第 6 章 DSP 的外设及应用

6.1 C55x 片内外设简介

C55x 的片内外设分为如下几类：

1.时钟与定时器

时钟与定时器包括时钟产生器、通用定时器、实时时钟以及看门狗定时器等。时钟产生器的功能是产生 CPU 的工作时钟，并提供 CLKOUT 时钟输出；通用定时器、实时时钟及看门狗定时器的功能是通过计数器为系统提供定时时钟和年、月、日、时、分、秒等时钟信号，以及监控系统正常运行的看门狗时钟，并能发出相应中断。

2．外部设备连接接口

外部设备连接接口包括主机接口、外部存储器连接接口等。主机接口主要用来为主控 CPU 和 C55x 处理器之间提供一条方便、快捷的并行连接接口，这个接口用来对 DSP 进行控制、程序加载、数据传输等工作；外部存储器接口主要用来同并行存储器连接，这些存储器包括 SDRAM、Flash、SRAM 存储器等，外部存储器接口还可以同外部并行设备进行连接，这些设备包括并行 A/D、D/A 转换器、具有异步并行接口的专用芯片，并可以通过外部存储器接口同 FPGA、CPLD 等连接；

3．信号采集

信号采集类的外设包括采集模拟信号的模/数转换器和提供数字信号输入、输出功能的通用输入/输出接口。模/数转换器为 DSP 提供了多通道模拟/数字转换能力，通用输入、输出接口可以完成数字信号的采集，当其被设置为输出模式时，可以通过这些接口对其他设备进行控制。

4．通信接口

C55x 处理器为用户提供了多种类型的通信接口，包括多通道缓冲串口、I^2C 接口、异步串口、USB 接口以及多媒体卡/SD 卡接口等。多通道缓冲串口可以连接串行存储器、A/D、D/A 转换器，并可以通过该接口实现与其他处理器的高速串行连接，多媒体卡/SD 卡接口可以用来扩展 SD 存储卡等移动存储设备，I^2C 接口、异步串口和 USB 接口为 DSP 通过了各种通用通信接口。

5．其他外设

其他外设包括 DMA 控制器、指令流水线等，这些外设主要用来辅助 CPU 工作，提高 DSP 的工作效率。

本章重点介绍时钟发生器、通用定时器、主机接口、外部存储器接口、DMA 控制器等内容。

6.2 时钟发生器

6.2.1 时钟模式寄存器

C55x 片内的时钟发生器可以从 CLKIN 引脚接收输入的时钟，将其变换为 CPU 及其外设所需要的工作时钟，工作时钟经过分频也能够通过引脚 CLKOUT 输出，供其他器件使用，如图 6-1 所示。时钟发生器内有一个数字锁相环（Digital Phase Lock Loop，DPLL）和一个时钟模式寄存器(CLKMD)。时钟模式寄存器用于控制时钟发生器的工作状态，如表 6-1 所示。

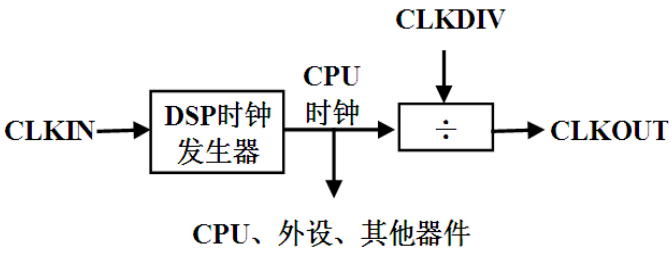

图 6-1 时钟输出

表 6-1 时钟模式寄存器（CLKMD）

位	字段	值	说明
15	Rsvd		保留
14	IAI	0 1	退出 IDLE 状态后，决定 PLL 是否重新锁定 PLL 将使用与进入 IDLE 状态之前相同的设置进行锁定 PLL 将重新锁定过程
13	IOB	0 1	处理失锁 时钟发生器不中断 PLL，PLL 继续输出时钟 时钟发生器自动切换到旁路模式，重新开始跟踪锁定后，又自动切换到锁定模式
12	TEST	0	必须保持为 0
11～7	PLL MULT	0～31	锁定模式下的倍频值，0～31
6～5	PLL-DIV	0～3	锁定模式下的分频值，0～3
4	PLL ENABLE	1 0	PLL 使能 使能，为锁定模式 禁止，为旁路模式
3～2	BYPASS DIV	00 01	旁路下的分频值 一分频 二分频

		10、11	四分频
1	BREAKLN	0 1	错误状态 PLL 失锁 锁定状态或有对 CLKMD 寄存器的写操作
0	LOCK	0 1	锁定状态 PLL 处于旁路模式 PLL 处于锁定模式

6.2.2 工作模式

在时钟模式寄存器中的第 4 位（PLL ENABLE 位）控制 PLL 的两种工作模式，分别是旁路模式和锁定模式。

1.旁路模式（BYPASS）

如果 PLL ENABLE=0，PLL 工作于旁路模式，PLL 对输入时钟信号进行分频，分频值由 BYPASS DIV 确定：

（1）如果 BYPASS DIV=00，输出时钟信号的频率与输入信号的频率相同，即一分频。

（2）如果 BYPASS DIV=01，输出时钟信号的频率是输入信号频率一半，即二分频。

（3）如果 BYPASS DIV=10 或 11，输出时钟信号的频率是输入信号频率 1/4，即四分频。

2.锁定模式（LOCK）

如果 PLL ENABLE=1，PLL 工作于锁定模式，输出的时钟频率由下面公式确定，PLL MULT 和 PLL DIV 说明见表 6-1：

$$\text{输出频率} = \frac{\text{PLL MULT}}{\text{PLL DIV} + 1} \times \text{输入频率}$$

6.2.3 CLKOUT 输出

CPU 时钟也可以通过一个时钟分频器提供 CLKOUT 信号，CLKOUT 的频率由系统寄存器（SYSR）中的 CLKDIV 确定：

（1）如果 CLKDIV=000b，CLKOUT 的频率等于 CPU 时钟的频率。

（2）如果 CLKDIV=001b，CLKOUT 的频率等于 CPU 时钟的频率的 1/2。

（3）如果 CLKDIV=010b，CLKOUT 的频率等于 CPU 时钟的频率的 1/3。

（4）如果 CLKDIV=011b，CLKOUT 的频率等于 CPU 时钟的频率的 1/4。

（5）如果 CLKDIV=100b，CLKOUT 的频率等于 CPU 时钟的频率的 1/5。

（6）如果 CLKDIV=101b，CLKOUT 的频率等于 CPU 时钟的频率的 1/6。

（7）如果 CLKDIV=110b，CLKOUT 的频率等于 CPU 时钟的频率的 1/7。

（8）如果 CLKDIV=111b，CLKOUT 的频率等于 CPU 时钟的频率的 1/8。

6.2.4 使用方法

通过对时钟模式寄存器的操作，可以根据需要设定时钟发生器的工作模式和输出频率，在设置过程中除了工作模式、分频值和倍频值以外，还要注意其他因素对 PLL 的影响。

1.省电（IDLE）

为了节省功耗，可以使时钟发生器处于省电状态，当时钟发生器退出省电状态时，PLL 自动切换到旁路模式，进行跟踪锁定，锁定后返回到锁定模式，时钟模式寄存器中与省电有关的位是 IAI。

