

ARQUITECTURA DE UN PIC

Introducción

El microcontrolador PIC16F84A de la marca Microchip es del tipo de 8 bits con EEPROM tipo flash reforzado encapsulado de 18 patas. El elevado rendimiento de este microprocesador de diseño avanzado permite realizar una gran cantidad de funciones y prestaciones como veremos a continuación.

Características del PIC

Esta familia de microcontroladores PIC se destaca por las siguientes prestaciones especiales:

- Solo se necesitan 35 instrucciones de una sola palabra para usarlo.
- Todas las instrucciones son de un solo ciclo con excepción de las ramificaciones del programa que son de dos ciclos.
- La velocidad de funcionamiento es de DC a 20MHz en la entrada del clock y de DC a 200 ns (nanosegundos) en el ciclo de instrucciones.
- La memoria del programa tiene capacidad para 1024 palabras (1kB).
- La memoria RAM de datos es de 68 bytes.
- La memoria de datos EEPROM es de 64 bytes.
- Las palabras de instrucciones tienen una extensión de 14 bits.
- Los bytes de datos tienen una extensión de 8 bits.
- Posee 15 registros de hardware de funciones especiales.
- Tiene una pila de hardware de ocho niveles.
- Procesa modos de acceso directo, indirecto y relativo.
- Tiene cuatro fuentes de interrupciones:
 - Pata externa RB0/INT
 - Exceso del temporizador TMR0
 - Interrupción al cambio PORTB<7:4>
 - Al completar la escritura del EEPROM de datos.

Características Periféricas

- 13 patas I/O con control individual de dirección.
- Fuente de alta corriente para excitación directa de LED«s»:

Arquitectura de un PIC

-25 mA máximo de sink (nodo) por pata

-25 mA máximo de source (c todo) por pata.

- Temporizador/contador de 8 bits TMR0 con preescaler (divisor de frecuencia) programable de 8 bits.

Características Especiales del Microcontrolador

- Memoria de programa de 1000 ciclos de escritura/borrado de flash reforzado.
- Memoria de datos EEPROM de 1.000.000 de ciclos de borrado/escritura típica.
- Retención de datos de EEPROM de >40 años.
- Programación en serie dentro del circuito (ICSP = In Circuit Serial Programming) mediante dos patas.

- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST).

- Watchdog Timer (WDT) con su propio oscilador RC incorporado en el chip para un funcionamiento confiable.

- Protección de códigos.
- Modo SLEEP para economizar consumo.
- Opciones seleccionables de oscilador.
- Tecnología CMOS con Flash reforzado y EEPROM.
- Tecnología de baja potencia y alta velocidad.
- Diseño estético completo.
- Rango amplio de tensiones de funcionamiento:
 - Comercial: 2,0 Volt a 5,5 Volt
 - Industrial: 2,0 Volt a 5,5 Volt.
- Consumo muy bajo:
 - < 2mA típico a 5 Volt, 4MHz
 - 15 A típico a 2 Volt, 32kHz
 - < 0,5 A típico de corriente en reposo a 2 Volt.

Patas del PIC

El microcontrolador PIC16C84 es un chip que como puede apreciarse en la figura 1, dispone de 18 patas.

La función que cumple cada una de ellas es la siguiente:

Patas 1, 2, 3 17 y 18 - RA0-RA4/TOCK1 : Es el PORT A. Corresponden a 5 líneas bidireccionales de E/S. Es capaz de entregar niveles TTL cuando la tensión de alimentación aplicada en VDD es de 5V – 5%. El pin RA4, si se programa como sa-

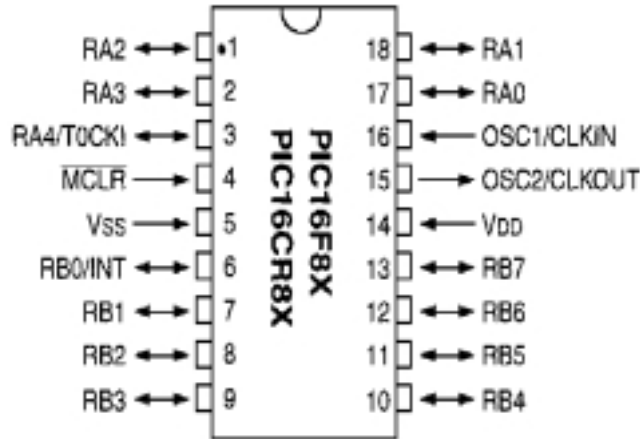


Figura 1

lida es de colector abierto. Como entrada puede programarse en funcionamiento normal o como entrada del contador/temporizador TMR0.

Pata 4 - MCLR / Vpp: Es una pata de múltiples aplicaciones, es la entrada de Reset si está a nivel bajo y también es la habilitación de la tensión de programación cuando se está programando el dispositivo. Cuando su tensión es la de VDD el PIC funciona normalmente.

Patas 5 y 14 - VSS y VDD: Son respectivamente las patas de masa y alimentación. La tensión de alimentación de un PIC está comprendida entre 2V y 6V aunque se recomienda no sobrepasar los 5,5V.

Dependiendo de la letra que posea al final, el PIC será la tensión admisible. Los modelos que contienen las letras C, F o CR admiten tensiones de alimentación comprendidas entre 4V y 4,5V como máximo hasta 5,5V a 6V como máximo. Los modelos que contienen las letras LC, LF o LCR admiten desde 2V a 6V.

Patas 6, 7, 8, 9, 10, 11, 12, 13 - RB0-RB7: Es el PORT B. Corresponden a ocho líneas bidireccionales de E/S. Pueden manejar niveles TTL cuando la tensión de alimentación aplicada en VDD es de 5V – 5%. RB0 puede programarse además como entrada de interrupciones externas INT. Los pines RB4 a RB7 pueden programarse para responder a interrupciones por cambio de estado. Las patas RB6 y RB7 se corresponden con las líneas de entrada de reloj y entrada de datos respectivamente, cuando está en modo programación del integrado.

Patas 15 y 16 - OSC1/CLKIN y OSC2/CLKOUT: Corresponden a los pines de la entrada externa de reloj y salida de oscilador a cristal respectivamente. Los distintos elementos de la familia 16X84X, dependiendo de la nomenclatura que utilizan tienen

Arquitectura de un PIC

distintas características de frecuencia máxima de funcionamiento, tipo de oscilador utilizado para generar frecuencias de reloj y margen de la tensión de alimentación.

Los microcontroladores PIC, permiten cuatro tipos de osciladores externos para aplicarles la frecuencia de funcionamiento. Durante el proceso de grabación, antes de introducir el programa en memoria, debe indicarse el tipo de oscilador empleado en los bits FSOC1 y FSOC2 de la Palabra de Configuración. Los tipos de osciladores que pueden utilizar los PIC16X54/84 son:

* *Oscilador de cristal o resonador de alta velocidad "HS" (High Speed Crystal/Resonator): Es un oscilador con una frecuencia comprendida entre 4MHz y 20MHz (figura 2).*

* *Oscilador o resonador cerámico "XT" (Crystal/Resonator): Es un oscilador estándar que permite una frecuencia de clock máxima de 4MHz.*

* *Oscilador a cristal o resonador cerámico de baja potencia "LP" (Low Power Crystal): Se trata de un oscilador de bajo consumo construido con un cristal de cuarzo o un resonador diseñado para trabajar con frecuencias comprendidas entre 30kHz y 200kHz.*

* *Oscilador RC (resistencia capacitor): También es posible construir un oscilador mediante la colocación de una resistencia y un capacitor externo. Se trata de un oscilador de bajo costo.*

Su circuito de aplicación es el que se muestra en la figura 3. Posee baja precisión la cual depende de la estabilidad de la red RC, pero es fácil de construir y de bajo precio lo que lo hace interesante para muchas aplicaciones. La resistencia debe tener un valor comprendido entre 5k Ω y 10k Ω .

Figura 2

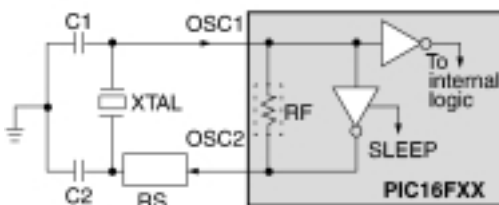
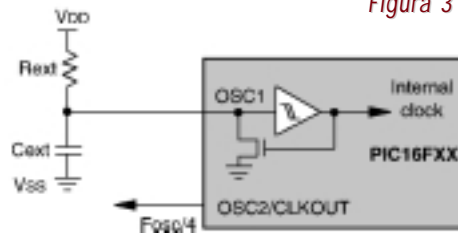


Figura 3



El circuito que opera con un cristal o un resonador se muestra en la figura 2 y depende de los valores de C1, C2 y del cristal para el buen funcionamiento del mismo. Se recomienda ver la tabla de datos del fabricante, aunque para los PIC16X54/84 los capacitores pueden ser del orden de los 27pF para casi todo el rango de frecuencias.

En la figura 3 se muestra un oscilador construido a partir de una celda RC.

Resumen del Dispositivo

El PIC16F84A, utilizado en muchas aplicaciones y que ha servido de base para la elaboración del primer texto de esta serie titulado: **Todo Sobre PICs**, pertenece a la familia de rango medio de los microcontroladores PIC. Un diagrama en bloques se encuentra en la figura 4.

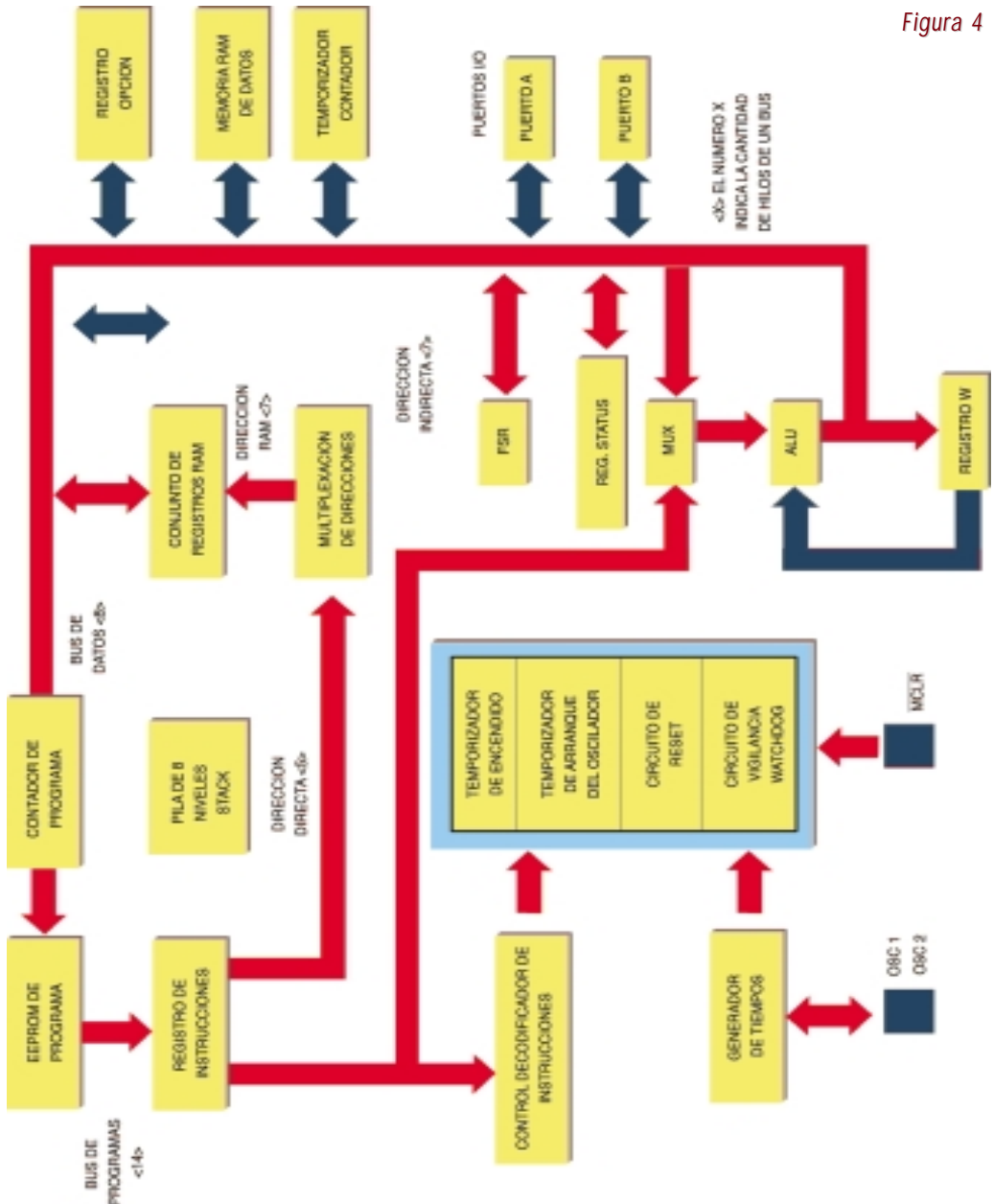


Figura 4

Ficha Didáctica

Antes de avanzar sobre la estructura interna de un PIC, vamos a realizar el primer resumen mediante una ficha tcnica que contiene la descripción interna del circuito integrado.

Los circuitos integrados programables (PIC) de Microchip son microcontroladores con una estructura interna como la mostrada en la figura 4.

Existe un grupo de bloques dedicados a mejorar el funcionamiento pero sin influir directamente en el flujo de señales. Vemos un temporizador de encendido, un temporizador de arranque del oscilador de CLOCK, un circuito de reset y un circuito llamado de vigilancia o WATCHDOG. Los dos primeros bloques procuran un arranque ordenado para no producir una carga al mismo tiempo sobre la fuente. El circuito interno de reset se encarga de volver a "cero" el funcionamiento del circuito cuando sea requerido. Por último, existe un circuito con un nombre curioso, si el lector sabe algo de inglés habrá traducido el nombre literalmente como "perro guardián". Su función es estar vigilante el máximo tiempo que tarda el microprocesador en completar su programa (o mejor sería decir, la derivación más larga de su programa) y en caso de superarse ese tiempo, provocar un reset automático porque el microprocesador se quedó trabado en alguna parte de su programa. También se dice que el microprocesador se quedó colgado o congelado.

