



CeTAD – Facultad de Ingeniería - UNLP

Descripción General de un Microcontrolador – (Módulos Básicos)

(Basado en Freescale HC08 y HCS08)

Cátedra de Circuitos digitales y Microprocesadores

Autores:

Ing. Jorge R. Osio
Ing. Walter Aróztegui
Ing. José A. Rapallini

Septiembre de 2011

Versión 2.0

INDICE.

1. Introducción

- 1.1. Módulos Disponibles en las Diferentes Líneas de Microcontroladores
- 1.2. Módulos Específicos de la Familia HC908QT/QY

2. Módulo de Interrupción Externa (IRQ)

- 2.1. Características
- 2.2. Descripción Funcional
 - 2.2.1. MODE = 1
 - 2.2.2. MODE = 0
- 2.3. Interrupciones
- 2.4. Modos de bajo Consumo
- 2.5. Módulo IRQ Durante el Quiebre de Interrupción
- 2.6. Señales de Entrada/Salida
 - 2.6.1. Pines de Entrada IRQ
- 2.7. Registros

3. Módulo de Auto Activación (AWU)

- 3.1. Características
- 3.2. Descripción Funcional
- 3.3. Modo Wait y Stop
- 3.4. Registros de Entrada/Salida
 - 3.4.1. Registro de datos del Puerto A (PTA).
 - 3.4.2. Registro de control y estado de la interrupción por teclado (KBSRC).
 - 3.4.3. Registro de habilitación de interrupción por teclado (KBIER).

4. Módulo de Interfaz del Temporizador (TIM)

- 4.1. Características
- 4.2. Convenciones de nombres de Pines
- 4.3. Descripción Funcional
 - 4.3.1. Preescalador del contador del TIM
 - 4.3.2. Captura de Entrada
 - 4.3.3. Comparador de Salida
 - 4.3.3.1. Comparador de Salida sin Buffer
 - 4.3.3.2. Comparador de Salida con buffer
 - 4.3.4. Modulación por Ancho de Pulso (PWM)

- 4.3.4.1. Generación de señal PWM sin buffer
- 4.3.4.2. Generación de señal PWM con buffer
- 4.3.4.3. Inicialización del PWM

4.4. Interrupciones

4.5. Modo Wait

4.6. TIM Durante el Quiebre de Interrupciones

4.7. Señales de Entrada/Salida

- 4.7.1. Pin de Clock del TIM (PTA2/TCLK)
- 4.7.2. Pines I/O de canales del TIM (PTA0/TCH0 y PTA1/TCH1)

4.8. Registros de Entrada / Salida

- 4.8.1. Registros de Control y Estado del TIM
- 4.8.2. Registros del Contador del TIM
- 4.8.3. Registros del Módulo Contador del TIM
- 4.8.4. Registros de control y Estado del canal del TIM
- 4.8.5. Registros del Canal de TIM

5. Módulo de Interrupción por Teclado (KBI)

5.1. Características

5.2. Descripción funcional

- 5.2.1. Operación de Teclado
- 5.2.2. Inicialización de Teclado

5.3. Modos Wait y Stop

5.4. Módulo Keyboard (Teclado) Durante el Quiebre de Interrupciones

5.5. Registros de entrada/Salida

- 5.5.1. Registro de control y estados del Teclado
- 5.5.2. Registro de Habilitación de Interrupción por Teclado

6. Módulo Conversor Analógico-Digital (ADC)

6.1. Introducción

6.2. Descripción

6.3. Descripción del Módulo ADC en el MCU HC908

6.4. Voltaje de conversión

6.5. Tiempo de conversión

6.6. Configuración de registros

- 6.6.1. Registro de estado y control del ADC (ADSCR)
- 6.6.2. Registro de datos del ADC
- 6.6.3. Registro de configuración del reloj para el ADC

Apéndice.

1. Registros de Control, de Estado y de Datos

2. Resets e Interrupciones

2.1. Procesamiento de Excepciones

2.1.1. Reconocimiento

2.1.2. Arbitraje

2.1.3. Stacking (apilado)

2.1.4. Búsqueda de Vectores

2.2. Enmascaramiento (Masking)

3. Factores de error en Conversores AD

3.1. Error de muestreo (Sampling Error)

3.2. Error por goteo del pin (Pin Leakage Error)

3.3. Error por ruido inducido (Noise Induced Error)

3.4. Error de cuantización (Quantization Error)

3.5. Error de escala cero (Zero Scale Error =Offset Error)

3.6. Error de escala plena (Full Scale Error)

3.7. Error diferencial no lineal (Differential non Linearity Error)

3.8. Error integral no lineal (Integral non Linearity Error)

3.9. Error por parpadeo en la conversión (Code Jitter Error)

3.10. Error por no monotonicidad (Non Monotonicity Error)

1. Introducción

La gran diferencia entre los Microprocesadores [5], [9] y [10] y los Microcontroladores es que estos últimos tienen espacio de memoria y una variedad de módulos periféricos que le permiten interactuar directamente con el mundo Exterior.

La memoria se puede dividir en memoria Flash (para el almacenamiento del programa), Memoria RAM (en donde se crean las variables) y Registros de control y estados.

Dentro de los módulos Periféricos se pueden distinguir dos grupos, aquellos que se encargan de contribuir, asegurar y monitorear el buen funcionamiento del MCU y los que permiten comunicar al MCU con otros dispositivos externos de diferentes maneras y mediante diferentes protocolos.

Dentro de los módulos que se encargan de asegurar el buen funcionamiento del MCU se encuentran el Módulo oscilador, el Módulo SIM, el Módulo COP y el LVI. Todos Estos módulos fueron desarrollados en el Apunte “Descripción General de un Microcontrolador – CPU” basado en los Microcontroladores de Freescale HC908 [1] y [2].

En este Apunte se desarrollarán el resto de los módulos que incluye la Familia HC908QT/QY y otros módulos, como el SCI, SPI, I²C y el USB, que aparecen muy comúnmente en las nuevas Familias de Microcontroladores. Básicamente se describirán todos aquellos módulos que le permiten al Microcontrolador interactuar con el mundo exterior.

1.1. Módulos Disponibles en las Diferentes Líneas de Microcontroladores

El objetivo de esta sección es comentar que cambios y que novedades se van incorporando en los Microcontroladores a medida que avanza la tecnología.

En este Apunte se detallarán los módulos principales que puede contener hoy en día un MCU, pero conjuntamente con estos Módulos de altísimas prestaciones se han incorporado otros de aplicaciones específicas que permiten a dichos dispositivos interactuar directamente con cualquier otro dispositivo [2] y [12].

Entre las interfaces más comunes disponibles en los nuevos MCUs se encuentran la Ethernet, CAN, USB, I²C, SPI (serial síncronico), SCI (serial asíncronico) [2], interfaces inalámbricas, etc.

Por otro lado, el avance de la tecnología ha logrado que estos dispositivos alcancen frecuencias de funcionamiento arriba de los 100 MHz e incorporen módulos para Procesamiento Digital de Señales, facilitando la realización de operaciones y la implementación de transformadas como la FFT y de filtros FIR.

Se ha llegado a un punto en donde algunos Microcontroladores permiten implementar un Sistema Operativo Embebido dentro de ellos, facilitando de esta manera la configuración y el uso de cualquiera de las interfaces disponibles en ellos.

1.2. Módulos Específicos de la Familia HC908QT/QY

Entre los Módulos específicos de la Familia HC908QT/QY [1], que se muestran en la figura 1, se tienen por un lado los que aseguran el correcto funcionamiento del MCU:

- **Módulo Oscilador:** Provee al MCU de una fuente de Clock. Esta fuente se puede implementar mediante un oscilador interno, un cristal externo y mediante un circuito RC.
- **Módulo SIM:** Generación y control de clock, control de Master reset, control de interrupciones, manejo de la temporización en el MCU.
- **Módulo COP:** Ayuda a recuperar la línea de ejecución de código del usuario.
- **Módulo LVI:** Realiza la supervisión de la fuente de alimentación.

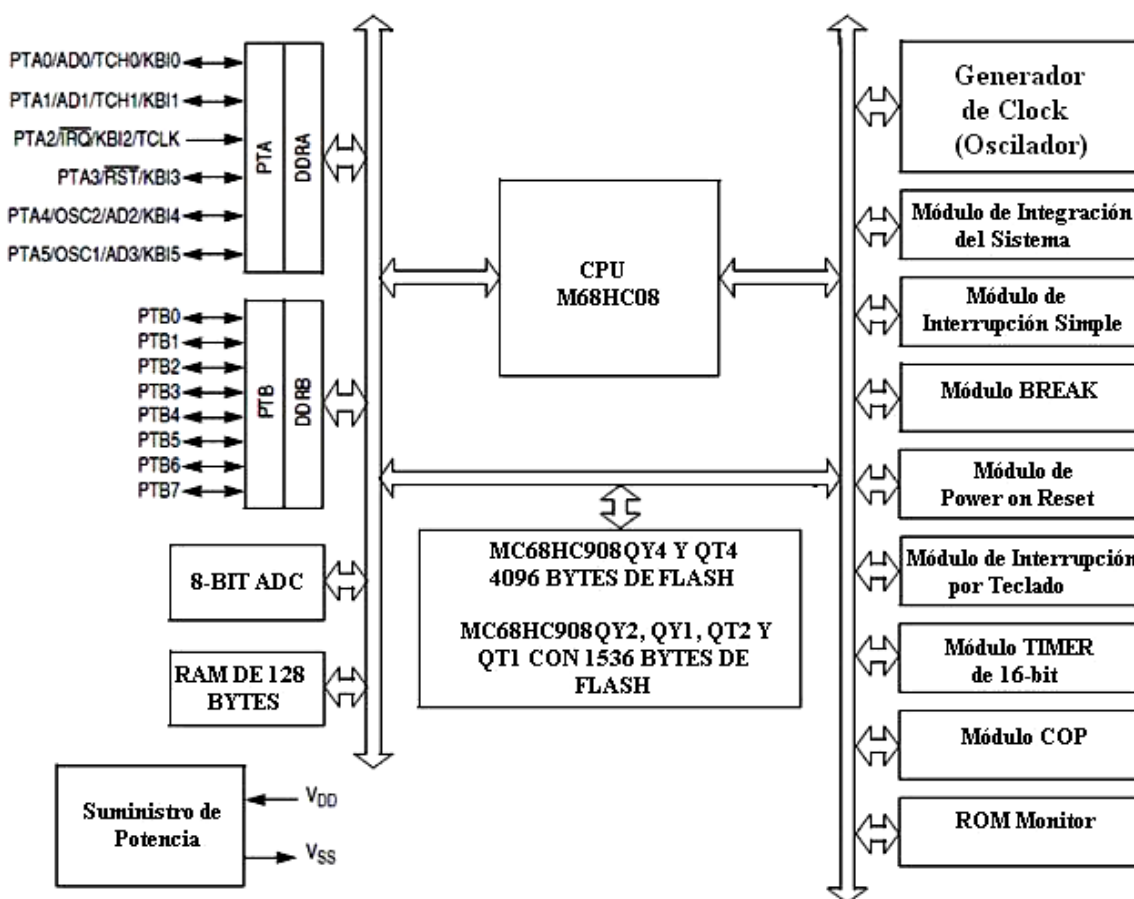


Figura 1. Diagrama en Bloques de la Línea de Microcontroladores HC908QT/QY

Por otro lado se tienen los módulos que permiten al MCU interactuar con otros dispositivos y cuyo funcionamiento se describirá en este Apunte. Los módulos que se detallan en los subsiguientes capítulos son:

- **Módulo IRQ:** Es un módulo que permite al MCU recibir interrupciones externas mediante un puerto de entrada.
- **Módulo AWU:** Este Módulo no provee una interfaz externa al MCU, pero interviene en todas aquellas operaciones en donde el MCU se queda inactivo esperando la llegada de algún evento externo.
- **Módulo TIM:** Provee Diferentes Modos de Temporización mediante Contadores, divisores de la frecuencia de Bus, Capturadores de entrada, Comparadores de Salida y hasta modulación por ancho de pulso PWM.
- **Módulo KBI:** Permite generar interrupciones por teclado externamente.
- **Módulo ADC:** Permite Digitalizar señales analógicas con una resolución de hasta 255 niveles.

2. Módulo de Interrupción Externa (IRQ)

El Pin IRQ (interrupción externa) [3], que comparte con PTA2 (entrada de propósito general) e interrupción por teclado (KBI) en el HC908Qy4, provee una interrupción enmascarable. La Figura 2 muestra el diagrama en bloques completo del MCU, en donde se resalta el Módulo IRQ y el pin de entrada correspondiente.

2.1. Características

Entre las características del IRQ se incluyen:

- Pin de interrupción externa, IRQ
- Bits de control de interrupción IRQ
- Detección de Interrupción programable por flanco solamente o por flanco y nivel.
- Detección de interrupción automática
- Pullup interno seleccionable.

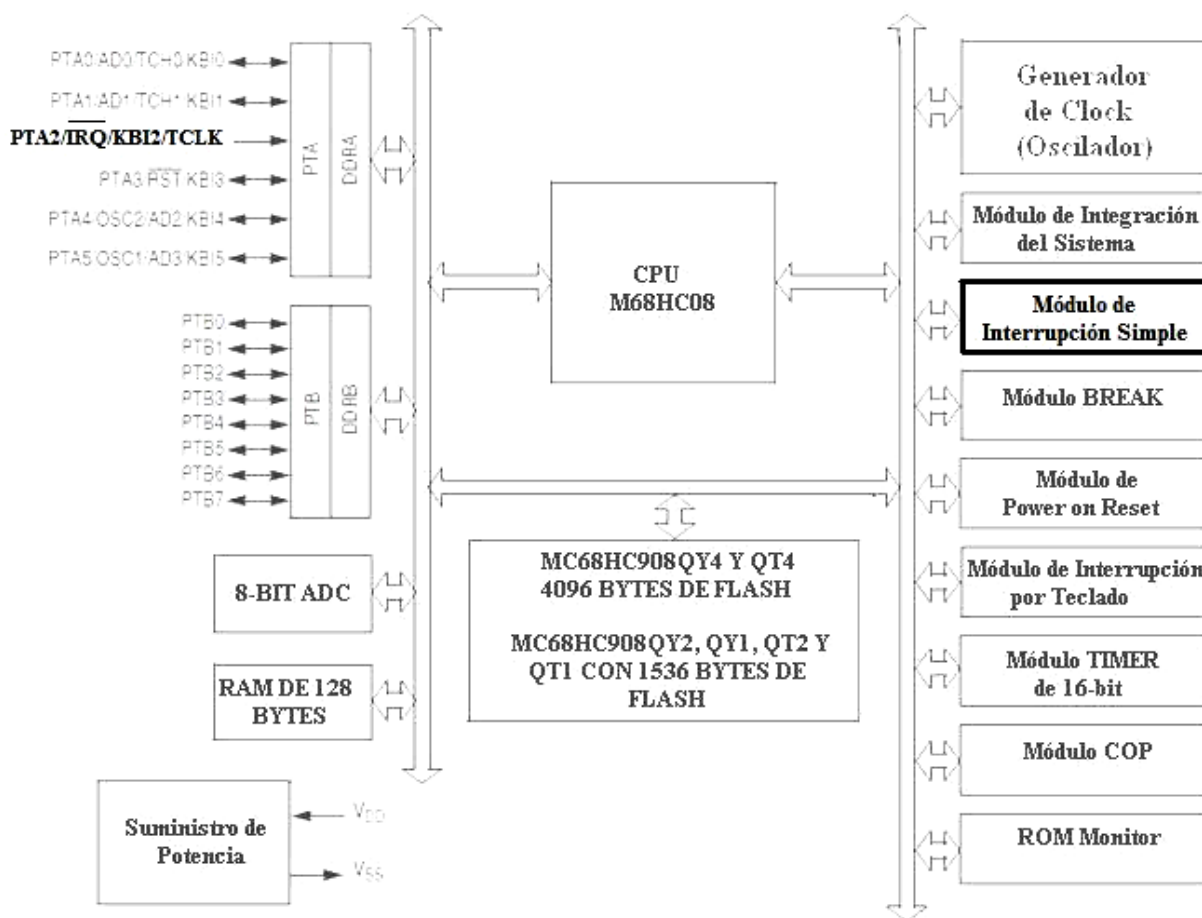
2.2. Descripción Funcional

La funcionalidad del pin IRQ se activa por ajuste del registro de configuración 2 (CONFIG2), específicamente el bit IRQEN. Un cero desactiva la función IRQ y el pin PTA2 tomará las demás funcionalidades. Un uno habilita la función IRQ.

