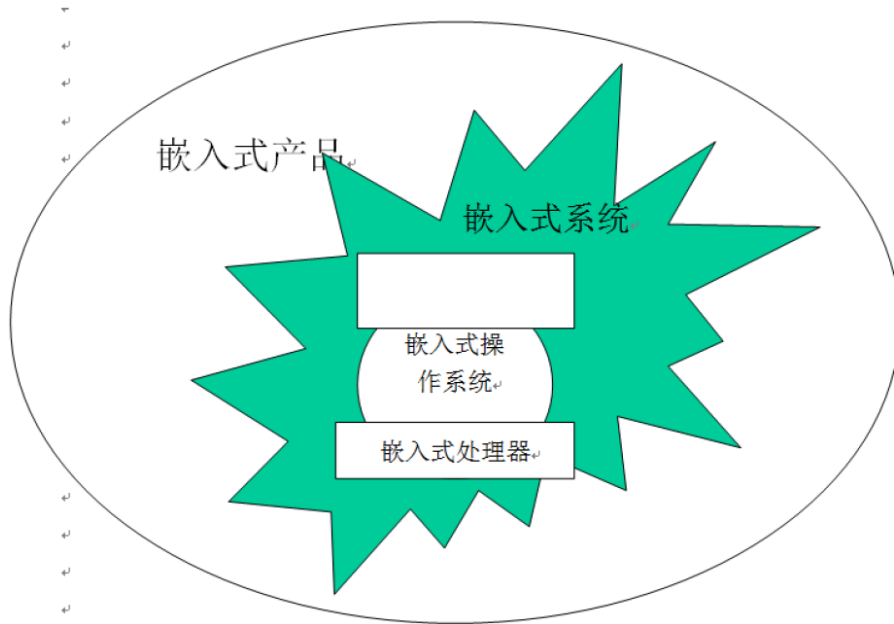


# 《嵌入式系统》图论题资料

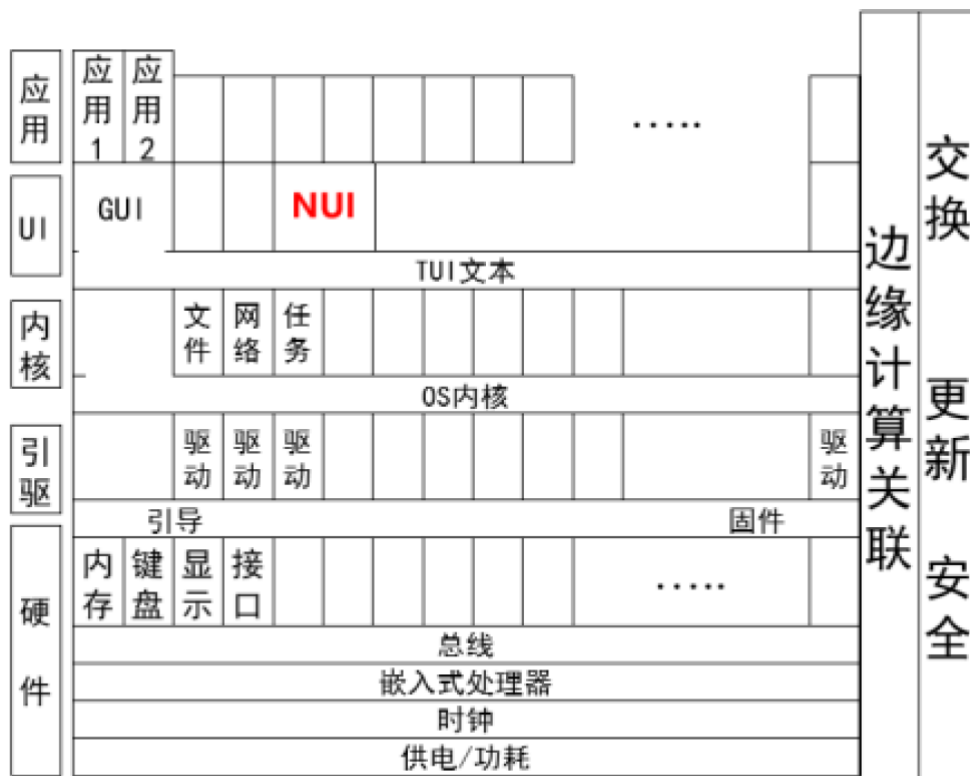
1.依照下图圆环套图概念关系，论述嵌入式处理器、嵌入式操作系统、嵌入式系统与嵌入式产品的相互关系？



