



PIC[®]

单片机入门与实战

张明峰 编著



北京航空航天大学出版社

PIC[®]单片机入门与实战

张明峰 编著

北京航空航天大学出版社

内 容 简 介

从 PIC 中档系列单片机的结构介绍入手,系统阐述 PIC 单片机开发过程的最基本步骤;通过引入设计实例,深入浅出地介绍 PIC 单片机的各主要功能模块。书中介绍的代码实例将以汇编指令为主,最后专门介绍 PIC 单片机的 C 语言编程。读者对象主要是对 PIC 单片机感兴趣的初学者或准备用 PIC 单片机进行产品开发的工程师,对已经熟悉 PIC 单片机开发的工程师也可以作为设计时的参考。

图书在版编目(CIP)数据

PIC[®] 单片机入门与实战 / 张明峰编著. — 北京: 北京
航空航天大学出版社, 2004. 9

ISBN 7-81077-508-1

I. P… II. 张… III. 单片微型计算机—基本知识
IV. TP368.1

中国版本图书馆 CIP 数据核字(2004)第 051306 号

PIC[®] 单片机入门与实战

张明峰 编著

责任编辑 胡晓柏

责任校对 陈 坤

*

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

北京市松源印刷有限公司印装 各地书店经销

*

开本:787 mm×960 mm 1/16 印张:27.75 字数:622 千字

2004 年 9 月第 1 版 2004 年 9 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-508-1 定价:39.00 元

前言

单片机,又称微控制器,对于现在从事电子技术应用的工程师们来说已经是很平常的一类器件了。在单片机应用逐渐渗入到各行各业的各类产品中的同时,制造单片机的半导体厂家也从十几年前屈指可数的几家发展到现在的几十家甚至更多。不同厂家提供了基于不同架构平台,具备不同功能特点的单片机,这就使得工程师们可以按照具体设计的要求,挑选一款最合适的芯片进行系统开发,在满足功能要求的同时最大限度地降低成本,亦即提高了自己产品的性能价格比。

在众多的单片机系列中,Microchip®公司生产的冠以 PIC®的 8 位单片机从以 Intel®公司 51 系列和 Motorola®公司 68 系列占绝对主导地位的 8 位单片机市场中脱颖而出。从 1989 年公司成立生产第一款 8 位单片机时的名不见经传,每年都持续快速成长,到 2003 年跃居 8 位单片机年出货量全球排名第一。要知道,PIC 单片机从一开始就是一款与众不同的芯片,它没有兼容当时任何一种市场上流行的单片机结构。在设计工程师们认识和应用的惯性下,在激烈的市场竞争的夹缝中,PIC 最终能取得如此的成功,其中必有道理。

Microchip 公司是第一个在 8 位单片机结构中实现精简指令集的厂家,并采用了程序空间和数据空间完全分离的哈佛总线结构。这种体系结构大大降低了 PIC 单片机的总体成本,同时提高了运行效率。另外,PIC 单片机不搞简单的功能堆积,Microchip 公司提供了不同系列不同型号的单片机。

前言

众多型号的单片机其运行性能有高有低,片上功能资源有多有少,但始终保持了高度一致的可移植性。这样,设计工程师就可以在最初定方案时挑选一款功能最恰当的芯片,随着开发的深入,能够随时方便地将整个设计移植到其它型号的单片机上,保证最终产品中使用的单片机不会有太多的资源浪费,提高了产品的性价比。

Microchip 公司的 PIC 单片机从一开始就走保证现场可编程灵活性之路。从早期的 OTP 芯片(用户一次可编程)到现在的 FLASH 芯片(闪存多次可编程),正好符合整个市场快速变化的需求。这也是 PIC 单片机能够取得成功的重要原因之一。现在的普遍情况是一个产品的生命周期短的只有几个月,长的也就是一两年。要想在变幻莫测的市场上获得成功,产品的上市时间和速度是必须要考虑的成本因素之一。仅从产品的上市速度和承担的风险这两方面考虑,以前靠掩膜定制单片机的做法已经不能适应这一现状,因此越来越多的用户转向使用 OTP 和 FLASH 形式单片机,以求加快产品的开发和上市速度,避免开发和生产中的风险,提高产品升级换代的灵活性。

另外,从产品的角度去看一个设计,必须要考虑售后的维护成本。这就要求作为控制核心的单片机具有非常高的可靠性,在各种不同的环境下都必须稳定可靠地工作。PIC 单片机能获得成功的另外一个重要原因就在于其高稳定性和可靠性。这与 Microchip 公司芯片设计的严格细致和制造过程的精益求精息息相关。

再来看国内单片机的应用现状,基本上还是以 51 内核系列为主。其实这也正常,51 系列在国内的根基非常深厚,现在学校的教学一般都是以 51 作为基本教材,51 系列芯片本身也在不断地变革和发展。随着市场的细分和产品设计所需,PIC 单片机在国内相关行业中的认知度迅速提高,其应用范围和市场占有率也随之上升。但基本上,PIC 不会是第一种被接触的单片机,我本人也不例外。我于 1999 年 6 月加入 Microchip 公司从事现场技术支持工作,在这之前,从来就没有用过 PIC 单片机。但经过一段时间在了解了 PIC 单片机的特点后,我就被其深深地吸引住了,时常发出这样的感叹:要是我以前做的一些设计换成现在的 PIC 单片机,系统的体积和成本要降低多少?可靠性又能提升多少?

