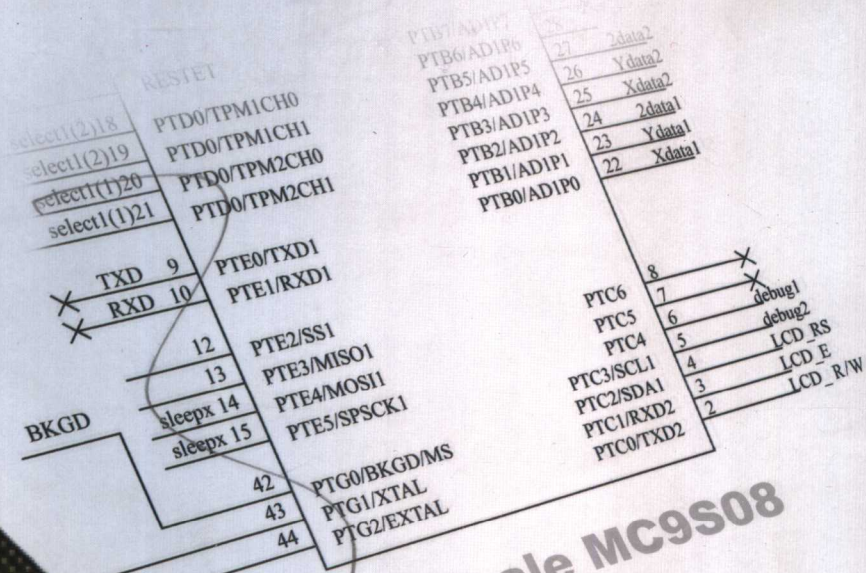


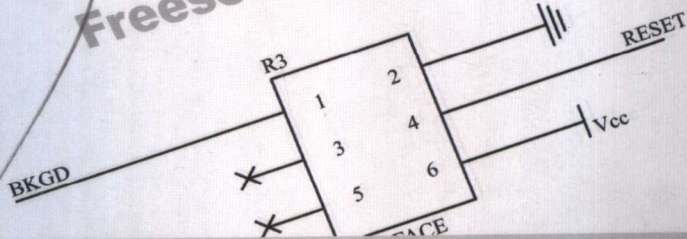
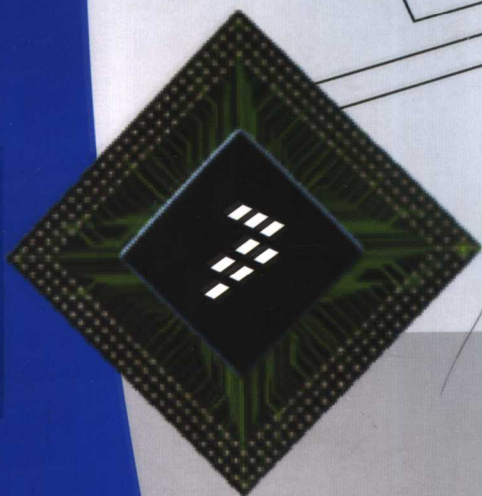


单片机认识与实践

邵贝贝 龚光华 等著



Freescale MC9S08



从零学习
搭建自己的开发系统

TP368.1
278

单片机认识与实践

邵贝贝 龚光华 等著

北京航空航天大学出版社

内 容 简 介

本书前8章是为高等院校电类低年级本科生“单片机认识与实践”课程撰写的教材,目的是使学生在尚不具备微机原理等相关知识的情况下也能尽快入门,尽早上手开始相关实践活动。书中强调单片机应用的多样性和个性化。从补充相关硬件知识开始,讲解如何设计基本硬件系统,从而在不使用仿真器的情况下搭建基本硬件开发条件;从编写简单的单片机监控程序开始,讲解如何使用C语言进行单片机编程,从而建立软件的开发环境,即从硬件、软件两个方面引导读者快速入门。后4章适用于嵌入式应用工程师,也可供高校研究生参考,首先讲述如何在8位单片机上运行嵌入式实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$,并以C语言的形式给出监控程序源代码,搭建起完整的嵌入式软件开发环境;然后以最新出现的S08单片机为例,提供各I/O模块的设备驱动程序;最后讨论应用系统的低功耗设计方法。书中还介绍目前先进的单片机BDM调试方式,这种在线开发方法同样适用于16位、32位单片机。

本书可作为高等院校电类相关专业本科生或研究生单片机课程的教材,也可作为单片机开发工程技术人员以及单片机爱好者的参考用书。

图书在版编目(CIP)数据

单片机认识与实践/邵贝贝等著. —北京:北京航空航天大学出版社,2006.8

ISBN 7-81077-797-1

I. 单… II. 邵… III. 单片微型计算机
IV. ①TP368.1

中国版本图书馆CIP数据核字(2006)第090302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。

