



Xinlinx大学计划课程

Zynq-7000 SoC设计导论

主讲：何宾

Email: hebin@mail.buct.edu.cn



Zynq-7000 SoC设计导论

主要内容

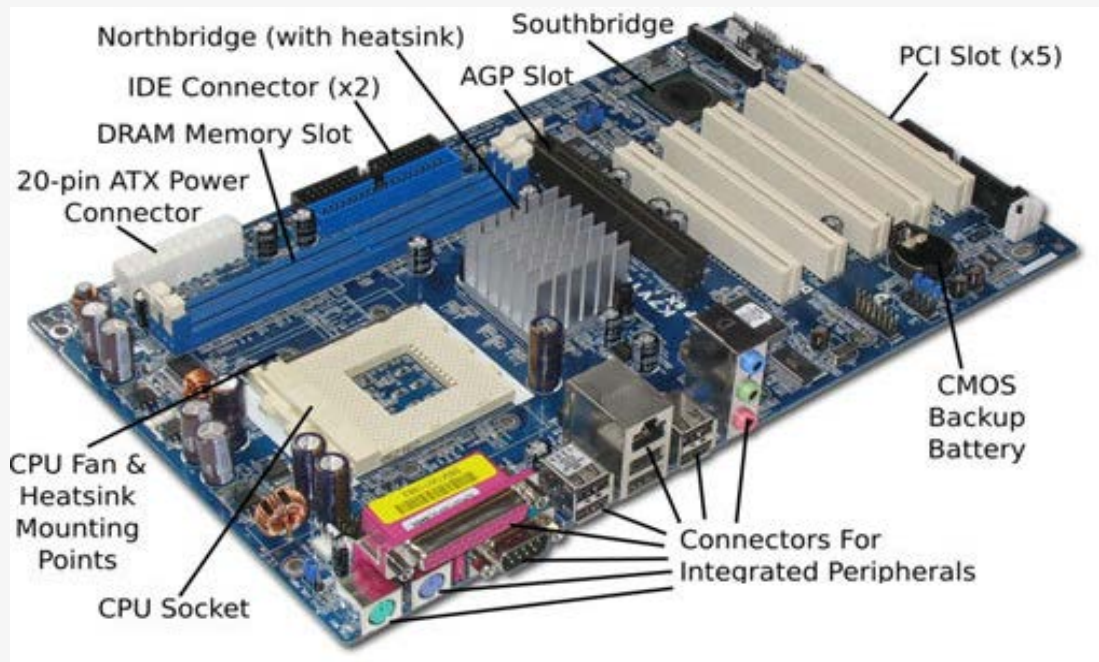
- 全可编程片上系统基础知识
- Zynq-7000 SoC功能和结构
- Zynq-7000 SoC在嵌入式系统中的优势
- Zynq-7000 SoC的Vivado设计流程

全可编程片上系统基础知识

--全可编程片上系统的演进

在二三十年前，构建一个嵌入式系统需要使用大量的器件。

- 典型的，对于一台个人电脑PC来说，其主板由大量的电子元器件、散热装置以及固定连接器组成。



全可编程片上系统基础知识

--全可编程片上系统的演进

这种传统的设计结构会带来以下几个方面的问题

- 增加了系统的整体功耗。
- 增加了系统的总成本。
- 降低了系统的可靠性和安全性。
- 系统维护成本较高。

全可编程片上系统基础知识

--全可编程片上系统的演进

随着半导体技术的不断发展，可以将构成计算机基本结构的大量元件集成到单个芯片中，比如：

- CPU内核
- 总线结构
- 功能丰富的外设控制器
- 模数混合器件ARM
 - ◆ CPU体系结构为基础的嵌入式处理器物理知识产权（Intellectual Property, IP）核

