

嵌入式系统原理与应用 提纲 by [阳光男孩](#) || 你的阳光学习频道

<http://study.yoursunny.com>

客服电话021-31001856转*018225008# || Provided as-is, absolutely no warranty

考试题型和范围

- 单选20、多选20、填空20、判断10——概述(不含历史)，软件(不含具体产品介绍)，硬件
- 简答20——Bootloader，设计方法，Linux
- 应用10——实例(根据需求选平台、开发方法)

下文中@后面文字为阳光男孩加的注释

一、嵌入式系统基础

(一)嵌入式系统概述

- 嵌入式计算机系统(简称嵌入式系统)：看不见的计算机，一般不能被用户编程，它有一些专用的I/O设备，对用户的接口是应用专用的
- 定义：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、功耗严格要求的计算机系统
- 体系结构 @单选题
 - 冯诺依曼结构：指令数据一起存储；桌面计算机采用
 - 哈佛结构：指令数据分开存储；嵌入式系统通常采用
- 系统层次比较 @填空题
 - 桌面计算机：硬件-固件-操作系统-应用程序
 - 复杂嵌入式系统：硬件-固件-操作系统-应用程序
 - 简单嵌入式系统：硬件-固件
- 嵌入式系统发展：1960年出现(航天)，1965年军用，1973年民用，1980年代完善，1990年代广泛应用 @历史不考~
- 嵌入式系统价值体现：外观，品牌，功能，重量，价格，质量，服务，综合

实时系统

- 嵌入式系统不一定是实时系统，实时系统一般都是嵌入式系统 @判断题
- POSIX实时系统：操作系统有能力在限定响应时间范围内，提供满足需求的服务
- 实时系统指标 @单选题
 - 响应时间：识别外部事件->响应
 - 生存时间：数据有效等待时间
 - 吞吐量：可处理时间总数/单位时间
- 实时系统响应时间在10 μ s数量级；响应时间可预测性比速度更重要
- 怎么达到实时的要求？ @多选题/简答题
 - 充分发挥硬件的功能
 - 微处理器的终端机制

- 简单的单线程循环程序
- 基于实时操作系统的复杂多线程程序
- 以硬件的方式实现软件

嵌入式系统分类

- 按确定性分类 @单选题，根据[01嵌入式系统概述.ppt]125页图片选择类别名称
 - 硬实时系统：响应时间不能满足会导致系统崩溃
 - 软实时系统：响应时间不能满足无致命错误
 - 严格实时系统：响应时间不能满足导致无法接受的低质量服务
 - 非实时系统
- 实时系统-按软件结构分类
 - 单线程程序
 - 简单轮询系统：软件结构简单，应用领域有限
 - 有限状态机：小系统结构简单，快速执行，应用领域有限
 - 事件驱动程序
 - 前后台系统 @填空题
 - 前台-中断级：中断服务程序处理异步事件；后台-任务级：应用程序，无限的循环
 - 性能指标：中断延迟时间，响应时间，恢复时间
 - 实时多任务系统：多线程，保证实时性，模块化好易维护，需使用RTOS增加开销
 - 多处理机系统
 - 紧耦合系统：共享内存交换信息
 - 松耦合系统：通讯链路交换信息

(二)嵌入式系统软件

- 软件体系结构 @多选题
 - 驱动层
 - 板级初始化程序：初始化CPU、存储器、中断控制器、DMA、定时器等
 - 与系统软件相关的驱动
 - 与应用软件相关的驱动
 - 操作系统层：(必须)嵌入式内核，(可选)嵌入式TCP/IP网络，嵌入式文件系统，嵌入式GUI系统，电源管理
 - 中间件层：(可选)嵌入式COBRA，嵌入式JAVA，嵌入式DCOM.....
 - 应用层：完成特定的工作
- 特征 @多选题/填空题
 - 固化存储于ROM，高质量高可靠性，高实时性，特殊开发平台，需要RTOS支持，开发人员是应用专家
- 实时软件要求 @填空题
 - 实时性，处理异步并发事件能力，快启动自恢复，应用程序与RTOS一体，独

立开发平台

- 软件组成 @多选题
 - Bootloader
 - 操作系统
 - 应用程序
- 嵌入式系统软件对存储器的需求 @填空题
 - ROM需求
 - 前后台系统：总代码量=应用程序代码
 - 多任务系统：总代码量=应用程序代码+内核代码(1~100K,8bit最小内核1~3K)
 - RAM需求
 - 多任务系统，无单独任务中断用栈：总RAM=应用程序需求+(任务栈需求+最大中断嵌套栈需求)×任务数
 - 多任务系统，中断用栈分离：总RAM=应用程序需求+内核数据区需求+各任务栈需求+最大中断嵌套栈需求

