

Descripción del sistema de memoria
y de entrada/salida en la placa
S3CEV40.

El contenido de este documento ha sido publicado originalmente bajo la licencia:
Creative Commons License (<http://creativecommons.org/licenses/by-nc-sa/3.0>)
Prácticas de Estructura de Computadores empleando un MCU ARM by Luis Piñuel y
Christian Tenllado is licensed under a Creative Commons Attribution-NonCommercial-
ShareAlike 3.0 Unported License.

1.1. Sistema de de memoria

La figura 1.1 representa de forma esquemática el subsistema de memoria de la placa empleada en el laboratorio¹. El funcionamiento del sistema de memoria depende en realidad de tres componentes de nuestro kit de laboratorio, el procesador (ARM7TDMI), el sistema en chip en el que se integra (S3C44B0X) y la placa sobre la que se coloca este último (S3CEV40). A continuación describiremos las principales características de cada uno de sus componentes.

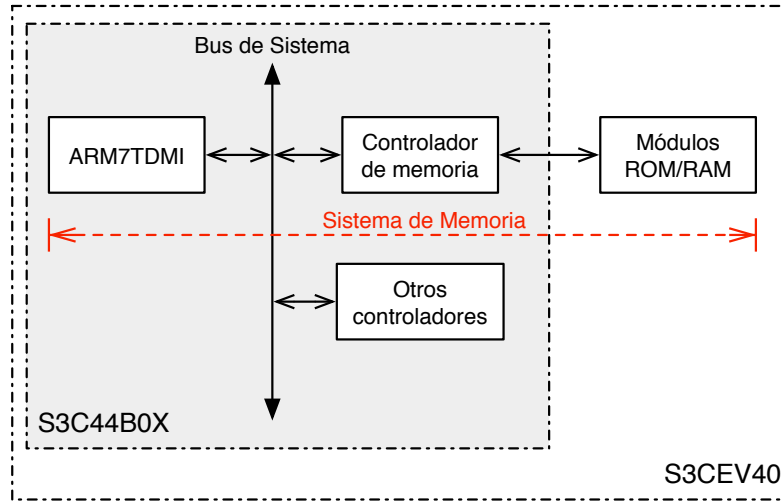


Figura 1.1: Sistema de memoria de la placa de laboratorio (Embtest S3CEV40).

1.1.1. El ARM7TDMI

El bus de direcciones del ARM7TDMI es de 32 bits, por lo que es capaz de direccionar potencialmente un espacio total de 4GB de memoria. Es preciso señalar además que este espacio de direcciones es compartido por el sistema de E/S (*E/S localizada en memoria*), por lo que las direcciones pueden referirse tanto a posiciones de memoria principal como a elementos internos de los controladores de E/S (registros o memoria local). En ambos casos el ARM7TDMI realiza el acceso del mismo modo y es responsabilidad del programador saber a qué dispositivo está asociada cada dirección. Esta asignación de rangos de direcciones a dispositivos suele recibir el nombre de *mapa de memoria del sistema*.

1.1.2. El controlador de memoria

El controlador de memoria es el responsable de actuar de interfaz entre los módulos de memoria externos (ROM o RAM) y el bus del sistema. Es fácil adivinar que el *mapa de memoria del sistema* viene determinado en gran medida por las características y configuración de este elemento del sistema de memoria.

¹Aunque en el *SoC* S3C44B0X dispone de un *Write Buffer* y una *Cache* de 8K, por defecto ambos están desactivados.

El comportamiento del controlador de memoria podría resumirse del siguiente modo. Cuando el procesador pone una dirección en el bus:

- Si ésta se encuentra dentro del rango del controlador, él se encarga de generar las señales necesarias para realizar el acceso al módulo de memoria que corresponda.
- Si, por el contrario, la dirección queda fuera de su rango de competencia, el controlador se inhibe, ya que supuestamente es responsabilidad de algún controlador de E/S atender a dicho acceso (lectura/escritura de un registro o memoria local).

Si el acceso a memoria falla (p.ej. se realiza el acceso a un banco para el que no hay asignado módulo de memoria), es responsabilidad del controlador detectar el error y generar la correspondiente excepción de *Abort*.

El controlador de memoria del S3C44B0X

En la placa del laboratorio (Embest S3CEV40), tanto el procesador ARM7TDMI como el controlador de memoria, así como otros controladores y dispositivos de E/S, se encuentran integrados dentro un mismo chip, el *System on Chip* S3C44B0X de Samsung.

El controlador de memoria del S3C44B0X reduce el espacio de direcciones efectivo a 256MB y lo divide en 8 fragmentos independientes, denominados *bancos*, tal y como ilustra la figura 1.2. Cada banco puede ser asignado a un chip de memoria externo distinto (módulo), aunque sólo los dos últimos bancos admiten cualquier tipo de memoria (SRAM, SRAM o SDRAM). Cuando se accede a una dirección dentro del rango de uno de estos bancos, además de las señales necesarias para realizar el acceso al módulo al que está asociado, el controlador de memoria activa una señal nGCS². Estas señales se utilizan externamente para seleccionar/activar el chip correspondiente.

De todo el espacio de memoria, la parte alta del primer banco, correspondiente al rango 0x01C00000-0x01FFFFFF, está reservada para los puertos de E/S de los controladores integrados dentro del S3C44B0X. Como hemos mencionado previamente, cuando el procesador realiza un acceso dentro este rango el controlador de memoria se inhibe (i.e. no genera señal alguna).

1.1.3. Los módulos de memoria

La placa que se emplea en el laboratorio (Embest S3CEV40) dispone de dos módulos de distinto tipo:

- Memoria ROM Flash de 1Mx16bits, ubicada en el banco 0 del S3C44B0X y cuyo rango de direcciones es [0x00000000-0x001FFFFFFF].
- Memoria SDRAM de 4Mx16bits, ubicada en el banco 6 del S3C44B0X y cuyo rango de direcciones es [0x0C000000-0x0C7FFFFFFF].

