

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

Tema 3 virtual para la asignatura  
MICROCOMPUTADORAS  
Profesor: Antonio Salvá Calleja  
Mayo de 2020

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

Todos los microcontroladores cuentan con la capacidad de poder ser configurados para que se generen interrupciones que se dan a un cierto intervalo de tiempo, esto es, el MCU entraría a la rutina de servicio periódicamente, siendo el intervalo entre entradas sucesivas a ésta programable. Esta funcionalidad es de gran utilidad en Instrumentación y Control; por ejemplo cuando se muestrea una señal. En este caso la rutina de servicio simplemente leería la salida del canal del convertidor analógico digital, a cuya entrada estuviera conectada la señal que se muestrea. El periodo entre interrupciones, obviamente sería el periodo de muestreo que se requiera en un momento dado.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

El hardware básico para la funcionalidad aquí esbozada, consiste en un contador binario, frecuentemente de 16 bits, para el cual:

1. Se puede configurar cual es la cuenta tope a la que puede llegar, para pasar a cero de nuevo, a esto se le llama **evento de overflow**.
2. Se conoce la frecuencia de la señal cuadrada TTL que acciona el contador.
3. Existe un bit testigo que se pone en uno lógico cuando se da el overflow. Ésto sería la bandera testigo de una instancia de interrupción, que se daría cada vez que hay un overflow del contador.

Por ejemplo, si se sabe que la cuenta tope del contador es 999, y que la frecuencia de la señal a la entrada del contador es 1 MHz, el intervalo de tiempo entre overflows sería de 1 mS. Ya que las cuentas del contador serían: 0,1,2,3,.....999,999; o sea, se darían mil periodos de 1 uS de la señal cuadrada de entrada al contador. **Nótese que la cuenta tope es el número de cuentas asociado menos 1.**

Para los microcontroladores, el contador aquí mencionado es parte de un periférico o módulo funcional denominado **Temporizador**. Un determinado MCU puede tener uno o varios temporizadores, cada uno con su respectivo contador.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

En la práctica, frecuentemente lo que se requiere es determinar cual es la cuenta tope (CT) requerida, para que el periodo entre overflows (Tovf), tenga un valor deseado, esto para un determinado valor de la frecuencia (Fck) de la señal que acciona el contador.

De acuerdo con lo explicado en la lámina anterior el número de cuentas requerido (NC) y la cuenta tope requerida estarían dados por:

$$NC = \frac{T_{ovf}}{T_{ck}} \dots\dots (1)$$

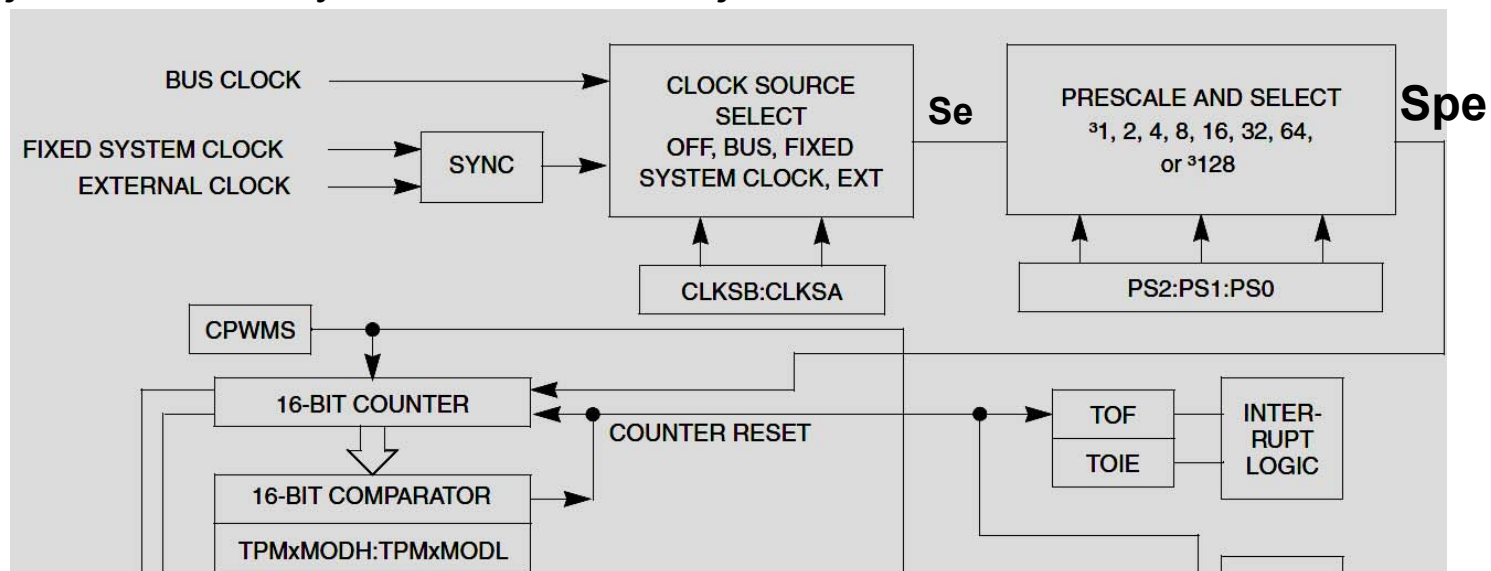
$$CT = NC - 1 \dots\dots (2)$$

Donde  $T_{ck} = 1/F_{ck}$  es el periodo de la señal aplicada al contador. Si el valor de NC obtenido no es entero, se debe considerar para éste al entero más próximo, esto haría que se produzca un error en el valor de Tovf respecto al valor requerido. En muchos casos prácticos este error suele ser pequeño y su efecto no afecta la funcionalidad de la aplicación implicada.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

