

E/S mediante interrupciones vectorizadas

Estructura de computadores

Grado en Informática

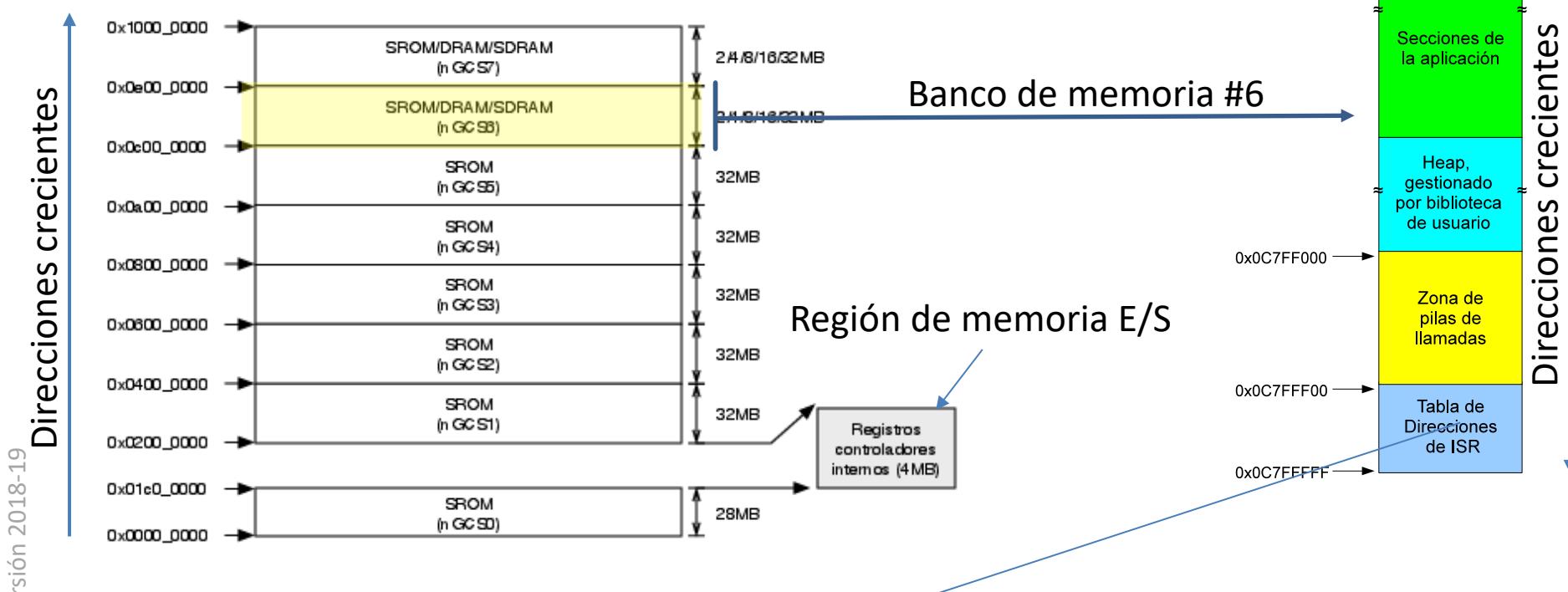
Universidad Complutense de Madrid



Bibliografía

- Este documento está basado en:
 - “Sistema de memoria y de entrada/salida en la placa S3CEV40” por Luis Piñuel y Christian Tenllado licenciado bajo [CC BY NC SA 3.0](#).
 - “USER'S MANUAL S3C44BOX 16/32-Bit RISC Microprocessor”, Samsung Electronics

Mapa de memoria



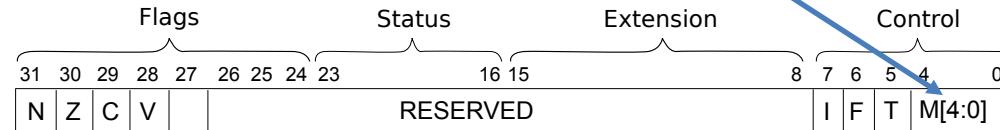
La tabla de ISR tendrá una entrada por vector de interrupción, que se utilizará para almacenar la dirección de la rutina que debe tratar dicha interrupción (o excepción interna)

Excepciones

- ARM posee excepciones autovectorizadas y vectorizadas
 - Cada fuente de excepción tiene asignada un vector y un modo de ejecución

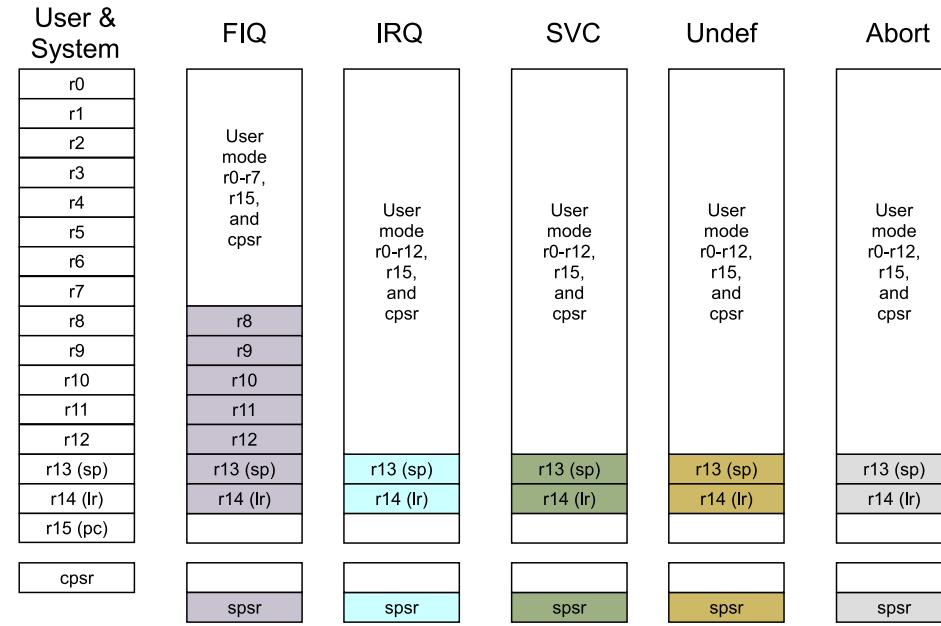
Prioridad	Excepción	Modo	Vector
1	Reset	SVC	0x00
2	Data Abort	Abort	0x10
3	FIQ	FIQ	0x1C
4	IRQ	IRQ	0x18
6	Prefetch Abort	Abort	0x0C
7	Instrucción no definida	Undef	0x04
8	SWI	SVC	0x08

Modo del procesador	Código	Uso
<i>usr</i>	10000	Ejecución de código de usuario
<i>fiq</i>	10001	Servicio de int. rápidas
<i>irq</i>	10010	Servicio de int. lentas
<i>svc</i>	10011	Modo protegido para sistema operativo (int. sw)
<i>abt</i>	10111	Procesado de fallos de acceso a mem
<i>und</i>	11011	Manejo de instrucc. indefinidas
<i>sys</i>	11111	Ejecución de tareas del SO



Excepciones. Modos ejecución

Prioridad	Excepción	Modo	Vector
1	Reset	SVC	0x00
2	Data Abort	Abort	0x10
3	FIQ	FIQ	0x1C
4	IRQ	IRQ	0x18
6	Prefetch Abort	Abort	0x0C
7	Instrucción no definida	Undef	0x04
8	SWI	SVC	0x08