Como podemos ver, el espacio de 256MB permitido por el controlador de memoria se reduce en nuestro caso a 10MB, 2 de ROM Flash y 8 de SDRAM.

²Por convenio, las señales cuyo nombre comienza con “n” se activan en baja.

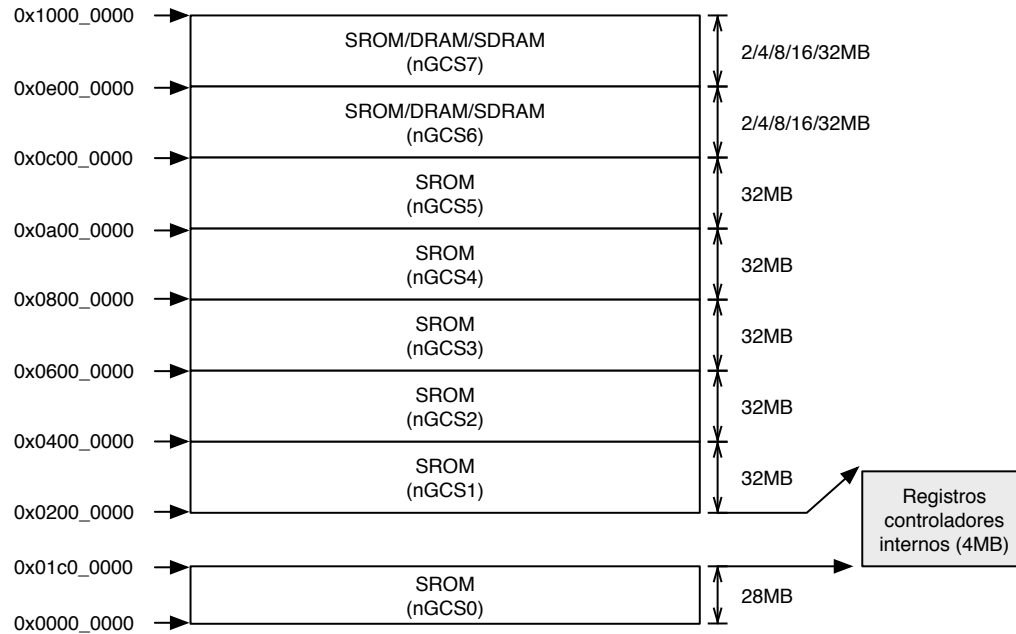


Figura 1.2: Mapa de memoria del S3C44B0X (SROM se refiere tanto a ROM como a SRAM).

1.2. Sistema de Entrada/Salida

La figura 1.3 representa de forma esquemática el sistema de entrada/salida de la placa empleada en el laboratorio. Sus principales componentes son el procesador ARM7TDMI, el controlador de interrupciones, los controladores de E/S – tanto internos como externos al Samsung S3C44B0X –, y la lógica de selección, que se sirve de las señales generadas por el controlador de memoria para seleccionar el chip externo con el que se desea trabajar. A continuación describiremos brevemente las principales características de cada uno de sus componentes.

1.2.1. Controladores y dispositivos de E/S

Salvo contadas excepciones, los dispositivos de E/S se gestionan a través de controladores específicos de E/S. Estos son los que actúan de interfaz entre el dispositivo y el bus de sistema y el controlador de interrupciones. Para comunicarse con el dispositivo, el procesador lee y escribe en los registros internos de estos controladores, a los que nos referiremos habitualmente como puertos de E/S.

Selección de dispositivos/controladores

Como hemos mencionado previamente, el procesador ARM7TDMI emplea E/S localizada en memoria y por lo tanto no es capaz de distinguir si una dirección se corresponde con un puerto de E/S o con una posición memoria. Anteriormente vimos que el controlador

