

<p>课程背景</p>
<p>FPGA 高级进阶班主要是介绍 FPGA 系统开发中的高级技巧，深入探讨如何提高 FPGA 设计的性能，如何优化设计规模，进而设计出高性能低成本的产品。同时介绍了 FPGA 和 DSP 芯片构成的高性能协同计算系统的软/硬件设计技术。课程中会结合实际的设计代码讲解并行设计技术，流水线设计技术等实用技巧，帮助学员短时间内理解和掌握这些高级技巧，并可以尽快应用到工程项目中去。</p>
<p>课程目标</p>
<p>本课程主要针对具备一定基础的学员，帮助学员快速提高技能，使之能够运用高级技巧快速自主地设计复杂 FPGA 系统或者 FPGA 和 DSP 协同工作的系统。</p>
<p>培养对象</p>
<p>具备一年左右的 FPGA 系统或者硬件系统开发设计经验的工程师，或者具有一定基础的电子类专业的大学生和研究生。</p>
<p>课程进度安排</p>
<p>第一阶段</p>
<ol style="list-style-type: none"> 1. 三个设计基本原则，包括面积和速度的平衡互换原则，硬件可实现原则和同步设计原则 2. 掌握 PS2 协议，掌握时钟线和数据线是如何控制数据的读写的 3. FPGA 设计仿真验证的原理和方法 4. ModelSim SE 仿真验证技巧 5. 仿真测试文件（Testbench）的设计方法 6. 异步 SRAM 的操作时序
<ol style="list-style-type: none"> 1. 实战训练一： 训练课题：“ModelSim 软件的使用”

实验要点：

- 1.1 ModelSim 软件工程创建
- 1.2 ModelSim 软件进行仿真
- 1.3 测试文件 (Testbench) 的设计方法

2. 实战训练二：

训练课题：“异步 SRAM 存储器读写操作”

实验要点：

- 2.1 SRAM 存储器的操作时序设计
- 2.2 复杂硬件开发的思想方法

第二阶段

1. AD/DA 转换器接口设计

- 1.1 选择正确的时钟采样边沿
- 1.2 选择适合的同步头检测方法
- 1.3 选择有效的缓存策略

2. 利用 FPGA 实现 RS-232C 串行接口

- 2.1 RS-232C 接口通讯原理和相关电气标准
- 2.2 RS-232C 的通讯协议的要点
- 2.3 TxD,RxD 是怎样控制数据传输的
- 2.4 RS-232C 接口的原理图剖析
- 2.5 软件和硬件握手的相关概念

3. MCU 与 FPGA 片上系统开发

- 3.1 FPGA 扩展 MCU 开发技术

3.1.1 FPGA 扩展方案及其系统设计技术

3.1.2 基于单片机 IP 软核的 SOC 设计方案

3.2 FPGA 扩展方案设计实例

3.2.1 串进并出/并进串出双向端口扩展模块设计

3.2.2 8 位四通道数据交换扩展模块设计

3.2.3 存储器读写的 FPGA 扩展模块设计

3.2.4 四通道 PWM 信号发生器接口模块设计

3.2.5 李萨如图波形发生器扩展模块设计

3.3 基于单片机核的 FPGA 片上系统设计

3.3.1 单片机扩展串进并出/并进串出模块的 SOC 设计

3.3.2 扩展 SRAM 模块的片上系统设计

3.3.3 扩展移相信号发生器模块的片上系统设计

4. FPGA 和 DSP 处理器联合应用的领域，视频、图像、人脸识别，指纹识别 的等领域
的现状与发展前景。

5. HPI 接口的分析，以及 FPGA 如何利用 HPI 接口对 DSP 进行控制。

6. 详细剖析 DSP 进行图像处理的关键技术（图像编码解码技术，图像处理的相关算法）

7. 案例分析：高清视频处理系统之图像阈值变换算法，及其达到的效果

1. 实战训练三：

训练课题：“RS232C 接口和 PC 主机的双向传输实验”

实验要点：

1.1 掌握 RxD,TxD 怎样控制数据传输的

1.2 掌握 RS232C 通讯原理、设计实现和硬件握手和软件握手的概念

1.3 通过 RS232 接口，FPGA 向 PC 主机发送字符串，PC 主机向 FPGA 发送数据，并使数据显示在数码管上。

2. 实战训练四：

训练课题：“AD 数据采集系统设计”

实验要点：

2.1 AD 转换器与 FPGA 接口设计

2.2 片上双口 RAM 的乒乓缓存

2.3 FPGA 与 PC 的通信

3. 实战训练五：

训练课题：“视频图像采集实验”

实验要点：

3.1 掌握编码解码芯片的工作原理

3.2 掌握视频图像处理解决方案，所要用到的接口、器件和原理

3.3 DSP 进行视频图像处理的流程。

4. 实战训练六：

训练课题：“视频图像处理的相关算法之一----图像反色实验”

