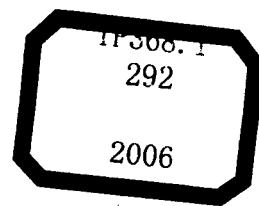


凌阳16位单片机

C语言开发

李晓白 秦红磊 编著
朱俊杰 潘涌泽



凌阳 16 位单片机 C 语言开发

李晓白 秦红磊 编著
朱俊杰 潘涌泽

北京航空航天大学出版社

内 容 简 介

凌阳 16 位单片机($\mu'nSP^TM$)是凌阳公司推出的新一代单片机,具有速度高,价格低,体积小,功耗低,可靠实用,简单易学等特点。本书以其代表型号 SPCE061A 为主线,讲述该系列单片机的内部结构、指令系统、开发工具,并结合开发环境自带的相关库函数和头文件,列举了大量的实例,详细地讲述了如何使用标准 C 语言来编程实现对微控制器的控制;列举了 SPCE061A 在实际工程中的两个应用以加深理解,强化应用。

本书体系完整,层次清晰,通俗易懂,例题丰富,可读性强,可作为本科生、专科生、职校生在电子实习、课程设计、电子竞赛、毕业设计过程中的参考用书,也可作为从事单片机开发与应用的工程技术人员及广大单片机爱好者的自学用书。

图书在版编目(CIP)数据

凌阳 16 位单片机 C 语言开发/李晓白等编著. —北京:
北京航空航天大学出版社, 2006. 9
ISBN 7-81077-872-2

I. 凌… II. 李… III. 单片微型计算机—C 语言—
程序设计 IV. TP368.1②TP312

中国版本图书馆 CIP 数据核字(2006)第 112534 号

© 2006, 北京航空航天大学出版社, 版权所有。

未经本书出版者书面许可, 任何单位和个人不得以任何形式或手段复制或传播本书内容。
侵权必究。

凌阳 16 位单片机 C 语言开发

李晓白 秦红磊 编著
朱俊杰 潘涌泽
责任编辑 王鑫光

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn> E-mail: bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本: 787×1092 1/16 印张: 23.5 字数: 602 千字

2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷 印数: 5 000 册

ISBN 7-81077-872-2 定价: 35.00 元

前 言

由于单片机具有多功能、低价位及小型化等优势,它们已被广泛地应用在消费电子、汽车电子、办公室自动化、通信及一般工业产品上。近些年来,单片机技术的发展已达到了相当高的水平,一些新型单片机层出不穷,技术日新月异。传统单片机(MCU)主要由外围部件、微控制器和存储器构成。新型单片机为了更加灵活地适应不同领域的应用,除了具有这些传统的数字功能外,还提供模拟功能,甚至射频功能,它们不仅是可编程的,而且是可动态重构的。它们是具有 PSoC 或 SoPC 特性的、真正的单片机。

单片机的智能和灵巧可能已经深深地吸引了你,然而众多初学者关心的是,如何能用较少的投入达到事半功倍的效果。显然,兴趣是学习的巨大动力,低成本的开发环境是学习的桥梁。当然,学习的目的终究是为了完成工程项目。做单片机项目开发,首先应该关注的是项目的功能和性能要求,即做需求分析。通过需求分析,明确实现这些功能和性能指标所需芯片的功能和结构资源、综合成本核算和产品集成度等要求,才能从琳琅满目、特色各异的芯片中选择合适的货架产品。凌阳科技股份有限公司开发的 16 位单片机 SPCE061A,是一款内含数字信号处理的微控制器,适用于初学者开发的、实用的、具有数字语音功能的产品,并能通过网上凌阳大学计划提供的正规开发环境和丰富的免费学习资料来获得帮助。

本书共 14 章。第 1 章介绍了凌阳 16 位单片机 SPCE061A 的性能结构和特点。芯片具有 DSP 功能的 16 位 RISC 内核,灵活、省电的锁相环时钟,丰富、实用的集成外设资源,特色数字语音功能。第 2 章给出了关于 C 语言的一些概念,为表述开发芯片各项功能做一些必要的准备。第 3~8 章详细讲述了 SPCE061A 单片机基本资源的 C 语言应用程序设计。包括系统设置、基本 I/O 端口设置、定时器/计数器设置、A/D 和 D/A 转换设置、串行接口 SIO 和 UART 的设置以及中断系统设置的 C 语言程序设计。第 9 章为该芯片的特色——数字语音应用 C 语言程序设计。第 10 章对于嵌入式实时操作系统 MiniOS 的特点、基本概念、编程及程序调试等进行了详细的说明。第 11、12 章分别介绍了编写程序使用的软件开发环境和检验程序使用的硬件平台——实验仪。第 13 章是 15 个基础实验,通过这些实验,读者可以深入了解 SPCE061A 的工作原理并训练自己的程序设计能力。第 14 章是应用实例,读者可通过数码录音笔的设计来检验自己的综合设计能力。另外,本书选择 CAN 总线组网技术介绍了扩展芯片外设资源的方法。所用程序均采用 C 语言描述,力求简明扼要。