Este bloque de circuitos no trabaja independientemente sino que requiere conexiones al exterior y al interior del dispositivo. Por ejemplo, no siempre son utilizados y es el programa quien determina su utilización y además ajusta sus parámetros. Esto se realiza a través del bloque de control o decodificador de instrucciones.

Analicemos ahora la sección de arriba a la izquierda en donde observamos la memoria de programa, el contador de programa, el registro de instrucciones y la pila o STACK de 8 niveles.

Cuando hablamos de registros nos referimos a pequeñas unidades de memoria transitoria, construida por lo general con un registro de desplazamiento. Son memorias volátiles que se utilizan para guardar información por un tiempo mínimo con el fin de realizar una operación compleja de varios pasos.

El contador de programa es el responsable de que el microprocesador vaya analizando las instrucciones en orden ascendente. El guarda el número de instrucción en el STACK y la instrucción misma le pasa registro de instrucciones desde donde se envía al resto del microprocesador.

El STACK es, en realidad, una pila de registros (en nuestro ejemplo hay 8) debido a que el programa puede tener derivaciones (en la jerga LOOPS, rulos o subprogramas). Cuando se termina de ejecutar un loop se debe volver al mismo punto del programa en donde se había producido la bifurcación y eso es posible porque ese número de instrucción quedó guardado en uno de los registros de la pila. Es común que un loop tenga, a su vez, un loop secundario y cuando se ejecuta ese loop secundario se debe volver al mismo punto del loop primario, eso se consigue

guardando ese número de instrucción del loop secundario en otro registro de la pila.

Analicemos ahora la sección inferior derecha. En ese sector se ubican los bloques responsables de efectuar operaciones matemáticas y lógicas binarias; recordemos que el nombre ALU proviene de Aritmetic Logic Unite (unidad aritmética y lógica). En este sector es imprescindible utilizar un registro ya que una operación aritmética o lógica siempre se efectúa entre dos números. Los números binarios que deben procesarse se toman de la memoria de datos, el primero se acumula en el registro de trabajo o registro W (de Work = trabajo) el segundo es el presente en el instante en que se invoca la memoria de datos.

Como las operaciones pueden ser encadenadas (cuando el resultado sirve como operando de la siguiente operación, tal como el caso de un producto) el registro W tiene un retorno a la ALU.

Vemos además que la ALU está comandada por el bloque MUX (MULTipleXador). En efecto, la ALU requiere que se le envíen números para procesar que le lleguen desde la memoria de datos pero antes se la debe predisponer para que efectúe la operación requerida (comparación, rotación de dígitos, etc.).

El registro de estado o estatus colabora durante las operaciones matemáticas.

Piense cómo opera Ud. para realizar una resta: primero ubica el primer número, luego el segundo y después comienza a analizar los bits menos significativos (las unidades), pero si el número de arriba es menor que el número de abajo, entonces toma prestado de la columna de las decenas, luego debe recordar esto porque el número de arriba en la columna de las decenas se redujo en una unidad.

En realidad, aunque se trate de una operación entre dos números su ejecución requiere guardar lo que se llama acarreo en otro registro y este no es otra cosa más que el registro STATUS.

En resumen, las características sobresalientes de los PICs son las siguientes:

Denominación: PIC1XXXX.

Longitud de palabra: 14 bits.

Memoria: EEPROM. Desde 512bytes de ROM y 32bytes de RAM a 4kB de ROM y 256bytes de RAM.

Interface I/O: 13 patas seleccionables de entrada/salida en los modelos estándar.

Arquitectura: Von Neumann (Instrucciones y datos juntos). La mayoría de las instrucciones se ejecutan en un ciclo. Pero posee un set de sólo 35 instrucciones de tipo RISC (Reduced Instruction Set Computer).

Alimentación: 4 a 6V.

Encapsulado: Múltiples versiones, lo normal es DIP de 18 patas.

Fácil de programar, fácil de cargar, extremadamente confiable, económico. Se encuentra gran cantidad de software y hardware sin cargo.

Ideal para Aplicaciones en servicio de equipos, diseños y construcción de "circuitos integrados a medida".

Arquitectura de un PIC

La memoria de programa contiene palabras de 1kB, lo que permite la formación de 1024 instrucciones, ya que cada palabra del programa posee el mismo ancho que cada instrucción. La memoria de datos RAM contiene 68 bytes. La memoria de datos EEPROM contiene 64 bytes.

Además existen 13 patas de I/O que se pueden configurar para el usuario en forma individual. Algunas patas son multiplexadas con otras funciones del dispositivo (tal como vimos anteriormente). Estas funciones incluyen:

- Interrupciones externas.
- Cambios en la interrupción de PORTB.
- Entrada de clock Timer0.

La Tabla 1 muestra las funciones de las patas del dispositivo con las descripciones y detalles de cada una.

Tabla 1-1. Descripción de las patas del PIC16F84A.

Nombre	N° DIP	N° SOIC	N° SSOP	Tipo I/O/P	Tipo buffer	Descripción
OSC1/CLKIN	16	16	18	I	ST/CMOS (3)	Entr. Xtal oscil. (instr. clock ext.)
OSC2/CLKOUT	15	15	19	O	--	SAL. Xtal oscil. (4)
MCLR	4	4	4	I/P	ST	Entr. Master clear
RA0	17	17	19	I/O	TTL	PORTA es un port I/O bidireccional
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	
RB0/INT	6	6	7	I/O	TTL/ST (1)	PORTB es un port I/O bidireccional
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	
RB5	11	11	12	I/O	TTL	
RB6	12	12	13	I/O	TTL/ST (2)	
RB7	13	13	14	I/O	TTL/ST (2)	
Vss	5	5	5,6	P	--	Ref. de masa para las patas lógicas e I/O
Vdd	14	14	15,16	P	--	Fuente positiva para patas lógicas e I/O

Leyendas:

I = Entrada O = Salida I/O = Entrada/Salida
P = Fuente -- = sin usar TTL = Entrada TTL
ST = Entrada Trigger (Disparador) Schmitt

Notas:

- (1) Este buffer es una Entrada de ST cuando se configura como interrupt externo.
- (2) Este buffer es una Entrada de ST cuando se usa en el modo de programación en serie.
- (3) Este buffer es una Entrada de ST cuando está configurado como oscilador RC y Entrada CMOS en las demás configuraciones.
- (4) En el modo de oscilador de cristal conecta al cristal o resonador. En el modo RC, OSC2 tiene como salida CLKOUT, que tiene una frecuencia de 1/4 de OSC1 e indica la tasa del ciclo de instrucciones.

Lo que Debe Saber Sobre los PICs

El PIC16X84 al igual que los demás miembros de su familia, se caracterizan por que:

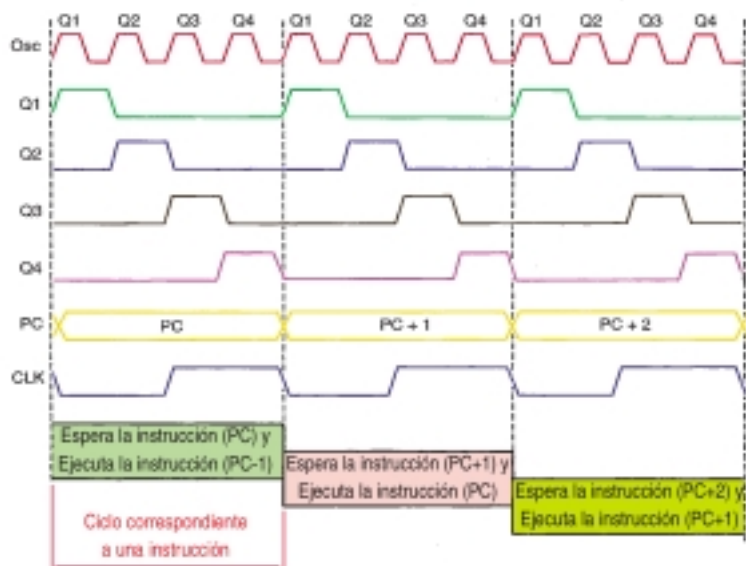
a) El Procesador es segmentado tipo "pipe-line": Lo que significa que aplica la técnica de segmentación que permite al procesador realizar simultáneamente la ejecución de una instrucción y la búsqueda del código de la siguiente instrucción. Esto permite que se pueda ejecutar una instrucción en un ciclo. (Cada ciclo de instrucción son cuatro ciclos de reloj).

Es decir, la mayoría de las instrucciones se realizan en un ciclo de contador de programa (ciclo de instrucción) excepto las instrucciones de salto que necesitan dos ciclos para ejecutarla.

Se determina el ciclo de instrucción dividiendo por cuatro la frecuencia del oscilador, elegida para el funcionamiento del microcontrolador tal como se observa en la figura 5.

Es decir, la señal que proviene del oscilador externo, conectado a los pines OSC1/CLKIN y OSC2/CLKOUT del microcontrolador, se divide en cuatro ciclos, obteniéndose así la señal requerida por el procesador interno para realizar las opera-

Figura 5



Ejemplo:	Acción 1	Acción 2	Acción 3	Acción 4	Acción 5	Acción 6
	Busca 1	Ejecuta 1				
1. MOVLW d'22'		Busca 2	Ejecuta 2			
2. MOVWF PORTB			Busca 3	Ejecuta 3		
3. CALL SUBRU				Busca 4	Salto del programa	
4. BSF PORTA,BIT3					Busca Subrutina	Ejecuta Subrutina

Arquitectura de un PIC

ciones. De esta manera se puede realizar la b squeda y ejecuci n de la instrucc i n.

El reloj de instrucc i n es el ciclo interno que posee el microcontrolador para cronometrar el tiempo de ejecuci n de las instrucciones.

Los pulsos entrantes del reloj son divididos por 4, generando diferentes se ales denominadas Q1, Q2, Q3 y Q4. El estado Q1 hace incrementar el contador de programa, Q2 y Q3, se encargan de la decodificaci n y ejecuci n de la instrucc i n y por ltimo, Q4 es la fase de b squeda de la instrucc i n. El c digo se almacena en el registro de instrucciones.

b) El procesador es de tipo RISC: Esto implica que el P puede operar con un juego de instrucciones m nimo (35 instrucciones). Las CPU«s atendiendo al tipo de instrucciones que utilizan pueden clasificarse en:

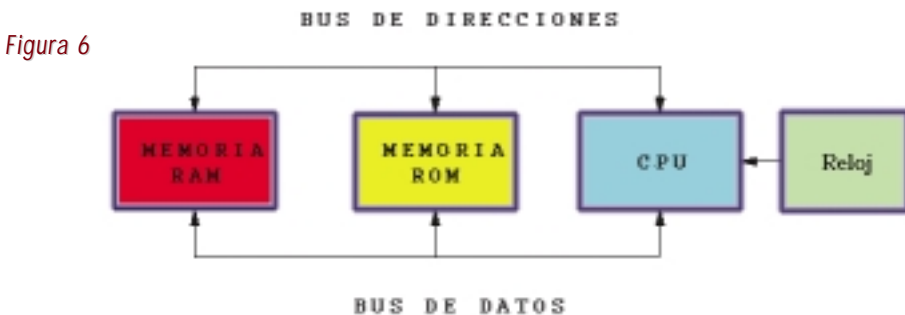
- *CISC: (Complex Instruction Set Computer). Procesadores de juego de instrucciones complejo, que disponen de un elevado n mero de instrucciones (unas 80), algunas de ellas complejas y potentes, pero que requieren muchos ciclos de m quina para ejecutar las instrucciones m s poderosas.*

- *RISC: (Reduced Instruction Set Computer). Controladores que manejan un juego de instrucciones reducido, en los que la cantidad de instrucciones es m nima (en nuestro caso 35). Las instrucciones son muy simples y suelen ejecutarse en un ciclo m quina. Adem s los RISC deben tener una estructura pipeline (vea nuevamente la figura 1) y ejecutan todas las instrucciones a la misma velocidad.*

- *SISC: (Specific Instruction Set Computer). Computadoras que manejan un set de instrucciones espec fico.*

c) Poseen una arquitectura tipo Harvard: Para la construcci n de microprocesadores existen b sicamente dos tipos de arquitecturas, una en la que los datos y la memoria del programa poseen la misma direcci n (denominada Von Neumann) y otra en la que existe un BUS de comunicaci n independiente para la memoria de datos y la memoria de programa (arquitectura Harvard).

La mayor a de los microprocesadores tradicionales se basan en la estructura tipo Von Neumann (figura 6), que se caracteriza por disponer de una nica memoria



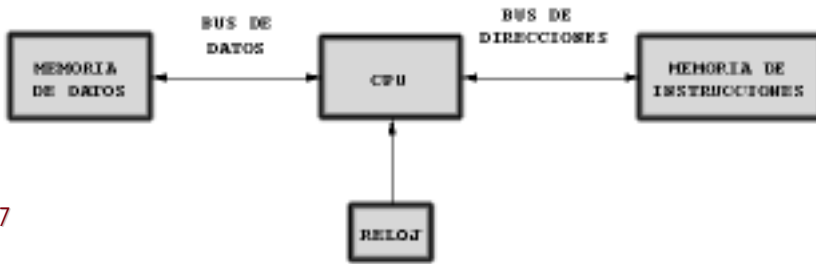


Figura 7

principal en la que se almacenan los datos y las instrucciones, teniendo en cuenta que existe un sistema de BUS de acceso formado por:

- Bus de datos
- Bus de direcciones
- Bus de control

El modelo Harvard, representado en la figura 7, dispone de dos memorias:

- Memoria de datos
- Memoria de Programa

En los procesadores basados en esta arquitectura cada memoria dispone de su respectivo bus, lo que permite, que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos y a la de programa (instrucciones). Como los buses son independientes estos pueden tener distintos contenidos en la misma dirección.

d) Son de arquitectura ortogonal basada en banco de registros: Esto significa que cualquier instrucción puede utilizar cualquier elemento de la arquitectura como fuente o destino. Además todos los elementos del sistema (temporizadores, puertos de entrada/salida, posiciones de memoria, etc.), están implementados físicamente como registros (de ahí el nombre banco de registros). Al tener manejo de banco de registros, la ALU (Unidad Aritmético-Lógica) efectúa sus operaciones con dos operandos, uno que proviene del registro W (Work -que en otras CPUs recibe el nombre de Acumulador-), y el otro que se encuentra en cualquier otro registro del P. En la figura 8 vemos el diagrama que indica cómo se ejecuta una instrucción, la ALU recibe datos del registro de trabajo y un registro MPX y el resultado puede ir a cualquier registro o al registro W.

