

# 第7章 多处理器系统

# 本章的内容

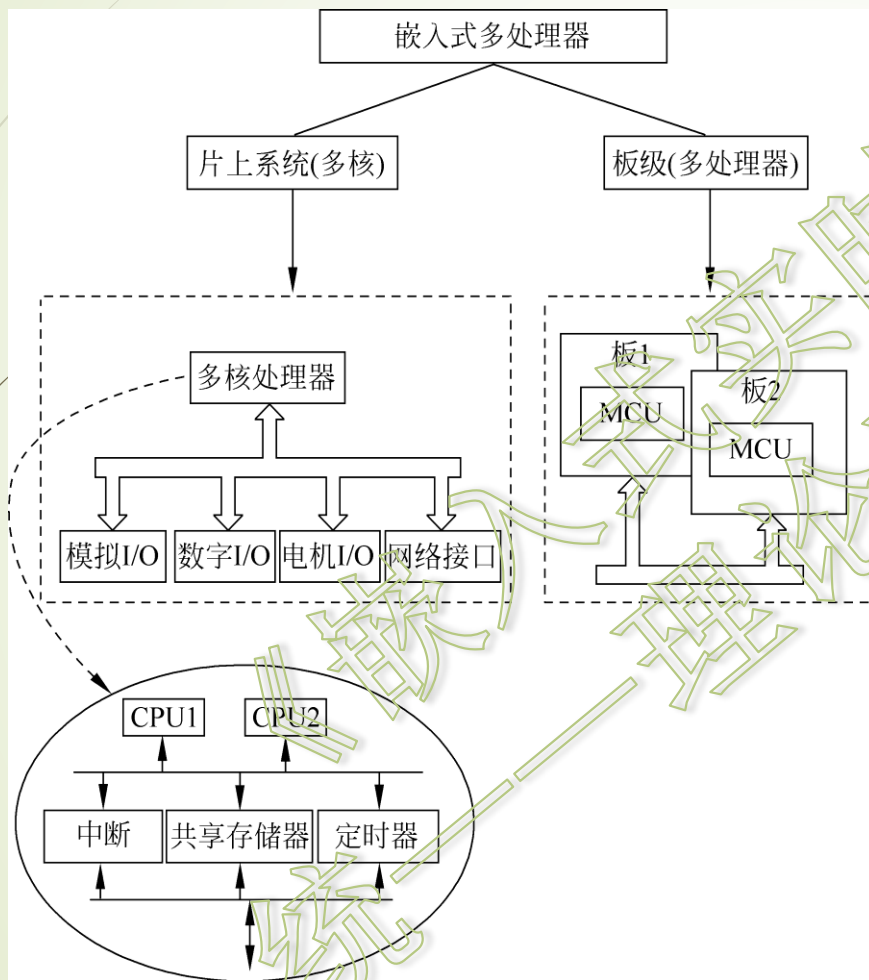
- 观察使用多处理器的原因。
- 研究可用的硬件芯片并描述其结构。
- 了解在多处理器设计中对软件进行分区和分配的实用方法。
- 描述控制多处理器软件执行的方法。
- 解释对称、非对称和绑定多处理技术的原理并阐述实践方法。

# 第一部分

## 第一部分Part 1

什么是嵌入式多处理器  
为什么需要嵌入式多处理器

# 嵌入式系统中的多处理器



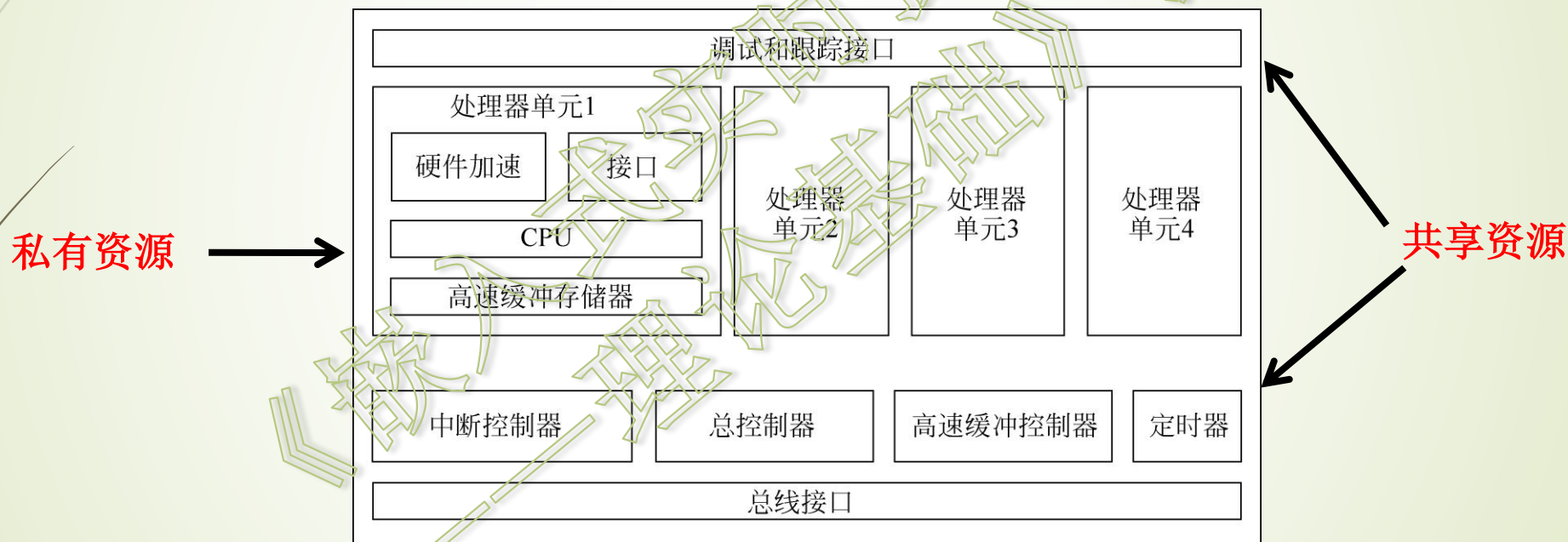
- 多核芯片提供更高性能和更小芯片尺寸。
- 这里多处理器被定义为是基于板级的为微型计算机系统。
- 每一个独立的板可能包含1个或者多个微控制器或者处理器。
- 这样的结构高度可伸缩 - 适合构建功能和性能强大的计算系统。
- 它们也可支持硬件上的处理器冗余。
- 多核和多处理器系统可以构造同构或者异构的体系架构。



