SoC微体系结构设计

计算机科学与技术学院
张剑贤

教学目标与任务

- 培养学生理论联系实际应用的能力，了解微处理器技术发展的最新情况。
- 本课程主要讲述SoC系统的概念、设计方法和关键技术的实现。主要包括：VHDL硬件描述语言、系统存储接口的实现方法、系统指令集实现方法、系统设计实例及工程问题及技术发展的现状等相关内容。

课程内容安排

- （1）SoC设计概论（2学时）
- （2）FPGA结构分析（1学时）
- （3）VHDL硬件描述语言（12学时+4学时）
- （4）加法器与乘除法器设计（2学时+12学时）
- （5）SoC微体系结构存储器设计实现方法（2学时+4学时）
- （6）SoC微体系结构系统指令集实现方法（2学时+ 8学时）
- （7）SoC微体系结构系统设计实例（10学时+ 12学时）

课程内容安排

- （8）SoC微体系结构验证与测试方法（1学时）
- （9）SoC微体系结构典型实例及技术发展（2学时）
- （10）SOC优化设计（2学时）

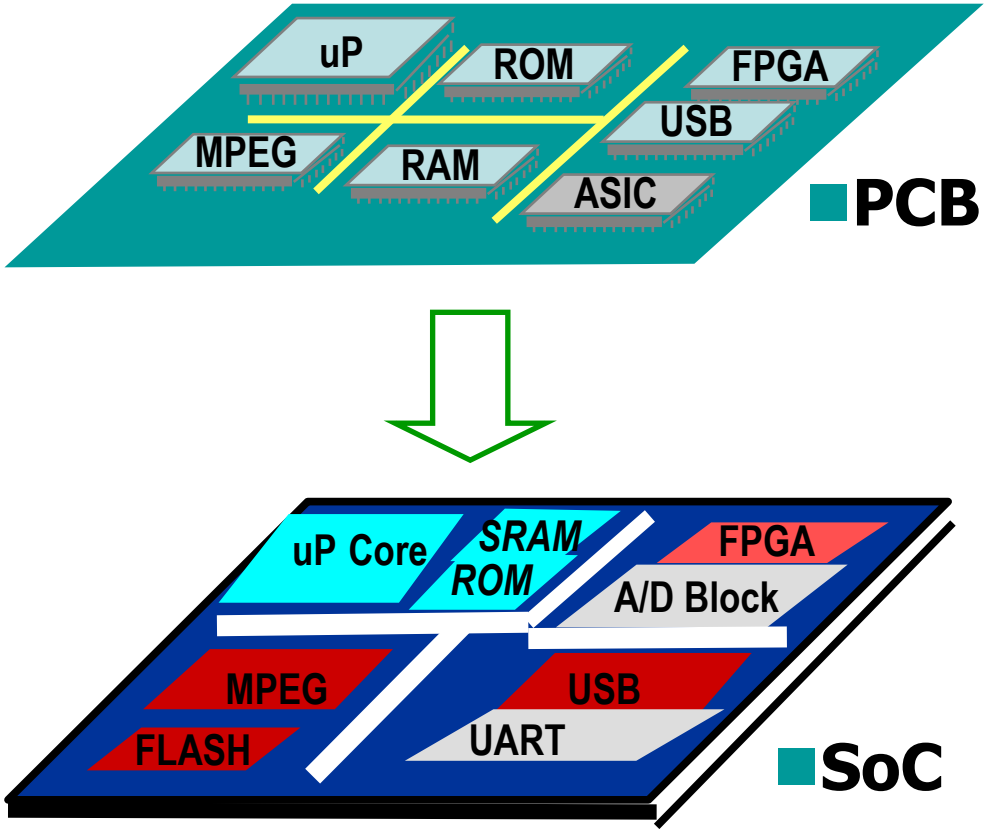
- 理论：36学时（18次课）
- 实验：40学时 （10次实验，2人一组）
- 考核方式：实验*50% + 期末考试50%

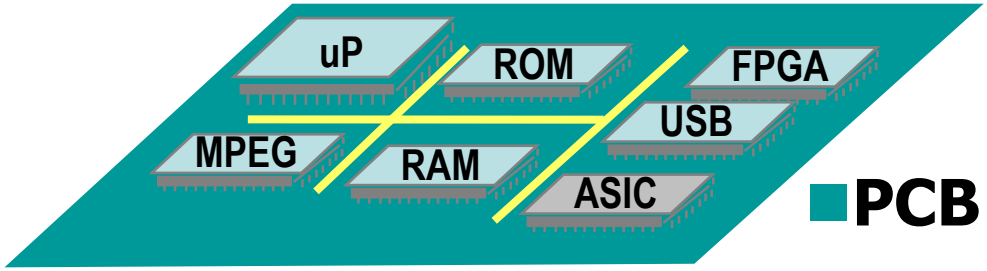
第一讲 SoC设计概论

- 1.1 SoC基本概念
- 1.2 SoC关键技术分析
- 1.3 SoC设计方法
- 1.4 SoC总线结构

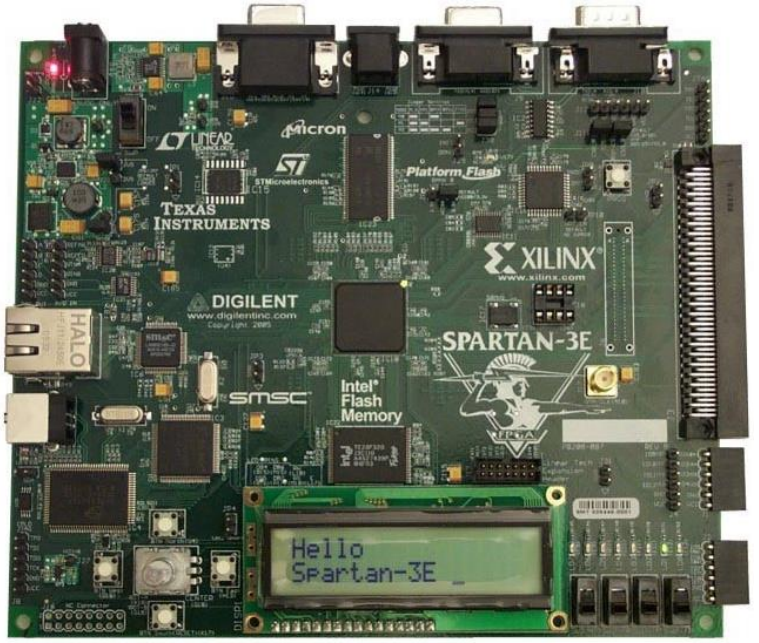
1.1 什么是SoC？

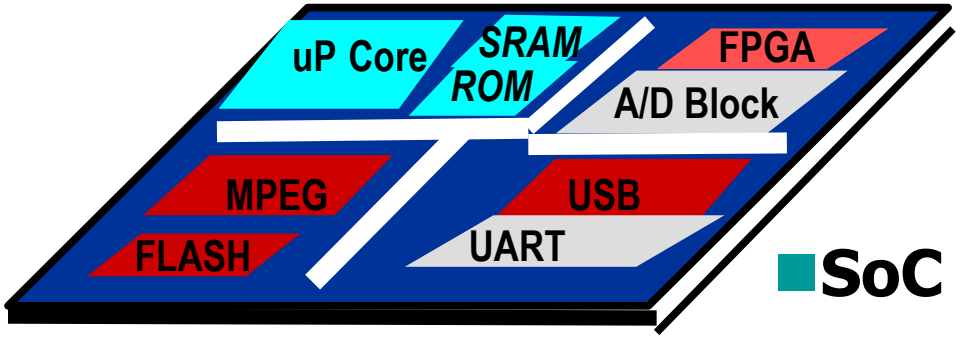
- 片上系统（System on Chip,SoC），是指在单一芯片上集成了数字电路、模拟电路、信号采集和转换电路、存储器、MPU、MCU、DSP、MPEG等，实现了一个系统的功能。


uP
ROM
FPGA
USB
MPEG
RAM
ASIC
PCB
uP Core
SRAM
ROM
FPGA
A/D Block
MPEG
USB
FLASH
UART
SoC

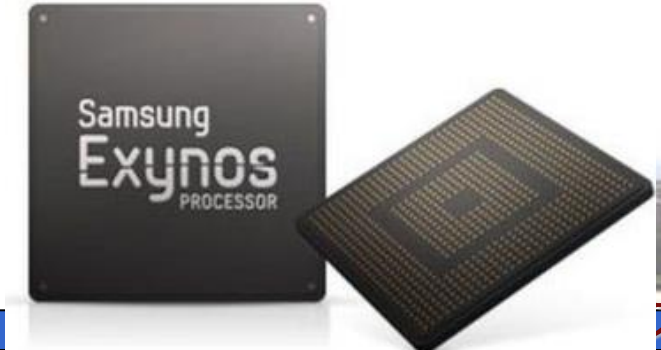

uP
ROM
FPGA
USB
MPEG
RAM
ASIC

PCB


TEXAS INSTRUMENTS
DIGILENT
XILINX
SPARTAN-3E
Intel Flash Memory
smsc
Hello
Spartan-3E


uP Core
SRAM
ROM
FPGA
A/D Block
MPEG
USB
FLASH
UART

SoC


Samsung
Exynos
PROCESSOR

1.1.1 SoC VS PCB

相对于PCB整机

- **微型化**：体积小、重量轻
- **工作速度↑**：传输路径短，寄生效应弱，芯片内部总线速度>>PCB板总线速度
- **功耗↓**：单个芯片功耗↑，但整个系统功耗↓，引线电容小，驱动能力要求低
- **可靠性↑**：焊点数↓，屏蔽效果好，干扰小

