

高等院校通用教材

ARM[®] 体系结构 及其嵌入式处理器



任 哲 等编著



北京航空航天大学出版社

TP332/127

2008

高等院校通用教材

ARM[®] 体系结构 及其嵌入式处理器

任 哲 等编著

北京航空航天大学出版社

内 容 简 介

本书在介绍微型计算机一般原理的基础上,重点介绍目前在应用中流行的 ARM RISC 体系结构及其嵌入式处理器,同时也介绍其软件的一般设计方法。

本书共分 13 章。第 1 章和第 2 章重点介绍微型计算机系统的基本构成及基本工作原理;第 3~5 章介绍 ARM 体系结构、指令系统及汇编语言程序设计的基本知识;第 6 章介绍 ARM 的 C/C++ 和汇编语言的混合编程方法;第 7 章在介绍微型机中断技术的基本概念基础上介绍 ARM 的中断系统及其特点;第 8~11 章介绍基于 ARM 体系结构的嵌入式处理器 LPC2000 的构成、特点及程序设计方法;第 12 章简要地介绍另一个常用的基于 ARM 体系结构的嵌入式处理器 S3C44B0X 的构成及其主要接口,以使读者对 ARM 体系结构嵌入式处理器有一个更全面的认识;第 13 章介绍 ARM 作为一个完整的微型计算机体系结构所应有的存储管理部分,为读者学习高档 ARM 处理器核建立必要的基础。

本书适合电气自动化、仪器仪表、电子技术等专业微机原理课程使用,也可作为其他以计算机嵌入式应用作为教学目标的专业教学用书。另外,对研究 ARM 体系结构及其嵌入式处理器的专业人士也有一定参考价值。

图书在版编目(CIP)数据

ARM 体系结构及其嵌入式处理器/任哲等编著. —北

京:北京航空航天大学出版社,2008. 1

ISBN 978-7-81124-192-1

I. A… II. 任… III. 微处理器, ARM—系统设计

IV. TP332

中国版本图书馆 CIP 数据核字(2008)第 006048 号

© 2008, 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。

ARM 体系结构及其嵌入式处理器

任 哲 等 编 著

责任编辑 王慕冰 王平豪

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:787×960 1/16 印张:22 字数:493 千字

2008 年 1 月第 1 版 2008 年 1 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-192-1 定价:38.00 元

前言

近年来,微型计算机由于具有体积小,质量轻,耗电少,可靠性高,价格低廉,结构灵活等特点,使得它在国民经济及人们的生活中得到了广泛的应用。目前,微型计算机的应用已经深入到科学计算、信息处理、事务管理、过程控制、仪器仪表、家用电器等方面,已经是人们生产、生活中不可或缺的工具、装置和设备。

正是应上述不同应用领域的多样化需要,随着计算机技术的发展和半导体制造工艺的进步,微型计算机的体系结构及其外观和形态在长期的演变和发展中,也出现了多样化的发展态势,一种与人们常见的通用计算机在应用和外观上都有重大区别的计算机系统——嵌入式系统出现了。

嵌入式系统是指以应用为中心,以计算机技术为基础,软件、硬件可裁剪,植入应用系统内部,适应应用系统对功能、可靠性、成本、体积和功耗严格要求的专用计算机系统。

由于嵌入式系统应用的广泛性、强大的信息处理功能以及对其宿主对象在性能、功能等方面的极大提升能力,引起了汽车、航空、航天、造船、仪器仪表、家用电器等领域的兴趣和重视。于是,各行各业对嵌入式系统人才的需求也日益高涨。

勿庸质疑,在我国高等教育已进入大众化的今天,培养应用型人才已是高等教育的一个重要职能和责无旁贷的任务。因此,各高等院校必须将嵌入式系统的教学内容以某种方式引入相关专业的课程体系。鉴于嵌入式系统是一个多学科融合的技术,它所涉及的领域极其宽广,所以,如何在高等院校开设相关课程是对高等教育工作者的一个巨大挑战。

嵌入式系统作为一种计算机系统,也由硬件和软件两部分组成。硬件的发展大体上可以分为初期的 MCU(单片机)和现在的 SoC 两个阶段。虽然目前 MCU 仍然在嵌入式低端应用中被广泛使用,但随着嵌入式高端应用的迫切需求,价格的降低,基于 SoC 的嵌入式处理器已逐渐上升为嵌入式应用的主流。