单片机认识与实践

邵贝贝 龚光华 等著

责任编辑 孔祥燮

*

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:787mm×960mm 1/16 印张:23 字数:515千字

2006年8月第1版 2006年8月第1次印刷 印数:5000册

ISBN 7-81077-797-1 定价:32.00元

前 言

本书是为高等院校电类低年级本科生“单片机认识与实践”课程而撰写的,讲述的是 8 位单片机的入门知识和开发方法。过去,我们电类本科的嵌入式应用课程是从大学三年级开始的,此时学生已经掌握了较多的电路方面的知识。为加强学生的实践环节,我们将单片机课程提前到大学二年级,以便在后两年的本科学习期间有更多的动手实践机会。这就要求本教材能使学生在不完全掌握数字电路、微机原理等基础知识的情况下,也能尽快上手,开始他们的实践活动,以便在实践过程中提高自主学习的能力。

本书的初稿及书中提到的实验系统,已经在清华大学本科二年级选修课程中试用过,证明通过 32 学时的课程和实验,我们的学生完全可以达到单片机入门的水平。

本书同样也适用于从事嵌入式应用开发的工程师们,特别是本书的后 4 章,讲述了如何构建单片机的开发环境以及自己制作单片机开发工具,具有一定的难度。对那些熟悉 8051 单片机且用惯了仿真器进行开发的工程师,这里介绍的在线开发方法是一个完全不同的技术思路。作者相信,一旦掌握了开发环境的搭建过程,就不会再想回到仿真器的老路上去了。

作为教材,本书力求浅显易懂,不求面面俱到,重点放在那些最基本的入门知识上。本书将以一个典型的通用 8 位机为例,说明单片机应用开发的一般方法,使读者能尽快将一个 8 位单片机的基本系统跑起来。对于单片机的业余爱好者,也可以把本书当作一本启蒙读物,从而寻找到“玩”单片机的乐趣。

随着半导体技术的飞速发展,单片机(微控制器)也变得越来越强大,它集成了更多的复杂的功能模块;与此同时,其入门学习的难度也大大增加了,对于刚刚进入高等教育的本科生来说,其需要学习的相关背景知识太多,容易陷入到繁杂的技术泥潭中。8 位单片机则相对简练、适中,容易从整体上理解和把握;同时 8 位单片机外围接口丰富,片内 RAM 资源适中,Flash 足够大,有足够的资源可以让学生进一步地提高。本书介绍了开发单片机的基本方法和流程:从设计单片机的最小硬件系统开始,再设法对系统初始化,打通串行口,然后用汇编语言编写简单的监控程序;再介绍汇编语言是如何与 C 语言接口的,如何用 C 语言编写程序,直到移植实时操作系统(RTOS) $\mu\text{C}/\text{OS}$ 。该方法同样适于 16 位和 32 位单片机。

俗话说,授之以鱼不如授之以渔。教学不同于培训,培训则是“授之以鱼”,目的是让用户尽快上手,一周乃至几天就可以开始针对具体项目进行开发;教学是“授之以渔”,授之以解决问题的方法,培养解决问题的能力,从而引导学生入门。本教材没有过多地纠缠于各种接口模块的原理和具体使用方法,而是通过授课与实验,要求学生能做到3个看懂:看懂原文手册,看懂硬件原理图,看懂别人编写的程序。有了这样的能力,就可以很容易地理解和使用功能各异的接口模块,能够自行学习掌握一种原先不熟悉的单片机。教学要符合认知的客观规律,回想自己是怎样进入单片机嵌入式应用这一行业的——是由简单到复杂,由8位到16位、32位;是从会编写几行简单的汇编和点亮一只LED开始的。要切记防止浮躁,也要教导学生防止浮躁,所能做到的仅仅是领学生入门而已。能指出一个大致正确的方向,不误人子弟已属难得。后来成为高手和专家的,不是课堂培养的,是他们通过自己的艰辛努力实现的。

虽然单片机技术发展迅速,但开设一门单片机教学的课程并非易事,需要教师、教材、实验器材等,特别是实验台,需要较大的投入。教师要有相当的科研经验,用单片机做过研发,还要有一定的表达能力。作为教学,稳定至关重要,从作者过去的教学经验来看,开设一门新课,至少需要3年的时间才能稳定和完善课程的内容。要善于解决稳定和发展快这一对矛盾,可以在保持课程大部分内容相对稳定的前提下,每年根据最新的技术发展更新一小部分内容。本教材也是在这样的思路指导下,从先前的相对稳定的教学平台上,结合几年教学过程中心得体会和最新的单片机进展而编写的一本升级教材。

