

从51到ARM

——32位嵌入式系统入门

● 赵星寒 刘 涛 编著

51—ARM



北京航空航天大学出版社

内 容 简 介

本书从 51 系列单片机的一般知识出发,将 ARM 处理器和 51 系列单片机进行对比,引导读者去理解和学习 ARM 处理器的知识。内容分为 3 部分:第 1 部分是前 4 章,从大家所熟悉的 51 系列单片机的基础知识开始,介绍 ARM 处理器的基本知识,包括 ARM 和 51 系列的对比、中断处理系统、寄存器和存储器结构等;第 2 部分是 5~9 章,详细比较 ARM 指令和 51 系列指令之间的差异,进一步阐述 ARM 指令的含义和使用方法,从简单的 51 系列编程经验出发,介绍 ARM 处理器软件编程方法;第 3 部分是后 3 章,介绍 ARM 处理器开发工具的使用。

任何一个学习过 51 系列单片机的技术人员,借助本书都会很容易地学会 ARM 处理器的一般知识,了解软件设计的基本方法,并且能够使用开发工具进行程序编辑、编译、连接和调试,成为一个初步懂得 ARM 并进行软件设计的工程师。

本书可供电子信息类大学生、研究生或电子设计工程师阅读参考。

图书在版编目(CIP)数据

从 51 到 ARM——32 位嵌入式系统入门/赵星寒等编著.

北京:北京航空航天大学出版社,2005.10

ISBN 7-81077-719-X

I. 从… II. 赵… III. 微处理器, RAM—系统设计

IV. TP332

中国版本图书馆 CIP 数据核字(2005)第 102390 号

从 51 到 ARM

——32 位嵌入式系统入门

赵星寒 刘 涛 编著

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:787 mm×960 mm 1/16 印张:25.5 字数:571 千字

2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-719-X 定价:38.00 元

前言

第 1：为什么非学 ARM 不可？

(1) 单片机技术的应用在更广泛的范围内把电子产品带到我们身边，把我们每一个人都和“电脑”紧紧相连。仔细观察我们周围的生活，无处不包含单片机的踪影。以 8051 为代表的单片机渗透到到我们生活的方方面面。但从 20 世纪 80 年代到现在，单片机始终在 8 位机的档次上徘徊，8 位的单片机始终主导着应用的潮流。16 位单片机虽然也曾经掀起过波浪，但很快就销声匿迹了。可时至今日，我们还能使用 8 位的单片机做些什么呢？在 20 世纪 80 年代，凡是使用 8051 系列单片机的产品都是先进技术的体现，现在人们当然不会这样去审视一个产品。我们再也无法把 51 系列和一个技术先进的产品相提并论。32 位单片机应用的高潮正悄悄到来。只有使用功能强大的 32 位单片机才能设计出轻巧多变的手机，才能设计出映像逼真的视频产品，才能设计出非同凡响的工业产品。

(2) 每个工科院校电子信息类的毕业生都异口同声地说：我会 51 系列单片机。当他说这句话的时候，他和别人是站在同一起跑线上的。可我们完全有可能和别人不站在同一起跑线上。我们有很多人，毕业近十年了，一直在搞产品研发，但也始终工作在 51 系列单片机这一档次上。我们再也不要以熟知 51 系列而引以为荣，不要以精通 51 系列而作为一门工作技能。

(3) 诚然，51 系列及同档次的单片机并没有退出应用市场，在今后很

前言

长的时间内都会继续主导潮流,但在高端产品的应用领域不会再发挥作用。应用 51 系列的产品将是大众化的产品,进行 51 系列设计的工程师也只是大众化的工程师。

你有什么理由不学 ARM 呢?

第 2: 学 ARM 为什么要从 51 系列学起?

(1) 在单片机没有出现的 20 世纪 80 年代以前,学习计算机是从单板机开始的。在单片机出现以后,学习计算机一般都以 51 系列为基础。从 51 系列的基本概念出发,去理解和学习其他更复杂的计算机就要容易得多,因为很多基本概念是相同的。

(2) 本书的目的在于帮助那些经验并不丰富的初学者,这些人一般只有 51 系列的经验而没有其他单片机经验,因此,只能以 51 系列为基础,去学习 ARM 处理器的知识。

第 3: 学 ARM 究竟有多难?