全可编程片上系统基础知识

--全可编程片上系统的演进

SoC的结构的优势体现在以下几个方面

- 改善性能
- 降低功耗
- 减少体积
- 降低总成本
- 可靠性提高

全可编程片上系统基础知识

--全可编程片上系统的演进

SoC的局限性，主要体现在

- 灵活性差
- 专用性强
- 设计复杂

SoC与MCU和CPU的比较

--SoC的特点

SoC的特点主要体现在以下几个方面

- 在SoC器件内，可以集成多个功能强大的处理器内核。
- 在SoC器件内，可以集成容量更大的存储器块、不同的I/O资源，以及其它外设。
- 随着半导体工艺的不断发展，在SoC器件内也集成了功能更强大的图形处理器单元（ Graphics Processing Unit , GPU ）、数字信号处理器（ Digital Signal Processor , DSP ）以及视频和音频解码器等。

SoC与MCU和CPU的比较

--SoC的特点

SoC的特点主要体现在以下几个方面

- 在基于SoC所构成的系统上，可以运行不同的操作系统。典型地，微软公司的Windows操作系统、Linux操作系统和谷歌公司的Android操作系统。
- 由于SoC强大的功能，因此它可以用于更高级的应用，比如：数字设备的主芯片（智能手机、平板电脑）。

SoC与MCU和CPU的比较

--CPU的特点

CPU的特点主要体现在以下几个方面

- 单个处理器核。当然对于英特尔公司量产的包含多个CPU核的芯片来说，已经不是传统意义上的CPU了，它已经体现出了SoC的影子。
- CPU可以用在绝大数的应用场合，但是需要外部额外的存储器和外设的支持。

SoC与MCU和CPU的比较

--MCU的特点

MCU的特点主要体现在以下几个方面

- 典型地，只有一个处理器内核。
- 内部包含了存储器块、基于的I/O和其他外设。
- MCU主要用于工业控制领域，比如：嵌入式应用。



可编程SoC系统技术特点

- 定制
- 延长产品生命周期
- 降低元件成本
- 硬件加速



全可编程片上系统中的处理器类型

根据不同的需求，全可编程片上系统内的处理器可分为

- 软核
- 硬核处理器

Zynq-7000 SoC功能和结构

--Xilinx Zynq-7000 SoC功能

与传统的FPGA和SoC相比，Zynq-7000 全可编程SoC：

- 提供了FPGA的灵活性和可扩展性。
- 提供了与专用集成电路（**Application-Specific Integrated Circuit, ASIC**）和专用标准产品（**Appciation-Specific Standard Product, ASSP**）相关的性能、功耗和易用性。

Zynq-7000 SoC功能和结构

--Xilinx Zynq-7000 SoC功能

Zynq-7000 SoC使得设计者能使用工业标准的工具在单个平台上实现高性能和低成本的应用

- 可扩展处理平台中的每个器件包含相同的PS。
- 不同器件所包含的PL和I/O资源不同。

Zynq-7000 SoC功能和结构

--Zynq-7000 SoC处理系统PS的构成

Zynq-7000 SoC处理系统PS的构成

- 应用处理器单元APU
- 存储器接口
 - ◆ DDR控制器
 - ◆ 四-SPI控制器
- 静态存储器控制器（Static Memory Controller, SMC），提供了NAND存储器和并行SRAM/NOR存储器的读写控制功能

Zynq-7000 SoC功能和结构

--Zynq-7000 SoC处理系统PS的构成

I/O外设

- 通用输入输出端口GPIO
- 两个三模式以太网控制器
- 两个USB控制器
- 两个SD/SDIO控制器
- 两个SPI控制器
- 两个CAN控制器两个UART控制器
- 两个I2C控制器
- 总共54个PS一侧的MIO引脚