Un nivel bajo aplicado al pin de requerimiento de interrupción externa (IRQ) puede generar un requerimiento de interrupción del CPU. La Figura 3 muestra la estructura del módulo IRQ. Las señales de Interrupción sobre el pin IRQ se mantienen en el latch IRQ (cerrojo IRQ). El latch IRQ se mantiene hasta que ocurra una de las siguientes acciones:

- Vector de carga IRQ — Un vector de carga IRQ genera automáticamente una señal de acuse de interrupción que borra el latch IRQ.
- Borrado por Software, el Software puede borrar el latch IRQ escribiendo un 1 al bit ACK en el registro de control y estado de interrupción (INTSCR).
- Reset — Al resetear se borra automáticamente el latch IRQ.

El pin de interrupción externa se dispara en un flanco descendente de reset fuera de un reset y es configurable por software para ser detectado por flanco descendente o por flanco descendente y nivel bajo. El bit MODE en INTSCR controla la sensibilidad de disparo del pin IRQ.



RST, IRQ: Estos pines tienen pullups internos (de 30K Ohms)

PTA[0:5]: Capacidad de alto suministro de corriente

PTA[0:5]: Estos pines tienen interrupciones por teclado programables y pullups

PTB[0:7]: No disponible en – MC68HC908QT1, MC68HC908QT2, y MC68HC908QT4

ADC: No disponible en MC68HC908QY1 y MC68HC908QT1

Figura 2. Diagrama en bloques que resalta el bloque IRQ y el pin correspondiente

Cuando se setea, el bit IMASK en las mascarar INTSCR se genera el requerimiento de interrupción de IRQ. No se almacenará un requerimiento de interrupción en la lógica de prioridad de interrupción a menos que IMASK sea borrado.

Se debe tener en cuenta que la máscara de interrupción (I) en el registro de código de condición (CCR) enmascara todos los requerimientos de interrupciones, incluyendo el requerimiento de interrupción IRQ.

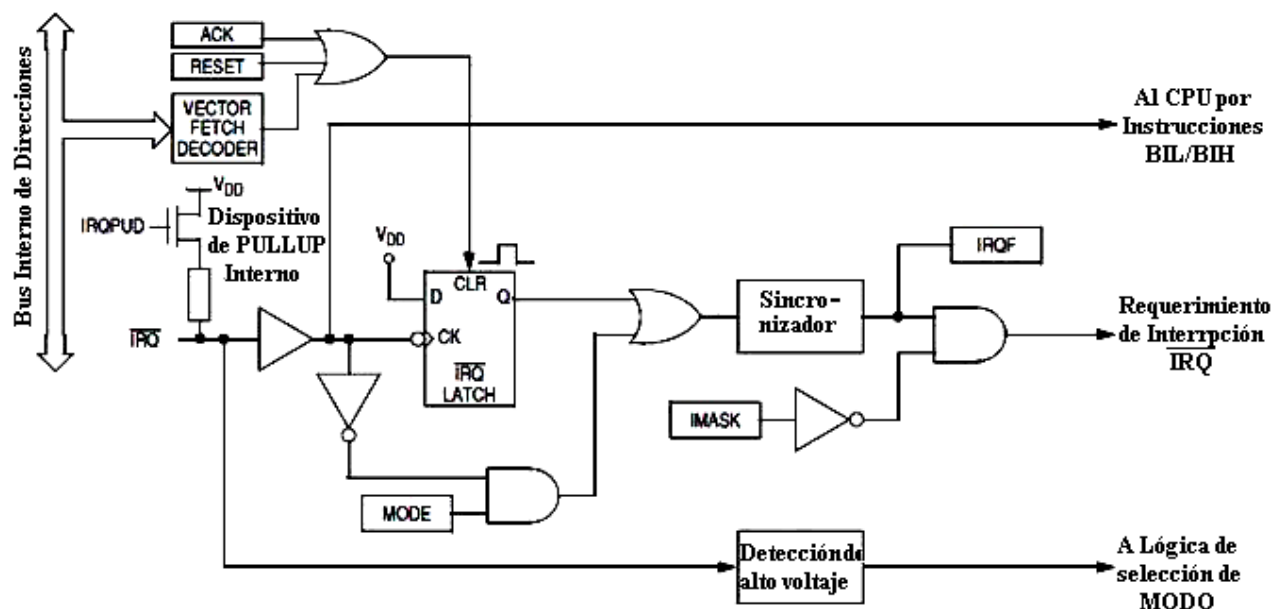


Figura 3. Diagrama en bloques del Módulo IRQ

2.2.1. MODE = 1

Si el bit Mode está seteado, el pin IRQ tendrá sensibilidad por flanco descendente y por nivel bajo. Con el Mode seteado, se deben realizar las siguientes acciones para borrar el requerimiento de interrupción IRQ:

- Retorno del pin IRQ a un nivel alto. Mientras el pin IRQ esté en bajo, el requerimiento de IRQ seguirá activo.
- El borrado del vector de carga IRQ (vector Fetch IRQ) o el borrado por software. Un vector fetch IRQ genera una señal de reconocimiento de interrupción para borrar el latch IRQ. El Software genera la señal de reconocimiento de interrupción escribiendo un 1 al ACK en INTSCR. El bit ACK es útil en aplicaciones que consultan el pin IRQ y precisan del software para borrar el latch IRQ. Escribiendo el ACK antes de abandonar una rutina de servicio de interrupción también se pueden prevenir interrupciones indeseadas causadas por ruido. Seteando el ACK no se afectarán las subsiguientes transiciones en el pin IRQ. Un flanco descendente que ocurre después de escribir el ACK guarda otro requerimiento de interrupción. Si la máscara del bit IRQ, IMASK, es borrada, la CPU [5] carga el contador de programa con la dirección del vector IRQ.

El vector fetch IRQ o el borrado por software y el regreso del pin IRQ a un nivel alto, pueden ocurrir en cualquier orden.

La solicitud de interrupción sigue pendiente mientras el pin IRQ está en nivel bajo. Un reset borrará el latch IRQ y el bit de control MODE, borrando así la interrupción incluso si el pin permanece en nivel bajo. Utilice la instrucción BIH o BIL para leer el nivel lógico sobre el pin IRQ.

2.2.2. MODE = 0

Si el bit MODE está en cero, el pin IRQ solo se activará con un flanco descendente. Con el MODE borrado, el borrado del vector fetch IRQ o un borrado por software borra inmediatamente el latch IRQ.

El bit IRQF en INTSCR se puede leer para chequear las interrupciones pendientes. El bit IRQF no es afectado por IMASK, lo que hace que sea útil en aplicaciones donde se prefiere hacer polling (lectura continua del estado del pin).

2.3. Interrupciones

La siguiente fuente de IRQ puede generar requerimientos de interrupción:

- flag de interrupción (IRQF) — el bit IRQF es seteado cuando el pin IRQ se encuentra configurado en el modo IRQ. El bit de la máscara de interrupción del IRQ, IMASK, se utiliza para habilitar o deshabilitar los requerimientos de interrupción por IRQ.

2.4. Modos de bajo Consumo

Las instrucciones WAIT y STOP ponen al MCU en modo de espera de bajo consumo. El módulo IRQ permanece activo en modo wait. Borrando el IMASK en INTSCR se habilita el requerimiento de interrupción IRQ para sacar al MCU fuera del modo wait. En el modo STOP sucede lo mismo que en modo wait durante un requerimiento de interrupción.

2.5. Módulo IRQ Durante el Requerimiento de Interrupción

El módulo de integración del Sistema (SIM) controla que los bits de estado en otros módulos puedan borrarse durante el estado de interrupción (break). El bit BCFE en el flag de break del registro de control (BFCR) habilita al software a borrar el bit de estado durante el estado de break.

Para permitir el borrado por software del bit de estado durante un requerimiento de interrupción, se debe poner en 1 el bit BCFE. Si el bit de estado se borra durante el estado de interrupción, se mantendrá borrado cuando el MCU salga de dicho estado.

Para proteger el bit de estado durante el estado de interrupción, se debe escribir un cero en BCFE. Con BCFE en cero (estado predeterminado), el software puede leer y escribir registros durante el estado de interrupción sin afectar los bits de estado. Algunos bits de estado tienen un procedimiento de dos pasos de borrado lectura/escritura. Si el software realiza el primer paso sobre uno de esos bits antes del requerimiento de interrupción, el bit no podrá cambiar durante el estado de interrupción hasta tanto BCFE sea borrado. Después de la interrupción, realizando el segundo paso se borra el bit de estado.

2.6. Señales de Entrada/Salida

El módulo IRQ comparte el pin con la interrupción por teclado, puertos de entrada/salida, y módulos de interfaz TIMER.

Cuando el IRQ está activado en el registro CONFIG2, las instrucciones BIH y BIL pueden ser utilizadas para leer el nivel lógico sobre el pin IRQ. Si la función IRQ está desactivada, estas instrucciones se comportan como si el pin IRQ tuviera un 1 lógico, independientemente del nivel

real sobre el pin. Por el contrario, cuando el IRQ está activado, el bit 2 del registro de datos del puerto A siempre se leerá como un 0.

Cuando se utiliza el disparo de interrupción sensible al nivel, se evitan las falsas interrupciones por enmascaramiento del requerimiento de interrupción en la rutina de interrupción. Se conecta un pullup interno a VDD en el pin IRQ; esto se puede desactivar seteando el IRQPUD en el registro CONFIG2 (\$001E).

2.6.1. Pines de Entrada IRQ

El pin IRQ provee una fuente de interrupción externa enmascarable. Además, contiene un dispositivo de pullup interno.

2.7. Registros

El registro de control y estado IRQ (INTSCR) controla y monitorea las operaciones del módulo. El registro INTSCR tiene las siguientes funciones:

- Muestra el estado del flag IRQ
- Borra el latch IRQ
- Enmascara el requerimiento de interrupción IRQ
- Controla la sensibilidad de disparo del pin de interrupciones IRQ

Address: \$001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQF	0	IMASK	MODE
Write:						ACK		
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figura 4. Registro de Control y Estado (INTSCR)

Flag IRQF: Este bit de estados de solo lectura es seteado cuando hay una interrupción IRQ pendiente.

- 1 = Interrupción pendiente IRQ
- 0 = No hay ninguna interrupción IRQ pendiente

Bit de acuse de requerimiento de Interrupción ACK: Escribiendo un 1 a este bit de solo escritura se borra el latch IRQ. ACK siempre se lee como 0.

Bit de mascara de interrupción IMASK: Escribiendo un 1 a este bit de lectura/escritura se deshabilita el requerimiento de interrupción IRQ.

- 1 = Requerimiento de interrupción IRQ deshabilitado
- 0 = Requerimiento de interrupción IRQ habilitado

Bit de selección por nivel/flanco MODE: Este bit de lectura/escritura controla la sensibilidad de disparo del pin IRQ.

- 1 = Requerimiento de Interrupción IRQ sobre flanco descendente y niveles bajos
- 0 = Requerimiento de Interrupción IRQ sobre flanco descendente únicamente

3. Módulo de Auto Activación (AWU)

Esta sección describe el módulo de auto activación del Microcontrolador (Auto Wake Up) [3] que genera una interrupción periódica durante el modo de parada (Stop) para salir automáticamente de este modo sin necesidad de una señal de interrupción externa. En la figura 5 se ve un diagrama en bloques de su estructura.

3.1. Características

Este módulo incluye:

- Una señal de interrupción interna con un bit separado de habilitación de interrupción compartiendo con el teclado el mismo vector de interrupción y el mismo bit de enmascaramiento de interrupción.
- Salida del modo de baja energía (Stop) sin señales externas.
- Tiempo de espera seleccionable.
- Oscilador interno dedicado, separado del resto de las fuentes de reloj del sistema.

3.2. Descripción Funcional

La función del AWU (Auto Wake Up) es generar pedidos de “activación” periódicos que haga salir a la unidad Microcontroladora del modo de parada o Stop.

Estos pedidos se tratan como un pedido de interrupción por teclado, salvo que no es necesaria una señal proveniente de un pin sino que es generada por una lógica interna.

Escribiendo el bit AWUIE en el registro de habilitación de interrupción por teclado, se habilita o deshabilita la entrada de interrupción de auto activación. Aplicando un 1 lógico a la entrada AWUIREQ (ver figura 5) con el pedido de interrupción de activación habilitado, fija el pedido de interrupción.

Este cerrojo o latch de auto activación AWUL puede leerse directamente desde el bit 6 del registro de datos del puerto A. Este bit es de solo lectura y no tiene asociado ningún registro del puerto A como el registro de dirección de dato o el de pullup para PTA6 y puede ser borrado mediante la escritura del bit ACKK en el registro de teclado KBSCR o bien durante un Reset. El bit AWUIE del registro de habilitación por teclado KBI no tiene efecto sobre la lectura del AWUL.

Entrando al modo de parada o Stop se habilitará la lógica de generación de auto activación; un oscilador RC interno, exclusivo de este módulo maneja el generador de pedido de activación, una vez que se alcanza el desborde (overflow) en el contador del generador se produce la señal de pedido AWUIREQ y esta se almacena y se envía a la lógica de la interrupción por teclado KBI. Tal pedido de interrupción solo se atenderá si está seteado el bit de habilitación de interrupción asociado AWUIE en el registro KBIER, se debe recordar que también comparte el mismo vector de interrupción que la de teclado.

El desborde de cuenta puede ser seleccionado desde dos opciones definidas por el bit COPRS en el registro de configuración CONFIG1. Este bit en realidad está “prestado” del módulo de operación correcta de la computadora (COP) en función de que éste se encuentra libre en el modo de parada o Stop.

Los valores típicos del pedido de activación periódico a temperatura ambiente son:

COPRS= 0 : 650ms @ 5V, 875ms @ 3V
COPRS= 1 : 16ms @ 5V, 22ms @ 3V

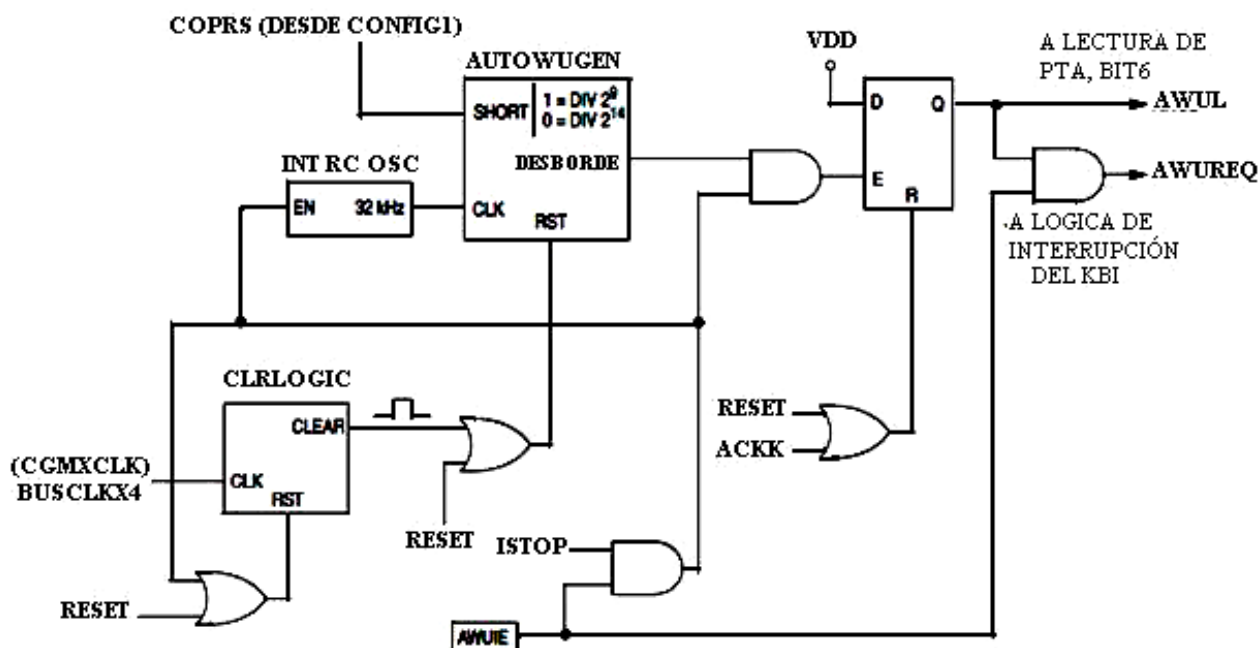


Figura 5. Generación lógica del requerimiento de interrupción de auto activación

El oscilador del Módulo AWU es altamente dependiente de la temperatura y la tensión de operación, por lo que el módulo no es recomendado para usar como una función de cronometraje o muy asociada al tiempo.

