

2.1 DSP 总体结构

2.1.1 概述

TMS320F28335 具有丰富的片上资源，快速的信息处理能力和强大的控制功能，其结构框图如图 2.1 所示。包括片内总线：程序总线、数据总线、DMA 总线和 CPU 内的寄存器总线（位于 CPU 内部）；中央处理器、时钟系统、存储单元、CPU 定时器、PIE 中断管理单元、仿真与调试端口、外部存储器接口 XINTF，以及众多片上外设（虚线框部分）。DSP 系统架构采用改进的哈佛结构，片上各功能单元均挂在程序总线、数据总线或 DMA 总线上，通过总线传输地址、数据和控制信息。

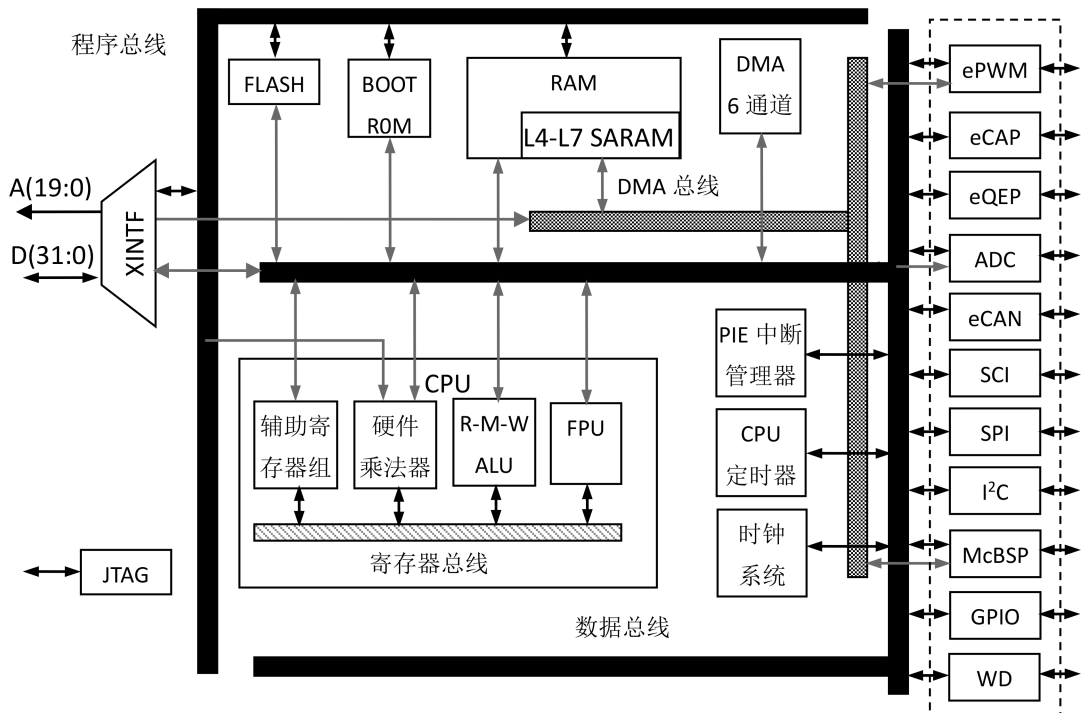


图 2.1 TMS320F28335 结构框图

2.1.2 片内结构及其功能

主要片上功能单元如下。

(1) CPU

TMS320F28335 的 CPU 采用 C28x+FPU 架构，包括 1 个 C28x 系列的 32 位定点运算内核，和 1 个单精度 32 位 IEEE-754 浮点处理内核 FPU。其中，32 位定点运算内核，又包括输入定标部分，可以进行 32×32 位或者 16×16 位乘法运算和乘累加运算的乘法部分和算术逻辑部分等。

(2) 存储单元

片内存储器是 DSP 核心功能单元之一，是影响 DSP 运行性能的重要因素。F28335 的片内存储器映射，包括 4M 的数据存储空间和 4M 的程序存储空间。片内存储空间又配置了各种存储器：256KW（W：16 位字长）FLASH；34KW 单周期访问 SARAM，其中 L4~L7 单元支持 DMA 直接存储访问模式；2MW+4KW 的外部存储器接口映射存储单元 XINTF：XINTF Zone0 4KW、XINTF Zone6 1MW、XINTF Zone7 1MW；外设寄存器帧 PF0~PF3；BOOT ROM，该存储区域又包括加载程序、定点/浮点数学表、产品的版本号和校验信息，以及一个 CPU 中断向量表（地址范围 0x3FFFC0~0x3FFFFFF），其中引导程序由 TI 公司出厂前设置并固化，可以根据 DSP 特定管脚的设置，在开机时实现不同模式下的程序加载。

(3) 片上外设

如图 2.1 所示，右侧虚线框内是重要的片上外设功能单元，包括如下片上外设：6 通道的 ePWM 单元，可以输出高达 18 路的 PWM 波信号；拥有 6 路并行捕获/比较通道的 eCAP 单元；2 个增强型正交编码器单元 eQEP；1 个 12 位精度的模/数转换器，拥有 16 路模拟信号输入通道；2 个 eCAN 总线接口单元；3 个异步通信接口（SCI）；1 个同步通信接口单元 SPI；1 个标准 I²C 通信接口单元；2 个多通道缓冲串行接口单元 McBSP；通用型数字输入输出引脚功能单元 GPIO，拥有 88 个可独立编程、可进行内部滤波的复用引脚。其中，ePWM、eQEP 和 eCAP 功能单元，主要针对电机控制与信号检测，是 F2833x 系列产品与众不同的特色设计。

(4) 基本功能单元

还有一些基本的功能单元，包括中断管理单元 PIE、时钟系统、CPU 定时器、电源系统等。其中中断管理单元 PIE，可以管理高达 96 个来自芯片内外的外设级中断，目前实际管理有 58 个；时钟系统，可以直接利用来自外部的标准时钟信号，也可以使用外部晶振，加内部电路一起形成振荡源，产生初始时钟信号，经过 PLL 锁相环的稳频、倍频，形成 CPU 输入时钟 CLKIN，为 WD 和 CPU，以及片上众多功能单元提供时钟信号；3 个 32 位的 CPU 定时器 T0、T1 和 T3；电源系统，DSP 的电源系统可以提供 3.3V、1.8、1.9V 三种不同的电压，以满足 DSP 在不同工作状态下的电源需求。

(5) 总线

F28335 采用改进的哈佛总线结构，几乎所有的功能单元均挂接在总线上，通过总线进行软件代码的读写以及数据信息的传递，其中寄存器总线位于 CPU 内部，是 CPU 各功能单元进行信息传输的高速通道，并与 CPU 外部总线相连接；DMA 总线是直接存储总线，不需要 CPU 的干预也不占用主程序执行时间，与 DMA 总线连接的功能单元，包括 DMA 控制器、SARAM（L4~L7）、ePWM、ADC、McBSP 等。

2.2 总线与流水线

2.2.1 总线及其结构

总线是 DSP 的重要组成部分之一，F2833x 总线包括片内总线和片外总线两部分。

如图 2.2 所示，左侧虚线框部分即为外部总线，由 20 位宽的地址线 A (19:0)、32 位宽的数据线 D (31:0) 和相关控制信号线组成，是标准的单总线结构，便于兼容单总线外部器件。

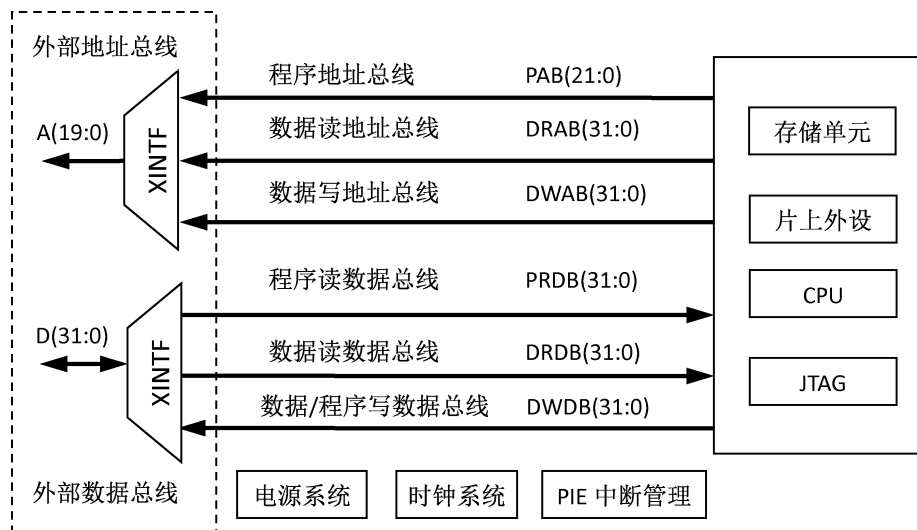


图 2.2 总线结构

DSP 片内总线与外部单总线结构不同，采用了改进的哈佛结构，包括地址总线和数据总线，上述总线又各自采用了分离的三总线结构，因此，共两类、六组总线，如图 2.2 所示。

地址总线：包括程序地址总线 PAB (22 位)、数据读地址总线 DRAB (32 位)、数据写地址总线 DWAB (32 位)；数据总线：包括程序读数据总线 PRDB (32 位)、数据读数据总线 DRDB (32 位)、数据/程序写数据总线 DWDB (32 位)。地址总线和数据总线各有 3 组总线。

其中，程序地址总线 PAB 为 22 位，所以，程序空间（读/写）的最大寻址范围 $2^{22}=4\text{MW}$ ；数据读地址总线 DRAB 和数据写地址总线 DWAB 均为 32 位，故数据空间的（读/写）最大寻址范围 $2^{32}=4\text{GW}$ 。但是，目前 F28335 片内只有 4M 大小的数据存储空间，所以 32 位数据地址总线仅使用了其中的低 22 位地址线（对应 4MW 寻址空间）。程序（空间）拥有自己独立的地址总线 PAB，数据（空间）拥有自己独立的读/写地址总线 DRAB/DWAB，片内程序空间与片内数据空间虽然大小相同、且起始与结束地址一致，但是却相互独立。

由于数据空间的读操作地址总线 DRAB，与数据空间写操作地址总线 DWAB，两者相互独立，所以数据的读/写操作可以并行进行；程序读数据总线 PRDB 与数据读数据总线 DRDB，两者相互独立，所以程序读取操作与数据读取操作也可以并行进行；程序写数据总线与数据写数据总线，则采用分时复用模式，共用一组写数据总线 DWDB。

改进的哈佛结构，又称三总线结构，是 F2833x 微控制器最显著的特征之一，大幅提升

了存储速度和运行效率。

注意：由于程序空间的地址读、写操作共用 PAB 地址总线，故程序空间的地址读/写寻址操作不能同时发生；同样，由于程序空间数据的写操作与数据空间数据的写操作共用 DWDB 数据总线，故上述写操作也不能同时发生。