Zynq-7000 SoC功能和结构

-- Zynq-7000 SoC可编程逻辑PL的构成

PL提供了用户可配置能力的丰富结构。主要体现在

- 可配置逻辑块 (Configurable Logic Block , CLB) 资源。
- 36Kb BRAM资源
- 数字信号处理DSP48E1资源
- 时钟管理单元 ,
- 可配置的I/O资源
- 模拟-数字转换器 (XADC)
- 低功耗的吉比特收发器 (仅Z-7030和Z-7045器件存在)
- 用于PCI-E设计的集成接口模块 (仅Z-7030和Z-7045器件可用)

Zynq-7000 SoC功能和结构

-- Zynq-7000 SoC内互联结构

PS内模块的互联。

- 在PS内，用于连接各个功能模块。
- 主要包含：
 - ◆ OCM互联
 - ◆ 中央互联

Zynq-7000 SoC功能和结构

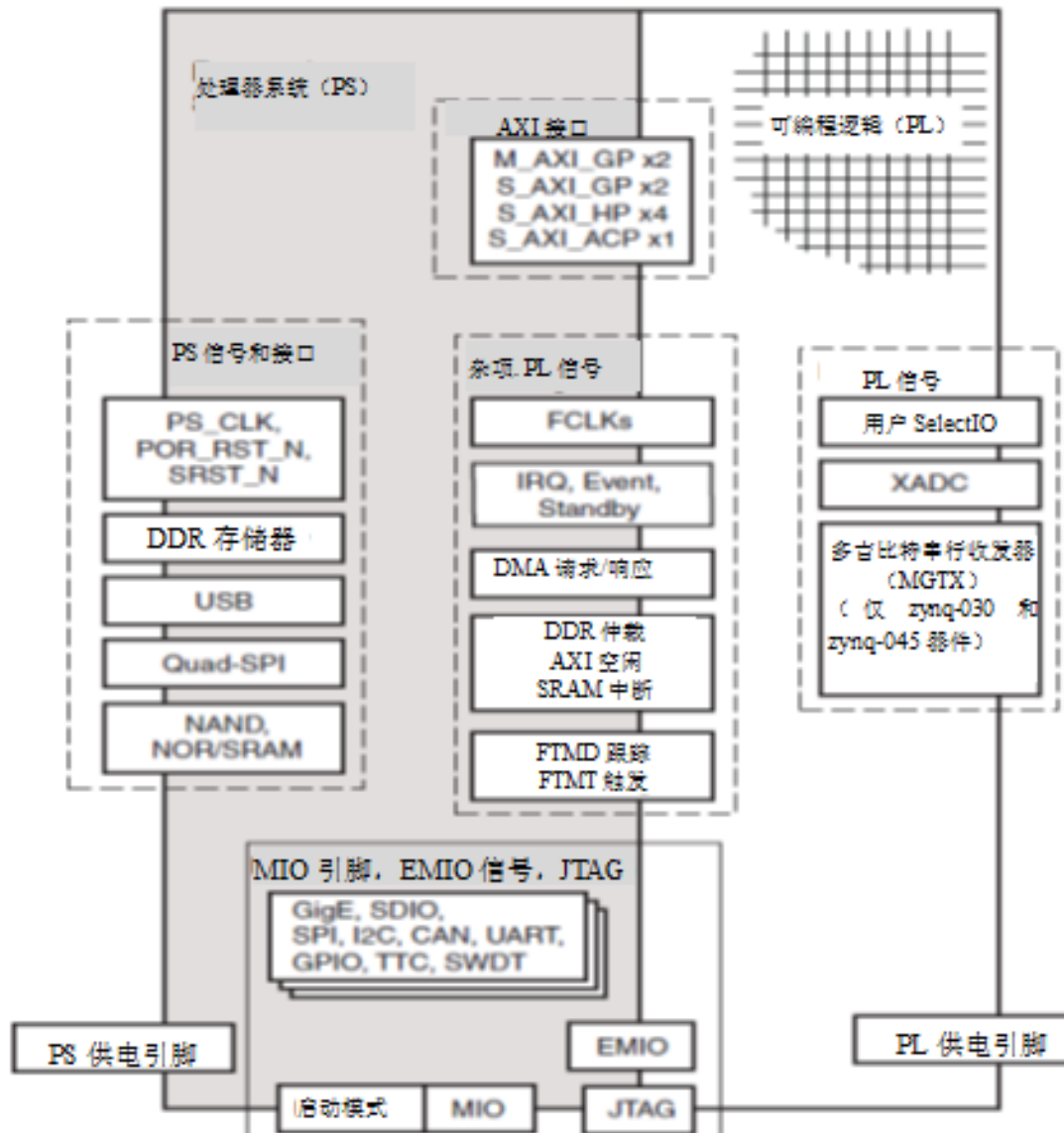
-- Zynq-7000 SoC内互联结构

PS-PL接口

- 提供了可用于PL内定制外设的所有可用信号。
- 通过PS-PL接口可以实现PL内定制的外设（也称为IP核）与PS内的Cortex-A9双核处理器及相关资源的连接。
- 在PS和PL之间提供了两种类型的接口：
 - ◆ 功能接口
 - ◆ 配置信号