# 同构的多处理器- 多核系统的实现案例

- 使用对称多处理器时，处理单元是相同的。
- 示例芯片：传统的ARM Cortex A9。

现代RK3568、I.MX6D 和STM32MP157，它们有两个或者以上架构相同的ARM Cortex-A核。

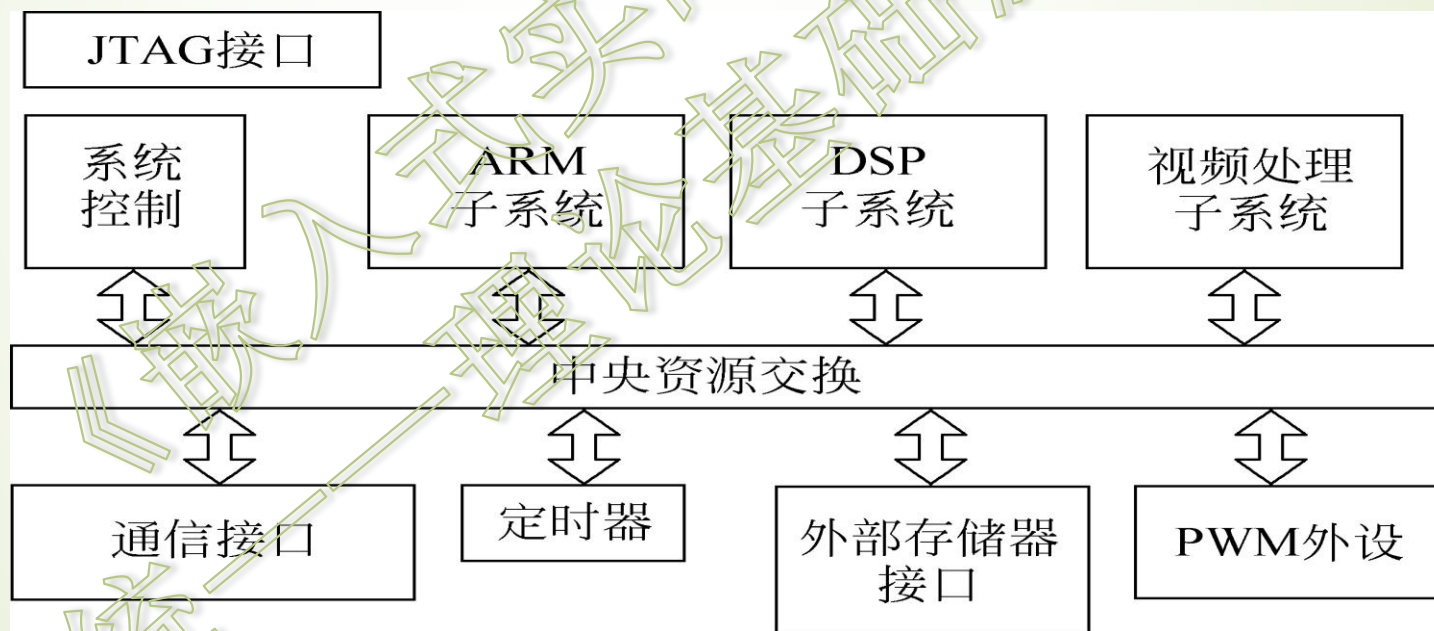


## 软件视角:

- ① 任务可以分配给处理器 - 处理器是“专用”资源。
- ② 任务可以在任何处理器上运行 - 处理器是“匿名”资源。

# 异构的多处理器——多核系统的实现案例

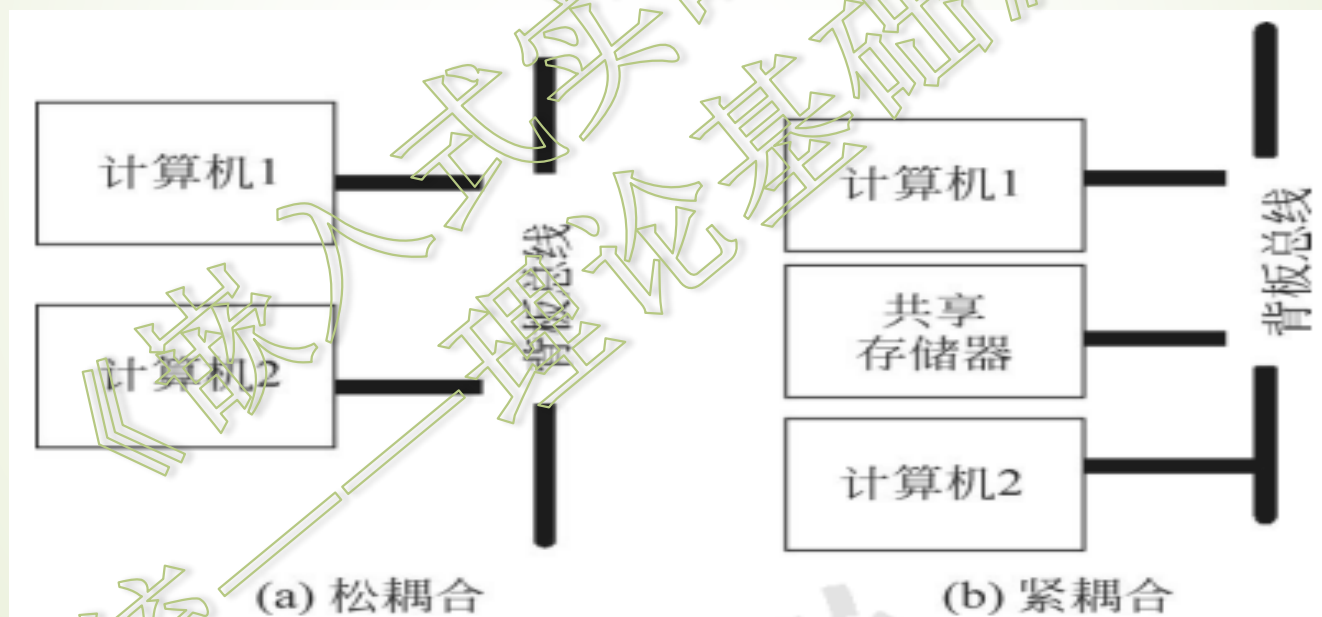
- 对于非对称多处理器，处理单元不同。
- 示例芯片：Texas TMS320DM6443。



