- 嵌入式系统有多种表现形式，包括计算机MCU、SOC片上系统、SOPC片上系统、GPU和FPGA等。
    - MCU(微控制器): 是最基本也是最常见的嵌入式系统形式,是**集成了CPU、ROM、RAM、IO口、定时器、中断控制器等组件的单一芯片**。MCU广泛用于电器电子产品的控制。
    - SoC(系统片上芯片): 集成了多个功能模块,可实现复杂计算任务。与MCU相比,SoC集成度更高,性能更强,但也更昂贵。典型的SoC如手机SoC、车用SoC等。
    - FPGA(现场可编程门阵列): 具有高度可编程性和可配置性的半定制芯片。FPGA可在上电后通过配置逻辑电路来实现特定功能，具有非常高的灵活性和可重构性，但成本较高。
    - ASIC(专用集成电路): 专门为某个应用设计的集成电路，性能高、功耗低、成本高，适合大规模生产。
    - GPP(通用处理器): 通用处理器是一种能够执行通用计算任务的处理器，比如Intel的x86、ARM的Cortex-A等。通用处理器不仅可以运行各种操作系统，还可以运行各种应用程序，因此具有非常广泛的适用范围。
    - GPU(图形处理器): 专门为图形处理而设计的处理器，能够高速地进行并行计算，在游戏、视频处理、科学计算等领域有广泛应用。
    - SoPC(System on Programmable Chip): 是**构建在FPGA基础上**的单芯片系统,具有高自由度和可配置性。SoPC通过用户自定义逻辑将IP模块集成在一起，可以在FPGA上实现各种数字电路和嵌入式系统。
- 不同的嵌入式系统表现形式适用于不同的应用场景，具有各自的**优缺点**。
    - MCU（微控制器）：集成度低，但成本低廉，适合于对资源要求不高的低功耗应用，例如家电、智能家居、玩具等。
    - SoC（系统片上芯片）：集成度高，性能强，适合于高性能、多功能、复杂计算的应用场景，例如智能手机、车载电子、智能摄像头等。成本较高，不适合于低成本、低功耗的应用场景。
    - FPGA（现场可编程门阵列）：具有高度可编程性和可重构性，适合于需要快速定制、快速原型设计的应用场景，例如数字信号处理、通信、高速计算等。
    - ASIC（专用集成电路）：适合于需要高性能、低功耗、大批量生产的应用场景，例如芯片卡、网络交换机、路由器等。需要长时间的设计和生产周期，不适合于快速原型设计和快速定制的应用场景。
    - GPP（通用处理器）：适合于需要高性能、通用性、可编程性的应用场景，例如桌面计算机、服务器、移动设备等。功耗较高，不适合于低功耗应用场景。
    - GPU（图形处理器）：适合于需要高性能图形处理的应用场景，例如游戏、虚拟现实、深度学习等。专门用于图形处理，不适合于通用计算任务。
    - SoPC（基于FPGA的系统芯片）：具有高自由度和可配置性，适合于需要快速原型设计、快速定制的应用场景，例如数字信号处理、嵌入式系统设计等。需要设计和配置的时间和成本较高，不适合于低成本应用场景。
- 计算机是软件和硬件的结合体制，软件和硬件可以相互转化。历史上第一台计算机全由硬件构成，现在绝大部分场景下，计算机都是软硬件综合起来的。
- **对于实时性要求很强的末端控制系统等特定场景，也可以完全用FPGA来实现，全硬件实现软件的功能。**
- **FPGA具有可编程性**，可以根据应用场景的需要进行**定制化开发**，从而达到更高的性能和可靠性。
- ARM芯片是一种广泛应用于嵌入式系统中的处理器架构
    - 国内有很多公司在做ARM芯片，包括**全志科技、瑞芯微**等。
    - 联发科技、展讯通信、华为海思也在做
- GPU也是嵌入式系统中常用的处理器
    - 国内的**景嘉微**是一个比较好的GPU芯片厂商。