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.依据组织型五层结构图，对比讨论嵌入式系统五层结构与可裁剪性的关系？同时说明其中TUI、GUI、NUI的异同处？



组织型嵌入式系统五层裁剪图

#### 与可裁剪性的关系：

- 硬件层：硬件层是系统的基础，提供了底层的物理设备和接口。可裁剪性体现在硬件的选择和配置上，可以根据系统的需求选择适合的处理器、传感器、执行器等硬件组件，并进行定制化设计，以满足特定的功能和性能要求。
- 驱动层：驱动层负责管理硬件与操作系统之间的交互，提供统一的接口供操作系统访问硬件资源。可裁剪性体现在驱动程序的定制和优化上，可以根据特定的硬件配置和系统需求开发定制的驱动程序，以提高系统的性能和稳定性。
- 内核层：内核层提供了系统的核心功能，包括任务调度、内存管理、设备驱动等。可裁剪性体现在操作系统的选择和定制上，可以根据系统的特点选择适合的嵌入式操作系统，并进行定制化配置，以满足系统的实时性、功耗和资源占用等需求。
- UI层：UI层提供了各种功能模块和服务，如通信协议栈、文件系统、数据库等。可裁剪性体现在中间件的选择和配置上，可以根据系统的需求选择适合的中间件组件，并进行定制化开发和集成，以支持系统的特定功能和应用场景。
- 应用层：应用层是系统的最顶层，包括用户界面、应用程序和业务逻辑。可裁剪性体现在应用软件的开发和定制上，可以根据用户需求和系统特点开发定制化的应用程序，以满足特定的功能和用户体验要求。

#### TUI、GUI、NUI的异同处：

不同点：

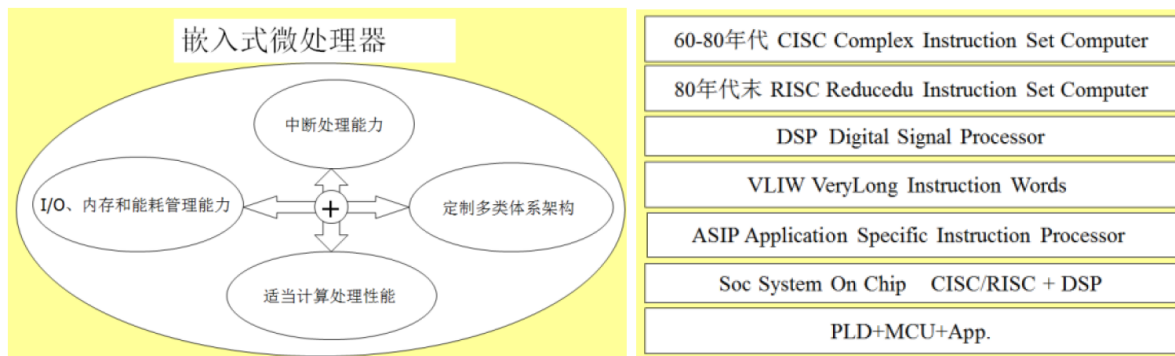
- 交互方式：
  - TUI：基于文本的用户界面，用户通过键盘输入和文本输出进行交互。通常使用命令行或终端窗口显示信息，并通过键盘输入命令或选项进行操作。
  - GUI：基于图形的用户界面，用户通过图形元素（如按钮、菜单、窗口等）进行交互，通常使用鼠标、触摸屏或其他指针设备进行操作。

- NUI: 自然用户界面, 通过自然的人体动作 (如手势、语音、眼神等) 进行交互, 不需要传统的输入设备 (如键盘、鼠标)。
- 用户体验:
  - TUI: 相对简单, 适用于一些需要快速执行命令或脚本的场景, 但通常交互体验较为受限。
  - GUI: 提供丰富的图形界面和交互元素, 使用户能够直观、友好地与系统进行交互, 适用于大多数桌面和移动应用程序。
  - NUI: 强调自然、直观的交互体验, 可以提供更加沉浸式和人性化的用户体验, 但在技术实现和用户适应方面可能存在挑战。

相同点:

- 都致力于提供用户友好的界面设计和良好的用户体验。
- 都提供了用户与计算机系统进行交互的方式
- 提供直观、易用的界面设计, 以帮助用户完成任务和实现目标。
- 被广泛应用于各种领域的软件和硬件系统中。

3.依据下图, 解释嵌入式处理器内涵, 并从中提炼5类主要指令系统, 做简单解释?

