

第1章 嵌入式系统概况

1.1 什么是嵌入式系统

关于嵌入式系统的定义可以说是众说纷纭，IEEE 给出的定义是：“嵌入式系统是用来控制、监控、或者辅助操作机器、装置、工厂等大规模系统的设备。”而维基百科给出的定义则是：“所谓嵌入式系统是指完全嵌入受控器件内部，为特定应用而设计的专用计算机系统。”国内学术界和工业界普遍接受的定义则是：“嵌入式系统是指以应用为中心，以计算机技术为基础，软件硬件可剪裁，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。”

总而言之，与巨型机、服务器、工作站和 PC 等通用计算平台不同，嵌入式系统是为特定应用设计的专用计算机系统。所谓通用计算平台，是指计算机的功能主要取决于所运行的软件系统，不同的软件决定了通用计算机系统的功能，比如在服务器上运行 Web 服务器，那么这台服务器的主要功能就是作为网页服务器，而如果运行的是打印服务，那么该服务器的主要功能就是打印服务器。嵌入式设备则通常具有一个非常特定的应用功能，比如即使在今天智能手机的功能已经非常强大，可以集成大量的应用，但是其作为电话的功能却必须是首先实现的。所以从最广义的角度上来看，只要是专用计算机系统都可以称为是嵌入式系统，甚至有种开玩笑似的定义，所谓嵌入式系统就是 PC 取反，所有非 PC 类的计算机系统都是嵌入式系统。甚至 PC 也是由嵌入式系统构成的，比如键盘控制器、硬盘的控制器构成的专用控制系统等等。

从嵌入式系统的定义我们可以知道嵌入式的应用其实无所不在，现代生活每天都在和嵌入式系统打交道。据 ARM 公司的预测，2013 年全球出货的消费电子类产品将达到 24 亿部（其中手机 14 亿部），而围绕工业控制、汽车电子、网络通信等传统嵌入式设备应用，2013 年全球出货的嵌入式微处理器将达到 110 亿颗。可以不夸张地说，嵌入式应用和相关技术支撑了全球电子信息类产业的大半壁江山。

与通用计算平台不同，嵌入式应用往往更加强调系统的高性能、低成本、低功耗以及实时性、可靠性等设计因素。尤其是对于消费电子这类面广量大，强调用户体验，并且采用电池供电的设备而言，系统的性能、成本以及功耗是必须考虑的优化目标。然而，高性能往往意味着高成本和高功耗，如何在性能、成本与功耗间找到折中是这类产品设计过程中必须面对的挑战。本书也试图重点介绍这三个设计维度的考量、优化与折中。

值得一提的是，传统上归入嵌入式系统设备的移动电话等消费电子产品，由于采用更加强大的处理器以及相关硬件和操作系统，在这些设备上集成的功能也越来越丰富，比如智能手机和平板电脑已经可以实现上网浏览、处理邮件、办公软件、游戏、高清视频、高保真音乐等功能，使得这类产品已经越来越具备通用计算平台的特征，也就是设备的功能取决于其上所安装的软件系统。在这类产品上嵌入式专用计算平台与通用计算平台的界限正在变得模糊。作者认为这种通用化的趋势会变得越来越明晰，智能手机等传统嵌入式设备会逐渐变为通用计算平台，这个通用化的过程其实也正是移动互联网兴起和普及的过程，而与此相应的是手机专用外设和众多以手机为终端的软件商（当然，也可能是服务商和内容提供商）的兴起，这个过程将像当年 PC 的兴起一样充满创新、机遇和激情。

1.2 嵌入式系统的应用与分类

从不同的分类标准我们可以对嵌入式系统应用进行不同的分类，本节将从实时性要求和应用领域两个维度对嵌入式应用进行分类，虽然任何分类方法都很难准确地划分纷繁复杂的不同应用和产品形态。另一方面，由于移动互联网和物联网的兴起，这两大领域的应用已经深刻改变了传统嵌入式系统应用的产品形态和商业模式，因此我们将用两小节的篇幅单独介绍。

1.2.1 基于实时性的分类

所谓实时性是指系统运行的正确性不仅仅取决于功能的正确完成，同时还取决于在规定的时间内完成该功能。按照系统对于实时性要求的严格程度，我们可以简单地将嵌入式系统划分为非实时系统、软实时系统和硬实时系统。

非实时系统。在非实时系统中，系统的功能正确性仅仅取决于功能是否正确执行，而与功能执行的时间无关。简单地说，比如我们在手机或平板电脑上打开一个 Word 文档并进行编辑，这个功能的正确性仅和文档是否正确打开、是否能够正常编辑并保存有关，而与打开文档所耗费的时间无关（打开文档耗时长短仅仅影响用户体验，而不影响功能）。

软实时系统。软实时系统的功能正确性不仅与功能执行有关，而且该功能必须在规定的时间内完成，否则将造成系统的功能不正常或故障，虽然这种不正常或故障并不会引起崩溃性和灾难性的后果。移动电话的语音编解码系统就是一个典型的软实时系统，电话的编解码系统必须在规定的时间内完成语音的采样、编码并在规定的时间内封装成为可传输的通信帧进行传输。如果系统不能在规定的时间内完成此项工作，就有可能造成通话质量的下降或停顿。另外，移动智能终端上的音乐播放软件和视频播放软件都必须在规定的时间内完成音频或视频文件的解码，否则将造成音乐或视频播放的卡顿。