El MCU mc9s08sh32 cuenta con dos temporizadores que el fabricante denota como TPM1 y TPM2. En la figura mostrada, tomada de la página 241 del documento mc9s08sh32.pdf, se muestra un esquema genérico de éstos, donde se aprecia únicamente lo relacionado con la generación de interrupciones periódicas. Ahí se aprecia que la señal de reloj, denotada como '**Spe**', que entra al contador, tiene una frecuencia que es el resultado de dividir entre uno de ocho posibles divisores, la frecuencia de otra señal, aquí denotada como '**Se**', la cual puede escogerse entre una de tres señales que son: el reloj de bus, una señal que el fabricante denomina reloj fijo del sistema y una señal de reloj externa.



# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Selección de preescalamiento

El divisor asociado con el preescalamiento es determinado con tres bits, denotados PS2, PS1 y PS0. En la tabla aquí mostrada, tomada de la página 247 del documento mc9s08sh32.pdf, se aprecia el valor que tiene el divisor de la frecuencia, que aquí denotamos como 'pe', para cada una de las ocho posibilidades asociadas con los tres bits aquí mencionados.

PS2:PS1:PS0	TPM Clock Source Divided-by (pe)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Selección de preescalamiento

Los tres bits PS2, PS1 y PS0, son parte de un registro del MCU denominado genéricamente como TPMxSC.

En la figura mostrada, tomada de la página 246 del documento mc9s08sh32.pdf, se muestra este registro con sus bits componentes. Se aprecia que los tres bits definitorios del preescalamiento son los tres bits menos significativos.

Al reset estos tres bits se inicializan como 000; por lo tanto, de acuerdo con la tabla mostrada en la lámina anterior, el valor por default para el preescalamiento es uno.

	7	6	5	4	3	2	1	0
R	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
W	0							
Reset	0	0	0	0	0	0	0	0

**Figure 16-7. TPM Status and Control Register (TPMxSC)**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Selección de señal de entrada 'Se'

La selección de la señal de entrada 'Se' está gobernada por los bits CLKSB y CLKSA. En la tabla mostrada, tomada de la página 247 del documento mc9s08sh32.pdf, se aprecia que valores deben tener estos bits, de acuerdo con la señal que se desee sea 'Se', en un momento dado.

Al reset, estos dos bits se inicializan como 00; y el contador del temporizador mostrará 0x0000 como cuenta, sin modificarse.

En los ejemplos ilustrativos de esta presentación, se usará como señal 'Se' al reloj de bus; por lo tanto, los bits CLKSB y CLKSA siempre se pondrán como 01.

CLKSB:CLKSA	TPM Clock Source to Prescaler Input (Se)
00	No clock selected (TPM counter disable)
<b>01</b>	<b>Bus rate clock</b>
10	Fixed system clock
11	External source



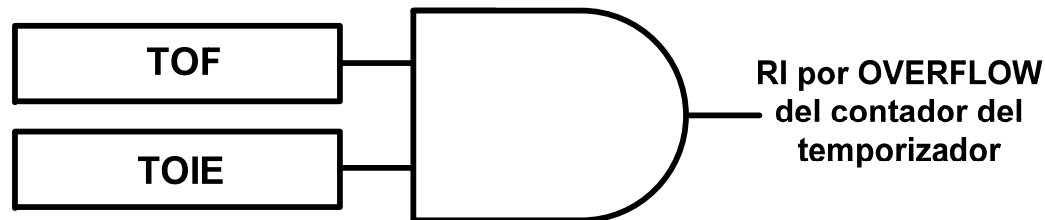
# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Habilitación de interrupciones cada que se da un overflow

Cada que se da un overflow se puede generar un requerimiento de interrupción, esto hará que el MCU entre periódicamente a la rutina de servicio asociada; esto es, el accionamiento asociado con el código colocado en ésta, se ejecutará repetitivamente, con una separación entre accionamientos igual al periodo de tiempo entre overflows, que en esta presentación se denota como **Tovf**.

En la figura se muestra la compuerta lógica, interna al MCU, cuya salida en uno lógico testifica el requerimiento de interrupción por overflow. Se aprecia que la bandera testigo asociada se denomina **TOF**, y que el habilitador local asociado se denomina como **TOIE**, ambos son respectivamente los bits 7 y 6 del registro TPMxSC. Véanse las láminas 10 a 13 de la presentación **Tema1\_concepto\_int.pdf**.

