

Temporizadores

Sistemas con Microprocesadores

Mg.Ing. Esteban Volentini (evolentini@herrera.unt.edu.ar)

<http://www.microprocesadores.unt.edu.ar/procesadores>

Cronograma

Actividad	Inicio	Descripción	Fin
Presentación	19/08	Reglamento de la Materia	✓
Tema 1	19/08	Estructura de las computadoras	✓
Tema 2	26/08	Proyecto con un microcontrolador	✓
Tema 3	30/08	Descripción funcional de microprocesador	✓
Tema 4	13/09	Programación en lenguaje ensamblador	✓
Tema 5	25/09	Descripción general de un microcontrolador	✓
Tema 6	27/09	Estructura general de microcontrolador	✓
Parcial	09/10	Primer examen parcial	✓
Tema 7	14/10	Sistema de Interrupciones	✓
Tema 8	21/10	Entradas y salidas digitales	✓
Tema 9	28/10	Entrada/salida con perifericos	✓
Tema 10	06/11	Temporizadores	←
Proyectos	25/11	Seminarios de Proyectos	
Parcial	04/12	Segundo examen parcial	

Timers

- ▶ Capítulo 32 – User Manual LPC43xx
- ▶ Hay hasta 4 timers, Timer0/1/2/3
- ▶ Cada timer a su vez se puede usar en 4 pins E/S, que se llaman canales.
- ▶ Todos los timers son iguales.
- ▶ Describiremos un Timer.
- ▶ Son configurables para generar comportamientos automáticos
 - ▶ Para liberar carga de procesamiento al procesador
 - ▶ Y que los programas sean más sencillos 😊.
 - ▶ La complejidad en el dispositivo y en la configuración. 😞

Diagrama de Bloques

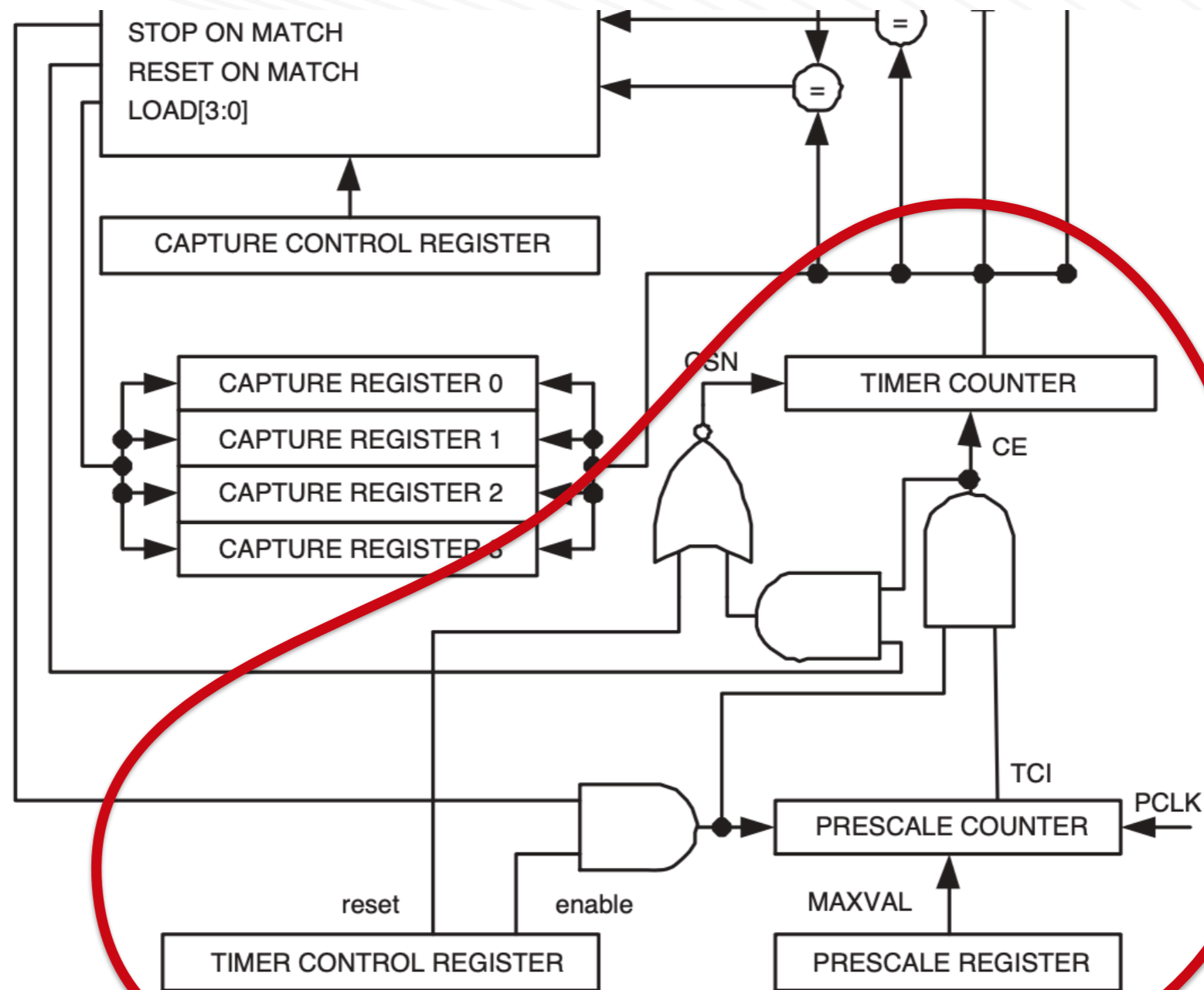


Fig 108. Timer block diagram

TIM - Timer

- ▶ El Núcleo es un contador de 32 bits.
 - ▶ Se llama Timer Counter – TC.
- ▶ El pulso de clock puede ser interno o externo.
 - ▶ Sirve para contar núm de veces suceso externo.
 - ▶ Un registro **TCCR** permite determinar el flanco y la fuente.
- ▶ El clock interno se obtiene del módulo de clock
 - ▶ El clock que se deriva a periféricos se llama PCLK
 - ▶ Esta frecuencia puede dividirse pasando por un divisor programable llamado prescaler - PC.
- ▶ TC en general corre de manera continua, su cuenta máxima llega a FFFFFFFF y vuelve a 0.
 - ▶ Puede haber eventos que lo paren y/o reseteen.
 - ▶ También se lo puede hacer vía programa.