硬实时系统。硬实时系统要求系统必须在规定的时间内完成规定的功能，否则将造成崩溃性的后果。火箭的控制系统可能是硬实时系统的最好例子，如果控制系统不能在规定的时间内完成对各路传感器传回数据的分析并做出控制响应的话，整个火箭系统将可能出现不可逆的灾难后果。硬实时系统的另外一个比较直观的例子是汽车的安全气囊。

关于嵌入式系统的实时性设计与分析是学术界和工业界研究的一个热门领域，随着系统复杂性的不断增加，如何保证任何情况下都能在规定的时间内做出正确的响应是一个巨大的挑战。

1.2.2 基于应用的分类

正如上一节所述，嵌入式系统的应用领域几乎涵盖了所有非 PC 的计算机系统，因此从应用领域对嵌入式系统进行分类是一件困难的事。通常情况下，我们将嵌入式应用领域划分为以下几大类。

消费电子类产品：毫无疑问，消费电子类产品是嵌入式应用领域最为面大量广、产品形态各异、竞争异常激烈的应用领域。我们可以将这个应用领域进一步细分为个人信息终端产品、办公自动化产品和家用电器类产品。个人信息终端应用包括手机、平板电脑、数码相机、数码摄像机、移动媒体播放器、个人游戏终端等。办公自动化类产品包括打印机、复印机、传真机等。家用电器包括电视机（含互联网电视）、家庭影院系统、机顶盒、冰箱、洗衣机等。

网络通信类产品：网络通信类产品构建了整个信息网络的基础，主要包括交换机、接入设备、路由器、防火墙、VPN 设备等等。

汽车电子类产品：汽车电子领域是嵌入式系统的传统应用领域，主要包括汽车的引擎控制系统、安全系统（防抱死系统、安全气囊）、车载导航系统和娱乐系统等等。由于车载环境比较恶劣（温度、

震动、灰尘等), 车载系统往往对设备的高可靠性、稳定性有着严格的要求。

工业控制类产品: 工控类应用领域也是一个非常宽泛的应用领域, 主要包括工控 PC、程控机床、智能仪表、生产线控制等等。另外, 我们有时也将交互式终端类产品归入工业控制类产品, 这类产品包括各种类型的非 PC 类的网络终端, 比如税控收款机、POS 机具、各种信息查询终端等等。

医疗电子类产品: 医疗电子类产品包括传统的医疗设备和医疗信息化所需要的各类设备。前者包括大型的 CT 机、核磁共振扫描, 也包括小型的生命体征监护仪、呼吸机、血压计等等。医疗信息化类产品涉及的面也非常广泛, 包括药品物流所需要的各种查询终端、病人住院信息查询终端、医生的电子病历等等。

军工及航天类产品: 军工和航天类产品通常情况下是作为武器系统或航天器系统的相关控制系统、导航系统等等, 涉及的面也非常广泛, 通常情况下这类产品都是硬实时系统, 而且对于系统的稳定性、健壮性有着非常高的要求。

1.2.3 移动互联网

移动互联网是一个庞大的系统, 包括移动智能终端(又可以分为智能手机、平板电脑等设备)、互联网络、网络服务(社交网络、游戏、及时通讯)等内容。传统上, 移动智能终端被划入嵌入式设备领域, 因为早期的手机采用嵌入式微处理器, 运行嵌入式操作系统, 其应用也比较单纯。但是现代的智能手机已经配备了功能强大的多核微处理器和外设, 并运行开放的手机操作系统, 用户可以根据自己的喜好选择安装不同的手机应用软件, 从这个角度上来说, 智能手机已经越来越像一个通用计算平台, 越来越多的公司和个人可以为流行的智能手机平台编写应用软件, 与此同时越来越多的能够和智能手机协同工作的硬件设备也层出不穷。因此, 部分研究者认为移动互联网终端, 尤其是运行开放操作系统的智能终端不能被划分为嵌入式系统产品, 而是应该作为一类独立的计算机类型。我们认为, 移动互联网终端依然保持了很多嵌入式系统的特征, 比如相比于桌面系统和服务器系统而言, 移动互联网终端的硬件系统依然需要尽可能采用低成本和低功耗的设计, 这使得虽然目前的移动互联网终端的硬件系统的功能已经越来越强大, 但总体上依然较同时代的桌面系统要低一个层次。也正因为如此, 对于移动互联网终端的底层软件与硬件的适配与优化就显得更为重要。另一方面, 虽然移动互联网终端已经出现向通用计算平台过渡的趋势, 但最基本的专用功能(比如智能手机的电话功能)依然是至关重要的。基于这两点考虑, 本书依然将移动互联网终端在广义上归为嵌入式系统。