几年来在具体从事 Microchip 产品技术支持工作的过程中,对现在工程师的状况也有了一些体会。绝大部分现在用 PIC 单片机的工程师都是从别的单片机体系结构转过来的,自然而然就有先入为主的惯性,在学习

PIC 单片机时自觉或不自觉地与其它单片机做简单的横向对比,而不注重掌握其内在的结构特点。其实这样做的结果是在用 PIC 单片机“模拟”其它单片机,根本就不能充分发挥出 PIC 单片机的固有特点和优点。在这个现实的世界里,几乎不可能有完美的东西,至少在众多型号的单片机中,没有一种型号是完美的,是可以用来包打天下的。作为我们设计工程师的职责就是选择一款最合适的芯片,充分发挥其固有的特点,实现产品的最优化设计。这就要求我们在进行设计时对芯片要有较深入细致的了解。

近几年来,针对 PIC 单片机的设计和应用,国内也陆续出版了一些参考书籍,这对 PIC 单片机的推广起了很大作用。在这一方面北航出版社走在了前面。但与介绍 51 系列的书籍相比,无论是数量还是质量方面都不能相提并论,这在一定程度上限制了 PIC 单片机的应用普及和技术提高。随着网络技术的发展,有了网上技术论坛,这给工程师们提供了一个可以相互交流的平台。我本人也热衷于参加各类技术讨论的活动,希望在技术上共同提高,但逐渐发现网上论坛中绝大部分的话题都是重复的、初级的,或基本概念性的问题。与其每次都要重复回答相同的问题倒不如先写下来,然后在需要时复制,或放到某一个地方作为参考。正好,北航出版社的胡晓柏编辑与我联系,约我写点关于 PIC 单片机开发和应用方面的东西,这与我前面的想法不谋而合,于是就有了这本书。

至于具体选题,考虑良久还是决定以 PIC 中档系列单片机为基础。因为 PIC 中档系列单片机的型号众多,价格适中,在国内的应用面最广;本地化的开发工具支持力度也最大,对广大初学用户而言进入门槛较低。针对编程语言的介绍将重点强调汇编语言编程的技巧,以便使读者更深入地体会 PIC 单片机体系结构和精简指令的优点。

作为技术工作者之一,我写这本书的出发点是想和大家一起分享在实际工作过程中积累的一些比较实用的经验体会,而不是什么教科书。对芯片数据手册上已经介绍的或别的参考书籍中已经讲到的内容将不作太多重复的介绍,大家在具体开发过程中一定要仔细阅读厂家提供的芯片数据手册。本书中如果出现与芯片数据手册描述相矛盾的地方,以数据手册为准。书中的绝大部分代码实例都通过实际电路验证,但不能保证没有错误,读者在引用这些代码或电路时必须自己负责验证,本人恕不承担任何连带责任。

感谢北航出版社的马广云博士和胡晓柏编辑在这整个写书过程中给我的支持和关心。也感谢安捷伦公司提供混合信号示波器,书中所有的电路波形即用安捷伦 54622D 混合信号示波器实测得到。由于我工作性质的原

因决定了此书只能利用我个人的业余时间编写,因此原本属于我和女儿嬉戏的时间就全部变成了我写书的时间,心里颇感内疚。我六岁的女儿也非常乖巧懂事,当我坐在电脑前敲击键盘写书时,从不会跑过来撒娇纠缠,因此我打心底里感谢她,爱她。

诚然,本人的经验和水平有限,时间上也比较匆忙,书中的错误和不妥之处在所难免,恳请广大读者批评指正。我的电子邮件是:Paul. Zhang@microchip. com。另外,有兴趣的读者可以登录 <http://www.21icbbs.com> 中嵌入式系统开发的 PIC 单片机论坛,或 <http://forum.microchip.com> 进入微芯公司的中文论坛,与我分享您相关的知识、经历及阅读心得。也可以从微芯公司网站主页 <http://www.microchip.com> 上获取更多资料。

张明峰

2004 年 5 月于上海

版 权 声 明

本书引用以下资料已得到其版权所有 者 Microchip Technology Inc. (美国微芯科技公司) 的授权。

PICmicro® Mid-Range MCU Family Reference Manual (DS33023A)