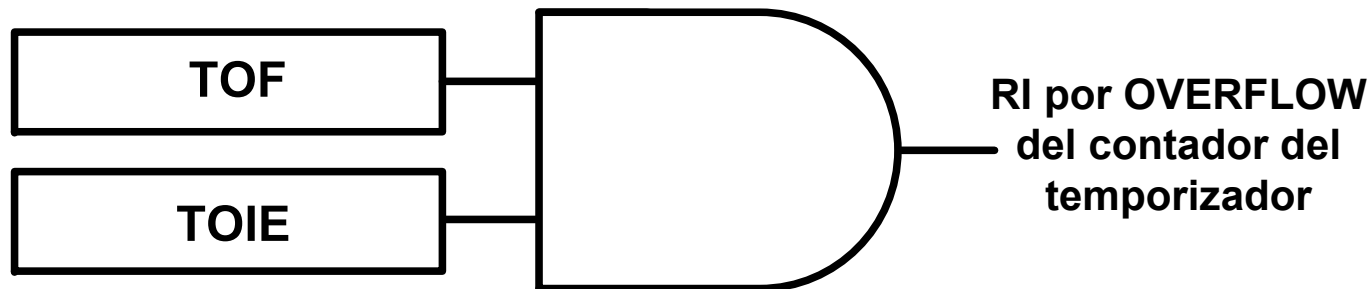


# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Habilitación de interrupciones cada que se da un overflow

De acuerdo con lo expresado en la lámina anterior, si un determinado programa va a usar interrupciones periódicas, en el bloque 1 de éste, deberá haber código que haga que el habilitador local TOIE tome el valor de uno lógico, además, en algún punto se debe hacer que la máscara global de interrupciones 'I' sea cero lógico, mediante la instrucción **CLI**. Para una mejor comprensión de lo expuesto en esta lámina, véanse las láminas 6 a 9 y 21 a 25 de la presentación **Tema1\_concepto\_int.pdf**.



# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Cálculo y colocación de la cuenta tope (CT) asociada con un determinado $T_{ovf}$

Asumiendo que la señal de entrada 'Se' al preescalador es el reloj de bus,  $T_{ck} = T_b$ , véase la figura presente en la lámina 7, y que se usa un determinado preescalamiento 'pe'. Considerando las ecuaciones (1) y (2) mostradas en la lámina 4, la cuenta tope 'CT' requerida para un determinado valor del tiempo entre overflows ' $T_{ovf}$ ', se determina mediante la siguiente ecuación:

$$CT = \text{Int}\left\{\frac{T_{ovf}}{peT_b}\right\} - 1 \dots \dots (3)$$

Donde  $T_b$  es el periodo asociado con la frecuencia del reloj de bus, además, se considera el hecho de que CT debe ser un entero.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Cálculo y colocación de la cuenta tope (CT) asociada con un determinado $T_{ovf}$

El valor de la cuenta tope obtenida mediante la ecuación (3), debe colocarse en un par de registros del temporizador denominados genéricamente como **TPMxMODH** y **TPMxMODL**, los cuales deberán contener respectivamente los bytes alto y bajo del valor de la CT requerida para un determinado  $T_{ovf}$ .

**Nótese que el valor de la cuenta tope CT deberá caber en 16 bits.**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Ejemplo 1: cálculo del valor de CT para un determinado Tovf

Supóngase que se desea que el tiempo entre overflows Tovf para el contador del temporizador sea 50 mS, si el reloj de bus tiene una frecuencia  $F_b = 20 \text{ MHz}$  ( $T_b = 50 \text{ nS}$ ), determinar los posibles pares de valores para la cuenta tope CT y el preescalamiento 'pe'.

Primero calculamos con  $pe = 1$ , usando la ecuación (3) considerando los valores explícitos implicados:

$$CT = Int\left\{\frac{50 \times 10^{-3}}{(50 \times 10^{-9}) \times (1)}\right\} - 1 = 999999$$

Dado que el valor obtenido no cabe en 16 bits, es claro que no se puede usar un preescalamiento unitario para los valores de Tovf y Tb implicados.

Por lo tanto, se debe aumentar el valor de 'pe'. Puede verse que 16, es el primer valor hacia arriba de 'pe', que conduce a un valor de CT que cabe en 16 bits, este valor se obtiene aplicando la ecuación (3) con  $pe = 16$  y el detalle de su cálculo se muestra en la siguiente lámina

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

### Ejemplo 1: cálculo del valor de CT para un determinado Tovf

Con  $pe = 16$  se tiene:

$$CT = Int\left\{\frac{50 \times 10^{-3}}{(50 \times 10^{-9}) \times (16)}\right\} - 1 = 62499 = 0xF423$$

Puede verse que si usamos  $pe = 32$  ó  $64$  ó  $128$ , los valores de CT obtenidos cabrían en 16 bits, y serían respectivamente: 31249, 15624 y 7811.

**De hecho, cualquiera de los 4 pares de valores para CT y 'pe' obtenidos, podrían usarse para que el tiempo entre overflows sea 50 mS, cuando**

**Fbus = 20 MHz**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## **TEMPORIZADORES DEL MCU MC9S08SH32**

### **Direcciones de los registros asociados con la generación de interrupciones periódicas con los temporizadores del MCU mc9s08sh32**

Los tres registros asociados con la generación de interrupciones periódicas con un temporizador del MCU mc9s08sh32, genéricamente el fabricante los denota como: TPMxSC, TPMxMODH y TPMxMODL. La literal 'x' denota el número de temporizador de que se trate en un momento dado. Dado que el MCU mc9s08sh32 tiene dos temporizadores, la literal 'x' será 1 para el temporizador uno, y 2 para el temporizador dos.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

**Direcciones de los registros asociados con la generación de interrupciones periódicas con los temporizadores del MCU mc9s08sh32**

Así, del mapa de memoria del MCU aquí mencionado, las direcciones y denotación de los dos tríos de bytes asociados con la funcionalidad de overflow, para cada uno de los dos temporizadores, presentes en el MCU son:

Temporizador	Registro	Dir (hex)
1	TPM1SC	0x20
1	TPM1MODH	0x23
1	TPM1MODL	0x24
2	TPM2SC	0x60
2	TPM2MODH	0x63
2	TPM2MODL	0x64



# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

**Inicialización de los registros del temporizador 1 para que éste genere tiempos entre overflows  $T_{ovf}$  determinados**

Pasos a seguir:

1. Emplendo la ecuación (3), determinar un par de valores viables para el preescalamiento 'pe' y la cuenta tope CT, considerando el valor del tiempo entre overflows  $T_{ovf}$  deseado y el periodo de bus  $T_b$  del reloj del MCU.
2. A partir del valor del preescalamiento 'pe' determinado en el paso anterior, usando la tabla 16-5 de la página 247 del manual del MCU, determinar el valor npe que representan los 3 dígitos binarios PS2, PS1 y PS0. Por ejemplo, si  $pe = 16$ ,  $npe = 4$ , si  $pe = 8$ ,  $npe = 3$ . Nótese que npe es el exponente al que hay que elevar el número dos para obtener el valor de pe.
3. Obtener el valor 'valsc' del byte que se debe cargar en el registro TPM1SC empleando la ecuación:

•

$$valsc = 64h + 8 + npe \cdots \cdots (4)$$

Donde h debe ser uno, si se desea que cada que se de un overflow, se genere un requerimiento de interrupción; en otro caso h debe ser cero.

**Nota:** Puede verse, aplicando un poco de aritmética binaria elemental, que si  $h = 1$ , el valor del bit 6 del byte que se va a cargar en el registro TPM1SC es uno lógico, por lo tanto, TOIE se pondrá en uno lógico, habilitándose localmente las interrupciones por overflow; por otra parte, si  $h = 0$ , TOIE quedará en cero lógico, no estando habilitadas interrupciones por overflow.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

**Inicialización de los registros del temporizador 1 para que éste genere tiempos entre overflows  $T_{ovf}$  determinados**

Ejemplo 2: Se desea que el temporizador 1 genere interrupciones periódicas cada 100 mS, suponiendo que  $F_{bus} = 20$  MHz ( $T_b = 50$  nS), determinar:

- Un par de valores viable para la cuenta tope CT y el preescalamiento 'pe'.
- El valor valsc que debe cargarse en el registro TPM1SC.
- Esbozar un código en ensamblador que configure el temporizador 1 para fines de lo especificado en el enunciado de este ejemplo.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

**Inicialización de los registros del temporizador 1 para que éste genere tiempos entre overflows Tovf determinados**

### Ejemplo 2: Solución

Paso 1: Empleando la ecuación 3 puede verse que el mínimo valor de 'pe' para que CT tenga un valor que quepa en 16 bits es 32, siendo  $CT = 62499 = 0xF423$ .

Paso 2: Dado que  $pe = 32$ , a partir de la tabla 16- 2 de la página 247 del manual del MCU, se ve que  $npe = 5$ .

Paso 3: Empleando la ecuación 4, considerando que se han de generar interrupciones ( $h=1$ ), cada vez que se de un overflow, el valor valsc que ha de cargarse en el registro TPM1SC es  
 $valsc = 64(1) + 8 + npe = 77 = 0x4D = 01001101$ .

Nótese que el bit 6 que corresponde al habilitador local TOIE se pondrá en uno lógico, lo que habilita localmente la instancia de interrupción por overflow del temporizador 1.

Un posible tramo de código en ensamblador para los fines de este ejemplo es:

```
mov #$f4,tpm1modh
mov #$23,tpm1modl ;carga cuenta tope CT

mov #$4d,tpm1sc ;pe = 32, toie = 1, Se = reloj de bus
```

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

Ejemplo 3: Se desea que empleando el temporizador 1 se produzca la siguiente cadencia en el ledpum de la tarjeta FACIL\_08SH, (250 ms encendido/250 mS apagado), suponiendo que  $F_{bus} = 20 \text{ MHz}$  ( $T_b = 50 \text{ nS}$ ), escribir un programa en ensamblador que realice esto empleando para ello el que se generen interrupciones periódicas cada 250 mS y en la rutina de servicio asociada, simplemente se complementa el bit PTA7, que es a donde está conectado el **ledpum**, véase la figura 4 del documento Gb\_facil\_08sh.pdf.

Empleando la ecuación (3) puede verse que el único par viable ( CT, pe ) es:  $pe = 128$  y  $CT = 39061 = 0x9895$ .

Dado el valor de 'pe' requerido,  $n_{pe} = 7$ , y ya que se requiere que se de un requerimiento de interrupción cada que haya un overflow, debemos usar  $h=1$  en la ecuación (4), por lo tanto, el valor de valsc que se debe cargar en el registro TPM1SC es:

$$valsc = 64(1) + 8 + 7 = 79 = 0x4F$$

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

Ejemplo 3: Continuación

De lo obtenido en la lámina anterior, el bloque 1 del programa podría ser:

```
bset 7,ptadd ;ptad7 es salida
```

```
mov #$98,tpm1modh
```

```
mov #$95,tpm1modl ;carga cuenta tope CT
```

```
mov #$4f,tpm1sc ;pe = 128, toie = 1, Se = reloj de bus, Tovf = 250 mS
```

```
cli ;habilita globalmente interrupciones
```

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

Ejemplo 3: Continuación

Dado que en este ejemplo todo lo hará la rutina de servicio de interrupción, el bloque 2 del programa será simplemente un salto a la misma línea.