Prescaler

- ▶ El contador es alimentado desde un prescaler (PC)
 - ▶ Este contador tiene un registro que limita su cuenta. Prescale Register (PR) que se carga por Sw.
- ▶ El prescaler cuenta hasta $PR+1$ y emite un pulso al contador.
- ▶ De esta manera la resolución del contador puede ser muy flexible.
 - ▶ Veremos ejemplos más adelante.

Registros de Configuración

Registro	Configura
CTCR	Clock Interno o Externo, flanco fuente externa
PR	Valor Máximo a contar por PC: Prescale Register

faacet

Diagrama de Bloques

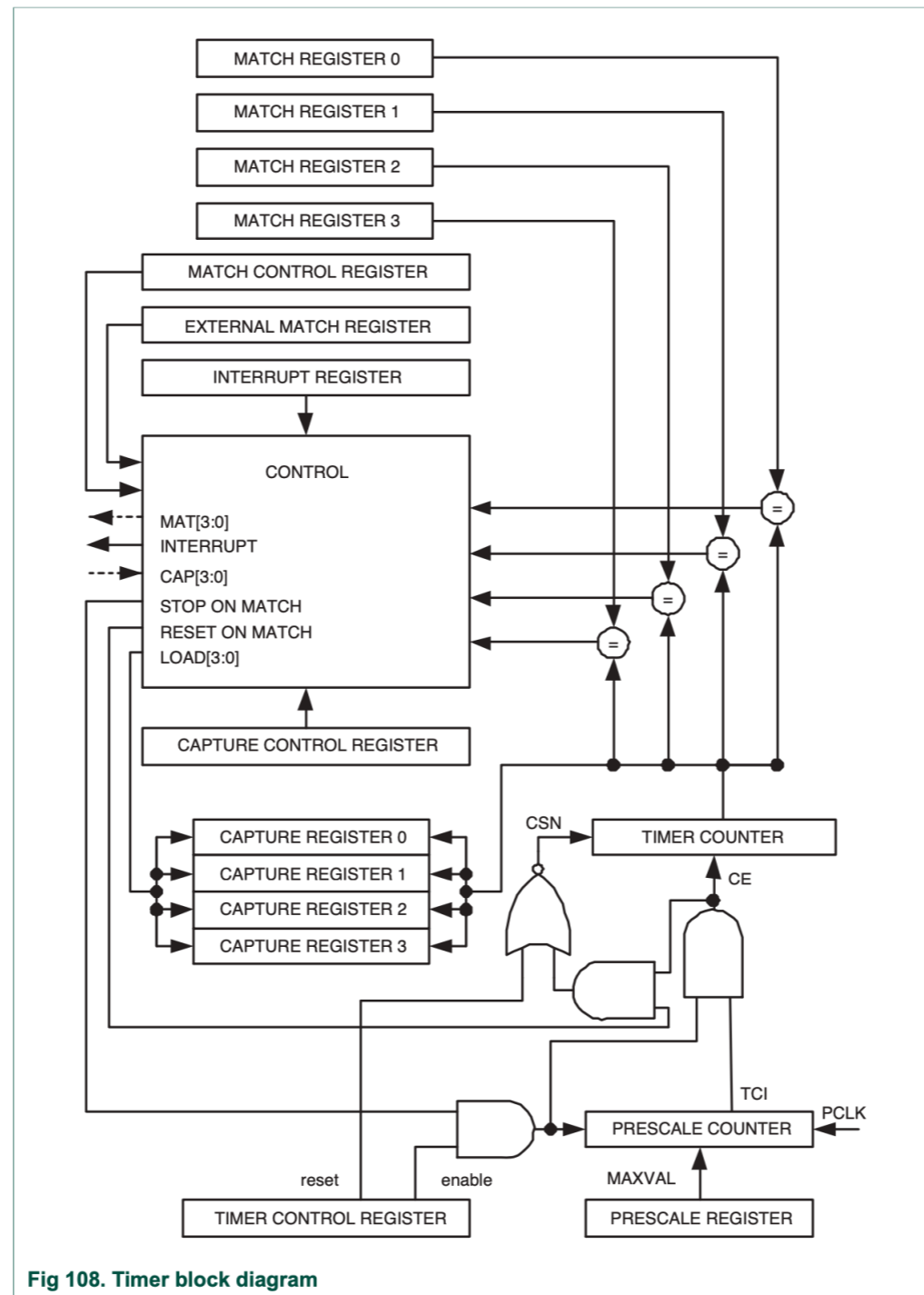
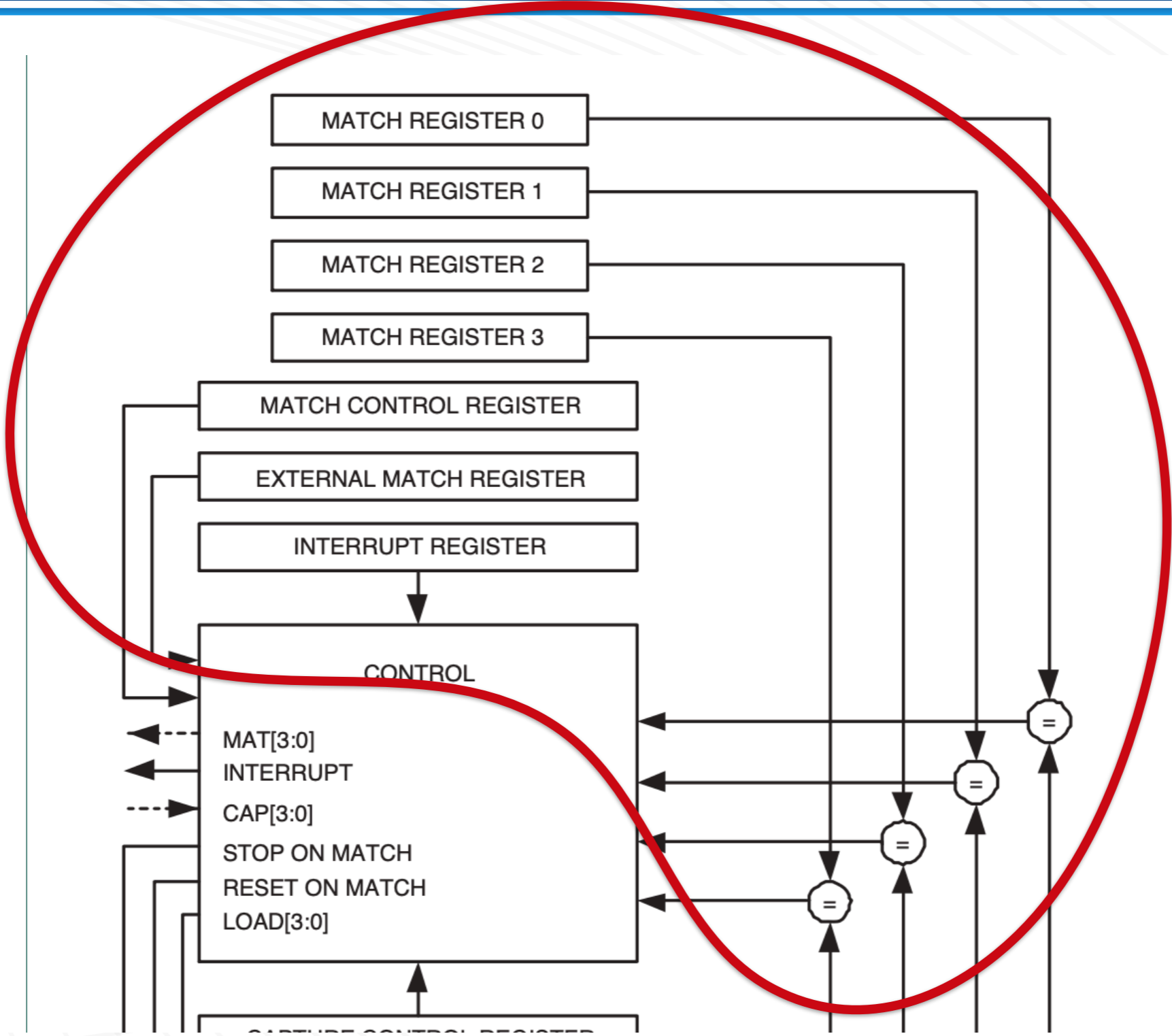


