第 1 章　计算机系统基础知识

本章首先概要介绍嵌入式系统，然后对计算机系统常用进位计数制、数据的表示和运算、计算机系统硬件基本组成和体系结构以及可靠性与系统性能评测等基础知识进行简要介绍。

1.1　嵌入式计算机系统概述

嵌入式系统的应用范围非常广泛，可以说除了桌面计算机和服务器外，所有计算设备都属于嵌入式系统，例如从便携式音乐播放器到航天飞机上的实时系统控制。

大多数商用的嵌入式系统都设计成专用任务的低成本产品，同时大多数的嵌入式系统都具有实时性的要求。有些功能需要非常快的主频，但大多数功能并不需要高速的处理能力。这些系统通过特定的器件和软件来满足实时性的要求。

简单地通过速度和成本来定义嵌入式系统是困难的，但对于大批量的产品而言，成本常常对系统设计起决定作用。通常，一个嵌入式系统的很多部分相对系统主要功能来说需要较低的性能，因此与通用 PC 相比，嵌入式系统能够使用一个满足辅助功能的合适的 CPU，从而可以简化系统设计，并降低成本。例如，数字电视的机顶盒需要处理每秒百万兆位的连续数据，但这些数据处理大部分是由定制的硬件来实现的，如解析、管理和编解码多个频道的数字影像。

对于大批量生产的嵌入式系统，如便携式音乐播放器或手机等，降低成本就成为最主要的问题。这些系统通常只有以下几个芯片：一个高度集成的 CPU，一个定制的芯片用于控制其他所有的功能，还有一个存储芯片。在这种设计中，每部分都设计成使用最小的系统功耗。

对于小批量的嵌入式应用，为了降低开发成本，常常使用 PC 体系结构，通过限制程序的执行时间或用一个实时操作系统来替换原先的操作系统。在这种情况下，可以使用一个或多个高性能的 CPU 来替换特殊用途的硬件。

嵌入式系统的软件通常运行在有限的硬件资源上：没有硬盘、操作系统、键盘或屏幕。软件一般也没有文件系统，如果有的话，也会采用 Flash 驱动器。如果有人机交互接口，也是一个小键盘或液晶显示器。

嵌入到机械设备中的嵌入式系统需要长期无故障连续运行，因此它的软件需要比 PC 中的软件更加仔细地开发及更加严格的测试。

根据 IEEE（国际电气电子工程师协会）的定义，嵌入式系统是“控制、监视或者辅

助设备、机器和车间运行的装置”。这主要是从应用上加以定义的，从中可以看出嵌入式系统是硬件和软件的综合体，还可以涵盖机械等附属装置。

目前国内一个普遍认同的嵌入式系统定义是：以应用为中心、以计算机技术为基础，软件硬件可裁剪，适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

可以这样认为，嵌入式系统是一种专用的计算机系统，作为装置或设备的一部分。通常，嵌入式系统是一个控制程序存储在 ROM 中的嵌入式处理器控制板。事实上，所有带有数字接口的设备，如手表、微波炉、录像机、汽车等，都使用嵌入式系统，有些嵌入式系统还包含操作系统，但大多数嵌入式系统都是单个程序实现整个控制逻辑。

信息时代和数字时代的到来，为嵌入式系统的发展带来了巨大的机遇，同时也向嵌入式系统厂商提出了新的挑战。目前，嵌入式技术与互联网（Internet）技术的结合正在推动着嵌入式系统的飞速发展，嵌入式系统的研究和应用产生了如下新的显著变化：

- 新的微处理器层出不穷，嵌入式操作系统自身结构的设计更加便于移植，能够在短时间内支持更多微处理器。
- 嵌入式系统的开发是一项系统工程，开发厂商不仅要提供嵌入式软硬件系统本身，同时还要提供强大的硬件开发工具和软件支持包。
- 通用计算机上使用的新技术、新观念开始逐步移植到嵌入式系统中，嵌入式软件平台得到进一步完善。
- 各类嵌入式 Linux 操作系统迅速发展，由于具有源代码开放、系统内核小、执行效率高、网络结构完整等特点，很适合信息家电等嵌入式系统的需要，目前已经形成了能与 Windows CE、Palm OS 等嵌入式操作系统进行有力竞争的局面。
- 网络化、信息化的要求随着 Internet 技术的成熟和带宽的提高而日益突出，以往功能单一的设备，如电话、手机、冰箱、微波炉等，其功能不再单一，结构变得更加复杂，网络互联成为必然趋势。
- 精简系统内核，优化关键算法，降低功耗和软硬件成本。
- 提供更加友好的多媒体人机交互界面。

嵌入式系统是为特定应用而设计的专用计算机系统，是由硬件子系统和软件子系统组成的，通过运行程序来协同工作。硬件是物理装置，软件则是程序、数据和相关文档的集合。

1．计算机硬件

基本的计算机硬件系统由运算器、控制器、存储器、输入设备和输出设备 5 大部件组成，随着网络技术的发展和应用，通信部件也成为计算机系统的基本组件。运算器和控制器及其相关部件已被集成在一起，统称为中央处理单元（Central Processing Unit，CPU）。CPU 是硬件系统的核心，用于数据的加工处理，能完成各种算术、逻辑运算及控制功能。

运算器是对数据进行加工处理的部件，它主要完成算术和逻辑运算。控制器的主要功能是从主存中取出指令并进行分析，以控制计算机的各个部件有条不紊地完成指令的功能。

存储器是计算机系统中的记忆设备，分为内部存储器（Main Memory，MM，简称内存、主存）和外部存储器（简称外存或辅存）。相对而言，内存速度快、容量小，一般用来临时存储计算机运行时所需的程序、数据及运算结果。外存容量大、速度慢，可用于长期保存信息。寄存器是 CPU 中的存储器件，用来临时存放少量的数据、运算结果和正在执行的指令。与内存储器相比，寄存器的速度要快得多。

习惯上将 CPU 和主存储器的有机组合称为主机。输入/输出（I/O）设备位于主机之外，是计算机系统与外界交换信息的装置。所谓输入和输出，都是相对于主机而言的。输入设备的作用是将信息输入到计算机中，输出设备则将运算结果按照人们所要求的形式输出到外部设备或存储介质上。

2．计算机软件

计算机软件是指为管理、运行、维护及应用计算机系统所开发的程序和相关文档的集合。如果计算机系统中仅有硬件系统，则只具备了计算的基础，并不能真正计算，只有将解决问题的步骤编制成机器可识别的程序并加载到计算机内存开始运行，才能完成计算。

软件是计算机系统中的重要组成部分，通常可将软件分为系统软件、中间件和应用软件等类型。系统软件的主要功能是管理系统的硬件和软件资源，应用软件则用于解决应用领域的具体问题，中间件是一类独立的系统软件或服务程序，常用来管理计算资源和网络通信，提供通信处理、数据存取、事务处理、Web 服务、安全、跨平台等服务。

3．计算机分类

从不同角度可对计算机进行不同的分类，个人移动设备、桌面计算机、服务器、集群计算机、超级计算机和嵌入式计算机是其中的一种分类方式。

（1）个人移动设备（Personal Mobile Device，PMD）。指一类带有多媒体用户界面的无线设备，如智能手机、平板电脑等。

（2）桌面计算机。桌面计算机的产品范围非常广泛，包括低端的上网本、台式计算机、笔记本计算机以及高配置的工作站，核心部件是基于超大规模集成电路技术的 CPU。台式计算机和笔记本计算机属于微型计算机，常用于一般性的办公事务处理和应用，工作站则是一种高档的微型计算机，通常配有高分辨率的大屏幕显示器及容量很大的内存储器和外部存储器，具备强大的数据运算与图形、图像处理能力，主要面向工程设计、动画制作、科学研究、软件开发、金融管理、信息服务、模拟仿真等专业应用领域。