# 多处理器基本通信结构

多计算机可以是：

- A) 松耦合 - 通过消息传递进行通信，或
- B) 紧耦合 - 通过共享内存进行通信。



## 第2部分

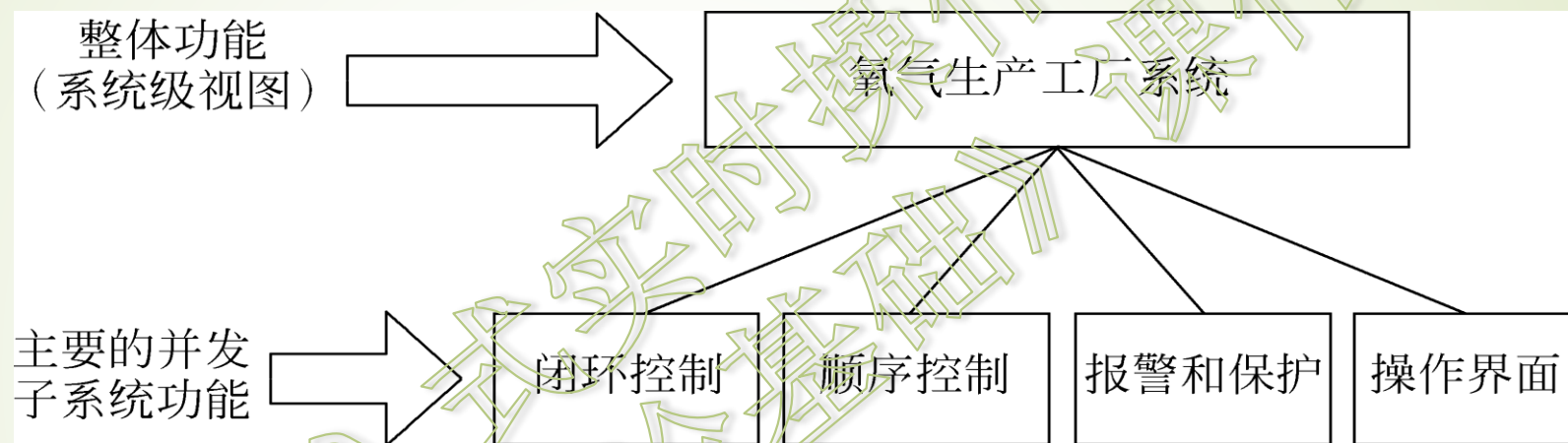
第2部分

软件问题

—  
基于系统功能的作业划分和分配



# 软件划分回顾——将软件构建为一组功能

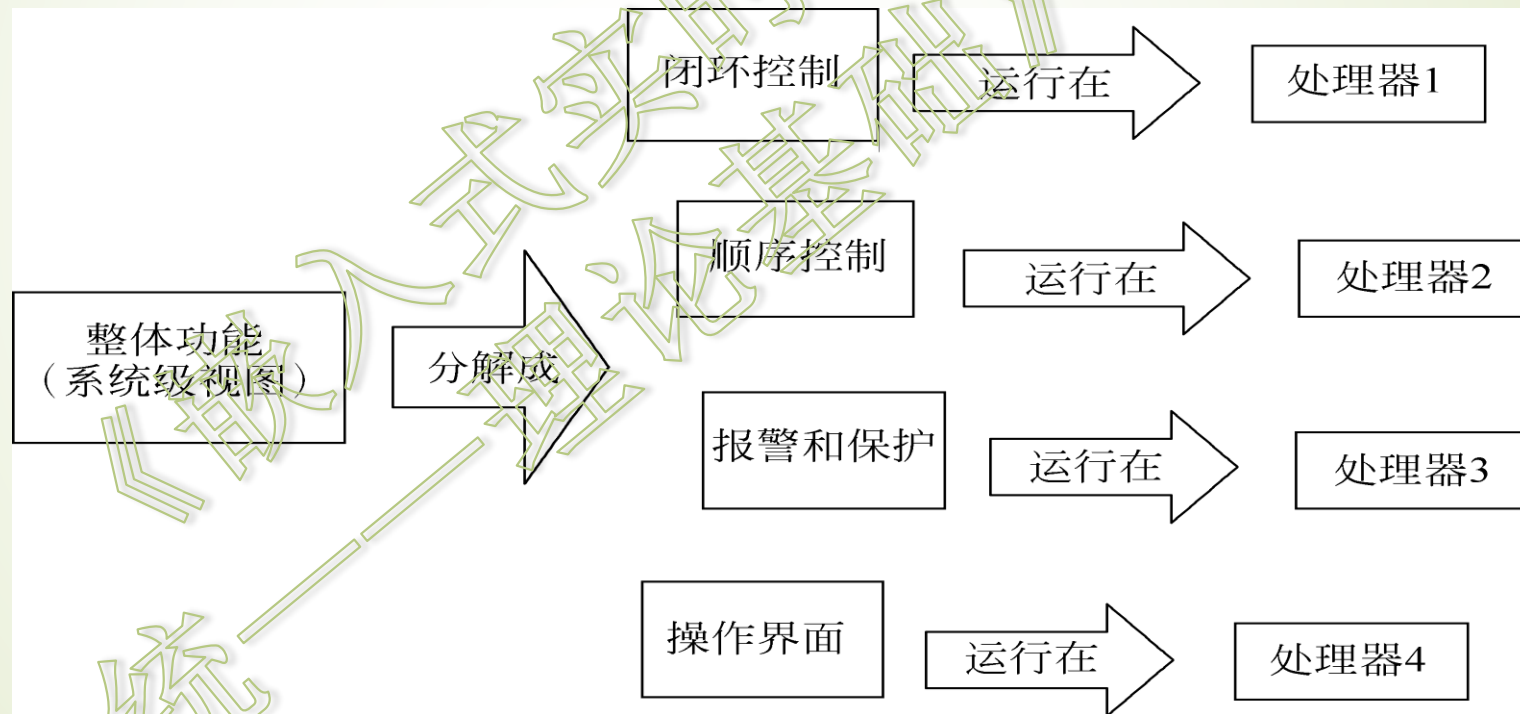


## 功能结构的基础

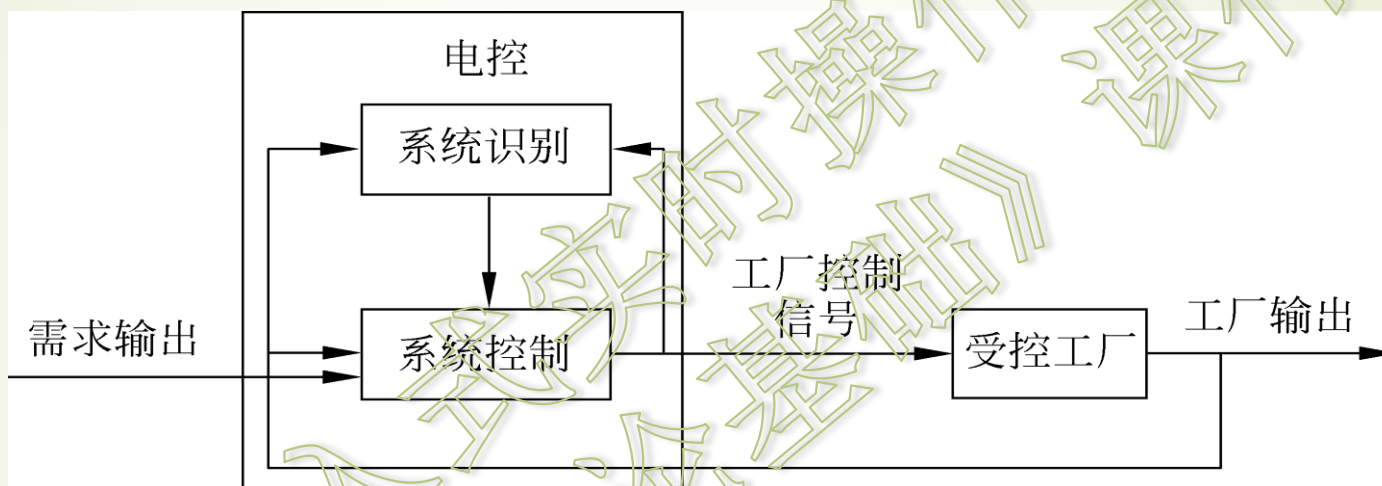
- ① 定义整个系统功能（“作业”）。
- ② 确定主要子系统。
- ③ 将它们作为单独的软件单元实现。
- ④ 将这些作为并发软件活动执行。