El requerimiento de wakeup se mantiene para permitir la identificación de la fuente de interrupción. El valor bloqueado, AWUL, pueden leerse directamente desde la posición del bit 6 del registro de datos del puerto A. Este es un bit de solo lectura que ocupa una posición de bit vacía en el puerto A. Para este bit no existen los registros asociados, tales como el de datos PTA6, el de dirección PTA6, y el de pullup. El latch puede borrarse escribiendo el bit ACKK en el registro KBSCR. El Reset también borra el latch. El bit AWUIE en el registro de habilitación de interrupción KBI no tiene efecto sobre la lectura del AWUL.

El oscilador y los contadores del módulo AWU permanecen inactivos durante los modos de operación normal, aún en el modo de espera WAIT y solamente se activan entrando en el modo de parada (modo stop).

3.3. Modo Wait y Stop

El modulo AWU mantiene inactivo el modo wait. Cuando el módulo AWU se activa (AWUIE = 1 en el registro de habilitación de interrupción por teclado) al ingresar al modo Stop, borrando el bit IMASKK en el registro de control y estado de teclado se habilita el requerimiento de interrupción por teclado para sacar al MCU del modo stop. El contador AWU inicia desde cero cada vez que se entra al modo stop.

3.4. Registros de Entrada/Salida

Como dijimos, el módulo AWU comparte registros con el módulo de interrupción por teclado KBI y con el Puerto A de entrada y salida. Así los registros de operación y monitoreo de operación del módulo AWU son:

3.10.1. Registro de datos del Puerto A (PTA).

El puerto A tiene además de los cerrojos de datos del puerto un cerrojo de dato para el estado del pedido de interrupción del módulo AWU.

Address: \$0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AWUL	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Write:								
Reset:	0	0	Unaffected by reset					


 = Unimplemented

Figura 6. Registro de datos del puerto A

Donde el AWUL (Auto Wake Up Latch) es un bit de solo lectura del cerrojo del pedido de interrupción de despertar automático.

- 1 = pedido de interrupción de despertar automático está pendiente.
- 0 = pedido de interrupción de despertar automático no está pendiente.

Los bits PTA5 a PTA0 no se usan conjuntamente con las características del módulo AWU y su funcionamiento deberá verse en el módulo del puerto A.

3.10.2. Registro de control y estado de la interrupción por teclado (KBSRC).

Este registro consta de:

1. Flags de pedido de interrupción por teclado o despertar automático.
2. Reconocimiento (Acknowledge) de pedido de interrupción de teclado/auto activación.
3. Enmascaramiento de los pedidos de interrupción teclado/despertar automático.

Address: \$001A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

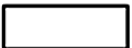
 = Unimplemented

Figura 7. Registro de control y estado de keyboard (teclado), (KBSCR)

Aquí, los bits 7 a 4 no son usados.

KEYF (Keyboard Flag Bit): Es un bit de solo lectura que es seteado cuando está pendiente un pedido de interrupción sobre el puerto A o el despertar automático. El Reset borra este registro.

1 = interrupción pendiente.

0 = interrupción no pendiente.

ACKK (Keyboard Acknowledge Bit): Escribiendo un 1 en este bit de solo escritura limpia los pedidos de interrupción sobre el puerto A o despertar automático. Siempre se lee como 0. El Reset también lo limpia.

IMASKK (Keyboard Interrupt Mask Bit): Escribiendo un 1 a este bit de lectura/escritura previene la generación de un pedido de interrupción desde el puerto A o despertar automático. El Reset borra este bit.

1 = pedido de interrupción enmascarado.

0 = pedido de interrupción no enmascarado.

MODEK: no tiene relación con el módulo AWU.

3.1.1. Registro de habilitación de interrupción por teclado (KBIER).

Este registro habilita o deshabilita al módulo de despertar automático para operar como una entrada de pedido de interrupción.

Address: \$001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figura 8. Registro de habilitación de interrupción por teclado (KBIER)

Aquí solo se usa el bit 6:

AWUIE: este bit de lectura/escritura habilita la interrupción de despertar automático como entrada para el pedido de interrupción. El Reset borra el AWUIE.

1 = despertar automático habilitado como entrada de pedido de interrupción.

0 = despertar automático deshabilitado como entrada de pedido de interrupción.

4. Módulo de Interfaz del Temporizador (TIM)

Esta sección describe el Módulo de interfaz de Temporizador (TIM). El TIM es un temporizador de dos canales, que proporciona referencia de tiempo con **captura de entrada (input capture)**, **comparador de salida (output compare)**, y funciones de **modulación por ancho de pulso PWM**. La Figura 10 muestra un diagrama en bloques del TIM [2] y [3].

4.1. Características

Entre las Características del TIM se incluyen las siguientes:

- Dos canales de input capture / output compare
 - Disparo mediante flanco ascendente, flanco descendente, o cualquier flanco para captura de entrada.
 - Acción de comparador de salida por seteo, borrado o conmutado.
- Generación de Modulación por ancho de pulso (PWM) por medio de buffers o sin ellos.
- Entrada de clock de TIM programable
 - Selección de divisor de la frecuencia de clock del bus interno con 7 frecuencias
 - Entrada de clock de TIM Externo
- Operación de módulo contador con límite de cuenta o de libre ejecución
- Conmutado de cualquier pin de canal por desborde
- Bits de reset y Stop en el contador del TIM

4.2. Convenciones de nombres de Pines

El TIM comparte dos pines de entrada/salida (I/O) con dos pines del puerto A. Los nombres completos de los pines del TIM se muestran en la tabla siguiente.

Tabla 1. Convenciones de nombres de pines

Nombres Genérico de pines del TIM:	TCH0	TCH1	TCLK
Nombres de pines del TIM:	PTA0/TCH0	PTA1/TCH1	PTA2/TCLK

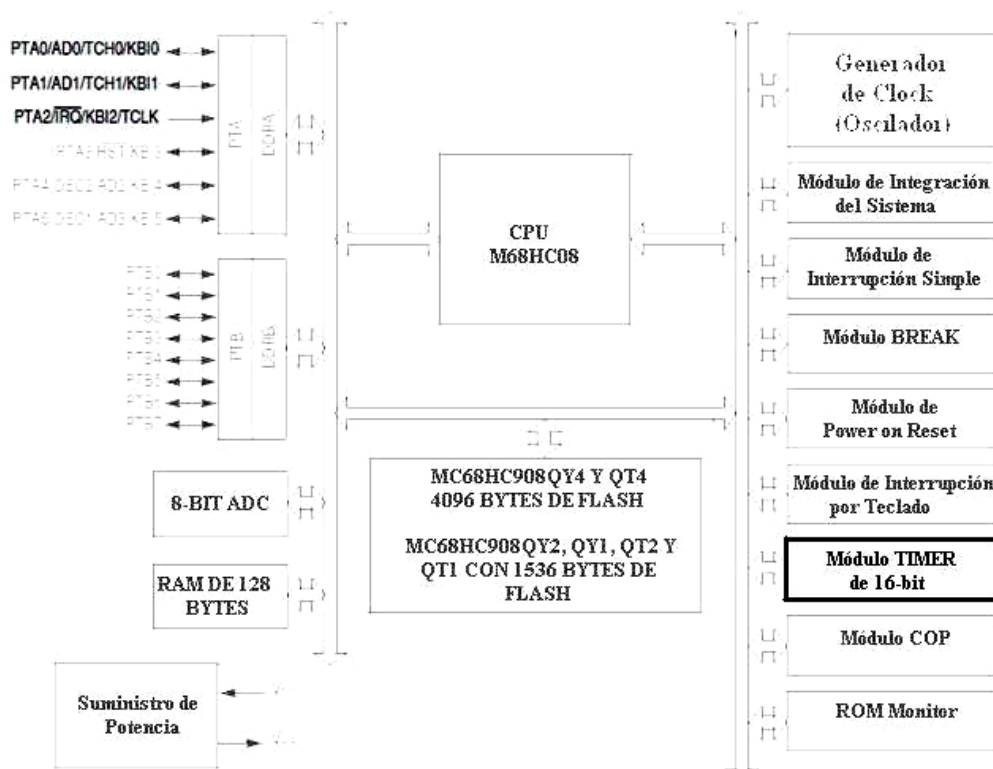


Figura 9. Diagrama en bloques que resalta los pines y bloques del Módulo TIM

4.3. Descripción Funcional

La figura siguiente muestra la estructura del TIM. El componente central del TIM es el contador de 16-bit que puede operar como un contador de ejecución libre o como un módulo contador con valor de cuenta preseleccionado. El contador del TIM provee el tiempo de referencia para las funciones de **captura de entradas** y de **comparación de salidas**. Los registros de Módulo contador, TMODH:TMODL, contienen el valor del módulo contador, indicando mediante el mismo, hasta que valor deberá incrementarse el contador. Por software se puede leer el valor del contador del TIM en cualquier momento sin afectar la secuencia de conteo. Los dos canales de TIM son programables independientemente como input capture u output compare.

4.3.1. Prescalador del contador del TIM

La fuente de clock del TIM es una de las siete salidas del prescalador o el pin de clock del TIM, TCLK. El prescalador genera siete tasas de clock desde el bus de clock interno. Los bits de selección de prescalador, PS[2:0], en el registro de control y estado del TIM (TSC) seleccionan la fuente de clock del TIM.



Con la función de captura de entrada, el TIM puede capturar el momento en que ocurre un evento externo. Cuando ocurre un flanco activo sobre el pin de un canal de captura de entrada, el TIM almacena el contenido del contador en el registro del canal del TIM, TCHxH:TCHxL. La polaridad del flanco activo es programable. La captura de entrada del TIM puede generar requerimientos de interrupción de la unidad central de procesos (CPU).

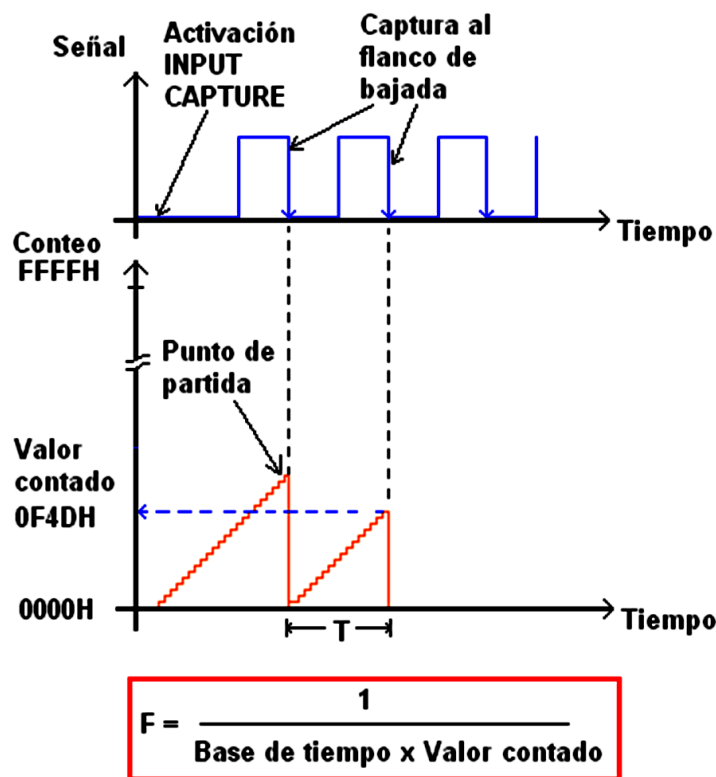


Figura 11. Esquema de funcionamiento de la “captura de entrada”

4.3.3. Comparador de Salida

Con la función de comparación de salida, el TIM puede generar un pulso periódico con polaridad, duración y frecuencia programables. En la figura 12 se muestra un esquema del funcionamiento de este Modo, cuando el contador alcanza el valor en los registros de un canal comparador de salida, el TIM puede fijar, borrar, o cambiar el estado del pin del canal. El comparador de salida del TIM puede generar requerimientos de interrupción de la CPU.

En este modo el Módulo puede generar una señal cuadrada con un ciclo de trabajo del 50% (es un modo particular del PWM).

4.3.3.1. Comparador de Salida sin Buffer

Cualquier canal comparador de Salida puede generar pulsos de salida sin buffer. Los pulsos son no bufereados porque cambiar el valor del comparador de salida requiere escribir el nuevo valor sobre el valor anterior que se encuentra en los registros del canal de TIM.

Una escritura no sincronizada de los registros del canal para cambiar un valor del comparador de salida podría causar el funcionamiento incorrecto durante 2 periodos de desborde del contador. Por ejemplo, escribir un nuevo valor antes de que el contador alcance el valor actual almacenado, pero después que el contador alcance el nuevo valor impide cualquier comparación durante ese periodo de desborde del contador. Asimismo, utilizando una rutina de interrupción de desborde para escribir un nuevo valor “menor que el anterior” del comparador de salida, quizás cause que la comparación se pierda. El TIM puede pasar el nuevo valor antes de que sea escrito.

Para sincronizar los cambios no bufereados en el valor del comparador de salida sobre el canal x se deben utilizar los siguientes métodos:

- Cuando se cambia a un valor menor, se deberá habilitar la interrupción en el comparador de salida del canal x y se deberá escribir el nuevo valor en la rutina de interrupción del comparador de salida. La interrupción del comparador de salida ocurre al final del pulso actual. La rutina de interrupción tiene hasta el final del periodo de desborde del contador para escribir el nuevo valor.
- Al cambiar a un valor mayor del comparador de salida, se deberán habilitar la interrupción por desborde y escribir el nuevo valor en la rutina de interrupción por desborde del TIM. La interrupción por desborde ocurre al final del periodo actual de desborde del contador. Al Escribir un valor mayor en la rutina de interrupción del comparador de salida (al final del pulso actual) podría causar dos comparaciones de salida a ocurrir en el mismo periodo de desborde del contador.

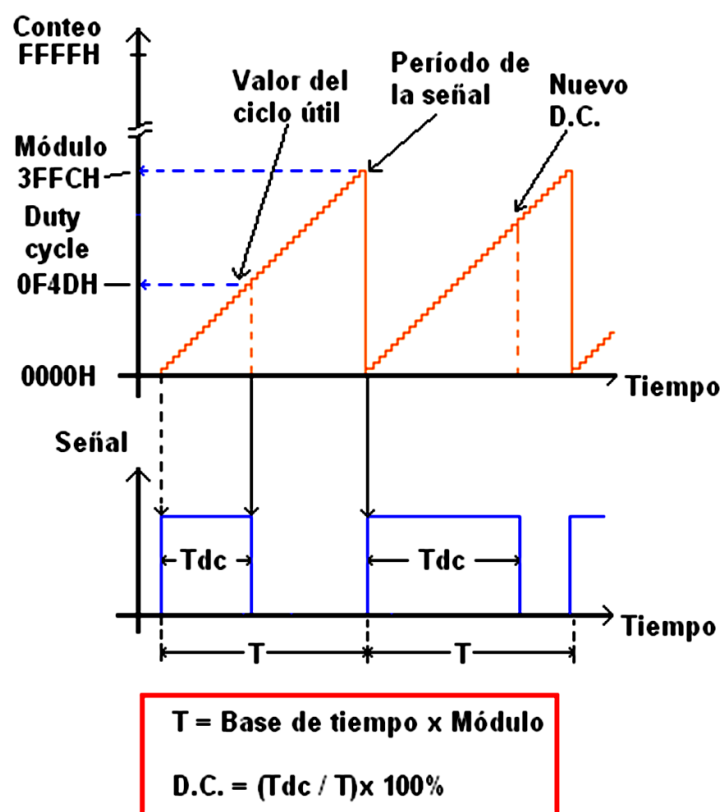


Figura 12. Esquema de Funcionamiento de Comparador de Salida / PWM

4.3.3.2. Comparador de Salida con buffer

Los Canales 0 y 1 pueden conectarse desde un canal comparador de salida bufereado cuya salida aparece sobre el pin TCH0. Los registros del canal TIM del par conectado controlan la salida alternadamente. Seteando el bit MS0B en el registro de control y estados del canal 0 (TSC0) se conecta el canal 0 y el canal 1.

El valor del comparador de salida en los registros del canal 0 controla la salida sobre el pin TCH0.

Escribiendo los registros del canal 1 se habilitan dichos registros para controlar sincrónicamente la salida después de los desbordes del TIM. Posteriormente, en cada desborde, los registros del canal (0 o 1) que controlan la salida son los últimos en escribirse. TSC0 controla y monitorea la función del comparador de salida bufereado, y quedará en desuso el registro de control y estado del canal 1 (TSC1). Mientras el bit MS0B está seteado, el pin del canal 1, TCH1, queda disponible como un pin I/O de propósito general.