El PIC 16X54/84 posee características importantes que lo hace útil para el diseño de circuitos generales. Entre las características más importantes podemos resaltar las siguientes:

- Memoria de programa EEPROM de 1Kx14 bits
- Memoria de datos dividida en 2 áreas:

Arquitectura de un PIC

rea RAM formada por 22 registros de prop sito espec fico (SFR) y 36 de prop sito general (GPR).

rea EEPROM formada por 64 bytes.

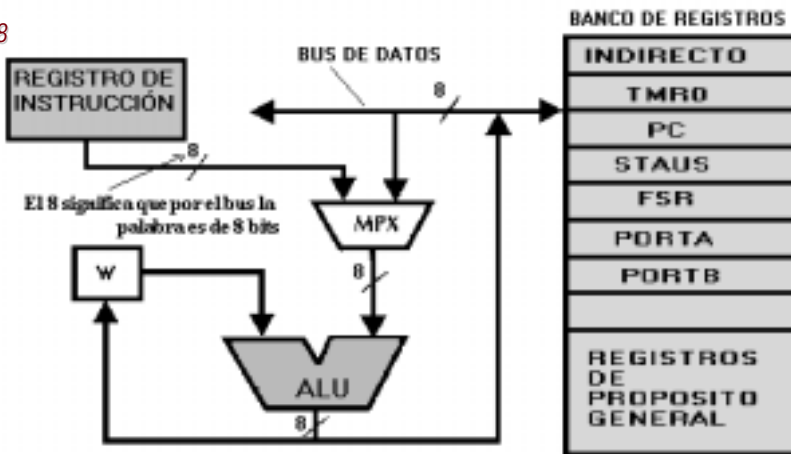
- ALU de 8 bits y registro de trabajo W del que normalmente recibe un operando que puede ser cualquier registro, memoria, puerto de Entrada/Salida o el propio c - digo de instrucc i n.

- Varios canales para conectar al bus de datos: PortA de 5 bits <RA0:RA4> , PortB de 8 bits <RB0:RB7>, Temporizador con Preescaler TMR0, etc.

- Contador de programa de 13 bit (lo que en teor a permitir a direccionar 4kB de memoria, aunque el 16X84 solo dispone de 1kB de memoria implementada).

- Pila de 8 niveles.

Figura 8



La arquitectura del PIC16X84 se mantiene para todos los microcontroladores de esta subfamilia, diferenci ndose unos de otros por las siguientes caracter sticas:

- **PIC 16F84:** La memoria de programa es de 1k, las palabras son de 14 bits, pero de tipo Flash. La memoria de datos RAM tiene 68 registros de tama o byte de prop sito general, en lugar de 36.

- **PIC16CR84:** La memoria de programa es de 1k con palabras de 14 bits tipo ROM y la de datos tiene iguales caracter sticas que el PIC16F84.

- **PIC16F83:** La memoria de programa es de 512 palabras de 14 bits y la RAM de datos tiene 36 bytes de registros de prop sito general.

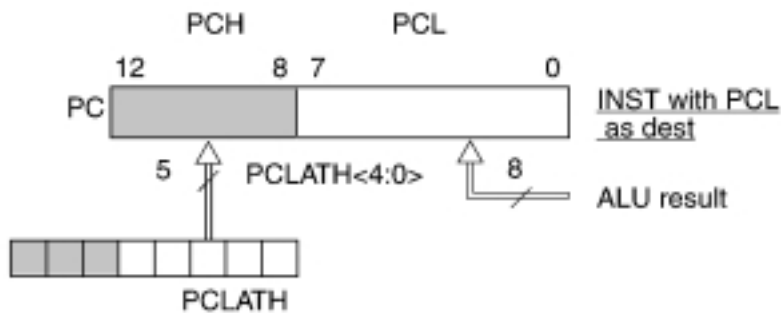
- **PIC16CR83:** Es igual que el PIC16F83, pero la memoria de instrucciones es de tipo ROM, o sea, s lo grabable durante el proceso de fabricaci n para ser utilizados con fines espec ficos (circuito integrado a medida).

La caracter stica m s importante del PIC16C84 es que su memoria de programa es del tipo EEPROM y en el caso del PIC16F84 la principal caracter stica es que su

memoria es del tipo flash, por lo dem s, otros dispositivos de esta familia disponen de m s memoria, tienen m s perif ricos, etc.

Los microcontroladores de la gama media disponen de un Contador de Programa (PC CP) de 13 bits, cuyos bits de menor peso corresponden a los 8 bits del registro PCL, implementado en la posici n de memoria RAM 02h (y duplicado en la posici n 82h),y los cinco bit de mayor peso del CP corresponden con los 5 bits de menor peso del registro PCLATCH, implementado en la posici n de memoria RAM 0Ah (y duplicado en la posici n 8Ah), lo que le permite direccionar hasta 8k x 14 bits dividido en p ginas de 2k x 14 bits. El PIC16C84 dispone de 1k x14 bits de memoria implementada, desde la posici n 0000h hasta la 03FFh, los 3 bit de mayor peso del CP no se utilizan, as pues la direcci n 20h, 320h, 420h, A20h,1420h, etc. se considera como la misma (figura 9).

Figura 9

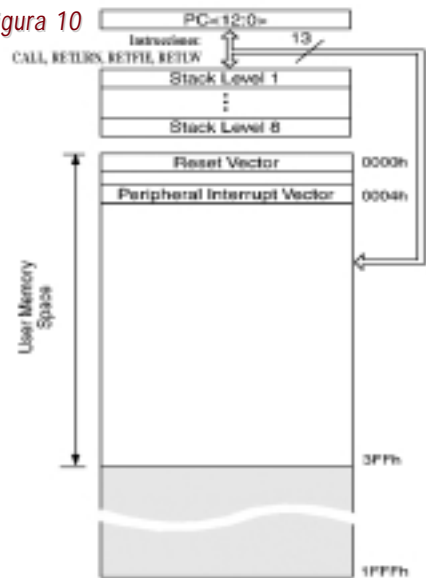


Un Reset provoca que se pongan a cero todos los bits de los registros PCL (parte baja del PC) y PCLATCH, forzando que la direcci n de inicio sea la 0000h. El vector de reset se almacena en la direcci n 0000h, mientras que el vector de interrupci n est en la direcci n 0004h (figura 10).

La memoria de programa de usuario propiamente dicha, comienza en la posici n 0005h y llega hasta la 03FFh.

La pila es una zona de memoria, que se encuentra separada tanto de la memoria de programa como de la de datos. Tiene una estructura LIFO (Last In First Out), lo que significa que el ltimo dato que se guarda es el primero que sale.

Figura 10



Arquitectura de un PIC

Tiene 8 niveles con una longitud de 13 bits cada uno. Su funcionamiento es como el de un buffer circular, de tal forma que el valor que se obtiene al realizar nueve desplazamientos, es igual al primer desplazamiento. La nica manera de cargar la Pila es a trav s de la instrucc i n CALL (llamada a subrutina) o por una interrupci n que hacen que con cada una de ellas, se cargue el contenido del PC en el valor superior de la Pila. Para recuperar el contenido de la Pila en el PC hay que ejecutar una instrucc i n RETURN, RETLW o RETFIE (vuelta del programa de atenci n a una subrutina o interrupci n).

Palabras de Configuraci n e Identificaci n

Los PIC de la gama media disponen de una palabra de configuraci n de 14 bits que se escribe durante el proceso de grabaci n del dispositivo y que debe hacerse de acuerdo con el sistema en el que se va a insertar. Dichos bits ocupan la posici n reservada de memoria de programa 2007h. La estructura de la palabra de configuraci n es la mostrada en la figura 11.

Figura 11

Posici n de Memoria 2007h
Reservada para la palabra de configuraci n

									PC	PWRTE	WDTE	FOSC<1	FOSC<0
Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 4: CP, bit de configuraci n de la protecci n

1= Protecci n de c digo desactivado

0= Protecci n de c digo activado

Bit 3: PWRTE, activaci n del temporizador "Power-Up"

1= Desactivado

0= Activado

Bit 2: WDTE, bit de configuraci n habilitaci n del Watchdog (WDT)

1: WDT activado

0: WDT desactivado

Bit 1 y Bit 0: FOSC<1:0>, tipo de oscilador empleado

11: Oscilador RC

10: Oscilador HS (8 - 20MHz)

01: Oscilador XT (100 KHz- 4MHz)

00: Oscilador LP (Bajo consumo 32- 200Hz)

Adem s dispone de cuatro posiciones de memoria de programa ubicadas en las direcciones <2000h:2003h>, reservadas para las Palabras de Identificaci n ID. Estas palabras que se escriben durante el proceso de grabaci n, s lo emplean los 4 bits de menos peso y se utilizan por el programador para indicar el c digo del dispositivo, el n mero de serie, la versi n del programa, etc.

Organizaci n de la Memoria

En el PIC16F84A existen dos bloques de memoria. Estos son la memoria del programa y la memoria de datos. Cada bloque posee su propio bus, de tal forma que el acceso a cada uno puede producirse durante el mismo ciclo del oscilador.

La memoria de datos puede dividirse m s a n en la RAM de fines generales y los Registros de Funciones Especiales (SFR).

El rea de la memoria de datos contiene tambi n los datos de la memoria EEPROM. Esta memoria no est directamente introducida en la memoria de datos, si no es registrada en forma indirecta. Esto significa que un puntero indirecto de direcciones especifica la direcci n de la memoria de datos EEPROM para escribir y leer. Los 64 bytes de la memoria de datos EEPROM poseen el rango de direcciones de 0h a 3Fh. M s detalles sobre la memoria EEPROM veremos en otro apartado.

Organizaci n de la Memoria de Programas.

El PIC16FXX posee un contador de programas de 13 bits capaz de direccionar un espacio de memoria de programas de 8k x 14. Para el Pic16F84A, los primeros 1k x 14 (0000h-03FFh) est n f sicamente implementados como vimos en la figura 10. Tal como se muestra, el acceso a una locaci n por encima de la direcci n f sicamente implementada, causar un plegado. Por ejemplo, las locaciones 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h y 1C20h tendr n la misma instrucci n.

El vector de reset est en 000h y el vector de interrupt en 0004h.

Nota: Como puede observar, estamos repitiendo algunos conceptos; esto lo hacemos porque tratamos de ver un tema desde distintos puntos de vista con el objeto de facilitar el aprendizaje del lector.

Organizaci n de la Memoria de Datos.

La memoria de datos est repartida en dos reas. El primero es el rea de Registros de Funciones especiales (SFR), mientras que el segundo es el Registro de Fines Generales (GPR). Los registros SFR controlan el funcionamiento del dispositivo. Partes de la memoria de datos est n colocados en bancos. Esto es v lido tanto para el rea de SFR como para el rea GPR. El rea de GPR es apilado para permitir m s que 116 bytes de RAM de uso general. Los reas de apilados del SFR se

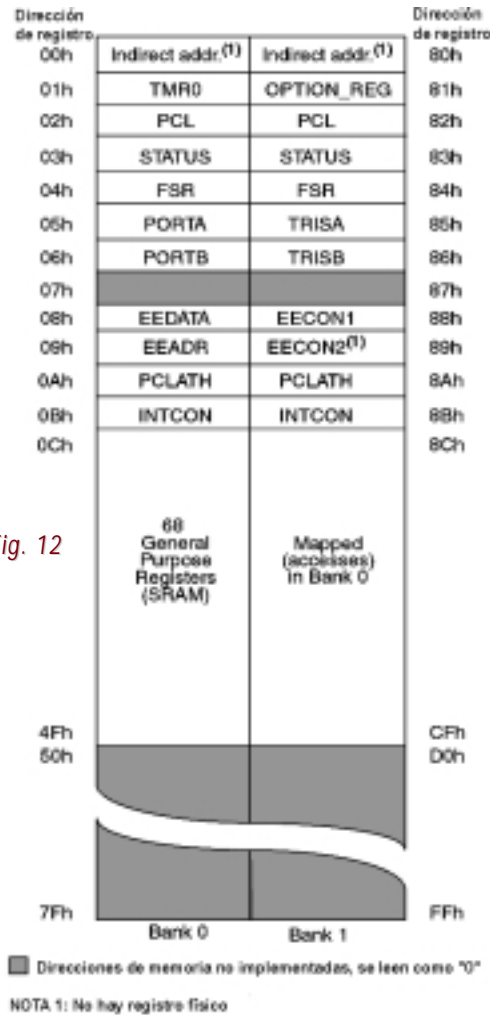
Arquitectura de un PIC

destinan a los registros que controlan las funciones periféricas. El apilado requiere el uso de bits de control para la selección de bancos. Estos bits de control se encuentran en el registro de STATUS. La figura 12 muestra la organización del mapa de datos de la memoria.

Las instrucciones MOVWF y MOVF permiten mover valores del registro W hasta cualquier locación en el archivo de registros (F) y viceversa.

Se puede acceder a la memoria de datos entera en forma directa usando la dirección absoluta de cada registro o en forma indirecta mediante el Registro de Selección de Archivos (FSR). El direccionamiento indirecto usa el bit RP0 para acceder a las áreas apiladas de la memoria de datos.

La memoria de datos está dividida en dos bancos que contienen los registros de uso general y los registros de funciones especiales. El Banco 0 es seleccionado eliminando el bit RP0 (STATUS<5>). El ajuste del bit RP0 selecciona el Banco 1. Cada Banco se extiende hasta 7Fh (128 bytes). Las primeras doce locaciones de cada Banco están reservadas para los Registros de Funciones Especiales (SFR). El resto son Registros de Uso general implementados como RAM.



Los Registros del PIC

Archivo de Registro de Uso General.

Cada Registro de Uso general (GPR) tiene un ancho de 8 bits y se accede en forma directa o indirecta mediante el FSR, como se explica más adelante.

Las direcciones GPR en el banco 1 están conectados con direcciones en el Banco 0. Como ejemplo, la dirección de la locación 0Ch o 8Ch, llamar el mismo GPR.

Registros de Funciones Especiales (SFR).

Los Registros de Funciones especiales de la Figura 13 son usados por la CPU y las funciones Periféricas para controlar el funcionamiento del dispositivo. Estos registros son RAM Estáticos (si bien mantenemos la figura en inglés para preservar el formato dado por los manuales de Microchip, también brindamos la versión en castellano).