ARM 处理器的确是一种很复杂的微处理器结构,但从应用的角度来说,只要有 51 系列的经验,学习 ARM 并不是很困难的事。

(1) 学习 ARM 处理器只学习与应用有关的部分。关于 ARM 处理器的很多资料,都是直接对 ARM 说明书的译文。这些说明书中很大一部分都是关于 ARM 结构的论述,这些内容是写给芯片设计工程师看的,学习这些内容对应用人员毫无用处。

(2) 应用 ARM 应该分为 3 个阶段:第 1 阶段是学习 ARM 的基本内容;第 2 阶段是学习以 ARM 为核的单片机;第 3 阶段才是应用。这与学习 51 系列单片机不同。学习 51 系列单片机时,学会了基本内容也就学会了单片机。本书所讲述的是 ARM 的基本内容。

(3) 与 51 系列的基本内容相近,ARM 的基本内容主要包括中断响应过程、ARM 指令集、存储器结构和汇编语言的程序结构,并没有比 51 系列更多的内容。从这几方面出发去学习 ARM 处理器,就会发现学习 ARM 并不是很困难的事。

第 4: 本书的内容安排

本书的前 4 章是对 ARM 处理器的一般介绍,内容并不多,把这些内容与 51 系列对比学习,会很快了解 ARM 处理器的基本原理。对于这些内容以外的关于 ARM 的知识,建议初学者暂时不要去研究。

第5章介绍 ARM 指令集,这一章的内容是汇编语言基础。第6章介绍 Thumb 指令集,它是 ARM 指令集的一个子集。在大多数情况下,因为所有的程序都可以使用 ARM 指令实现,初学者可以暂不学习 Thumb 指令集。但学过 ARM 指令集以后,学习 Thumb 指令集是一件很容易的事情。第7章讲述伪指令,使用 ARM 汇编语言必须熟悉伪指令。

第8章讲述软件的设计方法和汇编语言应用规则。第9章讲述 C 和 C++ 语言的应用。

在 ARM 应用中,学会使用开发工具是十分重要的。后面3章详细讲述开发工具的应用。第10章介绍 ARM 开发中一种常用的工具 ADS,介绍怎样使用 ADS 编辑和编译程序。第11章介绍怎样使用 ADS 连接程序。第12章介绍怎样使用 AXD 调试程序。

第5: 本书的目标

通过学习本书前4章的内容,可以了解 ARM 处理器的基本概念。通过学习5~9章的内容,可以学会为 ARM 系统编程的一般方法。通过后3章的学习,可以学会使用 ARM 开发工具。

学习本书以后,应该了解软件设计的基本方法,并且能够使用开发工具进行程序编辑、编译、连接和调试,成为一个初步懂得 ARM 并可以进行软件设计的工程师。

建议在学习本书时,应该重点学习后3章,并实际使用开发工具进行操作。只有这样,才可以使自己提高一大步。

本书的第9章由我的同事刘涛编写,其余均由我编写。在编写过程中,得到张君、邓耀武、郑玉峰三同仁及清华大学蒋东翔博士的大力帮助,也得到北京航空航天大学出版社的大力支持,在此一并感谢。由于接触 ARM 处理器时间不长,本书不可避免存在一些错误,欢迎广大读者指正。交流请使用 zxhjeket@yahoo.com.cn。

赵星寒

2005年9月于北京

目 录

第 1 章 几个基本概念

1.1 从 51 系列单片机到 ARM 处理器	1
1.2 ARM 不是单片机	2
1.3 51 系列单片机和 ARM 处理器的比较	3
1.4 以 ARM 为核的单片机	4
1.4.1 处理器内核只和外部地址相关	4
1.4.2 ARM 怎样组成一个单片机	6
1.4.3 一点启示	7

第 2 章 ARM 处理器基础

2.1 ARM 处理器的数据格式	8
2.1.1 基础知识	8
2.1.2 ARM 处理器的存储器结构	9
2.1.3 ARM 处理器的指令存储或数据存储	13
2.1.4 地址对准	15
2.2 处理器模式	15
2.3 ARM 内部寄存器	17
2.3.1 ARM 状态时的寄存器	17
2.3.2 Thumb 状态下的寄存器	25
2.4 异 常	27

