

Capítulo 4.

Jerarquía de memoria y sistemas de almacenamiento en las familias TMS320C3x y TMS320C6000.

1. RANGO DE DIRECCIONES Y MAPA DE MEMORIA	3
1.1 TMS320C3x.....	3
1.2 TMS320C6000.....	9
2. INICIALIZACIÓN Y VECTORES DE INTERRUPCIÓN.....	14
2.1 TMS320C3x.....	14
2.1.1 Fuentes de interrupción	15
2.1.2 Servicio de interrupciones	16
2.2 TMS320C6000.....	19
3. ARRANQUE DEL PROCESADOR.....	23
3.1 TMS320C3x.....	23
3.1.1 Carga desde Memoria Externa.....	24
3.1.2 Carga desde el puerto serie.....	25
3.1.3 TMS320C32: boot-loader flexible.....	25
3.2 TMS320C6000.....	26
4. MEMORIA CACHE´	30
4.1 TMS320C3x.....	30
4.2 TMS320C6000.....	34
4.2.1 Caché de instrucciones de primer nivel L1P.....	35
4.2.2 Caché de datos de primer nivel L1D	37
4.2.3 Caché de segundo nivel L2.....	40
4.2.4 Configuración de la caché	42
5. PERIFÉRICO DMA	44
5.1 TMS320C3x.....	46
5.1.1 Registros de control y configuración.....	46
5.1.2 Modos de operación: sincronización de los eventos DMA.....	49
5.1.3 Método de configuración.....	50
5.2 TMS320C6000.....	50
5.2.1 Modos de funcionamiento básico	52
5.2.2 Gestión de los accesos al bus del controlador DMA	58
5.2.3 El controlador DMA mejorado del TMS320C6000: EDMA.....	61
5.2.4 Transferencias QDMA	68

1. RANGO DE DIRECCIONES Y MAPA DE MEMORIA

1.1 TMS320C3x

El espacio total de memoria que es capaz de direccionar la familia TMS320C3x de Texas Instruments, es de 16M (unos 16 millones) de palabras de 32 bits. Dentro de este espacio total que es capaz de direccionar el DSP se incluyen las zonas de memoria internas de las que dispone el sistema (memoria RAM interna, ROM interna y registros de configuración de periféricos internos). En la figura 4.1, se muestran los buses y la estructura de acceso a memoria del sistema.

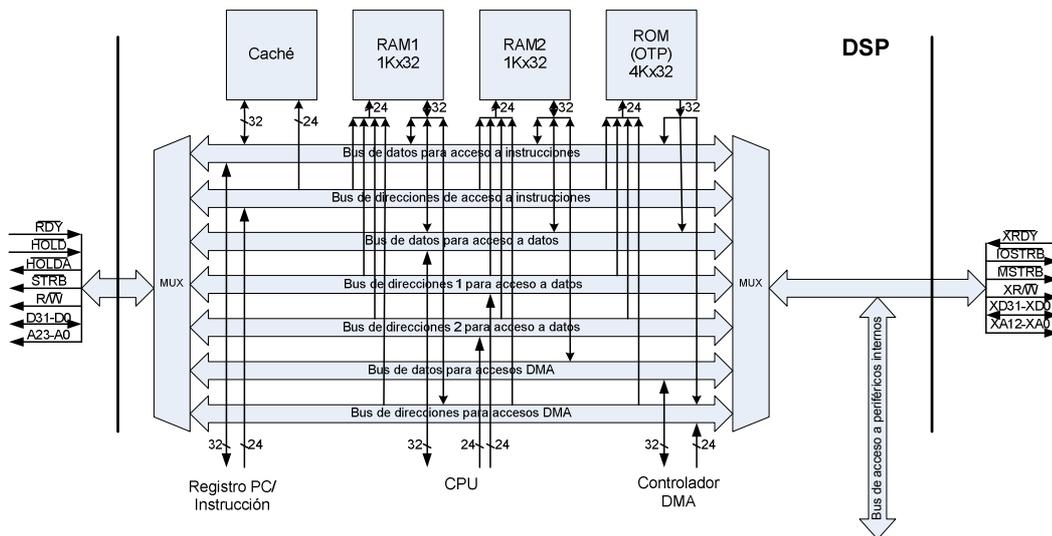


Fig. 4.1. Acceso a memoria: Buses del sistema

El sistema dispone de dos conjuntos de líneas para acceso al exterior, el denominado bus principal (formado por un bus de datos de 32 bits, un bus de direcciones de 24 bits y un bus de control de 5 bits) y el bus de expansión (formado por un bus de datos de 32 bits, un bus de direcciones de 13 bits y un bus de control de 4 bits). En realidad, el único DSP que dispone del doble sistema de buses para el acceso al exterior es el TMS320C30, el resto no dispone de bus de expansión.

En las figuras siguientes se muestran los mapas de memoria del TMS320C30, TMS320C31, TMS320C32 y TMS320C33. Se observa que cada DSP tiene dos posibles mapas de memoria. La diferencia estriba en la posibilidad de acceso o no a la ROM interna del sistema. La forma de seleccionar uno u otro mapa de memoria es imponiendo un nivel alto o bajo en el pin *MCBL/MP* de entrada a DSP. Esta característica define los dos modos de operación del DSP:

- **Microprocesador:** El sistema arranca autovectorizando desde la posición 000000H, que debe coincidir en este caso con dispositivos externos de memoria dispuestos (rutados y colocados en el mapa de memoria del sistema microprocesador) y programados por el usuario.

- **Microcomputador:** El sistema arranca autovectorizando desde la posición 000000H que en este caso coincide con la ROM interna del sistema. Obsérvese que para que el sistema TMS320C30 funcione, será preciso que el fabricante, Texas Instruments, programe en la fábrica, antes de entregar el DSP al usuario, la ROM interna de tipo OTP (*One Time Programmable*). Este proceso requiere que el fabricante genere máscaras especiales para programar la ROM interna con el programa suministrado por un usuario particular. Esto supone el encarecimiento del coste de fabricación del DSP, de forma que sólo será factible para usuarios que requieran un volumen muy elevado de DSPs, nunca para un usuario que sólo desee un DSP programado de fábrica. Para el TMS320C31, TMS320C32 y TMS320C33 el caso es diferente dado que la ROM interna ya está programada de fábrica (incorpora un programa denominado *Boot-Loader*) y no es posible contactar con el fabricante para su reprogramación (ROM de tipo OTP).

	64 Words	000000h	-----	64 Words	000000h
	Reservado a vectores de interrupción	00003Fh		Reservado a vectores de interrupción	00003Fh
		000040h		-----	000040F
				ROM interna	000FFFh
	STRB=0			-----	001000h
		7FFFFFFh		STRB=0	7FFFFFFh

MSTRB=0	800000h	MSTRB=0	800000h
8Kw	801FFFh	8Kw	801FFFh
Reservado	802000h	Reservado	802000h
8Kw	803FFFh	8Kw	803FFFh
IOSTRB=0	804000h	IOSTRB=0	804000h
8Kw	805FFFh	8Kw	805FFFh
Reservado	806000h	Reservado	806000h
8Kw	807FFFh	8Kw	807FFFh
Registros de configuración de periféricos internos	808000h	Registros de configuración de periféricos internos	808000h
6Kw	8097FFh	6Kw	8097FFh
RAM Block 0	809800h	RAM Block 0	809800h
1Kw	809BFFh	1Kw	809BFFh
RAM Block 1	809C00h	RAM Block 1	809C00h
1Kw	809FFFh	1Kw	809FFFh
STRB=0	80A000h	STRB=0	80A000h
	FFFFFFh		FFFFFFh
Modo Microprocesador		Modo Microcomputador	
MCBL/MP=0		MCBL/MP=1	

Fig. 4.2. Mapas de memoria del TMS320C30

64 Words	000000h		000000h
Reservado a vectores de interrupción	00003Fh	ROM Interna	
	000040h	Boot Loader	
			000FFFh

		Boot1	001000h
STRB=0			
		Boot2	
	7FFFFFFh	STRB=0	7FFFFFFh
Reservado	800000h	Reservado	800000h
32Kw	807FFFh	32Kw	807FFFh
Registros de configuración de periféricos internos	808000h	Registros de configuración de periféricos internos	808000h
	8097FFh		8097FFh

6Kw		6Kw	
RAM Block 0	809800h	RAM Block 0	809800h
1Kw	809BFFh	1Kw	809BFFh
RAM Block 1	809C00h	RAM Block 1	809C00h
1Kw	809FFFh	1Kw	809FFFh
	80A000h	STRB=0	80A000h
STRB=0		Boot3	
	FFFFFFh		FFFFFFh
Modo Microprocesador		Modo Microcomputador	
MCBL/MP=0		MCBL/MP=1	

Fig. 4.3. Mapas de memoria del TMS320C31

64 Words	000000h	64 Words	000000h
Reservado a vectores de interrupción	00003Fh	Reservado a vectores de interrupción	000FFFh
	000040h	BOOT1	001000h
STRB0		STRB0	
	7FFFFFFh		7FFFFFFh
Reservado	800000h	Reservado	800000h
32Kw	807FFFh	32Kw	807FFFh
Registros de configuración de periféricos internos	808000h	Registros de configuración de periféricos internos	808000h
6Kw	8097FFh	6Kw	8097FFh
Reservado	809800h	Reservado	809800h
26Kw	80FFFFh	26Kw	80FFFFh
	810000h	BOOT2	810000h
IOSTRB		IOSTRB	
128Kw	82FFFFh	128Kw	82FFFFh
Reservado	830000h	Reservado	830000h
314.5Kw	87FDFFh	314.5Kw	87FDFFh
RAM Block 0	87FE00h	RAM Block 0	87FE00h
256w	87FEFFh	256w	87FEFFh
RAM Block 1	87FF00h	RAM Block 1	87FF00h

	87FFFh		87FFFh
STRB0	88000h	STRB0	88000h
512Kw	8FFFFh	512Kw	8FFFFh
	90000h	BOOT3	90000h
STRB1		STRB1	
7.168Mw	FFFFFFh	7.168Mw	FFFFFFh
Modo Microprocesador		Modo Microcomputador	
MCBL/MP=0		MCBL/MP=1	

Fig. 4.4. Mapas de memoria del TMS320C32

64 Words	00000h		00000h
Reservado a vectores de interrupción	00003Fh	ROM Interna	
	000040h	Boot Loader	
			000FFFh
STRB=0		001000h
		Boot1	
	7FFFFFFh	STRB=0	
Bloque 2 de RAM	80000h	Boot2	
Interna			7FFFFFFh
16 Kw	803FFFh	Bloque 2 de RAM	80000h
Bloque 3 de RAM	80400h	Interna	
Interna		16KW	803FFFh
16 Kw	807FFFh	Bloque 3 de RAM	80400h
Registros de configuración de periféricos internos	80800h	Interna	
		16KW	807FFFh
	8097FFh	Registros de configuración de periféricos internos	80800h
	809800h		
RAM Block 0			8097FFh
1Kw		RAM Block 0	
	809BFFh	1Kw	
	809C00h		809BFFh
RAM Block 1			809C00h
1Kw		RAM Block 1	
	809FFFh	1Kw	
			809FFFh

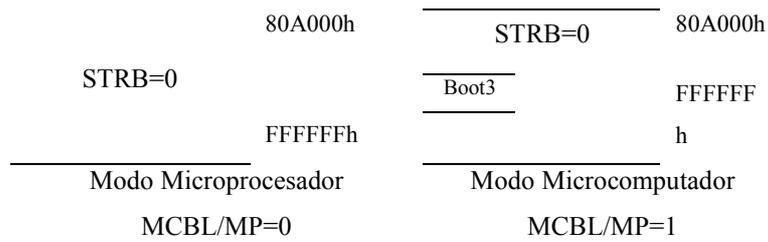


Fig. 4.5. Mapas de memoria del TMS320C33

Registros del Controlador DMA0 16 Words	808000h
Reservado o registros del Controlador DMA1 16 Words	808010h
Registros del Timer 0 16 Words	808020h
Registros del Timer 1 16 Words	808030h
Registros del Puerto Serie 0 16 Words	808040h
Registros del Puerto Serie 1 16 Words	808050h
Registros de control del Bus Principal y de Expansión 16 Words	808060h
Reservado	808070h
	8097FFh

Fig. 4.6. Mapa de memoria de los registros de control de los periféricos internos de los DSPs de la familia TMS320C3x

1.2 TMS320C6000

El espacio total de memoria que es capaz de direccionar la familia TMS320C6000 de Texas Instruments, es de 4Gbytes correspondientes a los 32 bits de direcciones internos. La tabla 4.1 muestra el mapa de memoria de los dispositivos TMS320C6201/C6204/C6205/C6701, la tabla 4.2 el del TMS320C6203 y la tabla 4.3 el del TMS320C6203. Los dispositivos TMS320C6201/C6701 disponen de 5 pines de entrada (BOOTMODE) que permiten seleccionar el mapa de memoria y el proceso de arranque. En el caso de los TMS320C6202/C6203/C6204 se sustituyen por la captura en latches de los cinco bits menos significativos del bus de datos de expansión (XD_{4,0}) durante el flanco de subida de la señal de entrada al DSP denominada reset. Análogamente, el TMS320C6205 usa los pines del bus de datos de la interfaz externa EMIF para determinar la configuración de arranque (ED_{4,0}) y otras configuraciones del dispositivo (ED_{31,5}) El TMS320C6211 dispone sólo de un mapa de memoria, pero el proceso de arranque se selecciona mediante los pines HD_{4,3} del periférico HPI en el flanco de subida de la señal reset.

Los dos modos de operación de las tablas (mapa 0 y mapa 1) corresponden a los modos de ejecución directa (mapa 0) y *boot* (mapa 1). En el modo ejecución directa, el programa comienza en la dirección 0h de memoria externa, mientras que en modo *boot* el programa se carga desde memoria externa, o desde un dispositivo externo, antes de comenzar la CPU la ejecución de las instrucciones del usuario ubicadas en la posición de memoria 0h, ahora perteneciente a memoria RAM interna.

Direcciones	Tamaño	Mapa 0	Mapa 1
00000000-0000FFFF	64K	Interfaz de memoria externa CE0	RAM de programa interna
00010000-003FFFFF	4M-64K	Interfaz de memoria externa CE0	Reservado
00400000-00FFFFFF	12M	Interfaz de memoria externa CE0	Interfaz de memoria externa CE0
01000000-013FFFFF	4M	Interfaz de memoria externa CE1	Interfaz de memoria externa CE0
01400000-0140FFFF	64K	RAM de programa interna	Interfaz de memoria externa CE1
01410000-017FFFFF	4M-64K	Reservado	Interfaz de memoria externa

			CE1
01800000-0183FFFF	256K	Registros EMIF	
01840000-0187FFFF	256K	Registros del controlador DMA	
01880000-018BFFFF	256K	Registros HPI / XBUS (sólo TMS320C6204)	
018C0000-018FFFFF	256K	Registros McBSP0	
01900000-0193FFFF	256K	Registros McBSP1	
01940000-0197FFFF	256K	Registros del temporizador 0	
01980000-019BFFFF	256K	Registros del temporizador 1	
019C0000-019FFFFF	256K	Registros del selector de interrupciones	
01A00000-01A3FFFF	256K	Reservado	
01A40000-01A8FFFF	320K	Reservado / Registros PCI (sólo TMS320C6205)	
01A90000-01FFFFFF	6M-576K	Reservado	
02000000-02FFFFFF	16M	Interfaz de memoria externa CE2	
03000000-03FFFFFF	16M	Interfaz de memoria externa CE3	
04000000-3FFFFFFF	1G-64M	Reservado	
40000000-4FFFFFFF	256M	Reservado / Bus de expansión XCE0 (sólo TMS320C6205)	
50000000-5FFFFFFF	256M	Reservado / Bus de expansión XCE1 (sólo TMS320C6205)	
60000000-6FFFFFFF	256M	Reservado / Bus de expansión XCE2 (sólo TMS320C6205)	
70000000-7FFFFFFF	256M	Reservado / Bus de expansión XCE3 (sólo TMS320C6205)	
80000000-8000FFFF	64K	RAM interna de datos	
80010000-FFFFFFF	2G-64K	Reservado	

Tabla 4.1. Mapa de memoria de los dispositivos TMS320C6201/6204/6205/6701.

Direcciones	Tamaño	Mapa 0	Mapa 1
00000000-0003FFFF	256K	Interfaz de memoria externa CE0	RAM de programa interna
00040000-003FFFFF	4M-256K	Interfaz de memoria externa CE0	Reservado
00400000-00FFFFFF	12M	Interfaz de memoria externa CE0	Interfaz de memoria externa CE0
01000000-013FFFFF	4M	Interfaz de memoria externa CE1	Interfaz de memoria externa CE0
01400000-0143FFFF	256K	RAM de programa interna	Interfaz de memoria externa CE1
01440000-017FFFFF	4M-256K	Reservado	Interfaz de memoria externa CE1
01800000-0183FFFF	256K	Registros EMIF	
01840000-0187FFFF	256K	Registros del controlador DMA	
01880000-018BFFFF	256K	Registros del bus de expansión	
018C0000-018FFFFF	256K	Registros McBSP0	
01900000-0193FFFF	256K	Registros McBSP1	
01940000-0197FFFF	256K	Registros del temporizador 0	
01980000-019BFFFF	256K	Registros del temporizador 1	
019C0000-019C01FF	512	Registros del selector de interrupciones	
019C0200-019CFFFF	256K-512	Registros de modos de bajo consumo	
01A00000-01A3FFFF	256K	Reservado	
01A40000-01A7FFFF	256K	Registros McBSP2	
01A80000-01FFFFFF	5.5M	Reservado	
02000000-02FFFFFF	16M	Interfaz de memoria externa CE2	
03000000-03FFFFFF	16M	Interfaz de memoria externa CE3	

04000000–3FFFFFFF	1G-64M	Reservado
40000000–4FFFFFFF	256M	Bus de expansión XCE0
50000000–5FFFFFFF	256M	Bus de expansión XCE1
60000000–6FFFFFFF	256M	Bus de expansión XCE2
70000000–7FFFFFFF	256M	Bus de expansión XCE3
80000000–8001FFFF	128K	RAM interna de datos
80020000–FFFFFFF	2G-128K	Reservado

Tabla 4.2. Mapa de memoria del dispositivo TMS320C6202.

Direcciones	Tamaño	Mapa 0	Mapa 1
00000000–0005FFFF	384K	Interfaz de memoria externa CE0	RAM de programa interna
00060000–003FFFFFFF	4M–384K	Interfaz de memoria externa CE0	Reservado
00400000–00FFFFFFF	12M	Interfaz de memoria externa CE0	Interfaz de memoria externa CE0
01000000–013FFFFFFF	4M	Interfaz de memoria externa CE1	Interfaz de memoria externa CE0
01400000–0145FFFF	384K	RAM de programa interna	Interfaz de memoria externa CE1
01460000–017FFFFFFF	4M–384K	Reservado	Interfaz de memoria externa CE1
01800000–0183FFFF	256K	Registros EMIF	
01840000–0187FFFF	256K	Registros del controlador DMA	
01880000–018BFFFF	256K	Registros del bus de expansión	
018C0000–018FFFFF	256K	Registros McBSP0	
01900000–0193FFFF	256K	Registros McBSP1	
01940000–0197FFFF	256K	Registros del temporizador 0	
01980000–019BFFFF	256K	Registros del temporizador 1	
019C0000–019C01FF	512	Registros del selector de interrupciones	
019C0200–019FFFFFFF	256K–512	Registros de modos de bajo consumo	
01A00000–01A3FFFF	256K	Reservado	
01A40000–01A7FFFF	256K	Registros McBSP2	
01A80000–01FFFFFFF	5.5M	Reservado	
02000000–02FFFFFFF	16M	Interfaz de memoria externa CE2	
03000000–03FFFFFFF	16M	Interfaz de memoria externa CE3	
04000000–3FFFFFFF	1G–64M	Reservado	
40000000–4FFFFFFF	256M	Bus de expansión XCE0	
50000000–5FFFFFFF	256M	Bus de expansión XCE1	
60000000–6FFFFFFF	256M	Bus de expansión XCE2	
70000000–7FFFFFFF	256M	Bus de expansión XCE3	
80000000–8007FFFF	512K	RAM interna de datos	
80080000–FFFFFFF	2G–512K	Reservado	

Tabla 4.3. Mapa de memoria del dispositivo TMS320C6203.