（3）服务器。不同于桌面计算机，服务器代替了传统的大型机，主要提供大规模和可靠的文件及计算服务，强调可用性、可扩展性和很高的吞吐率。

（4）集群/仓库级计算机。集群机是将一组桌面计算机或服务器用网络连接在一起，

运行方式类似于一个大型的计算机。将数万个服务器连接在一起形成的大规模集群称为仓库级计算机。

（5）超级计算机。超级计算机的基本组成在概念上与个人计算机无太大差异，但规格高，性能要强大许多，具有很强的计算能力，但是能耗巨大。我国的超级计算机主要有银河、天河、曙光、神威四个系列。例如，神威·太湖之光由 40 个运算机柜和 8 个网络机柜组成，共有 40 960 块处理器，每一块处理器相当于 20 余台常规笔记本计算机的计算能力。

（6）嵌入式计算机。嵌入式计算机是专用的，是针对某个特定的应用，如针对网络、通信、音频、视频或针对工业控制，对功能、可靠性、成本、体积、功耗有严格要求的计算机系统。常见的微波炉、洗衣机、数码产品、网络交换机和汽车中都采用嵌入式计算机技术。

1.2 数据表示

二进制是计算机系统广泛采用的一种数制。在计算机内部，数值、文字、声音、图形图像等各种信息都必须经过数字化编码后才能被传送、存储和处理。

1.2.1 进位计数制及转换

在采用进位计数的数字系统中，如果只用 r 个基本符号表示数值，则称其为 r 进制（Radix-r Number System），r 称为该数制的基数（Radix）。不同数制的共同特点如下。

（1）每一种数制都有固定的符号集。例如，二进制数制的基本符号为 0 和 1，而十进制数制的基本符号为 0，1，2，…，9。

（2）每一种数制都使用位置表示法。即处于不同位置的数符所代表的值不同，与它所在位置的权值有关。例如，十进制数 1234.55 可表示为：

$$1234.55 = 1\times10^3 + 2\times10^2 + 3\times10^1 + 4\times10^0 + 5\times10^{-1} + 5\times10^{-2}$$

计算机中常用的进位数制有二进制、八进制、十进制和十六进制，如表 1-1 所示。

表 1-1 常用计数制

进 位 制	二 进 制	八 进 制	十 进 制	十 六 进 制
规则	逢二进一	逢八进一	逢十进一	逢十六进一
基数	r=2	r=8	r=10	r=16
数符	0，1	0，1，2，…，7	0，1，2，…，9	0，1，2，…，9，A，B，…，F
权	2^i	8^i	10^i	16^i
形式表示符	B	O	D	H

对任何一种进位计数制，其表示的数都可以写成按权展开的多项式，在此基础上实现不同计数制的相互转换。

1）十进制计数法与二进制计数法的相互转换

在二进制计数制中，r=2，基本符号为 0 和 1。二进制数中的一个 0 或 1 称为 1 位(bit)。

将十进制数转换成二进制数时，整数部分和小数部分分别转换，然后再合并。十进制整数转换为二进制整数的方法是“除 2 取余”；十进制小数转换为二进制小数的方法是“乘 2 取整”。

【例 1-1】把十进制数 175.71875 转换为相应的二进制数。

算式	商	余数
175 / 2	87	1
87 / 2	43	1
43 / 2	21	1
21 / 2	10	1
10 / 2	5	0
5 / 2	2	1
2 / 2	1	0
1 / 2	0	1

$175_{10} = 10101111_2$

算式	乘积
0.71875×2	1.43750
0.4375×2	0.8750
0.875×2	1.750
0.75×2	1.50
0.5×2	1.0

$0.71875_{10} = 0.10111_2$

因此，$175.71875_{10} = 10101111.10111_2$

将十进制数写成按二进制数权的大小展开的多项式，按权值从高到低依次取各项的系数就可得到相应的二进制数。

$$\begin{aligned}175.71875_{10} &= 2^7 + 2^5 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} \\ &= 10101111.10111_2\end{aligned}$$

二进制数转换成十进制数的方法是：将二进制数的每一位数乘以它的权再相加，即可求得对应的十进制数值。

【例 1-2】把二进制数 100110.101 转换成相应的十进制数。

$$\begin{aligned}100110.101_2 &= 1\times2^5 + 0\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} \\ &= 32 + 0 + 0 + 4 + 2 + 0 + 0.5 + 0 + 0.125 \\ &= 38.625\end{aligned}$$

2）八进制计数法与十进制、二进制计数法的相互转换

八进制计数制的基本符号为 0，1，2，…，7。

十进制数转换为八进制数的方法是：对于十进制整数采用“除 8 取余”的方法转换为八进制整数；对于十进制小数则采用“乘 8 取整”的方法转换为八进制小数。

二进制数转换成八进制数的方法是：从小数点起，每三位二进制位分成一组（不足 3 位时，在小数点左边时左边补 0，在小数点右边时右边补 0），然后写出每一组的等值八进制数，顺序排列起来就得到所要求的八进制数。

依照同样的思想，将一位八进制数用三位二进制数表示，就可以直接将八进制数转换成二进制数。

二进制与八进制数、十六进制数之间的对应关系如表 1-2 所示。

表 1-2　二进制、八进制和十六进制数之间的对应关系

二进制	八进制	二进制	十六进制	二进制	十六进制
000	0	0000	0	1000	8
001	1	0001	1	1001	9
010	2	0010	2	1010	A
011	3	0011	3	1011	B
100	4	0100	4	1100	C
101	5	0101	5	1101	D
110	6	0110	6	1110	E
111	7	0111	7	1111	F

3）十六进制计数法与十进制、二进制计数法的相互转换

在十六进制计数制中，r=16，基本符号为 0，1，2，…，9，A，B，…，F。

十进制数转换为十六进制数的方法是：十进制数的整数部分“除 16 取余”，十进制数的小数部分“乘 16 取整”。

由于一位十六进制数可以用 4 位二进制数来表示，因此二进制数与十六进制数的相互转换就比较容易。二进制数转换成十六进制数的方法是：从小数点开始，每 4 位二进制数为一组（不足 4 位时，在小数点左边时左边补 0，在小数点右边时右边补 0），将每一组用相应的十六进制数符来表示，即可得到正确的十六进制数。如表 1.2 所示。

【例 1-3】将二进制数 10101111.10111 转换为相应的八进制数和十六进制数。

$$10\,101111.10111_2 = 010\ 101\ 111.101\ 110_2 = 257.56_8$$

$$10\,101111.10111 = 1010\ 1111.1011\ 1000 = AF.B8_{16}$$

1.2.2　数值型数据的表示

数据在计算机中表示的形式称为机器数，其特点是采用二进制计数制，数的符号用 0、1 表示，小数点隐含表示而不占位置。机器数对应的实际数值称为数的真值。

无符号数是指全部二进制位均代表数值，没有符号位。对于有符号数，其机器数的最高位是表示正、负的符号位，其余位则表示数值。若约定小数点的位置在机器数的最低数值位之后，则是纯整数；若约定小数点的位置在机器数的最高数值位之前（符号位之后），则是纯小数。