随着互联网技术以及移动智能终端技术的发展, 互联网已经渗透到现代社会的方方面面。据统计, 2009 年全球互联网的渗透率达到了全球人口的 23.8%, 而到 2015 年这个数字将达到 50%。作为互联网的终端设备, 2009 年共有 14 亿台电脑和 47 亿部移动电话, 并且越来越多的用户通过移动智能终端来访问互联网服务。而按照 ARM 公司的统计, 仅 2013 年一年, 全球出货的消费类移动设备的总出货量将达到 24 亿部, 其中仅移动电话一项就占 14 亿部。对于新一代移动互联网终端的产品特征, ARM 公司 CEO Tudor Brown 在 2012 曾预测至少应该包含以下内容(大部分功能在今天已经成为现实): 强大的无线互联网接入能力, 4~7 英寸的 LCD 显示屏, 支持随时在线, 支持常用的办公软件(比如邮件、字处理、电子表格、PPT 和 PDF 等), 支持高清视频、3D 游戏和 Hifi 音乐, 具有 8 小时以上的电池续航时间。

总的来说, 为了支撑这些产品的设计需求, 作为核心的移动终端嵌入式微处理器需要具备高集成、高性能、低能耗的特征。移动设备往往对产品的体积有着严格的限制, 因此为了在有限的空间内支持如此丰富的功能, 移动终端处理器必须将尽可能多的功能模块集成在一颗芯片中。比如, 除了主 CPU 外, 其还必须集成 GPU(图形处理器)、VPU(视频处理单元)、各类存储器接口(DDR, Nand Flash, SD 卡等)、各类通信模块与接口(3G/4G 移动通信, Wifi, USB OTG, 蓝牙, GPS 等)、高清显示控

制与接口（高清 LCDC，HDMI 接口）。如何将如此多的功能模块集成在一颗芯片，并保证每个功能模块的正常工作，已经成为现代 SoC 设计领域的一个重大挑战。另一方面，在保证处理器高集成度：高性能的同时，设计者还必须保证整个系统的低能耗。移动设备往往采用电池供电，降低系统能耗不仅能够延长电池在充电间隔的使用时间，同时也可以降低由于发热而带来的散热问题（毕竟，谁都不想要自己的智能电话需要配备散热风扇）。遗憾的是，系统对于高集成度与高性能的要求与低能耗的要求往往是一个矛盾，如何解决“既要马儿跑，又要马儿不吃草”的问题，成为移动智能终端处理器设计的另一个设计挑战。

1.2.4 物联网

物联网（The Internet of Things）是“万物沟通”的、具有全面感知、可靠传送、智能处理特征的连接物理世界的网络，实现了任何时间、任何地点及任何物体的连结。Kevin Ashton 最早在 1999 年提出了这个术语，虽然当时这个术语更多地用于供应链管理这个领域。今天，物联网可以帮助实现人类社会与物理世界的有机结合，使人类可以以更加精细和动态的方式管理生产和生活，从而提高整个社会的信息化能力。如果说传统的互联网是将全世界的计算机连接在一起的网络，而移动互联网则进一步将人们获取信息、处理信息、发布信息的终端从桌面计算机延伸到移动设备（包括智能手机、平板电脑、可穿戴设备等）的话，物联网则是将信息的获取和处理进一步延伸到几乎所有的物理设备（被称为智能对象，Smart Objects）。物联网的诞生将人类信息网络的神经末梢延伸到了几乎所有领域，包括农业、工业、交通、环境监测与保护、远程医疗、智能家居等。

物联网并不是独立于传统互联网的一个新兴网络，从某种意义上来说它实际构建在传统互联网之上。一般来说我们将物联网分为 3 层：第一层是物理感知层；第二层是网络传输层；第三层是应用层（见图 1-1）。物理感知层是物联网的最底层，负责采集物理世界的相关信息，并对来自上层应用的命令进行响应。采集物理世界的相关信息需要各种传感器来完成，比如温度传感器、光照传感器、摄像头、RFID 等等；为了响应上层应用发出的命令，有时还必须在该层配备相应的执行器（Actuators）。因此，物理感知层通常需要通过无线传感器网络（WSN，Wireless Sensor Network）或者传感器/执行器网络（SANET，Sensor/Actor Networks）来实现。可以说，物理感知层实际上是物联网的最本质特征之一，因为在此之前互联网的终端以桌面计算机和移动智能终端为主，而物联网的物理感知层将各种智能对象连接到互联网。所谓智能对象，可以将其定义为：

- 具有物理特性（大小、形状等）的实体；
- 具备最基本的通信功能，比如可以被其他通信实体发现，并可接受输入的信息和发出相应的回应；
- 拥有唯一的 ID 标识；
- 拥有至少一个名字和一个地址，名字用于人类识别，而地址用于机器间的通信；
- 具备一定的信息处理能力；
- 可以感知物理世界的相关信息（Sensor），或者可以对物理世界施加影响（Actuator）。

一个无线传感器网络通常由大量随机部署的传感器节点设备和一个或少数几个作为连接互联网的网关的汇总节点（Sink）构成。传感器节点只具有非常受限的计算能力、通信能力和电池容量。在网络拓扑结构上，由于受限于传感器节点的电池容量、处理能力以及部署方式，无线传感器网络通常采用 ad hoc^①网络的形式，也就是不需要预先布设的交换节点和路由节点，所有的通信节点在网络中地位相等，每个节点既要发送自己的信息也要承担转发来自其他节点信息的功能（Sink 节点除外，因为该类节点担负着将来自整个传感器网络的信息汇总传输到互联网的功能，因此该节点具有网关的作用）。