Pasos

1. Se genera la excepción
2. Se ejecuta la instrucción almacenada en la dirección de memoria indicada por el vector

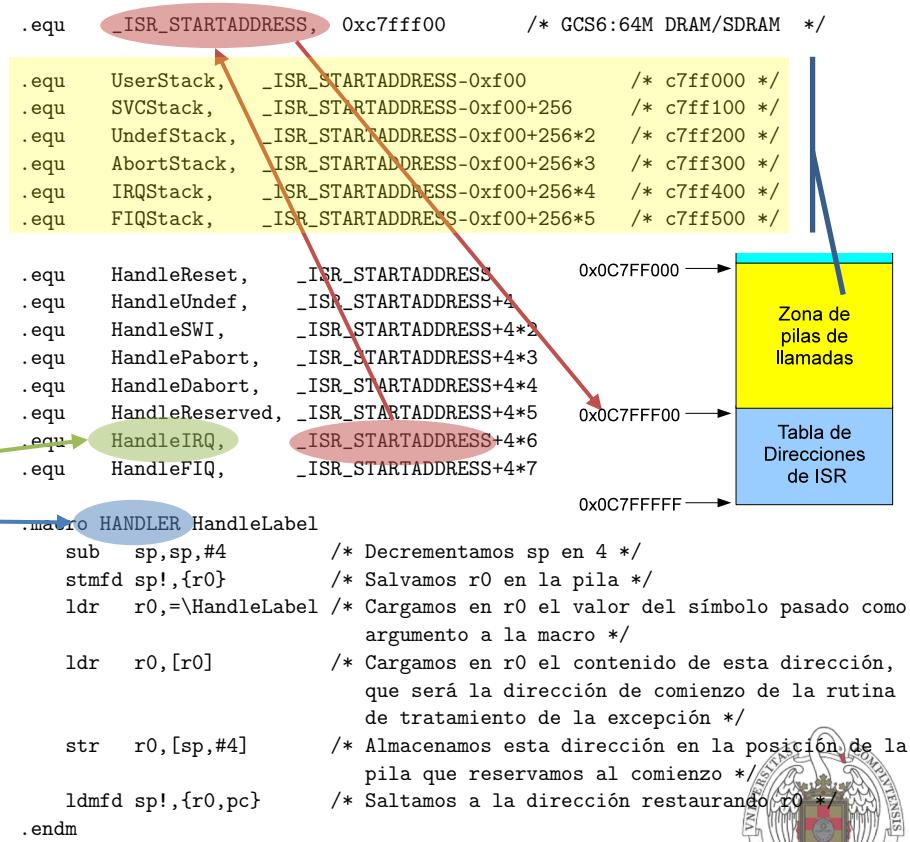
```
/*Comienzo del programa en dirección 0x00*/
start:
    b ResetHandler          /* 0x00 : vector de reset */
    b HandlerUndef           /* 0x04 : vector de Undef */
    b HandlerSWI              /* 0x08 : vector SWI */
    b HandlerPabort            /* 0x0C : vector de Pabort */
    b HandlerDabort            /* 0x10 : vector de Dabort */
    b .                      /* 0x14 : utilizado en ARMv6 */
    b HandlerIRQ              /* 0x18 : vector de IRQ */
    b HandlerFIQ              /* 0x1C : vector de FIQ */

    /*Más código que veremos en la práctica siguiente */
    .align

    HandlerFIQ:   HANDLER HandleFIQ
    HandlerIRQ:   HANDLER HandleIRQ
    HandlerUndef: HANDLER HandleUndef
    HandlerSWI:   HANDLER HandleSWI
    HandlerDabort: HANDLER HandleDabort
    HandlerPabort: HANDLER HandlePabort

    /*Más código que veremos en la práctica siguiente */

ResetHandler:
    /* Código de la rutina de reset */
```



Práctica 3. E/S mediante interrupciones vectorizadas

```

start:
    b ResetHandler      /* 0x00 : vector de reset */
    b HandlerUndef      /* 0x04 : vector de Undef */
    b HandlerSWI         /* 0x08 : vector SWI */
    b HandlerPabort     /* 0x0C : vector de Pabort */
    b HandlerDabort     /* 0x10 : vector de Dabort */
    b .                 /* 0x14 : utilizado en ARMv6 */
    b HandlerIRQ         /* 0x18 : vector de IRQ */
    b HandlerFIQ         /* 0x1C : vector de FIQ */

/*Más código que veremos en la práctica siguiente*/
.align

HandlerFIQ:    HANDLER HandleFIQ
HandlerIRQ:    HANDLER HandleIRQ
HandlerUndef:  HANDLER HandleUndef
HandlerSWI:    HANDLER HandleSWI
HandlerDabort: HANDLER HandleDabort
HandlerPabort: HANDLER HandlePabort

```

```

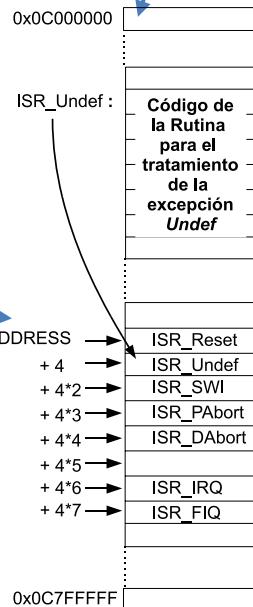
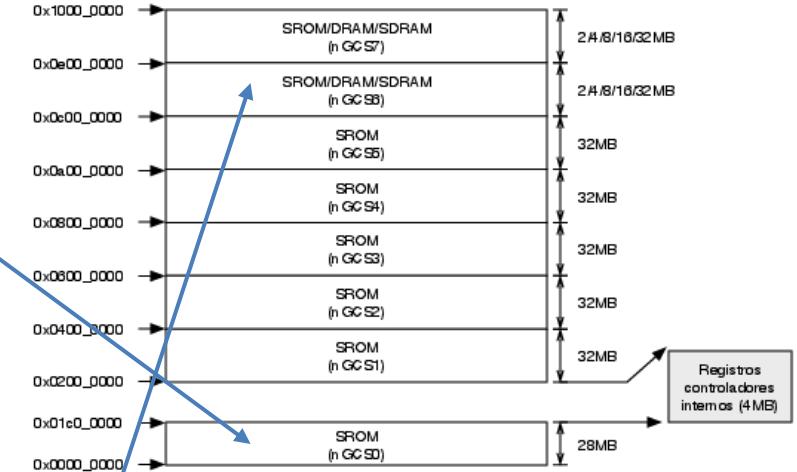
.equ _ISR_STARTADDRESS, 0xc7fff00 /* GCS6:64M DRAM/SDRAM */
.equ UserStack, _ISR_STARTADDRESS-0xf00 /* c7ff000 */
.equ SVCStack, _ISR_STARTADDRESS-0xf00+256 /* c7ff100 */
.equ UndefStack, _ISR_STARTADDRESS-0xf00+256*2 /* c7ff200 */
.equ AbortStack, _ISR_STARTADDRESS-0xf00+256*3 /* c7ff300 */
.equ IRQStack, _ISR_STARTADDRESS-0xf00+256*4 /* c7ff400 */
.equ FIQStack, _ISR_STARTADDRESS-0xf00+256*5 /* c7ff500 */

.equ HandleReset, _ISR_STARTADDRESS
.equ HandleUndef, _ISR_STARTADDRESS+4
.equ HandleSWI, _ISR_STARTADDRESS+4*2
.equ HandlePabort, _ISR_STARTADDRESS+4*3
.equ HandleDabort, _ISR_STARTADDRESS+4*4
.equ HandleReserved, _ISR_STARTADDRESS+4*5
.equ HandleIRQ, _ISR_STARTADDRESS+4*6
.equ HandleFIQ, _ISR_STARTADDRESS+4*7

.macro HANDLER HandleLabel
    sub sp,sp,#4 /* Decrementamos sp en 4 */
    stmdf spl,{r0} /* Salvamos r0 en la pila */
    ldr r0,=HandleLabel /* Cargamos en r0 el valor del símbolo pasado como argumento a la macro */
    ldr r0,[r0] /* Cargamos en r0 el contenido de esta dirección, que será la dirección de comienzo de la rutina de tratamiento de la excepción */
    str r0,[sp,#4] /* Almacenamos esta dirección en la posición de la pila que reservamos al comienzo */
    ldmfd sp!,{r0,pc} /* Saltamos a la dirección restaurando r0 */
.endm