Figura 13.A

Dirección	Nombre	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BANCO DE REGISTROS 0									
00h	INDF	Contenido utilizado de FSR a la dirección de memoria de datos (no es un registro físico)							
01h	TMR0	Contador de 8 bit en tiempo real							
02h	PCL	8 bit más bajos del Contador de Programa							
03h	STATUS	IRP	RP1	RP0	TO#	PD#	Z	DC	C
04h	FSR	Puntero indirecto de direccionamiento de datos							
05h	PORTA	--	--	--	RB4/TOCKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
07h		Posición no implementada, se lee como "00"							
08h	EEDATA	Registro de datos EEPROM							
09h	EEADR	Registro de direcciones EEPROM							
0Ah	PCLATH	--	--	--					
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIE
BANCO DE REGISTROS 1									
80h	INDF	Contenido utilizado de FSR a la dirección de memoria de datos (no es un registro físico)							
81h	OPTION	RBPU#	INTEDG	TOSC	TOSE	PSA	PS2	PS1	PS0
82h	PCL	8 bit más bajos del Contador de Programa							
83h	STATUS	IRP	RP1	RP0	TO#	PD#	Z	DC	C
84h	FSR	Puntero indirecto de direccionamiento de datos							
85h	TRISA	--	--	--	Registro de direccionamiento de datos del PORTA				
86h	TRISB	Registro de direccionamiento de datos del PORTB							
87h		Posición no implementada, se lee como "00"							
88h	EECON1	--	--	--	EEIF	WRERR	WRW N	WR	RD
89h	EECON2	Registro de control de EEPROM (no es un registro físico)							
8Ah	PCLATH	--	--	--	Buffer escrito con los 5 bit más altos del PC				
8Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIE

Figura 13.B

Addr	Name	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	Value on Power-on Reset	Value on all other resets (Notes)		
Bank 0													
00h	INDF	Uses contents of FSR to address data memory (not a physical register)									-----	-----	
01h	TMR0	8-bit real-time clock/counter									XXXX XXXX	UUUU UUUU	
02h	PCL	Low order 8 bits of the Program Counter (PC)									0000 0000	0000 0000	
03h	STATUS (2)	IFP	RP1	RP0	T0	PD	Z	DC	C	0001 1000	000q q000		
04h	FSR	Indirect data memory address pointer 0									XXXX XXXX	UUUU UUUU	
05h	PORTA (4)	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---X XXXX	---U UUUU		
06h	PORTB (5)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	XXXX XXXX	UUUU UUUU		
07h		Unimplemented location, read as 0									-----	-----	
08h	EEDATA	EEPROM data register									XXXX XXXX	UUUU UUUU	
09h	EEADR	EEPROM address register									XXXX XXXX	UUUU UUUU	
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC (†)					---	0 0000	---	0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000X	0000 0000		
Bank 1													
80h	INDF	Uses contents of FSR to address data memory (not a physical register)									-----	-----	
81h	OPTION_REG	REPU	INTEDG	T0CS	T0GE	PSA	PS2	PS1	PS0	1111 1111	1111 1111		
82h	PCL	Low order 8 bits of Program Counter (PC)									0000 0000	0000 0000	
83h	STATUS (2)	IFP	RP1	RP0	T0	PD	Z	DC	C	0001 1000	000q q000		
84h	FSR	Indirect data memory address pointer 0									XXXX XXXX	UUUU UUUU	
85h	TRISA	—	—	—	PORTA data direction register					---	1 1111	---	1 1111
86h	TRISB	PORTB data direction register									1111 1111	1111 1111	
87h		Unimplemented location, read as 0									-----	-----	
88h	EEDCON1	—	—	—	EEIF	WREPR	WREN	WR	RD	---0 X000	---0 q000		
89h	EEDCON2	EEPROM control register 2 (not a physical register)									-----	-----	
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC (†)					---	0 0000	---	0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000X	0000 0000		

Los Registros de Funciones Especiales pueden clasificarse en dos grupos:

- a) *n cleo* y
- b) *perif rico*.

Aquellos asociados con las funciones del *n cleo* se describen en esta Secci n. Los relacionados con el funcionamiento de las caracter sticas *perif ricas* se describen en la Secci n respectiva.

Leyendas de la figura 13 A y B:

- x = desconocido,
- u = sin cambio,
- = sin implementar, leer como cero
- q = valor depende de las condiciones.

Nota:

1. El byte superior del contador de programa no es directamente accesible. PCLATH es un registro esclavo para PC<12:8>. El contenido de PCLATH puede ser transferido al byte superior del contador de programa, pero el contenido de PC<12:8> no es transferido nunca a PCLATH.
2. Los bits de Status TO y PD del registro de STATUS no son afectados por el reset de MCLR.
3. Otros resets que no son POWER UP incluyen: reset externo por MCLR y el reset del temporizador de watchdog.
4. En todos los resets del dispositivo, estas patas son configuradas como entradas.
5. Este es el valor que se encontrar en el latch de la salida del port.

Registro de Status

El registro del STATUS contiene el status aritm tico de la ALU, el status de reset y el bit selector de bancos para la memoria de datos.

Como en todo registro, el registro de STATUS puede ser el destino de cualquier instrucc i n. Si el registro de STATUS es el destino de una instrucc i n que afecta los bits Z, DC o C, entonces la escritura de estos tres bits es desactivado. Estos bits son ajustados o anulados en concordancia con la l gica del dispositivo. Adem s, los bits TO y PD (la notaci n TO significa que el bit est activo con 1 o lo que es lo mismo: TO = negado de TO) no pueden ser escritos. Por lo tanto, el resultado de una instrucc i n con el registro de STATUS como destino, puede ser diferente al previsto.

Por ejemplo, CLRF STATUS anular los tres bits superiores y ajustar el bit Z. Esto dejar el registro de STATUS como 000u u1uu (donde u queda sin cambio).

S lo las instrucciones BCF, BSF, SWAPF y MOVWF deben ser usados para al-

Arquitectura de un PIC

terar el registro STATUS, como veremos en el capítulo dedicado al Set de Instrucciones, debido a que estas instrucciones no afectan ningún bit de status.

Notas:

1. Los bits IRP y RP1 (STATUS<:6>) no se usan en el PIC16F84A y deben ser programados como anulados. El uso de estos bits como bits R/W de uso general, no es recomendado, debido a que puede afectar la compatibilidad para arriba con productos futuros.

2. Los bits C y DC funcionan como bits de BORROW (BORROW es el negado de BORROW), en operaciones de resta. Ver ejemplos en las instrucciones de SUBLW y SUBWF del capítulo de Set de Instrucciones.

3. Cuando el registro de STATUS es el destino para una instrucción que afecta los bits Z, DC o C, entonces se impide la escritura de estos tres bits. El bit específico será actualizado en concordancia con la lógica del dispositivo.

En la Figura 14 vemos aspectos del registro de STATUS.

			○	○	○	○	○
R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-X	R/W-X	R/W-X
IRP	RP1	RP0	T0#	PD#	Z	DC	C
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

R = bit de lectura

W = bit de escritura

-X = Valor del bit después de un Reset

○ Flag

Figura 14

Veamos qué significado y qué valor puede tener cada bit del registro STATUS:

Bit 7, IRP: Bit de Selección de Banco de Registro, usado para direccionamiento indirecto. El bit IRP no es usado por el PIC16F84A. Mantenga IRP desactivado.

Bit 6-5, RP1:RP0: Bits de Selección de Banco de Registro, usado para direccionamiento directo.

00 = Banco 0 (00h — 7Fh)

01 = Banco 1 (80h — FFh)

Cada Banco posee 128 bytes. En el PIC16F84A se usa solo RP0. Mantenga.

RP1 se mantendrá desactivado.

Bit 4, TO : Bit de Time out (fin de tiempo).

1 = Después de Encendido, instrucción CLRWDT, o instrucción SLEEP (toma el estado 1).

0 = Cuando est en 0 es porque sucedi un Time out WDT (perro guardi n).

Bit 3, PD: Bit de Power Down.

1 = Despu s de Power up o por la instrucci n CLRWDT.

0 = Al ejecutar la instrucci n SLEEP.

Bit 2, Z: Bit cero.

1 = El resultado de una operaci n aritm tica o l gica es cero.

0 = El resultado de una operaci n aritm tica o l gica no es cero.

Bit 1, DC: Bit de llevar o prestar en las instrucciones ADDWF y ADDLW. Para prestar la polaridad es invertida.

1 = Cuando ocurre un exceso en el resultado del 4... bit de menos peso. Aca-
rreo en la suma.

0 = Cuando no ocurre un exceso en el resultado del 4... bit de menos peso.
No hay acarreo en la suma (en la resta es lo contrario).

Bit 0, C: Bit de carry/borrow (llevar/prestar) para las instrucciones ADDWF y
ADDLW.

1 = Cuando ocurre un exceso en el resultado del bit m s significativo. Aca-
rreo en la suma y no en la resta.

0 = Cuando no ocurre un exceso en el resultado del bit m s significativo.
Acarreo en la resta y no en la suma.

Nota: Para BORROW (prestar d gitos) se invierte la polaridad. Una resta se eje-
cuta al sumar el complemento de dos del segundo operando. Para instrucciones de
rotaci n (RRF, RLF), este bit es cargado con el bit de orden superior o inferior del
registro de fuente.

Recuerde que:

$BORROW = \overline{BORROW}$

$TO = \overline{TO}$

$PD = \overline{PD}$

Registro OPTION_REG.

Ocupa la posici n la 81h del banco de registro 1. Es recomendable darle otro
nombre por ejemplo "OPTION" en el programa ensamblador, para que el programa
ensamblador no d mensajes de error, ya que en los PIC de la gama baja existe la
instrucci n OPTION.

El registro OPTION_REG (o simplemente registro OPTION) es un registro para
escribir y leer que contiene varios bits de control para configurar el prescaler
TMR0/WDT, el interrupt INT externo, el TMR0 y el pull-up d bil de PORTB.

Arquitectura de un PIC

Nota:

Cuando el prescaler es asignado a WDT (PSA = 1), TMR0 tiene una asignación de prescalador de 1:1.

En la figura 15 vemos el Registro de OPTION_REG (dirección 81h).

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RPBU#	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

R = bit de lectura
 W = bit de escritura
 -1 = Valor del bit después de un Reset

Figura 15

bit 7, RPBU : Resistencia Pull-up, Puerto B, habilita el bit
 1: Desactivadas
 0: Activadas

bit 6, INTEDG: Flanco activo para el control de interrupciones
 1: Con flanco Ascendente
 0: Con flanco Descendente

bit 5, TOCS: Fuente de Reloj para TMR0
 1: Pulsos introducidos a través de RA4/T0CK1 (Contador)
 0: Pulsos de reloj interno Fosc/4 (Temporizador)

bit 4, TOSE: Tipo de flanco en T0CK1
 1: Incremento de TMR0 cada flanco descendente
 0: Incremento de TMR0 cada flanco ascendente

bit 3, PSA: Bit de asignación del prescaler divisor de frecuencia
 1: El divisor de frecuencia se asigna al WDT
 0: El divisor de frecuencia se asigna al TMR0

bit 2-0, PS2:PS0: Rango con el que actúa el divisor de frecuencia, este procede de acuerdo con la siguiente tabla:

PS2 PS1 PS0	Divisor del TMR0	Divisor del WDT
0 0 0	1:2	1:1
0 0 1	1:4	1:2

PS2 PS1 PS0	Divisor del TMR0	Divisor del WDT
0 1 0	1:8	1:4
0 1 1	1:16	1:8
1 0 0	1:32	1:16
1 0 1	1:64	1:32
1 1 0	1:128	1:64
1 1 1	1:256	1:128

El Registro INTCON

El registro INTCON es un registro de lectura y escritura que contiene los diferentes bits de activación de todas las fuentes de interrupción. Sus detalles se encuentran en la figura 16.

					0	0	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-X
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

R = bit de lectura
 W = bit de escritura
 -X = Valor del bit después de un Reset

0 - Flag

Figura 16

Las siglas de esta figura significan lo siguiente:

- R = Bit Legible
- W = Bit Escribible
- U = Bit sin implementar, leer como cero
- X = valor en reset POR.

Cada bit posee el significado y toma el valor que describimos a continuación:

Bit 7, GIE: Bit Global Interrupt Enable.

- 1 = Activa todos los Interrupt sin máscara.
- 0 = Desactiva todos los Interrupt.

Bit 6, EEIE: Bit EE Write Complete Interrupt Enable.

- 1 = Activa el EE Write Complete Interrupt.
- 0 = Desactiva el EE Write Complete Interrupt.

Bit 5, TOIE: Bit de TMR0 Overflow Interrupt Enable.

1 = Activa el TMR0 Interrupt.
0 = Desactiva el TMR0 Interrupt.

Bit 4, INTE: Bit de Interrupt Enable RB0/INT.

1 = Activa el Interrupt RB0/INT.
0 = Desactiva el Interrupt RB0/INT.

Bit 3, RBIE: Bit de RB Port Change Interrupt Enable.

1 = Activa el RB Port Change Interrupt.
0 = Desactiva el RB Port Change Interrupt.

Bit 2, T0IF: Bit de TMR0 Overflow Interrupt Flag.

1 = TMR0 tiene exceso y debe ser limpiado en software.
0 = TMR0 no tiene exceso.

Bit 1, INTF: Bit de RB0/INT Interrupt Flag.

1 = Ocurri un RB0/INT Interrupt.
0 = No ocurri un RB0/INT Interrupt.

Bit 0, RBIF: Bit de RB Port Change Interrupt Flag.

1 = Cuando por lo menos una de las patas de RB7 a RB4 cambi (debe ser limpiado en software).
0 = Ninguna de las patas RB7:RB4 cambi .

PCL y PCLATH

El contador de programas (PC de program counter , tambi n denominado CP en este texto) especifica la direcci n de la instrucc i n a buscar para la ejecuci n. El ancho del PC es de 13 bits. El byte bajo se denomina Registro PCL. Este Registro es legible y escribible. El byte alto se denomina Registro PCH. Este Registro contiene PC<12:8> bits y no es legible ni escribible en forma directa. Todas las actualizaciones del Registro PCH se canalizan a trav s del Registro PCLATH.