# 软件划分回顾——将软件构建为一组数据处理

- 实际运行时并发性取决于所使用的硬件平台。
- 下面显示的是四个功能分配到四个处理器。
- 这可以是多核或多处理器的实现



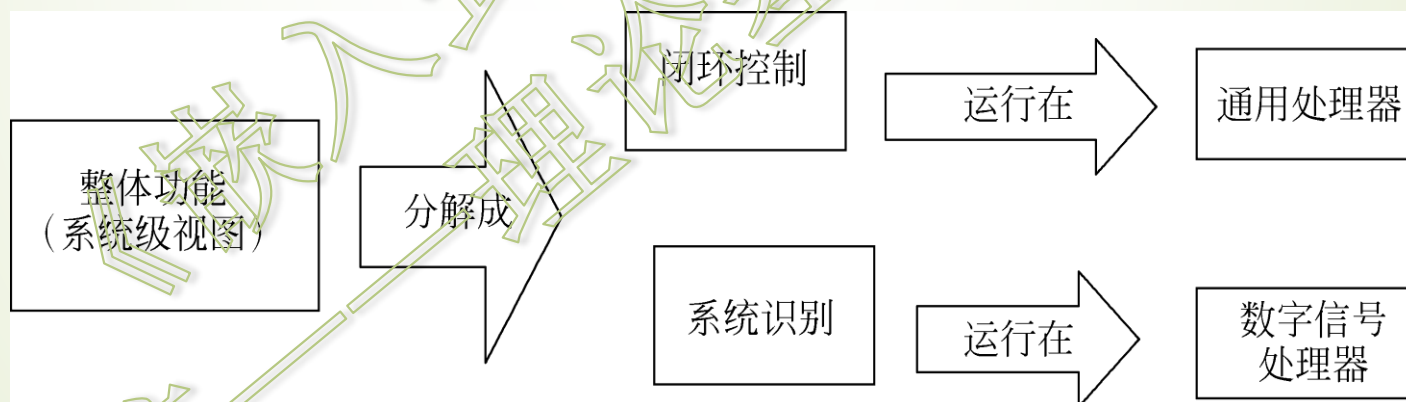
# 自适应控制系统案例



- 整体系统结构框图。
- 电子控制器由两大子系统组成：
  - ① 系统识别和
  - ② 控制器
- 它们作为两个不同的软件单元实现。

# 自适应控制系统——软件结构和分配

- 所选的硬件设备是非对称多处理器平台。
- 每个处理器运行一个软件任务。
- 这样的决定是基于各种任务的计算需求。





# 在一个异构多核处理器执行的案例

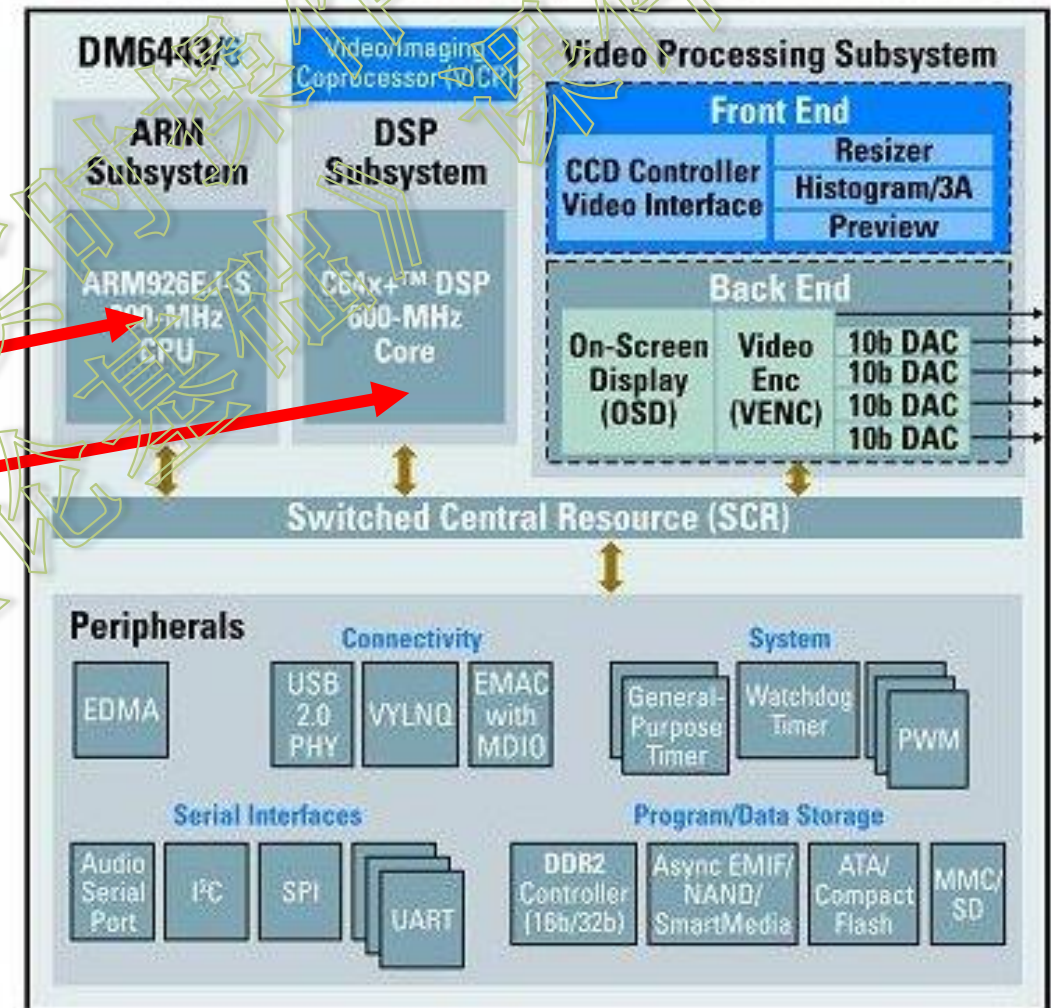
## 多核设计-异构的多处理器

- 每一个CPU核完成一个特定的工作