实验要点：

4.1 了解图像反色的原理和应用领域

4.2 了解图像反色的相关算法

5. 实战训练七：

训练课题：“视频图像处理的相关算法之二----图像阈值变换实验”

实验要点：

<p>5.1 了解图像阈值变换的原理和应用领域</p> <p>5.2 了解图像阈值变换的相关算法</p>
<p>第三阶段</p>
<p>1.PS2 键盘鼠标协议详解</p> <p>2.USB 开发</p>
<p>3. 实战训练八：</p> <p> 训练课题：“PS2 接口键盘、鼠标实验”</p> <p> 实验要点：</p> <p> 2.1 ModelSim SE 软件操作技巧演练</p> <p> 2.2 PS2 接口操作时序设计</p> <p> 2.3 基于模型的系统仿真方法</p> <p>4. 实战训练九：</p> <p> 训练课题：“USB 读取数据实验”</p> <p> 实验要点：</p> <p> 2.1 USB 读写方法</p> <p> 2.2 USB 硬件系统设计</p>
<p>课程背景</p>
<p>FPGA 技能提升班培训课程主要帮助学员尽快掌握 FPGA 的开发流程和设计方法，以工程实践为例，循序渐进的学习 FPGA 的集成开发环境，开发流程以及硬件电路设计等知识。每次课程都配有相关实战训练，每个实战训练题目都可以在 FPGA 硬件平台上进行下载验证。通过实战，学员可以更好的理解消化课堂知识，工程实践水平会得到迅速提高。</p>
<p>课程目标</p>

<p>培养学员迅速掌握和使用 FPGA 数字系统开发工具、开发流程，能够独立进行初步的 FPGA 系统设计。经过培训，学员可以掌握 HDL 语言的初步开发能力，并且解决 FPGA 产品开发过程中的常见问题，掌握基于 FPGA 的设计和调试方法。</p>
<p>培训对象</p>
<p>FPGA 系统的软件和硬件开发工程师；电子类专业的大学生和研究生；电子产品设计爱好者。</p>
<p>课程进度安排</p>
<p>第一阶段</p>
<p>了解 FPGA 系统设计的基础知识，掌握 FPGA 最小系统硬件电路设计方法，学会操作 QuartusII 软件来完成 FPGA 的设计和开发。</p>
<p>1 FPGA 概念</p> <p>1.1 FPGA 简单入门</p> <p>1.2 FPGA 应用领域</p> <p>1.3 FPGA 的优势</p> <p>1.4 开发流程</p> <p>2.FPGA 芯片的的结构</p> <p>2.1 FPGA 的结构、内部逻辑单元及接口</p> <p>2.2 主流低成本 FPGA 的结构、内部逻辑单元及接口</p> <p>2.3 FPGA 的布线策略</p> <p>3 开发工具简介</p> <p>3.1 软件下载和申请 license 申请</p> <p>3.2 Quartus II 的安装</p>

3.3 ModelSimAltera 的安装

3.4 USBBlaster 的驱动安装

4 FPGA .V 文件的编程规范

4.1 单个.v 文件的书写规范

4.2 多个.v 文件的书写规范

4.3 声明部分的编写规范

4.4 主体部分编写 , always、initial、function、task 的选择和使用

4.5 Always 语句块编写规范和注意要点

4.6 Always 内部功能的扩充和 Always 直接的通信和协调

5 .FPGA 关键电路的设计

5.1 FPGA 板级电路设计五要素

5.1.1 能量供应——电源电路

5.1.2 心脏跳动——时钟电路

5.1.3 状态初始——复位电路

5.1.4 灵活定制——配置电路

5.1.5 自由扩展——外设电路

5.2 FPGA 核心板设计

5.2.1 读懂器件手册

5.2.2 核心板电路设计架构

5.2.3 电源电路设计

5.2.4 时钟和复位电路设计

5.2.5 配置电路设计

- 5.2.6 SDRAM 电路设计
- 5.2.7 引脚分配规划和扩展 I/O 电路
- 5.3 扩展子板设计
 - 5.3.1 基本外设子板
 - 5.3.2 LCD 显示驱动子板
 - 5.3.3 VGA 显示驱动子板
 - 5.3.4 USB 和 UART 串口子板
 - 5.3.5 超声波与视频采集子板

1. 实战一：在 Altera 的 FPGA 开发板上运行一个接口实验程序-交通灯的设计实现，如何控制 Red,Green,Yellow 灯在南北东西各个方向的交替运作。

训练课题：“交通灯的设计实现”

实验要点：

- 1.1 Quartus II 工程创建及属性设置
- 1.2 Quartus II 源文件设计输入方式
- 1.3 Quartus II 约束设计
- 1.4 Quartus II 工程编译
- 1.5 Quartus II 功能仿真
- 1.6 Quartus II 时序仿真
- 1.7 Quartus II 硬件下

