

第1章 DSP 概述

1.1 DSP 概念

通常人们所讲到的 DSP 包含两个方面的内容，即数字信号处理（Digital Signal Processing）和数字信号处理器（Digital Signals Processor），两者都简称为 DSP。

数字信号处理（DSP）是研究用数字方法对信号进行分析、变换、滤波、检测、调制、解调及快速算法的一门技术学科。20世纪60年代以来，随着计算机和信息技术的飞速发展，数字信号处理技术应运而生并得到迅速发展。在过去的二十多年里，数字信号处理技术已经在通信等领域得到极为广泛的应用。数字信号处理是利用计算机或专用处理设备，以数字形式对信号进行采样、变换、滤波、估值、增强、压缩、识别等处理，以得到符合人们需要的信号形式。在高校的教学中，“数字信号处理”通常作为电子信息类专业的一门专业课而开设，主要讲述数字信号处理的算法与原理等。

数字信号处理器（DSP）是一种特别适合于进行数字信号处理运算的微处理器，其体系结构针对数字信号处理的操作需要进行了优化，其主要应用是实时快速地实现各种数字信号处理算法。DSP 的目标通常是测量、滤波或压缩连续的真实模拟信号。大多数通用微处理器也能成功地执行数字信号处理算法，但是专用的 DSP 通常具有更高的工作效率，因此它们更适合于便携式设备，如移动电话等。DSP 芯片的内部采用哈佛总线架构，具有专门的硬件乘法器，可以快速实现各种数字信号处理的算法。在当今的数字化时代背景下，DSP 已成为通信、计算机和消费类电子产品等领域的基础器件。本书主要讲述数字信号处理器（DSP），所以本书提到的 DSP 是指数字信号处理器，以后不再单独说明。

1.2 DSP 的发展历史与应用

DSP 的诞生是时代所需。在 DSP 出现之前，数字信号处理只能依靠微处理器来完成。但由于一般微处理器的处理速度较低，根本无法满足越来越大信息量的高速实时处理要求。因此，应用更快、更高效的信号处理方式成了日渐迫切的社会需求。

1978 年，AMI 公司发布了世界上第一个单片 DSP 芯片 S2811，但没有现代 DSP 芯片所必须有的硬件乘法器。

1979 年，美国 Intel 公司发布的商用可编程器件 920 是 DSP 芯片发展史上一个重要的里程碑，但其仍然没有硬件乘法器。

1980 年，日本 NEC 公司推出的 MPD7720 是第一个具有硬件乘法器的商用 DSP 芯片，从而被认为是第一块单片 DSP。

1982 年，TI（德州仪器）公司推出了 TMS32010 及其系列 DSP 产品。这些 DSP 产品采用微米工艺 NMOS 技术制作，虽功耗和尺寸稍大，但运算速度却比微处理器快了几十倍。这标志着 DSP 应用系统由大型系统向小型化迈进了一大步。至 20 世纪 80 年代中期，随着 CMOS 工艺的 DSP 芯片应运而生，其存储容量和运算速度都得到成倍提高，成为语音处理、图像处理技术的硬件基础。

20 世纪 80 年代后期至 90 年代，DSP 发展很快，集成度大幅提高，运算速度也进一步提高，

其应用范围逐步扩大到通信和计算机等领域。

进入 21 世纪后, DSP 芯片在性能上全面超越之前的产品, 同时基于商业目的不同发展出了诸多个性化的分支, 并开始逐渐拓展新的领域。

未来 DSP 将向以下几个方面继续发展与更新。

(1) DSP 芯片集成度越来越高

缩小 DSP 芯片尺寸一直是 DSP 的发展趋势。随着新工艺技术的引入, 越来越多的制造商开始改进 DSP 内核, 并且把多个 DSP 内核、MPU 内核及外围的电路单元集成在一个芯片上, 实现了 DSP 系统级的集成电路。

(2) 可编程 DSP 芯片将是未来主导产品

随着个性化发展的需要, DSP 的可编程化为用户提供了更多的灵活性, 满足了用户在同一个 DSP 芯片上开发出更多不同型号特征的系列产品, 也为用户对产品的功能升级提供了便利。例如, 为了提升产品性能, 冰箱和洗衣机等家用电器如今也可换成可编程 DSP 来对大功率电机进行控制。