案例：TMS320DM644x

- 高度集成的多核处理器架构
- 基于TMS320C64x+™ DSP核和ARM926 处理器核
- 闭环控制分配给ARM处理器。
- 系统识别分配给DSP。

## TMS320DM644x™ Processors



# 第3部分

第3部分

软件问题

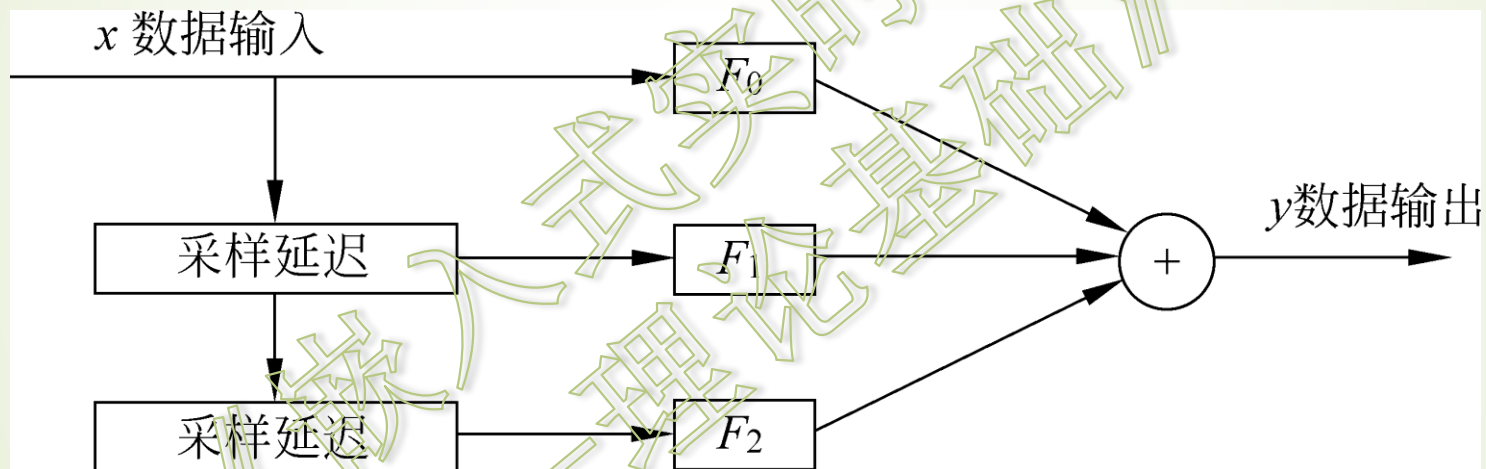
—  
基于一组数据操作的作业划分和分配

# 数字信号处理器算法案例

- 信号处理或滤波算法的简单示例如下：

$$y = F_0(x_0) + F_1(x_1) + F_2(x_2)$$

- 可以图示如下：



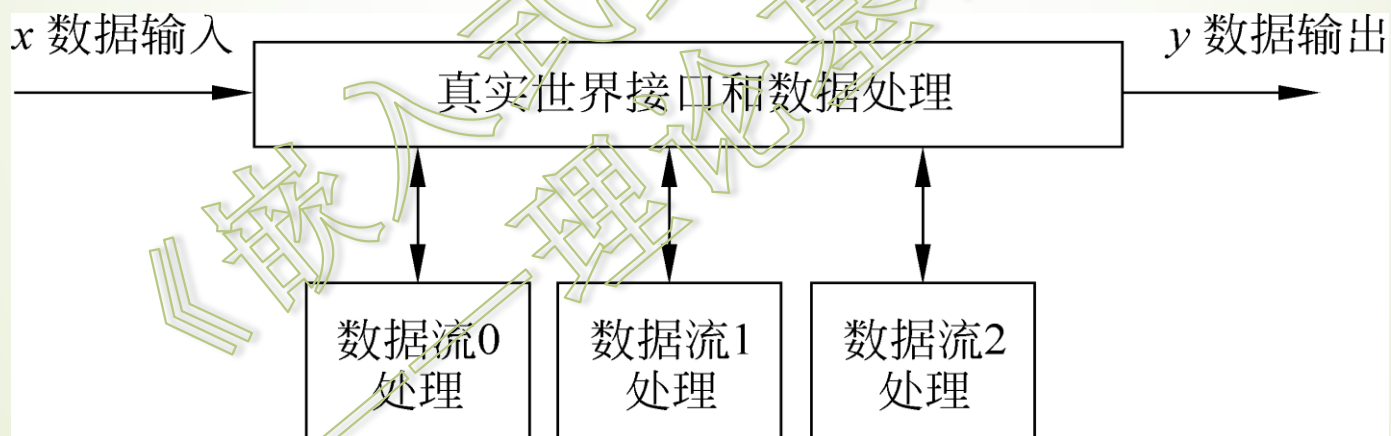
- 这种类型的处理在许多系统中都有需求，例如视频，音频，图像处理，雷达，声纳和模式识别。

