

Embedded Systems

2 - Development

嵌入式系统的定义

- ◆ 嵌入式系统是**嵌入到**对象体系中的以嵌入式计算机为核心的专用计算机系统。
- **基本要素：嵌入、专用性、计算机**

嵌入式系统的特点

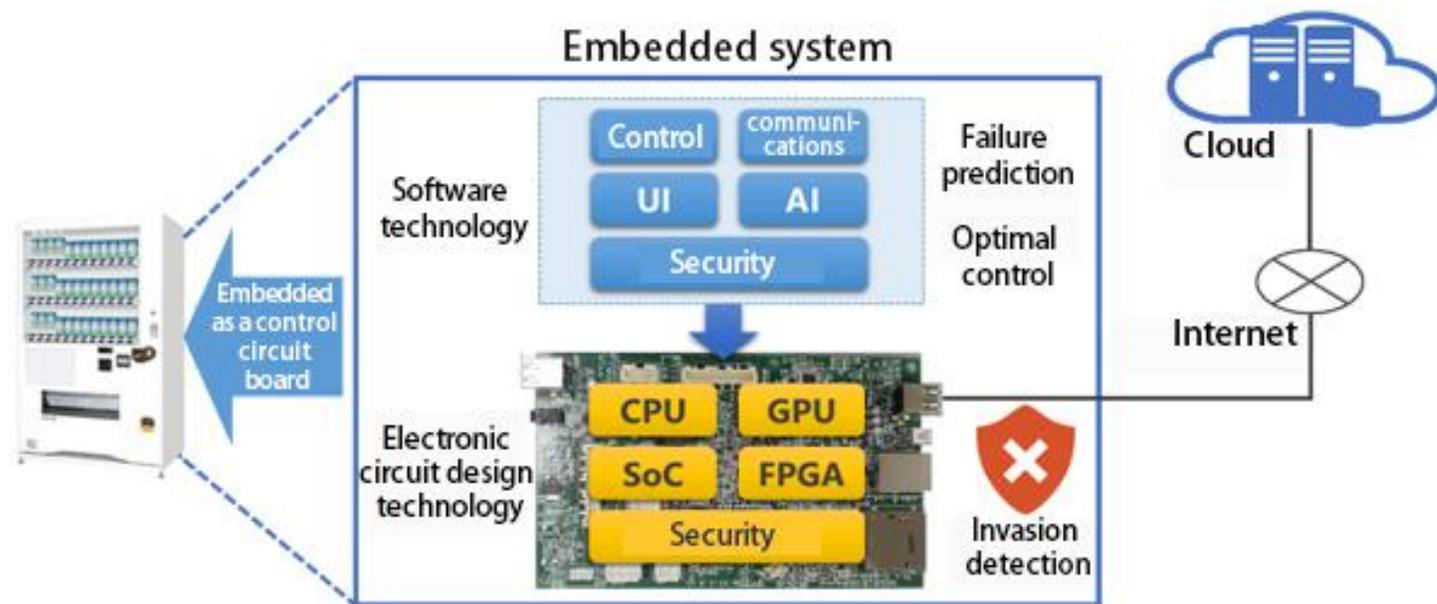
- “专用”的计算机系统；
- 运行环境差异很大；
- 比通用PC系统资源少；
- 低功耗、体积小、集成度高、成本低；
- 具有完整的系统测试和可靠性评估体系；
- 具有较长的生命周期；
- 需要专用开发工具和方法进行设计；
- 包含专用调试电路
- 多学科知识集成系统。

嵌入式系统的组成结构

- 1 嵌入式系统的硬件基本结构
- 2 嵌入式系统的软件基本结构
- 3 启动程序BootLoader介绍

嵌入式系统硬件基本结构

- 嵌入式系统的硬件架构如图下半部分所示。
- 嵌入式系统是“**量身定做**”的“**专用计算机应用系统**”。
- 硬件配置非常精简，除了微处理器和基本的外围电路以外，其余的电路都可以“**裁剪**”、“**定制化**”（Customize）。



嵌入式系统硬件基本结构介绍

- SoC——System on chip
- 可编程片上系统SOPC(System On Programmable Chip)。
- 现代嵌入式设计是以**处理器/SoC/SOPC**为核心，完成系统设计的。
 - DSP——**数字信号处理器(Digital Signal Processor)**
提高系统的信息处理能力，常外接DSP和DSP协处理器。