Direcciones	Tamaño	Descripción
00000000–0000FFFF	64K	RAM interna (L2)
00010000–017FFFFFFF	24M–64K	Reservado
01800000–0183FFFF	256K	Registros EMIF

01840000–0187FFFF	256K	Registros L2
01880000–018BFFFF	256K	Registros HPI
018C0000–018FFFFF	256K	Registros McBSP 0
01900000–0193FFFF	256K	Registros McBSP 1
01940000–0197FFFF	256K	Registros del temporizador 0
01980000–019BFFFF	256K	Registros del temporizador 1
019C0000–019FFFFF	256K	Registros selector de interrupciones
01A00000–01A3FFFF	256K	Registros EDMA RAM y EDMA
01A40000–01FFFFFF	6M-516K	Reservado
02000000–02000033	52	Registros QDMA
02000034–2FFFFFFF	736M–52	Reservado
30000000–3FFFFFFF	256M	Datos McBSP 0/1
40000000–7FFFFFFF	1G	Reservado
80000000–8FFFFFFF	256M	EMIF CE0
90000000–9FFFFFFF	256M	EMIF CE1
A0000000–AFFFFFFF	256M	EMIF CE2
B0000000–BFFFFFFF	256M	EMIF CE3
C0000000–FFFFFFF	1G	Reservado

Tabla 4.4. Mapa de memoria del dispositivo TMS320C6211.

La tabla 4.5 y la tabla 4.6 muestran los mapas de memoria del restos de dispositivos pertenecientes a la generación TMS320C67x. Igual que en el TMS320C6211, el proceso de arranque del TMS320C6711 y del TMS320C6713 se configura mediante los pines HD_{4,3} del periférico HPI, en el flanco de subida de la señal reset. El TMS320C6712 utiliza los pines de entrada BOOTMODE.

Direcciones	Tamaño	Descripción
00000000–0000FFFF	64K	RAM interna (L2)
00010000–017FFFFF	24M–64K	Reservado
01800000–0183FFFF	256K	Registros EMIF
01840000–0187FFFF	256K	Registros L2
01880000–018BFFFF	256K	Registros HPI
018C0000–018FFFFF	256K	Registros McBSP 0
01900000–0193FFFF	256K	Registros McBSP 1
01940000–0197FFFF	256K	Registros del temporizador 0
01980000–019BFFFF	256K	Registros del temporizador 1
019C0000–019C01FF	512	Registros selector de interrupciones
019C0200–019C0203	4	Registros de configuración del dispositivo (sólo TMS320C671xC/C671xD)
019C0204–019FFFFF	256K-516	Reservado
01A00000–01A3FFFF	256K	Registros EDMA RAM y EDMA
01A40000–01AFFFFFFF	768K	Reservado
01B00000–01B03FFF	16K	Registros GPIO (solo TMS320C671xC/C671xD)
01B04000–01B7BFFF	480K	Reservado
01B7C000–01B7DFFF	8K	Registros controlador PLL (sólo TMS320C671xC/C671xD)
01B7E000–01FFFFFF	4M+520K	Reservado
02000000–02000033	52	Registros QDMA

02000034–2FFFFFFF	736M–52	Reservado
30000000–33FFFFFF	64M	Datos McBSP 0 /Bus de datos de periféricos
34000000–37FFFFFF	64M	Datos McBSP 1 /Bus de datos de periféricos
38000000–3BFFFFFF	64M	Reservado
3C000000–7FFFFFFF	1G+64M	Reservado
80000000–8FFFFFFF	256M	EMIF CE0
90000000–9FFFFFFF	256M	EMIF CE1
A0000000–AFFFFFFF	256M	EMIF CE2
B0000000–BFFFFFFF	256M	EMIF CE3
C0000000–FFFFFFF	1G	Reservado

Tabla 4.5. Mapa de memoria del dispositivo TMS320C6711/6712.

Direcciones	Tamaño	Descripción
00000000–002FFFFF	192K	RAM interna (L2)
00030000–0003FFFF	64K	RAM interna/Cache
00040000–017FFFFF	24M-256K	Reservado
01800000–0183FFFF	256K	Registros EMIF
01840000–0185FFFF	128K	Registros L2
01860000–0187FFFF	128K	Reservado
01880000–018BFFFF	256K	Registros HPI
018C0000–018FFFFF	256K	Registros McBSP 0
01900000–0193FFFF	256K	Registros McBSP 1
01940000–0197FFFF	256K	Registros del temporizador 0
01980000–019BFFFF	256K	Registros del temporizador 1
019C0000–019C01FF	512	Registros selector de interrupciones
019C0200–019C0203	4	Registros de configuración del dispositivo
019C0204–019FFFFF	256K-516	Reservado
01A00000–01A3FFFF	256K	Registros EDMA RAM y EDMA
01A40000–01AFFFFF	768K	Reservado
01B00000–01B03FFF	16K	Registros GPIO
01B04000–01B3FFFF	240K	Reservado
01B40000–01B43FFF	16K	Registros I2C0
01B44000–01B47FFF	16K	Registros I2C1
01B48000–01B4BFFF	16K	Reservado
01B4C000–01B4FFFF	16K	Registros McASP0
01B50000–01B53FFF	16K	Registros McASP1
01B54000–01B7BFFF	160K	Reservado
01B7C000–01B7DFFF	8K	Registros PLL
01B7E000–01BBFFFF	264K	Reservado
01BC0000–01BFFFFF	256K	Registros emulación
01C00000–01FFFFFF	4M	Reservado
02000000–02000033	52	Registros QDMA
02000034–02FFFFFF	16M - 52	Reservado
03000000–2FFFFFFF	720M	Reservado
30000000–33FFFFFF	64M	Datos McBSP0
34000000–37FFFFFF	64M	Datos McBSP1
38000000–3BFFFFFF	64M	Reservado
3C000000–3C0FFFFF	1M	Datos McASP0
3C100000–3C1FFFFF	1M	Datos McASP1

3C200000–7FFFFFFF	1G+62M	Reservado
80000000–8FFFFFFF	256M	EMIF CE0
90000000–9FFFFFFF	256M	EMIF CE1
A0000000–AFFFFFFF	256M	EMIF CE2
B0000000–BFFFFFFF	256M	EMIF CE3
C0000000–FFFFFFF	1G	Reservado

Tabla 4.6. Mapa de memoria del dispositivo TMS320C6713.

2. INICIALIZACIÓN Y VECTORES DE INTERRUPCIÓN

2.1 TMS320C3x

Independientemente del modo de funcionamiento del DSP, la secuencia de operaciones siguientes a un reset hardware del sistema (aplicación de un 0 lógico en la entrada *RESET* del circuito integrado, reset externo no-enmascarable) es siempre la misma.

El sistema recoge el vector de reset (posición 000000h en el mapa de memoria del DSP) y copia el valor que existe en ese registro (que se ubica en un periférico de memoria externa suponiendo que el sistema funciona en modo microprocesador o en la ROM interna, si el sistema funciona en modo microcomputador) en el registro de contador de programa o PC de la CPU. Se dice por tanto que el sistema tiene el reset autovectorizado a la posición absoluta 000000h del mapa de memoria del DSP, dado que cuando aparece el reset, el sistema accede a la posición 000000h para recoger la dirección absoluta de la primera instrucción a ejecutar.

Al conectar la alimentación al DSP, el estado de los registros es desconocido. La función fundamental del reset, es llevar los registros del sistema a un estado inicial conocido, para lo cual es preciso mantener a nivel bajo (0 lógico) el pin de entrada *RESET* al circuito integrado durante al menos 10 ciclos de reloj del sistema. Un reset genera la siguiente reacción de los registros y líneas de entrada y salida del circuito integrado:

- Accesos al exterior: Las líneas de los buses de dirección y de datos ($D_0...D_{31}$, $A_0...A_{23}$, $XD_0...XD_{31}$, $XA_0...XA_{12}$) se ponen en alta impedancia, así como las líneas de control del tipo de acceso al exterior. (R/W , XR/W). Las líneas $STRB$, $MSTRB$ y $IOSTRB$ se ponen a 1. Los registros de control asociados al bus principal y de expansión se inicializan.
- Las líneas de los periféricos internos (temporizadores y puertos serie) así como las entradas o salidas digitales, $XF0$ y $XF1$ se ponen en alta impedancia. Los periféricos internos al DSP y sus registros de control se inicializan.
- Los registros internos de la CPU se inicializan: $IE=0$, $ST=0$, $IF=0$, $IOF=0$, etc. Al hablar de interrupciones en un microprocesador se distingue entre fuente que genera la interrupción y servicio que provoca.

2.1.1 Fuentes de interrupción

La forma de atención de las interrupciones es análoga al proceso de reset. Los DSPs de la familia TMS320C3x disponen de cuatro fuentes de interrupciones externas ($INT0...INT3$), dos fuentes de interrupción asociadas a los periféricos internos puertos serie síncronos, una fuente asignada a la transmisión y otra a la recepción ($XINT0$, $RINT0$ para el puerto serie 0 y $XINT1$, $RINT1$ para el puerto serie 1 que no está disponible en el TMS320C31, TMS320C32 y TMS320C33), así como una fuente de interrupción asociada a cada uno de los periféricos internos restantes, temporizadores y controlador de DMA ($TINT0$, $TINT1$ y $DINT$).

El esquema de generación de una interrupción en un DSP de esta familia, se muestra en las figuras 4.7 y 4.8. Las interrupciones externas ($INTn$) son activas por nivel bajo y se sincronizan con el ciclo máquina del sistema, $H1$ ó $H3$. Para garantizar que son detectadas, deben estar activas al menos durante 1 ciclo de $H1$ y para ser detectadas sólo una vez, deberán desactivarse antes de dos ciclos de reloj $H1$. Si permanecen activas más de dos ciclos de $H1$, pueden ser atendidas más de una vez (se generan lo que se conoce como interrupciones espúreas o indeseadas).

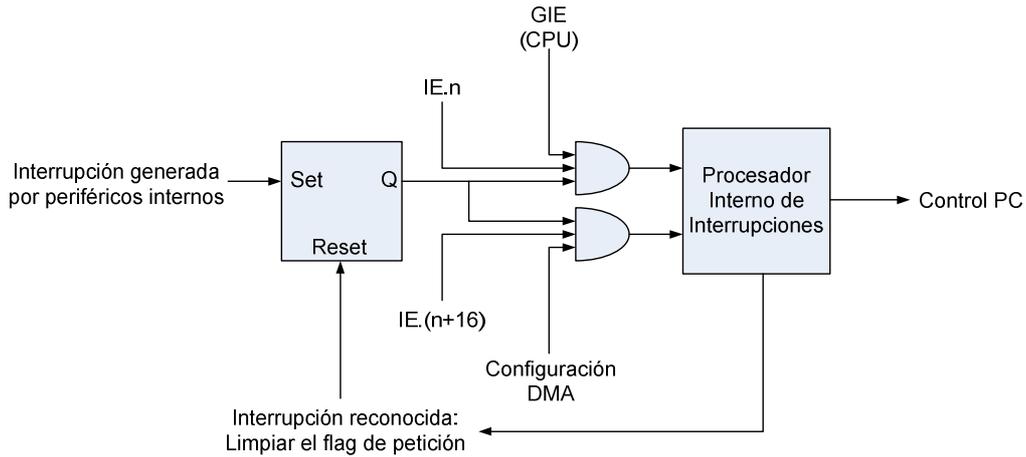


Fig. 4.7. Esquema de generación de interrupciones internas

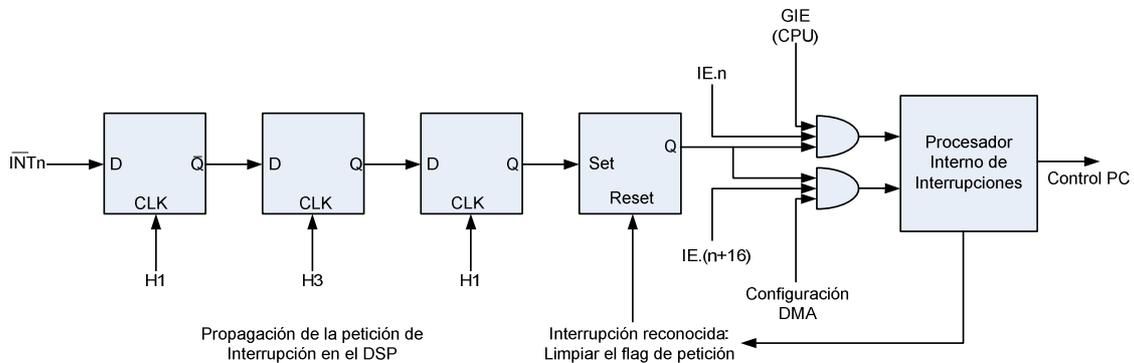


Fig. 4.8. Esquema de generación de interrupciones externas

2.1.2 Servicio de interrupciones

Si la interrupción tiene habilitada la generación de eventos CPU (está a 1 el bit correspondiente del registro IE y además el bit de habilitación global de interrupciones está también a 1) la instrucción que, cuando se detecta la interrupción, está siendo recogida no se ejecuta. La dirección de esta instrucción, registro PC, es almacenada en la dirección apuntada por el registro SP, puntero a la cima de la pila. Inmediatamente después de almacenar en la pila el PC, se pone a cero el bit GIE del registro ST (se desactiva el bit de habilitación global de las interrupciones) y se recoge el vector de interrupción asociado a la interrupción detectada que se almacena en PC. Se empieza, por tanto, a ejecutar la rutina de servicio de la interrupción. Al final de la rutina de

servicio de la interrupción deberá realizarse el retorno del contador de programa a la dirección de la instrucción que no se ejecutó. El retorno normal de la rutina de servicio lo realiza la instrucción RETI, que almacena en el registro PC el dato que aparece en la cima de la pila (deberá coincidir con el valor del registro PC que se almacenó en la pila al comenzar el proceso de atención de la interrupción) y que reactiva el bit GIE.

Si la interrupción tiene habilitada la generación de eventos DMA, está a 1 el bit correspondiente del registro IE y además se permite la sincronización de eventos DMA por interrupciones, sirve para lanzar las transferencias de tipo DMA.

Obsérvese, por tanto, que una interrupción puede generar un evento de tipo CPU (ruptura de la secuencia normal de ejecución en un programa) y otro de DMA. Ambos procesos son totalmente independientes y se ejecutan en paralelo.

Comentarios relacionados con el proceso de atención de interrupciones en la familia TMS320C3x:

- El flag GIE (del registro ST de la CPU) se pone a 0 al comenzar el proceso de atención de una interrupción. Por tanto, y en principio, no se produce anidamiento entre interrupciones. No es posible que, mientras se esté ejecutando la rutina de servicio de una interrupción, al producirse una nueva interrupción, ésta última se atienda antes de terminar de ejecutarse la rutina de servicio de la primera interrupción que apareció. Al terminar de ejecutar la rutina de servicio de la interrupción habrá que devolver el registro PC al valor que tenía antes de la llegada de la interrupción y, si se desea permitir la generación de nuevas interrupciones habrá que volver a activar el bit GIE (instrucciones RETI o RETS). Obsérvese que, dentro de la propia rutina de servicio de las interrupciones, el usuario puede poner vía software, el bit GIE a 1 permitiendo el anidamiento de las interrupciones.
- Los accesos en escritura al registro IF pueden causar fallos en la generación de las interrupciones (si se ha generado una interrupción y está lista para ser latched en IF a la par que se escribe en el mismo registro IF, el proceso de escritura tiene más prioridad, perdiéndose la interrupción generada lo que es

especialmente grave si la interrupción generada lo fue por un periférico interno como la DMA). El fabricante desaconseja el uso del *polling* o generación por software (escritura de 1's lógicos) en los flags del registro IF para provocar interrupciones así como su desactivación por software (escritura de 0's lógicos), aconsejando la generación de la interrupción y dentro de la rutina de servicio la distinción entre interrupción real o espúrea.

- La señal que genere una interrupción externa debe tener un 0 lógico (permanecer activa) un tiempo comprendido entre 1 y 2 ciclos H1 ó H3 (entre 2 y 4 ciclos de reloj del sistema) para garantizar que no se produzca más de una interrupción externa en el sistema (el flags del registro IF correspondiente a la interrupción externa lo limpia automáticamente la CPU generando una señal de reset asíncrono al entrar en la rutina de servicio de la interrupción).
- La atención de las interrupciones provoca que no se ejecute la instrucción que entra en el momento de la aparición de la interrupción en su fase de recogida (Fetch en la estructura de pipeline) siempre que ésta se encuentre en su primer ciclo máquina de recogida. Si la instrucción está en su fase de recogida pero en un ciclo máquina superior al primero (debido, por ejemplo, a la aparición de conflictos en la estructura *pipeline*) se termina de ejecutar antes de atenderse la rutina de servicio de la interrupción. Toda esta casuística, la gestiona la CPU de forma y manera transparente al usuario.
- Posibilidad de generar interrupciones de tipo software (excepciones) además de las interrupciones antes mencionadas. Se pueden generar hasta 32 interrupciones software empleando la instrucción "TRAPcond n", con $0 < n < 31$, que generaría una ruptura programada de la secuencia normal de control de forma análoga a como sucede en el caso de interrupciones de tipo hardware. Este tipo de interrupciones se generan independientemente del valor (0 ó 1) de GIE. Una instrucción de tipo TRAP, provoca además que el flag GIE se ponga a 0 inicialmente (sea o no cierta la condición *cond*, para generación de la excepción software), lo que puede provocar un error si posteriormente se produce un conflicto en la estructura de pipeline, en las fases de lectura (Read) o decodificación (Decode), asociadas a la propia instrucción TRAP, que dejaría

GIE=0 luego de ejecutarse la instrucción, aunque no se cumpla la condición ni se genere la interrupción software. Para evitar este error, el fabricante recomienda la inserción de dos instrucciones NOP antes del uso de la instrucción TRAP.

2.2 TMS320C6000

Existen tres tipos de interrupciones en la familia TMS320C6000 que se diferencian según sus prioridades:

Prioridad	Interrupción
Más alta	Reset
	NMI
	INT4
	INT5
	INT6
	INT7
	INT8
	INT9
	INT10
	INT11
	INT12
	INT13
	INT14
Más baja	INT15

Tabla 4.7. Interrupciones, ordenadas según su prioridad, asociadas a la familia TMS320C6000.

La máxima prioridad corresponde a la señal de RESET, de entrada al DSP y que provoca la inicialización del sistema microprocesador. La siguiente corresponde a la interrupción no enmascarable NMI y, a continuación, vienen las interrupciones enmascarables INT4-INT15. Las fuentes de interrupción correspondientes a las interrupciones INT4-INT15 pueden ser líneas de entrada al DSP, periféricos internos o interrupciones software.

- La interrupción de reset es activa a nivel bajo y debe permanecer así durante, al menos durante 10 ciclos de máquina para que se realice la secuencia de reset, que lleve al sistema a un valor inicial conocido. El paquete de recogida de servicio de la interrupción de reset debe ubicarse en la posición 0h.