2.DSP 复位

在 DSP 复位期间和复位之后，PLL 工作于旁路模式，输出的时钟频率由 CLKMD 引脚上的电平确定。

（1）如果 CLKMD 引脚为低电平，输出频率等于输入频。

（2）如果 CLKMD 引脚为高电平，输出频率等于输入频率的一半。

3.失锁

锁相环对输入时钟跟踪锁定之后，由于其他原因使其输出时钟发生偏移，即失锁。出现失锁现象后，PLL 的动作由时钟模式寄存器中的 IOB 确定。

6.2.5 时钟发生器应用

时钟发生器可以通过调用时钟发生器库函数来使用，首先要在头文件中包含 csl_pll.h 文件，接下来介绍利用库函数配置时钟发生器的方法。

首先要声明 PLL 配置结构，具体声明如下：

```
PLL_Config Config_PLL = {
1, /*iai 休眠后重新锁相 */
1,  /*iob 失锁后进入旁路模式并重新锁相    */
6,  /*pllmult   CLKIN * pllmult = DSP 主时钟*/
0  /*div   CLKOUT= DSP 主时钟/（div+1） */
};
```

之后运行配置函数：

```
PLL_config（&Config_PLL）;
```

也可以通过函数设置 PLL 频率：

```
PLL_setFreq（6, 1）;
```

通过 PLL_setFreq 函数可以复位 PLL 锁相环，并改变倍频和分频数从而得到所需的频率。

6.3 通用定时器

C55x 片内有两个 20 位软件可编程定时器，利用定时器可向 CPU 产生周期性中断或向 DSP 片外的器件提供周期信号。

6.3.1 结构框图

20 位的定时器由两部分组成：一个 4 位的预定标器(PSC)和一个 16 位的主计数器(TIM)，如图 6-2 所示。

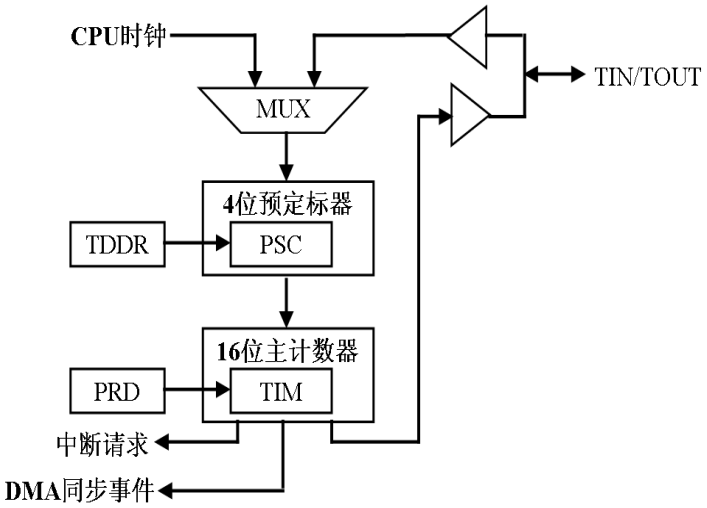

图 6-2 定时器结构图

定时器有两个计数寄存器（PSC，TIM）和两个周期寄存器（TDDR，PRD），在定时器初始化过程中，周期寄存器的内容拷贝到计数寄存器中。

6.3.2 工作原理

定时器的工作时钟可以来自 DSP 内部的 CPU 时钟，也可以来自引脚 TIN/TOUT。定时器控制寄存器（TCR）中的字段 FUNC 可以确定时钟源和 TIN/TOUT 引脚的功能，这样定时器的工作模式包括以下几种：

（1）当 FUNC=00b 时，TIN/TOUT 为高阻态，时钟源是内部时钟（CPU 时钟）。

（2）当 FUNC=01b 时，TIN/TOUT 为定时器输出，时钟源是内部时钟（CPU 时钟）。

（3）当 FUNC=10b 时，TIN/TOUT 为通用输出，时钟源是内部时钟（CPU 时钟）。

（4）当 FUNC=11b 时，TIN/TOUT 为定时器输入，时钟源是外部时钟。

在定时器中，预定标器由输入时钟驱动，PSC 在每个输入时钟周期减 1，当其减到 0 时，TIM 减 1，当 TIM 减到 0，定时器向 CPU 发送一个中断请求（TINT）或向 DMA 控制器发送同步事件。定时器发送中断信号或同步事件信号的频率可用下式计算：

$$TINT\text{频率} = \frac{\text{输入时钟频率}}{(TDDR+1)\times(PRD+1)}$$

通过设置 TCR 中的自动重装控制位 ARB，可使定时器工作于自动重装模式，当 TIM 减到 0，重新将周期寄存器（TDDR，PRD）的内容拷贝到计数寄存器（PSC，TIM）中，继续定时。

每个定时器包括 4 个寄存器，即主计数寄存器 TIM、主周期寄存器 PRD、定时器控制寄存器 TCR、定时器预定标寄存器 PRSC，表 6-2 至表 6-5 是有关这 4 个寄存器的说明。

表 6-2 主计数寄存器 TIM

位	字段	值	说明
15～0	TIM	0000h～FFFFh	主计数寄存器

表 6-3 主周期寄存器 PRD

位	字段	值	说明
15～0	PRD	0000h～FFFFh	当 TIM 必须重新装入时，将 PRD 的内容拷贝到 TIM 中

表 6-4 定时器控制寄存器 TCR

位	字段	值	说明
15	IDLEEN		省电使能位
		0	定时器不能处于省电状态
		1	如果省电状态寄存器中的 PERIS=1，定时器进入省电状态
14	INTEXT		时钟源从内部切换到外部标志位
		0	定时器没有准备好使用外部时钟源
		1	定时器准备使用外部时钟源
13	ERRTIM		定时器错误标志
		0	正常
		1	出错
12～11	FUNC		定时器工作模式选择位
10	TLB		定时器装载位
		0	TIM，PSC 不重新装载
		1	将 PRD、TDDR 分别拷贝到 TIM、PSC 中
9	SOFT		在调试时遇到断点定时器的处理方法
8	FREE		在调试时遇到断点定时器的处理方法
7～6	PWID		定时器输出脉冲的宽度
		00	1 个 CPU 时钟周期
		01	2 个 CPU 时钟周期
		10	4 个 CPU 时钟周期
		11	8 个 CPU 时钟周期
5	ARB		自动重装控制位
4	TSS		定时器停止状态位
		0	启动
		1	停止

3	C/P	0 1	定时器输出时钟/脉冲选择 输出脉冲 输出时钟
2	POLAR	0 1	时钟输出极性位 正极性 负极性
1	DATOUT	0 1	当 TIN/TOUT 作为通用输出引脚，该位控制引脚上的电平 低电平 高电平
0	Reserved	0	保留

表 6-5 定时器预定标寄存器 PRSC

位	字段	值	说明
15～10	Reserved		保留
9～6	PSC	0h～Fh	预定标计数寄存器
5～4	Reserved		保留
3～0	TDDR	0h～Fh	当 PSC 重新装入时，将 TDDR 的内容拷贝到 PSC 中