嵌入式系统软件基本结构

- 嵌入式系统软件结构一般包含四个层面：**板级支持包（BSP）层、实时操作系统（RTOS）层、应用程序接口（API）层、应用程序层。**
- **当设计一个简单的应用程序时，可以不使用操作系统。**对于功能简单仅包括应用程序的嵌入式系统一般不使用操作系统，**仅有应用程序和设备驱动程序。**
- 当设计较复杂的程序时，可能就需要一个操作系统（OS）来管理、控制内存、多任务、周边资源等等。

启动程序 BootLoader 介绍

- **BootLoader——开机程序**
- 相当于PC机中的BIOS（Basic Input /Output System）
- 系统加电复位后，几乎所有的CPU都从由复位地址上取指令。
- 因此在系统加电复位后，处理器将首先执行 BootLoader 程序。

启动程序BootLoader介绍

- BootLoader是系统加电后、操作系统内核或用户应用程序运行之前，首先必须运行的一段程序代码。
- 对于嵌入式系统来说，有的使用操作系统，也有的不使用操作系统，但在系统启动时都必须运行BootLoader。

嵌入式系统硬件组成及开发

- 嵌入式微处理器
- 典型32位嵌入式处理器介绍
- 嵌入式SoC/SOPC
- 嵌入式外围接口电路和设备接口

2.1 嵌入式处理器

- 嵌入式微处理器(Microprocessor)
- 嵌入式微控制器 (Microcontroller Unit)
- DSP (Digital Signal Processor)
- 片上系统SoC (System on Chip)

嵌入式微处理器

- 嵌入式微处理器(Microprocessor)就是和通用计算机的微处理器对应的CPU。
- 目前的嵌入式微处理器主要包括：、PowerPC、Motorola 68000、ARM、MIPS系列等等。

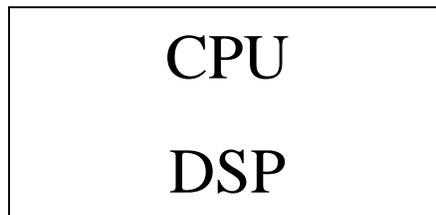
嵌入式微控制器

- 嵌入式微控制器(Microcontroller Unit, MCU)又称为**单片机**，它将CPU、存储器（少量的RAM、ROM或两者都有）和其它外设封装在同一片集成电路里。
- 常见的有**8051、P51XA、MCS-251等。单片机≠8051，单片机>8051。**
- 微控制器的特点是单片化，体积大大减小，从而使功耗和成本下降、可靠性提高。

嵌入式DSP

嵌入式DSP (Digital Signal Processor) 处理器有两个发展方向:

➤ **单片化设计**——嵌入式DSP处理器和嵌入式处理器集成，片上增加丰富的外设成为具有高性能DSP功能的SoC;



➤ 是在通用微处理器、微控制器或SoC中增加**DSP协处理器**，例如Intel的MCS-296和Siemens的TriCore。



典型嵌入式处理器介绍

- MIPS处理器
- PowerPC处理器
- Sparc处理器
- ARM处理器

嵌入式SoC

- 20世纪90年代后，“集成电路”级设计转向“集成系统”级设计(“IC”——“IS”)
- 目前嵌入式系统已进入单片系统SoC的设计阶段，并开始逐步进入实用化、规范化阶段，集成电路已进入SoC的设计流程。

嵌入式SoC发展的原因？

- 新产品在市场上的生命大为缩短，平均从36月到缩短为15-9月，要求产品的设计周期也大大缩短。
- 性能要求——高密度、高速度、高性能、小体积、低电压、低功耗。特别是低功耗。
- 成本价格、可靠性。

嵌入式SoC的特点

- 进入二十一世纪后，集成电路设计进入了高度集成的SoC时代。
- SoC设计技术优点：
 - 大幅度地提高系统的可靠性
 - 减少系统的面积和功耗
 - 降低系统成本
 - 极大地提高系统的性能价格比。

嵌入式SoC的设计理念

- 不是把所需的集成电路简单的二次集成到一个芯片上。
- 从整个系统性能要求出发，把微处理器、模型算法、芯片结构、外围器件各层次电路直至器件的设计紧密结合起来，并通过建立在全新理念上的系统软件和硬件的协同设计，在单个芯片上完成整个系统的功能。