- La interrupción no enmascarable, NMI, ocupa el segundo lugar en la escala de prioridades. Para que sea procesada, el bit NMIE (de habilitación de la interrupción no enmascarable) del registro de habilitación de interrupciones debe estar a '1'. Después del reset, este bit vale cero, aunque el usuario siempre tiene la posibilidad de habilitarlo por software (nunca de deshabilitarlo). Además, y para evitar el anidamiento automático de interrupciones, cada vez que la CPU atiende una interrupción no enmascarable pone el bit NMIE a '0'.
- El sistema dispone de doce interrupciones enmascarables, INT4-INT15, que pueden ser asociadas a entradas externas, periféricos internos, interrupciones *software* o bien declaradas no disponibles. Las condiciones que deben cumplirse para que estas interrupciones sean procesadas por la CPU son las siguientes:
 - El bit de habilitación global de interrupciones (GIE) en el registro de control de estado (CSR) debe estar a '1'.
 - El bit NMIE del registro de habilitación de interrupciones (IER) debe valer '1'.
 - El bit de habilitación de la interrupción correspondiente del registro IER debe valer '1'.
 - Si se cumplen las condiciones anteriores, el bit correspondiente en el registro de banderas de interrupción (IFR) se activa. Sin embargo, la rutina de atención de la interrupción no es procesada por la CPU si en dicho registro existe activo otro bit de una interrupción más prioritaria.

En los dispositivos TMS320C620x y TMS320C670x, las señales IACK e INUMx alertan, a cualquier dispositivo *hardware* externo que lo precise, que una interrupción está siendo procesada. La señal IACK indica que la CPU ha comenzado a procesar una interrupción y las señales INUMx (INUM0-INUM3) indican el número de la interrupción (posición del bit en el registro IFR) que está siendo procesada. Estas señales no están disponibles en los dispositivos TMS320C621x y TMS320C671x.

Cuando la CPU comienza la atención de una interrupción, accede a la tabla de servicio de interrupciones (IST). Está formada por 16 paquetes de recogida, cada uno de ellos con 8 instrucciones. Si la rutina de atención es muy reducida (de 8 o menos instrucciones), bastará con uno de estos paquetes para almacenarla. En caso contrario,

habría que insertar en él una instrucción de salto a otra zona de memoria que contenga el resto de la rutina de servicio.

Para cada interrupción existe un espacio reservado en la tabla IST de 32 bytes (espacio correspondiente a un paquete de recogida de instrucciones). Esta tabla es posible reubicarla mediante el registro puntero a la tabla de servicio de interrupciones (ISTP). Los registros de control de las interrupciones en la familia TMS320C6000 se muestran en la tabla 4.8.

Registro	Nombre	Descripción
CSR	Registro de control de estado	Permite habilitar o deshabilitar globalmente las interrupciones
IER	Registro de habilitación de las interrupciones	Permite habilitar las interrupciones de manera individual
IFR	Registro de banderas de petición de interrupciones	Muestra el estado de las interrupciones
ISR	Registro para la activación de las interrupciones	Permite habilitar las banderas del registro IFR manualmente
ICR	Registro para la desactivación de las interrupciones	Permite deshabilitar los flags del registro IFR manualmente
ISTP	Puntero a la tabla de servicio de las interrupciones	Puntero al comienzo de la tabla de servicio de las interrupciones
NRP	Puntero de retorno de interrupciones no enmascarables	Contiene la dirección de retorno cuando se atiende una interrupción no enmascarable.
IRP	Puntero de retorno de interrupciones enmascarables	Contiene la dirección de retorno cuando se atiende una interrupción enmascarable.

Tabla 4.8. Registros de control de las interrupciones de la familia TMS320C6000.

Dado que existen más fuentes de petición de interrupción que entradas de interrupción, es necesario multiplexar aquellas sobre las 12 entradas de interrupción enmascarables que posee la CPU. Las fuentes de petición de interrupción que posee la familia TMS320C6000 así como el código para ser seleccionadas, se detallan en la tabla 4.9.

Fuente	Código	Descripción
DSPINT	00000b	Interrupción HPI
TINT0	00001b	Interrupción del temporizador 0
TINT1	00010b	Interrupción del temporizador 1
SD_INT	00011b	Interrupción del temporizador SDRAM de la EMIF
EXT_INT4	00100b	Interrupción externa 4
EXT_INT5	00101b	Interrupción externa 5
EXT_INT6	00110b	Interrupción externa 6

EXT_INT7	00111b	Interrupción externa 7
DMA_INT0	01000b	Interrupción del canal 0 de la DMA
DMA_INT1	01001b	Interrupción del canal 1 de la DMA
DMA_INT2	01011b	Interrupción del canal 2 de la DMA
DMA_INT3	01011b	Interrupción del canal 3 de la DMA
XINT0	01100b	Interrupción de transmisión del McBSP0
RINT0	01101b	Interrupción de recepción del McBSP0
XINT1	01110b	Interrupción de transmisión del McBSP1
RINT1	01111b	Interrupción de recepción del McBSP1
-	10000b	Reservada
XINT2 PCI_WAKEUP	10001b	Interrupción de transmisión del McBSP2 (sólo TMS320C6202/C6203) y del bus PCI (sólo TMS320C6205)
RINT2 ADMA_HTL	10010b	Interrupción de recepción del McBSP2 (sólo TMS320C6202/C6203) y de la DMA auxiliar (sólo TMS320C6205)
-	10011-11111b	Reservados

Tabla 4.9. Fuentes y códigos de las interrupciones de la familia TMS320C6000.

Para multiplexarlas sobre las 12 entradas INT15-INT4 disponibles, se usan los registros de selección de interrupciones MUXH, MUXL y EXTPOL, según se indica en la tabla 4.10.

Dirección	Registro	Nombre	Descripción		
			Campo	Int.	Por defecto
019C0000h	MUXH	Multiplexor de interrupciones altas. Selecciona qué interrupciones se conectan a INT15-INT10.	MUXH[30:26]	INT15	00010b
			MUXH[25:21]	INT14	00001b
			MUXH[20:16]	INT13	00000b
			MUXH[14:10]	INT12	01001b
			MUXH[9:5]	INT11	01010b
			MUXH[4:0]	INT10	00011b
019C0004h	MUXL	Multiplexor de interrupciones bajas. Selecciona qué interrupciones se conectan a INT9-INT4.	MUXL[30:26]	INT9	01001b
			MUXL[25:21]	INT8	01000b
			MUXL[20:16]	INT7	00111b
			MUXL[14:10]	INT6	00110b
			MUXL[9:5]	INT5	00101b
			MUXL[4:0]	INT4	00100b
019C0008h	EXTPOL	Polaridad de las interrupciones externas. 0 significa flanco de subida y 1 flanco de bajada	Bit 3	EXT7	0
			Bit 2	EXT6	0
			Bit 1	EXT5	0
			Bit 0	EXT4	0

Tabla 4.10. Registros de selección y configuración de interrupciones en la familia TMS320C6000.

Las tablas 4.9 y 4.10 son válidas para los dispositivos TMS320C620x y TMS320C670x. En los dispositivos TMS320C621x y TMS320C671x existen las siguientes variaciones:

- En todos ellos, el código ‘01000b’ de la tabla 4.9. para el canal 0 de la DMA se refiere ahora a la EDMA.
- Los otros tres códigos, ‘01001b’, ‘01010b’ y ‘01011b’ de los otros tres canales de la DMA son reservados en el TMS320C6211, TMS320C6711 y algunas versiones del TMS320C6712. En otras versiones del TMS320C6712 y en el TMS320C6713 se usan para fuentes de interrupción del modo emulación.
- En algunas versiones del TMS320C6712 y en el TMS320C6713, las fuentes de interrupción de las interrupciones externas están asociadas al bloque GPIO (entradas y salidas de propósito general).

3. ARRANQUE DEL PROCESADOR

3.1 TMS320C3x

Todos los DSPs de la familia TMS320C3x disponen de un periférico de ROM interna del tipo OTP. Sin embargo, salvo en el caso del TMS320C30, esta ROM interna ya viene programada de fábrica con un programa denominado boot-Loader. El reset del TMS320C31, TMS320C32 ó TMS320C33, funcionando en modo microcomputador (arranque desde memoria ROM interna), provoca que la CPU del DSP lea la posición 000000h de la ROM interna. En dicha posición se encuentra la dirección de comienzo del programa denominado boot-loader ubicado, a su vez, en la ROM interna del DSP.

El programa boot-loader gestiona el trasvase de bloques de datos (código diseñado por el usuario) enviados desde otro sistema microprocesador y recibidos por el puerto serie de que dispone el DSP o ubicados en algún dispositivo externo de memoria (colocado en el mapa de memoria del DSP en una de las tres posiciones conocidas como BOOT1, BOOT2 ó BOOT3 y normalmente de tipo EPROM) hacia otras posiciones del mapa de memoria del DSP ocupadas por dispositivos de almacenamiento RAM.

El programa es realmente sencillo. Consiste en un bucle que espera una señal, generada por el usuario, que le indique que debe iniciar el trasvase de bloques de datos. Mientras el programa se encuentre en el bucle de espera, la única tarea que realiza el DSP es comprobar si el usuario ha generado alguna de las señales de inicio de trasvase de bloques de datos. El usuario dispone de cuatro líneas de entrada al DSP, INT0 a INT3, como señales para iniciar el trasvase de los bloques de datos. Inicialmente estas señales se usan como líneas digitales de entrada al programa y no como líneas de petición de interrupciones externas, y le sirven al usuario para indicarle al boot-loader hacia donde debe ir a buscar el código que va a cargar en memoria RAM ubicada en el mapa de memoria del DSP. Cuando una de estas líneas es activada por el usuario, el programa boot-loader inicia la carga programada, bien desde memoria externa (desde una dirección de tres posibles, BOOT1, BOOT2 ó BOOT3), bien mediante el puerto serie (el DSP estaría esperando datos que otro sistema digital, que puede ser otro DSP, le envía empleando un protocolo serie síncrono).

3.1.1 Carga desde Memoria Externa

El programa boot-loader accede a un dispositivo de memoria, normalmente de tipo EPROM, ubicado en BOOT1, BOOT2 ó BOOT3 en el mapa de memoria para realizar el trasvase de un bloque de datos (programa diseñado por el usuario) hacia otras posiciones del mapa de memoria del DSP. Antes de trasladar los bloques de código el programa boot-loader precisa que se le pasen una serie de parámetros de configuración o programación del tipo y modo de transferencia que debe realizar (cabecera del bloque de datos, primeros cuatro registros de 32 bits). La cabecera de un bloque de datos se muestra en la tabla 4.11. Estos valores de configuración son, el tipo de dispositivo de memoria al que se accede a recoger el bloque de datos (con un ancho de bus programable de 8, 16 ó 32bits), la palabra de control para la configuración del acceso a dichos periféricos de memoria a través del bus principal, el tamaño del bloque de datos que se desea transferir y, finalmente, la dirección de destino o de ubicación del bloque. Posteriormente aparece la primera instrucción del bloque de datos que se trasladará a otra zona del mapa de memoria del DSP.

Posición	Descripción	Valores Válidos
0	Tipo de memoria boot (8, 16 ó 32)	0x0, 0x10, 0x20
1	Configuración accesos por el bus principal	
2	Tamaño del bloque del programa (blk)	$0 < \text{Tamaño bloque} < 2^{24}$
3	Dirección de destino	Cualquier dirección de 24 bits es válida
4	Código de inicio del programa	Cualquier instrucción o dato de 32 bits

Tabla 4.11. Registros de configuración del programa boot-loader. Cabecera del bloque de código en memoria externa (BOOT1, BOOT2 ó BOOT3)

En definitiva, primero se configura el modo y tipo de transferencia, luego se realiza la misma. Se pueden realizar transferencias de múltiples bloques de datos desde las zonas BOOTs hacia otras zonas en el mapa de memoria del DSP. El boot-loader se termina añadiendo, al final del último bloque transferido, el código 00000000h que representa que el tamaño del siguiente bloque a transferir es 0 (no existe tal bloque en realidad). Obsérvese que, al menos, el primer bloque debe tener longitud mayor de cero para que tenga sentido invocar al programa boot-loader, si no fuese así el sistema arrancaría de forma impredecible.

3.1.2 Carga desde el puerto serie

El boot-loader configura automáticamente el puerto serie en modo de transferencia de datos de ancho 32 bits y en modo de transferencia de bits fixed-burst. Como en el caso de la carga desde memoria externa, se precisa de una cabecera para iniciar y programar la operación de carga. La cabecera, en este caso, sólo incluye el tamaño del bloque y la dirección de destino del bloque de datos.

3.1.3 TMS320C32: boot-loader flexible

El programa boot-loader incluido en la ROM interna del TMS320C32 presenta algunas mejoras respecto al incluido en el TMS320C31 y TMS320C33. Se trata de un programa que, al igual que el comentado anteriormente, permite trasladar un bloques de datos (código diseñado por el usuario) desde otro sistema microprocesador y recibidos por el puerto serie de que dispone el DSP o desde algún dispositivo externo de memoria (colocado en el mapa de memoria del DSP en una de las tres posiciones conocidas como BOOT1, BOOT2 ó BOOT3 y normalmente de tipo EPROM) hacia otras posiciones del mapa de memoria del DSP ocupadas por dispositivos de almacenamiento de tipo RAM.

En este caso, sin embargo, el trasvase realizado desde dispositivos externos presenta las siguientes diferencias respecto del boot-loader del TMS320C31 y TMS320C33:

- Las posiciones del rango de direcciones de memoria desde las que se trasvasarían los datos son: BOOT1=1000h, BOOT2=810000h y BOOT3=900000h, figura 4.4. Esto es así, para aprovechar las especiales características que ofrece el TMS320C32 en cuanto a los accesos externos (diferentes señales de activación del acceso externo correspondiente a las zonas BOOT1 -STRBO-, BOOT2 —IOSTRB— y BOOT3 -STRB1-).
- El trasvase desde memoria externa permite emplear, además, un protocolo de acceso a los periféricos de memoria externa de tipo "acuse de recibo", con la asignación de dos líneas digitales: señal de "dato listo para ser leído", XF1, y de "reconocimiento de dato leído", XF0.
- Como en el caso anterior, las señales INT0 a INT3 determinan el modo en que sí realiza la transferencia de información. En la tabla 4.12 se especifica el modo en el que realiza el trasvase de la información el programa boot-loader, en función de las líneas antes comentadas.

INT0	INT1	INT2	INT3	Modo de realizar el boot-loader
0	1	1	1	Desde memoria externa, 1000h.
1	0	1	1	Desde memoria externa, 810000h.
1	1	0	1	Desde memoria externa, 900000h.
1	1	1	0	Desde el puerto serie síncrono del DSP.
0	1	1	0	Desde memoria externa, 1000h, con protocolo "acuse de recibo" usando las líneas XF0 y XF1.
1	0	1	0	Desde memoria externa, 810000h, con protocolo "acuse de recibo" usando las líneas XF0 y XF1.
1	1	0	0	Desde memoria externa, 900000h, con protocolo "acuse de recibo" usando las líneas XF0 y XF1.

Tabla 4.12. Modo en que se realiza el trasvase de la información en función de las líneas de entrada al DSP INT0 a INT3.

3.2 TMS320C6000

Los dispositivos TMS320C62x y TMS320C67x proporcionan una amplia variedad de configuraciones de arranque, que determinan la secuencia de pasos que el

dispositivo realiza tras un reset. La configuración de arranque, que se realiza por *hardware* imponiendo un determinado valor en ciertas entradas externas del DSP, determinará:

- El mapa de memoria. En particular, definirá si la posición de memoria cero se corresponde con memoria interna o externa.
- El tipo de memoria externa existente en la posición de memoria cero (caso de que se haya definido dicha posición como perteneciente a memoria externa).
- El proceso de arranque usado para iniciar la memoria en la dirección cero, previo a que la CPU empiece a ejecutar código.

Mientras la señal de reset se encuentra a nivel bajo, todas las salidas están en alta impedancia. En el flanco de subida, se capturan las entradas externas BOOTMODE[4:0], que deberán estar fijadas a tierra o alimentación con las correspondientes resistencias de “*pull-down*” o “*pull-up*”, respectivamente. Dependiendo del dispositivo concreto de la familia TMS320C6000, estas entradas están separadas del resto de señales del DSP o coinciden con algunas de las líneas asignadas a algún periférico de modo que, durante el flanco de subida del reset, se capturan para determinar la configuración de arranque. Existen tres tipos de procesos de arranque:

- Arranque mediante ejecución directa (ejecutando la CPU un programa ubicado en memoria externa). La CPU comienza a ejecutar desde memoria externa ubicada en la posición 0h. Caso de utilizarse una memoria SDRAM, la CPU queda a la espera de que se complete el proceso de inicialización de dicha memoria. Este proceso de arranque no está soportado por los dispositivos TMS320C621x y TMS320C671x.
- Arranque en modo *boot-loader* desde memoria externa. Una sección de memoria externa se copia a la dirección 0h mediante el controlador DMA/EDMA. Mientras se realiza la transferencia, la CPU permanece en un estado de reset interno. El ancho de la memoria externa (normalmente de tipo no volátil) se puede seleccionar mediante las líneas BOOTMODE[4:3], tabla 4.13. En el caso de que el ancho sea inferior a 32 bits, la interfaz externa (EMIF) puede ir empaquetando bytes o palabras de 16 bits consecutivas hasta formar instrucciones de 32 bits que serán finalmente

copiadas. El proceso de arranque difiere ligeramente según el dispositivo de la familia TMS320C6000:

- TMS320C620x/C670x. La DMA copia 64Kbytes desde CE1 a la dirección 0, usando la configuración por defecto del acceso externo. Tras la transferencia, la CPU comienza a ejecutar desde la dirección 0.
- TMS320C621x/C671x/C64x. La EDMA copia 1Kbytes desde CE1 a la dirección 0, usando la configuración por defecto del acceso externo. Tras la transferencia, la CPU comienza a ejecutar desde la dirección 0.
- Arranque controlado desde dispositivo externo (*host boot*). La CPU permanece en un estado interno de reset y, durante ese período, un dispositivo externo puede iniciar el espacio de memoria de la CPU (incluyendo los registros de configuración de memoria externa) mediante el periférico HPI, el bus de expansión XBUS o el bus PCI (para aquellos dispositivos que lo posean). Una vez que la memoria externa ha sido configurada, el dispositivo externo puede acceder a cualquier sección que necesite. Finalmente, el dispositivo externo provoca que la CPU salga de su estado de reset, y que empiece a ejecutar desde la posición 0h.

La tabla 4.13 resume todas las posibles configuraciones de arranque de los dispositivos TMS320C620x/C670x en función del valor de BOOTMODE[4:0] (primera columna). En los dispositivos TMS320C6202/C6203/C6204, las líneas XD[4:0] se conectan a BOOTMODE[4:0] durante el reset, mientras que en el TMS320C6205 son las líneas ED[4:0] de la EMIF. En ambos casos, los valores impuestos en las correspondientes entradas externas determinan la configuración de arranque.

BOOTMODE[4:0]	Mapa de memoria	Memoria en la dirección 0h	Proceso de arranque
00000	Mapa 0	SDRAM: SDWID = 0	Ejecución directa
00001	Mapa 0	SDRAM: SDWID = 1	Ejecución directa
00010	Mapa 0	Interfaz asíncrona de 32 bits	Ejecución directa
00011	Mapa 0	SBSRAM (1/2 reloj de la CPU)	Ejecución directa
00100	Mapa 0	SBSRAM	Ejecución directa
00101	Mapa 1	Interna	Ejecución directa
00110	Mapa 0	Externa: valores por defecto	<i>Host boot</i> (HPI/XBUS/PCI)
00111	Mapa 1	Interna	<i>Host boot</i> (HPI/XBUS/PCI)

01000	Mapa 0	SDRAM: cuatro dispositivos de 8-bit (SDWID = 0)	8-bit <i>boot-loader</i>
01001	Mapa 0	SDRAM: dos dispositivos de 16-bit (SDWID = 1)	8-bit <i>boot-loader</i>
01010	Mapa 0	Interfaz asíncrona de 32 bits	8-bit <i>boot-loader</i>
01011	Mapa 0	SBSRAM (1/2 reloj de la CPU)	8-bit <i>boot-loader</i>
01100	Mapa 0	SBSRAM	8-bit <i>boot-loader</i>
01101	Mapa 1	Interna	8-bit <i>boot-loader</i>
01110-01111	-	Reservado	-
10000	Mapa 0	SDRAM: cuatro dispositivos de 8-bit (SDWID=0)	16-bit <i>boot-loader</i>
10001	Mapa 0	SDRAM: dos dispositivos de 16-bit (SDWID = 1)	16-bit <i>boot-loader</i>
10010	Mapa 0	Interfaz asíncrona de 32 bits	16-bit <i>boot-loader</i>
10011	Mapa 0	SBSRAM (1/2 reloj de la CPU)	16-bit <i>boot-loader</i>
10100	Mapa 0	SBSRAM	16-bit <i>boot-loader</i>
10101	Mapa 1	Interna	16-bit <i>boot-loader</i>
10110-10111	-	Reservado	-
11000	Mapa 0	SDRAM: cuatro dispositivos de 8-bit (SDWID = 0)	32-bit <i>boot-loader</i>
11001	Mapa 0	SDRAM: dos dispositivos de 16-bit (SDWID = 1)	32-bit <i>boot-loader</i>
11010	Mapa 0	Interfaz asíncrona de 32 bits	32-bit <i>boot-loader</i>
11011	Mapa 0	SBSRAM (1/2 reloj de la CPU)	32-bit <i>boot-loader</i>
11100	Mapa 0	SBSRAM	32-bit <i>boot-loader</i>
11101	Mapa 1	Interna	32-bit <i>boot-loader</i>
11110-11111	-	Reservado	-

Tabla 4.13. Resumen de las configuraciones de arranque de los dispositivos TMS320C620x/C670x.