本书的取材力求内容先进,涵盖经典,图文并茂,篇幅适中。

在编写过程中,北京航空航天大学沈士团教授、李驿华教授、谈展中教授给予了大力的支持和帮助,倪玉霞老师、邸瑞辉、孙小续等同学在收集和整理资料方面做了大量的工作,在此一并表示衷心感谢。

本书第 1~4 章由朱俊杰老师编写,第 5~9 章由秦红磊老师编写,第 10~14 章由李晓白老师编写,第 13 章由潘涌泽老师编写,李晓白对全书做了统稿。由于编者水平有限,差错在所难免,敬请读者批评指正。

作 者

2006 年 7 月于北航

目 录

第 1 章 SPCE061A 单片机简介	1
1.1 凌阳 16 位单片机	1
1.2 SPCE061A 简介	2
1.2.1 概 述	2
1.2.2 性 能	2
1.2.3 结构概览	2
1.2.4 芯片的引脚说明	3
1.2.5 开发方法	5
1.2.6 应用领域	5
1.3 $\mu'nSP$ 的内核结构	5
1.3.1 算术逻辑运算单元 ALU	6
1.3.2 寄存器组	7
1.3.3 数据总线和地址总线	9
1.4 SPCE061A 片内存储器结构	9
1.4.1 RAM	9
1.4.2 堆 栈	9
1.4.3 闪存 FLASH	10
第 2 章 C 语言的基础知识	14
2.1 数据与运算	14
2.1.1 数据与数据类型	14
2.1.2 常量、变量、运算符与表达式	14
2.2 流程控制语句	16
2.3 构造数据类型	19
2.3.1 数 组	20
2.3.2 指 针	21
2.3.3 结构体和共用体	23
2.4 函 数	25
2.4.1 函数的定义形式	25
2.4.2 形式参数与实际参数	26
2.4.3 函数的返回值	26
2.4.4 函数调用格式	26

2.4.5 函数调用规则	27
2.4.6 C 函数和汇编函数的相互调用	28
2.5 $\mu'nSP$ 的 C 语言嵌入式汇编	29
第 3 章 I/O 端口的 C 语言程序设计	34
3.1 I/O 端口的结构	34
3.2 I/O 端口设置的寄存器	34
3.2.1 A 口相应的寄存器	35
3.2.2 并行 I/O 口的组合控制	35
3.2.3 B 口相应的寄存器	36
3.2.4 B 口的特殊功能	37
3.3 I/O 端口设置的 C 库函数	37
3.4 I/O 端口的应用实例	39
第 4 章 系统设置的 C 语言程序设计	42
4.1 硬件结构	42
4.1.1 系统时钟	42
4.1.2 锁相环 PLL 振荡器	42
4.1.3 利用 B 口的特殊功能产生振荡信号	43
4.1.4 低电压复位/低电压监测	43
4.1.5 看门狗计数器	44
4.2 系统设置的寄存器	44
4.3 系统设置的 C 函数	46
4.4 系统设置的应用实例	48
第 5 章 定时器/计数器的 C 语言程序设计	50
5.1 定时器/计数器的结构	50
5.1.1 时基信号发生器	50
5.1.2 定时器/计数器	50
5.2 定时器/计数器控制的寄存器	52
5.3 定时器/计数器设置的 C 函数	54
5.4 定时器/计数器的应用实例	56
第 6 章 ADC 和 DAC 的 C 语言程序设计	58
6.1 ADC 和 DAC 的硬件特性	58
6.1.1 ADC 输入接口的结构	58
6.1.2 ADC 的直流电气特性	60
6.1.3 DAC 音频输出的结构	60
6.2 ADC 和 DAC 设置的寄存器	61

