

## 2 Periféricos

### 2.1 Temporizadores (*Timers*) 0, 1, 2, 3

Los *timers* están diseñados para funcionar tanto como temporizadores como contadores. Cuando funcionan como temporizadores, cada uno lleva a cabo una cuenta a través de un contador principal a una frecuencia previamente configurada. Si funcionan como contadores, la frecuencia dependerá de una señal externa. El modo de funcionamiento se configura en el registro CTCR.

Para controlar ese contador principal, se dispone de un contador secundario, llamado *prescaler*. Es el *prescaler* el que se va incrementando según la señal de reloj PCLK\_TIMERn (configurable en los registros PCLKSEL0 ó PCLKSEL1) si el *timer* es usado como temporizador (n=0, 1, 2, 3 según el *timer*); o según una señal externa si se usa como contador. Este contador cuenta de forma ascendente y su valor de desbordamiento es configurable en el registro PR. Cuando desborde, el contador principal se incrementará en una unidad, el *prescaler* se reseteará automáticamente y comenzará a contar de nuevo. El valor del *prescaler* se encuentra en el registro PC, y el del contador principal en el registro TC. Ambos se habilitan en el registro TCR.

Cuando los *timers* se usen como temporizadores, la frecuencia de cuenta depende de la configuración del *prescaler* y de la frecuencia de la señal PCLK\_TIMERn.

La única diferencia entre los *timer* es que al resetear, los *timers* 0 y 1 están alimentados (registro PCONP) y los *timers* 2 y 3 no lo están.

Los *timers* disponen cada uno de 2 pines de entrada de captura (CAP0[1:0] ... CAP3[1:0]) y 2 pines de salida tras un *match* salvo el *timer* 2, que cuenta con 4 pines de salida tras un *match* (MAT0[1:0], MAT1[1:0], MAT2[3:0], MAT3[1:0]).

Cuando se produzca un flanco en uno de los pines de captura (tanto el flanco como el pin de captura, se configuran en el registro CTCR), el valor del contador se carga en uno de los registros CRn, siendo n el número de registro de captura, que se corresponderá con el del pin de entrada de captura. Puede configurarse una interrupción cada vez que ocurra una captura, a través del registro CCR.

Se dice que se produce un *match* cuando el valor del contador es igual a alguno de los registros de *match* MRn. Pueden configurarse interrupciones para que tengan lugar cuando se produzca un *match*, en el registro MCR. Los pines de salida tras un *match*, pueden cambiar automáticamente de estado lógico cuando se produzca un *match*, según se configuren en el registro EMR.

Para que pueda producirse cualquier interrupción tras *match* ó captura, éstas deben de estar habilitadas previamente en el NVIC. Los flags de interrupción se encuentran en el registro IR, y no hay que olvidarse de borrarlos en la rutina de interrupción, para que pueda darse dicha interrupción de nuevo.

Pueden programarse varias salidas tras *match* en un mismo *timer*, funcionando todas en paralelo. En cambio, si en un mismo temporizador son programadas varias entradas de captura, entonces sólo se utilizará la que tenga el menor número de puerto.

El posible uso que se le puede dar a los temporizadores es del todo convencional: medida de tiempos o intervalos, demodulador de anchura de pulsos, etc...

Un *match* en MR0 ó MR1 pueden provocar un *request* de la GPDMA del microcontrolador.

A continuación se describen con más detalle los registros de configuración y uso de los timers:

**Tabla 111 Mapa de registros de los temporizadores 0, 1, 2 y 3.**

| Generic Name | Description   | Access | Reset Value | TIMERn Register/ Name & Address  |
|--------------|---|--------|-------------|--|
| IR           | Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending.  | R/W    | 0           | T0IR - 0x4000 4000<br>T1IR - 0x4000 8000<br>T2IR - 0x4009 0000<br>T3IR - 0x4009 4000     |
| TCR          | Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.   | R/W    | 0           | T0TCR - 0x4000 4004<br>T1TCR - 0x4000 8004<br>T2TCR - 0x4009 0004<br>T3TCR - 0x4009 4004 |
| TC           | Timer Counter. The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.  | R/W    | 0           | T0TC - 0x4000 4008<br>T1TC - 0x4000 8008<br>T2TC - 0x4009 0008<br>T3TC - 0x4009 4008     |
| PR           | Prescale Register. When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.  | R/W    | 0           | T0PR - 0x4000 400C<br>T1PR - 0x4000 800C<br>T2PR - 0x4009 000C<br>T3PR - 0x4009 400C     |
| PC           | Prescale Counter. The 32-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface. | R/W    | 0           | T0PC - 0x4000 4010<br>T1PC - 0x4000 8010<br>T2PC - 0x4009 0010<br>T3PC - 0x4009 4010     |
| MCR          | Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.   | R/W    | 0           | T0MCR - 0x4000 4014<br>T1MCR - 0x4000 8014<br>T2MCR - 0x4009 0014<br>T3MCR - 0x4009 4014 |
| MR0          | Match Register 0. MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.  | R/W    | 0           | T0MR0 - 0x4000 4018<br>T1MR0 - 0x4000 8018<br>T2MR0 - 0x4009 0018<br>T3MR0 - 0x4009 4018 |
| MR1          | Match Register 1. See MR0 description.  | R/W    | 0           | T0MR1 - 0x4000 401C<br>T1MR1 - 0x4000 801C<br>T2MR1 - 0x4009 001C<br>T3MR1 - 0x4009 401C |
| MR2          | Match Register 2. See MR0 description.  | R/W    | 0           | T0MR2 - 0x4000 4020<br>T1MR2 - 0x4000 8020<br>T2MR2 - 0x4009 0020<br>T3MR2 - 0x4009 4020 |
| MR3          | Match Register 3. See MR0 description.  | R/W    | 0           | T0MR3 - 0x4000 4024<br>T1MR3 - 0x4000 8024<br>T2MR3 - 0x4009 0024<br>T3MR3 - 0x4009 4024 |
| CCR          | Capture Control Register. The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.  | R/W    | 0           | T0CCR - 0x4000 4028<br>T1CCR - 0x4000 8028<br>T2CCR - 0x4009 0028<br>T3CCR - 0x4009 4028 |
| CR0          | Capture Register 0. CR0 is loaded with the value of TC when there is an event on the CAPn.0(CAP0.0 or CAP1.0 respectively) input.   | RO     | 0           | T0CR0 - 0x4000 402C<br>T1CR0 - 0x4000 802C<br>T2CR0 - 0x4009 002C<br>T3CR0 - 0x4009 402C |
| CR1          | Capture Register 1. See CR0 description.  | RO     | 0           | T0CR1 - 0x4000 4030<br>T1CR1 - 0x4000 8030   |

| Generic Name | Description   | Access | Reset Value | TIMERn Register/ Name & Address  |
|--------------|---|--------|-------------|--|
|              |   |        |             | T2CR1 - 0x4009 0030<br>T3CR1 - 0x4009 4030   |
| EMR          | External Match Register. The EMR controls the external match pins MATn.0-3 (MAT0.0-3 and MAT1.0-3 respectively).                          | R/W    | 0           | T0EMR - 0x4000 403C<br>T1EMR - 0x4000 803C<br>T2EMR - 0x4009 003C<br>T3EMR - 0x4009 403C     |
| CTCR         | Count Control Register. The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting. | R/W    | 0           | T0CTCR - 0x4000 4070<br>T1CTCR - 0x4000 8070<br>T2CTCR - 0x4009 0070<br>T3CTCR - 0x4009 4070 |

Destacar que cada registro tiene un equivalente en cada *timer* y que por lo tanto, todos los *timers* se configuran y se manejan de la misma manera.