La configuración de arranque del TMS320C6711 y del TMS320C6713 se fija mediante las entradas HD[4:3], de acuerdo a la tabla 4.14. Contempla el arranque desde un dispositivo HPI externo o desde una memoria externa. En este último caso, la EDMA realiza la transferencia de 1Kbyte de código de memoria externa la dirección de memoria 0h, siendo configurable el ancho de la memoria externa de arranque a 8, 16 ó 32 bits. La interfaz externa se encarga, automáticamente, de componer instrucciones de 32 bits a partir de las lecturas a las memorias señaladas. Finalmente, se empieza a ejecutar el código a partir de la dirección 0h.

Configuración	Descripción
HD[4:3]	00 – $\overline{CE}1$ con 32 bits, HPI boot
	01 – $\overline{CE}1$ con 8 bits, arranque desde memoria externa asíncrona

	10 – $\overline{CE1}$ con 16 bits, arranque desde memoria externa asíncrona
	11 – $\overline{CE1}$ con 32 bits, arranque desde memoria externa asíncrona

Tabla 4.14. Configuración de arranque de los dispositivos TMS320C6711/C6713.

El TMS320C6712, a diferencia de los dos anteriores, posee dos entradas dedicadas para configurar el arranque, según la tabla 4.15. En este caso, el arranque en modo emulación se utiliza para la depuración del código, y los otros dos modos no reservados hacen uso de la EDMA para realizar la copia de 1Kbyte de código desde memoria externa, situada en la zona $\overline{CE1}$, a la dirección 0h. Según la configuración elegida, la interfaz externa EMIF realiza cuatro accesos a posiciones de 8 bits o dos accesos a posiciones de 16 bits, para completar instrucciones de 32 bits. Una vez realizada la transferencia, la CPU sale del estado de reposo y comienza a ejecutar a partir de la dirección 0h.

Configuración	Descripción
BOOTMODE[1:0]	00 – Arranque en modo emulación
	01 – $\overline{CE1}$ con 8 bits, arranque desde memoria externa asíncrona
	10 – $\overline{CE1}$ con 16 bits, arranque desde memoria externa asíncrona
	11 – Reservado

Tabla 4.15. Configuración de arranque de los dispositivos TMS320C6712.

4. MEMORIA CACHÉ

4.1 TMS320C3x

Los DSPs de la familia TMS320C3x de Texas Instruments poseen un periférico interno destinado a la optimización, en caso necesario, de los accesos del sistema a memoria externa de programa. Este periférico consiste en una memoria capaz de almacenar hasta 64 instrucciones (de tamaño 64×32 bits) y un algoritmo de control del acceso a dicha memoria. Se usa para almacenar secciones de código repetidas, para

evitarle a la CPU el acceso al exterior a volver a recoger una instrucción que hace relativamente poco tiempo que se recogió. Esto reduce los accesos a memoria externa acelerando, en el tiempo, la ejecución de los programas almacenados en dispositivos de memoria externa lentos (que tienen por otro lado la ventaja de ser los más baratos). Cuando la CPU pretende recoger una instrucción entra en funcionamiento este periférico, siempre que se encuentre habilitado, que decide, de alguna forma, si es o no necesario el acceso al dispositivo externo donde se encuentra físicamente la instrucción.

Una de las principales ventajas e inconvenientes de este periférico, es que actúa de forma independiente y automática, el usuario no interviene. Usa un algoritmo denominado *LRU (Least Recently Used)* para la actualización del contenido de la memoria y como lógica de control del acceso al periférico.

La arquitectura de la memoria Caché se muestra en la figura 3.11. *La memoria Caché* se divide en dos segmentos de 32 *words* cada uno, con 32 posibles instrucciones cada uno. A cada segmento se le asocia otro registro de 19 bits denominado **SSA** (*Segment Start Address*), **SSA0** para el segmento 0 y **SSA1** para el segmento 1. A cada *word* de la Caché se le asocia un bit, denominado bit P, que indica si está o no presente una instrucción en ella.

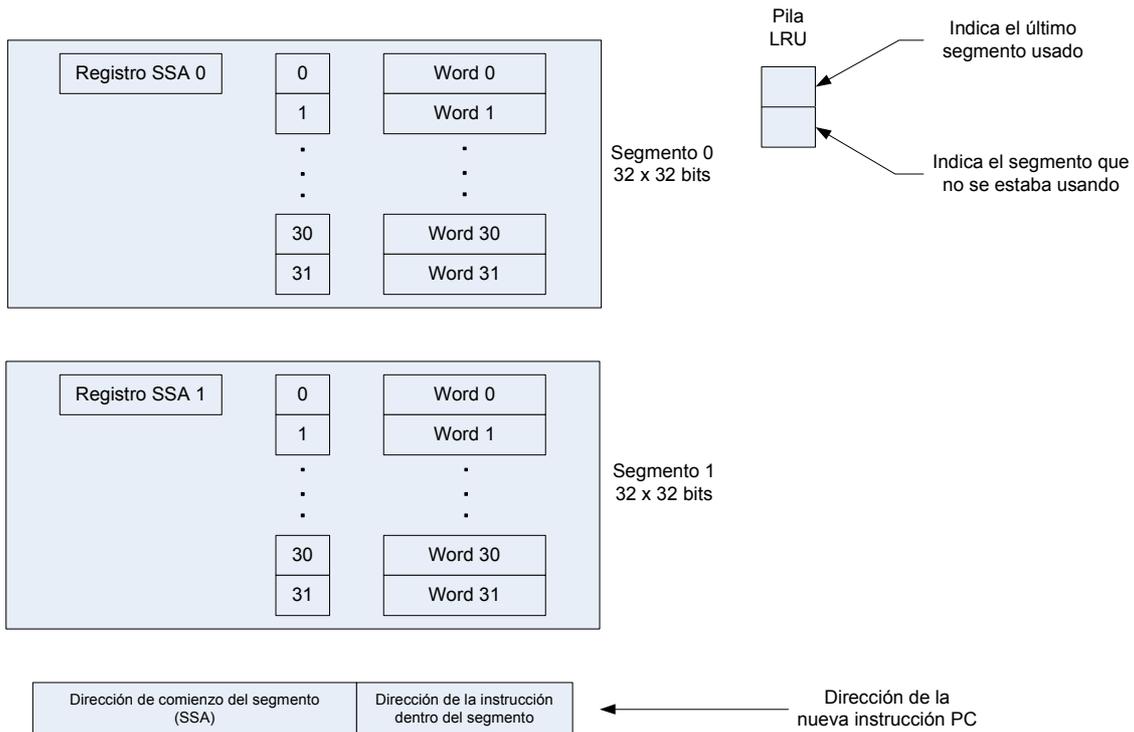


Fig. 4.9. Arquitectura del periférico de memoria caché

Cuando la CPU necesita recoger una instrucción ubicada en la memoria externa, el periférico caché realiza un chequeo para ver si la instrucción está contenida en su memoria. El Algoritmo de control de acceso a la memoria de la caché se basa en la división de la dirección de la nueva instrucción que se desea recoger, indicada por el registro PC, en dos partes:

- Los 19 MSBs del registro PC (bits más significativos de la dirección). Se usan para seleccionar entre uno de los dos segmentos de que dispone el periférico (SSA0 ó SSA1).
- Los 5 LSBs del registro PC (bits menos significativos de la dirección). Se usan para, buscar la instrucción dentro del segmento seleccionado, para seleccionar una posición de memoria de entre 32 posibles que dispone cada segmento.

Los 19 MSBs de la dirección de la nueva instrucción se comparan con los dos registros SSA. Si coincide con alguno de ellos, los 5 LSBs del registro PC se utilizan para comprobar si el bit P del segmento seleccionado está activo. El bit P lo utiliza la CPU para ver si la instrucción que desea recoger está presente en el segmento. Si no se

encuentra una coincidencia con los SSA de los 19 MSBs de la instrucción puede reemplazarse uno de los dos segmentos de la memoria caché con nuevos datos, instrucciones que se recojan de la memoria externa a partir de ese momento.

El algoritmo LRU es el que determina el segmento a reemplazar. Para ello se define una pila de 2 bits, *LRU-Stack*, que se emplea en determinar el segmento, de los dos posibles, menos usado recientemente por el periférico. Cuando hay que reemplazar datos de la caché se usa el segmento que más tiempo hace que no es accedido, determinado por el *elemento o bit que ocupa la parte inferior de la pila*. Una vez determinado el segmento menos empleado, las 32 banderas P de él son puestas a 0 el registro SSA correspondiente a ese segmento es reemplazado por los 19 MSBs de la dirección de la instrucción que se está recogiendo en esos momentos. Cada vez que el periférico accede a uno de los segmentos introduce en *la cima de la pila* el número correspondiente al mismo (si se usó el SSA0, se introduce en la parte alta de la pila del LRU el valor 0). Obsérvese que la pila tiene 2 bits, uno de los cuales vale siempre 1 y el otro 0, de forma y manera que si en la parte alta de la pila aparece un 0 en la baja tiene que haber un 1 y viceversa.

Cuando se resetea el sistema microprocesador, el periférico de memoria caché se inicializa con los siguientes valores:

- *Top pila=0*. El segmento 0 es el segmento que, después de un reset, se define como el último usado.
- *Bottom pila=1*. El segmento 1 es el segmento que, después de un reset, se define como el que no se ha usado recientemente.
- Todas las banderas P (una por cada word de la memoria caché) se ponen a 0 indicando que no hay ninguna instrucción en el registro correspondiente.

El algoritmo de acceso y actualización de la memoria caché se resume como sigue:

1. CACHE HIT: Se dice que se ha producido un Caché Hit si la memoria del periférico caché tiene la instrucción que la CPU está buscando. En este caso, la CPU no accede al exterior a recoger la instrucción sino que la recoge de la memoria del periférico caché.

El algoritmo de acceso a la instrucción se completa con la actualización de la pila de dos bits: En la cima de la pila (top pila) se introduce el número del segmento (0 ó 1) donde estaba la instrucción recogida.

2. CACHE MISS: Se dice que se ha producido un Caché Miss si la memoria del periférico caché no tiene la instrucción que la CPU está buscando. En este caso, el periférico no dispone de la instrucción por dos posibles causas:

- **WORD MISS:** Uno de los segmentos SSA coincide con los 19 MSBs de la dirección de la instrucción que se desea recoger pero la bandera P, asociada a los 5 LSBs de esa instrucción, no está activada. En este caso, la instrucción es recogida por la CPU de la memoria externa y se aprovecha para almacenarla en la propia caché. Los 5 LSBs de la dirección de la instrucción indican la word en la que se almacena la instrucción. La bandera P correspondiente a dicha posición de memoria de la caché se pone a 1. En la cima de la pila de dos bits aparece el número del segmento en el que se acaba de almacenar la instrucción recogida.
- **SEGMENT MISS:** Los 19 MSBs de la dirección de la instrucción no coinciden con ninguno de los registros SSA del periférico. En este caso, la instrucción se recoge de memoria exterior y, simultáneamente, se aprovecha para almacenarla en la caché. Ahora, al contrario que en el caso anterior, todas las banderas P correspondientes al segmento que indicaba la parte baja de la pila (bottom pila) son puestos a 0 (se limpian). Los 19 MSBs de la dirección de la instrucción accedida se copian al registro SSA del segmento inicializado, se activa la bandera P ligada a los 5 LSBs de la dirección de la instrucción recogida y, finalmente, se guarda la instrucción recogida en la posición de memoria seleccionada de la caché. Al final, el número del segmento activado se almacena en la cima de la pila (top pila).

4.2 TMS320C6000

En la figura 4.10 podemos observar la organización de la memoria de las generaciones TMS320C621x/C671x, con las dos memorias de nivel 1, L1P y L1D (de

programa y datos, respectivamente) de 4Kbytes. En el caso de la arquitectura de la generación TMS320C64x, con cada memoria L1 de 16Kbytes. Todas las caché y los caminos entre ellas son gestionados por el controlador de caché. El nivel 1, más próximo a la CPU, es accesible sin necesidad de estados de espera. El nivel 2 es configurable y puede dividirse en memoria SRAM y memoria caché. Su tamaño es variable según el elemento de la familia TMS320C6000, pero oscila entre 64 Kbytes y 1 Mbyte. Finalmente, el último nivel de la jerarquía lo ocupa la memoria externa, que puede alcanzar tamaños de varios Mbytes.

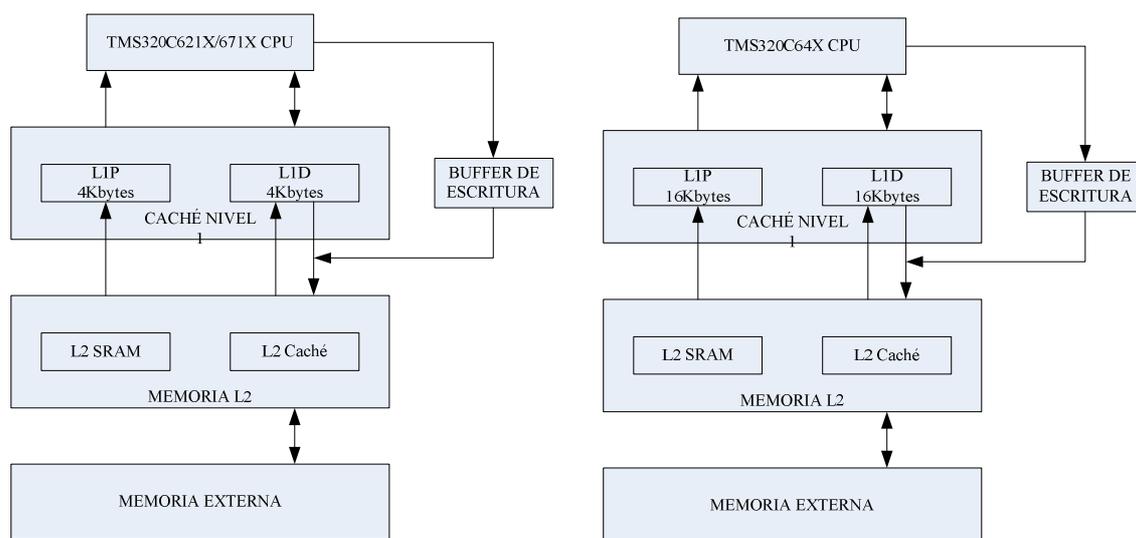


Fig. 4.10. Esquema general de la arquitectura de memoria de las generaciones TMS320C621x/C671/C64x.

4.2.1 Caché de instrucciones de primer nivel L1P

La caché L1P es una caché de instrucciones de correspondencia directa. La tabla 4.16 muestra las diferentes características de la caché L1P en las generaciones de la familia TMS320C6000.

Características	TMS320C621x/C671x	TMS320C64x
Organización	Correspondencia directa	Correspondencia directa
Protocolo	Alojamiento en lectura	Alojamiento en lectura
Tiempo de acceso de CPU	1 ciclo	1 ciclo
Capacidad	4 Kbytes	16 Kbytes
Tamaño de línea	64 bytes	32 bytes

Penalización de fallo	5 ciclos	8 ciclos
Segmentación del fallo	No	Sí

Tabla 4.16. Características de la caché L1P en las generaciones de la familia TMS320C6000.

En ambos casos, la caché es de alojamiento en lectura, lo que significa que únicamente se guardan datos en la caché cuando ocurre un fallo durante un acceso en lectura (algo lógico en este caso, debido a que se trata de una caché de instrucciones). Asimismo, son memorias rápidas, que no requieren de estados adicionales de espera en los accesos de la CPU. La principal diferencia entre las generaciones TMS320C621x/C671x y la TMS320C64x estriba en la capacidad y el tamaño de bloque, mayores en este último caso. La última característica de la tabla 4.16 hace referencia a la segmentación del fallo, es decir, a que el proceso de gestión de una serie de fallos pueda solaparse en el tiempo, lo que permite que la penalización de fallo sea menor en los fallos subsiguientes. Esta posibilidad sólo existe en la generación TMS320C64x.

La figura 4.11 muestra un esquema simplificado de caché L1P en la generación TMS320C671x, para la que vamos a particularizar el análisis. Esta caché posee una capacidad de 4 Kbytes organizada en 64 líneas de 64 bytes. A cada dirección de la memoria de programa le corresponde un lugar fijo en la caché. Por ejemplo, en esta figura se observa que las direcciones de la 0000h a 003Fh son siempre capturadas en la línea 0 de la caché y las direcciones de la 0FC0h a la 0FFFh en la línea 63. Dado que el tamaño de la caché es limitado, el rango de direcciones de la 1000h a la 103Fh vuelve a estar asignado a la línea 0, y así sucesivamente. Es importante reseñar que los 64 bytes que contiene cada línea de la caché, coincide exactamente con el tamaño de dos paquetes de recogida (*fetch packets*).

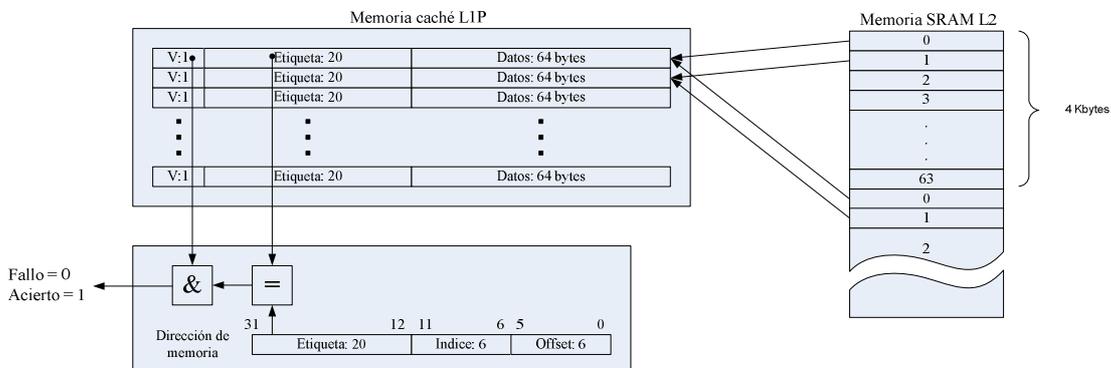


Fig. 4.11. Arquitectura de la caché L1P en la generación TMS320C671x.