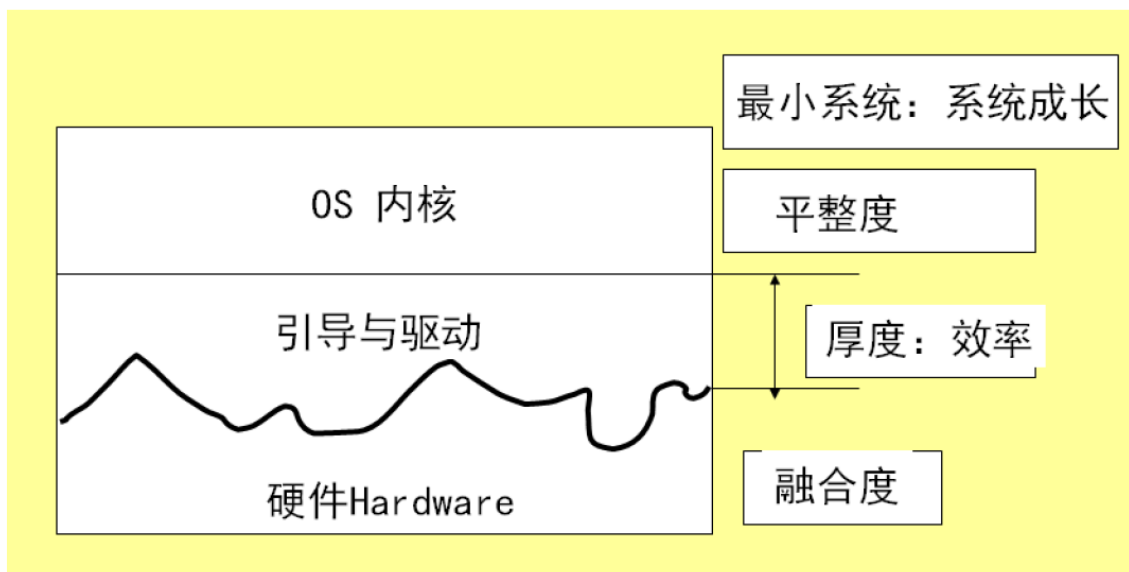


**嵌入式处理器内涵:** 上图从应用角度给出嵌入式微处理器的定义: 指具备强中断、I/O、内存和能耗管理能力, 具有定制多类体系架构特征, 适当计算处理性能的微处理器。而这些功能、特性由**中间的指令系统**支持, 微处理器与指令系统密不可分。

列举五类指令系统:

- **复杂指令集 (CISC):** 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。
- **精简指令集 (RISC):** 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。
- **超长指令字 (VLIW):** 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。
- **数据信号处理指令集 (DSP):** 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。
- **专用指令集 (ASIP):** 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.依据下面锯齿融合图: 论述引导与驱动的缓冲与分割作用? 解释其中融合度、厚度和平整度的基本含义?



#### 缓冲与分割作用：

- 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。
- 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。
- 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

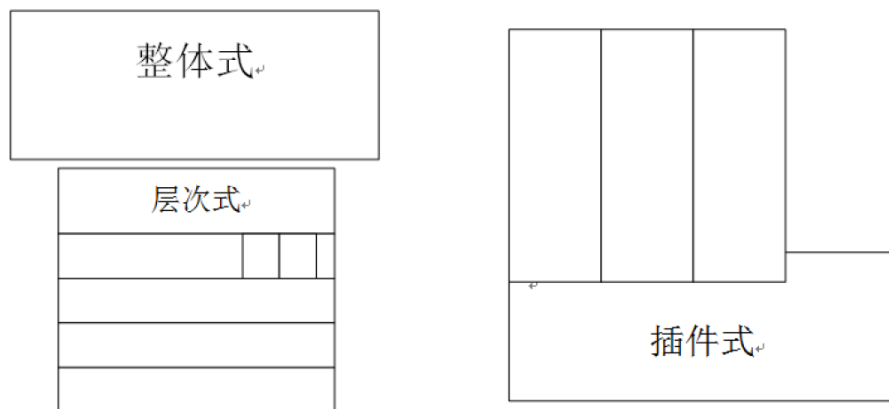
**厚度：**是指引导层的规模。一般来说，引导层的不稳定性与引导层的规模成正比

**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，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.依据下图示意，嵌入式操作系统三种典型结构：整体、层次、插件，论述层次与微内核OS各自的优缺点？