```

0x00000000	B ResetHandler
0x00000004	B HandlerUndef
0x00000008	B HandlerSWI
0x0000000C	B HandlerPabort
0x00000010	B HandlerDabort
0x00000014	B .
0x00000018	B HandlerIRQ
0x0000001C	B HandlerFIQ
0x00000020	
0x00000024	
0x00000028	
0x0000002C	
0x00000030	
0x00000034	
0x00000038	
0x00000040	
0x00000044	
0x00000048	
0x00000050	
0x00000054	
0x00000058	
0x00000060	
0x00000064	
0x00000068	
0x00000070	
0x00000074	
0x00000078	
0x0000007C	
0x00000080	
0x00000084	
0x00000088	
0x00000090	
0x00000094	
0x00000098	
0x000000A0	
0x000000A4	
0x000000A8	
0x000000B0	
0x000000B4	
0x000000B8	
0x000000C0	
0x000000C4	
0x000000C8	
0x000000D0	
0x000000D4	
0x000000D8	
0x000000E0	
0x000000E4	
0x000000E8	
0x000000F0	
0x000000F4	
0x000000F8	
0x000000F0	0x001FFFFF



Las direcciones donde se deben ubicar las direcciones de comienzo de las ISR en modo vectorizado están en 44b.h

Asignación

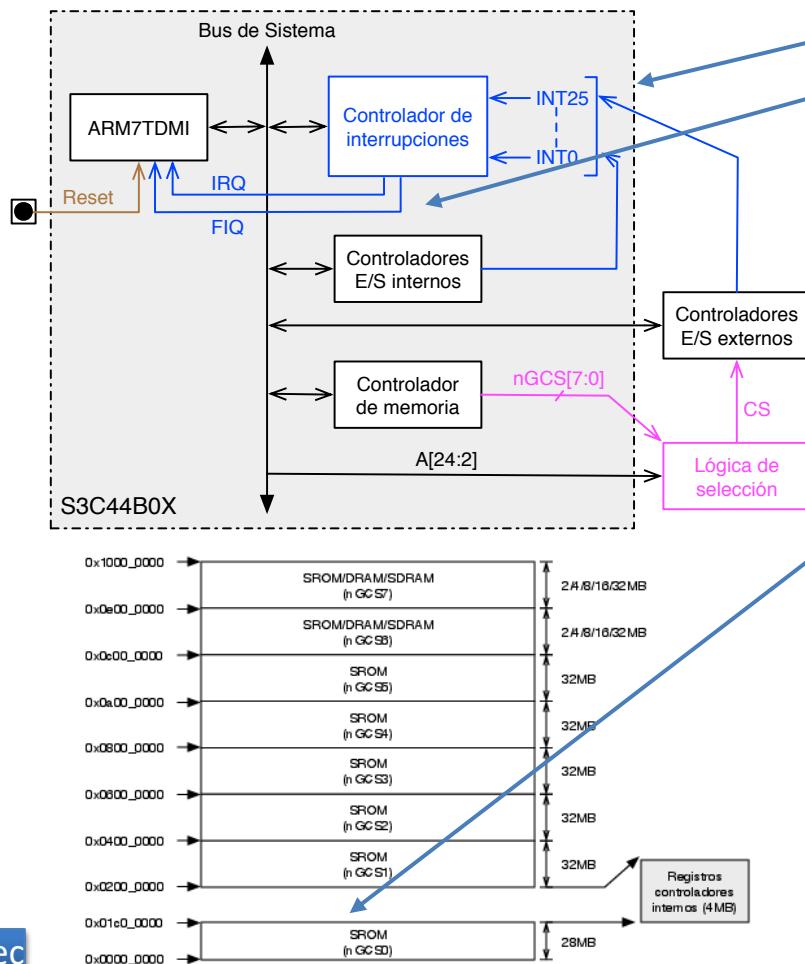
pISR_TIMERO = (unsigned) timer_ISR;

_ISR_STARTADDRESS+0x54



Controlador de interrupciones

- Elemento hardware externo al ARM7TDMI cuya función es ampliar y mejorar la gestión de interrupciones



- Hasta 26 fuentes de interrupción
- Solo 2 líneas llegan al procesador
- ¿Cómo identifica el procesador la fuente? Dos mecanismos
 - Vectorizadas**
 - Autovectorizadas
 - Registro INTPND que indica las líneas con interrupciones pendientes.

Register	Address	R/W	Description	Reset Value
INTPND	0x01E00004	R	Indicates the interrupt request status. 0 = The interrupt has not been requested 1 = The interrupt source has asserted the interrupt request	0x00000000

- ISPR interrupción en servicio. La de más alta prioridad

Register	Address	R/W	Description	Reset Value
I_ISPR	0x01E00020	R	IRQ interrupt service pending register	0x00000000

Registros controlador de interrupciones

Register	Address	R/W	Description	Reset Value
INTPND	0x01E00004	R	Indicates the interrupt request status. 0 = The interrupt has not been requested 1 = The interrupt source has asserted the interrupt request	0x00000000

Nº/Bit	Nombre	Fuente	Vector
25	EINT0	Interrupción externa 0	0x20
24	EINT1	Interrupción externa 1	0x24
23	EINT2	Interrupción externa 2	0x28
22	EINT3	Interrupción externa 3	0x2c
21	EINT4/5/6/7	Interrupciones externas 4, 5, 6 y 7	0x30
20	TICK	Interrupción de <i>tick</i> del RTC	0x34
19	ZDMA0	Interrupción del ZDMA0	0x40
18	ZDMA1	Interrupción del ZDMA1	0x44
17	BDMA0	Interrupción del BDMA0	0x48
16	BDMA1	Interrupción del BDMA1	0x4c
15	WDT	Interrupción del Watch-Dog Timer	0x50
14	UERR0/1	Interrupciones de error de las UART0/1	0x54
13	TIMER0	Interrupción del Timer0	0x60
12	TIMER1	Interrupción del Timer1	0x64
11	TIMER2	Interrupción del Timer2	0x68
10	TIMER3	Interrupción del Timer3	0x6c
9	TIMER4	Interrupción del Timer4	0x70
8	TIMER5	Interrupción del Timer5	0x74
7	URXD0	Interrupción de recepción de la UART0	0x80
6	URXD1	Interrupción de recepción de la UART1	0x84
5	IIC	Interrupción de controlador de bus IIC	0x88
4	SIO	Interrupción del controlador SIO	0x8c
3	UTXD0	Interrupción de envío de la UART0	0x90
2	UTXD1	Interrupción de envío de la UART1	0x94
1	RTC	Interrupción de alarma del RTC	0xa0
0	ADC	Interrupción <i>EOC</i> del conversor ADC	0xc0