A continuación paso a comentar cómo funcionan cada uno de estos registros:

- **Registro de interrupciones (TnIR):**

Los cuatro primeros bits de éste registro se corresponden con los flags de las interrupciones de *match*, desde el MR0 hasta el MR3. Los dos siguientes bits se corresponden con flags de captura, el CR0 y CR1. Si se genera una interrupción como resultado de una captura o de una *match* se pondrá a uno el bit correspondiente. Escribiendo un '1' se resetea la interrupción. Escribir un cero no tiene ningún efecto.

- **Registro de control de temporizador (TnTCR):**

Sólo los dos primeros bits de este registro son útiles (el resto reservados). El bit cero habilita el contador del *timer* (Counter Enable) y el contador de *prescale*. Cuando el bit 1 es puesto a nivel alto (Counter Reset), el contador del *timer* y el contador de *prescale* son reseteados en el próximo flanco positivo de PCLK, y quedan en ese estado hasta que es puesto a cero.

- **Registro de control de cuenta (TnCTCR):**

Los dos primeros bits sirven para especificar si funcionará como temporizador (00) o como contador (Counter/Timer mode). Si se da el último caso, además se especifica en qué flanco se producirá la cuenta: subida (01), bajada(10), o ambos (11).

Los dos bits siguientes (Count Input Select) sirven para especificar cuál de los pines de entrada de captura se va a utilizar (00: CAPn.0, 01: CAPn.1).

- **Registro de Contador de temporizador (TnTC):**

Este registro lleva a cabo la cuenta del temporizador, desde 0x00000000 hasta 0xFFFFFFFF (32 bits). Se incrementa cada vez que se desborda el contador de *prescale*. Cuando llega al valor máximo se resetea y continúa contando desde el principio. Éste desbordamiento no provoca interrupción, pero pueden (y deben) usarse los registros de *match* (MR0 ... MR3) para provocar una interrupción cuando la cuenta llegue a un valor determinado.

- **Registro de *prescale* (TnPR):**

En este registro de 32 bits se especifica el valor máximo de *prescale*. Cuando el *prescale* llegue a este valor desbordará, provocando un incremento en el contador del timer. Una vez haya desbordado, continuará contando desde cero.

- **Contador de *prescale* (TnPC):**

Este registro guarda el valor de cuenta del *prescaler*. Este valor es incrementado en cada flanco de PCLK. Cuando este contador alcanza el valor almacenado en el registro TnPR, entonces su valor es reseteado en el siguiente flanco de PCLK y se incrementa el valor del contador principal TnTC. De esta manera, si por ejemplo, TnPR es 0 el contador principal TnTC se incrementará en cada flanco de PCLK (TnPC desborda en cada cuenta), si TnPR es 1, TnTC se incrementará cada 2 flancos de PCLK (TnPC desborda cada dos cuentas), etc.

- **Registros de *match* (MR0 ... MR3):**

Los registros de *match* están comparándose constantemente con el valor del contador del temporizador (TC). Cuando el valor de cuenta del timer coincide con alguno de estos registros puede ocurrir alguna de estas acciones: un pin de salida de *match* cambia a nivel alto, bajo, o al contrario del que tenga (*toggle*), puede producirse un reset en el contador del timer, parar el contador ó generar una interrupción. Estas acciones son controladas por el registro MCR y EMR.

- **Registro de control de *match* (TnMCR):**

Se usan sólo los 12 primeros bits, siendo cada grupo de tres bits relativo a un determinado registro de match (los tres primeros bits hacen referencia al registro de match 0, los tres siguientes registro de match 1, etc).

Si el primer bit (MRnI) de cada grupo de tres está a '1', se produce una interrupción cuando ocurre un *match*. Si el primer bit es '0', la interrupción por *match* está deshabilitada en ese timer. Si el segundo bit (MRnR) está a '1', se produce el reseteo del timer tras el *match*. El tercer bit (MRnS) hace que el contador del temporizador y el contador de *prescale* se detengan al producirse un *match*. Esto provoca que TnCR[0] (Counter Enable) se ponga a cero.

Puede verse mejor en la siguiente tabla:

**Tabla 112 Registros de control de Match (T[0/1/2/3]MCR - addresses 0x4000 4014, 0x4000 8014, 0x4009 0014, 0x4009 4014)**

| Bit   | Symbol | Value | Description  | Reset Value |
|-------|--------|-------|--|-------------|
| 0     | MROI   | 1     | Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC.                                  | 0           |
|       |        | 0     | This interrupt is disabled   |             |
| 1     | MR0R   | 1     | Reset on MR0: the TC will be reset if MR0 matches it.  | 0           |
|       |        | 0     | Feature disabled.  |             |
| 2     | MROS   | 1     | Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC.                      | 0           |
|       |        | 0     | Feature disabled.  |             |
| 3     | MR1I   | 1     | Interrupt on MR1: an interrupt is generated when MR1 matches the value in the TC.                                  | 0           |
|       |        | 0     | This interrupt is disabled   |             |
| 4     | MR1R   | 1     | Reset on MR1: the TC will be reset if MR1 matches it.  | 0           |
|       |        | 0     | Feature disabled.  |             |
| 5     | MR1S   | 1     | Stop on MR1: the TC and PC will be stopped and TCR[0] will be set to 0 if MR1 matches the TC.                      | 0           |
|       |        | 0     | Feature disabled.  |             |
| 6     | MR2I   | 1     | Interrupt on MR2: an interrupt is generated when MR2 matches the value in the TC.                                  | 0           |
|       |        | 0     | This interrupt is disabled   |             |
| 7     | MR2R   | 1     | Reset on MR2: the TC will be reset if MR2 matches it.  | 0           |
|       |        | 0     | Feature disabled.  |             |
| 8     | MR2S   | 1     | Stop on MR2: the TC and PC will be stopped and TCR[0] will be set to 0 if MR2 matches the TC.                      | 0           |
|       |        | 0     | Feature disabled.  |             |
| 9     | MR3I   | 1     | Interrupt on MR3: an interrupt is generated when MR3 matches the value in the TC.                                  | 0           |
|       |        | 0     | This interrupt is disabled   |             |
| 10    | MR3R   | 1     | Reset on MR3: the TC will be reset if MR3 matches it.  | 0           |
|       |        | 0     | Feature disabled.  |             |
| 11    | MR3S   | 1     | Stop on MR3: the TC and PC will be stopped and TCR[0] will be set to 0 if MR3 matches the TC.                      | 0           |
|       |        | 0     | Feature disabled.  |             |
| 31:12 | -      |       | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | NA          |

- **Registro de *match* externo (TnEMR):**

En este registro se configura el cambio de estado lógico de los pines de *match* externo y proporciona información del estado lógico de dichos pines.