再版上述资料须经过其版权所有 者 Microchip Technology Inc. 的许可。
所有权保留。未得到该公司的书面许可, 不得再版或复制。

商 标 声 明

以下图案是 Microchip Technology Inc. 在美国及其他国家的注册商标:



以下文字是 Microchip Technology Inc. 的注册商标(状态:®):

Accuron, AmpLab, dsPIC, ENVOY, FilterLab, KEELog, KEELog Logo, Microchip Logo, Microchip Name and Logo, microID, MPLAB, MXDEV, MXLAB, PIC, PICmicro, PICMASTER, PICSTART, PowerSmart, PRO MATE, rPIC, SEEVAL, SmartShunt, *The Embedded Control Solutions Company*

以下文字是 Microchip Technology Inc. 的商标(状态:TM):

Analog-for-the-Digital Age, Application Maestro, dsPIC(development tools only), dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, ICEPIC, ICSP or In-Circuit Serial Programming, Migratable Memory, MPASM, MPLAB Certified Logo, MPLIB, MPLINK, MPSIM, Now Design It, PICDEM, PICDEM.net, PICKit, PICLAB, PICTail, PowerCal, PowerInfo, PowerMate, PowerTool, QuickASIC, rLAB, rPICDEM, Select Mode, Smart Seril, SmartTel, The Emerging World Standard, Total Endurance

以下文字是 Microchip Technology Inc. 的服务标记(状态:SM):

SQTP

以下所有其他商标的版权归各自公司所有:

PICC, PICC Lite, PICC-18, CWPIC, EWPIC



第1章 PIC 单片机的结构和指令系统

1.1 Microchip 单片机产品概述	1
1.2 PIC 中档单片机的基本特性	2
1.2.1 内核结构简述	3
1.2.2 指令流水线概念	6
1.2.3 寄存器文档概念	6
1.2.4 单字指令概念	7
1.3 PIC 中档系列单片机的指令集	9
1.3.1 字节操作指令详述	9
1.3.2 位操作指令详述	19
1.3.3 立即数操作指令详述	21
1.3.4 控制指令详述	24
1.4 PIC 单片机数据寄存器的寻址	28
1.4.1 数据寄存器直接寻址	28
1.4.2 数据寄存器间接寻址	31
1.5 PIC 单片机程序空间的寻址问题	33
1.5.1 PIC 单片机程序空间的直接跳转	33
1.5.2 PIC 单片机程序空间的间接跳转	36
1.6 PIC 中档系列单片机的中断机制	39
1.6.1 中断响应时的入口地址(中断向量)	40
1.6.2 中断优先级控制	40
1.6.3 中断现场保护和恢复	41

1.6.4 中断控制逻辑	44
--------------------	----

第2章 PIC 单片机的特色

2.1 PIC 单片机的工作时钟模式	47
2.1.1 石英晶体或陶瓷谐振器振荡	47
2.1.2 外部时钟	52
2.1.3 外部 RC 振荡	53
2.1.4 内部 RC 振荡	55
2.1.5 振荡模式的确定	56
2.2 PIC 单片机的复位过程	56
2.2.1 上电复位(POR)	56
2.2.2 低电压检测复位(BOR 或 BOD)	58
2.2.3 外部复位	60
2.2.4 看门狗复位	60
2.2.5 判别复位情形	61
2.3 PIC 单片机的低功耗休眠	62
2.4 PIC 单片机在线串行编程	64
2.5 PIC 单片机的芯片配置字	67

第3章 PIC 单片机的开发环境

3.1 MPLAB-IDE 的功能组成	70
3.1.1 MPLAB-IDE 综述	70
3.1.2 MPLAB-IDE 的安装	71
3.1.3 MPLAB-IDE 项目管理	72
3.1.4 MPLAB-IDE 源程序编辑	76
3.2 汇编编译器 MPASM™ 及其汇编程序编写的基本规则	77