嵌入式SoC举例

➤HMS30C7202是韩国现代公司开发的基于ARM720T内核、主频为70MHz、功能非常强大高集成度的片上系统，它片内外设的资源很多，广泛应用于PDA、智能电器、工业控制、网络设备、音频设备、电子图书、POS等。

可编程片上系统—SOPC

- 可编程逻辑器件——Programmable Logic Device (FPGA和CPLD)
- 基于FPGA技术的SOC设计方案——可编程片上系统SOPC
- SOPC是SOC的一种，同时具备软硬件在系统可编程的功能(可裁剪、可扩充、可升级)

3 嵌入式系统的软件组成及开发

3.1 嵌入式系统的软件层次结构

3.2 嵌入式操作系统

3.3 嵌入式系统的软件开发

3.1 嵌入式系统的软件层次结构

板级支持包层

实时操作系统层

应用程序接口层

应用程序层

3.2 嵌入式操作系统

- 1 操作系统与嵌入式操作系统
- 2 嵌入式实时操作系统
- 3 嵌入式操作系统基本管理功能
- 4 典型嵌入式操作系统介绍

操作系统与嵌入式操作系统

- 计算机由硬件和软件两部分组成；
- 操作系统可以解释或理解为**补平**硬件差异的界面或说**隐藏**硬件，让应用程序可以在上面运行。
- 嵌入式操作系统EOS(Embedded Operating System)是一段嵌入在目标硬件中的软件，用户的其他应用程序都建立在嵌入式操作系统之上。

嵌入式实时操作系统

- 嵌入式系统一般具有**实时**特点。
- 所谓实时系统，是指一个能够在指定或者确定的时间内完成系统功能以及对外部或内部、同步或异步事件作出响应的系统。

嵌入式实时操作系统

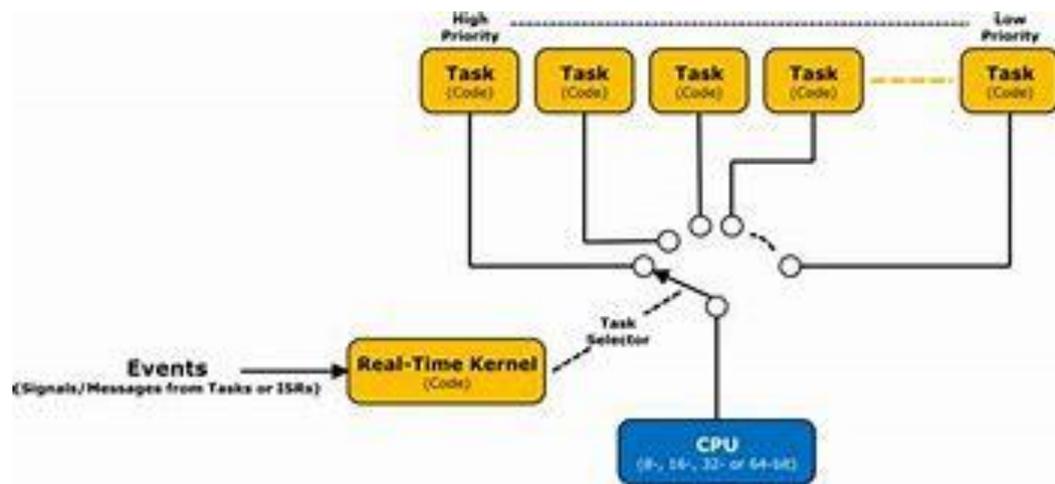
- 大多嵌入式系统应用在实时环境中，因此嵌入式操作系统大部分是实时操作系统（RTOS）。
- 嵌入式实时操作系统的主要特点是性能上的“实时性”，系统的正确性不仅依赖于计算的逻辑结果，也依赖于结果产生的时间。
- 实时操作系统可以根据实际应用环境的要求对内核进行**剪裁**和**重新配置**，组成可根据实际的不同应用领域而有所不同。但以下几个重要组成部分是不太变化的：**实时内核、网络组件、文件系统和图形接口**等。