外部总线，是 DSP 芯片对外部设备或外部扩展存储区进行操作时的总线，为了与其他外设兼容，采用了单总线结构，即只有一组 20 位宽地址总线和一组 32 位宽的数据总线，采用分时复用方法，因此，对外部总线操作速度相对缓慢。

2.2.2 流水线机制

DSP 内部采取了 8 级流水线机制，即每一条指令的执行过程分为 8 个阶段：取指 1 (F1)、取指 2 (F2)、译码 1 (D1)、译码 2 (D2)、读 1 (R1)、读 2 (R2)、执行 (E)、写 (W)，每个阶段分别完成不同的任务。上述每一个阶段，执行时间均需要 1 个机器周期，但是，并非每个阶段都是有效的，故一条指令的执行最多需要 8 个机器周期。但是，由于每个周期均激活 8 条指令，分别处于不同的流水线的不同阶段，而且每个周期均有指令执行。所以，总体效果是每个机器周期可以执行完一条指令，运行高效。通过流水线机制，DSP 实现了对海量数据的高速实时处理。

2.3 中央处理单元 (CPU)

F2833x 系列 DSP 中央处理单元，简称 CPU，采用定点 C28x+FPU 浮点架构，在原来 C28x 系列定点运算中央处理器单核基础上，增加了一个可以进行浮点运算的副处理器 FPU，形成双核机制。可以进行复杂的定点和浮点运算，包括 32×32 位、 16×16 位乘法运算、乘累加运算，具有 64 位的数据处理能力，可以处理高精度浮点运算。中央处理单元 (CPU) 由输入定标单元、硬件乘法器、算术逻辑单元、32 位辅助寄存器组和 FPU 等组成。

2.3.1 中央处理单元执行机构

输入定标部分、乘法部分和算术逻辑部分，组成了中央处理单元执行机构。来自总线的 16 位数据，首先进入输入定标部分，该部分是一个 32 位的移位寄存器，将输入的 16 位数据进行定标处理，转换为 32 位数据。定标过程中，可以进行符号扩展，该功能由状态寄存器 ST0 中的符号扩展位 SXM 控制，输入定标器工作不占用 DSP 运行周期。

乘法部分，由一个 32×32 位的硬件乘法器、32 位临时寄存器 XT、32 位乘积寄存器 P 和输出移位器等组成，与软件乘法运算不同，硬件乘法器不占用 CPU 时钟周期，可以完成 32×32 位、 16×16 位乘法运算，可以在单周期内完成乘累加 (MAC) 运算，显著提升了乘法运算速度。

临时寄存器 XT 是 32 位寄存器，用于存放 32 位被乘数，也可以作为两个独立的 16 位寄存器使用，即高 16 位寄存器 T 和低 16 位寄存器 TL；32 位的乘积寄存器 P，用于存放乘法运算结果，与 XT 寄存器类似，P 寄存器也可以拆分成两个独立的 16 位寄存器使用，即高 16 位寄存器 PH 和低 16 位寄存器 PL。

进行 32×32 位乘法运算时，XT 中的 32 位被乘数与指令中给出的 32 位乘数相乘，得到

64 位的乘积，可以将乘积的高 32 位或低 32 位放入 P 寄存器；进行 16×16 位乘法运算时，T 中的 16 位被乘数与指令中给出的 16 位乘数相乘，将 32 位的乘积结果，存放入乘积寄存器 P 或累加器 ACC。

算术逻辑部分，由 32 位算术运算单元 ALU、累加器 ACC 和 32 位输出移位器等组成，主要用于完成二进制补码算术运算和逻辑运算。其中，累加器 ACC 可以用作 32 位寄存器使用，也可以拆分成两个独立的 16 位寄存器使用，即 AH（高 16 位）和 AL（低 16 位），或者每 4 位为 1 组，进一步拆分为 4 个 8 位寄存器：AH.MSB、AH.LSB、AL.MSB、AL.LSB。32 位输出移位器，可以复制 ACC 的内容，移位后送至存储器。

2.3.2 中央处理单元寄存器组

DSP 中央处理单元 CPU 的寄存器组，数量多、使用灵活、功能强大。包括临时寄存器 XT、乘积寄存器 P、累加器 ACC、程序控制寄存器 PC、返回程序计数器 RPC、辅助寄存器 XAR0~XAR7、数据页指针寄存器 DP、堆栈指针寄存器 SP、状态寄存器 ST0/ST1、中断控制寄存器 IER、中断标志寄存器 IFR，以及调试中断寄存器 DBGIER 等。其中，部分寄存器介绍如下。

(1) 程序控制寄存器 PC 和 RPC

PC 是程序计数器，RPC 是返回程序计数器，均为 22 位的程序控制寄存器。在 8 级流水线操作流程中，D2 阶段是取指阶段，PC 存放 D2 阶段指令的地址；RPC 存放执行长调用指令 LCR 时的返回地址。

(2) 辅助寄存器 XAR0~XAR7

XAR0~XAR7 均为 32 位的辅助寄存器，有两大功能。一是间接寻址时，存放操作数的地址，最大可寻址 $2^{32}=4\text{GW}$ 的数据地址空间；二是可以作为通用寄存器使用，其中，辅助寄存器的低 16 位：AR0~AR7，可以作为独立的 16 位寄存器单独访问。寄存器组由辅助寄存器算术单元 ARAU 负责管理。

(3) 数据页面指针 DP

F28335 片内数据空间大小为 4MW，对应 22 位地址线，即 $2^{22}=4\text{MW}$ ，可以进行直接寻址。首先，将片内 4M 字的数据存储空间，划分为 $2^{16}=65536$ 页，页码序号 0~65535，对应 22 位地址线的高 16 位地址编码；每个页面，最多可存储 $2^6=64$ 个字，页内偏移序号 0~63，对应 22 位地址线的低 6 位地址编码。

总之，4M 片内数据空间的直接地址寻址，分为高 16 位的数据页寻址和低 6 位的页内寻址，通过“页→页内偏移量”两级寻址的方式，实现了对数据存储空间的低 4M 空间的完全寻址。其中，高 16 位地址线所对应的地址，被称为“数据页”地址，存放于数据页指针 DP 之中，低 6 位地址线所对应的页内地址，则由寻址指令中的偏移量直接给出。数据页面指针 DP 寻址，如表 2.1 所示。

表 2.1 数据空间低 4MW 分页

页码 (Page)	22 位地址线分页寻址		数据存储空间地址范围
	DP 值 (高 16 位)	页内偏移量 (低 6 位)	
第 0 页	0000 0000 0000 0000	000000~111111	0x00 0000~0x00 003F
第 1 页	0000 0000 0000 0001	000000~111111	0x00 0040~0x00 007F

续表

页码 (Page)	22 位地址线分页寻址		数据存储空间地址范围
	DP 值 (高 16 位)	页内偏移量 (低 6 位)	
第 2 页	0000 0000 0000 0010	000000~111111	0x00 0080~0x00 00BF
第 3 页	0000 0000 0000 0011	000000~111111	0x00 00C0~0x00 00FF
⋮	⋮	⋮	⋮
第 65533 页	1111 1111 1111 1101	000000~111111	0x3F FF40~0x3F FF7F
第 65534 页	1111 1111 1111 1110	000000~111111	0x3F FF80~0x3F FFBF
第 65535 页	1111 1111 1111 1111	000000~111111	0x3F FFC0~0x3F FFFF

(4) 堆栈指针 SP

堆栈指针 SP 是 16 位的寄存器, 最大可寻址空间 $2^{16}=64\text{K}$, 因此, 可以对片内数据存储空间的低 64K 字进行寻址。正常情况下, 堆栈的寻址方向, 从低地址向高地址方向增长, SP 永远指向下一个要存取的字, 即 SP 中存放的地址永远是下一步要操作的存储单元的地址。片内数据存储空间, 以字 (16 位) 为基本存储单元, 1 个 32 位数据占用 2 个存储单元, 所以, 使用 SP 指针进行 32 位访问时, 一般从偶地址开始, 倘若要存放 32 位数据, 要先存放低 16 位, 再存放高 16 位。

复位时, SP 指向 0x0400 存储单元。SP 的地址在 0x0000~0xFFFF 自动增/减变化。当 SP 值增加超过 0xFFFF 时, SP 会自动复位到 0x0000, 并向高地址循环; 当 SP 值小于 0x0000 时, 会自动从 0xFFFF 向低地址循环。

(5) 状态寄存器 ST0、ST1

DSP 有 2 个 16 位的状态寄存器 ST0 和 ST1。其中, ST0 寄存器的结构如下, 位描述如表 2.2 所示。

15	10	9	7	6	5	4	3	2	1	0
OVC/OVCU	PM	V	N	Z	C	TC	OVM	SXM		
R/W-000000	R/W-000	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

注: “R/W” 分别表示该位可读/可写, “-” 表示 DSP 复位后的值, 全书一致。

表 2.2 状态寄存器 ST0 的位描述

位	名称	说明
15~10	OVC/OVCU	溢出计数器。有符号运算时为 OVC, 保存 ACC 溢出信息; 无符号运算时为 OVCU, 加法运算时有进位则加; 减法运算时有借位则减
9~7	PM	乘积移位模式, 决定乘积在输出前如何移位
6	V	溢出标志, 反映操作结果是否引起保存结果的寄存器的溢出
5	N	负标志, 反映操作结果是否为负值
4	Z	0 标志, 反映操作结果是否为 0 值
3	C	进位标志, 反映加法操作结果是否产生进位, 或减法操作结果是否产生借位
2	TC	测试/控制位, 反映了位测试或归一化指令的测试结果
1	OVM	溢出模式位, 规定对 ACC 溢出结果是否进行调整
0	SXM	符号扩展位, 决定输入移位器对数据移位时, 是否需要进行符号扩展

ST1 寄存器的结构如下，位描述如表 2.3 所示。

15		13		12		11		10		9		8			
ARP				保留		M0M1MAP		保留		OBJMODE		AMODE			
R/W-000				R-0		R-1		R-0		R/W-0		R/W-0			
7		6		5		4		3		2		1		0	
IDLESTAT		EALLOW		LOOP		SPA		VMAP		PAGE0		DBGM		INTM	
R-0		R-0		R/W-0		R/W-0		R/W-1		R/W-0		R/W-1		R/W-1	

表 2.3 状态寄存器 ST1 的位描述

位	位域	说明
15~13	ARP	辅助寄存器指针，指示当前时刻工作的辅助寄存器 XAR0~XAR7
11	M0M1MAP	M0、M1 映射位。0-C27x 兼容模式；1-C28x 模式，仅供 TI 测试使用
9	OBJMODE	目标兼容模式位。0-C27x 模式；1-C28x 模式
8	AMODE	寻址模式位。0-C28x 模式；1-C2xLP 模式
7	IDLESTAT	空闲状态位（只读），执行 IDLE 指令时置位
6	EALLOW	受保护寄存器访问允许位。0-不允许；1-允许
5	LOOP	循环指令状态位。0-没有执行循环指令；1-CPU 执行循环指令
4	SPA	队列指针定位位。0-SP 指针没有定位到偶地址；1-SP 指针定位到偶地址
3	VMAP	CPU 中断向量表映射位。0-映射到最低地址；1-映射到最高地址
2	PAGE0	寻址模式设置位。0-堆栈寻址；1-直接寻址
1	DBGM	调试功能屏蔽位，置 1 时仿真器不能实时访问寄存器和存储器
0	INTM	中断屏蔽位，0-允许可屏蔽中断；1-禁止所有可屏蔽中断