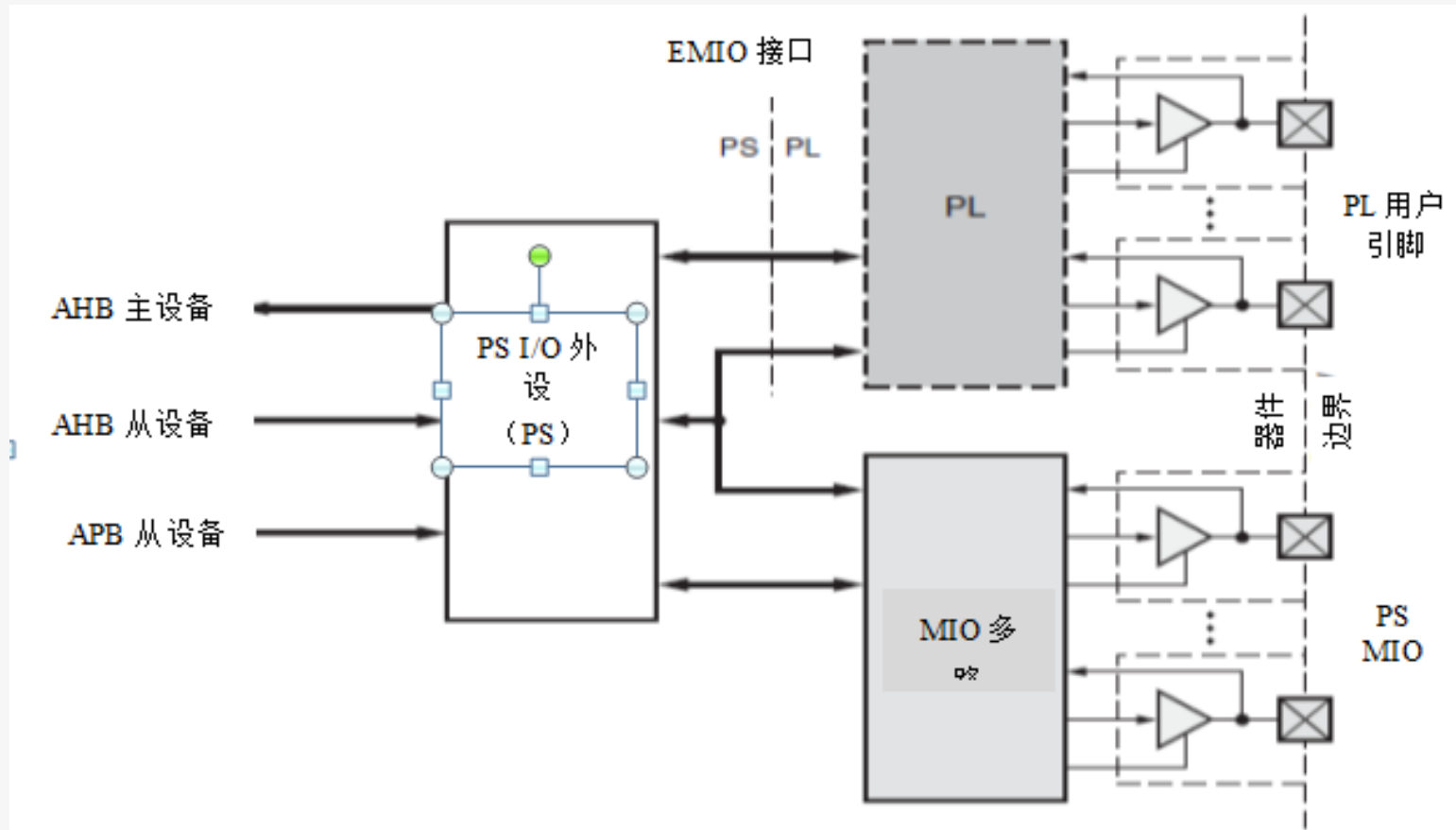
Zynq-7000 SoC功能和结构

-- Zynq-7000 SoC的供电引脚



Zynq-7000 SoC功能和结构

Zynq-7000 SoC内MIO到EMIO的连接



Zynq-7000 SoC功能和结构

-- MIO引脚分配考虑因素

当使用EMIO作为布线的另一个选择时，PS内外设的
最高时钟频率将降低。

- 两个MIO电压组
- 启动模式引脚
- I/O缓冲区输出使能控制
- 选择从SDIO设备启动Zynq-7000 SoC
- 静态存储器控制器（SMC）接口
- 四-SPI接口
- MIO[8:7]引脚

Zynq-7000 SoC功能和结构

-- MIO信号布线连接

由MIO_PIN_[53:0]配置寄存器控制MIO内的信号连接

在Zynq-7000 SoC中使用4级复用，用于控制MIO到MIO的各个输入/输出信号。MIO信号布线连接。

- 高速数据信号（比如：用于吉比特以太网的RGMII和用于USBULP1）只通过1级多路复用
- 低速信号线（比如：UART和I2C端口）可以通过所有4级多路复用。

Zynq-7000 SoC功能和结构

--Zynq-7000 内为PL分配的信号

可编程逻辑接口组在PS和PL之间包含其它接口

- 一个输入是由PL驱动
- 一个输出是由PS驱动

信号可能带有后缀，后缀“N”表示低有效信号，后缀“TN”表示低三态使能信号为连接到PL的输出。

连接到PL的输出信号，总是处于逻辑高或者逻辑低状态