6.3.3 使用方法

在定时器的工作过程中，要注意以下因素对定时器的影响。

1.初始化定时器

通用定时器的初始化过程如下：

（1）停止计时（TSS=1），定时器装载使能（TLB=1），并将周期寄存器（TIM，PSC）的内容拷贝到计数寄存器（PRD，TDDR）；

（2）将预定标计数周期数写入 TDDR；

（3）将主计数器周期数装入 PRD；

（4）关闭定时器装载（TLB=0），启动计时（TSS=0）。

2.停止/启动定时器

利用时钟控制寄存器中的 TSS 位可以停止或启动定时器。

（1）TSS=1，停止计时；

（2）TSS=0，启动计时。

3.DSP 复位

DSP 复位后定时器的寄存器将按照如下规则复位：

（1）停止定时（TSS=1）；

（2）预定标计数器值为 0；

（3）主计数器值为 FFFFh；

（4）定时器不进行自动重装（ARB=0）；

（5）idle 指令不能使定时器进入省电模式；

（6）仿真时遇到软件断点定时器立即停止工作；

（7）TIN/TOUT 为高阻态，时钟源是内部时钟（FUNC=00b）。

6.3.4 通用定时器的应用

通用定时器可以通过调用通用定时器库函数来使用,如果使用芯片支持库函数对通用定时器进行编程，则必须包含头文件 csl_timer.h。

首先定义通用定时器句柄和配置结构：

```
TIMER_Handle hTimer;
TIMER_Config Config_TIMER = {
0X0310,
/*; 载入 TCR0t:
; IDLE_EN = 0 (不允许空闲状态)
; FUNC = 00b (引脚为高阻态)
; TLB = 0 (TLB 被清除)
; FREE = 1 (遇到断点时时钟不停止)
; PWID = 00b (脉冲延迟一个 CPU 时钟周期)
; ARB = 1 (当 TIM 计数到 0 时重新载入 TIM 和 PSC)
; TSS = 1 (停止计数器)
; C/P = 0 (引脚输出为脉冲模式)
; POLAR = 0 (引脚信号开始为低)
; 其他为 0
*/
0X197,
/*prd = 407*/
0X0007
/* prsc = 7 TDDR=7*/
//;计数器每 3264(408*8) 个时钟周期输出:
};
```

接下来打开句柄：

```
hTimer = TIMER_open(TIMER_DEV0,0);//设置计数器 0
```

调用定时器配置函数对计数器初始化：

```
TIMER_config(hTimer,&Config_TIMER);
```

调用定时器开始函数使定时器开始工作：

```
TIMER_start(hTimer);
```

如果在程序中需要暂时停止定时器计数可以调用定时器停止函数：

```
TIMER_stop(hTimer);
```

当使能定时器中断时，则当定时中断发生时将运行定时中断服务程序：

```
interrupt void Timer0_Isr()
{
……
}
```

6.4 主机接口（EHPI）

主机接口可以让外部的主处理器直接访问 C55x 数字信号处理器内存映射中的部分内存，而无需 DSP 干预。通过主机接口还可以完成 DSP 的程序引导，DSP 向主机发出中断信号要求主机响应中断等功能。

EHPI 的连接方式有两种——非复用方式和复用方式。表 6-6 为 EHPI 接口的信号及其功能说明。

表 6-6 EHPI 接口功能说明

信号名称	类型	功能
HD[15:0]	输入/输出/高阻	主机数据总线 在非复用模式下，只传输数据信号；复用模式下传输数据和地址信号
HA[19:0]	输入	主机地址总线 复用模式下传输主机到 HPI 口的地址信号；复用模式下 HA[1]变成 HCNTL1，HA[2]变为 HAS_，其他引脚没有使用
HBE[1:0]	输入	主机字节选择信号，但在 TMS320VC5510 的 2.0 版本之后不再支持该信号
HCS_	输入	片选信号，低有效
HR/W_	输入	读/写信号
HDS1_，HDS2_	输入	数据选通信号，HPI 接口的数据选通信号是这两个信号的同或结果。选通信号至少应持续 2 个 CPU 周期，HDS1_和 HDS2_信号的连接根据主机选通信号而对应不同的接法。 主机有独立的读写选通信号，并且低有效，则一个接 HDS1_,另一个接 HDS2_； 主机只有一个低有效的选通信号，则这个信号接到 HDS1_、HDS2_之一，而另一个引脚接高电平； 主机有一个高有效地选通信号，则这个信号接 HDS1_或 HDS2_,另一个引脚接低电平
HRDY	输出	EHPI 就绪信号。当这个信号为低时，标志 EHPI 接口忙，主机应延长传输周

		期，信号为高时，表示传输过程已经结束，主机可以继续下一次传输。当 HCS_信号为高时，HRDY 信号总为高
HCNTL0、HCNTL1	输入	EHPI 口控制信号 非复用模式下 HCNTL0 为低时，EHPI 接口将访问数据存储器，为高时访问 EHPI 控制寄存器，HCNTL1 被地址线占用； 复用模式下，HCNTL1 和 HCNTL0 信号用来选择访问的寄存器类型 HCNTL[1:0] 寄存器访问类型 00 HPIC 读或写 01 HPID 读写，并且访问后地址自动增加 1 10 HPIA 读写 11 HPID 读写，访问后地址不增加
HAS_	输入	地址选通信号，该信号只在复用模式下起作用，这个信号使得 HCNTL[1:0]和 HR/W_信号可以在访问结束之前就消失
HMODE	输入	EHPI 模式选择信号，当为高时 EHPI 接口工作在非复用模式下，为低工作在复用模式下
RST_MODE	输入	复位模式信号，但该信号在高版本 C55x 处理器中已经不再支持
HINT_	输出	DSP 到主机中断信号，该信号受状态寄存器 ST3_55 中 HINT 位控制

6.4.1 EHPI 接口的非复用连接方式

非复用模式下，EHPI 口地址和数据分别使用单独的总线，接下来给出 C55x DSP 通过 EHPI 接口采用非复用方式访问另一个 C55x DSP 的信号连接图，如图 6-3 所示。

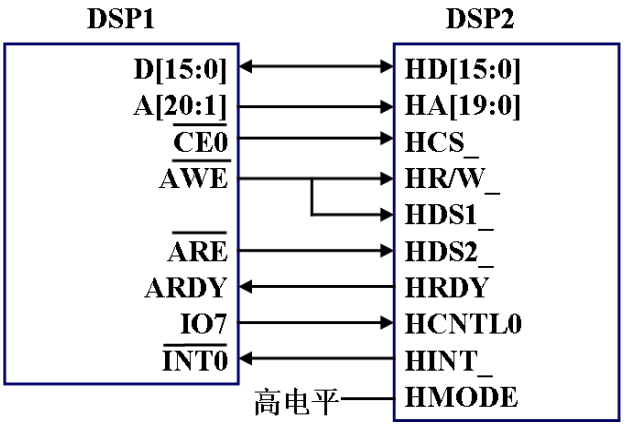

图 6-3 EHPI 口非复用模式连接

DSP1 的通用 IO 信号 IO7 用来选通数据寄存器或者控制寄存器，图中没有标出的 EHPI 口信号不连接即可。非复用连接方式下数据和地址分别使用不同的总线，地址信号不必再通过 EHPI 数据总线传递，访问更加方便、快捷。