SoC化实例:手机

■SoC化前

■SoC化后

1.1.2 SoC特点

- 优点

　　体积小、功耗低、可靠性高、成本低以及更完善的功能和更高的性能指标。

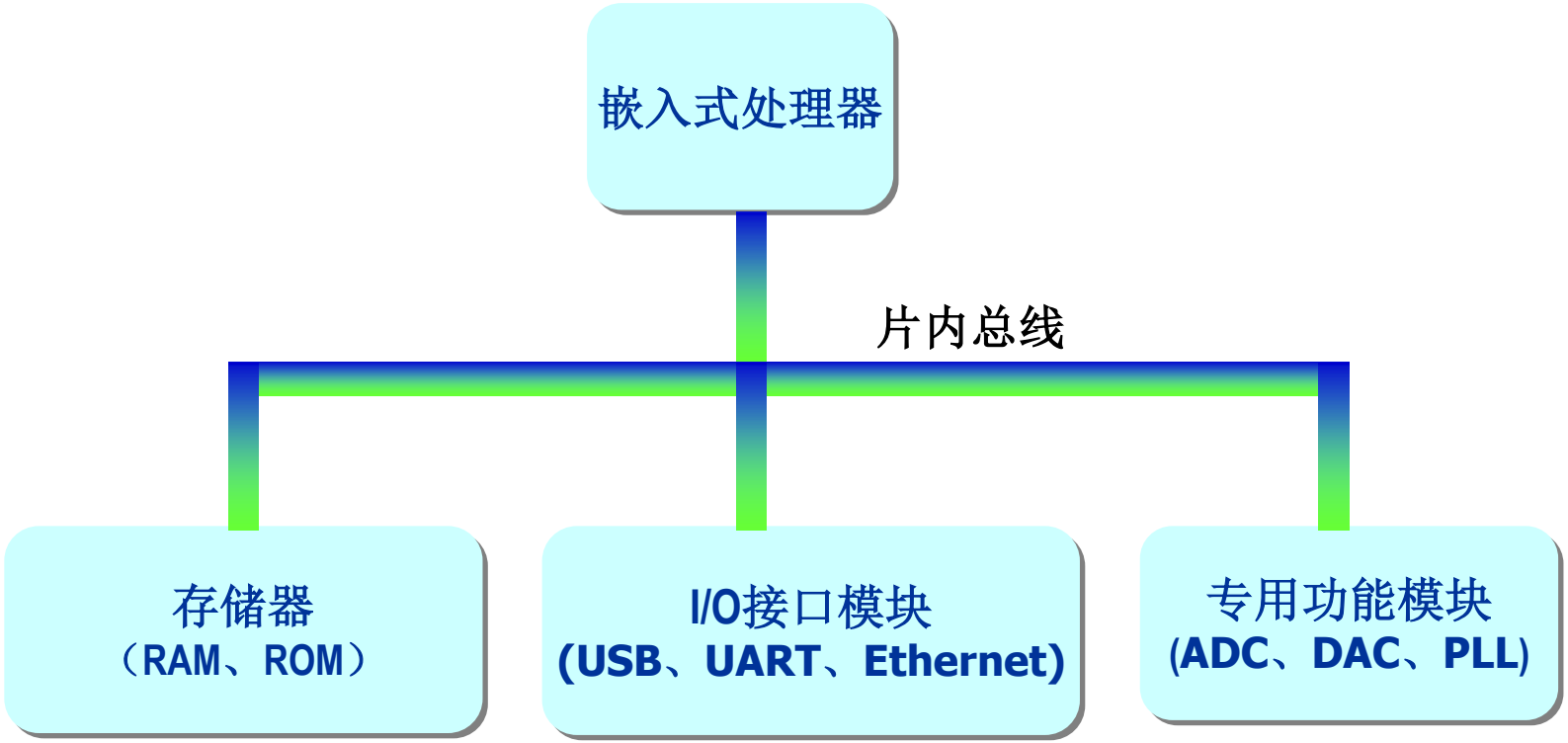
- 缺点

　　复杂性上升、设计成本高、开发时间长，完全改变了先前整机系统的总体设计方案。

1.1.3 SoC基本构成

- 嵌入式处理器核（如MPU、MCU或DSP）
- 存储器（如SRAM、SDRAM、Flash ROM）
- 专用功能模块（如ADC、DAC、PLL、2D/3D图形运算单元）
- I/O接口模块（如USB、UART、Ethernet等）等多种功能模块
- 片内总线（AMBA、Wishbone、Avalon等）

SoC基本结构


嵌入式处理器
片内总线
存储器
（RAM、ROM）
I/O接口模块
(USB、UART、Ethernet)
专用功能模块
(ADC、DAC、PLL)

1.1.4 SoC与计算机

- SoC是属于计算机与微电子学科交叉的新兴方向。
- 微电子方向注重电路级设计，包括管级电路设计、芯片版图设计、材料工艺实现等。
- 计算机方向注重于系统级设计，包括SoC结构、IP核间逻辑关系、片内总线结构设计、行为级/RTL级功能设计实现，FPGA验证、测试验证等。

1.1.4 SoC与计算机

- SoC是属于计算机与微电子学科交叉的新兴方向。
- 微电子方向注重电路级设计，包括管级电路设计、芯片版图设计、材料工艺实现等。
- 计算机方向注重于系统级设计，包括SoC结构、IP核间逻辑关系、片内总线结构设计、行为级/RTL级功能设计实现，FPGA验证、测试验证等。
- SoC强化了计算机基础理论知识之间的联系，为理论与实践的有机结合提供了有效的途径。