嵌入式操作系统基本管理功能

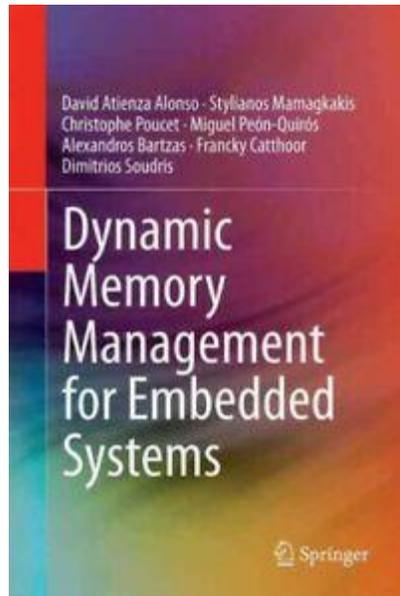
嵌入式操作系统相对于一般操作系统而言，仅指操作系统的**内核**（或者微内核），其他的诸如窗口系统界面或是通讯协议等模块，可以另外选择，目前大多数的嵌入式操作系统必须提供以下管理功能：

- 多任务管理
- 存储管理
- 周边资源管理
- 中断管理

多任务管理



存储管理

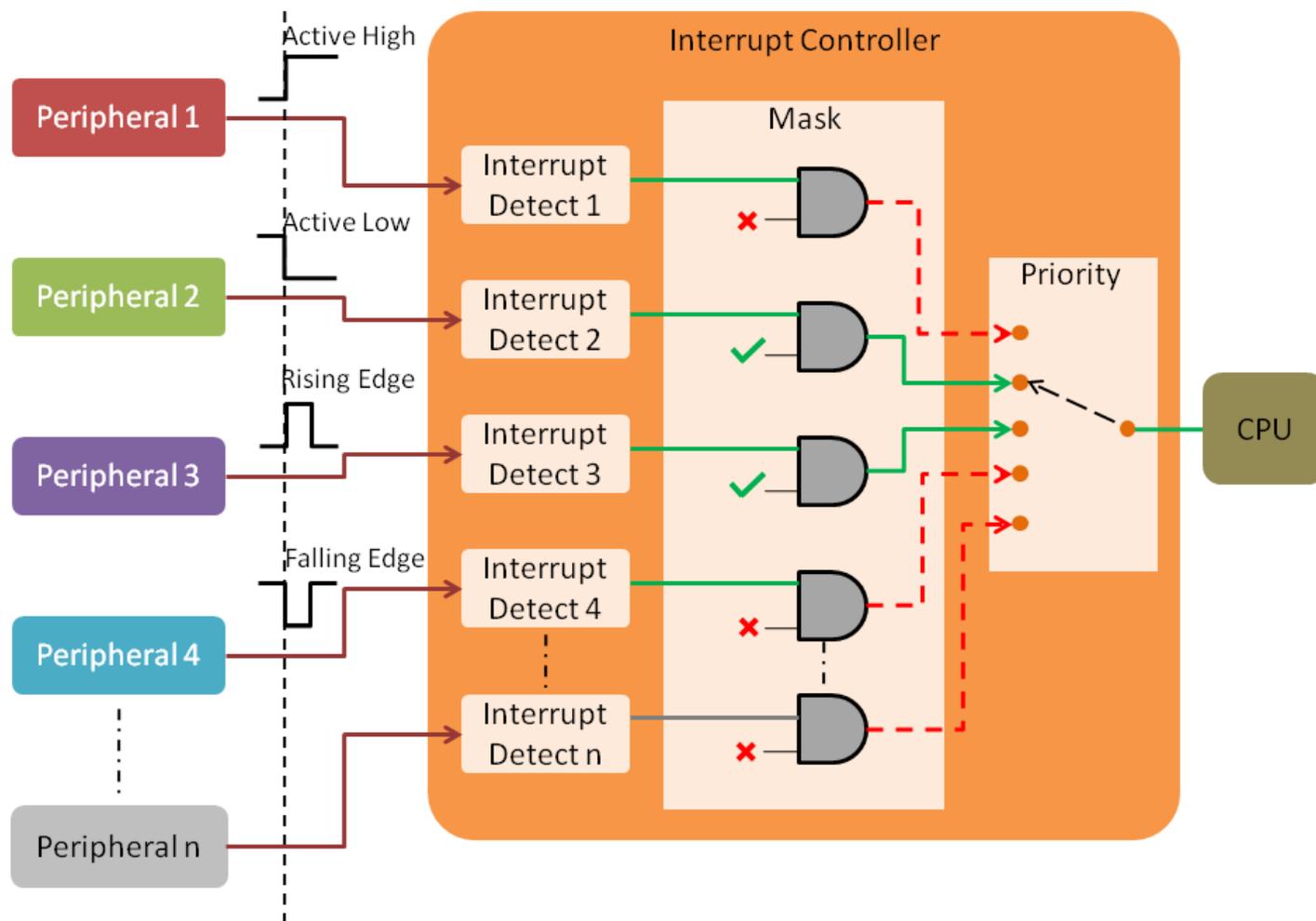


一般不使用虚拟内存
动态内存管理

周边资源管理



中断管理



典型嵌入式操作系统介绍

嵌入式操作系统的种类繁多，但大体上可分为两种——**商用型和免费型**。

- 商用型的操作系统主要有VxWorks、Windows CE、Psos、Palm OS、OS-9、LynxOS、QNX、LYNX、IOS、Symbian等
- 免费型的操作系统主要有Linux、 μ C/OS-III和Android
- FreeRTOS

典型嵌入式操作系统介绍

- Linux操作系统
- VxWorks嵌入式实时操作系统
- WinCE操作系统
- uC/OS-III 嵌入式操作系统
- Android操作系统
- IOS操作系统
- Symbian操作系统

嵌入式操作系统

常见的嵌入式操作系统

——嵌入式Linux

uClinux是一个完全符合GNU/GPL公约的操作系统，完全开放代码。uClinux从Linux 2.0/2.4内核派生而来，沿袭了主流Linux的架构，并针对没有MMU的CPU，并进行了优化。适用于没有MMU的处理器，例如ARM7TDMI或Flash的嵌入式系统。uClinux具有稳定、良好的移植性、优秀的网络功能、完备的对各种文件系统的支持、以及标准丰富的API等。