Durante la operación del comparador de salida bufereado, no se escriben nuevos valores de comparación de salida a los registros del canal actualmente activo. El software del usuario deberá hacer un seguimiento del canal actualmente activo para prevenir escribir un nuevo valor a dicho canal. Escribir los registros del canal activo es similar a generar comparaciones de salidas no bufereadas.

4.3.4. Modulación por Ancho de Pulso (PWM)

En este modo el Módulo puede generar una señal cuadrada, aplicada a un pin del MCU. A la señal generada se le puede controlar el ciclo útil (Duty Cycle), convirtiéndose en una señal PWM (Pulse Width Modulation).

El ciclo útil de la señal (D.C. Duty Cycle) es programado como un porcentaje entre el 0 y el 100% del período de la señal. Generalmente el ciclo útil se controla desde un canal del módulo temporizador y deberá ser un valor menor que el período de la señal.

Usando la característica de cambio de estado por desborde con un canal comparador de salida, el TIM puede generar una señal PWM [2]. El valor en los registros TMODH:TMODL del módulo, determina el periodo de la señal PWM. El pin del canal cambia su estado cuando el contador alcanza el valor de los registros del módulo contador. El tiempo entre desbordes es el período de la señal PWM.

Como muestra la figura siguiente, el valor del “comparador de salida” en los registros del canal del TIM determina el ancho de pulso de la señal PWM. El tiempo entre el comparador de salida y el desborde es el ancho de pulso. Se Programa el TIM para poner en estado “0 lógico” el pin del canal cuando el comparador de salida alcanzó el valor seteado, si el estado del pulso PWM es un 1 lógico (ELSxA = 0). Se Programa el TIM para poner en 1 lógico el pin del canal cuando el comparador de salida alcanza el valor seteado, si el estado del pulso PWM es un 0 lógico (ELSxA = 1).

El valor en los registros del módulo contador TIM y la salida del preescalador seleccionado determinan la frecuencia de salida PWM. La frecuencia de una señal PWM de 8-bit puede variar en un rango de 256. Escribiendo \$00FF (255) a los registros del módulo contador TMODH:TMODL, se produce un periodo de PWM de 256 veces el periodo de clock del bus interno, siempre y cuando se haya seteado el preescalador con el valor 000 (Ver detalle en el registro de control y estados del TIM, apartado 4.8.1.).

El valor en los registros del canal del TIM determina el ancho de pulso de la salida PWM. Si se escribe el valor \$0080 (128) en el registro del canal del TIM se produce un ciclo de 128/256 o 50%.

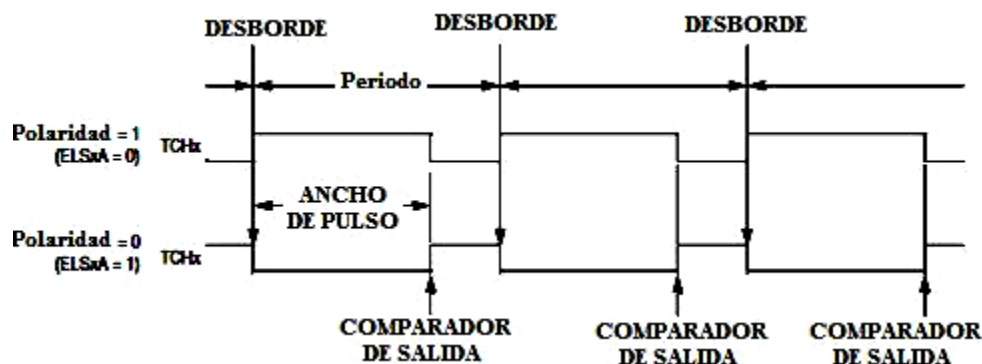


Figura 13. Periodo y ancho de pulso de PWM

4.3.4.1. Generación de señal PWM sin buffer

Cualquier canal comparador de salida puede generar pulsos PWM no bufereados. Los pulsos son no bufereados porque cambiar el ancho de pulso requiere escribir el nuevo valor de ancho de pulso sobre el valor actual en los registros del canal.

Una escritura no sincronizada de los registros del canal para cambiar el valor del ancho de pulso puede causar el mal funcionamiento durante dos periodos de PWM. La explicación de este mal funcionamiento se encuentra en la sección 4.3.3.1.

Para sincronizar los cambios no bufereados en el valor del ancho de pulso PWM sobre el canal x se deben utilizar los siguientes métodos:

- Cuando se cambia a un ancho de pulso menor, se deberá habilitar la interrupción en el comparador de salida del canal x y se deberá escribir el nuevo valor en la rutina de interrupción del comparador de salida. La interrupción del comparador de salida ocurre al final del pulso actual. La rutina de interrupción tiene hasta el final del periodo del PWM para escribir el nuevo valor.
- Al cambiar a un valor mayor del ancho de pulso, se deberán habilitar la interrupción por desborde y escribir el nuevo valor en la rutina de interrupción por desborde del TIM. La interrupción por desborde ocurre al final del periodo actual del PWM. Al Escribir un valor mayor en la rutina de interrupción del comparador de salida (al final del pulso actual) podría causar dos comparaciones de salida a ocurrir en el mismo periodo de PWM.

En la generación de señal PWM, no se programa el canal PWM para conmutar sobre el comparador de salida. Conmutando sobre el comparador de salida se impide la generación de un ciclo de trabajo confiable del 0% y se pierde la capacidad del canal de autocorrección en el caso de error de software o ruido. Conmutando sobre el comparador de salida también se puede provocar una generación de señal de PWM cuando se cambia el ancho de pulso PWM a un nuevo valor mucho mayor.

4.3.4.2. Generación de señal PWM con buffer

Los Canales 0 y 1 pueden conectarse desde un canal PWM bufereado cuya salida aparece sobre el pin TCH0. Los registros del canal TIM del par conectado controlan el ancho de pulso de la salida alternadamente. Seteando el bit MS0B en el registro de control y estados del canal 0 (TSC0) se conecta el canal 0 y el canal 1.

Los registros del canal 0 controlan el ancho de pulso sobre la salida del pin TCH0.

Escribiendo los registros del canal 1 se habilitan dichos registros para controlar sincrónicamente el ancho de pulso a iniciar en el próximo periodo PWM. En cada desborde, los registros del canal (0 o 1) que controlan el ancho de pulso son los últimos en escribirse. TSC0 controla y monitorea la función PWM bufereada, y dejará en desuso el registro de control y estado del canal 1 (TSC1). Mientras el bit MS0B está seteado, el pin del canal 1, TCH1, queda disponible como un pin I/O de propósito general.

Durante la generación de PWM bufereada, no se escriben nuevos valores de comparación de salida a los registros del canal actualmente activo. El software del usuario deberá hacer un seguimiento del canal actualmente activo para prevenir la escritura de un nuevo valor a dicho canal. Escribir los registros del canal activo es similar a generar señales PWM no bufereadas.

4.3.4.3. Inicialización del PWM

Para asegurar una correcta operación en la generación de señales PWM bufereadas o no, se debe seguir el siguiente procedimiento de inicialización:

1. En el Registro de control y estados del TIM (TSC):
 - a. Detener el contador del TIM seteando el bit de stop, TSTOP.
 - b. Resetear el contador y el preescalador seteando el bit de reset del Tim, TRST.
2. En los registros del modulo contador del TIM (TMODH:TMODL), se debe escribir el valor para el periodo de PWM deseado.
3. En los registros del canal x (TCHxH:TCHxL), se debe escribir el valor del ancho de pulso deseado.
4. En el registro de control y estados del canal x (TSCx):
 - a. Se debe escribir 0:1 (para comparación de salida o señales PWM no bufereadas) o 1:0 (para comparación de salida o señales PWM bufereadas) para los bits de selección de modo, MSxB:MSxA. Ver la tabla 3.
 - b. Se debe escribir 1 al bit de “conmutado sobre desborde” (toggle-on-overflow), TOVx.
 - c. Se debe escribir 1:0 (polaridad 1 — para borrar la salida en la comparación) o 1:1 (polaridad 0 — para setear la salida en la comparación) para seleccionar el flanco/nivel en los bits, ELSxB:ELSxA. La acción de salida en la comparación debe forzar la salida al complemento del nivel del ancho de pulso. Ver tabla 3.
5. En el registro de control y estados del TIM (TSC), se debe borrar el bit de stop del TIM, TSTOP.

Seteando la opción de conexión MS0B de los canales 0 y 1 y configurándolos para la operación de PWM bufereada. Los registros del canal 0 (TCH0H:TCH0L) controlan la salida del PWM bufereado. El registro de control y estados del TIM 0 (TSCR0) controla y monitorea la señal PWM desde los canales linkados. MS0B tiene prioridad sobre MS0A.

Borrando el bit de “cambio en desborde” (toggle-on-overflow), TOVx, se inhiben los cambios de la salida sobre los desbordes del TIM. Subsecuentemente el comparador de salida trata de forzar la salida a un estado quedando fijo y sin efecto. El resultado es una salida con ciclo de trabajo al 0%.

Seteando el bit de máximo ciclo de trabajo del canal x (CHxMAX) y seteando el bit TOVx se genera una salida a un ciclo de trabajo del 100%.

4.4. Interrupciones

Las siguientes Fuentes de TIM pueden generar requerimientos de interrupciones:

- **Flag de Desborde del TIM (TOF)** — El bit TOF es seteado cuando el contador del TIM alcanza el valor programado en los registros del Módulo contador. El bit de habilitación de interrupción por desborde del TIM, TOIE, habilita el requerimiento de interrupción del CPU por desborde del TIM. TOF y TOIE se encuentran en el Registro de control y estados del TIM.
- **Flags de canal del TIM (CH1F:CH0F)** — El bit CHxF es seteado cuando ocurre una captura de entrada o una comparación de salida sobre el canal x. El requerimiento de Interrupción del CPU en el canal x del TIM es controlado por el bit de habilitación de interrupción del canal x, CHxIE. El requerimiento de interrupción del CPU en el canal x del TIM se habilita cuando CHxIE =1. CHxF y CHxIE se encuentran en el registro de control y estados del canal x del TIM.

4.5. Modo Wait

La instrucción WAIT pone al MCU en modo de espera de bajo. El TIM se mantiene activo después de la ejecución de una instrucción WAIT. En modo Wait el registro TIM no puede ser accedido desde el CPU. Cualquier requerimiento de interrupción del CPU desde el TIM puede sacar al MCU fuera del modo wait.

4.6. TIM durante el requerimiento de Interrupciones

Un quiebre de interrupciones detiene el contador del TIM. En el módulo de integración del sistema (SIM) tanto los bits de control y estado en otros módulos pueden ser borrados durante el estado de quiebre. El bit BCFE en el registro de control del flag de quiebre (BFCR) habilita al software a borrar los bits de estados durante el estado de quiebre.

Para permitir al software borrar el bit de estados durante un quiebre de interrupción, se debe escribir un 1 en el bit BCFE. Si el bit de estado es borrado durante el estado de quiebre, se mantendrá borrado cuando el MCU salga del estado de quiebre.

Para proteger los bits de estado durante el estado de quiebre, se debe escribir un 0 al bit BCFE. Con BCFE a 0 (estado por defecto), el software puede leer y escribir los registros I/O durante el estado quiebre sin afectar los bits de estados. Algunos bits de estado (lectura/escritura) tienen un procedimiento de borrado de 2 pasos. Si el software realice el primer paso sobre un bit antes del quiebre, el bit no podrá cambiar durante el estado de quiebre mientras BCFE este en 0. Luego del quiebre, realizando el Segundo paso se borra el bit de estado.

4.7. Señales de Entrada/Salida

El Puerto A comparte tres de sus pines con el TIM. Los dos canales de TIM están en PTA0/TCH0 y PTA1/TCH1 y una fuente de clock alternativa está en PTA2/TCLK.

4.7.1. Pin de Clock del TIM (PTA2/TCLK)

PTA2/TCLK es un clock de entrada externo que puede ser una fuente de clock del contador en lugar del preescalador del clock de bus interno. Se selecciona la entrada PTA2/TCLK escribiendo unos a los 3 bits del preescalador, PS[2–0].

4.7.2. Pines I/O de los canales del TIM (PTA0/TCH0 y PTA1/TCH1)

Cada pin I/O del canal es programable independientemente de que sea un pin de captura de entrada o un comparador de salida. PTA0/TCH0 puede ser configurado como una salida bufereada del comparador de salida o un pin PWM bufereado.

4.8. Registros de Entrada / Salida

Los siguientes Registros I/O controlan y monitorean la operación del TIM:

- Registro de control y estados del TIM (TSC)
- Registro Contador del TIM (TCNTH:TCNTL)
- Registros de Módulo contador del TIM (TMODH:TMODL)
- Registros de control y Estados del Canal (TSC0 and TSC1)
- Registros del Canal de TIM (TCH0H:TCH0L y TCH1H:TCH1L)

4.8.1. Registros de Control y Estado del TIM

Los Registros de control y estados del TIM (TSC) son los siguientes:

- Habilitación de interrupciones por desborde del TIM
- Flags de desborde del TIM
- Stop el contador del TIM
- Resetea el contador del TIM
- Preescalador de clock para el contador del TIM

Address: \$0020

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST				
Reset:	0	0	1	0	0	0	0	0


 = Unimplemented

Figura 14. Registro de control y estados del TIM (TSC)

TOF — Bit de Flag de desborde del TIM: Este flag de lectura/escritura se setea cuando el contador alcanza el valor programado en los registros del Módulo contador. Para borrar el TOF se realiza una lectura del registro de control y estados verificando si el bit está seteado, y en ese caso se le escribe un cero. Si ocurre otro desborde antes de completar la secuencia de borrado, escribir un cero al TOF no tendrá efecto. De cualquier manera, un requerimiento de interrupción de TOF no se puede ignorar borrando el TOF. El reset borra el bit de TOF. Escribir un 1 al TOF no tendrá efecto.

- 1 = El contador del TIM ha alcanzado el valor del Módulo
0 = El contador del TIM no ha alcanzado el valor del Módulo

TOIE — Bit de habilitación de interrupción por desborde del TIM: Este bit de lectura/escritura habilita la interrupción por desborde del TIM cuando el TOF está seteado. El reset borra el bit TOIE.

- 1 = Habilita la Interrupción por desborde del TIM
0 = Deshabilita la interrupción por desborde del TIM

TSTOP — Bit de stop del TIM: Este bit de lectura/escritura detiene el contador del TIM. Se reanuda la cuenta cuando el TSTOP es borrado. El reset setea el bit TSTOP, deteniendo el contador del TIM hasta que el software borre dicho bit.

- 1 = Detiene el contador del TIM
0 = Activa el contador del TIM

TRST — Bit de reset de TIM: Seteando este bit de solo escritura se resetea el contador y el preescalador del TIM. El seteo de TRST no tiene efecto en ningún otro registro. La cuenta se reanuda desde \$0000. TRST se borra automáticamente después de que el contador del TIM es reseteado y siempre se lee como un 0. El reset borra el bit TRST.

- 1 = Borra el preescalador y el contador del TIM
0 = No tiene efecto

Nota: Seteando el TSTOP y el TRST simultáneamente se detiene el contador del TIM en un valor de \$0000.

PS[2:0] — Bits de Selección del preescalador: Estos bits de lectura/escritura seleccionan ya sea el pin PTA2/TCLK o una de las 7 salidas del preescalador como entrada al contador del TIM como muestra la Tabla 2. El reset borra los bits PS[2:0].

Tabla 2. Selección de pre-escalador

PS2	PS1	PS0	Fuente de Clock del TIM
0	0	0	Clock de bus interno ÷ 1
0	0	1	Clock de bus interno ÷ 2
0	1	0	Clock de bus interno ÷ 4
0	1	1	Clock de bus interno ÷ 8
1	0	0	Clock de bus interno ÷ 16
1	0	1	Clock de bus interno ÷ 32
1	1	0	Clock de bus interno ÷ 64
1	1	1	PTA2/TCLK

4.8.2 Registros Contador del TIM

Los 2 registros contadores de solo lectura contienen la parte alta y baja del valor en el contador del TIM. Leyendo el byte de la parte alta (TCNTH) se mantiene el contenido de la parte baja (TCNTL) en un buffer. Subsecuentemente leyendo TCNTL no se afecta el valor guardado de TCNTH hasta que TCNTH se lea. El reset borra los registros del contador del TIM. Seteando el bit de reset del TIM (TRST) también se borran los registros del contador.