世界上著名的半导体公司几乎都有大学计划,在各大大学设立了实验室。清华大学的这类实验室就有不止40个,这是非常好的教学资源。清华 Motorola MCU/DSP 应用开发研究中心就设在作者所在的工程物理系实验室里。目前在中国,纳入 Motorola 公司的大学计划并得到支持的大学已超过30所。Motorola 的半导体部分分离成立了 Freescale(飞思卡尔)公司后,这种支持会做得更好。目前不少大学都使用的是 MC9HC08GP32 IDK 教学平台,这个实验平台由 GP32 单片机的基本系统子板和目标母板组成。教材是清华大学出版社于2001年出版的《Motorola 微控制器 MC68HC08 原理及其嵌入式应用》或2005年出版的修订本《Motorola (Freescale) 控制器 MC68HC08 原理及其嵌入式应用》。使用 HC08GP32 单片机教学有两个缺点:一是没有 BDM 调试方式(这种调试方法对于单片机开发方法有革命性的意义);二是 RAM 太小,只有 512 字节,运行 RTOS 没有应用价值。

本教材描述的 MC9S08GT60 的基本系统板与原来的 GP32 IDK 目标母板的子板接口是兼容的,将 GP32 IDK 升级到 S08 只需要将子板更换为 S08 的子板即可,所需的费用是很低的。通常,单片机的教学可稳定很多年,实验台也应尽量稳定的时间长一些。既然讲述的是开发方法,使用8位、16位还是32位单片机原理是相通的。我们选用先进的 S08 单片机,使原来基于 HC08 单片机教学的实验系统可以直接升级到 S08 系统。

S08 是 Freescale 公司产品路线图上最新一代的单片机系列,面向低端和低成本的应用,S08 单片机继承了 Freescale 公司在其 16/32 位单片机系列中发展起来的诸多新的技术特性,

同时又保持了其相对简洁的结构和低廉的价格。以介绍 S08 单片机作为大学生低年级的入门教材,无疑对学生今后的发展有益。从 2006 年开始举行的第一届“飞思卡尔”杯全国大学生智能车竞赛,是教育部支持的大学生五大赛事之一。该竞赛选用了 Freescale 公司的 S12 单片机。S12 是 16 位单片机,比 S08 单片机复杂。而智能车的控制程序并不复杂,对 CPU 运算能力的需求也不高,用 S08 单片机控制智能车完全可以达到同样的效果。预期,今后的竞赛也可能加入包括 S08 在内的更多选择。

S08 单片机有多个系列几十个品种,而且仍然在不断发展之中。最简单的 S08 单片机只有 8 个引脚或 16 个引脚、128 字节的 RAM 和 2 KB 的 Flash,但是这样的单片机却有很高的集成度,用引脚复用的方式提供多种功能模块,为应用提供了极大的灵活性。本教材中选用的 MC9S08GT60 有 44 个引脚,多个 I/O 模块,是 S08 单片机中资源比较丰富的一款。相对丰富的片内资源可以减少对实验教学带来的限制,学生不必考虑资源复用、代码优化等相对难度较高的问题,而可专注于比较单一的实验内容。在实际产品开发中,根据实际资源使用情况可以换用资源较少、成本较低的单片机。

从 HC08 到 S08,单片机的功能有所增强。除了运行频率的提高和功能模块的增加、增强外,需要在教学上加以强调的主要有 RTOS 和后台调试 BDM 技术两个方面。

由于单片机系统的需求越来越复杂,集成电路技术发展又非常快,片上存储器容量不断扩大,所以使得片上系统的开发使用嵌入式 RTOS 成为可能。对于 90% 以上的学生,计算机只是工具,而不能是其专业。与各行各业开发单片机应用的工程师一样,他们不可能将主要精力花在研究计算机上,而是要把主要精力放在所研究的专业上,他们需要的是把计算机专家们研究的成果直接拿来使用。RTOS 可以将一个复杂的应用分解成多个任务,从而简化应用程序的设计;RTOS 可以保证系统的实时性达到或接近理论上可以达到的最好水平。而多任务本身会带来一系列的问题,主要是任务间的竞争、死锁、优先级反转、任务间同步与通信等。这些问题由 RTOS 为用户考虑了,用户只要知道 RTOS 的原理,会使用即可。这里的所谓“会使用”,并非如同在 PC 上会使用 DOS 或 Windows 那种意义上的会用那么简单。因为 RTOS 要经过裁剪再固化到单片机中,RTOS 内核的源代码是与用户的应用程序一同编译,并融入用户应用代码之中的,故这里讲的是在理解 RTOS 原理基础上的会使用。因此单片机课程中引入 RTOS 概念是一大趋势。RTOS 是要占用一定系统资源的,典型地, $\mu\text{C}/\text{OS}$ 或 $\mu\text{C}/\text{OS}-\text{II}$ 大约需要占用 2 KB 的 RAM 资源,我们选用的 Freescale 公司的 MC9S08GT60 单片机有 4 KB 片内 RAM 资源,用于 RTOS 教学已足够好了。虽然是以 8 位单片机为例运行 RTOS,但方法是具有典型性和普遍性的。