INTM 位是全局可屏蔽中断总控制位，位于 C28x 核心级中断逻辑部分，具体参阅 4.2.2。

2.4 FPU 及其寄存器组

2.4.1 FPU 协处理器

FPU 是 C28x 定点 CPU 的协处理器，二者之间可以进行数据交换。在定点中央处理器 CPU 的基础上，FPU 增加了支持 IEEE 单精度浮点操作的寄存器组和指令。FPU 的寄存器组，包括 8 个浮点结果寄存器 R0H~R7H、浮点状态寄存器 STF 和块重复寄存器 RB，均为 32 位。其中，RnH (n=0~7，本节下同) 和 STF，均有映射寄存器，可以在处理高级中断操作时，对上述浮点运算寄存器中的数据，进行快速保护和恢复。FPU 新增加的块重复指令是 RPTB，该指令允许重复执行一块代码，协处理器 FPU 通过块重复寄存器 RB，辅助实现块重复操作。

在进行定点与浮点数据转换时，注意时延操作，若用汇编程序实现浮点预算，需要在其后插入 NOP 空操作指令，若用 C/C++ 语言编程处理，则编译系统自动处理时延问题。

2.4.2 FPU 寄存器组

(1) 浮点状态寄存器 STF

浮点状态寄存器 STF 反映了浮点运算的结果，其结构如下，位域描述如表 2.4 所示。

31	30	10					9	8
SHDWS	保留					RND32	保留	
R/W-0	R-0					R/W-0	R-0	
7	6	5	4	3	2	1	0	
保留	TF	ZI	NI	ZF	NF	LUF	LVF	
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	

表 2.4 浮点状态寄存器 STF 位描述

位	位域	说明
31	SHDWS	映射模式状态位。SAVE 指令清零，RESTORE 指令置位，装载 STF 不影响该位
9	RND32	浮点运算取整模式位。0-向 0 取整；1-向最近的整数取整
6	TF	测试标志位，反映测试指令条件。0-条件为假；1-条件为真
5	ZI	0 整数标志位。0-整数非 0；1-整数为 0
4	NI	负整数标志位。0-整数非负；1-整数为负
3	ZF	浮点 0 标志位。0-浮点非 0；1-浮点为 0
2	NF	浮点负标志位。0-浮点非负；1-浮点为负
1	LUF	浮点下溢条件缓存标志。0-下溢条件未被缓存；1-下溢条件被缓存
0	LVF	浮点上溢条件缓存标志。0-上溢条件未被缓存；1-上溢条件被缓存

(2) 块重复寄存器 RB

RB 寄存器的结构如下，位描述如表 2.5 所示。

31	30	29	23	22	16	15	0
RAS	RA	RSIZE			RE	RC	
R-0	R-0	R-0			R-0	R-0	

表 2.5 块重复寄存器 RB 位域描述

位	位域	说明
31	RAS	块重复激活缓冲位。中断发生时，RA 内容复制到 RAS
30	RA	重复块激活位。执行 RPTB 指令置 1；块重复操作结束清 0。中断发生时将 RA 内容复制到 RAS，清零 RA；中断返回时 RAS 内容复制到 RA，清零 RAS
29-23	RSIZE	重复块大小
22-16	RE	重复块结束地址。执行 RPTB 指令时，RE=(PC+1+RSIZE) 的低 7 位
15-0	RC	重复计数器。块重复次数=(RC+1) 次

2.5 存储器与存储空间

2.5.1 概述

F2833x 采用改进的哈佛结构，存储空间划分为程序空间和数据空间，各自拥有独立的地址总线 and 数据总线。其中数据的读/写地址总线相互独立，程序与数据的读总线也相互独立，多总线运行机制，显著提升了数据的处理速度。同时，程序空间与数据空间本身又是重合的，包括 I/O 端口、DSP 引脚、各种片上外设的寄存器和片外存储器等，采取了统一编址模式。改进的哈佛结构，既保留了冯·诺依曼计算机体系结构的优点，增强了兼容性，又便于嵌入式操作系统运行。

TMS320F28335/28235 片内存储器映射，如图 2.3 所示。其程序地址总线为 22 位，故可寻址最大 4MW 程序存储空间，对应地址 0x000000~0x3FFFFFF；数据地址总线为 32 位，故可寻址最大 4GW 数据存储空间，由于 DSP 片内存储空间有限，因此，片内只有 4MW 数据存储空间。同时，可以通过外部地址总线 XA0~XA19 和外部数据总线 XD0~XD31 扩展片外存储空间，增加存储容量。

2.5.2 存储空间映射与存储空间配置

由图 2.3 可见，DSP 片内存储空间配置了多种类型的存储器，并对应不同的映射地址：FLASH (256KW)、SARAM (34KW)、OTP (1KW)、BOOT ROM (8KW)、外设帧存储器 PF0~PF3、外部存储器接口 XINTF Zone0、XINTF Zone6、XINTF Zone7 等。

起始地址	数据空间	程序空间	(接左图)	数据空间	程序空间
0x00 0000	M0 向量-RAM(32x32) (VMAP=0)		0x10 0000	XINTF Zone6 (4Kx16,XZCS0) (受保护) (DMA访问)	
0x00 0040	M0 SARAM(1Kx16)		0x20 0000	XINTF Zone7 (4Kx16,XZCS0)	
0x00 0400	M1 SARAM(1Kx16)		0x30 0000	FLASH (256Kx16, 安全区)	
0x00 0800	Peripheral Frame 0	保留	0x33 FFF8	128位保护密码	
0x00 0D00	PIE 向量-RAM (256x16)		0x34 0000	保留	
0x00 0E00	Peripheral Frame 0				
0x00 2000	保留				
0x00 4000	XINTF Zone 0(4Kx16,XZCS0) (受保护) DMA访问		0x38 0080	ADC 校准数据	
0x00 5000	Peripheral Frame 3 (受保护)	保留	0x38 0090	保留	
0x00 6000	Peripheral Frame 1 (受保护)		0x38 0400	User OTP (1Kx16, 安全区)	
0x00 7000	Peripheral Frame 2 (受保护)		0x38 0800	保留	
0x00 8000	L0 SARAM (4Kx16, 安全区, 双映射)		0x3F 8000	L0 SARAM (4Kx16, 安全区, 双映射)	
0x00 9000	L1 SARAM (4Kx16, 安全区, 双映射)		0x3F 9000	L1 SARAM (4Kx16, 安全区, 双映射)	
0x00 A000	L2 SARAM (4Kx16, 安全区, 双映射)		0x3F A000	L2 SARAM (4Kx16, 安全区, 双映射)	
0x00 B000	L3 SARAM (4Kx16, 安全区, 双映射)		0x3F B000	L3 SARAM (4Kx16, 安全区, 双映射)	
0x00 C000	L4 SARAM (4Kx16, DMA访问)		0x3F C000	保留	
0x00 D000	L5 SARAM (4Kx16, DMA访问)		0x3F E000	Boot ROM(8Kx16)	
0x00 E000	L6 SARAM (4Kx16, DMA访问)		0x3F FFC0	BROM 向量-ROM(32x32)	
0x00 F000	L7 SARAM (4Kx16, DMA访问)				
0x01 0000	保留				

图 2.3 TMS320F28335/28235 存储器映射

1. FLASH 与代码安全模块

FLASH 是可重复电擦除的存储空间，一般应用于程序调试，编译后的可执行程序代码，通过 JTAG 端口下载，并存储于 FLASH 之中。FLASH 同时被映射到程序存储空间和数据存储空间，既可以存放程序代码，也可以存储掉电后需要保护的重要数据。F28335 的 FLASH 共有 256KW 存储空间，片内具体地址区间为 0x300000~0x33FFFF，被细分为 8 个 32KW 的存储扇区 A~H，每个存储扇区均独立执行擦写操作，如表 2.6 所示。

表 2.6 F28335/F28235 闪存 FLASH 扇区划分与地址映射

扇区序号	地址范围	程序和数据空间
1	0x30 0000-0x30 7FFF	扇区 H (32K×16)
2	0x30 8000-0x30 FFFF	扇区 G (32K×16)
3	0x31 0000-0x31 7FFF	扇区 F (32K×16)
4	0x31 8000-0x31 FFFF	扇区 E (32K×16)
5	0x32 0000-0x32 7FFF	扇区 D (32K×16)
6	0x32 8000-0x32 FFFF	扇区 C (32K×16)
7	0x33 0000-0x33 7FFF	扇区 B (32K×16)
8	0x33 8000-0x33 FF7F	扇区 A (32K×16) 当使用 代码安全模块时，编程至 0x0000 引导至闪存进入点 (程序分支指令所在的位置) 安全密码 (128 位) (不要设定为零)
	0x33 FF80-0x33 FFF5	
	0x33 FFF6-0x33 FFF7	
	0x33 FFF8-0x33 FFFF	

注：①当代码安全密码被编辑时，0x33FF80~0x33FFFF 之间的所有地址不能被用作存储程序代码或数据，这些位置必须被设定为 0x0000。②如果代码安全特性未被使用，地址 0x33FF80~0x33FFFF 空间可以被用于存储代码或数据；③地址 0x33FF0~0x33FF5 共 6 个字的存储空间，可以存储数据，但是不能存储程序代码。

其中，FLASH 的最高 8 个字 0x33FFF8~0x33FFFF 共 128 位，为代码安全模块 CSM，可以设置 128 位的程序保护密码，以保护知识产权。

2. SARAM

F28335 拥有 34K×16 的 SARAM，即单周期访问 RAM，每个机器周期仅能访问一次。其中，M0、M1 大小均为 1KW；L0~L7 大小均为 4KW。M0 块位于 4M 存储空间的最低地址端，复位时，堆栈指针 SP 指向 M1 块起始地址 0x000400。上述存储块同时被映射到数据存储空间和程序存储空间，每个存储空间均可独立访问，以降低流水线拥堵。

其中 L0~L3 SARAM 是双映射，同时映射到 0x008000~0x00BFFF 和 0x3F8000~0x3FBFFF，L0~L3 内容受代码安全模块 CSM 保护；L4~L7 是单映射，映射范围为 0x00C000~0x00FFFF，可以直接进行 DMA 访问。