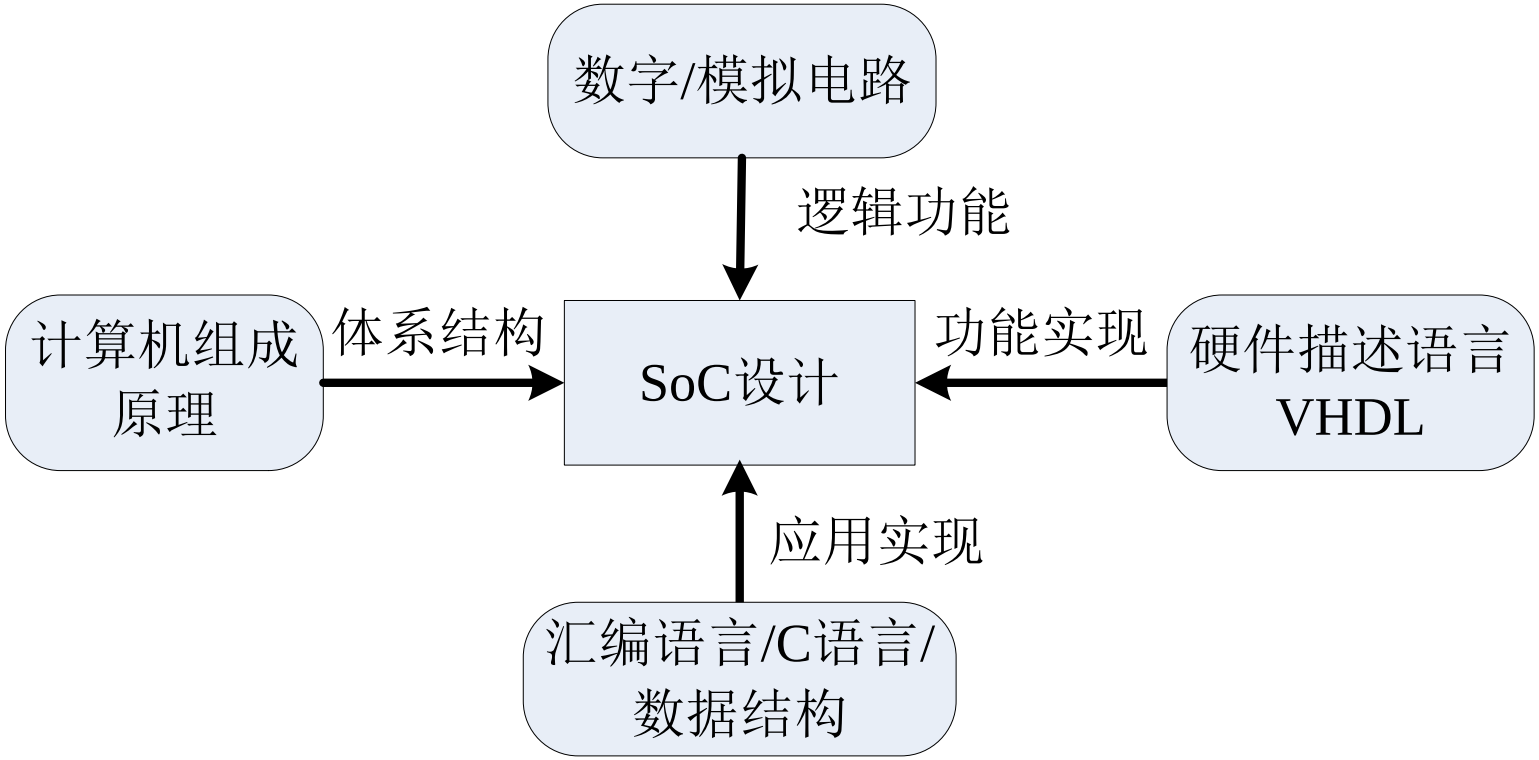
设置

1. SOC集成的IO模块包括以下哪几个

- A ADC
- B UART
- C SPI
- D USB

提交

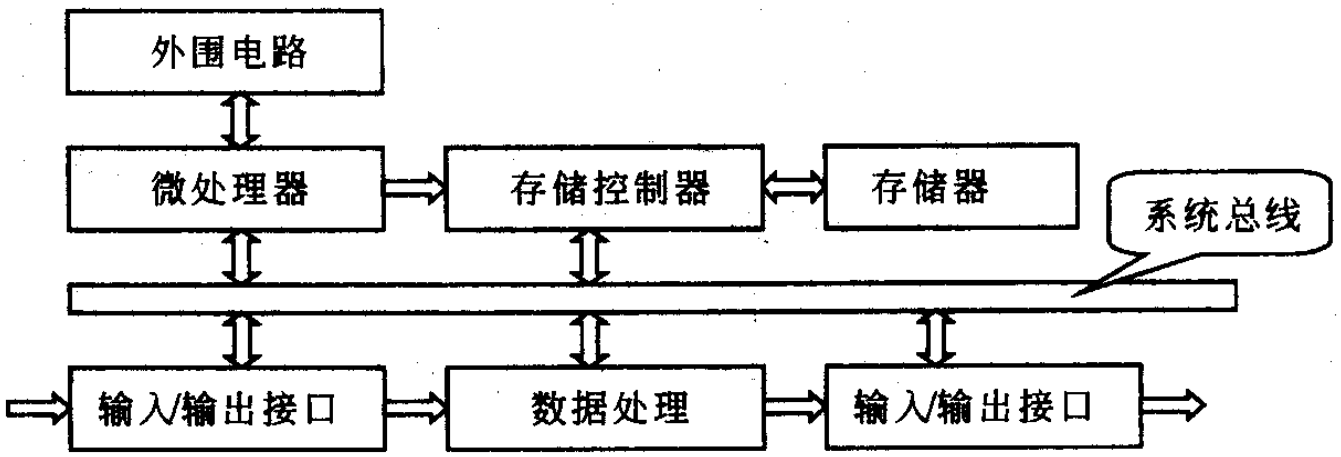
SoC与计算机相关基础课程



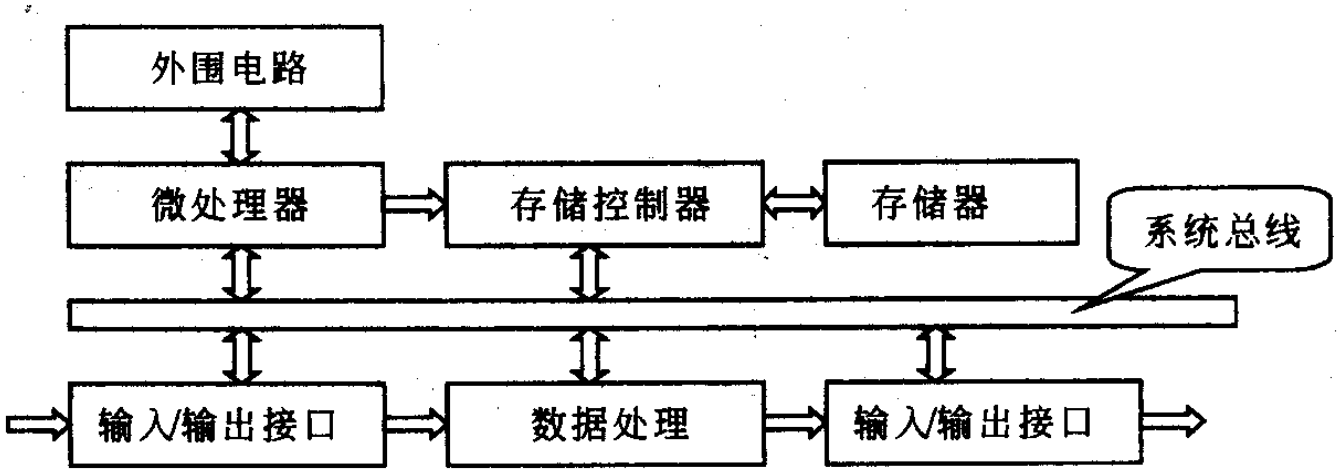
SoC类型

- 计算控制型
- 通信网络型
- 信号处理型

1.1.5 SoC类型--计算控制型

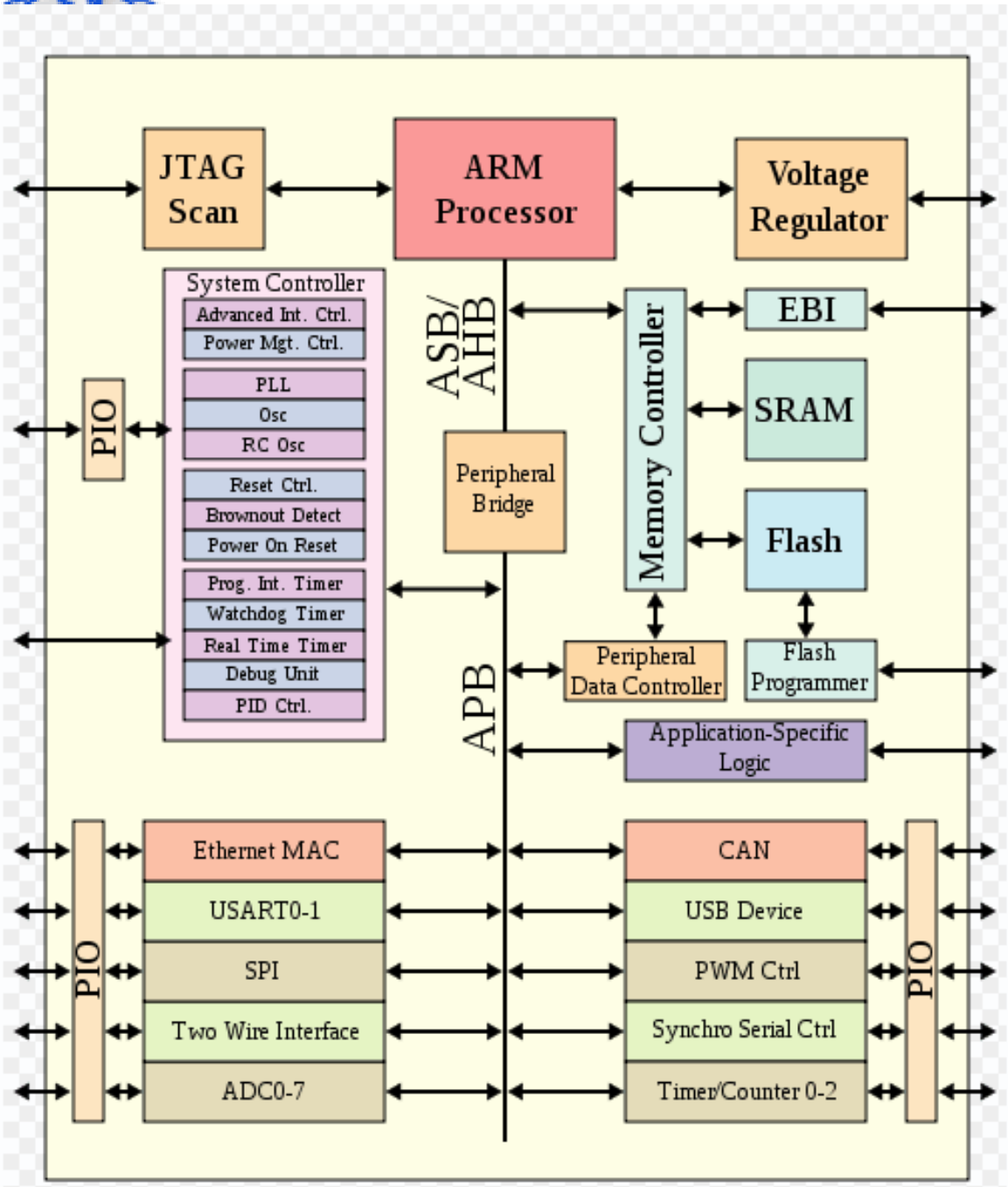


1.1.5 SoC类型--计算控制型

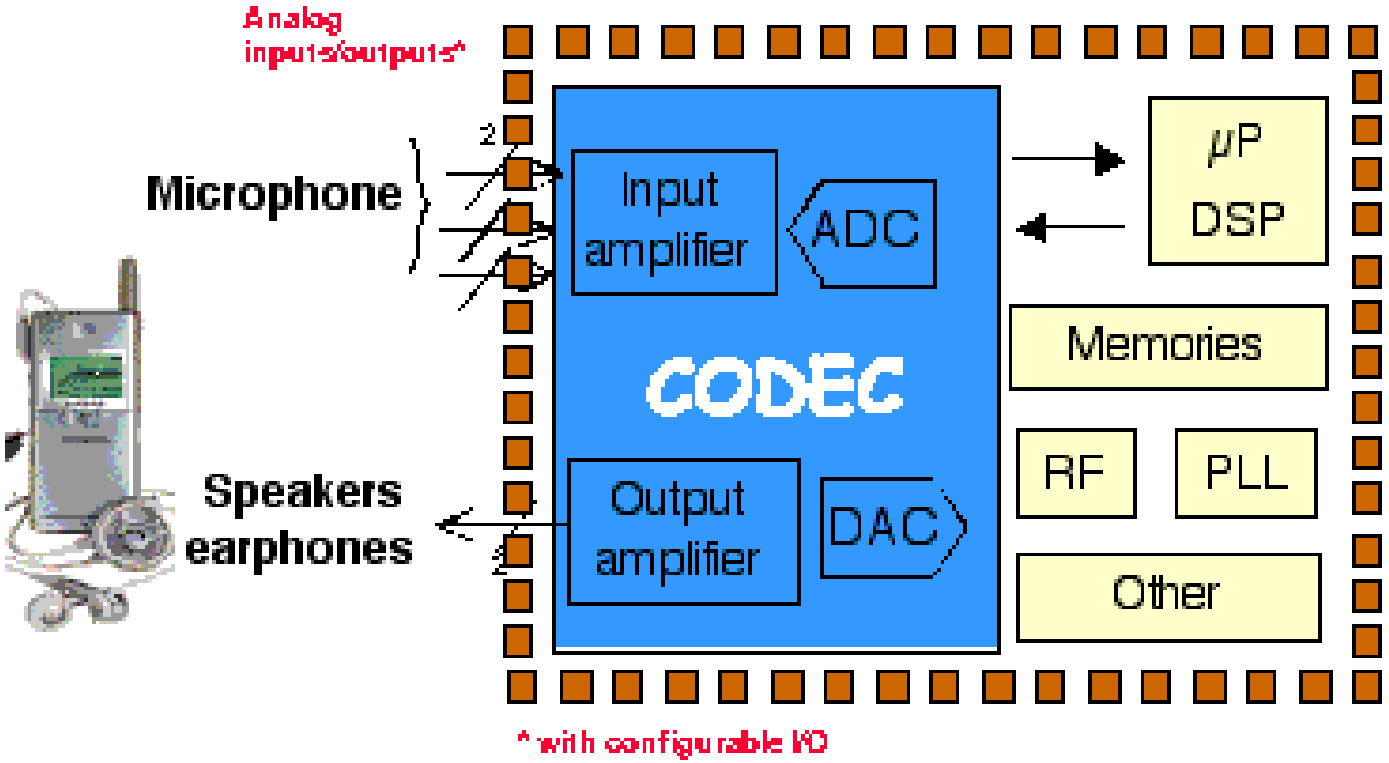


- **微处理器CPU**
  - CISC：硬件复杂，软件简化，指令执行效率低，功耗大，如Intel X86系列，微机操作系统（Windows系列），用于微机与工业机。
  - RISC：硬件简单，软件优化，指令执行效率高，功耗低，如ARM系列，嵌入式操作系统（如Palm OS、Windows CE），用于信息家电、个人电器、移动通信。
- **数字信号处理器DSP**
  - 通用DSP：强调高性能、高速，～1GHz
  - 嵌入式DSP：强调多功能、低功耗，～100MHz

计算控制型SoC


JTAG Scan
ARM Processor
Voltage Regulator
System Controller
Advanced Int. Ctrl.
Power Mgt. Ctrl.
PLL
Osc
RC Osc
Reset Ctrl.
Brownout Detect
Power On Reset
Prog. Int. Timer
Watchdog Timer
Real Time Timer
Debug Unit
PID Ctrl.
PIO
ASB/ AHB
Peripheral Bridge
Memory Controller
EBI
SRAM
Flash
Peripheral Data Controller
Flash Programmer
APB
Application-Specific Logic
Ethernet MAC
USART0-1
SPI
Two Wire Interface
ADC0-7
CAN
USB Device
PWM Ctrl
Synchro Serial Ctrl
Timer/Counter 0-2
PIO
PIO

