



C66x DSP(Digital Signal Processor)是最新一代定点和浮点 DSP,由 4 个乘法器组成,以实施单精度浮点乘法运算。C66x DSP 内核可同时运行多达 8 项浮点乘法运算,加之高达 1.4GHz 的时钟频率,使其具有很高的浮点处理性能。将多个 C66x DSP 内核与其他内核融合,即可创建出具有出众性能的多核片上系统(System-on-Chip, SoC)器件。

TMS320C6678(以下简写成 C6678)处理器具有 8 个 TMS320C66x 内核,内核工作主频为 1.4GHz 时,理论上具有 179.2GFLOP(22.4GFLOP $\times$ 8)和 358.4GMAC(44.8GMAC $\times$ 8)的处理性能。每个处理器内部有多级存储器: C66x 内核中有 L1P、L1D、L2SRAM;多核共享的有 MSM SRAM(Multicore Shared Memory SRAM)。处理器具有多核导航器、网络协处理器、数据包加速器、信号量、PLL 等多核共享的一些外部资源,同时提供如 SRIO(Serial Rapid IO)、PCIE(PCI Express)、EMIF(External Memory Interface)等多种外部接口。

66AK2Hx 处理器最多可包含 4 个 ARM Cortex-A15、8 个 TMS320C66x 高性能 DSP。66AK2H14/12/06 提供最高 5.6GHz(1.4GHz $\times$ 4) ARM 和 9.6GHz(1.2GHz $\times$ 8) DSP 处理性能。C66x DSP 内核中 L2 SRAM 容量不同,66AK2Hx 处理器 L2 容量为 1024KB, C6678 处理器 L2 容量为 512KB。

本章主要介绍基于 C66x DSP 内核的 DSP 处理器。首先概要性地介绍了 C6678 和 66AK2Hx 处理器,随后介绍了 C66x 处理器内核,然后以 C6678 处理器为例详细介绍了锁相环、外围设备、定时器、信号量、多核导航器等相关内容,最后给出了一些设计建议。

## 1.1 C6678 处理器

本节主要介绍 C6678 处理器,与 66AK 系列不同的是,其处理内核全部由 C66x 内核组成。

### 1.1.1 C6678 概览

TI 推出的 Keystone 架构的多核 DSP,片内集成多个处理内核。采用多核并行处理设计可以大大提高单个处理器的综合处理性能。

C6678 处理器平台具有 8 个 TMS320C66x 内核,每个内核内都有 L1P、L1D 和 L2 SRAM 存储器,多核共享的存储器为 MSM SRAM。

处理器具有多核导航器、网络协处理器、数据包加速器和信号量等多核共享的一些外部资源。多核导航器具有 8192 个多用途硬件队列,由队列管理器负责管理。

在内核主频为 1GHz 的情况下,L1 数据缓存和 L1 程序缓存通信带宽为 32GB/s,L2 缓存通信带宽为 16GB/s,多核共享存储器通信带宽为 64GB/s。DDR3 在主频 1333MHz 的情况下带宽为 10.664GB/s。各级 RAM 的容量分别是:L1 数据缓存和 L1 程序缓存为 32KB,L2 缓存为 512KB,多核共享存储器为 4MB,DDR3 最大支持 8GB。

### 1.1.2 外围设备

器件支持 SRIO、PCIe Gen2、HyperLink、千兆以太网(Gigabit Ethernet, GbE)等多种接口。SRIO 接口支持 4 通道 SRIO 2.1,每通道支持 1.24/2.5/3.125/5G 波特率传输。PCIe Gen2 单个接口支持 1 通道或 2 通道,每通道最高可支持 5G 波特率。HyperLink 接口支持与其他 Keystone 架构连接,最高支持 50G 波特率。Gigabit 网络(GbE)交换子系统,支持两个 SGMII 接口、支持 10/100/1000Mb/s 操作。

64 位 DDR3 接口支持 8GB 访问空间。16 位 EMIF 接口最大支持 256MB NAND FLASH 和 16MB NOR FLASH,最大可支持 1MB 异步 SRAM。C6678 处理器具有 2 个 TSIP、1 个 UART、1 个 I<sup>2</sup>C(Inter-Integrated Circuit)、16 个 GPIO(General-Purpose Input/Output)、1 个 SPI(Serial Peripheral Interconnect)接口、1 个 Semaphore 模块、16 个 64b 定时器、3 个片上 PLL。

C6678 器件架构如图 1.1 所示。

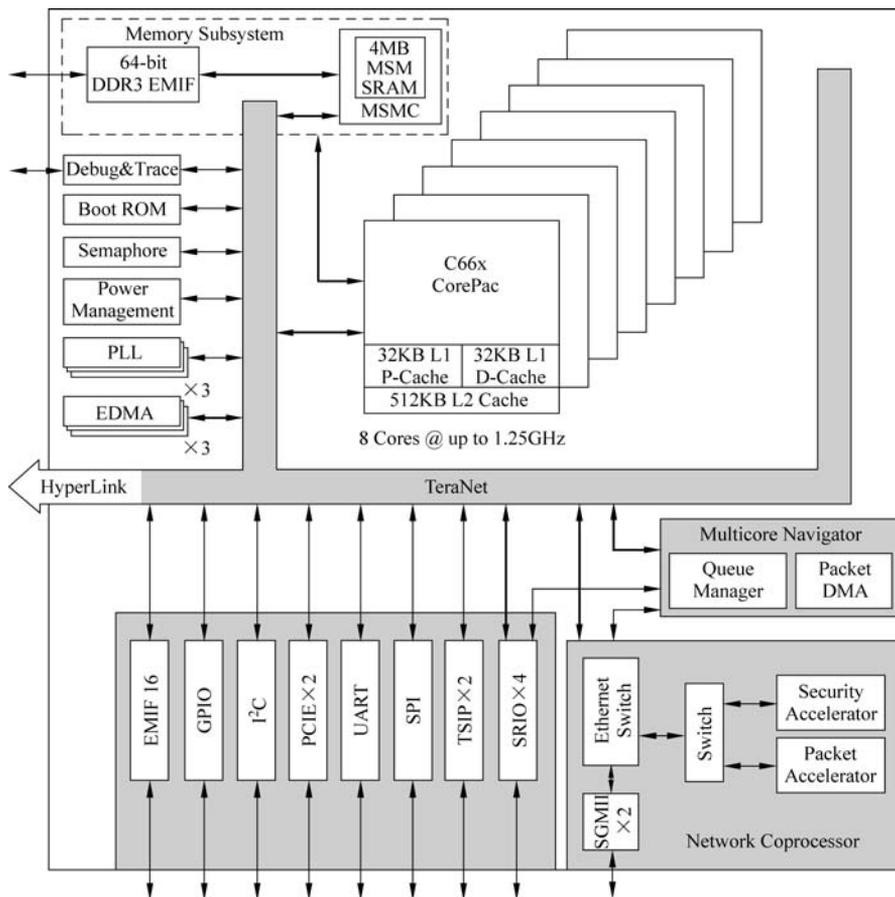


图 1.1 C6678 处理器架构

## 1.2 66AK 处理器

66AK2Hx 处理器平台基于 KeyStone II 架构,最多可包含 4 个 ARM Cortex-A15、8 个 C66x 高性能 DSP。与 C6678 不同的是,66AK 是具有 ARM 和 C66x 内核的 SoC 器件。

66AK2H14/12/06 提供最高 5.6GHz(1.4GHz×4)ARM 和 9.6GHz (1.2GHz×8)DSP 处理性能,并具有安全加速器、包加速器和网络交换功能,比多个芯片的解决方案更省电。

C66x 核载处理器内包含定点和浮点计算能力,运算能力是 38.4GMACS/核和 19.2GFLOPS/核(@ 1.2GHz 主频)。C66x 软件 100% 向下兼容 C64x+ 器件,包含 90 个新指令,其中 FPi (Floating Point instruction) 用于提升浮点运算能力、VPi (Vector math oriented Processing instruction) 用于提升矢量运算能力。