3. ADC 校准数据存储与用户 OTP

(1) ADC 校准数据存储，地址范围 0x380080~0x38008F，双映射存储区，厂家保留空间，用于 ADC 校准和系统测试。

(2) 用户 OTP

地址范围 0x380400~0x3807FF，大小 1KW，双映射存储区，可以存储用户需要保护的

非遗失性数据或程序代码，并受 CSM 密码保护，为一次性编程使用存储器，若无特殊需要，用户不要使用该区域。

4. BOOT ROM

BOOT ROM 又称引导 ROM，出厂时由厂家使用并固化了引导加载程序进行设定等信息，存储空间大小为 8KW，同时映射到数据与程序存储空间。其中，4 个 GPIO 端口 GPIO84~GPIO87 的电平状态决定引导模式，如表 2.7 所示。根据引导模式设置，引导加载软件在加电时自动选择具体的引导模式，包括正常引导模式、从外部连接下载新软件引导模式、在内部闪存/ROM 中编辑的引导软件等；同时，引导 ROM 还包含有 IQ 数学表、FPU 数学表、产品版本信息、CPU 中断向量表等。

表 2.7 引导模式设置与说明

引导模式	GPIO87/XA15	GPIO86/XA14	GPIO85/XA13	GPIO84/XA12	说明
F	1	1	1	1	跳转到闪存
E	1	1	1	0	SCI-A boot
D	1	1	0	1	SPI-A 引导
C	1	1	0	0	PC-A 引导
B	1	0	1	1	eCAN-A 引导
A	1	0	1	0	McBSP-A 引导
9	1	0	0	1	跳转到 XINTF×16
8	1	0	0	0	跳转到 XINTF×32
7	0	1	1	1	跳转到 OTP
6	0	1	1	0	并行 GPIO I/O 引导
5	0	1	0	1	并行 XINTF 引导
4	0	1	0	0	跳转至 SARAM
3	0	0	1	1	分支到检查引导模式
2	0	0	1	0	跳转到闪存，跳过 ADC 校准
1	0	0	0	1	跳转至 SARAM，跳过 ADC 校准
0	0	0	0	0	跳转至 SCI，跳过 ADC 校准

注：4 个 GPIO 端口 GPIO84~GPIO87 内部使用上拉电阻。

表 2.8 BOOT ROM 功能划分及空间映射

空间映射地址	BOOT ROM 存储空间	
	数据存储空间	程序存储空间
0x3FE000~0x3FEBDB	IQ 数学表	
0x3FEBDC~0x3FF27B	FPU 数学表	
0x3FF27C~0x3FF34B	保留	
0x3FF34C~0x3FF9ED	BOOT 下载功能区	

续表

空间映射地址	BOOT ROM 存储空间	
	数据存储空间	程序存储空间
0x3FF9EE~0x3FFFB8	保留	
0x3FFFB9~0x3FFFBF	版本、检测信息区	
0x3FFFC0	CPU 中断向量表（上电复位向量）	
0x3FFFC1~0x3FFFFF	CPU 中断向量表	

BOOT ROM 功能划分及空间映射如表 2.8 所示。其中，地址区间 0x3FFFC0~0x3FFFFF 是 CPU 中断向量表，大小 64W，存放着 32 个 CPU 软/硬件中断向量，每个中断向量占用 2 个字长的存储区域。其中，在中断向量表的首地址 0x3FFFC0 处，存放着 DSP 上电复位向量，该向量在出厂时被烧录为指向上电引导程序（BootLoader）。DSP 上电后自动读取该向量，将程序引导至 BootLoader 入口。根据上述引导模式设置，完成用户程序的加载和引导。有关中断向量知识，参阅 PIE 中断部分。

5. 外设寄存器帧 PF_n 与保护寄存器 EALLOW

(1) 外设寄存器帧 PF_n

外设寄存器帧 PF_n (n=0~3, 本节下同) 仅映射到数据存储区。除了个别寄存器之外，DSP 所有的片上外设寄存器、片上外设中断向量表、FLASH 存储器的控制、数据与状态寄存器等，都映射在外设寄存器帧 PF_n。PF0 主要包括 PIE、FLASH、XINTF、DMA、CPU 定时器、ADC（双映射）、CSM 等模块的寄存器；PF1 主要包括 GPIO、eCAN、ePWM、eCAP、eQEP 等外设模块的寄存器；PF2 主要包括系统控制寄存器、外部中断寄存器，以及 SCI、SPI、ADC（双映射）、I²C 等外设模块的寄存器；PF3 主要包括 McBSP 模块寄存器。其中，PF0 不受 EALLOW（Edit Allow）指令保护，而 PF1、PF2、PF3 则受 EALLOW 指令保护，PF3 可以进行 DMA 访问，如表 2.9 所示。

表 2.9 PF_n 及其对应寄存器

名称	PF3	PF2	PF1	PF0
地址	0x005000-0x005FFF	0x007000-0x007FFF	0x006000-0x006FFF	0x000800-0x000CFF 0x000E00-0x0001FFF
大小	4KW	4KW	4KW	7.25KW
外设 或功 能单 元寄 存器	McBSP-A 寄存器	系统控制寄存器	eCAN-A 寄存器	器件仿真寄存器
	McBSP-B 寄存器	SPI-A 寄存器	eCAN-B 寄存器	闪存寄存器
	ePWM _n +RPWM1	SCI-A 寄存器	ePWM _n +HRPWM _n 寄存器	代码安全模块寄存器
	ePWM _n +RPWM2	SCI-B 寄存器	eCAP _n 寄存器	ADC 寄存器（双映射）
	ePWM _n +RPWM3	SCI-C 寄存器	eQEP1 寄存器	XINTF 寄存器
	ePWM _n +RPWM4	外部中断寄存器	eQEP2 寄存器	CPU 定时器 T0/T1/T2 寄存器
	ePWM _n +RPWM5	ADC 寄存器	GPIO 寄存器	PIE 寄存器
	ePWM _n +RPWM6	I ² C-A 寄存器		PIE 矢量表
			DMA 寄存器	

注：①n=1~6, n 是指表中部分寄存器的下标；②映射到 PF3 的寄存器，可以直接 DMA 访问。

(2) 保护寄存器 EALLOW

由于 PF1~PF3 受 EALLOW 保护，所以对上述受保护的寄存器进行访问、更改设置之前，要引用编辑允许指令 EALLOW (Edit Allow) 开锁；操作完毕之后，要引用编辑禁止指令 EDIS (Edit Disable) 闭锁，对重要寄存器重新上锁保护。

注意：汇编指令 EALLOW 与 EDIS 要成对使用。上述指令实质是对状态寄存器 ST1 中的位域 ST1[EALLOW]进行置 1 (允许) 或置 0 (不允许) 操作。

6. 外部存储器接口 XINTF

DSP 可以通过 XINTF 单元连接外部程序或数据存储器，以扩大 DSP 芯片的存储空间。XINTF 属于单总线结构，可以与外部单总线结构的设备直接连接，增强对外部设备的兼容性。

XINTF 包括 20 位的外部地址总线 XA0~XA19、32 位的数据总线 XD0~XD31 和三条片选控制信号线 $\overline{\text{XZCS0}}$ 、 $\overline{\text{XZCS6}}$ 、 $\overline{\text{XZCS7}}$ 。三条片选信号线均低电平有效，分别对应三个不同的存储区域：区域 0 (XINTF Zone0)，寻址地址范围 0x004000~0x004FFF，大小 4K 空间；区域 6 (XINTF Zone6)，寻址地址范围 0x100000~0x1FFFFF，大小 1MW 空间；区域 7 (XINTF Zone7)，寻址地址范围 0x200000~0x2FFFFFF，大小 1MW 空间。通过 XINTF 便可以将 DSP 外部设备或器件中对应的扩展存储空间地址，与 DSP 片内对应区域的存储空间之间，建立起一一对应的映射关系。

譬如，若令 F28335 第 145 号引脚接地， $\overline{\text{XZCS0}}=0$ ，片选信号有效时，则 DSP 外部存储空间的首地址，便指向片内存储器的区域 0 (XINTF Zone0) 的起始地址 0x004000；同理，若令片选信号 $\overline{\text{XZCS6}}=0$ ，则外部的存储空间的起始地址便指向片内存储器的区域 6 (XINTF Zone6) 的起始地址 0x100000。通过外部存储器接口 XINTF，便可以像存储、读取片内存储单元的信息一样，对片外设备存储单元，直接进行存储或读取操作。

2.5.3 外部存储器接口 XINTF 应用实例

以模/数转换器 ADS8364 为例，阐述外部存储器接口 XINTF 的原理及其应用。

1. 并行模/数转换器 ADS8364

ADS8364 是 TI 公司设计生产的一款高速、低功耗的 16 位并行模/数转换器，具有较大的共模抑制比，适合于高噪声环境下对多路模拟信号进行并行高速采样，其引脚分布如图 2.4 所示。

ADS8364 拥有 6 个独立的模/数转换器模块，分别对应 6 路差分模拟输入通道 CH_A0、CH_A1、CH_B0、CH_B1、CH_C0、CH_C1，划分为 A 组、B 组和 C 组分别进行管理。对应 6 个 ADC 子模块和 6 路差分模拟输入通道，ADS8364 内部有 6 个对应的转换结果输出寄存器，包括 CHA0、CHA1、CHB0、CHB1、CHC0 和 CHC1。另外，ADS8364 外部的地址线 A0~A2；数据线 D0~D15；采样/保持信号 $\overline{\text{HOLDA}}$ 、 $\overline{\text{HOLDB}}$ 和 $\overline{\text{HOLDC}}$ ；以及外部时钟信号 CLK、片选信号 $\overline{\text{CS}}$ 、读使能信号 $\overline{\text{RD}}$ 和转换结束信号 EOC。

2. 外扩 ADC 电路设计

基于 ADS8364 的 DSP 信号采集系统电路原理图，如图 2.5 所示。

通过地址线 A0~A2 进行地址译码，配合相应控制信号，实现对转换结果寄存器 CHA0~CHC1 中的数据进行读取。地址译码与模拟输入通道、输出寄存器之间的对应关