Diagrama de Bloques



Match Registers

- ▶ Hay un total de 4 registros que se cargan por programa: MRO.0/.3
- ▶ Cuando el Contador iguala a un registro se puede producir un evento.
 - ▶ Se llama evento de igualdad o “match”.
 - ▶ Debe configurarse para que ello sea así.
 - ▶ Igual a = “match”. De allí el nombre.
- ▶ Con un match se puede configurar diversas acciones.

Acciones Internas de Igualdad

- ▶ Combinación de las siguientes.
 - ▶ Poner TC=0
 - ▶ Pedir Interrupción.
 - ▶ Parar TC y PC.
- ▶ Se puede configurar las acciones para cada igualdad (match). Los valores se cargan en:
 - ▶ MRO.0
 - ▶ MRO.1
 - ▶ MRO.2
 - ▶ MRO.3
- ▶ La configuración se realiza con MCR
 - ▶ Match Control Register.

Registros de Configuración

Registro	Configura
CTCR	Clock Interno o Externo, flanco, fuente externa
PR	Valor Máximo a contar por PC: Prescale Register
MCR	Si match: Interrupción y/o Reset y/o Stop PC y TC

taoet

Ejemplos de Uso

- ▶ Disparar una interrupción periódicamente.
- ▶ Establecer el período de TC.
 - ▶ ¿Cómo se haría?
- ▶ Cuando $TC = MR0.0/.3$ se genera una o más señales internas $MAT0[3:0]$.

Canales

- ▶ Cada timer puede asociarse a 4 señales de E/S.
 - ▶ Internas o Externas.
 - ▶ Veremos solo las externas comenzando por
 - ▶ Salidas: TO_MAT0/3
- ▶ Hay que configurar los pins para ello.
- ▶ Se pueden generar eventos externos o de salida. Se denomina modo “Output Compare”.
- ▶ Se configura External Match Register.
 - ▶ Set on Match
 - ▶ Reset on Match
 - ▶ Toggle on Match
 - ▶ No generar evento de salida.

Registros de Configuración

Registro	Configura
CTCR	Clock Interno o Externo, flanco, fuente externa
PR	Valor Máximo a contar por PC: Prescale Register
MCR	Si match: Interrupción y/o Reset y/o Stop PC y TC
EMR	Estado de la señal externa de Match, y acción ante match: 0, 1, Toggle, nada.

Aplicaciones “Output Compare”

- ▶ Generar un pulso.
- ▶ Generar una onda cuadrada con un determinado duty factor.
 - ▶ También con duty factor variable.
- ▶ Hay otro timer en MPU más específico para ello, pero nos arreglaremos solo con este.
- ▶ Los registros MRO-3 se pueden cargar
 - ▶ Con GPDMA (configurable) o CPU.
 - ▶ Para emitir señales sin intervención CPU.