随着单片机技术的进步,借用 JTAG 接口,在线调试单片机的方法越来越多地得到采用。MC9S12 单片机的 BDM 调试法能在单片机运行时对单片机动态调试,这在各类单片机中尚不多见,因此仅介绍在线调试原理和用法。对于片内 Flash 是空白的、新的单片机,总要有个简单的工具将监控程序下载下去。目前各种单片机都采用了 BDM (Background Debug

Mode)方式或使用 JTAG 接口。遗憾的是,目前对不同单片机,还没有统一的接口标准。虽然这类工具原理类似,都是通过单片机的某一个或几个不多的引脚,实现与 PC 机的通信,实现向 Flash 下载程序,但不同单片机的 BDM 工具差别很大。由于仿真实际上可以脱离硬件环境,在 PC 上进行,仿真后又可以通过 BDM 工具将程序下载到目标系统单片机的 Flash 中,故有人称这种 BDM 工具为仿真器。实际上这种 BDM 工具与仿真器完全是两个不同的概念,其价格远低于仿真器的价格。简单的 BDM 工具可以自己做,但需要熟悉相应的那种单片机。本教材中先介绍如何开发单片机,最后作为单片机应用的一个实例,介绍 BDM 调试工具和如何使用这种工具。因为 S08 单片机的 BDM 性能比 16 位 S12 的还要好,故也可以用 S08 单片机做成 S12 的 BDM 调试工具。这里讲述的开发方法的另一个优点是所有教学设备都可以自制。单片机的最小系统只有晶振、复位电路和 RS-232 驱动电路,成本很低;I/O 模块的实验台完全可以根据不同专业的需求自行设计,成本完全可控。最初仅需要一个 BDM 工具,而第二个 BDM 工具就可以自制。

4

本书第 1 章是单片机概论。介绍单片机的广泛选择和应用的个性化,然后选择一种有代表性的、技术是目前最先进的 8 位单片机 MC9S08GT60 为本教材实践环节的对象,简要说明其特点和开发思路,目的在于抵消长期以来使用 8051 及仿真器开发手段带来的负面效应,即以一种单片机加不同的接口芯片去适应各种应用,主张不使用仿真器的开发方法。

第 2 章用较少的篇幅、尽量通俗的语言讲述数字电路入门知识,以及如何用数字电路构建一个最简单的“概念 CPU”;从一个只有 15 条指令的概念 CPU,扩展到一个 8 位、256 条指令的真实 CPU。然后通过 CPU 从上电复位到执行程序的过程,说明电源、时钟在单片机中的作用,并给出与“驱动”相关的基本知识,旨在使那些没有学过数字电路的学生也能尽快开始单片机相关的实验,从而敢于动手触及单片机的硬件设计。

第 3 章讲述指令系统和汇编语言,是调试单片机硬件不可缺少的部分。由于指令太多,读者不易全面掌握,所以我们仅用调试硬件时最常用的十几条指令构成一些最有用的小程序,使读者入门,从读懂别人编写的程序开始,通过一学期的课程与实践,逐渐会用汇编语言编程。汇编语言对任何单片机的开发都是非常重要、必不可少的,是必须掌握的,但可以逐渐掌握。

第 4 章讲述如何设计一个单片机的基本硬件系统,即如何给单片机加上电源、时钟电路、复位电路和与人交互的串行通信驱动电路,让单片机能“跑”起来,并能听人指挥的基本硬件。复位向量概念的引入,使读者明白 CPU 是如何开始工作的。同时提及中断向量的概念,以备后用。

第 5 章从最简单的并行接口开始,重点介绍异步串行通信及 RS-232C 通信协议中与单片机开发有关的部分,讲述如何开启通信模块和初始化,以查询方式实现发送与接收,然后将接收改为中断方式。这个三步曲式的处理方法对单片机的任何 I/O 模块都有普遍意义。本章并不全面介绍 SCI 模块,只介绍实现与 PC 通信必须涉及的寄存器中最重要的那些位。同时告诉读者,如何将 PC 机设置成单片机系统的人机交互终端。以 SCI 中断为例,对中断做进