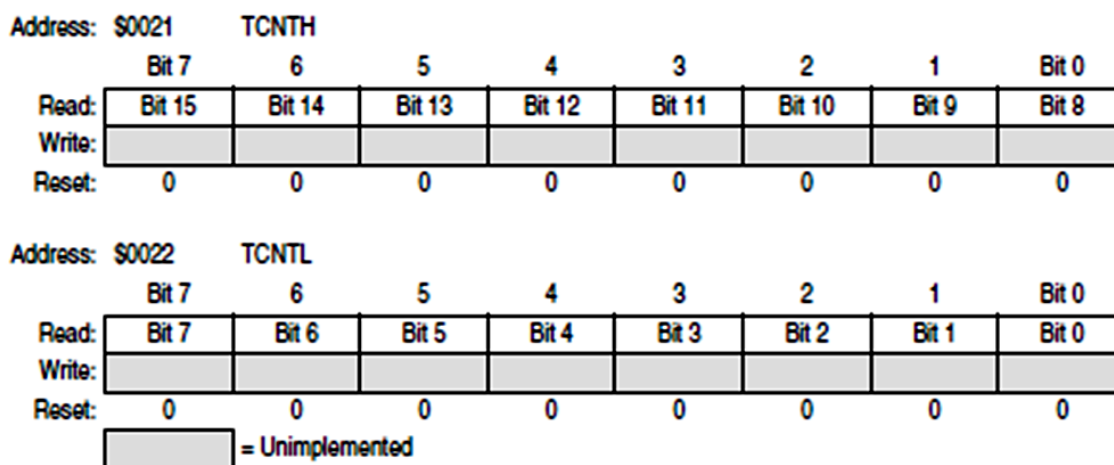


Figura 15. Registro Contador del TIM (TCNTH:TCNTL)

4.8.3 Registros del Módulo Contador del TIM

Los registros de lectura/escritura del Módulo TIM contienen el valor del módulo del contador del TIM. Cuando el contador del TIM alcanza el valor del Módulo, se setea el flag de desborde (TOF), y el contador reanuda la cuenta desde \$0000 hasta la próxima temporización de clock. Escribiendo el byte de la parte alta (TMODH) se inhibe el bit de TOF y la interrupción por desborde hasta la escritura del byte de la parte baja (TMODL). El reset setea los bits del modulo contador del TIM. Siempre es conveniente resetear los registros del cantador antes de la escritura de los registros del módulo contador.

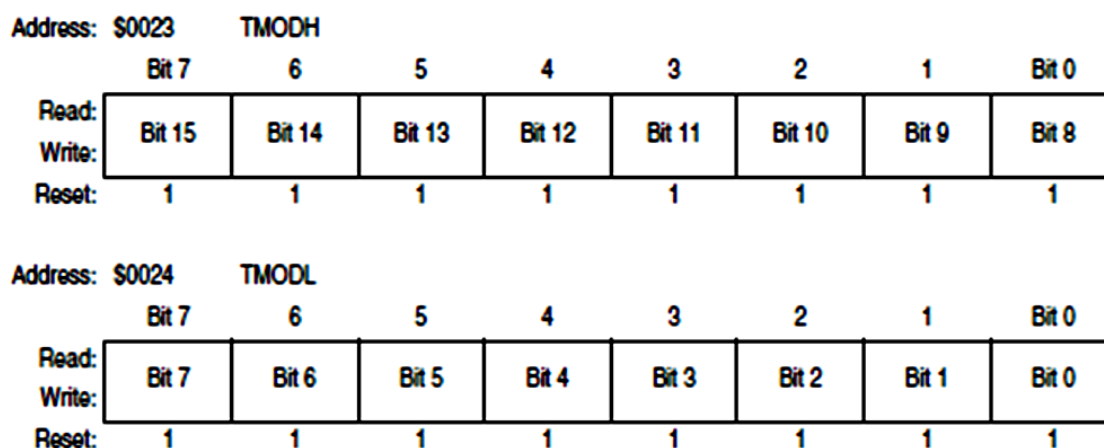


Figura 16. Registros del Módulo Contador del TIM (TMODH:TMODL)

4.8.4. Registros de control y Estado del canal del TIM

Cada uno de los registros de control y estado del canal de TIM están formados por:

- Flags del comparador de salidas y del capturador de entradas
- Habilidad de interrupciones del capturador de entrada y del comparador de salidas
- Selección de opciones de captura de entrada, comparador de salida, o PWM
- Selección de salida alta, baja, o conmutada sobre el comparador de salida

- Selección de disparo por flanco ascendente, flanco descendente o cualquier flanco en el captador de entrada activo.
- Selección de conmutado de Salida sobre el desborde del TIM
- Selección de ciclo de trabajo de PWM de 0% y 100%
- Selección de operación buffereada o no del comparador de salida/PWM.

Address: \$0025		TSC0						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Address: \$0028		TSC1						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figura 17. Registros de control y estado del canal del TIM (TSC0:TSC1)

CHxF — Bit de Flag del canal x: Cuando el canal x es un captador de entrada, este bit de escritura/lectura se setea cuando un flanco activo ocurre sobre el pin del canal x. Cuando el canal x es un comparador de entrada, CHxF se setea cuando el valor de los registros del contador del TIM coincide con el valor en los registros del canal x del TIM.

Para borrar el CHxF se debe verificar mediante la lectura del registro de control y estado del canal x que el CHxF esté seteado y luego escribir un 0 al mismo. Si ocurre otro requerimiento de interrupción antes de completada la secuencia de borrado, la escritura de un 0 a CHxF no tendrá efecto. De cualquier manera, un requerimiento de interrupción no se puede ignorar borrando el CHxF. El reset borra el bit CHxF. Escribir un 1 al CHxF no tendrá ningún efecto.

1 = Captura de entrada o comparación de Salida sobre el canal x

0 = No hay captura de entrada o comparación de Salida sobre el canal x

CHxIE — Bit de habilitación de Interrupción del Canal: Este bit de lectura/escritura habilita el requerimiento del servicio de interrupción del CPU sobre el canal x del TIM. El reset borra el bit CHxIE.

1 = Habilita el requerimiento de Interrupción del CPU en el Canal x

0 = Deshabilita el requerimiento de Interrupción del CPU en el Canal x

MSxB — Bit de selección de Modo B: Este bit de lectura escritura selecciona la operación de comparador de salida/PWM buffereado. El MSxB existe solo en el registro de control y estado del canal 0 del TIM.

Seteando el MS0B se deshabilita el registro de control y estado del canal 1 y se devuelve la condición de pin de entrada/salida de propósito general al TCH1. El reset borra el bit MSxB.

1 = Habilita la operación de comparador de salida/PWM buffereada

0 = Deshabilita la operación de comparador de salida/PWM buffereada

MSxA — Bit de selección de Modo A: Cuando ELSxB:A \neq 00, este bit de lectura/escritura selecciona ya sea la operación de captura de entrada o la operación de comparador de salida/PWM no bufereada. Ver tabla 3.

- 1 = Operación comparador de Salida/PWM no bufereada
- 0 = Operación de Captura de Entrada

Cuando ELSxB:A = 00, este bit de lectura/escritura selecciona el nivel inicial de salida del pin TCHx (Ver tabla 3). El reset borra el bit MSxA.

- 1 = Salida inicial en nivel bajo
- 0 = Salida inicial en nivel alto

Nota: Antes de cambiar la función del canal por escritura del bit MSxB o MSxA, se debe setear los bits de TSTOP y TRST en el registro de control y estado del TIM (TSC).

Tabla 3. Selección de modo, flanco y nivel

MSxB	MSxA	ELSxB	ELSxA	MODO	Configuración
x	0	0	0	preset de Salida	Pin bajo control del puerto; nivel de salida inicial en alto
x	1	0	0		Pin bajo control del puerto; Nivel de salida inicial en bajo
0	0	0	1	Captura de Entrada	Captura sobre flanco ascendente
0	0	1	0		Captura sobre flanco descendente
0	0	1	1		Captura sobre flanco ascendente o descendente
0	1	0	0	Comparador de Salida o PWM	Solo compara por software
0	1	0	1		Cambia la salida sobre el comparador
0	1	1	0		Borra la salida sobre el comparador
0	1	1	1		Setea la salida sobre el comparador
1	x	0	1	Comparador de Salida Bufereado PWM bufereado	Cambia la salida sobre el comparador
1	x	1	0		Borra la salida sobre el comparador
1	x	1	1		Setea la Salida sobre el comparador

ELSxB y ELSxA — Bits de selección de Flanco/Nivel: Cuando el canal x es un capturador de entrada, estos bits de lectura/escritura controlan la lógica de sensado de flanco sobre el canal x.

Cuando el canal x es un comparador de salida, ELSxB y ELSxA controlan el comportamiento de la salida del canal x cuando ocurre una comparación.

Cuando ELSxB y ELSxA son borrados, el canal x no se conecta al Puerto de entrada/salida, y el pin TCHx queda disponible como un pin de propósito I/O. La tabla 3 muestra cómo funcionan ELSxB y ELSxA. El reset borra los bits ELSxB y ELSxA.

Nota: Después de habilitar el registro del canal del TIM para la operación de captura de entrada y Seleccionando la sensibilidad del flanco, se debe borrar el CHxF para ignorar cualquier flag de detección de flanco erróneo.

TOVx — Bit de Cambio sobre Desborde: Cuando el canal x es comparador de salida, este bit de lectura/escritura controla el comportamiento de la salida del canal x cuando el contador del TIM desborda. Cuando el canal x es capturador de entrada, el bit TOVx no tiene efecto. El Reset borra el bit TOVx.

- 1 = El pin del canal x cambia en el desborde del contador del TIM.
- 0 = El canal no cambia en el desborde del contador.

CHxMAX — Bit de ciclo de trabajo Máximo del Canal x: Cuando el bit TOVx está en 1, seteando el bit CHxMAX se fuerza a un ciclo de trabajo del 100% de la señal PWM bufereada y no buffereada. Como muestra la Figura 18, el bit CHxMAX tiene efecto un ciclo después de ser seteado o borrado. La salida queda a un nivel de ciclo de trabajo del 100% hasta un ciclo después de haber borrado el bit CHxMAX.

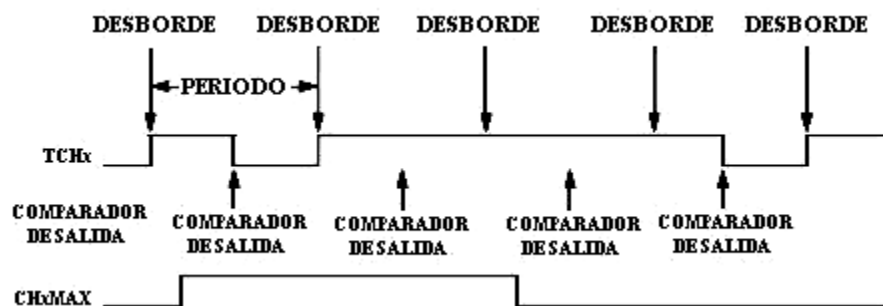


Figura 18. Ciclo de Trabajo máximo del canal x.

4.8.5. Registros del Canal de TIM

Estos registros de lectura/escritura contienen el valor del contador del TIM capturado de la función del Capturador de entrada o el valor de la comparación de salida de la función del comparador de salida. El estado del canal del Tim luego del Reset es desconocido.

Address: \$0026	TCH0H	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:									
Reset:		Indeterminate after reset							

Address: \$0027	TCH0L	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:									
Reset:		Indeterminate after reset							

Address: \$0029	TCH1H	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:									
Reset:		Indeterminate after reset							

Address: \$02A	TCH1L	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:									
Reset:		Indeterminate after reset							

Figura 19. Registros de Canal TIM (TCH0H/L:TCH1H/L)



En modo de captura de entrada ($MSxB:MSxA = 0:0$), leyendo el byte más significativo de los registros del canal x del TIM (TCHxH) se inhibe la captura de entrada hasta que se lea el byte menos significativo (TCHxL).

En el modo Comparador de Salida ($MSxB:MSxA \neq 0:0$), escribiendo el byte más significativo de los registros del canal x de TIM (TCHxH) se inhibe la comparación de salida hasta que se escriba el byte menos significativo (TCHxL).

5. Módulo de Interrupción por Teclado (KBI)

El módulo de interrupción por teclado permite disponer de seis interrupciones externas independientes, accesibles a través de los pines 0 a 5 del puerto A (PTA0 a PTA5).

5.1. Características

Este módulo incluye:

- Seis pines de interrupción por teclado con bits de habilitación separados y uno de enmascaramiento de interrupción.
- Dispositivo de pull-up configurable por software en el caso de que el bit del puerto sea configurado como entrada.
- Sensibilidad de la interrupción programable a flanco solo o flanco y nivel.
- Salida desde modos de bajo consumo.

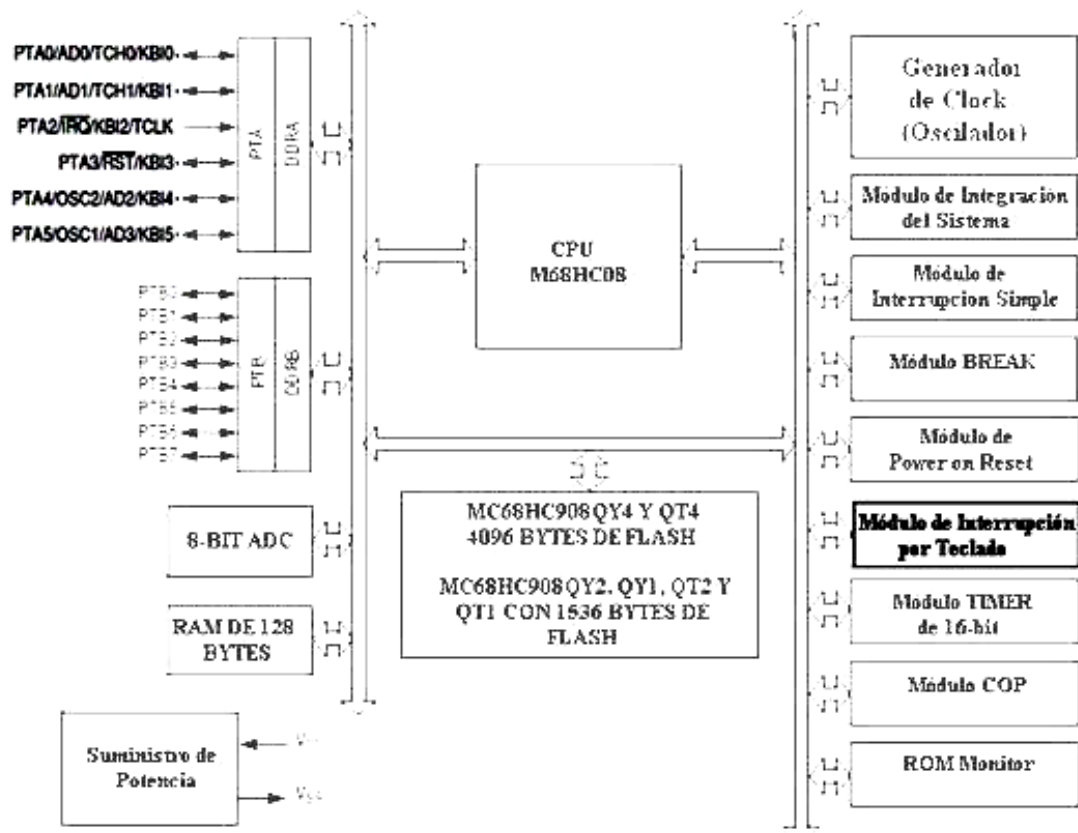


Figura 20. Diagrama en Bloques del HC908QY4 con el KBI resaltado.

5.2. Descripción funcional

El módulo de interrupción por teclado controla la habilitación y deshabilitación de las funciones de interrupción sobre seis pines del puerto A de forma independiente uno del otro.

5.2.1. Operación de Teclado

Escribiendo los bits KBIE0-KBIE5 en el registro de habilitación de interrupción por teclado, se habilita o deshabilita, independientemente, cada pin del puerto A como un pin de interrupción por teclado, esto activa automáticamente el respectivo dispositivo interno de pull-up desde el bit del registro de habilitación de entradas del puerto A (PTAPUEX).

Una interrupción por teclado se produce cuando una o más de éstas entradas se pone en estado bajo “0 lógico” después de estar todas en estado alto (1 lógico).

El bit MODEK del registro de control y estado de teclado, controla el modo de disparo de la interrupción por teclado:

- Si la interrupción por teclado es solo sensible a flanco, un flanco de bajada sobre el pin no almacena un pedido de interrupción si otro pin ya está en nivel bajo. Para prevenir la pérdida de un pedido de interrupción, el software puede deshabilitar la última entrada mientras está en nivel bajo.
- Si la interrupción por teclado es sensible al flanco descendente y nivel bajo, se producirá un requerimiento de interrupción cuando cualquier entrada de interrupción esté en nivel bajo.

