1. e概念
2. 普适计算

一种人们能够在任何时间、任何地点、以任何方式进行信息的获取与处理的计算；是强调和环境融为一体的计算；在普适计算模式下，计算机本身则从人们的视线中消失。

1. 嵌入式系统

嵌入式系统的定义有两种方式：

外包法：嵌入式系统是以应用为中心，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、功耗严格要求的专用计算机系统。

内涵法：嵌入式系统是以提高对象体系智能性、控制力和人机交互能力为目的，通过相互作用和内在指标评价的，嵌入到对象体系中的专用计算机系统。

1. 嵌入式操作系统

从用途上说，嵌入式操作系统是针对行业或一类应用定制的，具备封装、裁剪、融合特征的专用操作系统，应用在具有嵌入式系统的设备上。 从结构上讲，嵌入式操作系统是介于下层硬件、引导层和上层系统软件、应用软件之间的系统软件，管理硬件资源，并同时为上层应用提供可靠、高效的接口。

1. 嵌入式微处理器

嵌入式微处理器是指具备强的中断、IO、内存和能耗管理能力，具有定制多类体系架构的特征，适当计算处理性能的微处理器。

1. 傅氏变换与拉氏变换

傅立叶变换是一种分析信号的方法，它可分析信号的成分，也可用这些成分合成信号。

拉氏变换是一种为简化计算而建立的实变量函数和复变量函数间的一种函数变换。  
定义：  
f(t)是t的周期函数，如果t满足狄里赫莱条件：在一个以2T为周期内f(X)连续或只有有限个第一类间断点，附f（x）单调或可划分成有限个单调区间，则F（x）以2T为周期的傅里叶级数收敛，和函数S（x）也是以2T为周期的周期函数，且在这些间断点上，函数是有限值；在一个周期内具有有限个极值点；绝对可积。则有下图①式成立。称为积分运算f(t)的傅立叶变换，  
②式的积分运算叫做F(ω)的傅立叶逆变换。F(ω)叫做f(t)的像函数，f(t)叫做  
F(ω)的像原函数。F(ω)是f(t)的像。f(t)是F(ω)原像。

1. 数学与控制数学模型

数学模型是指人们为一定的目的，在一定的假设条件下，利用字母、数字、图表、图像、框图、结构图、数理逻辑等来描述系统特征及其内部联系与外界联系的模型。

控制数学模型：控制系统在工作状态时，描述系统变量间相互关系的动态性能的运动方程

1. Java虚拟机

虚拟机是类似于计算机的一个软件程序。它会像真正硬件处理器那样取出并执行程序指令，但两者的区别在于，虚拟机的指令执行过程发生在软件级而不是硬件级，即指令是由软件而不是硬件执行的。Java虚拟机可以定义为：执行经过编译的Java目标代码的计算机实现。

1. 最小系统

使一个含有微处理器的系统具有I/O、中断处理、基本计算能力等制造功能与用户功能，所需的最少的硬件及软件环境。（一般可认为，引导层和硬件层为一个最小的系统，或认为引导层、硬件层和操作系统内核为一个最小系统。）

1. 进程和线程

进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动。

线程是进程中的一个相对独立的实体，是被系统独立调度和分派的基本单位。

1. SDK

Soft Develop Kit,即软件开发工具包。一般是一些被软件工程师用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合。

1. 实时系统

实时系统是指计算结果的正确性不仅依赖于计算结果的逻辑正确性，还依赖于计算结果产生时间的计算机系统。（应用于嵌入式系统中，有能力在响应时间的边界内，提供满足要求的服务的OS）。

1. 云端系统

是指为能够通过连接云服务器扩展终端自身运算能力、存储能力或功能的软件或软硬件专用计算系统。(通过云端计算，可以更好的连接终端用户和云服务，应对多变的应用场景和网络环境，提供持续有效的服务，改善用户体验。）

1. 系统可靠性

系统可靠性是指系统在规定条件下和规定时间内，完成规定功能的能力。

1. 系统失效、错误与缺陷

错误（error）：计算机计算得到的、观察到的、测量到的数值或者条件和理论上得到的正确的数值或者条件之间存在的差异。

　　缺陷（defect）：程序或者软件中不正确的步骤、过程或者数据定义等。比如错误的语句或者错误的标量定义等。缺陷是错误的具体表现，可以是不正确的文档、程序段以及指令或者数据定义。

　　失效/失败（failure/fail）：软件系统或单元无法实现需求文档中规定的功能特性或者非功能特性。或者说单元/系统产生的结果与期望交付的服务或者结果存在偏差。外部的失效/失败是内部缺陷在执行测试软件时的外部反映。它是规范说明的期望值与实际观察到的值、现象等存在偏差。比如不正确的系统反应、系统崩溃、系统死机等等。

1. 系统缺陷静止、活跃与激活

静止：缺陷存在但不引起错误。

激活：一个输入使缺陷由静止态转为活跃态。

活跃：缺陷引发错误。

1. 物联网

是以普适计算作为理论基础，以互联网、传统网等为基础，以广义物前端感知为源，由应用驱动的，以综合提高可靠和效率、服务品质、降低能耗、增强大众用户体验为目的的规模化、设施化、公共基础化广义物联系统。

1. 开源许可证

许可证即授权条款。开源软件并非没有限制。最基本的限制就是开源软件强迫任何使用和修改该软件的人承认发起人的著作权和所有参与人的贡献。任何人拥有可以自由复制、修改、使用这些源代码的权利，不得设置针对任何人或团体领域的限制。不得限制开源软件的商业使用等。而许可证就是这样一个保证这些限制的法律文件。

1. 简答题
2. 简述嵌入式系统的主要应用领域？说明云端组织计算的特征？

嵌入式系统有五大应用领域：国防电子、汽车电子、医疗电子、机床电子、消费电子。比如在无线传感网络、火炮控制、精确制导、电子侦察以及电视机、冰箱、微波炉等领域和产品中都大量的使用嵌入式系统。（嵌入式系统的主要应用领域包括工业、军事、通信、运输、金融、医疗、气象、农业等。）

1）基于虚拟化技术快速部署资源或获得服务；

2）实现动态的、可伸缩的扩展；

3）按需求提供资源、按使用量付费；

4）通过互联网提供、面向海量信息处理；

5）用户可以方便地参与；

6）形态灵活，聚散自如；

7）减少用户终端的处理负担；

8）降低了用户对于IT专业知识的依赖。

1. 普适计算内涵思考是什么？普适计算的颠覆性体现在哪几个方面？如何理解穿戴设备是普适计算的最好诠释之一？

（1）内涵思考

基本思想：把计算机融入到环境中去，使人们关注的重点从操作工具转移到执行任务本身上来，可以在任意时间、使用任意设备、通过任意网络来获得所需的服务。

“消失”的技术：正如“书写”和“电”一样，由于这两种技术使用的普遍性导致它们变得非常平淡，以至忘记了这些技术对日常生活的巨大影响，犹如消失于人们的日常生活中。普适计算也将是那样一种消失的计算。

“不可见”的工具：一种好的工具是“不可见”的，它并不进入你的意识，你只是专注于任务而并非工具。普适计算也将是那样一种“不可见”的工具。

（2）颠覆性体现在：

简单而方便使用；

不要求人的注意力；

不隔绝于他人和现实活动；

（3）如何理解穿戴设备是普适计算的最好诠释之一？

可穿戴设备无障碍地享用计算能力和信息资源，它实现了信息空间和物理空间的融合。在这个融合的空间中，人们可以随时随地，透明地获得数字化的服务。其中“随时随地”是指人们在工作、生活的现场就可以获得服务，而不需离开这个现场去端坐在一个专门的计算机面前。而“透明”是指获得这种服务不需要花费很多注意力，即这种服务的访问方式是十分自然的甚至是用户本身注意不到的。这些都很好的诠释了普适计算。

1. 结合VxWork的成长路线图，如何理解嵌入式系统发展历史简介？

最早期的嵌入式系统没有操作系统，仅仅是将算法嵌入，通过A/D和D/A转换与外界通信；操作系统引入嵌入式开发后，系统拥有较大的灵活性，但此时嵌入式操作系统仅是内核，功能较少，移植性较差；后来嵌入式操作系统逐渐发展，可对一些关键设备进行实时监控；随着技术的提高，对设备的智能化有了更高的要求，实时性仅是对嵌入式系统最基本的要求，网络化才标志着设备智能度的提高；另外，为了降低成本，占有市场，各个操作系统厂商都努力提供一种通用的系统。

1. 嵌入式系统物理结构和逻辑架构有什么区别？系统功能与性能事物区别？

嵌入式系统的物理结构指的是看得到摸得着的物理上的一种结构，是各种元器件、连线、接口、程序、数据等实体性的元素按其联系的组合。物理结构是物质的，是整个系统的基础，是逻辑抽象的依据。

而逻辑架构则是非实体性的，更类似于意识层面上的结构，是人们抽象出来的，用于描述和抽象物理结构和实体运行过程的一种描述。逻辑结构一般用图形的描述方式体现系统运行的逻辑关系或者是系统组成间的逻辑关系的模型。