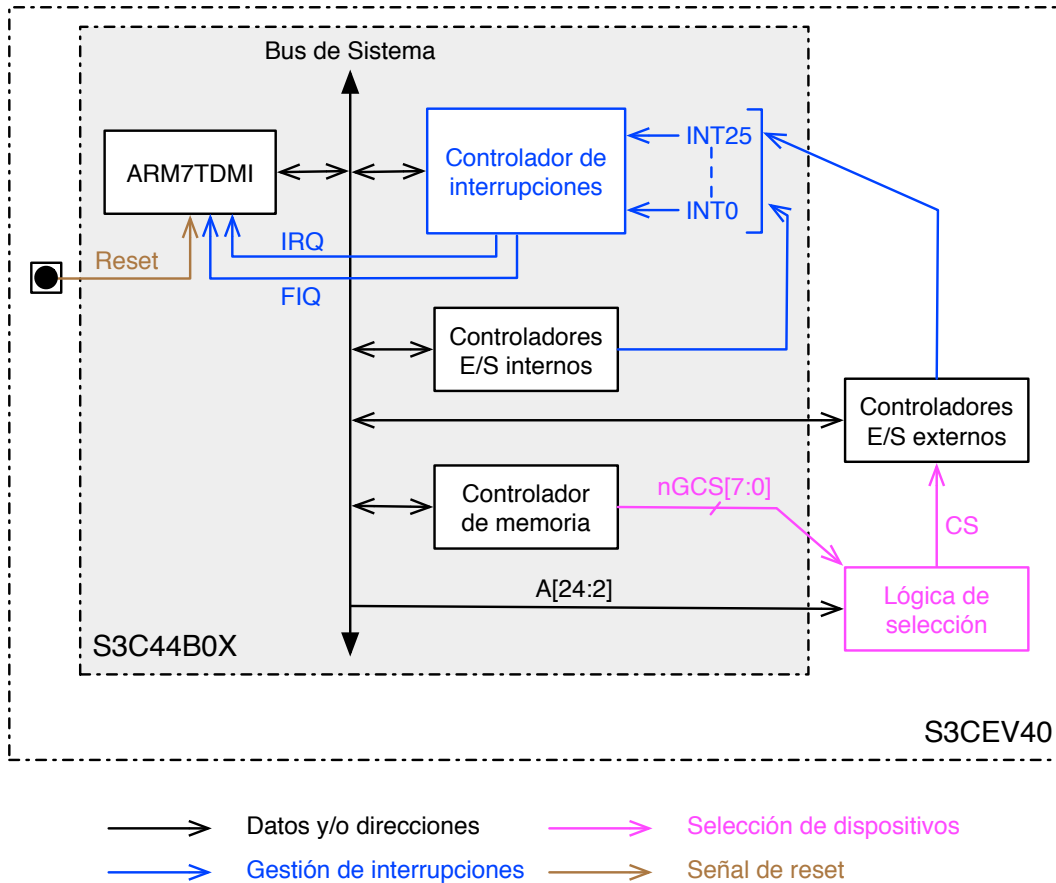


Figura 1.3: Sistema de E/S de la placa de laboratorio.

de memoria juega un papel esencial a la hora de discriminar entre ambos: asigna un rango de direcciones a los registros correspondientes a los controladores internos al S3C44B0X (de E/S, de memoria, de interrupciones, ...) y el resto de los 256MB del espacio de direcciones lo asigna a memoria y se encarga de su gestión, generando las señales necesarias para realizar la comunicación con el módulo y activando la señal nGCS correspondiente al banco (nGCS0-nGCS7).

Además de los controladores internos, existen en el sistema controladores de E/S externos. Los puertos de E/S correspondientes a estos controladores externos al S3C44B0X se ubican fuera del rango de direcciones correspondiente a los controladores internos y requieren de una lógica adicional para su selección (ver figura 1.3). Obviamente, no pueden asignarse puertos de E/S a bancos que tengan asignados módulos de memoria. Esta lógica de selección es la encargada de activar la señal de *Chip Select* (CS) correspondiente al dispositivo que se pretende acceder en función del valor de las señales nGCS0-7 y de los bits de direcciones del bus del sistema.

Dispositivos/controladores de la práctica

En la práctica 2, como el objetivo es familiarizarse con la E/S, vamos a emplear los controladores y dispositivos más sencillos de los que disponemos en la placa del laboratorio:

- *Internos al S3C44B0X:*
 - Controlador de pines de E/S de propósito general (GPIO)
 - Temporizadores con *modulación de ancho de pulso* (PWM Timer)
- *Externos al S3C44B0X:*
 - Display 8-segmentos
 - Diodos LEDs
 - Pulsadores

En las siguientes secciones, ampliaremos la información sobre cada uno de ellos, salvo sobre el controlador de interrupciones, que se tratará en el guión de la práctica.

1.3. Controlador de pines de E/S

El controlador de pines de E/S de propósito general (GPIO) mantiene el control de 71 pines multifuncionales, es decir, cada uno de estos pines puede ser utilizado para más de una función distinta. El GPIO permite configurar la funcionalidad escogida para cada uno de estos pines.

Los 71 pines están divididos en 7 grupos, a los que denominamos puertos, que son:

- 2 grupos de 9 bits de E/S (Puertos E y F)
- 2 grupos de 8 bits de E/S (Puertos D y G)
- 1 grupo de 16 bits de E/S (Puerto C)
- 1 grupo de 11 bits de E/S (Puerto B)
- 1 grupo de 10 bits de E/S (Puerto A)

Cada puerto es gestionado mediante 2-4 registros. El número concreto depende del puerto. Al realizar un *Reset* estos registros se cargan a un valor seguro para no dañar ni el sistema ni el hardware que pueda haber conectado a estos pines. En las prácticas vamos a utilizar sólo algunos de los pines de E/S de los puertos B y G. La tabla 1.1 recoge una relación de sus registros de datos y control, que describiremos en detalle a continuación. Para obtener una descripción completa es preciso consultar el manual del S3C44B0X [um-].

1.3.1. Puerto B

El puerto B tiene 11 bits, que admiten dos funcionalidades: podemos utilizar estos pines como pines de salida (para escribir un valor directamente en ellos) o podemos conectarlos a algunas señales generadas por el controlador de memoria (*i.e.* hacemos que estas señales salgan por estos pines fuera del S3C44B0X). La configuración del puerto se realiza a través de dos registros del GPIO:

Tabla 1.1: Registros de los puertos B y G

Registro	Dirección	R/W	Valor de Reset
PCONB	0x01D20008	R/W	0x7FF
PDATB	0x01D2000C	R/W	Undef
PCONG	0x01D20040	R/W	0x00
PDATG	0x01D20044	R/W	Undef
PUPG	0x01D20048	R/W	0x00
EXTINT	0x01D20050	R/W	0x00000000
EXTINTPND	0x01D20054	R/W	0x00

- *PCONB*, registro de control que selecciona la funcionalidad de cada uno de los pines. Su descripción está en la tabla 1.2.
- *PDATB*, registro de datos que permite escribir los bits que se pretenden sacar por el puerto (sólo útil para aquellos pines configurados como salida).

Tabla 1.2: Configuración de los pines del puerto B.

PCONB	Bit	Descripción	
PB10	[10]	0 = Salida	1 = nGCS5
PB9	[9]	0 = Salida	1 = nGCS4
PB8	[8]	0 = Salida	1 = nGCS3
PB7	[7]	0 = Salida	1 = nGCS2
PB6	[6]	0 = Salida	1 = nGCS1
PB5	[5]	0 = Salida	1 = nWBE3/nBE3/DQM3
PB4	[4]	0 = Salida	1 = nWBE2/nBE2/DQM2
PB3	[3]	0 = Salida	1 = nSRAS/nCAS3
PB2	[2]	0 = Salida	1 = nSCAS/nCAS2
PB1	[1]	0 = Salida	1 = SCLK
PB0	[0]	0 = Salida	1 = SCKE

En las prácticas utilizaremos dos pines del puerto B configurados como salida (los pines 9 y 10, como veremos más adelante), para encender o apagar dos LEDs conectados a ellos.

