

Diseño Basado en Microprocesadores

Tema 2. Microcontroladores

- 2.1. Introducción a los microcontroladores
- 2.2. Entradas/Salidas Digitales
- 2.3. Temporizadores
- 2.4. Excepciones
- 2.5. Conversión Analógica/Digital
- 2.6. Comunicación serie RS232C
- 2.7. Teclado, conversión D/A y sonido
- 2.8. Interfaz I2C

2.2. Temporizadores

2.2.1. Generalidades

2.2.2. Diagrama de los T/C

2.2.3. Posibilidades del registro Match

2.2.4. Entradas/Salidas asociadas a los timer

2.2.5. Mapa de memoria de los registros del timer

2.2.6. Ejemplos de operación del temporizador

2.2.7. Acceso a los registros

2.2.8. Proceso de configuración de los T/C

2.2.9. Ejemplo temporizador

2.2.10. Biblioteca "timer_lpc40xx.h"

2.2.11. Dependencia CPU-Timer

2.2.12. Dispositivos en paralelo. Comunicación por muestreo

2.2.13. Conteo de tiempo

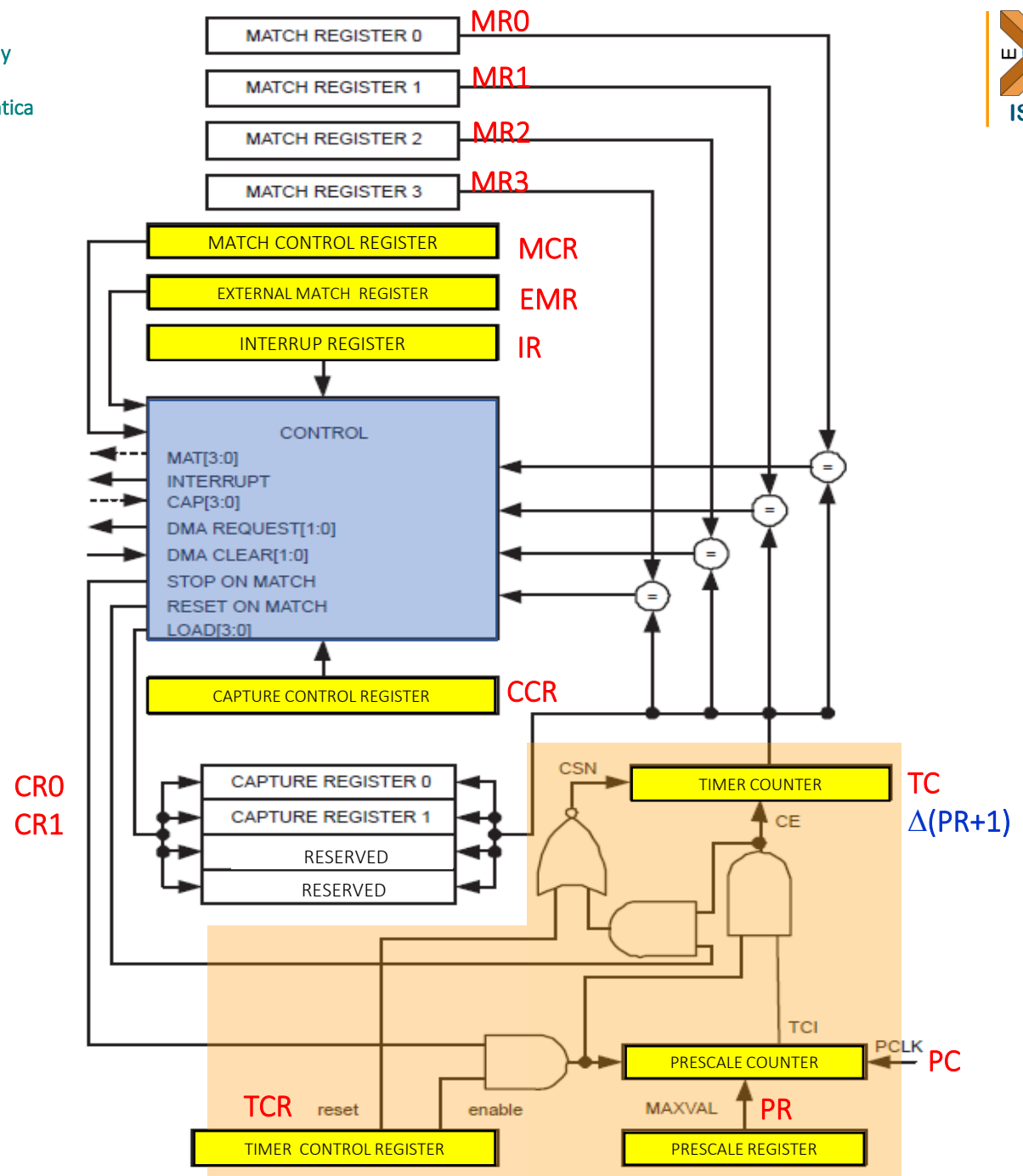
2.2.1. Generalidades

- 4 Temporizadores/Contadores T0 a T3 de 32 bits con preescalador de 32 bits.
- Temporizador cuenta tiempo, el contador cuenta eventos
- Los 4 T/C son idénticos, solo cambia dirección base
- En el Reset T0 y T1 se habilitan y T2 y T3 son deshabilitados.
- T0 a T3: 2 entradas de Captura de 32 bits por temporizador
2 salidas de Match de 32 bits
- T2: 2 salidas más de Match
- Modo captura o match pueden generar interrupciones los match pueden generar DMA.
- 4 registros de match en cada timer