66AK 是基于 C66x 多核 DSP 和 ARM 多核处理器架构的异构多核的 SoC 器件。66AK 器件架构如图 1.2 所示。

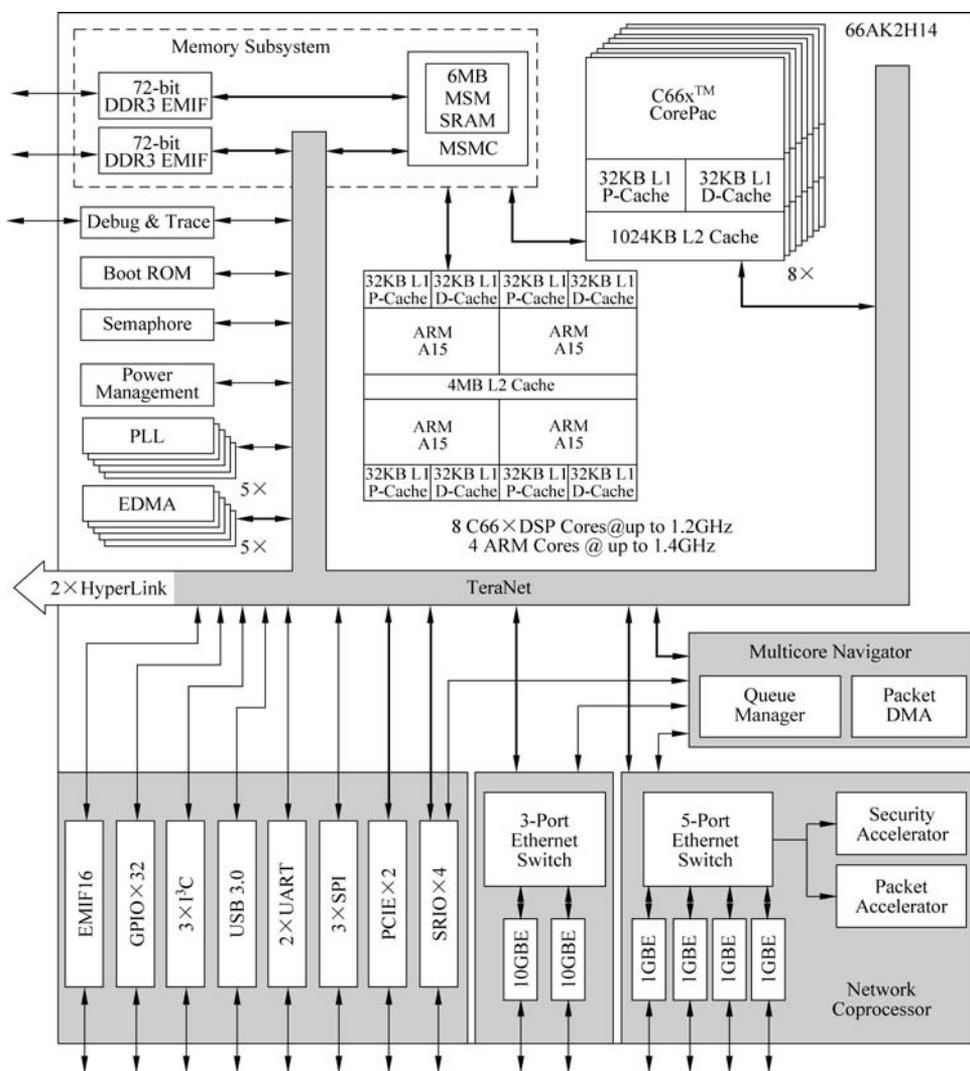


图 1.2 66AK 处理器架构

## 1.3 66AK2H14/12/06 和 C6678 各项功能对比

66AK2H14/12/06 和 C6678 各项功能的对比如表 1.1 所示。

表 1.1 C6678 和 66AK 对比

硬件配置		66AK2H14	66AK2H12	66AK2H06	C6678
核数	C66x DSP	8 个		4 个	8 个
	ARM Cortex-A15	4 个		2 个	0 个
外围设备数	10-GbE	2 个	—	—	—
	DDR3 存储控制器	2 个(72 位总线宽度)			1 个(64 位总线宽度)
	16-bit 异步 EMIF	1 个			1 个
	EDMA3	5 个(64 独立通道)			1 个(16 独立通道[内核时钟/2])
					2 个(64 独立通道[内核时钟/3])
	SRIO 1×/2×/4×	1 个			1 个
	HyperLink (4lanes)	2 个			1 个
	I <sup>2</sup> C	3 个			1 个
	SPI	3 个			1 个
	Tsip	—			2 个
	PCIE(2lanes)	1 个			1 个
	USB 3.0	1 个			—
	UART	2 个			1 个
	10/100/1000 Ethernet	4 个			2 个
	管理数据 I/O(MDIO)	1 个			1 个
	64 位定时器(可配置)	20 个 64 位或 40 个 32 位		14 个 64 位 或 28 个 32 位	16 个 64 位或 32 个 32 位
	GPIO	32 个			16 个
片上存储器组织	L1P 存储控制器(C66x)	32KB 每核			32KB 每核
	L1D 存储控制器(C66x)	32KB 每核			32KB 每核
	L2 缓存(C66x)	1MB 每核			512KB 每核
	L3 ROM (C66x)	128KB			128KB
	L1P (ARM Cortex-A15)	32KB 每核			—
	L1D (ARM Cortex-A15)	32KB 每核			—
	L2 缓存 (ARM Cortex-A15)	4096KB			—
	L3 ROM (ARM Cortex-A15)	256KB			—
	MSMC	6MB			4MB
频率	C66x	最高可达 1.2GHz			最高可达 1.4GHz
	ARM Cortex-A15	最高可达 1.4GHz			

66AK 系列在处理器种类、个数及性能上都有提升。66AK 的工艺是  $0.028\mu\text{m}$ , C6678 的工艺是  $0.040\mu\text{m}$ 。

两个处理器的加速器都是一样多的,每个处理器都有一个包加速器和一个安全加速器。

## 1.4 C66x 处理器内核

C66x 内核是 C6678 处理器的核心,用于完成高性能处理任务。

C66x 内核由以下组件组成: C66x DSP、一级程序存储器控制器(L1P)、一级数据存储器控制器(L1D)、二级存储器控制(L2)、内部 DMA(Internal Direct Memory Access, IDMA)、外部存储控制器(External Memory Controller, EMC)、扩展存储控制器(Extended Memory Controller, XMC)、带宽管理(BandWidth Management, BWM)、中断控制器和休眠控制器(Power-Down Controller, PDC)组成。

一级数据存储器控制器(L1D)、二级存储器控制(L2)、外部存储控制器(EMC)、扩展存储控制器(XMC)、带宽管理(BWM)在将第 4 章 C66x 存储器组织中介绍。内部 DMA (IDMA)在将第 6 章 DMA 传输中介绍。

C66x DSP 是最新一代定点和浮点 DSP。C66x DSP 通过提高 C674X 指令组结构提升性能。C66x DSP 内核如图 1.3 所示。

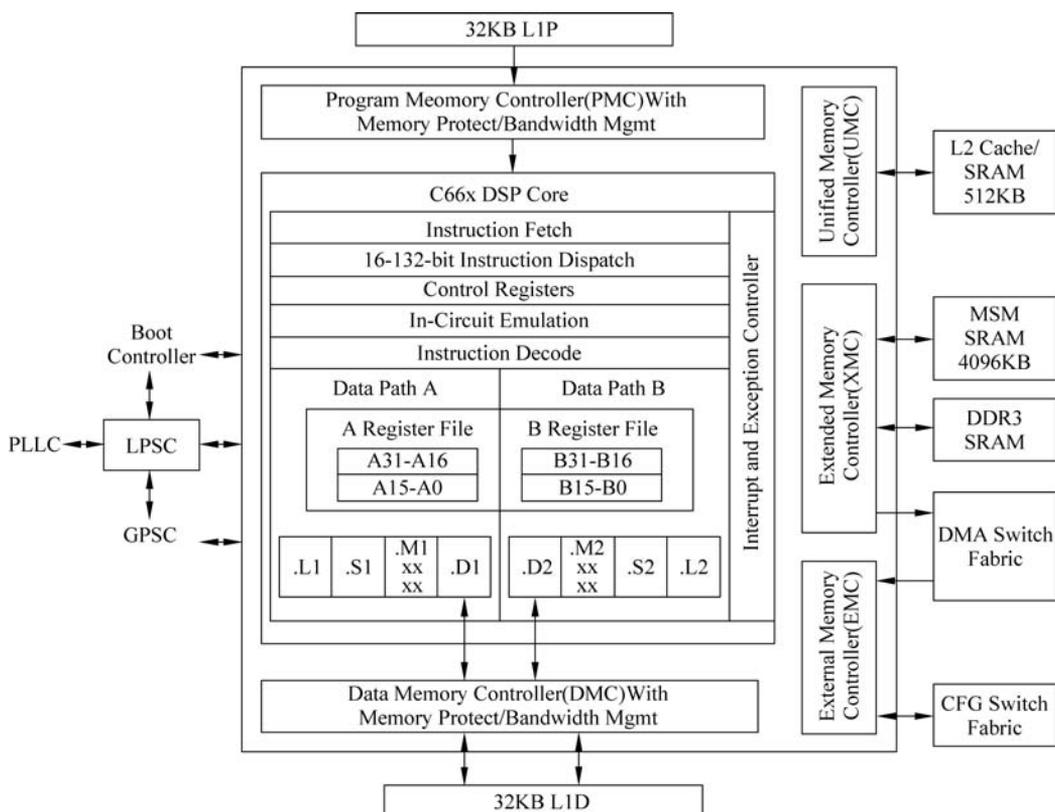


图 1.3 C66x 处理器内核

每个 C6678 内核具有两套处理单元和两套寄存器组，结构如图 1.4 所示。8 个功能单元(.M1、.L1、.D1、S1、M2、L2、D2、S2)都具备每个时钟周期执行一条指令的能力。M 功能单元执行所有乘法运算；S 和 L 单元执行一组通用的算术、逻辑和分支函数；D 单元主要完成从存储器加载(Load)数据到寄存器堆(Register File)，并从寄存器堆保存(Store)结果到存储器。通过多套处理单元并行及各个处理模块流水并行可以大大提高处理性能。