El campo índice de la dirección de memoria (bits del 11 al 5) determina la línea asignada a un bloque de instrucciones dentro de la caché. Para determinar si un dato se encuentra en la caché, el controlador compara el campo etiqueta (bits del 31 al 12) de la dirección de las instrucciones buscadas con la etiqueta de la línea seleccionada en la caché. En caso de coincidencia, se comprueba el bit de validez. Si el resultado es un cero, significa que el acceso ha sido fallido, en tanto que si es un uno significa que las instrucciones se encuentran efectivamente en la caché y que, en consecuencia, la búsqueda concluye con acierto. Una búsqueda en L1P que acabe en acierto se completa en un único ciclo máquina. Si el resultado es un fallo, la penalización asociada a ese fallo (acceso al nivel inferior L2 y transferencia del dato) es de 5 ciclos, siempre y cuando la instrucción se encuentre en ese nivel inferior (si no es así, la penalización sería mayor puesto que habría que buscarla en memoria externa). Para aprovechar la localidad espacial, cuando ocurre un fallo se recoge del nivel inferior todo un bloque de instrucciones (dos paquete de recogida en el TMS320C671x), ya que existe una alta probabilidad de que el siguiente paquete de ejecución pueda estar contenido en ese bloque capturado.

4.2.2 Caché de datos de primer nivel L1D

La caché L1D es una caché de datos asociativa de dos vías cuyas características se muestran en la siguiente tabla.

Características	TMS320C621x/C671x	TMS320C64x
Organización	Asociativa de dos vías	Asociativa de dos vías
Protocolo	Alojamiento en lectura, post-escritura	Alojamiento en lectura, post-escritura
Tiempo de acceso de CPU	1 ciclo	1 ciclo
Capacidad	4 Kbytes	16 Kbytes
Tamaño de línea	32 bytes	64 bytes
Penalización de fallo en lectura (L2 SRAM)	4 ciclos	6 ciclos
Penalización de fallo en lectura (L2 Caché)	4 ciclos	8 ciclos
Segmentación del fallo	No	Sí
Penalización de fallos consecutivos múltiples	4 ciclos	4 ciclos + 2 M ciclos

(L2 SRAM)		
Penalización de fallos consecutivos múltiples (L2 Caché)	4 ciclos	4 ciclos + 2 M ciclos
Fallo en escritura	Búfer de 4x32 bits	Búfer de 4x64 bits

Tabla 4.17. Características de la caché L1D en las generaciones de la familia TMS320C6000.

Igual que en el caso anterior, particularizaremos el estudio para la generación TMS320C671x. Se trata de una caché con una capacidad de 4 Kbytes (2 Kbytes por vía) y líneas que contienen 32 bytes de información. La principal diferencia con respecto a la caché de correspondencia directa es que para una dirección de la memoria de datos se dispone de dos vías que nos llevan a dos líneas alternativas de la caché. En la caché de correspondencia directa, cuando se accede a una posición de memoria cuyo índice apunta a una línea donde previamente se ha guardado el contenido de otra posición, se produce el desalojo de esta última. En la asociativa de dos vías, el desalojo no es necesario, debido a que se dispone de una segunda línea alternativa.

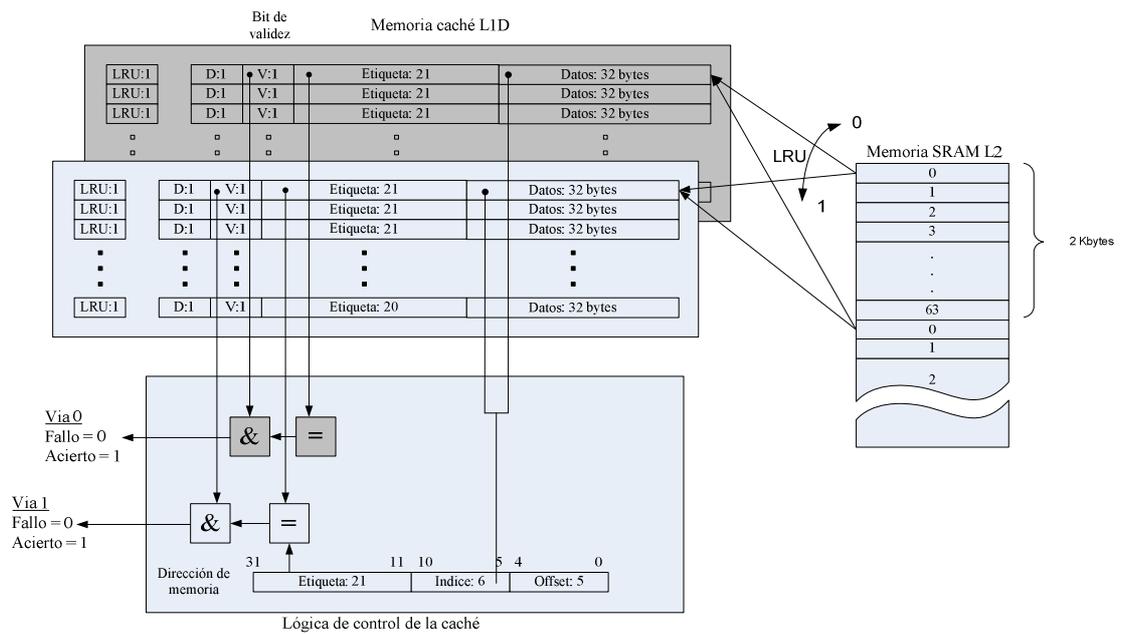


Fig. 4.16. Arquitectura de la caché L1D de la generación TMS320C671x.

La figura 4.16 muestra un esquema simplificado de la caché L1D de la generación TMS320C671x. En color gris oscuro aparece la vía 0 y en color gris claro la vía 1. El índice de la dirección buscada (bits del 10 al 5) apunta a dos líneas diferentes asociadas a las vías 0 y 1 de la caché. Si alguna de las líneas tiene ya datos almacenados,

no es necesario desalojarlos para ubicar nuevos datos cuyo índice apunte a la misma línea, aunque con una etiqueta diferente.

El estado del bit LRU se actualiza cada vez que se produce el alojamiento de una línea de alguna de las vías. Concretamente, cuando se accede a una vía, el bit LRU conmuta apuntando a la contraria. La idea es que el bit LRU siempre apunte a la vía que lleva más tiempo sin ser utilizada, para evitar el desalojo de la vía más recientemente utilizada. Cuando se produce un fallo en el acceso a la caché, se consulta el estado del bit LRU para determinar en que vía debe ser alojado el bloque de datos transferido desde el nivel jerárquicamente inferior. Si el bit vale 0, el nuevo bloque se alojará en la vía cero, ya que esa vía es la que lleva más tiempo sin ser usada, y de ese modo no se desaloja la línea de la vía 1, ocupada por otros datos que fueron alojados más recientemente. Análogamente, si el valor del bit LRU es 1, se utilizará la línea de la vía 1 para alojar el nuevo bloque de datos. El estado del bit LRU únicamente se consulta en caso de fallo, pero este bit se actualiza permanentemente, tanto si hay acierto o fallo como si la operación en cuestión es una lectura o escritura.

La caché L1D es sólo de alojamiento en lectura, lo que significa que sólo cuando se produce un fallo en la lectura de algún dato se produce el desalojo de una línea antigua para reemplazarla por otra nueva. Cuando lo que se produce es un fallo en la escritura, el dato se escribe directamente en la memoria física, de un nivel inferior en la jerarquía y que puentea la caché L1D. El tamaño del búfer es de 4x32 bits para la generación TMS320C621x/C671x, y de 4x64 bits para la generación TMS320C64x. Los fallos en escritura no ralentizan normalmente a la CPU, a no ser que el búfer se encuentre lleno, en cuyo caso la finalización del acceso se prolonga hasta que exista espacio suficiente en el búfer para dar entrada a estos datos. De modo indirecto, un fallo en escritura también puede provocar una ralentización de los fallos en lectura, ya que estos no serán procesados hasta que el búfer no esté completamente vacío. Esto es necesario, dado que se puede dar el caso de que la CPU acceda en lectura a una posición de memoria de datos que debe ser actualizada con el contenido del búfer (si la CPU accediese directamente, estaría leyendo el valor antiguo almacenado en esa posición, ya que no habría tenido tiempo de actualizarse).

Cuando se produce un acierto en escritura, el dato se escribe en la caché, aunque no se trasfiere a los niveles inferiores de memoria, ya que la estrategia que sigue la caché para mantener la coherencia de los datos es la de post-escritura. Para ello, y asociado a cada línea de la caché, existe un bit D (*dirty bit*) que indica si ha sido modificada por un acierto en escritura. Cuando una línea marcada con el bit D a 1 (es decir, que tuvo un acierto en escritura) necesita ser desalojada debido a que se ha producido un fallo en lectura (recuérdese que la caché es de sólo alojamiento en lectura), será necesario actualizar dicho dato en los niveles de la memoria jerárquicamente inferiores antes de proceder al desalojo. De ese modo, se garantiza la coherencia de los datos. Si el bit D hubiese estado a cero, eso implicaría que el contenido de la línea no ha sido modificado y que se puede proceder al desalojo de la misma sin mayor problema.

4.2.3 Caché de segundo nivel L2

La jerarquía de memoria de las generaciones TMS320C621x/C671x y TMS320C64x es la única que cuenta con el nivel L2 de memoria, que puede funcionar como memoria SRAM, caché o una mezcla de ambas. El motivo de introducir un segundo nivel de caché es suplir la diferencia existente, en cuanto a tamaño y tiempos de acceso, entre la caché de primer nivel y la memoria principal. Configurada como caché, el nivel L2 es una caché asociativa por vías, cuyas características se detallan en la tabla 4.18.

Características	TMS320C621x/C671x	TMSC64x
Organización	Asociativa de 1,2,3 ó 4 vías	Asociativa de 4 vías
Protocolo	Alojamiento en lectura y escritura, post-escritura	Alojamiento en lectura y escritura, post-escritura
Capacidad	16/32/48/64 Kbytes	32/64/128/256 Kbytes
Tamaño de línea	128 bytes	128 bytes

Tabla 4.18. Características de la caché L2 en las generaciones TMS320C621x/C671x y TMS320C64x de la familia TMS320C6000.

La capacidad como caché del nivel de memoria L2 depende de cómo haya sido configurada. Tras un reset, toda la memoria L2 está configurada como SRAM, aunque es posible, por *software*, habilitar una parte como caché (evidentemente, a costa de reducir la SRAM disponible).

Actúa como caché de nivel inferior a L1. Cuando se produce un fallo en lectura en L1, la dirección buscada pasa a L2. Siguiendo la misma operativa de etiquetas y bit de validez, se determina si el dato buscado está o no presente en L2. A diferencia de la caché L1, que era de sólo alojamiento en lectura, la caché L2 es de alojamiento en lectura y escritura, siguiendo además, en este último caso, la estrategia de post-escritura para garantizar la coherencia de los datos.

Consideremos una petición de lectura de la CPU a una posición de memoria externa que haya sido configurada como cacheable (existe la posibilidad de configurar una posición de memoria externa como cacheable o como no cacheable, es decir, la información transferida con la CPU queda o no alojada en la caché) que provoca un fallo en L1:

- Si esa dirección provoca también fallo en la caché L2, el bloque de datos correspondiente será transferido a la caché L2. El bit LRU determina la vía en la que debe situarse. Si la línea se halla marcada como *dirty* (bit D=1), lo primero que se hará será trasvasarla a memoria externa antes de recoger el nuevo bloque de datos. Para ello, es necesario que se compruebe antes si ese bloque de datos se encuentra contenido también en L1D. En ese caso, se debe actualizar el contenido de L2 antes de trasvasar la línea a la memoria, para mantener la coherencia de la caché. Posteriormente, se recoge el bloque de datos en la línea que corresponda de L2 y una porción de esa línea (los tamaños de línea pueden diferir) de la caché L2 (aquella que contiene la dirección referenciada), es enviada a la L1, que almacena la nueva línea en su caché. Finalmente, se envía el dato requerido a la CPU.
- Si la dirección referenciada provoca un acierto en L2, una parte de esa línea (aquella que contenga al dato accedido) es enviada a L1.

Algunas direcciones de memoria externa pueden ser configuradas como no-cacheables. En este caso, el dato requerido es, simplemente, enviado desde memoria externa a la CPU sin ser almacenado en ninguna de las cachés.

Si una petición de escritura, realizada por la CPU a una posición de memoria externa, provoca fallos en L1D, ésta pasada a L2 a través del búfer (como ya se vió anteriormente). En ese caso:

- Si L2 detecta un fallo para esta dirección, el correspondiente bloque de datos es recogido de memoria externa, modificado con el nuevo dato que escribe la CPU y almacenado en la línea asignada. Si esta línea contiene datos más actuales que los que existen en memoria externa (*dirty bit* activo), serán trasvasado a memoria externa antes de recoger el bloque nuevo.
- Si el acceso fue un acierto a L2, la correspondiente línea de L2 es actualizada con el dato que proviene de la CPU. Obsérvese que este dato no se actualiza en memoria externa, y que la línea en L2 se marca como *dirty*, dado que se sigue una estrategia de post-escritura.

En las posiciones de memoria marcadas como no-cacheables, los datos son actualizados en memoria externa directamente, sin ser almacenados en la caché.

4.2.4 Configuración de la caché

Los registros que intervienen en la configuración de la caché de dos niveles de las generaciones TMS320C621x/C671x son el registro de estado y de control de la CPU (CSR) y el registro de configuración de la caché (CCFG). En concreto, los diferentes campos de bits que se emplean en la configuración de la memoria caché son:

- El campo DCC (bits 4-2 del registro CSR) sólo admite los valores 000b y 010b, que configuran a L1D como caché asociativa de dos vías. El resto de valores están reservados.
- El campo PCC (bits 7-5 del registro CSR) sólo admite los valores 000b y 010b, que configuran a L1P como caché de correspondencia directa. El resto de valores están reservados.
- El campo L2MODE (bits 2-0 del registro CCFG). Determina qué porción de L2 funciona como SRAM y qué porción como caché, según la tabla 4.19.
- Los bits IP e ID (bits 9 y 8 del registro CCFG) que habilitan (a '0') o deshabilitan (a

‘1’) las caché L1P y L1D, respectivamente.

Campo L2MODE	TMSC6211/C6711/C6712	TMSC6713
000 - L2 caché deshabilitada	64K SRAM	256K SRAM
001 – caché de 1 vía (caché L2 16K)	48K SRAM	240K SRAM
010 – caché de 2 vías (caché L2 32K)	32K SRAM	224K SRAM
011 – caché de 3 vías (caché L2 48K)	16K SRAM	208K SRAM
100 – 110 – Reservado	-	-
111 – caché de 4 vías (caché L2 64K)	0K SRAM	192K SRAM

Tabla 4.19. Posibles configuraciones de la caché L2, según el campo L2MODE y para las generaciones TMS320C621x/C671x.

Los registros de los atributos de la memoria L2 (registros MAR, detallados en la tabla 4.20) determinan si los distintos rangos de direcciones de 16 Mbyte de memoria externa son, o no, cacheables: si el bit menos significativo vale ‘0’, el rango asignado a ese registro MAR no es cacheable (no puede ser capturado por la caché) y lo contrario si vale ‘1’.

Dirección	Acrónimo	Descripción
0184 8000h a 0184 81FCh	MAR0 a MAR127	Reservado
0184 8200h	MAR128	Rango EMIF CE0 8000000h–80FFFFFFh
0184 8204h	MAR129	Rango EMIF CE0 8100000h–81FFFFFFh
0184 8208h	MAR130	Rango EMIF CE0 8200000h–82FFFFFFh
0184 820Ch	MAR131	Rango EMIF CE0 8300000h–83FFFFFFh
0184 8210h a 0184 823Ch	MAR132 a MAR143	Reservado
0184 8240h	MAR144	Rango EMIF CE1 9000000h–90FFFFFFh
0184 8244h	MAR145	Rango EMIF CE1 9100000h–91FFFFFFh
0184 8248h	MAR146	Rango EMIF CE1 9200000h–92FFFFFFh
0184 824Ch	MAR147	Rango EMIF CE1 9300000h–93FFFFFFh
0184 8250h a 0184 827Ch	MAR148 a MAR159	Reservado
0184 8280h	MAR160	Rango EMIF CE2 A000000h–A0FFFFFFh
0184 8284h	MAR161	Rango EMIF CE2 A100000h–A1FFFFFFh
0184 8288h	MAR162	Rango EMIF CE2 A200000h–A2FFFFFFh
0184 828Ch	MAR163	Rango EMIF CE2 A300000h–A3FFFFFFh
0184 8290h a 0184 82BCh	MAR164 a MAR175	Reservado
0184 82C0h	MAR176	Rango EMIF CE3 B000000h–B0FFFFFFh
0184 82C4h	MAR177	Rango EMIF CE3 B100000h–B1FFFFFFh
0184 82C8h	MAR178	Rango EMIF CE3 B200000h–B2FFFFFFh
0184 82CCh	MAR179	Rango EMIF CE3 B300000h–B3FFFFFFh
0184 82D0h a 0184 83FCh	MAR180 a MAR255	Reservado

Tabla 4.20. Registros de atributos de memoria (MAR).

5. PERIFÉRICO DMA

Un periférico DMA (Direct Memory Access) es un periférico capaz de gestionar, sin coste alguno en tiempo de CPU para el microprocesador, transferencias de bloques de datos desde una posición en el mapa de memoria del sistema microprocesador (origen) hacia otra (destino).

Una transferencia DMA hace referencia al trasvase de un grupo de datos en el mapa de memoria del sistema digital. El número de datos que se van a trasvasar es uno de los parámetros característicos de la transferencia DMA. El periférico DMA dispone de un contador de datos transferidos que se inicializa al número de datos que se desean trasvasar. Cada vez que el periférico transfiere un dato en el mapa de memoria del sistema, disminuye en una unidad dicho contador. Se dice que la transferencia DMA ha concluido cuando el registro contador de datos transferidos alcanza el valor cero.

Cada dato trasvasado en el mapa de memoria implica dos operaciones elementales a realizar por parte del periférico DMA: una lectura del dato en la dirección origen (Read-DMA) y una escritura del mismo en la dirección destino (Write-DMA). Al final de cada proceso de tipo Read-DMA (Write-DMA) la dirección origen (destino) automáticamente se incrementa en una unidad, disminuye en una unidad o permanece constante.

El periférico DMA es un dispositivo que puede funcionar, en el sistema digital en el que se encuentre inmerso, como maestro o esclavo. El microprocesador puede acceder a él para configurar su modo de funcionamiento, como haría con cualquier otro periférico, comportándose, en ese momento, el periférico DMA como esclavo en el sistema. La característica fundamental que tiene este periférico es que, una vez configurado y habilitado, su funcionamiento normal consiste en realizar transferencias de información a través de los buses del sistema, sin que dicha transferencia suponga coste de CPU para el micro. Para poder realizar esa tarea, el periférico DMA necesita poder gobernar los buses y funcionar como maestro en el sistema digital. Obviamente, para no provocar conflictos en los accesos a los buses, sólo puede haber un maestro en el sistema.

Para evitar que aparezcan dos maestros que gobiernen las líneas de los buses del sistema se añade al bus de control de muchos microprocesadores un par de líneas, una de entrada (normalmente HLD) y otra de salida (normalmente HLDA), que implementan un protocolo de acuse de recibo. El protocolo que se establece implica que, por defecto, el maestro en el sistema es el microprocesador. Si algún periférico desea tomar el control de las líneas asociadas a los buses de datos, dirección y control, primero debe solicitarlo al maestro (esta petición se realiza activando la línea HLD de entrada al microprocesador). El microprocesador analiza internamente la petición y, si el usuario que ha programado el micro lo permite, atiende la petición (deja en triestado las líneas de los buses de dirección, control y datos). Una vez que el micro ha dejado de gobernar los buses del sistema le indica, activando la línea HLDA, al periférico que quería ser maestro (normalmente un periférico de tipo DMA) que puede gobernar los buses.