2.2.1. Generalidades. Aplicaciones

- Generación de retardos de tiempo precisos
- Generación de interrupciones periódicas
- Conteo de eventos externos
- Demodulación de ancho de pulso
- Generación de secuencias de pulsos arbitrarias

2.2.2. Diagrama de los Timers 0,1,2,3



2.2.3. Posibilidades de los registros Match

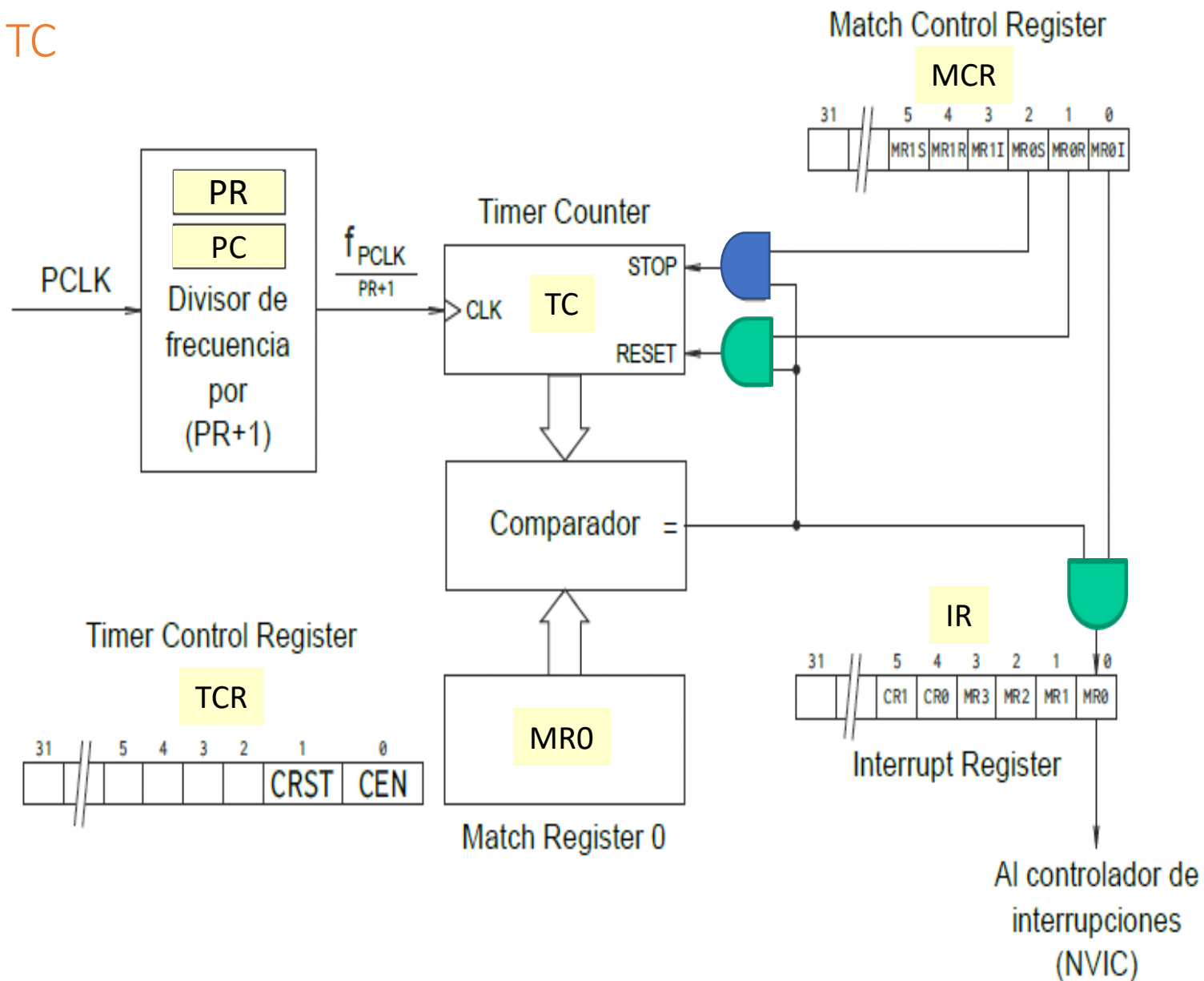
Registros Match

- Operación continua con generación de interrupción opcional según el match
- Parada del temporizador según el match con generación de interrupción opcional
- Reseteo del temporizador según el match con generación de interrupción opcional

Salidas asociadas a los registros match

- Poner nivel bajo según el match
- Poner nivel alto según el match
- Cambiar el nivel de salida según el match
- No hacer nada

Acciones sobre el TC



2.2.4. Entradas/Salidas asociadas a los Timer

En el IOCON se pueden seleccionar varios pines para las funciones CAP o MAT.