嵌入式操作系统

常见的嵌入式操作系统

——Win CE

Windows CE是微软开发的一个开放的、可升级的32位嵌入式操作系统，它是精简用户界面相当出色于Win32应用程序。CE不仅继承了传统Windows CE平台上可以使用Visual Basic、Visual C++等）、使绝大多数的应用软件只需简单的修改和移植就可以在Windows CE平台上继续使用。



嵌入式操作系统

常见的嵌入式操作系统

——VxWorks

VxWorks操作系统是美国 **WIND RIVER** 公司于1983年设计开发的一种嵌入式实时操作系统（RTOS），是嵌入式开发环境的关键组成部分。良好的持续发展能力、高性能的内核以及友好的用户开发环境，在嵌入式实时操作系统领域占据一席之地。它以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中，如卫星通讯、军事演习、弹道制导、飞机导航等，甚至在1997年4月登陆火星表面的火星探测器上也使用到了VxWorks。

嵌入式操作系统

常见的嵌入式操作系统

—— μ C/OS-III

μ C/OS-III是一个源码公开、**可移植、可固化、可裁剪、占先式**的实时多任务操作系统。其绝大部分源码是用ANSI C写的，使其可以方便的移植并支持大多数类型的处理器。 μ C/OS-III通过了联邦航空局 (FAA) 商用航行器认证。自1992年问世以来， μ C/OS-III已经被应用到数以百计的产品中。 **μ C/OS-III占用很少的系统资源**，并且在高校教学使用是不需要申请许可证。