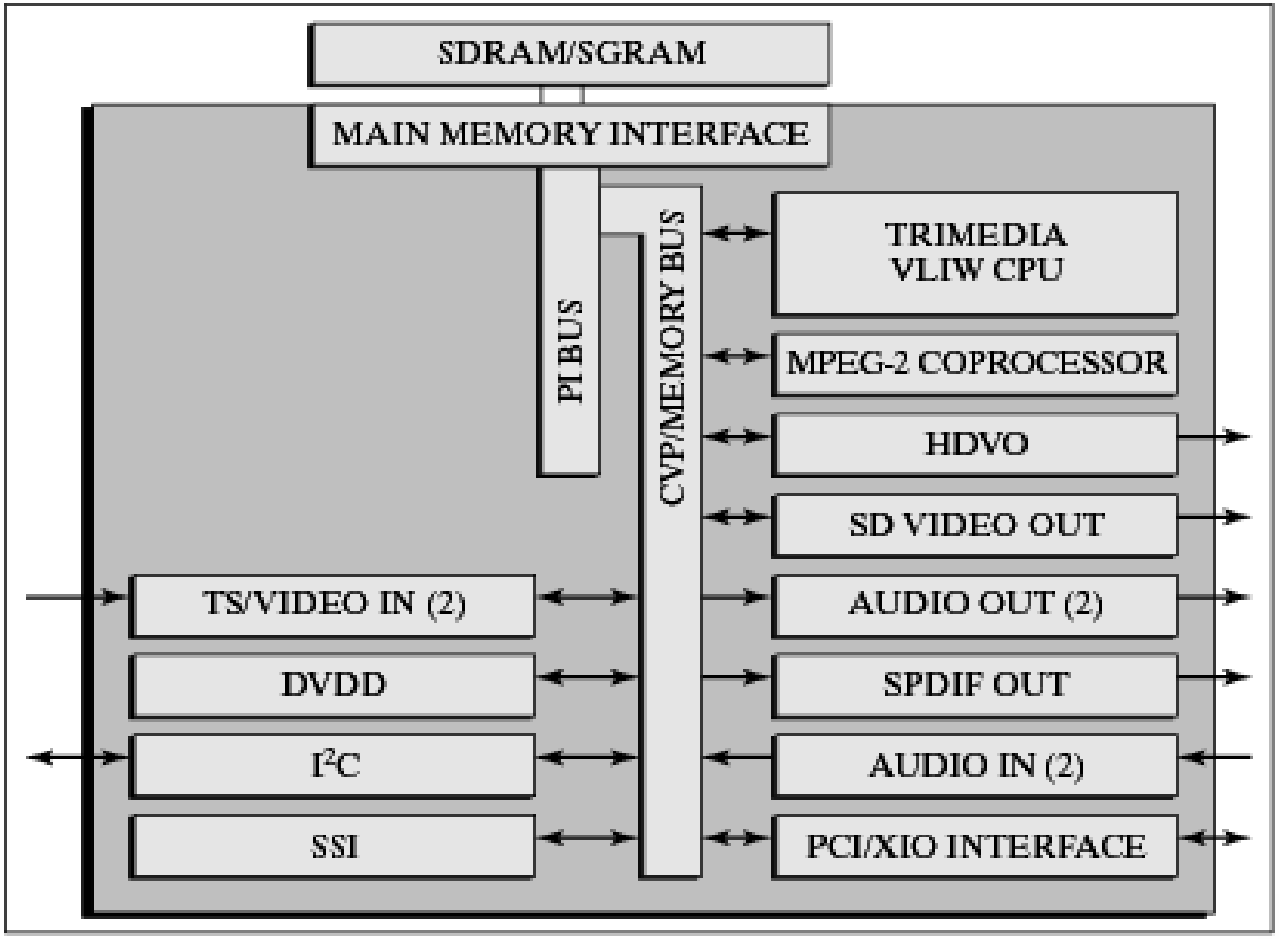
1.1.5 SoC类型——通信网络型



■移动电话手机的SoC

- **前放/功放：** 800MHz～5GHz，RF CMOS
- **基带：** 100MHz，带CPU、SRAM、ROM等，标准CMOS
- **调制：** GSM，CDMA，WLAN，TCP/IP，Bluetooth

1.1.5 SoC类型——信号处理型



- 编解码：语音PCM，音乐MP3，图片JPEG，视频MPEG
- 信号采集：声音（话筒），图像（CCD，CMOS）
- 信号输出：声音（扬声器），图像（LCD、CRT）

- 用于HDTV的飞利浦SoC芯片Nexperia

1.1.6 SoC应用领域

SoC的应用领域非常广泛

- 消费电子（包含白色家电和黑色家电，如数字电视、DVD、STB、家庭网关、MP3播放器）
- 通信设备（包含各种终端设备、接入设备和交换设备，如手机和路由器）
- 控制类设备（包含汽车电子、仪器仪表、军事电子、工业控制、医疗电子等，如智能化家用仪器仪表）

设置

2. 手机属于哪一种SOC类型

A 计算控制型

B 信号处理型

C 通信网络型

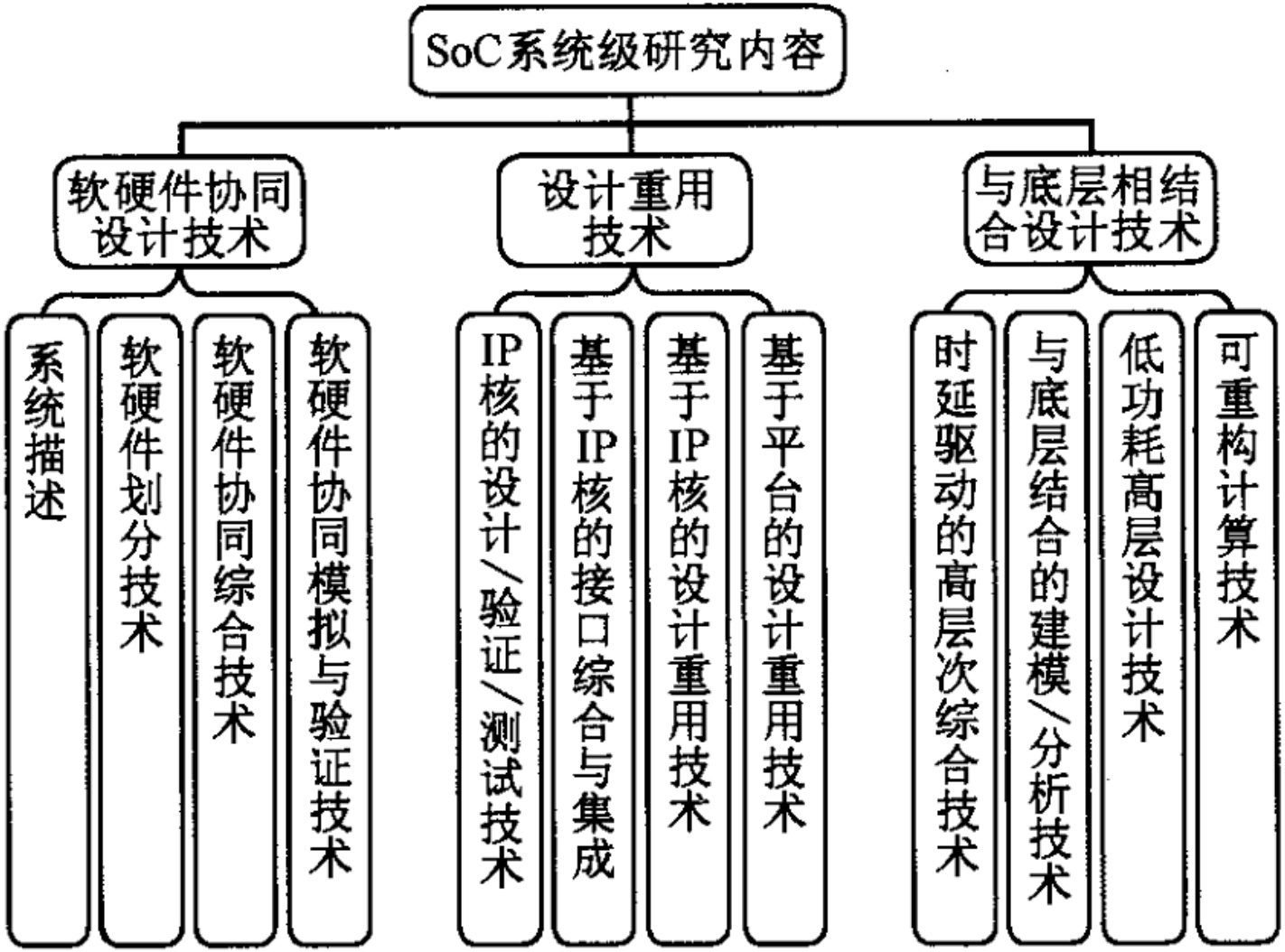
D 无线网络型

提交

1.1.7 SoC系统级研究内容

- 软硬件协同设计技术
- 设计重用技术
- 与底层相结合设计技术

1.1.7 SoC系统级研究内容



1.2 SoC设计关键技术

- 设计重用技术
- 低功耗设计技术
- 软硬件协同设计
- 总线架构
- 可测试性设计
- 设计验证
- 物理综合

1.2.1 设计重用技术

◆基于IP的模块级重用

建立在IP芯核基础上的，它是将已经验证的各种超级宏单元电路模块制成芯核，方便设计时使用。

◆基于平台的系统级重用

1.2.1 设计重用技术

◆基于IP的模块级重用

建立在IP芯核基础上的，它是将已经验证的各种超级宏单元电路模块制成芯核，方便设计时使用。

◆基于平台的系统级重用

- 平台是一组关于虚拟组件与体系结构框架的库，在平台中包含一些可集成的并且预先验证的软硬件IP、设计模型、EDA工具与软件配套工具、库单元等，同时定义了一套通过体系结构探索/集成/验证支持快速产品开发的设计方法学。
- 基于IP设计重用技术的扩展，延伸了设计重用的理念，强调系统级重用。基于平台的设计方法要求提供面向特定应用领域的设计模板。

1.2.1 设计重用技术

◆IP核是指经过反复验证过的、具有特定功能的，可重复利用的逻辑块或数据块，用于专用集成电路（ASIC）或者可编辑逻辑器件（FPGA）。

设置

3. IP核有哪几种类型?

- A 软核
- B 固核
- C 众核
- D 硬核

提交

IP芯核的分类

- 软核
- 固核
- 硬核

类型	定义	使用灵活性	时序性能 可预测性
Soft Cores (软核)	行为级或 RTL 级 HDL 源代码	灵活度高，可修改设计代码，与实现工艺无关。	IP 很难保护，时序性能无法保证，由使用者确定。
Firm cores (固核)	完成软核所有的设计外，还完成了门电路级综合和时序仿真等设计环节，一般以门电路级网表形式提交用户使用。	部分功能可修改，采用指定的实现技术，与实现工艺相关 。	关键路径时序可控制
Hard cores (硬核)	基于某种半导体工艺的物理设计，已有固定的拓扑布局和具体工艺，并已经过工艺验证，具有可保证的性能。提供电路物理结构掩模版图和全套工艺文件。	不能修改设计，必须采用指定实现技术。	包含工艺相关的布局和时序信息， IP 很容易保护，多数的处理器和存储器。

1.2.2低功耗设计技术

- 芯片功耗主要由开关功耗、短路功耗和漏电流功耗等组成。

1.2.2低功耗设计技术

- 芯片功耗主要由开关功耗、短路功耗和漏电流功耗等组成。
- 降低功耗要从SoC的多层次立体角度出发，研究电路实现工艺、输入向量控制、多电压技术、功耗管理技术以及软件低功耗技术等多方面综合解决。

1.2.2低功耗设计技术

- 芯片功耗主要由开关功耗、短路功耗和漏电流功耗等组成。
- 降低功耗要从SoC的多层次立体角度出发，研究电路实现工艺、输入向量控制、多电压技术、功耗管理技术以及软件低功耗技术等多方面综合解决。
- 功耗的降低是有限度的。首先是要限定在性能的约束范围内，否则功耗的降低可能会导致性能的大幅度降低。

1.2.2 低功耗设计技术

- 工艺级低功耗技术
- 电路级低功耗技术
- 逻辑（门）级低功耗技术
- RTL级（寄存器传输级）低功耗技术
- 体系结构级低功耗技术
- 算法级低功耗技术
- 系统级低功耗技术

(1)工艺级低功耗技术

- 降低电源供电电压，减少跳变功耗
  - 通过开发系统的并行性和流水线；
  - 根据用户对电路性能的不同要求，通过操作系统动态控制时钟频率和电源电压；
  - 根据性能的要求，实时改变供电电压。
- 多阈值工艺 MTCMOS (Multi-Threshold VT CMOS)
- 变阈值工艺VTCMOS (Variable Threshold VT CMOS)

(2)电路级低功耗技术

- 减摆幅 $P_s = AVCV_{swing}f$
- 电荷再循环总线结构（Charge Recycling Bus）它把整个电势差分几等份，利用总线各数据位电容上存储的电荷电势的变化来传输数据。