6.4.2 EHPI 接口的复用连接方式

EHPI 口如果采用复用连接方式，地址和数据则都将通过数据总线传递，接下来给出 PCI 总线控制器 PCI2040 同 C55x EHPI 口的连接图，如图 6-4 所示。

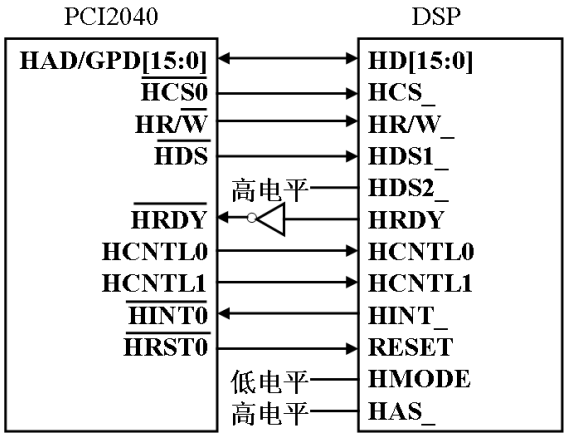

图 6-4 EHPI 口复用模式连接

PCI2040 是为 C54x 和 C6000 系列处理器通过 HPI 接口连接到 PCI 总线专门提供的，但由于 C55x 处理器的主机接口数据总线由 C54x 的 8 位变为 16 位，因此 C55x 是模拟 C6000 的 HPI 接口同 PCI2040 相连接的，由于 C6000 的 HRDY 信号为低有效，而 C55x 的 HRDY 为高有效,因此 C55x 的 HRDY 信号必须通过一个非门连接到 PCI2040 上。PCI2040 没有 HAS 信号，故而 C55x 中的 HAS_信号接到高电平。

6.4.3 EHPI 的寄存器

EHPI 有下列寄存器：数据寄存器（HPID）、地址寄存器（HPIA）和控制寄存器（HPIC），数据寄存器是一个 16 位寄存器，用来存放输入、输出的数据，在非复用方式下，该寄存器只起缓存作用，对主机来说该寄存器是透明的；而复用方式下，主机对 DSP 内存的访问都必须经过数据寄存器，DSP 再根据地址寄存器中的地址访问数据存储器。地址寄存器是一个 16 或 20 位寄存器，该寄存器保存复用方式下读写操作的地址，而地址寄存器也将根据 HCNTL1 和 HCNTL0 的状态决定访问结束后寄存器内的地址是否加 1，地址寄存器在非复用方式下不起作用。

控制寄存器 HPIC 控制数据的传输，主机还可以通过该寄存器向 DSP 发出中断，要求 DSP 响应中断，另外主机通过控制 HPIC 中的 RESET 位，在 DSP 复位引脚为高时可以控制 DSP 的复位或使 DSP 脱离复位状态。HPIC 寄存器具体说明如表 6-7 所示。

表 6-7 HPIC 寄存器

位	字段	值	说明
15～6	Reserved		保留
5	XADD		扩展地址使能位。在复用模式下如果使用 20 位地址，则须通过

		0 1	设置该位决定访问的是 HPIA 的 19～16 位还是 15～0 位 写到 HPIA 的 15～0 位 写到 HPIA 的 19～16 位
4～2	Reserved		保留
1	DSPINT	0 1	主机对 DSP 的中断申请位 清除 DSPINT 向 DSP 发出中断申请
0	RESET	0 1	复位 清除复位 使 DSP 停止进入复位状态

利用 RESET 位，主机可以通过软件使 DSP 进入复位状态，在该状态下，主机可以对 DSP 进行程序加载，加载完成之后清除复位标志，如果 DSP 设置的是 EHPI 引导，在主机清除复位标志后，DSP 接下来将从 10000h 地址开始执行程序。

6.5 外部存储器接口（EMIF）

6.5.1 EMIF 简介

本小节以 TMS320VC5509A 为例介绍 EMIF 接口，EMIF（external memory interface）控制 DSP 与外部存储器的数据传输，接口框图如图 6-5 所示。C55x 的外部存储器接口除了对异步存储器的支持以外，还提供了对同步动态存储器（SDRAM）的支持。

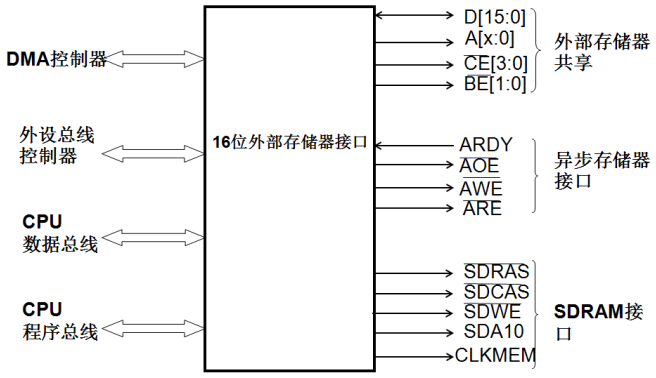

图 6-5 EMIF 结构框图

外部存储器映射分为四个 CE（chip enable）空间，分别为 CE0、CE1、CE2 和 CE3，如图 6-6 所示。CE0 空间位于最低地址，CE3 空间位于最高地址，每个 CE 空间最多可包含 4M 字节。CE3 空间中的一些地址可用于访问 DSP 内部的 ROM，状态寄存器 ST3_55 中的 MPNMC 位控制片内 ROM 的使用，MPNMC 为 1 时，CE3 空间长度为 4MB，MPNMC 为 0 时，CE3 空间后 32KB 空间被片上 ROM 占用。

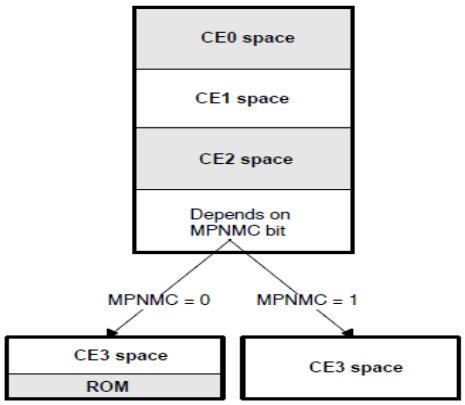

图 6-6 CE 空间

外部存储器接口为每个片选空间都提供了独立的片选控制寄存器 CEn_1、CEn_2、CEn_3 (n=0～3)，通过这些寄存器可以设置寄存器类型，读/写时序以及超时时钟周期数，如表 6-8、表 6-9、表 6-10 所示。同时，DSP 通过片选控制寄存器 CEn_1 中的 MTYPE 段（14-12 位）来设置访问类型，可以访问不同宽度的不同类型的存储器。

表 6-8 片选控制寄存器 1（CEn_1）

位	字段	值	说明
15	Reserved		保留
14～12	MTYPE		存储器类型
		000b	8 位宽异步存储器
		001b	16 位宽异步存储器
		010b	保留
		011b	16 位宽同步动态存储器(SDRAM)
		其他	保留
11～8	RDSETUP		读建立时间
		0,1,2	1 CPU 时钟周期
		3≤n≤15	n CPU 时钟周期
7～2	RDSTROBE		读选通时间
		0,1	1 CPU 时钟周期
		2≤n≤63	n CPU 时钟周期
1～0	RDHOLD		读保持时间
		0≤n≤3	n CPU 时钟周期