1.3.2. Puerto G

El puerto G tiene ocho bits que pueden configurarse de cuatro formas diferentes. La configuración del puerto se realiza mediante tres registros del GPIO:

- *PCONG*, registro de control que permite seleccionar la funcionalidad de cada pin, tal y como se describe en la tabla 1.3.
- *PDATG*, registro de datos que permite escribir o leer del puerto.
- *PUPG*, registro de configuración que permite activar (bit a '0') o no (bit a '1') una resistencia de *pull-up*³ por cada pin.

Tabla 1.3: Configuración de los pines del puerto G.

PCONG	Bits	Descripción	
PG7	15:14	00 = Entrada 10 = IISLRCK	01 = Salida 11 = EINT7
PG6	13:12	00 = Entrada 10 = IISDO	01 = Salida 11 = EINT6
PG5	11:10	00 = Entrada 10 = IISDI	01 = Salida 11 = EINT5
PG4	9:8	00 = Entrada 10 = IISCLK	01 = Salida 11 = EINT4
PG3	7:6	00 = Entrada 10 = nRTS0	01 = Salida 11 = EINT3
PG2	5:4	00 = Entrada 10 = nCTS0	01 = Salida 11 = EINT2
PG1	3:2	00 = Entrada 10 = VD5	01 = Salida 11 = EINT1
PG0	1:0	00 = Entrada 10 = VD4	01 = Salida 11 = EINT0

Como vemos en la tabla, podemos utilizar los bits del puerto G como pines de entrada o de salida o conectarlos a algunas señales internas del sistema. Alternativamente podemos configurar los pines del puerto para que activen las líneas **EINT*** de petición de interrupción. Esto permite que algún hardware externo conectado a estos pines pueda generar una interrupción por estas líneas. Cuando opera de esta forma este puerto admite diversas opciones para desencadenar la interrupción. La interrupción puede generarse cuando se detecte un nivel de voltaje en el pin o cuando se detecte un determinado flanco. Esto se configura mediante un cuarto registro del GPIO, *EXTINT* cuyo uso se describe en la tabla 1.4.

El controlador GPIO permite, además, descomponer en cuatro la línea EINT4/5/6/7 del controlador de interrupciones, cuyo funcionamiento se explicará más adelante. La línea EINT4/5/6/7 procedente del controlador de interrupciones se activa cuando alguna de estas cuatro líneas genera una interrupción (en realidad se hace una OR). El registro *EXTINTPND* del puerto G permite saber cuál de las cuatro líneas es la que ha generado la interrupción como describe la tabla 1.5.

Es preciso resaltar que para borrar la petición de interrupción por una de estas líneas de interrupción, la rutina de servicio debe borrar el bit correspondiente de este registro (escribiendo un '1', no un '0') antes de borrar el bit de INTPND del controlador de interrupciones.

Más adelante se explicará cómo utilizar el puerto G para leer los pulsadores o para generar una interrupción cuando se detecte que se ha pulsado uno de ellos.

³Una resistencia de *pull-up* pone la entrada a uno cuando está desconectada, evitando que se quede en un valor indefinido. De forma similar, una resistencia de *pull-down* pone la entrada a cero cuando está desconectada.

Tabla 1.4: Registro *EXTINT*.

EXTINT	Bit	Descripción
EINT7	[30:28]	Establece el método de señalización de EINT7. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos
EINT6	[26:24]	Establece el método de señalización de EINT6. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos
EINT5	[22:20]	Establece el método de señalización de EINT5. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos
EINT4	[18:16]	Establece el método de señalización de EINT4. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos
EINT3	[14:12]	Establece el método de señalización de EINT3. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos
EINT2	[10:8]	Establece el método de señalización de EINT2. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos
EINT1	[6:4]	Establece el método de señalización de EINT1. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos
EINT0	[2:0]	Establece el método de señalización de EINT0. 000 = Interrupción por nivel bajo 001 = Interrupción por nivel alto 01x = Disparado por flanco de bajada 10x = Disparado por flanco de subida 11x = Disparado por ambos flancos

1.4. LEDs y pulsadores

En la placa S3CEV40 hay dos pulsadores y dos LEDs conectados al sistema S3C44BOX como indica la figura 1.4.

Los LEDs se conectan a los pines 9 y 10 del puerto B. Como puede apreciarse en la figura 1.4, su ánodo se conecta a la alimentación (Vdd) por lo que para iluminar estos LEDs es necesario configurar los pines 9 y 10 como sólo salida (ver tabla 1.2) y escribir en ellos un '0' mediante el registro *PDATB*.

Los dos pulsadores se conectan a los pines 6 y 7 del puerto G. Configurando estos pines como entrada leeremos un 0 en el bit correspondiente del registro *PDATG* cuando el pulsador está pulsado. Para asegurarnos que los pines tomen el valor 1 lógico (Vdd) cuando el pulsador no está pulsado es necesario activar las resistencias de pull-up de estos pines mediante el registro *PUPG*. Sin embargo, cuando se produzca una pulsación en realidad no se producirá una bajada limpia de la tensión de Vdd a 0, sino que la tensión oscilará

Tabla 1.5: Significado de los bits del registro *EXTINTPND*.

EXTINTPND	Bit	Descripción
EXTINTPND3	[3]	Si EINT7 se activa, EXINTPND3 se pone a 1, y INTPND[21] se pone a 1.
EXTINTPND2	[2]	Si EINT6 se activa, EXINTPND2 se pone a 1, y INTPND[21] se pone a 1.
EXTINTPND1	[1]	Si EINT5 se activa, EXINTPND1 se pone a 1, y INTPND[21] se pone a 1.
EXTINTPND0	[0]	Si EINT4 se activa, EXINTPND0 se pone a 1, y INTPND[21] se pone a 1.