BLOQUE 2

**fin**

**bra fin**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

Ejemplo 3: Continuación

Estructura del BLOQUE 3

El bloque 3 contendrá únicamente la rutina de servicio de interrupción, el inicio de ésta se denota con la etiqueta 'servparp:'. El MCU pasará a ejecutarla cada 250 mS, el código en ésta debe:

- Regresar a cero la bandera testigo TOF. Cómo se hace esto, puede verse en la explicación alusiva al bit TOF, presente en la página 246 del manual del MCU.  
Ahí dice que para limpiar la bandera TOF se debe hacer una lectura del registro TPM1SC, seguido esto por una puesta a cero del bit 7 del mismo registro.
- Efectuar el accionamiento que se desea se haga cada 250 mS. Éste será simplemente complementar el bit pta7 del puerto A, sin afectar a los otros bits del mismo puerto. Esto se logra efectuando una operación or exclusivo con 1 lógico para el bit 7 del puerto y con cero lógico para los demás bits, empleando para ello a la instrucción EOR del MCU.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

Ejemplo 3: Continuación

Estructura del BLOQUE 3

La rutina de servicio podría ser:

```
servparp:      lda tpm1sc
                bclr 7,tpm1sc ; TOF ← 0

                lda ptad
                eor #$80
                sta ptad ;complementa el bit pta7

                rti ;Retorna de la interrupción
```



# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

Ejemplo 3: Continuación  
Estructura del BLOQUE 4

En la tabla 5-2 de la página 64 del manual del MCU, se ve que la instancia de interrupción por overflow para el temporizador 1 es la número 11. Ahí se aprecia que las direcciones naturales para la carga del vector de interrupción son FFE8 y FFE9, por lo tanto, para un dispositivo CHIPBAS8SH, las direcciones de carga del vector son D7E8 y D7E9. Entonces, el bloque 4 podría quedar como:

```
org $d7e8  
dw servparp
```

El programa fuente de este ejemplo está en el archivo **parp250ms.asm**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## TEMPORIZADORES DEL MCU MC9S08SH32

Ejemplo 4: Se desea emplear el temporizador 1 para efectuar cada cinco segundos un accionamiento, que consiste en complementar el bit pta7 del puerto A. Considérese que  $F_{bus} = 20 \text{ Mhz}$  ( $T_b = 50 \text{ nS}$ ). En este caso se requiere que  $T_{ovf}$  sea 5 segundos. Empleando la ecuación 3 se puede ver que aún cuando se use  $pe = 128$ , el valor de CT es 156249,

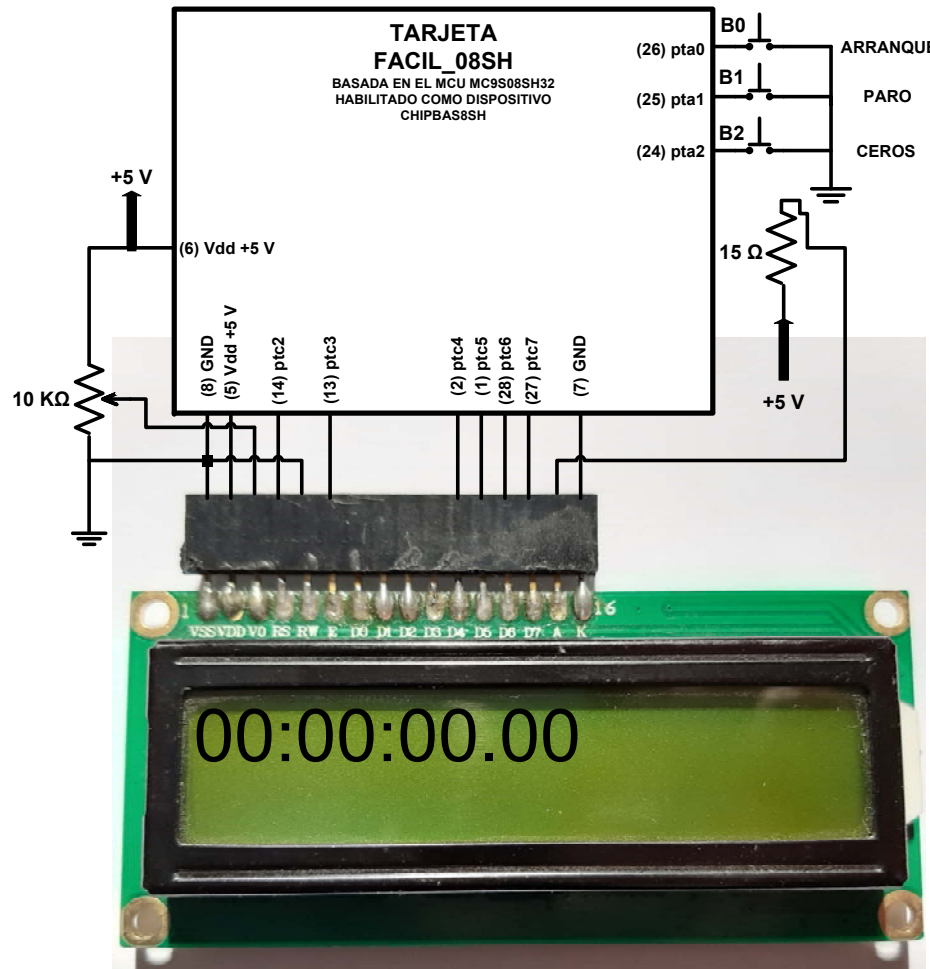
**¡que no cabe en 16 bits!**

Dado que no se puede lograr un par (CT,pe) para que  $T_{ovf}$  sea 5 segundos, para lograr el tiempo entre accionamientos deseado, se configura el temporizador para un intervalo entre overflows que sea un submultiplo exacto del tiempo entre accionamientos deseado. En este ejemplo se usó  $T_{ovf} = 250 \text{ mS}$ , y se usa un contador base 20 en RAM, que se incrementa cada que el MCU entra a la rutina de servicio asociada. Dado que esto sucede cada 250ms, al llegarse a la cuenta 20 habrán transcurrido cinco segundos, es entonces que se efectúa el accionamiento deseado y se regresa a cero el contador auxiliar, para que quede listo para el transcurso de otros cinco segundos.

