversión A/D – Bits PCFGs

Valor PCFG	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	V_{REF+}	V_{REF-}	Canales / Ref
0000	A	A	A	A	A	A	A	A	V_{DD}	V_{SS}	8/0
0001	A	A	A	A	V_{REF+}	A	A	A	RA3	V_{SS}	7/1
0010	D	D	D	A	A	A	A	A	V_{DD}	V_{SS}	5/0
0011	D	D	D	A	V_{REF+}	A	A	A	RA3	V_{SS}	4/1
0100	D	D	D	D	A	D	A	A	V_{DD}	V_{SS}	3/0
0101	D	D	D	D	V_{REF+}	D	A	A	RA3	V_{SS}	2/1
011X	D	D	D	D	D	D	D	D	V_{DD}	V_{SS}	0/0
1000	A	A	A	A	V_{REF+}	V_{REF-}	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	RA3	V_{SS}	6/0
1010	D	D	A	A	V_{REF+}	A	A	A	RA3	V_{SS}	5/1
1011	D	D	A	A	V_{REF+}	V_{REF-}	A	A	RA3	RA2	4/2
1100	D	D	D	A	V_{REF+}	V_{REF-}	A	A	RA3	RA2	3/2
1101	D	D	D	D	V_{REF+}	V_{REF-}	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	V_{DD}	V_{SS}	1/0
1111	D	D	D	D	V_{REF+}	V_{REF-}	D	A	RA3	RA2	1/2

Todas Entradas Analógicas

Todas Entradas/Salidas Analógicas

Combinación utilizada en la práctica

PIC – Conversión A/D – PIE1 y PIR1

Bit	Nombre	Descripción
7	-	
6	ADIE	Bit de habilitación de interrupción al finalizar una conversión A/D (A/DInterruptEnabled): <ul style="list-style-type: none">• 1 → habilita interrupción A/D.• 0 → deshabilita interrupción A/D.
5 a 0	-	

Bit	Nombre	Descripción
7	-	
6	ADIF	Bit de interrupción de finalización de conversión A/D (A/DInterruptFlag): <ul style="list-style-type: none">• 1 → conversión A/D completa.• 0 → conversión A/D en curso.
5 a 0	-	

PIC – Conversión A/D - Programación

Configurar el módulo A/D sin interrupción:

- Inicializar el registro ADCON1, seleccionando:
 - la combinación de los pines Analógicos y la tensión de referencia a utilizar (bits PCFG).
 - el formato del resultado (bit ADFM).
- Inicializar el registro TRISA:
 - los bits TRISA deben coincidir con la configuración del registro ADCON1.
- Inicializar el registro ADCON0, seleccionando:
 - la frecuencia del reloj para la conversión (bits ADCS).
 - el canal de entrada para la conversión (bits CHS).
 - la activación del módulo A/D (bit ADON).

PIC – Conversión A/D - Programación

Configurar el módulo A/D con interrupción:

- Seguir los pasos de configuración sin interrupción.
- Configurar el registro PIR1, poniendo en “0” el bit ADIF.
- Configurar el registro PIE1, poniendo en “1” el bit ADIE para habilitar las interrupciones A/D.
- Configurar el registro INTCON, estableciendo:
 - en “1” el bit PEIE para interrupciones de periféricos.
 - en “1” el bit GIE para las interrupciones generales.

PIC – Conversión A/D - Programación

Configurar el módulo A/D con interrupción:

- Seguir los pasos de configuración sin interrupción.
- Configurar el registro PIR1, poniendo en “0” el bit ADIF.
- Configurar el registro PIE1, poniendo en “1” el bit ADIE para habilitar las interrupciones A/D.
- Configurar el registro INTCON, estableciendo:
 - en “1” el bit PEIE para interrupciones de periféricos.
 - en “1” el bit GIE para las interrupciones generales.

PIC – Conversión A/D - Adquisición

- Configurar ADCON0:
 - seleccionar el canal a leer (solo si hay usa más de 1) en los bits CHS.
 - disparar la conversión poniendo en “1” el bit GO.
- Esperar el tiempo necesario de conversión:
 - Sin interrupción: realizando “polling” sobre el bit GO/DONE del puerto ADCON0.
 - Con interrupción: el microcontrolador invoca la rutina de interrupción cuando el bit GO/DONE se pone en “0”. Hay más de 1 dispositivo para interrumpir, se puede consultar el bit ADIF del registro PIR1 verificar que es de conversión A/D.
- Leer los 10 bits de los registros ADRESH:ADRESL
- Solo si habilitó interrupción, poner en “0” el bit ADIF del registro PIR1 para una nueva interrupción