第 3 章 ARM7TDMI 简介

3.1 概 述	29
3.2 结构和框图	30
3.3 3 级流水线结构	33
3.4 ARM 处理器的寻址方式	35
3.4.1 51 系列与 ARM 系列寻址方法的比较	35
3.4.2 寄存器寻址	36

目 录

3.4.3	立即寻址.....	36
3.4.4	寄存器移位寻址.....	37
3.4.5	寄存器间接寻址.....	39
3.4.6	变址寻址.....	40
3.4.7	多寄存器寻址.....	42
3.4.8	堆栈寻址.....	42
3.4.9	寻址方式的应用.....	43
3.4.10	ARM 指令寻址一览表	44

第 4 章 异常和中断

4.1	概 述.....	45
4.2	复位异常.....	46
4.3	未定义指令异常.....	47
4.4	软件中断异常.....	49
4.5	预取指中止异常.....	49
4.6	数据中止异常.....	50
4.7	中断请求(IRQ)异常	51
4.8	快速中断(FIQ)请求异常	52
4.9	中断源和中断优先级.....	53
4.10	异常中断的进入和退出	54

第 5 章 ARM 指令集

5.1	概 述.....	56
5.2	指令可选后缀.....	59
5.2.1	S 后缀	60
5.2.2	! 后缀.....	60
5.2.3	B 后缀	61
5.2.4	T 后缀.....	61
5.2.5	条件后缀.....	62
5.3	ARM 数据传送指令(存储器访问指令)	64
5.3.1	关于数据传送.....	64
5.3.2	数据传送中的几个问题.....	66
5.3.3	ARM 数据传送指令代码表	70
5.3.4	数据传送指令代码详述.....	71

5.4	ARM 分支指令	93
5.4.1	关于程序分支	93
5.4.2	ARM 分支指令详解	94
5.5	ARM 数据处理指令	98
5.5.1	ARM 数据处理指令的特点	98
5.5.2	第2操作数	99
5.5.3	ARM 数据处理指令代码表	101
5.5.4	ARM 数据处理指令详述	103
5.6	ARM 指令集中和协处理器有关的指令	120
5.7	ARM 指令集中的软件中断指令	125

第6章 Thumb 指令集

6.1	Thumb 指令的特点	129
6.2	Thumb 指令和 ARM 指令的比较	132
6.3	Thumb 指令集中的数据传送类指令	134
6.4	Thumb 指令集中的分支指令	146
6.5	Thumb 指令集中的数据处理指令	152
6.6	Thumb 指令集中的软件中断指令	164

第7章 伪指令

7.1	伪指令概述	166
7.2	与 ARM 指令相关的伪指令	167
7.3	与 Thumb 指令相关的伪指令	177
7.4	通用伪指令	180
7.4.1	为变量定义和赋值的伪指令	180
7.4.2	定义各类寄存器名称的伪指令	184
7.4.3	数据定义伪指令	187
7.4.4	控制程序流向的伪指令	196
7.4.5	其他伪指令	201

第8章 ARM 系统设计和程序设计

8.1	关于 ARM 处理器,我们学到了什么?	207
8.2	程序设计的几个问题	209
8.3	ARM 指令和 Thumb 指令的兼容性	214

目 录

8.4 汇编程序规则	221
8.4.1 组 成	222
8.4.2 书写格式	223
8.4.3 汇编语言中的表达式	224
8.5 汇编程序举例	231
第 9 章 C 和 C++ 语言的应用	
9.1 汇编语言与 C/C++ 语言的比较	237
9.2 C/C++ 语言的编程规范	238
9.2.1 注 释	238
9.2.2 命 名	239
9.2.3 编辑风格	239
9.3 如何编写针对 ARM 的高效 C 语言程序	240
9.3.1 以空间换时间	240
9.3.2 使用数学方法解决问题	242
9.3.3 使用位操作	242
9.3.4 嵌入汇编	243
9.4 C、C++ 语言与汇编语言的混合编程	244
9.4.1 在 C、C++ 程序中使用内嵌的汇编指令的语法格式	244
9.4.2 在 C、C++ 程序中使用内嵌的汇编指令的注意事项	245
9.4.3 内嵌汇编指令的应用举例	247
9.4.4 从汇编程序中访问 C 程序变量	248
9.4.5 在 C++ 程序中使用 C 程序头文件	249
9.4.6 汇编程序、C 程序以及 C++ 程序的相互调用举例	250
第 10 章 程序的编辑和编译	
10.1 程序设计过程	254
10.1.1 编辑程序	254
10.1.2 编译程序	255
10.1.3 连接和调试程序	257
10.2 ARM 处理器开发工具介绍	258
10.2.1 ARM 处理器仿真系统特点	258
10.2.2 ADS 集成开发环境	259
10.2.3 ADS 开发系统的系统资源	261