Si el bit MODEK está seteado, las entradas de interrupción por teclado son sensibles al “flanco descendente” y “nivel bajo” y para borrar un pedido de interrupción se deben cumplir las siguientes acciones:

- Búsqueda de vector o borrado por software: una búsqueda de vector genera una señal de reconocimiento de interrupción que borra el pedido de interrupción. El software puede generar esta señal de reconocimiento, escribiendo 1 en el bit de ACKK en el registro de control y estado de teclado (KBSCR); tal bit es útil en aplicaciones que chequean continuamente las entradas de interrupción por teclado (poll) y necesitan que el software borre los pedidos. Escribiendo el bit ACKK antes de dejar la rutina del servicio de interrupción se pueden prevenir las interrupciones espurias debidas al ruido. Setear este bit ACKK no afecta las subsecuentes transiciones sobre las entradas de interrupción por teclado, un flanco de bajada que ocurre después de escribir este bit almacena otro pedido de interrupción. Si se borra el bit de máscara de interrupción por teclado IMASKK, la unidad central de proceso (CPU) carga el contador de programa con la dirección del vector en las direcciones \$FFE0 y \$FFE1.
- Retorno de todas las entradas de interrupción por teclado a 1 lógico (mientras cualquiera de los pines de entrada de interrupción por teclado esté en 0, la interrupción permanece seteada).

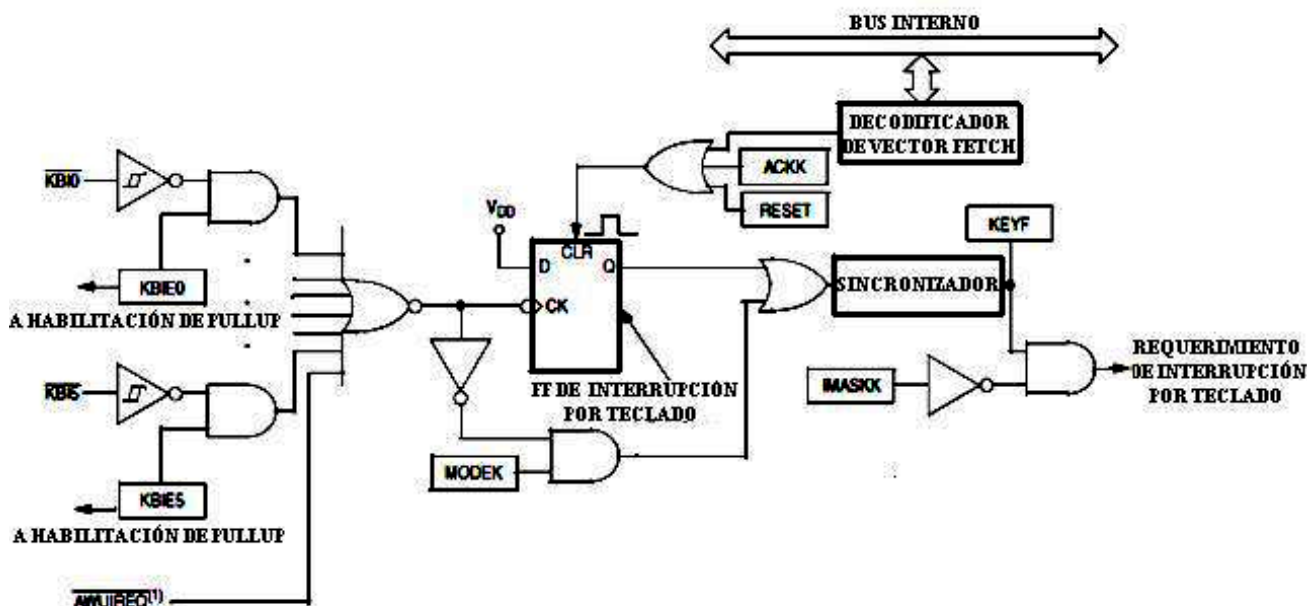


Figura 21. Diagrama en bloques del Sistema de Interrupción por Teclado

Si el bit MODEK está borrado, el pin de interrupción por teclado es sensible sólo a flanco descendente. Con este bit en cero, una búsqueda de vector o borrado por software, limpia inmediatamente los pedidos de interrupción.

Un RESET borra todos los pedidos de interrupción y el bit MODEK, aún cuando una entrada de interrupción permanezca en 0 lógico.

El bit de flag de teclado KEYF en el registro de control y estado de teclado puede ser usado para ver si queda pendiente un pedido de interrupción. El bit KEYF no se ve afectado por el bit de enmascaramiento de interrupción IMASKK lo que lo hace útil en aplicaciones donde se prefiere la encuesta (polling).

Para determinar el nivel lógico sobre un pin de interrupción por teclado, se usa el registro de dirección de datos para configurar el pin como entrada y luego leer el registro de datos.

En este caso, fijando un bit de habilitación de interrupción por teclado se fuerza al pin correspondiente a ser una entrada, con predominio por sobre el "registro de dirección de datos DDR, sin embargo el bit respectivo de este registro debe estar en 0 por software para poder leer el pin.

5.2.2. Inicialización de Teclado

Cuando un pin de interrupción por teclado se habilita, lleva un tiempo para que el pullup interno se ponga en 1 lógico, tiempo durante el cual puede ocurrir una falsa interrupción.

Para prevenir esta falsa interrupción se debe inicializar con:

1. Enmascarar la interrupción por teclado seteando el bit IMASKK en el registro de control y estado de interrupción por teclado.
2. Habilitar los pines KBI seteando los bits de KBIEx apropiados en el registro de habilitación de interrupción por teclado.

3. Escribir el bit ACKK en el registro de control y estado de interrupción por teclado para limpiar cualquier falsa interrupción.
4. Borrar el bit IMASKK.

Una señal de interrupción sobre un pin disparado por flanco puede ser aceptado inmediatamente después de habilitar el pin. En cambio, una señal de interrupción sobre un pin disparado por flanco y nivel debe ser aceptado después de un retardo que depende de la carga externa.

Otra manera de evitar una interrupción falsa puede ser:

1. Configurar los pines como salida seteando apropiadamente los bits del registro de dirección de datos A (DDRA).
2. Escribir unos en los bits apropiados del registro de datos del puerto A.
3. Habilitar los pines KBI seteando los bits apropiados del registro de habilitación de interrupción por teclado.

5.3. Modos Wait y Stop

El módulo de teclado permanece activo durante el modo Wait. Borrando el bit IMASKK del registro de control y estado de teclado, habilita el pedido de interrupción por teclado para sacar al Microcontrolador del modo Wait.

El módulo de teclado permanece activo también durante el modo de “parada o Stop”. De la misma manera que en modo espera, borrar el IMASKK habilita el pedido de interrupción para sacar al Microcontrolador del modo de stop.

5.4. Módulo Keyboard (Teclado) Durante el requerimiento de Interrupciones

El sistema de integración de módulos (SIM) controla si la interrupción de teclado almacenada puede ser borrada durante un estado de pausa o break.

El bit BCFE en el registro de control de flag de pausa (BFCR) habilita al software para limpiar los bits de estado durante un modo pausa. Para esto se debe escribir un uno en el bit BCFE. Si una interrupción almacenada (latch) se limpia durante un estado de pausa, permanece limpia aun cuando el Microcontrolador sale de este modo de pausa.

Para proteger el pedido de interrupción almacenado durante un break se debe poner a 0 el bit BCFE (que es su estado por defecto). De esta manera, escribir el bit de reconocimiento de interrupción por teclado ACKK del registro de control y estado de teclado durante un estado de break, no tiene efecto alguno.

5.5 Registros de entrada/Salida

Los siguientes registros de entrada y salida controlan y monitorean las operaciones sobre el módulo de interrupción por teclado:

- Registro de Control y Estado de Interrupción por Teclado (KBICSR).
- Registro de habilitación de interrupción por teclado (KBIER).

5.5.1 Registro de control y estados del Teclado

Este registro cuenta con:

- Flags de pedidos de interrupción por teclado.
- Reconocimientos de pedidos de interrupción por teclado.
- Enmascaramiento de pedidos de interrupción.
- Control del tipo de sensibilidad de disparo de las interrupciones por teclado.

Address: \$001A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figura 22. Registro de control y estados del teclado (KBSCR)

Bits 7–4: No son usados, son de solo lectura y permanecen en 0 por defecto.

KEYF —Bit de flag de teclado: Este bit de solo lectura es seteado cuando está pendiente un pedido de interrupción sobre el puerto A o auto wake-up. Un RESET borra este bit.

1 = interrupción por teclado pendiente.

0 = no hay interrupción por teclado pendientes.

ACKK — **Bit de confirmación de Teclado:** Bit de reconocimiento de interrupción por teclado, escribiendo un 1 en este bit de sólo escritura, se borran los pedidos de interrupción por teclado sobre el puerto A y auto wake-up. Al ser de solo escritura siempre se lee como 0 y el RESET lo borra.

IMASKK—**Bit de enmascaramiento de interrupción por teclado:** Escribiendo 1 en este bit de lectura/escritura se previene que se produzcan pedidos de interrupción por teclado o auto wake-up.

1 = pedidos de interrupción enmascarados.

0 = pedidos de interrupción no enmascarados.

MODEK — **Bit de Sensibilidad de Disparo del Teclado:** Este bit de lectura/escritura controla la sensibilidad de disparo de los pines de interrupción por teclado sobre el puerto A y despertar automático. RESET limpia este bit.

1 = pedidos de interrupción por teclado se producen en flancos de bajada y nivel bajo.

0 = pedidos de interrupción por teclado se producen sólo en flancos de bajada.

5.5.2. Registro de Habilidad de Interrupción por Teclado

El registro de habilitación de interrupción por teclado del puerto A habilita o deshabilita cada pin del puerto A o despertar automático para que funcione como una entrada de interrupción por teclado.

Address: \$001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0
	= Unimplemented							

Figura 23. Registro de habilitación de interrupción por teclado (KBIER)

KBIE5–KBIE0 — Bits de Habilitación de interrupción por Teclado del Puerto A: Cada uno de estos bits de lectura/escritura habilita al correspondiente pin asociado sobre el puerto A para realizar un pedido de interrupción. RESET limpia el registro.

1 = el pin KBIX es habilitado como pin de interrupción.

0 = el pin KBIX no está habilitado como pin de interrupción.

Nota: El bit AWUIE no se usa en conjunto con las interrupciones por teclado.

6. Módulo Conversor Analógico/Digital

6.1. Introducción

Antes de la descripción del módulo ADC del microcontrolador se analizarán las características y la utilidad de los conversores AD.

Como se describió anteriormente, los dispositivos Microcontroladores son puramente digitales, lo que no quita que puedan procesar señales analógicas. Para el procesamiento de señales analógicas en los MCUs se utilizan los Conversores AD, lo cuales digitalizan las señales analógicas para su posterior procesamiento dentro del microprocesador.

El proceso de convertir introduce una inevitable pérdida de información. Esta pérdida es inherente al proceso de discretizar las señales análogas y continuas, que finalmente serán llevadas a cantidades binarias.

La Figura 24 representa una señal análoga continua entre los puntos t_0 y t_1 , que desde el punto de vista de la magnitud será sometida a un número discreto de valores binarios y que la cantidad de valores se conoce con el nombre de Resolución del sistema [6].

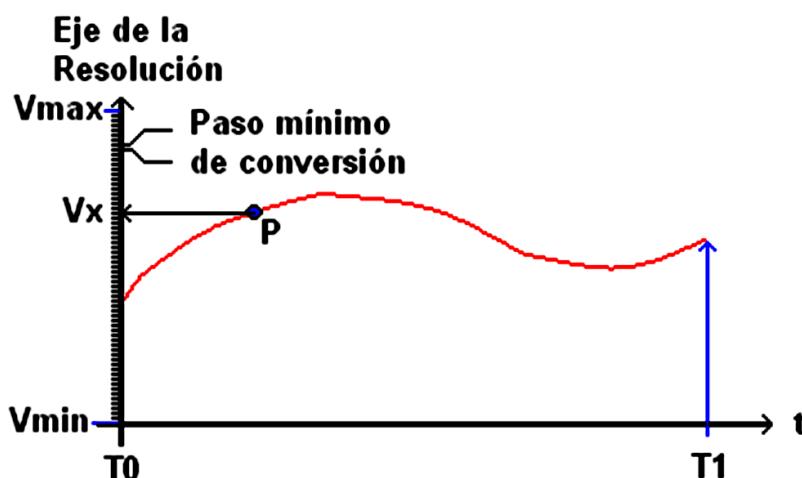


Figura 24. Eje de resolución en la conversión AD

La Resolución de un conversor analógico a digital es la cantidad de valores, discretos, en los cuales se interpreta la señal a digitalizar. Por ejemplo, para un procesador con un conversor A/D que tiene una resolución de 12 bits el número de valores discretos en los cuales se puede valorar a una señal, sería de $2^{12} = 4096$. El valor ideal para esos valores sería un número infinito, pero tecnológicamente es imposible.

Esos valores deben estar comprendidos dentro de dos límites, que forman la ventana de conversión o valores de referencia (V_{min} , V_{max}). Para la Figura 24, el punto P tiene una interpretación en el mundo de lo discreto y es de V_x .

Se recomienda que el usuario aproveche al máximo la resolución del sistema, adecuando la señal análoga para que excursione de la manera más completa en la ventana de conversión. Por ejemplo, una señal con un valor máximo de 100mV deberá ser amplificada por un factor de 30, para una ventana de conversión de 3V y de ésta manera aprovechar la resolución del sistema.

Para calcular el paso mínimo de conversión y por otro lado conocer el intervalo de pérdida de información, supóngase que se tiene una señal sometida a un conversor de 12 bits de resolución, un $V_{min} = 0V$ y un $V_{max} = 5V$. El paso mínimo de conversión está dado por:

$$\text{Paso mínimo} = (V_{\text{max}} - V_{\text{min}}) / \text{Resolución}$$

$$\text{Paso mínimo} = (5V - 0V) / 212$$

$$\text{Paso mínimo} = 1.22\text{mV}$$

El cálculo anterior indica que la diferencia en magnitud entre el resultado de una conversión y la inmediatamente superior (o inferior) es de 1.22mV. Todo valor que no sea múltiplo entero de un paso mínimo, se deberá aproximar al valor más cercano y es allí donde un conversor A/D ignora información del mundo analógico.

El Muestreo (sample) es otra característica importante de un conversor A/D y se refiere a la cantidad de muestras en la unidad de tiempo que se pueden procesar y convertir a cantidades discretas.

La Figura 25 presenta una señal analógica continua entre los puntos t_0 y t_1 , que desde el punto de vista del muestreo es sometida a un número finito de muestras y que cada muestra es tomada a un intervalo constante T , llamado período de muestreo [7].

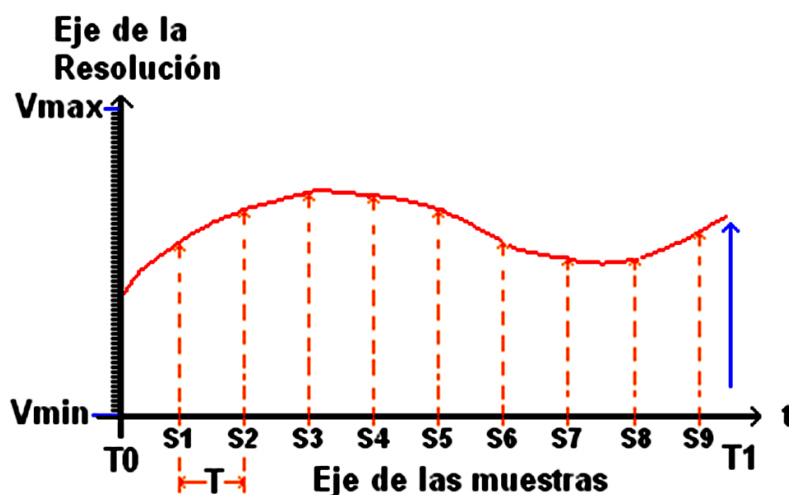


Figura 25. Muestreo en la conversión AD.

Al igual que en la resolución el muestreo introduce pérdida de información, debido a los valores que no son muestreados entre dos intervalos de muestreo contiguos. Idealmente la tasa de muestreo debería ser infinita, pero existen restricciones tecnológicas. Entonces, mientras menor sea la separación entre los S_i (T pequeño), mostrados en la Figura 25, más fiel será la señal digitalizada con respecto a la señal analógica original. Los factores de error que impiden que la señal digitalizada tenga diferencias respecto a la analógica se describen en el Apéndice 3.

6.2. Descripción

El ADC en el Microcontrolador M68HC08 es un conversor unipolar, de aproximaciones sucesivas disponible para una resolución de 8 bits o 8-/10-bits [6], garantizado para ambas opciones. Los Microcontroladores de Freescale M68HC08 incluyen módulos ADC con una gran variedad de opciones para el usuario, permitiendo de este modo diferentes aplicaciones [7].