一步说明。

第6章讲述单片机中其他重要的 I/O 模块：并行 I/O、同步串行 SPI 通信模块、A/D 转换模块和定时器模块。本章介绍了各个模块的基本用法和寄存器设置，但考虑到篇幅的限制并没有将其详细展开。在定时器部分，由定时器衍生出来的输入捕捉、输出比较、PWM 仅简单提及，待用户真正用到时自行研究解决，不作为本教材重点。有了定时功能，单片机系统可完成的实验将大为丰富，对于开发读者对应用的想象力极为有利，同时进一步加深中断的概念，为以后实时操作系统的时标做准备。

第7章讲述如何使用 C 语言开发单片机应用软件（这里假设读者学过 C 语言）。本章重点讲述嵌入式系统用的 C 编译器与标准的 ANSI 的 C 编译器有什么不同，以及如何快速掌握一个从来没有用过的交叉 C 编译器的方法。如何建立单片机的 C 语言程序运行环境，重点在于与汇编语言的参数传递和相互调用关系。这一部分对任何公司的 8/16/32 CPU 都有普遍的意义。

第8章给出与本教材配套的教学实验板上的实验，介绍典型单片机开发软件的安装使用方法。每个实验都给出了实验指导和示范代码。

第9章描述监控程序，是实现 CPU 与人通信的基本软件，也是单片机基本系统的软件部分。把监控程序下载到单片机中，单片机的基本系统就完成了。本章讲述监控程序至少应具备的基本功能和写监控程序的基本思路。而将我们用 C 语言编写好的一个监控程序的源代码清单放在附录 A 中，希望读者从读懂监控程序开始，逐渐学会用汇编语言编写单片机硬件相关部分的代码，进而有可能自己建立一套单片机的开发环境。

第10章介绍单片机的 BDM 调试方法，介绍两个单片机间是如何通过单线通信的，这是本书的重点和亮点，也是单片机调试技术的革命性发展。正是 BDM 技术，使单片机的在线动态调试成为可能。同时也告诉读者，单片机的程序是如何下载到单片机中的。本章介绍了单片机片内 Flash 存储器擦除、读/写方面的知识。具备这些知识，读者可以自己动手做一只 BDM 开发工具。S08 单片机的 BDM 方法来源于 16 位单片机 S12，并优于 S12 的方法。单片机的 BDM 方法大都借用测试超大规模集成电路用的 JTAG 接口。JTAG 接口本身有国际标准，而借用 JTAG 调试单片机则各公司尚无统一标准。因此该方法具有普遍意义，但通信規約不具有普适性。

第11章讲述如何将专门面向嵌入式应用的、源码公开的实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$ 移植到 8 位机 S08GT60 上。至此，一个完整的 8 位单片机应用开发环境就搭建起来了。

第12章在读者具备以上单片机基本知识的基础上，讲述单片机应用系统设计中涉及的一些深层次的问题，如时钟电路对系统噪声、功耗等的影响，引入 PLL、COP、EMC 等概念，讲述单片机本身在降低功耗方面的考虑，以及应用系统设计中应如何从硬件、软件的结合上考虑功耗问题。同时说明从实验到实际应用还有多远。

附录 A 给出监控程序使用说明和源代码。

附录 B 是作者所在实验室制作的 S08BDM 开发工具使用说明。

附录 C 列出 MC9S12GT60 单片机的存储器分配表、内部寄存器地址分配表和中断向量表。在编写程序时常常需要查找这些信息。

附录 D 是 S08CPU 指令机器码表和汇编指令。

书中使用的 C 编译器是 Codewarrior for S08 V5.0 教学版 (Special Edition), 可以从 Freescale 公司的官方网站上下载, 也可以联系器件代理商获取免费光盘。

本书前 8 章是单片机的入门知识, 后 4 章讲述单片机应用系统设计深层次的问题。对于低年级学生, 后 4 章可以略去。教师也可根据学时数、学生知识水平等确定以前一部分或后一部分为重点。无论以哪一部分为重点, 这 12 章应视为一个整体。

本书对于单片机入门基本够用了, 而进一步开发单片机内的各种功能模块, 则需要查询相关原文资料, 这些资料不难从网上下载。我们希望, 学生能逐步做到 3 个看懂: 看懂硬件原理图, 看懂别人编写的程序 (包括汇编和 C 程序), 看懂原文资料, 从而达到基本入门的目的, 以后的路, 他们会自己走。