操作系统基础 @略

嵌入式操作系统

- 实时操作系统内核 @多选题，[01嵌入式系统软件.ppt]90页
 - 核心：多任务环境；外围：文件系统接口，虚拟内存管理，网络协议栈，I/O系统接口
 - 最简内核(纳核)8K，只剩下任务的管理与调度
- 按实时性分类 @判断题
 - 强实时系统：ms~ μ s数量级，数控机床
 - 弱实时系统：s数量级，工程机械
- 实时操作系统：VxWorks,WinCE,Palm,Linux,ucLinux @产品不考
- 实时操作系统组成：交叉编译器，RTOS本身，软件调试工具包
- 实时操作系统特点 @填空题
 - 确定性：响应时间实时、可预测、可重复实现；中断请求->中断响应
 - 响应性：ISR(中断服务程序)执行时间；中断响应->中断结束
响应时间=确定性+响应性 @判断题
 - 用户控制：允许用户(开发者)精确控制任务优先级
 - 可靠性：避免性能丧失或降低
 - 软失败操作：最小化系统错误的影响

实时系统基本概念 @略

GUI 图形化对话接口

- GUI的主要特征：Windows,Icons,Menu,Pointing devices @填空题/多选题
- 嵌入式GUI的特点：体积小，功能强，算法简洁占资源少，可靠性高，模块化便于移植

(三)嵌入式系统硬件

- 硬件平台架构：CPU，总线，内存，I/O设备 @多选题
- 开发板，BSP，PLD，FPGA，ASIC，COTS商业通用组件(传感器等)，特定处理器(DSP、MCU、MPU) @判断题
- 基于PC平台的嵌入式系统：价格低廉，易于采购，开发环境丰富，熟悉；需要大量硬件资源，非实时 @应用题

嵌入式处理器

- 指令集 @判断题
 - 复杂指令集(CISC)：大量指令和寻址方式；桌面机CPU采用
 - 精简指令集(RISC)：只含最有用的指令，确保数据通道高效执行，CPU硬件结构简单；ARM、DSP采用
 - 并行指令集(EPIC)
 - 超长指令集(VLIW)
- 指令系统设计
 - 指令系统设计 @填空题
 - 决定于应用、性能、代码密度、方言
 - 包括符号指令设计、编码设计
 - 指令的类型
 - 寻址方式
 - 传输类指令
 - 运算类指令
 - 系统类指令
 - 指令的编码
 - 与代码密度、功耗、译码器相关
 - 垂直编码译码器简单、效率低；非垂直编码译码器复杂、效率高 @判断题
- 可重配置处理器，不可重配置处理器
- 处理器分类
 - 指令长度：4,8,16,32,64
 - 集成度：MPU,单片机,SoC
 - 用途
 - 通用处理器：x86,ARM,MIPS,PowerPC
 - 专用处理器：DSP,IXP,IOP
 - 指令集
- 影响CPU性能的因素
 - 流水线：几个指令并行执行
 - 超标量：采用多条流水线结构

- 高速缓存：保存部分主存内容的拷贝
 - 数据cache, 指令cache, 混合cache
 - cache命中, 通写、回写
- 总线和总线桥
- 嵌入式处理器分类
 - 嵌入式微处理器MPU: PowerPC, Motorola 68000, ARM
 - 与通用处理器相比, 只保留嵌入式应用相关的硬件
 - 用于执行单一特定程序
 - 嵌入式微控制器MCU: 8051, MCS-251
 - 单片机: 内部集成ROM RAM 总线 I/O A/D D/A 串口 Flash等各种必要功能和外设
 - 嵌入式DSP: 适合连续的数据流处理、高精度复杂运算
 - SoC/RSoC: 结合许多功能区块, 做在一个芯片上; 软硬件无缝结合
 - 通用处理器: x86, AMD
 - 内存可编程, 通用寄存器, 通用算术逻辑单元

常见嵌入式处理器

- ARM
 - IP核: ARM RISC处理器是SoC的重要部件, 连接了数码世界的软、硬件
 - ARM系列: ARM7, ARM9, StrongARM, XScale.....
 - ARM优势: RISC 32位处理器, 价格低廉, 合作伙伴众多, 低功耗
- MIPS
 - ISA指令集体系, 支持64位整数