6.2.1	MIC 输入的存储单元 P_ADC(读/写)(7014H)	61
6.2.2	ADC 的控制端口 P_ADC_Ctrl(读/写)(7015H)	61
6.2.3	ADC 多通道控制单元 P_ADC_MUX_Ctrl(读/写)(702BH)	62
6.2.4	模拟电压输入的存储单元 P_ADC_MUX_Data(读)(702CH)	63
6.2.5	DAC 数据存放的单元	63
6.2.6	DAC 音频输出方式的控制单元 P_DAC_Ctrl(写)(702AH)	63
6.3	ADC 和 DAC 设置的 C 函数	64
6.4	ADC 和 DAC 的应用实例	68
第 7 章	串行接口 SIO 和 UART 的 C 语言程序设计	71
7.1	串行口的硬件特性	71
7.2	串行口控制的寄存器	72
7.3	串行口设置的 C 函数	75
7.4	串行口应用实例	79
第 8 章	中断系统的 C 语言程序设计	82
8.1	中断系统	82
8.1.1	中断源	82
8.1.2	中断优先级和中断入口地址	84
8.2	中断控制	84
8.2.1	中断控制的寄存器	84
8.2.2	中断响应过程	86
8.3	中断控制的相关 C 函数	88
8.4	中断控制指令	89
8.5	中断系统的应用实例	90
8.5.1	单中断源应用	90
8.5.2	多中断源应用	94
第 9 章	凌阳音频的 C 语言程序设计	98
9.1	压缩分类和压缩算法	98
9.1.1	凌阳音频压缩算法的编码标准	98
9.1.2	压缩分类	98
9.1.3	凌阳常用的音频形式和压缩算法	98
9.1.4	语音压缩方法	98
9.1.5	在开发环境 $\mu'nSP^TM$ IDE 里添加语音资源	100
9.2	常用的应用程序接口 API 的功能介绍及应用	103
9.2.1	概 述	103
9.2.2	SACM_A2000	104
9.2.3	SACM_S480	112

9.2.4	SACM_MS01	119
9.2.5	SACM_DVR	126
9.3	语音辨识	133
9.4	本章 API 函数中所占用的寄存器	140
第 10 章	嵌入式实时操作系统	142
10.1	MiniOS 的特点	142
10.2	操作系统概述以及 OS 的一些基本概念	143
10.3	内核的启动	146
10.4	头文件及内核配置	149
10.5	通信与同步	151
10.5.1	信号量	151
10.5.2	邮 箱	157
10.5.3	消息队列	159
10.5.4	小 结	161
10.6	中断过程及处理	164
10.6.1	中断处理过程	164
10.6.2	中断中的通信	167
10.6.3	内存管理	169
10.6.4	驱动程序	170
10.6.5	综合应用	171
10.7	MiniOS 程序的调试	179
10.7.1	系统用的全局性调试信息	180
10.7.2	与任务相关的全局性调试信息	181
10.7.3	与中断相关的全局性调试信息	182
10.7.4	事件的结构	183
10.7.5	内存控制块	184
10.8	MiniOS API 列表	184
10.8.1	初始化部分	184
10.8.2	任务部分	185
10.8.3	事件部分	189
10.8.4	中断处理部分	197
10.8.5	内存管理部分	199
10.8.6	其 他	201
第 11 章	开发环境简介	203
11.1	概 述	203
11.2	菜单、工具栏	204
11.3	窗 口	207

11.3.1	Workspace 窗口	208
11.3.2	Output 窗口	208
11.3.3	Edit 窗口	209
11.3.4	Memory 窗口	209
11.3.5	Register 窗口	209
11.3.6	Command 窗口	210
11.3.7	Watch 窗口	210
11.3.8	Disassembly 窗口	210
11.4	工 程	211
11.4.1	工作区窗口	211
11.4.2	工程的操作	211
11.4.3	工程的设置	212
11.4.4	工程的编制	214
11.4.5	运行程序	215
11.4.6	加载程序	215
11.4.7	调试器	215
11.5	程序示例	216
第 12 章	实验仪	220
12.1	系统概述	220
12.1.1	功能特点	220
12.1.2	应用领域	221
12.2	实验仪原理图及模块说明	221
12.2.1	电 源	222
12.2.2	SPCE061A 芯片周边电路	222
12.2.3	I/O 口电路	223
12.2.4	音频输入	224
12.2.5	音频输出	224
12.2.6	单路 0~3.3 V 可调电路	225
12.2.7	SPR4096 构成的 SIO 存储电路	225
12.2.8	异步串口通信电路	225
12.2.9	1×8 键盘	226
12.2.10	4 位数码管及 LED 指示灯显示电路	226
12.2.11	LCD	227
第 13 章	基础实验	229
实验 1	I/O 口实验	229
实验 2	定时器 A/B 实验	231
实验 3	FIQ 中断实验	233