Los cuatro primeros bits de este registro (EMn) se corresponden con el valor lógico de los pines de salida tras un *match*, para conocer dicho estado si fuese necesario. El bit 0 se corresponde con MATn[0], el bit 1 con MATn[1], etc.

El bit/pin que cambiará de estado se corresponderá con el registro de match que provoque el cambio, es decir, un match provocado por el registro T1MR0, provocará una salida externa de match en el pin MAT1[0]. Por ello hay que recordar que sólo el timer 2 dispone de 4 pines de salida tras match, una por cada registro MR0...MR3.

Para elegir qué tipo de salida externa se produce (cambio a nivel alto, bajo o *toggle*) están los ocho bits siguientes en grupos de dos bits (campo llamado EMCn), 00: no ocurre nada, 01: bit/pin a nivel bajo, 10: bit/pin a nivel alto, 11: toggle (cambio de nivel).

Puede verse con más claridad en las siguientes tablas:

**Tabla 113 Registro de match externos (T[0/1/2/3]EMR - addresses 0x4000 403C, 0x4000 803C, 0x4009 003C, 0x4009 403C)**

| Bit   | Symbol | 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           |
| 1     | EM1    | External Match 1. When a match occurs between the TC and MR1, this bit can either toggle, go low, go high, or do nothing, depending on bits 7:6 of this register. This bit can be driven onto a MATn.1 pin, in a positive-logic manner (0 = low, 1 = high).   | 0           |
| 2     | EM2    | External Match 2. When a match occurs between the TC and MR2, this bit can either toggle, go low, go high, or do nothing, depending on bits 9:8 of this register. This bit can be driven onto a MATn.2 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.3 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.   | 0           |
| 7:6   | EMC1   | External Match Control 1. Determines the functionality of External Match 1.   | 0           |
| 9:8   | EMC2   | External Match Control 2. Determines the functionality of External Match 2.   | 0           |
| 11:10 | EMC3   | External Match Control 3. Determines the functionality of External Match 3.   | 0           |
| 31:12 | -      | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.  | NA          |

Tabla 114 Valores de configuración de los registros de match externos

| EMR[11:10], EMR[9:8],<br>EMR[7:6], EMR[5:4] | Function  |
|---|---|
| 00  | Do Nothing.   |
| 01  | Clear the corresponding External Match bit/output to 0 (MATn.m pin is LOW if pinned out). |
| 10  | Set the corresponding External Match bit/output to 1 (MATn.m pin is HIGH if pinned out).  |
| 11  | Toggle the corresponding External Match bit/output.                                       |

- **Registros de captura (CRO, CR1):**

Cuando se produce un determinado evento en uno de los pines de captura (cambio a nivel alto, bajo o simplemente un cambio de nivel cualquiera) el valor del contador del temporizador correspondiente, se carga en estos registros. Esto se configura desde el registro CCR.

- **Registro de control de captura (TnCCR):**

Este registro tiene útiles los 6 primeros bits, siendo cada grupo de tres bits relativo a cada uno de los posibles pines de captura (los tres primeros bits hacen referencia a CAP0 y los tres siguientes a CAP1). Ambos grupos de tres bits funcionan de la misma manera.

El primer bit (CAPnRE) provoca la carga del valor de cuenta en el registro CRn, cuando tiene lugar un flanco de subida en CAPn. El segundo bit (CAPnFE) tiene la misma función, salvo que en este caso es en flanco de bajada. Si el tercer bit (CAPnI) es '1', se habilita la interrupción. Así cada vez que se produzca un flanco cualquiera en el pin determinado, se producirá una interrupción.



La siguiente tabla puede resultar aclaratoria:

**Tabla 115 Registro de control de captura (T[0/1/2/3]CCR - 0x4000 4028, 0x4000 8028, 0x4009 0028, 0x4009 4028)**

| Bit  | Symbol | Value | Description   | Reset Value |
|------|--------|-------|---|-------------|
| 0    | CAPORE | 1     | Capture on CAPn.0 rising edge: a sequence of 0 then 1 on CAPn.0 will cause CR0 to be loaded with the contents of TC.  | 0           |
|      |        | 0     | This feature is disabled.   |             |
| 1    | CAPOFE | 1     | Capture on CAPn.0 falling edge: a sequence of 1 then 0 on CAPn.0 will cause CR0 to be loaded with the contents of TC. | 0           |
|      |        | 0     | This feature is disabled.   |             |
| 2    | CAPOI  | 1     | Interrupt on CAPn.0 event: a CR0 load due to a CAPn.0 event will generate an interrupt.                               | 0           |
|      |        | 0     | This feature is disabled.   |             |
| 3    | CAP1RE | 1     | Capture on CAPn.1 rising edge: a sequence of 0 then 1 on CAPn.1 will cause CR1 to be loaded with the contents of TC.  | 0           |
|      |        | 0     | This feature is disabled.   |             |
| 4    | CAP1FE | 1     | Capture on CAPn.1 falling edge: a sequence of 1 then 0 on CAPn.1 will cause CR1 to be loaded with the contents of TC. | 0           |
|      |        | 0     | This feature is disabled.   |             |
| 5    | CAP1I  | 1     | Interrupt on CAPn.1 event: a CR1 load due to a CAPn.1 event will generate an interrupt.                               | 0           |
|      |        | 0     | This feature is disabled.   |             |
| 31:6 | -      |       | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.    | NA          |

En la siguiente figura puede verse a modo de ejemplo de funcionamiento el esquema de lo que podría ser el *TIMER/COUNTER0* ó *TIMER/COUNTER1*:

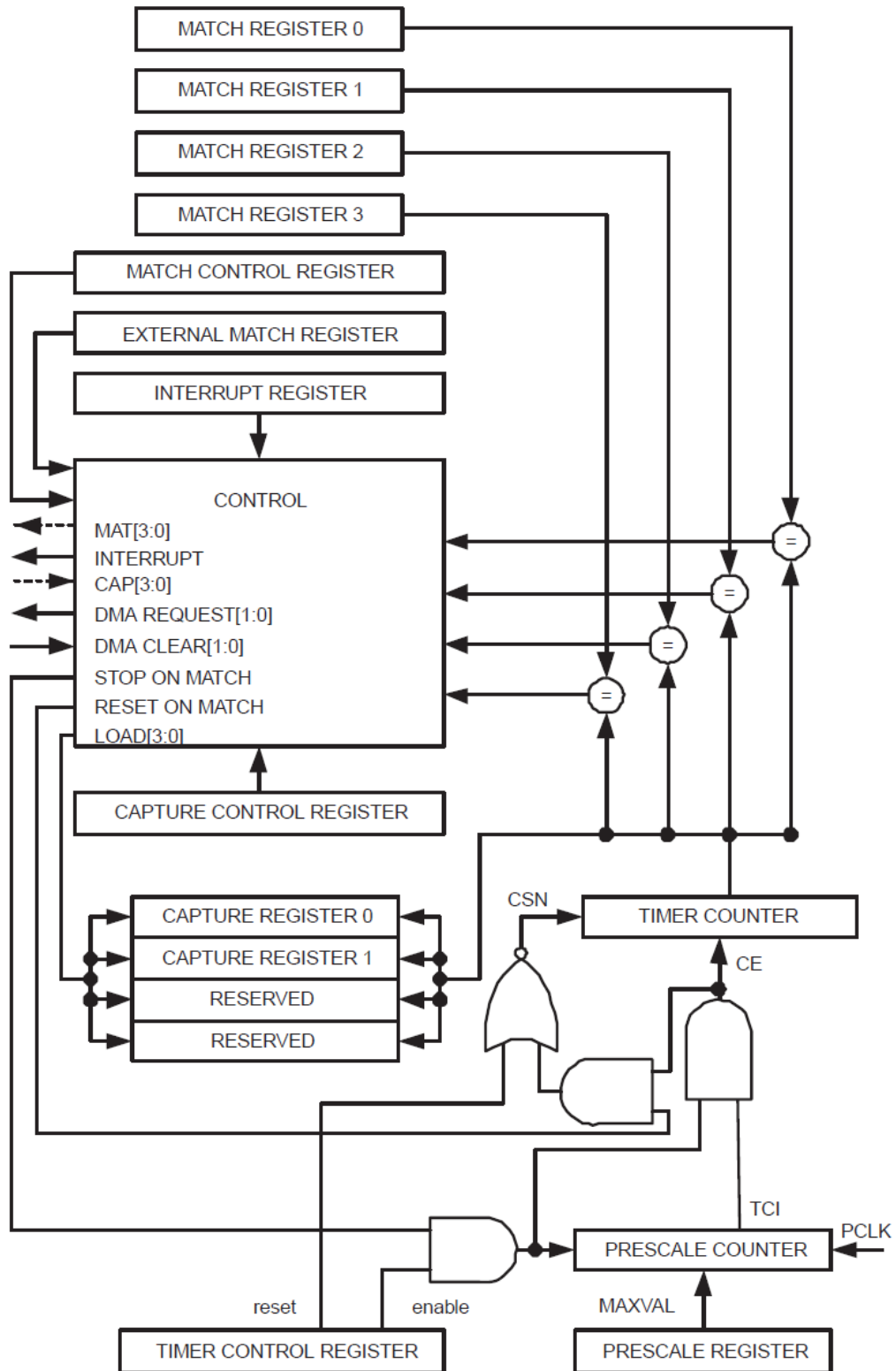


Fig. 32 Arquitectura del Contador/Temporizador 0 ó 1

## 2.2 Timer de interrupción periódica (RIT)

Este timer se utiliza para generar interrupciones periódicas. Se trata de un timer de 32 bits que cuenta a una determinada frecuencia marcada desde el PCLK (*Peripheral Clock Divider* en el apartado II.1.4.1). Cuando la cuenta (registro RICOUNTER) alcanza un valor determinado por el registro RICOMPVAL, se produce una interrupción.

En el registro de configuración del timer (RICTRL) se encuentra el flag de interrupción y los de habilitación del timer, y se establece si el timer se resetea tras una interrupción. En el registro RIMASK se puede escribir '1' en los bits que se deseen obviar (no tener en cuenta) para que dé lugar la comparación. Con ello se puede, por ejemplo, forzar una interrupción.

A continuación se muestra el mapa de registros del RIT:

**Tabla 116 Mapa de memoria de los registros del RIT**

| Name      | Description   | Access | Reset value | Address     |
|-----------|---|--------|-------------|-------------|
| RICOMPVAL | Compare register  | R/W    | 0xFFFF FFFF | 0x400B 0000 |
| RIMASK    | Mask register. This register holds the 32-bit mask value. A '1' written to any bit will force a compare on the corresponding bit of the counter and compare register. | R/W    | 0           | 0x400B 0004 |
| RICTRL    | Control register.   | R/W    | 0xC         | 0x400B 0008 |
| RICOUNTER | 32-bit counter  | R/W    | 0           | 0x400B 000C |

Como ya se ha explicado la función de los registros RICOMPVAL, RIMASK y RICOUNTER, se describe con detalle a continuación el registro RICTRL:

**Tabla 117 Registro de control del RIT (RICTRL, 0x400B0008)**

| Bit  | Symbol   | Value | Description   | Reset value |
|------|----------|-------|---|-------------|
| 0    | RITINT   |       | Interrupt flag  | 0           |
|      |          | 1     | This bit is set to 1 by hardware whenever the counter value equals the masked compare value specified by the contents of RICOMPVAL and RIMASK registers. Writing a 1 to this bit will clear it to 0. Writing a 0 has no effect. |             |
|      |          | 0     | The counter value does not equal the masked compare value.  |             |
| 1    | RITENCLR |       | Timer enable clear  | 0           |
|      |          | 1     | The timer will be cleared to 0 whenever the counter value equals the masked compare value specified by the contents of RICOMPVAL and RIMASK registers. This will occur on the same clock that sets the interrupt flag.          |             |
|      |          | 0     | The timer will not be cleared to 0.   |             |
| 2    | RITENBR  |       | Timer enable for debug  | 1           |
|      |          | 1     | The timer is halted when the processor is halted for debugging.   |             |
|      |          | 0     | Debug has no effect on the timer operation.   |             |
| 3    | RITEN    |       | Timer enable.   | 1           |
|      |          | 1     | Timer enabled.<br><b>Remark:</b> This can be overruled by a debug halt if enabled in bit 2.   |             |
|      |          | 0     | Timer disabled.   |             |
| 31:4 | -        | -     | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.  | NA          |

El campo RITINT es el flag de interrupción. Se pone a 1 por hardware. El software debe escribir '1' para ponerlo a cero.

Si el campo RITENCLR está a '1', cuando se produce una interrupción, el contador (RICOUNTER) se resetea. Si está a '0', la interrupción se produce igualmente, pero el contador no se resetea. Sigue contando hasta el valor máximo, y al alcanzar éste continúa desde 0 sin interrumpir.

El campo RITENBR a '1' hace que el timer se detenga cuando el microcontrolador entre en modo de depuración. Si está a '0', el contador continúa como si nada ocurriese.

El campo RITEN a '1' habilita el timer, siempre que el microcontrolador no entre en modo depuración (RITENBR a '1').

### 2.3 Watchdog

El *watchdog* o “perro guardián” es un temporizador que hay que ir reseteando a lo largo de la ejecución del programa, con el fin de que la cuenta no llegue nunca a desbordar. Esto se hace con el fin de evitar que el sistema quede colgado debido a estados erróneos.

El *watchdog* es un contador decreciente de 32 bits con un preescaler prefijado en el valor 4, cuyo valor mínimo que puede alcanzar es 0xFF (255). Si el contador alcanza este valor pueden ocurrir dos cosas según su modo de funcionamiento (configurable en el registro WDMOD): interrupción de *watchdog* o reseteo de la CPU.