^①“ad hoc”这个词来自拉丁语，意为“for this”。

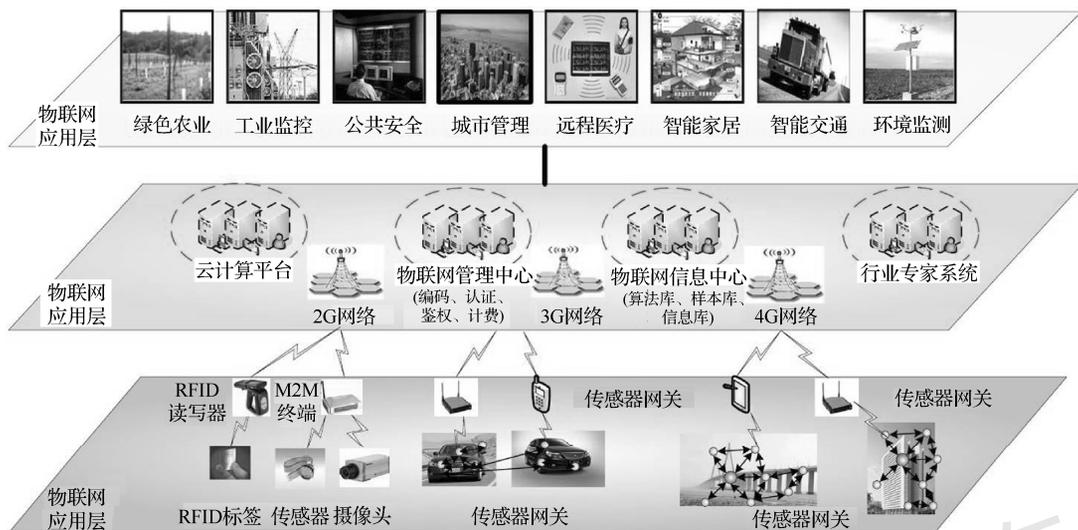


图 1-1 物联网

物联网的第二层是网络传输层。事实上，这里的网络层就是指传统的互联网，借助感知层的网关节点，物联网将底层采集的物理世界信息汇入互联网（也就是运行 TCP/IP 协议的全球网络）。互联网为这些信息提供传输通道和安全控制的同时，也为物联网应用提供了云存储平台和云计算平台。

物联网的最上层是应用层。如果说感知层和网络层提供机制保障的话，应用层才是真正的策略层。比如，微型传感器节点布置在农田当中，动态自组织成网络，采集农田作物的实时信息，如温度、光照度、土壤酸碱度等，并将采集的数据通过无线传感器网络返回到监控中心，监控中心将参照农作物种植的最优化环境数据，采取相应举措，如启动灌溉设备等。再如，如今的工业厂房规模日益增大，厂房车间等的实时环境成为影响安全生产和产品质量的重要因素。单靠人力去监测厂房设备及环境信息会耗费大量的资源。如果在广大的厂房空间中布置好一定数量的传感器节点，这些节点就能组成网络实时采集设备及环境信息，如温度、压力、烟雾浓度、有害气体含量等，并将信息实时反馈到监控中心。

从系统的层面来看，物联网应该具备以下特征：

- 支持异构设备。物联网中包含了大量的异构智能对象，这些对象在计算能力和通信能力上都具有很大的不同。为了管理这些异构设备物联网，应该在体系结构和协议两个层面都提供支持；
- 通过近距离无线通信技术实现无所不在的数据交换；
- 低功耗解决方案。由于大量的传感器接点是通过电池或者其他非传统电源方式（比如被动 RFID 采用感应的方式获取能量）供电的，所以低功耗技术和能量获取技术对于物联网具有举足轻重的作用；
- 位置与跟踪能力。通过短距离无线通信技术，物联网中的通信节点可以被定位（甚至跟踪移动轨迹），这个特性对于物流和产品生命周期管理等应用（这类应用已广泛采用 RFID 技术）具有非常重要的作用；
- 自组织能力。一方面由于传感器节点布置的随机性，另一方面由于这些节点在计算能力、通信能力和电池容量上的限制，传感器网络通常采用 ad hoc 网络的形式进行组织，这就要求系统中的各个节点能够自组织起网络的通信方式，比如动态的路由选择；
- 内建的安全与隐私保护机制。因为物联网与物理世界的紧密关联，因此必须从系统的最底层保证安全性和隐私。

通过以上分析，我们可以看出在整个物联网系统中真正与嵌入式系统相关的部分主要集中在物理感知层，主要包括嵌入式微控制器及相关软件、短距离无线通信、传感器等，而其他两层则更多地与传统的计算机科学相关。

1.3 嵌入式系统的产业链

嵌入式系统产业涉及现代电子信息产业的方方面面，几乎涵盖了电子信息产业的所有内容，构成了一个纷繁复杂的产业生态环境，形成了错综复杂的产业生态链。从大的方面进行划分，这个生态环境可以分为芯片设计与制造、方案设计与软件、整机设计与生产以及运营与服务这几个方面（如图 1-2 所示）。