鉴于上述形势,作为培养应用型人才的高等院校如何应对呢?这就不得不对这类院校计算机教育的历史做一分析。

笔者认为,从教学的角度来看,微型计算机原理课程承担着两项任务:一是使学生在学校学习期间以某种机型为背景了解计算机的体系结构;二是为应用计算机解决实际问题打下必要的基础。这两个任务的实现,在计算机专业和非计算机专业中还各有侧重,大体上是两种模式:

➤ 以了解计算机体系结构为主,为设计应用软件奠定基础的模式——微型计算机原理课



程的内容为 x86 或 x86 + 单片机。

► 以掌握计算机硬件结构为主,为计算机嵌入式应用奠定基础的模式——微型计算机原理课程的内容为 8 位单片机或 8 位单片机 + 16 位单片机。

前一种是大多数计算机专业所采用的模式,而后一种则是大多数非计算机专业所采用的模式。

必须肯定,过去上述两种模式都成功地为社会培养了大量的应用型人才,他们正在我国各领域为社会做着贡献。但也必须清楚,按前一种模式培养的人才,由于硬件方面的薄弱直接导致了与硬件紧密相关的系统软件(例如操作系统、编译软件等)掌握、了解得甚为浅薄,从而最终影响了高质量应用软件的设计和开发能力。而按后一种模式培养的人才,由于不能或不适合使用操作系统,不了解或不掌握具有现代计算机特征且极为简单的 MCU 的原理和硬件结构,所以使得他们只能进行一些低端应用的开发,在高端应用上则力不从心。

目前,基于 SoC 的处理器出现,给我国高等院校培养应用型人才的计算机教育提供了对课程体系及内容进行改革的良好机遇。理由如下:

- 由于嵌入式应用的特点,使得计算机应用进入了极为个性化的时代,不会也不可能由某个公司或国家对该项技术进行垄断,ARM 公司与芯片厂商之间的合作关系就是一个例子。因此,就给计算机专业的学生进行计算机底层技术的教育及学生毕业后在底层发展提供了机遇,从而也为他们在顶层的发展奠定了良好的基础。
- 由于基于 SoC 的处理器从体系结构上更接近于现代计算机体系结构的发展方向,把它作为非计算机专业微型计算机原理的主要内容,更有利于学生对计算机体系结构进行全面的了解和掌握,并为他们今后进行高端嵌入式应用打好基础。

基于上述理由,笔者认为,计算机应用专业,尤其是以嵌入式应用为目标的计算机应用专业的微型计算机原理课程的内容应该是:基于 SoC 的嵌入式处理器 + x86 或者是 x86 + 基于 SoC 的嵌入式处理器的模式。而非计算机专业的微型计算机原理课程的内容应该是:基于 SoC 的嵌入式处理器 + 单片机。这里“+”后面的内容最好由教师提供参考书让学生自学。

那么这样的微机原理课程的内容应该如何来构建呢?当然,这门课程还应该以讲授硬件及与硬件密切相关的汇编语言程序设计为主,但将来学生实际开发工作是以高级语言来进行的,所以对于汇编语言的程序设计内容要适当减少,以腾出课时讲解高级语言与汇编语言的混合编程。又由于现代嵌入式系统都使用了或大或小的操作系统,而处理器的某些部件(例如 Cache、MPU、MMU 等)必须在操作系统的配合下才能发挥其作用,因此,操作系统的基本概念及其引导也应该是这门课程的重要内容。由于这部分内容的处理是课程内容改革的一个难点,所以笔者曾经为此征询了一些专家的意见。一个比较统一的认识是,对于操作系统来说,本课程应该把操作系统的引导与装载作为重要内容,而操作系统的其他内容还应该放在操作系统课程中去解决;对于 Cache、MPU、MMU 则只介绍其硬件功能,而软件部分只给出必要的

流程图,具体的编程则应让学生通过阅读实际操作系统(例如 Linux)的相关代码去了解;至于进程及进程切换则通过学习操作系统 $\mu\text{C}/\text{OS}$ 等小型操作系统去了解;至于以哪种体系结构作为课程内容的背景,则应该以目前我国应用较普遍的 ARM 体系为主较为合适。