为了便于运算，带符号的机器数可采用原码、反码、补码和移码等不同的编码方法。

1．原码、反码、补码和移码

1）原码表示

数值 X 的原码记为$[X]_{原}$，如果机器字长为 n（即采用 n 个二进制位表示数据），则最高位是符号位，0 表示正号，1 表示负号，其余的 n–1 位表示数值的绝对值。数值零的原码表示有两种形式：$[+0]_{原}$=00000000，$[-0]_{原}$=10000000。

【例 1-4】若机器字长 n 等于 8，则

$[+1]_{原}$=00000001　　$[-1]_{原}$=10000001

$[+127]_{原}$=01111111　　$[-127]_{原}$=11111111

$[+45]_{原}$=00101101　　$[-45]_{原}$=10101101

$[+0.5]_{原}$=0◊1000000　　$[-0.5]_{原}$=1◊1000000（其中◊表示小数点所在位置）

2）反码表示

数值 X 的反码记作$[X]_{反}$，如果机器字长为 n，则最高位是符号位，0 表示正号，1 表示负号，其余的 n–1 位表示数值。正数的反码与原码相同，负数的反码则是其绝对值按位求反。数值 0 的反码表示有两种形式：$[+0]_{反}$=00000000，$[-0]_{反}$=11111111。

【例 1-5】若机器字长 n 等于 8，则

$[+1]_{反}$=00000001　　$[-1]_{反}$=11111110

$[+127]_{反}$=01111111　　$[-127]_{反}$=10000000

$[+45]_{反}$=00101101　　$[-45]_{反}$=11010010

$[+0.5]_{反}$=0◊1000000　　$[-0.5]_{反}$=1◊0111111（其中◊表示小数点所在位置）

3）补码表示

数值 X 的补码记作$[X]_{补}$，如果机器字长为 n，则最高位为符号位，0 表示正号，1 表示负号，其余的 n–1 位表示数值。正数的补码与其原码和反码相同，负数的补码则等于其反码的末尾加 1。在补码表示中，0 有唯一的编码：$[+0]_{补}$=$[-0]_{补}$=00000000。

【例 1-6】若机器字长 n 等于 8，则

$[+1]_{补}$=00000001　　$[-1]_{补}$=11111111

$[+127]_{补}$=01111111　　$[-127]_{补}$=10000001

$[+45]_{补}$=00101101　　$[-45]_{补}$=11010011

$[+0.5]_{补}$=0◊1000000　　$[-0.5]_{补}$=1◊1000000（其中◊是小数点的位置）

相对于原码和反码表示，n 位补码表示法有一个例外，当符号位为 1 而数值位全部为 0 时，它表示整数-2^{n-1}，即此时符号位的 1 既表示负数又表示数值。

设计补码时，有意识地引用了模运算在数理上对符号位的处理，即利用模的自动丢弃实现了符号位的自然处理。

4）移码表示

移码表示法是在数 X 上增加一个偏移量来定义的，常用于表示浮点数中的阶码。如果机器字长为 n，在偏移量为 2^{n-1} 时，只要将补码的符号位取反便可获得相应的移码表示。

偏移量也可以是其他值。采用移码表示时，码值大者对应的真值就大。

【例 1-7】若机器字长 n 等于 8，偏移量为 2^7，则

$[+1]_{移}$=10000001　　$[-1]_{移}$=01111111

$[+127]_{移}$=11111111　　$[-127]_{移}$=00000001

$[+45]_{移}$=10101101　　$[-45]_{移}$=01010011

$[+0]_{移}$=10000000　　$[-0]_{移}$=10000000

2．定点数和浮点数

1）定点数

所谓定点数，就是表示数据时小数点的位置固定不变。小数点的位置通常有两种约定方式：定点整数（纯整数，小数点在最低有效数值位之后）和定点小数（纯小数，小数点在最高有效数值位之前）。

设机器字长为 n，各种码制表示下的带符号数的范围如表 1-3 所示。当机器字长为 n 时，定点数的补码和移码可表示 2^n 个数，而其原码和反码只能表示 2^n-1 个数（0 表示占用了两个编码），因此，定点数所能表示的数值范围比较小，运算中很容易因结果超出范围而溢出。

表 1-3　机器字长为 n 时各种码制表示的带符号数的范围

码　制	定 点 整 数	定 点 小 数
原码	$-\left(2^{n-1}-1\right)$ ～ $+\left(2^{n-1}-1\right)$	$-\left(1-2^{-(n-1)}\right)$ ～ $+\left(1-2^{-(n-1)}\right)$
反码	$-\left(2^{n-1}-1\right)$ ～ $+\left(2^{n-1}-1\right)$	$-\left(1-2^{-(n-1)}\right)$ ～ $+\left(1-2^{-(n-1)}\right)$
补码	-2^{n-1} ～ $+\left(2^{n-1}-1\right)$	-1 ～ $+\left(1-2^{-(n-1)}\right)$
移码	-2^{n-1} ～ $+\left(2^{n-1}-1\right)$	-1 ～ $+\left(1-2^{-(n-1)}\right)$

2）浮点数

浮点数是指小数点位置不固定的数，浮点表示法能表示更大范围的数。

在十进制中，一个实数可以写成多种表示形式。例如，83.125 可写成 $10^3\times0.083125$ 或 $10^4\times0.0083125$ 等。同理，一个二进制数也可以写成多种表示形式。例如，二进制数 1011.10101 可以写成 $2^4\times0.101110101$、$2^5\times0.0101110101$ 或 $2^6\times0.00101110101$ 等。

一个含小数点的二进制数 N 可以表示为更一般的形式：

$$N=2^E\times F$$

其中，E 称为阶码，F 为尾数，这种表示数的方法称为浮点表示法。

在浮点表示法中，阶码通常为带符号的纯整数，尾数为带符号的纯小数。浮点数的表示格式一般如下：

阶符	阶码	数符	尾数

显然，一个数的浮点表示不是唯一的。当小数点的位置改变时，阶码也相应改变，因此可以用多种浮点形式表示同一个数。

浮点数所能表示的数值范围主要由阶码决定，所表示数值的精度则由尾数决定。

为了提高数据的表示精度，当尾数的值不为 0 时，规定尾数域的最高有效位应为 1，这称为浮点数的规格化表示，否则需修改阶码左移或右移小数点的位置，使其变为规格化数的形式。

3）工业标准 IEEE 754

IEEE 754 是由 IEEE 制定的有关浮点数的工业标准，被广泛采用。该标准的表示形式如下：

S	*P*	*M*

其中，*S* 为数的符号位，为 0 时表示正数，为 1 时表示负数；*P* 为指数（阶码），用移码表示（偏移值为 $2^{p-1}-1$，*p* 为阶码的位数）；*M* 为尾数，用原码表示。

对于阶码为全 0 或全 1 的情况，IEEE 754 标准有特别的规定：若 *P* 为全 0 且 *M* 为 0，则表示真值±0（正负号和数符位有关）。如果 *P* 为全 1 且 *M* 是 0，则这个数的真值为±∞（与符号位有关）；如果 *P* 为全 1 并且 *M* 不是 0，则规定其不是一个数（NaN）。

目前，计算机中主要使用三种形式的 IEEE 754 浮点数，如表 1-4 所示。

表 1-4　三种形式的 IEEE 754 浮点数格式

参　　数	单精度浮点数	双精度浮点数	扩充精度浮点数
浮点数字长	32	64	80
尾数长度	23	52	64
符号位长度	1	1	1
阶码长度	8	11	15
指数偏移量	+127	+1023	+16 383
可表示的实数范围	$10^{-38}\sim10^{38}$	$10^{-308}\sim10^{308}$	$10^{-4932}\sim10^{4932}$