- Cuando más de un pin es seleccionado para la salida MAT, todos los pines son iguales
- Cuando más de un pin es seleccionado para la entrada CAP, se usa el pin con el menor número de puerto

Pin	Type	Description
T0_CAP1:0 T1_CAP1:0 T2_CAP1:0 T3_CAP1:0	Input	Capture Signals- A transition on a capture pin can be configured to load one of the Capture Registers with the value in the Timer Counter and optionally generate an interrupt. Capture functionality can be selected from a number of pins. When more than one pin is selected for a Capture input on a single TIMER0/1 channel, the pin with the lowest Port number is used Timer/Counter block can select a capture signal as a clock source instead of the PCLK derived clock. For more details see Section 24.6.11 .
T0_MAT1:0 T1_MAT1:0 T2_MAT3:0 T3_MAT1:0	Output	External Match Output - When a match register (MR3:0) equals the timer counter (TC) this output can either toggle, go low, go high, or do nothing. The External Match Register (EMR) controls the functionality of this output. Match Output functionality can be selected on a number of pins in parallel.

2.2.5. Mapa de memoria de los registros del timer (I)

Nombre	Descripción	Acceso	Reset	Offset
IR	Registro de interrupción. Borra la interrupción cuando se escribe. Cuando se lee identifica cuál de las 8 fuentes de interrupción está pendiente. (4 MR Y 4 CR)	R/W	0x0	0x000
TCR	Registro de control del temporizador. El TC puede ser deshabilitado y reseteado por el TCR.	R/W	0x0	0x004
TC	Registro temporizador. Es incrementado cada PR+1 ciclos de PCLK. Es controlado por el TCR.	R/W	0x0	0x008
PR	Registro Preescala. Cuando el Contador de preescala es igual a este valor, el siguiente pulso de reloj incrementa el TC y borra el PC.	R/W	0x0	0x00C
PC	Contador preescala. El contador se incrementa hasta el valor PR. Cuando PR es alcanzado , el TC es incrementado y el PC borrado. El PC es observable y controlable por el bus interfase.	R/W	0x0	0x010

2.2.5. Mapa de memoria de los registros del timer (II)

Nombre	Descripción	Acceso	Reset	Offset
MCR	Registro de control match. Se usa para controlar si una interrupción es generada y si el TC es reseteado cuando ocurre un match.	R/W	0x0	0x014
MR0	Registro match 0. El MR0 puede ser habilitado por el MCR para resetear el TC, parar el TC y el PC y/o generar una interrupción cada vez que hay un match.	R/W	0x0	0x018
MR1	Registro match 1. Ídem.	R/W	0x0	0x01C
MR2	Registro match 2. Ídem.	R/W	0x0	0x020
MR3	Registro match 3. Ídem.	R/W	0x0	0x024

2.2.5. Mapa de memoria de los registros del timer (III)

Nombre	Descripción	Acceso	Reset	Offset
CCR	Reg. control captura. Controla los flancos de las entradas de captura para cargar los registros de captura y la generación de una interrupción en la captura.	R/W	0x0	0x028
CR0	Registro de captura 0. CR0 es cargado con el valor de TC cuando hay un evento en el CAPn:0 en la entrada.	RO	0x0	0x02C
CR1	Registro de captura 1. Ídem.	RO	0x0	0x030
EMR	Registro externo del Match. Controla los pines asociados al Match MATn:0	R/W	0x0	0x03C
CTCR	Selecciona entre el modo temporizador y contador y en modo contador selecciona la señal y el flanco para conteo.	R/W	0x0	0x070

IR registro de interrupción

Si una interrupción es generada, el bit se pone a 1. Escribiendo un 1 en el bit se borra la señal de interrupción. En el caso de los registros match también se borra la petición DMA.

Table 540. Interrupt Register (IR - addresses 0x4000 4000 (TIMER0), 0x4000 8000 (TIMER1), 0x4009 0000 (TIMER2), 0x4009 4000 (TIMER3)) bit description

Bit	Symbol	Description	Reset Value
0	MR0INT	Interrupt flag for match channel 0.	0
1	MR1INT	Interrupt flag for match channel 1.	0
2	MR2INT	Interrupt flag for match channel 2.	0
3	MR3INT	Interrupt flag for match channel 3.	0
4	CR0INT	Interrupt flag for capture channel 0 event.	0
5	CR1INT	Interrupt flag for capture channel 1 event.	0
31:6	-	Reserved. Read value is undefined, only zero should be written.	-

TCR registro de control del Timer y TC contador del Timer

El registro TC es incrementado cuando el contador de preescala alcanza su cuenta. Cuando el TC alcanza 0xFFFFFFFF empieza por 0x00000000

Table 542. Timer counter registers (TC - addresses 0x400 4008 (TIMER0), 0x4000 8008 (TIMER1), 0x4009 0008 (TIMER2), 0x4009 4008 (TIMER3)) bit description

Bit	Symbol	Description	Reset value
31:0	TC	Timer counter value.	0

Bit	Symbol	Description	Reset Value
0	CEN	When 1, the Timer Counter and Prescale Counter are enabled for counting. When 0, the counters are disabled.	0
1	CRST	When 1, the Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero.	0
31:2	-	Reserved. Read value is undefined, only zero should be written.	NA

PR registro de preescala y PC contador de preescala

- PR especifica el valor máximo del contador de preescala
- PC es el contador de preescala. Se incrementa cada PCLK. Cuando se alcanza el valor de PR el TC se incrementa en 1 y el PC se resetea

Table 543. Timer prescale registers (PR - addresses 0x4000 400C (TIMER0), 0x4000 800C (TIMER1), 0x4009 000C (TIMER2), 0x4009 400C (TIMER3)) bit description