- 多处理器在这些情况下表现出色。



# 信号处理器软件的分配

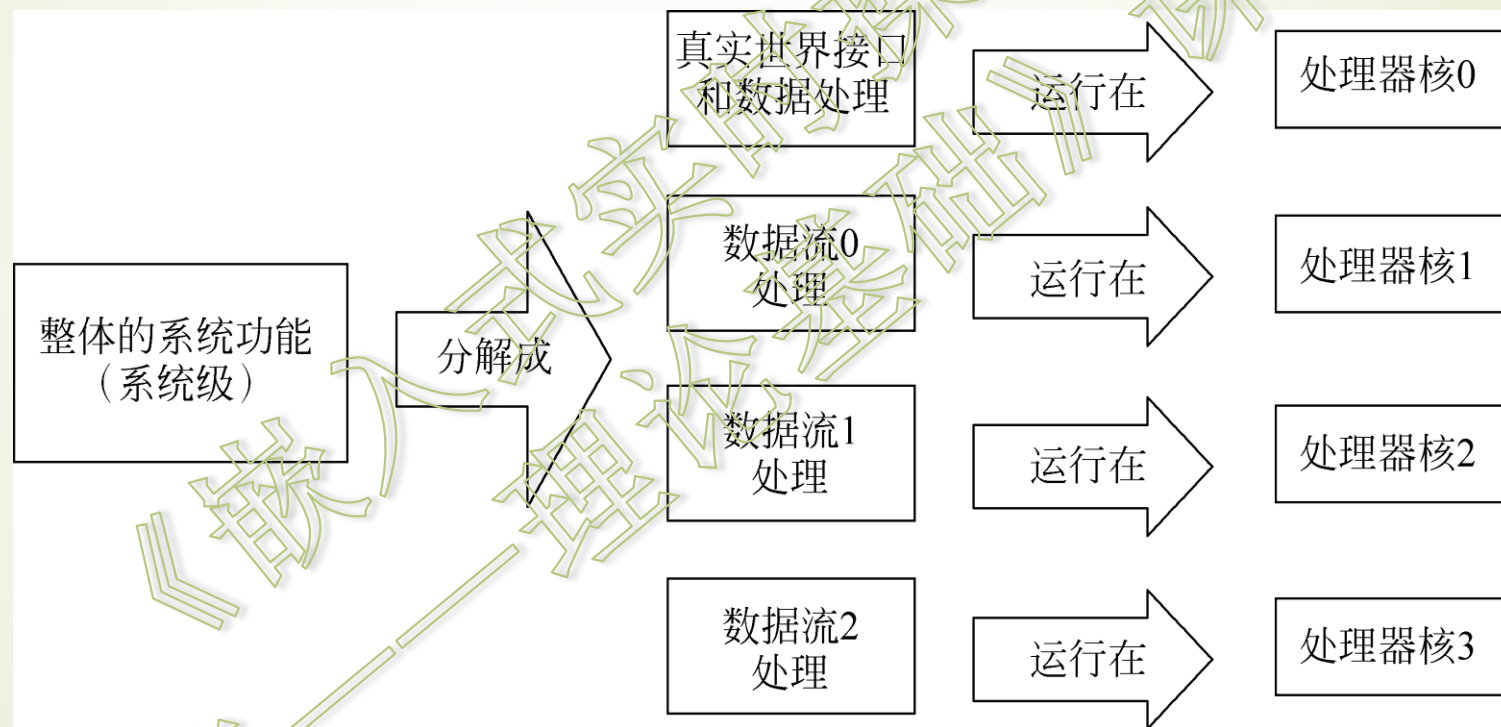
- 按照功能分配并不是构建软件的最佳方式。
- 相反，分配主要基于软件执行的数据处理流程。
- 但请注意，需要额外的操作来支持接口、数据处理等。