0x00000000	B ResetHandler
0x00000004	B HandlerUndef
0x00000008	B HandlerSWI
0x0000000C	B HandlerPabort
0x00000010	B HandlerDabort
0x00000014	B .
0x00000018	B HandlerIRQ
0x0000001C	B HandlerFIQ
0x00000020	
0x00000024	
0x00000028	
0x0000002C	
0x00000030	
0x00000034	
0x00000038	
0x0000003C	
0x00000040	
0x00000044	
0x00000048	
0x0000004C	
0x00000050	
0x00000054	
0x00000058	
0x00000060	
0x00000064	
0x00000068	
0x0000006C	
0x00000070	
0x00000074	
0x00000078	
0x0000007C	
0x00000080	
0x00000084	
0x00000088	
0x0000008C	
0x00000090	
0x00000094	
0x000000A0	
0x000000B0	
0x000000C0	
0x000000D0	
0x000000E0	
0x000000F0	
0x000000F4	
0x000000F8	
0x000000FC	
0x000000F0	
0x000000F4	
0x000000F8	
0x000000FC	
0x00000100	
0x00000104	
0x00000108	
0x0000010C	
0x00000110	
0x00000114	
0x00000118	
0x0000011C	
0x00000120	
0x00000124	
0x00000128	
0x0000012C	
0x00000130	
0x00000134	
0x00000138	
0x0000013C	
0x00000140	
0x00000144	
0x00000148	
0x0000014C	
0x00000150	
0x00000154	
0x00000158	
0x0000015C	
0x00000160	
0x00000164	
0x00000168	
0x0000016C	
0x00000170	
0x00000174	
0x00000178	
0x0000017C	
0x00000180	
0x00000184	
0x00000188	
0x0000018C	
0x00000190	
0x00000194	
0x00000198	
0x0000019C	
0x000001A0	
0x000001A4	
0x000001A8	
0x000001AC	
0x000001B0	
0x000001B4	
0x000001B8	
0x000001BC	
0x000001C0	
0x000001C4	
0x000001C8	
0x000001CC	
0x000001D0	
0x000001D4	
0x000001D8	
0x000001DC	
0x000001E0	
0x000001E4	
0x000001E8	
0x000001EC	
0x000001F0	
0x000001F4	
0x000001F8	
0x000001FC	
0x00000200	
0x00000204	
0x00000208	
0x0000020C	
0x00000210	
0x00000214	
0x00000218	
0x0000021C	
0x00000220	
0x00000224	
0x00000228	
0x0000022C	
0x00000230	
0x00000234	
0x00000238	
0x0000023C	
0x00000240	
0x00000244	
0x00000248	
0x0000024C	
0x00000250	
0x00000254	
0x00000258	
0x0000025C	
0x00000260	
0x00000264	
0x00000268	
0x0000026C	
0x00000270	
0x00000274	
0x00000278	
0x0000027C	
0x00000280	
0x00000284	
0x00000288	
0x0000028C	
0x00000290	
0x00000294	
0x00000298	
0x0000029C	
0x000002A0	
0x000002A4	
0x000002A8	
0x000002AC	
0x000002B0	
0x000002B4	
0x000002B8	
0x000002BC	
0x000002C0	
0x000002C4	
0x000002C8	
0x000002CC	
0x000002D0	
0x000002D4	
0x000002D8	
0x000002DC	
0x000002E0	
0x000002E4	
0x000002E8	
0x000002EC	
0x000002F0	
0x000002F4	
0x000002F8	
0x000002FC	
0x00000300	
0x00000304	
0x00000308	
0x0000030C	
0x00000310	
0x00000314	
0x00000318	
0x0000031C	
0x00000320	
0x00000324	
0x00000328	
0x0000032C	
0x00000330	
0x00000334	
0x00000338	
0x0000033C	
0x00000340	
0x00000344	
0x00000348	
0x0000034C	
0x00000350	
0x00000354	
0x00000358	
0x0000035C	
0x00000360	
0x00000364	
0x00000368	
0x0000036C	
0x00000370	
0x00000374	
0x00000378	
0x0000037C	
0x00000380	
0x00000384	
0x00000388	
0x0000038C	
0x00000390	
0x00000394	
0x00000398	
0x0000039C	
0x000003A0	
0x000003A4	
0x000003A8	
0x000003AC	
0x000003B0	
0x000003B4	
0x000003B8	
0x000003BC	
0x000003C0	
0x000003C4	
0x000003C8	
0x000003CC	
0x000003D0	
0x000003D4	
0x000003D8	
0x000003DC	
0x000003E0	
0x000003E4	
0x000003E8	
0x000003EC	
0x000003F0	
0x000003F4	
0x000003F8	
0x000003FC	
0x00000400	
0x00000404	
0x00000408	
0x0000040C	
0x00000410	
0x00000414	
0x00000418	
0x0000041C	
0x00000420	
0x00000424	
0x00000428	
0x0000042C	
0x00000430	
0x00000434	
0x00000438	
0x0000043C	
0x00000440	
0x00000444	
0x00000448	
0x0000044C	
0x00000450	
0x00000454	
0x00000458	
0x0000045C	
0x00000460	
0x00000464	
0x00000468	
0x0000046C	
0x00000470	
0x00000474	
0x00000478	
0x0000047C	
0x00000480	
0x00000484	
0x00000488	
0x0000048C	
0x00000490	
0x00000494	
0x00000498	
0x0000049C	
0x000004A0	
0x000004A4	
0x000004A8	
0x000004AC	
0x000004B0	
0x000004B4	
0x000004B8	
0x000004BC	
0x000004C0	
0x000004C4	
0x000004C8	
0x000004CC	
0x000004D0	
0x000004D4	
0x000004D8	
0x000004DC	
0x000004E0	
0x000004E4	
0x000004E8	
0x000004EC	
0x000004F0	
0x000004F4	
0x000004F8	
0x000004FC	
0x00000500	
0x00000504	
0x00000508	
0x0000050C	
0x00000510	
0x00000514	
0x00000518	
0x0000051C	
0x00000520	
0x00000524	
0x00000528	
0x0000052C	
0x00000530	
0x00000534	
0x00000538	
0x0000053C	
0x00000540	
0x00000544	
0x00000548	
0x0000054C	
0x00000550	
0x00000554	
0x00000558	
0x0000055C	
0x00000560	
0x00000564	
0x00000568	
0x0000056C	
0x00000570	
0x00000574	
0x00000578	
0x0000057C	
0x00000580	
0x00000584	
0x00000588	
0x0000058C	
0x00000590	
0x00000594	
0x00000598	
0x0000059C	
0x000005A0	
0x000005A4	
0x000005A8	
0x000005AC	
0x000005B0	
0x000005B4	
0x000005B8	
0x000005BC	
0x000005C0	
0x000005C4	
0x000005C8	
0x000005CC	
0x000005D0	
0x000005D4	
0x000005D8	
0x000005DC	
0x000005E0	
0x000005E4	
0x000005E8	
0x000005EC	
0x000005F0	
0x000005F4	
0x000005F8	
0x000005FC	
0x00000600	
0x00000604	
0x00000608	
0x0000060C	
0x00000610	
0x00000614	
0x00000618	
0x0000061C	
0x00000620	
0x00000624	
0x00000628	
0x0000062C	
0x00000630	
0x00000634	
0x00000638	
0x0000063C	
0x00000640	
0x00000644	
0x00000648	
0x0000064C	
0x00000650	
0x00000654	
0x00000658	
0x0000065C	
0x00000660	
0x00000664	
0x00000668	
0x0000066C	
0x00000670	
0x00000674	
0x00000678	
0x0000067C	
0x00000680	
0x00000684	
0x00000688	
0x0000068C	
0x00000690	
0x00000694	
0x00000698	
0x0000069C	
0x000006A0	
0x000006A4	
0x000006A8	
0x000006AC	
0x000006B0	
0x000006B4	
0x000006B8	
0x000006BC	
0x000006C0	
0x000006C4	
0x000006C8	
0x000006CC	
0x000006D0	
0x000006D4	
0x000006D8	
0x000006DC	
0x000006E0	
0x000006E4	
0x000006E8	
0x000006EC	
0x000006F0	
0x000006F4	
0x000006F8	
0x000006FC	
0x00000700	
0x00000704	
0x00000708	
0x0000070C	
0x00000710	
0x00000714	
0x00000718	
0x0000071C	
0x00000720	
0x00000724	
0x00000728	
0x0000072C	