表 6-9 片选控制寄存器 2（CEn_2）

位	字段	值	说明
15～14	RDEXHLD		读延长保持时间

		0≤n≤3	n+1 CPU 时钟周期
13～12	WREXHLD	0≤n≤3	写延长保持时间 n+1 CPU 时钟周期
11～8	WRSETUP	0,1,2 3≤n≤15	写建立时间 1 CPU 时钟周期 n CPU 时钟周期
7～2	WRSTROBE	0,1 2≤n≤63	写选通时间 1 CPU 时钟周期 n CPU 时钟周期
1～0	WRHOLD	0≤n≤3	写保持时间 n CPU 时钟周期

表 6-10 片选控制寄存器 3（CEn_3）

位	字段	值	说明
15～8	Reserved		保留
7～0	TIMOUT	0 1≤n≤255	超时字段 超时功能被禁止 当 ARDY 信号为低超过 n 个时钟周期，则发生超时错误

6.5.2 异步存储器接口方案

异步存储器的类型多种多样，它们既包括静态随机存储器、闪存存储器、只读存储器等，又有先入先出存储器、双端口存储器等，这些存储器有着不同的特点，可以根据需要灵活选用。

不同的 DSP 封装格式与异步存储器的连接有所区别，与不同位宽的异步存储器连接也是不同的。BGA 封装格式的 DSP 与 16 位宽异步存储器连接如图 6-7（a）所示，LQFP 封装格式的 DSP 与 16 位宽异步存储器连接如图 6-7（b）所示。BGA 封装格式的 DSP 与 8 位宽异步存储器连接如图 6-8（a）所示，LQFP 封装格式的 DSP 与 8 位宽异步存储器连接如图 6-8（b）所示。EMIF 与异步存储器连接各引脚的含义如表 6-11 所示。

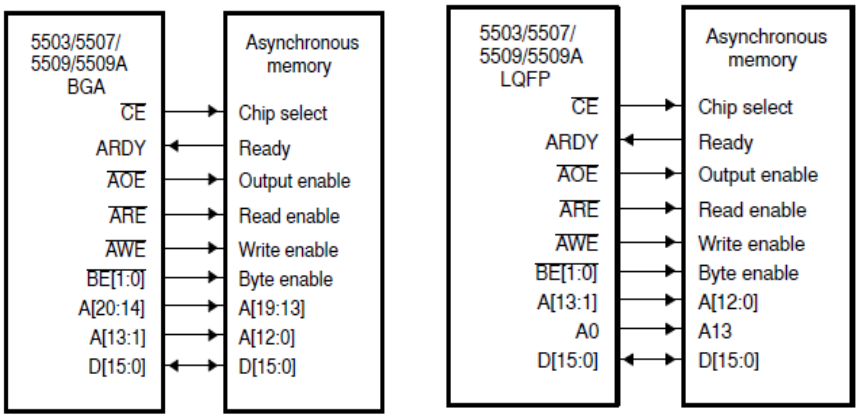

图 6-7（a）与 16 位宽异步存储器连接（BGA 封装）图 6-7（b）与 16 位宽异步存储器连接（LQFP 封装）

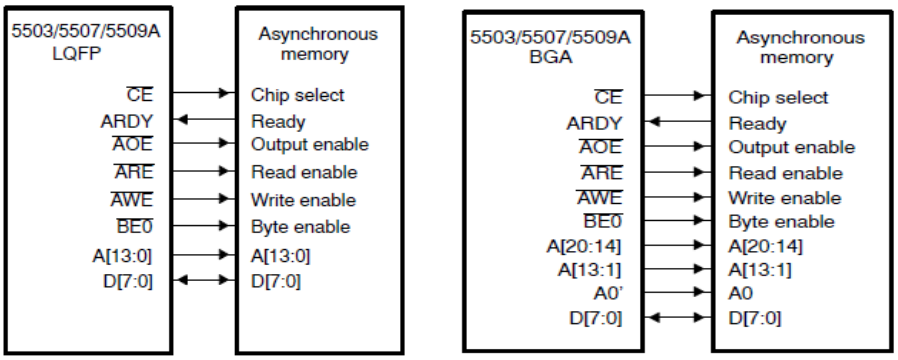

图 6-8（a）与 8 位宽异步存储器连接（BGA 封装）图 6-8（b）与 8 位宽异步存储器连接（LQFP 封装）

表 6-11 EMIF 的引脚含义

EMIF 的引脚	功能
CE	低电平有效的片选信号，用于指定要访问的外部空间。
ARDY	异步访问就绪指示，使 EMIF 可以延缓异步访问速度。
AOE	低电平有效的异步输出使能信号，连接异步存储器的输出使能引脚。
ARE	低电平有效的异步读使能信号。
AWE	低电平有效的异步写使能信号。
BE	低电平有效的字节选择信号，用于指定要访问的字节位置。
A	地址总线。
D	数据总线。

6.5.3 SDRAM 接口方案

C5509 的 EMIF 能支持容量为 64Mbit 或 128Mbit 宽度为 16 位的 SDRAM。SDRAM 的引脚映射和寄存器配置表如表 6-12 所示，SDRAM 接口专用信号包括 SDRAM 行选通信号 SDRAS、列选

通信号 SDCAS 和写使能信号 SDWE，SDA10 信号在 ACTV 命令时作为行地址信号，在读写操作时作为预加电使能信号，在 DCAB 命令下为高，保持模式下为高阻态。

表 6-12 SDRAM 的引脚映射和寄存器配置表

SDRAM	配置位 SDACC SDSIZE SDWID	CE 空间	接口	边界/行地址	列地址
64M 位 4M×16 位	0 0 0	CE0-CE1	SDRAM	BA[1:0] 和 A[11:0]	A[7:0]
		CE2-CE3	EMIF	A[14:12], SDA10, 和 A[10:1]	A[8:1]
128M 位 8M×16 位	0 1 0	CE0-CE3	SDRAM	BA[1:0] 和 A[11:0]	A[8:0]
			EMIF	A[14:12], SDA10, 和 A[10:1]	A[9:1]

其中配置位 SDSIZE、SDWID 位于 SDRAM 控制寄存器 1 中的第 10～9 位，配置位 SDACC 位于 SDRAM 控制寄存器 2 的第 10 位。SDRAM 控制寄存器 1 的功能如表 6-13 所示，SDRAM 控制寄存器 2 的功能如表 6-14 所示。

表 6-13 SDRAM 控制寄存器 1

位	字段	值	说明
15～11	TRC	0～31	从刷新命令 REFR 到 REFR/MRS/ACTV 命令间隔 CLKMEM 周期数
10～9	SDSIZE:SDWID		SDRAM 类型
		00	4M x16 位（64M 位）
		10	8M x16 位（128M 位）
		其他	保留
8	RFEN		刷新使能
		0	禁止刷新
		1	允许刷新
7～4	TRCD	0～15	从 ACTV 命令到 READ/WRITE 命令 CLKMEM 周期数
3～0	TRP	0～15	从 DCAB 命令到 REFR/ACTV/MRS 命令 CLKMEM 周期数