在 IEEE 754 标准中，对于单精度浮点数和双精度浮点数，约定小数点左边隐含有一位，通常这位数就是 1，因此尾数为 1.××…×。

【例 1-8】利用 IEEE 754 标准将数 176.0625 表示为单精度浮点数。

解：首先将该十进制数转换成二进制数，即$176.0625_{10}=10110000.0001_2$，其次对二进制数进行规格化处理，即 10110000.0001=1◊01100000001×2^7。这就保证了最高位为 1，而且小数点应当在◊位置上，将最高位去掉并扩展为单精度浮点数所规定的 23 位尾数，得到 01100000001000000000000。

然后求阶码，上述表示中指数为 7，用移码表示为 10000110（偏移量是 127，因此偏移后的指数值为 7＋127=134）。

最后，得到176.0625_{10}的单精度浮点表示形式：

0 10000110 01100000001000000000000

1.2.3 其他数据的表示

各类数据的表示都有相应的基本字符集，任何字符在计算机中都必须转换成二进制表示形式，称为字符编码。

1．十进制数与字符的编码表示

用 4 位二进制代码表示一位十进制数，称为二-十进制编码，简称 BCD 编码。因为2^4=16，而十进制数只有 0～9 这 10 个不同的数符，故有多种 BCD 编码。根据 4 位代码中每一位是否有确定的权来划分，可分为有权码和无权码两类。

应用最多的有权码是 8421 码，即 4 个二进制位的权从高到低分别为 8、4、2 和 1。无权码中常用余 3 码和格雷码（有多种编码形式）。余 3 码是在 8421 码的基础上，把每个数的代码加上 0011 后构成的。格雷码的编码规则是相邻的两个代码之间只有 1 位不同。

常用的 8421BCD 码、余 3 码、格雷码与十进制数的对应关系如表 1-5 所示。

表 1-5 8421BCD 码、余 3 码、格雷码与十进制数的对应关系

十 进 制 数	8421BCD 码	余 3BCD 码	格 雷 码
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0011
3	0011	0110	0010
4	0100	0111	0110
5	0101	1000	0111
6	0110	1001	0101
7	0111	1010	0100
8	1000	1011	1100
9	1001	1100	1101

2．ASCII 码

美国标准信息交换代码（American Standard Code for Information Interchange，ASCII）被国际标准化组织 ISO 采纳，成为一种国际通用的信息交换用标准代码。基本的 ASCII 码采用 7 个二进制位，即 $d_6d_5d_4d_3d_2d_1d_0$ 对字符进行编码：低 4 位组 $d_3d_2d_1d_0$ 用作行编码，高 3 位组 $d_6d_5d_4$ 用作列编码。基本的 ASCII 字符代码表如表 1-6 所示。

表 1-6　7 位 ASCII 代码表

$d_3d_2d_1d_0$ 位（低 4 位）	$d_6d_5d_4$ 位（高 3 位）							
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	,	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	'	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	DI	US	/	?	O	↓	o	Del

根据 ASCII 码的构成格式，可以很方便地从对应的代码表中查出每一个字符的编码。例如，字符 0 的 ASCII 码值为 0110000（$2^5+2^4=48$），字符 a 的 ASCII 码值为 1100001（$2^6+2^5+2^0=97$）。

3．汉字编码

计算机中处理汉字时，必须先将汉字代码化，即对汉字进行编码。汉字处理包括汉字的编码输入、汉字的存储和汉字的输出等环节。

西文是拼音文字，基本符号比较少，比较容易编码，在计算机系统中输入、内部处理、存储和输出都可以使用同一代码。汉字种类繁多，编码比拼音文字困难，而且在一个汉字处理系统中，输入、内部处理、存储和输出对汉字代码的要求不尽相同，所以采用的编码也不同。汉字信息处理系统在处理汉字和词语时，关键的问题是要进行一系列的汉字代码转换。

1）输入码

中文字数繁多，字形复杂，字音多变，常用汉字就有 7000 个左右。为了能直接使用西文标准键盘输入汉字，必须为汉字设计相应的编码方法，汉字的输入码主要分为三类：数字编码、拼音码和字形码。

（1）数字编码。数字编码就是用数字串代表一个汉字的输入，常用的是国标区位码。国标区位码将国家标准局公布的 6763 个两级汉字分成 94 个区，每个区 94 位，区码和位码各两位十进制数字。例如，“中”字位于第 54 区 48 位，区位码为 5448。

汉字在区位码表的排列是有规律的。在 94 个分区中，1～15 区用来表示字母、数字和符号，16～87 区为一级和二级汉字。一级汉字以汉语拼音为序排列，二级汉字以偏旁部首进行排列。使用区位码方法输入汉字时，必须先在表中查找汉字对应的代码，才能输入。数字编码输入的优点是无重码，而且输入码和内部编码的转换比较方便，但是数字码有难以记忆的缺点。

（2）拼音码。拼音码是以汉语读音为基础的输入方法。由于汉字同音字太多，输入重码率很高，因此，按拼音输入后还必须进行同音字选择，会影响输入速度。

（3）字形编码。字形编码是以汉字的形状确定的编码。汉字总数虽多，但都是由一笔一划组成，全部汉字的部件和笔划是有限的。因此，把汉字的笔划部件用字母或数字进行编码，按笔划书写的顺序依次输入，就能表示一个汉字，五笔字型、表形码等便是这种编码法。

2）内部码

汉字内部码（简称汉字内码）是汉字在设备和信息处理系统内部存储、处理、传输汉字用的代码。汉字数量多，用一个字节无法区分，采用国家标准局 GB 2312—1980 中规定的汉字国标码，两个字节存放一个汉字的内码，每个字节的最高位置 1，作为汉字机内码。由于两个字节各用 7 位，因此可表示 16 384 个可区别的机内码。以汉字“大”为例，国标码为 3473H，两个字节的高位置 1，得到的机内码为 B4F3H。

GB 18030—2005《信息技术中文编码字符集》是我国最新的内码字符集，与 GB 2312—1980 完全兼容，支持 GB 13000 及 Unicode 的全部统一汉字，共收录汉字 70244 个。

3）字形码

汉字字形码是表示汉字字形的字模数据，通常用点阵、矢量函数等方式表示，用点阵表示字形时，汉字字形码指的就是这个汉字字形点阵的代码。字形码也称字模码，是用点阵表示的汉字字形码，它是汉字的输出方式。根据输出汉字的要求不同，点阵的多少也不同。简易型汉字为 16×16 点阵，高精度型汉字为 24×24 点阵、32×32 点阵、48×48 点阵等。

字模点阵的信息量是很大的，所占存储空间也很大，以 16×16 点阵为例，每个汉字就需要 32 字节用于机内存储。字库中存储了每个汉字的点阵代码，当显示输出时才检索字库，输出字模点阵得到字形。

汉字的矢量表示法是将汉字看作由笔画组成的图形，提取每个笔画的坐标值，这些坐标值就可以决定每一笔画的位置，将每一个汉字的所有坐标值信息组合起来就是该汉字字形的矢量信息。显然，汉字的字形不同，其矢量信息也就不同，每个汉字都有自己的矢量信息。由于汉字的笔画不同，则矢量信息就不同。所以，每个汉字矢量信息所占的内存大小不一样。同样，将每一个汉字的矢量信息集中在一起就构成了汉字库。当需要汉字输出时，利用汉字字形检索程序根据汉字内码从字模库中找到相应的字形码。