Para resetear el contador es necesario realizar una secuencia de *feed* similar a la que se vió en el caso de los PLLs (a realizar en el registro WDFEED). Cuando el contador es reseteado se carga el valor contenido en el registro WDTC cuyo valor mínimo es 0xFF y continúa la cuenta hacia atrás.

Si se desea leer el valor de cuenta, se puede hacer desde el registro WDTV, pero hay que tener en cuenta que el valor obtenido difiere del valor que la CPU está leyendo en ese momento, debido al tiempo que se tarda en procesar el dato extraído.

En el registro WDCLKSEL puede establecerse la fuente de reloj del *watchdog* entre las posibles: WD\_PCLK (divisor de frecuencia de la CPU para los periféricos, que se tiene que activar en el registro PCLKSELO), RTC e IRC.

Por todo lo explicado puede establecerse unos márgenes de funcionamiento comprendidos entre los siguientes valores:  $T_{WDCLK} \times 265 \times 4 \leq TWD \leq T_{WDCLK} \times 2^{32} \times 4$

De esta manera, el modo de usar el *watchdog* sería el siguiente:

1. Colocar el valor a cargar en el *watchdog*, en el registro WDTC.
2. Configuración del modo de funcionamiento en el registro WDMOD.
3. Habilitación del *watchdog* con secuencia *feed* (escritura en el registro WDFEED).
4. Resetear el *watchdog* mediante sentencia *feed* para que nunca llegue la cuenta al valor mínimo.

Si el *watchdog* se encuentra configurado en modo de reseteo de la CPU, y el contador finaliza, la CPU es reseteada como si de un reset externo se tratase.

Para saber si el reseteo de la CPU ha sido debido al *watchdog*, puede leerse el flag WDTOF del registro WDMOD. Éste flag debe ser puesto a cero (o “limpiado”) mediante software.

El *watchdog*, aunque usa para la cuenta la fuente de reloj configurada en el registro WDCLKSEL, para la actualización de los registros WDMOD y WDTC, utiliza el reloj del APB, lo que hace que los cambios tengan efecto 3 ciclos del WDPCLK después de haber sido cambiados.

**Tabla 118 Mapa de memoria de los registros del *watchdog***

| Name     | Description  | Access | Reset value | Address        |
|----------|--|--------|-------------|----------------|
| WDMOD    | <i>Watchdog</i> mode register. This register contains the basic mode and status of the <i>Watchdog</i> Timer.  | R/W    | 0           | 0x4000<br>0000 |
| WDTC     | <i>Watchdog</i> timer constant register. This register determines the time-out value.  | R/W    | 0xFF        | 0x4000<br>0004 |
| WDFEED   | <i>Watchdog</i> feed sequence register. Writing 0xAA followed by 0x55 to this register reloads the <i>Watchdog</i> timer with the value contained in WDTC. | WO     | NA          | 0x4000<br>0008 |
| WDTV     | <i>Watchdog</i> timer value register. This register reads out the current value of the <i>Watchdog</i> timer.  | RO     | 0xFF        | 0x4000<br>000C |
| WDCLKSEL | <i>Watchdog</i> clock source selection register.   | R/W    | 0           | 0x4000<br>0010 |

Ahora se verá con más detalle cada uno de éstos registros:

- **Registro de modo de *watchdog* (WDMOD, 0x4000 0000):**

Los campos WDEN y WDRESET (bits 0 y 1 respectivamente) controlan el modo de funcionamiento:

**Tabla 119 Posibles modos de funcionamiento del *watchdog***

| WDEN | WDRESET | Mode of operation   |
|------|---------|---|
| 0    | X       | Debug/Operate without the <i>Watchdog</i> running.  |
| 1    | 0       | <i>Watchdog</i> interrupt mode: debug with the <i>Watchdog</i> interrupt but no WDRESET enabled. When this mode is selected, a <i>watchdog</i> counter underflow will set the WDINT flag and the <i>Watchdog</i> interrupt request will be generated.   |
| 1    | 1       | <i>Watchdog</i> reset mode: operate with the <i>Watchdog</i> interrupt and WDRESET enabled. When this mode is selected, a <i>watchdog</i> counter underflow will reset the microcontroller. Although the <i>Watchdog</i> interrupt is also enabled in this case (WDEN = 1) it will not be recognized since the <i>watchdog</i> reset will clear the WDINT flag. |

El bit 2 se corresponde con el flag WDTOF, que es activado por hardware cuando la cuenta llega al final. Debe ser borrado por software.

El bit 3 es el flag de interrupción WDINT, que se pone a '1' cuando entra en la interrupción. Es puesto a '0' tras cualquier reset (bien por el *watchdog*, o bien por la fuente de reset externa). Existe la posibilidad de deshabilitar la interrupción de *watchdog* desde la propia rutina de interrupción, para que no vuelva a interrumpir (para evitar, por ejemplo, el reseteo de la CPU).

Si se produce interrupción por *watchdog*, hay que realizar un RESET para poner a funcionar el sistema de nuevo (borrado del flag WDINT).

- **Registro de cuenta del *watchdog* (WDTC, 0x4000 0004):**

Este registro determina el tiempo máximo de cuenta del *watchdog*. Cada vez que el *watchdog* sea reseteado (con una secuencia de *feed*) su contenido será cargado en el contador (registro WDTV). Su valor mínimo es 0xFF (intervalo mínimo:  $T_{WDCLK} \times 265 \times 4 \leq TWD$ ).

- **Registro de *feed* (WDFEED, 0x4000 0008):**

Para establecer la configuración del *watchdog* (tras configurar el registro WDMOD) y realizar su reseteo, es necesario realizar una secuencia de *feed*. Dicha secuencia se realiza escribiendo en este registro 2 veces. La primera debe escribirse 0xAA, y a continuación debe escribirse (sobrescribiendo lo anterior) 0x55. Si el *watchdog* estuviese habilitado y funcionando, y se accede a cualquier otro registro del mismo, se produciría un error de *feed*, lo que causaría la interrupción de *watchdog* y tal vez el reseteo (dependiendo del modo de funcionamiento) de la CPU. Además mientras se realiza la secuencia de *feed* no se debe acceder a ninguna otra parte, por ello lo más recomendable es deshabilitar las interrupciones, hacer *feed* y volverlas a habilitar. El reset es generado en el 2º ciclo del WDPCLK siguiente al acceso erróneo durante la secuencia de *feed*.

- **Registro de cuenta del *watchdog* (WDTV, 0x4000 000C):**

En este registro de sólo lectura se lleva a cabo la cuenta decreciente del *watchdog*. Cuando se lee de él, el valor obtenido difiere del valor que lee en ese momento la CPU, debido al tiempo que lleva procesar la información obtenida. Cuando se obtiene la información han pasado 6 ciclos de WDPCLK más otros 6 del APB.