Zynq-7000 SoC功能和结构

--Zynq-7000 内为PL分配的信号

Zynq-7000 内为PL分配的信号

- 时钟
- 复位
- 中断信号
- 事件信号
- 空闲 AXI , DDR Urgent/Arb , SRAM中断信号
- DMA Req/Ack信号

Zynq-7000 SoC在嵌入式系统中的优势

--使用PL实现软件算法

使用PL加速的优势

- 性能
- 功耗
- 延迟

Zynq-7000 SoC在嵌入式系统中的优势

--使用PL实现软件算法

实现PL加速器的方法

- 设计者可以使用Verilog HDL或者VHDL在Zynq-7000 SoC内的PL中创建定制IP，用于算法的加速实现。有经验的设计者可以直接使用C代码设计一个模型，通过使用Xilinx提供的Vivado HLS综合工具在Zynq-7000 SoC内的PL中创建一个高效算法的硬件实现IP核。

Zynq-7000 SoC在嵌入式系统中的优势

--使用PL实现软件算法

影响PL加速的因素

- I/O速率
- 资源限制
- 延迟限制

Zynq-7000 SoC在嵌入式系统中的优势

--使用PL实现软件算法

降低功耗

- 使用的是较短的、低容性的本地连接路径。而在一个本地汇编命令行的方式中，需要将数据从一个操作单元传输到另一个传输单元。而在Cortex-A9处理器上实现算法功能要求从本地缓存或者外部存储器中取出指令和数据，运行程序得到结果后，将结果写回到寄存器，或者存储器系统，这需要通过一个较长的、较高容性的接口。