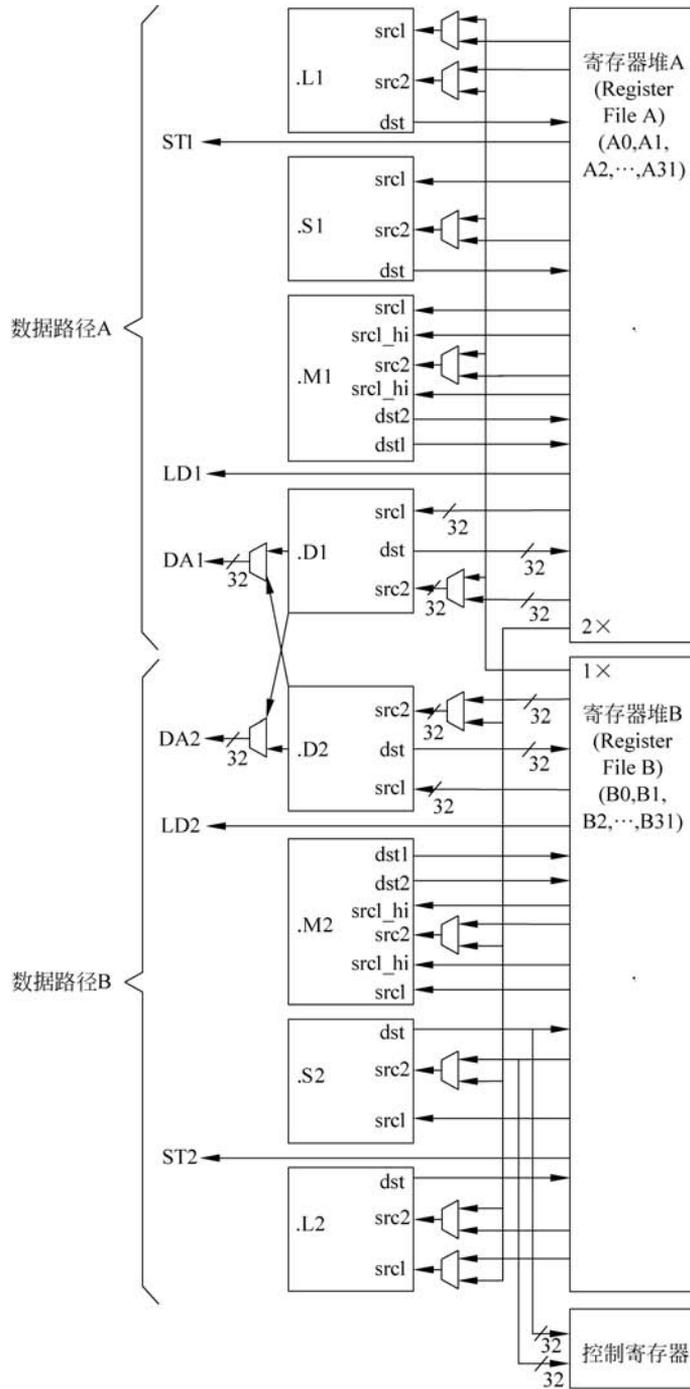


图 1.4 C66x CPU 数据路径

如图 1.5 所示的 TI 最新 C66x 内核,具有同 C64x+内核相同的基本 A & B 结构。值得注意的是: .M 单元的 16 位乘法器已增至每个功能单元 16 个,从而实现内核原始计算能力提升 4 倍。C66x DSP 实现的突破性创新使得由 4 个乘法器组成的各群集可协同工作,以实施单精度浮点乘法运算。C66x DSP 内核可同时运行多达 8 项浮点乘法运算,加之高达 1.4GHz 的时钟频率,使其具有很高的浮点处理性能。将多个 C66x DSP 内核进行完美整合,即可创建出具有出众性能的多核片上系统设备。

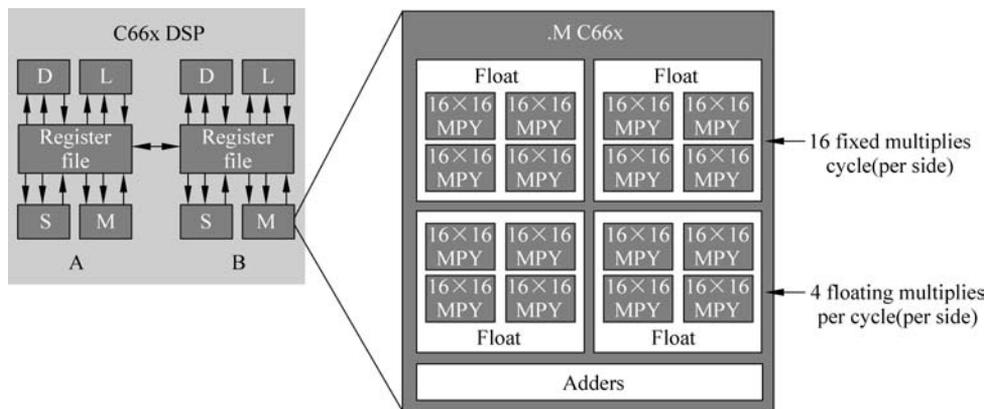


图 1.5 C66x 乘法单元

尽管与浮点处理相比,DSP 定点处理更快,但却不得不为特定算法在开发时间上付出代价。为使定点和浮点组件都能同时实现最佳性能,该款 C66x DSP 内核支持定点与浮点运算指令。

浮点指令包括:

- (1) 单精度复数乘法。
- (2) 矢量乘法。
- (3) 单精度矢量加减法。
- (4) 单精度浮点-整数之间的矢量变换。
- (5) 支持双精度浮点算术运算(加、减、乘、除及与整数间的转换)。

最新定点指令可实现最佳的矢量信号处理,其中包括:

- (1) 复数矢量和矩阵乘法,诸如针对矢量的 DCMPY 以及针对矩阵乘法的 CMATMPYR1。
- (2) 实矢量乘法。
- (3) 增强型点积计算。
- (4) 矢量加减法。
- (5) 矢量位移。
- (6) 矢量比较。
- (7) 矢量打包与拆包。

## 1.5 电源休眠控制器

电源休眠控制器(Power-Down Controller,PDC)可由软件驱动,对所有内核组件进行休眠管理。DSP 可以休眠全部或部分 C66x 内核。本节介绍 C66x 内核电源休眠管理及其特征。

### 1.5.1 C66x 内核电源休眠管理介绍

C66x 内核支持关掉 C66x 内核的部分功能模块或休眠整个内核。设计师可以结合这些特征设计系统,以降低系统电源需求。

表 1.2 列出了 C66x 内核可用的电源休眠特征以及如何/何时应用的概要描述。

表 1.2 C66x 内核电源休眠特征

电源休眠特征	如何/何时应用
L1P 存储器	当 SPLOOP 指令执行时
L2 存储器	保留直到访问(Retention Until Access,RTA),存储器提供基于页的动态唤醒
Cache 控制硬件	当 Cache 被禁用时
DSP	在发出 IDLE 指令时
整个 C66x 内核	通过 PDC 和 IDLE 使能

### 1.5.2 电源休眠管理特征

#### 1. L1P 存储器

当内核从 SPLOOP 缓冲执行指令时,L1P 存储器动态地休眠。该特征是动态使能的,并对用户透明。在完成了 SPLOOP 指令后,当 DSP 重新从 L1P 存储器取数时,L1P 存储器被唤醒。换句话说,当 L1P 没有被访问时,L1P 休眠。