Diagrama de Bloques

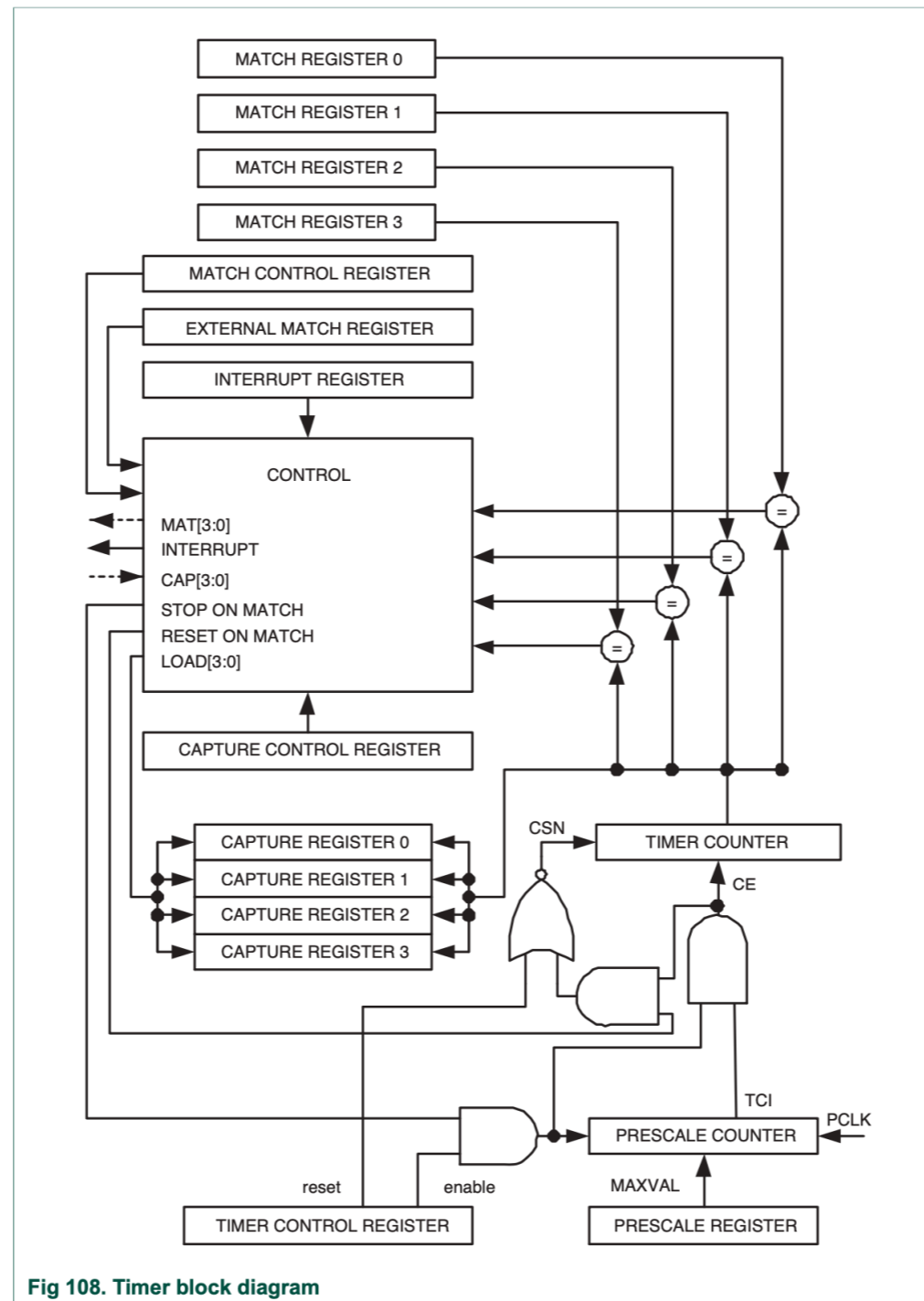
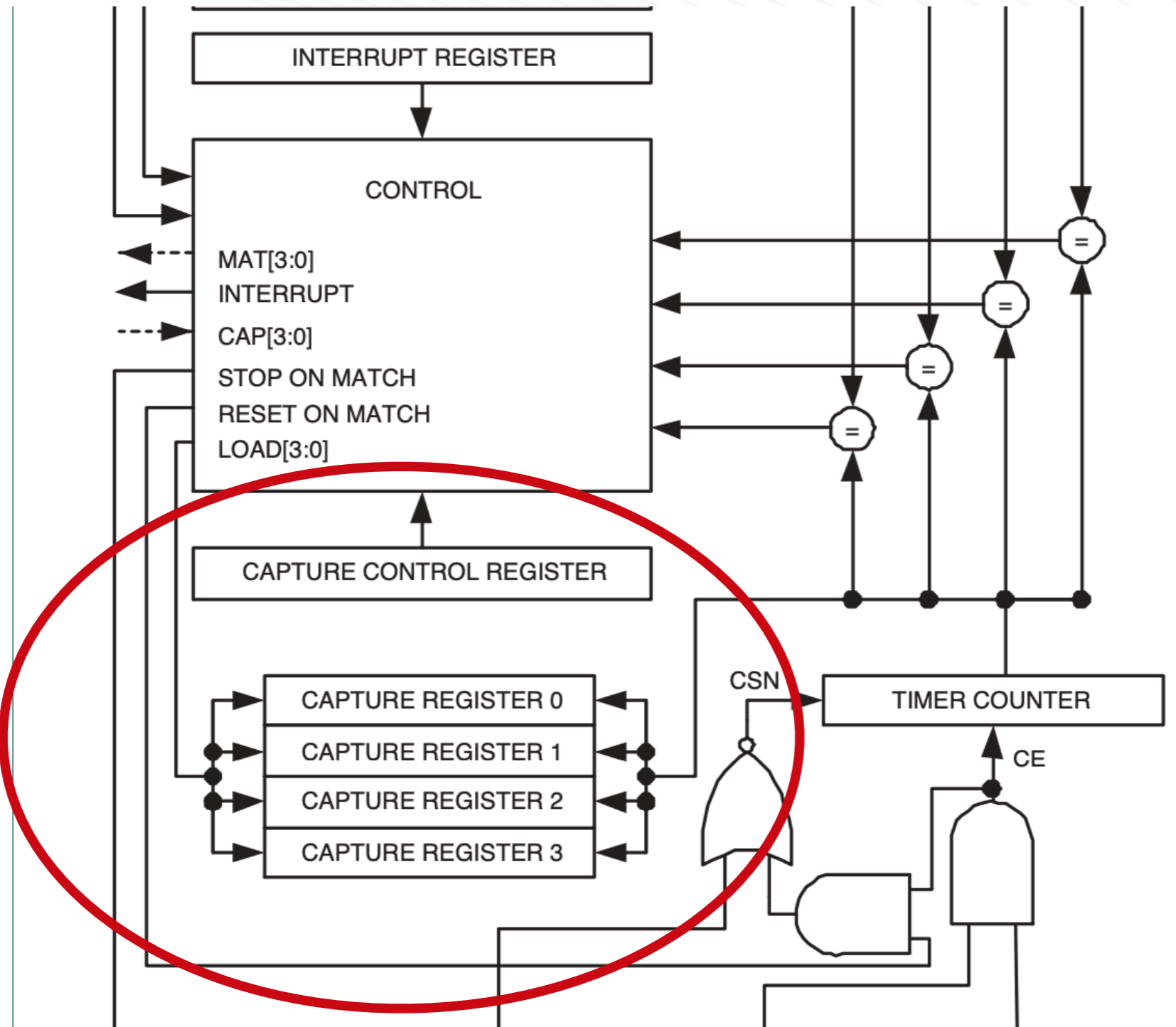


Fig 108. Timer block diagram

Diagrama de Bloques



Eventos Externos de Entrada

- ▶ Se debe configurar los pins para entrada: T0_CAP0/3.
- ▶ Se puede definir una combinación de los siguientes eventos (CCR – capture control Register).
 - ▶ Flanco positivo
 - ▶ Flanco negativo
 - ▶ Pedido de Interrupción.
- ▶ Con el evento se “captura” el valor del TC en un registro.
 - ▶ Hasta 4 registros para realizar capturas.
- ▶ Se dice que se opera en “Input Capture”

Registros de Configuración

Registro	Configura
CTCR	Clock Interno o Externo, flanco, fuente externa
PR	Valor Máximo a contar por PC: Prescale Register
MCR	Si match: Interrupción y/o Reset y/o Stop PC y TC
EMR	Estado de la señal externa de Match, y acción ante match: 0, 1, Toggle, nada.
CCR	reg que captura, f. positivo / negativo / toggle /Int.