STACK (PILA)

La pila permite que ocurra una combinaci n de hasta 8 llamadas de programa e interrupciones. La pila contiene la direcci n de retorno de esta rama de la ejecuci n de programas.

Los dispositivos de rango medio poseen una pila de hardware de una profundidad de 8 niveles por 13 bits de ancho. El espacio de la pila no es parte de ning n programa ni de espacio de datos y el puntero de la pila no es legible ni escribible. El contador PC (CP) es colocado a la pila v a PUSH cuando una instrucc i n CALL es ejecutada o cuando una orden de interrupt produce una ramificaci n. La pila ejecu-

ta la orden POP cuando se ejecutan las rdenes RETURN, RETLW o RETFIE. PCLATH no se modifica cuando la pila ejecuta PUSH o POP.

Despu s de ejecutar ocho veces la orden PUSH, la novena sobrescribe el valor que fuera almacenado del primer PUSH. El d cimo PUSH sobrescribe el segundo PUSH y as sucesivamente.

REGISTROS INDF y FSR

El registro INDF no es un registro f sico. Dirigirse a INDF en realidad produce la direcci n del registro que est contenido en el registro FSR. FSR es un puntero. Esto se denomina Direccionamiento Indirecto.

EJEMPLOS:

1: Direccionamiento Indirecto.

- El archivo de registro 05 contiene el valor 10h.
- El archivo de registro 06 contiene el valor 0Ah.
- Cargue el valor 05 al registro FSR.
- La lectura del registro INDF devuelve el valor de 10h.
- Incremente el valor del registro INDF por uno (FSR = 06).
- La lectura del registro INDF devuelve ahora el valor de 0Ah.

La lectura indirecta de INDF mismo (FSR = 0), producir 00h. Escribir al registro indirectamente resulta en una no-operaci n, si bien el bit de STATUS puede quedar afectado.

Un programa simple para desactivar las locaciones 20h-2Fh del RAM, usando el direccionamiento indirecto, puede verse en el siguiente ejemplo

	Movlw	0x22	; cargo al registro W con el n mero 22 ; en hexadecimal
	Movwf	FSR	; transfiero el contenido de W al registro FSR
PROX	Clrf	INDF	; borro el contenido del registro INDF
	Incf	FSR	; incremento en una unidad el contenido ; del registro FSR
	Btfss	FSR,4	; que el CP salte una unidad si el bit 4 ; del FSR es 1
	Goto	PROX	; si el bit 4 de FSR = 0, entonces vuelva ; a PROX

Una direcci n efectiva de 9 bits se obtiene por la concatenaci n del registro FSR de ocho bits con el bit IRP (STATUS<7>), como vemos en la figura 17.

Sin embargo, IRP no es usado en el PIC16F84A.

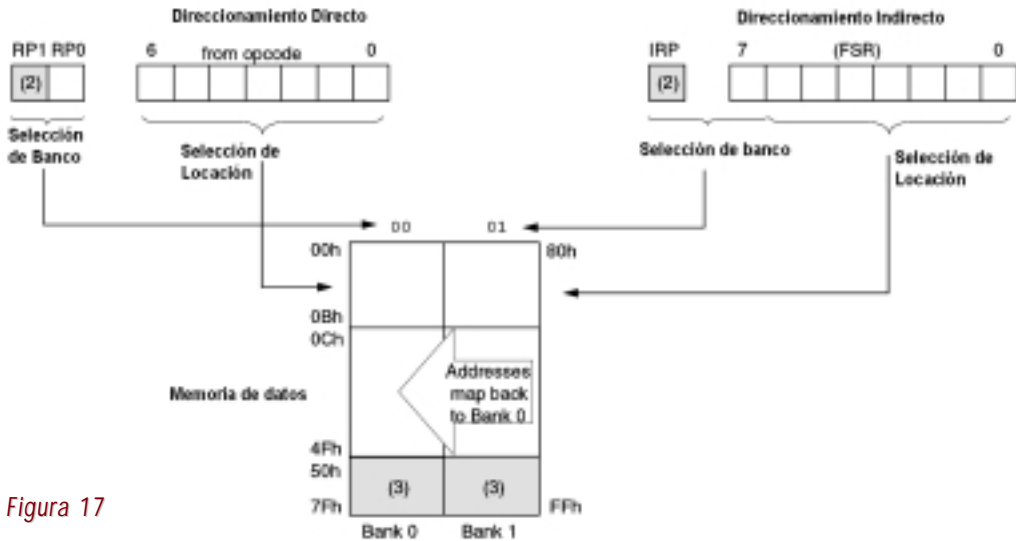


Figura 17

PORTS I/O

Algunas patas de ports I/O están multiplexadas con una función alternativa para las prestaciones periféricas del dispositivo. En general, cuando se activa un periférico, esta pata no debe usarse como pata I/O de uso general.

Registros PORTA y TRISA

PORTA es un puerto bidireccional de un ancho de 5 bits. El registro de direcciones de datos correspondiente es TRISA. Ajustar el bit de TRISA a 1, hará de la pata correspondiente de PORTA una entrada, lo que significa que el driver de salida correspondiente adquiere un modo de alta impedancia. Desactivar un bit de TRISA (= 0) convierte la pata correspondiente de PORTA en salida, lo que significa que el contenido del latch de salida se encuentra en la pata seleccionada.

NOTA: En un Reset de Fuente (POWER RESET), estas patas son configuradas como entradas y leen como 0.

En la lectura, el registro de PORTA lee el status de las patas, mientras que al escribir, se escribe en el latch del port.

Un latch o cerrojo es un circuito que permite retener datos en una posición de preparación hasta que sea pedido, generalmente por medio de otro circuito.

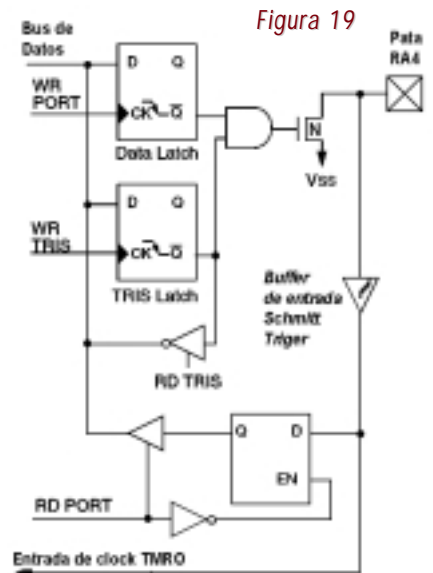
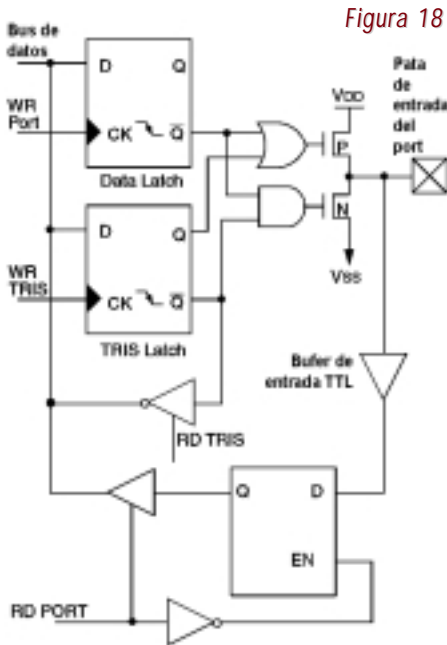
Todas las operaciones de escritura son del tipo de *lectura-modificación-escritura*. Esto significa que escribir a un port implica que las patas del port son leídas, es-

te valor es modificado, y despu s escrito al latch del port. La pata RA4 es multiplexada con el temporizador, *entrada Timer0* de la entrada del m dulo del clock y se transforma en la pata RA4/T0CKI. Esta pata RA4/T0CKI es la entrada de un disparador Schmitt y posee una salida de colector abierto. Las dem s patas de port RA poseen niveles de entrada TTL y son drivers completos del tipo CMOS. En la figura 18 vemos el diagrama en bloques de una de las patas RA3 a RA0.

Para inicializar el PORTA, podemos realizar la siguiente rutina:

```

CLRF    PORTA           ; borro el contenido del registro PORTA
BSF     STATUS, RPO    ; pone a 1 el bit RPO del registro STATUS
MOVLW  B 0000 1111    ; cargo a W con el n mero 0x0F
MOVWF  TRISA           ; selecciono a las patas RA3 a RA0 como salidas
                           ; y la pata RA4 como entrada.
    
```



En el ejemplo del programa, las dos primeras instrucciones no tienen nada que ver con la definición de las patas del puerto A, en realidad la inicialización comienza con la tercera instrucción o línea de programa. En la figura 19 se observa el diagrama de la pata RA4.

Una síntesis de la función que cumple cada pata del PORT se muestra en la siguiente tabla:

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Entrada/Salida
RA1	bit1	TTL	Entrada/Salida
RA2	bit2	TTL	Entrada/Salida
RA3	bit3	TTL	Entrada/Salida
RA4/TOCKI	bit4	ST	Entrada/Salida = Clock externo para el TMR0 La salida es tipo colector abierto

Legend: TTL = TTL input, ST = Schmitt Trigger input

Tabla de Registros Asociados

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor sobre Power-on Reset	Valor sobre los demás Resets
09h	PORTA	—	—	—	RA4/TOCKI	RA3	RA2	RA1	RA0	---x xxxx	---0 xxxx
09h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---0 1111	---0 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are unimplemented, read as '0'

Registros PORTB y TRISB

PORTB es un port bidireccional con un ancho de 8 bits. El registro de direcciones de datos correspondiente es TRISB. Si se ajusta el bit TRISB = 1, la pata PORTB correspondiente se transforma en entrada, quiere decir que coloca el driver de salida correspondiente en el modo de alta impedancia.

Limpiar el bit TRISB (= 0), transforma la pata correspondiente de PORTB en salida, quiere decir que el contenido del latch de salida aparece en la pata seleccionada.

Para inicializar el PORTB podemos ejecutar la siguiente sentencia:

```

MOVLW    B 0010 1111    ; cargo a W con el número 0x0F
MOVWF    TRISB          ; selecciono las patas RB0 a RB3 y RB5 como salidas,
                        ; mientras que las patas RB4, RB6 y RB7
                        ; ser n entradas.
    
```

Cada una de las patas de PORTB posee un resistor de polarización interno. Un bit de control único puede activar todos los resistores. Se realiza esto limpiando el bit RBPU (OPTION<7>).

Este circuito es desactivado automáticamente cuando la pata del port es configurada como salida. Estos circuitos son desactivados en un reset de encendido.

Cuatro de las patas de PORTB, RB7:RB4, poseen una característica de interrupt al cambiar. Solo las patas configuradas como entradas pueden producir este interrupt.

Esto significa que toda pata RB7 a RB4 configurada como salida está excluida de la comparación de interrupt al cambiar. Las patas de entrada de RB7 a RB4, se comparan con el valor anterior asentado en el latch de la última lectura de PORTB. Las salidas de desequilibrio de RB7 a RB4 son combinadas en una función lógicamente OR para generar el cambio del RB Port con el bit de indicación RBIF (INTCON<0>) como flag.

Esta interrupción puede despertar el dispositivo de SLEEP. El usuario puede limpiar el interrupt en la rutina de service en la siguiente manera:

- a) Toda lectura o escritura de PORTB. Esto termina con la condición de desequilibrio.
- b) Limpie el bit del flag RBIF.

Una condición de desequilibrio hará ajustar el bit de flag RBIF. La lectura de PORTB termina con la condición de desequilibrio y permite la limpieza del bit RBIF. La prestación de interrumpir al cambiar es recomendada para operaciones de despertar al oprimir una tecla y para operaciones donde PORTB solo es usado para la prestación de interrupt al cambiar. No se recomienda el escrutinio (polling) de PORTB mientras se usa la prestación de interrupt al cambio. En la figura 20 se puede observar el diagrama en bloque correspondiente al sistema de patas RB4 a RB7, mientras que la figura 21 muestra lo propio para las patas RB0 a RB3.

Figura 20

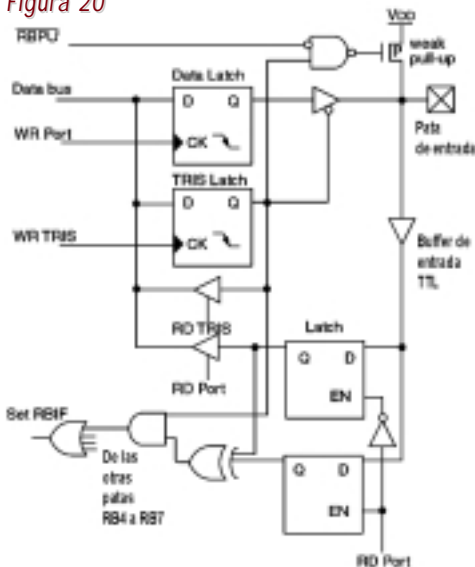
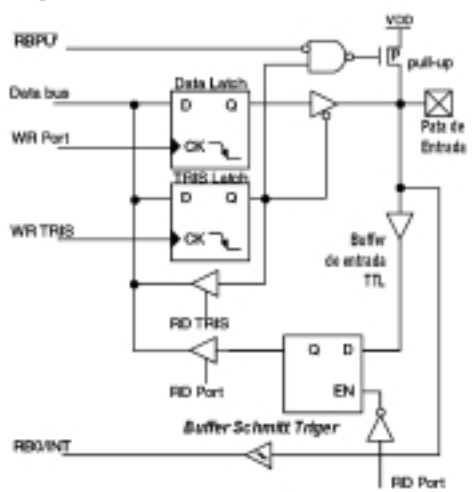


Figura 21



En la siguiente tabla vemos las funciones de PORTB.