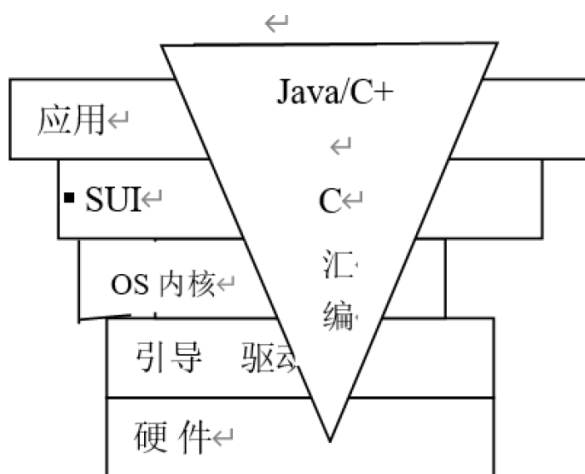


**整体内核**（单内核，大内核），是一种几乎涵盖了所有操作系统功能模块，如进程调度、进程间通信、内存管理、设备管理、文件系统、网络系统的内核结构。大部分模块遵守特定的接口规范，相互协调。所有模块在编译时链接在一起，形成一个可执行文件。运行时所有操作系统功能模块处于内核态，而其他用户程序和系统程序处于用户态，通过系统调用可以切换到内核态。但调试困难，难于裁剪，稳定性相对较差。如WinCE就是采用此结构。

**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。

**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。

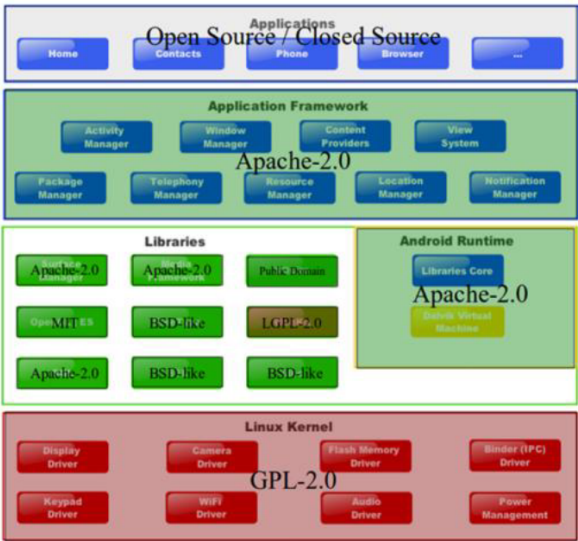
6.依据此图描述系统五层结构与编程语言位置关系？给出相应关系的简单理由？



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。

7.以Android系统中开源软件许可使用为例，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？



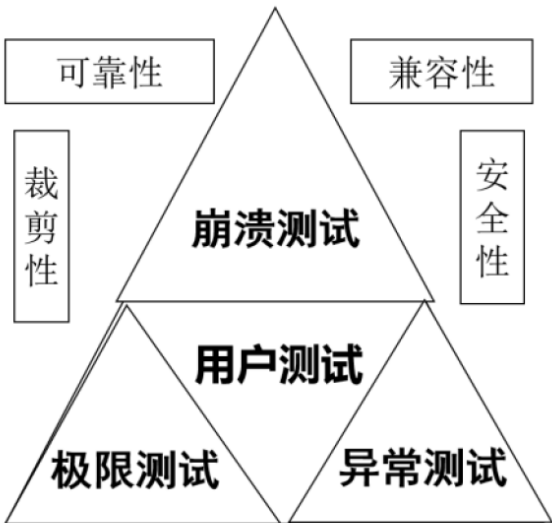
软件许可证：

- **Apache许可证:** 即Apache License，是一个由Apache软件基金会(ASF)发布的自由软件许可证。Apache许可证最初为Apache Web服务器而撰写,Apache许可证在Apache社区内外被广泛使用；Apache基金会下属所有项目都使用Apache许可证；许多非Apache基金会项目也使用了Apache许可证。
- **GPL许可证:** GNU通用公共许可协议，是被广泛使用的自由软件许可证，给予了终端用户运行、学习、共享和修改软件的自由。GPL是一个Copyleft许可证，这意味着只要项目的某个部分（如动态链接库）以GPL发布，则整个项目以及派生作品只能以相同的许可条款分发，这与宽松自由软件许可证有所区别。GPL是第一个普遍使用的Copyleft许可证。根据GPL许可的优异自由软件程序的例子有Linux内核和GNU编译器集合（GCC）。

