

# 第 1 章 嵌入式系统概述

## 1.1 嵌入式系统的定义

嵌入式系统通常指非计算机系统，是一种具有计算机功能但又不称之为计算机的设备或器材。嵌入式系统以应用为中心，是一种软硬件可裁减的，适应应用系统对功能、可靠性、成本、体积、功耗等综合性能严格要求的专用计算机系统。简单地说，嵌入式系统集系统的硬件与应用软件于一体，具有软件代码少、自动化程度高、响应速度快等特点，特别适合要求实时和多任务的体系。

嵌入式系统主要由嵌入式处理器、相关支撑硬件、嵌入式操作系统及应用软件系统等组成，它是可独立工作的“器件”。

## 1.2 嵌入式系统的应用

嵌入式系统广泛应用于生活中的各种电子电器设备，如 PDA（个人数字助理）、移动计算设备、电视机顶盒、手机、数字电视、微波炉、数字相机、家庭自动化系统、电梯、空调、安全系统、自动售货机、蜂窝式电话、消费电子设备、汽车、工业自动化仪表与医疗仪器等。

因此，我们又可以将嵌入式系统定义为“嵌入到特定对象体系中的专用计算机系统”。

## 1.3 嵌入式系统的特点

按照嵌入式系统的定义，嵌入式系统有 3 个基本特点，即“嵌入性”“计算机”与“专用性”。

“嵌入性”：“嵌入性”由早期微型机时代的嵌入式计算机应用而来，专指将计算机嵌入到对象体系中，实现对对象体系的智能控制。当嵌入式系统变成一个独立应用产品时，可将嵌入性理解为内部嵌有微处理器或计算机。

“计算机”：“计算机”是对对象系统进行智能化控制的根本保证。随着单片机向 MCU、SoC 发展，片内计算机外围电路、接口电路、控制单元日益增多，“专用计算机系统”逐渐演变成为“内含微处理器”的现代电子系统。与传统的电子系统相比较，现代电子系统由于内含微处理器，具备对对象系统的计算机智能化控制能力。

“专用性”：“专用性”是指在满足对象控制要求和环境要求条件下的软硬件裁减性。嵌入式系统的软硬件配置必须依据嵌入对象的要求，设计成专用的嵌入式应用系统。

## 1.4 嵌入式微处理器

### 1.4.1 嵌入式处理器简介

嵌入式处理器种类繁多，有 ARM(Advanced RISC Machines)、MIPS(Microprocessor without Interlocked Piped Stages)、PPC(Power PC)等多种架构。到目前为止，ARM 芯片在微处理器市场的份额已达到 75%。

ARM，既可以认为是一个公司的名字，也可以认为是对一类微处理器的通称，还可以认为是一种技术的名字。ARM 公司是 32 位嵌入式 RISC 微处理器技术的领导者，自从 1990 年创办以来，公司基于 ARM 技术 IP 核的微处理器的销售量已经超过 100 亿片。

ARM 公司并不生产芯片，而是出售芯片技术授权。其合作公司针对不同需求搭配各类硬件部件（如 UART、SDI、PC 等），设计不同的 SoC 芯片。

ARM 公司在技术上的开放性使得它的合作伙伴既有世界顶级的半导体公司，也有各类中、小型公司。随着合作伙伴的增多，ARM 处理器得到了更多的第三方工具和软件支持，又使整个系统成本降低，新品上市时间加快，从而具备更大的竞争优势。

基于 ARM 的处理器以其高速度、低功耗、低成本等优点得到非常广泛的应用，它可以应用于以下领域：

（1）工业控制领域：作为 32 位的 RISC 架构，基于 ARM 核的微控制器芯片不仅占据了高端微控制器市场的大部分市场份额，同时也向低端微控制器应用领域扩展。ARM 微控制器利用其低功耗、高性价比等优点，向传统的 8 位/16 位微控制器发起了挑战。