10.2.4	系统软件开发的步骤	263
10.3	程序的编辑	263
10.3.1	源程序是可移植的	263
10.3.2	在 CodeWarrior for ADS 环境下编辑源文件	266
10.3.3	建立工程项目总结	271
10.3.4	建立 C 语言和 C++ 语言源文件	271
10.3.5	文件的管理	272
10.3.6	编辑画面的设置	273
10.3.7	ADS 编辑功能介绍	279
10.3.8	屏幕分割设置	282
10.4	程序的编译	283
10.4.1	有两种方式可以实现程序的编译	283
10.4.2	命令行中的可选项和图形方式中的可选项	286
10.4.3	ADS 中的可选项	290
10.5	汇编器的应用	291
10.5.1	ARM 汇编器 armasm 中的可选项	291
10.5.2	ARM 汇编器应用	298
10.5.3	应用举例	300
10.6	编译器的应用	303
10.6.1	ARM 编译器中的可选项	303
10.6.2	ARM 编译器的应用	317

第 11 章 程序的连接

11.1	几个与连接有关的概念	320
11.2	连接器的应用	326
11.3	连接器选项设置	334
11.4	连接器应用举例	347

第 12 章 ARM 调试工具 AXD

12.1	调试工具 AXD	361
12.1.1	ARM 调试工具介绍	361
12.1.2	调试工具 AXD 的使用方法	363
12.2	在调试过程中使用断点	371
12.2.1	设置断点	371

目 录

12.2.2	执行到断点处.....	374
12.2.3	查找和修改断点.....	374
12.2.4	断点管理下拉菜单.....	374
12.2.5	观察断点.....	375
12.2.6	删除断点.....	376
12.3	AXD 中其他工具介绍	376
12.3.1	使用观测项.....	376
12.3.2	在调试程序时观察变量.....	381
12.3.3	在调试程序时使用符号表.....	383
12.3.4	使用观测点.....	384
12.3.5	工具栏中的工具.....	387
12.4	数据格式.....	389
12.4.1	设置数据格式.....	389
12.4.2	设置默认的数据显示格式.....	393

参考文献

第 1 章

几个基本概念

1.1 从 51 系列单片机到 ARM 处理器

Intel 公司从 1980 年推出 8 位 51 系列单片机,发展到今天已经形成了一个应用广泛、门类齐全的庞大家族。中国计算机基础教育一般以 51 系列单片机为基础。因此,工科院校电子信息类的毕业生都声称会 51 系列单片机。

51 系列单片机经过多年的开发和应用,不仅在教育体系上比较完整,而且在应用上也有众多的、功能强大的仿真工具支持,再加多年的技术积累,富有经验的工程师比比皆是。因此 51 系列单片机在工业、民用、医疗器械和家用电器等各个领域都得到了广泛的应用。

随着科技的发展,人们对数字化的要求越来越高,同时,先进的芯片的设计日新月异。从这两个方面出发,人们对单片机的性能、速度、存储量、通信能力、功能的多样性、开发的方便程度及耗电的多少等不断提出更高的要求。ARM 处理器就是近年来发展十分迅猛的一种体系结构。

从 51 系列单片机到 ARM 处理器是一个很大的跨度。原因是 ARM 处理器完全不同于 51 系列单片机,其结构十分复杂。一个设计经验不多的工程师从 51 系列的经验出发去学习 ARM 很困难。而对于那些只有实验室经验的工科院校的学生来说,理解 ARM 的体系结构真正是纸上谈兵。但总不能等到有了经验再去学习。怎样从已有的 51 系列单片机的经验出发去理解和学习 ARM,正是本书所要完成的任务。之所以这样提出问题,主要是因为 51 系列普及的程度较大,如果能充分利用已有的知识,由此及彼地去学习 ARM,理解 51 系列和 ARM 中相近的部分,将会收到事半功倍的效果。但愿本书能成为学习 ARM 的一个台阶和入门。