Registros controlador de interrupciones

Register	Address	R/W	Description	Reset Value
INTPND	0x01E00004	R	Indicates the interrupt request status. 0 = The interrupt has not been requested 1 = The interrupt source has asserted the interrupt request	0x00000000

Nº/Bit	Nombre	Fuente	Vector
25	EINT0	Interrupción externa 0	0x20
24	EINT1	Interrupción externa 1	0x24
23	EINT2	Interrupción externa 2	0x28
22	EINT3	Interrupción externa 3	0x2c
21	EINT4/5/6/7	Interrupciones externas 4, 5, 6 y 7	0x30
20	TICK	Interrupción de <i>tick</i> del RTC	0x34
19	ZDMA0	Interrupción del ZDMA0	0x40
18	ZDMA1	Interrupción del ZDMA1	0x44
17	BDMA0	Interrupción del BDMA0	0x48
16	BDMA1	Interrupción del BDMA1	0x4c
15	WDT	Interrupción del Watch-Dog Timer	0x50
14	UERR0/1	Interrupciones de error de las UART0/1	0x54
13	TIMER0	Interrupción del Timer0	0x60
12	TIMER1	Interrupción del Timer1	0x64
11	TIMER2	Interrupción del Timer2	0x68
10	TIMER3	Interrupción del Timer3	0x6c
9	TIMER4	Interrupción del Timer4	0x70
8	TIMER5	Interrupción del Timer5	0x74
7	URXD0	Interrupción de recepción de la UART0	0x80
6	URXD1	Interrupción de recepción de la UART1	0x84
5	IIC	Interrupción de controlador de bus IIC	0x88
4	SIO	Interrupción del controlador SIO	0x8c
3	UTXD0	Interrupción de envío de la UART0	0x90
2	UTXD1	Interrupción de envío de la UART1	0x94
1	RTC	Interrupción de alarma del RTC	0xa0
0	ADC	Interrupción <i>EOC</i> del conversor ADC	0xc0

Register	Address	R/W	Description	Reset Value
INTCON	0x01E00000	R/W	Interrupt control Register	0x7

INTCON	Bit	Description	initial state
Reserved	[3]	0	0
V	[2]	This bit disables/enables vector mode for IRQ 0 = Vectored interrupt mode 1 = Non-vectored interrupt mode	1
I	[1]	This bit enables IRQ interrupt request line to CPU 0 = IRQ interrupt enable 1 = Reserved Note : Before using the IRQ interrupt this bit must be cleared.	1
F	[0]	This bit enables FIQ interrupt request line to CPU 0 = FIQ interrupt enable (Not allowed vectored interrupt mode) 1 = Reserved Note : Before using the FIQ interrupt this bit must be cleared.	1

Register	Address	R/W	Description	Reset Value
INTMOD	0x01E00008	R/W	Interrupt mode Register 0 = IRQ mode 1 = FIQ mode	0x00000000

Register	Address	R/W	Description	Reset Value
INTMSK	0x01E0000C	R/W	Determines which interrupt source is masked. The masked interrupt source will not be serviced. 0 = Interrupt service is available 1 = Interrupt service is masked	0x07fffff

Modo vectorizado



Registros controlador de interrupciones

Register	Address	R/W	Description	Reset Value
INTPND	0x01E00004	R	Indicates the interrupt request status. 0 = The interrupt has not been requested 1 = The interrupt source has asserted the interrupt request	0x00000000

Nº/Bit	Nombre	Fuente	Vector
25	EINT0	Interrupción externa 0	0x20
24	EINT1	Interrupción externa 1	0x24
23	EINT2	Interrupción externa 2	0x28
22	EINT3	Interrupción externa 3	0x2c
21	EINT4/5/6/7	Interrupciones externas 4, 5, 6 y 7	0x30
20	TICK	Interrupción de <i>tick</i> del RTC	0x34
19	ZDMA0	Interrupción del ZDMA0	0x40
18	ZDMA1	Interrupción del ZDMA1	0x44
17	BDMA0	Interrupción del BDMA0	0x48
16	BDMA1	Interrupción del BDMA1	0x4c
15	WDT	Interrupción del Watch-Dog Timer	0x50
14	UERR0/1	Interrupciones de error de las UART0/1	0x54
13	TIMER0	Interrupción del Timer0	0x60
12	TIMER1	Interrupción del Timer1	0x64
11	TIMER2	Interrupción del Timer2	0x68
10	TIMER3	Interrupción del Timer3	0x6c
9	TIMER4	Interrupción del Timer4	0x70
8	TIMER5	Interrupción del Timer5	0x74
7	URXD0	Interrupción de recepción de la UART0	0x80
6	URXD1	Interrupción de recepción de la UART1	0x84
5	IIC	Interrupción de controlador de bus IIC	0x88
4	SIO	Interrupción del controlador SIO	0x8c
3	UTXD0	Interrupción de envío de la UART0	0x90
2	UTXD1	Interrupción de envío de la UART1	0x94
1	RTC	Interrupción de alarma del RTC	0xa0
0	ADC	Interrupción <i>EOC</i> del conversor ADC	0xc0

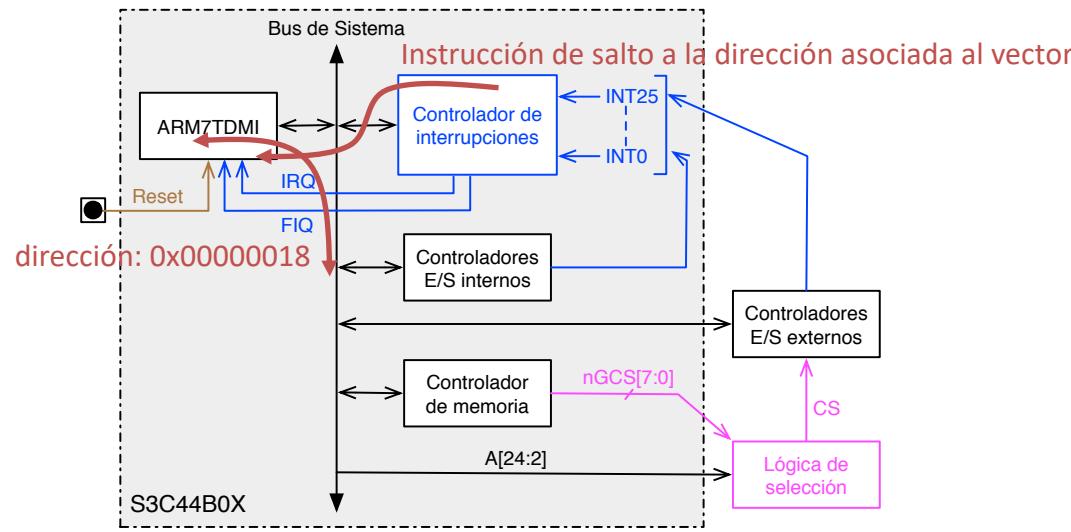
Register	Address	R/W	Description	Reset Value
EXTINTPND	0x01D20054	R/W	External interrupt pending Register	0x00

PUPS	Bit	Description
EXTINTPND3	[3]	If EINT7 is activated, EXINTPND3 bit is set to 1, and also INTPND[21] is set to 1.
EXTINTPND2	[2]	If EINT6 is activated, EXINTPND2 bit is set to 1, and also INTPND[21] is set to 1.
EXTINTPND1	[1]	If EINT5 is activated, EXINTPND1 bit is set to 1, and also INTPND[21] is set to 1.
EXTINTPND0	[0]	If EINT4 is activated, EXINTPND0 bit is set to 1, and also INTPND[21] is set to 1.