(3) 从定点 DSP 占据主流向浮点 DSP 发展

目前市场上所销售的 DSP 芯片, 已从前些年占据主流 16 位的定点可编程 DSP 芯片, 向 32 位浮点 DSP 芯片发展, 32 位浮点 DSP 芯片将成为主流。

1.3 典型 DSP 产品简介

目前, 世界上 DSP 芯片制造商主要有 TI (德州仪器)、ADI (模拟器件公司) 和 Motorola (摩托罗拉) 等公司, 其中 TI 公司作为行业龙头, 占据了绝大部分的市场份额, ADI 和 Motorola 公司也有一定的市场。

TI 公司在 1982 年成功推出了其第一代 DSP 芯片 TMS32010。由于 TMS320 系列 DSP 芯片具有价格低廉、简单易用和功能强大等特点, 逐渐成为目前最有影响、最为成功的 DSP 系列产品之一。

TI 公司的 DSP 产品主要包括 C2000、C5000 和 C6000 这 3 个系列。C2000 系列主要用于数字控制系统; C5000 (C54x、C55x) 系列主要用于低功耗和便携式的无线通信终端产品, 如 C5000 系列中的 TMS320C54x 系列 DSP 被广泛应用于通信和个人消费电子领域; C6000 系列主要用于高性能复杂的数字图像处理系统。

C2000 系列产品主要面向数字控制和运动控制等, 主要包括:

- ① 32 位定点系列 DSP (240x 基础上升级), 主要包括 TMS320x280x/281x/282xx;
- ② 32 位浮点系列 DSP, (C28x+FPU)(Delfino)TMS320x283xx;
- ③ Piccolo 小封装系列 DSP (低价格+高性能), 主要包括 TMS320F2802x/2803x/2806x;
- ④ Concerto 系列 DSP, (ARM+C28x 内核) TMS320F28M35x。

本书以 C2000 系列的 TMS320F28335 作为主要内容进行介绍。

1.4 TMS320F28335 芯片简介

1.4.1 内部功能结构

TMS320F28335 的内部功能结构如图 1-1 所示, 主要由 CPU、多种类型的存储器和外设等组成, 它们之间通过总线连接在一起。

TMS320F28335 内部有 32 位 CPU 和 IEEE-754 单精度浮点单元 FPU; 256KW (千字) 的 Flash 存储器等; 增强 ePWM 单元等片上外设和总线系统等。

(1) 高性能 CPU (C28x+FPU)

- 32 位算术逻辑单元 (ALU);
- 32×32 位 (双 16×16 位) 的乘法器;
- 间接寻址的 8 个 32 位辅助寄存器和辅助算术单元 (ARAU);
- IEEE-754 单精度浮点单元 FPU。

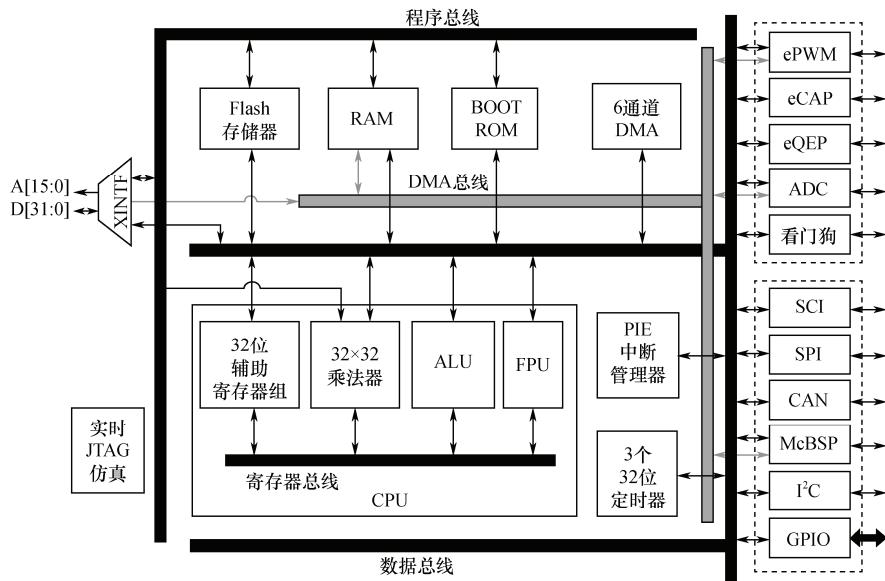


图 1-1 TMS320F28335 内部功能结构图

(2) 片上存储器

- 多达 512KB (256K×16 位) 的 Flash 存储器;
- 2KB (1K×16 位) 的 OTP 型只读存储器;
- 68KB (34K×16 位) 的单口随机访问 SARAM;
- 128 位的安全密钥;
- 16KB (8K×16 位) 的 BOOT (引导) ROM, 支持软件引导模式;
- 标准的数学表。