本书所提及的 51 系列单片机,不是指 51 系列的那一个型号,而是指 51 系列的基本

特征。

一般称 51 系列为单片机,单片机就是把中央处理器(Center Processing Unite)、存储器(RAM/ROM)和输入/输出设备(Input/Output)集成在一个芯片内的芯片。它具有微型计算机的最基本的功能,是一个可以独立运行的最小的智能系统。一般称其为 Single Micro-computer,也可以称为微型控制器(Microcontroller)。把单片机作为一个控制系统的核心进行产品设计,使产品的性能得到提高,这在 20 世纪 80 年代的确是一件很大的成就。它的出现极大地促进了计算机在各个领域的应用。但随着科技的发展,更大规模的集成电路在芯片设计方面得到了进一步的推广和应用,单片机的功能变得更加强大,这时人们把这些功能更加强大的单片机称为微处理器(Microprocessor)。因此微处理器与单片机一般并没有多大区别,只不过微处理器功能更强大,特别是数据处理功能更加强大。

从 51 系列到 ARM 处理器虽然跨度很大,但这之间并没有一个台阶,比如学习 16 位的单片机是不是更利于学习 32 位的 ARM? 其实这是没有多大帮助的,因此,学习 ARM 必须从头学起,从自己的经验出发学起。

1.2 ARM 不是单片机

ARM 不是单片机,而只是一个单片机的内核。ARM 是 Advanced RISC Machines 的缩写。其中 RISC 原文是 Reduced Instruction Set Computer,意为精简指令集计算机。同时,ARM 也是设计 ARM 处理器的公司的简称。

单片机最主要的特征就是本身能组成最小系统,可独立运行,并具有完整的功能;而 ARM 则不能。严格地说,ARM 本身不是一个产品,不是一个芯片。

因此,ARM 和单片机不同,它仅仅是单片机中的中央处理器。一般称其为 ARM 处理器结构。这个微处理器结构拥有自己的指令代码集,可以通过执行代码控制外部设备。给这个微处理器设计出各种外部设备,并把它们用总线连接到一起,就组成了不同结构和性能的单片机。目前,世界上很多芯片生产商都利用 ARM 处理器设计自己的单片机。之所以这样做,主要有以下几个原因:

- ◆ 利用成功的内核开发自己的单片机,可以极大地缩短产品的开发周期(因为内核的设计比外围设备复杂得多)并降低开发风险。
- ◆ ARM 公司没有自己的产品,只是转让技术。正是因为它的任务的单一性,才使 ARM 内核技术更趋于完善,成为高性能的微处理器结构。
- ◆ 不同厂家的微处理器采用同一种内核,这对于应用工程师是十分有利的。因为不同的单片机具有相近的结构和相同的指令代码。

ARM 是一个不断发展的微处理器家族。目前,在这个家族中,主要有 5 个产品系列:

1 几个基本概念

ARM7、ARM9、ARM9E、ARM10 和 SecurCore。

而以 ARM 为核的单片机也在不断发展。目前,全世界较大的芯片生产商大都在应用 ARM 技术。以 ARM 为核的单片机在国内也得到越来越多的应用。

所谓以 ARM 为核,就是把 ARM 作为中央处理器,根据需要设计出外围功能模块,用总线把这些功能模块和 ARM 核连接在一起,组成一个单片机。这个单片机由 ARM 核控制,ARM 核执行指令,并根据指令对外围设备发出各种控制命令。

现在,也有以 51 系列为核的单片机,其设计主要基于 51 系列具有广泛应用这样一个背景。在这样的设计中,把使用 51 系列指令、具备 51 系列结构的单片机设计得功能更加强大,内部存储区更多,输入/输出线增加几倍,速度可达 40 Mb/s。但这种单片机仍然是一个 8 位机,功能仍然受到 51 系列的局限。