目 录

3.2.1	汇编编译器 MPASM 简介	77	4.3.7	ICE2000 高级触发功能	140
3.2.2	MPASM 的语法简要说明	78	4.3.8	ICE2000 触发输入/输出同步功能	148
3.2.3	MPASM 的伪指令	79	4.3.9	ICE2000 代码执行覆盖检查功能	149
3.2.4	MPASM 内的直接运算符	86	4.3.10	关于 ICE2000 的其他有用信息	151
3.2.5	MPASM 的宏指令	88	4.4	芯片编程烧写工具 PICSTART-Plus 和 PROMATE-II	153
3.2.6	程序模板	95	4.4.1	PICSTART-Plus	154
3.2.7	IDE 内 MPASM 编译环境的设定	97	4.4.2	PROMATE-II	157
第 4 章 PIC 单片机的开发工具			4.5	第三方开发工具	159
4.1	软件模拟器 MPLAB-SIM	101	4.5.1	第三方仿真工具	159
4.1.1	软件模拟器的安装和启动	101	4.5.2	第三方编程工具	159
4.1.2	软件模拟器的程序运行控制方式	102	第 5 章 PIC 单片机的 I/O 口及其灵活应用		
4.1.3	断点的设定和取消	104	5.1	PIC 单片机 I/O 口的基本特性	160
4.1.4	光标控制程序运行	105	5.1.1	完全双向的 I/O 引脚	161
4.1.5	软件模拟器的跑表功能	105	5.1.2	端口输入/输出设定	161
4.1.6	运行结果的观察窗口	107	5.1.3	端口的读和写	161
4.1.7	模拟器的信号激励功能	111	5.1.4	端口的输出驱动能力	162
4.1.8	软件模拟器的局限	117	5.1.5	端口的“读—修改—写”问题	163
4.2	在线调试器 MPLAB-ICD2	117	5.1.6	引脚端口的 ESD 保护	165
4.2.1	ICD2 的基本功能及其优缺点	118	5.2	PIC 单片机 I/O 口的设定方法	166
4.2.2	ICD2 与计算机的连接方式	119	5.3	PIC 单片机 PORTA 端口	167
4.2.3	ICD2 与目标板的连线方式	120	5.3.1	PORTA 的特点	167
4.2.4	ICD2 和目标板的供电问题	122	5.3.2	RA4 的特殊性	168
4.2.5	MPLAB-IDE 环境下 ICD2 的启用	123	5.4	PIC 单片机 PORTB 端口	169
4.2.6	用 ICD2 进行系统调试	126	5.4.1	PORTB 综述	169
4.2.7	ICD2 的辅助信息	128	5.4.2	RB0/INT 引脚的中断功能	170
4.2.8	用 ICD2 让芯片独立工作	130	5.4.3	PORTB 的 RB7~RB4 引脚状态变化中	170
4.3	硬件仿真器 MPLAB-ICE2000	130	5.5	PIC 单片机 PORTC 端口	176
4.3.1	ICE2000 的基本功能	131	5.6	PIC 单片机 PORTD 和 PORTE 端口	177
4.3.2	ICE2000 与计算机的连接	132	5.7	PIC 单片机 I/O 引脚的应用技巧	178
4.3.3	ICE2000 与目标板的连接	132	5.7.1	输入输出分时复用	179
4.3.4	启动 ICE2000 仿真器	133	5.7.2	普通 I/O 引脚测电压值	180
4.3.5	ICE2000 的基本调试功能	136			
4.3.6	ICE2000 对代码执行的跟踪功能	136			

5.7.3 普通 I/O 引脚测电阻值	184	7.5.1 输入电压信号的幅度	228
5.7.4 普通 I/O 引脚进行超限电压检测	190	7.5.2 输入电压信号的内阻	229
第 6 章 PIC 单片机的定时器资源及其应用		7.5.3 输入信号的抗混叠滤波	230
6.1 定时器 TMR0	192	7.6 输入信号的采样过程	232
6.1.1 TMR0 综述	192	7.7 AD 转换时钟的选择	234
6.1.2 与 TMR0 相关的控制寄存器	193	7.8 参考电压的选取	235
6.1.3 TMR0 的运作	194	7.9 休眠时进行 A/D 转换	236
6.1.4 TMR0 中断	195	7.10 特殊事件触发进行 A/D 转换	238
6.1.5 TMR0 用于外部脉冲计数	197	7.11 10/12 位分辨率的 ADC 模块	241
6.1.6 TMR0 的预分频器	198	7.11.1 10/12 位分辨率 A/D 模块的工作原理	241
6.1.7 TMR0 应用注意事项	199	7.11.2 ADCON1 寄存器的内容扩充	242
6.2 定时器 TMR1	205	7.11.3 A/D 转换结果的格式问题	243
6.2.1 TMR1 综述	205	7.11.4 参考电压的考虑	244
6.2.2 与 TMR1 相关的控制寄存器	206	7.11.5 采样时间的考虑	245
6.2.3 TMR1 作为定时器	207	7.12 8/14 引脚单片机上的 A/D 模块	245
6.2.4 TMR1 作为同步计数器	207	7.12.1 8/14 引脚单片机 A/D 模块综述	246
6.2.5 TMR1 作为异步计数器	208	7.12.2 ADCON0 寄存器的变化	246
6.2.6 TMR1 的内部振荡器	208	7.12.3 ADCON1 寄存器的变化	247
6.2.7 TMR1 与 CCP 模块配合	208	7.12.4 ANSEL 寄存器的引入	248
6.2.8 TMR1 的具体应用	209	7.12.5 其他注意事项	249
6.3 定时器 TMR2	216	7.13 A/D 模块的应用技巧	249
6.3.1 TMR2 综述	216	7.13.1 模拟/数字引脚分配时的问题	249
6.3.2 与 TMR2 相关的控制寄存器	216	7.13.2 合理实现采样时间	252
6.3.3 TMR2 的工作方式	218	7.13.3 转换结果的软件数字滤波方法	253
6.3.4 TMR2 的应用	218	第 8 章 PIC 单片机的 USART 通信模块及其使用	
第 7 章 PIC 单片机的模/数转换模块及其使用		8.1 USART 简介	258
7.1 PIC 单片机片上 ADC 模块综述	221	8.2 USART 模块关键寄存器介绍	258
7.2 ADC 相关控制寄存器介绍	222	8.2.1 TXSTA 数据发送控制及状态寄存器	258
7.2.1 ADCON0 控制寄存器	222	8.2.2 RCSTA 数据接收控制及状态寄存器	259
7.2.2 ADCON1 控制寄存器	223	8.2.3 SPBRG 波特率控制寄存器	261
7.2.3 ADRES 结果寄存器	224	8.2.4 相关的中断控制寄存器	261
7.2.4 A/D 转换中断相关的寄存器	224		
7.3 设定模拟信号输入引脚	225		
7.4 A/D 转换过程说明	226		
7.5 被测输入信号的参数要求	228		