Bit	Symbol	Description	Reset value
31:0	PR	Prescale counter maximum value.	0

Table 544. Timer prescale counter registers (PC - addresses 0x4000 4010 (TIMER0), 0x4000 8010 (TIMER1), 0x4009 0010 (TIMER2), 0x4009 4010 (TIMER3)) bit description

Bit	Symbol	Description	Reset value
31:0	PC	Prescale counter value.	0

MCR registro de control del match y MR0 a MR3 registros de match

El registro de match se compara continuamente con el TC. Cuando el TC alcanza el valor del MRn, se disparan las acciones definidas en el MCR

Bit	Symbol	Value	Description	Reset Value
0	MR0I		Interrupt on MR0	0
		1	Interrupt is generated when MR0 matches the value in the TC.	
		0	Interrupt is disabled	
1	MR0R		Reset on MR0	0
		1	TC will be reset if MR0 matches it.	
		0	Feature disabled.	
2	MR0S	1	Stop on MR0	0
		1	TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC.	
		0	Feature disabled.	
----- ----- ----- ----- -----				
9	MR3I		Interrupt on MR3	0
		1	Interrupt is generated when MR3 matches the value in the TC.	
		0	This interrupt is disabled	
10	MR3R		Reset on MR3	0
		1	TC will be reset if MR3 matches it.	
		0	Feature disabled.	
11	MR3S		Stop on MR3	0
		1	TC and PC will be stopped and TCR[0] will be set to 0 if MR3 matches the TC.	
		0	Feature disabled.	
31:12	-		Reserved. Read value is undefined, only zero should be written.	NA

Table 546. Timer match registers (MR[0:3], addresses 0x4000 4018 (MR0) to 0x4000 4024 (MR3) (TIMER0), 0x4000 8018 (MR0) to 0x4000 8024 (MR3) (TIMER1), 0x4009 0018 (MR0) to 0x4009 0024 (MR3) (TIMER2), 0x4009 4018 (MR0) to 0x4009 4024 (MR3)(TIMER3)) bit description

Bit	Symbol	Description	Reset value
31:0	MATCH	Timer counter match value.	0

CCR registro de control de captura y CR0 a CR1 registro de captura

El registro de captura se asocia a un pin y cuando se produce un evento en ese pin, se realiza la captura del TC.

Bit	Symbol	Description	Reset value
31:0	CAP	Timer counter capture value.	0

Bit	Symbol	Value	Description	Reset Value
0	CAP0RE		Capture on CAPn.0 rising edge	0
		1	A sequence of 0 then 1 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	
		0	This feature is disabled.	
1	CAP0FE		Capture on CAPn.0 falling edge	0
		1	A sequence of 1 then 0 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	
		0	This feature is disabled.	
2	CAP0I		Interrupt on CAPn.0 event	0
		1	A CR0 load due to a CAPn.0 event will generate an interrupt.	
		0	This feature is disabled.	
3	CAP1RE		Capture on CAPn.1 rising edge	0
		1	A sequence of 0 then 1 on CAPn.1 will cause CR1 to be loaded with the contents of TC.	
		0	This feature is disabled.	
4	CAP1FE		Capture on CAPn.1 falling edge	0
		1	A sequence of 1 then 0 on CAPn.1 will cause CR1 to be loaded with the contents of TC.	
		0	This feature is disabled.	
5	CAP1I		Interrupt on CAPn.1 event	0
		1	A CR1 load due to a CAPn.1 event will generate an interrupt.	
		0	This feature is disabled.	
31:6	-		Reserved. Read value is undefined, only zero should be written.	NA