显示屏

- LCD液晶显示屏
 - 段式液晶
 - 字符型液晶
 - 图形点阵型液晶
- 触摸屏

存储系统

- RAM: SRAM, DRAM
- ROM: Masked ROM, OTP PROM, EPROM, EEPROM
- Flash: NOR Flash, NAND Flash
 - Flash设备按块读写, 每块可写入100000次
 - Flash设备模拟为512字节的块设备时, 为了提供平均损耗, 设备与文件系统上的逻辑位并不一一对应 @判断题
 - 开发板上的Flash内存逻辑分区: Boot Flash + Application Flash

总线

- 总线的主要参数
 - 带宽(=工作时钟频率×位宽) @填空题
 - 位宽
 - 工作时钟频率
- 总线分类
 - 按接口分类 @多选题“下列哪些是内部总线：A VXI,B I2C,C SCI,D PCI”
 - 片内总线：处理器内部ALU与寄存器连接
 - 局部总线：CPU引脚的延伸
 - 系统总线：计算机系统各插件板的连接总线；ISA,PCI,CompactPCI
 - 内部总线：I2C,I2S,SPI,SCI
 - 外部总线：计算机系统之间、计算机与外设之间连接；RS-232,IEEE1394,USB
 - 现场总线：CAN
 - 按功能分类 @填空题
 - 地址总线：决定了寻址范围
 - 数据总线：决定了同时传送的数据位数
 - 控制总线
 - 电源和地线
 - 备用线

接口

- 并口，串口，USB

JTAG

- Joint Test Action Group
- 使得集成电路固定在PCB上，只通过边界扫描便可以测试

板级支持包BSP

- 标准开发板上运行的一部分软件，支持操作系统的引导与开发
- BSP特点
 - 硬件相关性
 - 操作系统相关性
- 架构：引导程序，硬件抽象层，驱动，配置文件
- 功能
 - 嵌入式系统初始化
 - 片级初始化：初始化微处理器
 - 板级初始化：初始化其他硬件
 - 系统级初始化：初始化操作系统

- 硬件相关的设备驱动程序

二、嵌入式系统高级知识

(一)Bootloader

- **Bootloader:** 在操作系统内核运行之前运行的一段小程序，它初始化硬件设备、建立内存空间映射图，从而将系统的软硬件环境带到合适的状态，为调用操作系统内核作好准备
- 嵌入式系统启动过程 @简答题
 1. 系统上电或复位，从0x0地址开始执行
 - bank的划分：把外部总线分成不同的bank空间，对应不同的地址；每个bank可以对应一个ROM、Flash、DRAM等设备
 - 0x0地址(CPU也可能规定其他地址)保存有启动代码，通常对应bank0(连接Boot Flash)
 2. Bootloader stage1任务：用汇编语言编写
 - 硬件设备初始化：关中断，设置CPU频率，RAM初始化，LED(或其他可以表明系统状态的简单设备)初始化，关闭cache
 - 为stage2准备RAM地址空间，将stage2代码拷贝到RAM中，设置堆栈指针，跳转到stage2入口
 3. Bootloader stage2任务：用C语言编写，但不能动态链接glibc(允许静态链接)
 - 初始化本阶段要用到的硬件设备：一个串口进行I/O，计时器
 - 建立系统内存映射：配置其他bank，设置端口、外设工作模式
 - 加载操作系统内核和根文件系统映像
 - 为内核设置启动参数，调用操作系统内核
 4. 操作系统启动
- Bootloader的制作
 - 原有Bootloader下载新Bootloader
 - 将Flash ROM插在编程器上
 - 使用JTAG接口
- 操作模式
 - 启动加载模式(自主模式)：从嵌入式系统的某个存储设备(通常是Flash ROM)加载操作系统到RAM
 - 下载模式：通过串口或以太网下载操作系统内核与根文件系统；只用于开发(第一次安装和升级)