- **Registro de selección de fuente de reloj (WDCLKSEL, 0x4000 0010):**

Los dos primeros bits de éste registro (campo WDSSEL) sirven para especificar la fuente de reloj de la cuenta del *watchdog* (WDPCLK), de la siguiente forma:

**Tabla 120 Registro de selección de fuente de reloj del *watchdog* (WDCLKSEL, 0x40000010)**

| Bit  | Symbol | Value | Description   | Reset value |
|------|--------|-------|---|-------------|
| 1:0  | WDSSEL |       | These bits select the clock source for the <i>Watchdog</i> timer as described below.<br><b>Warning:</b> Improper setting of this value may result in incorrect operation of the <i>Watchdog</i> timer, which could adversely affect system operation. If the WDLOCK bit in this register is set, the WDSSEL bit cannot be modified. | 0           |
|      |        | 00    | Selects the Internal RC oscillator ( <i>irc_clk</i> ) as the <i>Watchdog</i> clock source (default).  |             |
|      |        | 01    | Selects the APB peripheral clock ( <i>watchdog pclk</i> ) as the <i>Watchdog</i> clock source.  |             |
|      |        | 10    |   |             |
|      |        | 11    | Reserved, this setting should not be used.  |             |
| 30:2 | -      | -     | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.  | NA          |
| 31   | WDLOCK | 0     | This bit is set to 0 on any reset. It cannot be cleared by software.  | 0           |
|      |        | 1     | Software can set this bit to 1 at any time. Once WDLOCK is set, the bits of this register cannot be modified.   |             |

Una vez se selecciona una fuente de reloj, hay que dejarla fijada escribiendo '1' en el campo WDLOCK (bit 31). Los datos de éste registro no pueden ser modificados hasta que no ocurra un reset (lo que provoca que WDLOCK se ponga a '0').

Todas las posibles fuentes de reloj pueden funcionar en modo de bajo consumo "sleep", pero sólo el IRC puede funcionar en modo "deep sleep", reseteando la CPU y despertando al dispositivo.

## 2.4 Generador de señales PWM

El generador de señales PWM es un periférico que se encarga de generar señales cuadradas, cuyas características (ciclo de trabajo, periodo) son configuradas por el usuario. Para este fin dispone de seis posibles salidas y dos entradas de captura, por si en vez de usar el temporizador interno, se usa otro externo (modo contador). También puede usarse como un timer estándar.

La arquitectura del generador de señales PWM está basada en los temporizadores estándar vistos en el apartado 2.1. Para contar el tiempo, hay un contador interno que cuenta al ritmo establecido por el divisor de frecuencia PCLK y en su control intervienen distintos registros de *match*. Hay dos modos de control: control por un único flanco o control con dos flancos.

En ambos casos el registro de match 0 sirve para controlar el ciclo completo de la señal PWM, de manera que el contador interno es reseteado cuando se produzca un *match*.

En el caso de control mediante un flanco, basta sólo un registro de match más para generar una señal PWM, de manera que en éste se especifique el momento en el que la señal PWM cambiará de nivel. El siguiente cambio se producirá al finalizar el ciclo. Si se controla mediante dos flancos serán necesarios dos registros adicionales de match, de forma que cada uno indica el momento en el que se producirá cada flanco. Cada salida nueva de señal PWM requiere un registro de match en el caso de control con un flanco, o de dos registros de match para el control con dos flancos.



En la siguiente imagen puede verse un esquema del funcionamiento del generador de señales PWM. Puede observarse que, salvo la parte derecha donde están los biestables, el resto del circuito es prácticamente igual al de los temporizadores estándar.

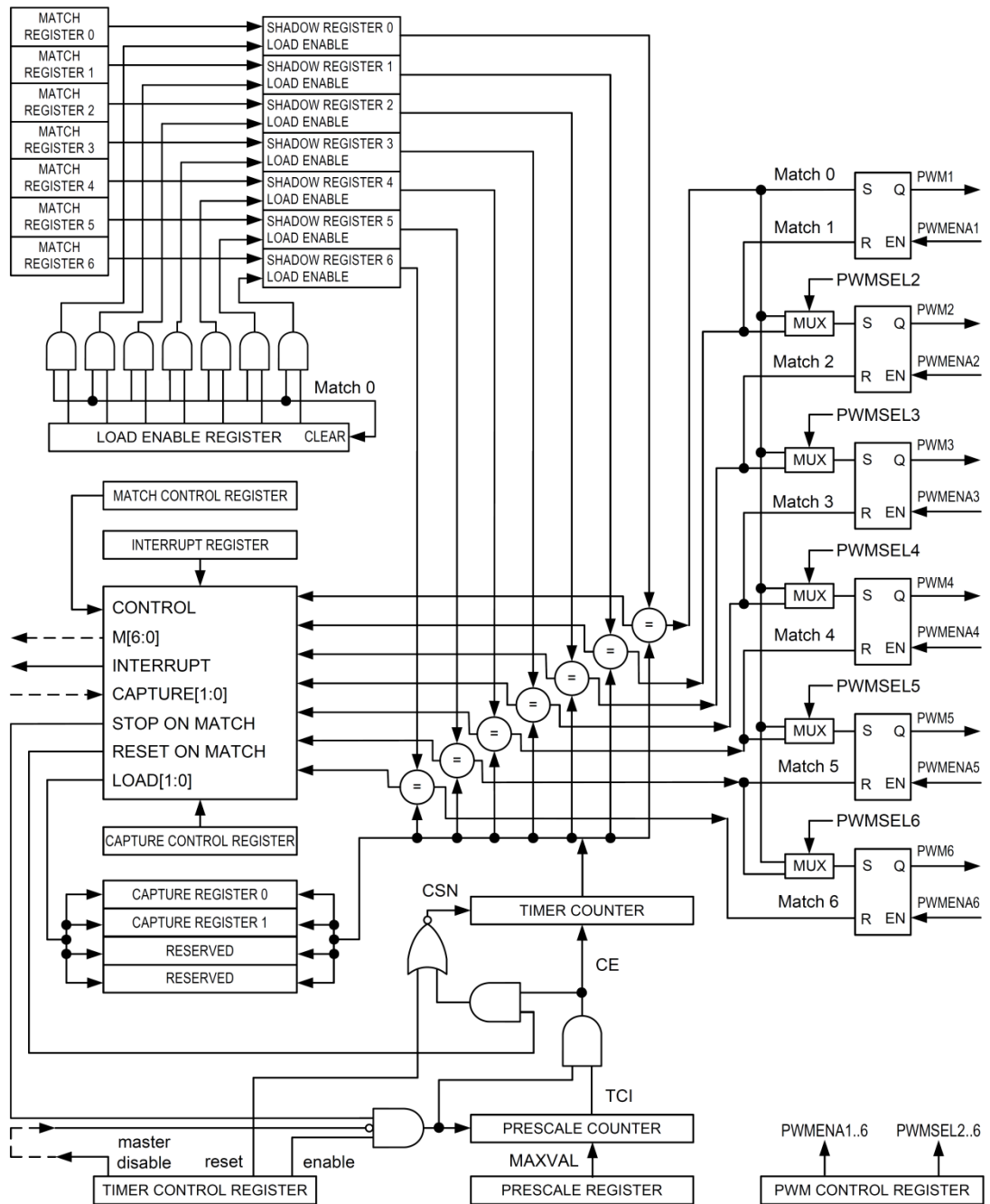


Fig. 33 Arquitectura interna del bloque PWM

En la siguiente tabla se muestran las posibles combinaciones de los 7 registros de *match*, para las diferentes elecciones de puerto de salida:

**Tabla 121 Posibles configuraciones de los registros de match para el uso de las múltiples salidas**

| PWM Channel | Single Edge PWM (PWMSELn = 0) |          | Double Edge PWM (PWMSELn = 1) |          |
|-------------|-------------------------------|----------|-------------------------------|----------|
|             | Set by                        | Reset by | Set by                        | Reset by |
| 1           | Match 0                       | Match 1  | Match 0 <sup>1</sup>          | Match 1  |
| 2           | Match 0                       | Match 2  | Match 1                       | Match 2  |
| 3           | Match 0                       | Match 3  | Match 2 <sup>2</sup>          | Match 3  |
| 4           | Match 0                       | Match 4  | Match 3                       | Match 4  |
| 5           | Match 0                       | Match 5  | Match 4 <sup>2</sup>          | Match 5  |
| 6           | Match 0                       | Match 6  | Match 5                       | Match 6  |

<sup>1</sup> Identical to single edge mode in this case since Match 0 is the neighboring match register. Essentially, PWM1 cannot be a double edged output.

<sup>2</sup> It is generally not advantageous to use PWM channels 3 and 5 for double edge PWM outputs because it would reduce the number of double edge PWM outputs that are possible. Using PWM 2, PWM4, and PWM6 for double edge PWM outputs provides the most pairings.

Normas que hay que tener en cuenta para el control del generador de señales PWM con un único flanco:

1. Todas las salidas PWM se pondrán a nivel alto al comienzo del ciclo PWM, aunque su valor de match sea 0.
2. Cada salida PWM se pondrá a nivel bajo cuando ocurra un match. Si no hay match (porque su valor es superior al de la duración del ciclo), la salida quedará a nivel alto de forma indefinida.

Normas que hay que tener en cuenta para el control del generador de señales PWM con doble flanco:

1. Los valores de match para el siguiente ciclo PWM son usados al final del actual ciclo PWM, salvo en lo expuesto en la norma 3.
2. Un valor de match igual a cero o un ciclo de señal PWM igual a cero, tienen exactamente el mismo efecto, excepto en el caso expuesto en la norma 3.
3. Cuando un valor de match está cambiando, si alguno de los “viejos” valores de match es igual al tamaño de ciclo de la señal PWM, se usa una vez más si ninguno de los nuevos valores es igual a 0 o al tamaño de ciclo.
4. Si un cambio a nivel alto en la salida PWM es requerido al mismo tiempo que un cambio a nivel bajo, el paso a nivel bajo tiene preferencia.
5. Si un valor de match está fuera de rango (es mayor que el tiempo de ciclo), la salida permanecerá en el mismo estado de forma indefinida.

En la siguiente tabla puede verse el mapa de registros del generador de señales PWM:

**Tabla 122 Mapa de registros del generador de señales PWM**

| Generic Name | Description   | Access | Reset Value | PWMn Register        |
|--------------|---|--------|-------------|----------------------|
| IR           | Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending.  | R/W    | 0           | PWM1IR - 0x40018000  |
| TCR          | Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.   | R/W    | 0           | PWM1TCR - 0x40018004 |
| TC           | Timer Counter. The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.  | R/W    | 0           | PWM1TC - 0x40018008  |
| PR           | Prescale Register. The TC is incremented every PR+1 cycles of PCLK.   | R/W    | 0           | PWM1PR - 0x4001800C  |
| PC           | Prescale Counter. The 32-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented. The PC is observable and controllable through the bus interface.   | R/W    | 0           | PWM1PC - 0x40018010  |
| MCR          | Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.   | R/W    | 0           | PWM1MCR - 0x40018014 |
| MR0          | Match Register 0. MR0 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC sets any PWM output that is in single-edge mode, and sets PWM1 if it's in double-edge mode. | R/W    | 0           | PWM1MR0 - 0x40018018 |
| MR1          | Match Register 1. MR1 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC clears PWM1 in either edge mode, and sets PWM2 if it's in double-edge mode.                 | R/W    | 0           | PWM1MR1 - 0x4001801C |
| MR2          | Match Register 2. MR2 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC clears PWM2 in either edge mode, and sets PWM3 if it's in double-edge mode.                 | R/W    | 0           | PWM1MR2 - 0x40018020 |
| MR3          | Match Register 3. MR3 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC clears PWM3 in either edge mode, and sets PWM4 if it's in double-edge mode.                 | R/W    | 0           | PWM1MR3 - 0x40018024 |
| CCR          | Capture Control Register. The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.  | R/W    | 0           | PWM1CCR - 0x40018028 |
| CR0          | Capture Register 0. CR0 is loaded with the value of the TC when there is an event on the CAPn.0 input.  | RO     | 0           | PWM1CR0 - 0x4001802C |
| CR1          | Capture Register 1. See CR0 description.  | RO     | 0           | PWM1CR1 - 0x40018030 |
| CR2          | Capture Register 2. See CR0 description.  | RO     | 0           | PWM1CR2 - 0x40018034 |
| CR3          | Capture Register 3. See CR0 description.  | RO     | 0           | PWM1CR3 - 0x40018038 |
| MR4          | Match Register 4. MR4 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC clears PWM4 in either edge mode, and sets PWM5 if it's in double-edge mode.                 | R/W    | 0           | PWM1MR4 - 0x40018040 |
| MR5          | Match Register 5. MR5 can be enabled in the MCR to reset the TC,  | R/W    | 0           | PWM1MR5 -            |

| Generic Name | Description   | Access | Reset Value | PWMn Register                |
|--------------|---|--------|-------------|------------------------------|
|              | stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC clears PWM5 in either edge mode, and sets PWM6 if it's in double-edge mode.                          |        |             | 0x40018044                   |
| MR6          | Match Register 6. MR6 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC.<br>In addition, a match between this value and the TC clears PWM6 in either edge mode. | R/W    | 0           | PWM1MR6 -<br>0x4001 8048     |
| PCR          | PWM Control Register. Enables PWM outputs and selects PWM channel types as either single edge or double edge controlled.  | R/W    | 0           | PWM1PCR -<br>0x4001 804C     |
| LER          | Load Enable Register. Enables use of new PWM match values.  | R/W    | 0           | PWM1LER -<br>0x4001 8050     |
| CTCR         | Count Control Register. The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.   | R/W    | 0           | PWM1CTCR<br>-<br>0x4001 8070 |

En el registro PWM1IR se encuentran los flags de interrupción. Puede haber tres posibles fuentes de interrupción: interrupciones por *match* (PWMMRn, siendo n=0..6), o por captura (PWMCAPn, siendo n=0,1). En todos los casos se ponen a '1' por hardware (al interrumpir), y deben de ponerse a '0' mediante software, haciendo posible así una nueva interrupción.