EMR registro externo del match

Permite el control y conocer el estado de los pines externos del match

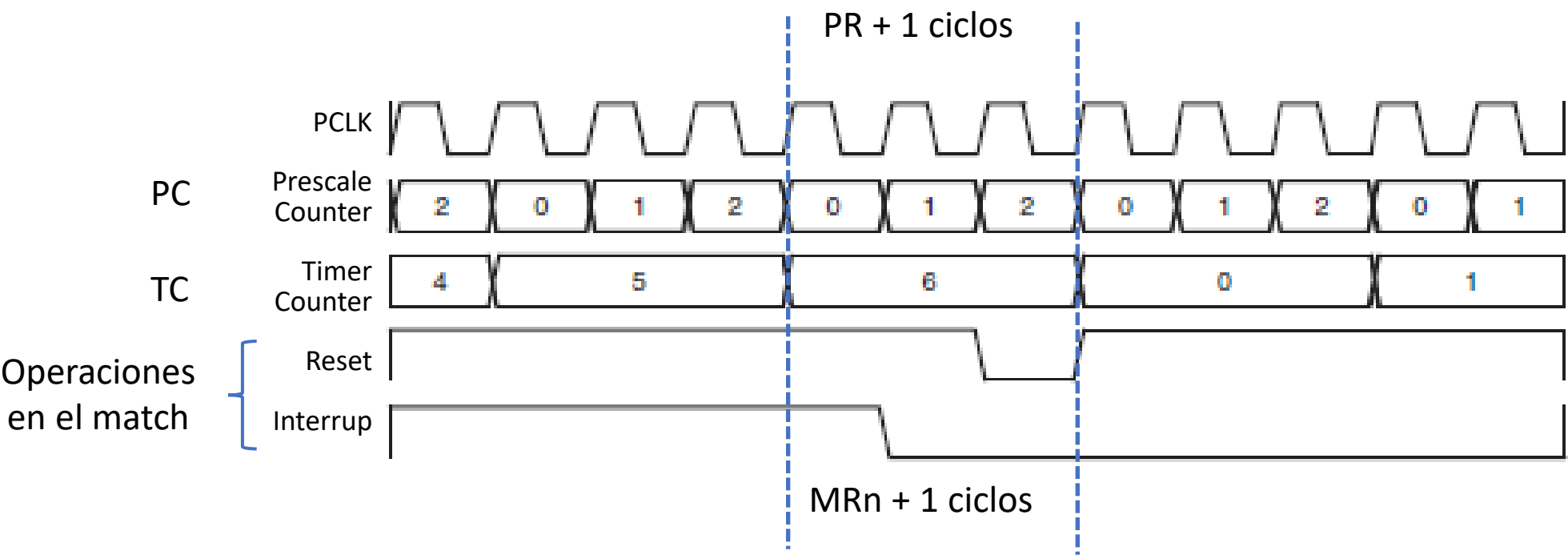
Bit	Symbol	Value	Description	Reset value
0	EM0		External Match 0. When a match occurs between the TC and MR0, this bit can either toggle, go low, go high, or do nothing, depending on bits 5:4 of this register. This bit can be driven onto a MATn.0 pin, in a positive-logic manner (0 = low, 1 = high).	0
----- ----- ----- ----- -----				
3	EM3		External Match 3. When a match occurs between the TC and MR3, this bit can either toggle, go low, go high, or do nothing, depending on bits 11:10 of this register. This bit can be driven onto a MATn.0 pin, in a positive-logic manner (0 = low, 1 = high).	0
5:4	EMC0		External Match Control 0. Determines the functionality of External Match 0.	00
	0x0		Do Nothing.	
	0x1		Clear the corresponding External Match bit/output to 0 (MATn.m pin is LOW if pinned out).	
	0x2		Set the corresponding External Match bit/output to 1 (MATn.m pin is HIGH if pinned out).	
	0x3		Toggle the corresponding External Match bit/output.	
----- ----- ----- ----- -----				
11:10	EMC3		External Match Control 3. Determines the functionality of External Match 3.	00
	0x0		Do Nothing.	
	0x1		Clear the corresponding External Match bit/output to 0 (MATn.m pin is LOW if pinned out).	
	0x2		Set the corresponding External Match bit/output to 1 (MATn.m pin is HIGH if pinned out).	
	0x3		Toggle the corresponding External Match bit/output.	
31:12	-		Reserved. Read value is undefined, only zero should be written.	NA

CTCR registro de control de cuenta

Selecciona entre modo Contador y Timer y en modo contador selecciona el pin y el tipo de flanco para la cuenta

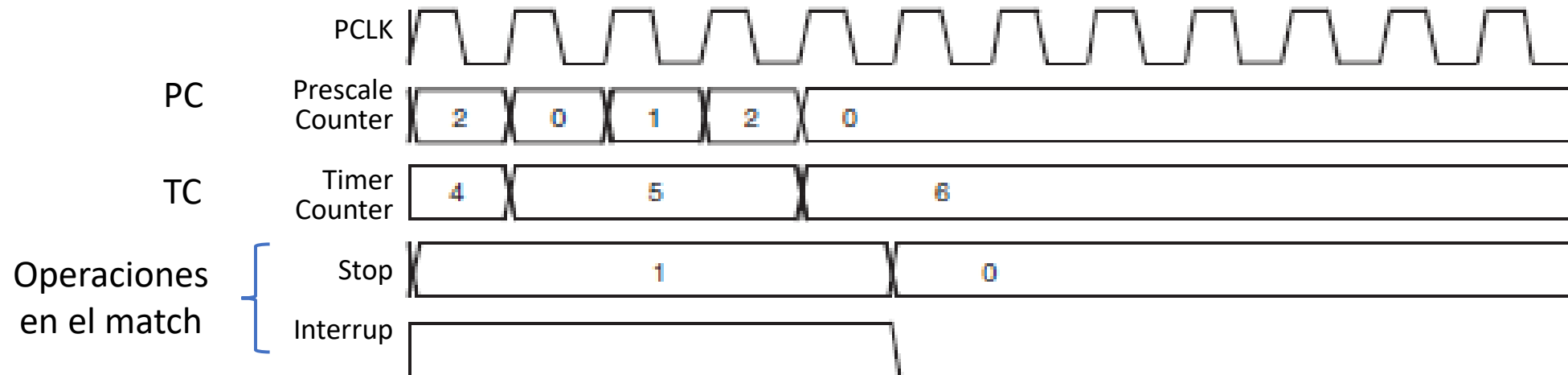
Bit	Symbol	Value	Description	Reset Value
1:0	CTMODE		Counter/Timer Mode This field selects which rising PCLK edges can increment Timer's Prescale Counter (PC), or clear PC and increment Timer Counter (TC). Timer Mode: the TC is incremented when the Prescale Counter matches the Prescale Register.	00
		0x0	Timer Mode: every rising PCLK edge	
		0x1	Counter Mode: TC is incremented on rising edges on the CAP input selected by bits 3:2.	
		0x2	Counter Mode: TC is incremented on falling edges on the CAP input selected by bits 3:2.	
		0x3	Counter Mode: TC is incremented on both edges on the CAP input selected by bits 3:2.	
3:2	CINSEL		Count Input Select When bits 1:0 in this register are not 00, these bits select which CAP pin is sampled for clocking. Note: If Counter mode is selected for a particular CAPn input in the TnCTCR, the 3 bits for that input in the Capture Control Register (TnCCR) must be programmed as 000. However, capture and/or interrupt can be selected for the other 3 CAPn inputs in the same timer.	0
		0x0	CAPn.0 for TIMERN	
		0x1	CAPn.1 for TIMERN	
		0x2	Reserved	
		0x3	Reserved	
31:4	-		Reserved. Read value is undefined, only zero should be written.	NA

2.2.6. Ejemplos de operación del temporizador (1)



Ciclo temporizador con PR=2, MRn=6 y reset e interrupción habilitados por el match.