系统功能是指系统具有的完成任务的能力，是一种行为状态，是反映系统与外部环境的关系，表达出系统的性质和行为。

性能则是对系统的一种或一类功能的量化指标，是表明系统功能好坏强弱的参数。

1. 微处理器指令系统的基本要求是什么？指令系统CISC与RISC的优缺点？
2. 完备性。完备性是指用汇编语言编写各种程序时，指令系统直接提供的指令足够使用。

2. 有效性。有效性是指利用该指令系统所编写的程序能够高效率地运行。

3. 规整性。规整性包括指令系统的对称性、指令格式和数据格式的一致性。

4. 兼容性。兼容性要求要能做到“向上兼容”，即在低档机上运行的软件可以在同系列高档机上运行。

CISC：  
–优点：指令越多功能越强，强调代码效率，容易和高级语言接轨。可以对存储器直接操作，实现从存储器到存储器的数据转移，可加入DSP指令。  
–缺点：指令太多不易记忆；CPU内部结构复杂造成频率不高；指令执行速度慢。  
RISC：  
–优点：指令少容易记忆，尽量将操作码和操作数用1个16位数或32位数表示，指令整齐。CPU时钟频率可以做得很高，指令执行速度快。  
–缺点：同样功能的程序，产生的代码量比较大；不能对存储器直接访问，不能实现存储器到存储器的数据转移。

1. ARM系列嵌入式处理器的特点是什么？简述ARM的七种运行模式和ARM和Thumb状态？

（RISC的特点：指令条数和寻址方式少；单一机器周期操作；存取操作限制；简单固定的指令结构；指令流水线；高速度、高可靠性、低成本。ARM微处理器除具有RISC的特点外还具有：

1].在每条数据处理指令当中，都控制算术逻辑单元（ALU）和移位器，以使ALU和移位器获得最大的利用率；

2].自动递增递减的寻址模式，以优化程序中的循环；

3.]同时Load和Store多条指令，以增加数据吞吐量；

4].所有指令都条件执行，以增大执行吞吐量。）

1]、体积小、低功耗、低成本、高性能；

2]、支持Thumb（16位）/ARM（32位）双指令集，能很好的兼容8位/16位器件；

3]、大量使用寄存器，指令执行速度更快；

4]、大多数数据操作都在寄存器中完成；

5]、寻址方式灵活简单，执行效率高；

6]、指令长度固定；流水线处理方式。

1].用户模式（usr）：ARM处理器正常的程序执行状态；

2].快速中断模式（fiq）：用于高速数据传输或通道管理；

3].外部中断模式（irq）：用于通用的中断处理；

4].管理模式（svc）：操作系统使用的保护模式；

5].数据访问终止模式（abt）：当数据或指令预取终止时进入该模式，用于虚拟存储及存储保护；

6].系统模式（sys）：运行具有特权的操作系统任务；

7].未定义指令中止模式（und）：当未定义指令执行时进入该模式，可用于支持硬件协处理器的软件仿真。

ARM体系结构在V4T及其以上版本定义了称为Thumb指令集的16位指令集。Thumb指令集的功能是32位ARM指令集的功能子集。Thumb指令集在性能和代码大小之间提供了出色的折中。

正在执行Thumb指令集的处理器是工作在Thumb状态下的。同样，正在执行ARM指令集的处理器是工作在ARM状态下。ARM状态下的处理器不能执行Thumb指令，在Thumb状态下的处理器也不能执行ARM指令。必须确保处理器不接受对当前状态来说为错误指令集的指令。每个指令集都包括切换处理器状态的指令。ARM处理器总是在ARM状态下开始执行代码。

1. 简述ARM V1-V7版本特征？说明ARM处理器结构的六大特征？

ARM版本Ⅰ： V1版架构   
 该版架构只在原型机ARM1出现过，只有26位的寻址空间，没有用于商业产品。

ARM版本Ⅱ： V2版架构   
 该版架构对V1版进行了扩展，例如ARM2和ARM3（V2a）架构。包含了对32位乘法指令和协处理器指令的支持。 版本2a是版本2的变种，ARM3芯片采用了版本2a，是第一片采用片上Cache的ARM处理器。同样为26位寻址空间，现在已经废弃不再使用。   
ARM版本Ⅲ ： V3版架构   
 V3版架构（ 目前已废弃 ）对ARM体系结构作了较大的改动，比如：寻址空间增至32位（4GB）；增加了程序状态保存寄存器SPSR；增加了从异常处理返回的指令功能等。

ARM版本Ⅳ ： V4版架构   
 V4版架构在V3版上作了进一步扩充，。V4不再强制要求与26位地址空间兼容，而且还明确了哪些指令会引起未定义指令异常。   
ARM版本Ⅴ ： V5版架构   
 V5版架构是在V4版基础上增加了一些新的指令。这些新增命令有：计数前导零CLZ指令，BRK中断指令，带有链接和交换的转移BLX指令等。

ARM版本Ⅵ ： V6版架构   
 V6版在降低耗电量地同时，还强化了图形处理性能。通过追加有效进行多媒体处理的SIMD(Single Instruction, Multiple Data，单指令多数据 )功能，将语音及图像的处理功能提高到了原型机的4倍。

ARM版本Ⅵ ： V7版架构

V7采用了THUMB-2技术和NEON技术，支持改良了浮点运算，支持动态编译器。  
Von & Havard

Little & Big

ARM &Thumb

DSP & Javazell

Secure & Mali

Sys & Debug

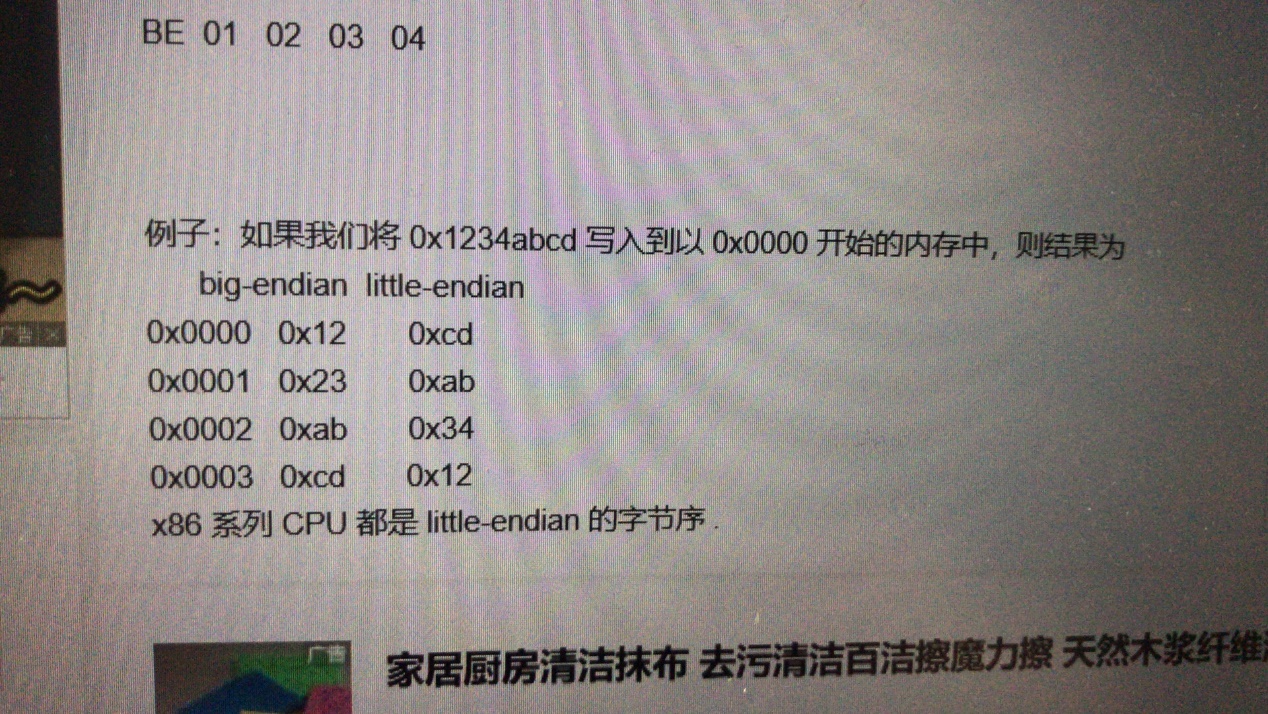
1. 简述litte&big多字节数据存储方式区别，并以12H，34H，ABH，CDH为例分别示意存放顺序？

big endian是指低地址存放最高有效字节（MSB），而little endian则是低地址存放最低有效字节（LSB）。

12H：0x00010010

Big endian:00 01 00 10

Little endian:10 00 01 00



1. 如何理解公式 P∝CV2F，进而说明嵌入式系统如何进行能耗优化控制的方法？锁相环有何用途？

P∝CV2F使用的前提是在数字集成电路中，CMOS电路静态功耗非常低，与动态功耗相比可忽略不计，故可以写成该形式。公式中，P为CMOS芯片的动态功耗，C为CMOS芯片的负载电容，V为CMOS芯片的工作电压，F为CMOS芯片的工作频率。

通过上式可发现，降低CMOS电路的工作电压比降低时钟频率更能降低活动功耗。因为功耗与电压时指数幂的关系，而与时钟频率只是线性关系。因此，选定满足电子系统中所有电驴模块的最低要求电压，对降低能耗至关重要。

锁相环是一种以相位分割为基础的可编程倍频器，用以调节系统运行时钟，从而控制能耗。

1. 怎样看待复位和开门狗的关系？

手动复位是在有需要时直接给CPU一个信号，使其直接从头开始执行程序，而看门狗是在每次开机或关机时由电容放电使CPU复位，使程序下次运行处于最佳状态，它实质上是一个由MCU进行复位的定时器，提高了系统的抗干扰能力。一般情况下，嵌入式系统同时应用这两种技术。