目 录

8.2.5 TXREG 和 RCREG 寄存器	261	9.2.1 SSPSTAT 同步接口状态和控制寄存器	315
8.3 USART 波特率设定	262	9.2.2 SSPCON 同步接口控制寄存器	317
8.3.1 异步通信时的波特率设定	262	9.2.3 SSPBUF 数据寄存器	319
8.3.2 同步通信时的波特率设定	267	9.2.4 SSPADD 地址寄存器	319
8.3.3 SPBRG 寄存器的写效应	267	9.2.5 SSPCON2 同步接口控制寄存器 2	319
8.4 USART 模块的异步通信	267	9.2.6 与 SSP 相关的中断控制寄存器	320
8.4.1 异步通信数据格式	267	9.3 SSP 模块的 SPI 通信方式	321
8.4.2 异步通信数据发送过程	268	9.3.1 使用 SPI 通信接口	321
8.4.3 异步通信数据接收过程	273	9.3.2 SPI 通信引脚设置	323
8.4.4 常用的异步通信的协议	279	9.3.3 SPI 通信时的连接方式	323
8.5 USART 模块同步通信主模式	283	9.3.4 SPI 主模式通信	324
8.5.1 同步通信主模式发送	283	9.3.5 SPI 从模式通信	325
8.5.2 同步通信主模式接收	284	9.3.6 SPI 从模式选择控制通信	326
8.6 USART 模块同步通信从模式	285	9.3.7 SPI 应用实例介绍	327
8.6.1 同步从模式数据发送	285	9.4 SSP 和 MSSP 模块的 I ² C 模式	337
8.6.2 同步从模式数据接收	286	9.4.1 I ² C 从模式	339
8.7 通信时的数据缓冲技巧	286	9.4.2 MSSP 模块的 I ² C 主模式	342
8.7.1 接收环形 FIFO 缓冲队列	286	9.4.3 I ² C 通信程序范例	344
8.7.2 发送缓冲队列	287	9.4.4 I ² C 总线硬件处理	357
8.8 通信数据的校验方法	288	9.5 软件实现同步串行通信	358
8.8.1 奇偶位校验	288	9.5.1 软件实现 SPI 主模式通信	359
8.8.2 累加和校验	289	9.5.2 软件实现 I ² C 主模式通信	361
8.8.3 循环冗余校验(CRC)	290	第 10 章 PIC 单片机的 CCP 模块及其应用	
8.9 报文通信时的简单协议	293	10.1 CCP 模块简介	367
8.10 异步串行通信的波特率自适应技术探讨	298	10.2 与 CCP 模块相关的控制寄存器	368
8.10.1 标准波特率穷举法	298	10.3 CCP 模块的输入捕捉模式	370
8.10.2 码元宽度实时检测法	301	10.3.1 CCPx 引脚配置	371
8.11 多机通信的寻址方式	304	10.3.2 改变不同的捕捉模式	371
8.11.1 9 位数据传输	305	10.3.3 脉冲沿捕捉时的预分频	371
8.11.2 数据包传输	305	10.3.4 休眠时进行捕捉	371
8.12 软件实现异步串行通信	305	10.3.5 复位时的情形	372
8.12.1 三倍速采样法	306	10.3.6 捕捉模式应用实例介绍	372
8.12.2 起始位中断捕捉定时采样法	313	10.4 CCP 模块的比较输出模式	374
第 9 章 PIC 单片机同步串行接口及其应用		10.4.1 比较模式下的 CCPx 引脚	374
9.1 同步串行接口简介	315		
9.2 SSP 接口模块控制寄存器	315		