**注意:** 当整个内核电源休眠时,L1P 也被休眠。

#### 2. L2 存储器

C66x 内核不支持用户控制的 L2 动态电源休眠。对于 Keystone 器件,L2 是保留直到访问的存储器类型,只有被访问时才唤醒对应的一个块,访问结束后又把那个块重新置回低泄漏模式。L2 存储器以基于页的唤醒方式进行动态休眠,其自身自动处理休眠管理功能。

**注意:** 当整个内核电源休眠时,L2 存储器也被休眠。

#### 3. Cache 电源休眠模式

当 L1D、L1P 或 L2 Cache 没有使能时,它们保持休眠模式。

**注意:** 当整个内核电源休眠时,三个 Cache 控制器被休眠。

#### 4. DSP 电源休眠

DSP 可以通过执行一个 IDLE 指令进行休眠,也可以被中断唤醒。

#### 5. C66x 内核电源休眠

C66x 内核电源休眠通常被称为静态休眠,这种模式通常需要较长的时间;而用在较短时间的场合,称为动态休眠。

C66x 内核电源休眠完全通过软件控制,通过对 C66x 内核电源休眠指令控制寄存器(PDCCMD)中的(MEGPD)位编程设置,用户可以实施控制。在 C66x 内核电源休眠中,需要按照以下顺序进行:

- (1) 设置电源休眠指令控制寄存器(PDCCMD)的 MEGPD 位为 1,使能电源休眠。
- (2) 使能用户想唤醒的 C66x 内核的 DSP 中断,禁止所有其他中断。
- (3) 执行 IDLE 指令。

内核电源休眠时,C66x 内核保持休眠状态,直到被步骤(2)使能的中断唤醒。

**注意:**当 C66x 内核休眠时,如果出现一个到 L1D、L1P 或 L2 存储器的 DMA 访问,PDC 唤醒所有存储控制器;当 DMA 访问完成后,PDC 会重新休眠所有存储控制器。

## 1.6 锁相环及其设置

锁相环(Phase Locked Loop,PLL)是处理器的时钟源,控制着 C6678 处理器中 C66x 内核、各外围设备的时钟。

### 1.6.1 主 PLL 和 PLL 控制器

主 PLL 由标准 PLL 控制器控制。PLL 控制器负责管理处理器系统时钟的时钟比、对准和选通功能。如图 1.6 所示为主 PLL 和 PLL 控制器的功能框图。

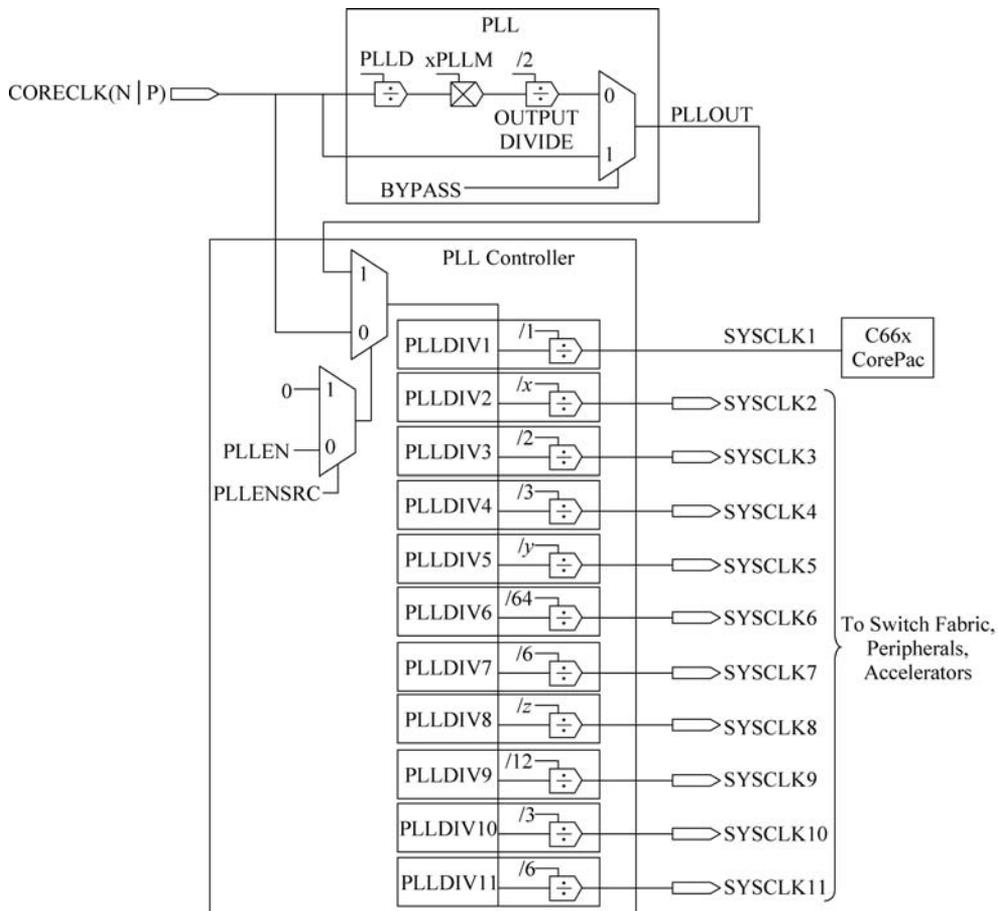


图 1.6 主 PLL 和 PLL 控制器

**注意：**主 PLL 控制寄存器可以被器件中的任何主设备访问。倍频器的 PLLM[5:0]位被 PLL 控制器中的 PLLM 寄存器控制, PLLM[12:6]位被器件级 MAINPLLCTL0 寄存器控制。输出除数和 PLL 旁路(Bypass)逻辑被 SECCTL 寄存器相应的域控制。在 C6678 器件里,只有 PLLDIV2、PLLDIV5 和 PLLDIV8 是可编程的。

主 PLL 用于驱动内核、交叉开关网络(Switch Fabric)和大多数外围设备的时钟(除了 DDR3 和网络协处理器(PASS))。主 PLL 的 PLL 控制器管理不同的时钟分频器、对准和同步。

主 PLL 的 PLL 控制器具有一些 SYSCLK 输出,每个 SYSCLK 具有一个相应的分频器对 PLL 输出的时钟分频。

**注意：**除了在下面描述中明确提到可编程的 SYSCLK 外,其他时钟分频器不是可编程的。

SYSCLK1: 用于内核的全比例时钟。

SYSCLK2:  $1/x$  比例时钟,用于内核(仿真)。默认的比例是  $1/3$ ,这是可编程的,范围从  $1$  到  $32$ ,该时钟最大不能超过  $350\text{MHz}$ 。SYSCLK2 可以被软件关掉。

SYSCLK3:  $1/2$  比例时钟,用于 MSMC 时钟、HyperLink、CPU/2 SCR、DDR EMIF 和 CPU/2 EDMA。

SYSCLK4:  $1/3$  比例时钟,用于交叉开关网络和高速外围设备。Debug\_SS 和 ETBs 也会使用这个时钟。

SYSCLK5:  $1/y$  比例时钟,只用于系统追踪(System Trace)模块。默认比例是  $1/5$ ,可以被配置,最大配置时钟是  $210\text{MHz}$ 、最小配置时钟是  $32\text{MHz}$ 。SYSCLK5 可以被软件关掉。

SYSCLK6:  $1/64$  比例时钟(emif\_ptv),被用于驱动 DDR3 EMIF PVT 补偿缓冲。

SYSCLK7:  $1/6$  比例时钟,用于慢速外围设备和资源的系统输出引脚。

SYSCLK8:  $1/z$  比例时钟,该时钟被用作系统中的慢速系统时钟,默认的比例是  $1/64$ ,可以被编程设置为  $1/24 \sim 1/80$ 。

SYSCLK9:  $1/12$  比例时钟,用于 SmartReflex。

SYSCLK10:  $1/3$  比例时钟,只用于 SRIO。

SYSCLK11:  $1/6$  比例时钟,只用于 PSC。

主 PLL 用两个芯片级寄存器——主 PLL 控制寄存器 0(MAINPLLCTL0)和主 PLL 控制寄存器 1(MAINPLLCTL1)来实现配置。MAINPLLCTL0 的组成结构如图 1.7 所示,寄存器说明见表 1.3; MAINPLLCTL1 的组成结构如图 1.8 所示,寄存器说明见表 1.4。

31	24	23	19	18	12	11	6	5	0
BWADJ[7:0]		Reserved		PLLM[12:6]		Reserved		PLLD	
RW-0000 0101		RW-0000 0		RW-000000		RW-000000		RW-000000	

图 1.7 MAINPLLCTL0 的组成结构

RW: 读和写; W: 只写; R: 只读;