鉴于国内大多数介绍 ARM 体系结构的书籍是以工程技术人员为读者群的,起点相对较高,不适合教学使用。虽然有些书籍是为培训而编写的教材,但缺乏教学所需要的系统性。因此,笔者根据自己的教学体验试图编写一本教学用书,但由于可借鉴的资料较少,所以也就是一个尝试而已。

参加本书编写的作者有任哲、范忠诚、陈微和房红征,由任哲负责本书的统稿。

在本书的编写过程中,作者参考、借鉴了大量相关资料(见参考文献)及网络资源,并引用了其中一些文字和代码,在此谨对这些作者表示衷心的感谢。

由于作者水平有限,书中难免出现错误,希望广大读者能给予批评。

联系方式:renzhe71@sina.com。

作 者

2007 年 7 月

目 录

第 1 章 微型计算机基础知识

1.1 微型计算机的发展历程	1
1.2 数字电路与计算机	2
1.2.1 数据在计算机中的表示——二进制	2
1.2.2 运算器及二进制数运算	2
1.2.3 数据的存储	8
1.2.4 指令及指令译码	10
1.2.5 程序及程序计数器 PC	11
1.2.6 计算机执行程序的过程	12
1.3 微型计算机系统的基本组成	13
1.3.1 冯·诺依曼计算机结构	14
1.3.2 计算机的硬件系统	14
1.3.3 计算机的软件系统	17
1.4 计算机体系结构的发展	17
1.4.1 RISC	18
1.4.2 指令流水线	19
1.4.3 高速缓存	21
1.4.4 协处理器	21
1.4.5 片上系统	21
1.5 微型计算机的两种主要应用方向	21
1.5.1 桌面系统	21
1.5.2 嵌入式系统	22
1.6 计算机程序设计语言	22
1.6.1 低级语言	22
1.6.2 高级语言	23
1.7 计算机常用标准编码	23
1.7.1 ASCII 码	23
1.7.2 BCD 码	24



1.8 习 题	24
---------	----

第 2 章 总线、存储器和接口

2.1 总 线	26
2.1.1 总线的基本概念	26
2.1.2 系统总线	27
2.1.3 系统总线结构	28
2.1.4 片内总线	30
2.2 存储器	30
2.2.1 半导体存储器的一般结构	31
2.2.2 随机读/写存储器	32
2.2.3 只读存储器	35
2.2.4 存储器的逻辑表示	36
2.3 外部设备及接口	36
2.3.1 外部设备及其特点	37
2.3.2 I/O 接口电路的功能	38
2.3.3 I/O 设备接口电路的基本结构	42
2.3.4 外部设备与处理器的联络和数据传输	42
2.4 常用接口电路	45
2.4.1 并行接口电路	45
2.4.2 串行接口电路	46
2.4.3 定时器/计数器	51
2.5 习 题	52

第 3 章 ARM 体系结构

3.1 ARM 及其嵌入式处理器的研发与生产方式	54
3.1.1 SoC 与嵌入式处理器	54
3.1.2 嵌入式处理器的研发和生产方式	55
3.2 ARM 处理器核的结构	58
3.3 冯·诺依曼结构及哈佛结构在 ARM 中的应用	59
3.4 ARM 处理器的运行模式	60
3.5 ARM 的两种工作状态	61
3.6 ARM 处理器的寄存器	61
3.6.1 ARM 状态下寄存器的组织方式	61



3.6.2 Thumb 状态下寄存器的组织方式	65
3.7 存储器的组织	66
3.8 ARM 体系结构的其他特点	68
3.8.1 灵活方便的协处理器接口	68
3.8.2 嵌入式的在线仿真调试	69
3.8.3 低电压低功耗的设计	69
3.9 ARM 体系结构的版本及处理器系列	69
3.9.1 ARM 体系结构的版本	69
3.9.2 ARM 处理器系列	70
3.10 习 题	71
 第 4 章 ARM 指令系统	
4.1 寻址方式	73
4.1.1 立即寻址方式	73
4.1.2 寄存器寻址方式	74
4.1.3 寄存器间接寻址方式	74
4.1.4 变址寻址	75
4.2 ARM 指令集概述	75
4.2.1 指令的基本格式	76
4.2.2 指令的可选操作	76
4.2.3 指令的第 2 源操作数	78
4.3 ARM 指令集	82
4.3.1 数据传送指令	82
4.3.2 算术运算指令	83
4.3.3 逻辑运算指令	90
4.3.4 测试与比较指令	92
4.3.5 跳转指令	94
4.3.6 程序状态寄存器访问指令	96
4.3.7 加载/存储指令	97
4.3.8 批量数据加载/存储指令	101
4.3.9 数据交换指令	103
4.3.10 异常产生指令	104
4.4 Thumb 指令集简介	104
4.5 习 题	105