- FPGA是一种可编程硬件
  - 国外的主要厂商包括**Xilinx、Altera、Lattice、Microsemi**
  - 国内的**复旦微**是一家FPGA芯片厂商。
- 单片机51系列是一种经典的微控制器芯片
  - 单片机51系列是一种经典的微控制器芯片，是由**英特尔**推出的一款8位微控制器，应用广泛，具有低功耗、易于编程、成本低等特点。
  - 国内有很多厂家在做，其中**宏晶电子和STC**是比较有名的厂商。
- 国内的嵌入式系统产业已经比较成熟，有很多能力强、产品质量好的厂家能够提供各种类型的芯片产品，与国外的产品相当甚至更好。
- ARM是一种处理器架构，也是一种技术的总称，同时也是一家公司的名字。
- ARM公司本身并不生产SOC或MCU，而是向全球100多家半导体制造商出售IP CORE或ISA（指令集架构）。
- ARM芯片最开始的命名规则为ARM、ARM7、ARM9、ARM11，后来发展出了基于Cortex架构的三个系列：Cortex-A、Cortex-R和Cortex-M。
  - Cortex-A系列适用于**高性能**场景，如手机等；
  - Cortex-R系列适用于**实时性**要求较高的场景；
  - Cortex-M系列适用于**低功耗**、成本较低的场景。
- STM32是**意法半导体公司**购买了ARM的IP CORE生产的MCU，属于**Cortex-M**系列。
- 对于实时性要求较高的系统，我们需要对硬件有基本的了解。
  - 快速处理器：选择高性能的处理器可以提高系统的响应速度和处理能力。例如，基于**ARM Cortex-R**系列的微控制器，能够满足严格的实时性要求，具有硬实时（Hard Real-Time）设计，提供高性能、可靠性和可扩展性，被广泛应用于汽车、工业控制、医疗设备等领域。

    实时操作系统（RTOS）：采用实时操作系统可以提高系统的响应速度和稳定性。实时操作系统可以提供任务调度、中断处理、内存管理等功能，以确保系统的实时性和可靠性。例如，**FreeRTOS和uC/OS**是一款轻量级的实时操作系统，适用于各种嵌入式系统，能够提供快速、稳定、可靠的实时性能。

    快速存储器：选择快速的存储器可以提高系统的数据处理速度和数据传输速度。例如，使用SRAM可以比使用DRAM更快地处理数据，SRAM的读写速度比DRAM快得多，而且不需要刷新操作。使用高速的闪存可以提高数据传输速度，例如，**UFS（Universal Flash Storage）**是一种高速闪存，具有更高的传输速度和更低的延迟，适用于高速数据传输的场景。

    快速总线：选择高速的总线可以提高系统的数据传输速度和响应速度。例如，使用USB 3.0可以比使用USB 2.0更快地传输数据，USB 3.0的传输速度可以达到5Gbps，是USB 2.0的10倍以上。使用**PCI Express**可以提供更高的带宽和更快的数据传输速度，PCI Express的传输速度可以达到32Gbps，是PCI的4倍以上。

    实时监测和控制电路：使用实时监测和控制电路可以提高系统的实时性和稳定性。例如，使用实时监测电路可以实时监测电压、温度、湿度等参数，以保证系统的稳定性和可靠性。使用实时控制电路可以实时控制电流、电压、功率等参数，以确保系统的安全性和可靠性。例如，**电源管理芯片（PMIC）**可以实时监测和控制电源输出，以保证电源的稳定性和可靠性。
- **ARM和x86是两种不同的处理器架构，硬件架构上有很多不同之处。**
  - 架构：ARM架构由英国公司**ARM Holdings**开发，而x86架构由**英特尔（Intel）**和**AMD（Advanced Micro Devices）**等公司开发。
  - 指令集：ARM和x86采用不同的指令集。**ARM处理器使用RISC**（Reduced Instruction Set Computing）指令集，而**x86处理器使用CISC**（Complex Instruction Set Computing）指令集。与CISC指令集相比，RISC指令集更加简单，指令长度一般为固定长度，执行速度更快。
  - 思想：**ARM架构的设计思想是低功耗、高效率和可移植性**，因此它广泛应用于**移动设备和嵌入式系统。**而**x86架构的设计思想则是高性能、多功能和兼容性**，因此它主要用于**个人电脑和服务器**等领域。
  - 寄存器：**ARM处理器拥有16个通用寄存器**，而**x86处理器则拥有8个通用寄存器。**此外，ARM处理器还有一些专用寄存器，如程序计数器（PC）和状态寄存器（CPSR）等。
  - 栈：**自由选择：Arm（但个别指令仅支持向低）；而在x86处理器中，栈则是从高向低地址增长的。**

arm和x86区别：

1、追求不同。X86主要追求性能，但会导致功耗大，不节能，而ARM则是追求节能，低功耗，但和X86相比性能较差。

2、领域区别。ARM主要应用于移动终端之中，类如手机，平板等，而X86则是主要应用于Intel，AMD等PC机，X86服务器中。

3、本质区别。X86使用CISC(ComplexInstructionSetComputer，复杂指令集计算机)，ARM使用RISC(ReducedInstructionSetComputer，精简指令集计算机)，ARM英文全称AdvancedRISCMachine。

- 4、与RISC的不同。C是复杂指令集CPU，指令较多，因此使得CPU电路设计复杂，功耗大，但是对应编译器的设计简单。

5、典型代表。X86结构主要是Intel、AMD等PC电脑；ARM主要是移动终端，IBM的PowerPC。

6、扩展能力不同。X86结构的电脑采用“桥”的方式与扩展设备进行连接，而且x86结构的电脑出现了近30年，其配套扩展的设备种类多、价格也比较便宜，所以x86结构的电脑能很容易进行性能扩展，如增加内存、硬盘等。ARM结构的电脑是通过专用的数据接口使CPU与数据存储设备进行连接，所以ARM的存储、内存等性能扩展难以进行，所以采用ARM结构的系统，一般不考虑扩展。基本奉行“够用就好”的原则。