(entre Vdd y 0) hasta que finalmente se estabilice en 0. Solemos decir que se producen rebotes de la señal. Esto puede hacer que detectemos varias pulsaciones cuando sólo se ha producido una. Para filtrar estos rebotes debemos esperar un tiempo cuando detectemos una pulsación. El tiempo a esperar puede ajustarse empíricamente, pero 100ms suele ser suficiente. Podemos hacer esta espera con una función Delay que básicamente consiste en dos bucles anidados que no hacen nada (cuerpo del bucle interno vacío).

Alternativamente, podemos hacer llegar los pines 6 y 7 del puerto G, conectados a los pulsadores, a las líneas EXINT6 y EXINT7 del controlador de interrupciones estableciendo el valor adecuado de *PCONG* (ver tabla 1.3). Para generar interrupciones mediante estos pulsadores, es necesario, además, activar las resistencias de *pull-up* de estos pines mediante el registro *PUPG* (para mantener constante el valor de entrada cuando el pin está al aire), o de lo contrario el comportamiento sería imprevisible. Asimismo es necesario configurar el tipo de interrupción (nivel alto/bajo, flanco subida/bajada/ambos) mediante el registro *EXTINT* del puerto.

1.5. Temporizadores

El S3C44B0X tiene 6 temporizadores de 16 bits, conectados según indica la figura 1.5. Cada uno puede ser configurado para operar por interrupciones o por DMA. Los temporizadores 0, 1, 2, 3 y 4 tienen asociada la capacidad de generar una onda cuadrada de la frecuencia con la que haya sido programado el temporizador, modulada por ancho de pulsos (PWM, pulse width modulation), son las señales TOUT* de la figura 1.5. Estas señales pueden sacarse por algún pin para transmitir una señal por modulación de ancho de pulsos, controlar un motor, etc. Esta funcionalidad no está disponible en el temporizador 5, que sólo puede utilizarse para temporización interna y no tiene pin de salida. Los registros utilizados para el manejo de los temporizadores se recogen en la tabla 1.6.

Funcionamiento de los temporizadores

Los temporizadores son contadores descendentes que pueden ser inicializados con un determinado valor. Una vez configurados e inicializados, cada ciclo de reloj interno se decrementan. Cuando llegan a 0 generan una interrupción que podemos utilizar para realizar algunas tareas de forma periódica.

Como podemos ver en la figura 1.5, cada par de temporizadores (0-1, 2-3, 4-5) comparte un módulo de pre-escalado y un divisor de frecuencia. El divisor de los 4 primeros temporizadores tiene cinco señales divididas distintas: 1/2, 1/4, 1/8, 1/16 y 1/32. Los temporizadores 4 y 5 tienen un divisor con cuatro señales de reloj divididas: 1/2, 1/4, 1/8 y

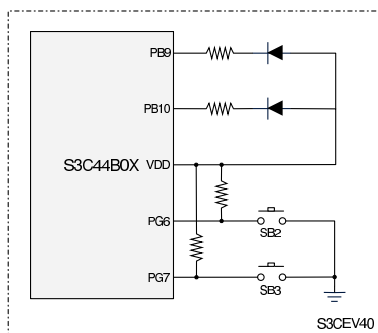


Figura 1.4: Esquema de conexión de pulsadores y LEDs de la placa Embest S3CEV40.

1/16, y una entrada adicional $TCLK/EXTCLK$, que sirve para realizar la cuenta de una señal distinta de la señal de reloj (por ejemplo para contar pulsos externos al sistema, obtenidos de un pin). Como vemos, los divisores se alimentan con la salida de los módulos de pre-escalado con lo que la frecuencia de cuenta se construye con un proceso de división en dos etapas, como detallaremos posteriormente.

Cada temporizador (excepto el 5) tiene un par de registros asociados, $TCNTn$ y $TCMPn$. El registro de cuenta ($TCNTn$) se inicializa a un determinado valor y se decrementa en cada ciclo mientras el temporizador esté activo. El registro de comparación ($TCMPn$) se inicializa a otro valor y como su propio nombre indica se emplea para comparar con el registro de cuenta. El resultado de esta comparación se emplea para controlar la señal de salida. Al comienzo de cada cuenta esta señal tiene un valor 0, cuando el contador alcanza el valor del registro de comparación la señal toma el valor 1. De esta forma podemos controlar exactamente la anchura de los pulsos (PWM), es decir el número de ciclos que debe estar a 0 y a 1.

Los temporizadores pueden funcionar en dos modos: *auto-reload* y *one-shot*. En modo *auto-reload* cuando el temporizador llega a cero, su valor inicial se vuelve a cargar automáticamente y se comienza una nueva cuenta atrás. Esto sirve para generar de forma muy precisa interrupciones periódicas. En el modo *one-shot*, cuando el contador llega a cero se detiene la cuenta.