第 5 章 ARM 汇编

5.1 汇编器与汇编语言	106
5.2 ARM 伪指令	107
5.2.1 段定义伪指令	107
5.2.2 符号定义伪指令	108
5.2.3 程序中的标号	110
5.2.4 数据定义伪指令	111
5.2.5 汇编控制伪指令	115
5.2.6 其他常用伪指令	116
5.3 宏与宏指令	121
5.3.1 宏	121
5.3.2 宏指令	122
5.4 汇编语言的规范	124
5.4.1 汇编语句格式	124
5.4.2 汇编语言的表达式和运算符	124
5.5 习 题	128

第 6 章 ARM 程序设计

6.1 ARM 工程	129
6.2 ARM 汇编语言程序设计	130
6.2.1 段	130
6.2.2 分支程序设计	131
6.2.3 循环程序设计	135
6.2.4 子程序及其调用	136
6.2.5 汇编程序访问全局 C 变量	137
6.3 ARM 程序框架	138
6.3.1 初始化程序部分	139
6.3.2 初始化部分与主应用程序部分的衔接	139
6.3.3 ARM 开发环境提供的程序框架	140
6.4 C 与汇编之间的函数调用	140
6.4.1 ATPCS 简介	140
6.4.2 C 程序可调用汇编函数实例	143
6.4.3 汇编程序调用 C 函数实例	144

6.5 C/C++语言和汇编语言的混合编程	145
6.5.1 内联汇编	145
6.5.2 嵌入式汇编	146
6.5.3 内联汇编代码与嵌入式汇编代码之间的差异	148
6.6 习 题	148

第7章 中断和异常

7.1 中断和异常的基本概念	151
7.1.1 中断和异常	151
7.1.2 中断请求信号的屏蔽	152
7.1.3 中断优先级及中断嵌套	153
7.1.4 中断服务程序	154
7.1.5 中断向量和中断向量表	155
7.1.6 中断的处理过程	156
7.2 ARM的中断(异常)	157
7.2.1 ARM的中断(异常)向量表	157
7.2.2 ARM中断(异常)的管理	158
7.2.3 ARM中断(异常)运行模式	160
7.2.4 中断(异常)的响应过程及返回	161
7.3 习 题	165

第8章 LPC2000 系列嵌入式处理器

8.1 LPC2000 嵌入式处理器概貌	166
8.2 引脚连接模块	168
8.3 通用可编程并行数据接口 GPIO	169
8.4 LPC2000 的存储器	170
8.4.1 存储空间的分配	170
8.4.2 内部存储器映射	171
8.4.3 I/O 接口端口映射	172
8.4.4 Flash 存储器加速模块	172
8.5 外部存储器控制模块 EMC	173
8.5.1 EMC 的结构	173
8.5.2 EMC 的寄存器	174
8.5.3 外部存储器的连接	175



8.6	LPC2000 的时钟	177
8.6.1	LPC2000 时钟模块	177
8.6.2	celk 时钟参数的设置	178
8.6.3	pclk 时钟参数的设置	179
8.7	定时器	180
8.7.1	定时器的原理和功能	180
8.7.2	定时器的寄存器	181
8.7.3	定时器的初始化	184
8.8	LPC2000 的脉宽调制器 PWM	184
8.8.1	LPC2000 脉宽调制器原理	184
8.8.2	脉宽调制器的寄存器	186
8.8.3	LPC2000 脉宽调制器的使用示例	187
8.9	看门狗	188
8.9.1	看门狗的结构及原理	188
8.9.2	看门狗的寄存器	189
8.9.3	看门狗的使用	190
8.10	功率控制模块	190
8.11	LPC2000 的 UART 接口	191
8.11.1	UART 接口的结构	191
8.11.2	UART 接口的寄存器	193
8.11.3	UART1 接口与 Modem	196
8.11.4	UART 接口的初始化	197
8.12	LPC2000 的 SPI 接口	198
8.12.1	SPI 简介及 LPC2000 的 SPI 接口逻辑	198
8.12.2	LPC2000 SPI 接口的寄存器	199
8.13	LPC2000 的 I ² C 接口	201
8.13.1	I ² C 总线简介	201
8.13.2	LPC2000 的 I ² C 总线接口	204
8.13.3	I ² C 接口的工作过程	207
8.14	A/D 转换器	208
8.15	LPC2000 的中断管理	210
8.15.1	外部中断通道	210
8.15.2	中断控制器 VIC	213
8.16	习 题	218