2.2.6. Ejemplos de operación del temporizador (2)



Ciclo temporizador con $PR=2$, $MRn=6$ y parada e interrupción habilitados por el match.

2.2.7. Acceso a los registros (1)

```
typedef struct{
    __IO uint32_t      IR;                /*!< Offset: 0x000 Interrupt Register (R/W) */
    __IO uint32_t      TCR;               /*!< Offset: 0x004 Timer Control Register (R/W) */
    __IO uint32_t      TC;                /*!< Offset: 0x008 Timer Counter Register (R/W) */
    __IO uint32_t      PR;                /*!< Offset: 0x00C Prescale Register (R/W) */
    __IO uint32_t      PC;                /*!< Offset: 0x010 Prescale Counter Register (R/W) */
    __IO uint32_t      MCR;               /*!< Offset: 0x014 Match Control Register (R/W) */
    __IO uint32_t      MR0;               /*!< Offset: 0x018 Match Register 0 (R/W) */
    __IO uint32_t      MR1;               /*!< Offset: 0x01C Match Register 1 (R/W) */
    __IO uint32_t      MR2;               /*!< Offset: 0x020 Match Register 2 (R/W) */
    __IO uint32_t      MR3;               /*!< Offset: 0x024 Match Register 3 (R/W) */
    __IO uint32_t      CCR;               /*!< Offset: 0x028 Capture Control Register (R/W) */
    __I  uint32_t      CR0;               /*!< Offset: 0x02C Capture Register 0 (R/ ) */
    __I  uint32_t      CR1;               /*!< Offset: 0x030 Capture Register 1 (R/ ) */
    uint32_t          RESERVED0[2];
    __IO uint32_t      EMR;               /*!< Offset: 0x03C External Match Register (R/W) */
    uint32_t          RESERVED1[12];
    __IO uint32_t      CTCR;              /*!< Offset: 0x070 Count Control Register (R/W) */
} LPC_TIM_TypeDef;
```

2.2.7. Acceso a los registros (2)

```
#define LPC_APB0_BASE    (0x40000000UL)
```

```
#define LPC_APB1_BASE    (0x40080000UL)
```

```
#define LPC_TIM0_BASE    (LPC_APB0_BASE + 0x04000)
```

```
#define LPC_TIM1_BASE    (LPC_APB0_BASE + 0x08000)
```

```
#define LPC_TIM2_BASE    (LPC_APB1_BASE + 0x10000)
```

```
#define LPC_TIM3_BASE    (LPC_APB1_BASE + 0x14000)
```

```
#define LPC_TIM0          ((LPC_TIM_TypeDef *) LPC_TIM0_BASE)
```

```
#define LPC_TIM1          ((LPC_TIM_TypeDef *) LPC_TIM1_BASE)
```

```
#define LPC_TIM2          ((LPC_TIM_TypeDef *) LPC_TIM2_BASE)
```

```
#define LPC_TIM3          ((LPC_TIM_TypeDef *) LPC_TIM3_BASE)
```

2.2.8. Proceso de configuración de los T/C (1)

0. Detener el Timer

1. Resetear el Timer
2. Configurar el Timer para generar un "match" cada periodo de tiempo deseado:
 - a) Seleccionar una operación "interrupt on match", "reset on match" y/o "stop on match" con MRn (n=0..3): De esta forma, cada vez que el valor del registro contador del Timer alcance el valor programado en el registro de match n de este Timer, el registro contador volverá a 0 y/o se parará y se activará el flag de petición de interrupción del Timer. La interrupción del Timer no estará habilitada, pero se sondeará el flag de petición de interrupción por software para detectar el momento en el que el valor del Timer iguala al del registro de match.

2.2.8. Proceso de configuración de los T/C (2)

b) Programar una preescala:

Esto hará que el Timer Counter TC se incremente cada periodo de tiempo deseado.