-n: 表示复位后的值(下同)。

表 1.3 MAINPLLCTL0 说明

位	域	描 述
31~24	BWADJ[7:0]	BWADJ [11:8]和 BWADJ [7:0]分别位于 MAINPLLCTL0 和 MAINPLLCTL1 寄存器,寄存器组合(BWADJ [11:0])必须被配置一个值
23~19	Reserved	保留
18~12	PLLM[12:6]	一个 13 位总线,用于选择倍频因子(此为高位部分)
11~6	Reserved	保留
5~0	PLLD	一个 6 位总线,用于选择分频因子

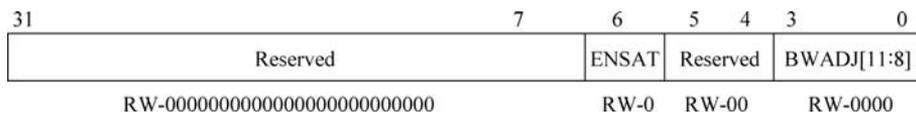


图 1.8 MAINPLLCTL1 的组成结构

表 1.4 MAINPLLCTL1 说明

位	域	描 述
31~7	Reserved	保留
6	ENSAT	需要被设置为 1,用于正确操作 PLL
5~4	Reserved	保留
3~0	BWADJ[11:8]	BWADJ [11:8]和 BWADJ [7:0]分别位于 MAINPLLCTL0 和 MAINPLLCTL1 寄存器。寄存器组合(BWADJ [11:0])必须被设置等于 PLLM[12:0]值的一半,如果 PLLM 是一个奇数则向下取整,如 PLLM=15,那么 BWADJ =7

**注意:** 当 PLL 控制器中 GO 操作被发起时,MAINPLLCTL0 寄存器 PLLM[12:6]位必须被先写,然后写控制器的 PLLM 寄存器 PLLM[5:0]位,以获得完整的 13 位寄存器值的锁存。详细信息参考 *Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide*。PLL 二级控制寄存器(SECCTL)如图 1.9 所示,寄存器说明见表 1.5。

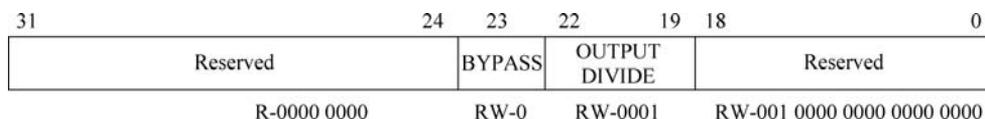


图 1.9 PLL 二级控制寄存器

表 1.5 PLL 二级控制寄存器说明

位	域	描 述
31~24	Reserved	保留
23	BYPASS	主 PLL 旁路使能: 0=主 PLL 旁路禁止; 1=主 PLL 旁路使能
22~19	OUTPUT DIVIDE	输出分频比例: 0h=÷1,分频比例为 1; 1h=÷2,分频比例为 2; 2h-Fh= Reserved
18~0	Reserved	保留

其他相关寄存器详见相关手册。

### 1.6.2 DDR3 PLL

DDR3 PLL 为 DDR3 存储控制器产生的接口时钟。当上电复位后,在加载配置期间、使能和使用之前,DDR3 PLL 被程序设置为一个有效的频率。如图 1.10 所示为 DDR3 PLL 功能结构图。

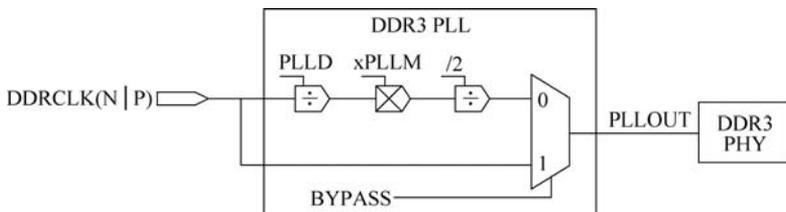


图 1.10 DDR3 PLL 功能结构图

#### 1. DDR3 PLL 控制寄存器

DDR3 PLL 被用于驱动 DDR PHY EMIF,没有使用 PLL 控制器。通过使用 Bootcfg 模块中的 DDR3PLLCTL0 和 DDR3PLLCTL1 寄存器,可以控制 DDR3 PLL。这些存储器映射寄存器(Memory-Mapped Registers,MMRs)存在于 Bootcfg 空间内部。为了写这些寄存器,软件必须使用 KICK0/KICK1 寄存器完成一个解锁流程。如图 1.11 所示为 DDR3 PLL 控制寄存器 0(DDR3PLLCTL0)的组成结构图,表 1.6 为其说明。

31	24	23	22	19	18	6	5	0
BWADJ[7:0]	BYPASS	Reserved			PLLM		PLLD	
RW-0000 1001	RW-0	RW-0001			RW-0000000010011		RW-000000	

图 1.11 DDR3PLLCTL0 的组成结构

表 1.6 DDR3PLLCTL0 说明

位	域	描述
31~24	BWADJ [7:0]	BWADJ[11:8]和 BWADJ[7:0]分别位于 DDR3PLLCTL0 和 DDR3PLLCTL1 寄存器。寄存器组合(BWADJ[11:0])必须被程序设定为等于 PLLM[12:0]值的一半(如果 PLLM 是一个奇数则向下取整),例如 PLLM=15,那么 BWADJ=7
23	BYPASS	使能旁路模式: 0=旁路禁止; 1=旁路使能
22~19	Reserved	保留
18~6	PLLM	一个 13 位总线,用于选择倍频因子
5~0	PLLD	一个 6 位总线,用于选择分频因子

如图 1.12 所示为 DDR3 PLL 控制寄存器 1(DDR3PLLCTL1)的组成结构图,表 1.7 为其说明。

31	14	13	12	7	6	5	4	3	0
Reserved		PLL_RST	Reserved		ENSAT	Reserved		BWADJ[11:8]	
RW-000000000000000000		RW-0	RW-000000		RW-0	R-0		RW-00000	

图 1.12 DDR3PLLCTL1 的组成结构

表 1.7 DDR3PLLCTL1 说明

位	域	描 述
31~14	Reserved	保留
13	PLL_RST	PLL 复位位： 0 = PLL 复位被释放； 1 = PLL 复位被确认
12~7	Reserved	保留
6	ENSAT	需要被设置为 1, 用于正确操作 PLL
5~4	Reserved	保留
3~0	BWADJ [11:8]	BWADJ [11:8] 和 BWADJ [7:0] 被分配在 DDR3PLLCTL0 和 DDR3PLLCTL1。寄存器组合 (BWADJ [11:0]) 必须被程序设定为等于 PLLM [12:0] 值的一半 (如果 PLLM 是一个奇数则向下取整), 如 PLLM = 15, 那么 BWADJ = 7

## 2. DDR3 PLL 器件特定信息

如图 1.10 所示, DDR3 PLL (PLLOUT) 的输出被二分频并直接接到 DDR3 存储控制器。

DDR3 PLL 在上电复位时失锁, 当 RESETSTAT 引脚拉高时被锁定, 在任何其他复位时都不会失锁。

主 PLL 和 PLL 控制器必须总是在 DDR3 PLL 之前被初始化, 必须遵照如下顺序初始化 DDR3 PLL:

- (1) 在寄存器 DDR3PLLCTL1 中, 写 ENSAT = 1 (用于最佳的 PLL 操作)。
- (2) 在寄存器 DDR3PLLCTL0 中, 写 BYPASS = 1 (设置 PLL 旁路)。
- (3) 在寄存器 DDR3PLLCTL1 中, 写 PLL\_RST = 1 (PLL 被复位)。
- (4) 程序设置 DDR3PLLCTL0 寄存器中的 PLLM 和 PLLD。
- (5) 程序设置 DDR3PLLCTL0 中的 BWADJ [7:0] 和 DDR3PLLCTL1 寄存器中的 BWADJ [11:8]。BWADJ 值必须被设置为  $((\text{PLLM} + 1) \gg 1) - 1$ 。
- (6) 基于参考时钟, 至少等待  $5\mu\text{s}$  (PLL 复位时间)。
- (7) 在寄存器 DDR3PLLCTL1 中, 写 PLL\_RST = 0 (PLL 复位被释放)。
- (8) 至少等待  $500 \times \text{REFCLK 周期} \times (\text{PLLD} + 1)$  (PLL 锁定时间)。
- (9) 在寄存器 DDR3PLLCTL0 中, 写 BYPASS = 0 (切换到 PLL 模式)。

**注意:** 对于 PLL 中的任何寄存器, 软件必须总是按照“读-修改-写”的顺序执行。这是为了确保只有相关的寄存器位被修改, 剩下的位 (包括保留位) 不会被影响。

### 1.6.3 PASS PLL

PASS PLL 产生网络协处理器的时钟接口。通过使用 PACLKSEL 引脚, 用户可以选择 PASS PLL 输入源, 从 CORECLK 输出时钟参考源或 PASSCLK 参考源中选择一个作为