表 6-14 SDRAM 控制寄存器 2

位	字段	值	说明
15～11	Reserved		保留
10	SDACC	0	只能为 0，表示 SDRAM 数据总线宽度为 16 位
9～8	TMRD	0～3	ACTV/DCAB/REFR 延迟 CLKMEM 周期数
7～4	TRAS	0～15	SDRAS 信号有效时持续 CLKMEM 周期数

3～0	TACTV2ACTV	0～15	SDRAS 到 SDRAS 有效延迟 CLKMEM 周期数

C55x 与不同容量的 SDRAM 连接关系如图 6-9 与图 6-10 所示。与 64Mbit（4M×16bit）的 SDRAM 的连接关系图如图 6-9 所示，一片 SDRAM 的容量为 64Mbit，而一个片选空间只有 32Mbit，则需要占用 2 个连续的片选空间。但在连接片选信号时只需要连接第一个 CEn 信号即可，在本例中只需要连接 CE0 信号，而 CE1 信号不需要连接。其它没有用到的 CEn 信号（如 CE2、CE3）可以供其他存储器件使用。但是需要注意的是 CE 空间只能使用前两个空间或者后两个空间，比如可以使用 CE0 和 CE1、或者 CE2 和 CE3，但是不能使用 CE1 和 CE2；同时选中的 2 个 CE 空间对应的片选控制寄存器中的 MTYPE 段需要设置为 011。

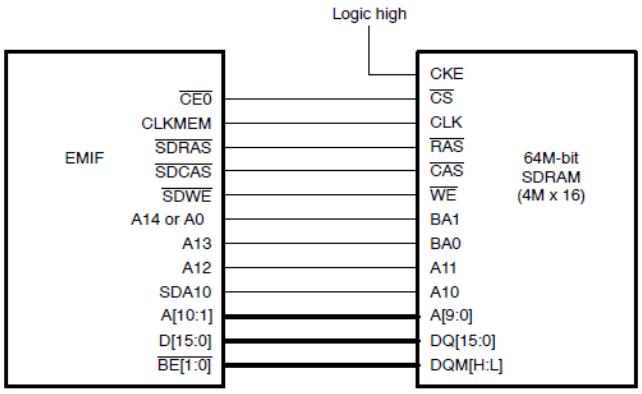

图 6-9 C55x 与 64Mbit 的 SDRAM 连接图

C55x 与 128Mbit（8M×16bit）的 SDRAM 的连接关系图如图 6-10 所示，128Mbit 的 SDRAM 将占用所有片选空间，而片选信号只需连接 CE0 信号即可，但应注意 CE1、CE2、CE3 信号不能被其他的存储器所使用。同时所有片选控制寄存器中的 MTYPE 段需要设置为 011。

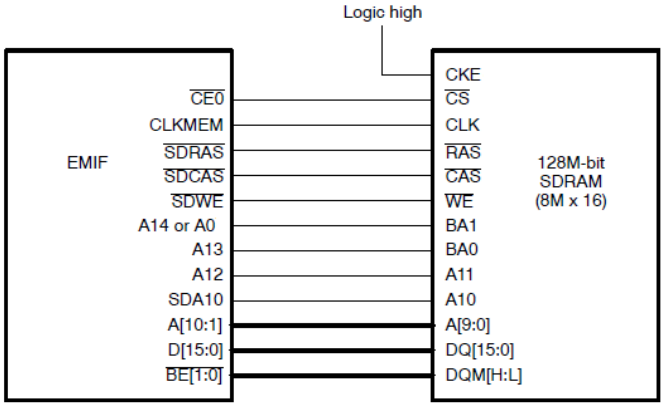

图 6-10 C55x 与 128Mbit 的 SDRAM 连接图

6.6 DMA 控制器

6.6.1 概述

DMA（Direct Memory Access）控制器可以无需 CPU 介入而在内部存储器、外部存储器及片上外设之间传送数据，HPI 接口也使用 DMA 辅助端口传送数据。DMA 具有如下的特点：

（1）DMA 可以独立于 CPU 工作；

（2）有 4 个标准端口（port）与 DARAM、SARAM、外部存储器和外设相连；

（3）一个辅助端口用于 HPI 和存储器之间的数据传送；

（4）具有 6 个通道；

（5）可以设置每个通道的优先级；

（6）每个通道的传输可以由选定事件触发；

（7）当操作完成之后，DMA 控制器可向 CPU 发出中断。

DMA 控制器与 DSP 其他部件的连接框图如图 6-11 所示。

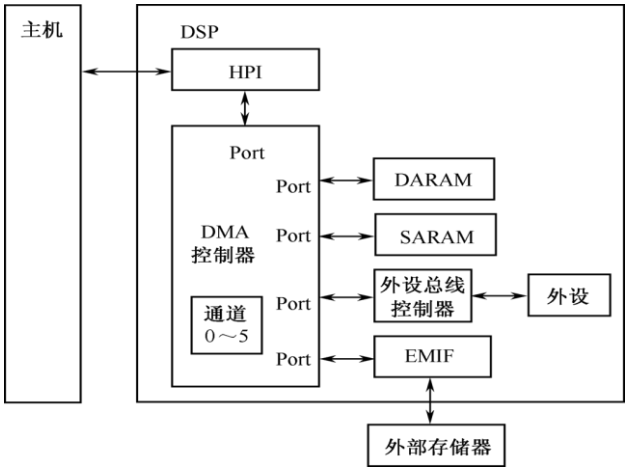

图 6-11 DMA 控制器与其他部件的连接框图

6.6.2 通道传输

DMA 控制器有 6 个通道，用于 4 个标准端口之间的数据传送，每个通道可以从某个端口读取数据，也可以将数据写入某个端口。每个通道有一个 FIFO 缓冲区。如图 6-12 所示，使数据的传输包括两个阶段：端口读取和端口写入。DMA 先从源端口读取数据，并将其放到通道的 FIFO 缓冲区里，然后再从 FIFO 缓冲区取出写入目的端口。

图 6-12DMA 通道传输

DMA 控制器的寄存器有两套：一套是配置寄存器，供 CPU 写入所需的配置值；另一套是工作寄存器，供 DMA 工作时使用。所以，DMA 通道正在进行数据传输时，CPU 可以写入下次传输的配置参数，而不影响正在进行的传输。