3.3 嵌入式系统的软件开发

- 嵌入式软件开发的特点和技术挑战
- 嵌入式软件开发环境
- 嵌入式应用软件开发的基本流程
- 嵌入式软件开发的可移植性和可重用性

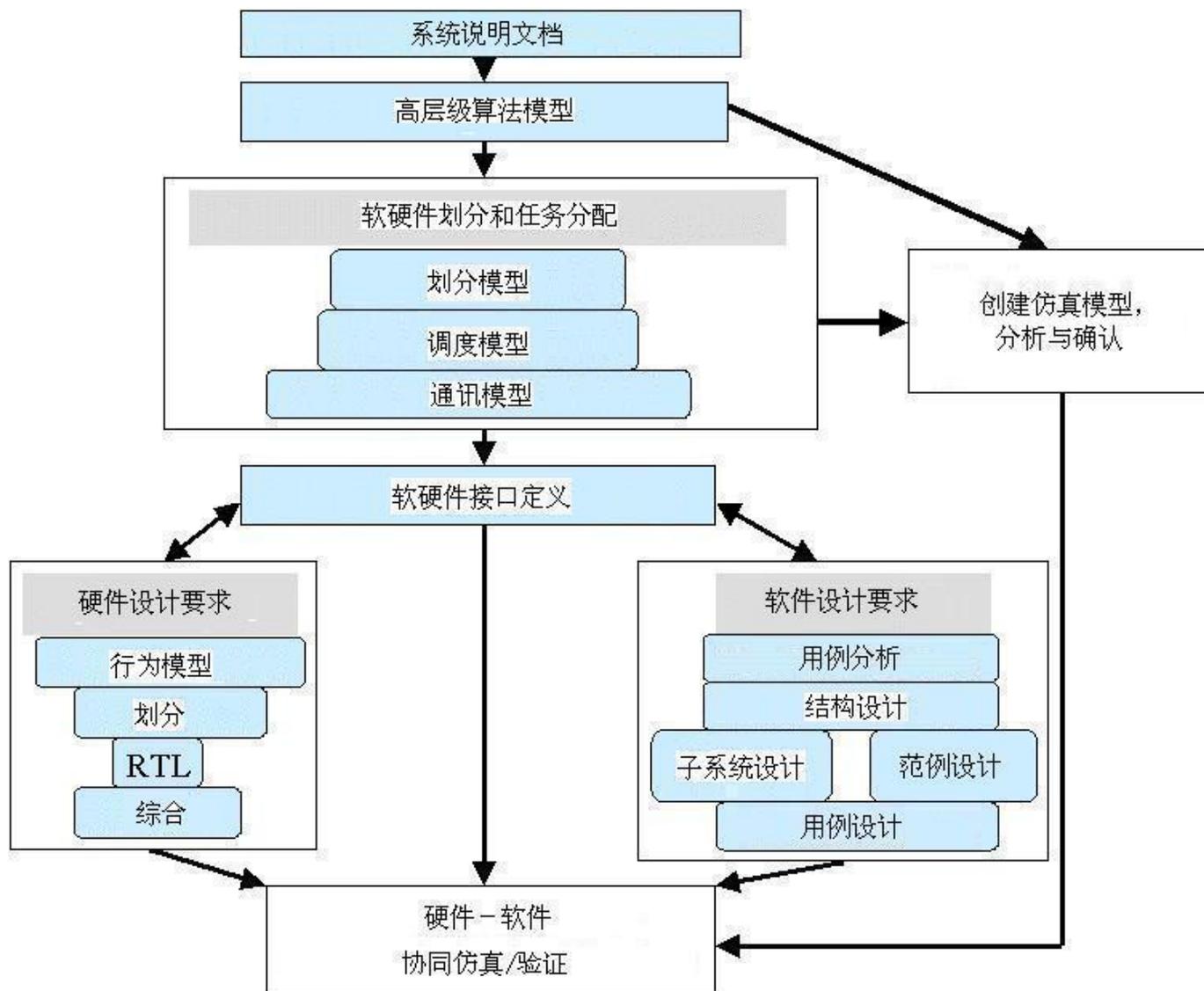
嵌入式软件开发的特点和技术挑战

- 嵌入式软件开发需要软件开发环境和工具
- 嵌入式软硬件必须协同设计
- 需要新的任务设计方法
- 开发过程完成后，系统应用程序代码需要固化到系统中进行功能、性能和可靠性测试
- 技术挑战：软件的要求更高，开发工作量和难度更大

软硬件协同设计

这种方法的特点是在设计时，从系统功能的实现考虑，把**实现时的软硬件同时考虑进去**，硬件设计包括芯片级“功能定制”设计。既可以最大限度的利用有效资源、缩短开发周期，又能取得更好的设计效果。

软硬件协同设计



嵌入式软件开发环境

- 嵌入式系统应用软件是跨平台开发的，或称为交叉开发的。
- 交叉开发是指在一台通用计算机上进行软件的编辑编译，然后下载到嵌入式设备中运行调试的开发方式。通常采用宿主机/目标机模式。
- 开发计算机称为宿主机，嵌入式设备称为目标机。
- 交叉开发环境由宿主机上的交叉开发软件和宿主机到目标机的调试通道组成。