Nombre	Bit	Tipo de Buffer	Función
RB0/INT	bit0	TTL/ST	Pata de entrada / salida o interrupción externa Programable internamente para pull-up
RB1	bit1	TTL	Pata de entrada / salida. Programable internamente para pull-up
RB2	bit2	TTL	Pata de entrada / salida. Programable internamente para pull-up
RB3	bit3	TTL	Pata de entrada / salida. Programable internamente para pull-up
RB4	bit4	TTL	Pata de entrada / salida (con cambio por interrupción). Programable internamente para pull-up
RB5	bit5	TTL	Pata de entrada / salida (con cambio por interrupción). Programable internamente para pull-up

Arquitectura de un PIC

Nombre	Bit	Tip de Buffer	Función
RB6	bit6	TTL/ST ⁽²⁾	Pata de entrada / salida (con cambio por interrupción). Programable internamente para pull-up. Clock para prog. serie
RB7	bit7	TTL/ST ⁽²⁾	Pata de entrada / salida (con cambio por interrupción). Programable internamente para pull-up. Dato para prog. serie

Registros Asociados

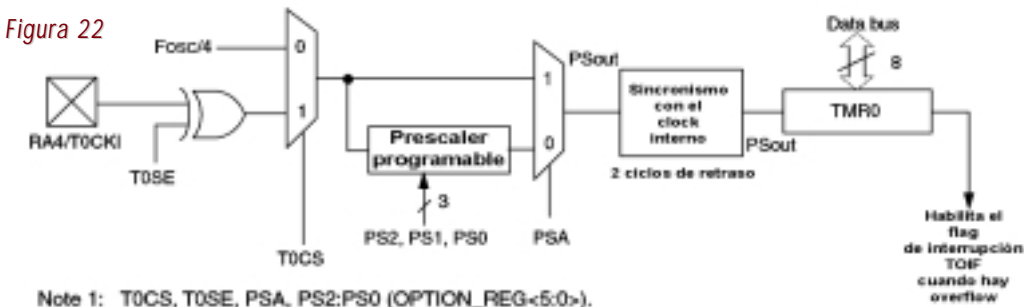
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	XXXX XXXX	XXXX XXXX
08h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
8th	OPTION_REG	RBP0	INTE03	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Módulo Temporizador TIMER0

El módulo temporizador/contador posee las siguientes prestaciones:

- Temporizador/contador de 8 bits.
- Legible y escribible.
- Selector de clock interno o externo.
- Prescalador de 8 bits programable por software.
- Interrupt por exceso de FFh a 00h.

En la figura 22 vemos un diagrama en bloques simplificado del temporizador.



El temporizador TIMER0 puede funcionar como temporizador o como contador.

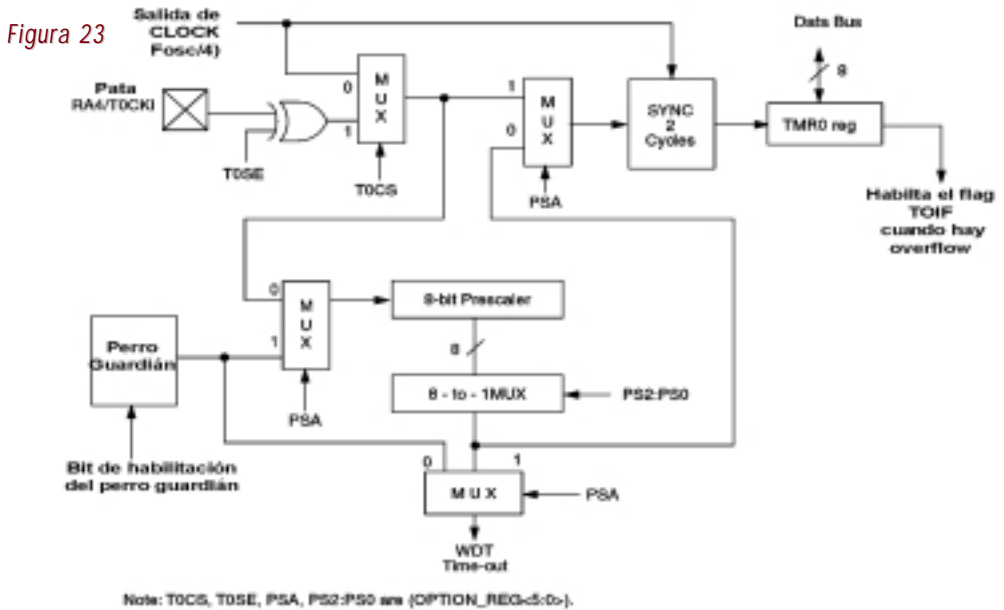
El modo de temporizador se selecciona limpiando el bit T0CS (OPTION_REG<5>). En el modo de temporizador, el módulo Timer0 incrementa cada ciclo de instrucciones sin división de frecuencia. Si se escribe en el registro TMR0, el incremento es inhibido durante los siguientes ciclos de instrucciones. El usuario puede evitar eso escribiendo un valor ajustado al registro TMR0.

El modo de contador se selecciona al ajustar el bit T0CS (OPTION_REG<5>). En el modo de contador, el módulo Timer0 incrementa en el flanco ascendente o descendente de la pata RA4/T0CKI. El flanco ascendente es determinado por el bit Timer Source Edge Select, T0SE (OPTION_REG<4>). Limpiando el bit T0SE selecciona el borde ascendente.

Cuando se usa una entrada de clock externo en el Timer0, se deben cumplir ciertos requisitos para que el clock externo pueda ser sincronizado con la fase del clock interno (Tosc). Además existe un retardo en el incremento real del Timer0 después de la sincronización.

Prescaler (Divisor de Frecuencia)

Se dispone de un contador de 8 bits como prescaler en el módulo Timer0, o como post escalador para el temporizador Watchdog, respectivamente. En la Figura 23 vemos este esquema.



Para mayor simplicidad, nos referimos a este contador como prescaler. Observe que solo existe un divisor que es compartido mutuamente en forma exclusiva por el módulo del Temporizador y el Temporizador del Watchdog. De esta manera, una asignación del prescaler para el módulo del temporizador significa que no hay divisor para el Temporizador del Watchdog y viceversa.

El prescaler no es legible ni escribible.

Los bits PSA y PS2:PS0 (OPTION_REG<3:0>) determinan la asignación y la ta-

sa de la divisi n. Al limpiar el bit PSA se asignar el prescaler al m dulo del Timer0. Cuando ste es asignado al m dulo Timer0, los valores de divisi n son seleccionables en 1:2, 1:4,..... 1:256.

El ajuste del bit PSA asignar el prescaler al Temporizador Watchdog (WDT). Cuando el prescaler es asignado a WDT, los valores de divisi n son seleccionables en 1:1, 1:2, 1:128.

Cuando est asignado al m dulo Timer0, todas las instrucciones escritas al registro TMR0, como por ejemplo: CLRF 1, MOVWF 1, BSF 1,x ...y otros, limpiar n el prescaler. Cuando est asignado al WDT, una instrucci n CLRWDT limpiar el prescaler junto con WDT.

NOTA: Escribir a TMR0 cuando el prescaler est asignado a Timer0, limpiar el conteo del prescaler, pero no modificar su asignaci n.

Cabe aclarar que la asignaci n del prescaler est completamente bajo el control del software y por lo tanto puede ser cambiado durante la ejecuci n del programa.

NOTA: Para evitar un RESET no intencional del dispositivo, es necesario ejecutar una secuencia de instrucciones espec ficas cuando se cambia la asignaci n del prescaler del Timer0 a WDT. Esta secuencia debe ser seguida a n cuando WDT est inactivo.

Interrupci n de Timer0

La interrupci n de TMR0 se genera cuando el registro TMR0 excede su capacidad de FFh a 00h. Este exceso ajusta el bit T0IF (INTCON<2>). La interrupci n puede ser enmascarada limpiando el bit T0IE (INTCON<5>). El bit T0IF se limpia por software por la rutina de service del m dulo Timer0, antes de reactivar esta interrupci n. La interrupci n TMR0 no puede despertar al procesador de SLEEP, debido a que el temporizador est desconectado durante SLEEP.

En la siguiente tabla vemos los registros asociados con el Timer.

Registros Asociados con el TIMER0

Address	Name	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	Value on POR, BOR	Value on all other resets
01h	TMR0	Timer0 module's register								XXXX XXXX	XXXX XXXX
08h,8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION_REG	RSPO	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

Memoria de Datos EEPROM

La memoria de datos EEPROM se puede leer y escribir durante el funcionamiento normal del P, para el rango completo de Vdd. Esta memoria no est ordenada

en forma de mapa directamente en el espacio de los archivos del registro sino que es direccionada indirectamente mediante los Registros de Funciones Especiales (SFR). Existen cuatro SFR usados para leer y escribir en esta memoria. Estos registros son:

- EECON1
- EECON2 (este registro no est implementado f sicamente)
- EEDATA
- EEADR

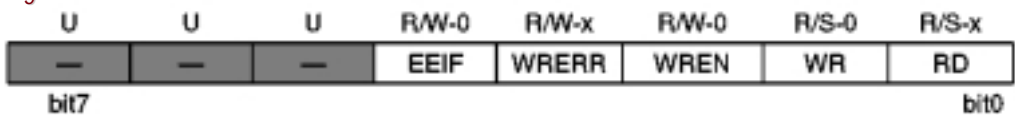
EEDATA contiene los datos de 8 bits para leer y escribir, y EEADR contiene las direcciones de las locaciones del EEPROM que son accedidos. Los dispositivos PIC16F84A poseen 64 bytes de datos de EEPROM con un rango de direcciones que abarca de 0h a 3Fh.

La memoria de datos EEPROM permite leer y escribir bytes. Una escritura de byte borra autom ticamente la locaci n y escribe los datos nuevos (borra antes de escribir). La memoria de datos EEPROM est preparada para ciclos elevados de leer y escribir. El tiempo de escribir es controlado por un temporizador incorporado en el chip. El tiempo de escritura puede variar con la tensi n y temperatura y tambi n entre una unidad y otra. Para l mites exactos s rvase referir a las especificaciones de corriente alterna.

Cuando el dispositivo posee protecci n codificada, la unidad CPU puede continuar de leer y escribir los datos en la memoria del EEPROM. El programador del dispositivo no tiene m s acceso a la memoria.

En la figura 24 vemos el registro EECON1 en su direcci n 88h.

Figura 24



Las siglas de esta figura tienen el siguiente significado:

- R = Bit Legible
- W = Bit Escribible
- S = Bit Ajustable
- U = Sin Implementar, lea como cero,
- n = Valor de acuerdo al Reset POR (donde n puede ser x, o).

El significado y estado de los bits de la memoria son los siguientes:

Bits 7, 6 y 5: Sin Implementar, lea como cero.

Bit 4, EEIF: Bit indicador de EEPROM Write Operation Interrupt (FLAG).

1 = La operaci n de escritura est cumplida, debe ser borrado en el software.

0 = La operaci n de escribir no est cumplida o no empez a n.

Bit 3, WRERR: Bit indicador de Error en EEPROM.

1 = Una operaci n de escribir termin en forma prematura. Causado por un RESET MCLR o un RESET WDT durante el funcionamiento normal.

0 = La operaci n de escribir est terminada.

Bit 2, WREN: Bit de Activar la Escritura.

1 = Permite ciclos de escritura.

0 = Inhibe la escritura al EEPROM.

Bit 1, WR: Bit de Control de Escritura.

1 = Inicia un ciclo de escritura. El bit es anulado por hardware una vez terminada la operaci n. El bit WR s lo puede ser ajustado por software, pero no borrado.

0 = El ciclo de escritura al EEPROM est completo.

Bit 0, RD: Bit de Control de Lectura.

1 = Inicia una lectura del EEPROM. La lectura ocupa s lo un ciclo.

RD es borrado con hardware. El bit RD s lo puede ser ajustado por software, pero no borrado.

0 = No comienza una lectura del EEPROM.

Lectura de la Memoria de Datos de EEPROM

Para leer una locaci n de memoria de datos, el usuario debe escribir la direcci n al registro EEADR y despu s ajustar el bit de control RD (EECON1<0>). El dato est disponible en el siguiente ciclo en el registro EEDATA y por lo tanto puede ser le do en la instrucc i siguiente. EEDATA conservar este valor hasta otra lectura o hasta que sea escrito por el usuario durante una operaci n de escribir.

Ejemplo:

Lectura de EEPROM de Datos. Veamos como podr a ser la secuencia de un programa para leer una locaci n de memoria:

BCF	STATUS, RPO	; pone a cero el bit 5 (RPO) del registro STATUS, ; dicho bit se usa para direccionamiento directo y, ; en este caso, elegimos el banco 0
MOVLW	0x20	; cargo al registro W con la direcci n de la memoria ; de datos que voy a leer
MOVWF	EEADR	; muevo el contenido del registro W al registro ; EEADR que es el registro de direccionamiento de ; la EEPROM, dicho de otra manera, estoy ; realizando el direccionamiento para leer

BSF	STATUS, RPO	; pone a uno el bit 5 (RPO) del registro STATUS, ; dicho bit se usa para direccionamiento directo y, ; en este caso, elegimos el banco 1
BSF	EECON1, RD	; pone a uno el bit 0 (RD) de control de lectura del ; registro de la memoria de datos (EECON1), esto ; significa que se inicia la lectura del dato ; contenido en la direcci n que hemos ; cargado en EEADR
BCF	STATUS, RPO	; pone a cero el bit 5 (RPO) del registro STATUS, ; dicho bit se usa para direccionamiento directo y, ; en este caso, elegimos el banco 0
MOVF	EEDATA, W	; como en la instrucci n anterior dijimos que ; har amos un direccionamiento directo y EEDATA ; es el registro de datos donde est la informaci n ; le da, ahora, con la instrucci n MOVF hemos ; cargado al registro de trabajo con el dato le do ; de la memoria de datos.

Escritura de la Memoria de Datos EEPROM

Para escribir una locaci n de memoria de datos EEPROM, el usuario debe escribir la direcci n primero al registro EEADR y los datos al registro EEDATA. Despues el usuario debe seguir una secuencia espec fica para iniciar la escritura de cada byte. Cabe aclarar que si deseo leer o escribir una direcci n de memoria tambi n es posible nombrar una variable , tema que analizaremos m s adelante.