(3)门级低功耗技术

- 主要通过低电压实现低功耗技术，主要采用互补CMOS实现来实现。

(4)寄存器传输级（RTL）低功耗技术

- RTL 低功耗技术主要从降低不希望的跳变入手。
- 降低的方法主要是消除其产生的条件，如延迟路径平衡、用时钟信号同步减少故障、结构重构。

(5)体系结构级低功耗技术

- 并行技术
- 流水线技术
- 预计算技术

(6)算法级低功耗技术

- 总线翻转译码技术
- 编码技术

(7)系统级低功耗技术

- 门控时钟技术
- 异步电路技术

设置

4. 体系结构级低功耗技术包括哪些

A 编码技术

B 流水线技术

C 并行技术

D 预计算技术

提交

1.2.3软硬件协同设计技术

- 在传统的设计方法中，硬件和软件是分开进行的，最终的集成要在硬件投片完成后才能进行，在软件中不能纠正的设计错误只能通过硬件的修改和重新投片来解决，严重影响了投放市场的时间，提高了设计成本。

1.2.3软硬件协同设计技术

- 在传统的设计方法中，硬件和软件是分开进行的，最终的集成要在硬件投片完成后才能进行，在软件中不能纠正的设计错误只能通过硬件的修改和重新投片来解决，严重影响了投放市场的时间，提高了设计成本。
- 软硬件协同设计方法强调软件和硬件设计开发的并行性和相互反馈，克服了传统方法中把软件和硬件分开设计带来的种种弊端，能协调软件和硬件之间的制约关系，达到系统高效工作的目的。

1.2.3 软硬件协同设计关键技术

- 系统建模
- 软硬件划分技术
- 软硬件协同综合
- 软硬件协同仿真与验证

1.2.3.1系统建模

- 目的是在最高抽象层次上利用某种高级语言， 如C/C++，SystemC或统一建模语言（UML）等描述整个系统行为，获取用户功能需求和约束要求，验证需求分析的正确性。
- 全面描述系统功能，精确建立系统模型，深入挖掘软硬件之间的协同性。
- 明确体现性能描述、功能特点、技术指标、约束条件等因素。

(1)系统描述模型

- 离散事件模型
- 有限状态机模型
- 通信进程网络模型
- Petri网模型
- 任务流图模型
- 控制数据流图模型

(2)系统模型要求

- 采用形式化规范，应用逐步细化求精的思想，实现可变粒度的层次化任务描述能力；

(2)系统模型要求

- 采用形式化规范，应用逐步细化求精的思想，实现可变粒度的层次化任务描述能力；
- 并通过控制机制指导控制相关性，捕获其并发性、时序与通信关系；

(2)系统模型要求

- 采用形式化规范，应用逐步细化求精的思想，实现可变粒度的层次化任务描述能力；
- 并通过控制机制指导控制相关性，捕获其并发性、时序与通信关系；
- 将系统模型与底层实现相关联， 通过一系列的细化与变换规则，完成功能任务到实现的映射；

(2)系统模型要求

- 采用形式化规范，应用逐步细化求精的思想，实现可变粒度的层次化任务描述能力；
- 并通过控制机制指导控制相关性，捕获其并发性、时序与通信关系；
- 将系统模型与底层实现相关联， 通过一系列的细化与变换规则，完成功能任务到实现的映射；
- 支持快速生成系统原型，有利于在系统级进行功能验证与性能评价。

设置

5. 以下哪种模型适合描述并发、竞争及同步的特性

A 有限状态机

B 离散事件

C Petri网

D 任务流图

提交

1.2.3.2 软硬件划分

- 软硬件划分是在系统描述与建模层次的分析结果上，将系统功能合理地划分为软件和硬件实现部分，使系统性能与成本最优。
- 划分结果力求提高速度、缩小面积、降低成本、减少功耗。
- 软硬件划分是一个NP难问题。

1.2.3.2 软硬件划分

- 根据SoC系统需求，结合成本、功耗、面积、实时性、和可靠性等性能参数，研究满足系统约束的各种优化算法的目标函数，探讨各种优化算法的初始解的生成、参数设置及收敛条件，设计软硬件划分的最优化算法。

1.2.3.2 软硬件划分

模块名称	软件实现		硬件实现	
	软件成本	软件功耗	硬件成本	硬件功耗
A	5	8	10	2
B	10	10	20	5
C	8	15	10	8

纯硬件实现：成本40，功耗15

纯软件实现：成本23，功耗33

1.2.3.2 软硬件划分

模块名称	软件实现		硬件实现	
	软件成本	软件功耗	硬件成本	硬件功耗
A	5	8	10	2
B	10	10	20	5
C	8	15	10	8

纯硬件实现：成本40，功耗15

纯软件实现：成本23，功耗33

23≤系统成本≤40

15≤系统功耗≤33

设计要求：系统成本≤35，系统功耗≤20

设置

6.满足设计要求（系统成本≤35，系统功耗≤20）的软硬件划分方案是

A A软件，B硬件，C硬件

B A硬件，B硬件，C软件

C A硬件，B软件，C硬件

D A软件，B硬件，C软件

提交

1.2.3.2 软硬件划分

模块名称	软件实现		硬件实现	
	软件成本	软件功耗	硬件成本	硬件功耗
A	5	8	10	2
B	10	10	20	5
C	8	15	10	8

设计要求：系统成本≤35，系统功耗≤20
A模块硬件实现，成本10，功耗2
B模块软件实现，成本10，功耗10
C模块硬件实现，成本10，功耗8
系统成本=30，系统功耗=20

1.2.3.3软硬件协同综合

- 软硬件协同综合是利用设计中的各种资源(如系统模型、软/硬件模块等) 生成最优的通信体系结构，实现从功能到结构再到实现的转换，同时满足系统性能与代价约束。
- 通信体系结构综合—软硬件接口
- 软件综合—软件构件
- 硬件综合—硬件IP

1.2.3.4软硬件协同仿真与验证

- 系统评估与验证是检验SoC设计的逻辑、功能、时间特性等是否满足用户需求的过程。
- 模块/IP核级验证
- 软硬件协同仿真验证
- FPGA验证

1.2.3.4软硬件协同仿真与验证

- 黑盒验证通过设计顶层接口，验证哪些与设计实现技术无关的功能，不能直接访问设计内部状态，可控性差、可测性差。

1.2.3.4软硬件协同仿真与验证

- 黑盒验证通过设计顶层接口，验证哪些与设计实现技术无关的功能，不能直接访问设计内部状态，可控性差、可测性差。
- 白盒验证保证设计实现相关技术的功能正确实现，黑盒的补充，对内部结构完全可控可见，但是不可移植。

1.2.3.4软硬件协同仿真与验证

- 黑盒验证通过设计顶层接口，验证哪些与设计实现技术无关的功能，不能直接访问设计内部状态，可控性差、可测性差。
- 白盒验证保证设计实现相关技术的功能正确实现，黑盒的补充，对内部结构完全可控可见，但是不可移植。
- 灰盒验证根据设计的内容结构写Testcase，从设计顶层接口进行控制与观察，验证是否实现了一些主要特性，而不关心设计方法。

设置

7.软硬件协同综合包括哪些

A 软件模块到硬件模块接口综合

B 硬件模块到软件模块接口综合

C 上位机软件综合

D 硬件IP核综合

提交

1.2.4总线结构

- 对SoC上芯核和电路模块等的互连常采用单总线、多总线和片上网络的方式。
- SoC总线规范需要定义各个模块之间的初始化、仲裁、请求传输、响应、发送接收等过程中的驱动、时序、策略等关系。

总线结构特点

- SoC总线要尽可能简单。
- SoC的总线应有较大的灵活性。
- SoC的总线要尽可能降低功耗。

1.2.5可测试性设计

面临的最大挑战是如何降低测试成本。

SoC芯核的测试方法

- 并行直接接入
- 串行扫描链接入
- 设置专门的针对芯核的测试结构

1.2.6设计验证

- 设计验证是SoC设计中不可或缺的重要组成部分。
- 验证的目的是确保所设计的SoC满足系统规范中定义的功能要求，这是保证SoC设计正确性的关键。

1.2.6设计验证

- 设计验证是SoC设计中不可或缺的重要组成部分。
- 验证的目的是确保所设计的SoC满足系统规范中定义的功能要求，这是保证SoC设计正确性的关键。

➢ IP核或电路模块的验证
➢ SoC的全功能验证
➢ 软硬件协同验证
➢ FPGA验证

1.2.7物理综合

物理综合过程分为初始规划、RTL规划和门级规划等多个阶段。

- 信号完整性与时序收敛分析

➢ 完整性问题（串扰效应、噪声问题、天线效应、电迁移、自热问题以及电压降问题）

➢ 时序收敛与设计参数相互依赖（时序与面积、面积与功耗、时序与布局）

1.3 SoC系统级设计方法

◆自顶向下

美国加州大学Irvine分校嵌入式系统研究小组的基于 SpecC的逐层细化求精设计方法。

◆自底向上

法国TIMA实验室系统级综合小组的基于组件的多处理器核SoC设计方法。

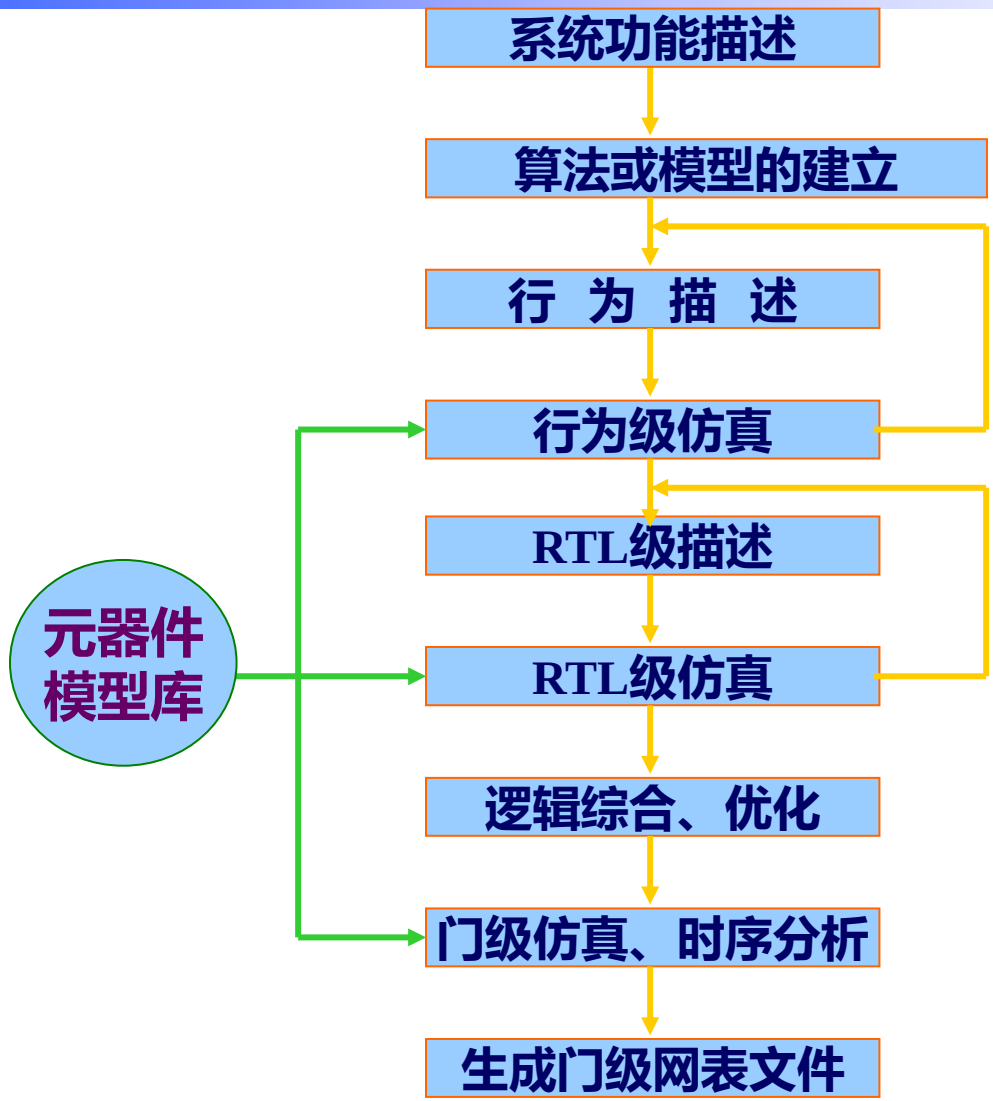
◆上下结合(中间相遇)