10.4.2 软件中断模式	375	11.6.2 调用层次的控制	408
10.4.3 特殊事件触发	375	11.6.3 函数类型声明	409
10.4.4 休眠时的比较输出状态	375	11.6.4 中断函数的实现	409
10.4.5 比较输出模式应用实例	375	11.6.5 标准库函数	411
10.5 CCP 模块的 PWM 模式	381	11.7 PICC 定义特殊区域值	411
10.5.1 PWM 周期	382	11.7.1 定义工作配置字	411
10.5.2 PWM 占空比(高电平持续宽度)	382	11.7.2 定义芯片标记单元	412
10.5.3 占空比调整的绝对分辨率	383	11.8 MPLAB-IDE 中实现 PICC 的编译选项设 置	413
10.5.4 PWM 工作模式设定	384	11.8.1 选择单片机型号	413
10.5.5 PWM 应用实例	384	11.8.2 PICC 普通编译选项(General)设定	413
第 11 章 PIC 单片机的 C 语言编程		11.8.3 PICC 全局选项设定(PICC Global)	414
11.1 PIC 单片机 C 语言编程简介	394	11.8.4 C 编译器选项设定(PICC Compiler)	415
11.2 Hitech-PICC 编译器	395	11.8.5 连接器选项设定(PICC Linker)	417
11.3 MPLAB-IDE 内挂接 PICC	395	11.8.6 汇编器选项设定(PICC Assembler)	419
11.4 PIC 单片机的 C 语言源程序基本框架	397	11.9 C 和汇编混合编程	420
11.5 PICC 中的变量定义	398	11.9.1 嵌入行内汇编的方法	420
11.5.1 PICC 中的基本变量类型	398	11.9.2 汇编指令寻址 C 语言定义的全局变量	421
11.5.2 PICC 中的高级变量	399	11.9.3 汇编指令寻址 C 函数的局部变量	422
11.5.3 PICC 对数据寄存器 bank 的管理	400	11.9.4 混合编程的一些经验	425
11.5.4 PICC 中的局部变量	400		
11.5.5 PICC 中的位变量	401		
11.5.6 PICC 中的浮点数	402		
11.5.7 PICC 中变量的绝对定位	403		
11.5.8 PICC 的其他变量修饰关键词	403		
11.5.9 PICC 中的指针	405		
11.6 PICC 中的子程序和函数	408		
11.6.1 函数的代码长度限制	408		

后 记

参考文献

第 1 章

PIC 单片机的结构和指令系统

1.1 Microchip 单片机产品概述

现今市场上常见的各类 8 位单片机,从存储器组织体系结构角度区分,不外乎有“冯·诺曼”和“哈佛”结构两种。“冯·诺曼”结构原理如图 1-1 所示,其程序存储器和数据存储在物理分配上属于同一空间。因为是 8 位机,即数据总线的宽度为 8 位,所以读取指令和存取数据只能共用同一 8 位总线,数据流量受到一定的限制。

“哈佛”结构原理如图 1-2 所示,其程序存储器和数据存储在物理空间上完全独立,读取指令的总线和存取数据的总线也完全分开。这样的结构让指令总线和数据总线的宽度不一样成为可能;数据总线宽度一样还是 8 位,但针对程序空间读取指令的总线宽度可以是任意合适的位数。另外,很关键的一点是由于总线独立,读取指令和存取操作数就可以同时进行,即可以引入指令的流水线机制,以提高单片机内的数据流量,提高代码运行效率。