（2）无线通信领域：目前已有超过 85% 的无线通信设备采用了 ARM 技术，ARM 以其高性能和低成本等优点，在该领域的地位日益巩固。

（3）网络应用：随着宽带技术的普及，采用 ARM 技术的 ADSL 芯片正逐步获得竞争优势。此外，ARM 在语音及视频处理上进行了优化，并获得广泛支持，也对 DSP 的应用领域发起了挑战。

（4）消费类电子产品：ARM 技术在目前流行的数字音频播放器、数字机顶盒和游戏机中得到广泛采用。

（5）成像和安全产品：现在流行的数码相机和打印机中绝大部分采用了 ARM 技术，手机中的 32 位 SIM 智能卡也采用了 ARM 技术。

除此以外，ARM 微处理器及技术还被应用到许多其他的领域，并将在未来取得更广泛的应用。

### 1.4.2 嵌入式处理器的硬件架构

#### 1. 冯·诺依曼结构

1945 年，冯·诺依曼首先提出了“存储程序”的概念。后来，人们把利用这种概念

设计的电子计算机统称为“冯·诺依曼结构”计算机。“冯·诺依曼结构”的处理器使用同一个存储器，并由同一个总线进行传输。

“冯·诺依曼结构”处理器具有以下几个特点：

- 必须有一个存储器；
- 必须有一个控制器；
- 必须有一个运算器，用于完成算术运算和逻辑运算；
- 必须有输入和输出设备，用于进行人机通信。

冯·诺依曼的主要贡献就是提出并实现了“存储程序”的概念。由于指令和数据都是二进制码，指令和操作数的地址又密切相关，因此当初选择这种结构是自然而然的。但是，这种指令和数据共享同一总线的结构，使得信息流的传输成为限制计算机性能的瓶颈，影响了数据处理速度的提高。

在典型情况下，完成一条指令需要 3 个步骤：取指令、指令译码和执行指令。如图 1.1 所示，这是一个最简单的对存储器进行读写操作的指令，指令 1 至指令 3 均为存、取数指令。对于“冯·诺依曼结构”处理器，由于取指令和存取数据要从同一个存储空间进行存取，经由同一总线进行传输，因此它们无法重叠执行，只有一个完成后再进行下一个。

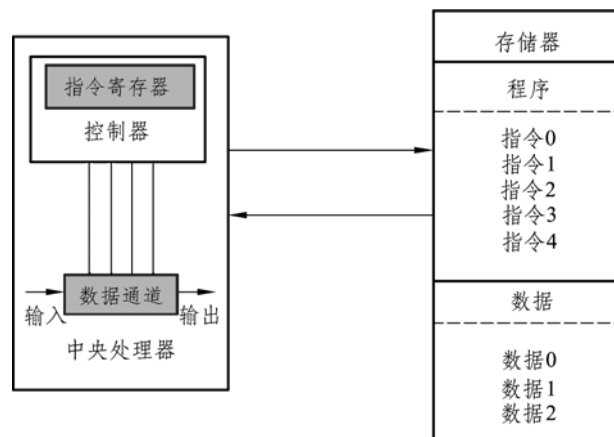


图 1.1 冯·诺依曼体系结构模型

ARM7 系列微处理器采用“冯·诺依曼结构”。

## 2. 哈佛结构

哈佛结构是一种将程序指令存储和数据存储分开的存储器结构。哈佛结构是一种并行体系结构，它的主要特点是将程序和数据存储在不同的存储空间（即程序存储器和数据存储器是两个独立的存储器）中，每个存储器独立编址、独立访问。与两个存储器相对应的是系统的 4 条总线：程序的数据总线与地址总线、数据的数据总线与地址总线。哈佛结构的微处理器通常具有较高的执行效率，其程序指令和数据指令分开组织和存储，执行时可以预先读取下一条指令。