El programa fuente para este ejemplo está en el archivo **parp5seg.asm**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

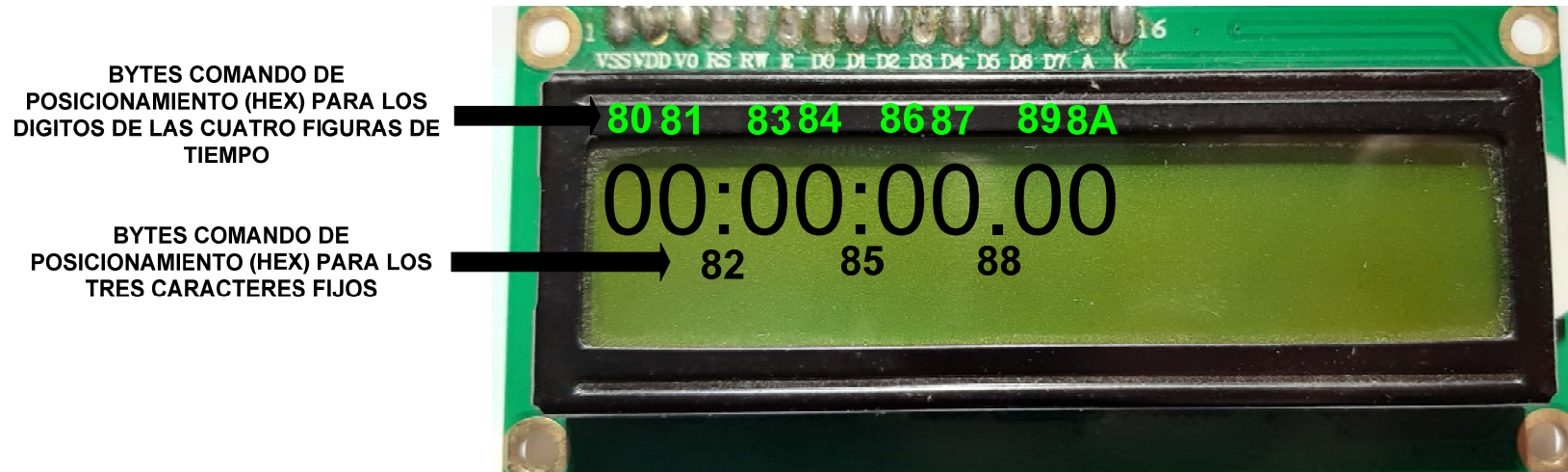
Ejemplo 5: Escribir un programa en ensamblador, que implique al temporizador 1 del MCU, de modo que al ejecutarse, se valide la funcionalidad del cronómetro mostrado en la figura. Ahí se aprecia que éste deberá contar con un botón de arranque, ligado con pta0; un botón de paro, ligado con pta1; y un botón de puesta a cero, ligado con pta2. La resolución deberá ser de horas, minutos, segundos y centésimas. El tiempo deberá correr de las 00:00:00.00 a las 23:59:59.99. Las cuatro figuras de tiempo se deben desplegar en un LCD de 16x2 a partir de la columna 1 del renglón 1. El programa fuente asociado deberá almacenarse en el archivo **cronobotb.asm**



# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5

Por ser necesario para fines del despliegado de los caracteres fijos, y de los asociados con las cuatro figuras de tiempo, aquí se muestran los bytes comando de posicionamiento asociados con éstos.



Nota: A partir de la ecuación (1) mostrada en la lámina 11 del archivo Tema2\_manejo\_de\_lcd.pdf, puede verse fácilmente que los bytes de posicionamiento para las columnas 1 a 16 del renglón 1 respectivamente son: 0x80, 0x81, 0x82, ..., 0x8E y 0x8F. En lo que toca al renglón 2, los bytes de posicionamiento para las columnas 1 a 16 respectivamente son: 0xC0, 0xC1, 0xC2, ..., 0xCE y 0xCF.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5: Código básico que deberá ejecutarse cada centésima de segundo

Las cuatro figuras de tiempo se asocian con sendos contadores en la página cero de la RAM del MCU, denominados como: **conthor** para las horas, que deberá contar de 0 a 24; **contmin** para los minutos, que deberá contar de 0 a 60; **contseg** para los segundos, que deberá contar de 0 a 60; y **cont100** para las centésimas que deberá contar de 0 a 100.

Dado que el mínimo intervalo de tiempo implicado es una centésima de segundo, o sea, 10 mS, se configura el temporizador 1 de modo que se de una interrupción por overflow cada 10 mS.