输入。在上电复位时, PASS PLL 为旁路模式(BYPASS Mode), 在使能和使用前需要被重新设置到一个有效频率。PASS PLL 的示意图如图 1.13 所示。

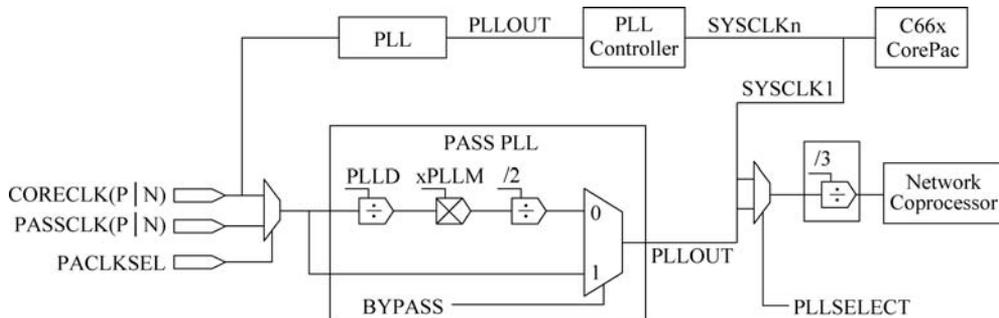


图 1.13 PASS PLL 示意图

## 1.7 C6678 处理器接口通信相关外围设备

C6678 具有丰富的外围设备, 可以协助处理器内核完成很多功能, 如高速通信接口(HyperLink、SRIO、PCIE 等)、低速接口(I<sup>2</sup>C、UART 等)、通用目的输入输出(General-Purpose Input/Output, GPIO)等。本节介绍 C6678 处理器中与接口通信相关的外围设备。

### 1.7.1 I<sup>2</sup>C 外围设备

TMS320C6678 器件包含一个 I<sup>2</sup>C(Inter Integrated Circuit)外围设备模块, 并提供一个 DSP 与其他器件用 I<sup>2</sup>C 连接的接口。该接口遵从飞利浦半导体 I<sup>2</sup>C bus 规范(版本 2.1), 外部部件可以通过 2 线串行总线与 DSP 连接, 实现最多 8 位数据的收发。

**注意:** 当使用 I<sup>2</sup>C 模块时, 应确保在 SDA 和 SCL 引脚上有外部上拉电阻。

I<sup>2</sup>C 模块可以被 DSP 用来控制本地外围器件(DAC、ADC 等), 与一个系统中其他控制器通信, 或用来实现一个用户接口。通常, I<sup>2</sup>C 在系统中也用于健康管理。I<sup>2</sup>C 模块结构图如图 1.14 所示, 接口简单、通用性好。

I<sup>2</sup>C 模块具有以下特点。

(1) 符合飞利浦半导体 I<sup>2</sup>C 总线规范(2.1 版)。

① 支持字节格式传输。

② 7 位和 10 位寻址模式。

③ 通用广播(General Call), 当发出[0000000]的地址信息后, 所有 I<sup>2</sup>C 上的从设备(Slave)都要对此做出反应, 该机制适合用在主设备(Master)要对所有的从设备进行广播性讯息更新与沟通的场合, 是一种总体、批次的运作方式。

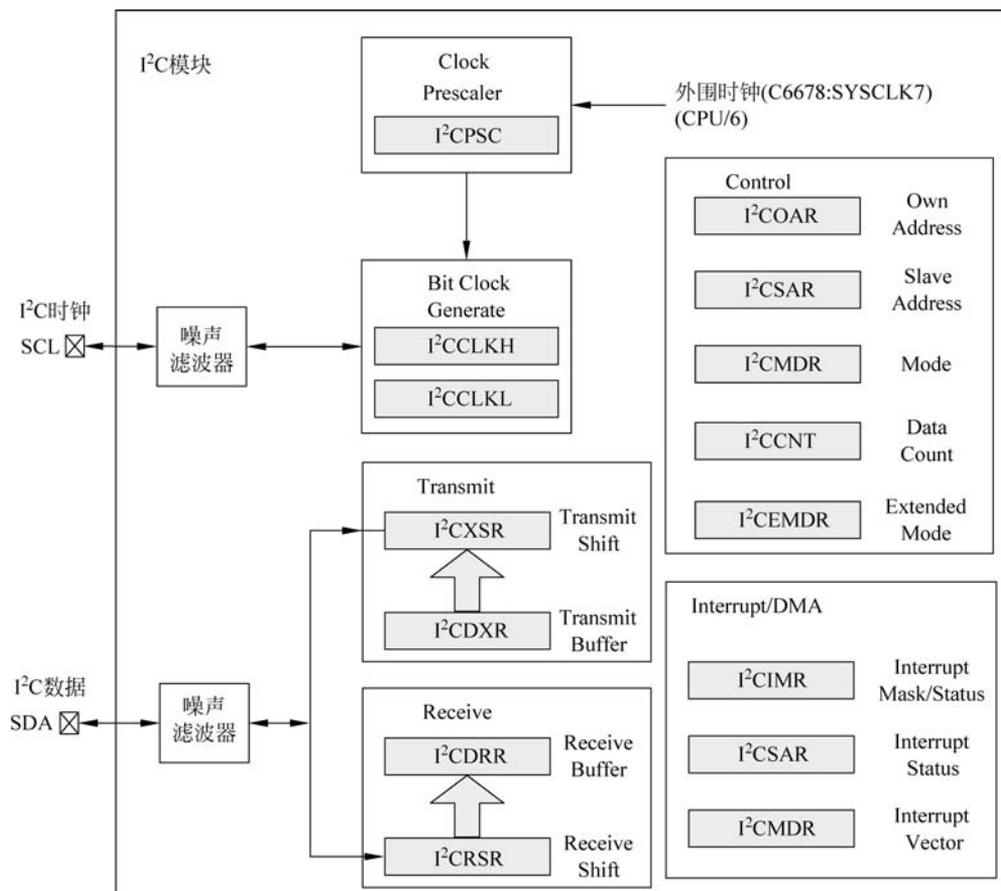
④ START 字节模式。

⑤ 支持多个主发送(Master-Transmitter)和从接收模式(Slave-Receiver)。

⑥ 支持多个从发送(Slave-Transmitter)和主接收(Master-Receiver)模式。

⑦ 主发送/接收和接收/发送模式结合。

⑧ I<sup>2</sup>C 数据传输速率从 10kb/s 到 400kb/s。

图 1.14 I<sup>2</sup>C 模块组成框图

(2) 2~7 位格式传输。

(3) 自由数据格式(Free Data Format, FDF)模式,在 FDF 数据格式中,START 条件后最初的一些位是数据字(Data Word)。在每个数据字之后插入一个 ACK 位,根据 ICMDR (I<sup>2</sup>C Mode Register)的位计数(Bit Count, BC)位,数据字可以是 1~8 位。不发送地址或数据方向位。因此,发送者和接收者都必须支持 FDF 数据格式,并且在整个传输过程中数据的方向必须是恒定的。

(4) 一个读 DMA 事件和一个写 DMA 事件,事件可由 DMA 使用。

(5) 7 个 CPU 可以使用的中断。

I<sup>2</sup>C 模块由以下主要模块组成。

(1) 串行接口:一个数据引脚(SDA)和一个时钟引脚(SCL)。

(2) 数据寄存器:在 SDA 引脚与 CPU 或 EDMA 控制器之间传输的数据,数据寄存器用于临时保存接收和发送的数据。

(3) 控制和状态寄存器。

(4) EDMA 总线接口,使 CPU 和 EDMA 控制器能够访问 I<sup>2</sup>C 模块寄存器。

(5) 时钟同步器,同步 I<sup>2</sup>C 输入时钟(来自时钟发生器(Clock Generator))和 SCL 引脚

上的时钟,并与不同时钟速度的主机同步数据传输。

(6) 一个预分频器(Prescaler),用于划分驱动至 I<sup>2</sup>C 模块的输入时钟。

(7) SDA 和 SCL 两个引脚上各有一个噪声滤波器。

(8) 处理 I<sup>2</sup>C 模块(当它是主模块时)和另一个主模块之间仲裁的仲裁器。

(9) 中断生成逻辑,以便中断可以发送到 CPU。

(10) EDMA 事件生成逻辑,以便 EDMA 控制器中的行为可以与 I<sup>2</sup>C 模块中的数据接收和数据传输同步。

I<sup>2</sup>C 外围设备的寄存器如表 1.8 所示。CPU 或 EDMA 控制器将数据写入 ICDXR(I<sup>2</sup>C Data Transmit Register),并从 ICDRR(I<sup>2</sup>C Data Receive Register)读取接收到的数据。当 I<sup>2</sup>C 模块被配置为发送时,写入 ICDXR 的数据被复制到 ICXSR(I<sup>2</sup>C Transmit Shift Register),并在 SDA 引脚上一次移位一位。当 I<sup>2</sup>C 模块被配置为接收器时,接收的数据被转移到 ICRSR(I<sup>2</sup>C Receive Shift Register)中,然后被复制到 ICDRR 中。