Se puede ver la posición de cada campo en la siguiente tabla:

**Tabla 123 Registro de flags de interrupción (PWM1TCR, address 0x4001 8004)**

| Bit   | Symbol            | Description  | Reset Value |
|-------|-------------------|--|-------------|
| 0     | PWMMR0 Interrupt  | Interrupt flag for PWM match channel 0.  | 0           |
| 1     | PWMMR1 Interrupt  | Interrupt flag for PWM match channel 1.  | 0           |
| 2     | PWMMR2 Interrupt  | Interrupt flag for PWM match channel 2.  | 0           |
| 3     | PWMMR3 Interrupt  | Interrupt flag for PWM match channel 3.  | 0           |
| 4     | PWMCAP0 Interrupt | Interrupt flag for capture input 0   | 0           |
| 5     | PWMCAP1 Interrupt | Interrupt flag for capture input 1   | 0           |
| 7:6   |                   | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | NA          |
| 8     | PWMMR4 Interrupt  | Interrupt flag for PWM match channel 4.  | 0           |
| 9     | PWMMR5 Interrupt  | Interrupt flag for PWM match channel 5.  | 0           |
| 10    | PWMMR6 Interrupt  | Interrupt flag for PWM match channel 6.  | 0           |
| 31:11 | -                 | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | NA          |

El registro PWM1TCR sirve para configurar el contador y activar el funcionamiento del generador de señales PWM. Un '1' en el bit 0 (Counter Enable) habilita el contador interno y un '0' lo deshabilita. El bit 1 (Counter Reset) resetea el contador con un '1' en el siguiente flanco de PCLK. El bit 3 (PWM Enable) habilita con un '1' el funcionamiento como generador PWM, ya que si no, el periférico puede usarse como un timer común.

Los registros PWM1TC y PWM1PC son los registros de cuenta y de prescaler, respectivamente. El valor máximo de cuenta de prescaler se guarda en el registro PWM1PR.

En el registro PWM1CTCR se especifica el modo de funcionamiento del generador de señales PWM. Los dos primeros bits (Counter/Timer mode) establecen si va a funcionar como timer ("00"), o contador ("01": en flanco de subida, "10": en flanco de bajada, "11": en ambos flancos de PCLK). Los bits 2 y 3 (Count Input Select) activan los pines PCAP1.0 y PCAP1.1 respectivamente, como entrada de disparo de captura.

El registro PWM1MCR configura el funcionamiento de los registros de *match*, de la misma manera que ocurría en los temporizadores estándar. Cada tres bits se hace referencia a un registro de *match*, desde el match 0 al 6 (los 21 primeros bits, del 0 al 20). El primer bit de cada grupo de tres (PWMMRnI) hace referencia a la posibilidad de interrumpir tras match, el segundo (PWMMRnR) la de resetear el contador y el tercero (PWMMRnS) la de parar el contador; todo ello se activa escribiendo '1', y se desactiva escribiendo '0'.

En los registros de match, desde el PWM1MR0 hasta el PWM1MR6, se escriben los valores que provocarán match cuando sean iguales al del contador.

El registro CCR sirve para configurar los pines de entrada del disparo de captura. Los tres primeros bits hacen referencia al pin PCAP1.0, y los tres siguientes al PCAP1.1. En ambos casos el primer bit (Capture onCAP1.n rising edge) especifica la captura en flanco de subida, el segundo (Capture onCAP1.n falling edge) por flanco de bajada y el tercero (Interrupt on CAP1.n event) en ambos flancos. Si el modo contador es seleccionado en el registro CTCR para alguno de los pines de captura, los tres bits de este registro que hacen referencia a esta entrada deben estar configurados como 000.

Los registros PWM1CR0 y PWM1CR1, almacenan el valor del contador interno (registro PWM1TC) cuando ocurre un evento de captura.

El registro PWM1PCR controla el modo en que se controlará la señal PWM. Desde el bit 2 al 6 (PWMSEL2 .. PWMSEL6) se pueden habilitar con un '1' el control por doble flanco en el canal de salida especificado, o por un único flanco con un '0'. Además con los bits del 9 al 14 (PWMENA1 .. PWMENA.6) se habilitarán las posibles canales de salida, del 1 al 6, respectivamente escribiendo '1' donde corresponda.

Puede verse más claro en la siguiente tabla:

**Tabla 124 Registro de control de salidas PWM (PWM1PCR - address 0x4001 8050)**

| Bit   | Symbol  | Value | Description  | Reset Value |
|-------|---------|-------|--|-------------|
| 1:0   | Unused  |       | Unused, always zero.   | NA          |
| 2     | PWMSEL2 | 1     | Selects double edge controlled mode for the PWM2 output.   | 0           |
|       |         | 0     | Selects single edge controlled mode for PWM2.  |             |
| 3     | PWMSEL3 | 1     | Selects double edge controlled mode for the PWM3 output.   | 0           |
|       |         | 0     | Selects single edge controlled mode for PWM3.  |             |
| 4     | PWMSEL4 | 1     | Selects double edge controlled mode for the PWM4 output.   | 0           |
|       |         | 0     | Selects single edge controlled mode for PWM4.  |             |
| 5     | PWMSEL5 | 1     | Selects double edge controlled mode for the PWM5 output.   | 0           |
|       |         | 0     | Selects single edge controlled mode for PWM5.  |             |
| 6     | PWMSEL6 | 1     | Selects double edge controlled mode for the PWM6 output.   | 0           |
|       |         | 0     | Selects single edge controlled mode for PWM6.  |             |
| 8:7   | -       |       | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | NA          |
| 9     | PWMENA1 | 1     | The PWM1 output enabled.   | 0           |
|       |         | 0     | The PWM1 output disabled.  |             |
| 10    | PWMENA2 | 1     | The PWM2 output enabled.   | 0           |
|       |         | 0     | The PWM2 output disabled.  |             |
| 11    | PWMENA3 | 1     | The PWM3 output enabled.   | 0           |
|       |         | 0     | The PWM3 output disabled.  |             |
| 12    | PWMENA4 | 1     | The PWM4 output enabled.   | 0           |
|       |         | 0     | The PWM4 output disabled.  |             |
| 13    | PWMENA5 | 1     | The PWM5 output enabled.   | 0           |
|       |         | 0     | The PWM5 output disabled.  |             |
| 14    | PWMENA6 | 1     | The PWM6 output enabled.   | 0           |
|       |         | 0     | The PWM6 output disabled.  |             |
| 31:15 | Unused  |       | Unused, always zero.   | NA          |

Para cambiar el valor del pulso de la señal PWM hay que hacerlo mediante el registro PWM1LER. De este registro se usan los siete primeros bits (Enable PWM Match n Latch), correspondiendo cada uno con la habilitación de *match* de cada registro (del 0 al 6). Para actualizar los valores de los registros de *match*, en primer lugar hay que escribir en los mismos el nuevo valor. Después hay que escribir '1' en los campos del registro PWM1LER correspondientes a los registros de *match* modificados. Finalmente en el próximo flanco de PCLK se actualizará la señal PWM.