如图 1.2 所示，如果采用哈佛结构处理以上同样的 3 条存取数指令，由于取指令和存

取数据分别经由不同的存储空间和不同的总线，使得各条指令可以重叠执行，从而有效克服了数据流传输的瓶颈，提高了运算速度。

ARM9、ARM10 和 ARM11 使用哈佛结构。

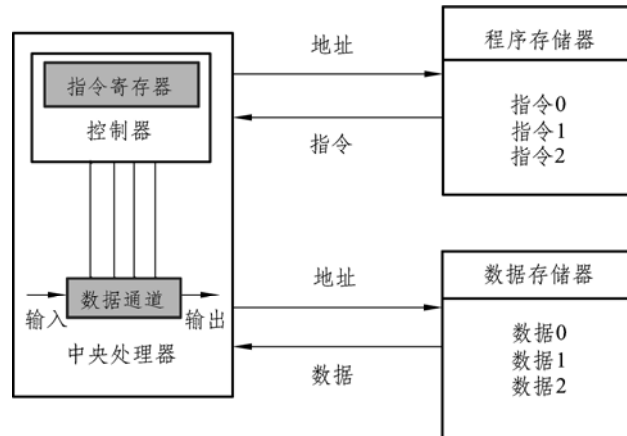


图 1.2 哈佛体系结构模型

### 1.4.3 RISC 体系结构

传统的 CISC (Complex Instruction Set Computer, 复杂指令集计算机) 结构有其固有的缺点，即随着计算机技术的发展而不断引入新的复杂的指令集，为支持这些新增的指令，计算机的体系结构会越来越复杂。然而，在 CISC 指令集的各种指令中，其使用频率却相差悬殊；大约有 20% 的指令会被反复使用，占整个程序代码的 80%；而余下的 80% 的指令却不经常使用，在程序设计中只占 20%。

RISC (Reduced Instruction Set Computer, 精简指令集计算机) 并非只是简单地减少指令，而是把着眼点放在如何更加简单合理地提高运算速度上。为达到上述目的，RISC 结构采取了以下措施：优先选取使用频最高的简单指令，避免复杂指令；将指令长度固定，指令格式和寻址方式种类减少；以控制逻辑为主，不用或少用微码控制等。

RISC 体系结构具有如下特点：

- 采用固定长度的指令格式，指令格式整齐、简单，基本寻址方式只有 2~3 种。
- 使用单周期指令，便于流水线操作。
- 大量使用寄存器，数据处理指令只对寄存器进行操作，只有“加载/存储”指令可以访问存储器，提高了指令的执行效率。
- 所有的指令都可根据前面的执行结果决定是否被执行，提高了指令的执行效率。
- 可用“加载/存储”指令批量传输数据，以提高数据的传输效率。
- 可在一条数据处理指令中同时完成逻辑处理和移位处理。
- 在循环处理中使用地址的自动增减来提高运行效率。

### 1.4.4 ARM 微处理器的寄存器和指令

#### 1. ARM 微处理器的寄存器结构

ARM 处理器共有 37 个寄存器，这些寄存器包括：

- 31 个通用寄存器（包括程序计数器（PC 指针）），均为 32 位的寄存器。
- 6 个状态寄存器，用以标识 CPU 的工作状态和程序的运行状态，均为 32 位的寄存器，目前只使用了其中的一部分。

ARM 处理器有 7 种不同的处理器模式，在每一种处理器模式下均有一组相应的寄存器与之对应。即在任意一种处理器模式下，可访问的寄存器包括 15 个通用寄存器（R0 ~ R14）、1~2 个状态寄存器和程序计数器。在所有的寄存器中，有一部分可以在 7 种处理器模式下共用，而另一部分在不同的处理器模式下不可共用。

## 2. ARM 微处理器的存储格式

ARM 体系结构将存储器看作是从 0 地址开始的字节的线性组合。0~3 字节放置第 1 个存储的字数据，第 4~7 字节放置第 2 个存储的字数据，依次排列。作为 32 位的微处理器，ARM 体系结构所支持的最大寻址空间为 4 GB（ $2^{32}$  字节）。

