

嵌入式系统简介

本章学习目标

本章将学习下面的内容：

- “系统”以及“嵌入式系统”的定义
- 比较嵌入式系统和一般用途的计算系统
- 系统中嵌入的处理器
- 嵌入式系统的处理单元包括：
 - 微处理器
 - 微控制器(嵌入式微控制器和 ATME1、ARM、RISC 或 CISC 微控制器)和数字信号处理器(DSP)
 - 单用途处理器，例如协处理器加速器、图形处理器和直接内存访问控制器
- 系统中的嵌入式硬件单元，包括构建块(VLSI 内核或电路板上的块)
- 该单元包括电源、时钟振荡器、复位、watchdog 定时器复位、系统定时器、实时时钟、系统内部和外部存储器、输入输出端口、设备、数字-模拟转换器、模拟-数字转换器、LED、LCD 或触摸屏显示单元、多路转换器、多路信号分离器和中断处理器
- 软件最终嵌入为 ROM 或闪存的字节，称为 ROM 映像
- 用于开发嵌入式软件的语言，例如：
 - 机器语言和汇编语言
 - 高级语言、Linux 或实时操作系统
- 介绍嵌入式系统设计
- 介绍嵌入式系统体系结构
- 介绍嵌入式系统模型
- 嵌入式系统的 3 种类型
- 嵌入式系统设计人员需要掌握的技能
- 嵌入式系统应用示例

1.1 嵌入式系统

1.1.1 系统

定义

系统是一种根据固定的计划、程序或规则集处理、组织或者执行一项或多项任务的方式。系统的另一个定义是：也是一种安排方法，其所有单元按照一定的计划或程序装配在一起，并共同工作。

示例 1-1

手表是一个时间显示系统。其零件包括硬件、表针和电池，以及漂亮的表盘、底盘和表链。这些零件以特定方式组织在手表系统中，显示系统会显示每一秒的实时时间，并且每一秒都连续更新时间显示。它遵守一系列规则。其中一条规则是：所有的表针都顺时针转动。时间显示系统还有其他规则。

1.1.2 嵌入式系统

定义

嵌入式系统的一种定义如下：

嵌入式系统是一种在计算机硬件中嵌入软件的系统，该系统专用于某个应用或者应用或产品的特定部分，或大型系统的一个组件。

有许多书籍都以不同方式定义了嵌入式系统。下面给出其他文献对嵌入式系统的定义：

Computers as Components – Principles of Embedded Computing System Design 一书的作者 Wayne Wolf 认为：“什么是嵌入式计算系统？如果不严格地定义，它是任何一台包含可编程计算机的设备，它本身却不是一台通用计算机”，“使用微处理器构造的传真机或时钟就是一种嵌入式计算系统”。

Embedded Microcontrollers 一书的作者 Todd D. Morton 认为：“嵌入式系统是一种电子系统，它包含微处理器或微控制器，但是我们不认为它们是计算机——计算机隐藏或嵌入在系统中。”

Embedded Software Primer 一书的作者 David E. Simon 认为：“人们使用嵌入式系统这个术语，指的是隐藏在任意产品中的计算机系统。”

An Introduction to the Design of Small Scale Embedded System with examples from PIC, 80C51 and 68HC05/08 Microcontrollers 一书的作者 Tim Wilmshurst 认为：(1) “嵌入式系统是这样一个系统，它的首要功能并不是计算，而是受嵌入其中的计算机控制的一个系统。计算机可能是微处理器或微控制器。“嵌入”暗示它存在于整个系统中，从外部观察不到，是形成更大整体的一个不可分割的部分。”(2) “嵌入式系统是一种基于微控制器、软件驱动的可靠实时控制系统，以自治、人工或网络方式进行交互，对各种物理变量进行操作，存在于各种环境中，在竞争激烈的市场上出售”。

1.1.3 嵌入式系统和通用计算系统

计算机是通用计算系统的一个示例。计算机是一个具有下列部分或更多组成部分的系统。

1. 计算机(通用计算系统)的组件

- (1) 微处理器。
- (2) 一组特殊用途的处理器，例如图像、输入输出和网络处理器。
- (3) 大型存储器。
- (4) I/O 单元，例如触摸屏、调制解调器、附有调制解调器的传真机等。
- (5) 输入单元，例如键盘、鼠标、数字转换器、扫描仪等。
- (6) 输出单元，例如触摸屏、LCD 显示屏、视频监视器、打印机等。
- (7) 网络单元，例如以太网卡、基于前端处理器的驱动程序、总线驱动器等。
- (8) 操作系统(OS)。

存储器包括如下部分：

- (1) 处理器中的缓存。
- (2) 主存储器(半导体存储器：RAM、ROM)。
- (3) 辅助存储器(硬盘、磁盘和磁带中的磁性存储器，CD-ROM 中的光存储器，便携式计算机中的记忆棒)，辅助存储器中的通用用户软件和应用软件。

OS 允许通过程序使用硬件资源。OS 支持同时执行多个程序。OS 调度和运行任务(如系统输入输出)，使计算性能最大化。

通用计算系统的特点

通用计算系统有如下特点：

- (1) 程序先加载到 RAM 中，再运行通用程序、多用户程序和多个任务。
- (2) 需要操作系统(OS)。
- (3) 快速运行复杂的算法。
- (4) 提供一组复杂的图形或触摸屏用户界面(GUI)。

通用计算系统的限制

这种系统在执行程序任务时，对存储器没有限制，也没有最后期限的限制。

2. 嵌入式系统的组件

嵌入式系统有三个主要组成部分：

- (1) 与计算机相似的硬件。处理器可以嵌入到处理器核心中，硬件包括嵌入式存储器、外围设备和输入输出设备。它嵌入了主应用软件。应用软件可以并行执行多个任务(一组任务)。图 1-1 给出了嵌入式系统硬件中的组成单元。

(2) 软件通常嵌入在 ROM(只读存储器)、闪存或媒介卡中。与计算机不同，系统通常不需要辅助硬盘和 CD 存储器。

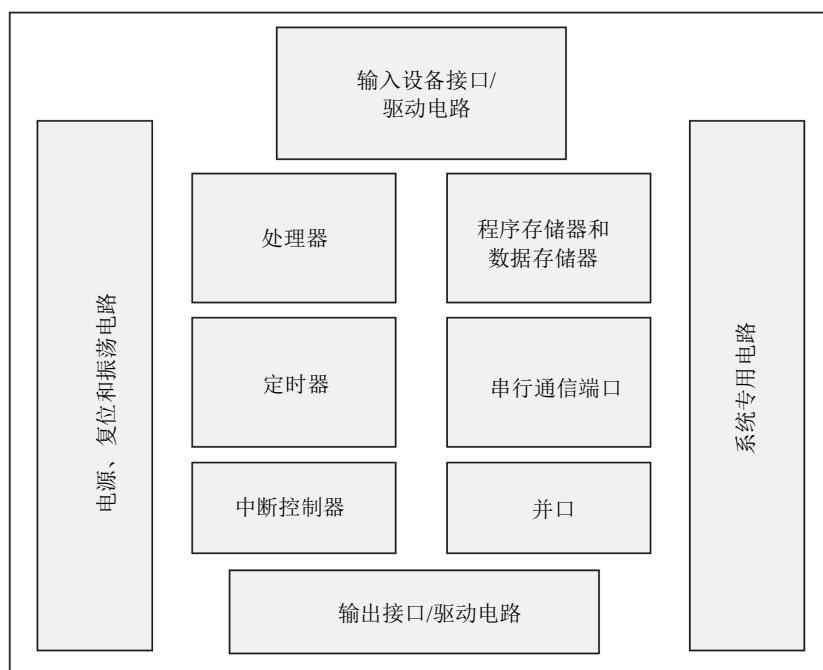


图 1-1 嵌入式系统硬件的组成

(3) 它嵌入了实时操作系统(RTOS)，RTOS 管理应用软件，控制对系统资源的访问。它允许在指定的时间间隔完成程序任务的执行。

嵌入式系统的特点

嵌入式系统具有以下特点：

- (1) 程序预先加载或嵌入到 ROM(只读存储器)或闪存中。
- (2) 系统实时运行。任务根据优先级来运行。系统响应事件，在预定的时间间隔内中断，根据优先级调度响应。系统控制任务、事件的延迟，也管理中断和响应。延迟指要求响应的时间点和实际开始执行的时间点之间的间隔。
- (3) 一组专用的函数。
- (4) 专用的复杂算法。
- (5) 专用的复杂预编程硬件、图形和其他用户界面(GUI)。
- (6) 多重速率的操作具有预定的不同时间约束，完成不同的操作，例如音频、视频、数据、网络流和事件(例如屏幕触摸、开关 ON、外部输入)。

嵌入式系统的约束

嵌入式系统的软件设计受三个条件的限制：

- (1) 可用的系统存储器。
- (2) 可用的处理器速度。
- (3) 需要系统连续运行，因此，系统需要在运行时限制能源的消耗。程序运行时，要把能源的消耗降到最低。

嵌入式系统的设计必须考虑性能、功率、尺寸、设计和生产成本等限制条件。

1.2 嵌入系统中的处理器

处理器是任何计算系统中重要的组成单元，是嵌入式系统的核心。

指令在处理器的指令集中定义，在程序中按顺序执行。程序按照指令从存储器中提取的顺序执行它们。处理器具有两个基本单元：程序流控制单元(CU)和执行单元(EU)。处理器运行一组指令的“提取-执行”循环。

处理器采用如下形式：

- (1) 单VLSI(Very Large Scale Integrated, 超大规模集成电路)芯片
- (2) ASIP(Application Specific Instruction-set Processor, 专用指令集处理器)中的核
- (3) ASIC (Application Specific Integrated Circuit, 专用集成电路)中的核
- (4) SoC(System-on-Chip, 片上系统)中的核

核是 VLSI 或 VVLSI 芯片上功能电路的一部分。片上系统表示一个芯片上带有所有核、软件、系统的数字和模拟电路的 VLSI 或 VVLSI 芯片。

嵌入式系统中的处理器芯片或核可以是下列之一：

(1) **微处理器或嵌入式处理器**：一种通用处理器(General Purpose Processor, GPP)。通用处理器的指令集不是专为特定应用设计的，例如 Intel 奔腾处理器。

(2) **专用指令集处理器(ASIP)**。ASIP 是一个处理器带有为特定应用设计的指令集的 VLSI 芯片或核。

- a. 微控制器
- b. 数字信号处理器(Digital Signal Processor, DSP)
- c. 输入输出(IO)、媒介、网络或特定域的处理器
- (3) **作为附加处理器的专用处理器**。专用处理器的示例如下：
 - a. 协处理器(Coprocessor)，如用于图形处理、浮点处理、加密、解密、离散余弦变换和逆变换、TCP/IP 协议栈和网络连接功能的协处理器。
 - b. 加速器(Accelerator)，如 Java 代码加速器。
 - c. 控制器，如外围设备控制器、DMA 控制器、总线控制器。
 - d. 集成到专用集成电路(ASIC)或超大规模集成电路(VLSI)中的 GPP 核或 ASIP 核，或者 VLSI(ASIC)芯片中集成了处理器单元的 FPGA 核。
 - e. 专用系统处理器(ASSP)。
 - f. 使用 GPP 的多核处理器和多处理器系统，例如 Intel i3 处理器。

1.2.1 微处理器

微处理器是一个 VLSI 芯片，芯片中有一个 CPU，还可以有其他附加的单元(例如缓存、浮点处理算术单元、流水线和超标量单元)，这样可以加快指令的处理速度。

CPU 是一个集中取指令和使用处理器处理一组通用指令的单元。微处理器的“提取-执行”周期由时钟限制。在早期处理器中，该时钟的频率是 4MHz。新一代处理器的时钟频率已经达到 800MHz 到 3.2GHz，处理器还具有多个核和线程来处理指令。