Aplicaciones “Input Capture”

- ▶ Medir ancho de un pulso.
 - ▶ Evento: ambos flancos e Interrupción.
 - ▶ Diferencia entre dos capturas.
- ▶ Medir período de una señal.
 - ▶ Evento: flanco positivo e Interrupción.
- ▶ Medir tiempo entre dos señales (RE).
 - ▶ RE – flanco positivo = rising edge.
 - ▶ Evento: flancos positivos de dos entradas distintas e interrupción en la más tardía.

Bajando a Detalle, veremos:

- ▶ Descripción Registros del Timer
- ▶ Ejemplos eventos internos.
- ▶ Ejemplos de Input Capture.
 - ▶ Para medir ancho de un pulso positivo.
- ▶ Ejemplos de Output Compare.
- ▶ LPC43xx User Manual, cap 32, pag 1433
 - ▶ Direcciones reg. de configuración.
 - ▶ Todos ocupan una palabra
 - ▶ Están seguidos para cada TIM.
 - ▶ Comienzan en 0x4008.4000 para TIM0.

Timer Count Control Register

- ▶ CTCR0 – Con reset CTCR0=0. Solo 4 bits usados.
 - ▶ Selecciona fuente de reloj para el contador TC
 - ▶ CINSEL selecciona cual entre 4 fuentes.

Bit	Símbolo	Valor	Acción
1:0	CTMODE	0	Cuenta con PCLK en FP
		1	Flanco Positivo en Fuente Externa
		2	Flanco Negativo en Fuente Externa
		3	Ambos flancos en Fuente Externa
3:2	CINSEL	0	Usar CAPn.0 como fuente de reloj externa
		1	Usar CAPn.1 como fuente de reloj externa
		2	Usar CAPn.2 como fuente de reloj externa
		3	Usar CAPn.3 como fuente de reloj externa

Ej: Duración Máxima TC

- ▶ Supongamos PCLK a 50 MHz.
- ▶ Duración Máxima
 - ▶ TC llega a máximo.
 - ▶ PC llega a máximo con PR=FFFFFFFF.
 - ▶ Período $T_{TC} = 2^{32} / 50 \times 10^6 = 85$ seg.
 - ▶ El máx sería $2^{32} \times 85$ seg !!!
- ▶ El mín sería con PR=1, cuento 2.
 - ▶ $T_{TC} = 2 / 50 \times 10^6 = 40$ nseg
 - ▶ MR1=1 → match1, con reset. Cuenta 2.
 - ▶ El mínimo sería 80 nseg.

Timer Interrupt Register - IR

- ▶ Uno por cada Timer: IR0-3
- ▶ Contiene flags que indican con “1” que se pide interrupción:
 - ▶ 4 por Match (MR0INT_0 – MR3INT_0)
 - ▶ 4 por Captura (CR0INT_0 – CR3INT_0)
 - ▶ Match en Nibble 0, Captura en Nibble 1
- ▶ Registro de 32 bits.
 - ▶ Solo se usan 8 bits.
- ▶ Para resetear una interrupción
 - ▶ Escribir “1” en el bit de IR correspondiente

Registros de Configuración

Registro	Configura
CTCR	Clock Interno o Externo, flanco, fuente externa
PR	Valor Máximo a contar por PC: Prescale Register
MCR	Si match: Interrupción y/o Reset y/o Stop PC y TC
EMR	Estado de la señal externa de Match, y acción ante match: 0, 1, Toggle, nada.
CCR	reg que captura, f. positivo / negativo / toggle /Int.
TIR	Quién pide interrupción (4 capturas y 4 Matchs) Escribir 1 en un flag pone a 0.

En verde los registros que ya se describieron en detalle

TCR – Timer Control Register

- ▶ Solo se usan 2 bits.
- ▶ $b_0 \rightarrow CEN=1$ cuenta, 0 parado.
- ▶ $b_1 \rightarrow CRST=1 \rightarrow TC=0 / PC=0$
 - ▶ Ambas acciones sincrónicas. Ocurren en próximo flanco positivo de PCLK.
 - ▶ Solo puede salir de 0 si $b_0=1$ y $b_1=0$.

Registros de Configuración

Registro	Configura
CTCR	Clock Interno o Externo, flanco, fuente externa
PR	Valor Máximo a contar por PC: Prescale Register
MCR	Si match: Interrupción y/o Reset y/o Stop PC y TC
EMR	Estado de la señal externa de Match, y acción ante match: 0, 1, Toggle, nada.
CCR	reg que captura, f. positivo / negativo / toggle /Int.
TIR	Quién pide interrupción (4 capturas y 4 Matchs) Escribir 1 en un flag pone a 0.
TCR	Parar, resetear, comenzar a contar.

PR – Prescale Register

- ▶ Se carga con el valor máximo a alcanzar por PC.
- ▶ PC genera señal de clock a TC cuando pasa de valor máximo a 0.
 - ▶ Si $PC=PR \Rightarrow PC=0$
 - ▶ T del clock de TC = $PM+1$

Match Control Reg - MCR

- ▶ Contiene 3 bits por cada match reg: MR0.(0-3).
 - ▶ Los primeros 3 para MR0
 - ▶ Los siguientes 3 para MR1 y así...

Bit	Símbolo	Valor	Acción
0	MR0I	1	Interrumpe con Match0
1	MR0R	1	Reset TC con Match0
2	MR0S	1	Stop TC con Match0
3	MR1I	1	Idem para Match1
...			
11	MR3S	1	Stop TC con Match3

Timer Match Registers

- ▶ MRO.0, MRO.1, MRO.2, MRO.3
- ▶ 4 Registros de 32 bits.
- ▶ Se escriben valores para generar match con TC.

Señales Internas del Timer

- ▶ Con cada evento de captura
 - ▶ Hasta 4.
 - ▶ Se generan señales internas CAP0[3:0]
- ▶ Con cada evento de comparación
 - ▶ Hasta 4.
 - ▶ Se generan señales internas MAT0[3:0]
- ▶ Se puede o no asociar a pins externos
 - ▶ con registros de configuración de cada pin.
 - ▶ Se usan como entradas a la MEF del timer

Capture Control Reg - CCR

- ▶ Definir cual registro de captura usar – 0/3.
- ▶ En qué flanco (FP, FN, AF – ambos)
- ▶ Se produce o no una interrupción
- ▶ En reset todos los valores en 0.
- ▶ ¿Cómo se habilita captura en ambos flancos?
- ▶ Se puede configurar pins externos como CAPn.0 a CAPn.4.