第 9 章 LPC2000 外部电路

9.1 复位电路	220
9.2 人机交互设备	221
9.2.1 键盘与触摸屏	221
9.2.2 显示器	225
9.3 习 题	230

第 10 章 LPC2000 的固件

10.1 LPC2000 的重映射存储区	231
10.2 LPC2000 的固件及其作用	232
10.3 LPC2000 引导块	235
10.3.1 LPC2000 引导块的地址映射	235
10.3.2 向量表的重映射	236
10.3.3 LPC2114/2124 的 Boot Block 的工作流程	236
10.3.4 有效用户程序的识别	237
10.4 RAM 空间的重映射	238
10.5 习 题	239

第 11 章 LPC2000 程序的组织

11.1 用户程序的启动部分	241
11.2 启动文件	242
11.2.1 汇编语言文件 Startup.s	242
11.2.2 C 语言文件 Target.c	245
11.3 μ C/OS-II 操作系统的使用(移植)	246
11.3.1 μ C/OS-II 简介	246
11.3.2 文件 OS_CPU.H 的移植	247
11.3.3 文件 OS_CPU.C 的移植	251
11.3.4 文件 OS_CPU.A.S 的移植	254
11.4 用户程序的存储与运行	257
11.5 习 题	259

第 12 章 S3C44B0X 处理器简介

12.1 S3C44B0X 的结构	260
-------------------------	-----





12.2 S3C44B0X 的存储器	260
12.2.1 S3C44B0X 存储映射	260
12.2.2 S3C44B0X 用于存储管理的寄存器	262
12.3 时钟和功耗管理	265
12.3.1 时钟设置	265
12.3.2 功耗管理控制	266
12.3.3 功耗管理控制寄存器	267
12.4 中断管理	268
12.4.1 中断源与中断系统结构	268
12.4.2 中断控制寄存器	268
12.4.3 中断源优先排队模块	270
12.5 DMA 控制器	272
12.5.1 ZDMA 和 BDMA	273
12.5.2 与 ZDMA 相关的寄存器	274
12.5.3 与 BDMA 相关的寄存器	276
12.6 LCD 控制器	278
12.6.1 概 述	278
12.6.2 LCD 控制器的控制寄存器	280
12.6.3 帧缓冲区起始地址寄存器	282
12.7 I ² S 总线接口	284
12.7.1 I ² S 总线接口框图	284
12.7.2 I ² S 接口工作模式	286
12.7.3 I ² S 接口的特殊功能寄存器	286
12.8 S3C44B0X 的引脚与 I/O 端口	289
12.8.1 端口配置寄存器 PCON	289
12.8.2 端口数据寄存器 PDATA~PDATAG	290
12.8.3 上拉寄存器 PUPC~PUPG	290
12.9 习 题	290

第 13 章 ARM 高级存储管理

13.1 高速缓冲存储器	291
13.1.1 Cache 的基本概念和工作原理	291
13.1.2 Cache 与主存之间的关系	294
13.1.3 Cache 的写缓冲器	295



13.2 协处理器	296
13.2.1 ARM 的协处理器	297
13.2.2 协处理器操作指令	297
13.3 ARM 存储器保护单元 MPU	299
13.3.1 ARM 的 MPU	300
13.3.2 协处理器 CP15 及保护区域的定义	300
13.3.3 保护区域的重叠应用	306
13.3.4 MPU 应用示例	306
13.4 虚拟存储与 MMU	314
13.4.1 虚拟存储空间与物理内存空间	314
13.4.2 地址映射机构	315
13.4.3 页表的缓存——快表	318
13.4.4 页表的存储及二级页表	319
13.4.5 采用虚拟存储技术的优点	321
13.5 ARM 的虚拟存储管理	322
13.5.1 ARM 的页表结构	322
13.5.2 访问控制	326
13.5.3 MMU 的配置	328
13.5.4 页 Cache 和写缓冲器的设置	328
13.5.5 快 表	329
13.5.6 MMU 异常	330
13.6 习 题	331
附录 S3C44B0X 引脚描述	332
参考文献	335