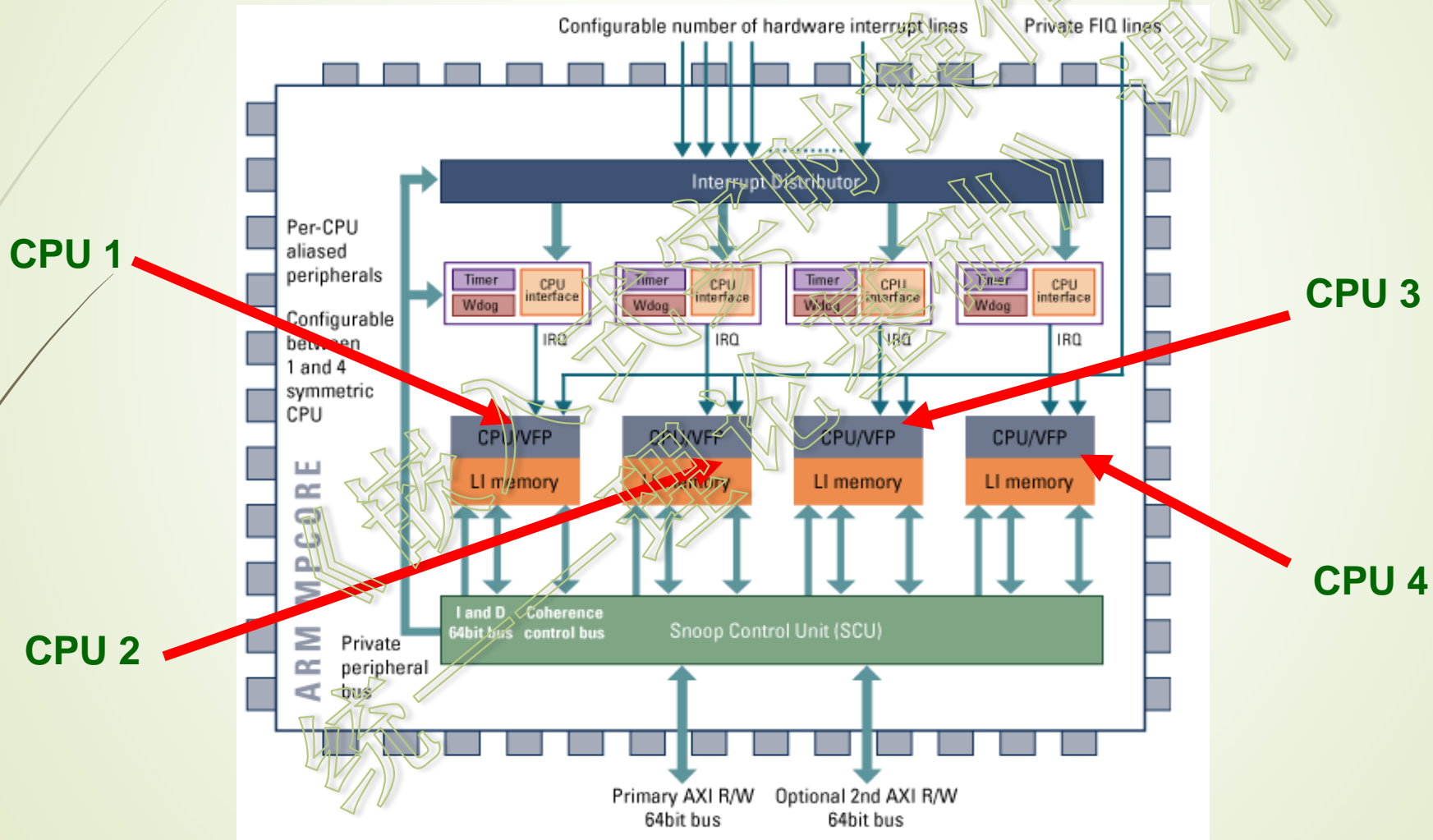
# 给处理器分配软件的实例

设计决策：使用四核对称多处理器。



# 支持SMP的 SoC 处理器核设计- ARM 11

- 使用对称多处理器时，处理单元是相同的。
- ARM 11是对称多处理器（多核）的一个示例。

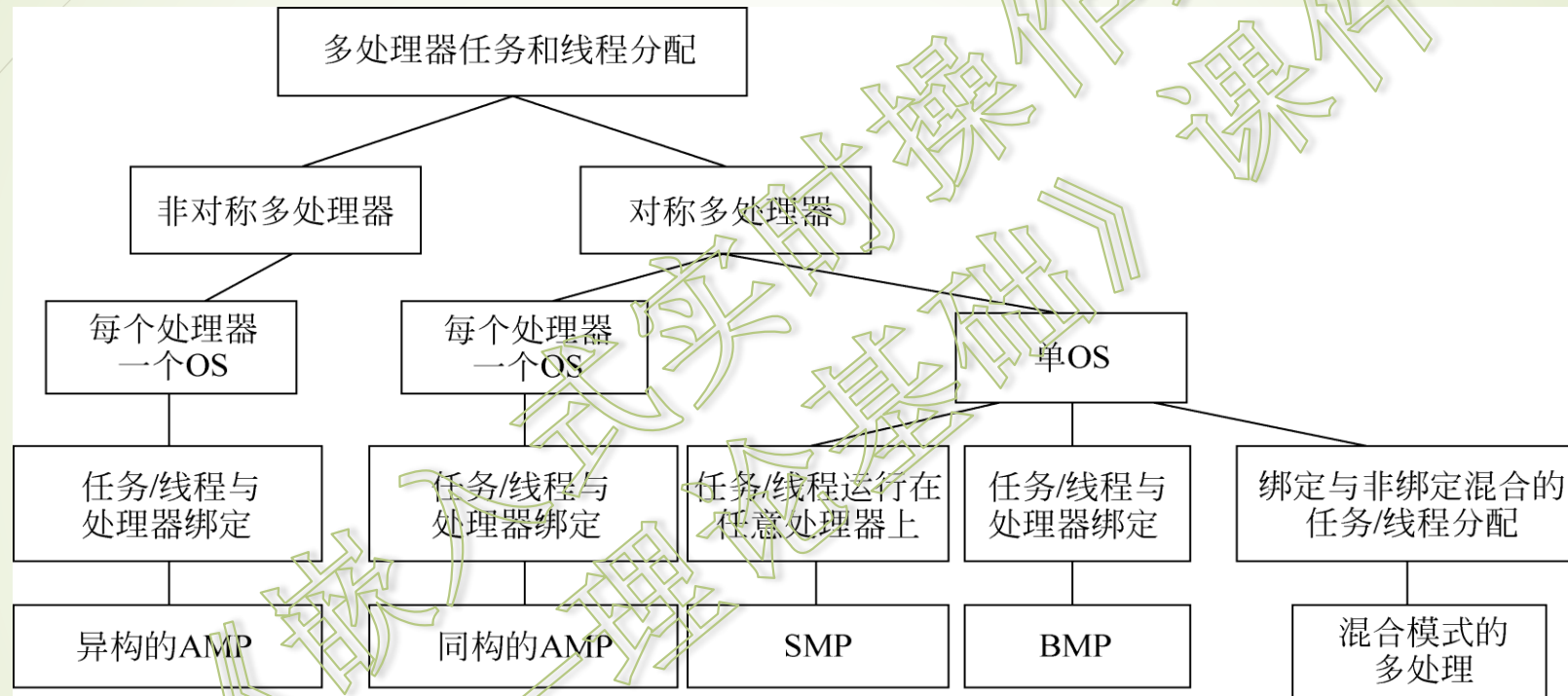


# 第4部分

第4部分

软件控制  
与  
执行问题

# 多处理器任务和线程分配



注释:

AMP — 同构多处理器  
SMP — 异构多处理器  
BMP — 绑定多处理器



# 非对称多处理器（AMP）特点

## AMP具备：

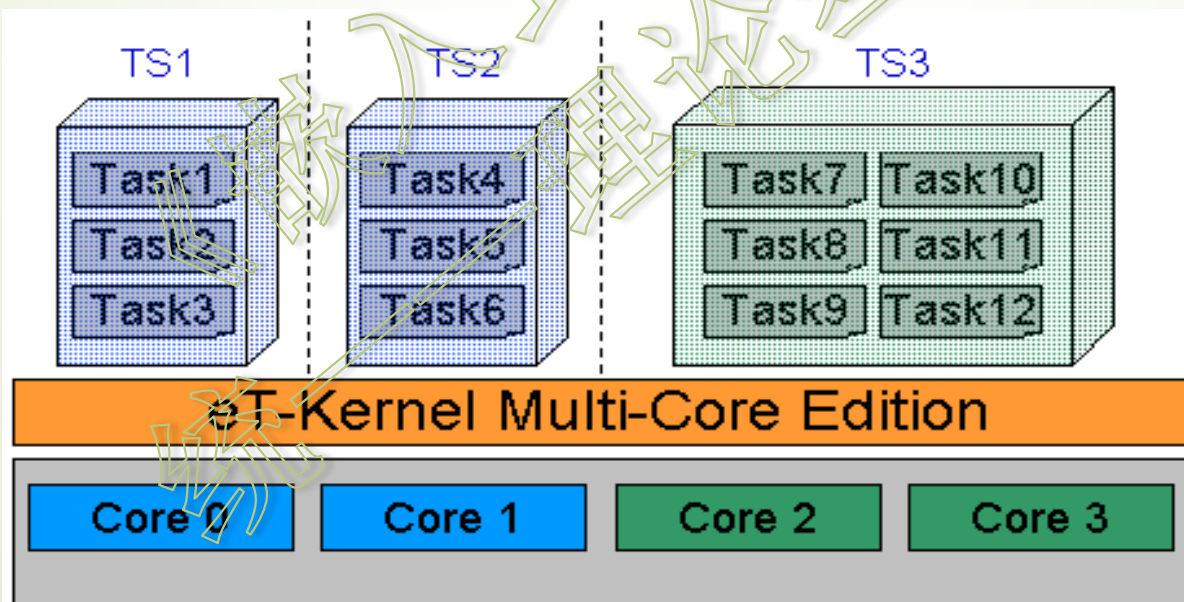
- 每个CPU内核执行专用作业，但其
- 操作与单个处理器设计完全相同。
- 行为和可预测性类似于单处理器设计。
- 如果软件适合特定类型的硬件，请在非对称多处理器上使用异构AMP。
- 如果需要混合使用操作系统或调度方法，请在对称多处理器上使用同构AMP。
- 所有方法都应该提高性能。