系，如表 2.10 所示。

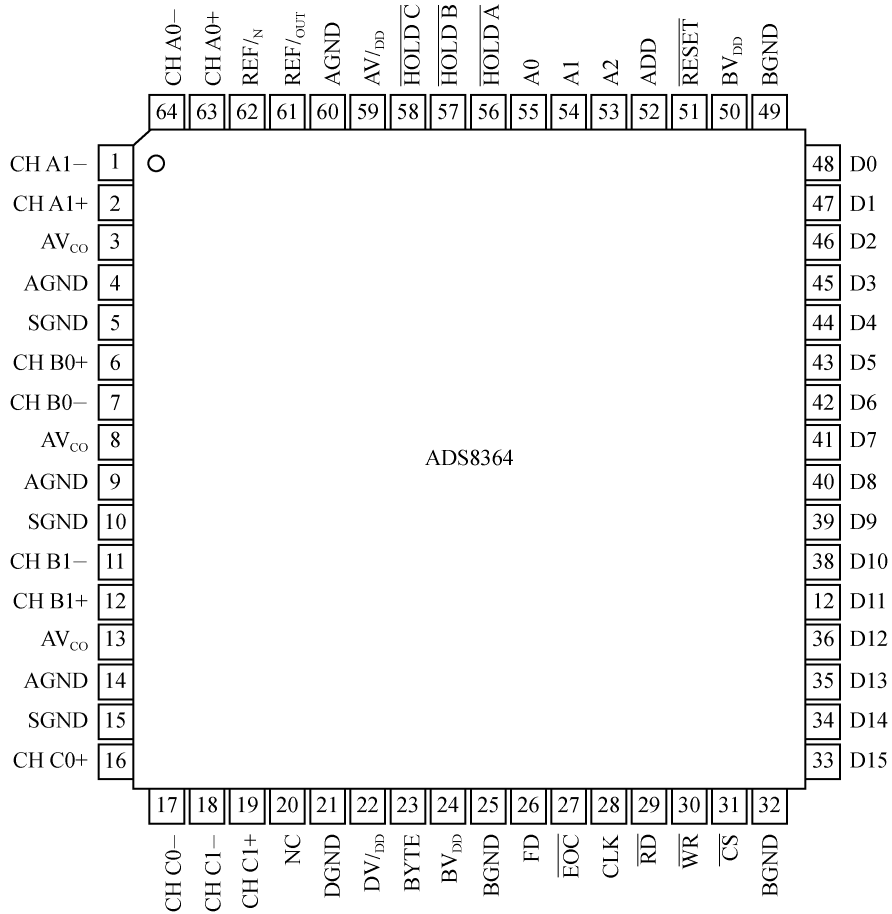


图 2.4 ADS8364 引脚分布图

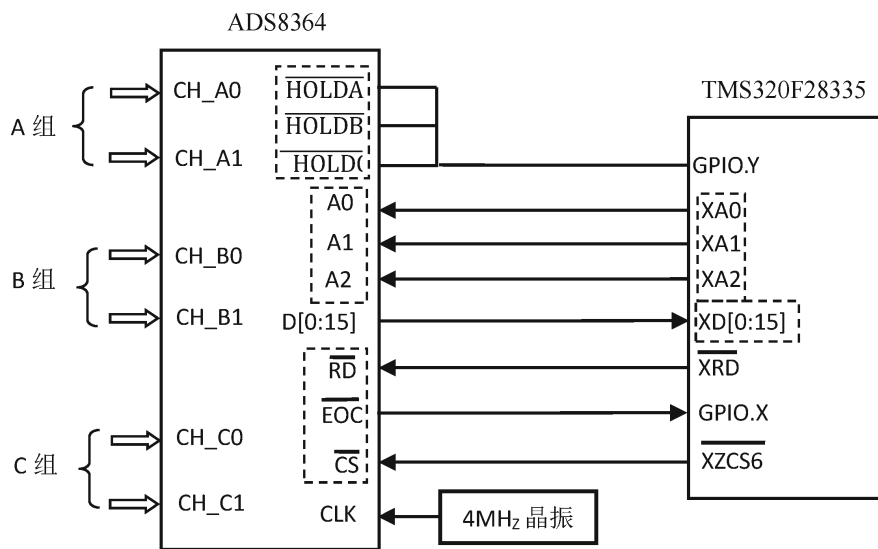


图 2.5 DSP 与 ADS8364 电路连接原理图

表 2.10 地址译码与模拟输入通道/输出寄存器对应关系表

地址译码			模拟输入通道	数据读取	
A2	A1	A0		读取模式	输出寄存器
0	0	0	CH_A0	直接地址读取	CHA0
0	0	1	CH_A1		CHA1
0	1	0	CH_B0		CHB0
0	1	1	CH_B1		CHB1
1	0	0	CH_C0		CHC0
1	0	1	CH_C1		CHC1
1	1	0	所有通道	循环读取	CHA0~CHC1
1	1	1	所有通道	FIFO 读取	CHA0~CHC1

3. 工作原理

如图 2.5 所示，F28335 与 ADS8364 之间通过地址线、数据线，以及控制信号线相连接。

通过采样/保持信号 $\overline{\text{HOLDA}}$ 、 $\overline{\text{HOLDB}}$ 和 $\overline{\text{HOLDC}}$ ，可以分别启动 ADS8364 片内对应的 ADC 子模块工作（A 组、B 组或 C 组）；DSP 外部 20 根地址总线中的 $\text{XA0} \sim \text{XA2}$ ，分别与 ADS8364 的地址线 $\text{A0} \sim \text{A2}$ 对应连接，通过地址译码，可以选择相应的转换结果输出寄存器。

在本项目中，选择 XINTF 单元的 $\overline{\text{XZCS6}}$ 端口作为 ADS8364 的片选控制信号，当使能 $\overline{\text{XZCS6}}$ 为低电平时，便激活了片内 XINTF Zone6 存储区，其对应寻址地址范围 $0\text{x}100000 \sim 0\text{x}01\text{FFFFFF}$ ，大小 1MW 存储空间。根据 XINTF 工作原理，ADS8364 内部 6 个转换结果输出寄存器，在片内存储器的区域 6（XINTF Zone6）空间的映射地址依次为 $0\text{x}100000 \sim 0\text{x}100005$ 。由于 ADS8364 芯片只有 6 个转换结果输出寄存器，因此地址线 $\text{A0} \sim \text{A2}$ 已经足够对其进行片选，且仅占用 DSP 的三根地址线 $\text{XA0} \sim \text{XA2}$ 。另外， $\overline{\text{XZCS0}}$ 、 $\overline{\text{XZCS6}}$ 、 $\overline{\text{XZCS7}}$ 又分别是 DSP 片内存储器区域 0（XINTF Zone0）、区域 6（XINTF Zone6）和区域 7（XINTF Zone7）的片选信号。ADS8364 地址线、输出寄存器与 DSP 片内映射地址之间的关系，如表 2.11 所示。

表 2.11 ADS8364 地址译码、输出寄存器与 F28335 空间映射关系

ADS8364 地址译码			ADS8364 模拟输入通道	ADS8364 输出寄存器	片内寄存器区域 0/6/7 地址映射		
A2	A1	A0			$\overline{\text{XZCS0}} = 0$ XINTF Zone0	$\overline{\text{XZCS6}} = 0$ XINTF Zone6	$\overline{\text{XZCS7}} = 0$ XINTF Zone7
0	0	0	CH_A0	CHA0	0x004000	0x100000	0x200000
0	0	1	CH_A1	CHA1	0x004001	0x100001	0x200001
0	1	0	CH_B0	CHB0	0x004002	0x100002	0x200002
0	1	1	CH_B1	CHB1	0x004003	0x100003	0x200003
1	0	0	CH_C0	CHC0	0x004004	0x100004	0x200004
1	0	1	CH_C1	CHC1	0x004005	0x100005	0x200005
1	1	0	—	—	—	—	—
1	1	1	—	—	—	—	—

利用外部 ADC 集成芯片 ADS8364 进行多路采样时, 通过 XINTF 接口, 可以很方便地在片内映射地址中, 直接读取 ADS8364 输出存储器中的转换数据。

2.6 时钟系统

2.6.1 时钟系统概述

1. 系统结构

时钟系统是 DSP 重要组成部分, 是保障系统正常运行的神经网络, 主要由时钟振荡器、锁相环 PLL、看门狗 WD、高/低速外设定标器、系统时钟控制寄存器, 以及若干控制和状态寄存器等组成。F2833x 时钟系统结构框图, 如图 2.6 所示。

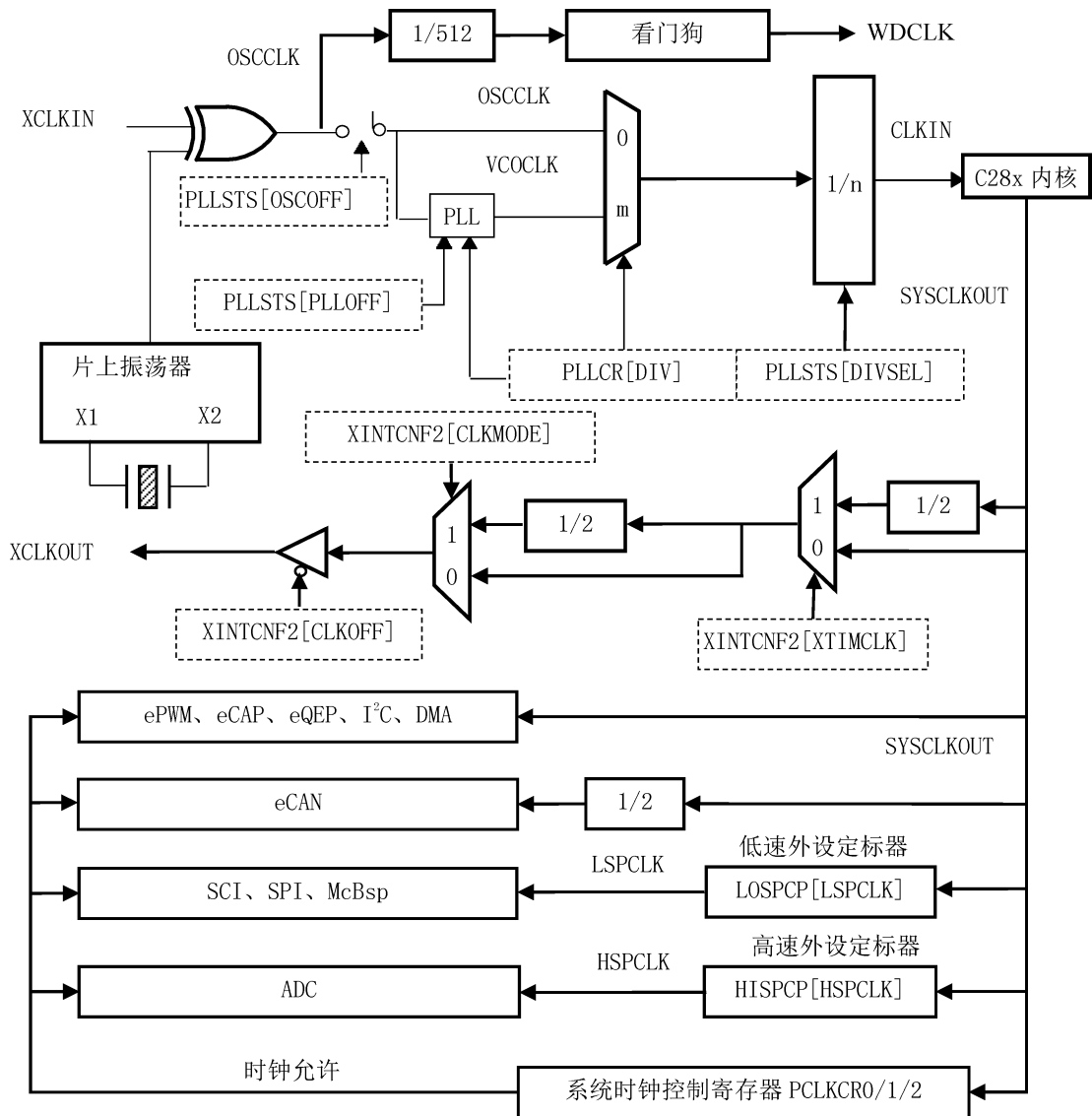


图 2.6 TMS320F2833x 时钟系统原理框图

2. 外部时钟 OSCCLK

时钟源位于时钟系统的前端，产生外部时钟 OSCCLK。目前，有 2 种时钟源模式。其一，通过外部时钟输入引脚 XCLKIN，输入 1 路幅值为 3.3V 或 1.9V 的标准时钟信号，作为 DSP 时钟系统的基础时钟；其二，利用 DSP 时钟引脚 X1 和 X2，通过外接石英晶体和 2 个高频电容，与片内振荡电路一起，组成一个时钟振荡器（Oscillator）系统，从而产生振荡信号，即基础时钟。具体实现电路如图 2.7 所示，其中图（b）和图（c）为从外部引进标准时钟信号模式；图（a）则是搭建时钟振荡器产生 OSCCLK 模式，DSP 应用系统多选择该模式产生基础时钟信号，其中外接晶振参数为 30MHz，高频电容参数为 24pF。