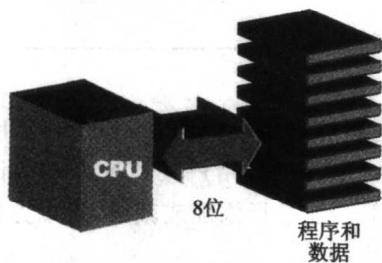


图 1-1 “冯·诺曼”结构示意图

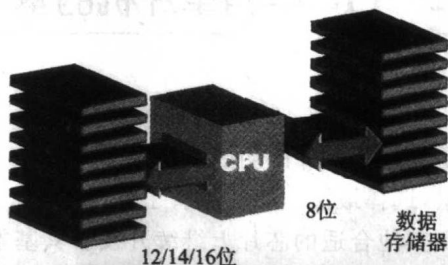


图 1-2 “哈佛”结构示意图

1 PIC 单片机的结构和指令系统

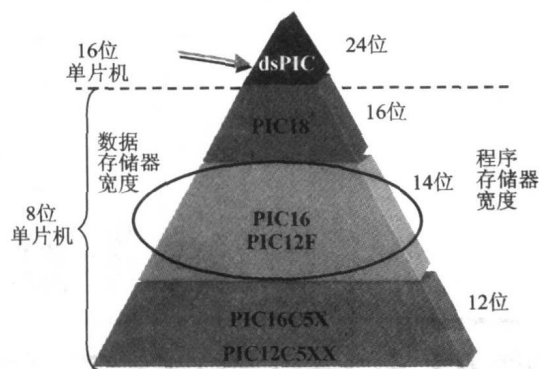


图 1-3 PIC 系列单片机金字塔

Microchip 公司推出的 PIC 系列 8 位单片机内核正是基于“哈佛”结构。在其整个 8 位机家族中分低/中/高三档，对应的内部指令宽度分别是 12/14/16 位，见图 1-3 金字塔形。12 位指令宽度的低档单片机结构简单，价格低廉，适用于一些控制任务相对简单且对价格敏感的低端消费类产品，代表型号如 PIC16C5x 系列的 PIC16C54C、PIC16C57C 等；PIC12C5xx 系列的 PIC12C508A、PIC12C509 等。14 位指令宽度的中档单片机是目前型号最丰富的一个系列，其片内功能模块种类其

全，组合灵活多变，价位适中，应用领域及其广阔，代表型号如 PIC16F7x 系列的 PIC16F73/74/76/77、PIC16F87xA 系列的 PIC16F873A/874A/876A/877A 等。16 位指令宽度的高档单片机在命名上以 PIC18 开头，整个内核体系在 PIC16 系列基础上有很大提高，最高指令流可达 10 MIPS（单条指令执行时间为 100 ns），达到部分 16 位单片机的性能，一般用于较高端的产品设计，是 Microchip 今后在 8 位单片机上发展的主要方向。dsPIC® 系列是 Microchip 最新推出的 16 位高性能单片机，其内核中包含有一个完整的定点 DSP，数据运算和处理能力大大增强，应用领域主要定位于汽车电子、复杂的电机控制、数据压缩和信号分析等高端场合。

尽管 PIC 单片机分那么多种类，但不同家族不同型号间有着无缝的迁徙路径。这完全得益于 PIC 单片机内核设计的一致性，使其指令上下兼容，具备相同的功能模块操控方式，更重要的是其表现出的电气特性也一致。目前，国内单片机的应用方向还主要定位在中低端的产品设计中，自然 PIC16 系列成为广大设计工程师的关注焦点。本书内容就以 PIC 家族中最典型的中档系列为立足点展开深入讨论，但相关话题对低档和高档系列的开发应用也有借鉴意义。

1.2 PIC 中档单片机的基本特性

所有 PIC12/16 中档单片机都有一个同样的运算控制和执行内核，各个型号所不同的是其配属的外围功能模块各不相同。这样做的目的是可以让用户按照具体产品设计的要求，选择最恰当的一款着手设计工作。按设计任务对功能进行添加或裁减，可以方便地移植到其他合适的芯片上继续开发。其基本特性包括：

- “哈佛”结构，功能强大的 RISC 单片机内核；

1 PIC 单片机的结构和指令系统

- 8 层硬件堆栈用于保护和恢复程序计数器；
- 工作频率 0~20 MHz, 用户可选的多种时钟模式；
- 引入指令执行流水线机制, 指令顺序执行时 1 条指令只需 1 个指令周期, 程序分支跳转需 2 个指令周期；
- 全部 35 条指令, 每条指令占 1 个字, 程序字长 14 位；
- 程序空间最大物理可寻址范围 8 192(8K) 程序字；
- 片上寄存器最大物理可寻址范围 512 字节, 使用寄存器文档的概念；
- 丰富的外围功能模块；
- 片内或片外多种信号中断机制；
- 片上具备独立看门狗。

1.2.1 内核结构简述

由图 1-4 可见, PIC 单片机内部指令空间和数据空间(RAM 寄存器)完全独立。程序存储器是由一组 EPROM 或 FLASH 存储单元实现的, 用于存放指令代码的空间, 最终的用户程序将通过烧写器编程烧入固化于此。EPROM 类的单片机一般都是一次可编程(OTP), 适用于稳定的批量生产; 也有特殊带石英窗口封装的开发芯片提供, 用户可以用紫外线擦除后进行多次烧写。FLASH 类的单片机可以电擦除, 重复烧写次数大于 10 万次, 无论是试验开发还是大量生产都非常灵活方便。某些增强型 FLASH 的芯片其代码空间在程序运行时还可以由代码控制自编程, 可以方便实现系统远程在线升级。Microchip 目前的产品发展战略是廉价可靠的 FLASH 工艺, 故大量的 PIC 单片机都将会是 FLASH 类型。作为产品开发, FLASH 芯片应该是首选。

程序计数器 PC 用来控制指令的执行顺序。在单片机内核设计时, PC 计数器的宽度最多为 13 位, 故理论上能执行的指令代码最多为 8K 字。对于某一特定型号的单片机, 其片内并不一定有全部 8K 字的程序空间, 有可能指令空间只有 1K/2K/4K 字不等。此时, 指令寻址就只需 PC 的若干低位数据即可(高位无效)。指令的运行永远从当前 PC 指向的程序单元读取指令, 读取当前指令后 PC 自动加 1, 准备读取下一条指令。如果所有的指令都是顺序执行的, 当执行到程序空间的最后一条指令后, PC 会自动回 0, 程序将从地址 0x000 处重新开始。