Bit	Símbolo	Valor	Acción
0	CAP0RE	1	Flanco Positivo en CAPn.0
1	CAP0FE	1	Flanco Negativo en CAPn.0
2	CAP0I	1	Interrumpe con la captura en CAPn.0
3	CAP1RE	1	Flanco positivo en CAPn.1
...			
11	CAP3I	1	Interrumpe con la captura en CAPn.3

Timer Capture Registers

- ▶ CRO.0, CRO.1, CRO.2, CRO.3
- ▶ 4 Registros de 32 bits.
- ▶ Se guardan valores de TC cuando se producen capturas definidas en CCR
 - ▶ Esta acción es automática con el evento.

External match registers

- ▶ Informan status de cada igualdad.
 - ▶ 4 bits, uno por cada igualdad EM0.0,EM0.1,EM0.2,EM0.3.
 - ▶ Se pueden configurar como salidas.
 - ▶ Cada señal MAT0($i=0/3$) se puede asociar a un único pin de salida.
 - ▶ En ese caso da el valor de la salida.

External Match Register

- ▶ 2 bits para configurar la funcionalidad de las señales de igualdad MAT0[3:0].
 - ▶ No hacer nada.
 - ▶ MAT0.0=0 (pin salida en 0, con match).
 - ▶ MAT0.0=1. (idem anterior pero en 1).
 - ▶ MAT0.0= \neg MAT0.0
 - ▶ También para MAT0.1, 2 y 3.

Descripción EMR

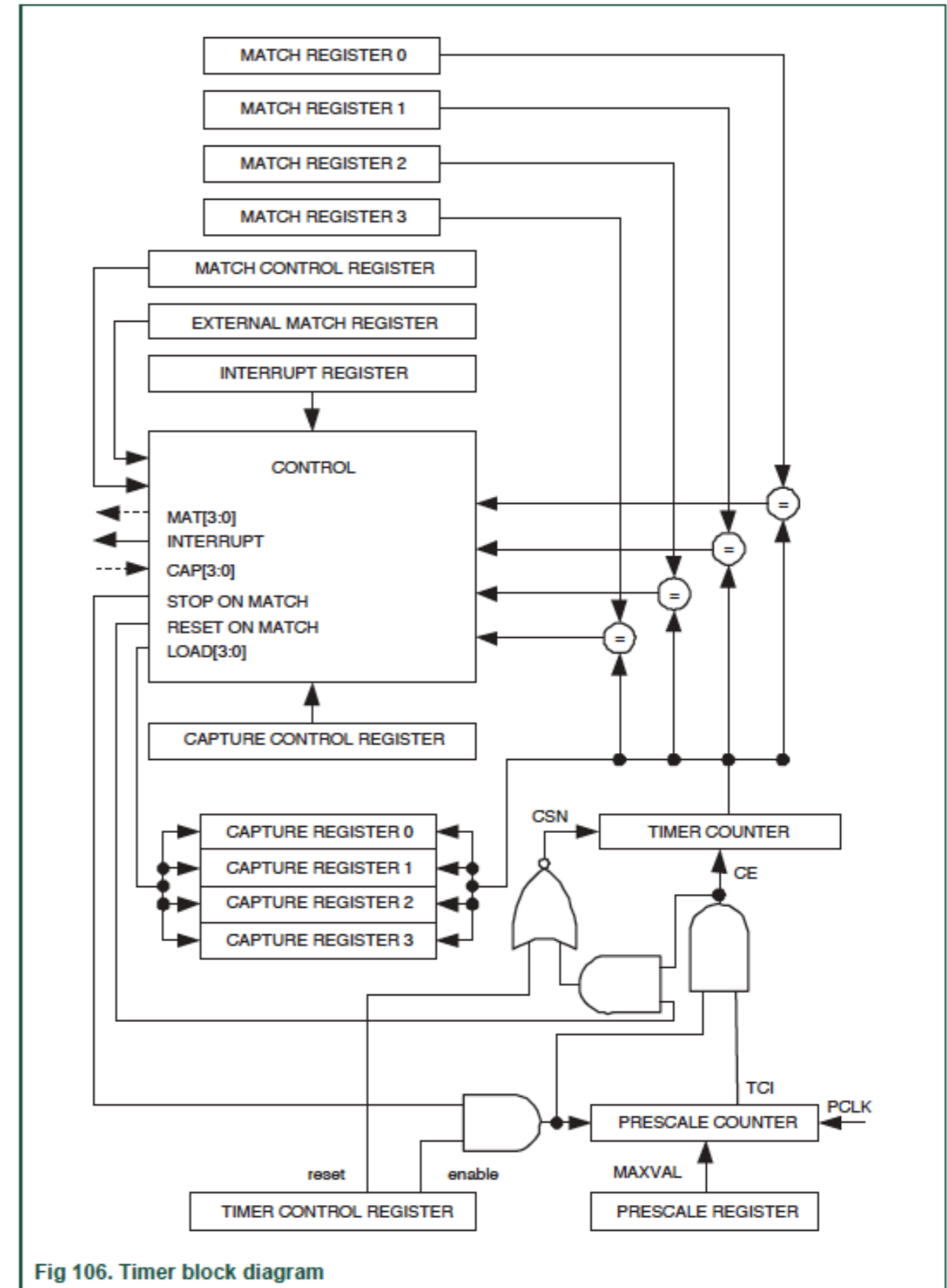
Bit	Símbolo	Valor	Acción
0	EM0	*(5:4)	Estado de MATn.0
1	EM1	*(7:6)	Estado de MATn.1
2	EM2	*(9:8)	Estado de MATn.2
3	EM3	*(11:10)	Estado de MATn.3
5:4	EMC0		External Match Control
		0	No hacer nada
		1	MATn.0 = 0
		2	MATn.0 = 1
		3	Invertir MATn.0
7:6	EMC1		Idem EMC0 para MATn.1
9:8	EMC2		Idem MATn.2
11:10	EMC3		Idem MATn.3
31:12	Reservados		no escribir 1's en reservados

Registros de Configuración

Registro	Configura
CTCR	Clock Interno o Externo, flanco, fuente externa
PR	Valor Máximo a contar por PC: Prescale Register
MCR	Si match: Interrupción y/o Reset y/o Stop PC y TC
EMR	Estado de la señal externa de Match, y acción ante match: 0, 1, Toggle, nada.
CCR	reg que captura, f. positivo / negativo / toggle /Int.
TIR	Quién pide interrupción (4 capturas y 4 Matchs) Escribir 1 en un flag pone a 0.
TCR	Parar, resetear, comenzar a contar.
CR0-3	Cuatro registros de captura de 32 bits.
MR0-3	Cuatro registros de Match de 32 bits.