（看门狗即监控定时器，在嵌入式系统内部使用看门狗技术是一种防止尖峰脉冲干扰的有效方法。当侵入的尖峰脉冲干扰使程序编码的一位（或数位）发生改变时，程序所呈现的外在表现可能为“飞掉”，此时，利用看门狗技术可帮助系统自动恢复正常运行。）

1. PC BIOS和ISA总线在PC产业的发展过程中起到了哪些作用？为什么嵌入式系统中也加入了BIOS环节？

PC结构中的BIOS的出现产生了具有重大意义的变革，它促使了硬件与软件就得以分离，使得PC软件得以独立发展，而且由于有了BIOS，增加了I/O扩展槽，才有了外部总线，促使I/O设备得以应用。

ISA总线使寻址范围、数据总线宽度、中断处理能力有了很大的提高，具备了良好的兼容性，为以后更先进总线的诞生打好了基础。

BIOS在嵌入式系统中具有极其重要的作用，主要表现在一下三方面：

1. 自检及初始化。这部分工作是负责启动计算机，主要完成三件事：对硬件进行检测；创建中断向量、设置寄存器、对一些外部设备进行初始化和检测、设置BIOS的参数；引导操作系统。

2. 程序服务处理，这些工作主要是为应用程序和操作系统服务的。服务内容主要与I/O设备有关，例如读磁盘，文件输出到打印机等。

3. 中断处理，在开机时，BIOS会传送所有硬件设备的中断向量号给CPU。当用户发出使用某个设备的指令后，CPU就会根据中断向量号使用相应的硬件来完成工作。

1. 简述U-boot两步启动过程？Stage1&Stage2
2. 第一阶段的功能

Ø  硬件设备初始化

Ø  加载U-Boot第二阶段代码到RAM空间

Ø  设置好栈

Ø  跳转到第二阶段代码入口

1. 第二阶段的功能

Ø  初始化本阶段使用的硬件设备

Ø  检测系统内存映射

Ø  将内核从Flash读取到RAM中

Ø  为内核设置启动参数Ø  调用内核

1. 评述层次和微内核嵌入式操作系统的优缺点？

微内核结构：

优点：结构规范，易于裁剪、编程和调试；扩展和移植性强；降低内存固定使用量；实时性强。

缺点：整体效率低；资源管理复杂。

层次内核体系结构：

优点：各模块之间的组织结构和依赖关系清晰明确，适合进行系统功能的划分；分层思想有利于组织操作系统的开发、调试和诊断，也使得系统的修改和扩展相对容易。

缺点：对某一层功能的修改往往影响到相关层次，使得指定和裁剪系统功能相对困难；底层代码和硬件直接相关，功能越强，关系越紧；与微内核结构相比比较庞大，移植较复杂。

1. 简述指令集 CISC&RISC 与 OS 进程&线程，在提高嵌入式系统并行性上的价值？

无论指令集还是进程线程，都是计算机基础资源分配的最小单位，其单位粒度的大小，直接影响并行程度。

1. 怎样评测嵌入式操作系统的实时性？

实时性能指标包括任务切换时间、中断延迟时间和系统响应时间。

任务切换时间：该指标主要受微处理器控制，在进行任务切换时，保存和恢复CPU所花费的时间与CPU的寄存器个数有关。

中断延迟时间：由最大中断屏蔽时间，得到只想ISR（中断服务例程）向量的时间、存储所有寄存器的时间和执行中断处理的时间。

系统响应时间：该时间是系统在发出处理要求到系统给出应答信号的时间，从整体上评价操作系统。

实时性测量方法：Rhealston方法、进程调度间隙时间法、三维表示法、Real/Stong Benchmark方法。

方法说明：

Rhealstone方法（六元素测量法）：对EOS的六个关键操作的时间量进行评测，即任务切换时间，抢占时间，中断延迟时间，信号量混洗时间，死锁解除时间、数据包吞吐率。

　　进程调度间隙时间法：从中断产生到由中断激活的实时任务开始执行之间的时间间隔。

三维表示法：测试EOS的计算能力、中断处理能力、I/O吞吐率。然后可用一个三维图形来表达三个特性之间的依赖关系。

Real/Stong Benchmark方法：为一个纯软件的基准程序，属于合成工作负载，包含了系统响应能力、系统抢占能力和系统I/O吞吐能力三个测试。

1. 简述Linux内核与Android之间的关系？

Android（安卓)本质上是一个基于 Linux 内核上面运行的 java 虚拟机，实际上就是一个解释程序。它相当于一个应用程序，应用程序要运行需要一个平台，这个平台是Linux内核，它俩的关系就是一种依赖关系。

1. 如何理解Java语言的安全性、程序兼容性、JVM起到了什么作用？

Java语言的安全性可以从两个方面得到保证。

一方面在Java语言里，象指针和释放内存等C++功能被删除，避免了非法内存操作。

另一方面当Java用来创建浏览器时，语言功能和一些浏览器本身提供的功能结合起来，使它更安全。

Java语言在机器上面执行前，要经过很多次的测试。它经过代码校验、检查代码段的格式、检测指针操作、对象操作是否过分以及试图改变一个对象的类型。

虚拟机是由一套完整的规范定义的。JVM保证了Java代码在任何系统上到能运行。凡是符合Java虚拟机规范的实现都是百分之百兼容的。JVM规范对其实现做出了具体规定。JVM规范定义了虚拟机体系结构、支持的数据类型和类结构文件。

1. 怎样理解基于IOS编程的HIG有关UI方面的5项准则要求？

（1）一致性原则

坚持以用户体验为中心设计原则，界面直观、简洁，操作方便快捷，用户接触软件后对界面上对应的功能一目了然、不需要太多培训就可以方便使用本应用系统。

（2）准确性原则

使用一致的标记、标准缩写和颜色，显示信息的含义应该非常明确，用户不必再参考其它信息源。

（3）布局合理化原则

在进行UI设计时需要充分考虑布局的合理化问题，遵循用户从上而下，自左向右浏览、操作习惯，避免常用业务功能按键排列过于分散，以造成用户鼠标移动距离过长的弊端。多做“减法”运算，将不常用的功能区块隐藏，以保持界面的简洁，使用户专注于主要业务操作流程，有利于提高软件的易用性及可用性。

（4）系统操作合理性原则

尽量确保用户在不使用鼠标(只使用键盘)的情况下也可以流畅地完成一些常用的业务操作，各控件间可以通过Tab键进行切换，并将可编辑的文本全选处理。

（5）系统响应时间原则

系统响应时间应该适中，响应时间过长，用户就会感到不安和沮丧，而响应时间过快也会影响到用户的操作节奏，并可能导致错误。

1. 可测性设计对于嵌入式系统设计有什么帮助？

可测性设计是为了降低电路测试费用和提高故障覆盖率，在电路设计阶段对原电路进行修改，但不影响其性能的设计方法。嵌入式系统设计中需要采用结构化设计方法，将系统划分为几个模块，而每个模块都需要进行测试以满足要求，引入可测性设计可以保证系统的可靠性和稳定性。为保证测试的有效性，驱动的开发需要有一定的相互独立性。系统开发达到一定程度时，可进行整机测试。而且软硬件测试的有机结合使得嵌入式系统的设计更加方便、可靠。

1. 为什么系统设计中，可测性设计对于复杂系统非常重要？

可测性设计是为了降低电路测试费用和提高故障覆盖率，在电路设计阶段对原电路进行修改，但不影响其性能的设计方法。复杂系统中软件的开发往往是与相应的硬件设备同步进行的，硬件的可靠性与软件的运行息息相关，引入可测性设计可以保证系统的可靠性和稳定性；复杂系统设计常将系统划分为几个模块，而每个模块都需要进行测试，引入可测性设计对每个模块进行测试从而简化了复杂系统的设计。

1. 软件测试基本机理是什么？复杂的软件系统能够保证没有Bug吗？嵌入式软件的基本质量是什么？
2. 机理：软件测试是使用人工或自动的手段测试某个软件系统的过程。目的在于检验它是否满足规定的要求或弄清预期结果与实际内容之间的差别。

（软件测试的方法包括白盒测试和黑盒测试。白盒测试：按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常运行，检测程序中的每条通路是否都按照预定要求工作。黑盒测试：也称功能测试，是通过测试来检测每个功能是否都能正常使用，完全不考虑程序内部结构和内部特性。）

1. 不能保证。测试程序是不能发现所有Bug的，只能将软件系统中可能存在的Bug找出来，但是并不是所有的Bug都能被发现。但没有致命的BUG存在，模拟的不完整。
2. 嵌入式软件的基本质量：可靠性、剪裁性、兼容性、安全性（另一种答案：安全性足够高，系统长时间可靠、实时性）。
3. GIT系统的基本内涵？

Git 是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项 目。Git 是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。Git是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的 一个开放源码的版本控制软件。