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.依据此图，结合自己本组的主题设计简要说明嵌入式系统测试种类及相关内涵？



种类：



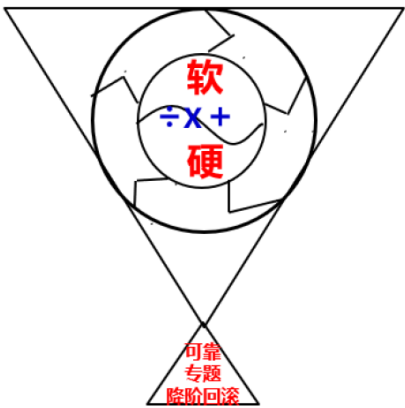
功能测试、极限测试、崩溃测试、异常测试，以及可靠性、裁剪性、兼容性、安全性测试

内涵：

功能测试是嵌入式系统测试的最基本部分，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。

9.依据下图，简述嵌入式系统可靠性，说明系统降阶含义？

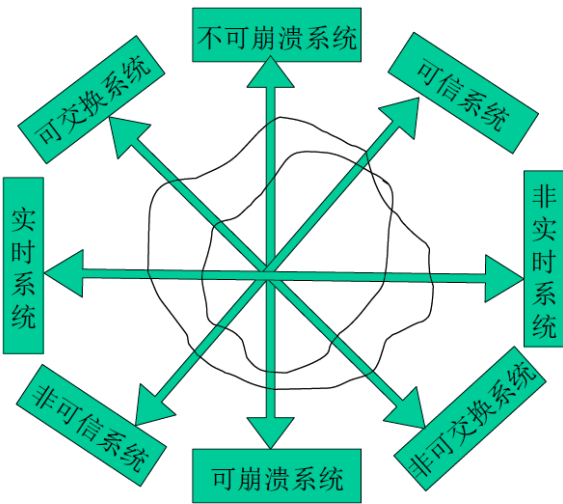
嵌入式系统可靠性



- 1. **硬件**：嵌入式系统的硬件组件应该具有高可靠性，硬件组件应该能够在不同的环境条件下长时间稳定运行，并且能够抵御各种外部干扰和故障。
- 2. **软件**：嵌入式系统的软件应该具有高可靠性，软件应该能够正确地响应各种输入和事件，并且能够处理异常情况和错误。
- 3. 通过**洞察除、放大乘、包容加三法则**，将软件与硬件交织在一起，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。循环往复，周而复始，使得整个系统动静结合，可靠性大大提高。

嵌入式系统中的"降阶"通常指的是将系统的性能、功能或者复杂度进行降级或者缩减，以达到特定的要求或者限制。本图利用系统回滚的方式，实现降阶。

10.从系统辩证的角度，讨论嵌入式系统分类？结合本组主题进行简要说明。

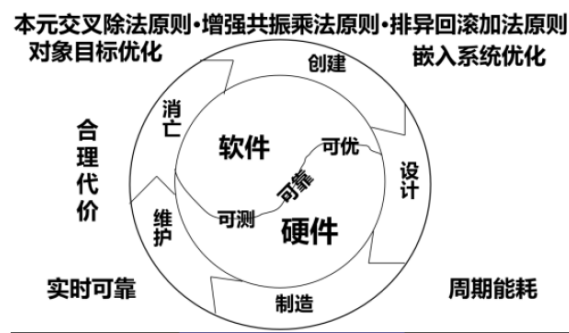


此图为设计结构的八角图，也称为蜘蛛图。

根据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，可崩溃系统与不可崩溃系统四种。**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。

每条对角线线的端点是分类中对立特性的两种系统。对一个嵌入式操作系统，在资源相同的条件下，**只能在对立特性的要求上权衡**，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。

11.依据下图总结本课程的主体内容关联关系？由此简单解释嵌入式系统的生命周期？



**关联关系：**

该图类似于太极阴阳图，是对于整个嵌入式系统的高度概括和总结。

**生命周期：**

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，由维护到消亡，由消亡再到创建。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，以此得到了嵌入式系统的生命周期。