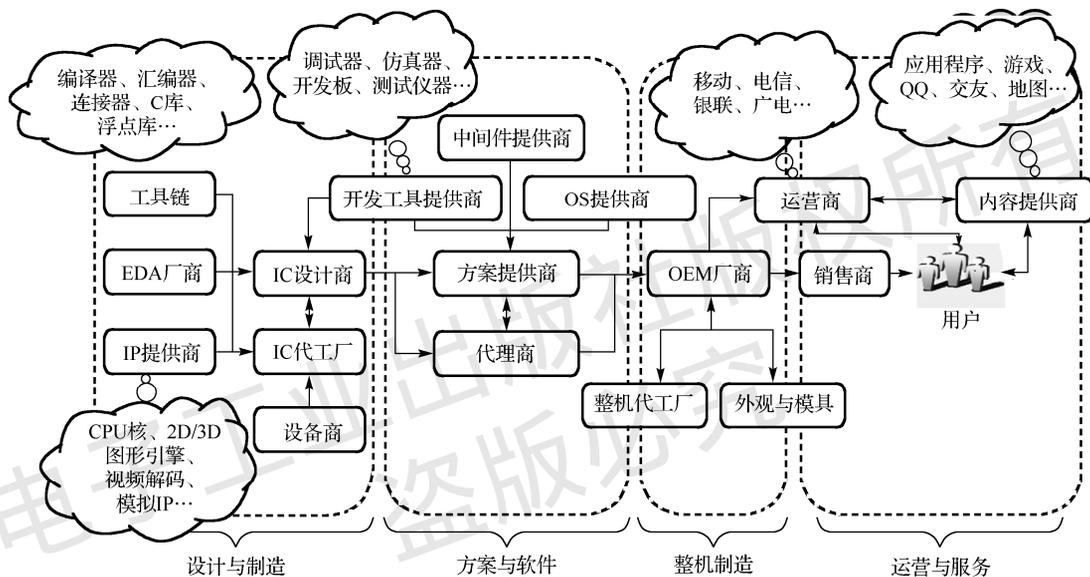


图 1-2 嵌入式系统的产业链

芯片设计与制造又可以分为 IC 设计商（Fabless）、IC 代工厂（包括 Foundry、封测厂等）、IP 提供商（IP Vendor）等。当然国际半导体巨头厂商中有仅负责 IC 设计的纯 Fabless 厂商，比如 Mavell、Broadcom 等；也有整合设计与生产封测于一体的 IDM（集成设备制造商，Integrated Device Manufacture）厂商，比如 Intel、Samsung、ST、NXP、FreeScale 等；还包括仅负责生产或封测的厂商，比如 TSMC（台积电）、SMIC（中芯国际）、日月光等。IP 提供商中有只提供知识产权 IP 核的公司（如 ARM），也有除提供 IP 外同时推出自己芯片的厂商（如高通等公司）。当然，除了前述的几类厂商外，IC 代工厂还需要向相应的设备商购买用于生产、测试、封装的专用设备。IC 设计商也需要在采购所需 IP 外向相应的 EDA（电子设计自动化，Electronic Design Automation）厂商购买用于设计的 EDA 工具，比如 EDA 产业的两大巨头 Synopsys 和 Cadence，当然这些 EDA 公司有时不仅销售 EDA 设计工具，同时也销售自己的 IP 和提供设计服务（Design Service）。

芯片设计和制造出来以后还必须由相应的方案提供商（Design House）基于这些芯片开发相应的方案设计。当然，随着竞争的日益激烈，越来越多的芯片提供商在推出新芯片的同时也会给出相应的非常完善的参考设计（这部分工作有时由芯片提供商自己提供，有时也会委托第三方的方案商进行设计）。另外，通常情况下，芯片提供商会委托芯片代理商进行芯片的推广工作，由芯片代理商直接面对

芯片用户并提供必要的技术支持。为了设计需要，方案提供商往往还会采购第三方的相关工具和软件，比如开发工具提供商、操作系统提供商（比如风河公司、微软公司、Mona Vista 公司）和嵌入式软件中间件提供商（比如媒体解码库、地图导航软件、嵌入式数据库）等。

芯片方案商所提供的解决方案最终被应用于某个整机厂商（OEM 商）的整机产品设计，通常情况下现有的 OEM 厂商仅负责产品的策划、外观设计（最后产品的模具设计一般委托给第三方的模具设计厂商）、功能定义、品牌包装、市场推广等工作，具体的整机产品生产都外包给专门的整机代工厂负责代工，OEM 厂商可能只负责整机的组装和测试。

最终的整机产品将通过相应的销售商渠道推向市场。当然，随着移动互联网的兴起，越来越多的嵌入式整机产品（尤其是消费类电子产品）还需要订购通信运营商（比如中国移动、中国电信、中国联通）和内容服务商（比如腾讯、新浪、京东等）所提供的通信服务和内容服务。

在整个嵌入式系统的产业链分工中，一方面各个厂商的定位与分工越来越细、越来越专业，另一方面产业链上的分工也在出现逐渐融合的趋势。比如，原来专门研发嵌入式 CPU IP 的 ARM 公司越来越关注最终的用户体验，并与众多整机厂商进行合作以保持其 CPU 的架构创新能够满足最终用户的使用体验。再比如很多传统的整机厂商也开始涉足运营和内容服务，并采用全新的营销模式，这其中的代表包括国产小米手机。另外，随着芯片提供商开始为 OEM 厂商提供完整的整机解决方案（Turn Key 方案），进一步压缩了传统方案提供商的市场空间。在这种融合的趋势中做得最彻底的可能是苹果公司了，为了给用户提供更好的用户体验，苹果的 iPhone、iPad 产品不仅采用苹果自主设计的处理器芯片，自己设计所有的硬件系统，采用苹果自己的 iOS 操作系统，苹果还通过 AppStore 和 iTunes 等网络平台为用户提供应用软件销售、数码音乐销售、电子书销售等在线服务（当然，苹果将产业链中利润较薄的整机生产环节外包给了富士康这样的整机代工厂）。