Las cuentas de los cuatro contadores deberán iniciarse a cero en el bloque 1 del programa y deberán actualizarse **cada centésima (10 mS)**, por lo tanto, la rutina de servicio de interrupción deberá:

1. Regresar a cero la bandera testigo TOF
2. Se incrementa el contador de centésimas (cont100), si éste no ha llegado 100, se pasa al paso 6, pero si éste llega a 100, esto es, transcurrió ya un segundo más, se regresa a cero cont100 y se pasa al paso 3
3. Se incrementa el contador de segundos (contseg), si éste no ha llegado 60, se pasa al paso 6, pero si éste llega a 60, esto es, transcurrió ya un minuto más, se regresa a cero contseg y se pasa al paso 4
4. Se incrementa el contador de minutos (contmin), si éste no ha llegado 60, se pasa al paso 6, pero si éste llega a 60, esto es, transcurrió ya una hora más, se regresa a cero contmin y se pasa al paso 5
5. Se incrementa el contador de horas (conthor), si éste no ha llegado 24, se pasa al paso 6, pero si éste llega a 24, esto es, transcurrió ya un día más, se regresa a cero conthor y se pasa al paso 6
6. Retorna de la interrupción

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5: Código de la rutina de servicio de interrupción

Aquí se muestra un posible código para validar la subrutina de servicio de interrupción, que es parte del bloque 3 del programa , y cuyo inicio se denota con la etiqueta 'sertof:'

\*\*\*\* Subrutina de servicio interrupción \*\*\*\*

```
sertof:  lda tpm1sc
        bclr 7,tpm1sc ;tof <-- 0
        inc cont100
        lda cont100
        cmp #$64 ;Si cont100 no ha llegado a 100 sale,
        bne salir ;si cont100 llega a 100,limpia cont100 y pasa a incrementar contseg
        clr cont100
        inc contseg
        lda contseg
        cmp #$3C ;Si contseg no ha llegado a 60 sale,
        bne salir ;si contseg llega a 60,limpia contseg y pasa a incrementar contmin
        clr contseg
        inc contmin
        lda contmin
        cmp #$3C ;Si contmin no ha llegado a 60 sale,
        bne salir ;si contmin llega a 60,limpia contmin y pasa a incrementar conthor
        clr contmin
        inc conthor
        lda conthor
        cmp #$18 ;Si conthor no ha llegado a 24 sale,
        bne salir ;si conthor llega a 24,limpia conthor
        clr conthor
salir:   rti
```

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5

Aplicando la ecuación (3), puede verse que, para que  $T_{ovf}$  sea 10 mS, un par viable ( $p_e, CT$ ) es  $p_e=8$ , lo que implica que  $n_{pe}$  sea 3, y  $CT = 24999 = 0x61a7$ .

Por otra parte, al inicializar el temporizador no se debe habilitar la interrupción, sólo se debe configurar lo propio, para que  $T_{ovf}$  sea 10 mS, ya que será la detección de la opresión del botón de arranque (B0), el evento que habilite localmente la interrupción haciendo **TOIE = 1**, y la opresión del botón paro (B1), el evento que deshabilite localmente la interrupción haciendo **TOIE = 0**. Por lo tanto, para fines de la inicialización del temporizador usado, el valor del byte **valsc** que se debe cargar en el registro **TPM1SC**, se obtiene empleando la ecuación (4) con  $n_{pe} = 3$  y  $h = 0$ ; o sea , **valsc = 11 = 0x0b**.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5

### Sentencias equ requeridas

El programa a desarrollar implica datos a usar como pueden ser entre otros: los códigos ASCII de los caracteres fijos, la cuenta tope, las localidades de RAM donde residirán los cuatro contadores asociados con las figuras de tiempo. Esta información se declara en las sentencias equ presentes en el encabezado del programa, éstas se muestran a continuación:

```
asciip  equ $2e ;Código ASCII del caracter '.'
ascii2p equ $3a ;Código ASCII del caracter ':'
conthor equ $90
contmin equ $91
contseg equ $92
cont100 equ $93
tpm1sc  equ $20
tpm1modh equ $23
ctopec  equ $61a7 ;Cuenta tope (CT)
pardig  equ $95
pardig2 equ $96
```

```
ptad  equ $00
ptadd equ $01
ptape equ $1840
```



# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5 Continuación

:

**\*\*\* Sentencias equ para definir bytes de posicionamiento requeridos \*\*\***

**bypos2p1 equ \$82 ;byte comando para posición de caracter ':' entre horas y minutos**

**bypos2p2 equ \$85 ;byte comando para posición de caracter ':' entre minutos y segundos**

**bypospto equ \$88 ;byte comando para posición de caracter '.' entre segundos y centésimas**

**byposhor equ \$80 ;byte comando de posicionamiento para las horas**

**byposmin equ \$83 ;byte comando de posicionamiento para los minutos**

**byposseg equ \$86 ;byte comando de posicionamiento para los segundos**

**byposcent equ \$89 ;byte comando de posicionamiento para las centésimas**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## **Ejemplo 5: Estructura del bloque 1 del programa asociado con la realización del cronómetro mostrado en la lámina 27**

Este bloque deberá efectuar las siguientes acciones de inicialización y configuración de recursos del MCU, empleados para fines de la realización del cronómetro

1. Habilitar resistencias de pull-up internas para los tres bits pta0, pta1 y pta2 que reciben el nivel lógico de los botones B0, B1 y B2
2. Inicializa el desplegado (LCD)
3. Colocar en las columnas del LCD que corresponda los caracteres fijos ':' y '.'
4. Configura temporizador 1 de modo que,  $T_{ovf} = 10 \text{ mS}$  sin habilitar interrupciones por overflow
5. Inicializa a cero las cuatro figuras de tiempo, horas, minutos, segundos y centésimas
6. Habilita globalmente las interrupciones