1. 简述嵌入式系统的基本设计流程？

系统需求分析-----体系结构设计-----硬件、软件设计-----软硬件集成-----系统集成

1. 描述评价嵌入式操作系统的10大方面？

十个方面分别是时域、资源占用、能耗、干扰、智能、控制、融合、封装、功能性、可靠性。

时域：系统完成规定任务所需要的时间必须满足的条件。

资源占用：系统在正常运行的各个阶段，所需的硬件资源的种类，数量及多少。

能耗：系统在运行的各种状态下所消耗的电能的速率。

干扰：嵌入式产品在加入了嵌入式系统后，对原有产品性能及其他指标的影响程度。

智能：统能够完成的操作种类多少、解决问题的复杂程度以及自我完善、更新、发展的能力。

控制：系统的控制过程的快速、准确和精细程度以及操作此控制作用的便捷性。

融合：指嵌入式系统和嵌入式产品其他部分相匹配、结合的能力。

封装：简单说，封装就是打包，把一系列相关的东西打包在一起，把细节隐藏，只把简单的、稳定的接口提供给用户。

功能性：拥有多种功能

可靠性：在预定的操作和环境条件下，在一定的时期或是某一时刻，系统专一并正确的执行任务的可依赖程度。

1. 嵌入式的可靠性主要是指哪些属性？几种提高嵌入式系统可靠性的手段各有什么特点，如何选取合适的手段用于系统设计？

可用性：随时可以向使用者提供正确服务。

可信性：可以持续的提供正确服务。

安全性：不会对使用者和环境造成灾难性的损害。

完整性：不会出现意外的系统部分替换。

可维护性：可以进行修改和维修。

如何正确合理的处理缺陷，是提高嵌入式系统可靠性的关键。

处理缺陷的方法：

1. 缺陷避免：避免缺陷出现或被引入系统。
2. 缺陷包容：在缺陷出现时，避免出现系统服务失效。包括错误发现和恢复。
3. 缺陷取出：减少缺陷的数量，降低失效的严重性。
4. 缺陷预测：预测缺陷的数量，分布，可能产生的结果等。
5. 以下面代码为例，说明软件缺陷发生与状态激活的变化？如何抑制该缺陷，做简单说明？

float func(float f1,float f2)

{

float result;

result=1/(f1-f2);

return result;

}

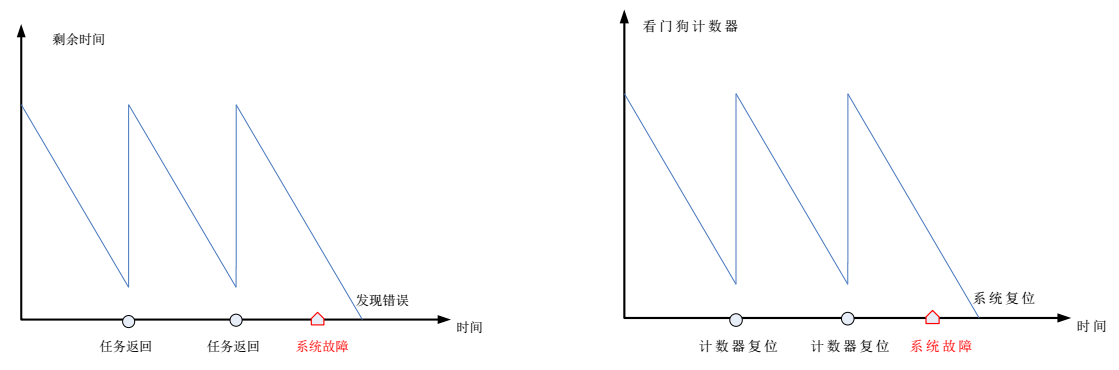
a,确保每个被发现的缺陷都能够被解决；

b,这里解决的意思不一定是被修正，也可能是其他处理方式（例如，在下一个版本中修正或是不修正），总之，对每个被发现的BUG的处理方式必须能够在开发组织中达到一致；

c,收集缺陷数据并根据缺陷趋势曲线识别[测试过程](http://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E8%BF%87%E7%A8%8B" \t "http://baike.baidu.com/_blank)的阶段；决定测试过程是否结束有很多种方式，通过缺陷趋势曲线来确定测试过程是否结束是常用并且较为有效的一种方式；

d,收集缺陷数据并在其上进行数据分析，作为组织的过程财富。

1. 依下图说明时间刻度检查方法用于发现缺陷的原由，并以此说明软件看门狗，对于提高系统可靠性的价值？



嵌入式系统中软件看门狗实现多线程监测以及运用/proc文件系统实现监测数据采集的原理和方法，讨论了使用UML对自监测系统进行面向对象的分析以及设计，并将该方案应用到TI公司的开发板实验。测试结果表明，它可以有效的实现对被监测进程的每一个线程的监测，并且可以对系统的平均负载、内存剩余量和指定接口收发包 情况进行监测，确保系统的长时间的运行。

1. 嵌入式系统优化的三大方面？

（1）等效性原则

优化前后的代码实现的功能要完全一致

1. 有效性原则

优化后的代码一定要比原先的代码运行速度快活着占用存储空间小，或者二者兼有，否则就是毫无意义的优化

（3）经济性原则

很多代码性能低下的部分原因也是由于硬件性能的限制，比如将文件压缩存放以节约存储成本。优化要在现有的条件下考虑，不要以更换存储空间的大小来换取解压的时间。优化要付出较小的代价，很多程序员在做优化的时候，抱怨设备的性能有限，要求提高设备的性能，这样只能是本末倒置。

1. 广义行业物联网，以基础教育物联系统为例，阐明系统的数据聚能价值？

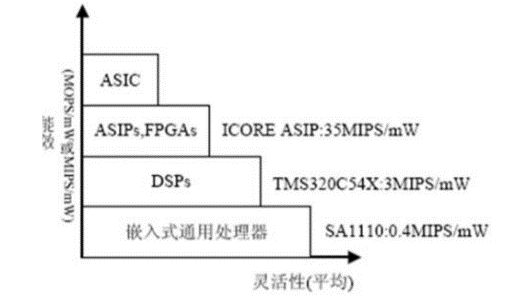
通常建设行业基础垂直的物联网，要考虑到基础赋能问题，否则缺乏系统前进的基本动力。比如基础教育物联系统，通过生态精准批改或生态修订，便可以萃取出学生与学生之间相互学习、帮助的动力，使得基础作业数据聚能上传，并产生精准匹配、分享的基础自动力。

1. 综合思考题
2. 计算载体与计算系统的关系？如何看待手工、机械、电子计算，以及未来的光、生物和量子计算？

计算系统包含计算载体。计算系统还包括在计算载体上运行的程序、数据，以及各部分和部分之间组织起来的设计思想等。计算载体只是承载思想和数据运行的实体而已。

人类的计算方法和工具自古以来不断的发展进步。由一开始的手工、机械计算，到现在的电子计算，而未来将会出现光、生物和量子计算。计算的载体不断发展变化着，一方面说明了人类科学技术的进步，认识和改造世界的能力的加强。我们通过对于客观世界的认识和改造，也在延伸着人本身的能力。另一方面则说明，物质世界的本质具有数学特性，具有一致性，不管工具如何发展，计算的本质并没有发生改变。人类的数学与计算就是一种对于宇宙本质规律的抽象与总结。因此无论载体是否变为光和生物、量子，计算速度不管如何提高，计算的本质是不变的。

1. 解读下图不同类型硬件的能耗级别，说明 INTEL 与 ARM 架构的不同能耗观？生物、 量子计算大量降低能耗？



Intel处理器架构的能耗观是以现有工艺资源追求最大通用计算能力为目标的，ARM则是以完成一定的计算能力，实现能耗最小为目标的。生物、量子计算可以大量降低能耗，生物计算的能量消耗仅相当于普通计算机的十亿分之一。

1. 怎样理解云端组织型嵌入式系统的编程特征？依据下图提示，说出并解释苹果 App 应该审核中 5 个主要方面要求, 如何理解苹果云端类产品的封闭式开放生态特征？



云端系统可以简单的理解为云服务与终端计算并举形成的计算系统。

云计算操作系统，是云计算后台数据中心的整体管理运营系统（也有人认为云计算系统包括云终端操作系统，例如现在流行的各类手机操作系统，这与先行的单机操作系统区别不大，在此不做讨论），它是指构架于服务器、存储、网络等基础硬件资源和单机操作系统、中间件、数据库等基础软件管理海量的基础硬件、软件资源之上的云平台综合管理系统。

1. 一致性原则。
2. 准确性原则。
3. 布局合理化原则。
4. 操作系统合理性原则。
5. 系统响应时间原则。

苹果的iphone和ipad产品都是典型的云端组织型嵌入式产品，其封闭特征是指全球范围内只有一个云支持服务系统（itunes+APPstore），开放是指通过其开发工具平台，及一系列的指南、审查机构，可以有众多的开发机构、开发者在其平台上进行大量的有商业价值的应用开发。