1.3 51 系列单片机和 ARM 处理器的比较

ARM 处理器作为单片机的内核和 51 系列单片机内核有哪些相近和差异呢?下面以 89C52 和 ARM7 为例进行比较,如表 1.1 所列。

表 1.1 51 系列和 ARM 处理器的比较

51 系列单片机内核	ARM7 系列处理器
8 位代码指令	32 位代码指令(兼容 16 位代码)
8 位数据总线	32 位数据总线
16 位地址总线	32 位地址总线
6 个中断源	7 个中断源(含复位)
工作寄存器(R0~R7)4 组	共 37 个寄存器
程序计数器	程序计数器
状态寄存器	状态寄存器
累加器 A 和 B	37 个都可以做累加器
寻址范围 16 位地址宽度	寻址范围 32 位地址宽度
不能预取指	三级流水线预取指
1 种工作模式	7 种工作模式
不支持协处理器	支持协处理器
不支持 JTAG 调试	支持 JTAG 调试

其实这种比较没有太多意义,我们只是想把它们联系起来。在后面的学习中,将会发现它们之间的差异。

1 几个基本概念

举例来说,51 系列单片机只有 6 个中断源,而 ARM 系列单片机可多达上百个中断源,虽然 ARM 处理器只有 7 个中断源。再举例说,有的以 ARM 为核的单片机复位中断也是可以屏蔽的,这可能是在 ARM 中也把复位作为中断处理的原因,但从 51 系列的结构出发,这是一个很难理解的事情。

51 系列单片机是一个 8 位的单片机,因而只能执行 8 位宽的指令。ARM 是一个 32 位的结构,不但可以执行 32 位的指令(称为 ARM 指令集),也可以执行 16 位的指令(称为 Thumb 指令集)。

虽然 ARM 处理器要比 51 系列强大得多,但它们仍有很多相似之处,在学习中可以借鉴的有:

- ◆ 指令的处理——取指、译码和执行过程是一样的;
- ◆ 中断的处理过程——响应中断、分配向量、保护现场和退出中断是一样的;
- ◆ 指令的种类和指令的含义是相近的,比如数据传送、数据相加/减等;
- ◆ 程序的编程方法是相同的,比如程序分支、子程序调用等;
- ◆ 寄存器的使用是相近的,比如堆栈指针、程序计数器(PC)和程序状态寄存器等;
- ◆ 程序的处理是相近的,主要指程序的编辑、编译、连接,目标代码的生成和程序的调试等。
- ◆ 一些通用概念是相同的,比如复位、复位后的状态、中断、地址、指令代码和数据等。在后面的学习中,要注意它们之间最主要的区别在于:
- ◆ ARM 只是一个单片机内核;
- ◆ ARM 是一个 32 位处理器,而 51 系列只是 8 位,在数据访问的处理方式上相差很大;
- ◆ ARM 处理器有多种工作模式,而 51 系列只有一种;
- ◆ ARM 处理器内寄存器只有一种,而 51 系列有多种;
- ◆ ARM 不但有 32 位的 ARM 指令集,还包含 16 位的 Thumb 指令集。