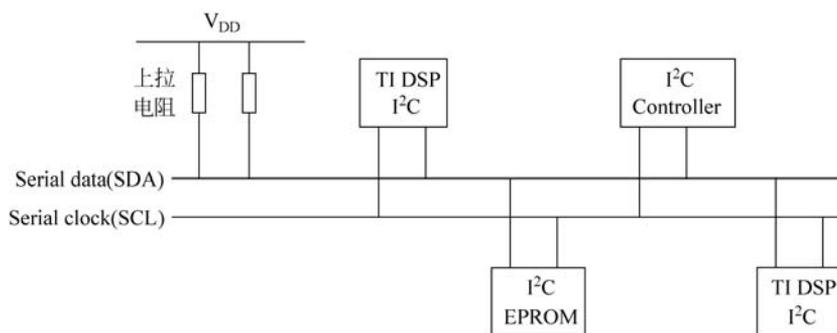
表 1.8 I<sup>2</sup>C 外围设备的寄存器

Hex 地址	寄存器	说明
0253 0000	ICOAR	I <sup>2</sup> C 自己的地址寄存器(Own Address Register)
0253 0004	ICIMR	I <sup>2</sup> C 中断屏蔽/状态寄存器(Interrupt Mask/Status Register)
0253 0008	ICSTR	I <sup>2</sup> C 中断状态寄存器(Interrupt Status Register)
0253 000C	ICCLKL	I <sup>2</sup> C 时钟低(low)分频器寄存器(Clock Low-Time Divider Register)
0253 0010	ICCLKH	I <sup>2</sup> C 时钟高(high)分频器寄存器(Clock High-Time Divider Register)
0253 0014	ICCNT	I <sup>2</sup> C 数据计数寄存器(Data Count Register)
0253 0018	ICDRR	I <sup>2</sup> C 数据接收寄存器(Data Receive Register)
0253 001C	ICSAR	I <sup>2</sup> C 从地址寄存器(Slave Address Register)
0253 0020	ICDXR	I <sup>2</sup> C 数据发送寄存器(Data Transmit Register)
0253 0024	ICMDR	I <sup>2</sup> C 模式寄存器(Mode Register)
0253 0028	ICIVR	I <sup>2</sup> C 中断向量寄存器(Interrupt Vector Register)
0253 002C	ICEMDR	I <sup>2</sup> C 扩展模式寄存器(Extended Mode Register)
0253 0030	ICPSC	I <sup>2</sup> C 预分频器寄存器(Prescaler Register)
0253 0034	ICPID1	I <sup>2</sup> C 外围识别寄存器 1(Peripheral Identification Register) [值: 0x0000 0105]
0253 0038	ICPID2	I <sup>2</sup> C 外围识别寄存器 2[值: 0x0000 0005]
0253 003C- 0253 007F	Reserved	保留

I<sup>2</sup>C 总线是支持多主模式的多主总线。这允许多个设备能够连接到 I<sup>2</sup>C 总线,并能控制该总线。每个 I<sup>2</sup>C 设备由一个唯一的地址识别,并且可以根据设备的功能作为发送器或接收器工作。除了作为发送器或接收器外,连接到 I<sup>2</sup>C 总线的设备在执行数据传输时也可以被视为主设备或从设备。

请注意,主设备是在总线上启动数据传输并生成时钟信号以允许该传输的设备。在这个传输过程中,由该主机寻址的任何设备都被认为是从机。图 1.15 为多个 I<sup>2</sup>C 模块连接的示例,用于从一个设备到其他设备的双向传输。

I<sup>2</sup>C 的更详细信息见 *KeyStone Architecture Inter-IC Control Bus(I<sup>2</sup>C)User Guide*。

图 1.15 多个 I<sup>2</sup>C 模块连接图

### 1.7.2 SPI 外围设备

SPI 模块提供一个 DSP 和其他兼容 SPI 器件的接口,该接口的主要功能是连接 SPI ROM 用于引导(Boot),也可以连接其他芯片级组件(如温度传感或 I/O 扩展)。C6678 SPI 模块只支持主模式。

SPI 是一个高速同步串行输入/输出端口,允许的串行比特流可设置长度为 2~16 位,以可设置的比特传输率移入和移出器件。SPI 通常用于器件和外部外设之间的通信。典型的应用包括诸如移位寄存器、显示驱动器、SPI EPROM 和模数转换器等设备与外部 I/O 或外围扩展的接口。C6678 SPI 支持 3-pin 和 4-pin 两种模式。对于 4-pin 芯片选择模式, C6678 最多支持两个芯片选择。

SPI 具有以下功能。

- (1) 16 位移位寄存器。
  - (2) 16 位接收缓冲寄存器 (SPIBUF) 和 16 位 SPI 仿真寄存器 (SPI Emulation Register, SPIEML)。
  - (3) 16 位传输数据寄存器 (SPIDAT0) 和 16 位传输数据和格式选择寄存器 (SPIDAT1)。
  - (4) 8 位波特时钟发生器。
  - (5) 串行时钟 (SPICLK) I/O 引脚。
  - (6) 从输入、主输出 (SPISIMO) I/O 引脚。
  - (7) 从输出、主输入 (SPISOMI) I/O 引脚。
  - (8) 多个从芯片选择 ( $\overline{\text{SPISCS}}[n]$ ) I/O 引脚(仅 4-pin 模式)。
  - (9) 可编程 SPI 时钟频率范围。
  - (10) 可编程字符长度(2~16 位)。
  - (11) 可编程时钟相位(延迟或无延迟)。
  - (12) 可编程时钟极性(高或低)。
  - (13) 中断能力。
  - (14) 支持 DMA(读/写同步事件)。
  - (15) 高达 66MHz 的工作频率。
- SPI 允许软件对以下选项进行编程。

- (1) SPICLK 频率(SPI 模块 Clock/2 至 SPI 模块 Clock/256)。
- (2) 3-pin 和 4-pin 选项。
- (3) 字符长度(2~16 位)和移位方向(MSB/LSB 优先)。
- (4) 时钟相位(延迟或无延迟)和极性(高或低)。
- (5) 主模式下传输之间的延迟。
- (6) 主模式下芯片选择建立和保持时间(Chip Select Setup and Hold Time)。
- (7) 主模式下芯片选择保持(Chip Select Hold)。

SPI 不支持以下功能。

- (1) 多缓冲区模式。
- (2) 并行模式或奇偶校验。
- (3) SPIENA Pin。
- (4) SPI 从机模式。
- (5) GPIO 模式。

SPI 功能框图如图 1.16 所示。

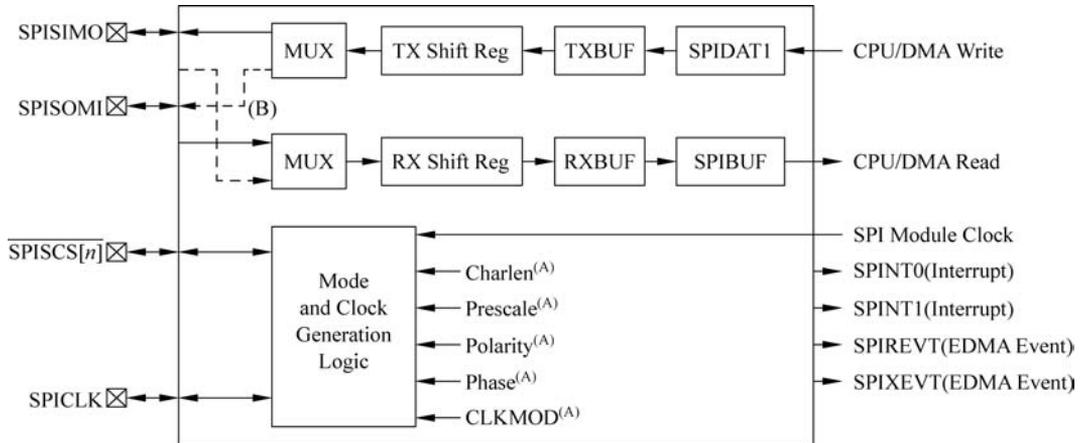


图 1.16 SPI 功能框图

注：(A) 指示由 SPI 寄存器位控制的记录。(B) 实线表示 SPI 主模式的数据流。

SPI 引脚的描述如表 1.9 所示。

表 1.9 SPI 引脚描述

引 脚	输入输出类型	功 能
SPISIMO	输出	主模式串行数据输出
SPISOMI	输入	主模式下的串行数据输入
SPICLK	输出	主模式下的串行时钟输出
SPISCS[n]	输出	主模式下从端芯片选择输出