1. 如何理解嵌入式系统软件设计中的自测、优化和体验？

自测：测试编程者一定要按照测试贯穿始终的原则，在设计编程时要安排自测点和自测程序，不要将测试工作完全推到测试环节。

优化：对系统要有精进，不断优化的意识和行动。

体验：要重视用户的使用体验。

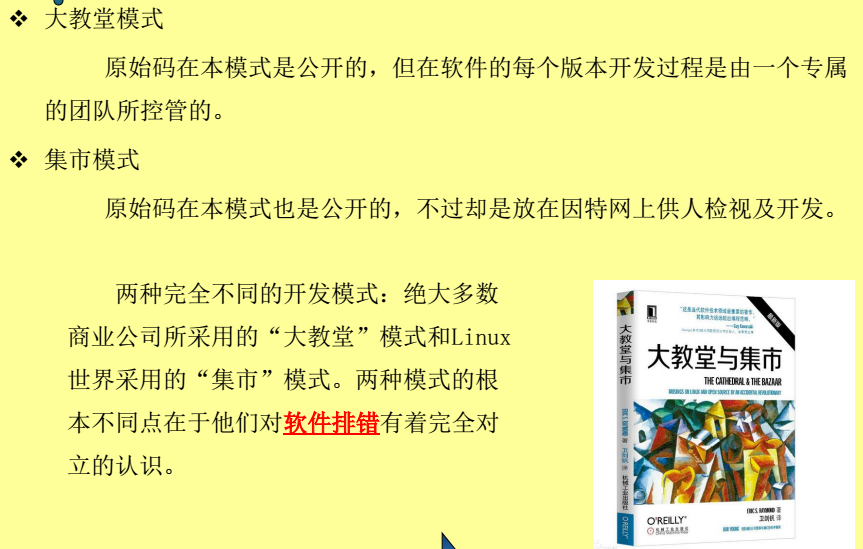
1. 为什么大数据时代更需要直觉思维模式？为什么嵌入式系统可靠性分析和性能优化，不能用结构性思维？

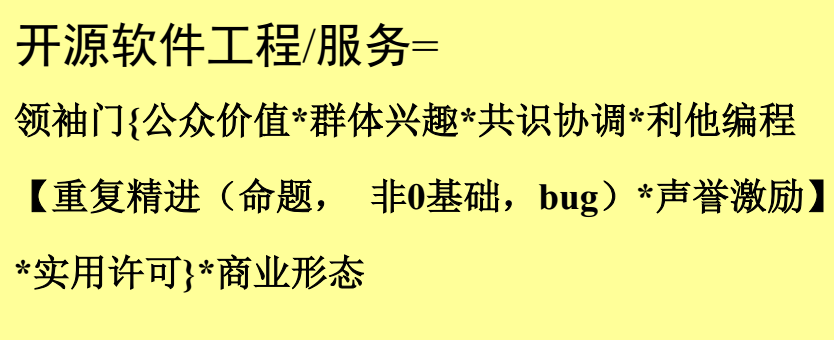
大数据的直觉思维，就是不是用所谓理性的数学和逻辑模式进行的因果思维，而是类似于神经网络的感性模式。

嵌入式系统的可靠性分析和性能优化，都属于嵌入式系统的综合性问题处理，属于在层次和模块化基础上的垂直型思考处理的内容，所以不适合结构性思维。

1. 过本课程RevolutionOS纪录片观看与《大教堂与集市》阅读，说明自由或开源软件中的利他编程，在代码优化及基本质量控制中的作用？

利他编程是一个高层阶的大概念，讲到了编程的目的、价值归宿取向问题，但凡以利他编程驱动的事务，要有更为强烈的行业公共基础价值，其编程测试通常是自发的一体化的，专业化的，而且在合适的时候可以相对频繁的使用重写方式进行，这在利己编程中通常是不可思议的。基本质量可理解为基础质量包括代码组织、版本控制和系统应用的可靠、性能、架构兼容，以及UI等方面



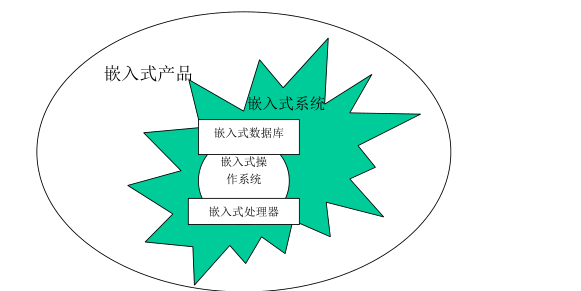


**三、 图解讨论题**

**1． 园环套图概念关系：嵌入式处理器/嵌入式操作系统/嵌入式数据库/嵌入式系**

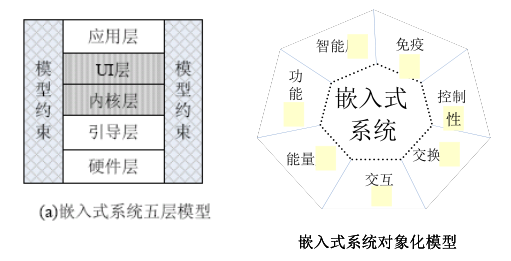
**统/嵌入式产品相互关系**

嵌入式微处理器：指具备强的中断、I/O、内存和能耗管理能力、具有定制多类体系构架的特征，适当计算处理性能的微处理器



关系：嵌入式产品包含了嵌入式系统，但并不是嵌入式系统。嵌入式系统包含嵌入式数据库、嵌入式操作系统、嵌入式处理器，其中嵌入式操作系统是嵌入式系统的核心，嵌入式处理器是硬件基础，嵌入式数据库是嵌入式系统的上层应用。

**2． 讨论嵌入式系统五层结构与对象化模型的关系？**

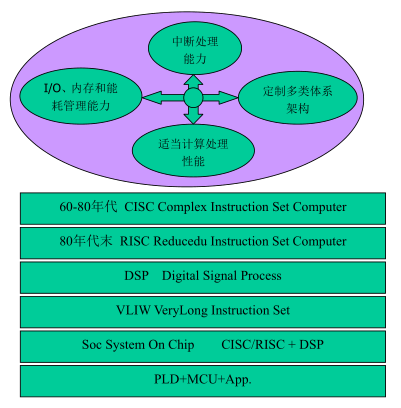


1. 看问题的角度不同。五层模型综合了系统的物理与逻辑架构，兼顾了系统宏观与微观的层次要素，体现了系统的思想，是由内到外的。对象化仅仅是逻辑上的，反映了系统的应用角度和内在形态，在逻辑上是完全的，也是系统的。但对于五层结构来说，它反映的是五层结构的逻辑应用的一个侧面，是由外向内的。

2. 层次间关系不同。嵌入式五层模型是一个层次分明、结构严谨的框架，且具有自己的约束条件，各层顺序严格规定，每一层只与相邻的层发生关系，比环形结构更加系统。对象化环是一个相对松散的结构，各个模块间是并列的关系，符合逻辑应用的侧重点。

3. 侧重点、功能不同。五层模型侧重系统整体的设计与分析，对象化环模型则侧重于逻辑应用。

**3. 依据下图如何理解嵌入式处理器内涵，及其指令系统分类？**



上图从应用角度给出嵌入式微处理器的定义：指具备强中断、I/O、内存和能耗管理能力，具有定制多类体系架构特征，适当计算处理性能的微处理器。而这些功能、特性由中间的指令系统支持，微处理器与指令系统密不可分。

指令系统从产生到现在的形态主要有五种风格；复杂指令集（CISC），精简指令集（RISC），超长指令字（VLIW），数据信号处理指令集（DSP），专用指令集（ASIP）。

CISC是一种为了便于编程和提高记忆体访问效率的晶片设计体系，常用指令只占20%，效率不高。

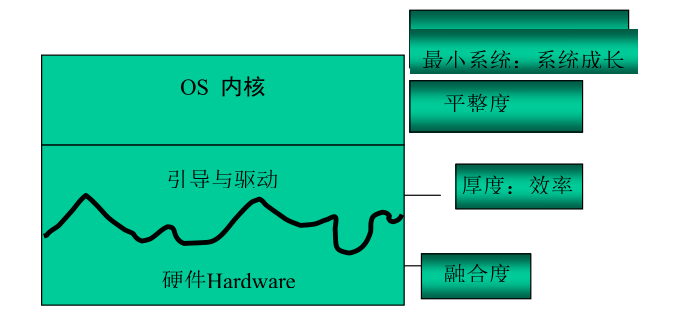
RISC是为了提高处理器运行速度而设计的晶片设计体系。关键技术在于流水线（pipelining）操作：在一个时钟周期里完成多条指令。提高了执行速度和可靠性，成本较低，但对编译器要求较高。

VLIW由编译器将多条可同时发送的指令并排在一条超长指令字中，并行执行多个操作，扩展了指令并行度。

DSP面向特殊应用，具有独特的体系结构，其芯片采用改进的哈佛结构，指令系统为流水线操作，数字信号处理可通过软件修改处理参数，灵活性强。具有良好的多机并行运行特性，采用专用硬件乘法器，电压较低。

ASIP的核心思想是，针对特定应用设计专用的指令集，再根据设计好的指令集优化处理器的体系机构，使优化后的体系结构更适合执行专用指令，进一步提高指令执行效率。

**4． 锯齿融合图：引导与驱动的缓冲与分割原则解释**



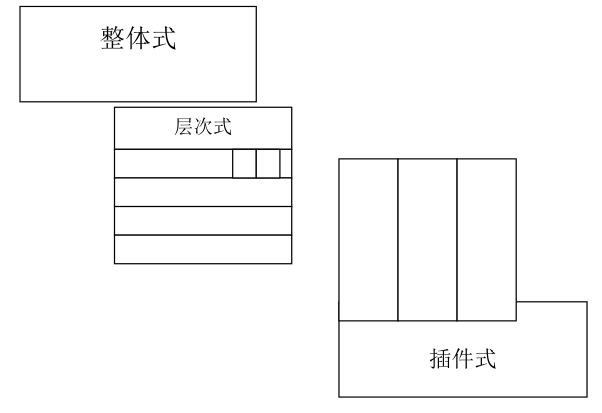
1.启动前操作系统仅仅是存储在不挥发介质中的代码，从启动的一刻开始，操作系统将被装载到内存中，成为完整功能的运行系统。这个装载的过程就是引导。引导是系统启动和运转的第一步，是系统从硬件走向系统软件的开始，从这一刻开始，系统将从固化的代码和静态的电路转化为动态运行的程序和具有功能的系统。2.为了保证系统功能可以正常调用，所做的初始化准备就是驱动。3.顺利进行引导与驱动两部分工作就是引导层要做的工作。