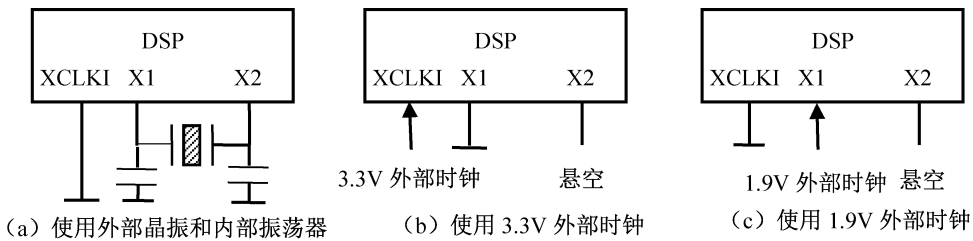


图 2.7 时钟源方案

3. 工作原理

来自时钟源的外部时钟 OSCCLK，首先分为 2 路时钟信号，其中 1 路时钟信号经过 512 分频后，送入看门狗单元，通过看门狗单元可以对外输出 WDCLK 时钟；另 1 路时钟信号，由 PLL 状态寄存器的 PLLSTS[OSCOFF]位域控制，当 PLLSTS[OSCOFF]=0 时通道闭合，OSCCLK 时钟进入锁相环系统，此时，OSCCLK 时钟可以直接通过锁相环系统，进入后续分频环节，也可以经过 PLL 单元进行倍频，其中 PLL 单元由 PLLSTS [PLLOFF]位域使能，倍频系数则由 PLLCR[DIV]控制。当 PLLSTS[PLLOFF]=0 时，使能锁相环 PLL，OSCCLK 信号在 PLL 中，经过锁相和倍频后，成为相位和频率稳定的高频时钟信号。OSCCLK 时钟可以选择直通或经过 PLL 锁相环进行锁相和倍两种方式，进入 1/n 分频环节，分频系数则由 PLLSTS[DIVSEL]位域设置，通过倍频和分频环节产生符合 CPU 内核要求的时钟信号 CLKIN，CLKIN 时钟信号进入 C28x 内核。最后从 C28x 内核输出高质量的高频时钟信号 SYSCLKOUT，供系统使用，SYSCLKOUT 时钟又称为系统时钟。

SYSCLKOUT 经过一系列倍频、分频或直通处理，被送到 DSP 各个功能模块。SYSCLKOUT 直接输入到 ePWM、eCAP、eQEP、I²C、DMA 等模块；经过 1/2 分频，被送到到 eCAN 模块；经过高速外设定标器 HISPCP 分频处理后，成为 HSPCLK 时钟，送至 ADC 模块；经过低速外设定标器 LOSPCP 分频处理后，送至 SCI、SPI 和 McBSP 模块；另外，SYSCLKOUT 可以直接旁路输出或经过 1 次 1/2 分频或 2 次 1/2 分频后，通过 DSP 引脚 XCLKOUT 输出，供片外设备作为时钟源使用；具体分频系数，由控制寄存器 XINTCNF2 的 XINTCNF2[XTIMCLK]和 XINTCNF2[CLKMODE]位域设置。

2.6.2 时钟系统子模块

1. 寄存器

时钟系统中各子模块的寄存器，如表 2.12 所示。

表 2.12 时钟系统寄存器

序号	寄存器名称	大小 ×16	功能描述	序号	寄存器名称	大小 ×16	功能描述
1	PLLSTS	1	PLL 状态寄存器	7	HISPCP	1	高速外设定标寄存器
2	PLLCR	1	PLL 控制寄存器	8	LOSPCP	1	低速外设定标寄存器
3	PCLKCR0	1	外设时钟控制寄存器 0	10	SCSR	1	系统控制与状态寄存器
4	PCLKCR1	1	外设时钟控制寄存器 1	11	WDCNTR	1	看门狗计数寄存器
5	PCLKCR2	1	外设时钟控制寄存器 2	12	WDKEY	1	看门狗复位密钥寄存器
6	LPMCR0	1	低功耗模式控制寄存器 0	113	WDCR	1	看门狗控制寄存器

2. 锁相环 PLL

外部时钟 OCSCLK 是相对低速的基础时钟，为了给 C28x 内核以及其他片上外设提供高品质时钟信号，必须经过锁相环 PLL 单元进行相位和频率处理。首先通过锁相环 PLL 可以使时钟信号相对于参考信号保持恒定相位，达到锁相目的，同时，通过 PLL 控制寄存器的 PLLCR[DIV]位域，设置倍频系数，通过状态寄存器 PLLSTS[DIVSEL]位域，设置分频系数，从而调整输出频率。

PLLCR 是 16 位的控制寄存器，通过 PLLCR[DIV]设置倍频系数值，PLLCR 结构如下所示。

15	4	3	0
保留		DIV	
R-0		R/W-0	

PLLSTS 是 16 位的状态寄存器，通过 PLLSTS[DIVSEL]位域设置分频系数值，PLLSTS 结构如下，位域描述如表 2.13 所示。

15						9	8	7
保留						DIVSEL		
R-0						R/W-0		
6	5	4	3	2	1	0		
MCLKOFF	OSCOFF	MCLKCLR	MCLKSTS	PLLOFF	保留	PLLOCKS		
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0		

表 2.13 锁相环状态寄存器 PLLSTS 的位域描述

位	位域	说明
8-7	DIVSEL	分频系数位。00 或 01-4 分频；10-2 分频；11-不分频 ^①
6	MCLKOFF	振荡器时钟丢失检测位。0-使能；1-关闭
5	OSCOFF	锁相环单元输入时钟关闭位。0-打开；1-关闭
4	MCLKCLR	丢失时钟清零复位位。0-无效；1-清零、复位丢失时钟检测电路
3	MCLKSTS	丢失时钟状态指示位。0-无时钟丢失；1-时钟丢失
2	PLLOFF	PLL 单元关闭位。0-使能 PLL；1-关闭 PLL
0	PLLOCKS	PLL 锁相指示位。0-锁相中；1-完成锁相，并进入稳态

注：①只有 PLL 被旁路或关闭时，才能使用该功能；

PLLCCR[DIV]位域完成倍频设置，PLLSTS[DIVSEL]位域完成分频设置，两者配合完成 CLKIN 频率调整。PLL 模块设置与 CLKIN 频率调整，如表 2.14 所示。

表 2.14 PLL 模块设置与 CLKIN 频率

PLLCCR[DIV]	倍频 m=	PLLSTS[DIVSEL]=00、01	PLLSTS[DIVSEL]=2	PLLSTS[DIVSEL]=3
		分频系数 n=4	分频系数 n=2	分频系数 n=1 ^①
0000 (PLL 旁路)	无	CLKIN= OSCCLK/4	CLKIN= OSCCLK/2	CLKIN=OSCCLK
0001	1	CLKIN= (OSCCLK×1) /4	CLKIN= (OSCCLK×1) /2	CLKIN= OSCCLK×1
0010	2	CLKIN= (OSCCLK×2) /4	CLKIN= (OSCCLK×2) /2	CLKIN=OSCCLK×2
0011	3	CLKIN= (OSCCLK×3) /4	CLKIN= (OSCCLK×3) /2	CLKIN=OSCCLK×3
0100	4	CLKIN= (OSCCLK×4) /4	CLKIN= (OSCCLK×4) /2	CLKIN=OSCCLK×4
0101	5	CLKIN= (OSCCLK×5) /4	CLKIN= (OSCCLK×5) /2	CLKIN=OSCCLK×5
0110	6	CLKIN= (OSCCLK×6) /4	CLKIN= (OSCCLK×6) /2	CLKIN=OSCCLK×6
0111	7	CLKIN= (OSCCLK×7) /4	CLKIN= (OSCCLK×7) /2	CLKIN=OSCCLK×7
1000	8	CLKIN= (OSCCLK×8) /4	CLKIN= (OSCCLK×8) /2	CLKIN=OSCCLK×8
1001	9	CLKIN= (OSCCLK×9) /4	CLKIN= (OSCCLK×9) /2	CLKIN=OSCCLK×9
1010	10	CLKIN= OSCCLK×10) /4	CLKIN= (OSCCLK×10) /2	CLKIN=OSCCLK×10
1011~1111	保留	保留	保留	保留

注意：①只有 PLL 被旁路或关闭时，才能使用该功能；②系统上电或复位后，PLL 处于旁路状态；③PLL 模块重新设置前，要先关闭看门狗，待 PLL 系统运行稳定后，再启动看门狗。

其中，PLLCCR[DIV]=0000，则 PLL 旁路，若向 PLLCCR[DIV]写入 10 以内的非零值，即 PLLCCR[DIV]≠0000 时，倍频系数即为 DIV 位域二进制数所对应的十进制数值 m (m ≤ 10) 1~10，频率倍频计算公式为 OSCCLK × m；PLLSTS 为 16 位状态寄存器，其中，PLLSTS [DIVSEL]为分频系数的有效位域，PLLSTS[DIVSEL]=00 或 01，分频系数为 4；PLLSTS [DIVSEL]=10，分频系数为 2；PLLSTS[DIVSEL]=11，若 PLL 被旁路或关闭，则分频系数为 1，否则不启用该功能。计算公式 CLKIN= (OSCCLK × m)/n，其中 m 和 n 分别是倍频与分频系数值。

3. 高/低速外设定标器 HISPCP/LOSPCP

高速外设定标器 HISPCP 和低速外设定标器 LOSPCP，均为 16 位寄存器，低 3 位有效。结构如下，位域描述如表 2.15 所示。

高速外设定标器 HISPCP:

15	3	2	0
保留			HSPCLK
R-0			R/W-001

低速外设定标器 LOSPCP:

15	3	2	0
保留			LSPCLK
R-0			R/W-010

表 2.15 高/低速外设设定标器设置

名称	位域状态及分频系数							
[HSPCLK]或 [LSPCLK]	000	001	010	011	100	101	110	111
位域值	0	1	2	3	4	5	6	7
分频系数 K	1	2	4	6	8	10	12	14

高、低速外设设定标器分频系数设置，如表 2.15 所示。当位域为“000”时，分频系数为 1，当位域为其他状态时，分频系数为其对应的十进制值的 2 倍。系统复位后两者默认分频系数不同，HSPCLK 默认频率为 $f_{\text{SYSCLKOUT}}/2$ ，LSPCLK 默认频率为 $f_{\text{SYSCLKOUT}}/4$ 。

4. 外设时钟控制寄存器

所有片上外设的时钟功能，分别通过 3 个外设控制寄存器 PCLKCR0/PCLKCR1/PCLKCR3 进行设置，每个片上外设在上述控制寄存器中均有对应的一个功能位域，可以独立设置。

位域描述：0-不使能外设时钟；1-使能外设时钟。为了降低功耗，一般将不使用的片上外设时钟关闭。具体应用示例，参见例题 4.1。上述控制寄存器的结构和位域描述如下所示。

(1) 外设时钟控制寄存器 PCLKCR0

15	14	13	12	11	10	9	8
ECANB ENCLK	ECANA ENCLK	MCBSPB ENCLK	MCBSPA ENCLK	SCIBEN CLK	SCIAEN CLK	保留	SPIAEN CLK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
7	6	5	4	3	2	1	0
保留	SCICENCLK	I2CAENCLK	ADCENCLK	TBCLKSYNC	保留		
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	

(2) 外设时钟控制寄存器 PCLKCR1

15	14	13	12	11	10	9	8
EQEP2 ENCLK	EQEP1 ENCLK	ECAP6 ENCLK	ECAP5 ENCLK	ECAP4 ENCLK	ECAP3 ENCLK	ECAP2 ENCLK	ECAP1 ENCLK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
保留	EPWM6 ENCLK	EPWM5 ENCLK	EPWM4 ENCLK	EPWM3 ENCLK	EPWM2 ENCLK	EPWM1 ENCLK	
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

(3) 外设时钟控制寄存器 PCLKCR3

15	14	13	12	11	10
保留	GPIOINENCLK		XINTFENCLK		CPUTIMER2ENCLK
R-0	R/W-1		R/W-0		R/W-1
9	8	7		0	
CPUTIMER1ENCLK	CPUTIMER0ENCLK	保留			
R/W-1	R/W-1	R-0			

5. 看门狗 WD

(1) 概述

看门狗模块 (Watch Dog, WD), WD 实质上是 1 个 8 位的连续增计数模式计数器, 主要功能是监控程序是否正常运行, WD 启动以后, 便对经过前置分频的时钟信号进行周期计数。若程序运行正常, 通过周期性或非周期性地向 WD 计数器写“0x55+0xAA”进行“喂狗”, 使其及时复位, 可以避免 WD 因持续增计数导致计数器溢出而产生中断或使系统复位。相反, 当程序跑飞或进入死循环时, WD 便会因不能及时复位而产生溢出中断信号 $\overline{\text{WDINT}}$ 或直接使系统复位。

(2) 系统结构

WD 结构如图 2.8 所示, 包括 4 部分: ①前置分频与 8 位周期计数器; ②“喂狗”和系统复位; ③软件中断; ④复位/中断信号发生器。

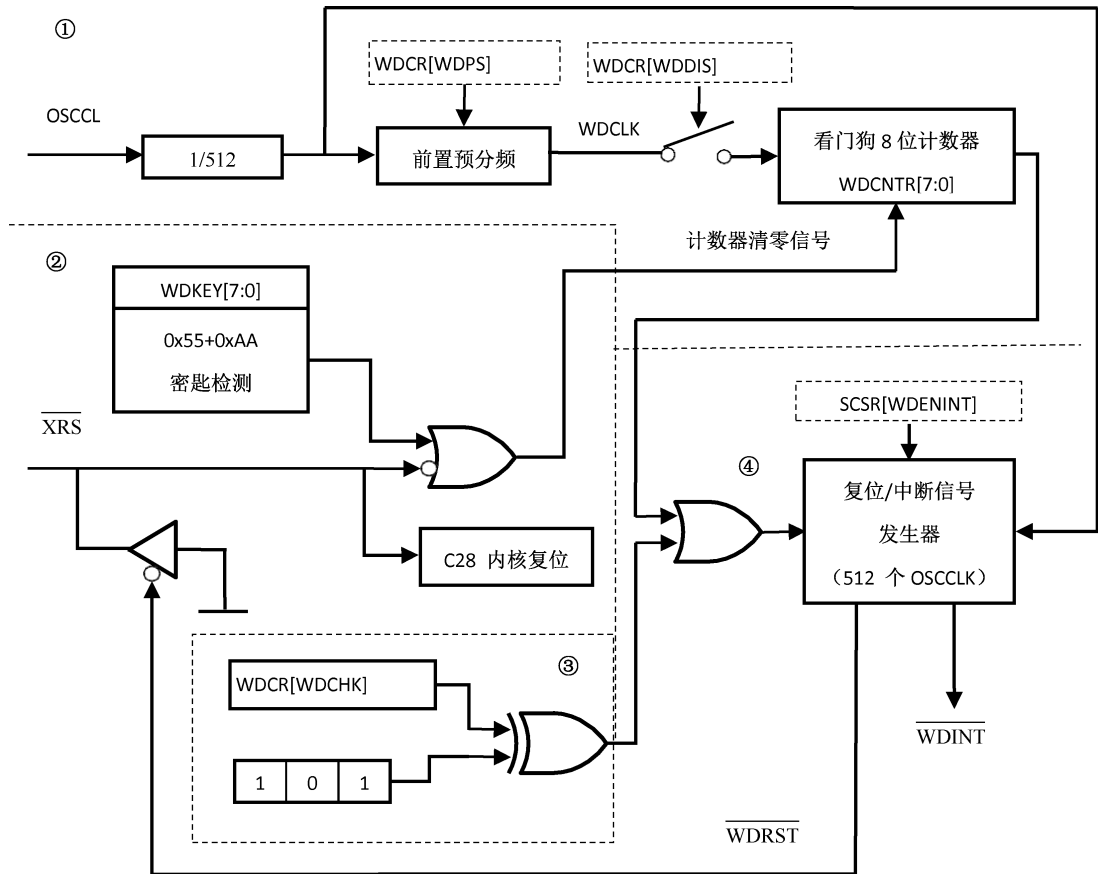


图 2.8 WD 结构图

(3) 看门狗 WD 寄存器

WD 的寄存器包括系统控制与状态寄存器 SCSR、计数寄存器 WDCNTR、密码复位寄存器 WDKEY 和控制寄存器 WDCR。

① 系统控制与状态寄存器 SCSR, 其结构和位域描述如下。

15	保留			8
R-0				
7	3	2	1	0
保留	WDINTS		WDENINT	WDOVERRIDE
R-0	R-1		R/W-0	R/W1C-1

其中，低3位为有效位域，其余为保留位。

WDOVERRIDE: WD 重写位，写0无效，该位是 WDCR[WDDIS]位域是否可以改变的控制位。0-不能改变；1-可以改变。复位信号 RESET 可以置位 WDCR[WDDIS]。

WDENINT: WD 中断允许位：0-禁止中断 $\overline{\text{WDINT}}$ ；1-使能中断 $\overline{\text{WDINT}}$ 。

WDINTS: WD 中断状态位。0- $\overline{\text{WDINT}}$ 有效；1- $\overline{\text{WDINT}}$ 无效。

② 计数寄存器 WDCNTR

WD 计数寄存器 WDCNTR 是 16 位寄存器，低 8 位有效，WDCNTR[WDCNTR]保存着当前 WD 计数器的值，其余位为保留位，结构如下。

15	8	7	0
保留		WDCNTR	
R-0		R-0	

③ 密码复位寄存器 WDKEY

WD 密码复位寄存器 WDKEY 是 16 位寄存器，低 8 位有效，其余位为保留位，结构如下。

15	8	7	0
保留		WDKEY	
R-0		R/W-0	

向 WDKEY[WDKEY]写入“0x55+0xAA”，则清零 WDCNTR，写入其他值无效。

④ 控制寄存器 WDCR

WD 控制寄存器 WDCR 是 16 位寄存器，低 8 位有效，其余位为保留位，结构如下，位域描述如表 2.16 所示。

15	8	7	6	5	3	2	0
保留	WDFLAG	WDDIS	WDCHK		WDPS		
R/W1C-0	R/W-0	R/W-0	R/W-0		R/W-0		

表 2.16 控制寄存器 WDCR 的位域描述

位	位域	说明
7	WDFLAG	复位状态标志位。0-引脚 $\overline{\text{XRS}}$ 或上电引发复位，写 1 可清零；1- $\overline{\text{WDRST}}$ 引发复位
6	WDDIS	WD 禁止位。0-使能 WD 模式；1-禁止 WD 模式
5-3	WDCHK	检测位。写入任何非“101”值，均引发复位或中断事件，可以产生软件复位
2-0	WDPS	前置预分频位。000-直通；001~111-位域值为 n，对应分频系数 2^{n-1} ，详见表 2.17

WDCR[WDPS]是 WDCNTR 前置预分频设置位域，000-不分频（默认值），001~111-若对

应十进制值为 n ，则对应分频系数为 2^{n-1} ，对应计算公式 $WDCLK = OSCCLK/512/2^{n-1}$ 。

表 2.17 前置预分频系数表

WDPS	分频系数	WDCLK	WDPS	分频系数	WDCLK
000	1	OSCCLK/512/1	100	8	OSCCLK/512/8
001	1	OSCCLK/512/1	101	16	OSCCLK/512/16
010	2	OSCCLK/512/2	110	32	OSCCLK/512/32
011	4	OSCCLK/512/4	111	64	OSCCLK/512/64

注：WDCR[WDPS]为前置预分频系数位，OSCCLK 时钟在进行 512 分频之后，进行第二次分频。

(4) WD 工作原理

如图 2.8 所示，外部时钟 OSCCLK 经过 512 分频后，1 路时钟信号送到复位/中断信号发生器；另一路时钟信号则送到前置预分频器，进行二次分频形成 WDCLK，具体分频系数由 WDCR[WDPS]位域决定，参见表 2.17。WD 的核心器件是 8 位计数器 WDCNTR，WDCLK 时钟由 WDCR[WDDIS]位域控制，若 WDCR[WDDIS]=0，则 WDCLK 输入 WDCNTR，WDCNTR 复位受两个信号控制，一是软件复位指令“0x55+0xAA”，二是外部引脚 \overline{XRS} 产生的硬件低电平复位信号。