实验 4	IRQ0/IRQ1/IRQ2 中断实验	237
实验 5	IRQ4 中断实验	239
实验 6	IRQ5 中断实验	243
实验 7	IRQ6 中断实验	246
实验 8	UART 实验	249
实验 9	A/D 转换实验	252
实验 10	双通道 D/A 实验	254
实验 11	一路输入的录音实验	255
实验 12	32K FLASH 读/写实验	257
实验 13	语音实验	260
实验 14	4 位 7 段 LED 数码管显示实验	264
实验 15	LCD 显示实验	267
第 14 章	SPCE061A 单片机应用实例	276
14.1	数码录音笔	276
14.1.1	开发背景	276
14.1.2	硬件结构框图	276
14.1.3	参考程序	276
14.2	Can 总线的控制	287
14.2.1	Can 总线概述	287
14.2.2	BasicCan 模式下网络测试功能的实现	289
14.2.3	测试平台应用说明	291
14.2.4	操作示例	293
14.2.5	程序流程图	294
14.2.6	参考程序	295
附录 A	C-Lib 中的函数集	335
附录 B	μ^nSP 的指令集	345
附录 C	μ^nSP 汇编伪指令集	352
附录 D	μ^nSP 编译相关错误信息	354
附录 E	端口速查表	362
附录 F	符号约定	364
参考文献		365

第 1 章 SPCE061A 单片机简介

1.1 凌阳 16 位单片机

随着单片机功能集成化的发展,其应用领域也逐渐由传统的控制扩展为控制处理、数据处理以及数字信号处理(DSP, Digital Signal Processing)等领域。凌阳的 16 位单片机就是为适应这种发展而设计的,其 CPU 内核采用凌阳最新推出的 $\mu'nSP^{\text{TM}}$ (Microcontroller and Signal Processor) 16 位微处理器芯片。围绕 $\mu'nSP$ 所形成的 16 位 $\mu'nSP$ 系列单片机(以下简称 $\mu'nSP$ 家族)采用的是模块式集成结构,它以 $\mu'nSP$ 内核为中心集成不同规模的 ROM、RAM 和功能丰富的各种外设接口部件,如图 1.1 所示。

$\mu'nSP$ 家族具有共享的微处理器核心,并可视需求增减功能模块。结合 $\mu'nSP$ 核心与不同的功能模块,便形成各种不同系列的产品,以配合多样化的应用场合。这样做可以使 $\mu'nSP$ 家族的系列产品具有更强的功能和更低的成本。

$\mu'nSP$ 家族有以下特点:

① 体积小,集成度高,可靠性好,且易于扩展。 $\mu'nSP$ 家族把各功能部件模块化地集成在一个芯片里,内部采用总线结构,因而减少了各功能部件之间的连线,提高了其可靠性和抗干扰能力。另外,模块化的结构易于系统扩展,可以适应不同用户的需求。

② 具有较强的中断处理能力。 $\mu'nSP$ 家族的中断系统支持 10 个中断向量及 10 余个中断源,适合实施应用领域。

③ 具有高性能价格比。 $\mu'nSP$ 家族片内带有高寻址能力的 ROM、静态 RAM 和多功能的 I/O 口。另外, $\mu'nSP$ 还提供了能加速数字信号处理的指令——高速的 16 位 $\times$ 16 位乘法运算指令以及内积(乘法)运算指令,使得 $\mu'nSP$ 家族在复杂的数字信号处理时更加便利,又比专用的 DSP 芯片廉价。

④ 具有功能强、效率高的指令集。 $\mu'nSP$ 指令集的格式紧凑,执行迅速,并且其指令集提供了对高级语言的支持,这可以大大缩短产品的开发时间。

⑤ 低功耗,低电压。 $\mu'nSP$ 家族采用 CMOS 制造工艺,并提供了可有软件控制的 3 种节电模式,即弱振模式(Weak)、暂停模式(Halt)和备用模式(Standby),大大地降低了其耗电量。另外, $\mu'nSP$ 家族的工作电压范围大,能在低电压供电时正常工作,且能用电池供电;这对于便携式的应用具有特殊的意义。

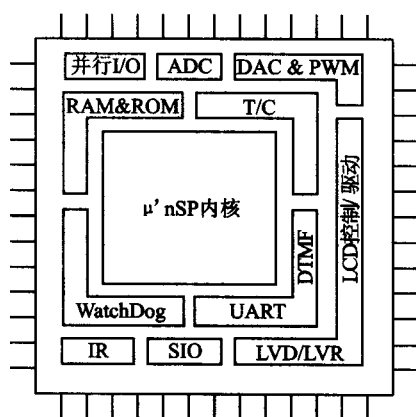


图 1.1 $\mu'nSP$ 家族的模块式结构

以 $\mu'nSP$ 为内核的单片机有 SPCE ***、SPT660 * 等系列,本书中使用的开发系统是 SPCE061A 和 IDE1.8.4,所有例程均在此环境下开发。