融合度是指引导层对于下级的硬件层控制和兼容能力。由于硬件的参差不齐，所以融合度不同，才有了引导与驱动这一层，厚度是指引导层的规模。一般来说，引导层的不稳定性与引导层的规模成正比。平整度是指引导层对上级的中间件层所提供的操作接口的统一性和友好性。

引导层的作用之一就是屏蔽掉底层硬件的差异性，为上层的操作系统和各种程序调用提供统一的接口。如果引导层任务太多，虽然会向上提供一个平整的接口，但效率会明显下降。如果引导层太薄，效率会提高，但设备无关性会降低，应结合具体应用分析。引导层的平整度、厚度和融合度是衡量引导层的重要指标。

最小系统分两种，一是OS内核，引导和驱动，二是引导和驱动。它是一个含有微处理器的系统。具有制造功能和用户功能所需的最少硬件及软件环境，是嵌入式系统能工作的最低要求，不具有智能性，是系统成长的基础。

**5． 嵌入式操作系统典型结构构造图：整体、层次、插件**

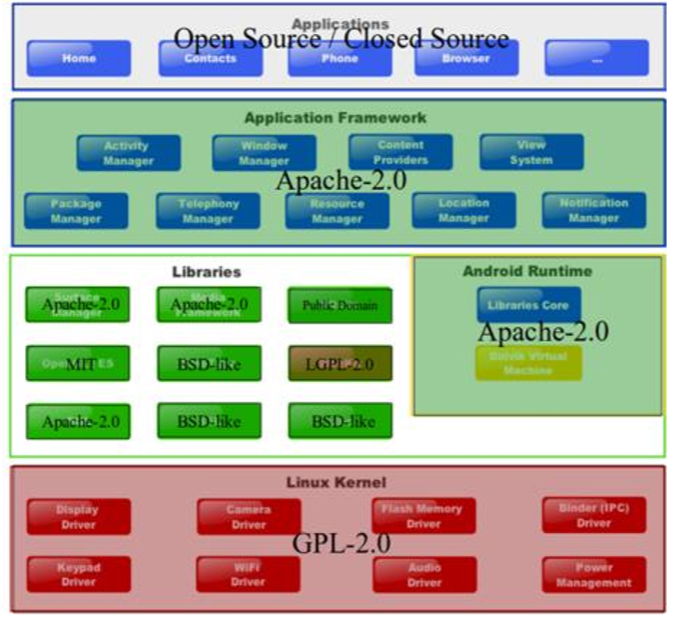


整体内核（单内核，大内核），是一种几乎涵盖了所有操作系统功能模块，如进程调度、进程间通信、内存管理、设备管理、文件系统、网络系统的内核结构。大部分模块遵守特定的接口规范，相互协调。所有模块在编译时链接在一起，形成一个可执行文件。运行时所有操作系统功能模块处于内核态，而其他用户程序和系统程序处于用户态，通过系统调用可以切换到内核态。但调试困难，难于裁剪，稳定性相对较差。如WinCE就是采用此结构。

插件式结构（微内核结构），是一种类似于C/S模式的内核结构。微内核运行在核心态，提供所有操作系统的基本操作。结构规范，易于裁剪和编程，且调试简单。扩展性和移植性、鲁棒性好，但整体效率低，资源管理复杂。如QNX就是采用此结构。

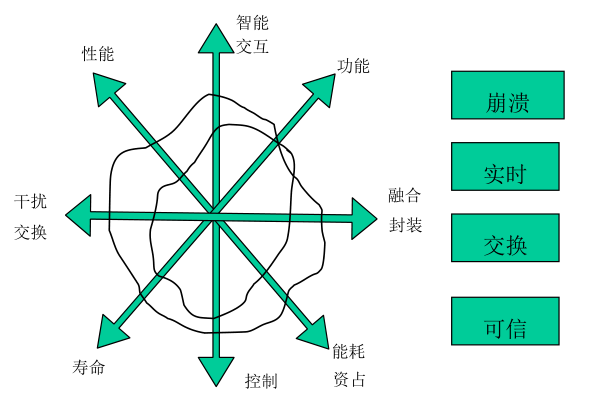
层次内核结构是一种基于分层思想的内核结构。它把操作系统的所有功能划分为若干模块，按功能流程的调用次序排列成若干层。（下层模块封装内部细节并向上提供统一的抽象和接口，上层模块调用下层模块提供的接口，各个层级间的实现比较独立。各个模块间的组织结构和依赖关系清晰明确，适合进行系统功能的划分）分层思想有利于组织操作系统的开发，易于调试和诊断，系统的修改和扩展相对容易。但是对系统功能进行定制和裁剪相对困难，且它是一个理论化的结构，实际中是没有完全符合层次的操作系统。

**6．以 android 系统中开源软件使用为例，说明其中使用的几种软件许可证？进而说明软件许 可证交叉使用的行业价值？**

****

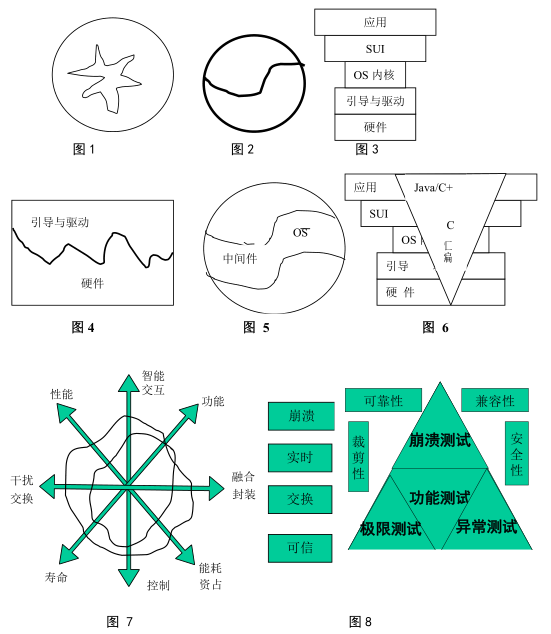
**软件许可证的最大行业价值是让自由开源软件既能独立发展，又能与私有、商业软件有效对接，共同发展，解决了发展的排他问题。**

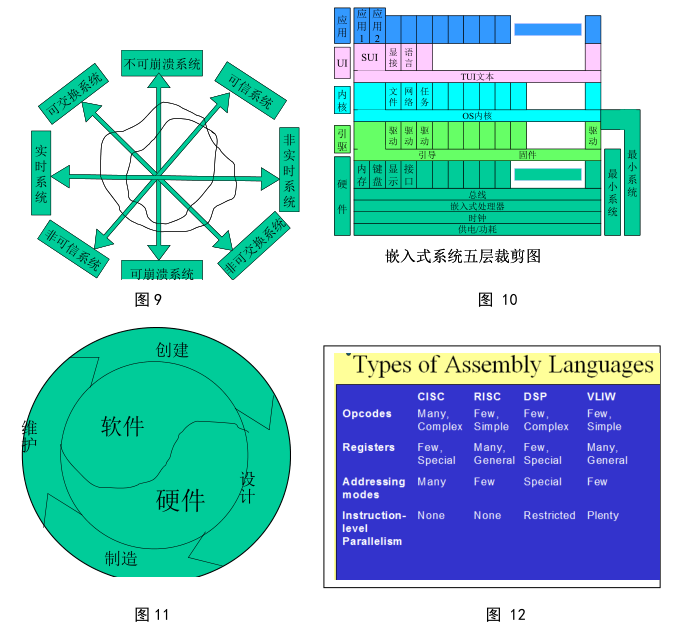
**8.从系统辩证的角度，分析嵌入式系统结构蜘蛛图相互关系？**



？？？为了更好的衡量对比操作系统在各个方面的异同，引入了多位分析法或蜘蛛图法的定性分析方法。每条评价线的端点是系统中顾此失彼的两种性能要求，而整条线长代表了资源量。对一个嵌入式操作系统，在资源相同的条件下，只能在对立性能的要求上权衡，对于不同的嵌入式系统，就产生了不同的形状，如图中的两个环，代表两个不同的系统，每一种系统在对立的两个性能指标间所占的比例不同，体现了两种系统的差异。

**四、 图论简述题**



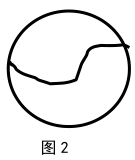


**图1===========================================================================**



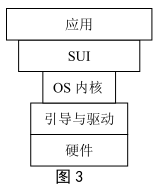
答：该图形象的表示了对象体系与嵌入式系统之间的关系，对象体系“包含”嵌入式系统，而嵌入式系统“嵌”在对象体系之中。不规则的图形代表嵌入式系统，是嵌入式产品的核心，而外围包围他的是嵌入式产品。嵌入式系统在嵌入式产品处于中心的位置，决定了嵌入式产品的主要特性，并且其外形的不规则说明了嵌入式系统本身的性能是多样而且有其不确定的一面。