ARM启动过程 @略

U-BOOT @略

(二)ARM处理器

@略

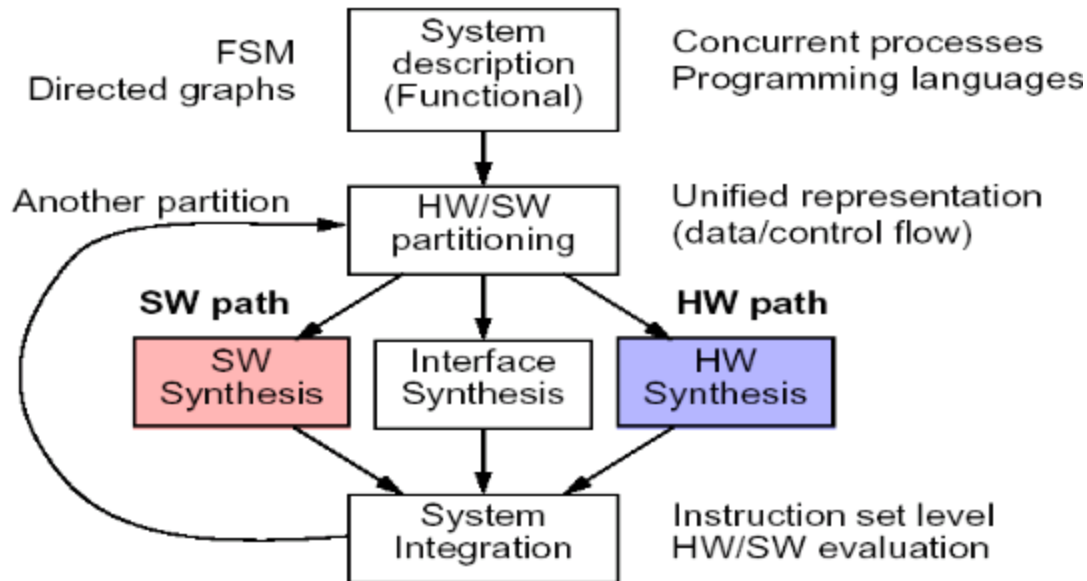
三、嵌入式系统设计

(一)系统级设计



- 计算机设计：实时性，响应性，安全可靠，苛刻环境适应性，低成本
- 系统级设计：产品功能，系统安全可靠，物理系统的控制
- 生命周期：市场需要→概念设计→产品设计、生产流程设计→生产→发布→支持→升级→淘汰
- 商业模式：设计和生产费用，循环周期，产品系列
- 设计原则

(二)软硬件协同设计



- 软硬件协同设计：软硬件共同设计目的是为硬件和软件的协同描述、验证和综合提供一种集成环境
- 传统设计方法(系统级设计)的存在的问题 @简答题
 - 缺少统一的软硬件表示方法
 - 软硬件划分依靠先验定义
 - 不能够验证整个系统
 - 软硬件边界不兼容问题难以发现
 - 缺少成熟的设计流程
 - 上市时间困难（容易拖期）
 - 难以更改描述
- 软硬件协同设计理论体系
 - 系统任务描述
 - 软硬件划分
 - 软硬件协同综合
 - 软硬件协同仿真
- 软硬件共同设计带来的好处
 - 缩短开发周期
 - 取得更好的设计效果
 - 满足苛刻的限制
 - 平台=芯片+开发系统、典型应用+服务和技术支持
- 软硬件协同设计的基本需求 @简答题
 - 统一的软硬件描述方式：目前较为缺乏
 - 交互式软硬件划分技术
 - 完整的软硬件模型基础
 - 正确的验证方法：目前较为缺乏

- 软硬件协同设计内容
 - 设计描述
 - 设计建模
 - 设计空间的研究和划分
 - 合成与优化
 - 设计验证
 - 设计实现

(三) 软硬件协同划分

@根本没讲过，略

四、Linux嵌入式系统开发

- Linux实时实现方法
 - 实时内核调度实时任务，无实时任务时实时内核调度标准内核运行普通进程
 - 将标准内核修改为支持实时任务和普通进程
- Linux为什么成为嵌入式操作系统的主流
 - 免费
 - 稳定
 - 网络功能强
 - 开发工具丰富
 - 大量文档
- 嵌入式系统开发模式 @简答题
 - 本机开发
 - 交叉开发
 - 仿真开发
- Linux的内核格式 @判断题
 - vmlinux: 未压缩的可执行文件
 - zImage: 经压缩，1MB实模式内存
 - bzImage: 经压缩，无1MB限制，现在常用

@我不认为这章要考，做过实验~，因此其他内容略

五、嵌入式系统产品开发实例

@应用题；别忘了第一章的“嵌入式系统价值体现”~

- 硬件平台选择
 - 性能：建立原型系统测试效率能否满意，参考其他同类产品
 - 价格
- 软件平台选择
 - WinCE, Linux(开源嵌入式Linux、商业嵌入式Linux、普通Linux), VxWorks

- 无操作系统，条件为：任务简单，单任务
- 开发工具选择
- Bootloader选择
- 设计方法：状态机，中断，状态机+中断
 - 实时性要求
 - 系统复杂度：设计复杂度，实现复杂度
 - 系统可扩展性