7、功耗不同。ARM是为了低功耗设计，而x86则是为了高性能。而功耗会影响稳定性、散热成本、产品体积及续航能力等太多方面。

- ARM有七种工作模式，分别是**用户模式、系统模式、监管模式、服务模式、中断模式、异常模式和未定义模式**。每种模式有不同的特点和限制，需要清楚了解。
  - 用户模式（User Mode）：用户模式是处理器最常使用的模式，**它只能访问受保护的指令和数据，不能直接访问操作系统的资源**。这种模式适用于普通应用程序的运行。
  - 系统模式（System Mode）：系统模式是处理器最高的特权模式，它可以**执行所有指令**，包括特权指令和访问操作系统的资源。这种模式适用于操作系统内核的运行和系统级别的任务。
  - 监管模式（Supervisor Mode）：监管模式是类似于系统模式的特权模式，但权限更低。它可以执行一些特权指令和访问一些受保护的资源。这种模式适**用于操作系统内核的子系统或驱动程序的运行**。
  - 服务模式（Service Mode）：服务模式是一种特殊的模式，**用于支持调试和性能分析工具的使用**。它可以执行一些特殊的指令，以便进行调试和性能分析。
  - 中断模式（Interrupt Mode）：当处理器接收到中断信号时，会进入中断模式。在该模式下，处理器**会执行中断处理程序，以响应中断事件**。
  - 异常模式（Abort Mode）：当处理器发生异常事件时，会进入异常模式。在该模式下，处理器会执行异常处理程序，以处理异常事件。异常事件包括指令或数据存取错误、未定义指令、软件中断等。
  - 未定义模式（Undefined Mode）：当处理器执行未定义的指令或指令序列时，会进入未定义模式。在该模式下，处理器会停止执行，并触发异常事件。
- 嵌入式系统中的操作系统可以分为实时系统和非实时系统，带操作系统和不带操作系统等不同类型。
  - 强实时操作系统一般用于对实时性要求较高的场景，软实时操作系统则对实时性要求相对较低。
  - μCOSII是一个简单的实时操作系统，有休眠、就绪、运行、中断、挂起等状态。不同状态之间可以相互转换，需要清楚了解转换规则。
- 当操作系统主程序被中断时，会进入中断状态，任务恢复后可能会发生状态转换。之前的高优先级任务可能会由挂起状态变为就绪状态，需要注意优先级的调度。

- μCOSII中有64个任务，其中有两个缺省任务，一个是最低优先级的空闲任务，一个是统计任务，可以删除。
- 任务优先级的高低可以通过优先级任务组和就绪表来判断，不同优先级任务的就绪情况可能不同。
- 操作系统管理每个任务的任务控制块，通过任务控制块的指针列表和优先级来找到任务本身的堆栈地址和代码段地址。