Ejemplo

Escritura de EEPROM de Datos. Veamos c mo podr a ser la secuencia de un programa para escribir una locaci n de memoria:

BSF	STATUS, RPO	; pone a uno el bit 5 (RPO) del registro STATUS, ; dicho bit se usa para direccionamiento directo y, ; en este caso, elegimos el banco 1
MOVLW	0x18	; cargo al registro de trabajo con la direcci n ; de la memoria donde escribir el dato
MOVWF	EEADR	; en la direcci n de registro de escritura ; coloco la direcci n donde escribir el dato
MOVLW	0x12	; cargo a W con el n mero 12 en hexadecimal que ; ser el dato que voy a escribir en la posici n 18h
MOVWF	EEDATA	; escribimos el dato en EEDATA

Arquitectura de un PIC

	BSF	EECON1,WREN	; solicito el permiso de escritura y ; comienza la secuencia de escritura	
Secuencia requerida para realizar la escritura	MOVLW	0x55	; cargo a W con 55h	
	MOVWF	EECON2	; se escribe el dato 12h en EECON2	
	MOVLW	0xAA	; cargo a W con AAh	
	MOVWF	EECON2	; se escribe AAh en EECON2	
	BSF	EECON1, WR	; comienza la escritura	
	WI	BTFSC	EECON1,WR	; espera a que termine la escritura
		GOTO	WI	;
		BCF	STATUS,R0	; selecci n del banco 0

La escritura no se inicia si las secuencias arriba mencionadas no se cumplen exactamente para cada byte. Escribir 55h a EECON2, escribir AAh a EECON2, despu s ajustar el bit WR. Recomendamos especialmente desactivar las interrupciones durante este segmento del c digo.

Adem s, el bit WREN en EECON1 debe ser ajustado para activar la escritura. Este mecanismo impide la escritura accidental al EEPROM de datos debido a inesperadas ejecuciones del c digo, por ejemplo por programas perdidos. El usuario debe mantener el bit WREN limpio en todo momento, excepto cuando est actualizando EEPROM: el bit WREN no se limpia con hardware.

Despu s de haber iniciado una secuencia de escritura, la limpieza del bit WREN no afectar el ciclo de escritura. El bit WR no podr ser ajustado hasta que el bit WREN est ajustado.

Al completar el ciclo de escritura, el bit WR es limpiado por el hardware y el bit indicador de EE Write Complete Interrupt (EEIF) tambi n se ajusta. EEIF debe ser limpiado por software.

Verificaci n de Escritura

De acuerdo a la aplicaci n, la pr ctica de buena programaci n puede indicar que los datos escritos a la EEPROM de Datos deben ser verificados. Este procedimiento debe ser usado cuando un bit de la EEPROM est muy cerca de su l mite especificado. Generalmente una falla de la EEPROM consiste en un bit que fue escrito como 0 , pero se lee como 1 , debido a problemas.

Para verificar un dato escrito, debemos primero leer la posici n de memoria grabada (si tomamos el ejemplo de reci n deber amos leer el dato contenido en la direcci n 22h); posteriormente deber amos cargar al registro W con el dato que esperamos encontrar, realizar una resta entre el dato le do (en EEDATA) y el contenido

de W, luego por medio del flag Z del registro de STATUS, nos fijamos si la diferencia es cero; si esto ocurre, el programa continuar normalmente. Hay muchas formas de hacer este proceso, pero una alternativa ser a:

BCF	STATUS, RPO	; pone a cero el bit 5 (RPO) del registro STATUS, ; dicho bit se usa para direccionamiento directo y, ; en este caso, elegimos el banco 0
MOVLW	0x22	; cargo al registro W con la dirección de la memoria ; de datos que voy a leer, en este caso 22h
MOVWF	EEADR	; muevo el contenido del registro W al registro ; EEADR que es el registro de direccionamiento de ; la EEPROM, dicho de otra manera, estoy ; realizando el direccionamiento para leer
BSF	STATUS, RPO	; pone a uno el bit 5 (RPO) del registro STATUS, ; dicho bit se usa para direccionamiento directo y, ; en este caso, elegimos el banco 1
BSF	EECON1, RD	; pone a uno el bit 0 (RD) de control de lectura del ; registro de la memoria de datos (EECON1), esto ; significa que se inicia la lectura del dato ; contenido en la dirección que hemos ; cargado en EEADR, en EEDATA tendrá el ; número 15h (si está todo bien), si hubo un error ; de escritura no habrá ese valor
BCF	STATUS, RPO	; pone a cero el bit 5 (RPO) del registro STATUS, ; dicho bit se usa para direccionamiento directo y, ; en este caso, elegimos el banco 0
MOVLW	0x15	; cargo a W con 15h
SUBWF	EEDATA, W	; resto el contenido de EEDATA con el de W
BTSS	STATUS, Z	; si el flag Z del STATUS es 1 es porque el ; resultado de la resta es cero (está todo bien), ; luego, la siguiente instrucción se ignora y el ; programa continúa normalmente. Si Z = 0, ; entonces se ejecuta la próxima instrucción
GOTO	ERROR	; el programa irá a una etiqueta que ejecuta una ; rutina que indica que hubo error.
MOVLW	; el programa continúa normalmente

Arquitectura de un PIC

En la próxima tabla vemos los Registros y Bits asociados con la EEPROM de Datos.

Registros Asociados con la Memoria de Datos

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
00h	EECON1	EEPROM data register								XXXX XXXX	XXXX XXXX
01h	EEADR	EEPROM address register								XXXX XXXX	XXXX XXXX
80h	EECON1	—	—	—	EEF	WRERR	WREN	WR	RD	---0 XXX0	---0 0000
81h	EECON2	EEPROM control register 2								-----	-----

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by data EEPROM.

La CPU del PIC

Un microcontrolador se diferencia de otros tipos de procesadores por tener circuitos especiales para atender las necesidades de ciertas aplicaciones en tiempo real. El PIC16F84A, por ejemplo, posee una cantidad importante de este tipo de prestaciones cuyo propósito es aumentar al máximo la confiabilidad del sistema, minimizar los costos por la eliminación de componentes externos, proveer modos operativo de economía del consumo y ofrecer una buena protección de datos. Estas prestaciones son:

- Selección del oscilador.
- Reset
 - Reset al Encendido (POR)
 - Temporizador de encendido (PWRT)
 - Temporizador de arranque del Oscilador
- Interrupciones
- Temporizador Watchdog (WDT - perro guardián).
- Sleep (modo de operación en bajo consumo)
- Protector de Código.
- Locaciones de Identificación (ID).
- Programación en serie (en la confección del programa).

El PIC16F84A posee un temporizador Watchdog que puede ser desactivado solamente mediante bits de configuración. Para mayor confiabilidad funciona con su propio oscilador RC. Existen dos temporizadores que ofrecen el retardo necesario al aplicar alimentación al PIC. Uno es el Temporizador de arranque del oscilador (Oscillator Start-up Timer = OST), cuya función es la de mantener el chip en Reset hasta que el oscilador a cristal se estabiliza. El otro es el Temporizador de Encendido

(Power-up Timer = PWRT) que provee un retardo fijo de 72ms (valor nominal) al aplicar tensión VDD. Este dispositivo mantiene el dispositivo en Reset durante el tiempo de estabilización de la fuente de alimentación. Con estos dos temporizadores incorporados, la mayoría de las aplicaciones no necesita circuitos de Reset externos.

El modo de SLEEP ofrece una forma de apagado de muy bajo consumo. El usuario puede despertar desde SLEEP mediante un Reset externo, fin de tiempo del Temporizador de Watchdog o una Interrupción. Se proveen varias opciones para los osciladores para poder adaptar el componente a diferentes aplicaciones. La opción de oscilador RC ahorra costos del sistema aunque no es estable, mientras que la opción con cristal ahorra consumo, es muy estable pero es más costosa. Como vimos, en la palabra de configuración del PIC se debe decir qué tipo de oscilador se va a emplear.

Bits de Configuración

Tal como vimos anteriormente en este capítulo, los bits de configuración pueden ser programados para leer 0 o quedar sin programación para leer 1 y de esta manera seleccionar diferentes configuraciones del dispositivo. Estos bits se encuentran en la ubicación de memoria del programa 2007h.

A continuación vemos otro aspecto de la palabra de configuración:

Palabra de Configuración en la Línea 2007h

R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	PWRT	WDTE	FOSC1	FOSC0		
bit13															bit0

Las referencias son las siguientes:

- R = bit que puede ser leído
- P = bit programable
- u = sin cambio

Ya hemos hablado de la función que cumple cada bit. La dirección 2007h está fuera del espacio de la memoria del programa de usuario y pertenece al espacio de la memoria de test especial y configuración (2000h — 3FFFh). Este espacio sí es accesible durante la programación.

Configuraciones del Oscilador

Tipos de Osciladores

El PIC16F84A puede funcionar con cuatro modos diferentes de oscilador. El usuario puede programar dos bits de configuración (FOSC1 y FOSC0) para seleccionar uno de estos cuatro modos:

Arquitectura de un PIC

- LP = Cristal de Baja Potencia
- XT = Cristal o Resonador
- HS = Cristal o Resonador de Alta Velocidad
- RC = Oscilador RC

Oscilador de Cristal o Resonador de Cer mica

En los modos XT, LP o HS se conecta un cristal o resonador cer mico a las pata OSC1/CLKIN y OSC2/CLKOUT para establecer la oscilaci n de acuerdo a la figura 2 de este cap tulo.

El dise o del oscilador del PIC16F84A requiere el uso de un cristal de corte paralelo. El uso de un cristal de corte serie puede dar oscilaciones con una frecuencia fuera del rango especificado por el fabricante del cristal. En los modos XT, LP o HP, el dispositivo permite la conexi n de una fuente externa para excitar la pata OSC1/CLKIN, como vemos en la figura 25.



Figura 25

Oscilador RC

Para aplicaciones insensibles a variaciones temporales, el uso de la opci n de oscilador RC ofrece ventajas adicionales. La frecuencia del oscilador RC es una funci n de la tensi n de la fuente, del valor de R_{ext} , del valor de C_{ext} y de la temperatura de funcionamiento. Adem s de estos par metros, tambi n puede haber una variaci n de la frecuencia del oscilador debido a las variaciones normales de los par metros del proceso. Adem s, las diferencias en las capacidades de conductores del dispositivo puede introducir variaciones, especialmente con valores muy bajos de C_{ext} . El usuario debe tomar en cuenta asimismo variaciones debido a las tolerancias de los componentes externos R y C usados. La figura 3 de este cap tulo, muestra c mo conectar una combinaci n RC al PIC16F84A.

Reset

El PIC16F84A distingue entre varios tipos de reset.

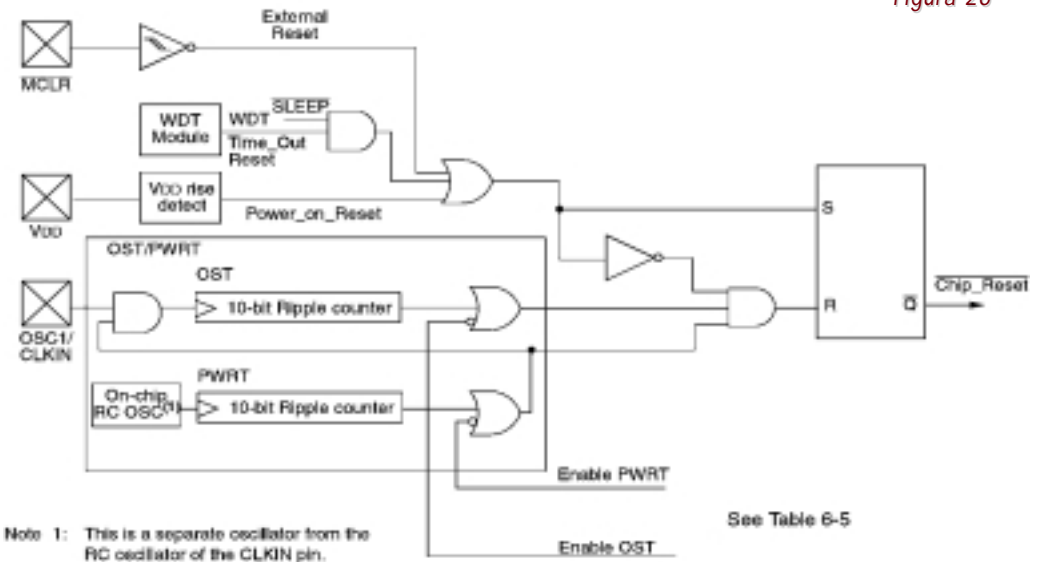
- Reset al encender (Power-on Reset POR).
- Reset MCLR durante el funcionamiento normal.
- Reset MCLR durante SLEEP.
- Reset WDT durante el funcionamiento normal.
- Reset WDT de despertar durante SLEEP.

Nota: La notación MCLR implica que el reset se activa con un 0 (negado), es decir:

$$\overline{\text{MCLR}} = \text{MCLR}$$

La figura 26 muestra un diagrama en bloques simplificado del circuito de reset incorporado en el chip. El paso de reset MCLR posee un filtro de ruido para ignorar pulsos pequeños. Las especificaciones eléctricas fijan los requisitos del ancho de pulso necesario para la pata MCLR.

Figura 26



Algunos registros no son afectados por ninguna condición de reset, su status es desconocido en un reset POR y sin cambios en cualquier otro reset. La mayoría de los demás registros se colocan en un estado de reset en resets POR, MCLR o WDR durante el funcionamiento normal y en un reset MCLR durante SLEEP. No son afectados por un reset WDT durante SLEEP debido a que este reset es visto como resumen del funcionamiento normal. En la siguiente tabla vemos la descripción de condiciones de reset para el contador de programas PC y el registro de STATUS.

Condición de Reset desde el Contador de Programa y el Registro Status

CONDICIÓN	PC	STATUS
Reset de Encendido	000h	0001 1xxxx
MCLR Reset durante la operación normal	000h	000u wuuu
MCLR Reset durante el modo SLEEP	000h	0001 0uuu
WDT Reset (durante la operación normal)	000h	0000 1uuu
WDT Wake-up	PC + 1	uuu0 0uuu
Interrupción para despertar del modo SLEEP	PC + 1 ⁽¹⁾	uuu1 0uuu

Arquitectura de un PIC

Leyenda:

u = sin cambios,

x = desconocido.

Nota 1. Cuando el Despertar (volver de SLEEP) se debe a una Interrupción y el bit GIE es ajustado, el contador PC se carga con el vector de interrupt (0004h).