(3) 总线架构与流水线

- 哈佛 (Harvard) 总线架构, 地址总线和数据总线各 3 组;
- 8 级流水线操作, 提高处理速度。

(4) 中断控制

- 8 个外部中断;
- PIE 模块支持 96 (58) 个外设中断。

(5) 系统时钟

- 工作频率高达 150MHz (指令周期 6.67ns);
- 支持动态改变锁相环倍频系数, 具有片上振荡器和看门狗。

(6) 供电要求

- 内核电压为 1.9/1.8V; 输入、输出电压为 3.3V。

(7) 增强的控制外设

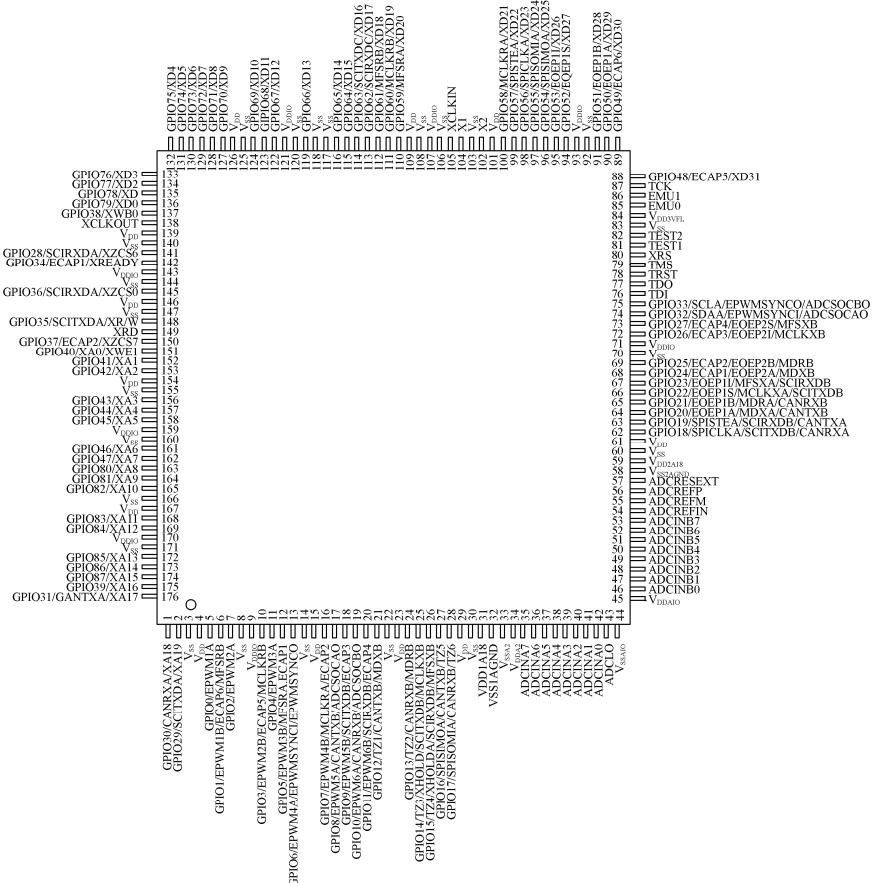
- EV 分解成 ePWM、eCAP 和 eQEP，且互不干扰，可实现复杂的信号输出。

(8) 其他外设

- 3 个 32 位 CPU 定时器；
- 1 个 16 通道 12 位 A/D 转换器；
- 3 个串行异步数字通信接口 (SCI)；
- 1 个串行外设接口模块 (SPI)；
- 2 个增强现场总线通信 (eCAN) 模块；
- 2 个多通道缓冲串口 (McBSP)；
- 1 个内部集成电路总线接口 (I²C)；
- 88 个可独立编程的 I/O 口；
- 6 通道 DMA 控制器；
- 16 或 32 位的外部接口 (XINTF)。

1.4.2 封装与引脚

TMS320F28335 由于封装不同，引脚数量也有所不同，其封装主要有 LQFP、PBGA 和 BGA。这 3 种。LQFP 有 176 个引脚，PBGA 有 176 个引脚，BGA 有 179 个引脚，其中 LQFP 封装如图 1-2 所示。