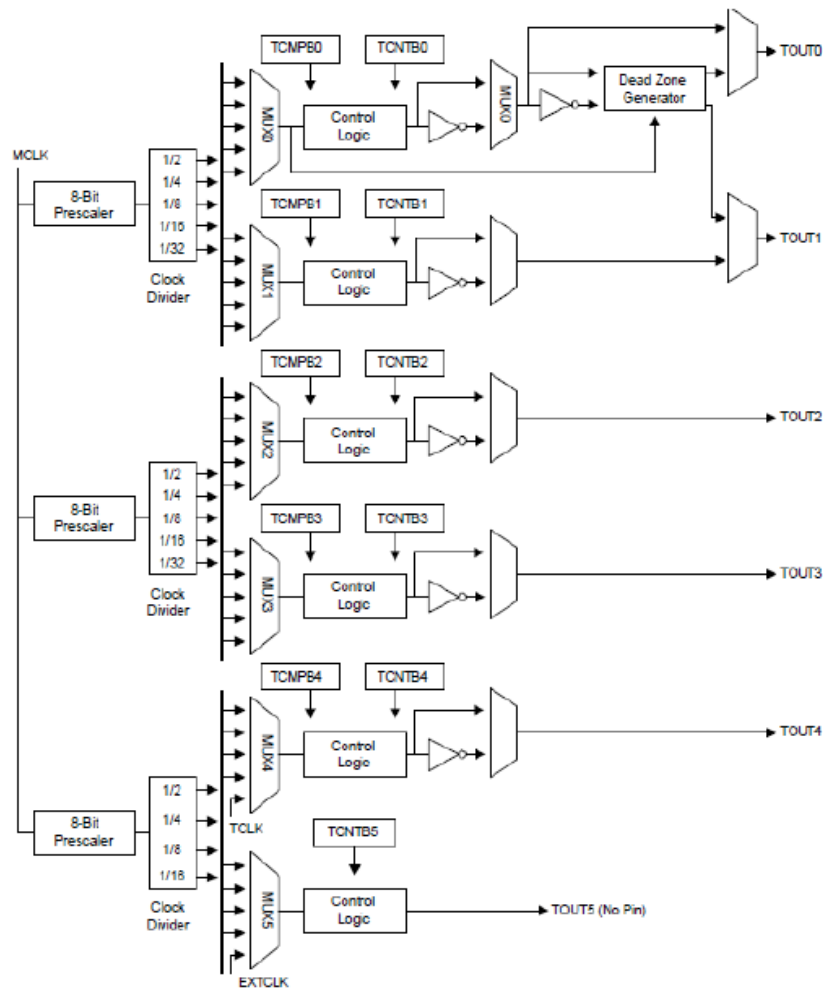


Figura 1.5: Esquema de conexión de los temporizadores del S3C44B0X.

La figura 1.6 representa el diagrama temporal de un ejemplo de funcionamiento habitual de los temporizadores. Se utiliza un sistema de doble buffer, que permite al usuario modificar el valor de recarga del temporizador mientras está contando, sin alterar o parar el contador. Normalmente el temporizador se inicializa escribiendo un valor en el buffer del registro de cuenta (TCNTBn), otro valor en el buffer del registro de comparación (TCMPBn) y activando el manual update bit, como veremos en el siguiente apartado. Mientras el contador está en marcha, se modifica el valor del registro cuenta asociado (TCNTn) y el valor del buffer del registro de cuenta no se verá alterado. Si se está en modo auto-reload, el valor del buffer del registro de cuenta (TCNTBn) se utilizará para recargar TCNTn y el valor del buffer del registro de comparación (TCMPBn) se utilizará para recargar TCMPn cuando la cuenta llegue a cero.

El valor actual de cuenta puede ser leído del registro de observación (TCNTOn). Si se lee TCNTBn no se obtiene el valor actual de la cuenta sino el valor de recarga.

Tabla 1.6: Registros para la utilización de los temporizadores

Registro	Dirección	R/W	Valor de Reset
TCFG0	0x01D50000	R/W	0x00000000
TCFG1	0x01D50004	R/W	0x00000000
TCON	0x01D50008	R/W	0x00000000
TCNTB0	0x01D5000C	R/W	0x00000000
TCMPB0	0x01D50010	R/W	0x00000000
TCNTO0	0x01D50014	R	0x00000000
TCNTB1	0x01D50018	R/W	0x00000000
TCMPB1	0x01D5001C	R/W	0x00000000
TCNTO1	0x01D50020	R	0x00000000
TCNTB2	0x01D50024	R/W	0x00000000
TCMPB2	0x01D50028	R/W	0x00000000
TCNTO2	0x01D5002C	R	0x00000000
TCNTB3	0x01D50030	R/W	0x00000000
TCMPB3	0x01D50034	R/W	0x00000000
TCNTO3	0x01D50038	R	0x00000000
TCNTB4	0x01D5003C	R/W	0x00000000
TCMPB4	0x01D50040	R/W	0x00000000
TCNTO4	0x01D50044	R	0x00000000
TCNTB5	0x01D50048	R/W	0x00000000
TCNTO5	0x01D5004C	R	0x00000000

Inicialización de los temporizadores

Como el valor del registro TCMPBn se copia en el registro de cuenta TCNTn sólo cuando TCNTn alcanza cero, el temporizador no puede ser inicializado simplemente escribiendo en TCMPBn. Para que el valor que escribimos en TCMPBn pase directamente a TCNTn debemos activar el *manual update bit*. La secuencia de inicialización del temporizador sería por tanto:

1. Escribir los valores iniciales de cuenta y comparación en TCNTBn y TCMPBn respectivamente.
2. Activar el *manual update bit* del temporizador.
3. Activar el bit de comienzo (*start bit*) al mismo tiempo que se desactiva el *manual update bit*.

Configuración de los temporizadores

La señal de reloj efectiva para los temporizadores es:

$$F = MCLK / ((\text{valor de pre-escalado} + 1)(\text{valor del divisor}))$$

donde, MCLK es la señal interna de reloj, el valor de pre-escalado está en el intervalo 0–255 y el valor del divisor: 2, 4, 8, 16, 32. El valor de pre-escalado para los temporizadores se configura en el registro TCFG0, descrito en la tabla 1.7.