美国加州大学Berkeley分校CAD研究小组的基于平台的设计方法。

1.3 SoC系统级设计方法

系统级设计方法	设计模型的优点	局限性说明
自顶向下细化求精设计方法 UCI:SpecC	符合软件开发者、硬件设计师设计思路； 易于定义层次关系，明确层次行为、结构和语义； 易于开发建模/划分/综合/仿真工具.	较强地依赖于某种系统级设计语言； 设计重用的效率降低，开发的产品只能定制于某一种应用.
自底向上搭积木设计方法 TIMA:CBD	易于遵循设计重用思路； 能简化设计流程，加快设计速度； 能扩充对体系结构探索的能力.	系统的集成较为困难，通信接口综合问题比较严重； 较强地依赖与底层环境的支撑.
上下结合分而治之设计方法 UCB:PBD	易于遵循计算与通信、行为与结构分离设计原则； 与具体的设计语言无关； 开发的产品适用于一类应用，有较强的可编程性、灵活性、扩展性.	平台定义比较复杂，需兼顾软件开发与硬件设计； 难于开发面向平台的自动化综合与验证工具.

SoC设计流程


系统功能描述
算法或模型的建立
行 为 描 述
行为级仿真
RTL级描述
元器件
模型库
RTL级仿真
逻辑综合、优化
门级仿真、时序分析
生成门级网表文件

1.4 SoC总线结构

- 在芯核互连的形式上，主要有共享总线、点对点连接、多总线等方式。
- 共享总线方式是通过不同地址的解码来完成不同主、从部件的互连，以及总线重用。

1.4 SoC总线结构

- 在芯核互连的形式上，主要有共享总线、点对点连接、多总线等方式。
- 共享总线方式是通过不同地址的解码来完成不同主、从部件的互连，以及总线重用。
- 多总线方式采用多种实现方式：按不同速率对总线分段；采用独立的读写总线；采用多个并行的总线；采用分层总线构架、采用交换矩阵或互连网络。

设置

8. SOC总线包括哪些

- A PCIe
- B AMBA
- C OCP
- D Wishbone
- E Avalon

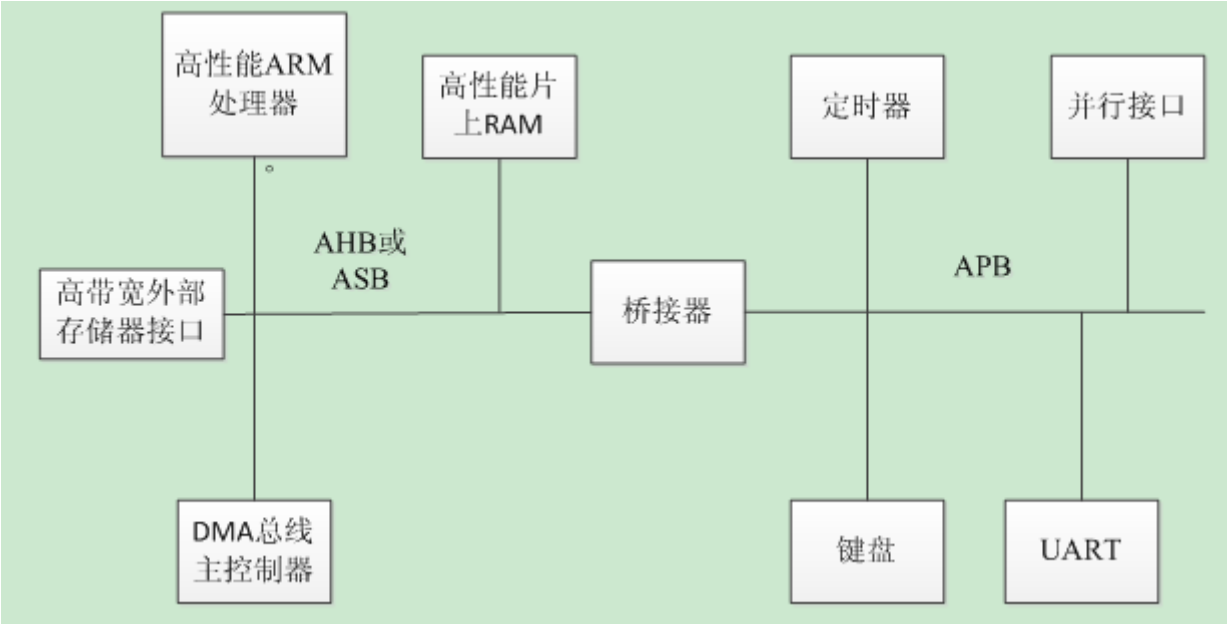
提交

1.4典型SoC片上总线

- AMBA
- Core Connect
- Avalon
- Wishbone
- OPC

1.4.1 AMBA总线

- ARM公司推出的片上总线，定义了三种可以组合使用的不同类型的总线：先进高性能总线(AHB)、先进系统总线(ASB)和先进外设总线(APB)。



(1)先进高性能总线（AHB）

- AHB适合于高性能和高时钟频率的系统模块，主要用于连接高性能和高吞吐量设备之间的连接，如CPU、片上存储器、DMA设备和DSP或其他协处理器等。
- 其主要特性有:单个时钟边沿操作，非三态的实现方式，支持多个主控制器，支持突发传输，支持分段传输，可配置32~ 128位总线宽度，支持字节、半字和字的传输。

(2)先进系统总线（ASB）

- AMBA的先进系统总线（ASB）适合于高性能的系统模块。具有以下特性:突发传送，流水方式工作，支持多总线主设备。
- 典型的ASB系统包括ASB主设备、ASB从设备、ASB译码器、ASB仲裁器。

(3)先进系统总线（ASB）

- ASB基本工作流程为：
  - 主设备请求使用总线。
  - 仲裁器决定授权哪个主设备占用总线。
  - 主设备一旦被授权，则启动传输。
  - 译码器用地址线的高位来选择从设备。
  - 从设备返回传输响应给主设备，数据在主设备和从设备之间传输。

(4)先进外设总线（APB）

- AMBA的先进外设总线适合于任何低带宽，并且无需高性能总线接口的外围器件，进行数据通信。

(5)AXI总线

- AXI协议是新一代AMBA3.0标准，总线带宽利用率高，功能丰富。
- 单向通道体系结构
- 支持多项数据交换
- 独立的地址和数据通道
- 增强的灵活性

AXI与AHB

总线名称	AMBA 3 AXI	AMBA 2 AHB
数据线宽度（位）	8,16,32,64,128,256,512,1024	32, 64, 128, 256
地址线宽度（位）	32	32
体系结构	多主/从设备 仲裁机制	多主/从设备 仲裁机制
数据线协议	支持流水/分裂传输 支持猝发传输 支持乱序访问 字节/半字/字	支持流水/分裂传输 支持猝发传输 字节/半字/字
数据对齐方式	大端/小端对齐 支持非对齐操作	大端/小端对齐 不支持非对齐操作
时序	同步	同步
互接	多路	多路
支持互接	不支持三态总线 分开的读/写数据线	不支持三态总线 分开的读/写数据线

AXI与AHB

总线名称	AMBA 3 AXI	AMBA 2 AHB
数据线宽度（位）	8,16,32,64,128,256,512,1024	32, 64, 128, 256
地址线宽度（位）	32	32
体系结构	多主/从设备 仲裁机制	多主/从设备 仲裁机制
数据线协议	支持流水/分裂传输 支持猝发传输 支持乱序访问 字节/半字/字	支持流水/分裂传输 支持猝发传输 字节/半字/字
数据对齐方式	大端/小端对齐 支持非对齐操作	大端/小端对齐 不支持非对齐操作
时序	同步	同步
互接	多路	多路
支持互接	不支持三态总线 分开的读/写数据线	不支持三态总线 分开的读/写数据线

AXI与AHB

总线名称	AMBA 3 AXI	AMBA 2 AHB
数据线宽度（位）	8,16,32,64,128,256,512,1024	32, 64, 128, 256
地址线宽度（位）	32	32
体系结构	多主/从设备 仲裁机制	多主/从设备 仲裁机制
数据线协议	支持流水/分裂传输 支持猝发传输 支持乱序访问 字节/半字/字	支持流水/分裂传输 支持猝发传输 字节/半字/字
数据对齐方式	大端/小端对齐 支持非对齐操作	大端/小端对齐 不支持非对齐操作
时序	同步	同步
互接	多路	多路
支持互接	不支持三态总线 分开的读/写数据线	不支持三态总线 分开的读/写数据线