特别要指出的是, 单片机的开发方法仅靠读书是不可能掌握的, 实践环节最重要, 一定要动手做。从大学二年级开始学习单片机入门课, 到研究生毕业时至少有 5、6 年的学习成长时间, 而单片机课程的课内学时是非常有限的, 仅依靠课内的学习和实验不可能完全掌握单片机开发所需的全部知识, 需要在进一步学习的基础上进行实际项目的训练。清华大学有专门的 SRT 计划 (Student Research and Training Program) 对此进行支持, 学生可以设想一个单片机应用的原理性演示题目, 花半年到一年的课余时间, 设计一块印刷线路板, 焊上元器件, 再实现预想的功能。在 SRT 计划中出现的优秀作品可拿到每年校庆的学生科技展上评奖, 甚至参加全国大学生“挑战杯”的角逐。这样的学生在进入研究生阶段可以很快进入科研课题, 甚至在本科毕业设计阶段就能进入角色, 并最终成为嵌入式系统设计的高手。

陈萌萌、张首钊、张乐平等研究生做过本课程助教, 参与了部分章节的编写工作; 何峰设计了相关硬件系统; 卓开阔编写了本书提供的监控程序的源代码和详细的注释, 并且测试了所有的实验示例代码; 龚光华老师协助我完成了对全书的校订。参加本书编写工作的还有曾鸣、宫辉、薛涛、安鹏、金鑫等。本教材的编写工作还得到了 Freescale 公司金功九博士和 Roger Fan 先生的大力支持, 在此表示感谢。

作者殷切希望, 这本教材能引导读者入门, 能使他们从较高的起点出发, 沿着正确的成才之路顺利前行, 去实现他们的理想。

邵贝贝

2006 年 6 月

目 录

第 1 章 单片机的广泛选择

1.1 单片机世界	1
1.2 开发单片机应用不再需要仿真器	4
1.3 Freescale 单片机	5
1.4 Freescale 公司的 32 位单片机	6
1.4.1 以 68K、CPU32 为 CPU 的 32 位单片机	6
1.4.2 以 ColdFire 为 CPU 的 32 位单片机	7
1.4.3 用于控制的以 PowerPC 为 CPU 的 32 位单片机	8
1.4.4 用于通信的以 PowerPC 为 CPU 的 32 位单片机	8
1.4.5 ARM 系列 32 位单片机	9
1.5 Freescale 公司的 DSP 型单片机	10
1.5.1 普通 16 位 DSP 型单片机	10
1.5.2 使用增强型内核的 16 位 DSP 型单片机	14
1.5.3 StarCore 系列 16 位 DSP 型单片机	15
1.5.4 24 位 DSP 型单片机	15
1.6 Freescale 公司的 16 位单片机	16
1.6.1 MC68HC12 系列单片机	16
1.6.2 MC9S12 系列单片机	17
1.6.3 MC9S12X 系列单片机	20
1.7 Freescale 公司的 8 位单片机	20
1.7.1 MC68HC08 系列单片机	20
1.7.2 MC9S08 系列单片机	24
1.7.3 MC9RS08 系列单片机	28

目 录

1.8	相关软件开发工具	30
第2章 数字电路基础及 S08 单片机简介		
2.1	数字电路基础	31
2.1.1	高速 CMOS 电路与逻辑电平	31
2.1.2	“与非”门和“或非”门	32
2.1.3	三态门	33
2.1.4	组合逻辑电路	34
2.1.5	D 触发器与时序电路	34
2.1.6	数的表示法与运算	35
2.1.7	中央处理器示意	36
2.1.8	存储器	38
2.1.9	中断与复位	40
2.2	真实单片机——MC9S08GT60	40
2.2.1	S08 CPU 结构	40
2.2.2	S08 CPU 寄存器	41
2.2.3	MC9S08GT60 功能模块结构	43
2.2.4	MC9S08GT60 单片机存储器组织	45
2.2.5	MC9S08GT60 的复位与中断	46
2.2.6	MC9S08GT60 的引脚与封装	47
2.2.7	MC9S08GT60 复位及系统寄存器设置	50
第3章 S08 单片机的指令系统及汇编语言程序设计		
3.1	指令与寻址方式	51
3.1.1	指令助记符、操作码和操作数	51
3.1.2	汇编指令集	52
3.1.3	指令按功能分类	52
3.1.4	寻址方式	58
3.2	S08 汇编语言程序设计	63
3.2.1	机器语言和汇编语言	63
3.2.2	汇编语言源程序的格式	64
3.2.3	汇编管理指令	67
3.2.5	S 记录	72
3.2.6	汇编语言程序设计和举例	74