Zynq-7000 SoC在嵌入式系统中的优势

--使用PL实现软件算法

实时减负

- MicroBlaze辅助实时处理
- PL中断服务
- HW状态机

Zynq-7000 SoC在嵌入式系统中的优势

--使用PL实现软件算法

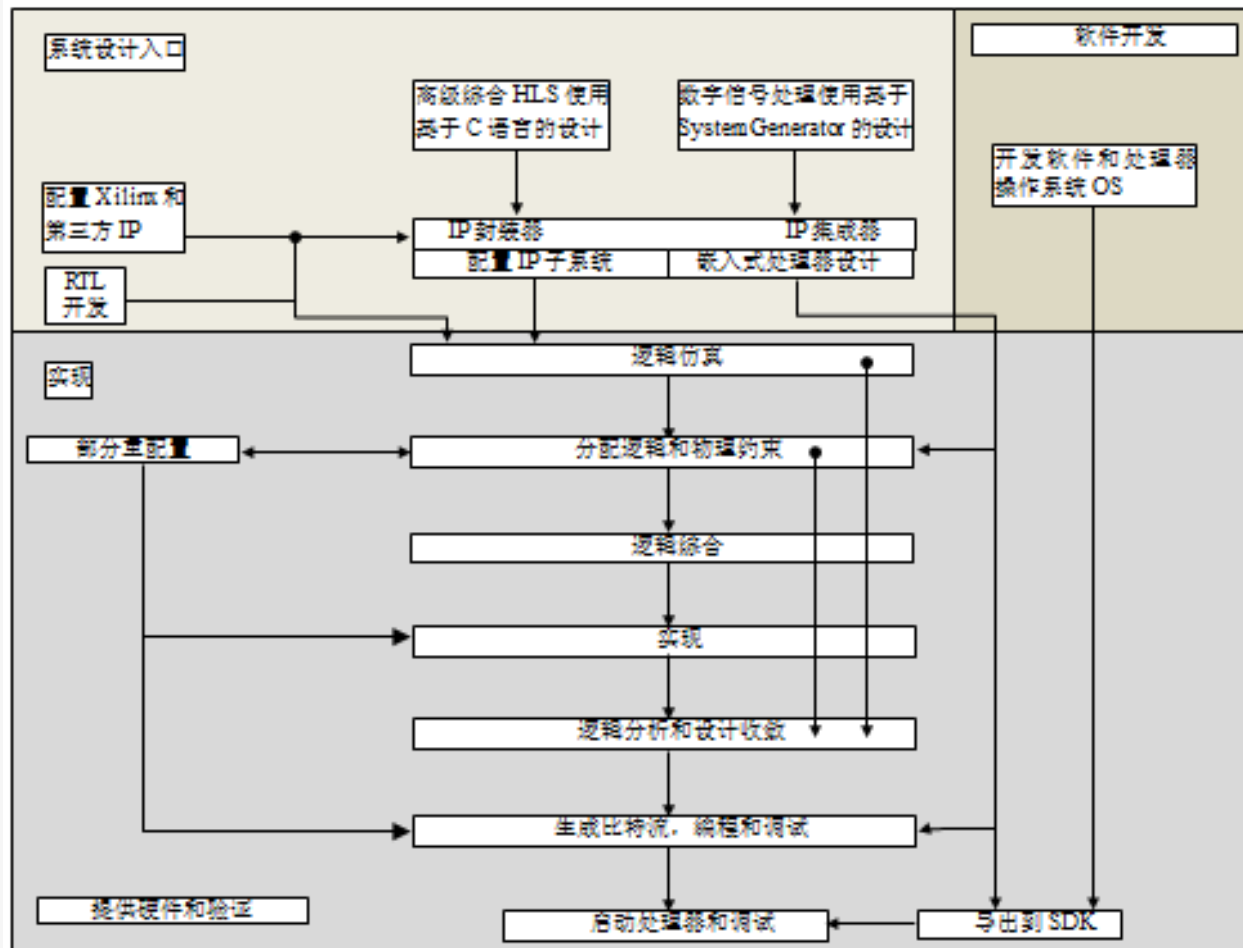
可重配置计算

- 当需要提供新的硬件加速器时，可以重新配置Zynq-7000 SoC内的可编程逻辑。当重新配置整个器件或者重新配置PL所选择的一部分逻辑时，这就允许将加速器功能库保存在磁盘、Flash存储器或者DRAM中，然后当需要时再将这些加速器功能库加载到PL中。Zynq-7000 SoC的PS可以通过它的PCAP接口实现重配置过程，并且负责PL资源的管理和分配。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 的Vivado设计流程

Vivado的IP设计和系统级设计集成



Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

使用RTL或者网表的设计流程

- Vivado设计套件中提供了不同的设计入口点用于支持不同的设计流程。
 - ◆ RTL流程
 - ◆ 第三方综合流程
- Vivado设计套件可以通过创建多个运行来评价不同的综合和实现选项，或者时序和物理约束，或者实现配置，它们对最终设计结果的影响。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

IP子系统设计

- Vivado IP集成器环境中提供用户通过AMBA AXI4互联协议将不同IP组合在一起构成IP子系统的能力。
- 通过使用块设计类型接口，用户可以交互的配置和连接IP，并且很容易通过使用符合正确DRC的连线，将类似原理图符号的IP连接在一起。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

IP子系统设计

- 与传统基于RTL的连接相比，这种方法节省了大量的时间。在Vivado设计套件中，提供了连接自动化以及DRC的能力，用于保证实现正确的配置和连接。