1.2 SPCE061A 简介

1.2.1 概 述

SPCE061A 是继 SPCE500A 之后,凌阳科技推出的又一款 16 位结构的微控制器。与 SPCE500A 不同的是,在存储器资源方面考虑到用户较少资源的需求以及便于程序调试等功能,SPCE061A 里只内嵌 32K 字的闪存(FLASH)。较高的处理速度使 $\mu'nSP$ 能够非常容易、快速地处理复杂的数字信号。因此,与 SPCE500A 相比,以 $\mu'nSP$ 为核心的 SPCE061A 微控制器是适用于数字语音识别应用领域产品的一种经济的选择。

1.2.2 性 能

SPCE061A 的主要性能特点如下:

- 16 位 $\mu'nSP$ 微处理器;
- 工作电压:VDD 为 2.6~3.6 V(CPU),VDDH 为 VDD~5.5 V(I/O);
- CPU 时钟为 0.32~49.152 MHz;
- 内置 2K 字的 SRAM;
- 内置 32K 字的 FLASH;
- 可编程音频处理;
- 系统处于备用状态下(时钟处于停止状态),耗电小于 2 μA @3.6 V;
- 2 个 16 位可编程定时器/计数器(可自动预置初始计数值);
- 2 个 10 位 DAC(D/A 转换)输出通道;
- 32 位通用可编程输入/输出端口;
- 14 个中断源,包括定时器 A/B、时基及 2 个外部时钟源输入;
- 具备触键唤醒的功能;
- 使用凌阳音频编码 SACM_S240 方式(2.4 kbps),能容纳 210 s 的语音数据;
- 锁相环 PLL 振荡器提供系统时钟信号;
- 32768 Hz 实时时钟;
- 7 通道 10 位电压 A/D 转换器(ADC)和 1 个单通道的声音专用 A/D 转换器;
- 声音 A/D 转换器输入通道,内置麦克风放大器和自动增益控制(AGC)电路;
- 具备串行设备接口;
- 具有低电压复位(LVR)功能和低电压监测(LVD)功能;
- 内置在线仿真电路 ICE(In-Circuit Emulator)接口;
- 具有保密能力;
- 具有 WatchDog 功能(由具体型号决定)。

1.2.3 结构概览

SPCE061A 的结构如图 1.2 所示。

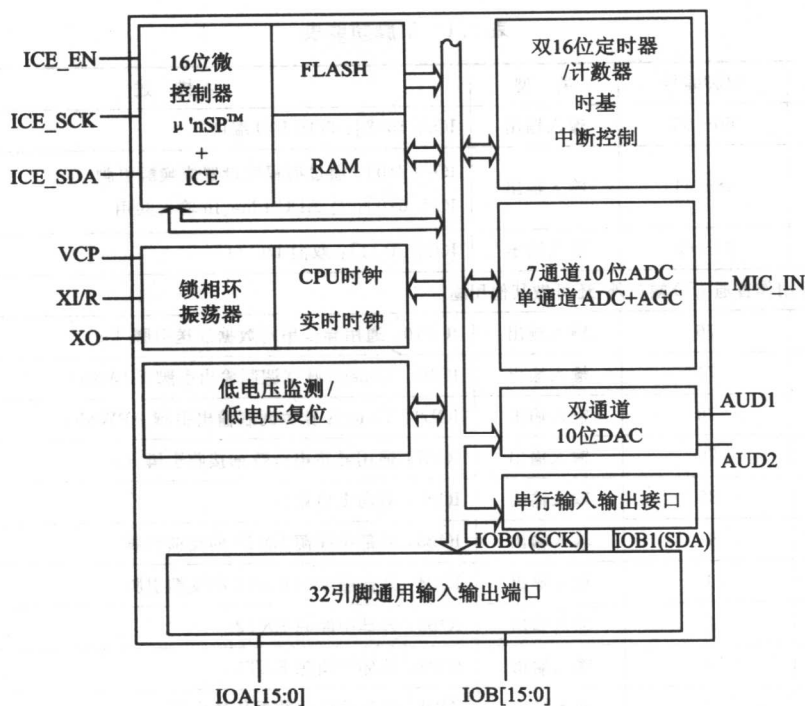
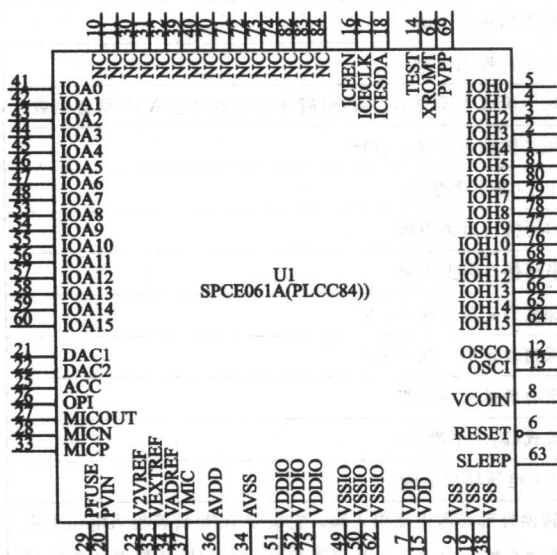