注意:

当软件位于外部存储芯片或辅助存储器上时, 微处理器就用作通用处理器。

由于具备下列特点, 可给嵌入式设计使用微处理器或通用处理器芯片或核: (1)用预定义的通用指令集中已知可用的指令进行处理, 可加快系统开发速度。(2)如果印制板和 I/O 接口是根据 GPP 设计的, 可以通过只修改印制板上 ROM 或闪存中的嵌入式软件, 应用于一个新系统。(3)现有的编译器或框架便于用高级语言开发嵌入式软件。(4) 现有的 OS 和处理器专用的 API, 以及先前为其他应用设计的代码, 能够加快一个新系统的开发速度。设计成本也会降低。

示例 1-2

1. Intel 80x86(也称为 x86)处理器是 8086 的 32 位后继处理器(这里, x 的意思是将 8086 扩展为 32 位)。80x86 系列的例子是 Intel 80386、80486 和 32 位微处理器经典奔腾(Pentium)系列。IBM PC 使用 80x86 系列处理器, 嵌入系统的 PC 把这些集成进来, 以完成特定任务, PC 中嵌入式系统的示例有图形加速器、磁盘控制器、总线控制器和网络接口。
2. 苹果、安卓和黑莓手机使用 1.5GHz 双核处理器。
3. Intel i3 4010U 是双核四线程处理器, 它有 3MB 缓存, 小功率 DDR 存储器, 操作频率是 1.70GHz, 包含 Intel Graphics 4400。Intel i7 4707R 是四核八线程处理器, 它有 6MB 缓存, 小功率 DDR 存储器, 操作频率是 3.20GHz, 包含 Intel Iris Pro Graphics 5200。
4. 许多移动手持设备都使用 ARM 9、ARM 10、ARM 11 处理器。
5. 许多服务器使用 SPARC 系列的微处理器。

1.2.2 微控制器

微控制器是 VLSI 或 SoC 中的一个集成芯片或核, 把处理器、存储器和其他一些硬件单元连接在一起, 因此它的功能类似于微计算机。图 1-2 给出了微控制器中的功能电路(实线框中的部分)。图中还给出了应用专用的单元(虚线框中的部分)。这些单元出现在给定微控制器系列的特定版本中。

正如微处理器或处理器是计算系统的基本组成部分一样, 微控制器是控制电路或通信电路的基本组成部分。微控制器是一个单芯片 VLSI 单元(也称为“微计算机”), 这个单元虽然计算能力有限, 却改进了输入输出功能, 还具备一些片上功能单元(关于各种功能硬件单元的介绍参见 1.3 节)。

微控制器尤其适用于实时控制应用的嵌入式系统。该系统具有处理器、片上程序存储器和设备。

图 1-3 给出了在小型、中型和大型嵌入式系统中普遍使用的微控制器。系统设计者在选择微控制器作为处理单元之前, 必须考虑特定应用的具体特征。

几个最新的微控制器也有超标量处理单元和双核, 因此具有更强大的计算能力。

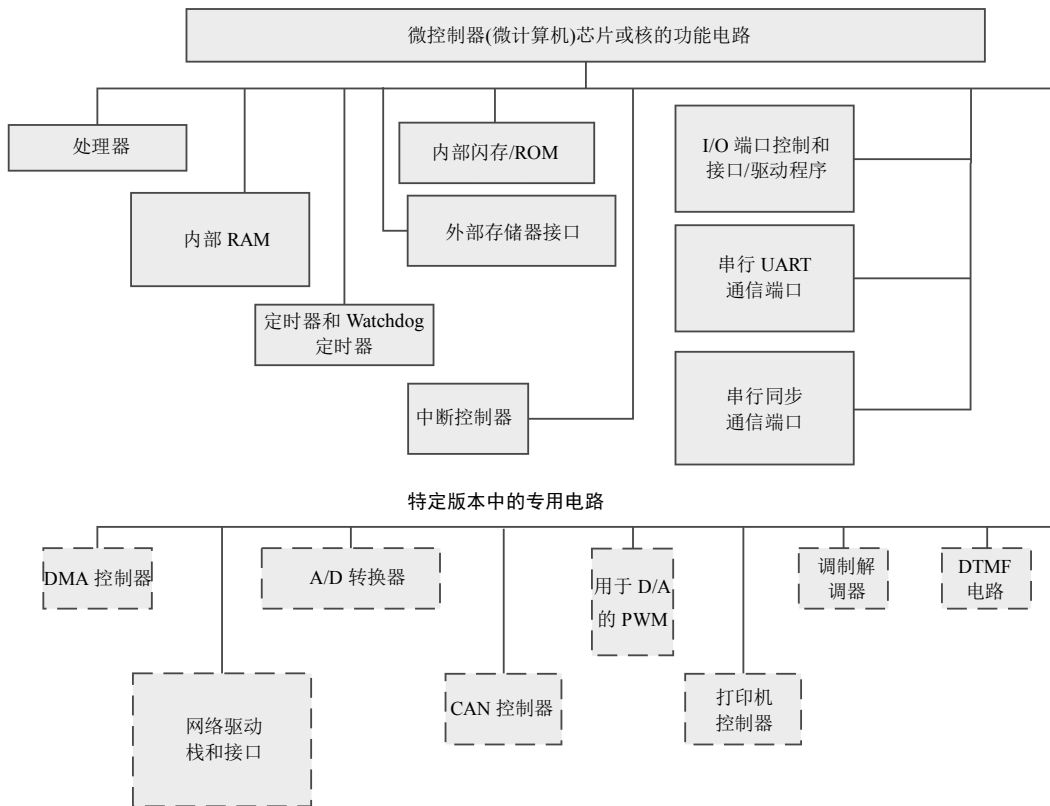


图 1-2 嵌入式系统中的微控制器芯片和核中的各种功能电路(实线框中的部分), 以及微控制器系统的特定版本的应用专用单元(虚线框中的部分)

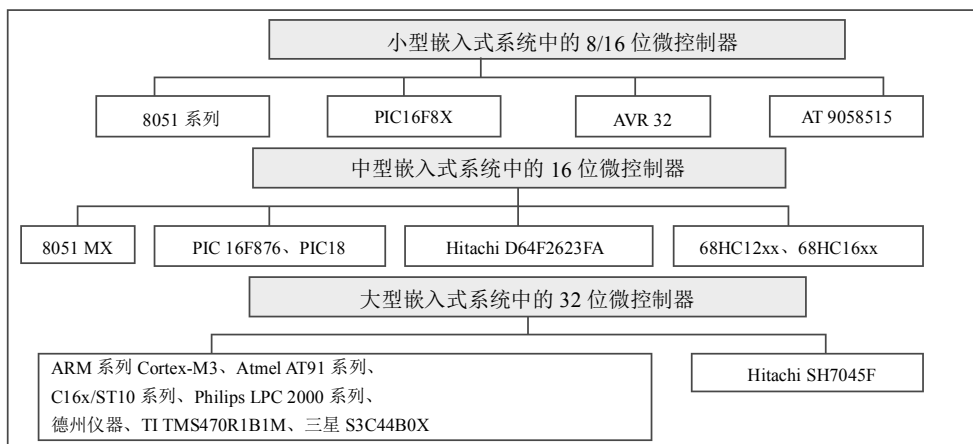


图 1-3 在小型、中型和大型嵌入式系统中普遍使用的微控制器

注意：

当小型嵌入式软件或嵌入式软件的一部分位于芯片的内部存储器时，以及当需要一些片上功能单元(例如中断处理程序、端口、定时器、ADC、PWM、CAN 控制器、ZigBee 或 USB 接口)时，要使用微控制器。

用在嵌入式系统中的一些重要微控制器芯片有 8051 和 80251 版本、ATMEL mega 系列 8/168/324/1284/2560、Freescale 68HS12 和更高的微芯片 PIC16F84 或 16C76、16F876、18 和 24, ARM、Intel、Nexperia、三星和 ST 微电子的 ARM9、ARM10、ARM11 处理器的微控制器版本。

1. 嵌入式微控制器

嵌入式微控制器表示嵌入所有需要的存储器和设备中的微控制器, 因此, 不需要外部接口电路。嵌入式微控制器芯片或核之所以用于设计, 是因为它有如下特点: (1)微控制器是一个集成芯片, 具有处理器、存储器和其他几个硬件单元。(2)集成开发环境(IDE)有助于快速开发系统。(3)经过良好测试和调试的处理器专用 API 允许重用先前为其他应用设计的代码, 能加快新系统的开发速度。设计成本也会减少。

2. ATMEL

ATMEL 制作了一些 8051 系列微控制器, 它具备闪存块。闪存的优点是: 在开发阶段, 软件可以重复存储, 提供了服务内编程的能力。该设计可以通过模拟和调试进行完善, 直到完成最终的系统。

ATMEL 微控制器的示例是 AT89C51RD2/ED2。它是 80C51 CMOS 单芯片 8 位微控制器的高性能 CMOS 闪存版本。它包含 64KB 的闪存块, 用于存储代码和数据。闪存可以用两种模式编程: 并行和串行。在编程过程中, 会在内部生成编程电压。

1.2.3 ARM

ARM 是 GPP, 其示例有 ARM7 和 ARM9。这些处理器的核心可以嵌入 VLSI 芯片或 SoC(片上系统)。ARM 处理器 VLSI 体系结构可用作 CPU 芯片或集成到 VLSI 或 SoC 中。

ARM 通过核上的 RISC 体系结构提供了 CISC 功能。ARM7、ARM9 及其 DSP 改进版本的核可嵌入系统(参见 <http://www.ti.com/sc/docs/asic/modules/arm7.htm> 和 [arm9.htm](http://www.ti.com/sc/docs/asic/modules/arm9.htm))。

ARM 在新的 GPP 中集成了其他特性(如 DSP), 它们有几个来源, 例如 Intel 和 Texas。典型的 ARM9 应用有机顶盒、有线调制解调器、无线设备(如移动手机)。

ARM9 有单个循环 16×32 乘法累积单元。它的操作频率是 200MHz, 它使用 $0.15\mu\text{m}$ GS30 CMOS, 有一个 5 段管线。它与 RISC 协同使用。

ARM、Intel、Nexperia、三星和 ST 微电子制作了 ARM9、ARM10 和 ARM11 处理器的微控制器版本。

1.2.4 RISC

RISC 微处理器提供了指令的快速处理, 每个指令都在单个时钟循环中处理。而且, 除了前面提及的功能的巨大改进之外, 处理指令集中指令的速度也有很大提升。系统需要进行大量计算(例如语音处理系统)时, 就使用 RISC。新一代的微处理器具有 RISC 核和 CISC 功能, 示例是 ARM7/ARM9 和 ARM11。

对于基于 RISC 的嵌入式系统, 重要的微控制器 VLSI 核或芯片是 Philips、三星和 ST 微电子、ARM 和 Texas 改进的 ARM9/ARM7 微控制器。这些微控制器都具有 RISC 核和 CISC 功能。

1.2.5 CISC

CISC 微处理器有许多寻址模式和指令,它提供了内存地址中的一个字与寄存器中的字相乘的功能。它的硬件比较简单。执行单元包括一个控制存储器,该控制存储器为所处理的每个指令保存了微指令和纳指令。

复杂指令的循环次数要大于寄存器的移动或相加指令。系统需要用许多寻址模式和较简单的 VLSI 核来执行操作时,使用 CISC。示例是 80×86 微处理器。

嵌入式系统的重要微控制器 VLSI 核或芯片是 Intel 8051、80251、PIC 16F84 或 16C76 或 PIC 18 和 Hitachi H8x 系列。