- 将这些IP块设计来进行验证、封装，最后将其作为单个的设计源。在Vivado设计套件中，允许在一个设计工程使用块设计。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

IP子系统设计

- Vivado IP集成器环境中提供用户通过AMBA AXI4互联协议将不同IP组合在一起构成IP子系统的能力。
- 通过使用块设计类型接口，用户可以交互的配置和连接IP，并且很容易通过使用符合正确DRC的连线，将类似原理图符号的IP连接在一起。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

IP子系统设计

- 在Vivado设计套件中，提供了连接自动化以及DRC的能力，用于保证实现正确的配置和连接。
- 将这些IP块设计来进行验证、封装，最后将其作为单个的设计源。在Vivado设计套件中，允许在一个设计工程使用块设计。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

嵌入式处理器硬件设计

- 创建一个嵌入式处理器硬件设计涉及到Vivado设计套件提供的IP集成器特性。
- 在一个Vivado IP集成器块设计中，用户可以例化、配置和组合处理器核，以及它的接口。
- 在用户连接不同的IP时，Vivado设计套件将强制这些IP之间的连接符合相关的规范，并且在连接的过程中提供连接帮助。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

嵌入式处理器硬件设计

- 当在Vivado设计套件中对设计处理完成后，就可以将Vivado中的硬件设计导入到软件开发工具SDK中。
- 在SDK中，用户可以完成软件的开发和对软件代码的验证。在Vivado设计套件中提供了仿真和调试特性，这允许用户同时使用SDK和Vivado内的工具对设计进行仿真和验证。

Zynq-7000 SoC在嵌入式系统中的优势

--Zynq-7000 SoC的Vivado设计流程

使用模型和高级综合的DSP设计

- 基于模型的DSP设计
- 基于HLS的DSP设计
- 脱离上下文的设计流程
- I/O引脚规划和布局
- 设计分析和验证
- 器件编程和硬件验证
- 部分可重配置