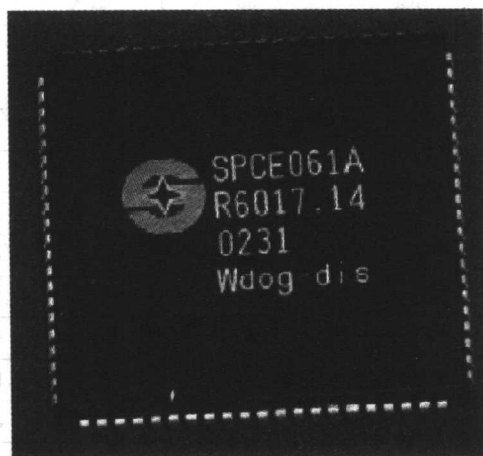
图 1.2 SPCE061A 的结构

1.2.4 芯片的引脚说明

SPCE061A 有 PLCC84 和 QFP80 两种封装。封装形式为 PLCC84 的共有 84 个引脚，其中包括空脚 15 个，其余引脚功能说明如表 1.1 所列。QFP80 封装在引脚方面只是比 PLCC84 封装少了 4 个空脚。SPCE061A PLCC84 封装排列及实物图见图 1.3。



(a) PLCC84封装排列图



(b) SPCE061A实物图

图 1.3 SPCE061A PLCC84 封装排列及实物图

表 1.1 引脚功能表

引脚名称	引脚编号	类 型	描 述
IOA[15:8]	60~53	输入输出	IOA[15:8]: 双向 I/O 端口
IOA[7:0]	48~41	输入输出	IOA[7:0]: 通过编程可设置成唤醒引脚 IOA[6:0]: 与 ADC Line_In 输入共用
IOB[15:11]	64~68	输入输出	IOB[15:11]: 双向 I/O 口
IOB[10:0]: 除用作普通 I/O 端口外,还可做其他用途			
IOB10	76	输入输出	IOB10: 通用异步串行数据发送引脚 Tx
IOB9	77	输入输出	IOB9: TimerB 脉宽调制输出引脚 BPWMO
IOB8	78	输入输出	IOB8: TimerA 脉宽调制输出引脚 APWMO
IOB7	79	输入输出	IOB7: 通用异步串行数据接收引脚 Rx
IOB6	80	输入输出	IOB6: 双向 I/O 端口
IOB5	81	输入输出	IOB5: 外部中断源 EXT2 的反馈引脚
IOB4	1	输入输出	IOB4: 外部中断源 EXT1 的反馈引脚
IOB3	2	输入输出	IOB3: 外部中断源 EXT2
IOB2	3	输入输出	IOB2: 外部中断源 EXT1
IOB1	4	输入输出	IOB1: 串行接口的数据传送引脚
IOB0	5	输入输出	IOB0: 串行接口的时钟信号
DAC1	21	输出	DAC1 数据输出引脚
DAC2	22	输出	DAC2 数据输出引脚
OSC31I	13	输入	32768 Hz 晶振输入引脚
OSC32O	12	输出	32768 Hz 晶振输出引脚
AGC	25	输入	AGC 的控制引脚
MINC	28	输入	麦克风负向输入引脚
MICP	33	输入	麦克风正向输入引脚
MICOUT	27	输出	麦克风 1 阶放大器输出引脚,引脚外接电阻决定 AGC 增益倍数
OPI	26	输入	麦克风 2 阶放大器输入引脚
XICE	16	输入	激活 ICE(高电平激活)
XICECLK	17	输入	ICE 串行接口时钟引脚
XICESDA	18	输入输出	ICE 串行接口数据引脚
PFUSE	29	输入	程序保密设定脚。接 +5 V
PVIN*	20	输入	程序保密设定脚。接 GND
XRESB	6	输入	低电平有效的复位引脚
SLEEP	63	输出	睡眠模式(高电平激活)
VREF2	23	输出	2 V 参考电压输出引脚
VRTPAD	35	输入	A/D 转换外部参考电压引入脚。它决定 A/D 转换输入电压上限值。若该点输入一个 2.5 V 参考电压,则 A/D 转换电压输入范围为 0~2.5 V。外部 A/D 最高参考电压小于 3.3 V

续表 1.1

引脚名称	引脚编号	类 型	描 述
VCM	34	输出	ADC 参考电压输出引脚
VMIC	37	输出	麦克风电源
VCP	8	输入	锁相环压控振荡器的阻容输入
XROMT、PVPP、 XTEST	61、69、14		出厂测试用引脚
VDDH	51、52、75	输入	I/O 电平参考。若该点输入一个 5 V 的参考电压,则 I/O 输入输出高电平为 5 V
VDD	7	输入	锁相环电压
VSS	9	输入	锁相环地
VSS	19、24	输入	模拟地
VSS	38、49、50、62	输入	数字地
VDD	15、36	输入	数字电源