# 同构的多处理器(SMP)特点

SMP具备:

- 作业/任务动态分配给处理器内核（运行时）。
- 总计算负载分布在每个内核上。
- 一个操作系统控制所有 CPU 内核上的程序。
- 核心间通信和同步不需要特殊的操作系统服务。

RTOS 实例 - eT-Kernel for multicore processors



[www.esol.co.jp](http://www.esol.co.jp)

# SMP 潜在的问题

- 无法预测单个进程的执行顺序。
- 无法预测哪个 CPU 内核执行什么进程。
- 对于某些进程，可能无法保证实时行为。
- 如果从现有的单处理器设计移植到多处理器，可能会出现同步或互斥问题。
- 业界已开发绑定多处理（BMP）以最大程度地减少此类问题。



# 绑定多处理器(BMP)和多处理器混合模式

## BMP具备:

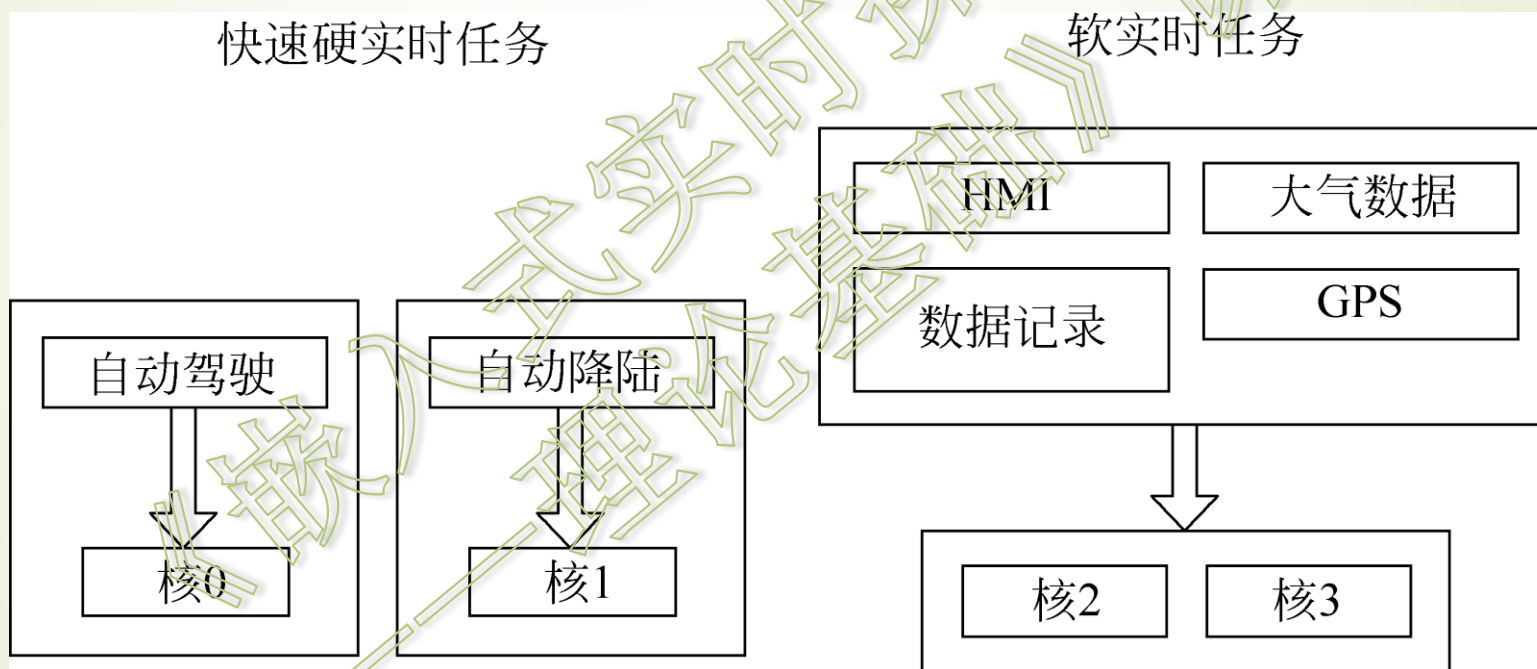
- 软件运行在一个同构多处理器上。
- 软件可以运行在任意一个处理器上。

## 混合模式具备:

- SMP和BMP的混合。
- 一些专用任务运行在指定的处理器上。
- 专用任务多是关键任务，可以工作在BMP模式。
- 非专用任务运行在SMP模式。



# 混合模式系统软件分配的案例



本章结束

# 版权说明和联系方式

- 课件由《嵌入式实时操作系统——理论基础》一书翻译团队成员何灵渊、何小庆、张爱华和付元斌编写，图书原作者提供原始素材，清华大学出版社提供部分图片。课件可用于非商业场合，比如教学和研究课题，商业使用需联系作者。
- 如果你是高校老师，希望以本书内容开设课程，需要全套的课件（PPT格式）可以联系：
  - ① 本书译者：何小庆老师 [xiaoqinghe@live.com](mailto:xiaoqinghe@live.com) 或者添加何老师微信 allanhexq（注明你的姓名+学校+专业）
  - ② 本书责编：刘星 [liux@tup.tsinghua.edu.cn](mailto:liux@tup.tsinghua.edu.cn) 电话：83470219 QQ：1468682976

注意： 需提供教材订购证明，获取全书的PPT课件。