6.6.3 HPI 访问配置

HPI 和 DMA 通道的关系由 DMAGCR 寄存器中的 EHPIEXCL 位确定，如图 6-13 所示。

（1）当 EHPIEXCL=0，HPI 和 DMA 通道共享 DARAM、SARAM 和 EMIF。

（2）当 EHPIEXCL=1，HPI 独占 DARAM 和 SARAM，DMA 通道只能访问 EMIF 和外设。

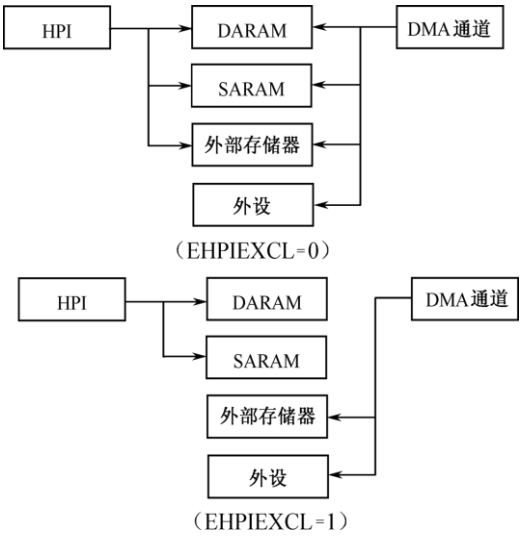

图 6-13 HPI 和 DMA 配置

DMA 通道和 HPI 具有可编程的优先级。通过寄存器 DMACCR 中的 PRIO 位可以设置每个通道的优先级，通过寄存器 DMAGCR 中的 EHPIPRIO 位可以设置 HPI 的优先级。当所有高优先级任务完成或停止时，DMA 控制器为低优先级任务提供服务。在 DSP 复位后，所有通道和 HPI 为低优先级。不管优先级如何设定，端口对通道和 HPI 的检测按照固定的顺序循环：0，1，2，3，4，5，HPI，0，1，2，3，4，5，HPI。

6.6.4 DMA 传输配置

1.数据传输单位

DMA 通道传输的数据单位有 4 种：

（1）字节（Byte）：一个字节是 DMA 通道最小的数据传输单位。

（2）单元（Element）：若干个字节构成的数据传输单位称为一个单元，一个单元可以是 8 位、16 位或 32 位的。一个单元的传输是不能被中断的。

（3）帧（Frame）：若干个单元构成的数据传输单位称为一帧。在一帧的传输过程中不能被中断。

（4）块（Block）：若干个帧构成的数据传输单位称为一个块，每个通道一次或多次传输一个块。在块的传输过程中可以被中断。

对于 DMA 的每个通道，可以定义一块中包括帧的个数，一帧中单元的个数，一个单元中字节的个数。

2.数据打包

DMA 控制器具有数据打包功能，比如选择 8 比特的数据传输而目的端口是 32 位的数据总线，可以将 4 个 8 比特的数据片打包成一个 32 比特的数据包进行传输，这样会提高 DMA 的传输速率。DMA 控制器的数据打包功能通过参数寄存器 DMACSDP 中的 DST(SRC)PACK 字段设定。

（1）当 DST(SRC)PACK=0 时，不打包；

（2）当 DST(SRC)PACK=1 时，对数据打包后再传输。

3．端口

DMA 通道传输的目的端口和源端口由参数寄存器 DMACSDP 中的 DST(SRC)字段来确定。

（1）当 DST(SRC)= xx00 时，目的（源）端口为 SARAM；

（2）当 DST(SRC)= xx01 时，目的（源）端口为 DARAM；

（3）当 DST(SRC)= xx10 时，目的（源）端口为 EMIF；

（4）当 DST(SRC)= xx11 时，目的（源）端口为 Peripheral。

4．数据源和目的地址

DMA 控制器采用字节地址，一个 DMA 通道的数据源起始地址由源起始地址寄存器 DMACSSAL 和 DMACSSAU 指定，其中 DMACSSAL 存放低 16 位地址，DMACSSAU 存放高位地址；目的起始地址由目的起始地址寄存器 DMACDSAL 和 DMACDSAU 指定，其中 DMACDSAL 存放低 16 位地址，DMACDSAU 存放高位地址。

DMA 通道在数据传输过程中的地址修改方式由 DMACCR 寄存器中的 DST(SRC) AMODE 字段确定。

（1）当 DST(SRC)AMODE=00 时，目的（源）地址为固定地址，用于单元的传输；

（2）当 DST(SRC)AMODE=01 时，目的（源）地址在每个单元传输完后自动增加。根据数据的位数是 8 位、16 位还是 32 位，地址分别增加 1、2 或 4。

（3）当 DST(SRC)AMODE=10 时，目的（源）地址在每个单元传输完后自动增加一个索引值，索引值由单元索引寄存器 DMACEI/DMACSEI 确定。

（4）当 DST(SRC)AMODE=11 时，目的（源）地址在每个单元传输完后按单元索引和帧索引自动增加，索引值由单元索引寄存器 DMACEI/DMACSEI 和帧索引寄存器 DMACFI/DMACSFI 确定，又称为双索引。

6.6.5 DMA 控制器的寄存器

DMA 控制器的寄存器如表 6-15 所示，其中有三个全局控制寄存器 DMAGCR、DMAGSCR 和 DMAGTCR 对所有的通道进行控制，此外还有通道配置寄存器用于对每个通道进行控制。

表 6-15 DMA 控制器的寄存器

寄存器	数量	说明

DMAGCR	只有 1 个	全局控制寄存器
DMAGSCR*	只有 1 个	全局软件兼容寄存器，用于控制 DMA 获得目的单元、目的帧索引的方式
DMAGTCR*	只有 1 个	全局超时控制寄存器，用于使能或禁止 SARAM 和 DARAM 端口的超时计数器
DMACCR	每通道 1 个	通道控制寄存器，用于配置优先级等
DMACICR	每通道 1 个	通道中断寄存器，用于中断使能
DMACSR	每通道 1 个	状态寄存器
DMACSDP	每通道 1 个	源和目的参数寄存器，用于配置数据块参数
DMACSSAL	每通道 1 个	源起始地址寄存器（低地址）
DMACSSAU	每通道 1 个	源起始地址寄存器（高地址）
DMACDSAL	每通道 1 个	目的起始地址寄存器（低地址）
DMACDSAU	每通道 1 个	目的起始地址寄存器（高地址）
DMACEN	每通道 1 个	单元数量寄存器
DMACFN	每通道 1 个	帧数量寄存器
DMACEI/DMACSEI	每通道 1 个	单元索引寄存器/源单元索引寄存器
DMACFI/DMACSFI	每通道 1 个	帧索引寄存器/源帧索引寄存器
DMACDEI*	每通道 1 个	目的单元索引寄存器
DMACDFI*	每通道 1 个	目的帧索引寄存器
DMACSAC*	每通道 1 个	源地址计数寄存器
DMACDAC*	每通道 1 个	目的地址计数寄存器

其中*标注的寄存器只在 TMS320VC5503/5507/5509A/C5510 中使用,在 TMS320VC5509 中不使用。

1.DMA 全局控制寄存器（DMAGCR）

DMA 全局控制寄存器（DMAGCR）的参数及说明如表 6-16 所示。

表 6-16 DMA 全局控制寄存器（DMAGCR）

位	字段	值	说明
15～4	Reserved		保留
3	Reserved	1	保留（通常写入 1）
2	FREE		遇到断点时的处理位
		0	停止 DMA 传输
		1	继续 DMA 传输
1	EHPIEXCL		HPI 的配置位