Modo vectorizado

Modo vectorizado

- If ARM7TDMI receives the IRQ interrupt request from the interrupt controller, ARM7TDMI executes an instruction at 0x00000018. In vectored interrupt mode, the interrupt controller will load branch instructions on the data bus when ARM7TDMI fetches the instructions at 0x00000018. The branch instructions let the program counter be a unique address corresponding to each interrupt source.
- The interrupt controller generates the machine code for branching to the vector address of each interrupt source. For example, If EINT0 is IRQ, the interrupt controller must generate the branch instruction which branches from 0x18 to 0x20. So, the interrupt controller generates the machine code, 0xea000000.
- The user program code must locate the branch instruction, which branches to the corresponding ISR (interrupt service routine) at each vector address.



Temporizador

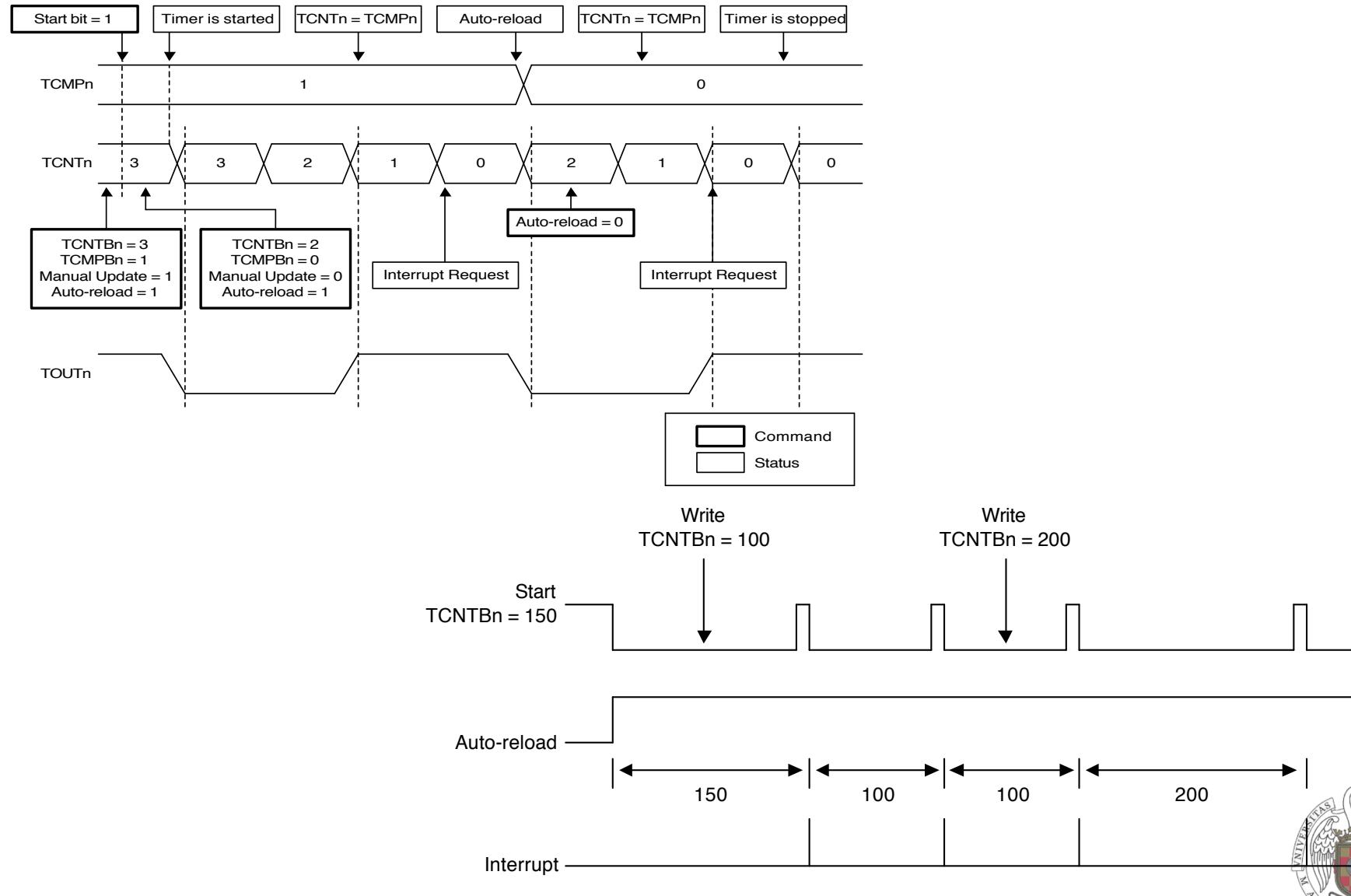
- S3C44BOX tiene 6 temporizadores (0 a 5) de 16 bits
 - Temporizadores 0 a 4 tienen salida con función Pulse Width Modulation
 - Temporizador 5 es un temporizador interno sin señal de salida
- Cada uno de los temporizadores tiene asociado tres registros visibles

TCNTB1	0x01d50018	R/W	Timer Count Buffer 1
TCMPB1	0x01d5001c		Timer Compare Buffer 1
TCNTO1	0x01d50020	R	Timer Count Observation 1

y dos registros no visibles: TCNT* y TCMP*

- Funcionamiento
 - Se inicializa el valor del temporizador (escribiendo TCNTB* y TCMPB*)
 - Los valores se cargan en en TCNTO* y en los registros internos cuando el temporizador es habilitado (*manual update bit*) y con cada ciclo de un reloj “generado” a partir del reloj primario se decrementa el temporizador
 - Cuando TCNT* se iguala a TCMP* se aserta la señal de salida
 - Dos modos de funcionamiento: auto-reload y one-shot
 - Cuando TCNT* = 0 se genera una interrupción (bits 8 a 13 de INTPND)

Funcionamiento temporizador



Registros temporizador

Register	Address	R/W	Description	Reset Value
TCON	0x01D50008	R/W	Timer control register	0x00000000