外部引脚 \overline{XRS} 是硬件产生的低电平复位信号，如图 2.8 所示，单元电路④中 \overline{WDRST} 信号也是低电平复位信号（512 个 OSCCLK 时钟周期的低电平信号），上述低电平信号可以同时引发 C28 内核复位。在单元电路④中，主要设备是复位/中断信号发生器，其触发信号有 2 路来源，1 路是 WD 计数器 WDCNTR 计数溢出产生的信号；另 1 路是 WDCR[WDCHK]非“101”赋值引发的软件中断信号。当上述 2 路信号之一有效时，会触发复位/中断信号发生器，该信号发生器的具体输出信号，则由系统控制与状态寄存器 SCSR[WDENINT]控制，SCSR[WDENINT]=0 时， \overline{WDRST} 有效，SCSR[WDENINT]=1 时， \overline{WDINT} 有效。注意，上述低电平信号的持续时间，要保持 512 个 OSCCLK 时钟的低电平时间。

另外，外部引脚 \overline{XRS} 是双向的。当看门狗 WD 计数溢出时，可以引发复位信号 \overline{WDRST} ，从而触发系统复位，同时，看门狗可以通过 \overline{XRS} 管脚向外输出复位信号，以实现与外围设备的同步复位。

2.6.3 低功耗模式

为了降低能耗，在 DSP 系统空闲时可以令其部分或全部时钟停止工作，进入不同等级的低功耗模式。具体的低功耗模式选择和设置，由低功耗模式控制寄存器 LPMCR0 控制。

1. 控制寄存器 LPMCR0

LPMCR0 是 16 位寄存器，其结构如下，位域描述如表 2.18 所示。

15	14	8	7	2	1	0
WDINTE	保留			QUALSTDBY	LPM	
R/W-0	R-0			R/W-1	R/W-0	

表 2.18 低功耗控制寄存器 LPMCR0 的位域描述

位	位域	说明
15	WDINTE	WD 中断唤醒允许位。0-不允许从 STANDBY 模式唤醒（默认）；1-允许从 STANDBY 模式唤醒（SCSR[WDENINT]=1）
7-2	QUALSTDBY	STANDBY 模式唤醒所需要的 GPIO 信号有效电平保持时间。设 K 为该位域对应的十进制值，则保持时间为 (K+2) 个 OSCCLK 时钟周期，默认值为 2 个 OSCCLK
1-0	LPM	低功耗模式选择位。00-IDLE（默认）；01-STANDBY；1x-HALT

2. 低功耗模式及其特征

低功耗模式由 LPMCR0[LPM]位域决定，共有 3 种低功耗模式：00-IDLE（默认）；01-STANDBY；1x-HALT。具体描述如表 2.19 所示。

表 2.19 系统低功耗模式

LPM =	低功耗 模式	时钟及其状态			退出条件	说明
		OSCCLK	CLKIN	SYSCLKOUT		
00	IDLE	ON	ON	ON	\overline{XRS} 、 \overline{WDINT} 及任何允许中断	
01	STANDBY	ON	OFF	OFF	\overline{XRS} 、 \overline{WDINT} 、GPIOA 端口信号、Debugger	WD 仍然运行
10 11	HALT	OFF	OFF	OFF	\overline{XRS} 、GPIOA 端口信号、Debugger	OSCillator、PLL、WD 关闭

(1) IDLE 低功耗模式

若 LPM=00，执行 IDLE 指令后，系统进入 IDLE 低功耗模式，此时 DSP 处于空转状态，包括 NMI 在内任何被允许的中断，均可以使之退出低功耗模式。

(2) STANDBY 低功耗模式

若 LPM=01，执行 IDLE 指令后，系统进入 STANDBY 低功耗模式。CLKIN 停止工作，与系统时钟 SYSCLKOUT 有关的所有时钟信号均停止工作，但是时钟振荡器 OSCillator 和看门狗 WD 仍在运行。在进入 STANDBY 低功耗模式之前，必须先进行如下操作。

① 使能 PIEIER1[WAKEINT]，开放 WD 中断 \overline{WDINT} 和低功耗中断。

② 根据需要在 GPIOLPMSEL 寄存器中设置 GPIOA 端口唤醒信号，在 LPMCR0[LPM] 中设置低电平需要的保持时间。

被选中的 GPIO 信号、 \overline{XRS} 、 \overline{WDINT} 均可以使其退出低功耗模式。

(3) HALT 低功耗模式

若 LPM=10 或 11，执行 IDLE 指令后，系统进入 HALT 低功耗模式。在此低功耗模式下，包括 OSCillator 在内所有时钟信号均停止工作。但是在进入该模式之前，要使能 PIEIER1[WAKEINT]、选择并使能 GPIOLPMSEL，同时，使能 PIE[ENPIE]和 ST1[INTM]。注意，还要求 PLLSTS[MCLKSTS]=0，因此要检测 PLLSTS[MCLKSTS]位域状态。

总之，在进入上述低功耗模式之前，要确保所规定的中断能够唤醒并退出低功耗状态。

2.7 电源与复位系统

2.7.1 电源系统

1. 概述

电源是 DSP 应用系统的重要功能单元，按照电压数值分类，有 3.3V、1.9V 和 1.8V 三类；从对负载电路供电性质分析，又分为模拟电压源和数字电压源两种。在实际应用中，一般将 DSP 的 1.9V、1.8V 两种电源的引脚采用 1.9V 电压统一供电，所以 DSP 电源至少需要提供模拟电压 3.3V、1.9V 和数字电压 3.3V、1.9V，共 4 种供电电源。

其中，CPU、时钟系统和大部分片内外设等逻辑电路，需要 1.8V/1.9V 供电；I/O 口采用 3.3V 供电，以增强驱动能力。

同时，为了确保 DSP 上电稳定可靠运行，要优先给 DSP 的 1.8/1.9V 内核电源引脚上电，然后再给 DSP 的 I/O 口部分的电源上电，或同时上电。

2. 电源设计

在实际应用中，DSP 应用系统需要外接+5V 或以上电源，通过 DC-DC 变换，产生所需要的各类直流电压。因此，DSP 电源系统设计，普遍采用两级电源变换模式，电源电路如图 2.9 所示。

第一级，AC-DC 电压变换，可以采用标准的手持端设备供电电源，注意功率及输出电流。电源参数（仅供参考）：①交流输入端：交流电压 100~240VAC，频率 50Hz，电流 0.5A；②直流输出端：直流电压 5VDC，直流电流 1.5~2A。

第二级，DC-DC 电压变换，采用 DC-DC 直流电压变换模块（芯片），即利用第一级电源提供的+5V 直流电压，通过专用电源芯片及辅助电路，获得 3.3V 和 1.8/1.9V 电压，再通过电源隔离滤波技术，形成 DVDD3.3、AVDD3.3、DVDD1.8/1.9、AVDD1.8/1.9 几种满足 DSP 供电需求的电压源。

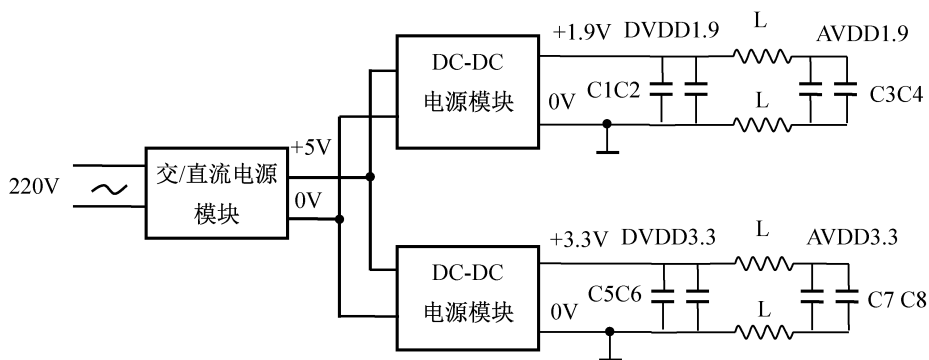


图 2.9 DSP 电源电路图

在 DSP 电源系统设计中，注意电源变换芯片的选型，并考虑负载功率、电流等，要满足 DSP 芯片以及外围电路的功率要求，同时要做好过压、过流保护设计。模拟负载电路要使用模拟电压源供电，数字负载电路使用数字电压源供电，做好模拟电压源与数字电压源之间的隔离设计，减小高频信号的干扰，提高电压源供电质量。对于 DSP 片内 ADC 模块，当使用片外参考电压模式时，要根据稳压要求使用高质量稳压芯片设计稳压电压源，为 ADC

模块提供高质量的片外稳压电源。

2.7.2 复位系统

复位功能是 DSP 控制器的一种工作机制，当系统上电、掉电或软件运行中出现异常时，采取的一种安全保护措施。

DSP 具体复位措施有：①系统上电自动复位，针对 DSP 系统开机启动情况；②开机后，通过专用复位按键进行手动复位；③软件运行过程中，看门狗 WD 计数溢出引发的系统复位。

DSP 复位信号有 2 个，其一，从外部复位引脚 \overline{XRS} 输入的低电平复位信号；其二，看门狗 WD 计数溢出产生的低电平复位信号 \overline{WDRST} 。如图 2.8 所示， \overline{WDRST} 信号控制电子开关，当 \overline{WDRST} 有效时，与复位引脚 \overline{XRS} 连通的信号线在 DSP 内部接地，并保持 512 个 OSCCLK 时钟周期的低电平保持时间，从而触发系统复位。外部复位引脚 \overline{XRS} 具有双向导通功能，看门狗 WD 的复位信号 \overline{WDRST} ，可以通过该引脚向外部输出复位信号，从而使外围相关器件在 \overline{WDRST} 信号的控制下，实现同步复位。

复位信号发生后，可以将 CPU 寄存器和各片上外设的寄存器，设置为默认值。并在系统复位中断的引导下，系统恢复至特定程序段或从头开始运行。

习题与思考题

- 2.1 概述 F28335 硬件结构及其组成，并对片上外设按照其功能进行简单分类。
- 2.2 简述 DSP 总线特征，详细列出 3 组地址总线和 3 组数据总线。
- 2.3 DSP 片内、片外地址总线宽度及对应最大可寻址范围分别是多少？
- 2.4 DSP 内存采用页寻址模式，可直接寻址的范围是多少？共分为多少页？每页的最大偏移量是多少？
- 2.5 详细分析 F28335 存储空间映射及其分布，分析 FLASH 存储空间范围，说明代码安全模块及 BOOT 向量（64W）存储区各自功能。
- 2.6 论述外部存储器接口的物理意义，以及片内片外统一编址基本原理。
- 2.7 DSP 时钟系统有多种设计方案，最常用的是哪种？若使用外部晶振模式，说明其基本参数值。
- 2.8 简述 PLL 频率配置原理。
- 2.9 DSP 时钟系统中，各功能单元的频率使能由控制寄存器设置，若启动 ADC 模块，简述外部频率设置与使能过程。
- 2.10 针对 F28335，并参阅其引脚分布资料，说明其电源系统电压分类？
- 2.11 概述看门狗模块的作用及工作原理。



课件



代码