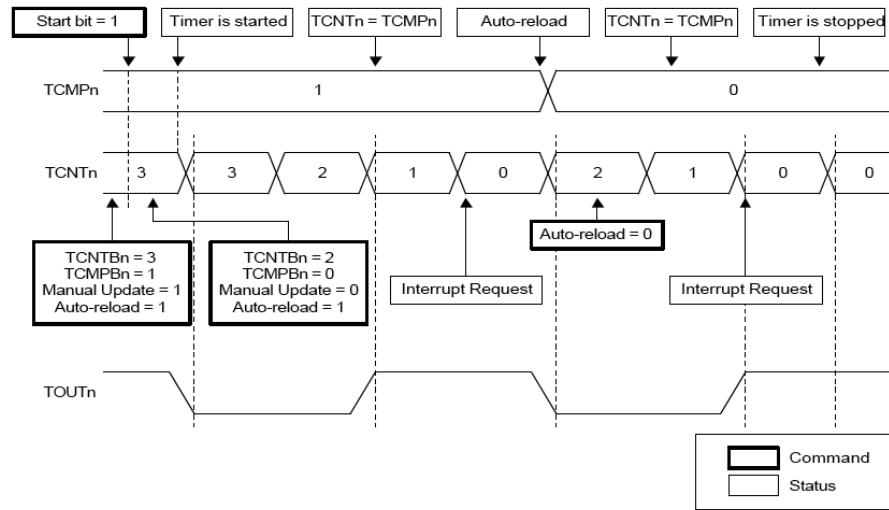


Figura 1.6: Diagrama temporal del funcionamiento de los temporizadores del S3C44B0X.

Tabla 1.7: Registro *TCFG0*.

Función	Bits	Descripción
Longitud de la zona muerta	[31:24]	Estos 8 bits determinan la zona muerta. La unidad de tiempo de la zona muerta es la misma que la del temporizador 0.
Pre-escalado 2	[23:16]	Estos ocho bits determinan el factor de pre-escalado de los temporizadores 4 y 5.
Pre-escalado 1	[15:8]	Estos ocho bits determinan el factor de pre-escalado de los temporizadores 2 y 3.
Pre-escalado 0	[7:0]	Estos ocho bits determinan el factor de pre-escalado de los temporizadores 0 y 1.

El valor del divisor se configura con el registro *TCFG1*, así como la asignación de un temporizador al canal de petición del DMA. La descripción del registro aparece en la tabla 1.8.

El control de los temporizadores (puesta en marcha, parada, actualización, modo auto-reload, etc) se realiza mediante el registro *TCON*. Su funcionalidad está descrita en la tabla 1.9.

Finalmente, para manejar los temporizadores debemos utilizar los registros de buffer, tanto de cuenta como de comparación, y el registro de observación.

Para ampliar la información sobre los temporizadores es preciso consultar el manual del S3c44B0X [um-].

1.6. Display 8-segmentos

En la placa S3CEV40 también tenemos un display de 8 segmentos, que consta de 7 LEDs para conformar cualquier dígito hexadecimal y otro LED adicional para el punto

Tabla 1.8: Registro *TCFG1*.

Función	Bits	Descripción
modo DMA	[27:24]	Selecciona el canal de DMA 0000 = No seleccionado 0001 = Temporizador0 0010 = Temporizador1 0011 = Temporizador2 0100 = Temporizador3 0101 = Temporizador4 0110 = Temporizador5 0111 = Reservado
MUX 5	[23:20]	Selecciona el MUX para el Timer5. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = EXTCLK
MUX 4	[19:16]	Selecciona el MUX para el Timer4. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = TCLK
MUX 3	[15:12]	Selecciona el MUX para el Timer3. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32
MUX 2	[11:8]	Selecciona el MUX para el Timer2. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32
MUX 1	[7:4]	Selecciona el MUX para el Timer1. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32
MUX 0	[3:0]	Selecciona el MUX para el Timer0. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32

decimal. Este display es de tipo ánodo común, lo significa que para encender los LEDs hay que ponerlos a cero. La Figura 1.7 muestra la conexión del display a la lógica de selección externa. Los LEDs del display se conectan al bus de datos del procesador a través de un latch triestado (el chip 74LS573). La selección de este dispositivo se lleva a cabo por medio de la señal CS6, generada por un decodificador de 3 a 8 (chip 74LV138), que es habilitado con la señal nGCS1 según el esquema de la figura 1.7. Esta lógica de selección es común a todos los dispositivos ubicados en el Banco-1. La tabla 1.10 muestra el rango de direcciones y la señal de selección que le asigna a cada uno de ellos. Como podemos ver, el latch conectado al display de 8 segmentos se activa con la señal CS6, cada vez que accedemos a alguna dirección del rango 0x0214_0000-0x0217_FFFF. Mientras la entrada LE (Latch Enable) permanezca activa el latch dejará pasar el byte definido por los 8 bits menos significativos del bus de datos. Cuando esta señal se desactiva el latch mantiene el último valor que le ha llegado. Sin embargo, el banco 1 tiene una anchura de 8 bits. Esto quiere decir que si hacemos una escritura de tamaño palabra el controlador de memoria enviará por separado los 4 bytes que componen la palabra. Como el procesador se configura como little-endian se escribirían los bytes en orden de su peso (0,1,2 y 3), quedando en el latch el byte más significativo, en lugar del menos significativo. Por este motivo es muy importante que siempre se realicen escrituras de tamaño byte en el display de 8 segmentos, utilizando la instrucción strb.

Tabla 1.9: Registro *TCON*.