4．Unicode

为了统一地表示世界各国的文字，国际标准化组织 1993 年公布了“通用多八位编码

第 2 章　嵌入式系统硬件基础知识

本章主要介绍嵌入式系统所涉及的硬件知识，重点是嵌入式微处理器、嵌入式存储体系、嵌入式系统的输入/输出接口、嵌入式系统通信接口等方面进行的硬件接口基础知识。

2.1　数字电路基础

2.1.1　信号特征

现代计算机内部的电子元器件都是数字式的。数字式的电子元器件工作状态是二值电平：高电平和低电平。这也是计算机采用二进制的主要原因。通常不指定具体的电平值，而是采用信号来表示，如，用“逻辑真”“1”或“确定”来表示高电平，而用“逻辑假”“0”或“不确定”来表示低电平。1 和 0 称为互补信号。

根据电路是否具有存储功能，将逻辑电路划分为两种类型：组合逻辑电路和时序逻辑电路。组合逻辑电路不含存储功能，它的输出值仅取决于当前的输入值；时序逻辑电路含存储功能，它的输出值不仅取决于当前输入状态，还取决于存储单元中的值。

2.1.2　组合逻辑电路和时序逻辑电路

1．组合逻辑电路

所谓组合逻辑电路，是指该电路在任一时刻的输出，仅取决于该时刻的输入信号，而与输入信号作用前电路的状态无关。组合逻辑电路一般由门电路组成，不含记忆元器件，输入与输出之间无反馈。常用的组合逻辑电路有译码器和多路选择器等。

1）真值表

由于组合电路中不包含任何存储单元，所以组合电路的输出值可由当前输入值完全确定。这种确定的对应关系可以由真值表（true table）来描述。例如，对于有 n 个输入的逻辑电路，对应的真值表有 2^n 种输入组合，每一种输入组合表示一组输入状态集，分别对应一个确定的输出。

通常，真值表能够完全描述任何一种组合逻辑函数，但是表的大小随着输入个数的增加呈指数增长，而且不够清晰。

2）布尔代数

描述逻辑函数的另外一种方法是逻辑表达式，可以通过布尔代数（Boolean algebra）

实现。布尔代数中有 3 种典型的操作符：OR、AND 和 NOT。

- OR（“或”）操作符，记为“+”，也称为逻辑和（logical sum）。如 $A+B$，若 A 和 B 中至少有一位为 1 时，则结果为 1。
- AND（“与”）操作符，记为“ • ”，也称为逻辑乘（logical product）。如 $A \cdot B$，当且仅当输入值都为 1 时，其结果才为 1。
- NOT（“非”）操作符，记为“ $\overline{A}$ ”，也称为逻辑非。当输入为 0 时，输出为 1；当输入为 1 时，输出为 0。

3）门电路

门电路可以实现基本的逻辑功能。基本的门电路如图 2-1 所示，包括与门、或门和非门。

图 2-1　基本门电路

通常在信号的输入或输出端加上一个“。”表示对输入/输出信号取非。任何一个逻辑表达式都可以用与门、非门和或门的组合来表示。如果允许某个门电路取非，那么任何一个逻辑图函数都可以仅用与门或仅用或门实现。常见的两种反向门电路为 NOR 和 NAND，它们分别对应或门、与门的取非。NOR 和 NAND 的门电路称为全能门电路，因为任何一种逻辑函数可以用这种门电路得以实现。

4）译码器

译码器又称为解码器（decoder），译码器是一种多输入多输出的组合逻辑网络，它有 n 个输入端，m 个输出端。与译码器对应的是编码器（encoder），它实现的是译码器的逆功能。译码器的框图如图 2-2 所示。

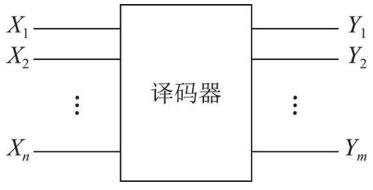


图 2-2　译码器

每输入一个 n 位的二进制代码，在 m 个输出端中最多有一个有效。译码器的输入端和输出端之间应满足下列关系：

$$m \leqslant 2^n$$

$m=2^n$ 时，称为全译码；当 $m<2^n$ 时，称为部分译码。

5）数据选择器和数据分配器

数据选择器又称多路开关，它是以“与或”门或“与或非”门为主的电路。它可以在选择信号的作用下，从多个输入通道中选择某一个通道的数据作为输出。常见的数据选择器有二选一、四选一、八选一、十六选一等。

数据选择器除有选择输入信号的功能外，还可利用它实现任意组合逻辑函数。例如四选一的数据选择器可以实现三个变量的组合逻辑函数，2*n* 个数据输入的多路选择器可实现 *n*+1 个变量的组合逻辑函数。

数据分配器又称多路分配器，它有一个输入端和多个输出端，其逻辑功能是将一个输入端的信号送至多个输出端中的某一个，简称 DMUX，作用与 MUX 正好相反。数据分配器的核心部分实际上是一个带有使能端的全译码器，可以把数据分配器理解为是输出受 X 控制的译码器。

2．时序逻辑线路

所谓时序逻辑电路，是指电路任一时刻的输出不仅与该时刻的输入有关，而且还与该时刻电路的状态有关。因此，时序逻辑电路中必须包含记忆元器件。触发器是构成时序逻辑电路的基础。常用的时序逻辑电路有寄存器和计数器等。

1）时钟信号

时钟信号是时序逻辑的基础，它用于决定逻辑单元中的状态何时更新。时钟信号是指固定周期并与运行无关的信号量，时钟频率（Clock Frequency，CF）是时钟周期的倒数。如图 2-3 所示，时钟周期（Clock cycle Time）由两部分内容组成：高电平和低电平。时钟边沿触发信号（Edge-triggered 时钟周期 locking）意味着所有的状态变化都发生在时钟边沿到来时刻。

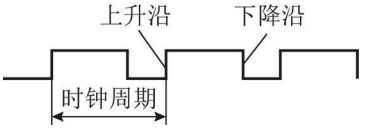


图 2-3　时钟信号

在边沿触发机制中，只有上升沿或下降沿才是有效信号，才能控制逻辑单元状态量的变。至于到底是上升沿还是下降沿作为有效触发信号，则取决于逻辑设计的技术。

同步是时钟控制系统中的主要制约条件。同步就是指在有效信号沿发生时刻，希望写入单元的数据也有效。数据有效则是指数据量比较稳定（不发生改变），并且只有当输入发生变化时数值才会发生变化。由于组合电路无法实现反馈，所以只要输入量不发生变化，输出值最终会是一个稳定有效的量。

2）触发器

触发器种类很多。按时钟控制方式来分，有电位触发、边沿触发、主-从触发等方式。

按功能分类，有 R-S 型、D 型、J-K 型等功能。同一功能触发器可以由不同触发方式

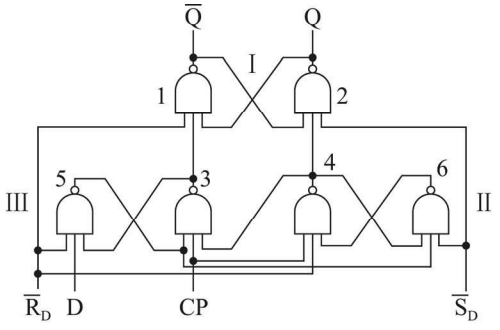
来实现。对使用者来说，在选用触发器时，触发方式是必须考虑的因素。因为相同功能的触发器，若触发方式选用不当，系统达不到预期设计要求。这里将以触发方式为线索，介绍几种常用的触发器。

（1）电位触发方式触发器。当触发器的同步控制信号 E 为约定“1”或“0”电平时，触发器接收输入数据，此时输入数据 D 的任何变化都会在输出 Q 端得到反映；当 E 为非约定电平时，触发器状态保持不变。鉴于它接收信息的条件是 E 出现约定的逻辑电平，故称它为电位触发方式触发器，简称电位触发器。