A partir de ese momento el que controla, como maestro, las líneas de los buses del sistema digital es el periférico DMA. El periférico realiza, entonces, la transferencia de información (programada previamente por el microprocesador) en el mapa de memoria del sistema digital. Esta transferencia de información es un proceso repetitivo que consiste en un acceso en lectura a una posición de memoria (el periférico DMA impone, en las líneas asociadas al bus de direcciones del sistema, el valor de la dirección de la posición a la que desea acceder, activa la señal de lectura asociada al bus de control y copia el valor que aparece en el bus de datos en un registro interno del periférico) y un acceso posterior en escritura a otra zona de memoria (el periférico DMA impone, en las líneas asociadas al bus de direcciones del sistema, el valor de la dirección de la posición a la que accede, activa la señal de escritura asociada al bus de control y copia el valor que aparece en el registro interno del periférico en el bus de datos). La dirección del bloque de memoria que se desea copiar, el tamaño de dicho bloque (que indica cuantas veces consecutivas accede en lectura y en escritura la DMA como maestro del sistema digital) y la dirección donde se desea copiar el bloque de memoria son datos que programó el microprocesador cuando accedió, para configurarlo, al periférico DMA. Una vez que el periférico DMA es maestro, el microprocesador no puede efectuar accesos al exterior por los buses del sistema hasta que la DMA no pase a

ser esclavo, cosa que ocurre cuando se concluye el trasvase programado.

La utilidad que ofrece un periférico de este tipo es descargar al microprocesador de la realización de operaciones de trasvase de datos. Un periférico de este tipo tiene interés sólo si el usuario tiene previsto desarrollar aplicaciones en las que se realicen gran cantidad de trasvases de datos en el mapa de memoria del sistema digital.

Tanto la familia TMS320C3x, como la familia TMS320C6000, ofrece periféricos DMA internos que realizan transferencias de bloques de información desde una posición en el mapa de memoria del sistema microprocesador (origen) hacia otra (destino).

5.1 TMS320C3x

5.1.1 Registros de control y configuración

El periférico DMA dispone de cuatro registros, de 32 bits cada uno, dedicados al control y configuración del mismo y ubicados en una determinada zona del mapa de memoria del DSP. Estos registros pueden ser accedidos en lectura o escritura por el usuario.

- Registro global de control y configuración del periférico. Determina el modo de operación del periférico. Indica, además, el estado en el que se encuentra la transferencia DMA. En la figura 4.17 se muestra este registro y en las tablas 4.21, 4.22, 4.23, 4.24 y 4.25 se indican las funciones de cada bandera del mismo.
- Registro dirección fuente de la transferencia DMA. Indica la dirección en la que se encuentra el dato que se va a leer en la transferencia DMA. De los 32 bits del registro sólo tienen interés los 24 LSB (este DSP sólo dispone de 24 líneas de dirección).

- Registro dirección destino de la transferencia DMA. Indica la dirección en la que se desea almacenar el dato leído por el periférico DMA. De los 32 bits del registro sólo tienen interés los 24 LSB.
- Registro contador de transferencias DMA. Indica el número de transferencias que debe realizar el periférico (número de datos que se reubicarán en el mapa de memoria del DSP).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XX	XX	XX	XX	TCINT	TC	SYNC	DEC DST	INC DST	DEC SRC	INC SRC	STAT	START			

Fig. 4.17. Registro global de control y configuración del periférico DMA de la familia TMS320C3x.

Bit	Nombre	Valor de Reset	Función que realiza
1-0	START	00	Controla el estado en el que la DMA arranca y para.
3-2	STAT	00	Indica el estado de la DMA.
4	INCSRC	0	Si INCSRC=1, incrementa, después de cada lectura DMA, la dirección del registro fuente para la transferencia DMA.
5	DECSRC	0	Si DECSRC=1, decrementa, después de cada lectura DMA, la dirección del registro fuente para la transferencia DMA.
6	INCDST	0	Si INCDST=1, incrementa, después de cada escritura DMA, la dirección del registro destino para la transferencia DMA.
7	DECDST	0	Si DECDST=1, decrementa, después de cada escritura DMA, la dirección del registro destino para la transferencia DMA.
9-8	SYNC	00	Indica la forma de lanzar los eventos DMA (lecturas DMA y/o escrituras DMA).
10	TC	0	Afecta a la operación del contador de transferencias DMA. Si TC=0, las transferencias DMA no concluyen cuando el contador de transferencias alcance el valor 0. Si TC=1, las transferencias DMA concluyen al alcanzar el contador de transferencias el valor 0.
11	TCINT	0	Si TCINT=1, cuando el contador de transferencias DMA alcance el valor 0 se activa la petición de interrupción DMA. En caso contrario, no se activa.
31-12	Reservado	0-0	Se lee siempre un cero.

Tabla 4.21. Funciones de los bits del registro de control y configuración del periférico DMA de la familia TMS320C3x.

START	Función que realiza
0-0	Valor de reset. Los ciclos de lectura o escritura DMA que aún no hayan terminado, se completan. La DMA no realiza más transferencias y se inicializa: cuando se lance un nuevo evento DMA, comienza una nueva transferencia DMA.

0-1	Si se ha efectuado o ha comenzado un evento de lectura o escritura DMA cuando aparece este valor, se completa dicho evento antes de pararse la DMA. Si no ha comenzado ningún evento, la DMA se vuelve inactiva.
1-0	Si ha comenzado una transferencia DMA, se completa (incluyendo los eventos de lectura y escritura DMA) antes de pararse el periférico. Si no ha comenzado ninguna transferencia, la DMA se vuelve inactiva.
1-1	La DMA arranca desde el reset o desde el estado previo en que estaba.

Tabla 4.22. Inicio de eventos DMA. Funciones de los bits START del registro de control y configuración del periférico DMA de la familia TMS320C3x.

STAT	Función que realiza
0-0	Valor de reset. La DMA está parada al final de una transferencia DMA (entre un evento de escritura y otro de lectura).
0-1	La DMA está parada en la mitad de una transferencia DMA (entre un evento de lectura y otro de escritura).
1-0	Reservado.
1-1	La DMA está ocupada (realizando un evento de lectura, uno de escritura o esperando la llegada de una petición de interrupción para lanzar eventos programados).

Tabla 4.23. Estado del periférico DMA. Funciones de los bits STAT del registro de control y configuración del periférico DMA de la familia TMS320C3x.

SYNC	Función que realiza
0-0	SIN SINCRONIZACIÓN. Valor de reset. La sincronización de los eventos DMA con las peticiones de interrupción, está inhabilitada.
0-1	SINCRONIZACIÓN DE LA FUENTE. La sincronización de los eventos de lectura de la DMA se realiza con las peticiones de interrupción. La DMA genera un evento de lectura (completando una transferencia completa DMA) cuando se produce la petición de una interrupción habilitada para la generación de eventos DMA.
1-0	SINCRONIZACIÓN DEL DESTINO. La sincronización de los eventos de escritura de la DMA se realiza con las peticiones de interrupción. La DMA genera un evento de escritura (completando una transferencia completa DMA e iniciando otra) cuando se produce la petición de una interrupción habilitada para la generación de eventos DMA.
1-1	SINCRONIZACIÓN DE LA FUENTE Y DEL DESTINO. La sincronización de los eventos de lectura y escritura de la DMA se realiza con las peticiones de interrupción. La DMA genera un evento de lectura o escritura (según corresponda) cuando se produce la petición de una interrupción habilitada para la generación de eventos DMA. Una transferencia DMA se realiza cuando se producen dos peticiones de interrupción que tengan habilitada la generación de eventos DMA.

Tabla 4.24. Modo de funcionamiento del periférico DMA: sincronización de los eventos. Funciones de los bits SYNC del registro de control y configuración del periférico de la familia TMS320C3x.

TCINT-TC	Función que realiza
0-0	La CPU no es notificada cuando se completa una transferencia DMA (el registro contador de la transferencia pasa a valer cero). Se continúan realizando transferencias DMA cuando pasa a valer cero el registro contador de transferencias DMA.
0-1	La CPU no es notificada cuando se completa una transferencia DMA (el registro contador de la transferencia pasa a valer cero). Se dejan de realizar transferencias DMA cuando pasa a valer cero el registro contador de transferencias DMA.
1-0	La CPU es notificada, vía activación de la bandera de petición de interrupción

	asociada al periférico DMA, cuando se completa una transferencia DMA (el registro contador de la transferencia pasa a valer cero). Se continúan realizando transferencias DMA cuando pasa a valer cero el registro contador de transferencias DMA.
1-1	La CPU es notificada, vía activación de la bandera de petición de interrupción asociada al periférico DMA, cuando se completa una transferencia DMA (el registro contador de la transferencia pasa a valer cero). Se dejan de realizar transferencias DMA cuando pasa a valer cero el registro contador de transferencias DMA.

Tabla 4.25. Modos de conclusión de las transferencias DMA programadas en la familia TMS320C3x.

5.1.2 Modos de operación: sincronización de los eventos DMA

El usuario puede programar cuatro formas de realización de las transferencias DMA, según la sincronización impuesta entre la fuente y el destino (campo SYNC del registro de control y configuración del periférico).

Las transferencias de tipo DMA se realizan mediante una parada controlada al final de uno, los dos o ninguno de los eventos de los que se compone la transferencia DMA (*read_DMA* y *write_DMA*). Las peticiones de las interrupciones habilitadas para generar eventos DMA son las que se encargan de relanzar, cuando dichas peticiones aparecen, las transferencias DMA:

- SYNC=00. Sin sincronización entre la fuente y el destino. Las interrupciones habilitadas para generar eventos DMA son ignoradas, no afectan a los procesos de transferencia que se realizan cuando no haya conflictos en la estructura *pipeline*. No existe ningún control sobre cuando se realizan las operaciones elementales asociadas a la DMA.
- SYNC=01. Sincronización de la fuente. La DMA no realiza transferencias mientras no llegue alguna petición de interrupción que tenga habilitada la generación de eventos de tipo DMA. Cuando llega, se inicia la transferencia DMA, realizándose primero un evento de lectura (*read-DMA*) y, a continuación, se lleva a cabo un evento de escritura (*write-DMA*). Existe, por tanto, un control sobre cuando se realizan las operaciones elementales de la DMA que consiste en insertar una parada controlada del periférico DMA al comienzo de los eventos *read-DMA*.
- SYNC=10. Sincronización del destino. Como en el caso anterior, la DMA no realiza

transferencias mientras no llegue alguna interrupción que tenga habilitada la generación de eventos de tipo DMA. Cuando llega, se inicia la transferencia DMA, realizándose primero un evento de escritura (*write-DMA*) y, a continuación, se lleva a cabo un evento de lectura (*read-DMA*). La parada que controla la transferencia DMA se ubica, en este caso, al comienzo de cada evento *write-DMA*.

- SYNC=11. Sincronización de la fuente y el destino. Antes de cada evento de lectura o escritura DMA hay que esperar a que ocurra una interrupción que tenga habilitada la generación de eventos de tipo DMA. Existe, por tanto, dos paradas controladas, situadas al comienzo de los eventos *read-DMA* y *write-DMA*, que temporizan la realización de las operaciones elementales del periférico DMA.

5.1.3 Método de configuración

Mientras no se programe, la DMA permanece inactiva (después de un *reset* el periférico DMA para las transferencias en curso y no realiza trasvase de datos). Para poder usar el periférico DMA del DSP, primero es necesario configurarlo. La configuración debe realizarse, según aconseja el fabricante, manteniendo inactivo el dispositivo. En primer lugar, se debe parar el periférico poner los dos bits START del registro global de control y configuración de la DMA a cero. Posteriormente, se realiza la configuración del periférico manteniendo los bits START a cero se escribe en el registro global de control y configuración de la DMA, se programa la dirección de origen y destino así como el número de datos a transferir, y, finalmente, se habilita el periférico se ponen las banderas START a un valor distinto de 00.

5.2 TMS320C6000

El controlador DMA de la familia TMS320C6000, que aparece en la mayoría de los dispositivos de las versiones TMS320C620x y TMS320C670x, realiza básicamente las mismas funciones que el controlador DMA de la familia TMS320C3x con la diferencia de soportar cinco canales, frente a sólo uno en el TMS320C3x, y de disponer de métodos de transferencia más versátiles, que redundan en un aumento considerable

del rendimiento del sistema. De hecho, cualquier aplicación de procesamiento digital de señales se implementa apoyándose en la potencia de estos controladores, para mover bloques de memoria de manera concurrente con la ejecución de un algoritmo por parte de la CPU. El controlador DMA de la familia TMS320C6000 puede realizar: transferencias de elementos de una zona de memoria a otra (hay que recordar que los periféricos están mapeados como posiciones de memoria), transferencias de una trama formada por varios elementos y transferencias por bloques en las que cada uno de estos bloques está formado por varias tramas.

Las características básicas de este controlador DMA se resumen en:

- Alto rendimiento: la información se puede trasvasar usando el reloj de la CPU.
- Cuatro canales disponibles, que aportan otros cuatro contextos diferentes para las operaciones DMA. A ellos hay que añadir un canal auxiliar que permite al controlador atender peticiones del periférico HPI.
- Posibilidad de utilizar cada canal DMA para realizar dos transferencias completas de tipo DMA: operaciones que permite tener, en realidad, dos canales en uno. Este modo de funcionamiento se denomina *split*.
- Transferencias de bloques de datos, compuestos por múltiples tramas cada una de las cuales se compone, a su vez, de varios elementos. Tanto el número de tramas que componen un bloque como el número de elementos que componen la trama son programables.
- Prioridad programable. Cada uno de los canales puede configurarse como más o menos prioritario que la CPU. Aunque la prioridad relativa de los canales 0 a 3 de la DMA está fijada (el canal más prioritario es el 0 y el menos prioritario el 3), el orden de prioridad del canal auxiliar puede establecerse en cualquier nivel desde más prioritario que el canal 0 hasta menos prioritario que el canal 3.
- Variación de las direcciones de los datos origen y destino programable. Los registros de dirección de origen y destino pueden mantenerse fijos o incrementarse y decrementarse, después de transferirse un elemento y una trama, en un valor programable. Además, el valor programado puede cambiar de una transferencia a otra.

- Acceso completo al espacio de memoria interna de datos y programa, de los periféricos, del bus EMIF y del bus de expansión.
- Configuración del ancho del elemento transferido: 8, 16 ó 32 bits.
- Posibilidad de recarga del controlador, para que se inicialice de nuevo una vez finalizada la transferencia en curso. Modo de funcionamiento denominado autoinicialización.
- Sincronización de las transferencias, que pueden iniciarse en función de múltiples eventos programables por el usuario.
- Generación de interrupciones. Cada vez que se completa una transferencia, o cuando se produzca algún error, el controlador puede provocar una interrupción a la CPU.

5.2.1 Modos de funcionamiento básico

La tabla 4.26 muestra los registros de control del periférico. Los registros de control primario, secundario y del canal auxiliar permiten configurar el modo de funcionamiento de los canales de la DMA, mientras que los registros de dirección fuente y destino permiten especificar la ubicación de las posiciones de memoria entre las que se realizará el trasvase de datos. Cada canal DMA se puede lanzar independientemente, de manera manual o automática, mediante el procedimiento conocido como autoinicialización. Una vez que la transferencia ha comenzado, los canales se pueden parar o pausar mediante la modificación de los bits START en los registros PRICTL (en cuyo caso, es necesario esperar a que el campo STATUS, en el mismo registro, refleje la nueva situación):

- El arranque manual se realiza escribiendo el valor '01' en el campo START. Previamente, hay que haber inicializado los registros correspondientes: SRC0 a SRC3 (registros de dirección fuente de los canales 0 a 3), DST0 a DST3 (registros de dirección destino de los canales 0 a 3) y XFRCNT0 a XFRCNT 3 (registros de cuenta de transferencias de los canales 0 a 3).
- La operación de pausa se realiza escribiendo el valor '10' en el campo START. En este caso, se completarán aquellas transferencias en las que se haya leído el elemento (se ha generado un evento *read_DMA*).

- La operación de parada se realiza escribiendo '00' en el campo START.
- Cuando el controlador de DMA ha completado la transferencia de información, es necesario reprogramarlo. Este proceso se puede realizar de forma automática, por medio de los registros globales de la DMA en los que se pueden programar recargas de los registros del controlador para que éste siga realizando transferencias. Así, se puede programar la autoinicialización para tener un modo de funcionamiento continuo siendo, por tanto, necesario modificar el contenido de los registros de recarga. Para activar el modo de autoinicialización se deben fijar los bits START del registro PRICTL al valor '11'. Los registros internos del controlador GBLADDRB, GBLADDRC, GBLADDRD, GBLIDXA, GBLIDXB y GBLCNTA, GBLCNTB se encargan de almacenar las direcciones, los índices y los parámetros de recarga.

Dirección	Acrónimo	Nombre
01840000	PRICTL0	Registro de control primario del canal 0 de la DMA.
01840004	PRICTL2	Registro de control primario del canal 2 de la DMA.
01840008	SECCTL0	Registro de control secundario del canal 0 de la DMA.
0184000C	SECCTL2	Registro de control secundario del canal 2 de la DMA.
01840010	SRC0	Registro de dirección fuente del canal 0 de la DMA.
01840014	SRC2	Registro de dirección fuente del canal 2 de la DMA.
01840018	DST0	Registro de dirección destino del canal 0 de la DMA.
0184001C	DST2	Registro de dirección destino del canal 2 de la DMA.
01840020	XFRCNT0	Registro de cuenta de transferencias del canal 0 de la DMA.
01840024	XFRCNT2	Registro de cuenta de transferencias del canal 2 de la DMA.
01840028	GBLCNTA	Registro global A de recarga de la cuenta de la DMA.
0184002C	GBLCNTB	Registro global B de recarga de la cuenta de la DMA.
01840030	GBLIDXA	Registro A de índice global.
01840034	GBLIDXB	Registro B de índice global.
01840038	GBLADDRA	Registro A de dirección global de la DMA.
0184003C	GBLADDRB	Registro B de dirección global de la DMA.
01840040	PRICTL1	Registro de control primario del canal 1 de la DMA.
01840044	PRICTL3	Registro de control primario del canal 3 de la DMA.
01840048	SECCTL1	Registro de control secundario del canal 1 de la DMA.
0184004C	SECCTL3	Registro de control secundario del canal 3 de la DMA.
01840050	SRC1	Registro de dirección fuente del canal 1 de la DMA.
01840054	SRC3	Registro de dirección fuente del canal 3 de la DMA.
01840058	DST1	Registro de dirección destino del canal 1 de la DMA.
0184005C	DST3	Registro de dirección destino del canal 3 de la DMA.
01840060	XFRCNT1	Registro de cuenta de transferencias del canal 1 de la DMA.
01840064	XFRCNT3	Registro de cuenta de transferencias del canal 3 de la DMA.
01840068	GBLADDRC	Registro C de dirección global de la DMA.
0184006C	GBLADDRD	Registro D de dirección global de la DMA.
01840070	AUXCTL	Registro de control auxiliar de la DMA.

Tabla 4.26. Ubicación de los diferentes registros de control de la DMA de la familia TMS320C6000.

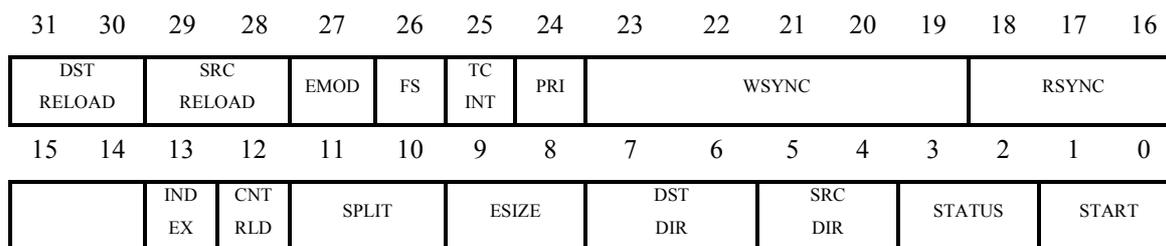


Fig.4.18. Registro de control primario de la DMA (PRICTL) de la familia TMS320C6000.