1.2.6 SoC

通用应用的嵌入式系统在一个硅芯片上设计,用于所有需要的模拟和数字电路、处理器和软件。基于 SoC 的系统示例是机顶盒和移动电话。

片上系统(SoC)是嵌入式系统的一次设计创新。SoC 是 VLSI 芯片上的一个系统,它嵌入了所有需要的硬件和软件设计单元。典型的 SoC 嵌入了如下组件:

- (1) 嵌入式处理器 GPP 或 ASIP 核
- (2) 专用处理核或多个处理器
- (3) 网络总线协议核
- (4) 加密函数单元
- (5) 离散余弦变换,用于信号处理应用
- (6) 存储器
- (7) 多个标准的源解决方案,称为 IP(Intellectual Property)核
- (8) 可编程的逻辑设备和 FPGA(Field Programmable Gate Array, 现场可编程门阵列)核
- (9) 其他逻辑回路
- (10) 模拟单元

1.2.7 数字信号处理器(DSP)

数字信号处理器(DSP)是一个处理器核或芯片,由应用程序用于处理数字信号。DSP 作为 ASIP,是单芯片 VLSI 单元,它拥有微处理器的计算能力,有 MAC(Multiply and Accumulate, 乘法和累加)单元。现在,典型的 MAC 有 16×32 MAC 单元。

DSP 的示例有 TMS320Cxx 和 OMAP。设计 ASIC 解决方案时,ARM9 会与 DSP 集成。

一个示例是与 DSP 的集成,即 Texas 中的 TMS320C55x。(参见 <http://www.ti.com/sc/docs/asic/modules/arm7.htm> 和 [arm9.htm](http://www.ti.com/sc/docs/asic/modules/arm9.htm))。

1.2.8 专用处理器

专用处理器在嵌入式系统中使用,其示例如下:

示例 1-3

嵌入式系统中使用的专用处理器有下列几种:

1. **控制器**。用于总线或外围设备、DMA、触摸屏或显示控制器。
2. **协处理器**。用于浮点处理。
3. **图形处理器**。图像中包含了大量像素。图像格式是 1/4 通用中间格式(Quarter-CIF)。这种格式的图像有 144×176 (横坐标 x 轴和纵坐标 y 轴) 个像素。视频帧有 525×625 个像素, 视频图像适配器(VGA)格式有 $640 \times 480 = 307\,200$ 个像素, 这种格式用于电子邮件和网页。系统需要一个图形处理器, 它用于游戏应用程序, 它从图形存储缓存中显示图形, 控制着图像或图像一部分的移动(在屏幕上转移位置)和旋转。图形处理器的示例有 Adreno 200(用于黑莓手机和三星 Galaxy)和 iPhone5 中的 PowerVR SGX 543MP3(三核图形)处理器。
4. **像素协处理器**。高分辨率图片的格式有: 2592×1944 像素=5 038 848 像素、 2592×1728 像素=3.2M 像素、 2048×1536 像素=3M 像素、 1280×960 像素=1M 像素。在数码相机中, 需要像素协处理器, 该处理器会直接显示图像, 或者经过向右旋转、向左旋转、向上旋转、向下旋转, 在切换到前一幅或后一幅图像后显示。
5. **加密引擎**。这种处理器运行适当的算法, 为安全传输的数据流加密。
6. **解密引擎**。这种处理器运行适当的算法, 对接收终端收到的加密数据流进行解密。
7. **离散余弦变换(DCT)和逆变换(DCIT)处理器**。用于语音和视频处理。
8. **协议栈处理器**。在把应用数据发送到网络之前, 协议栈必须准备就绪。协议栈有许多标题字(header word)。接收端根据接收到的协议栈, 接收相应的应用数据。TCP/IP 协议栈处理器用于处理 TCP/IP 网络数据。
9. **网络处理器**。网络处理器的功能是建立完成连接, 发送和接收应答, 发送和接收重(转)发请求, 并检测和更正已接收数据帧的错误。网络处理器的功能包括所有协议栈处理的功能。
10. **加速器(例如 Java 代码加速器)**。加速器通过提前执行对 Java 程序中下一个对象的即时编译(just-in-time compilation), 加快运算速度。
11. **CODEC(编码器和解码器)**。经过 CODEC 处理, 声音、语音、图像、视频信号或位串在编码后保存或传输, 或者这些保存和接收的位串或信号经过解码后显示或播放。CODEC 还具有对声音、语音、图像和视频信号进行压缩和解压缩的功能。例如, 在文件或数据流中, 编码像素数据。编码器执行这个操作, 并把编码的数据存储在内存中。解码器检索已编码的数据, 进行显示或播放。编码的文件或数据, 例如 jpg 或 mp3 文件数据, 在播放过程中由解码器解码, 或者从内存中检索, 以显示图片、播放视频或执行其他操作。CODEC 是一个处理器电路, 对输入信息、位串或信号进行编码, 对经过编码的信息、位串或信号进行解码, 产生全套位串或原信号。
12. **JPEG CODEC**。这是一种用于 jpg 压缩和解压缩的处理器。联合图像专家组(Joint Photographic Experts Group, JPEG)是国际电信联盟电信标准部(ITU-T)和国际标准组织(ISO)的一名成员。
13. **MPEG CODEC**。运动图像专家组(Motion Pictures Experts Group, MPEG)提出了针对视频的 CODEC 标准。MPEG3 CODEC 是一个处理 mp3 压缩和解压缩的处理器。MPEG2、

3 或 4 对音频或视频数据压缩后才能存储或传输，解压缩后才能恢复或播放文件。使用 GPP 运行嵌入式软件时，需要一个独立的 DSP，让系统执行实时处理功能。

注意：

专用处理器用于专门的应用、计算，或者作为外围设备、直接内存访问和总线的控制器。

1.3 系统中的嵌入式硬件单元和设备

软件嵌入硬件里。硬件包括电路板、ASIC、SoC 上的许多构建块以及处理器。

1.3.1 构建块

系统中的硬件一般包含如下构建块和设备：

1. 电源

嵌入式系统中各个单元的操作范围都属于以下 4 种之一： $5.0V \pm 0.25V$ 、 $3.3V \pm 0.3V$ 、 $2.0V \pm 0.2V$ 和 $1.5V \pm 0.2V$ 。

2. 时钟振荡电路和时钟单元

时钟控制着执行一条指令的时间。频率依赖于处理器电路的执行速率。

3. 系统定时器

对定时器电路进行适当的配置，可以作为系统时钟(system-clock)。系统时钟周期性地生成系统中断，系统时钟中断允许在 OS 中定期执行系统监控操作。系统时钟可以 1 秒生成 60 次中断。

注意：

系统需要系统时钟，在系统的 OS 中，系统时钟还定期驱动系统监控功能。微控制器为时钟提供了定时器电路。

4. 实时时钟

系统需要实时时钟(RTC)。该时钟还驱动定时器执行系统中的各种计时和计数任务，更新系统中的时间和日期。微控制器还提供了用作计数和定时装置的定时器电路。

5. 复位电路、加电复位和 watchdog 定时器复位

复位电路允许一开始使用开关或信号启动系统。复位也可以使用处理器的指令进行。加电复位电路允许在打开系统的电源时启动系统。在某组指令集中执行的时间超过了预设的时间间隔时，watchdog 定时器复位就允许系统重新启动。

示例 1-4

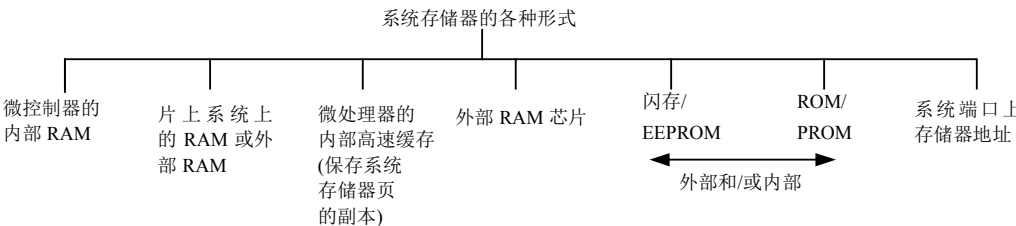
- 1. 考虑一个控制温度的系统。假设当程序开始执行时，传感器输入端工作正常。然而，在达到想要的温度之前，传感器电路产生了某种错误。如果系统没有复位，控制器将继续传送电流，而不停止。因此需要 watchdog 定时器复位，该复位可以让系统退出当前被挂起的状态。
- 2. 考虑另一个例子：一个控制机器人的系统。假设机器人胳膊中的接口发动机控制电路在运行时产生了错误。这种情况下，如果没有 watchdog 定时器，机器人的胳膊会继续运动。如果不停止，机器人会折断自己的胳膊！

根据复位开关、复位指令或加电复位进行的复位和 watchdog 复位，都可以从相同的指令或不同的启动指令执行。

6. 存储器

一个系统中有各种类型的存储器。图 1-4 给出了系统中出现的存储器的各种形式。它们是：

- (1) 内部 RAM
- (2) 内部 ROM/PROM/EPROM
- (3) 外部 RAM，存储临时数据和堆栈(在大多数系统中)
- (4) 高速缓存(在流水线和超标量微处理器中)
- (5) 内部 EEPROM 或闪存
- (6) 记忆棒(或卡)：在数码相机和移动系统中存储视频、图像、歌曲、语音和大存储对象。
- (7) 外部 ROM 或 PROM，存储嵌入式软件(在几乎所有不是基于微控制器的系统中)
- (8) 端口的 RAM 内存缓冲区



示例 1-5

- 1. 8051 微控制器中的 256 或 512 字节，用于寄存器、临时数据和堆栈。
- 2. 4KB 或 16KB 的 EPROM，用于 8051 微控制器中的程序。
- 3. 闪存允许把处理结果保存在非易失存储器中。手机联系人号码和数据存储在闪存中。闪存定期保存系统的状态，它会存储图像、歌曲或语音。这些数据在使用适当的格式压缩后存储起来。
- 4. 索尼记忆棒 Micro(M2)的尺寸是 15×12.5×1.2 mm，闪存空间是 2GB。它的数据传输率是 160Mbps(每秒 Mega bit)，PRO-HG 是 480Mbps，写入速率是 120Mbps。

表 1-1 列出了嵌入式系统中存储器的功能。ROM、PROM 或 EPROM 中嵌入了系统专用的软件。

表 1-1 嵌入式系统中存储器的功能

所需要的存储器	功 能
ROM、EPROM 或闪存	用来存储应用程序，处理器从中取出指令代码。存储系统用于引导、初始化的代码，存储初始的输入数据和字符串。存储 RTOS 的代码。存储指向不同中断服务例程(ISR)的指针(地址)
RAM(内部 RAM 或外部 RAM)和用于缓冲区的 RAM	在程序运行时存储变量和堆栈。例如，在语音或图像应用中用来存储输入和输出缓冲区
记忆棒	将闪存记忆棒插入移动计算系统和数码相机中。存储高分辨率视频、图像、歌曲或经过适当压缩的语音，以及大量永久性数据
EEPROM 或闪存	存储非易失性的处理结果
高速缓存	在外部主存储器预先存储指令和数据的副本，在快速处理期间临时存储处理结果

系统可将下列程序数据嵌入(放置到)微控制器的内部闪存、ROM 或 PROM 中，也可以放置在外部分存、ROM 或 PROM 中：引导程序、初始化数据、初始屏幕显示的字符串或系统的初始状态、各种任务程序、ISR 和操作系统内核。系统中的 RAM 用来保存临时数据，以及程序运行时所需要的堆栈和缓冲区。系统使用闪存存储非易失性结果。