电位触发器具有结构简单的优点。在计算机中常用它来组成暂存器。

（2）边沿触发方式触发器。具有如下所述特点的触发器称为边沿触发方式触发器，简称边沿触发器。触发器是时钟脉冲 CP 的某一约定跳变（正跳变或负跳变）来到时的输入数据。在 CP=1 及 CP=0 期间以及 CP 非约定跳变到来时，触发器不接收数据。

常用的正边沿触发器是 D 触发器，图 2-4 给出了它的逻辑图及功能表。



功能表

$\overline{R}_D$	$\overline{S}_D$	CP	D	Q	$\overline{Q}$
0	1	×	×	0	1
1	0	×	×	1	0
1	1	↑	0	0	0
1	1	↑	1	1	0

图 2-4　D 触发器逻辑图

下面比较边沿触发器和电位触发器。

电位触发器在 E=1 期间来到的数据会立刻被接收。但对于边沿触发器，在 CP=1 期间来到的数据，必须“延迟”到该 CP=1 过后的下一个 CP 边沿来到时才被接收。因此边沿触发器又称延迟型触发器。

边沿触发器在 CP 正跳变（对正边沿触发器）以外期间出现在 D 端的数据和干扰不会被接收，因此有很强的抗数据端干扰的能力而被广泛应用，它除用来组成寄存器外，还可用来组成计数器和移位寄存器等。

至于电位触发器，只要 E 为约定电平，数据来到后就可立即被接收，它不需像边沿触发器那样保持到约定控制信号跳变到来才被接收。

（3）触发器的开关特性。描述触发器的参数很多，其中既有描述传输延迟的参数，也有描述各输入波形宽度要求的参数，还有描述各输入波形之间时间配合要求的参数。如果在使用时不能满足参数的要求，电路就不能正常地工作。

3）寄存器与移位器

寄存器主要用来接收信息、寄存信息或传送信息，通常采用并行输入一并行输出的

方式。由于一个触发器仅能寄存一位二进制代码，所以要寄存 *n* 位进制代码，就需要具备 *n* 个触发器。随着组成寄存器的触发器的触发方式不同，寄存器也有不同的触发方式，最常用的是正跳沿触发的 D 触发器，这种寄存器的各位在同一时刻（CP 脉冲的上升沿作用下）接收信息。也有一些寄存器的信息接收是通过电位信号（使能 G）控制的，即高电平触发，这种寄存器又称为锁存器，其主要用途是把一些短暂的信号锁存（锁住并保存）起来，以达到时间上的扩展。

寄存器中除具有若干触发器以外，还应有门电路构成的控制电路，以保证信息的正确接收、发送和清除。

在时钟信号控制下，将所寄存的信息向左或向右移位的寄存器称为移位寄存器。按照信息移动方向的不同，移位寄存器可以分为单向（左移或右移）及双向移位寄存器。按照信息的输入/输出方式不同，移位寄存器可以分为：串行输入—串行输出、串行输入—并行输出和并行输入—串行输出 3 种工作方式。从移位寄存器的外部特征来看，串行输入—串行输出的移位器仅需要一条数据输入线和一条数据输出线，而串行输入—并行输出的移位器需要一条数据输入线和多条数据输出线，并行输入—串行输出的移位器需要多条数据输入线和一条数据输出线。将串行输入信息变换成并行输出信息的过程，称为“串—并变换”，反之，将并行输入信息变换为串行输出信息的过程，称为“并—串变换”，这在计算机的接口电路中使用十分广泛。

2.1.3 信号转换

1．数字集成电路的分类

按照开关元件的不同，数字集成电路可以分为两大类：一类是双极型集成电路，采用晶体管作为开关元件，管内参与导电的有电子和空穴两种极性的载流子。另一类采用绝缘栅场效应晶体管作开关元件，称为金属氧化物半导体（Metal-oxide Semiconductor，MOS）集成电路。这种管子内部只有一种载流子——电子或空穴参与导电，故又称单极型集成电路。

晶体管−晶体管逻辑电路（Transistor-Transistor Logic，TTL）是目前双极型数字集成电路中用得最多的一种。它具有比较快的开关速度、比较强的抗干扰能力以及足够大的输出幅度，并且带负载能力也比较强，所以得到了最为广泛的应用。在双极型数字集成电路中，除了 TTL 电路以外，还有二极管−三极管逻辑（Diode-Transistor Logic，DTL）、高阈值逻辑（High Threshold Logic，HTL）、发射极耦合逻辑（Emitter Coupled Logic，ECL）和集成注入逻辑（Integrated Injection Logic，IL）等几种逻辑电路。

ECL 电路中的三极管工作在非饱和状态，是一种非饱和电路，有极高的工作速度，此外它还具有输出阻抗低，带负载能力强，电路内部开关噪声低，使用方便灵活等优点。它的主要缺点是：噪声容限低，电路功耗大，输出电平的稳定性较差。目前 ECL 电路主要用于高速、超高速数字系统中。

按照所用 MOS 管类型的不同，可分为 3 种：由 PMOS 管构成的 PMOS 集成电路；由 NMOS 管构成的 NMOS 集成电路；由 PMOS 管和 NMOS 管构成的互补（Complementary）MOS 集成电路，简称 CMOS。PMOS 和 NMOS 组件中各只含有一种 MOS 管，习惯上称它们为 MOS 集成电路，以与 CMOS 集成电路相区别。

PMOS 集成电路问世较早，但由于其速度低，现已很少使用；NMOS 集成电路速度稍高，且直流电源电压较低，在工艺上可以制造出开启电压较低的器件，故 NMOS 集成电路仍在使用中。CMOS 电路由于其静态功耗极低，工作速度较高，抗干扰能力强，故被广泛采用。

2．常用电平接口技术

虽然 TTL 电路具有很多优点，但它毕竟不能满足生产中不断提出来的各种特殊要求，例如高速、高抗干扰、低功耗等，因而在工程中还经常用到 ECL、CMOS 等数字集成电路。在微机测控系统中，习惯于用 TTL 电路作为基本电路元件，根据需要可能采用 CMOS、ECL 等芯片，因此存在 TTL 电路与这些数字电路的接口问题。

ECL 的特点是速度快，但抗干扰性能差，功耗也高；TTL 的应用广泛，成本低廉，有许多种类可供选择；CMOS 功耗最低，抗干扰性能优良，不仅适用于中、小规模集成电路，而且在大规模集成组件中应用也很普遍。

下面讨论 TTL 与 ECL 和 CMOS 之间的电平转换接口。

1）TTL 与 ECL 电平转换接口

ECL 电路电压一般为−5.2V（CE10K 系列），逻辑高电平输出为 VOH=−0.9V，低电平为−1.75V；对 CE100K 系列电源电压为−4.5V，输出高电平为 VOH=−0.955V，低电平 VOL=−1.705V。

（1）TTL→ECL 转换。利用集成芯片 CE1024 即可完成 TTL 到 ECL 的电平转换。它有一个公共的选通脉冲输入端 B，若 B 为低电平，ECL 的所有 Y 为低电平，$\overline{Y}$ 为高电平。

（2）ECL→TTL 转换。CE10125 为四 ECL—TTL 电平转换器，它的输入与 ECL 电平兼容，具有差分输入和抑制±1V 共态干扰输入能力，输出是 TTL 电平。如果有某路不用时，须将其一个输入端接到 VBB 端上，以保证电路的工作稳定性。