1.4 嵌入式系统的知识体系

嵌入式系统的本质是专用计算机系统，因此从知识体系上来看嵌入式系统相关技术的知识体系与计算机和电子信息类专业的内容是一致的。就一个具体的应用来看，在应用和物理世界之间存在着巨大差距，为了解决应用的需要，我们需要将应用的问题划分成若干个层次进行解决。一般而言，这些层次包括：器件、电路、寄存器传输级、微架构、指令架构、操作系统、编程语言、算法和最终的应用（如图 1-3 所示）。

所有的嵌入式系统最终都是由数量庞大的电子器件组成的，这些器件最主要的是以 CMOS 工艺为代表的 MOS 晶体管，在器件层面需要研究和学习的内容包括器件原理、器件结构、器件的制造工艺等内容。

器件组成了电路，电路是实现基本逻辑功能的基本单元，在电路层面需要学习的主要知识包括电路分析、电路与系统、数字电路和模拟电路等。

所谓寄存器传输级（RTL, Register Transfer Level）是指以寄存器或门电路为单位对整个电路系统的描述。寄存器传输级进一步对一个计算机系统的逻辑电路进行了抽象，使得设计人员可以以逻辑门和触发器为单位对系统进行设计，在现代数字集成电路的设计过程中，设计人员通常采用硬件描述语言（比如 VHDL 或 Verilog 语言）在寄存器传输级来描述数字电路的逻辑并进行仿真，在基本功能和时序正确后再通过综合（Synthesis）将其转化成为逻辑门和触发器的网表（Schematic，可以理解为基于逻辑门和触发器的电路图）。

微架构是指对寄存器传输级的进一步抽象，如果说寄存器传输级是对电路的抽象的话，微架构实际上已经是电路功能模组的进一步抽象。比如，CPU 内部的流水线设计、分支预测机制的实现、乱序执行机制和指令多发射机制等等。微架构实质上就是 CPU 或其他计算引擎的具体实现。

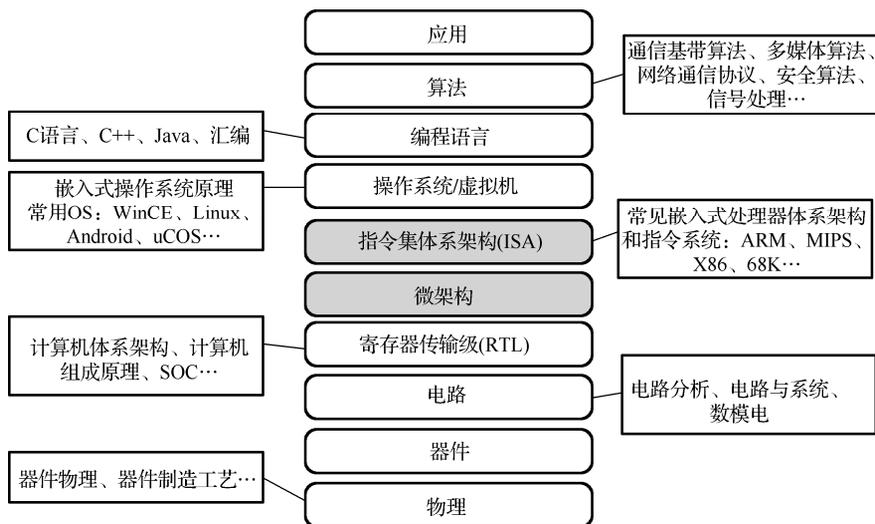


图 1-3 应用的层次

指令集体系架构（ISA, Instruction Set Architecture）是软件和硬件的分水岭，一般而言指令集体系架构之上的层次属于软件层面，指令集体系架构之下则属于硬件层面。指令集体系架构定义了某款 CPU 的指令集，而一款 CPU 是否能够满足某类特定应用的需求在很大程度上取决于指令集的设计。一般而言，指令集系统的设计分为两个流派，一是 CISC 架构，一是 RISC 架构。CISC 架构追求指令集的功能强大，让更多的复杂功能可以通过单条指令完成。但是由于 CISC 架构需要大量的额外硬件去实现使用频度非常低的部分指令，使得其成本、功耗的代价相对比较高，因此从 20 世纪 80 年代开始出现了 RISC（精简指令集）架构，该架构用相对简单得多的硬件实现功能简单的指令集，通过提高主频来弥补指令功能的欠缺。由于 RISC 架构在能效比方面能够取得更好的结果，因此现代嵌入式微处理器设计中采用 RISC 设计思想的实现占据了主流（即使像 Intel 这样传统的 CISC 处理器在其微架构设计上也大量借鉴了 RISC 的设计思想）。我们将在第 4 章详细讨论处理器的指令集体系架构。