第二阶段

熟练掌握硬件描述语言(Verilog HDL)是 FPGA 工程师的基本要求。通过本节课程的学习,学员可以了解目前最流行的 Verilog HDL 语言的基本语法,掌握 Verilog HDL 语言中最常用的基本语法。通过本节课程学习,学员可以设计一些简单的 FPGA 程序,掌握组合逻辑和时序逻辑电路的设计方法。通过实战训练,学员可以对 Verilog HDL 语言有更深入的理解和认识。

Verilog HDL 数字设计

1. 层次建模的概念
2. 模块和端口
3. 门级建
4. 数据流建模
5. 行为级建模
6. 任务和函数
7. 实用建模技术
8. Verilog HDL 操作数和操作符
9. Verilog HDL 和 VHDL 语言的对比
10. Verilog HDL 循环语句
11. Verilog HDL 程序的基本结构
12. Verilog HDL 语言的数据类型和运算符
13. Verilog HDL 语言的赋值语句和块语,阻塞和非阻塞赋值语句的区别
14. Verilog HDL 语言的条件语句,包括 IF 语句和 CASE 语句的典型应用
15. Verilog HDL 语言的其他常用语句
16. Verilog HDL 语言实现组合逻辑电路

17. Verilog HDL 语言实现时序逻辑电路

1. 实战训练二：

训练课题：“顺序执行状态机设计实验”

实验要点：

- 1.1 Quartus II 软件操作
- 1.2 组合逻辑电路设计实现
- 1.3 IF 语句和 CASE 语句的使用

2. 实战训练三：

训练课题：“跑马灯设计实现”

实验要点：

- 2.1 Quartus II 软件操作
- 2.2 时序逻辑电路设计实现
- 2.3 分频原理和实现方法

3. 实战训练四：

训练课题：“7 段数码管测试实验-以动态扫描方式在 8 位数码管“同时”显示 0-7”

实验要点：

- 3.1 Quartus II 软件操作
- 3.2 了解如何按一定的频率轮流向各个数码管的 COM 端送出低电平，同时送出对应的数据给各段。
- 3.3 介绍多个数码管动态显示的方法。

第三阶段

虽然利用第二阶段课程学到的 HDL 基本语法可以完成大部分的 FPGA 功能，但相对复

杂的 FPGA 系统设计中，如果能够合理的应用 Verilog HDL 的高级语法结构，可以达到事半功倍的效果。通过第三天课程的学习，学员可以掌握任务 (TASK)，函数 (FUNCTION) 和有限状态机 (FSM) 的设计方法，可以更好的掌握 FPGA 的设计技术。此外，本节课程还介绍了 QuartusII 软件的两个常用的高级工具 - SignalTAP，可以提高 FPGA 设计和调试的效率。

1. TASK 和 FUNCTION 语句的应用场合
2. Verilog HDL 高级语法结构 - 任务 (TASK)
3. Verilog HDL 高级语法结构 - 任务 (FUNCTION)
4. 有限状态机(FSM)的设计原理及其代码风格
5. 逻辑综合的原则以及可综合的代码设计风格
6. SignalTap II 在线逻辑分析仪使用方法
7. FPGA 编程思想梳理和升华

第四阶段

随着 FPGA 芯片的性能和密度不断提高，基于 FPGA 的 SOPC 系统正在逐渐成熟并且在很多领域得到了应用。第四阶段课程主要给学员介绍 Altera 公司基于 NIOSII 软核的 SoPC 系统设计流程和方法。通过硬件开发板上的 SoPC 系统设计实验，学员能够体会 SoPC 技术给系统设计带来的灵活性。最后通过 FPGA 综合设计实验，学员完成对四天学习内容的回顾和总结

1. 基于 FPGA 系统组成原理和典型方案
2. Altera 公司的解决方案
3. FPGA 的编程思想的总结
4. FPGA 硬件开发的思路

- 5. FPGA 调试方法
- 6. SOPC 开发思路和技巧
- 7. NIOS 开发流程和技巧
- 8. SOPC Build CPU 软核的搭建
- 9. NIOS+SOPC Builder+Quartus 的联合使用案例

1. 实战训练九：

训练课题：“NIOS+SOPC Builder+Quartus 的联合使用”

实验要点：

- 1.1 SOPC 调试方法
- 1.2 SOPC 软件开发流程
- 1.3 NIOS+SOPC Builder+Quartus 的联合开发实验

2. 实战训练十：

训练课题：“SOPC 软核综合设计实验”

训练内容：针对一个综合性实验题目，学员独立完成需求分析，结构设计，代码设计，仿真验证和程序下载固化。

实验要点：

- 2.1 复杂软核的构建
- 2.2 复杂软核的 Nios 编程
- 2.3 软核组织和裁剪