Esquema Timer

- ▶ TC cuenta a partir de PC.
- ▶ PR valor máximo de PC.
- ▶ TCR para enable y reset.
- ▶ TC se compara con hasta 4 MR0/3.
- ▶ El resultado va al control.
- ▶ Las señales de Captura entran al Control.
- ▶ Se captura en hasta 4 registros CR0/3
- ▶ MCR y CCR da señales al control
- ▶ Para eventos, capturas e interrupción.



Resolución y Rango.

- ▶ Resolución: tiempo mínimo que se puede contar.
 - ▶ Está dada por $PR+1$, pues establece TTC.
 - ▶ Es variable, dependiendo del valor máximo de PR.
- ▶ Rango: tiempo máximo a contar.
 - ▶ $Rango = Resolución \cdot n$ ($n-1 =$ cuenta máx. en ciclos T)
 - ▶ Cuando $TC = n-1$ se establece un evento de igualdad.
 - ▶ Un registro de igualdad debe cargarse con el valor $n-1$.
 - ▶ Se configura este MATCH para resetear TC. El contador vuelve a cero automáticamente al contar n pulsos.
 - ▶ Queda configurado como “free running”. El periodo de TC es igual al Rango.

Ej. Esperar t por Interrupción

1. Ajustar rango y precisión según especificaciones del problema.
 2. Inicializar Timer:
 1. Poner en 0 y parar contador **TCR**
 2. Escribir en **MRO.1** el valor n-1.
 3. $MCR0[5:3]=3$ (interrumpe con Match1, resetea TC, no detiene el contador)
 4. Arrancar contador y quitar Reset **TCR**
- ▶ **$TIRO[1]=1$ en ISR (servicio Interrupción)**
 - ▶ **Solo funciona por Int.**
 - ▶ A menos que se enmascare la interrupción en NVIC y se haga polling en IR
 - ▶ **Funciona de manera similar a SysTick. Se puede esperar n Interrupciones.**

Ej. Medir Periodo de Señal Externa

1. Revisar rango y precisión Ok, configurar “free running”
 2. Configurar el canal para evento por flanco ascendente, captura en CRO e interrupción. “**CCRO=5**”
 3. Siempre antes de configurar se para y resetea y luego se arranca el contador de nuevo.
 4. **E1**: Primer flanco.
 - Escribir $TIR[4]=1$: quita INT. Leer CRO y guardar en V1.
 5. **E2**: Segundo flanco:
 - $TIR[4]=1$
 - Leer CRO.0 en R2, restar de V1, multiplicar por precisión y obtener t y guardarlo en V1.
 - ¿Puede ser que $R2 < R1$? ¿Los tiempos son exactos?
- ▶ **E1 y E2 se pueden detectar solo por Int.**
- ▶ A menos que se enmascare la interrupción en NVIC y se haga polling en IR.

Ej. Cálculo Ancho de un Pulso

1. Ajustar rango y resolución de acuerdo a especificaciones, free running sin INT.
 2. Configurar el canal para evento por ambos flancos. **CCRO**
 - Habilitar Interrupción y captura en CRO.0.
 - **Mide un pulso positivo o negativo, el primero.**
 - **Con cada evento no olvidar poner IR[4] a 0.**
 3. E1: Primer flanco: leer CRO.0 en R2, guardar en V.
 4. E2: Segundo flanco:
 - Leer CRO.0 en R1, $R0=R1-V$, multiplicar por precisión y obtener T. Guardar en V.
- E1 y E2 se detectan por polling o Int.
- Por Polling enmascarando INT en NVIC

Ej. Cálculo Ancho de un Pulso

- ▶ Estructura de Datos:
 - ▶ NE – eventos transcurridos.
 - ▶ 0 – espera primer evento.
 - ▶ 1 – espera segundo evento.
 - ▶ VALOR va a contener Ancho del Pulso (AP)
- ▶ Rango y Resolución – suponer PCLK=40 MHz:
 - ▶ R=8 mseg (resolución). AP<10 seg (Rango 2 veces AP).
- ▶ Usaremos Interrupciones.
- ▶ Configuración Prescale Register (PR).
 - ▶ $(PR+1) \times T = 8 \text{ ms}$. $T = 1/40 = 25 \text{ ns}$. $PR = 320 - 1 \rightarrow 319$.
- ▶ Cuenta mínima en TC = $(10 \text{ seg} / 8 \text{ ms}) = 1250$
 - ▶ MR1 = 2500 - 1 provocará un reset de TC.

Configuración Pin

- ▶ Pin de entrada T0_CAP0 (Canal 0, T0) corresponde a P1_13.
 - ▶ Mode=4 (es FUNC4=T0_CAP0 pag 397)
 - ▶ EPD=0 (sin pull down).
 - ▶ EPUN=1 (sin pull up).
 - ▶ EHS=0 (pendiente media, no se usa).
 - ▶ EZI=1 (habilitar entrada).
 - ▶ ZIF=0 (filtro ruido habilitado)
- ▶ SPSP1_13= 0x00000054
- ▶ Repasar configuración pag 411 manual.
- ▶ El pin se configuró como entrada.

Configuración Timer0, Canal0

- ▶ Parar y limpiar el contador: TCR=2
- ▶ CCR[2:0] (capture control reg)=7
 - ▶ Ambos flancos e interrumpen.
 - ▶ No tocar resto de los bits de ningún registro.
- ▶ Estado Interrupción: TIR.
 - ▶ ISR debe escribir 1 en TIR[4] para ponerlo a 0
 - ▶ No tocar el resto de los bits
- ▶ Count Timer Control Register: CTCR
 - ▶ Dejar en valor por reset: 0.
 - ▶ Usa clock interno PCLK para contar.
- ▶ MR0.1=2499, PR=319.
- ▶ MCR[5:3] (match control register)=2
 - ▶ Reset on MR1, sin interrupción, habilitar cuenta.

Configuración

- ▶ Configurar en NVIC
 - ▶ INT #12 para TIMER0, todos sus eventos.
 - ▶ Vector Int. Habilitar Int. Prioridad. Quitar posible int pendiente en NVIC.
- ▶ $TIR[4] \leftarrow 1$ (timer int. register), $TIR[1] \leftarrow 1$.
 - ▶ Limpia Int. del dispositivo por las dudas.
- ▶ $NE=0$ (inicializar núm eventos)
- ▶ VALOR – variable que contendrá la duración en T
- ▶ Habilitar Int en general (en CPU y $PRIMASK=0$)
- ▶ Quitar Reset y Habilitar contador:
 - ▶ $TCR=1$
- ▶ Escribir programa de Inicialización como ejercicio en la casa.