Bit	Nombre	Tras reset	Función que realiza
1-0	START	00	Controla el funcionamiento del canal correspondiente de la DMA. Si START=00, la DMA deja de transferir información. Si START=01, la DMA comienza a transferir información en modo de no autoinicialización. Si START=10, el controlador de DMA se pone en pausa. Si START=11, el controlador comienza a transferir información en modo de autoinicialización.
3-2	STATUS	00	Indica el estado del controlador de DMA. Si STATUS=00, el controlador está parado. Si STATUS=01, está funcionando sin haberse autoinicializado. Si STATUS=10, está en pausa. Si STATUS=11, el controlador está funcionando habiéndose autoinicializado.
5-4	SRCDIR	00	Indica cómo se modifica la dirección fuente en cada transferencia. Si SRCDIR=00, la dirección fuente no cambia. Si SRCDIR=01 o SRCDIR=10, se incrementa o decrementa, respectivamente, según el número de bytes que tenga el elemento a transmitir. Si SRCDIR=11 se usa el registro de índice global de la DMA como elemento de modificación.
7-6	DSTDIR	00	Funcionalidad idéntica a la de SRCDIR, pero manejando la dirección de destino.
9-8	ESIZE	00	Tamaño del elemento a transferir. '00' indica 1 byte, '01' 2 bytes, '10' cuatro bytes y el valor '11' está reservado.
11-10	SPLIT	00	Activación/desactivación del modo <i>split</i> . Con '00', está desactivado y con '01', '10' y '11' se utilizan los registros A, B y C de la dirección global de la DMA como elementos de almacenamiento de la dirección.
12	CNT_RELOAD	0	Indicación de recarga cuando se está en el modo de autoinicialización. Con '0', se utiliza el registro global A de recarga de la cuenta de la DMA y con '1', el registro global B de recarga de la cuenta de la DMA.
13	INDEX	0	Selecciona el registro de índice global a utilizar. Con INDEX=0, se utiliza el registro A y con INDEX=1 el B.
18-14	RSYNC	0-0	Selecciona el evento que sincronizará la operación <i>read DMA</i> .
23-19	WSYNC	0-0	Selecciona el evento que sincronizará la operación <i>write DMA</i> .
24	PRI	0	Selecciona la prioridad entre el canal correspondiente del controlador DMA y la CPU. Con PRI=0, la CPU es prioritaria y con PRI=1 el canal del controlador DMA es el de mayor nivel de prioridad.
25	TCINT	0	Con TCINT=1, se habilita la interrupción cuando finaliza una

			transferencia completa del controlador DMA. Con TCINT=0, está deshabilitada.
26	FS	0	Sincronización de la trama. Con FS=0, está deshabilitada. Con FS=1, se utiliza el evento especificado en RSYNC para sincronizar una trama.
27	EMOD	0	Modo de emulación. Con EMOD=0, el controlador DMA sigue funcionando durante una parada del emulador. Con EMOD=1, el controlador se queda en pausa.
29-28	SRCRELOAD	00	Configura la dirección fuente a utilizar para realizar la recarga en una autoinicialización. Con SRCRELOAD=00, no se recarga. Con SRCRELOAD=01, se utiliza el registro B de dirección global de la DMA. Con SRCRELOAD=10, se utiliza el registro C. Con SRCRELOAD=11, se usa el registro D.
31-30	DSTRELOAD	00	Configura la dirección destino a utilizar para realizar la recarga en una autoinicialización. Con DSTRELOAD=00, no se recarga. Con DSTRELOAD=01, se utiliza el registro B de dirección global de la DMA. Con DSTRELOAD=10, se utiliza el registro C. Con DSTRELOAD=11, se usa el registro D.

Tabla 4.27 Función de los bits del registro PRICTL del controlador de la DMA de la familia TMS320C6000.

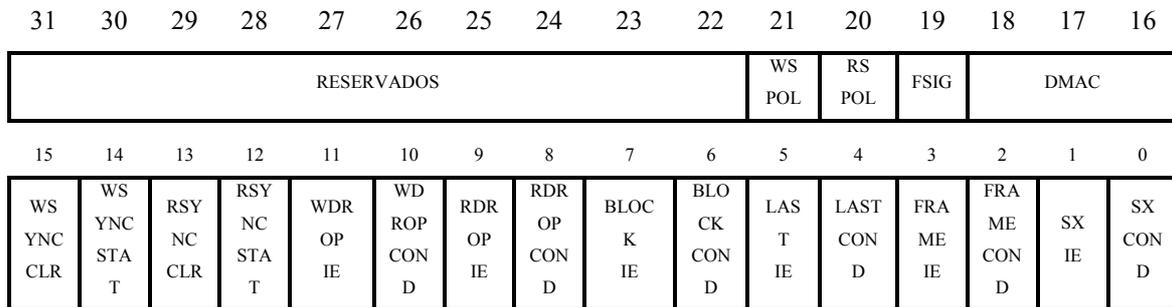


Fig. 4.19. Registro de control secundario (SECCTL) del controlador de la DMA, familia TMS320C6000.

Bit	Nombre	Tras reset	Función que realiza
0	SX COND	0	Modo de funcionamiento <i>split</i> en transmisión. Si SX COND=0, no se detecta y si SX COND=1, sí se detecta.
1	SX IE	0	Habilitación de interrupción por sobreescritura en modo <i>split</i> .
2	FRAME COND	0	Condición de trama completa. Si FRAME COND=0 no se detecta y si FRAME COND=1, sí se detecta.
3	FRAME IE	0	Habilitación de interrupción por trama completa. Si FRAME IE=1, se habilita, y con '0', se deshabilita.
4	LAST COND	0	Condición de detección de la última trama. Con LAST COND=1, se detecta.
5	LAST IE	0	Habilitación de la interrupción por última trama. Con LAST IE=1, se habilita, y con '0' se deshabilita.
6	BLOCK COND	0	Condición de detección de último bloque. Con BLOCK COND=1, se habilita.

7	BLOCK IE	0	Habilitación de interrupción por detección del último bloque. Con BLOCK IE=1, se habilita, con '0' se deshabilita.
8	RDROP COND	0	Condición de detección de un evento de sincronización de lectura cuando el anterior no ha sido borrado.
9	RDROP IE	0	Habilitación del evento RDDROP. Con RDROP=1 se habilita, con '0' se deshabilita.
10	WDROP COND	0	Condición de detección de un evento de sincronización de escritura cuando el anterior no ha sido borrado.
11	WDROP IE	0	Habilitación del evento WDDROP. Con WDROP=1 se habilita, con '0' se deshabilita.
12	RSYNC STAT	0	Estado de la sincronización en lectura. RSYNC STAT=0 indica que no se ha recibido la sincronización para leer el elemento. Un '1' indica que se ha recibido.
13	RSYNC CLR	0	Escribiendo un '1' en este bit, se pone a cero el bit RSYNC STAT.
14	WSYNC STAT	0	Estado de la sincronización en escritura. WSYNC STAT=0 indica que no se ha recibido la sincronización para escribir el elemento. Un '1' indica que se ha recibido.
15	WSYNC CLR	0	Escribiendo un '1' en este bit, se pone a cero el bit WSYNC STAT.
18-16	DMAC EN	000	Configuración del estado del pin DMAC. Las opciones válidas son: DMAC EN=000, mantiene DMAC a cero. DMAC EN=001, mantiene DMAC a uno. DMAC EN=010, DMAC refleja el estado de RSYNC STAT. DMAC EN=011, DMAC refleja el estado de WSYNC STAT. DMAC EN=101, DMAC refleja el estado de FRAME COND. DMAC EN=100, DMAC refleja el estado de BLOCK COND.
19	FSIG	0	No aplicable a TMS320C6201/C6701. Configuración del modo de detección de la señal FS: Por flanco (FSIG=0) o por nivel (FSIG=1).
20	RSPOL	0	Polaridad del evento de lectura DMA asociado a las interrupciones externas. Con RSPOL=0, el evento es activo a nivel alto y con RSPOL=1 a nivel bajo. Este campo sólo es válido si se han seleccionado las interrupciones externas EXT_INTx como eventos para sincronizar las transferencias DMA.
21	WSPOL	0	Polaridad del evento de escritura DMA asociado a las interrupciones externas. Con WSPOL=0, el evento es activo a nivel alto y con WSPOL=1 a nivel bajo. Este campo sólo es válido si se han seleccionado las interrupciones externas EXT_INTx como eventos para sincronizar las transferencias DMA.
31-22	Reservados	0-0	Se lee siempre un cero.

Tabla 4.28. Elementos del registro de control secundario (SECCTL) de la DMA de la familia TMS320C6000.

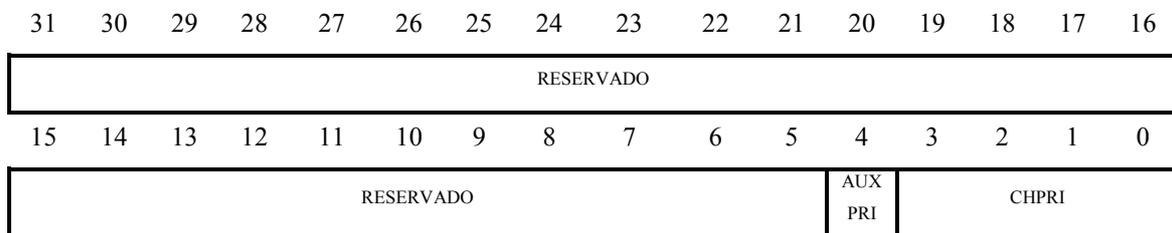


Fig. 4.20. Registro de control del canal auxiliar de la DMA (AUXCTL), familia TMS320C6000.

Bit	Nombre	Tras reset	Función que realiza
3-0	CHPRI	0000	Prioridad asignada al canal auxiliar de la DMA. Los valores válidos son: CHPRI=0. El canal auxiliar es el más prioritario (más prioritario que el canal 0). CHPRI=1. El canal auxiliar es el segundo más prioritario (menos prioritario que el canal 0 y más que el canal 1). CHPRI=2. El canal auxiliar es el tercero más prioritario (menos prioritario que el canal 1 y más que el canal 2). CHPRI=3. El canal auxiliar es el cuarto más prioritario (menos prioritario que el canal 2 y más que el canal 3). CHPRI=4. El canal auxiliar es el menos prioritario de los canales DMA (menos prioritario que el canal 3).
4	AUXPRI	0	Selecciona la prioridad entre el canal auxiliar del controlador DMA y la CPU. Con AUXPRI=0, la CPU es prioritaria y con AUXPRI=1 el canal auxiliar del controlador DMA es el de mayor nivel de prioridad.
31-5	Reservados	0-0	Se lee siempre un cero.

Tabla 4.29. Elementos del registro de control del canal auxiliar de la DMA (AUXCTL) en la familia TMS320C6000.

El registro SRC contiene la dirección de 32 bits que se utiliza para obtener el dato que se desea transferir y el registro DST contiene la dirección de 32 bits donde se almacenará dicho dato. Lógicamente, estos registros están duplicados en cada uno de los canales que posee el controlador DMA. El controlador DMA de la familia TMS320C6000 organiza el bloque de datos a transferir en un número de tramas programable, que componen la transferencia, y en un número de elementos también programable que se incluyen en cada trama. Esta información se especifica en el registro XFRCNT. El campo de 16 bits denominado ELEMENT COUNT indica el número de elementos que hay en cada trama y, por tanto, se puede programar en él un valor de hasta 65535. El campo de 16 bits denominado FRAME COUNT indica el número de tramas de la transferencia y su valor máximo, al igual que en el caso anterior, es 65535. A medida que el controlador DMA realice transferencias, este registro se irá decrementando. Cada vez que se transfiere un elemento, la dirección fuente y destino se pueden incrementar o decrementar en función del campo de 16 bits denominado ELEIDX de los registros índice globales de la DMA (GBLIDXA o GBLIDXB). El valor contenido en dicho campo permite incrementar la dirección del siguiente elemento a transferir, respecto de la dirección del elemento transferido, en un valor que varía de -32768 hasta 32767. De forma análoga, al final de la transferencia de una trama, el controlador DMA actualiza la dirección del siguiente elemento a transferir en función del campo de 16 bits denominado FRMIDX de los registros índice globales de la DMA

(GBLIDXA o GBLIDXB). El valor contenido en dicho campo permite incrementar la dirección del siguiente elemento a transferir, respecto de la dirección del último elemento de una trama transferido, en un valor que varía de -32768 hasta 32767.

El controlador DMA soporta sincronización en lectura y escritura de acuerdo a los eventos indicados en la tabla 4.30. Así, por ejemplo, se puede programar que el controlador realice la transferencia de una palabra cuando se produce una interrupción del temporizador 0 (TINT0). Además, soporta sincronización a nivel de trama, de forma que la transferencia de la trama también se puede sincronizar con los eventos indicados en la tabla. El tipo de sincronización se selecciona con los bits RSYNC y WSYNC de los registros PRICTRL0 a PRICTRL3, cuando FS=1.

Código del evento	Tipo	Descripción
00000	None	Sin sincronización.
00001	TINT0	Interrupción temporizador 0
00010	TINT1	Interrupción temporizador 1.
00011	SD_INT	Interrupción del temporizador de gestión de la SDRAM conectada al EMIF.
00100	EXT_INT4	Interrupción externa 4.
00101	EXT_INT5	Interrupción externa 5.
00110	EXT_INT6	Interrupción externa 6.
00111	EXT_INT7	Interrupción externa 7.
01000	DMA_INT0	Interrupción del canal DMA 0.
01001	DMA_INT1	Interrupción del canal DMA 1.
01010	DMA_INT2	Interrupción del canal DMA 2.
01011	DMA_INT3	Interrupción del canal DMA 3.
01100	XEVT0	Evento de transmisión del McBSP 0.
01101	REVT0	Evento de recepción del McBSP 0.
01110	XEVT1	Evento de transmisión del McBSP 1.
01111	REVT1	Evento de recepción del McBSP 0.
10000	DSPINT	Interrupción del HOST al DSP.
10001	XEVT2	Evento de transmisión del McBSP 2.
10010	REVT2	Evento de recepción del McBSP 2.
Resto de valores	Reservados	

Tabla 4.30. Eventos que utiliza el controlador DMA de la familia TMS320C6000 para sincronizar las transferencias.

5.2.2 Gestión de los accesos al bus del controlador DMA

La figura 4.21 muestra la estructura interna del *hardware* de gestión de acceso a los buses del DSP en el controlador DMA que se incorpora en los dispositivos

TMS320C6201/C6701 y TMS320C6702. Como se puede ver, existen dos registros asociados a los canales 0 a 3 y un registro asignado al canal auxiliar, denominados *holding registers*. Además, y exclusivamente para operaciones DMA que se realicen por ráfagas, existe una memoria FIFO de nueve posiciones. Cualquier operación de lectura realizada sobre el EMIF, el bus de expansión, la memoria de datos y programa, el HPI o el bus de periféricos se almacena en estos *holding registers*, que actúan a modo de FIFO de dos posiciones (excepto para el canal auxiliar que sería de una única posición). Lógicamente, si el proceso de lectura se realiza tan rápidamente que no se vacía el contenido de estos registros, se pierde rendimiento en la arquitectura. De esta manera, y orientado exclusivamente a los modos de ráfaga (*burst*), se incluye una FIFO de nueve posiciones que permite almacenar hasta nueve datos procedentes de los buses mencionados. Para tener acceso a la FIFO, lo que es gestionado por el controlador DMA, es necesario que se cumplan las siguientes condiciones:

1. No se debe haber activado la opción de sincronización en lectura o escritura mediante eventos.
2. La FIFO debe estar vacía de elementos de otro canal.
3. El canal debe tener una prioridad mayor que otros canales que también cumplan las condiciones anteriores.

La utilización de la FIFO permite realizar las operaciones de lectura de una manera independiente de las operaciones de escritura, a la velocidad del reloj de la CPU, permitiendo la obtención de un elevado rendimiento en la transferencia de los datos.

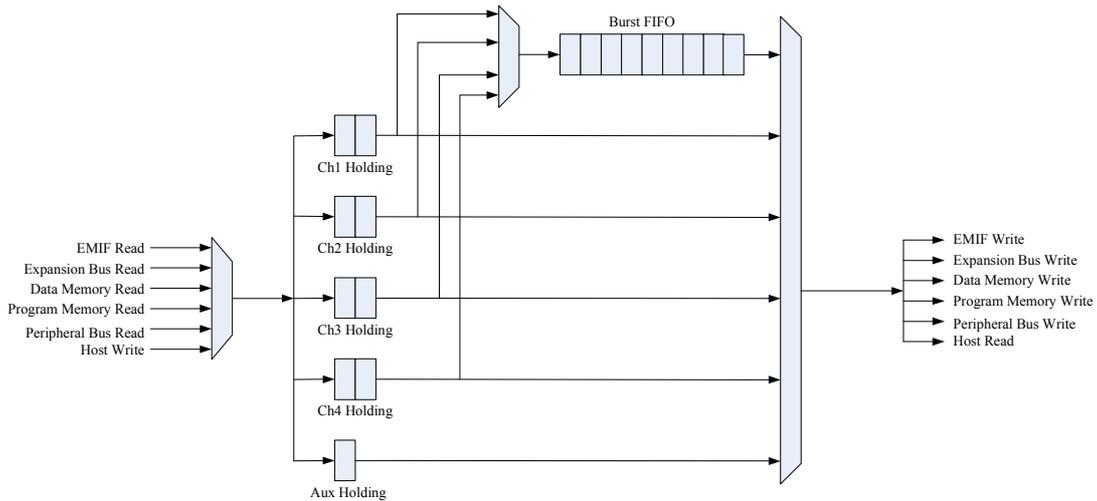


Fig. 4.21. Gestión del acceso al bus en el controlador DMA de los dispositivos TMS320C6201/C6701/C6702.

Para aumentar el rendimiento del controlador en los dispositivos TMS320C6202/C6203/C6204/C6205, se ha eliminado el arbitraje necesario para compartir la FIFO y se ha dotado a cada canal de una FIFO propia, a excepción del canal auxiliar. La estructura interna de estos controladores DMA se muestra en la figura 4.22.

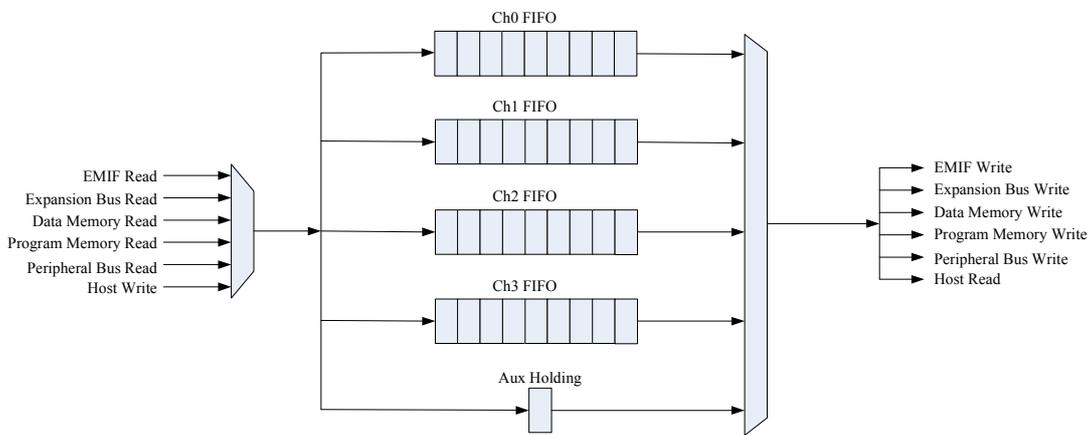


Fig. 4.22. Estructura mejorada del acceso a los bus en el periférico DMA de los dispositivos TMS320C6202/C6203/C6204/C6205.

5.2.3 El controlador DMA mejorado del TMS320C6000: EDMA

El controlador de acceso directo a memoria mejorado (*Enhanced* DMA ó EDMA) es una evolución del controlador DMA que desarrolla Texas Instruments para los dispositivos TMS320C6201 y TMS320C6701 y que también está disponible en los dispositivos TMS320C621x/C671x/C64x.