$CCLK = 120 \text{ MHz}$; $PCLK = CCLK/2 = 60 \text{ MHz} = 60000 \times 10^3 \text{ Hz} \rightarrow \text{PeripheralClock}$

$PR=0 \rightarrow 1 \cdot (1/\text{PeripheralClock}) \text{ s}$

$PR=1 \rightarrow 2 \cdot (1/\text{PeripheralClock}) \text{ s}$

$PR=2 \rightarrow 3 \cdot (1/\text{PeripheralClock}) \text{ s}$

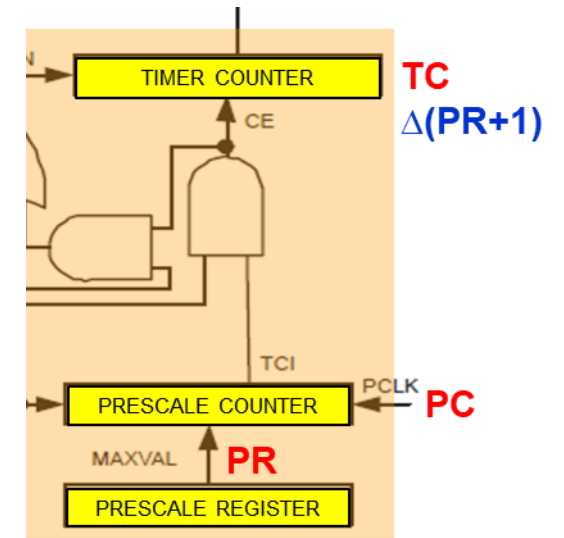
 $PR=n \rightarrow (n+1) \cdot (1/\text{PeripheralClock}) \text{ s} = \Delta TC \text{ s}$

$PR = (\text{PeripheralClock} \cdot \Delta TC(s)) - 1$

Caso $\Delta TC=1\text{ms}$; $\Delta TC \text{ s} = 10^{-3} \text{ s}$; $PR = (\text{PeripheralClock} \cdot 10^{-3}) - 1$

Caso $\Delta TC=0,1\text{ms}$; $\Delta TC \text{ s} = 10^{-4} \text{ s}$; $PR = (\text{PeripheralClock} \cdot 10^{-4}) - 1$

Caso $\Delta TC=1\mu\text{s}$; $\Delta TC \text{ s} = 10^{-6} \text{ s}$; $PR = (\text{PeripheralClock} \cdot 10^{-6}) - 1$



2.2.8. Proceso de configuración de los T/C (3)

- c) Programar en el registro de match n el periodo de tiempo deseado.**
El registro de match debe cargarse con un valor inferior en una unidad al valor deseado.

$$TC\ s = \Delta TC\ s * (MRn + 1) = (PR + 1) * (MRn + 1) / \text{PeripheralClock}$$

$$MRn = [TC\ s * \text{PeripheralClock} / (PR + 1)] - 1 \quad (MRn \neq 0)$$

$$\text{Caso } \Delta TC = 1\text{ms}; \Delta TC\ s = 10^{-3}\ s; PR + 1 = (\text{PeripheralClock} * 10^{-3})$$

$$MRn = [TC\ ms * 10^{-3} * \text{PeripheralClock} / (PR + 1)] - 1 = TC\ ms - 1$$

$$\text{Caso } \Delta TC = 0,1\text{ms}; \Delta TC\ s = 10^{-4}\ s; PR + 1 = (\text{PeripheralClock} * 10^{-4})$$

$$MRn = [TC\ ms * 10^{-3} * \text{PeripheralClock} / (PR + 1)] - 1 = 10 * TC\ ms - 1$$

$$\text{Caso } \Delta TC = 1\mu s; \Delta TC\ s = 10^{-6}\ s; PR + 1 = (\text{PeripheralClock} * 10^{-6})$$

$$MRn = [TC\ ms * 10^{-3} * \text{PeripheralClock} / (PR + 1)] - 1 = 1000 * TC\ ms - 1$$

2.2.8. Proceso de configuración de los T/C (4)

3. **Esperar** la coincidencia de que el contador del Timer alcance el valor programado en el registro de match n. Para ello se comprueba el flag de petición de interrupción situado en el registro de interrupción del Timer (IR)
4. **Borrar el bit** de petición de interrupción

2.2.9. Ejemplo temporizador (1)

```
/*=====
```

Programa de ejemplo de programación del timer 0 en C. Se programa el timer 0 y su registro de match 0 para temporizar un periodo de 1 segundo con incremento del TC de 0,1ms. El LED conectado al pin P0[13] cambia su estado cada vez que transcurre este tiempo.

```
=====*/
```

```
#include <LPC407x_8x_177x_8x.h>
```

```
int main(void)
```

```
{ /* Programar el Timer 0 para que provoque un evento "match" cada  
segundo*/
```

```
/* Resetear el timer 0.*/
```

```
LPC_TIM0->TCR = 0; LPC_TIM0->TC = 0; LPC_TIM0->PC = 0;
```

```
/* Seleccionar "interrupt on match" y "reset on match" con MR0.*/
```

```
LPC_TIM0->MCR |= 3;
```

2.2.9. Ejemplo temporizador (2)

```
/* Programar una preescala con  $\Delta TC$  de 1 ms en el PR.*/  
    LPC_TIM0->PR = (PeripheralClock * 10-4) - 1;  
/* Programar en el registro match el número deseado de milisegundos de  
intervalo. */  
    LPC_TIM0->MR0 = 10 * 1000 - 1;  
/* Configurar el pin P0[13] como salida.*/  
    LPC_GPIO0->DIR |= (1<<13);  
/* Asegurar el bit de IR a 0.*/  
    LPC_TIM0->IR = 1;  
/* Poner en marcha el timer.*/  
    LPC_TIM0->TCR = 1;  
    while (1)  
    { while (!(LPC_TIM0->IR & 1);  
        LPC_TIM0->IR = 1;  
        LPC_GPIO0->PIN ^= (1<<13);  
    }  
}
```