嵌入式软件开发环境

- 交叉开发软件一般为一个整合编辑、编译、汇编、链接、调试、工程管理及函数库等功能模块的**集成开发环境IDE (Intergrated Development Environment)**。
- 嵌入式交叉开发环境的宿主机到目标机的调试通道一般有以下四种：
 - 在线调试 (On-Chip Debugging, OCD) 或在线仿真 (On-Chip Emulator)
 - * 基于JTAG的ICD (In-Circuit Debugger)
 - * 背景调试模式 (BDM)
 - 在线仿真器ICE (In-Circuit Emulator)
 - ROM监控器 (ROM monitor)
 - ROM仿真器 (ROM Emulator)

嵌入式软件开发环境

2) 软件模拟环境

➤ 软件模拟环境也称为指令集模拟器ISS(Instruction Set Simulator)

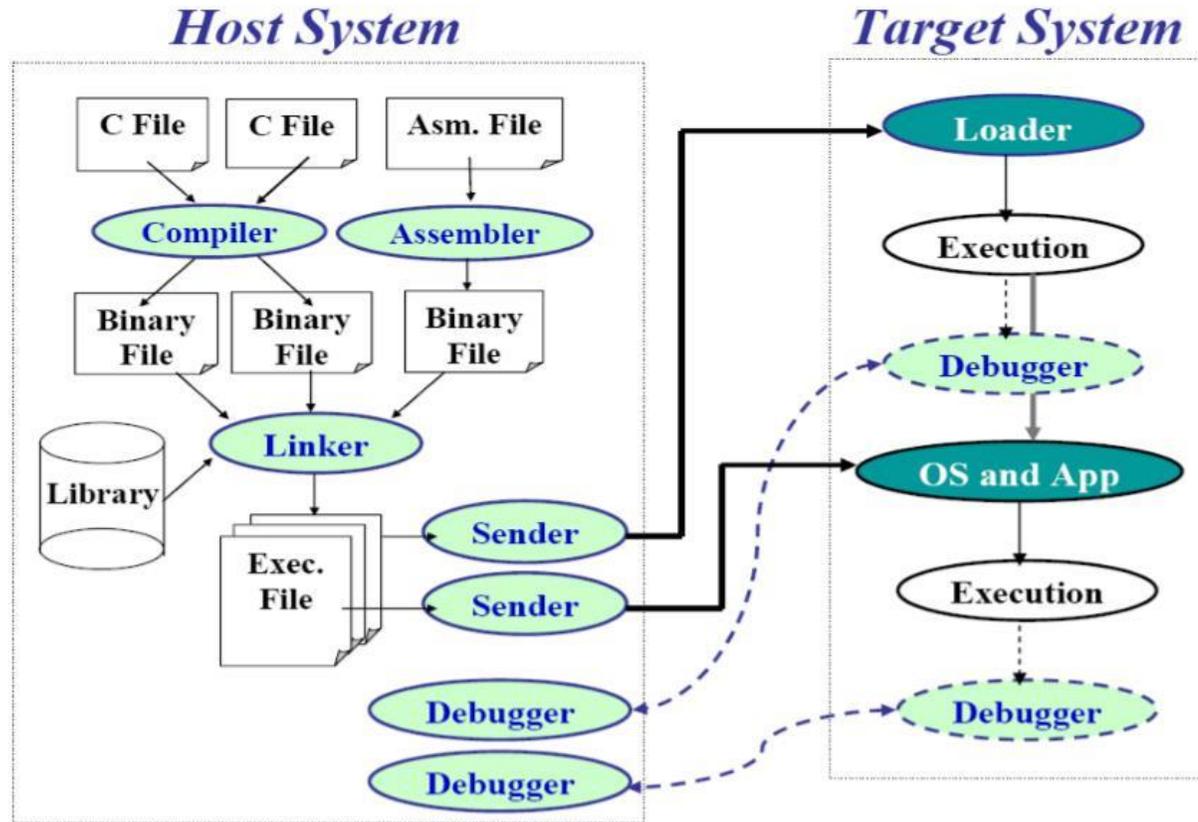
➤ 软件模拟不可能完全代替真正的硬件环境，这种模拟调试只能作为一种初步调试，主要是用作用户程序的模拟运行，用来检查语法、程序的结构等简单错误，用户最终还必须在真实的硬件环境中实际运行调试，完成整个应用的开发。