Seudocódigo ISR

1. Asegurar que $TIRO[4]=1$ – sino otro motivo...
2. Limpiar pedido INT. $TIRO[4] \leftarrow 1$
3. $R1=CR0.0$ (capture register)
4. $NE=NE+1$ (nuevo evento)
5. Si $NE=2 \rightarrow$ ir a 8
6. $VALOR=R1$ // caso primer flanco
7. Retornar
8. $RO=(R1-VALOR)$ //caso: segundo flanco.
9. $CCR0[2:0]=0$ // no captura ni interrumpe.
10. Si $RO>0$ ir a 12 //Contador no volvió a 0.
11. $RO=RO+MRO+1$ //corregir OVF
12. $VALOR=RO*8$
13. Retornar

¿Cómo sabe P.Principal que tiene el resultado?

¿Por qué no se deja los valores en registros?

Antes de retornar se podría revisar $TIRO \neq 0$ en ese caso hay otro motivo también pendiente de tratamiento. ¿Y si no se revisa qué pasa?

Ej. Cálculo Ancho de un Pulso

```
//Rutina de interrupción (INTTIM0 debe estar escrito en Vector int)
INTTIM0  LDR    R1,=TIR0      //dirección reg. Estado de interrupción.
        LDR    R2,[R1]
        ANDS   R3,R2,#0x10   //Interrumpe por CAP0?
        BEQ    otro         //caso no desarrollado, si se usara otro canal
        STR    R3,[R1]      //poner a cero flag interrupción
        LDR    R2,=CR0_0
        LDR    R1,[R2]^-    //leer CR0 en R1
        LDR    R2,=NE
        LDR    R3,[R2]      //leer NE
        ADD    R3,R3,#01
        STR    R3,[R2]      //NE=NE+1
        LDR    R2,=VALOR
        CMP    R3,#01
        BNE    E2           //¿es el primer o segundo flanco?
        MOVE   R0,R1        //caso primer flanco
        B      FIN
E2       LDR    R3,=CCR0     //registro de control de captura,
        LDR    R4,[R3]
        AND    R4,#0xFFFFF8 //no interrumpe ni captura
        STR    R4,[R3]
        LDR    R0,[R2]      //Cargar el valor
        SUBS   R0,R1,R0     //Calcular la diferencia
        ITT    MI          //revisar si no pasa contador por su máximo
        LDRMI  R3,=MR0_1   //en ese caso sumar cuenta maxima mas 1.
        LDRMI  R3,[R3]^-
        ADDMI  R0,R3
        ADDMI  R0,#1
        LSL    R0,#3        //multiplico por 8 (rango es 8 msg)
FIN      STR    R0,[R2]     //guardar valor
        BX    LR
```

Para pensar

- ▶ ¿La demora de una interrupción influye sobre el tiempo medido?
- ▶ Si el pulso es muy breve en relación al tiempo entre dos interrupciones
 - ▶ ¿Hay problema?
- ▶ Si se quiere medir ancho solo de pulsos positivos
 - ▶ ¿Cómo se hace?

Output Compare: Generar Pulso

- ▶ Se debe configurar para que un pin esté asociado al canal como salida.
- ▶ Debe haber 2 eventos: E1 y E2
 - ▶ Uno para generar primer flanco y otro para segundo flanco.
- ▶ Inicialización para generar primer evento (E1).
 - ▶ Parar el contador y reset.
 - ▶ Escribir comienzo del pulso en MR0.0
 - ▶ Configurar EMR para toggle on compare, MCR para Interrupt.
 - ▶ Lanzar el contador.
- ▶ Detección Evento E1: Interrupción
 - ▶ $TIR[0] \leftarrow 1$ – para poner flag de INT en 0.
 - ▶ Escribir nuevo valor a comparar en MR0 (ancho del pulso).
- ▶ Si E2 debe producirse muy pronto:
 - ▶ Puede no haber tiempo para una interrupción.
 - ▶ En este caso se puede usar DMA para cambiar MR0.

Generación de un Pulso.

1. Revisar resolución r , en función ancho del pulso P – ej 1%.
2. Configurar Canal.
 1. Dar valor inicial a MRO.0, cualquiera, ej: $MRO=100$
 2. Toggle on Compare en MRO.0
 - t se expresa en cantidad de pulsos de Timer.
 3. Lanzar contador.
 4. Cuando iguala a MRO.0, calcular $MRO.0=(MRO.0 + P)$
 - P – duración del pulso en ciclos T de TC.
 5. Escribir MRO.0.
 6. En el siguiente evento, parar el contador.
 7. Puede ser por Polling o Interrupciones.
- ¿Si t mayor que máximo M de TC?
 - El máximo es $0xFFFF.FFFF$.
 - Es muy grande para que ocurra.

Ej. Configurar Señal Periódica.

1. Parar y Limpiar Contador.
2. Configurar T del contador (PR).
 - ▶ En función de una cierta resolución .
 - ▶ Arrancar Contador.
3. Configurar MRO.0 para duración pulso positivo,
 - ▶ Reset salida en EMR, interrupción en MCR.
4. Con primer evento configurar MRO de acuerdo a T de la señal periódica.
 - MRO.0 = duración en ciclos del período de la señal.
 - Set on MRO - en EMR.
 - Reset counter on Match de MRO e Interrupción - MCR
5. Vaya a 3.
 - ▶ Cambiando MRO con cada INT de MRO se logra una señal en PWM.

Generar Señal Periódica

- 1. Evento E0: Reset on match.**
 - Int: Configurar MR0.0 para nuevo evento E1.
 - MR0_R=MR0.0 (respaldo)
 - MR0.0=PERIODO. Interrumpir en MR0. Set on match. Reset Timer.
- 2. Evento E1: Set on Match.**
 - Interrupción por MR0. MR0 cambia la salida a 1.
 - Configurar MR0 para evento E0.
 - MR0.0=MR0_R. Interrumpir en MR0. Reset on Match.
- 3. Señal Periódica: Se repiten 1 y 2...**
- 4. Comienza recién en el segundo T. ¿Por qué?**
- 5. Si usáramos MR0.1 en lugar de MR0.0 en E1, sería más fácil? No se puede.**