		0 1	与通道共享 独占内部 RAM
0	EHPIPRIO	 0 1	HPI 优先级位 低优先级 高优先级

2.DMA 通道控制寄存器（DMACCR）

DMA 通道控制寄存器（DMACCR）的参数及说明如表 6-17 所示。

表 6-17 DMA 通道控制寄存器（DMACCR）

位	字段	值	说明
15～14	DSTAMODE	00～11	目的地址修改模式
13～12	SRCAMODE	00～11	源地址修改模式
11	ENDPROG	 0 1	通道传输配置位 在传输过程中可以对配置寄存器进行编程 编程结束
10	Reserved	0	保留，该位一定为 0
9	REPEAT	 0 1	多次传输配置时的重复条件位 在本次传输结束后，只有当 ENDPROG=1，才装入新的配置值，开始下次传输 在本次传输结束后，立即装入新的配置值，开始下次传输
8	AUTOINIT	 0 1	多次传输配置时的自动初始位 自动初始禁止 自动初始使能
7	EN	 0 1	通道使能位 禁止 使能
6	PRIO	 0 1	通道优先级位 低优先级 高优先级
5	FS	 0 1	帧/单元同步位 单元同步 帧同步
4～0	SYNC		同步事件

3.DMA 状态寄存器（DMACSR）

DMA 状态寄存器（DMACSR）的参数及说明如表 6-18 所示。

表 6-18 DMA 状态寄存器（DMACSR）

位	字段	值	说明
15～7	Reserved		保留
6	SYNC		同步事件状态
		0	DMA 控制器已完成之前服务请求
		1	同步事件已发生
5	BLOCK		整块状态位
		0	整块事件未发生，或者块已经被清零
		1	整块已经被传输
4	LAST		上一帧状态位
		0	上一帧事件未发生，或者上一帧已经被清零
		1	上一帧已经被传输
3	FRAME		整帧状态位
		0	整帧事件未发生，或者帧已经被清零
		1	整帧已经被传输
2	HALF		半帧状态位
		0	半帧事件未发生，或者半帧已经被清零
		1	帧的前一半已经被传输
1	DROP		同步事件丢弃状态位
		0	同步事件丢弃未发生，或者丢弃已经被清零
		1	同步事件丢弃已经发生
0	TIMEOUT		超时状态位
		0	超时未发生，或者超时已经被清零
		1	超时已经发生

4.源和目的参数寄存器（DMACSDP）

源和目的参数寄存器（DMACSDP）的参数及说明如表 6-19 所示。

表 6-19 源和目的参数寄存器（DMACSDP）

位	字段	值	说明
15～14	DSTBEN		目的端口突发使能位
		00、01	目的端口突发禁止
		10	目的端口突发使能
		11	保留

13	DSTPACK		目的端口打包使能位
		0	禁止
		1	使能
12～9	DST		目的端口类型位
		0000	目的端口为 SARAM
		0001	目的端口为 SARAM
		0010	目的端口为 EMIF
		0011	目的端口为 Peripherals
		其他	保留
8～7	SRCBEN		源端口突发使能位
		00、01	禁止
		10	使能
		11	保留
6	SRCPACK		源端口打包使能位
		0	禁止
		1	使能
5～2	SRC		源端口类型
		0000	源端口为 SARAM
		0001	源端口为 SARAM
		0010	源端口为 EMIF
		0011	源端口为 Peripherals
		其他	保留
1～0	DATATYPE		数据传输单位
		00	8 位
		01	16 位
		10	32 位
		11	保留

5.起始地址寄存器

起始地址寄存器包括源起始地址寄存器（DMACSSAL 和 DMACSSAU）、目的起始地址寄存器（DMACDSAL 和 DMACDSAU），参数及说明如表 6-20 所示。

表 6-20 起始地址寄存器

寄存器	位	字段	值	说明
DMACSSAL	15～0	SSAL	0000h～FFFFh	源起始地址低 16 位
DMACSSAU	15～0	SSAU	0000h～00FFh 0100h～FFFFh	源起始地址高位 保留
DMACDSAL	15～0	DSAL	0000h～FFFFh	目的起始地址低 16 位
DMACDSAU	15～0	DSAU	0000h～00FFh 0100h～FFFFh	目的起始地址高位 保留

6.单元和帧数量寄存器

单元和帧数量寄存器包括单元数量寄存器（DMACEN）、帧数量寄存器（DMACFN），参数及说明如表 6-21 所示。

表 6-21 单元和帧数量寄存器

寄存器	位	字段	值	说明
DMACEN	15～0	ELEMENTNUM	0000h 0001h～FFFFh	保留 每帧的单元数量（1～65535）
DMACFN	15～0	FRAMENUM	0000h 0001h～FFFFh	保留 每块的帧数量（1～65535）

7.单元和帧索引寄存器

单元和帧索引寄存器包括单元索引寄存器（DMACEI/DMACSEI,DMACDEI）、帧索引寄存器（DMACFI/DMACSFI, DMACDFI），参数及说明如表 6-22 所示。

表 6-22 单元和帧索引寄存器

寄存器	位	字段	值	说明
DMACEI/DMACSEI	15～0	ELEMENTNDX	-32768～32767	源单元索引值
DMACDEI	15～0	ELEMENTNDX	-32768～32767	目的单元索引值
DMACFI/DMACSFI	15～0	FRAMENDX	-32768～32767	源帧索引值
DMACDFI	15～0	FRAMENDX	-32768～32767	目的帧索引值

6.6.6 DMA 控制器的应用

DMA 控制器可以通过调用 DMA 控制器库函数来使用，首先要在头文件中包含 csl_dma.h 文件，DMA 配置结构名为 DMA_Config。

1.声明配置结构之后，需要调用 DMA_open 函数，初始化 DMA 句柄：

```
DMA_Handle myhDMA;
myhDMA = DMA_open（DMA_CHA0, 0）;/* 打开 DMA 通道 0 */
```

2.接下来调用 DMA_config 函数对 DMA 进行配置：

```
myconfig.dmacssal =
（DMA_AdrPtr）(((Uint16)（myconfig.dmacssal）<<1）&0xFFFF）;
myconfig.dmacdsal =
（DMA_AdrPtr）(((Uint16)（myconfig.dmacdsal）<<1）&0xFFFF）;
myconfig.dmacssau - （((Uint32） &src） >> 15） & 0xFFFF;
myconfig.dmacdsau - （((Uint32） &dst） >> 15） & 0xFFFF;
DMA_config（myhDMA, &myConfig）; /* 配置通道 */
```

3.配置完成之后，调用 DMA_start（）函数开始 DMA 传送：

```
DMA_start（myhDMA）; /* 开始传送 */
```

4.等待 DMA 状态寄存器的帧状态位指示传输结束：

```
while（!DMA_FGETH（myhDMA, DMACSR, FRAME））
{
}
```

5.DMA 传输结束后关闭句柄：

```
DMA_close（myhDma）; /* 关闭通道 */
```

习题

1. C55x 的片内外设分为哪几类？

2. C55x 的时钟发生器的工作模式有哪几种？如何计算输出频率？

3. C55x 的定时器结构图是什么？如何计算定时器发送中断信号或同步事件信号的频率？

4.举例并画图说明 C55x DSP 通过 EHPI 接口采用非复用方式访问另一个 C55x DSP 的信号连接图。

5. 说明外部存储器映射分为哪几个空间？若想访问 DSP 内部的 ROM 应该如何设置？

6. 简述 DMA 的含义及其特点。