注: * 将 PFUSE 接 5 V, PVIN 接 GND, 并维持 1 s 以上, 即可将内部保险丝熔化, 此后无法读取和向闪存加载数据。

1.2.5 开发方法

SPCE061A 的开发是通过在线调试器 PROBE 实现的。PROBE 既是一个编程器(即程序烧写器), 又是一个实时在线调试器。它可以替代在单片机应用项目的开发过程中常用的软件工具——硬件在线实时仿真器和程序烧写器。PROBE 利用了 SPCE061A 片内置的在线仿真电路 ICE(In-Circuit Emulator)接口和凌阳公司的在线串行编程技术, 工作于凌阳 IDE 集成开发环境软件包下, 其 5 芯的仿真头直接连接到目标电路板上 SPCE061A 的相应引脚, 在目标电路板上的 CPU(SPCE061A)直接调试、运行用户编制的程序。PROBE 的另一头是标准 25 针打印机接口, 直接连接到计算机打印口与上位机通信, 在计算机 IDE 集成开发环境软件包下, 完成在线调试功能。

1.2.6 应用领域

SPCE061A 的主要应用领域如下:

- 语音识别类产品;
- 有语音功能的仪器仪表、家电产品及自动售货机;
- 智能语音交互式玩具;
- 高级亦教亦乐类玩具;
- 儿童电子故事书类产品;
- 通用语音合成器类产品;
- 需较长语音持续时间类产品。

1.3 μ' nSP 的内核结构

μ' nSP 的核心结构由总线、ALU 算术逻辑运算单元、寄存器组、中断系统及堆栈等部分组

成。其结构如图 1.4 所示。

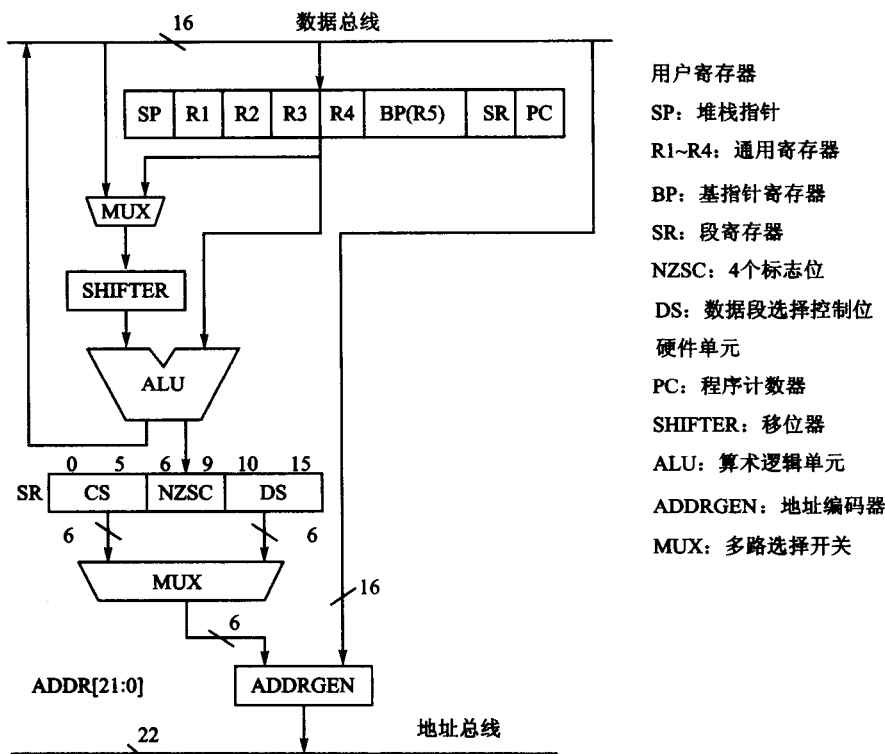


图 1.4 $\mu'nSP$ 的内核结构

1.3.1 算术逻辑运算单元 ALU

$\mu'nSP$ 的 ALU 在运算能力上很有特色,它不仅能做 16 位的基本算术逻辑运算,也能做带移位操作的 16 位算术逻辑运算;同时在数字信号处理方面,提供了高速的 16 位 $\times$ 16 位乘法运算和内积(乘加)运算。

1. 16 位算术逻辑运算

与大多数 CPU 一样, $\mu'nSP$ 提供了基本的算术运算与逻辑操作指令,如加法、减法、比较、取补、异或、或、与、测试、写入和读出等 16 位算术逻辑运算及数据传送操作。