7. 输入、输出、I/O 端口、IO 总线和 IO 接口

系统通过输入端口从物理设备获得输入，例如：

示例 1-6

1. 系统可从触摸屏、小键盘或键盘的按键、传感器和变频电路获得输入。
2. 系统中的控制电路从端口、传感器和变频电路获得输入。
3. 信号接收器或网卡从通信系统获得输入(通信系统可以是传真机、调制解调器或广播服务)。
4. 网络的信号也可在端口上接收。
5. 考虑巧克力自动售卖机中的系统。它从小键盘或触摸屏中获得用户指令。它从一个端口获得输入，这个端口收集儿童投入的硬币。
6. 考虑移动电话中的系统。用户通过触摸或按键直接或间接(通过从存储器中重新调用号码)输入移动电话号码。键盘通过输入端口与系统连接。外部信号是通过内部天线进入的输入。

处理器根据每个输入端口的存储器缓冲地址(也称为端口地址)来识别这些端口。正如一个字节或字的存储单元通过地址来标识一样，每个输入端口也是用地址标识的。系统通过对端口地址的读操作来获得输入。

系统还具有输出端口，通过这些端口可向外部发送输出字节，例如：

示例 1-7

1. 输出结果可输出到 LED(发光二极管)、LCD(液晶显示器)或触摸屏面板上。例如, 计算机或移动电话系统向 LCD 或触摸显示屏发送输出数字或 SMS 消息。
2. 系统可以向打印机发送输出数据。
3. 输出结果可以发送给通信系统或网络。
4. 控制系统向报警器、驱动器、熔炉或锅炉发送输出数据。
5. 机器人向其各种电动机发送输出数据。

每个输出端口都是通过本身的存储器缓冲地址(称为端口地址)标识的。系统通过一个写操作向端口地址发送输出数据。

通用端口可以同时进行输入和输出(I/O)操作, 例如:

示例 1-8

移动电话系统通过一条无线通信通道既可以发送输出数据, 又可以接收数据。移动计算系统中的触摸屏系统通过用户触摸屏幕上显示的菜单或按键, 就可以发送输出数据和接收输入数据。

每个 I/O 端口也是通过地址标识的, 在这个地址上可以进行读写操作。

端口可以和系统地址以及数据总线进行串行通信或并行通信。串行通信使用一条比特位的数据线, 比特流在连续的时间槽中顺序传输。通用异步收发(UART)是串行通信常用的协议。在并行通信中, 多条数据线并行发送多个比特位的数据。

一个系统端口可能需要向多条通道发送输出数据。这时就要使用信号分离器和多路器。

8. 总线

一个系统的处理器可能要与其他几个设备和系统连接。总线就是连接着多个设备、硬件单元和系统, 负责任意两个单元之间通信的一套公用线路。在这两个单元通信期间, 其他单元都处于未连接状态。

总线通信协议规定了总线上的信号通信方式。总线分为串行总线和并行总线, 在一次传输中分别传输单个和多个数据比特位。

总线协议还规范了下列内容: (1)多个设备需要通过总线通信时的仲裁方法; (2)每个设备的总线轮询方式; (3)设备菊花链的形式, 使总线可根据链上设备的优先级授权给设备。

示例 1-9

对于分布式单元或系统的网络互联, 有不同类型的串行和并行总线协议可供使用, 例如, I²C、CAN、USB、ISA、EISA 和 PCI。对于系统的无线网络互联, 可以使用 802.11、IrDA、蓝牙以及 ZigBee 协议。

注意:

系统通过并行或串行端口与外部物理设备和系统相连。信号分离器和多路器通过一条通用路径可以与多条通道进行信号通信。系统经常通过 I/O 总线与其他设备和系统进行网络互连: 例如, I²C、CAN、USB、ISA、EISA 和 PCI 总线。

9. 数字到模拟转换器(DAC)

DAC 是一个用于将 8 位、10 位或 12 位数字输入转换为模拟输出的电路。DAC 操作在微控制器中的脉宽调制器(Pulse Width Modulation, PWM)单元和积分器电路的帮助下完成。

10. 模拟到数字转换器(ADC)

ADC 是将模拟输入转换为 4 位、8 位、10 位或 12 位数字输出的电路。模拟输入作用于 ADC 正负管脚，并依照参考电压转换为数字输出。当输入等于正负参考电压的差值时，所有输出位都等于 1；当输入等于负参考电压(通常为 0V)时，所有输出位都等于 0。

嵌入式系统微控制器中的 ADC 单元可以有多通道。它可以连续从与不同模拟源内部相连的各个管脚获得输入。

注意：

系统必须为自动控制和信号处理应用提供必要的接口电路。DAC 单元和 ADC 单元允许控制和处理信号。

11. LCD、LED 和触摸屏显示

系统需要一个接口电路和软件来显示状态。

LED 用于显示系统的 ON 状态。闪烁的 LED 可以表示一项特定的任务即将结束或者正在运行。它还可以表示正在等待信息的状态。

屏幕可以显示状态或消息。屏幕可以显示单行、多行字符或闪烁。LSI(Lower Scale Integrated Circuit, 小规模集成电路)在 LCD 矩阵显示中用作显示控制器。

触摸屏既是一个输入设备，又是一个输出设备。它可以用来输入命令、选择菜单或予以应答。用手指或手写笔物理接触屏幕位置就可以输入信息。手写笔是一个细铅笔形的物体，可以拿在手上像笔一样使用。屏幕显示选项或命令的磁贴。它显示磁贴和键按钮，而不是菜单、对话框和图标。LSI 用作触摸屏控制器，就像计算机视频显示器一样。

注意：

系统必须提供必要的接口电路和软件，供 LCD 显示控制器、LED 接口端口和触摸屏 IO 接口输出使用。

12. 小键盘或键盘

小键盘或键盘是获得用户输入的重要设备。触摸屏为移动计算系统提供了虚拟小键盘，它是显示在 LCD 屏幕显示器上的小键盘，用户通过触摸就可以输入按键。在 LCD 屏幕显示器上，命令显示为磁贴，用户可在磁贴上通过触摸输入命令。

小键盘或键盘可以与系统交互。系统可以提供必要的接口电路以及软件，使系统能够从小键盘、键盘或触摸屏控制器直接获得输入。

13. 中断处理器

一个系统可能有许多设备，系统处理器必须为每个中断事件运行一个适当的中断服务例程

(Interrupt Service Routine, ISR), 从而控制和处理每个设备的请求。每个系统中都必须有一个中断处理机制, 处理来自系统中各个事件或过程的中断, 以及处理同时产生的多个等待服务的中断。

示例 1-10

1. 当预设时间到期时, 定时器设备会发送一个超时中断; 而在当前时间等于预设时间时, 定时器设备会发送一个比较中断。
2. 假设要传送来自键盘、打印机或触摸屏的数据, 每个动作都会产生一个中断。
3. 外围设备接口接收到输入数据或传输缓冲区为空时, 就会产生一个中断。

操作系统会控制并处理中断, 为中断运行例程。

注意:

系统必须有一个中断处理电路, 在物理设备、系统、软件指令发生中断和软件异常时执行 ISR。操作系统也控制中断的处理。

1.3.2 嵌入板

电路板设计为使用 Intel 8051 和 80251、ATMEL mega 系列 8/168/324/1284/2560、Freescale 68HS12 和更高的 Microchip PIC 16F84 或 16C76、16F876、18 和 24、ARM、Intel、Nexperia、三星和 ST 微电子 ARM 的微处理器或微控制器板。该电路都需要构建块。嵌入式软件会下载到这个板上。该板用模拟器和调试器软件来测试。

1.4 嵌入式系统中的软件和编程语言概述

软件是嵌入式系统的核心。

1.4.1 嵌入式软件 ROM 映像

通用计算系统把软件加载到 RAM 后, 运行程序。嵌入式系统处理器和系统都需要软件, 该软件是系统中给定应用所特定的。系统的处理器处理指令代码和数据。在设计的最后阶段, 这些代码和数据放在存储器(ROM 或闪存)中, 软件支持系统中的专用应用。

所有任务都使用 ROM 或闪存中的指令执行。在设计的最后阶段, 软件也称为 ROM 映像。正如图像是像素独特的排列方式一样, 嵌入式软件也是指令和数据独特的字节排列方式。

每条代码或数据只以位或字节的形式出现。根据要执行的任务, 系统需要每一个 ROM 地址上的字节。因此, 机器可实现的软件文件类似于一个表格, 该表格的每一行包含存储器的地址和字节。这些字节存储在系统存储器的每个地址中。这个表必须作为目标硬件的 ROM 映像存在。

图 1-5 给出了系统存储器中的 ROM 映像。这个映像包括引导程序、堆栈地址指针、程序计数器地址指针、应用程序、ISR、RTOS、输入数据和向量地址。

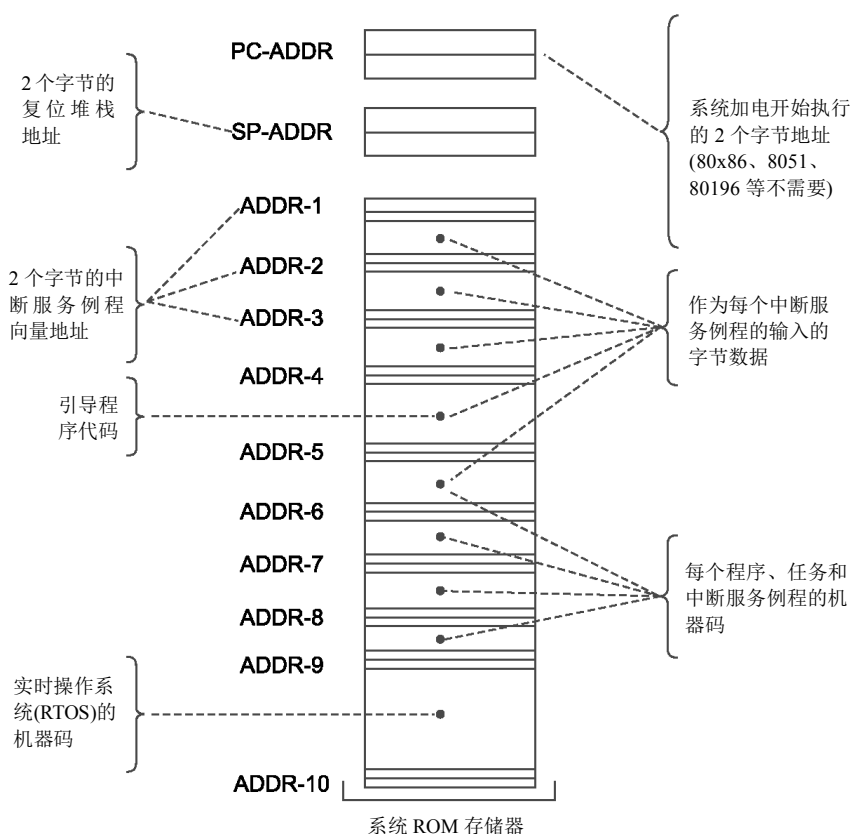


图 1-5 系统 ROM 存储器中嵌入的软件、RTOS、数据和向量地址

注意：

最终阶段软件也称为 ROM 映像。系统的最终机器可将软件嵌入 ROM(或 PROM)中,就像画框中的图像。为创建 ROM 映像,必须定义每个地址上的字节。通过修改这个映像,相同的硬件平台可执行不同的操作,可用于完全不同的应用或同一系统的升级版本。

1.4.2 用机器码编写软件

在以这种形式进行编码的过程中,程序员定义地址以及每一个地址上相应的字节或位。在配置某个物理设备或子系统时,使用基于机器码的编码方式。

示例 1-11

1. 某种机器码或位可以配置收发器,使之以特定的 Mbps 或 Gbps 进行信息传输,并使用特定的总线协议和网络协议。
2. 使用某种代码或位来配置处理器中的控制寄存器或设备寄存器。这种设备的示例是定时器,它拥有定时器模式和控制寄存器。
3. 在处理特定代码段的过程中,可以启用或禁用其内部缓存。禁用缓存,会减少处理过程中的电耗。