- 优先级反转是指高优先级任务等待低优先级任务所占用的资源，导致低优先级任务优先级变高，从而影响高优先级任务的执行。可以通过使用信号量或者禁止抢占的方式来消除优先级反转，保证高优先级任务不会被低优先级任务所阻塞。
- ARM处理器有7个异常类型，分别是复位异常、未定义指令异常、软件中断异常、抢占中断异常、数据中止异常、指令中止异常和外部中断异常。
- 每个异常类型都有一个中断向量地址，中断向量地址是一个字，放的是对应异常的中断服务程序的入口地址。当某个异常发生时，处理器会根据对应的向量地址跳转到中断服务程序的入口地址。
- 在ARM中，每条指令都包括取指周期、译码周期和执行周期。当发生异常时，可能还会包括中断周期。指令存储在内存中，处理器需要从内存中取指令并进行译码和执行操作，其中取指令的过程需要花费时间。
- ARM架构大量使用寄存器，操作数可以来自于指令本身（立即数寻址）或者寄存器（寄存器寻址）。
- 在指令执行过程中，ARM不需要访问内存，因此处理速度较快。
- 当指令执行完毕时，CPU会自动查询外部设备是否有中断请求，如果有且CPU允许中断，那么CPU会响应中断并执行中断服务程序。此时指令执行过程中可能会增加一个中断周期。
- 对于同一台计算机上的同一段程序，在不同的时刻执行同一条指令时，可能会出现中断周期的差异。中断周期是由中断请求引起的，可能会影响指令的执行。
- 在x86汇编中，中断周期需要保存断点、关中断、找到中断服务程序入口地址，通常使用push指令将CS、IP和PSW等信息保存到内存中。由于push指令涉及到内存的操作，因此中断在x86中的速度较慢。
- 在ARM架构中，每种异常模式都有自己私有的寄存器，如链接计数器LR和状态保存寄存器SPSR等。可以使用LR寄存器保存被中断的那条指令的下一条指令地址，使用SPSR保存中断前的程序状态。由于状态的断点是从寄存器到寄存器，因此ARM在实现中断引起的过程中不需要访问内存，速度较快。
- 异常是指在指令执行的过程中发生的中断，需要在CPU内部执行一些微操作来响应异常。
- 在中断周期中，需要进行一系列操作来保存断点。具体步骤如下：
    1. 使用当前异常模式下的LR寄存器保存被中断程序的下一条指令地址。
    2. 使用当前模式下的SPSR寄存器保存被中断程序工作段的CPSR。
    3. 将CPSR的模式位修改为当前模式（如普通中断模式、快速中断模式、软中断模式等）。
    4. 将状态设置为ARM状态。
    5. 关闭普通中断，如果当前中断是快速中断，则需要关闭快速中断。
    6. 将PC指向中断服务程序对应的中断向量地址，在中断向量地址中取中断服务程序的入口地址。
- 这些操作需要使用汇编语言进行描述，可以使用伪指令等方式来实现。
- 在硬件设计方面，需要对所使用的ARM芯片的架构有比较清晰的了解。
- 所学的ARM芯片为ARM 7TDMI S3C44B0X，具有32位地址线，可以接4 GB内存空间。
- 由于嵌入式系统一般用不了4GB的内存空间，因此实际上只能够引出256M空间，**分为八个块**，每个块大小为32M字节。
- 每个块可以分为SRAM、DRAM、FLASH等类型，总线的宽度和速度可以通过存储器控制器进行调试和配置。
- ARM架构是基于RISC的，它的IO指令和内存是共享空间的。所有IO的配置、存储器控制器、中断控制器以及IO设备的配置都是通过第一块里面最高的4M的寄存器的地址进行配置。
- 存储器控制器和中断控制器是嵌入式系统中重要的组成部分。
- 存储器控制器是一个硬件模块，用于管理嵌入式系统中的存储器，包括内部RAM和外部存储器。它可以控制存储器的读写、时序等操作。
- 中断控制器也是一个硬件模块，用于管理嵌入式系统中的中断。中断控制器通常包括一个主模块和多个从模块，可以控制中断的优先级、向量中断或非向量中断等。
- 对于普通中断，可以通过中断控制器将其扩展到多个优先级进行配置，以便更好地管理中断。此外，还可以将中断设置为向量中断或非向量中断，具体取决于嵌入式系统的需求。

- I/O方面的基本概念包括发送和接收。RS 232C是全双工的。发送和接收都可以引发中断。串口通常使用中断方式处理单字节数据。
- I2C是用于系统间信息交互的通信协议。它只有两根线：SDA和SCL。多个主设备可以通过这两根线实现半双工通信。在同一时刻，只能进行发送或接收操作。在使用I2C总线进行通信时，需要确保总线空闲，并避免竞争。
- 当主设备要访问从设备时，需要先给出从设备的地址。从设备地址是七位。第8位决定读或写。在使用I2C总线时，需要明确七位地址高位在前还是低位在前。I2C总线的工作原理及其如何知道总线是否有竞争，如何知道自己是否与其他设备存在竞争，以及如何获取和放弃总线使用权都需要清楚了解。
- 看门狗是一种监控系统正常运行的机制。可以通过中断或复位来恢复正常。通常将看门狗放置在优先级最低的任务中，以确保其正常工作，并衡量任务调度是否达到最优状态。
- ARM有七种模式，其中包括用户模式、系统模式、监管模式、中断模式、快速中断模式、抢占式模式和未定义模式。此外，ARM还有两种状态：ARM状态和THUMB状态，可以用不同的编程方式进行编程。其中THUMB编程相对于ARM编程来说更加简单，适用于一些资源有限的系统。
- 程序阅读题：
    - 阅读给定的程序代码，并在代码中加入注释，解释程序的功能和结果。
    - 分析程序中使用的指令，以确定程序的作用和功能。
    - 总结就是语句语法含义和整体模块含义
    - **涉及到C调汇编、汇编调C的问题，因此需要清楚参数传递和返回的规则。**
- 系统设计题目：
    - 根据给定的功能要求，设计出一个满足要求的嵌入式系统。
    - 考虑硬件和软件两个方面
        - 对于硬件方面，需要**画出硬件方块图，以展示系统的结构和组成部件**；
        - 对于软件方面，需要**画出流程图或者使用伪指令进行描述，以展示系统的运行流程和功能实现方式。**
        - 在设计过程中，还需要考虑如何**利用所学知识去实现一些改变的功能。**