例如: 单片机为 PIC16C712, 其程序空间为 1K(1 024) 字, 最后一条指令的地址是 0x3FF。PC 计数器内只有低 10 位有效, 虽然 $0x3FF + 1 = 0x400$, 但只取低 10 位的值, 0x400 即等同于 0x000。

单片机为 PIC16F73, 其程序空间为 4K(4 096) 字, 最后一条指令的地址是 0xFFFF。PC 计数器内只有低 12 位有效, 虽然 $0xFFFF + 1 = 0x1000$, 但只取低 12 位的值, 0x1000 即等同

1 PIC 单片机的结构和指令系统

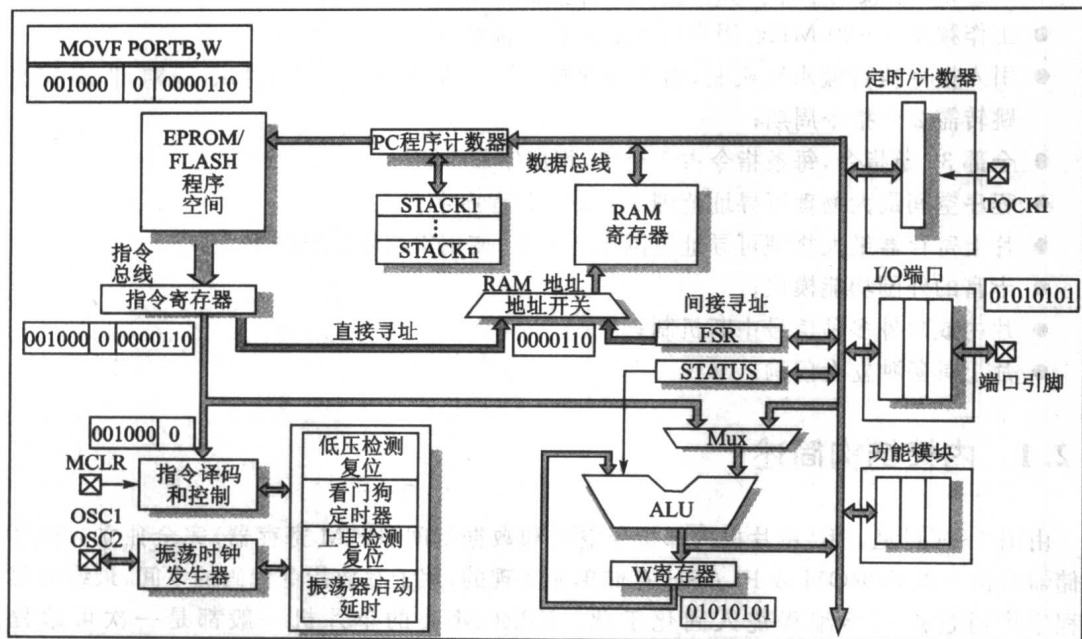


图 1-4 PIC 单片机内核结构图

于 0x000。

通过指令的执行可以强行改变 PC 计数器的值，这就是程序运行时的分支或跳转。任何时候发生任何形式的复位，PC 值都被置为 0x000；中断入口地址是 0x004。

在图 1-4 中还有一部分资源和 PC 计数器密切相关，那就是堆栈区。PIC 单片机内的堆栈是独立于任何其他存储单元的一组特殊存储区域，中档系列其堆栈有 8 层深度。堆栈的每个存储单元数据位宽度和 PC 一致，惟一用途就是在发生子程序调用或中断跳转时，把当前 PC 值（即返回地址）压栈保存在栈顶，原先栈顶和其下单元的内容被顺序往下移一个单元位置，最后一个单元的内容被丢弃；当子程序调用结束返回或中断结束返回时，从堆栈中弹出最顶部的值到 PC 中，顶部其下的单元内容顺序往上移一个单元，程序从刚才中断处继续往下运行。

例如：例 1-1 的代码在地址为 0x0122 处有指令要调用子程序，调用前堆栈内的内容如图 1-5(a) 所示。当执行了子程序调用的指令后，其下一条返回地址被压栈，堆栈的内容变成如图 1-5(b) 所示，然后 PC 赋值为 0x0200，程序就跳到地址 0x0200 处执行子程序 SUB1；SUB1 处只有一条返回指令，子程序 SUB1 返回指令执行完毕后，栈顶的内容弹出到 PC，返回后 PC=0x0123，程序从地址 0x0123 处继续执行，最后堆栈的内容如图 1-5(c) 所示。