SPI 工作在主模式。SPI 总线主设备是驱动 SPICLK、SPISIMO 和可选的 SPISCS[n] 信号的设备,因而启动 SPI 总线传输。SPI 全局控制寄存器 1(SPIGCR1)中的 CLKMOD 和 MASTER 位选择主模式。在主模式下,SPI 支持如下两个选项。

- (1) 3-pin 选项;
- (2) 4-pin 带芯片选择选项。

3-pin 选项是基本的时钟、数据输入和数据输出 SPI 接口,并使用 SPICLK、SPISIMO 和 SPISOMI 引脚。3-pin SPI 连接关系图见图 1.17。SPI 总线主模式是驱动 SPICLK 信号并启动 SPI 总线传输的设备。在 SPI 主模式下,SPISOMI 引脚输出缓冲区处于高阻抗状态,SPICLK 和 SPISIMO 引脚输出缓冲区已启用。

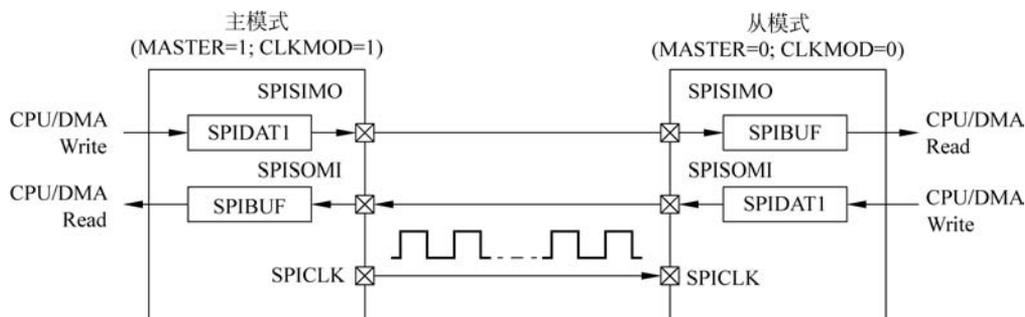


图 1.17 3-pin SPI 连接关系图

在带有 3-pin 选项的主模式下,DSP 将发送数据写入 SPI 发送数据寄存器 (SPIDAT0 [15:0] 或 SPIDAT1[15:0])。这将启动传输,一系列时钟脉冲将在 SPICLK 引脚上驱动出来。SPICLK 引脚上的每个时钟脉冲都会导致主 SPI 设备和从 SPI 设备同时传输一个位(在两个方向)。CPU 写入 SPIDAT1 中的配置位(不是写入 SPIDAT1[15:0])不会导致新的传输。当选定的位数被发送时,接收的数据被传输到 SPI 接收缓冲寄存器 (SPIBUF) 以便 CPU 读取。数据以 SPIBUF 格式右对齐存储。

4-pin 带芯片选择选项,添加用于支持单个 SPI 总线上多个 SPI 从设备的  $\overline{\text{SPISCS}}[n]$  引脚。4-pin SPI 连接关系图见图 1.18。 $\overline{\text{SPISCS}}[n]$  应通过配置 SPI 引脚控制寄存器 0 (SPI Pin Control Register 0, SPIPC0) 将引脚配置为功能引脚。在 SPI 主模式下,SPISOMI 引脚输出缓冲区处于高阻抗状态,SPICLK、SPISIMO 和  $\overline{\text{SPISCS}}[n]$  引脚输出缓冲区已启用。

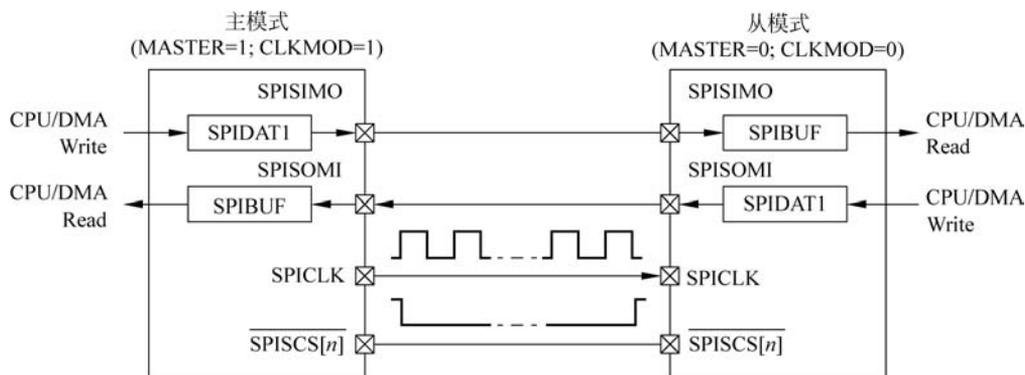


图 1.18 4-pin SPI 连接关系

在主模式下, $\overline{\text{SPISCS}}[n]$  引脚用作输出,并在选择特定从设备时切换。然而,这在支持多个  $\overline{\text{SPISCS}}[n]$  引脚的设备上最有用。SPI 只支持一个  $\overline{\text{SPISCS}}[n]$ ,因此该引脚在主模式

下的用处有限。在实际应用中,需要通用的 I/O (GPIO) 引脚来支持多个从设备芯片的选择。

SPI 寄存器的逻辑地址范围为 0x20BF0000~0x20BF01FF, SPI 寄存器的描述如表 1.10 所示。

表 1.10 SPI 寄存器的描述

偏移地址	寄存器	描述
0h	SPIGCR0	全局控制寄存器 0(Global Control Register 0): 包含模块的软件复位位
4h	SPIGCR1	全局控制寄存器 1(Global Control Register 1): 控制模块的基本配置
8h	SPIINT0	中断寄存器(Interrupt Register): Interrupts、Error、DMA 和其他功能的使能位
Ch	SPIVLV	级别寄存器(Level Register): SPI 中断级别在此寄存器中设置
10h	SPIFLG	标志寄存器(Flag Register): 描述操作期间多个事件的状态
14h	SPIPC0	引脚控制寄存器 0(Pin Control Register 0): 确定引脚是否作为通用 I/O 或 SPI 功能引脚运行
38h	SPIDAT0	发送数据寄存器 0(Transmit Data Register 0)
3Ch	SPIDAT1	发送数据寄存器 1(Transmit Data Register 1): 具有格式选择的发送数据寄存器
40h	SPIBUF	接收缓冲寄存器(Receive Buffer Register): 保留接收的字
44h	SPIEMU	接收缓冲区仿真(Receive Buffer Emulation): SPIBUF 的镜像, 读取不清除标志寄存器
48h	SPIDELAY	Delay Register: 设置 SPISCS[n] 模式, SPISCS[n] 传输前/传输后延迟时间
4Ch	SPIDEF	芯片选择默认寄存器(Chip Select Default Register): 仅在 SPISCS[n] 解码模式下设置 SPISCS[n] 信号高/低有效
50h	SPIFMT0	Format 0 Register: 数据字 Format 0 的配置
54h	SPIFMT1	Format 1 Register: 数据字 Format 1 的配置
58h	SPIFMT2	Format 2 Register: 数据字 Format 2 的配置
5Ch	SPIFMT3	Format 3 Register: 数据字 Format 3 的配置
60h	INTVEC0	中断向量寄存器 0(Interrupt Vector Register 0): INT0 的中断向量
64h	INTVEC1	中断向量寄存器 1(Interrupt Vector Register 1): INT1 的中断向量

SPI 的更详细信息见 *KeyStone Architecture Serial Peripheral Interface (SPI) User Guide*。

### 1.7.3 HyperLink 外围设备

C6678 包含 HyperLink 总线用于芯片接口, 这是一个 4 线(Lane) SerDes 接口, 最高通信速率为每线 12.5Gb/s, 支持的数据率包括 1.25Gb/s、3.125Gb/s、6.25Gb/s、10Gb/s 和 12.5Gb/s。

**注意:** HyperLink 必须用直流耦合器连接, 详见 *KeyStone Architecture HyperLink User Guide*。

HyperLink 提供了一个高速、低延迟和引脚数少的通信接口, 扩展了两个 KeyStone 器件之间内部基于 CBA 3.x(Common Bus Architecture)的事务。它可以模拟当前使用的所