La siguiente tabla brinda una descripción completa de los estados de reset para todos los registros:

Condición de RESET frente a todos los Registros

Registro	Dirección	Reset durante el encendido	MCLR WDT	Despertar o volver de SLEEP
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	---- --	---- --	---- --
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC + 1 ⁽²⁾
STATUS	03h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA ⁽⁴⁾	05h	---x xxxx	---u uuuu	---u uuuu
PORTB ⁽⁵⁾	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu ⁽²⁾
INDF	80h	---- --	---- --	---- --
OPTION_REG	81h	1111 1111	1111 1111	uuuu uuuu
PCL	82h	0000h	0000h	PC + 1
STATUS	83h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EECON1	88h	---0 x000	---0 q000	---0 uuuu
EECON2	89h	---- --	---- --	---- --
PCLATH	8Ah	---0 0000	---0 0000	---u uuuu
INTCON	8Bh	0000 000x	0000 000u	uuuu uuuu ⁽²⁾

Leyenda:

u = sin cambios

x = desconocido

- = bit sin implementar, le doy como 0,

q = valor depende las condiciones.

Notas 1: Uno o más bits de INTCON serán afectados para causar despertar.

Nota 2: Cuando el despertar se debe a un interrupt y el bit GIE es ajustado, el contador PC se carga con el vector de interrupt (0004h).

Nota 3: Los valores para cada condición específica están listados en la tabla de condiciones de reset para el contador de programas PC y el registro de STATUS.

Nota 4. En todo reset del dispositivo, estas patas son configurados como entradas.

Nota 5: Este es el valor que se encuentra en el latch del port de salida.

Los bits TO y PD son ajustados o limpiados en forma diferente en diferentes situaciones de reset. Estos bits se usan en software para determinar la naturaleza del reset.

Reset de Encendido (POR)

Un pulso de reset de encendido se genera en el chip cuando se detecta una elevación de VDD en el rango de 1,2 a 1,7 Volt. Para aprovechar el POR, solo debe conectar la pata MCLR directamente o a través de un resistor a VDD. Esto elimina los componentes RC externos que se necesitan generalmente para crear un reset de encendido. Se necesita un tiempo de crecimiento mínimo para VDD para que funcione correctamente. Las especificaciones respectivas surgen de las Especificaciones Elctricas.

Cuando el dispositivo comienza su funcionamiento normal al salir de la condición de reset, es necesario asegurar los parámetros (tensión, frecuencia, temperatura, etc) en sus valores normales para garantizar el funcionamiento. Si estas condiciones no son cumplidas, el dispositivo debe mantenerse en reset hasta que se normalice. En la Figura 27 vemos un circuito de RESET de encendido externo para una aplicación lenta de VDD.

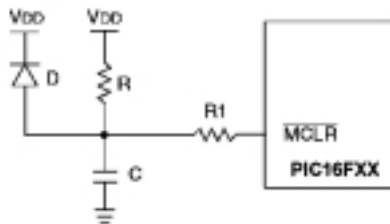


Figura 27

El reset de encendido (POR) no produce un reset interno cuando VDD cae. Sobre el circuito de la figura anterior, debemos decir lo siguiente:

1. El circuito de Power-on Reset (externo) solo es necesario cuando la tasa de crecimiento de VDD es demasiado lenta. El diodo D ayuda a descargar el capacitor rápidamente al apagar el equipo.

2. El resistor $R < 40\text{ k}\%$ es recomendado para estar seguro que la caída de tensión sobre R no exceda 0,2 volt. La máxima corriente de fuga sobre la pata MCLR

est especificada en 5 A. Una ca da de tensi n mayor producir la degradaci n del nivel de Vih en la pata MCLR .

3. El resistor R1 = 100‰ a 1000‰ limita toda corriente que circula en MCLR desde un capacitor externo C en el supuesto que haya una ruptura de MCLR debido a ESD o EOS.

Temporizador de Arranque del Oscilador (OST)

El OST (Oscillator Start-up Timer = OST) tiene la funci n de mantener el chip en Reset hasta que el oscilador a cristal se estabiliza y entra en funciones en los modos XT, LP y HS del oscilador.

Temporizador de Encendido (PWRT)

El PWRT (Power-up Timer) es un circuito interno que provee un retardo fijo de 72ms (valor nominal) al aplicar tensi n VDD nicamente. Este dise o mantiene el dispositivo en Reset durante el tiempo de estabilizaci n de la fuente de alimentaci n. Con estos dos temporizadores incorporados, la mayor a de las aplicaciones no necesitan circuitos de Reset externos.

El Perro Guardi n (WDT)

Se trata de un contador de 8 bits que act a como temporizador y tiene el objetivo de generar un reset a todo el sistema cuando se desborda su valor. Su control de tiempos es independiente del oscilador principal y se basa en una red RC. La temporizaci n nominal con la que se halla programado el Perro guardi n es de 18ms, pero puede aumentarse utilizando el Divisor de frecuencia, con el cual, trabajando en el rango mayor, puede alcanzar hasta 2,3 segundos.

Para evitar que se desborde el WDT y genere un reset, hay que recargar o refrescar su cuenta antes de que llegue a completarse. Este refresco, que en realidad consiste en ponerlo a 0 para iniciar la temporizaci n, se consigue por software con las instrucciones CLRWDT y SLEEP. El dise ador debe analizar el programa de trabajo y situar alguna de estas dos instrucciones en sitios estrat gicos por los que pasa el flujo de control antes que transcurra el tiempo que controla el Perro Guardi n. De esta manera, si el programa se cuelga (bucle infinito, espera de acontecimiento que no se produce, etc.), no se refresca a tiempo al Perro guardi n y se produce una reinicializaci n.

La instrucc i n CLRWDT borra simplemente el valor de WDT, reiniciando la cuenta. Sin embargo, la instrucc i n SLEEP, adem s de borrar el WDT, detiene a todo el sistema entrando en un modo de trabajo en el que el consumo es m nimo (modo de Reposo o de bajo consumo). Si no se desactiva al Perro guardi n cuando se entra

en el modo de Reposo, al acabar su conteo provocar un reset y se saldrá de dicho modo. Otra forma de salir del modo de Reposo es activando la patita MCLR. Como hemos visto, para desactivar al Perro Guardián, hay que poner un 0 en el bit 2 (WDTE) de la Palabra de configuración.

Existe la posibilidad de que el Divisor de frecuencia opere con el TMR0 o con el WDT, según el valor que tenga el bit PSA. Los impulsos de conteo pasan por el Divisor antes de aplicarse al TMR0 (prescaler). Por el contrario, los impulsos pasan primero por el WDT y luego por el Divisor (Post-divisor).

Interrupciones del Micro

Las interrupciones son el mecanismo más importante para la conexión del microcontrolador con el mundo exterior, sincronizando la ejecución de programas con acontecimientos externos.

Las interrupciones son una especie de subrutinas de las cuales se diferencian en los procedimientos que las ponen en marcha. Por otra parte las subrutinas se ejecutan cada vez que en el programa aparece una instrucción CALL, mientras que las interrupciones se ponen en marcha al aparecer en cualquier instante un evento externo al programa, es decir por un mecanismo de hardware. El PIC16X84/PIC, por ejemplo, dispone de 4 posibles fuentes de interrupción:

1. *Activación del pin RB0/INT*
2. *Desborde del temporizador TMR0*
3. *Cambio de estado en una de las 4 patas de más peso (RB7:RB4) del PORTB*
4. *Finalización de la escritura en la EEPROM de datos.*

Cuando se produce alguno de estos estados, se origina una petición de interrupción, que si se acepta, guarda el valor del contador de programa (PC - CP) actual en la Pila, pone a cero el bit GIE (Global Interrupt Enable), lo que prohíbe cualquier otra interrupción y se carga el PC con el valor 0004H, que es la posición del vector de interrupción, y comienza a ejecutarse el programa de atención a la interrupción que se encuentra a partir de esta dirección.

Una interrupción está controlada mediante dos bits. Uno de ellos actúa como señalizador o flag que indica si se ha producido o no la interrupción, y la otra funciona como bit de permiso o inhibición de la interrupción en sí.

Los bits de control que se encuentran en el registro INTCON (0Bh - 8Bh) habilitan y configuran las interrupciones.

Para que se pueda producir una interrupción por alguna de estas fuentes, el bit

correspondiente debe estar en 1, mientras que los bits se alizadores o flags que est n en los registros INTCON y EEDATA (08h) informan si se ha producido la interrupci n cuando stos se ponen a 1. Cualquiera de las interrupciones tambi n puede sacar al procesador del modo de reposo SLEEP.

El bit GIE (Global Interrupt Enable) es el de activaci n global del permiso de interrupci n, y se borra autom ticamente cuando aparece una interrupci n, con el objeto de evitar que se produzca otra interrupci n mientras se est atendiendo a la primera. Cuando por programa se retorna de la interrupci n con una instrucc i n RETFIE, el bit GIE se vuelve a activar poni ndose a 1. Para el resto de los flags o bits indicadores de interrupci n, no est previsto mecanismo de puesta a cero, por lo que es el programa de atenci n a la interrupci n el que debe realizar el tratamiento de la correspondiente interrupci n y adem s, el que debe poner el o los flags de indicaci n de interrupci n a 0. De no ser as, no se podr salir de la rutina de atenci n a la interrupci n.

Esta familia de PICs posee un vector de interrupci n en la direcci n 0004h; esto quiere decir que, sea cual sea la fuente de la interrupci n, el PC se carga con 0004h. Por lo tanto, el programa de atenci n a la interrupci n debe encargarse de comprobar el estado de cada uno de los flags para saber cu l es la fuente de interrupci n y actuar seg n el caso.

Si se necesita preservar alg n otro registro adem s del PC, cuando se atiende una interrupci n, debe ser el propio programa de atenci n a la interrupci n el que se encargue de salvar su estado al inicio de la rutina y de devolverlos al final del mismo, de igual modo que se hac a en las subrutinas.

Por lo dicho, el P debe realizar determinados procesos autom ticamente mientras que otros se ejecutan por programa, estos procesos son:

1. Cuando se activa una interrupci n, el flag correspondiente se activa. Si el bit de permiso correspondiente est en 1 y el bit de habilitaci n de todas las interrupciones (GIE) est en 1, se produce la interrupci n.

2. Para evitar que se produzca otra interrupci n mientras se est atendiendo a otra anterior, el bit GIE se pone en 0.

3. El valor del PC se guarda en la PILA

4. El PC se carga con el valor 0004h, que es el vector de interrupciones

5. El programador, debe comenzar la rutina de atenci n a la interrupci n con un salto a la posici n de memoria donde se encuentra el programa de atenci n a dicha interrupci n, seguidamente se guardan todos los registros que puedan ser modificados por sta, luego si est n habilitadas varias v as de interrupci n, se debe verificar el valor de los flag para determinar la causa de la interrupci n.

6. La rutina de interrupci n lleva al programa a la subrutina correspondiente.

7. Antes de retornar al programa principal se deben reestablecer los valores que ten an los registros antes de producirse la interrupci n y se deben borrar por software los flags que indican las fuentes de las interrupciones.

8. Cuando se llega a la última instrucción de la rutina de interrupción, RETURN, se carga el PC con el valor que se guardó inicialmente en la PILA y el bit GIE se pone automáticamente en 1.

Interrupción Externa (INT)

La fuente de interrupciones INT es sumamente importante para atender eventos externos en tiempo real. Cuando en la línea RB0/INT se hace una petición de interrupción, entonces, de forma automática, el bit INTF del registro INTCON se pone en 1 y si el bit GIE=1. Luego se pone en marcha el mecanismo de atención a la interrupción. Mediante el bit INTDEG del registro OPTION, se puede seleccionar el flanco activo de RB0/INT, ya que con esta puesta en 1 el flanco activo es el de subida y cuando está en 0 el flanco activo es el de bajada.

El programa de atención a la interrupción antes de regresar al programa principal debe borrar el flag INTF, para que no se repita el proceso de atención a la interrupción cuando se ejecuta la instrucción de retorno de interrupción RETFIE.

Interrupción por Desborde del TMR0

Para activar la interrupción del TMR0, los bits T0IE y GIE del registro INTCON deben estar en 1; bajo estas condiciones cuando el temporizador TMR0 se desborda al pasar de FFh a 00h, se activa el flag TOIF del registro INTCON.

Si no se carga de nuevo TMR0 cuando se desborda, éste sigue contando desde 00h hasta FFh. Este registro puede escribirse o leerse en cualquier momento, pero hay que tener en cuenta que al escribir sobre él, se pierden dos ciclos de reloj para mantener el sincronismo.

Cuando se carga el registro TMR0 con un valor XXh, éste cuenta FFh-XXh impulsos y el tiempo que tarda en hacerlo viene dado por la expresión:

$$\text{Temporización} = 4 \cdot T_{osc} \cdot (256 - N_{10}). \text{ Rango del Divisor de Frecuencia}$$

Interrupción por Cambio de Estado en las patas RB7 y RB4

Para activar la interrupción por cambio de nivel en las patas RB4 y RB7, los bits RBIE y GIE del registro INTCON deben estar a 1. Cuando esto es así y se produce un cambio de nivel en cualquiera de los pines RB4-RB7, se activa el flag RBIF del registro INTCON. Este tipo de interrupciones está especialmente pensada para el control de un teclado matricial 4 x 4 de 16 teclas.

Interrupción por Finalización de la Escritura en la EEPROM de Datos

El área de EEPROM dispone de 64 bytes donde se pueden almacenar datos que no se pierden al desconectar la alimentación. La memoria EEPROM no está mapeada

da en la zona de memoria de datos donde se encuentran los registros SFR y GPR. Para poder leerla y escribirla durante el funcionamiento normal del microcontrolador hay que utilizar los registros especiales EEDATA, EEADR, EECON1 y EECON2, tal como ya hemos visto. Recuerde que el Registro EEADR, se encuentra en la posición de memoria 09h del banco 0, en él se carga directamente la dirección a la que acceden de la EEPROM de datos. Las 64 posiciones de un byte ocupan las direcciones de un mapa que comienza en la posición 00h y termina en la 3Fh, por eso los 2 bits de más peso del registro EEADR siempre valen 0.

Los bit RD y WR indican respectivamente lectura o escritura. No hay que ponerlos a 0 si lo a 1. Se borran automáticamente cuando la operación de lectura ha sido completada.

El registro EECON2 no está implementado físicamente, por lo que es imposible leerlo (si se intenta leer, todos sus bits se ponen a 0). Se emplea como dispositivo de seguridad durante el proceso de escritura de la EEPROM, para evitar las interferencias en el tiempo que precisa su desarrollo. La seguridad se consigue escribiendo los valores concretos 55h y AAh. Un ciclo de escritura en una posición EEPROM de datos tiene una duración de 10 ms, que es un tiempo muy grande para la velocidad del procesador.