Función	Bits	Descripción
Timer 5 auto reload on/off	[26]	Este bit determina el auto-reload para el Temporizador 5. 0 = One-shot 1 = Interval mode (auto reload)
Timer 5 manual update	[25]	Este bit determina el <i>manual update</i> del Temporizador 5. 0 = No operation 1 = Update TCNTB5
Timer 5 start/stop	[24]	Este bit determina el <i>start/stop</i> del Temporizador 5. 0 = Stop 1 = Start for Timer 5
Timer 4 auto reload on/off	[23]	Este bit determina el auto-reload para el Temporizador 4. 0 = One-shot 1 = Interval mode (auto reload)
Timer 4 output inverter on/off	[22]	Este bit determina el inversor de salida para el Temporizador 4. 0 = Inverter off 1 = Inverter on for TOUT4
Timer 4 manual update	[21]	Este bit determina el <i>manual update</i> del Temporizador 4. 0 = No operation 1 = Update TCNTB4, TCMPB4
Timer 4 start/stop	[20]	Este bit determina el <i>start/stop</i> del Temporizador 4. 0 = Stop 1 = Start for Timer 4
Timer 3 auto reload on/off	[19]	Este bit determina el auto-reload para el Temporizador 3. 0 = One-shot 1 = Interval mode (auto reload)
Timer 3 output inverter on/off	[18]	Este bit determina el inversor de salida para el Temporizador 3. 0 = Inverter off 1 = Inverter on for TOUT3
Timer 3 manual update	[17]	Este bit determina el <i>manual update</i> del Temporizador 3. 0 = No operation 1 = Update TCNTB3, TCMPB3
Timer 3 start/stop	[16]	Este bit determina el <i>start/stop</i> del Temporizador 3. 0 = Stop 1 = Start for Timer 3
Timer 2 auto reload on/off	[15]	Este bit determina el auto-reload para el Temporizador 2. 0 = One-shot 1 = Interval mode (auto reload)
Timer 2 output inverter on/off	[14]	Este bit determina el inversor de salida para el Temporizador 2. 0 = Inverter off 1 = Inverter on for TOUT2
Timer 2 manual update	[13]	Este bit determina el <i>manual update</i> del Temporizador 2. 0 = No operation 1 = Update TCNTB2, TCMPB2
Timer 2 start/stop	[12]	Este bit determina el <i>start/stop</i> del Temporizador 2. 0 = Stop 1 = Start for Timer 2
Timer 1 auto reload on/off	[11]	Este bit determina el auto-reload para el Temporizador 1. 0 = One-shot 1 = Interval mode (auto reload)
Timer 1 output inverter on/off	[10]	Este bit determina el inversor de salida para el Temporizador 1. 0 = Inverter off 1 = Inverter on for TOUT1
Timer 1 manual update	[9]	Este bit determina el <i>manual update</i> del Temporizador 1. 0 = No operation 1 = Update TCNTB1, TCMPB1
Timer 1 start/stop	[8]	Este bit determina el <i>start/stop</i> del Temporizador 1. 0 = Stop 1 = Start for Timer 1
Dead zone enable	[4]	Este bit determina la operación de zona muerta. 0 = Disable 1 = Enable
Timer 0 auto reload on/off	[3]	Este bit determina el auto-reload para el Temporizador 0. 0 = One-shot 1 = Interval mode(auto reload)
Timer 0 output inverter on/off	[2]	Este bit determina el inversor de salida para el Temporizador 0. 0 = Inverter off 1 = Inverter on for TOUT0
Timer 0 manual update on/off	[1]	Este bit determina el <i>manual update</i> del Temporizador 0. 0 = No operation 1 = Update TCNTB0, TCMPB0
Timer 0 start/stop	[0]	Este bit determina el <i>start/stop</i> del Temporizador 0. 0 = Stop 1 = Start for Timer 0

Tabla 1.10: Rango asignado a cada uno de los dispositivos ubicados en el Banco-1.

Dispositivo	CS	Dirección
USB	CS1	0x0200_0000 - 0x0203_FFFF
Nand Flash	CS2	0x0204_0000 - 0x0207_FFFF
IDE (IOR/W)	CS3	0x0208_0000 - 0x020B_FFFF
IDE (KEY)	CS4	0x020C_0000 - 0x020F_FFFF
IDE (PDIAG)	CS5	0x0210_0000 - 0x0213_FFFF
8-SEG	CS6	0x0214_0000 - 0x0217_FFFF
ETHERNET	CS7	0x0218_0000 - 0x021B_FFFF
LCD	CS8	0x021C_0000 - 0x021F_FFFF

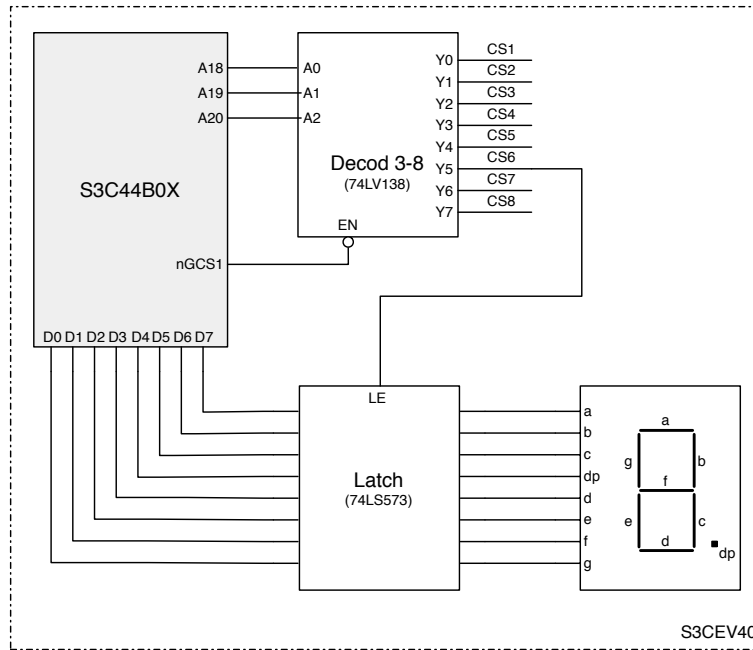


Figura 1.7: Esquema de conexión del display de 8 segmentos en la placa S3CEV40. A[20:18] son bits de direcciones y D[7:0] son bits de datos del bus del sistema del S3C44B0X. Es conveniente recordar que la señal nGCS1 del controlador de memoria se activa a baja cuando se referencia una dirección del Banco-1.

Bibliografía

[um-] S3c44b0x risc microprocessor product overview. Accesible en http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly_id=229&partnum=S3C44B0. Hay una copia en el campus virtual.