嵌入式软件开发环境

3) 评估电路板——开发板

一般用来作为开发者使用的学习板、实验板，可以作为应用目标板出来之前的软件测试、硬件调试的电路板

Development Environment



嵌入式应用软件开发的基本流程

基于交叉开发环境的嵌入式应用软件开发主要分如下五个步骤：

- 开发环境的建立
- 源代码编辑阶段
- 交叉编译和链接
- 重定位和下载
- 联机调试

嵌入式应用软件开发的基本流程

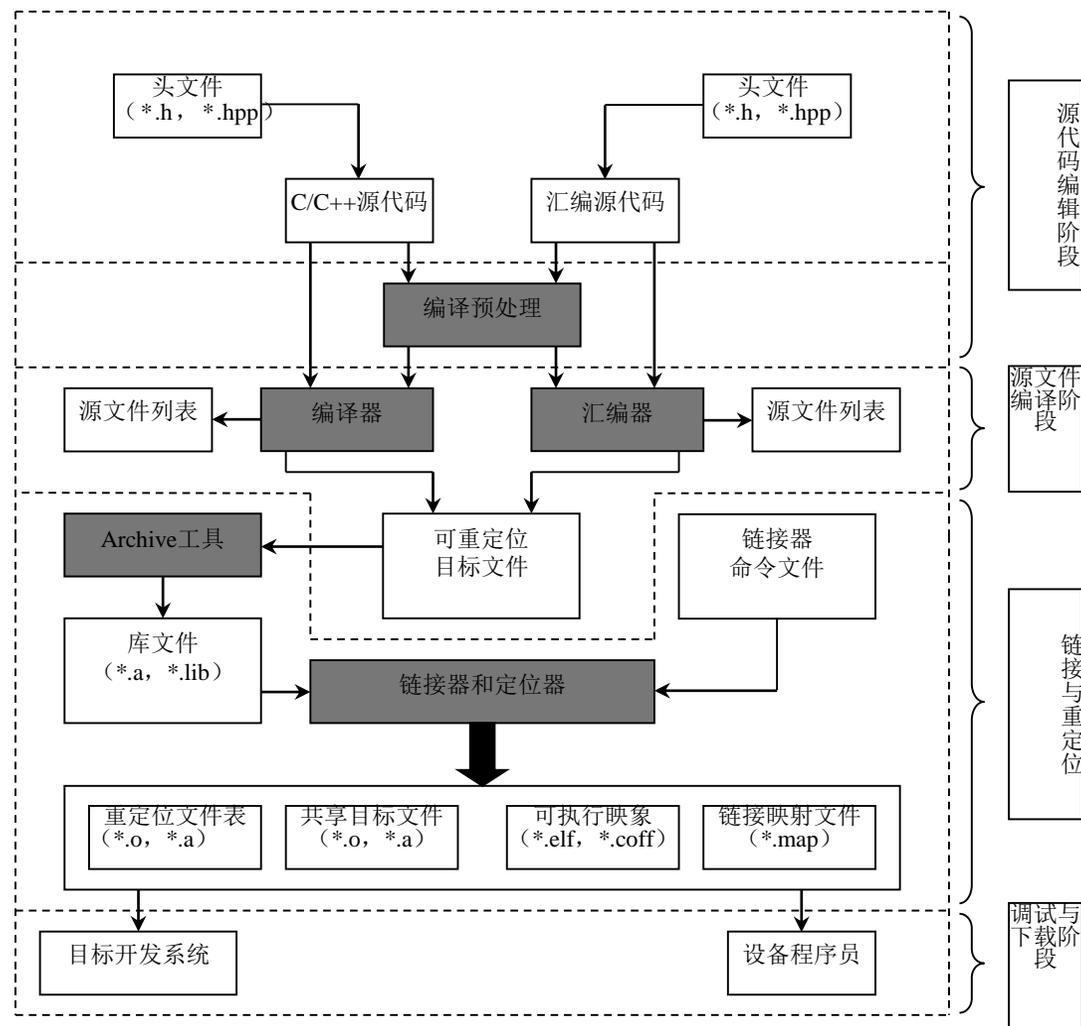


图1-7 嵌入式软件开发流程图

嵌入式软件开发的可移植性和可重用性

在确保软件的正确性、实时性的前提下，必须关注软件的可移植性和可重用性。

采用下面的方法可以提高应用软件的可移植性和可重用性。

- 多用高级语言少用汇编语言
- 将不可移植部分局域化
- 提高代码的可重用性