然而,只能在特定的情形下用机器码进行编码,这是很浪费时间的,因为程序员必须首先

熟悉处理器指令集，还要记住这些指令及其机器码。

1.4.3 用特定于处理器的汇编语言编写软件

当程序员完全理解处理器及其指令集后，就可以用汇编语言(assembly language)编写程序或者特定的一小部分程序。汇编器是开发汇编语言代码时使用的软件。

用汇编语言编写所有代码是非常耗时的。完全用汇编语言编码只适用于少数简单的小型系统，例如玩具、机器人或者数据采集系统。

图 1-6 给出了将汇编语言程序转换为机器可实现的软件文件，最后得到 ROM 映像文件的过程。

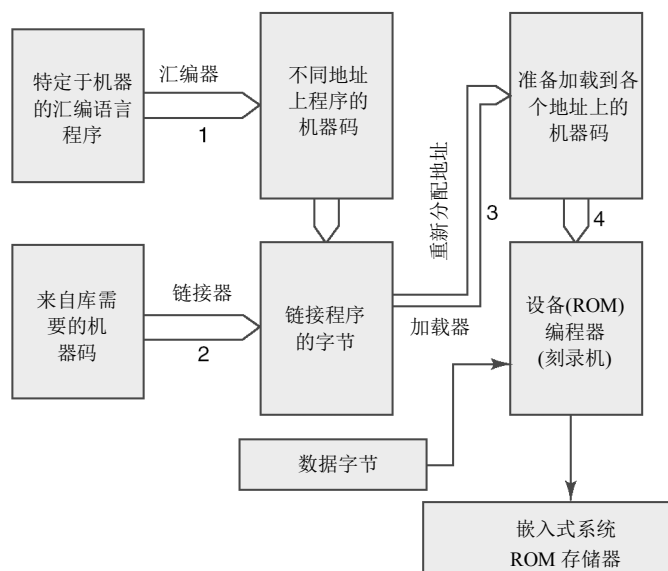


图 1-6 将汇编语言程序转换为机器码文件，最后在闪存或 ROM 上烧制 ROM 映像的过程

汇编语言编码对于配置物理设备来说非常有用，例如端口、行显示接口、ADC 和 DAC 以及从缓冲区读取或发送数据。这些代码称为设备驱动程序低级代码。它们能够运用特定处理器或设备的特性，并提供一种优化的编码方式。

注意：

汇编代码允许使用特定于处理器的特性。在 3 个步骤中，分别使用“汇编器”、“链接器”和“定位器”创建一个文件。该文件具有标准格式的 ROM 映像。设备编程器最终将这个映像烧写到 PROM 或 EPROM 中。从映像创建的掩膜允许 ROM 成为 IC 芯片的形式。

1.4.4 用高级语言编写软件

多数情况下，将所有代码都用汇编语言编写是很浪费时间的。因此，软件可以用高级语言开发，例如 C、C++、Visual C++ 或 Java 语言。程序员完全没必要知道处理器指令集。

Linux、实时操作系统或.NET 框架也用于嵌入式编程。在大多数嵌入式系统中，若系统同

时处理多个过程，任务有实时限制和最后期限时，就使用 RTOS。

图 1-7 展示了典型的嵌入式 C 软件中的不同编程层。这些层是：(1)预处理器指令。(2)主函数。(3)中断服务例程(函数)。(4)多进程或任务，如任务 1~N。(5)操作系统内核和调度程序。(6)标准库函数、协议函数和栈分配函数。

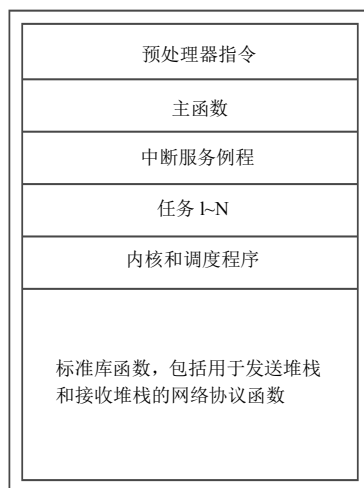


图 1-7 嵌入式 C 软件中的各种编程层

C、C++、Java 和 Visual C++是用于软件开发的语言。C 程序有多个层：处理器指令、主函数、任务函数和库函数、中断服务例程和操作系统内核(调度程序)。编译器生成目标文件。使用链接器和定位器，为目标硬件生成 ROM 映像文件。

第 8 章将介绍嵌入式软件的编码过程中使用的不同模型。

1. 高级语言的优点

考虑下面的问题：

示例 1-12

计算 127、29 和 40 的和，并输出平方根。

可用于所有处理器的 C 语言程序示例如下：

```
# include <stdio.h>
# include <math.h>
void main (void)
{int i1, i2, i3, a; float result;
i1=127; i2=29; i3=40; a = i1 + i2 + i3; result = sqrt (a);
printf (result);}
```

显然，求平方根的代码需要编写许多行，只能由熟练的汇编语言程序员完成。使用高级语言编写这个程序比使用汇编语言简单得多。C 程序有一个特征，当使用某种特定处理器特性，并编写特定部分的代码时，可以加入汇编指令，例如端口设备驱动程序。

2. 编译和嵌入高级语言程序

图 1-8 给出了将一个 C 程序转换为 ROM 映像文件的过程。编译器(compiler)生成目标代码。

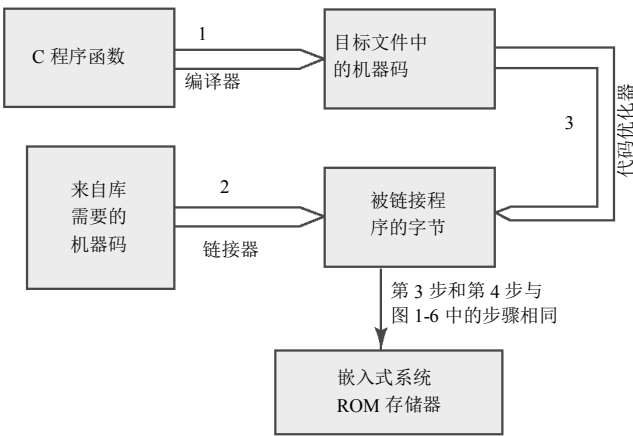


图 1-8 将 C 程序转换为 ROM 映像文件的过程

编译器根据处理器指令和其他说明对代码进行汇编。作为编译的最后一个步骤，嵌入式系统的 C 编译器必须使用代码优化器(code-optimizer)。优化器在链接之前，对代码进行优化。编译后，链接器(linker)将目标代码与其他必要的代码链接在一起。例如，链接器将某些函数代码包含进来，如 printf 和 sqrt 代码。设备管理和驱动程序代码(设备控制代码)也是在这个阶段链接的：例如，打印机设备管理和驱动程序代码。链接后，创建 ROM 映像文件的其他步骤与图 1-6 中所示的步骤相同。

1.5 嵌入式系统的设计过程

所谓设计，就是用简单形式表现系统。嵌入式系统的设计过程包括如下步骤：

- 抽象(Abstraction)：为了设计抽象出系统。

示例 1-13

1. 巧克力自动售卖系统可以抽象为带有如下单元的机器：
 - a. 带有小键盘的用户界面，供用户输入所选的巧克力
 - b. 获取用户钱币的机制
 - c. 交付所需巧克力的机制
 - d. 显示系统，允许用户与机器交互
2. 智能卡用作信用卡和银行 ATM 机的取款卡，可以抽象为带有如下单元的卡：
 - a. 卡中的通信硬件和嵌入式软件
 - b. 把持卡人的身份和银行授权交付给读卡器硬件的卡

- 硬件和软件体系结构(Hardware and Software architecture)：完成体系结构的设计，这有助于理解硬件单元及其接口、软件组件和数据流路径。

- **附加功能属性(Extra functional property):** 待开发系统的附加功能属性应该在设计中充分加以理解。
- **相关系统系列设计(System related family of design):** 在设计过程中, 应该对早期开发的相关系统系列加以考虑。
- **模块化设计(Modular Design):** 使用模块化设计概念。系统设计是:
 - a. 将硬件分解为要使用的、接口化的、互连接的单元
 - b. 将软件分解为要编码的模块(组件)
- **映射(Mapping):** 根据软件和硬件需求映射为各种表示形式。
- **用户界面设计(User Interface Design):** 用户界面设计是设计的重要组成部分。用户的界面是按照每个用户的需求、环境和对系统功能的分析进行设计的。
- **改进(Refinement):** 每个单元和组件的设计都需要反复改进, 直到它们最适合嵌入式软件和硬件小组的实现为止。

软件设计过程可能需要使用工具。图 1-9 列出了巧克力自动售卖系统的设计单元。

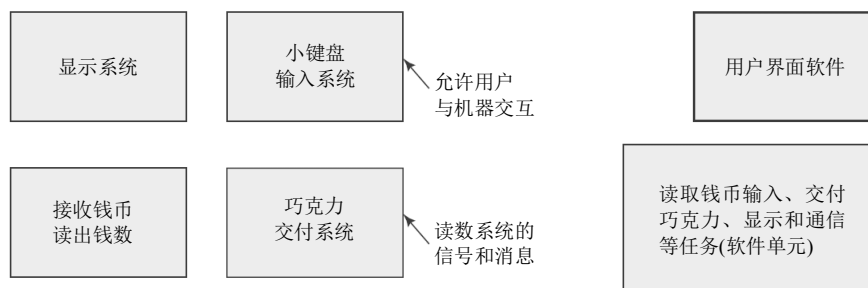
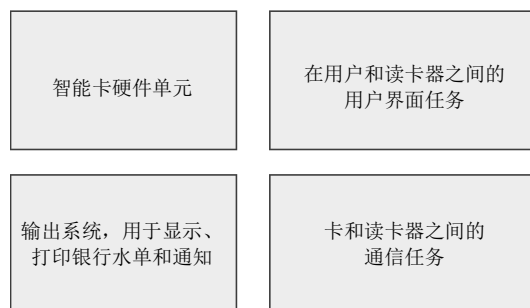