**图2===========================================================================**



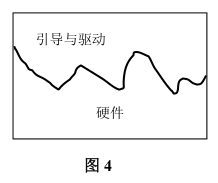
答：该图类似于中国古代的太极图，有一种此消彼长，互相制约的关系。可以表示存储器和MCU的关系，数据和程序的关系，软件和硬件的关系等等比如说存储器和MCU的关系，它们构成了嵌入式系统很大的一部分，而且决定了系统功能及性能的强弱。有的应用要求微处理器有比较快的处理速度，而数据量比较少，因此只需要很少的存储器；有的应用要处理大量数据，需要大容量存储器，而并不需要很快的微处理器。要处理好这两者之间的关系。

**图3===========================================================================**



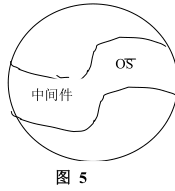
答：是嵌入式系统的五层结构模型。五层模型最底层为硬件层，这是系统架构的基础，所有的功能实现与应用都是以此为基础的。硬件层以上是引导层，它是静止的硬件与动态运行的嵌入式系统之间的过渡。硬件层与引导层一起可以构成一个最小系统。引导层之上是内核层，实质上就是指嵌入式操作系统。第四层是SUI是计算机和使用者之间的对话接口。最上层为用户层，是面向用户的，大多数程序都在这一层进行工作。其中，内核层与SUI这两层又统称为中间件。内核层实际上就是指操作系统，UI层为图形用户界面接口（GUI）及其他系统与用户接口。

**图4===========================================================================**



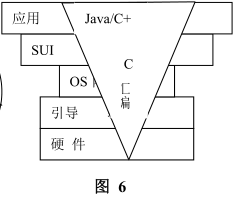
答：该图显示出了引导层与驱动和硬件之间的关联。引导层与驱动在硬件的上方，是五层结构的第二层。硬件通过引导层与驱动才能够被上层的系统及软件使用，可以说引导与驱动就相当于硬件对于上层的接口。而上层的系统也只有通过引导与驱动才能够访问到硬件。这两层之间的结合部分并不是平滑的，而是互相渗透的嵌入式的关系。

**图5===========================================================================**



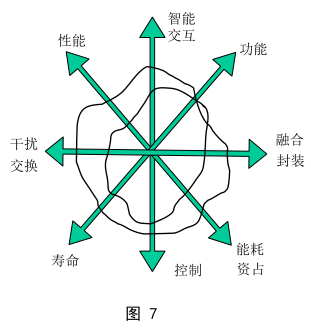
答：图中的夹层将圆分为两部分，夹在中间的是中间件，属于嵌入式系统的第三和第四层。系统需要通过中间件与硬件进行交互。对于使用者来说，此环节越薄越好，而对于系统设计和延续的角度来讲，则希望它是越大越好。嵌入式系统正是靠着这种你中有我、我中有你的相互包含的关系，才能达到体积功耗的最小化，性能速度则是最大化。正是所谓的嵌入式系统的“硬中有软，软中有硬”的特殊性。

**图6===========================================================================**



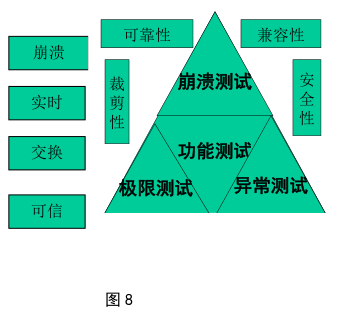
答：嵌入式系统的五层结构中与用户最近的一层就是第五层应用层。而应用层则是由各种程序组成的，这些程序又出自于各种不同的程序设计语言，比如Java、C+、C、汇编语言等。该图把程序及语言放在五层结构的上面，意思是对于普通的用户和消费者而言，这些才是看得到用得到的部分，它们遮盖住了嵌入式系统的复杂结构，是嵌入式系统与用户的统一接口。任何系统的根本目的都是为了对数据进行处理，而这种处理正是通过程序这种特殊的数据来实现的。如何保证应用程序及数据的开放和兼容性，是行业面临的基础问题。因此最理想的情况下是软件及程序能够具有非常好的兼容和移植性，能够使用户看不出复杂的底层结构上的差别。

**图7===========================================================================**



为了更好的衡量对比操作系统在各个方面的异同，引入了多位分析法或蜘蛛图法的定性分析方法。每条评价线的端点是系统中顾此失彼的两种性能要求，而整条线长代表了资源量。对一个嵌入式操作系统，在资源相同的条件下，只能在对立性能的要求上权衡，对于不同的嵌入式系统，就产生了不同的形状，如图中的两个环，代表两个不同的系统，每一种系统在对立的两个性能指标间所占的比例不同，体现了两种系统的差异。

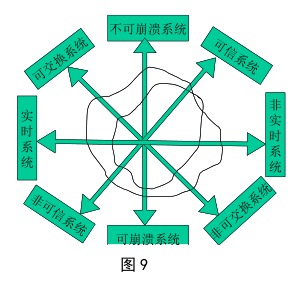
**8===========================================================================**



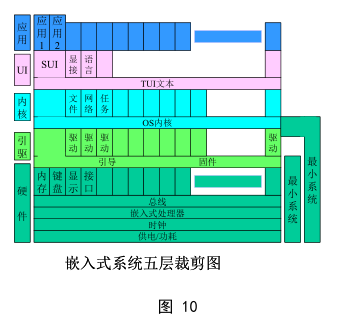
答：该图画出了嵌入式系统的测试方法：功能测试、极限测试、崩溃测试、异常测试，以及可靠性、裁剪性、兼容性、安全性测试。

功能测试是嵌入式系统测试的最基本成分，同时崩溃测试、极限测试、异常测试也是非常重要的测试环节。而且可靠性测试可以测试系统的可靠程度，裁剪性测试则测试出了系统的裁剪性，兼容性测试可看出系统的兼容程度，而异常测试则是测试系统是否会发生异常。只有这样才能尽量使系统具有最好的稳定性。

**图9===========================================================================**



答：此图为设计结构的八角图，也称为蜘蛛图。嵌入式系统的特性可以分为可信系统与非可信系统，可交换系统与非可交换系统，实时系统与非实时系统，可信系统与非可信系统四种。任何一个操作系统都不可能同时满足任何对角的系统特性，这就必须要根据实际的需求来选择各个属性的程度，来组建一个最符合用户需求的嵌入式系统。在实际的工作中，往往也需要靠经验和对规律的总结来设定到底采用哪些系统方式。

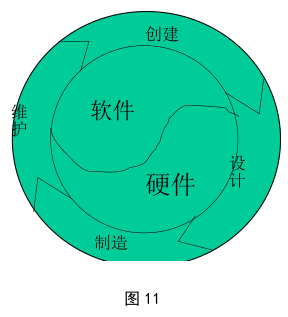
**图10=========****==================================================================**

答：该图是嵌入式系统的五层裁剪图，用以对嵌入式系统进行抽象、概括和描述，方便对系统进行实验、分析与预测，并且作为一个框架，用来协调各层标准的制定。嵌入式系统的五层结构分别是应用层、UI层、内核层、引导驱动层和硬件层。

最底层的硬件层包括供电、时钟、处理器、总线，还有内存、键盘、显示、接口部分。引导驱动层下面是引导及固件，上面则由驱动作为内核的接口。硬件和引导驱动层共同构成了最小系统。

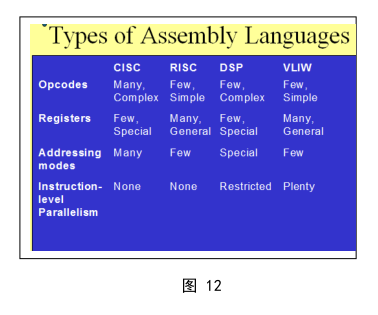
再上一层是内核层，包括相对底层的OS内核部分，它连接着内核与驱动。也包括文件、网络、任务等。UI层为使用者界面，包括SUI、文件、网络、任务。而应用层则是与消费者最接近的一层，是由各种应用程序需组成。