指令集体系架构之上的层次属于软件层面，主要包括操作系统、编程语言、算法和应用。嵌入式操作系统已经成为现代嵌入式系统设计不可或缺的软件平台，与传统的桌面操作系统主要作为用户的使用平台不同，嵌入式操作系统则主要是作为设计人员的开发平台。目前常见的嵌入式操作系统主要包括：嵌入式 Linux、微软的 Windows CE 系列、谷歌的 Android、风河的 VxWorks、开源的 uCOS、ThreadX 等。早期的嵌入式编程语言主要是 C 语言和汇编语言，但随着嵌入式系统的功能越来越强大，嵌入式软件的规模和复杂度越来越大，C++、Java 等面向对象语言，甚至 XML、Python 等脚本语言也得到了越来越广泛的应用。算法是系统完成一个实际功能所需要采用的方法，在嵌入式系统中所采用的算法与系统的功能有关。比如，对于个人移动互联网终端类（如智能手机）的应用，系统必须包含数字语音处理与压缩算法、移动通信基带算法、多媒体编解码算法（含音频、视频、图像等）、安全认证算法、网络通信协议等内容。事实上，上述任何一个算法所涉及的内容都是一个专门的学科方向所研究的内容。

从上面的分析可以看出，嵌入式系统所涉及的知识几乎涵盖了整个电子信息类专业的所有内容。完全掌握这些知识，尤其是仅仅通过一门课程、一本教材的学习，如果说不是不可能的话也是极其困难的任务。好在对于大多数读者而言，至少在前期的学习实践过程中或多或少对于这些层次的相关内容进行了专门的学习，而本书的目的就是在读者具备相关知识的基础上，给出一个完整的嵌入式系统的框架和脉络，为读者的后续学习奠定基础。本书的主要内容将重点关注指令集体系架构、微架构和

寄存器传输级三个层次，使读者站在 SoC 设计者的视角学习嵌入式系统的构成。至于电路、器件和物理三个层次的内容，电子类专业的读者应该在先修课程中学习。编程语言、操作系统的相关内容本书会简单介绍，这部分的详细内容将在后续课程中进一步学习（对于计算机专业类的读者，这部分内容其实是先修课程的内容）。至于算法层面，正如前面所说，任何一个具体的算法实际上都是一门独立的学问，需要读者根据未来工作的需要有选择地进一步学习。

1.5 案例：MP3 播放器

本节将以一个 MP3 音乐播放器作为案例来讨论一个简单嵌入式系统的工作过程。我们假设该 MP3 播放器通过键盘进行操作，并且有一个彩色液晶屏作为显示输出。用户的音乐文件（MP3 文件）存放在系统的 Flash 存储器中，系统的 RAM 存储器采用外挂 DRAM 实现，系统的音频 Codec 通过 IIS 或者 AC97 接口与主芯片连接，解码方式为软解码，系统架构如图 1-4 所示。

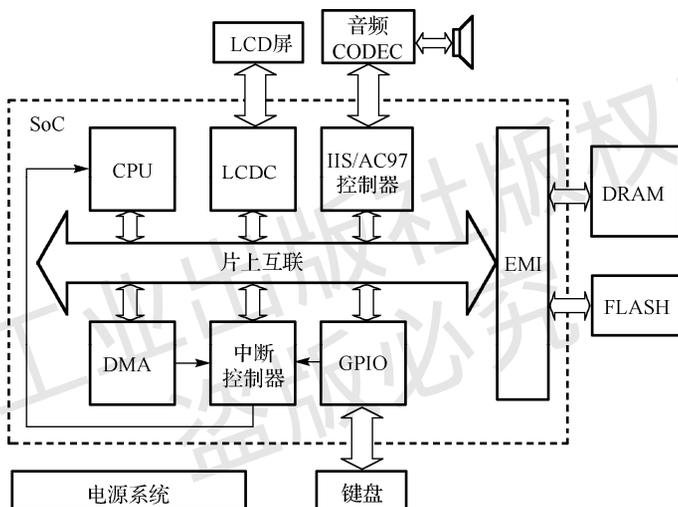


图 1-4 MP3 播放器系统架构

当然，在一个真实的 MP3 播放器实现中，系统可能高度集成，比如 Codec 和电源系统是集成在主芯片 SoC 中的，系统采用专用硬件解码而不是软解（这样可以获得更高的效率和更低的功耗），甚至键盘控制也专门集成了键盘控制器，而不需要通过 GPIO 进行管理。不过为了讨论的方便，我们采用图 1-4 的架构。

假设用户通过键盘操作选择了一首歌曲，并选择播放该文件，系统将如何完成这个任务呢？我们将按照系统处理的顺序介绍这个过程。

① 用户键盘操作最终体现为 GPIO 口上的一个中断信号，GPIO 向中断控制器提出中断请求，在中断控制器完成中断仲裁后，由中断控制器向 CPU 发出中断请求，CPU 在响应这个外部中断后，通过查询中断控制器中的相应寄存器可以获知本次中断由 GPIO 发起，系统进入 GPIO 的中断处理代码，发起对 GPIO 组成的键盘系统进行扫描以获得具体的键值，并以信号、事件或消息的形式通知操作系统。至此完成本次中断服务程序。

② 操作系统在获得键盘消息后，由内核调度相应的任务（线程）对该消息进行处理。通常这个任务是负责 MP3 音乐播放的主线程（将负责所有的用户操作响应、界面显示等）。主线程根据用户选择的文件，将通过文件系统读取将要播放的 MP3 文件。