2.2.10. Biblioteca “timer_lpc40xx.h”

```
void timer_inicializar(LPC_TIM_TypeDef *timer_regs)
void timer_retardo_ms(LPC_TIM_TypeDef *timer_regs,
                      uint32_t retardo_en_ms)
void timer_retardo_us(LPC_TIM_TypeDef *timer_regs,
                      uint32_t retardo_en_us)
void timer_iniciar_ciclos_ms(LPC_TIM_TypeDef *timer_regs,
                             uint32_t periodo_en_ms)
void timer_esperar_fin_ciclo(LPC_TIM_TypeDef *timer_regs)
void timer_iniciar_conteo_ms(LPC_TIM_TypeDef *timer_regs)
```

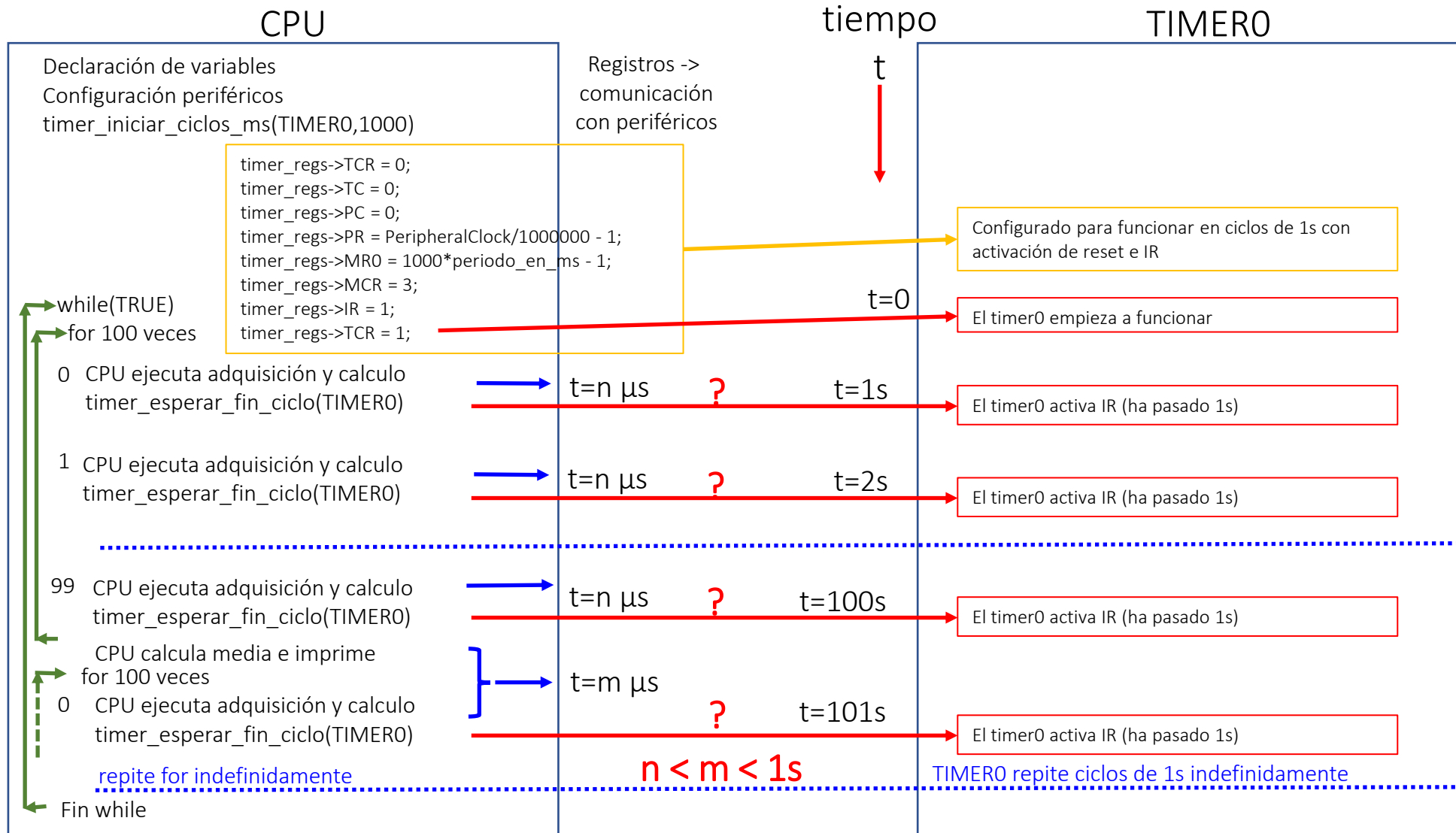
2.2.11. Dependencia CPU-Timer

```
timer_retardo_ms(TIMER0,500); /*La CPU tiene que esperar a que  
la función termine, es decir a que el timer cuente 500ms */
```

```
timer_iniciar_ciclos_ms(TIMER0,500); /* La CPU arranca el  
timer pero no espera a que termine, lo deja contando */
```

```
timer_esperar_fin_ciclo(TIMER0); /* La CPU espera a que el  
timer termine */
```

2.2.12. Dispositivos en paralelo. Comunicación por muestreo



Evolución en el tiempo de la ejecución de instrucciones en la CPU y su relación con el timer

2.2.13. Conteo de tiempo

```
timer_iniciar_conteo_ms(TIMER0); /* El timer empieza a  
contar tiempo y el TC se leerá cuando interese */
```

```
/* Se puede capturar el valor de TC de acuerdo a ciertas  
señales externas */
```