图 1-9 巧克力自动售卖系统的设计单元

图 1-10 显示了智能卡的设计单元。



在卡和读卡器之间交换键和信息

图 1-10 智能卡的设计单元

1.6 嵌入式系统的体系结构

体系结构意味着系统结构计划的表示。嵌入式系统的体系结构就是嵌入式系统结构中软硬件单元计划的表示。

硬件体系结构可表示为单元及其接口的块图。软件体系结构可以表示为软件组件和数据流路径的块图。

体系结构显示了设计系统时需要的单元，显示设计过程中需要的接口。接着设计硬件单元的接口，然后在硬件单元中嵌入、集成软件，构成嵌入式系统。

图 1-11 显示了巧克力自动售卖系统的软硬件体系结构。

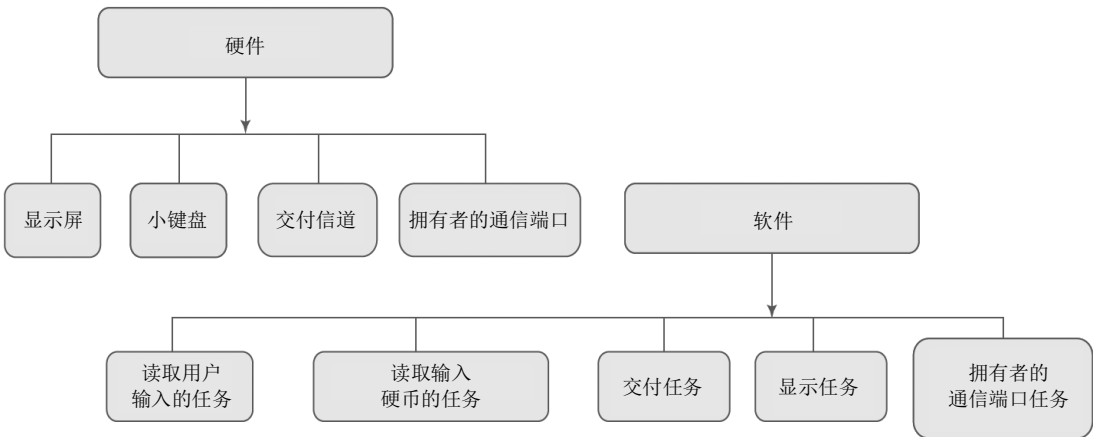


图 1-11 巧克力自动售卖系统的软硬件体系结构

图 1-12 显示了智能卡的软硬件体系结构。

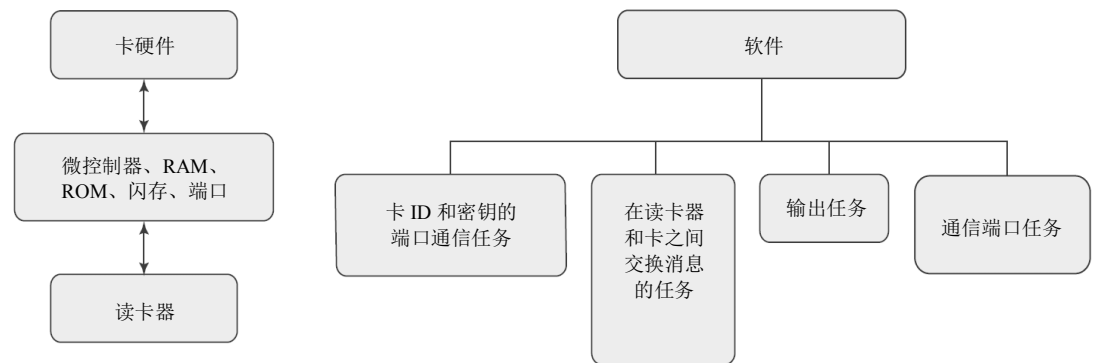


图 1-12 智能卡的软硬件体系结构

体系结构描述语言(ADL)用于通过如下项表示软件体系结构：

- (1) 控制层次结构
- (2) 结构划分
- (3) 数据结构和层次结构
- (4) 软件过程

1.7 嵌入式系统的模型

模型也是用简单形式表现系统。

示例 1-14

1. 网络系统的 OSI 模型有 7 层：

- a. 应用层
- b. 显示层
- c. 会话层
- d. 传输层
- e. 网络层
- f. 数据链接层
- g. 物理层

2. 嵌入式系统软件可建模为包含如下层：

- a. 应用层包含应用程序接口和用户界面
- b. 系统软件层包含中间件、操作系统和设备驱动程序
- c. 硬件层包括处理器、存储器、IO 单元和显示单元

中间件意味着软件，它允许从一个系统中输入，输出到另一个系统中。例如，移动系统中中间件允许从无线系统输入，输出到移动系统上。

图 1-13 显示了巧克力自动售卖系统的模型。

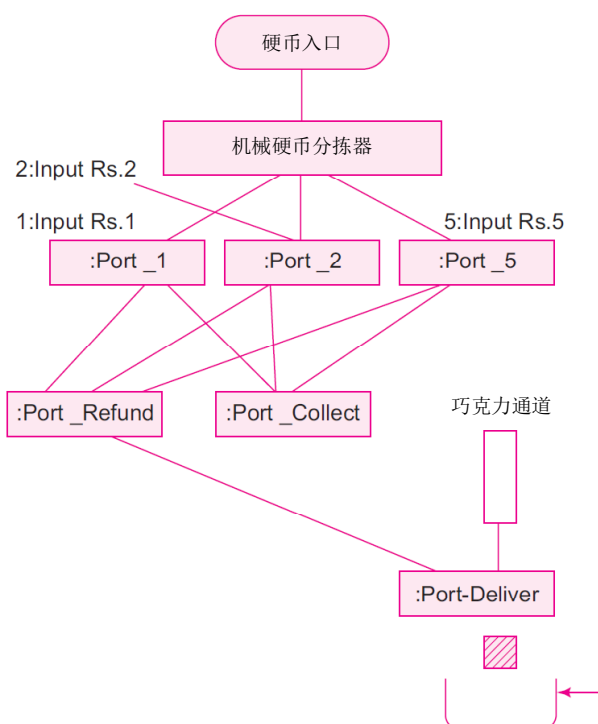


图 1-14 显示了智能卡系统的模型。

嵌入式软件设计过程用到多种不同的程序模型，如下所示：

(1) 顺序程序模型

- (2) 面向对象编程模型
- (3) 针对数据路径的有限状态机
- (4) 多线程模型
- (5) 针对进程、线程和任务的并发处理

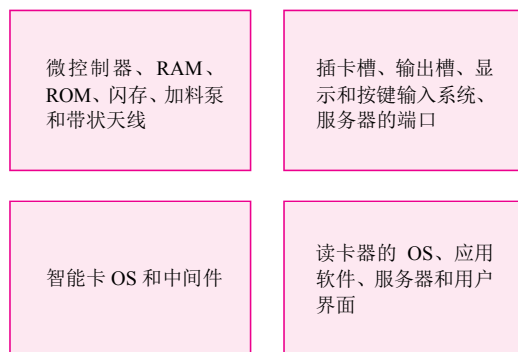


图 1-14 智能卡系统的模型

1.8 嵌入式系统的分类

考虑到设计的复杂度和系统性能，可将嵌入式系统分成以下三种类型：

1. 小型嵌入式系统

- (1) 采用一个8位或16位的微控制器或 DSP 设计
- (2) 硬件和软件复杂度很小，需要进行板级设计。
- (3) 可以是电池驱动的。
- (4) 可以使用专用的集成开发环境(IDE)，或者把所使用的处理器系列作为主编程工具(IDE有项目管理器、设备选择、代码生成、编译器、汇编器、库、加载器、定位器和实验室编程器、内部电路模拟、模拟和调试工具)。

(5) 通常使用 C 或 MATLAB(C 程序被编译为汇编程序，然后将可执行代码存放到系统存储器的适当位置)。

- (6) 可能是多任务，也可能不是，可能使用 RTOS，也可能不使用。

2. 中型嵌入式系统

(1) 采用一个或几个 16 位或 32 位的微控制器、DSP 或 RISC 设计，例如 ARM 系列微处理器和微控制器。

- (2) 还可以运用已有的专用处理器设计。
- (3) 使用实时操作系统(RTOS)处理多任务。
- (4) 使用软件工具提供硬件复杂性的解决方案。

(5) 对于复杂的软件设计，可使用如下编程工具：C/C++/Visual C++/Java 语言、RTOS、源代码工程管理工具、模拟器、调试器和集成开发环境(IDE)。

(6) 这些系统的 VLSI 设计形式使用容易获得的 IP(后面将介绍)来完成各种功能,例如总线控制器和 USB 接口。

这些系统的硬件和软件复杂度都比较大。

3. 复杂嵌入式系统

复杂嵌入式系统的软件和硬件都非常复杂,复杂系统是:

- (1) 在 32 位或 64 位的 RISC 上设计。
- (2) 需要硬件和软件的协同设计。
- (3) 需要在最终系统中集成组件。
- (4) 对于复杂的软件功能(如加密和解密算法、离散余弦变换和逆变换算法、TCP/IP 协议栈和网络驱动程序),可能需要在硬件中以 IP 核的方式实现,硬件的实现需要额外的处理速度。
- (5) 可能需要新的开发工具。
- (6) 可能在 SoC(片上系统)上。

这些系统用于需要协同设计硬件和软件、在最终系统中集成组件的顶尖应用。复杂系统的示例是移动计算系统。

1.9 嵌入式系统设计者需要具备的技能

嵌入式系统设计者必须使用已有的工具,并在给定的规范、费用和时间框架内开发产品。

1. 小型嵌入式系统设计者需要具备的技能

前文介绍的作者 Tim Wilmshurst 说过,开发小型嵌入式系统的个人或团队需要具备下列技能:“完全理解微控制器,具备计算机体系结构、数字电子设计、软件工程、数据通信、控制工程、电动机和制动器、传感器和测量、模拟电子设计、IC 设计和生产的基本知识。”

硬件领域需要的基础知识包括微处理器、微控制器、DSP、硬件单元、软件工程、数据通信、控制工程、电动机和制动器、传感器和测量、模拟电子设计、IC 设计和生产。

具有了使用 OS 编写软件的技能,就能使用硬件资源,让应用软件运行在硬件上。OS 支持多任务。OS 有设备驱动程序、文件系统或设备管理。嵌入式系统常常需要实时操作系统(RTOS)以完成实时调度、编程和多任务。需要 RTOS、C 编程、MATLAB、模拟器和调试器。

特定的情形下需要特定的技能。例如,当设计控制系统时,需要具备控制工程的知识;当设计系统接口时,需要具备模拟电子设计的知识。

2. 中型嵌入式系统设计者需要具备的技能

对于设计中型的嵌入式系统来说,C/C++/Java 语言编程和 RTOS 编程以及程序建模是必须具备的技能。下面的知识对于设计者来说是很关键的。(1)任务或线程以及 RTOS 对它们的调度。(2)协同式和抢占式调度。(3)Intel 处理器通信功能。(4)共享数据的使用,以及对临界区和可重入函数的编程。(5)信号量、邮箱、队列、套接字以及管道的使用。(6)中断延迟和满足任务最后期限的处理。(7)各种 RTOS 函数的使用。(8)物理和虚拟设备驱动程序的使用。

对于将要使用的特定微控制器，设计者必须使用具有应用编程接口(API)的 RTOS 编程工具。各种功能的处理方法，如存储器分配、定时器、设备驱动程序和中断处理机制，已经作为 RTOS 的 API 存在了。

设计者只需要知道硬件组织方式和这些 API 的用法即可。这样，微控制器或处理器就为设计者提供了一个小的系统元素，设计者只需要具备很少的知识就足够了。

3. 复杂嵌入式系统设计者需要具备的技能

嵌入式系统硬件工程师应该具备硬件单元的全面技能，还要具备 C/C++ 和 Java 语言、RTOS 和其他编程工具的基本知识。软件工程师应该具备硬件的基础知识，全面掌握 C 语言、RTOS 和其他编程工具的知识。最终的最佳设计方案要通过系统集成来获得。设计团队的成员必须协同设计，并解决硬件和软件设计以及系统集成中的高层次复杂问题。