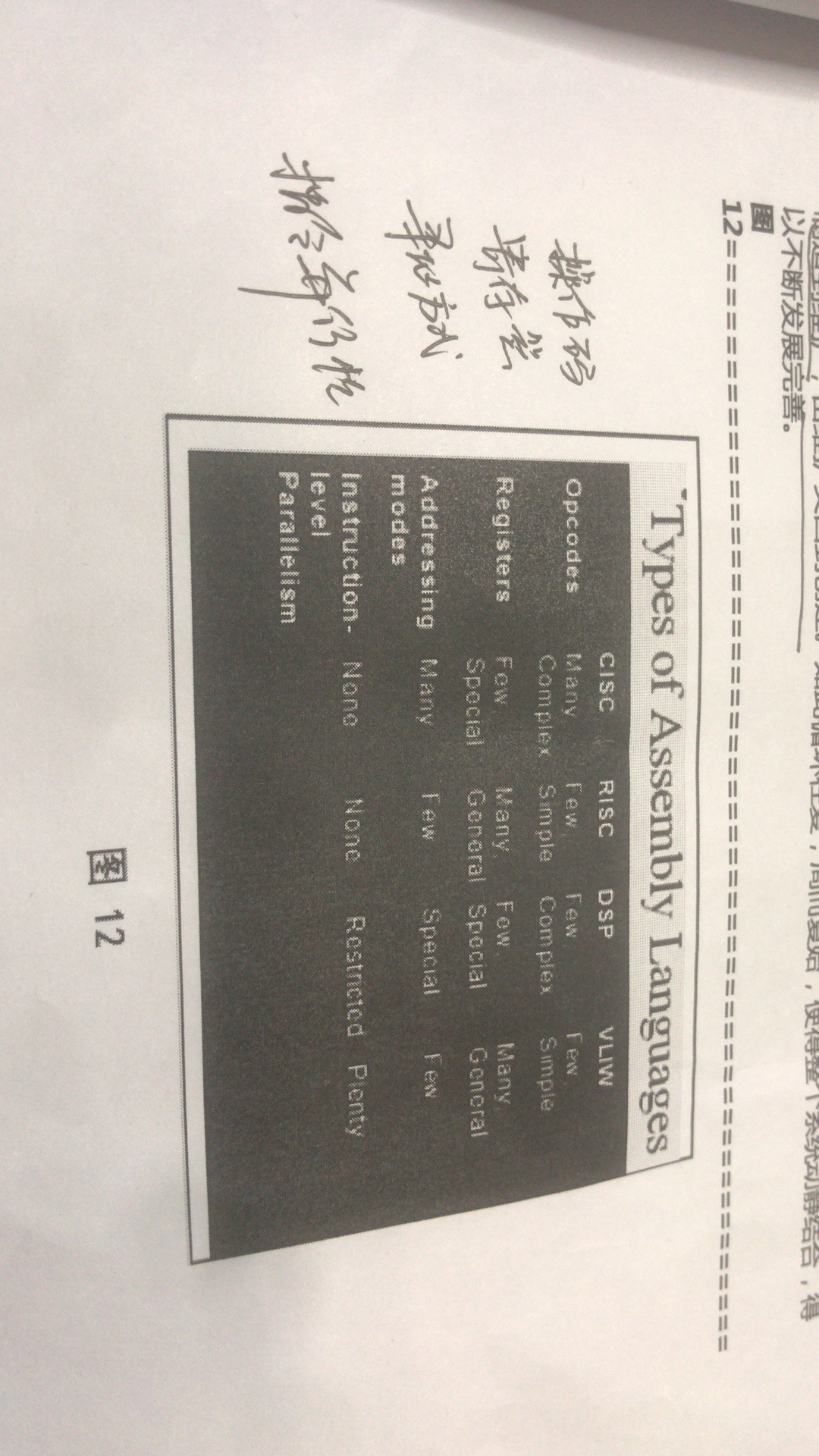
**图11===========================================================================**

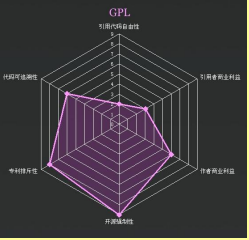


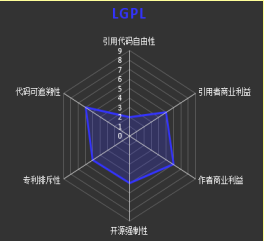
答：该图类似于太极阴阳图，是对于整个嵌入式系统的总括。中间的硬件和软件是两个不同的部分，但又通过某些方式交织在一起，互相对立，又互相包含。外层则是围绕着整个嵌入式系统的软硬件的一种动态的流程与方法，从创建到设计，再由设计到制造，由制造到维护，由维护又回到创建。如此循环往复，周而复始，使得整个系统动静结合，得以不断发展完善。

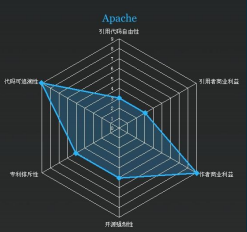
**图12===========================================================================**





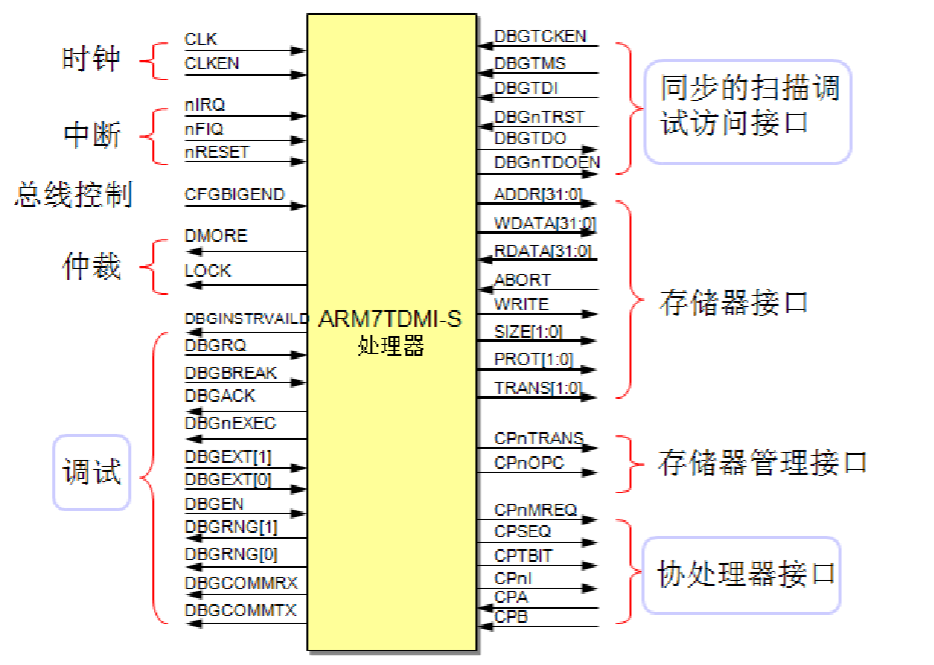






**五、 实验讨论题**

**1．简述 ARM 处理器结构的六大特征？及其下图 ARM7 处理器芯片的管脚分布描述？**

****

Von & Havard

Little & Big

ARM &Thumb

DSP & Javazell

Secure & Mali

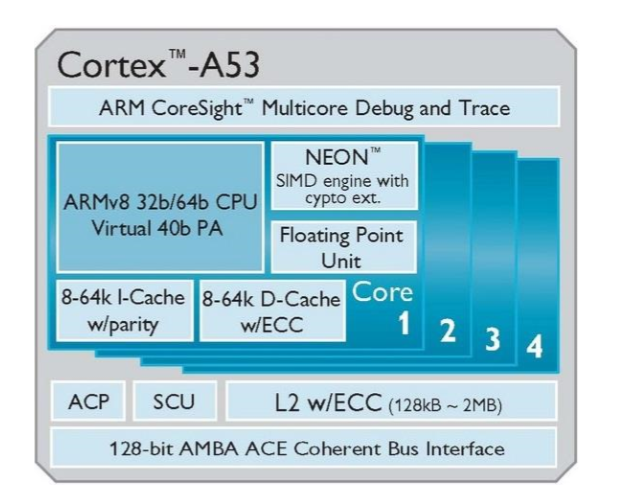
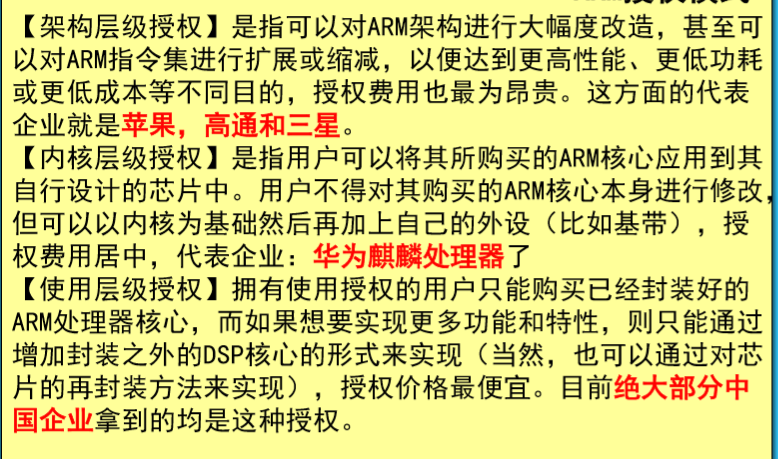
Sys & Debug

数据总线、地址总线、时钟类、显示控制

**2．依据下图如何理解实验中使用的 S5P6818 嵌入式处理器架构中使用的 CortexA53八核类型？ARM产品体系中其他两种类型及特点是什么？ARM体系中 使用的架构层级授权、内核层级授权、使用层级授权三种方式分别是什么含义？**

**StrongARM Intel XScale**

**Xscale 处理器是基于ARMv5TE体系结构的解决方案，是 一款全性能、高性价比、低功耗的处理器。它支持16位 的Thumb指令和DSP指令集，已使用在数字移动电话、个 人数字助理和网络产品等场合。**

****

**3．请叙述项目实验的基本步骤？为什么说嵌入式系统设计要采用结构化设计，而 应用可以采用对象化设计？结合实验说明一下嵌入式系统性能优化的内涵？**

**4．用语言描述实验控制箱硬件电路系统的特征？**



**5. 实验过程中，如果更换 MCU，那些环节的工具会发生变化？结合本组特点说明软硬件部分哪些可以裁剪掉？**

**6．以自己任务实验为例说明优化裁剪界面？按照目标系统的设计要求，请描述系统（硬件和软件）裁剪内容？为什么？**

**7. 结合实验过程，说明对嵌入式系统可靠性方面的理解？**

**可用性 可信性 安全性 完整性 可维护性**