引脚主要分为电源接口类引脚、地址总线接口引脚、数据总线接口引脚、JTAG 仿真接口引脚、片内外设和通信接口引脚。详细说明请参考 TI 公司的产品手册。

输入引脚兼容 3.3V CMOS 电平信号（不能承受 5V 电压），输出引脚兼容 3.3V，输出缓冲驱动器的驱动电流典型值为 4mA，外部存储器接口引脚的驱动能力可达 8mA。

1.4.3 状态寄存器

TMS320F28335 的 CPU 寄存器主要有程序计数器 PC 和 RPC、辅助寄存器 XAR0~XAR7、数据页面指针 DP、堆栈指针 SP、状态寄存器 ST0 和 ST1 等。采用高级语言编程时，其他寄存器直接应用不多，在此仅就状态寄存器作出说明，其他请参考 TMS320F28335 数据手册。

1. 状态寄存器 ST0

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								

位域	名称	说 明
15~10	OVC/OVCU	溢出计数器。有符号运算时为 OVC，用以保存 ACC 的溢出信息；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溢出
5	N	负标志，反映某些操作中运算结果是否为负
4	Z	零标志，反映操作结果是否为零
3	C	进位位，反映加法运算是否产生进位，或者减法运算是否产生借位
2	TC	测试/控制位，反映位测试 (TBIT) 或归一化 (NORM) 指令的测试结果
1	OVM	溢出模式位，规定是否需要对 ACC 溢出结果进行调整
0	SXM	符号扩展位，决定输入移位器对数据移位时是否需要符号扩展

2. 状态寄存器 ST1

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
IDLESTAT	EALLOW	LOOP	SPA	VMAP	PAGE0	DBGM
R-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1

位域	名称	说 明
15~13	ARP	辅助寄存器指针，指示当前时刻的工作寄存器
11	M0M1MAP	M0 和 M1 的映射位，C28x 模式下为 1，C27x 兼容模式下为 0（仅供 TI 产品测试用）
9	OBJMODE	目标兼容模式位，用于在 C28x（该位为 1）和 C27x 模式（该位为 0）间选择
8	AMODE	寻址模式位，用于在 C28x（该位为 0）和 C2xLP 寻址模式（该位为 1）间选择
7	IDLESTAT	空闲状态位，只读。执行 IDLE 指令时置位，下列情况复位：执行中断；CPU 退出 IDLE 状态；无效指令进入指令寄存器或某个外设复位后
6	EALLOW	受保护寄存器访问允许位，对仿真寄存器或受保护寄存器访问前要将该位置 1
5	LOOP	循环指令状态位，CPU 执行循环指令时，该位置位
4	SPA	队列指针定位位，反映 CPU 是否已把堆栈指针 SP 定位到偶地址
3	VMAP	向量映射位，用于确定将 CPU 的中断向量表映射到最低位置（该位为 0）还是最高地址（该位为 1）
2	PAGE0	寻址模式设置位，用于直接寻址（该位为 1）和堆栈寻址（该位为 0）间选择
1	DBGM	调试功能屏蔽位，该位置位时，仿真器不能实现访问存储器和寄存器
0	INTM	中断屏蔽位，即可屏蔽中断的总开关，该位为 1，所有可屏蔽中断被禁止

复位时，即清除了 EALLOW 位，启用 EALLOW 保护。受保护时，CPU 对受保护寄存器的所有写入都将被忽略，只允许 CPU 读取、JTAG 读取和 JTAG 写入。如果通过执行 EALLOW 指令设置了该位，则允许 CPU 自由地写入受保护寄存器。修改寄存器后，可以通过执行 EDIS 指令清除 EALLOW 位，再次保护它们。

1.4.4 存储器与存储空间

如图 1-3 所示为 TMS320F28335 的存储器映射图，可见其数据空间和程序空间是统一的，由于程序地址总线为 22 位，因此最大寻址空间为 $2^{22}=4\text{MW}$ （地址范围 0x000000~0x3FFFFF）。对于数据存储空间，尽管图 1-3 中仅画出了与程序存储空间地址重合的区域，但由于片内数据地址总线为 32 位，故最大可寻址空间为 $2^{32}=4\text{GW}$ 。