设置

9. AXI总线是哪个公司提出的总线

A Xilinx

B Intel

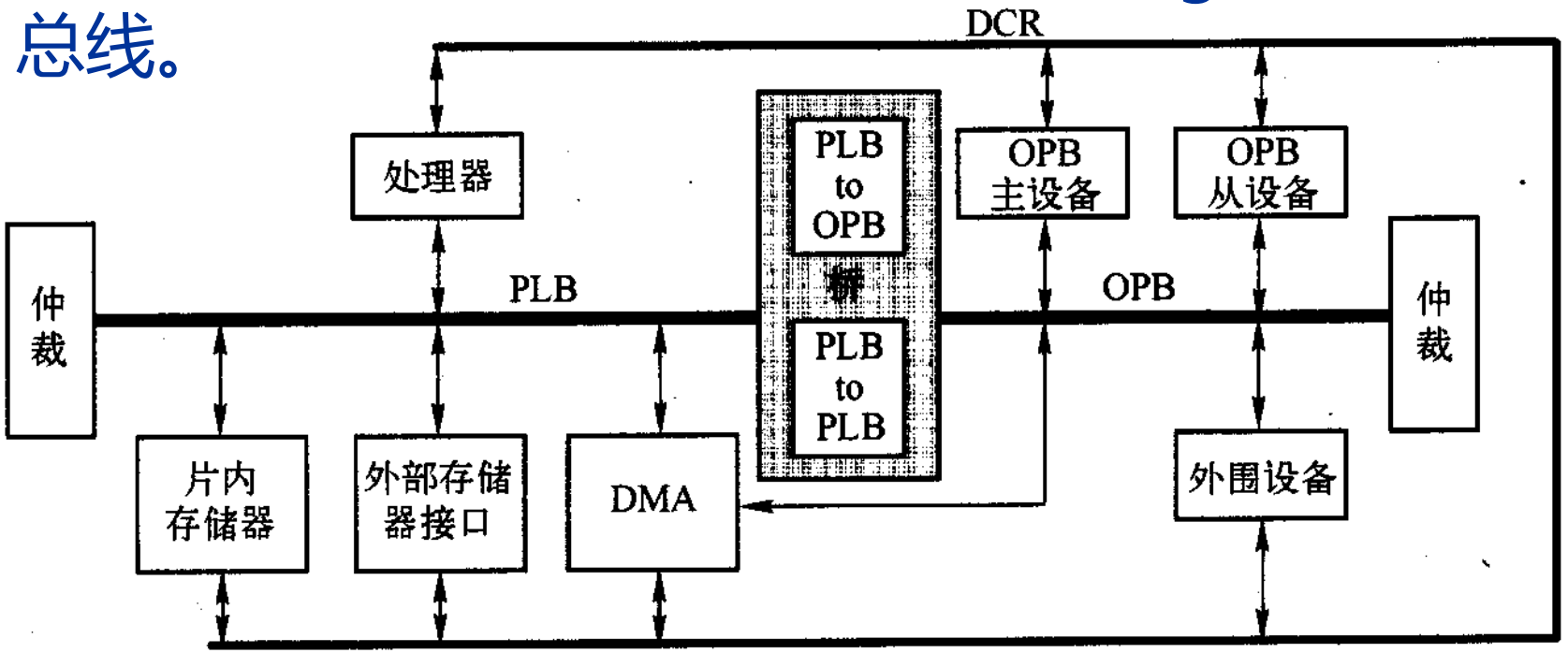
C ARM

D AMD

提交

1.4.2 CoreConnect总线

CoreConnect总线是IBM公司开发的片上总线系统，包括处理器本地总线PLB(Processor Local Bus)、片上外围总线OPB(On-Chip Peripheral Bus)、一个总线桥、两个判优器，以及一个设备控制寄存器(DCR)(Device Control Register Bus)总线。



1.4.2 CoreConnect总线

- PLB（Processor Local Bus）是高性能总线，通过总线接口单元来访问存储器设备，为总线传输的主要发出者和接受者之间提供高带宽、低延迟的连接。

1.4.2 CoreConnect总线

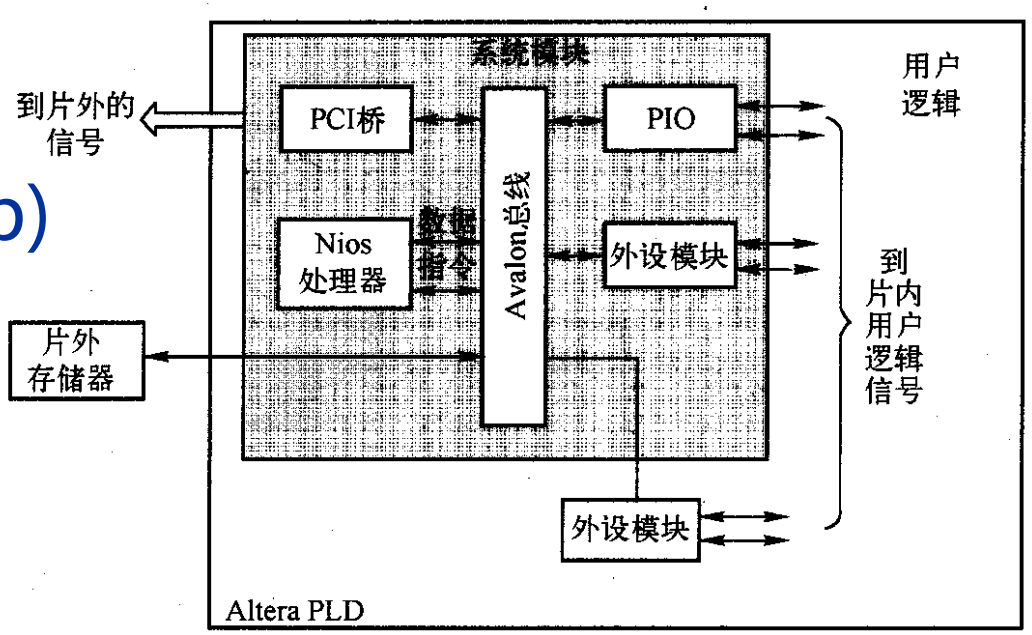
- OPB（On-Chip Peripheral Bus）用于连接低性能设备，如各种外围接口。为连接具有不同的总线宽度及时序要求的外设和存储器提供了一条途径，并尽量减小对PLB性能的影响。

1.4.2 CoreConnect总线

- DCR（Device Control Register）用于访问和配置PLB和OPB总线设备的状态和控制寄存器，用来规范CPU通用寄存器设备，控制寄存器之间传输数据。
- DCR总线在内存地址映射中取消了配置寄存器，减少取操作，增加了处理器内部总线带宽。

1.4.3 AVALON总线

Avalon总线是Altera公司设计的用于SOPC (System On Programmable Chip)中，连接片上处理器和其它IP模块的一种简单的总线协议，它规定了主部件和从部件之间进行连接的端口和通信的时序。



Avalon总线特点

- 开放性。接口协议简单，容易学习，易于理解。
- 简单性。提供一个易于理解的总线接口协议，使用独立的地址、数据、控制线，提供与片上逻辑的最简单的接口。
- 支持高达128位的数据宽度，支持2的非偶数次幂宽度的地址和数据通道。

Avalon总线特点

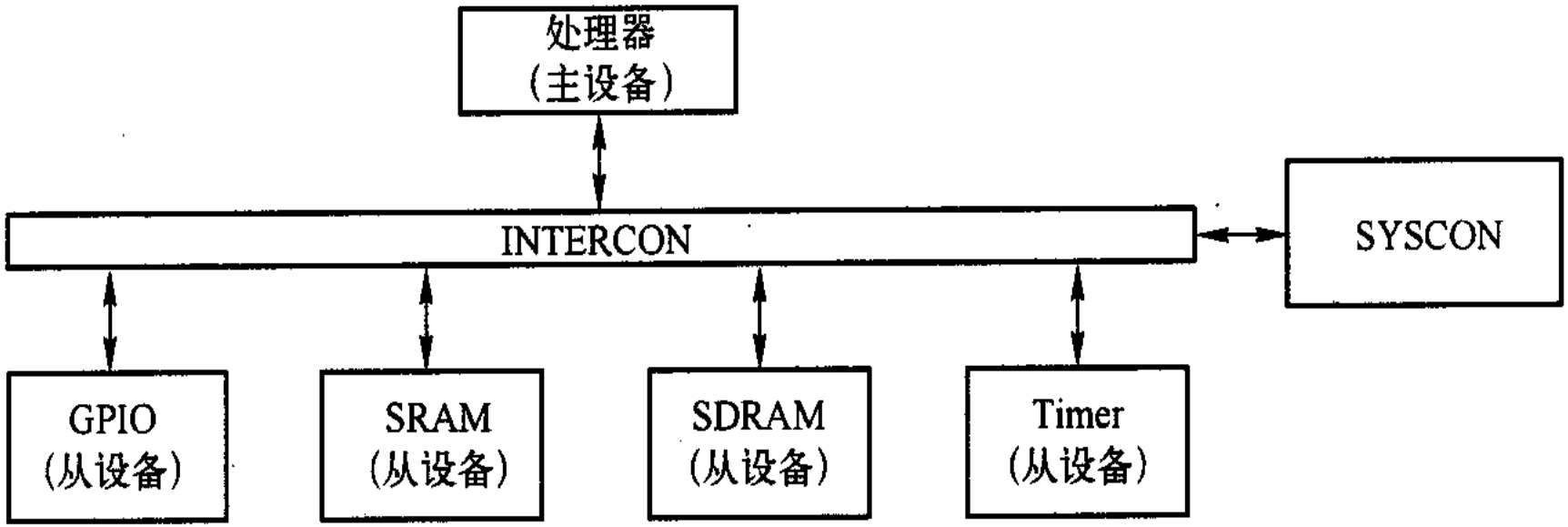
- 开放性。接口协议简单，容易学习，易于理解。
- 简单性。提供一个易于理解的总线接口协议，使用独立的地址、数据、控制线，提供与片上逻辑的最简单的接口。
- 支持高达128位的数据宽度，支持2的非偶数次幂宽度的地址和数据通道。
- 对同步操作的支持。所有的Avalon外设接口与Avalon交换结构的时钟同步，不需要复杂的握手/应答机制。
- 支持动态地址对齐。可处理具有不同数据宽度的外设之间的数据传输。资源占用少，减少片内逻辑资源的占用。

Avalon总线数据传输

- Avalon从端口传输
- Avalon主端口传输
- 流水线传输
- 流传输
- 三态传输
- 突发传输

1.4.4 Wishbone总线

- 最先由Silicore公司开发的片上总线系统，现在已移交OpenCores组织维护。
- Wishbone接口在IP核模型之间定义了一组标准的信号和总线周期。通过在IP核之间创建一个通用的接口，可以提高系统的可移植性和可靠性。



1.4.4 Wishbone总线

- INTERCON定义了主设备和从设备之间的连接方式，SYSCON用于产生系统时钟和复位信号,有4种连接方式，即点对点、数据流、共享总线和交叉连接。

1.4.4 Wishbone总线

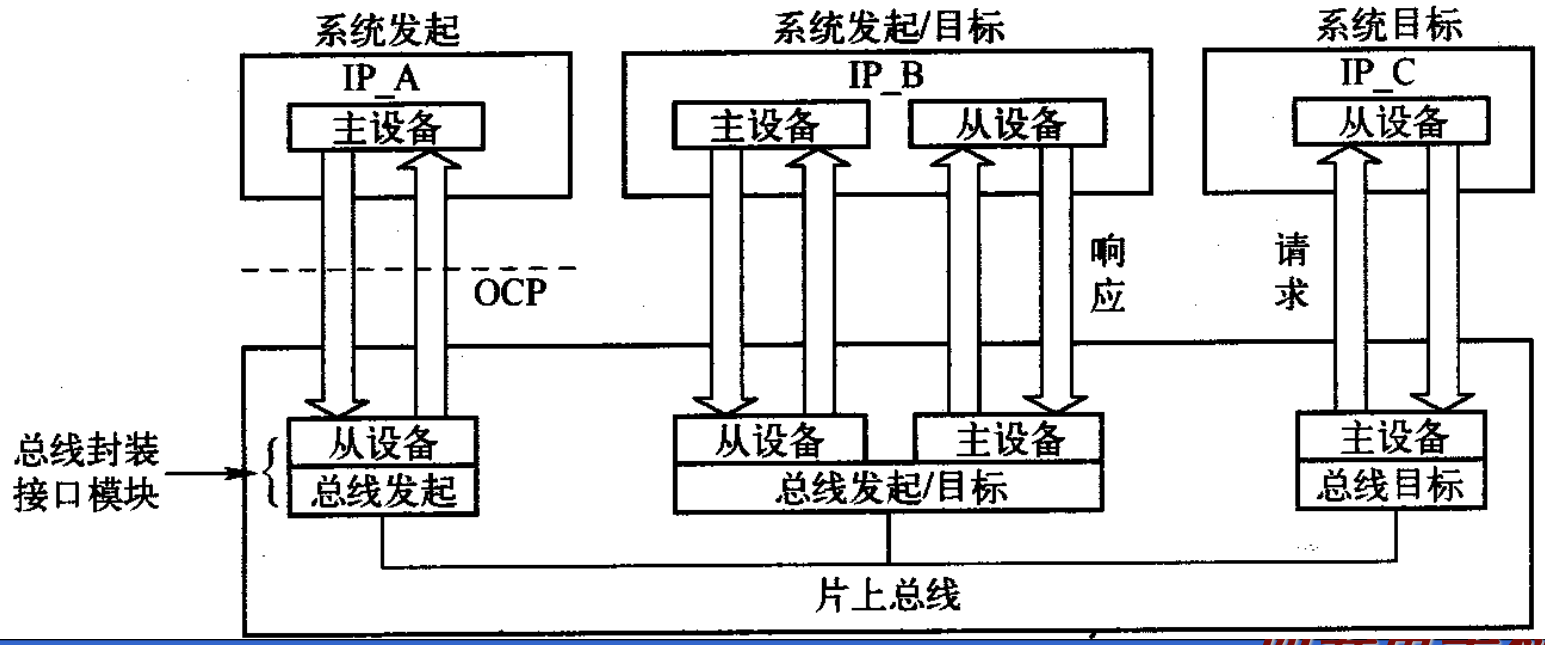
- INTERCON定义了主设备和从设备之间的连接方式，SYSCON用于产生系统时钟和复位信号,有4种连接方式，即点对点、数据流、共享总线和交叉连接。
- Wishbone总线规范可用于软核、固核和硬核，对开发工具和目标硬件没有特殊要求，并且几乎兼容已有所有的综合工具，可以使用多种硬件描述语言来实现。