2. 带移位操作的 16 位算术逻辑运算

移位运算包括算术右移 ASR、逻辑左移 LSL、逻辑右移 LSR、循环左移 ROL 及循环右移 ROR。从图 1.4 中可以发现在 $\mu'nSP$ 的 ALU 前面串接了一个移位器 SHIFTER,也就是说,操作数在经过 ALU 的算术逻辑操作前可先进行移位处理,然后再经 ALU 完成算术逻辑运算操作。因此, $\mu'nSP$ 的指令系统里专有一组复合式的移位算术逻辑操作指令,用来完成移位和算术逻辑操作两项功能。程序设计者可利用这些复合式指令撰写更精简的程序代码,进而增加程序代码密集度(Code Density)。在微控制器应用中,如何增加程序代码密集度是非常重要的问题。提高程序代码密集度可以减少程序代码的大小,进而减少 ROM 或 FLASH 的需求,以降低系统成本与增加执行效能。

3. 16 位×16 位的乘法运算和内积运算

除了普通的 16 位算术逻辑运算指令外, $\mu'nSP$ 还提供了高速的 16 位×16 位乘法运算指令 MUL 和 16 位内积运算指令 MULS。二者都可以用于有符号数相乘(signed×signed)或无符号数与有符号数相乘(unsigned×signed)的运算。在 ISA1.1 指令集下, MUL 指令只需花费 12 个时钟周期, MULS 指令花费 $10n+6$ 个时钟周期, 其中 n 为乘积求和的项数。例如, “MR=[R2] * [R1], 4”表示求 4 项乘积的和, MULS 指令只需花费 $46(10 \times 4 + 6 = 46)$ 个时钟周期。这两条指令大大提升了 $\mu'nSP$ 的数字信号处理能力, 为应用于复杂的数字信号处理运算方面提供了便利条件。

1.3.2 寄存器组

$\mu'nSP$ 的 CPU 寄存器组一共有 8 个 16 位寄存器, 可分为通用寄存器和专用型寄存器两大类。通用寄存器包括 R1~R4, 作为算术逻辑运算的源和目标寄存器。专用寄存器包括 SP、BP、SR 及 PC, 是与 CPU 特定用途相关的寄存器。

1. 通用寄存器 R1~R4(General-purpose Registers)

R1~R4 通常可分别用于数据运算或传送的源及目标寄存器。而寄存器 R4、R3 配对使用可组成一个 32 位的乘法结果寄存器 MR, 其中 R4 为 MR 的高字符组, R3 为 MR 的低字符组, 用于存放乘法运算或内积运算结果。

2. 堆栈指针寄存器 SP(Stack Pointer)

SP 是用来记录堆栈地址的寄存器, SP 指向堆栈的顶端。堆栈是一个先进后出的内存结构, $\mu'nSP$ 的堆栈结构是由高地址向低地址的方向储存的。当 CPU 执行压栈/出栈指令(push/pop)、子程序调用/返回指令(call/retf)以及进入中断服务子程序 ISR(Interrupt Service Routine)或从 ISR 返回指令(reti)时, SP 会自动减少(压栈)或增加(弹栈)来表示堆栈指针的移动。 $\mu'nSP$ 堆栈的大小限制在 2K 字的 SRAM 内, 即地址为 $0x000000 \sim 0x0007FF$ 的内存范围中。

3. 基址指针寄存器 BP(Base Pointer)

$\mu'nSP$ 提供了一种方便的寻址方式, 即基址寻址方式[BP+IM6]。程序设计者可通过 BP 来存取 ROM 与 RAM 中的数据, 包括局部变量(Local Variable)、函数参数(Function Parameter)及返回地址(Return Address)等, 这在 C 语言程序中是特别有用的。BP 除了上述用途外, 也可作为通用寄存器 R5, 用于数据运算或传送的源及目标寄存器。因此, 在程序中, BP 与 R5 是共享的, 均代表基址指针寄存器。

4. 程序计数器 PC(Program Counter)

它的作用与一般微控制器中的 PC 相同, 是用来记录程序目前执行位置的寄存器, 以控制程序走向。CPU 每执行完一个指令, 就会改变 PC 的值, 使其指向下一条指令的地址。在 $\mu'nSP$ 里, 16 位的 PC 寄存器与 SR 寄存器的 CS 字段, 共同组成一个 22 位的程序代码地址。

5. 状态寄存器 SR(Status Register)

SR 内含许多字段, 每个字段都有特别的用途, 如图 1.4 所示。其中包含两个 6 位的区段选择字段: CS(Code Segment)和 DS(Data Segment), 它们可与其他 16 位的寄存器结合在一