由图 1-3 可见，其片内配置了各种类型的存储器：Flash（256KW）、SARAM（34KW）、BOOT ROM（8KW）、OTP（1KW），另外还预留了用于外扩存储器的空间。

在实际编程应用中，通过 CMD 文件来组织程序和数据存储空间的分配工作，具体参见第 9 章中关于 CMD 文件的内容。

1.4.5 TI 公司 DSP 命名规则

为了标示产品开发周期所处的阶段，TI 公司为所有 TMS320 器件和支持工具的部件分配了分类前缀。每个 TMS320TMMCU 商用系列成员都具有以下 3 个分类前缀中的一个：TMX、TMP 或 TMS（如 TMS320F28335）。TMS320 名称中主要包含 6 部分内容，具体含义如下。

① 分类前缀：TMX=实验器件；TMP=原型器件；TMS=合格器件。

② 系列号：320=TMS320 系列。

③ 工艺：C=COMS；E=COMS EEPROM；F=Flash EEPROM；LC=低电压 CMOS（3.3V）；LF=Flash EEPROM（3.3V）；VC=低电压 CMOS（3V）等。

④ 子系列类型：2xxxx=2000 DSP 系列；5xxxx=5000 DSP 系列；6 xxxx=6000 DSP 系列。

⑤ 封装类型：PGA，64 脚 TQFP；PGE，144 脚 TQFP；PZ，100 脚 TQFP；PGF，176 脚 LQFP 等。

⑥ 温度范围（默认 0~70°C）：L=0~70°C；A=-40~85°C；S=-40~125°C；Q=-40~85°C 等。

例如，TMS320F2xxxxPGFS 命名含义如图 1-4 所示。



图 1-3 TMS320F28335 的存储器映射图

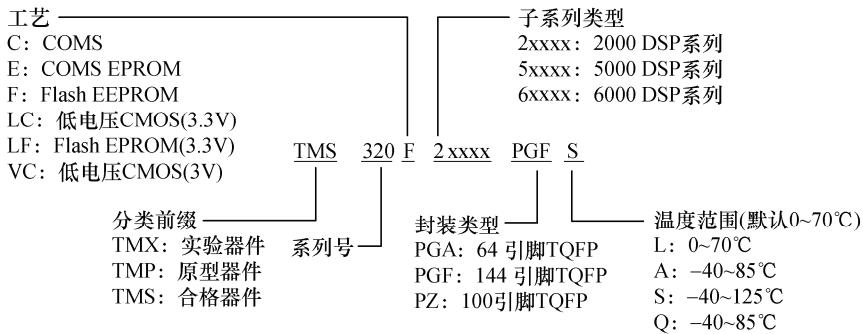


图 1-4 命名含义图

1.5 TMS320F28335 编程的一些说明

1.5.1 混合编程

DSP 的汇编指令系统相对复杂，但效率高；高级语言编程方便，但效率不如汇编语言。所以在有些场合，经常直接插入汇编语言或使用汇编语言+高级语言混合编程。

在 C 语言中直接嵌入汇编语言的格式为：

```
asm("汇编语句")
```

1. 与状态寄存器有关的混合编程

ST1[INTM]: 总中断屏蔽位。DSP2833x_Device.h 文件中混合编程程序代码为：

```
#define DINT asm("setc INTM") //使 ST1[INTM]=1,屏蔽所有可屏蔽中断
#define EINT asm("clrc INTM") //使 ST1[INTM]=0,使能所有可屏蔽中断
```

ST1[EALLOW]: 受保护寄存器访问允许位。DSP 中有些寄存器受 EALLOW 位保护，所以在访问前需解除保护。混合编程程序代码为：

```
#define EALLOW asm("EALLOW") //解除保护,相当于 ST1[EALLOW]置 1
#define EDIS asm("EDIS") //重新保护,相当于 ST1[EALLOW]清 0
```

2. 其他常用的混合编程

```
#define ESTOP0 asm("ESTOP0") //仿真停止
#define ERTM asm("clrc DBGM") //使能调试功能
#define DRTM asm("setc DBGM") //屏蔽调试功能
```