1.4 以 ARM 为核的单片机

1.4.1 处理器内核只和外部地址相关

先从 51 系列单片机说起。

图 1.1 的虚框内为 51 系列单片机的简化框图,虚框外为连接在单片机总线上的外部设备 A。

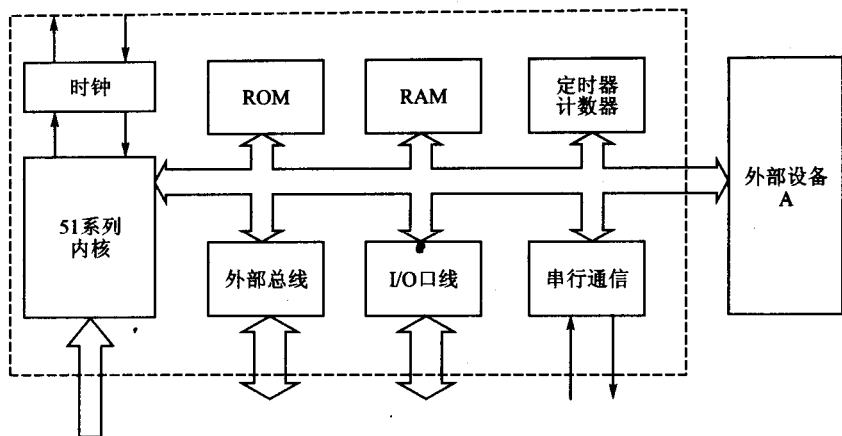


图 1.1 51 系列单片机框图

在单片机内,通过总线与内核(中央处理器)连接在一起的有各种外围(片内)设备,如定时器/计数器、数据存储器和串行通信模块等。对中央处理器而言,一般可以把这些统称为外围设备(片内)。对中央处理器来说,这些外围设备和外部设备并没有太大区别,中央处理器对片内外围设备的操作是通过对寄存器的读/写来完成的,而对片外的外部设备的操作是通过对存储器的读/写来完成的。无论片内还是片外,对中央处理器来说,其操作都是对某一地址的操作。例如:

```
STAR:  MOV    DPTR, #20H           ;把外部地址值赋给 DPTR
        MOVX   A, @DPTR           ;对外部设备读操作
        MOVC   A, @A+DPTR         ;对外部设备读操作
        MOVX   @DPTR, A           ;对外部设备写操作

        MOV    TMOD, #020H        ;对内部设备写操作
        MOV    SCON, #0E0H        ;对内部设备写操作
```

在上例中,对片外设备的操作是通过把地址赋给 DPTR 来实现的,而对片内设备的操作是直接完成的。在这个例子中, TMOD 的地址是 89H, 而 SCON 的地址是 98H。对内核来说,这都是一次对总线的操作。

对设计者来说,不同的地址意味着不同的设备。对中央处理器来讲,对所有外部设备(片内和片外)的操作都是对地址的操作。从这个结论去学习 ARM,将对以后的学习有很大的帮助。

1.4.2 ARM 怎样组成一个单片机

像 51 系列的中央处理器(内核)一样,ARM 不是单片机,是一个只能够做单片机内核的中央处理器。在本书中,我们把它称为 ARM 处理器。在以后的学习中,我们要记住:ARM 处理器仅仅是一个单片机内核,它的任何功能实现都离不开外围设备。由于它仅仅是一个单片机内核,因此,由它所构成的单片机因侧重不同而性能相差很大。ARM 组成的单片机结构如图 1.2 所示。

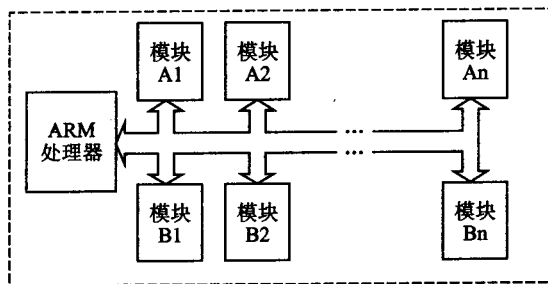


图 1.2 以 ARM 为核的单片机结构

在这种单片机中,可以设计出各种功能模块,比如 51 系列中所包含的串行通信模块、定时器/计数器模块等。由于这种模块设计相对简单,又很灵活,移植比较容易,因此,以 ARM 为核的单片机在同一家族中兼容性好。同时,由于模块是连接在总线上的,模块的多少对 ARM 本身没有影响,因此,在一个片内可以设计出很多不同功能的模块。正因为如此,以 ARM 为核才容易设计出功能强大的单片机。

在图 1.2 中,模块也可以是完全相同的或是有几个是相同的,这对系统并没影响。同 51 系列相似,ARM 处理器对外部模块的操作仅仅是对不同地址的操作。每个模块都有自己的功能寄存器,例如可能有命令寄存器、状态寄存器、中断寄存器和中断屏蔽寄存器等。每个模块的寄存器在总线上都有固定的地址,ARM 通过对这些地址的操作来实现对这些寄存器的控制和检测,也就是实现对这些模块控制和检测,从而实现各种功能。

在这种设计中,ARM 核只是把外部模块作为不同的地址对象,各个模块的不同对 ARM 处理器来说,仅仅是地址不同而已。

如图 1.3 所示是一个以 ARM 为核的单片机框图的局部。

从这个图可以看出,左上角的 ARM 处理器只是通过 32 位总线和外围设备连接在一起。