Algunas de las características del módulo ADC incluyen:

- De 4 a 24 canales ADC con una gran variedad de opciones de por medio.
- Dos opciones para la resolución: están disponibles los módulos ADC de 8 bits o 8-/10-bits, dependiendo de la familia del Microcontrolador.
- El tipo de conversión es adaptable a cada aplicación: Permite *conversión de una muestra o continua*, con la opción de *conversión auto-scan* en algunos Microcontroladores.
- Soporta ambos métodos de programación: Incluye una *conversión completa por flag* y una *conversión completa por interrupciones*, permitiendo al usuario elegir un método por llamada selectiva o uno basado en interrupciones.
- Frecuencia seleccionable del reloj ADC: Incluye un preescalador de reloj, y algunos Microcontroladores incluyen la opción de elegir entre el reloj de bus o el cristal externo como reloj de entrada.
- Flexibilidad: El ADC 8-/10-bits contiene cuatro opciones para justificar el resultado de la conversión (10 bits justificados a izquierda, 10 bits justificados a derecha, 10 bits justificados a izquierda con signo, y el modo de truncado de 8 bits).

Los principales registros del módulo ADC son:

1. El registro de control y estado ADC (ADSCR)
 - Flags (banderas) para una conversión ADC completa (el bit 7, COCO)
 - habilitación de interrupciones ADC (el bit 6, AIEN)
 - Selección de conversiones continuas ADC o una muestra ADC (el bit 5, ADCO)
 - Selección de uno de los canales ADC para ser escaneado (bit 5:0, CH4:CH0)
2. El registro del reloj ADC (ADCLK o ADICLK)
 - Selección del preescalador del reloj (bit 7:5, ADIV2:ADIV0)
 - Selección del reloj de entrada para el reloj ADC interno (el bit 4, ADICLK)¹
 - Selección de cuatro modos de justificación de resultado (bit 3:2, MODE1:MODE0)¹
3. El registro de datos ADC (ADR)
 - Se actualiza con el resultado cada vez que un ADC termina la conversión, y puede ser ya sea:
 - a. De un byte de largo para ADCs de 8 bits.
 - b. De dos bytes de largo para ADCs 8-/10-bits conteniendo los datos en el formato seleccionado (ADRH-ADRL o ADRH0:El ADRL0)¹
4. Los registros de datos para el modo auto scan del ADC (ADRL1-ADRL3), que son registros de un byte de largo para canales ADC1 a ADC3 cuando el ADC opera en modo auto scan.¹
5. El registro de control auto scan del ADC (ADASCR)
 - Define el número de canales de auto scan a ser usado en el modo auto scan (bit 2:1, AUTO1: AUTO0)¹
 - Activa el modo auto scan (bit 0, ASCAN)¹

¹ Característica opcional

6.3. Descripción del Módulo ADC en el MCU HC908

Existen cuatro canales que pueden ser utilizados como entrada del conversor a través de los pines PTA0, PTA1, PTA4, y PTA5.

Un multiplexor analógico permite seleccionar uno de los cuatro canales como entrada del conversor (ADCVIN). ADCVIN es convertido por el método de aproximaciones sucesivas.

La resolución del conversor es de 8 bits. Cuando se realiza la conversión, el resultado es escrito en el registro de datos ADR y setea un flag o genera una petición de interrupción.

Los pines de E/S de propósitos generales PTA0, PTA1, PTA4, y PTA5 son compartidos con el conversor como canales de entrada de la señal a medir. Seleccionando uno de los canales como entrada del conversor a través del registro ADSCR, la función del pin como pin de E/S deja de tener efecto. El resto de los pines son controlados por la lógica del puerto de E/S y podrán ser utilizados como pines de E/S de propósito general.

Escribir en el registro de datos del puerto (PTA) o en el registro de dirección de datos (DDRA) no tendrá efectos sobre el pin seleccionado como entrada del ADC. Leer desde el pin del puerto cuando este es usado como entrada del ADC retornará un 0 si el correspondiente bit en el DDRA contiene un 0. Si el bit del DDRA contiene un 1, se lee el valor del registro de datos del puerto.

6.4. Voltaje de conversión

La entrada del conversor debe estar entre Vdd y Vss para que la conversión sea lineal. Si la entrada supera Vdd, el resultado de la conversión siempre será \$FF y si disminuye por debajo de Vss el resultado será \$00. La señal de entrada debe mantenerse estable durante el tiempo de conversión.

6.5. Tiempo de conversión

El conversor requiere 16 (o 17, si no existe sincronismo entre el reloj del Bus y el del conversor) ciclos de reloj del conversor para completar una conversión. La conversión comienza en el primer flanco ascendente inmediatamente después de seleccionar el canal, escribiendo en el registro ADSCR.

$$\text{Tiempo de conversión} = \frac{16 \text{ ciclos de reloj ADC}}{\text{Frecuencia de reloj ADC}}$$

6.6. Configuración de registros [7]

6.6.1. Registro de estado y control del ADC (ADSCR)

Este registro es muy importante dentro de las características del conversor AD, ya que contienen el flag COCO que indica cuando se ha completado una conversión. También tiene un flag de interrupción y los bits que permiten seleccionar el canal del conversor a utilizar.

Address: \$003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COCO	AIEN	ADCO	CH4	CH3	CH2	CH1	CH0
Write:	R							
Reset:	0	0	0	1	1	1	1	1
	R	= Reserved						

Figura 26. Registro de estado y control del ADC (ADSCR)

COCO: Bit de conversión completa (solo lectura). Si el bit AIEN se encuentra seteado (habilitación de interrupción) el bit COCO siempre será 0. Si el bit AIEN es 0, el COCO se pondrá a 1 cuando se complete una conversión y 0 cuando no se haya completado. Para borrar esta bandera debe realizarse una lectura sobre el registro de datos ADR.

AIEN: Bit de habilitación de interrupción. Si este bit es seteado, se genera una interrupción cuando el ADC finaliza una conversión. Leyendo el registro ADR o escribiendo en el ADSCR el flag de pedido de interrupción será borrado. Si el bit se encuentra en 0, una conversión no generará interrupción.

ADCO: Bit de conversión continua. Seteado producirá una conversión continua del canal seleccionado (los datos son sobrescritos sin importar si fueron leídos o no). Borrado generará una sola conversión del canal.

CH[4:0]: Bits de selección de canales. La configuración de estos se resume en el siguiente cuadro.

Tabla 4. Selección de canal por multiplexado.

CH4	CH3	CH2	CH1	CH0	Canal ADC	Selección de entrada
0	0	0	0	0	ADC0	PTA0
0	0	0	0	1	ADC1	PTA1
0	0	0	1	0	ADC2	PTA4
0	0	0	1	1	ADC3	PTA5
0	0	1	0	0	-	No usado ²
↓	↓	↓	↓	↓	-	
1	1	0	1	0	-	
1	1	0	1	1	-	reservado
1	1	1	0	0	-	No usado
1	1	1	0	1	-	V_{DDA} ³
1	1	1	1	0	-	V_{SSA} ³
1	1	1	1	1	-	ADC desactivado

Para apagar el ADC se deben colocar todos unos en los bits CH[4:0]

² Si se selecciona un canal no usado, el resultado de la conversión ADC será desconocido.

³ Los niveles de voltaje suministrados desde el nodo de referencia interno, como se especifica en la tabla, se usan para verificar la operación del conversor ADC, ambas en test de producción y para aplicaciones del usuario.

6.6.2. Registro de datos del ADC

Address: \$003E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	Indeterminate after reset							

 = Unimplemented

Figura 27. Registro de Datos del ADC

Registro de datos (ADR)

Este registro es refrescado con el resultado de cada conversión. Se recomienda que la primera conversión generada inmediatamente después de la activación del conversor sea descartada.

6.6.3. Registro de configuración del reloj para el ADC

Address: \$003F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ADIV2	ADIV1	ADIV0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figura 28. Registro de configuración de clock para el ADC.

Registro de configuración del reloj (ADICLK)

Los bits ADIV2, ADIV1, y ADIV0 son utilizados como divisores de la frecuencia de bus [8]. La frecuencia de reloj del ADC se debe encontrar entre las frecuencias mínimas y máximas de la señal de reloj de entrada. La forma de configurar este registro se resume en la siguiente tabla.

Tabla 5. Razón de división de reloj del ADC.

ADIV2	ADIV1	ADIV0	Tasa de reloj del ADC
0	0	0	Bus clock /1
0	0	1	Bus clock /2
0	1	0	Bus clock /4
0	1	1	Bus clock /8
1	x	x	Bus clock /16

Apéndice.

1. Registros de Control, de Estado y de Datos

A continuación se describen todos los registros de Datos, de Estado y de Control del MCU HC908, En cada Registro se detalla el nombre, la dirección de memoria en que se encuentra, el estado por defecto y el nombre de cada uno de los bits que forma cada registro.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$0000	Port A Data Register (PTA)	Read:	R	AWUL	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
\$0001	Port B Data Register (PTB)	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
\$0002	Unimplemented									
\$0003	Unimplemented									
\$0004	Data Direction Register A (DDRA)	Read:	R	R	DDRA5	DDRA4	DDRA3	0	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
			= Unimplemented		R = Reserved		U = Unaffected			

Figura 29. Registros de Puertos I/O

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0006	Unimplemented								
\$000A	Unimplemented								
\$000B	Port A Input Pullup Enable Register (PTAPUE)	Read: OSC2EN Write: 0	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0	
		Reset: 0	0	0	0	0	0	0	0
\$000C	Port B Input Pullup Enable Register (PTBPUE)	Read: PTBPUE7 Write: PTBPUE6	PTBPUE5	PTBPUE4	PTBPUE3	PTBPUE2	PTBPUE1	PTBPUE0	
		Reset: 0	0	0	0	0	0	0	0
\$000D	Unimplemented								
\$0019	Unimplemented								
\$001A	Keyboard Status and Control Register (KBSCR)	Read: 0 Write: 0	0	0	KEYF	0	IMASKK	MODEK	
		Reset: 0	0	0	0	0	0	0	0
\$001B	Keyboard Interrupt Enable Register (KBIER)	Read: 0 Write: 0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Reset: 0	0	0	0	0	0	0	0
\$001C	Unimplemented								
\$001D	IRQ Status and Control Register (INTSCR)	Read: 0 Write: 0	0	0	IRQF	0	IMASK	MODE	
		Reset: 0	0	0	0	0	0	0	0
\$001E	Configuration Register 2 (CONFIG2) ⁽¹⁾	Read: IRQPUD Write: IRQEN	R	OSCOPT1	OSCOPT0	R	R	RSTEN	
		Reset: 0	0	0	0	0	0	0 ⁽²⁾	
		1. One-time writable register after each reset. 2. RSTEN reset to 0 by a power-on reset (POR) only.							
\$001F	Configuration Register 1 (CONFIG1) ⁽¹⁾	Read: COPRS Write: LVISTOP	LVIPTD	LVIWRD	LVIOR3	SSREC	STOP	COPD	
		Reset: 0	0	0	0 ⁽²⁾	0	0	0	
		1. One-time writable register after each reset. 2. LVIOR3 reset to 0 by a power-on reset (POR) only.							
\$0020	TIM Status and Control Register (TSC)	Read: TOF Write: 0	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Reset: 0	0	1	0	0	0	0	0
\$0021	TIM Counter Register High (TCNTH)	Read: Bit 15 Write: Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Reset: 0	0	0	0	0	0	0	0
		= Unimplemented R = Reserved U = Unaffected							

Figura 30. Registros de Módulos KBI, IRQ, Config y TIM

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0022	TIM Counter Register Low (TCNTL)	Read: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:							
		Reset:	0	0	0	0	0	0	0
\$0023	TIM Counter Module Register High (TMODH)	Read: Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:							
		Reset:	1	1	1	1	1	1	1
\$0024	TIM Counter Module Register Low (TMDL)	Read: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:							
		Reset:	1	1	1	1	1	1	1
\$0025	TIM Channel 0 Status and Control Register (TSC0)	Read: CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0						
		Reset:	0	0	0	0	0	0	0
\$0026	TIM Channel 0 Register High (TCH0H)	Read: Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:							
		Reset:	Indeterminate after reset						
\$0027	TIM Channel 0 Register Low (TCH0L)	Read: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:							
		Reset:	Indeterminate after reset						
\$0028	TIM Channel 1 Status and Control Register (TSC1)	Read: CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0						
		Reset:	0	0	0	0	0	0	0
\$0029	TIM Channel 1 Register High (TCH1H)	Read: Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:							
		Reset:	Indeterminate after reset						
\$002A	TIM Channel 1 Register Low (TCH1L)	Read: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:							
		Reset:	Indeterminate after reset						
\$002B ↓ \$0035	Unimplemented								
\$0036	Oscillator Status Register (OSCSTAT)	Read: R	R	R	R	R	R	ECGON	ECGST
		Write:							
		Reset:	0	0	0	0	0	0	0
\$0037	Unimplemented	Read:							
\$0038	Oscillator Trim Register (OSCTRIM)	Read: TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:							
		Reset:	1	0	0	0	0	0	0

 = Unimplemented
 R = Reserved
 U = Unaffected

Figura 31. Registros de Módulos TIM y Oscilador.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0039 ↓ \$003B	Unimplemented								
\$003C	ADC Status and Control Register (ADSCR)	Read: C000 Write: R	AIEN	ADCO	CH4	CH3	CH2	CH1	CH0
\$003D	Unimplemented								
\$003E	ADC Data Register (ADR)	Read: Bit 7 Write: Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
\$003F	ADC Input Clock Register (ADICLK)	Read: ADIV2 Write: ADIV1	ADIV0	0	0	0	0	0	0
\$FE00	Break Status Register (BSR)	Read: R Write: R	R	R	R	R	R	SBSW See note 1	R
		Reset: 0	0	0	0	0	0	0	0
		1. Writing a 0 clears SBSW.							
\$FE01	SIM Reset Status Register (SRSR)	Read: POR Write: 1	PIN	COP	ILOP	ILAD	MODRST	LVI	0
\$FE02	Break Auxiliary Register (BRKAR)	Read: 0 Write: 0	0	0	0	0	0	0	BDCOP
\$FE03	Break Flag Control Register (BFCR)	Read: BCFC Write: R	R	R	R	R	R	R	R
\$FE04	Interrupt Status Register 1 (INT1)	Read: 0 Write: R	IF5	IF4	IF3	0	IF1	0	0
\$FE05	Interrupt Status Register 2 (INT2)	Read: IF14 Write: R	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3)	Read: 0 Write: R	0	0	0	0	0	0	IF15
\$FE07	Reserved	Read: R Write: R	R	R	R	R	R	R	R

 = Unimplemented
 R = Reserved
 U = Unaffected

Figura 32. Registros de Módulos ADC, SIM y Estados de Interrupciones

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$FE08	FLASH Control Register (FLCR)	Read: 0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE09	Break Address High Register (BRKH)	Read: Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE0A	Break Address low Register (BRKL)	Read: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE0B	Break Status and Control Register (BRKSCR)	Read: BRKE	BRKA	0	0	0	0	0	0
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE0C	LVI Status Register (LVISR)	Read: LVIOUT	0	0	0	0	0	0	R
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE0D ↓ \$FE0F	Reserved for FLASH Test	R	R	R	R	R	R	R	R
\$FFBE	FLASH Block Protect Register (FLBPR)	Read: BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:							
		Reset:	Unaffected by reset						
\$FFBF	Reserved	R	R	R	R	R	R	R	R
\$FFC0	Internal Oscillator Trim Value (Optional)	Read: TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:							
		Reset:	Unaffected by reset						
\$FFC1	Reserved	R	R	R	R	R	R	R	R
\$FFFF	COP Control Register (COPCTL)	Read:	LOW BYTE OF RESET VECTOR						
		Write:	WRITING CLEARS COP COUNTER (ANY VALUE)						
		Reset:	Unaffected by reset						
			= Unimplemented			R = Reserved		U = Unaffected	

Figura 33. Registros de Módulos de Memoria, LVI, Oscilador Interno y COP

2. Resets e Interrupciones

Los Resets [11] y [13] inicializan al CPU a un estado conocido. Todos los resets son manejados por medio del Modulo de Integración del Sistema (S.I.M) [5].

Tipos de Resets:

- Power On
- COP
- Illegal Address
- External Pin
- LVI
- Illegal Opcode

Las Interrupciones [14] suspenden el procesamiento normal para que el CPU pueda realizar algunos servicios requeridos.