1.10 示例嵌入式系统

嵌入式系统具有非常广泛的应用。其中几个应用领域是电信、智能卡、导弹和卫星、计算机网络、数字消费类电子及自动化。图 1-15 给出了嵌入式系统在这些领域中的应用。

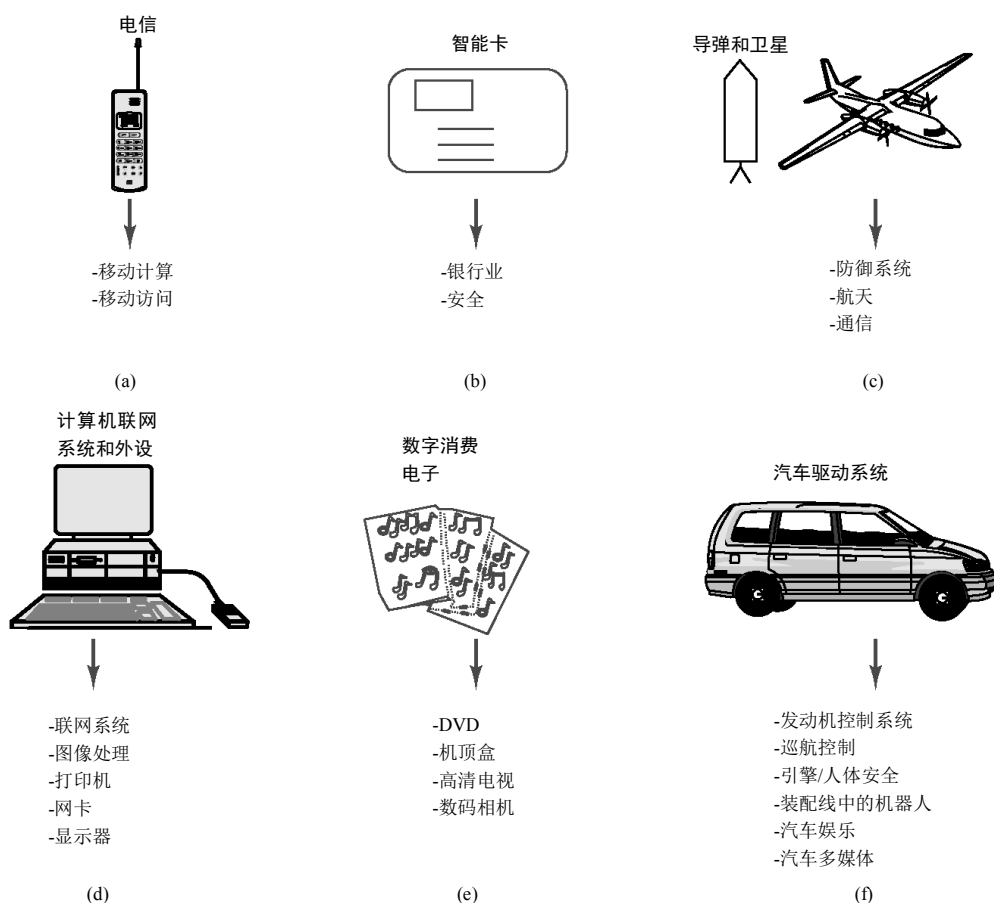


图 1-15 嵌入式系统在各个领域中的应用

小型嵌入式系统应用的几个示例如下所示：

- (1) 巧克力自动售卖机
 - (2) 用于机器人系统的步行发动机控制器
 - (3) 销售终端点
 - (4) 洗涤系统
 - (5) 烹饪系统
 - (6) 多功能玩具
 - (7) 基于微控制器的单显示或多显示数字板仪表，用于显示电压、电流、电阻和频率
 - (8) 键盘控制器
 - (9) 串口卡
 - (10) CD 驱动和硬盘驱动控制器
 - (11) 计算机外设控制器，如 CRT 显示控制器、键盘控制器、DRAM 控制器、DMA 控制器、打印机控制器、激光打印机控制器、LAN 控制器、磁盘驱动控制器
 - (12) 传真机、影印机、打印机或扫描仪
 - (13) 远程电视(控制器)
 - (14) 发动机控制系统，如 DC 发动机、机器人和 CNC 机的速度和位置的准确控制；自动应用，如闭环引擎控制、动态行驶性能控制、防锁制动系统监测器
 - (15) 电子数据采集和监督控制系统
 - (16) 电子器械，如工业过程控制器
 - (17) 电子智能重量显示系统和工业湿度记录器及控制器
 - (18) 信号波形、电表或水表读取数字存储系统
 - (19) 光谱分析器
 - (20) 生物医学系统，如 ECG LCD 显示及记录器、血细胞记录器及分析器、病人监控系统
- 中型嵌入式系统的几个示例如下所示：

- (1) 计算机网络系统，如路由器、服务器的前端处理器、交换机、网桥、网络集线器及网关
 - (2) 网络设备
 - (3) 娱乐系统，如可视游戏和音乐系统
 - (4) 银行系统，如银行 ATM 和信用卡、刷卡机
 - (5) 自动信号跟踪器和目标跟踪器
 - (6) 通信系统，如移动通信 SIM 卡、数字寻呼机、便携式电话、有线电视顶盒、FAX 收发机
 - (7) 图形加速器
 - (8) 图像过滤、图像处理、模式识别机、语音处理和视频处理
 - (9) 可视游戏
 - (10) 嵌入式防火墙/路由器
- 复杂嵌入式系统的示例如下所示：
- (1) 移动或平板电脑(电话计算系统)
 - (2) 无线 LAN 嵌入式系统

(3) 视频、交互视频、宽带 IPv6(Internet Protocol Version 6)Internet 和其他产品、实时视频和音频或者多媒体处理系统

(4) 使用高速(大于 400MHz)、极高速(10Gbps)和大带宽的接口和网络系统: 路由器、LAN、交换机(和网关)、SAN(存储区域网络)、WAN(广域网)

(5) 安全产品及高速网络安全

(6) GB 速率的加密产品

本章小结

- 嵌入式系统是一种在计算机硬件中嵌入软件的系统, 该系统专用于某个应用, 或者应用或产品的特定部分, 或大型系统的一个组件。
- 嵌入式系统处理器可以是多个系列的微处理器中选择一个通用处理器。此外, 系统处理器也可以是一个专用指令处理器(ASIP), 例如微控制器、嵌入式处理器和数字信号处理器(DSP), 系统还使用专用处理器。示例有图形处理器、总线控制器、用户界面处理器、CODEC 和触摸屏输入输出控制器。
- 嵌入式系统需要电源、时钟和复位电路、接口、输入输出(IO)端口、串行 UART 以及其他端口。端口接收输入, 发送输出, 连接外围设备。系统需要显示单元、小键盘(或键盘)、定时器和 watchdog 定时器。嵌入式系统需要中断控制单元。
- 嵌入式系统把软件映像嵌入(定位)到 ROM 中。
- 嵌入式软件通常使用高级语言 C、C++、Java 或 Visual C++编写的, 需要加入、启用或禁用某些编程特征。C 和 C++语言还有助于合并汇编语言代码。
- 任何计算系统都使用 OS。OS 允许使用硬件资源, 让应用软件运行在硬件上。OS 支持多任务。OS 包括设备驱动程序、文件系统或设备管理。嵌入式系统常常需要实时操作系统(RTOS)以完成实时调试、编程和多任务。
- 在嵌入式系统的开发 and 设计阶段, 需要集成开发环境和一些软件工具。
- 介绍了嵌入式系统的设计、体系结构设计和建模。
- 大量应用和产品都使用了嵌入式系统。
- 根据软硬件的复杂度, 把嵌入式系统分为 3 类。设计该系统需要多种技能。

关键词及其定义

- **ADC:** 一个按照需要, 根据参考电压将处于正极和负极之间的模拟输入转换为 8 位、10 位或 12 位数字信号的电路。当输入等于正负参考电压的差值时, 所有输出位都等于 1; 当输入等于负参考电压(通常为 0)时, 所有输出位都等于 0。
- **ASIP(Application Specific Instruction Set Processor, 专用指令集处理器):** 含有专用于 VLSI 芯片上特定应用指令集的处理器, 例如微控制器、DSP、I/O、媒体、网络或其他特定领域处理器。
- **汇编器(Assembler):** 将汇编语言软件翻译为机器码, 放在 .exe(可执行)文件中的程序。

- **ASSP(Application Specific System Processor, 专用系统处理器):** 系统特定任务的处理单元, 如图像处理、压缩和解压缩, 它通过总线与主处理器一起集成在嵌入式系统中。
- **总线(Bus):** 总线就是连接着多个设备和硬件单元, 负责任意两个单元之间通信的一套公用并行线路。
- **总线通信协议:** 规定了总线上的信号通信方式, 还规范了多个设备需要通过总线通信时的仲裁方法, 以及设备请求对总线的轮询方式。
- **缓存(Cache):** 处理单元中执行快速读写的片上存储器。它与处理器的执行单元关联。它存储了指令或数据。它提前从 ROM 或 RAM 中取出这些指令或数据, 再把数据写回 RAM。有了缓存, 处理器执行单元就不必等待来自外部总线的指令和数据。它仍以更快的速度把数据写回 RAM。
- **时钟(Clock):** 由一个振荡电路产生的固定频率脉冲, 控制处理过程中的所有操作以及系统中的所有定时参考。
- **时钟频率:** 每秒脉冲的次数。频率依赖于处理器电路的需求。如果处理器需要 100MHz 的时钟, 则它的最小指令执行时间是这个值的倒数, 处理器处理每个指令需要一个循环, 时间是 10ns。
- **CODEC(编码器和解码器):** CODEC 是一个实现对输入信息、位串或信号进行编码, 以及对经过编码的信息、位串或信号进行解码, 产生全套位串或原信号的电路。CODEC 可对声音、语音、图像和视频信号进行编码或解码。CODEC 还具有对声音、语音、图像和视频信号进行压缩和解压缩的功能。
- **编码器(Coder):** 编码器是 CODEC 电路的一部分, 用于对输入信息、位串或信号进行编码。
- **编译器(Compiler):** 一个按照处理器规范、从高级语言生成机器码的程序。这些机器码称为目标代码。
- **周期(Cycle):** 一个周期包括从 RAM/ROM 中取出指令、在处理器执行和写回运行结果。
- **DAC:** 将数字信号(8 位、10 位或 12 位)转换为与参考电压成比例的模拟信号的转换器。当所有输入位都为 1 时, 模拟输出为正负参考管脚电压的差值; 当所有输入位都为 0 时, 模拟输出等于负参考管脚电压(通常为 0V)的值。
- **解码器(Decoder):** 解码器电路对输入地址进行解码, 从多个输出中选择一个激活。它用于对多个可编址单元进行选择。解码器还可以对已编码数据进行解码, 生成输出数据、信号或信息。
- **信号分离器(Demultiplexer):** 拥有多个通道数字输出的数字电路。信号分离器接收到的输入中包含通道地址位, 使其通过通道地址位可以确定输出被发送到哪个通道。信号分离器获得输入后, 从多个输出通道选择一个把输入发送给该通道。一个信号分离器只能向多条通道中的一条输出数字信号。
- **设备驱动程序(Device Driver):** 高级语言函数, 实现对设备的打开(配置)、连接、绑定、侦听、读、写或关闭操作。调用这些函数, 会执行软件中断服务例程。设备 ISR 在外围设备的控制寄存器(或字)编程之后运行, 使设备获得输入或输出。它们从状态寄存器中读取寄存器状态, 从设备获取输入, 并把输出写入设备。该例程在设备中断时执行。