在小型系统中，ECL 和 TTL 可能均使用+5V 电源，此时需用分立元件来实现接口。

2）TTL 与 CMOS 电平转换接口

CMOS 反相器当其使用电源电压为 5V 时，输出低电平电压最大值为 0.05V，高电平最小值为 4.95V，输出低电平电流最小为 0.5mA，高电平电流最小为−0.5mA；对于带缓冲门的 CMOS 电路，当供电电源电压为 5V 时，VIL≤1.5V，VIH≥3.5V。而对于不带缓冲门的 CMOS 电路，VIL≤1V，VIH≥4V。

（1）TTL→CMOS 转换。由于 TTL 电路输出高电平的规范值为 2.4V，在电源电压为 5V 时，CMOS 电路输入高电平 VIH≥3.5V。这样就造成了 TTL 与 CMOS 电路接口

上的困难。解决的办法是在 TTL 电路输出端与电源之间接一上拉电阻 R。上拉电阻 R 的取值由 TTL 的高电平输出漏电流 IOH 来决定，不同系列的 TTL 应选用不同的 R 值。一般有：

- 74 系列，4.7kΩ≥R≥390Ω。
- 74H 系列，4.7kΩ≥R≥270Ω。
- 74L 系列，27kΩ≥R≥1.5kΩ。
- 74LS 系列，12kΩ≥R≥820Ω。

如果 CMOS 电路的电源电压高于 TTL 电路的电源电压。同时 CMOS 电路应使用具有电平移位功能的电路，如 CC4504、CC40109 及 BH017 等，至于 CMOS 电路的电源电压可在 5～15V 范围内任意选定。

（2）CMOS→TTL 转换。关于 CMOS 到 TTL 的接口，由于 TTL 电路输入短路电流较大，就要求 CMOS 电路在 VOL 为 0.5V 时能给出足够的驱动电流，因此需使用 CC4049、CC4050 等作为接口器件。

2.1.4 可编程逻辑器件

随着微电子技术的发展，设计与制造集成电路的任务已不完全由半导体厂商来独立承担。系统设计师们更愿意自己设计专用集成电路（Application Specific Integrated Circuit，ASIC）芯片，而且希望 ASIC 的设计周期尽可能短，最好是在实验室里就能设计出合适的 ASIC 芯片，并且立即投入实际应用之中，因而出现了现场可编程逻辑器件（Field Programmable Logic Device，FPLD），其中应用最广泛的当属现场可编程门阵列（Field Programmable Gate Array，FPGA）和复杂可编程逻辑器件（Complex Programmable Logic Dvice，CPLD）。

早期的可编程逻辑器件只有可编程只读存储器（PROM）、紫外线可擦除只读存储器（EPROM）和电可擦除只读存储器（EEPROM）3 种。由于结构的限制，它们只能完成简单的数字逻辑功能。

其后，出现了一类结构上稍复杂的可编程芯片，即可编程逻辑器件（PLD），它能够完成各种数字逻辑功能。典型的 PLD 由一个“与”门和一个“或”门阵列组成，而任意一个组合逻辑都可以用“与-或”表达式来描述，所以，PLD 能以乘积和的形式完成大量的组合逻辑功能。

这一阶段的产品主要有可编程阵列逻辑（Programmable Array Logic，PAL）和通用阵列逻辑（Generic Array Logic，GAL）。PAL 由一个可编程的“与”平面和一个固定的“或”平面构成，“或”门的输出可以通过触发器有选择地被置为寄存状态。PAL 器件是现场可编程的，它的实现工艺有反熔丝技术、EPROM 技术和 EEPROM 技术。还有一类结构更为灵活的逻辑器件是可编程逻辑阵列，它也由一个“与”平面和一个“或”平面构成，但是这两个平面的连接关系是可编程的。PLA 器件既有现场可编程的，也有掩膜

可编程的。在 PAL 的基础上，又发展了一种通用阵列逻辑，如 GAL16V8 和 GAL22V10 等。它采用了 EEPROM 工艺，实现了电可擦除、电可改写，其输出结构是可编程的逻辑宏单元，因而它的设计具有很强的灵活性，至今仍有许多人使用。这些早期的 PLD 器件的一个共同特点是可以实现速度特性较好的逻辑功能，但其过于简单的结构也使它们只能实现规模较小的电路。

为了弥补这一缺陷，20 世纪 80 年代中期。Altera 和 Xilinx 公司分别推出了类似于 PAL 结构的扩展型复杂可编程逻辑器件和与标准门阵列类似的现场可编程门阵列，它们都具有体系结构和逻辑单元灵活、集成度高以及适用范围宽等特点。这两种器件兼容了 PLD 和通用门阵列的优点，可实现较大规模的电路，编程也很灵活。与门阵列等其他专用集成电路相比，它们又具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无需测试、质量稳定以及可实时在线检验等优点，因此被广泛应用于产品的原型设计和产品生产（一般在 10000 件以下）之中。几乎所有应用门阵列、PLD 和中小规模通用数字集成电路的场合均可应用 FPGA 和 CPLD 器件。

2.2　嵌入式微处理器基础

嵌入式操作系统硬件架构的核心是处理器（Central Processing Unit，CPU），负责从内存中取指并执行。在每个 CPU 的指令执行周期中，包括从内存取出指令，解码确定类型和操作数后再执行该条指令；然后再取指、解码并执行下一条指令，循环往复直至程序执行完毕。处理器的架构可采用冯 •诺依曼结构（见图 2-5）或者哈佛结构（见图 2-6）。

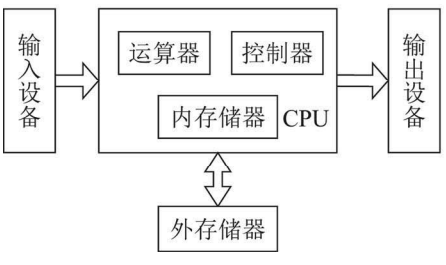


图 2-5　冯 • 诺依曼结构

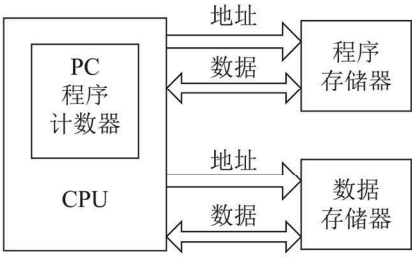


图 2-6　哈佛结构

每个 CPU 都有其一套可执行的专门指令集。早期的计算机硬件结构比较简单，因而其指令系统很简单，指令种类较少，功能较简单。随着计算机硬件成本的降低，促使 CPU 的设计者在指令系统中增加了更多指令，其功能更强，当然其设置更为复杂。指令系统愈加复杂。通常将具有复杂指令系统的计算机称为复杂指令集计算机（Complex Instruction Set Computer，CISC），与之相对为精简指令集计算机（Reduced Instruction Set Computer，RISC）。两者的特点及对比见表 2-1 所示。

表 2-1 复杂指令集和精简指令集的比较

条　目	CISC	RISC
价格	由硬件完成部分软件功能，硬件复杂性增加，芯片成本高	由软件部分完成部分硬件功能，软件复杂性增加，芯片成本低
性能	减少代码尺寸，增加指令的执行周期数	使用流水线降低指令的执行周期数，增加代码尺寸
指令集	指令系统复杂，指令数目多达 200～3000 条	只设置使用频度高的一些简单指令，复杂指令的功能由多条简单指令组合而实现
高级语言支持	硬件完成	软件完成
寻址方式	较多的寻址方式	寻址方式种类少
控制器	大多采取微程序控制器实现	用硬件实现，采用组合逻辑控制器
寄存器数量	通用寄存器较少	大量的通用寄存器