Fuentes de Interrupciones:

- IRQ external PIN
- SCI
- TIM
- CGM (PLL)
- IRQ / KBI
- SPI
- ADC
- SWI

2.1. Procesamiento de Excepciones

Los Resets e Interrupciones son excepciones del CPU (al proceso normal), o sea “rompen” con la secuencialidad normal de la ejecución de un programa (ver figura 34). Determinando que tipo de manejo se requiere, se llama al proceso de excepción. El procesado de la excepción es manejado a través de tareas discretas.

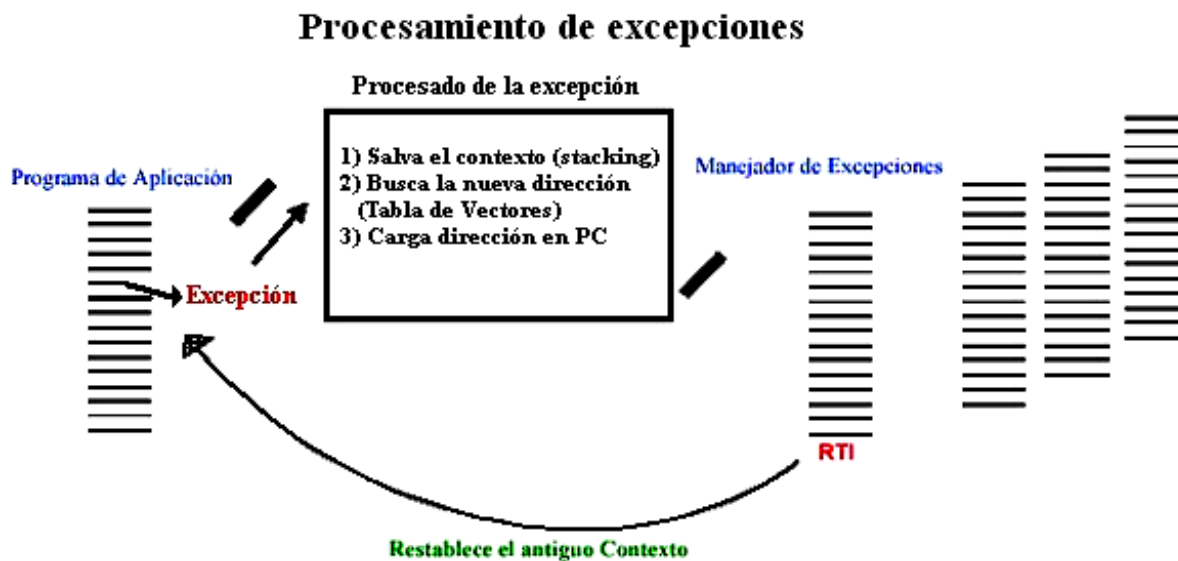


Figura 34. Procesamiento de excepciones.

Diferencias para el reconocimiento en el resets y las interrupciones

- Arbitraje
- Stacking (apilado)
- Búsqueda de vectores.

2.1.1. Reconocimiento

Detecta Reset o Interrupciones pendientes.

Resets:

- Es reconocida y ejecutada inmediatamente una vez ingresada (no hay sincronismo, es una excepción “imperativa”).

Interrupciones:

- Es reconocida durante el último ciclo de la instrucción corriente.
- En el caso que ingrese durante el último ciclo, será reconocida durante el último ciclo de la próxima instrucción.
- Actúa después del último ciclo de la instrucción corriente.
- Hay sincronismo.

2.1.2. Arbitraje

Resets:

Determinación de igual y más alta prioridad.

- No hay arbitraje

Interrupciones:

- Difiriendo prioridades, más bajas que los resets.
- Realizado por el SIM

2.1.3. Stacking (apilado)

Hace el salvado de la información del CPU cuando se atiende un requerimiento de interrupción, es por eso que se aconseja utilizar con cuidado la pila dentro de una interrupción.

Resets:

- No se realiza stacking (apilado), ya que no es necesario salvar registros.
- Se resetea el estado del CPU (a estados conocidos).

Interrupciones:

- Stacks (apila) registros del CPU (PC, X, A, CCR) como se muestra en la figura 35.
- El registro “H” no es apilado para mantener compatibilidad con líneas de Microcontroladores anteriores. Para apilarlo se deben usar las instrucciones PULL H y PUSH H.



Figura 35. Mapa de la pila en RAM (registros del CPU).

2.1.4. Búsqueda de Vectores

Resets:

Todos usan el mismo vector (igual dirección)

- Puede determinarse la fuente de reset examinando un registro del SIM llamado **SRSR** (Sim reset Status Register).

Interrupciones:

- El vector depende de la fuente de interrupción (Ver figura 36).

Busqueda de vectores

Tabla de Vectores del HC908GP32

\$FFDC - \$FFDD	Timebase
\$FFDE - \$FFDF	ADC conv. complete
\$FFE0 - \$FE1	Keyboard pins
\$FFE2 - \$FFE3	SCI Tx Empty /complete
\$FFE4 - \$FFE5	SCI Rx full/idle
\$FFE6 - \$FFE7	SCI Errors
\$FFE8 - \$FFE9	SPI Tx Empty
\$FFEA - \$FFEB	SPI Rx full/overflow
\$FFEC - \$FFED	TIM2 Overflow
\$FFEE - \$FFEF	TIM2 Channel 1
\$FFF0 - \$FFF1	TIM2 Channel 0
\$FFF2 - \$FFF3	TIM1 overflow
\$FFF4 - \$FFF5	TIM1 Channel 1
\$FFF6 - \$FFF7	TIM1 Channel 0
\$FFF8 - \$FFF9	CGM (PLL)
\$FFFA - \$FFFB	IRQ pin
\$FFFC - \$FFFD	SWI instruction
\$FFFE - \$FFFF	RESET

Más baja prioridad

Más alta prioridad

Figura36. Tabla de vectores

2.2. Enmascaramiento (Masking)

Habilita / Deshabilita el procesamiento de excepciones.

Resets:

- NO puede ser enmascarado (No puede impedirse.)

Interrupciones:

- Pueden ser enmascaradas (pueden impedirse).
- El bit I habilita/deshabilita TODAS las interrupciones a procesar.
- Mascaras Locales en periféricos que permiten enmascarar individualmente interrupciones.
- Disparo de Interrupciones Externas por Flanco o por Flanco y Nivel.

3. Factores de Error en la Conversión AD

A continuación se describen los errores inherentes a la conversión analógica a digital [7] y [8].

3.1. Error de muestreo (Sampling Error)

Este tipo de error está asociado al modelo de entrada del conversor A/D y tiene que ver con la impedancia y capacitancia características de entrada versus la impedancia característica de la señal a convertir. La Figura 37 muestra un modelo típico de impedancia de entrada de un conversor A/D.

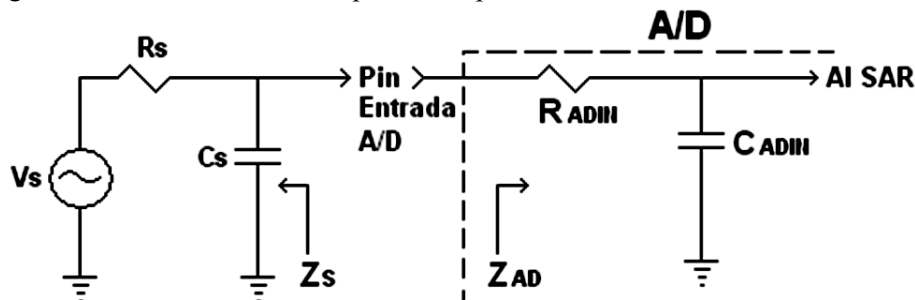


Figura 37. Modelo simplificado de entrada de un A/D.

Ejemplo: Para un conversor A/D cuya impedancia típica es de $R_{ADIN} = 7K$ y $C_{ADIN} = 5.5pF$, en donde muestras cercanas a $\frac{1}{4} LSb^4$ (*Less Significant bit*) con una resolución de 12 bits, pueden ser realizadas con la mayor velocidad de conversión de 3.5 ciclos de un reloj de 8MHz.

Lo anterior es posible siempre y cuando la impedancia característica (Z_s) de la señal aplicada al pin A/D, permanezca cercana a un valor de 2K, que sería el equivalente a la impedancia característica del A/D (Z_{AD}).

Para impedancias más altas de la fuente a digitalizar, se recomienda disminuir la velocidad de conversión (incremento en el tiempo de muestreo).

Por último, es importante que el usuario deberá siempre conocer las limitaciones de velocidad del conversor y aplicar el teorema de Nyquist-Shannon a la hora de digitalizar.

3.2. Error por goteo del pin (Pin Leakage Error)

Es posible que se presente una corriente de goteo debida a una alta resistencia de la fuente análoga a digitalizar.

El usuario deberá garantizar que R_s siempre esté por debajo de:

$$V_{DDAD} / (2^N \times I_{LEAK})$$

En donde:

V_{DDAD} : Voltaje de alimentación del A/D

N: Resolución de la conversión A/D

I_{LEAK} : Corriente de goteo (dato del manual)

Esta condición garantiza que el error por goteo estará por debajo de de $\frac{1}{4} LSb$.

⁴ Significa que para una resolución de 12 bits este error tendrá un peso del 0.024%.

3.3. Error por ruido inducido (Noise Induced Error)

El ruido en una conversión analógica a digital ocurre durante el muestreo (sample) o el proceso de conversión. El ruido inducido es inevitable, pero puede ser minimizado y para esto se recomiendan las siguientes acciones:

- Ubicar un condensador de 0.1uF, con baja ESR⁵ entre VREFH y VREFL.
- Ubicar un condensador de 0.1uF, con baja ESR entre VDDAD y VSSAD.
- Si la fuente VDD usa un aislamiento inductivo, es necesario ubicar un condensador adicional de 1uF entre VDDAD y VSSAD.
- Si VREFL se encuentra al nivel de VSS, es necesario garantizar una buena unión eléctrica.
- No hacer conmutaciones de pines I/O durante una conversión.

Existen situaciones en donde el ruido radiado o conducido (EMI) es intenso, para esto se hacen las siguientes recomendaciones:

- Ubicar un condensador de 0.01uF, con baja ESR a la entrada del canal de conversión, pero se debe tener en cuenta que esto altera la velocidad de muestreo.
- Hacer algoritmos que promedien las muestras, para suavizar los cambios ruidosos.
- Utilizar un reloj asíncrono como fuente del ADC y promediar las muestras.

3.4. Error de cuantización (Quantization Error)

Para una resolución de 12 bits, un paso de conversión (code) corresponde a una 1/4096 parte en la ventana de conversión. Este paso es el delta entre un código y el vecino inmediato, entonces la relación que define este paso será:

$$LSb = (VREFH - VREFL) / 2^N$$

En esta relación se encuentra un error inherente de cuantización, debido al resultado de la digitalización. Por ejemplo para un ADC con un error de $\pm \frac{1}{2} LSb$ en una resolución de 10 bits y con una ventana de conversión de 5V, el error asociado a un paso de conversión será de:

$$1 LSb = 5V/1024$$

$$1 LSb = 4.88mV$$

De donde:

$$\text{Error de cuantización} = \pm 2.44mV$$

3.5. Error de escala cero (Zero Scale Error =Offset Error)

Está definido como la diferencia entre el ancho del paso actual de conversión y el paso de conversión teórico, esto para el primer paso de conversión. La Figura 38 ilustra el fenómeno de error de escala cero.

⁵ El término corresponde a la resistencia serie equivalente.

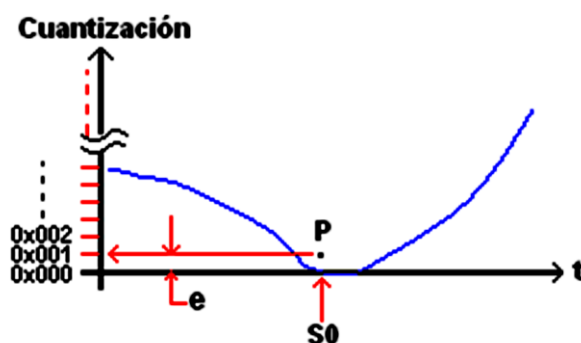


Figura 38. Error por escala de cero.

La muestra tomada en el punto **P** debería arrojar un valor de cuantización de 0x000, para la muestra tomada en **S0** (valor teórico). Como se puede apreciar el resultado real fue de 0x001, es decir un error de cero (**e**) de 1LSb.

3.6. Error de escala plena (Full Scale Error)

Está definido como la diferencia entre el ancho del paso actual de conversión y el paso de conversión teórico, esto para el último paso de conversión. La Figura 39 ilustra el fenómeno de error de escala plena.

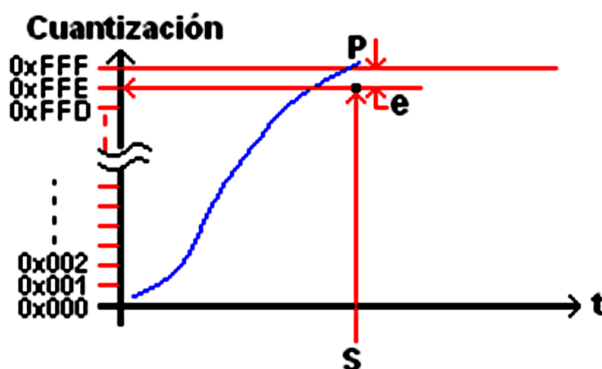


Figura 39. Error por escala plena.

La muestra tomada en el punto **P** debería arrojar un valor de cuantización de 0xFFF, para la muestra tomada en **S** (valor teórico). Como se puede apreciar el resultado real fue de 0xFFE, es decir un error de escala plena (**e**) de 1LSb.

3.7. Error diferencial no lineal (Differential non Linearity Error)

Está definido como la peor diferencia entre el valor teórico y el valor actual, de un paso de conversión. Esto aplica para toda conversión en todo el rango de la escala de conversión.

3.8. Error integral no lineal (Integral non Linearity Error)

Está definido como el más alto valor (valor absoluto) de la suma de los errores diferenciales acumulados.

3.9. Error por parpadeo en la conversión (Code Jitter Error)

Se presenta cuando en ciertos puntos, un voltaje de entrada determinado es convertido de uno a dos valores, durante un muestreo repetitivo.

Idealmente, cuando un voltaje de entrada es infinitesimalmente pequeño respecto de un voltaje de transición (paso mínimo de conversión), el convertidor arrojará el código más pequeño.

Es usual que pequeñas cantidades de ruido en el sistema, puedan causar que el convertidor tme un valor indeterminado (valor equivocado entre dos códigos consecutivos). El error típico de *jitter* es de $\pm \frac{1}{2}$ LSb. Para minimizar el error por *jitter* se recomienda hacer un promedio de las medidas.

3.10. Error por no monotonicidad (Non Monotonicity Error)

Se presenta cuando el conversor arroja un código bajo ante un voltaje alto de entrada.

Referencias:

- [1] Web Site "Freescale Semiconductors", www.freescale.com, 2007.
- [2] Douglas H. Summerville, "Embedded Systems Interfacing for Engineers using the Freescale HCS08 Microcontroller II: Digital and Analog Hardware Interfacing", State University of New York at Binghamton, Morgan y Claypool Publishers, 2009.
- [3] Data sheet: MC68HC908QY4 Microcontrollers, Rev. 5, 07/2005
- [4] Daniel Di Lella, Curso de Microcontroladores HC908, Electrocomponentes S.A., 2005
- [5] Reference Manual: CPU Central Processor Unit Microprocesador, CPU08RM, Rev. 4, 02/2006.
- [6] J. Feddeler, Lucas Bill, "ADC Definitions and Specifications", División de sistemas de Ingeniería, Austin, Texas, febrero de 2003.
- [7] Referente Manual: ADC - Analog-to-Digital Converter, ADCRM, Motorota, 1996.
- [8] Luis Reynoso Covarrubias, "Programming the Analog-to-Digital Converter on M68HC08 Microcontrollers", Freescale semiconductor, Nota de aplicación, Julio de 2005.
- [9] H. Taub: "Circuitos Digitales y Microprocesadores", Mc Graw-Hill, 1989.
- [10] Fernando I. Szklanny, Hoarcio Martinez Del Pezzo: "Introducción a los Microprocesadores", Editorial Arbó, 1979.
- [11] A. C. Dowton: "Computadores y Microprocesadores: Componentes y Sistemas", Versión en español de Ernesto morales Peake, Addison-Wesley, 1993.
- [12] James W. Bignell y Robert L. Donovan, "Electrónica Digital". México: Editorial Continental, 2005.
- [13] Ross Mitchell, "Resetting MCUs", Nota de aplicación, Motorola, East Kilbride, Scotland, 1993.
- [14] Mark Martinets, "Interrupt Handling Considerations When Modifying EEPROM on HC08 Microcontrollers", Nota de aplicación, Motorola, agosto de 2002.