ARM 体系结构可以用两种方法存储字数据，分别称为小端格式（Little Endian）和大端格式（Big Endian）。

### 1) 小端格式

每个 32 位的字是由 4 个字节组成。对这 32 位的数据进行存储时，如果将低字节的内容保存在低地址空间里，高字节的内容保存在高地址空间里，这样的存储方式称为小端格式。小端格式的存储格式如图 1.3 所示。

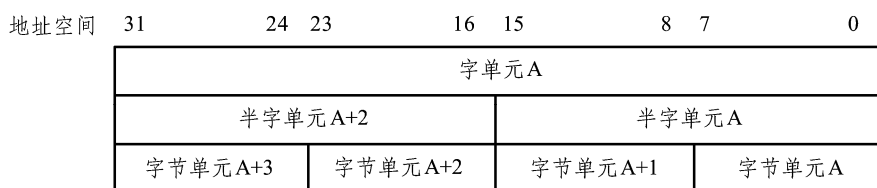


图 1.3 小端格式

### 2) 大端格式

大端格式和小端格式的存储方式刚好相反。在大端格式中，字数据的高字节存储在低地址空间中，而字数据的低字节则存储在高地址空间中。大端格式的存储格式如图 1.4 所示。



图 1.4 大端格式

ARM 系列微处理器默认采用小端格式。

## 3. ARM 微处理器的指令结构

ARM 微处理器在较新的体系结构中支持两种指令集 :ARM 指令集和 Thumb 指令集。其中,ARM 指令的长度为 32 位,Thumb 指令的长度为 16 位。Thumb 指令集为 ARM 指令集的功能子集,与等价的 ARM 代码相比较,可节省 30%~40%以上的存储空间,同时具备 32 位代码的所有优点。

#### 4. ARM 微处理器的运行模式

ARM 微处理器支持 7 种运行模式,分别为:

- (1) 用户模式 (usr): ARM 处理器正常的程序执行状态。
- (2) 快速中断模式 (fiq): 用于高速数据传输或通道处理。
- (3) 外部中断模式 (irq): 用于通用的中断处理。
- (4) 管理模式 (svc): 操作系统使用的保护模式。
- (5) 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,可用于虚拟存储和存储保护。
- (6) 系统模式 (sys): 运行具有特权的操作系统任务。
- (7) 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。

### 1.5 嵌入式系统的主要开发软件

#### 1.5.1 Keil $\mu$ VISION

Keil 公司开发的 ARM 开发工具 MDK (Microcontroller Development Kit),用于开发基于 ARM 核系列微控制器的嵌入式应用程序,如图 1.5 所示。它适合不同层次的开发者使用,包括专业的应用程序开发工程师,也包括嵌入式软件开发的入门者。MDK 内含工业标准的 Keil C 编译器、宏汇编器、调试器、实时内核等组件,支持所有基于 ARM 的设备。



图 1.5 Keil  $\mu$ Vision 启动界面

#### 1.5.2 IAR

IAR 的 Embedded Workbench 是一种增强型、一体化、嵌入式、集成开发环境,它集成了开发嵌入式系统所需要的文件编辑、项目管理、编译、链接和调试工具。IAR 的

Embedded Workbench 系列适用于开发基于 8 位、16 位以及 32 位微处理器的嵌入式系统，针对多种不同的目标处理器，在相同的集成开发环境中进行基于不同 CPU 的嵌入式系统应用程序开发。

### 1.5.3 ADS

ADS ( ARM Developer Suite )，是由 Metrowerks 公司在 1993 年开发的 ARM 处理器下最主要的开发工具，成熟版本为 ADS1.2。ADS 是全套的实时开发软件工具，包编译器生成的代码密度和执行速度优异。ADS1.2 支持 ARM10 之前的所有 ARM 系列微控制器。

#### 【习 题】

1. 嵌入式处理器的硬件架构有哪几类？
2. ARM 微处理器的运行模式有哪几种？



扫码看课后习题答案+  
名师教学 PPT 课件