第4章 单片机基本系统设计

4.1 芯片选型——初识 MC9S08GT60	81
4.2 设计规划	84
4.3 基本系统的电路设计	85
4.3.1 电源电路	85
4.3.2 时钟电路	86
4.3.3 复位电路	86
4.3.4 BDM 调试接口	88
4.3.5 RS-232 驱动电路	89
4.3.6 外部接口定义	89
4.4 系统综合	90

第5章 异步串行通信

5.1 串行通信协议 RS-232 标准	93
5.1.1 串行数据格式	95
5.1.2 RS-232 电缆的连接方法	95
5.1.3 通信速率	96
5.1.4 ASCII 码	96
5.2 SCI 功能简介	98
5.2.1 SCI 发送器	99
5.2.2 SCI 接收器	99
5.3 SCI 寄存器	100
5.3.1 波特率寄存器	100
5.3.2 串行通信控制寄存器 1(SCI1C1)	100
5.3.3 串行通信控制寄存器 2(SCI1C2)	101
5.3.4 串行通信控制寄存器 3(SCI1C3)	102
5.3.5 串行通信状态寄存器 1(SCI1S1)	102
5.3.6 串行通信状态寄存器 2(SCS2)	103
5.3.7 串行通信数据寄存器(SCIxD)	103
5.4 SCI 的软件开发方法	104
5.4.1 SCI1 的初始化	104
5.4.2 SCI 的查询方式	105
5.4.3 SCI 的中断方式	106

目 录

5.5	SCI 应用	108
第 6 章 单片机片内 I/O 模块		
6.1	并行 I/O 端口	111
6.1.1	并行口简述	111
6.1.2	并行口应用举例	112
6.2	定时器	114
6.2.1	定时器的结构及基本功能	114
6.2.2	输入捕捉	117
6.2.3	输出比较	118
6.2.4	脉宽调制输出	118
6.3	同步串行外设接口 SPI	119
6.3.1	SPI 的数据流动	121
6.3.2	SPI 功能	121
6.3.3	SPI 引脚	122
6.3.4	SPI 寄存器	124
6.4	A/D 转换模块	127
6.4.1	A/D 控制寄存器	128
6.4.2	A/D 状态和控制寄存器	130
6.4.3	A/D 数据结果寄存器	131
6.4.4	A/D 引脚允许寄存器	131
第 7 章 用 C 语言开发应用程序		
7.1	C 语言是开发单片机应用软件的有力工具	132
7.2	开发嵌入式应用的 C 编译器的特点	133
7.2.1	关于初始化变量	134
7.2.2	函数的可重入性问题	135
7.3	交叉编译和 C 语言程序运行环境的建立	135
7.3.1	应用程序的构成与模块化程序结构	139
7.3.2	全局变量与局部变量	140
7.3.3	函数的结构与函数间参数的传递	141
7.3.4	C 语言中的 I/O 语句	143
7.3.5	程序模块的框架与组织	144
7.3.6	程序的链接与定位	145

7.4	嵌入式应用中的 I/O 函数	146
7.4.1	关于“Hello, World”	146
7.4.2	自己编写 printf()函数	147
第 8 章 HCS08 实验开发系统及实验指导		
8.1	HCS08 实验开发系统简介	158
8.1.1	HCS08 实验开发系统特点	158
8.1.2	HCS08 实验开发系统套件	159
8.1.3	HCS08 实验开发系统对 PC 的要求	159
8.1.4	HCS08 实验开发系统与 PC 的连接	159
8.2	HCS08 实验系统设计框架	159
8.2.1	目标评估母板与单片机的接口定义	160
8.2.2	8 位数字量输入电路	160
8.2.3	8 位数字量显示电路	161
8.2.4	A/D 模拟电平产生电路	161
8.2.5	数码管显示电路	161
8.2.6	键盘中断信号产生电路	161
8.2.7	IRQ 中断信号产生电路	163
8.2.8	SPI 输入电路	163
8.2.9	SPI 输出电路	164
8.2.10	MC68HC908QY4 编程/调试电路	164
8.2.11	面包板	165
8.3	基于 HCS08 实验开发系统的实验	165
8.3.1	实验 1——熟悉监控程序与实验环境	165
8.3.2	实验 2——并行口实验	167
8.3.3	实验 3——机器语言和汇编程序	168
8.3.4	实验 4——使用 CodeWarrior for S08 汇编编译器	169
8.3.5	实验 5——用汇编语言显示“Hello World”	173
8.3.6	实验 6——用 SCI 的中断方式接收字符	175
8.3.7	实验 7——使用 CodeWarrior for S08 的 C 编译器编程	177
8.3.8	实验 8——用 C 语言实现 SCI 通信与 LED 数码管驱动	182
8.3.9	实验 9——定时器中断实验	184
8.3.10	实验 10——定时中断程序设计	186
8.3.11	实验 11——通用 I/O 及 IRQ 中断程序设计	189