El dispositivo incluye canales para 16 eventos, en los dispositivos TMS320C621x y TMS320C671x, y 64 eventos, en los TMS320C64x. Como características más novedosas indicar que este controlador gestiona las transferencias de datos entre la memoria caché de nivel L2 y los periféricos internos del DSP o los dispositivos externos conectados con la CPU mediante el controlador EMIF o el interfaz HPI. Además, es capaz de enlazar varias transferencias consecutivas y de gestionar las prioridades de los distintos canales que puede atender.

Internamente el controlador EDMA está formado por un conjunto de registros de gestión de eventos e interrupción, un codificador de eventos, un bloque de memoria RAM (que almacena los parámetros de configuración y enlace o recarga de cada canal) y un generador de direcciones. Cuando se produce un evento, éste se decodifica disparando el canal de EDMA correspondiente y enviando la información almacenada en la RAM de parámetros al generador de direcciones de acceso. Aunque la EDMA está orientada a la gestión de transferencias con los periféricos, también se puede utilizar para el movimiento rápido de bloques de datos cuando así son solicitados por la CPU (operación que se denomina QDMA, *Quick DMA*, y que se analizará brevemente al final de este capítulo).

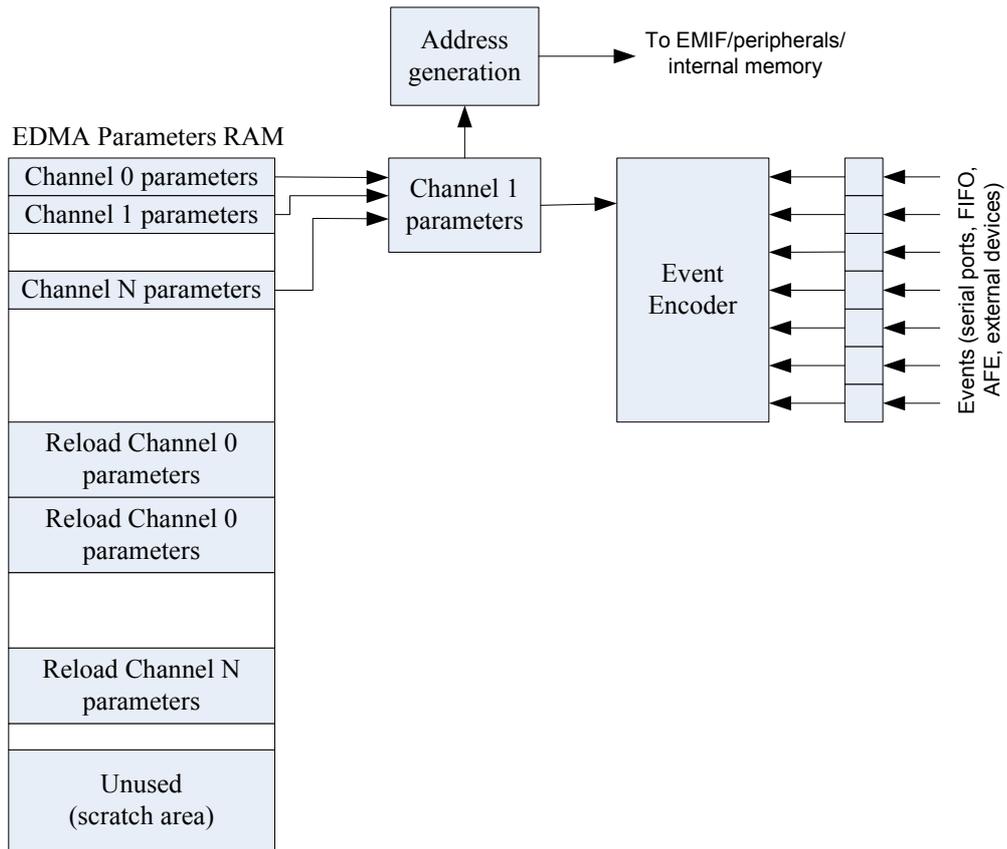


Fig. 4.23. Estructura interna del controlador EDMA.

El controlador EDMA permite realizar los siguientes tipos de transferencias:

1. Transferencias de tramas, entendiéndose éstas como elementos que pueden ser contiguos o escalonados en las direcciones de memoria del DSP.
2. Transferencias de *arrays* o cadenas de información, conjunto de elementos contiguos en memoria.
3. Transferencias de bloques o conjunto de tramas (transferencias de una dimensión o 1D) o arrays (transferencias de dos dimensiones o 2D).

El periférico no tiene, como tal, una batería de registros que configuren las transferencias, sino que existe una zona de memoria de 2 Kbytes de RAM, donde se deben programar los parámetros de cada una de las transferencias asociadas a los canales EDMA. Los parámetros asignados a cada canal son seis palabras de 32 bits, existiendo la posibilidad de programar hasta 85 entradas diferentes de las cuales, en los

dispositivos TMS320C621x/C671x se utilizan un mínimo 16 y en los TMS320C64x de 64. De esta manera, se pueden programar las transferencias asociadas a cada canal y los parámetros de transferencias que se utilizarán para llevar a cabo transferencias enlazadas. Así, los registros de dirección fuente (SRC) y destino (DST) contienen los valores de las direcciones entre las que se realizará el movimiento de datos. El parámetro ELECNT, de 16 bits, contiene el número de elementos asociados a cada trama, si la transferencia es del tipo 1D, o *array*, si la transferencia es del tipo 2D. El parámetro FRMCNT indica el número de tramas, menos una, en un bloque si la transferencia es del tipo 1D o el número de *arrays*, menos uno, si la transferencia es del tipo 2D. El parámetro ELEIDX proporciona el desplazamiento para obtener el siguiente elemento en la transferencia de una trama, por lo que sólo tiene sentido en transferencias del tipo 1D, en las que está permitido el movimiento de elementos no contiguos en memoria. El parámetro FRMIDX contiene el desplazamiento entre tramas asociadas al mismo bloque, en transferencias del tipo 1D, o el desplazamiento entre *arrays* asociados al mismo bloque, en transferencias del tipo 2D. El parámetro ELERLD indica el valor a recargar en el campo ELECNT cuando se transfiere el último elemento de una trama y el parámetro LINK contiene 16 bits que permiten formar la dirección donde se encuentran los parámetros de configuración del canal que se deben recargar cuando finalice la transferencia en curso (la dirección se compone anexando el valor 0x01A0 al valor contenido en éste parámetro).

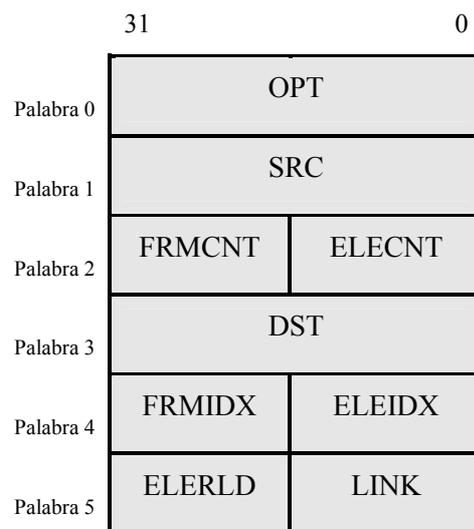


Fig. 4.24. Parámetros de configuración de un canal de EDMA.

El parámetro de configuración denominado OPT permite configurar el comportamiento del canal EDMA.

Bit	Nombre	Tras reset	Función que realiza
0	FS	0	Sincronización de trama. Si FS=0, el canal se sincroniza para elementos o <i>arrays</i> . Si FS=1, el canal está sincronizado para tramas.
1	LINK	0	Con LINK=0, el enlazado esta deshabilitado. Con LINK=1, cuando finaliza la transferencia en curso se reprograma una nueva con los parámetros de la dirección especificada en la dirección de enlace (parámetro de configuración denominado LINK).
2	PDTD	0	Modo <i>peripheral device transfer</i> (PDT) para el destino. Si PDTD=0, la operación de escritura PDT está desactivada. Si PDTS=1, está activada. El modo PDT, sólo válido en el TMS320C64x, permite mover datos entre dispositivos que están ubicados en el EMIF sin necesidad de realizar una operación de lectura y escritura del EMIF por parte del controlador EDMA: con una única operación, la información se transfiere de un dispositivo a otro.
3	PDTS	0	Modo PDT para la fuente u origen. Si PDTS=0, la operación de lectura PDT está desactivada. Si PDTS=1 está activada.
4	Reservado	0	Se lee siempre un cero.
10-5	ATCC	0-0	Sólo se aplica en los dispositivos TMS320C64x. Código alternativo de transferencia completa. Código de 5 bits que proporciona información sobre la finalización de un movimiento intermedio en una transferencia de un bloque.
11	Reservado	0	Se lee siempre un cero.
12	ATCINT	0	Sólo se aplica en los dispositivos TMS320C64x. Habilitación de interrupción por la aparición de las condiciones intermedias. Con ATCINT=0, está deshabilitado, y con ATCINT=1, está habilitado.
14-13	TCCM	00	Sólo se aplica en los dispositivos TMS320C64x. Bits (los MSBs) que completan el campo TCC, formando el código de transferencia completada.
15	Reservado	0	Se lee siempre un cero.
19-16	TCC	0000	Código de transferencia completada.
20	TCINT	0	Habilitación de la generación de interrupción al finalizar una transferencia.
22-21	DUM	00	Modo de actualización de la dirección de destino. Si DUM=00, la dirección se mantiene fija. Si DUM=01, se incrementa en función de los bits FS y 2DD. Si DUM=10, se decrementa en función de los bits FS y 2DD. Si DUM=11, se modifica en función de los campos ELEIDX y FRMIDX, dependiendo de los bits FS y 2DD.
23	2DD	0	Dimensión del destino. Si 2DD=0, el destino es de una dimensión y si 2DD=1, la destino es de dos dimensiones.
25-24	SUM	00	Modo de actualización de la dirección de origen. Si SUM=00, la dirección se mantiene fija. Si SUM=01, se incrementa en función de los bits FS y 2DS. Si SUM=10, se decrementa en función de los bits FS y 2DS. Si SUM=11, se modifica en función de los campos ELEIDX y FRMIDX, dependiendo de los bits FS y 2DS.
26	2DS	0	Dimensión de la fuente. Si 2DS=0, se maneja información en una

			dimensión, y si 2DS=1, la información manejada es de dos dimensiones.
28-27	ESIZE	00	Tamaño del elemento a mover. Si ESIZE=00, es 1 byte. Si ESIZE=01, son 2 bytes. Si ESIZE=10, es una palabra con cuatro bytes (8 bytes en algunas versiones del TMS320C64x). La opción ESIZE=11, está reservada.
31-29	PRI	000	Código que especifica el nivel de prioridad del canal. En los TMS320C621x/C671x, los valores no reservados son: <ul style="list-style-type: none"> • PRI=000: prioridad reservada para peticiones a la caché L2 que no se puede utilizar para transferencias normales. • PRI=001: transferencia EDMA de alta prioridad. • PRI=010, transferencias EDMA de baja prioridad. En los TMS320C64x, los valores no reservados son: <ul style="list-style-type: none"> • PRI=000: prioridad urgente. • PRI=001: transferencia de prioridad alta. • PRI=010: transferencias de prioridad media. • PRI=011, transferencias de prioridad baja.

Tabla 4.31. Bits del parámetro OPT de configuración de un canal de la familia TMS320C6000.

Una transferencia de tipo 1D se caracteriza por transferir un conjunto de elementos de tamaño programable (una trama) que no tienen por qué estar localizados en posiciones contiguas de memoria. Los elementos están separados según la información especificada en el campo ELEIDX, que aparece en los parámetros de configuración de la EDMA, mientras que el parámetro FRMIDX especifica el salto que hay que dar desde el último elemento que compone una trama para localizar el comienzo de la siguiente. La sincronización de la transferencia de los elementos en este modo de funcionamiento se puede hacer de dos maneras:

- Cada vez que se produce un evento se transfiere un elemento.
- Cuando se activan las transferencias sincronizadas por trama lo que implica que cada vez que se produce un evento se transfiere la trama entera.

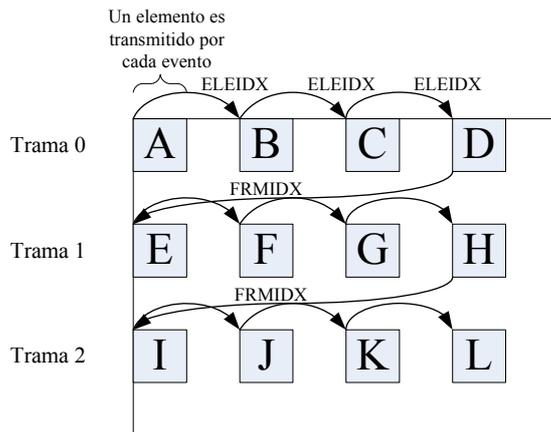


Fig. 4.24. Transferencia 1D sin sincronización de trama.

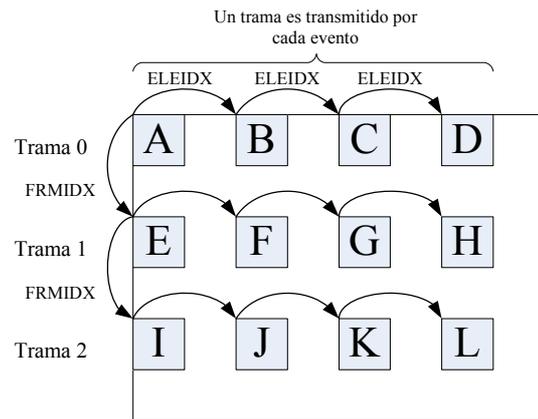


Fig. 4.25. Transferencia de 1D con sincronización de trama.

Las transferencias 2D son muy útiles en procesamiento de imágenes. En este caso, los elementos son contiguos en el rango de direcciones de memoria del DSP y el campo ELEIDX no se utiliza. Cada bloque está formado por un número de *arrays* que, a su vez, contienen varios elementos. La transferencia EDMA de tipo 2D se sincroniza también de dos formas:

- Cuando se configura la sincronización a nivel de *array*, cada vez que se produce un evento se transfiere un *array* completo.
- Cuando se configura la sincronización a nivel de bloque, cada vez que se produce un evento se transfiere el bloque completo.

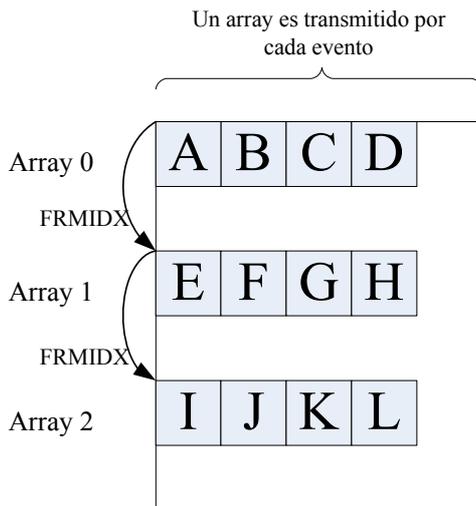


Fig. 4.26. Transferencia tipo 2D sincronizada a nivel de array.

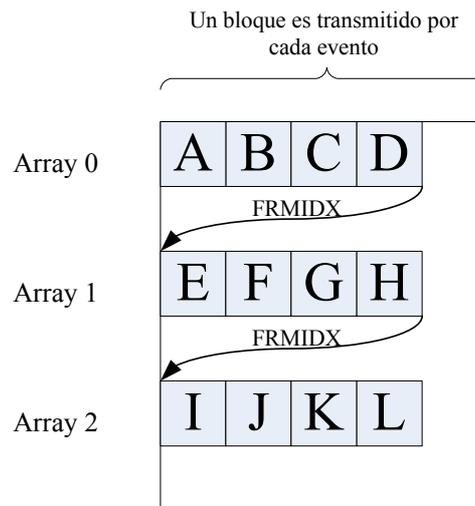


Fig. 4.27. Transferencia tipo 2D sincronizada a nivel de bloque.

Dentro del controlador EDMA existe un conjunto de registros de procesamiento de eventos que tienen como misión fundamental habilitar, o deshabilitar, los eventos que se producen y activarlos, o desactivarlos, de forma manual. Estos registros se muestran en la tabla 4.32. Una vez que un evento se ha activado, éste se desactiva sólo cuando el controlador EDMA realiza la acción oportuna. No obstante, existe también la posibilidad de desactivarlos manualmente usando los registros correspondientes.

Registro	Acrónimo	Descripción
TMS320C621x/C671x		
<i>Event Register</i>	ER	Almacena los 16 posibles eventos que se producen en un DPS. Este registro dispone de 16 bits en su parte baja, denominados EVT0 a EVT15, que indican la aparición de alguno de estos eventos, incluso si están deshabilitados.
<i>Event Enable Register</i>	EER	Permite habilitar o deshabilitar los eventos.
<i>Event Clear Register</i>	ECR	Desactivación manual de los eventos.
<i>Event set register</i>	ESR	Activación manual de los eventos.
TMS320C64x		
<i>Event Register Low</i> <i>Event Register High</i>	ERL ERH	Tienen como misión almacenar los 64 posibles eventos que se producen.
<i>Event Enable Register Low</i> <i>Event Enable Register High</i>	EERL EERH	Permiten habilitar o deshabilitar los eventos.
<i>Event Clear Register Low</i> <i>Event Clear Register High</i>	ECRL ECRH	Desactivación manual de los eventos.
<i>Event set registre Low</i> <i>Event set registre High</i>	ESRL ESRH	Activación manual de los eventos.
<i>Event polarity register low</i> <i>Event polarity register high</i>	EPRL EPRH	Programación del flanco de activación de los eventos.

Tabla 4.32. Registros de estado y de configuración de eventos de la EDMA.

Existen, básicamente, dos métodos para iniciar una operación EDMA:

1. El primero se basa en activar manualmente el bit correspondiente en el registro ESR y, por tanto, no tiene ningún tipo de sincronización.
2. El segundo, el método más habitual, se basa en la activación por los diferentes periféricos del evento correspondiente en el registro ER.

Para que el evento sea atendido por el controlador EDMA se debe activar el correspondiente bit en el registro EER. Los códigos de eventos disponibles en los DSPs TMS320C621x y TMS320C671X se muestran en la tabla 4.33 (los 64 códigos asociados a los eventos en los dispositivos TMS320C64x no se suministran por simplicidad y pueden encontrarse en la documentación técnica proporcionada por el fabricante).

Código de evento	Acrónimo	Descripción
0	DSPINT	Interrupción del Host (HPI) al DSP.
1	TINT0	Interrupción temporizador 0.
2	TINT1	Interrupción temporizador 1.
3	SD_INT	Interrupción del timer EMIF SDRAM.
4	EXT_INT4	Interrupción externa pin 4.
5	EXT_INT5	Interrupción externa pin 5.
6	EXT_INT6	Interrupción externa pin 6.
7	EXT_INT7	Interrupción externa pin 7.
8	EDMA_TCC8	Interrupción fin de EDMA, código '1000'.
9	EDMA_TCC9	Interrupción fin de EDMA, código '1001'.
10	EDMA_TCC10	Interrupción fin de EDMA, código '1010'.
11	EDMA_TCC11	Interrupción fin de EDMA, código '1011'.
12	XEVT0	Transmisión del McBSP0.
13	REVT0	Recepción del McBSP0.
14	XEVT1	Transmisión del McBSP1.
15	REVT1	Recepción del McBSP1.

Tabla 4.33. Códigos de los eventos EDMA en los DSPs TMS320C621x/C671x.

5.2.4 Transferencias QDMA

La familia TMS320C6000 dispone de un tipo de transferencia rápida DMA, denominada QDMA, que realiza el controlador EDMA. Es un tipo de transferencia de información asociada a bloques completos, a diferencia de los tipos de transferencia

comentados para el controlador EDMA, que están orientados al intercambio de datos entre periféricos y memoria sincronizados con eventos *hardware*. Las transferencias QDMA se basan en un movimiento de bloques controlado por *software*, a voluntad del usuario, sin que exista la posibilidad de enlazar las peticiones y que se programan mediante un conjunto de registros con una estructura similar a la de los parámetros asociados a la RAM del controlador EDMA.