En las siguientes dos láminas se detalla el código que corresponde a las acciones de inicialización

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

**Ejemplo 5: Estructura del bloque 1 del programa asociado con la realización del cronómetro mostrado en la lámina 27**

**Código en el bloque 1, asociado con las acciones de inicialización 1, 2 y 3**

```
lda #$07 ;Pta2,pta1 y pta0  
sta ptape ;tienen 'pull-ups' internos.
```

```
jsr inilcd ;Inicializa LCD
```

**\*\* Coloca caracteres fijos ':' y '.' \*\*\*\*\***

```
lda #bypos2p1  
jsr escom4 ;Posiciona siguiente escritura en renglón 1 columna 3  
lda #ascii2p  
jsr escdat4 ;Coloca caracter ':' en renglón 1 columna 3
```

```
lda #bypos2p2  
jsr escom4 ;Posiciona siguiente escritura en renglón 1 columna 6  
lda #ascii2p  
jsr escdat4 ;Coloca caracter ':' en renglón 1 columna 6
```

```
lda #bypospto  
jsr escom4 ;Posiciona siguiente escritura en renglón 1 columna 9  
lda #asciip  
jsr escdat4 ;Coloca caracter '.' en renglón 1 columna 9
```

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

**Ejemplo 5: Estructura del bloque 1 del programa asociado con la realización del cronómetro mostrado en la lámina 27**

**Código en el bloque 1, asociado con las acciones de inicialización 4, 5 y 6**

**\*\* Configura temporizador 1 para que  $T_{ovf} = 10$  mS sin habilitar interrupciones por overflow \*\*\*\*\***

```
ldhx #modc
sthx tpm1modh
mov #$0b,tpm1sc ;toie--0,clksb:clksa <--01, pe=8,tovf=10 ms.
```

**\*\*\* Inicializa a cero los cuatro contadores en RAM,**

**\*\*\* asociados con las cuatro figuras de tiempo.**

**\*\*\* Horas está en conthor,minutos está en contmin**

**\*\*\* segundos está en contseg y centésimas está en cont100**

```
clr conthor
clr contmin
clr contseg
clr cont100
```

```
cli ; habilita interrupciones globalmente
```

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5

### **Estructura del bloque 2 del programa asociado con la realización del cronómetro mostrado en la lámina 27**

Las acciones que debe efectuarse mediante el código que integra este bloque son:

1. Checar si alguno de los tres botones está oprimido, si es el caso llevar a cabo la acción que corresponda acorde con el botón que se oprimió, después de esto, pasar al paso 2. Si ningún botón está oprimido, pasar directamente al paso 2.
2. Desplegar en el LCD las cuatro figuras de tiempo expresadas en decimal. Para esto se invoca una subrutina cuyo inicio se denota con la etiqueta 'desptemp:'.
3. Regresar al paso 1

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5

**Estructura del bloque 2 del programa asociado con la realización del cronómetro mostrado en la lámina 27**

Aquí se muestra el código que integra al bloque 2 del programa

**lazo: brclr 0,ptad,arranque**

**brclr 1,ptad,paro**

**brclr 2,ptad,ceros**

**desple: bsr desptemp ;Esta subrutina despliega las figuras de tiempo en el LCD. Se ejecuta en aprox 42.5 mS**

**bra lazo**

```
arranque: bset 6,tpm1sc ;toie <-- 1
```

**bra desple**

```
paro:  bclr 6,tpm1sc ;toie <-- 0
```

**bra desple**

**ceros: clr conthor**

## clr contmin

## clr contseg

```
clr cont100
```

**bra desple**

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5

Estructura del bloque 3 del programa asociado con la realización del cronómetro mostrado en la lámina 27

El bloque 3 del programa está integrado por los siguientes componentes:

- **Subrutina de servicio de interrupción por overflow**, ésta se mostró previamente en la lamina 30.
- **Subrutina desptemp**, que despliega las cuatro figuras de tiempo en el LCD. Se auxilia de una subrutina auxiliar denominada auxdesp.
- **Subrutina con\_a**, que convierte un byte menor que 100 a su representación en decimal, como un par de códigos ASCII. Esta subrutina se invoca cuatro veces desde la subrutina desptemp, para poder desplegar las cuatro figuras de tiempo en decimal en el LCD.
- **Subrutinas inilcd, escom4, escdat4 y copiadis**, contenidas en el archivo **rutsswlcsh32\_20mhz.asm**, que se incluye con una sentencia `$include`.

El detalle y funcionalidad de las subrutinas que integran el bloque 3 puede verse en el archivo **coronobotb.asm**, que contiene el programa que valida el cronómetro.

# INTERRUPCIONES PERIÓDICAS EN LOS MICROCONTROLADORES

## Ejemplo 5

**Estructura del bloque 4 del programa asociado con la realización del cronómetro mostrado en la lámina 27**

El bloque 4 del programa está integrado únicamente por la declaración que valida la colocación del vector asociado con la instancia 11 de interrupción del MCU, cuyo evento asociado es el overflow del temporizador 1. A partir de la tabla 5-2 de la página 64 del manual del MCU, y considerando la ubicación de los vectores de interrupción para un dispositivo CHIPBAS8SH, es fácil ver que la declaración de la colocación del vector de interrupción, es la que se muestra a continuación:

```
org $d7e8 ;Colocación del vector de interrupción  
dw sertof ;propio del evento de overflow del temporizador 1
```

- **El programa completo del ejemplo 5, está en el archivo cronobotb.asm**