③ 文件系统在接收到主线程给出的文件读取命令后将通过底层驱动程序访问物理的存储介质，通常情况下驱动程序将配置 DMA 控制器，启动其中的某个传输通道，DMA 在启动后将按照配置内容从外部存储器接口（EMI）读取 Flash 存储器中的文件内容到 DMA 通道的内部 FIFO，并将 FIFO 中的内容通过 EMI 传输到相应的 DRAM 缓冲区。由于受限于内部 FIFO 的大小限制，每次 FIFO 传输完成后，DMA 控制器将发起中断，通知 CPU 重新配置新的 DMA 参数继续完成后续的数据传输，直到文件的内容全部（或部分）传输到 DRAM 中的媒体数据缓冲区。

④ 当 DRAM 中的媒体数据缓冲区就绪后，操作系统将启动负责解码的任务（解码线程），解码线程由 CPU 从 DRAM 的缓冲区取出需要解码的数据并进行解码计算，并将解码后的数据写入 DRAM 的输出缓冲区。

⑤ 当 DRAM 中的输出缓冲区有数据后，操作系统将启动播放任务（播放线程），播放线程将配置 DMA 控制器的另一个数据传输通道，该通道从 DRAM 输出缓冲区中取出数据存放在 DMA 的 FIFO 中，并将 FIFO 中的数据传输给 IIS 控制器，该控制器将按照 IIS 标准的规定将解码后的音频数据传输给片外的 Codec 芯片，Codec 芯片将输出的 PCM 数据通过 D/A 转换为音频信号，并最终播放出来。这个过程和步骤③类似的是，DMA 控制器将不断地在一次 FIFO 数据空的情况下发起中断，CPU 重配 DMA 继续后续的数据传输。

⑥ 播放线程在播放的过程中将不断地通过操作系统提供的任务间通信机制，不断通知主线程播放的相关信息，主线程根据这些信息不断修改显示界面的内容（比如时间、歌词等）。主线程配置 LCDC 不断从 DRAM 中的显示缓冲区将数据刷新到 LCD 屏上，更新显示内容。另外，对于电池供电的设备而言，主线程可能还需要负责系统的低功耗控制。

通过上述分析不难发现，在 MP3 播放的过程中至少存在 5 条数据流：从 Flash 到 DRAM 的数据流，CPU 从 DRAM 取数据进行解码并将结果写回 DRAM 的数据流，从 DRAM 缓冲区到 IIS 控制器的数据流，LCDC 不断周期性地从 DRAM 取显示缓冲区数据的数据流，CPU 在这个过程中不断从 DRAM 中取指令的数据流。这 5 条数据流都需要通过 SoC 片上的片上互连网络和外部存储器接口（EMI）进行数据传输，因此 EMI 以及外部存储器的效率以及片上互连网络的效率直接决定了整个系统的效率。

另一点需要说明的是，从软件角度看，我们可以通过 3 个任务（或线程）协同工作的方式来解决 MP3 播放的任务。其中主线程负责维护用户界面和控制，解码线程负责具体的数据解码工作，而播放线程则负责将解码后的数据传输到具体的 IIS 控制器，三个线程间通过操作系统提供的任务间通信机制进行同步。其中解码线程与播放线程属于典型的生产者与消费者关系。当然，不同的实现可以采用不同的方案，甚至在简单系统中通过单任务也可以完成所有上述工作。

综上所述，即使像 MP3 播放器这样简单的系统也是一个由硬件和软件协同工作的系统，当然从产品设计的角度看，一款 MP3 产品的设计将更加复杂，包括产品的策划、市场、外观设计、生产、销售渠道和售后服务等，绝不仅仅是我们提到的技术实现部分的内容。

思考题

1. 在本章的案例中，LCDC 需要不断地从 DRAM 存储器中取出显存中的数据并将这些数据按照一定的协议传输到 LCD 屏，屏上集成的驱动芯片将按照 LCD 的时序将这些数据显示出来，假设屏的分辨率是 800×600 ，每个像素需要采用 24 位表示（RGB 三个分量各占一个字节），请计算每帧 LCD 数据的大小。假设为了使屏幕的显示不出现抖动，LCDC 需要每秒刷新显示 50 帧，请计算 LCDC 需要每秒从 DRAM 中取多少数据？

2. 在本章的案例中，音频数据按照 44.1K 的频率进行采样，每个采样点需要占用 16 位，按照左右两个声道进行采样，请计算 DMA 每秒需要向 IIS 控制器传输多少解码后的数据？这个过程需要占用多少片上互连的带宽？

3. 请说说能耗与功耗的关系? 通过降低系统的功耗是否一定能够降低系统的能耗? 电池使用时间的长短主要取决于系统的能耗还是系统的功耗?

扩展阅读

- [1] Barry, Peter and Patrick Crowley, Modern embedded computing: designing Connected, Pervasive, Media-Rich Systems[Book], ELSEVIER, 2012.
- [2] John L. Hennessy, David A. Patterson. 计算机体系结构量化研究方法(第 5 版). 贾洪峰译. 北京: 人民邮电出版社, 2013.
- [3] Hisa Ando. 支撑处理器的技术. 李剑译. 北京: 电子工业出版社, 2012.
- [4] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, ImrichChlamtac, Internet of things: Vision, applications and research challenges [J], Ad Hoc Networks 10 (2012) 1497-1516.

电子工业出版社版权所有
盗版必究