第 1 章

微型计算机基础知识

自 1946 年世界上出现第一台数字电子计算机 ENIAC 之后,它就以强大的运算能力使世界感到震惊。其后,经半个多世纪的飞速发展,计算机技术应用已经深入到人们生产和生活的各个方面,以至于不管你是否情愿,都必须要与之打交道。

本章的主要内容有:

- 数字电子电路是计算机硬件的基础;
- 二进制数制及机器数;
- 有符号数的表示及溢出的概念;
- 存储器的基本电路及其作用,存储器存储单元的地址;
- 运算器、控制器的基本概念;
- 指令及指令系统,计算机程序的执行过程;
- 计算机的硬件组成;
- SISC 和 RISC 架构;
- 微型计算机的桌面系统应用与嵌入式应用。

1.1 微型计算机的发展历程

根据计算机所使用的电子器件,计算机的发展经历了电子管、晶体管、集成电路、大规模集成电路和超大规模集成电路等几个阶段。同时,计算机在体系结构、组成方式及其体积、性能、价格、应用领域等方面也发生了重大变化。目前,计算机可分为巨型机、大型机、中型机、小型机和微型机五类。其中,人们接触最多的是被叫做“微机”的微型计算机。

微型计算机出现在 20 世纪 70 年代。1971 年,以 Intel 公司设计的世界上第一个微处理器芯片 Intel 4004 为标志,数字电子计算机的发展便进入了微型计算机时代。微型计算机一出现,就表现出了强大的生命力,在短短的几十年间就经过了从第一代到第五代的发展历程:第一代是以 4004、4040 和 8008 微处理器为代表的字长为 4 位和 8 位的微型计算机;第二代是以微处理器 Z80、I8085、M6800 和 Apple-II 等为代表的中高档 8 位机;第三代是以 8086、8088、80286 微处理器为代表的 16 位机;第四代是以 80386、80486、Pentium、Pentium II、



Pentium III、Pentium IV 等微处理器为代表的 32 位机；第五代则是以 Itanium、MIPS 和其他具有精简指令架构(RISC)的微处理器为代表的、目前正在飞速发展的 64 位机。

1.2 数字电路与计算机

数字电子计算机,是一种用电子装置构成,能够按照机器操作者输入的命令进行数字信息处理的机器。从功能上看,它具有两个基本功能:一是表示和存储数字数据的功能;二是对数字数据进行运算的功能。从物理构成的角度来看,它是一种由数字电子器件和电路构成的运算装置。

1.2.1 数据在计算机中的表示——二进制

作为计算机的硬件基础——数字电路,只能用电路信号端子的电信号(电平)来表示数字信息,即规定,低电平代表数码 0(低电平也叫做电路的“0”状态),高电平代表数码 1(高电平也叫做电路的“1”状态),所以它的一个信号端也就只能表示两个数码——0 和 1。但是,如果电路有两个信号端子,那么这两个端子就可以有 4 个状态:00、01、10 和 11。如果用 00 表示十进制数 0,用 01 表示十进制数 1,用 10 表示十进制数 2,用 11 表示十进制数 3,那么这个电路就能表达 4 个十进制数了。这就是说,一个数字电路的信号端子越多,其状态就越多,所能表达的数也就越多。稍微分析一下就会知道,一个具有 n 个信号端子的数字电路所具有的状态数为 2^n ,那么它所能表达的数的个数就为 2^n 个。

例如有 8 个信号端子的数字电路就可以有如下的 256 个状态,并可像下面那样表示 256 个十进制数 0~255:

0000 0000 表示十进制数 0;

0000 0001 表示十进制数 1;

0000 0010 表示十进制数 2;

0000 0011 表示十进制数 3;

0000 0100 表示十进制数 4;

⋮

1111 1111 表示十进制数 255。

从上面左边的表示中可以看到,电路端子的表示是以“逢 2 进 1”为加法运算规则,以“借 1 当 2”为减法运算规则的二进制数制。

1.2.2 运算器及二进制数运算

计算机的核心任务是进行数的运算,该功能是由一个叫做“算术逻辑运算单元(ALU)”的电路来实现的。