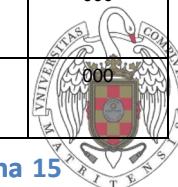
TCON	Bit	Description	initial state
Timer 5 auto reload on/off	[26]	This bit determines auto reload on/off for Timer 5. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 5 manual update (note)	[25]	This bit determines the manual update for Timer 5. 0 = No operation 1 = Update TCNTB5	0
Timer 5 start/stop	[24]	This bit determines start/stop for Timer 5. 0 = Stop 1 = Start for Timer 5	0
Timer 4 auto reload on/off	[23]	This bit determines auto reload on/off for Timer 4. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 4 output inverter on/off	[22]	This bit determines output inverter on/off for Timer 4. 0 = Inverter off 1 = Inverter on for TOUT4	0
Timer 4 manual update (note)	[21]	This bit determines the manual update for Timer 4. 0 = No operation 1 = Update TCNTB4, TCMPB4	0
Timer 4 start/stop	[20]	This bit determines start/stop for Timer 4. 0 = Stop 1 = Start for Timer 4	0
Timer 3 auto reload on/off	[19]	This bit determines auto reload on/off for Timer 3. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 3 output inverter on/off	[18]	This bit determines output inverter on/off for Timer 3. 0 = Inverter off 1 = Inverter on for TOUT3	0
Timer 3 manual update (note)	[17]	This bit determine manual update for Timer 3. 0 = No operation 1 = Update TCNTB3, TCMPB3	0
Timer 3 start/stop	[16]	This bit determines start/stop for Timer 3. 0 = Stop 1 = Start for Timer 3	0
Timer 2 auto reload on/off	[15]	This bit determines auto reload on/off for Timer 2. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 2 output inverter on/off	[14]	This bit determines output inverter on/off for Timer 2. 0 = Inverter off 1 = Inverter on for TOUT2	0
Timer 2 manual update (note)	[13]	This bit determines the manual update for Timer 2. 0 = No operation 1 = Update TCNTB2, TCMPB2	0
Timer 2 start/stop	[12]	This bit determines start/stop for Timer 2. 0 = Stop 1 = Start for Timer 2	0

Register	Address	R/W	Description	Reset Value
TCFG0	0x01D50000	R/W	Configures the three 8-bit prescalers	0x00000000

TCFG0	Bit	Description	Initial State
Dead zone length	[31:24]	These 8 bits determine the dead zone length. The 1 unit time of the dead zone length is equal to the 1 unit time of timer 0.	0x00
Prescaler 2	[23:16]	These 8 bits determine prescaler value for Timer 4 & 5	0x00
Prescaler 1	[15:8]	These 8 bits determine prescaler value for Timer 2 & 3	0x00
Prescaler 0	[7:0]	These 8 bits determine prescaler value for Timer 0 & 1	0x00

Register	Address	R/W	Description	Reset Value
TCFG1	0x01D50004	R/W	6-MUX & DMA mode selecton register	0x00000000

TCFG1	Bit	Description	Initial State
DMA mode	[27:24]	Select DMA request channel 0000 = No select (all interrupt) 0001 = Timer0 0010 = Timer1 0011 = Timer2 0100 = Timer3 0101 = Timer4 0110 = Timer5 0111 = Reserved	000
MUX 5	[23:20]	Select MUX input for PWM Timer5. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = EXTCLK	000
MUX 4	[19:16]	Select MUX input for PWM Timer4. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = TCLK	000
MUX 3	[15:12]	Select MUX input for PWM Timer3. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32	000
MUX 2	[11:8]	Select MUX input for PWM Timer2. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32	000
MUX 1	[7:4]	Select MUX input for PWM Timer1. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32	000
MUX 0	[3:0]	Select MUX input for PWM Timer0. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = 1/32	000



Interrupción pulsador

- Pulsadores asociados a los pines 6 y 7 del puerto G
- Definir estos pines como entradas de interrupciones externas activas a flanco de bajada

PUPG	Bit	Description
PG[7:0]	[7:0]	0: the pull up resistor attached to the corresponding port pin is enabled. 1: the pull up resistor is disabled.

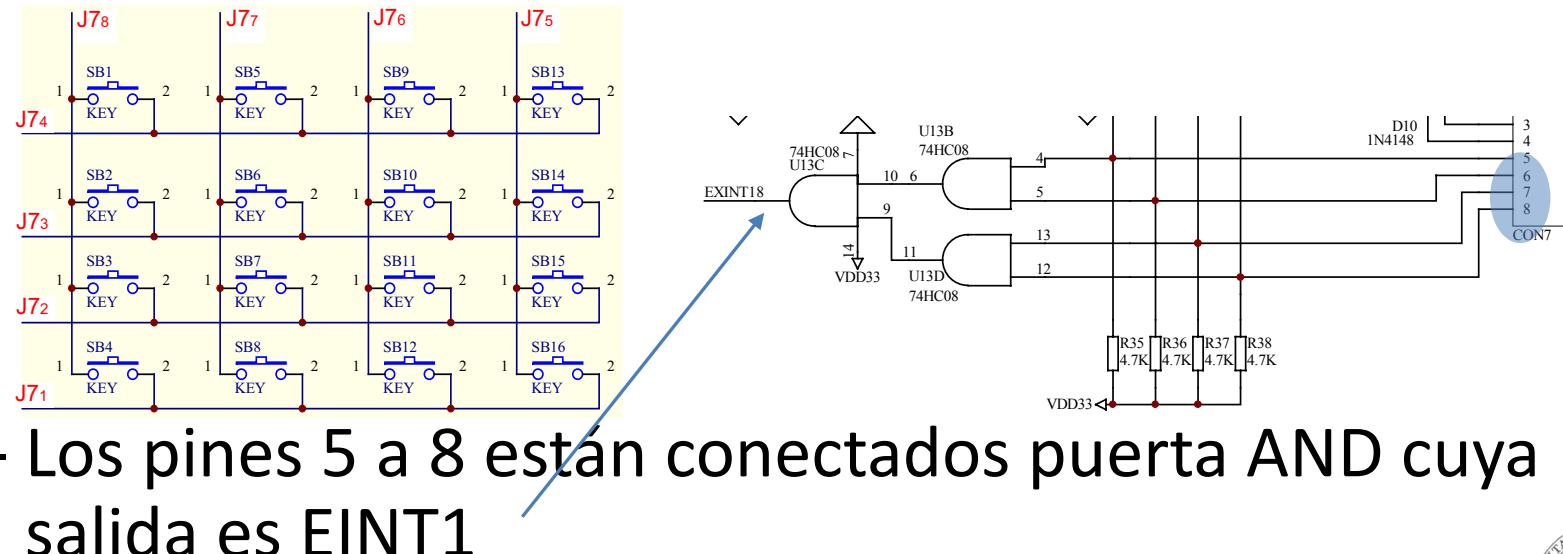
Register	Address	R/W	Description	Reset Value
EXTINT	0x01D20050	R/W	External Interrupt control Register	0x000000

EXTINT	Bit	Description			
EINT7	[30:28]	Setting the signaling method of the EINT7.	000 = Low level interrupt 01x = Falling edge triggered 11x = Both edge triggered	001 = High level interrupt 10x = Rising edge triggered	

PCONG	Bit	Description	EINT6	[26:24]	Description
PG7	[15:14]	00 = Input 10 = IISLRCK	01 = Output 11 = EINT7		Setting the signaling method of the EINT6.
PG6	[13:12]	00 = Input 10 = IISDO	01 = Output 11 = EINT6		000 = Low level interrupt 01x = Falling edge triggered 11x = Both edge triggered
PG5	[11:10]	00 = Input 10 = IISDI	01 = Output 11 = EINT5		001 = High level interrupt 10x = Rising edge triggered
PG4	[9:8]	00 = Input 10 = IISCLK	01 = Output 11 = EINT4		
PG3	[7:6]	00 = Input 10 = nRTS0	01 = Output 11 = EINT3		
PG2	[5:4]	00 = Input 10 = nCTS0	01 = Output 11 = EINT2		
PG1	[3:2]	00 = Input 10 = VD5	01 = Output 11 = EINT1		
PG0	[1:0]	00 = Input 10 = VD4	01 = Output 11 = EINT0		