- **设备管理器(Device Manager):** 操作系统中的一个软件组件, 管理多个设备及其驱动程序。
- **设备编程器(Device Programmer):** 它从定位器生成的文件中获得输入, 并烧录可熔断链接, 接着 ROM 存储数据和系统代码, 作为系统软件的 ROM 映像。
- **嵌入式系统(Embedded System):** 嵌入式系统是一种在计算机硬件中嵌入软件的系统, 该系统专用于某个应用, 或者应用或产品的特定部分, 或大型系统的一个组件。
- **文件(File):** 一种数据集, 将记录(字符或字)发送给数据目的端(如程序结构), 并存储来自于数据源(如程序结构)的数据。计算机中的文件还可存储在硬盘中。
- **文件系统(File System):** 文件系统指定了文件创建、调用、命名、使用、复制、保存和删除的方式。
- **FPGA:** 芯片上的现场可编程门阵列。芯片中有大量的阵列, 每一个元素都有可熔断的链接。阵列的每一个元素包含许多XOR、AND、OR、多路器、信号分离器和三态门。通过从存储器中传输数据对链接进行适当的编程, 可以在芯片上构造复杂的数字电路设计。
- **GPP(General-Purpose Processor, 通用处理器):** 来自几个系列微处理器的处理器, 执行通用处理器指令集中指定的指令, 具有立即可以使用的编译器, 能够用高级语言编程。
- **输入输出(IO)端口(Input Output port):** 系统从这些端口获得输入或者发送输出。通过这些端口, 将触摸屏、外部设备和外部系统与系统连接起来。
- **中断控制器(处理器)(Interrupt Controller/Handler):** 一个单元, 控制(处理)来自中断源的中断引起的处理器操作。
- **内核(Kernel):** 一个具有如下功能的 OS 程序组件——存储器分配和解除分配、任务调度、进程间通信, 以及通过使用信号、异常(错误)处理信号、信号量、队列、邮箱、管道和套接字有效管理共享存储器的访问、I/O 管理、文件系统、中断控制(处理器)、设备驱动程序和设备管理。
- **掩膜和 ROM 掩膜(Mask and ROM Mask):** 在工厂中为了制作芯片而创建。ROM 掩膜是从 ROM 映像文件中生成的。
- **存储器(Memory):** 它存储了所有的程序、输入数据和输出数据。处理器从存储器中取出指令运行, 并在每条指令运行完毕后将结果传送给它。
- **记忆棒(Memory Stick):** 记忆棒(卡)是存储视频、图像、歌曲和语音的存储单元, 它可用作数码相机和移动系统中的大容量存储器。例如, 索尼的 Micro 记忆棒(M2)尺寸为 $15 \times 12.5 \times 1.2 \text{ mm}^2$, 却有 2GB 的闪存空间。
- **中间件(Middleware):** 这是个软件, 允许从一个系统输入, 再输出到另一个系统中。
- **多路器(Multiplexer):** 一个有大量数字输入的数字电路, 其输入来自多条通道, 把输出发送给一条选择的通道。从输入到输出的通道是通过通道地址位来确定的。该地址位在多路器的输入上应用。多路器从多条输入通道中选择一条通道的输入, 将数据发送给输出通道。
- **LCD:** 液晶显示, 在 3V~4V、50Hz 或者 60Hz 的操作电压脉冲、电流最大为 50 μA 的条件下吸收或发出光。多段和多线 LCD 单元用来高亮显示数字、字符、图表和短信息, 功耗很低。

- **LED:** 发光二极管, 发出电压在 1.6V~2V 之间、电流为 8mA~15mA 之间的红色、绿色、黄色或红外线光的二极管。多段和多线 LED 单元用来高亮显示数字、字符、图表和短信息。
- **链接器(Linker):** 一个将编译过的代码与其他代码链接起来, 并为加载器或定位器提供输入的程序。
- **加载器(Loader):** 一个程序, 重新分配用于加载到系统 RAM 存储器中的物理存储器地址。重新分配是有必要的, 因为在计算机处理过程的给定瞬间, 可用的存储器地址可能并不是从 0x0000 开始的。加载器是计算机中 OS 的一部分。
- **定位器(Locator):** 在 ROM 存储器的实际地址中重新分配应用程序已链接的文件和 RTOS 代码的程序。它创建一个标准格式(称为 hex 格式)的文件。这个文件称为 ROM 映像。
- **微控制器(Microcontroller):** 具有处理器的单元。根据应用的需求, 提供存储器、定时器、watchdog 定时器、中断控制器、ADC 或 PWM 等。
- **调制解调器(Modem):** 将送出的比特调制为电话线中常用的脉冲, 将输入的脉冲解调为输入信息的比特。
- **多路器(Multiplexer):** 一个数字电路, 其数字输入来自多个通道。多路器一次仅发送一个通道输出。输出中的通道地址位与输入中的通道地址位相同。
- **多任务调度(Multitasking):** 由调度器指导, 处理不同任务的代码。
- **物理设备(Physical Device):** 连接到系统端口的设备, 如打印机或键盘。
- **管道(Pipe):** 一种虚拟设备, 发送来自一个数据源(如程序结构)的字节流, 并将数据传递给数据目的端(如打印机)。
- **进程(Process):** 一个程序、任务或线程, 具有独立内存分配, 具有针对特定工作的一个或多个函数或过程。进程可以与其他任务共享内存(数据)。在 OS 指导下, 一个处理器可以分别或并行执行多个进程。
- **处理器(Processor):** 一个电路, 执行指令周期, 指令周期实现一个过程或一组过程, 或者执行程序指令。
- **PWM(Pulse Width Modulator, 脉宽调制器):** 根据输入位调制脉宽的调制器, 提供脉宽与需要的模拟输出成比例的脉冲。结合了 PWM 输出后, 就完成了 DAC 操作。
- **实时操作系统(real-time operating system):** 用于实时编程和调度、进程和存储器管理、设备驱动程序、设备管理和多任务调度的操作系统软件。
- **RAM:** 是一个随机访问读写存储器, 处理器用来存储可变的程序和数据, 断电后存储器中的数据将消失。
- **寄存器(Register):** 与处理器相关, 临时存储来自存储器以及在指令处理过程中执行单元的变量值。
- **复位(Reset):** 是一个处理器状态, 在这个状态中处理器寄存器获得初始值, 并且从这个状态开始执行一个初始化程序。这个程序也通常在加电时运行。
- **复位电路(Reset Circuit):** 一个强制进入复位状态的电路, 从加电的一个短周期内激活。当复位被激活后, 处理器生成一个复位信号, 提供给其他需要复位的系统单元。这样可以对接受信号的单元复位。

- **ROM**: 一个只读存储器, 放置了下列数据—— 嵌入式软件、初始数据、字符串以及操作系统或 RTOS。
- **手写笔(Stylus)**: 是一个细铅笔状的物体, 可以拿在手上像笔一样使用。屏幕显示选项或命令、菜单、对话框和图标, 就像计算机视频显示器一样。
- **系统(System)**: 系统是一种按照固定的计划、程序或规则来处理、组织或执行一项或多项任务的方式。
- **片上系统(System on Chip, SoC)**: VVLSI 或 VLSI 芯片上的一个系统, 具有所有需要的数字电路和模拟电路, 如在移动电话中。
- **磁贴(Tile)**: 触摸屏上的一个矩形, 就像桌面屏幕中的图标。触摸磁贴会在系统上执行一个命令。
- **定时器(Timer)**: 该设备在后续的时间间隔提供超时脉冲, 在编写好的时间间隔中断系统。在预设时间到期、溢出、当前时间等于(或超过)预设时间以及捕捉到事件时, 定时器会产生中断。它提供了系统时钟和实时操作、调度时间。
- **触摸屏(Touch Screen)**: 触摸屏既是一个输入设备, 又是一个输出设备。它可以用来输入命令、选择菜单或进行应答。使用手指或手写笔通过物理接触屏幕上的位置执行输入操作。
- **UART**: 通用异步收发器。
- **虚拟设备(Virtual Device)**: 一个通过编程可以像操作物理设备一样打开、关闭、读和写的文件, 或者类似于管道的设备。例如连接和断开物理设备的程序, 以及输入输出程序。
- **VLSI 芯片(VLSI Chip)**: 制作在硅片上的最多为 100 万晶体管的大规模集成电路。
- **VVLSI 芯片(VVLSI Chip)**: 制作在硅片上的最多为 1000 万晶体管的大规模集成电路。
- **watchdog 定时器**: 当程序挂起超过一定时间后, 该定时器定时结束, 并将处理器复位。

复习题

- (1) 分别对系统和嵌入式系统给出定义。
- (2) 下列器件的基本结构单元是什么? 请列举出来。(a)微处理器, (b)嵌入式处理器, (c)微控制器。
- (3) 微控制器与通用处理器(GPP)的主要区别是什么?
- (4) (a)只有定点算术单元的处理器和(b)一个附加了浮点算术处理单元的处理器的优缺点各是什么?
- (5) 微控制器与 DSP 的区别是什么?
- (6) 解释基于收敛技术的嵌入式系统中专用处理器的使用, 例如具有邮件客户端、Internet 连接和图像画面下载功能的智能移动电话。
- (7) 为什么新一代系统能在低电压下($\leq 2V$)操作处理器和 IO(大约为 3.3V)?
- (8) 处理器复位和系统复位的作用是什么?
- (9) 解释为什么需要 watchdog 定时器和监视时间结束后要复位。

- (10) 嵌入式系统中 RAM 的作用是什么？
- (11) 为什么对于嵌入式系统的设备，我们需要多种行为和多种控制任务？用一个嵌入式系统示例——移动电话——来加以解释。
- (12) 何时需要多任务调度 OS？
- (13) 何时需要 RTOS？
- (14) 实时视频处理需要硬实时限制的复杂嵌入式系统。为什么？请加以解释。
- (15) 为什么处理器系统总是需要“中断处理器(中断控制器)”？
- (16) 为什么在嵌入式系统中程序存在于 ROM 中？给出 ROM 映像的定义，并解释一个系统中 ROM 映像的每个部分。
- (17) 链接器所起的作用是什么？为什么在计算机系统中使用加载器，在嵌入式系统中使用定位器？
- (18) 下列术语的含义是什么：中断、系统时钟、实时时钟、外围设备、总线协议。
- (19) 绘制巧克力自动售卖机所需软硬件单元的体系结构。
- (20) 绘制智能卡所需软硬件单元的体系结构。

实践练习题

- (1) 从附录 B “参考文献”中列出的书籍中寻找嵌入式系统的定义，并将定义和参考书列表成表格，定义放在第一列，参考书放在第二列。
- (2) 将嵌入式系统分类为小型、中型和复杂系统。现在，再将嵌入式系统重新分类为具有和没有实时(响应时间受限)要求的系统，并对每一类给出 10 个示例。
- (3) 列出各种类型的存储器以及它们在下列系统中的应用：机器人、电子智能重量显示系统、ECG LCD 显示及记录器、路由器、数码相机、语音处理、智能卡、嵌入式防火墙/路由器、邮件客户端卡、具有冲突控制和超时传输控制(jabber)的收发系统(冲突的意思是当网络中其他系统都不使用网络时的接收和发送。超时传输控制的意思是控制网络中连续的随机数据流，这些数据流最终会阻塞网络)。
- (4) 以表格的形式列出下列每个系统所需的硬件单元：相机、便携式计算机和机器人。
- (5) 给出嵌入式系统的两个示例，它们需要下列一个或多个单元：(a)DAC, (b)ADC, (c)LCD 显示, (d)LED 显示, (e)小键盘, (f)脉冲拨号电路, (g)调制解调器, (h)收发器, (i)触摸屏用户界面。
- (6) 有一个 10 位的 ADC。它的参考电压为 $V_{\text{ref-}}=0.0\text{V}$, $V_{\text{ref+}}=1.023\text{V}$ 。当输入为(a) -0.512V , (b) $+0.512\text{V}$, (c) $+2.047\text{V}$ 时，ADC 的输出分别是什么？在下列三种情况下，ADC 的输出分别是什么：(a) $V_{\text{ref-}}=0.512\text{V}$, $V_{\text{ref+}}=1.023\text{V}$; (b) $V_{\text{ref-}}=1.024\text{V}$, $V_{\text{ref+}}=2.047\text{V}$; (c) $V_{\text{ref-}}=-1.024\text{V}$, $V_{\text{ref+}}=+2.047\text{V}$ 。
- (7) 以表格形式列出使用下列编码语言的优缺点：(a)机器码；(b)汇编语言；(c)C；(d)C++；(e)visual basic 或 C++；(f)Java；(g)Linux。
- (8) “设计嵌入式系统的成本可能是设计处理器和硬件单元的数千倍”，解释这句话。