1.5.2 TMS320F28335 外设寄存器使用说明

为方便使用 C 语言开发的用户，TI 公司为外设寄存器提供了硬件抽象层方法，其思路是采用结构体和位域定义的形式定义片内外设寄存器，以方便访问寄存器或寄存器的某些位，然后在编译时将其映射到 DSP 数据空间对应的地址。外设寄存器的硬件抽象层描述包含在 include 文件夹中，其中的头文件 (.h) 给出了与片内外设寄存器对应的结构体和位域的定义，source 文件夹下的源文件 DSP2833x_GlobalVariableDefs.c 给出了寄存器结构体变量的段分配，链接命令文件 DSP2833x_Headers_nonBIOS.cmd 给出了段映射情况。

使用时，首先在主程序前应加入相应的头文件，然后在程序中即可对寄存器进行访问。例程如下：

```
#include "DSP2833x_Device.h"
```

```

#include "DSP2833x_Examples.h" //首先包括头文件,放于主程序前
void main(void)
{
    .....
    EALLOW; //宏指令,允许访问受保护寄存器,不是所有寄存器都用,视寄存器是否受 EALLOW 保护而定;
    GpioCtrlRegs.GPDIR.bit.GPIO7 = 0x1; //将 GPIO7 设置为输出口
    GpioDataRegs.GPADATA.bit.GPIO7= 0x1; //从 GPIO7 输出高电平
    EDIS; //宏指令, 恢复寄存器的保护状态
    .....
}

```

访问寄存器的格式主要有两种，即位访问和寄存器整体访问。对外设寄存器进行位访问操作的格式：外设结构体.寄存器共同体.bit.具体位域；对外设寄存器进行整体访问操作的格式：外设结构体.寄存器共同体.all，这样的方法类似于访问文件，即“根目录\子目录\文件”的形式。

bit 是头文件中定义的结构体变量，其中包含多个位域。每个位域可能是一位，也可能是几位，根据实际寄存器中具体的功能需要而定。下面是寄存器访问举例。

(1) 访问系统时钟寄存器的格式

SysCtrlRegs.PLLSTS.bit.DIVSEL = 2;	//对指定的位域访问
SysCtrlRegs.HISPCP.all = 0x0001;	//对指定的寄存器整体访问

(2) 访问 CPU 定时器寄存器的格式

CpuTimer2Regs.TPRH.all=0;	//对寄存器 TPRH 赋 0
CpuTimer1Regs.TCR.bit.TSS=1;	//对 TCR 寄存器的 TSS 置 1

(3) 访问中断寄存器的格式

PieCtrlRegs.PIECTRL.bit.ENPIE=1;	//允许从 PIE 向量表中读取中断向量
PieCtrlRegs.PIEIER3.bit.INTx1=1;	//允许 PIE 中断 INT3.1 即 ePWM1 中断

(4) 访问 I/O 寄存器的格式

GpioCtrlRegs.GPAPUD.bit.GPIO0 = 0;	//使能内部上拉
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 1;	//配置为 EPWM1A 功能

(5) 访问 ePWM 寄存器的格式

EPwm1Regs.ETSEL.bit.INTEN=1;	//允许中断使能 ePWM1 中断
EPwm1Regs.ETCLR.bit.INT=1;	//ETCLR.bit.INT 清除标志位

(6) 访问 eCAP 寄存器的格式

ECap1Regs.ECCLR.all = 0xFFFF;	//对 ECCLR 赋 0,即写 1 清 0
ECap1Regs.ECCTL1.bit.CAPLDEN =0x0;	//对 ECCTL1 的 CAPLDEN 位赋 0

(7) 访问 ADC 寄存器的格式

AdcRegs.ADCMAXCONV.all = 0x0003;	//设置最大通道数
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0;	//设置排序寄存器

(8) 访问 SCI 寄存器的格式

ScibRegs.SCICCR.all =0x0007;	//设置数据格式
ScibRegs.SCICCTL2.bit.TXINTENA = 1;	//使能发送中断

注意：对寄存器赋值时，可采用十进制数、二进制数和十六进制数。不同之处在于，十进

制数不用标注，二进制数和十六进制数则需要标注。例如：

```
ScibRegs.SCICCR.all =0x0007;           //十六进制设置数据格式
ScibRegs.SCICCR.all =00000111B;         //二进制设置数据格式
ScibRegs.SCICCR.all =7;                  //十进制设置数据格式
```

在实际编程中，由于二进制数较长而很少采用，十进制数和十六进制数采用较多。

思考与练习题

- 1-1 TI 公司的 DSP 产品主要有哪几个系列？各自的特点及应用领域是什么？
- 1-2 TMS320F28335 有哪些主要特点？
- 1-3 说明 TMS320F28335 命名的含义。