在实际使用中，人们发现：典型程序中 80%的语句仅使用到指令系统中 20%的指令，而且使用频度较高的指令都是简单的基本指令。复杂指令的设计更为复杂，使得其执行速度受限。精简指令集计算机（RISC），其设计特点是简化指令集，只设置使用频度高的一些简单指令，复杂指令功能由多条简单指令组合来实现。

在嵌入式系统中，通过访问内存得到指令或者数据的时间远大于 CPU 执行指令所花费的时间。因此在 CPU 内部都有一些用来保存关键变量和临时数据的寄存器。一般而言，CPU 主要由如下部件构成。

1）通用寄存器组

以 Intel 8086 CPU 为例，其内部包含 8 个通用寄存器 AX、BX、CX、DX、BP、SP、SI、DI。它们均为 16 位寄存器，功能上略有不同：AX、BX、CX、DX 为通用数据寄存器；BX、BP 为基址寄存器；SI、DI 用于寄存器间接、变址寻址功能；SP 为堆栈指针，用于堆栈操作。

2）运算器

核心部件为算术逻辑单元（Arithmetic&logical Unit，ALU），可完成各种算术和逻辑运算，同时配合 ALU 工作的有暂存器。

3）控制器

构成 CPU 的另一重要部件，主要有以下几种：

（1）程序计数器（Program Counter，PC），存放着下一条需要执行指令的内存地址。

（2）程序状态字（Program Status Word，PSW），存放着指令执行结果状态及一些特定标志，例如溢出标志 OF、进位标志 CF 等。

（3）指令寄存器（Instruction Register，IR），存放当前执行指令。

（4）时序部件，用于产生所需时序信号。

随着计算机的发展，现有 CPU 内部还集成了高速缓存（Cache）、流水线等部件。

2.2.1　嵌入式微处理器的结构和类型

1．8 位、16 位、32 位处理器的体系结构特点

1）常用 8 位处理器的体系结构特点

8 位微处理器是指使用 8 位数据总线的微处理器。大部分的 8 位微处理器有 16 位的地址总线，其能够访问 64KB 的地址空间，而 8 位的数据总线则可以通过多重内存存取的方式来处理更多的数据。最早的 8 位微处理器是 1973 年由 Intel 公司开发的 8080 微处理器芯片，随后各大厂商也陆续推出 8 位微处理器，如 Zilog 公司的 Z80、Motorola 公司的 6800、National 半导体公司的 NSC800 及 Intel 公司的 8085 等。

由于 8 位微处理器具有低成本、可扩充内存及接口设备等特点，目前仍然在嵌入式系统领域得到广泛应用。8 位的微处理器有许多种不同时代的产品，其中有两个比较著名，一个是 Intel 公司推出的 8048，另一个则是 Fairchild 及 Mostek 公司推出的 3870。Intel 公司的 8048 在当时是一种新的体系结构，并未延续其他已存在的微处理器体系结构，因此在指令集及体系结构的开发上变的有些困难，但因为它是定位在具可伸缩性并且低成本的产品控制单元，所以至今仍被广泛地使用。另外，其所衍生的第二代产品 8051，更是目前应用最广泛的 8 位微处理器系列。Intel 的 8041 及 8042 是延续 8048 的系统，并作为从处理器（Slave Processor）使用。8044 是 8051 的延续微处理器，它包含了一个额外的链表接口，可以连到主微处理器，做其他的数据处理。

2）常用 16 位处理器的体系结构特点

继 8 位的微处理器后，许多厂商为了满足更复杂的应用，推出了 16 位微处理器。16 位微处理器是指内部总线宽度为 16 位的微处理器。16 位微处理器的操作速度及数据吞吐能力在性能上比 8 位微处理器有较大的提高，它的数据宽度增加了一倍，实时处理能力更强，主频更高，集成度、RAM 和 ROM 都有较大的增加，而且有更多的中断源，同时配置了多路的 A/D 转换通道和高速处理单元，适用于更复杂的控制系统。

Intel 公司的 8086 是第一款 16 位微处理器，当时 IBM 公司推出的个人计算机都是采用 8086 作为个人计算机的数据处理及控制核心。8086 微处理器延续了 Intel 公司之前的 8080 及 8085 微处理器的基本体系结构，再加上一些增强式的硬件体系结构与指令集。Intel 公司随后又在 1982 年 2 月，推出了第二代的 8086 产品 80286 微处理器，集成了以往许多微处理器需额外加上的外围设备组件，包括：一个时钟产生器、两个直接内存访问信道、一个中断信号控制器、3 个可程序化计时单元、可程序化芯片选择逻辑单元以及一个等待状态产生器；并且和 8086 及 8088 微处理器的软件兼容，因而受到市场的欢迎。

目前 16 位微控制器以 Intel 公司的 MCS-96/196 系列、TI 公司的 MSP430 系列和 Motorola 公司的 68H12 系列为主，它们主要应用于便携式设备、工业控制及智能仪器仪表等。

3）常用 32 位处理器的体系结构特点

32 位处理器采用 32 位的地址和数据总线，其地址空间达到了 4GB。目前主流的 32 位嵌入式微处理器系列主要有 ARM 系列、MIPS 系列、PowerPC 系列等。属于这些系列的嵌入式微处理器产品很多，有千种以上。

（1）ARM。作为一种 RISC 体系结构的微处理器，ARM 处理器具有 RISC 体系结构的典型特征，同时具有以下特点：

- 在每条数据处理指令当中，都控制算术逻辑单元 ALU 和移位器，以使 ALU 和移位器获得最大的利用率。
- 自动递增和自动寻址模式，以优化程序中的循环。
- 同时执行 Load 和 Store 多条指令，以增加数据吞吐量。
- 所有指令都可以条件执行，以执行吞吐量。

这些是对基本 RISC 体系结构的增强，使得 ARM 处理器可以在高性能、小代码尺寸、低功耗和小芯片面积之间获得好的平衡。

ARM 的数据类型：

- 字（Word）：在 ARM 体系结构中，字的长度为 32 位，而在 8 位/16 位处理器体系结构中，字的长度一般为 16 位。
- 半字（Half-Word）：在 ARM 体系结构中，半字的长度为 16 位，与 8 位/16 位处理器体系结构中字的长度一致。
- 字节（Byte）：在 ARM 体系结构和 8 位/16 位处理器体系结构中，字节的长度均为 8 位。

ARM 微处理器支持 7 种运行模式：

- 用户模式（USR）：ARM 处理器正常的程序执行状态。
- 快速中断模式（FIQ）：用于高速数据传输或通道处理。
- 外部中断模式（IRQ）：用于通用的中断处理。
- 管理模式（SVC）：操作系统使用的保护模式。
- 数据访问终止模式（ABT）：当数据或指令预取终止时进入该模式，可用于虚拟存储及存储保护。
- 系统模式（SYS）：运行具有特权的操作系统任务。
- 定义指令中止模式（UND）：当未定义的指令执行时进入该模式，可用于支持硬件协处理器的软件仿真。

（2）MIPS。MIPS 32 架构刷新了 32 位嵌入式处理器的性能标准。它是 MIPS 科技公司下一代高性能 MIPS-Based 处理器 SoC 发展蓝图的基础，并向上兼容 MIPS 64 位架构。MIPS 架构拥有强大的指令集、从 32 位到 64 位的可扩展性、广泛的软件开发工具以及众多 MIPS 科技公司授权厂商的支持，是领先的嵌入式架构。

MIPS 32 架构是以前的 MIPS I 和 MIPS II 指令集架构的扩展集，整合了专门用于嵌