1.4.4 Wishbone总线

- INTERCON定义了主设备和从设备之间的连接方式，SYSCON用于产生系统时钟和复位信号,有4种连接方式，即点对点、数据流、共享总线和交叉连接。
- Wishbone总线规范可用于软核、固核和硬核，对开发工具和目标硬件没有特殊要求，并且几乎兼容已有所有的综合工具，可以使用多种硬件描述语言来实现。
- Wishbone接口规范具有简单、开放、高效、利于实现等特点而且完全免费，并没有专利保护。

1.4.5 OCP总线

- 由OCP-IP (0pen Core ProtocolInternational Partnership)国际组织提出的片上总线，为了在SoC设计中实现IP核的即插即用而制订的片上总线规范，不依赖于特定处理器内核的总线协议。



1.4.5 OCP总线

- OCP总线规定了数据和控制信号，以及测试信号，使用同步单向信号来简化系统设计和时序分析。
- OCP总线也支持流水线操作，采用主从结构，并且通过线程标识符管理方式实现并发传送，增加了数据吞吐率。

1.4.5 OCP总线

- OCP总线规定了数据和控制信号，以及测试信号，使用同步单向信号来简化系统设计和时序分析。
- OCP总线也支持流水线操作，采用主从结构，并且通过线程标识符管理方式实现并发传送，增加了数据吞吐率。
- OCP标准是目前惟一一个无所有权，公开许可，并给出IP核系统级综合要求的以核为中心的协议，克服了反复定义、校验、证明和兼容接口的复杂性。仅定义IP核与IP核或者IP核与片上总线之间的封装接口。

五种SoC总线综合应用比较

总线名称	AMBA	Wishbone	CoreConnect	Avalon	OCP
适用器件	FPGA、PLD、ASIC	FPGA、PLD、ASIC	FPGA、PLD、ASIC	Altera系列 PLD	FPGA、PLD、ASIC
适用范围	高性能嵌入式系统	高性能或小型嵌入式系统	高性能嵌入式系统	用于 Altera 公司软核系统中	高性能或小型嵌入式系统
可用资源	……彼此都提供了丰富的 IP 核……				
使用条件	声称免费，但需要授权协议	完全免费	声称免费，但需要授权协议	声称免费，但需要授权协议	完全免费

五种SoC总线性能比较

总线名称	AMBA	Wishbone	CoreConnect	Avalon	OCP
互联方式	共享总线	共享总线 交叉总线 点对点总线	共享总线	共享总线	点对点总线
主控制器	多个	多个	多个	多个	单个
数据线宽度/位	32－512	32－512	32－512	32－512	用户可配置
地址空间/位	64	64	64	64	32
请求响应	同步	同步	同步	同步	异步
数据传输方式	……都可以按字节、半字、字几种方式传输……				
事务传输方式	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 突发
数据对齐方式	……都有大端对齐和小端对齐两种方式……				
仲裁机制	系统定义	用户自定义	系统定义	系统定义	无仲裁

五种SoC总线性能比较

总线名称	AMBA	Wishbone	CoreConnect	Avalon	OCP
互联方式	共享总线	共享总线 交叉总线 点对点总线	共享总线	共享总线	点对点总线
主控制器	多个	多个	多个	多个	单个
数据线宽度/位	32－512	32－512	32－512	32－512	用户可配置
地址空间/位	64	64	64	64	32
请求响应	同步	同步	同步	同步	异步
数据传输方式	……都可以按字节、半字、字几种方式传输……				
事务传输方式	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 突发
数据对齐方式	……都有大端对齐和小端对齐两种方式……				
仲裁机制	系统定义	用户自定义	系统定义	系统定义	无仲裁

五种SoC总线性能比较

总线名称	AMBA	Wishbone	CoreConnect	Avalon	OCP
互联方式	共享总线	共享总线 交叉总线 点对点总线	共享总线	共享总线	点对点总线
主控制器	多个	多个	多个	多个	单个
数据线宽度/位	32－512	32－512	32－512	32－512	用户可配置
地址空间/位	64	64	64	64	32
请求响应	同步	同步	同步	同步	异步
数据传输方式	……都可以按字节、半字、字几种方式传输……				
事务传输方式	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 突发
数据对齐方式	……都有大端对齐和小端对齐两种方式……				
仲裁机制	系统定义	用户自定义	系统定义	系统定义	无仲裁

五种SoC总线性能比较

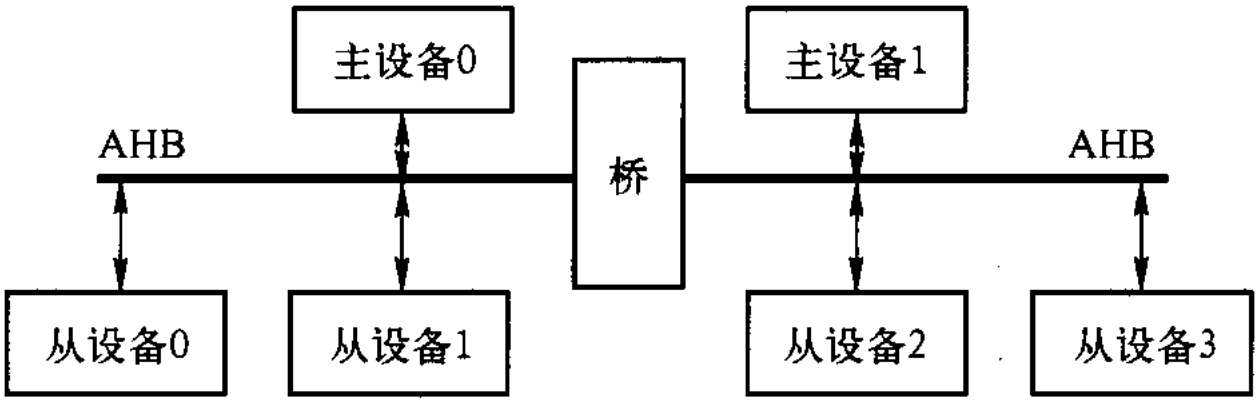
总线名称	AMBA	Wishbone	CoreConnect	Avalon	OCP
互联方式	共享总线	共享总线 交叉总线 点对点总线	共享总线	共享总线	点对点总线
主控制器	多个	多个	多个	多个	单个
数据线宽度/位	32－512	32－512	32－512	32－512	用户可配置
地址空间/位	64	64	64	64	32
请求响应	同步	同步	同步	同步	异步
数据传输方式	……都可以按字节、半字、字几种方式传输……				
事务传输方式	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 突发
数据对齐方式	……都有大端对齐和小端对齐两种方式……				
仲裁机制	系统定义	用户自定义	系统定义	系统定义	无仲裁

五种SoC总线性能比较

总线名称	AMBA	Wishbone	CoreConnect	Avalon	OCP
互联方式	共享总线	共享总线 交叉总线 点对点总线	共享总线	共享总线	点对点总线
主控制器	多个	多个	多个	多个	单个
数据线宽度/位	32－512	32－512	32－512	32－512	用户可配置
地址空间/位	64	64	64	64	32
请求响应	同步	同步	同步	同步	异步
数据传输方式	……都可以按字节、半字、字几种方式传输……				
事务传输方式	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 分离/突发	流水/ 突发
数据对齐方式	……都有大端对齐和小端对齐两种方式……				
仲裁机制	系统定义	用户自定义	系统定义	系统定义	无仲裁

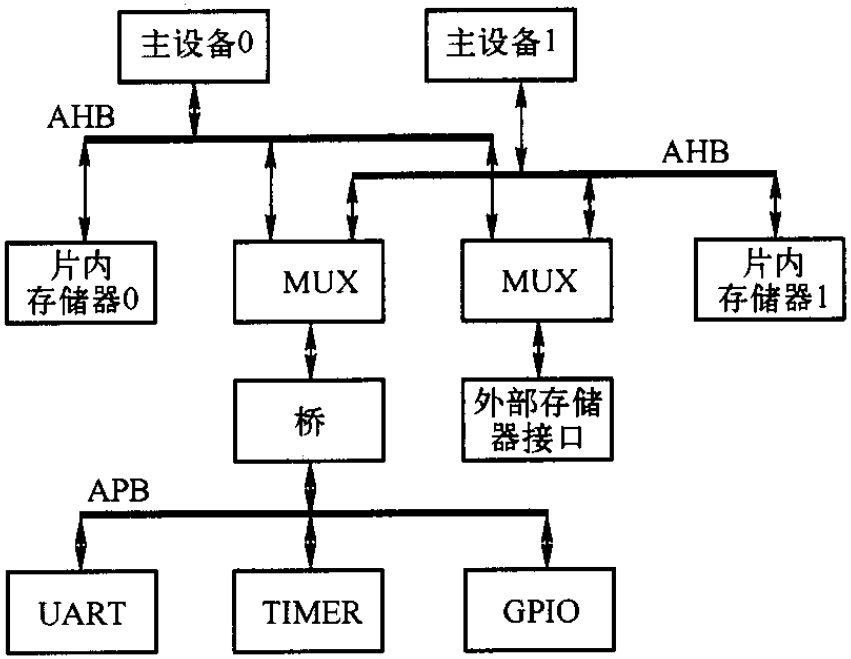
复杂总线结构

- 双总线结构：主设备可以快速访问本地总线的从设备，也可以通过总线桥访问相邻总线的从设备。



复杂总线结构

- 复杂总线结构：两条高速总线AHB，一条低速总线APB，哪个主设备要访问低速设备或者外部存储器需要通过仲裁来决定。



设置

10. 真正可以免费试用的SoC总线协议是

A AMBA

B OCP

C CoreConnect

D Wishbone

提交