Gestión del teclado

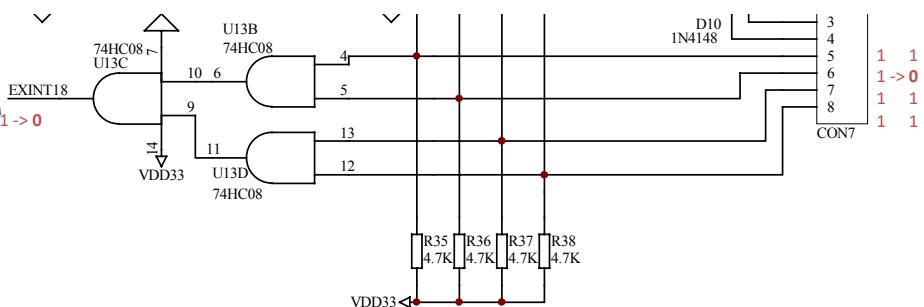
- El teclado consta de 8 pines conectados a un conector llamado J7
 - Los pines 5 a 8 están conectados a resistencias de pull-up (3.3V)
 - Los pines 1 a 4 están a pull-down (0V)



Funcionamiento del teclado

- Al pulsar una tecla se cortocircuita fila y columna de forma que la columna correspondiente toma el valor 0. A consecuencia:
 - La señal EINT1 cambia de 1 a 0. Si se configura el puerto G adecuadamente este cambio provoca una interrupción por la línea IRQ

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



Funcionamiento del teclado

- Al pulsar una tecla se cortocircuita fila y columna de forma que la columna correspondiente toma el valor 0. A consecuencia:
 - RTI debe identificar la tecla enviando una tensión de baja por una línea horizontal dejando el resto a tensión de alta y comprobar si alguna de las líneas verticales toma la tensión de baja.
 - ¿Cómo?
 - Leer direcciones de memoria. A partir del dato leído se puede identificar la tecla pulsada

Dirección	Dato				
	0x7	0xB	0xD	0xE	0xF
0xFD	SB1	SB5	SB9	SB13	-
0xFB	SB2	SB6	SB10	SB14	-
0xF7	SB3	SB7	SB11	SB15	-
0xEF	SB4	SB8	SB12	SB16	-

- Ojo: la lectura de la dirección de memoria provocará una nueva interrupción. Al final de la RTI hay que borrar las interrupciones la interrupción asociada a EINT1 generada espuriamente durante la RTI.

ISR en C

■ Pasos

1. Definir la ISR usando la directiva `__attribute__((interrupt ("TYPE")))` donde **TYPE** puede ser **IRQ, FIQ, ABORT, UNDEF, SWI**
 - De esta forma el compilador sabe qué prólogo y epílogo debe utilizar, los registros visibles, ...
2. Registrar la ISR. Esto es, ubicar la dirección de comienzo de la ISR en la posición de memoria asignada.
 - Si es autovectorizada solo hay que registrar en la dirección (`_ISR_STARTADDRESS+0x18`)
 - Si es vectorizada hay que registrar en la dirección asociada a la interrupción específica. Por ejemplo, la ISR asociada al TIMER0 hay que registrarla en la dirección
`(_ISR_STARTADDRESS+0x54)`

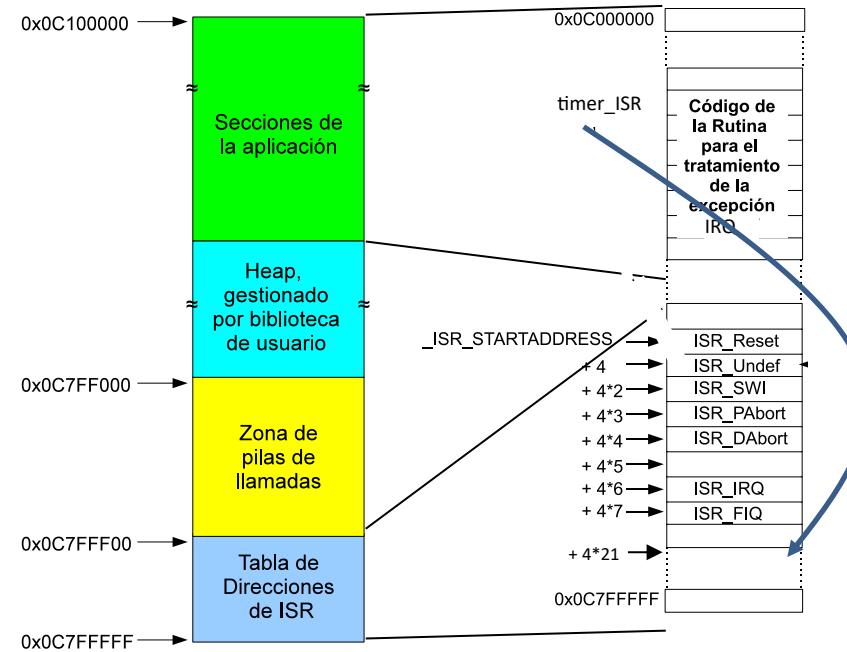
■ Identificación

1. Si es autovectorizada, la ISR debe identificar la fuente leyendo el registro `I_ISPR`
2. Si es vectorizada, cada ISR está asociada a una fuente salvo EINT4/5/6/7 Y UERR0/1. Para ellas la ISR debe identificar la fuente en otro registro.

ISR en C. Ejemplo

- Definir y registrar interrupción vectorizada asociada al timer0

1. Definir: `void timer_ISR(void) __attribute__ ((interrupt ("IRQ")));`
2. Registrar: `pISR_TIMER0 = (unsigned) timer_ISR;`



ISR en P2. Setup

- Las fuentes de interrupción de tipo IRQ son:
 - Timer0
 - Pulsadores
 - Teclado
- Configuración de interrupciones E/S
 - Timer0
 - Ajustar prescaler
 - Ajustar divider
 - Definir valor inicial y modo de cuenta
 - Pulsadores
 - Asociar pulsación interruptor con interrupciones EINT
 - Interrupción por flanco de bajada
 - Habilitar pull-up
 - Teclado
 - Asociar pulsación interruptor con interrupciones EINT
 - Interrupción por flanco de bajada
 - Habilitar pull-up
- Configuración interrupciones ARM
 - Habilitar IRQ y declararlas como vectorizadas
 - Deshabilitar FIQ
 - Asociar interrupciones EINT4567, TIMERO y EINT1 con el modo IRQ
 - Habilitar las interrupciones EINT4567, TIMERO y EINT1

ISR en P2. Identificación

- ¿Cómo se identifican las fuentes de interrupción?
 - Definir las tres ISR usando __attribute__ (ejemplo transparencias anteriores)
 - Registrar la ISR con el vector asociado (ejemplo transparencia anterior)
- Dentro de la ISR asociada a los botones
 - Leer registro de interrupciones externas pendientes (**0x1d20054**)
 - Botón 1 en EXTINTPND[2]
 - Botón 2 en EXTINTPND[3]
- Dentro de la ISR asociada al teclado
 - Leer las direcciones de memoria asociada a cada fila. En función del valor returnedo, decodificar la tecla pulsada.
 - Borrar la interrupción espuria que se genera durante la lectura.
- ISR TIMER0 no requiere identificación adicional