目 录

8.3.12	实验 12——键盘中断程序设计	190
8.3.13	实验 13——SCI 中断与 A/D 转换程序设计	192
第 9 章 建立实用的在线调试环境		
9.1	建立单片机的运行环境——系统初始化	197
9.2	实现人机交互——串口通信	199
9.3	控制 CPU 的行为——利用 CPU 固定映像	200
9.4	查看和修改 RAM 区	203
9.5	下载目标程序	205
9.5.1	S 格式文件	205
9.5.2	向 RAM 下载程序	206
9.5.3	在 RAM 程序中设置断点	206
9.5.4	Flash 编程基本方法	208
9.5.5	将 S 格式文件下载到 Flash 区	213
9.6	Flash 的块保护和保密设置	215
9.6.1	块保护	215
9.6.2	加密设置	217
9.6.3	加密方法	218
9.6.4	解密方法	218
9.7	实现断点功能——在监控程序中使用 DBG 模块	219
9.8	中断向量重新定向	223
9.9	帮助信息	224
第 10 章 BDM 后台调试模式		
10.1	BDM08 概述	227
10.2	进入 BDM 模式	228
10.3	BDM 通信协议及底层软件	230
10.3.1	BDM 相关寄存器	230
10.3.2	BDM 时钟的同步	231
10.3.3	BDM 基本指令——读、写单字节	233
10.3.4	BDM 指令的组织	237
10.4	BDM 简单应用	240
10.4.1	用 BDM 对 Flash 编程	240
10.4.2	通过 BDM 显示存储器内容	243

10.5 使用 DBG 模块完成复杂调试	244
10.5.1 利用 DBG 来增加两个硬件断点	244
10.5.2 利用 DBG 捕获指定地址的信息	245
10.5.3 获取指定条件下的程序分支跟踪信息	246
10.5.4 实时获取运行记录	246
第 11 章 嵌入式实时操作系统 $\mu\text{C}/\text{OS} - \text{II}$ 在 HCS08 上的实现	
11.1 实时系统	249
11.2 前后台系统	250
11.3 嵌入式实时操作系统	250
11.3.1 不可剥夺型内核的多任务实时系统	251
11.3.2 可剥夺型内核的多任务实时系统	251
11.3.3 使用嵌入式实时操作系统的优缺点	252
11.4 嵌入式实时操作系统 $\mu\text{C}/\text{OS} - \text{II}$	253
11.5 在 HCS08 上移植 $\mu\text{C}/\text{OS} - \text{II}$	254
11.6 为内核编写与硬件相关的代码	257
11.6.1 OS_CPU.H	257
11.6.2 OS_CPU_C.C 和 OS_CPU_A.ASM	259
11.6.3 产生时钟节拍中断	265
11.6.4 编写自己的任务代码	266
11.6.5 启动 $\mu\text{C}/\text{OS} - \text{II}$	268
第 12 章 时钟和低功耗模式	
12.1 HCS08 的内部时钟发生模块	269
12.1.1 时钟发生模块的 4 种工作模式及特点	271
12.1.2 时钟发生模块的初始化	272
12.2 看门狗	278
12.3 HCS08 单片机的低功耗设计	279
12.3.1 单片机中功耗的来源	279
12.3.2 HCS08 的低功耗模式	279
12.3.3 降低单片机的供电电压	282
12.3.4 嵌入式应用的低功耗设计	284

目 录

附录 A S08Monitor 用户手册和源代码

- A.1 S08Monitor 命令行方式指令列表 291
- A.2 S08Monitor 命令行方式指令详解 291
- A.3 利用监控代码编写自己的程序 295
- A.4 S08 监控程序(C语言版) 296

附录 B THUBDM08 系统使用说明书

- B.1 系统硬件设置说明 317
- B.2 Serial Monitor 方式说明 319
- B.3 命令行方式说明 323

附录 C HCS08GT60 资源速查

- C.1 S08GT60 存储空间分配图 330
- C.2 寄存器汇总 331
- C.3 中断向量表 335

附录 D S08CPU 指令集

- D.1 助记符指令表 337
- D.2 机器码指令表 346

参考文献 349