



信息工程系

教 案

课程名称： 可编程逻辑器件应用

教 师： 陈凯斌

总学时： 54

理论学时： 0

实训学时： 54

上课班级： 电子 3+241、电子自主 241

授课学期： 2025-2026 第 2 学期

第 1 章：课程介绍与 FPGA 概述(3 课时)

教学目标

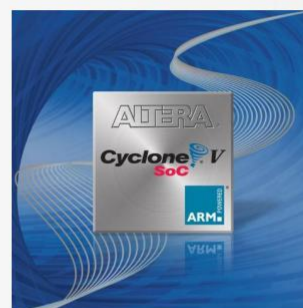
1、知识目标：

了解本课程的教学安排、考核方式、实训要求及学习重点。

掌握 FPGA 的基本定义、特点及应用场景，理解 FPGA 与 ASIC、ARM、DSP 的区别。

◆ FPGA是什么

- FPGA就是“可反复编程的逻辑器件”。它取自 Field Programmable Gate Array这四个英文单词的首个字母，译为“现场（Field）可编程（Programmable）逻辑阵列（Gate Array）”。
- 1985年，Ross Freeman发明了现场可编程门阵列（FPGA），它是一块全部由“开放式门”组成的计算机芯片。采用该芯片，工程师可以根据需要进行灵活编程，以满足不断发展的协议标准或规范。



Altera公司的Cyclone V Soc FPGA器件

了解 FPGA 的主流厂商、产品系列及 FPGA 开发的基本流程。

2、能力目标：

能够清晰梳理本课程的学习脉络，明确各阶段学习重点与实训任务。

能够准确区分 FPGA 与其他同类芯片的核心差异，结合应用场景选择合适的芯片类型。

3、素质目标：

培养学生对 FPGA 技术的学习兴趣和自主学习意识。

引导学生树立“理论联系实际、严谨务实”的学习态度，为后续课程学习奠定基础。

4、课程思政目标：

培养爱国情怀：通过介绍我国在 FPGA、集成电路领域的发展成就和技术突破，引导学生认识核心技术自主可控的重要性，激发为国家数字电子技术进步贡献力量的使命。

感。

树立责任意识：强调 **FPGA** 技术在工业控制、智能终端等关键领域的应用价值，引导学生认识专业学习与社会发展的紧密联系，树立专业责任感。

重难点内容

1、核心重点：

FPGA 的定义、核心特点（可编程性、并行性、高集成度等）及应用领域。

FPGA 与 **ASIC**、**ARM**、**DSP** 在设计流程、灵活性、成本、功耗等方面的差异。

2、教学难点：

理解 **FPGA** 的可编程特性与 **ASIC** 的固定逻辑特性的本质区别。

梳理 **FPGA** 开发的完整流程（需求分析、电路设计、代码编写、仿真验证、板级调试、下载固化），建立工程实践的整体认知。

教学方法与过程

采用讲授法、演示法、互动法相结合，具体过程如下：

1、导入：展示 **FPGA** 在各领域的实际应用案例，提问引导学生思考，引出本章主题，激发学习兴趣。

2、新课讲解：解读教学大纲明确学习目标；讲解 **FPGA** 定义、特点、主流厂商；对比 **FPGA** 与同类芯片差异，梳理 **FPGA** 开发完整流程。

◆ **FPGA、ARM和DSP**

- **ARM** (Advanced RISC Machines) 是微处理器行业的一家知名企业，设计了大量高性能、廉价、耗能低的RISC处理器、相关技术及软件。
- **DSP** (digital singnal processor) ，即数字信号处理器，是一种独特的微处理器，有自己的完整指令系统，能够进行高速、高吞吐量的数字信号处理。
- **ARM**虽然有很多外设，**DSP**虽然具备强大的信号运算能力，但是 **FPGA**可以称作 “All Programmable” 。

3、演示与互动：现场展示 **FPGA** 开发板及简单控制效果，提问检验学习效果，解答学生疑问。

4、总结：梳理本章重点知识，强调“理论+实践”的学习重点，引导学生重视后续实训。

5、布置作业：查阅资料介绍 **FPGA** 最新发展趋势及典型应用案例；熟悉教学大纲并整理学习重点笔记。

教学资源：《勇敢的芯 伴你玩转 Altera **FPGA**》、《**FPGA** 从入门到精通(实战篇)》

第 2 章：逻辑设计基础（一）(3 课时)

教学目标

1、知识目标：

掌握逻辑代数的基本运算、基本定律和常用公式。

学会运用逻辑代数进行逻辑函数的公式法化简方法。

掌握卡诺图的绘制方法，能够运用卡诺图化简 2-4 变量的逻辑函数。

理解逻辑函数化简的意义，为后续 FPGA 逻辑设计奠定理论基础。

2、能力目标：

能够熟练运用逻辑代数基本定律和公式，对简单的逻辑函数进行化简。

能够独立绘制 2-4 变量的卡诺图，根据逻辑函数表达式填充卡诺图，并按照化简规则得到最简逻辑表达式。

3、素质目标：

培养学生的逻辑思维能力和严谨的推理能力。

引导学生总结化简技巧，培养自主学习和归纳总结的能力。

4、课程思政目标：

培养工匠精神：通过逻辑函数化简的反复练习，引导学生树立精益求精的工匠精神，注重细节，追求最简、最优的逻辑设计方案。

培养严谨态度：强调逻辑推理的严谨性，引导学生在化简过程中认真核对每一步运算，避免出现逻辑错误，树立严谨的治学态度。

重难点内容

1、核心重点：

逻辑代数的基本运算（与、或、非、复合逻辑运算）、基本定律和常用公式。

- 输入x和y进行或运算后得到结果z。
- 或运算的原则就是“遇1则1，全0则0”

NOT

$z = \sim x$

非门符号和真值表

x	0	1
z	1	0

- 输入x和y进行与运算后得到结果z。
- 与运算的原则就是“遇0则0，全1则1”

AND

$z = x \& y$

与门符号和真值表

x	0	0	1	1
y	0	1	0	1
z	0	0	0	1

- 输入x经过非门后，输出z为x的取值反向。
- 如输入x=0则z=1，反之，输入x=1则z=0。

OR

$z = x | y$

或门符号和真值表

x	0	0	1	1
y	0	1	0	1
z	0	1	1	1

逻辑函数的公式法化简步骤和技巧。

2-4 变量卡诺图的绘制方法及化简规则（合并相邻项、消去冗余变量）。

2.1 公式化简

核心思路：用上面 9 条定律反复“消项→并项→去冗”。

2.2 卡诺图（K-map）

操作步骤：

1. 写真值表，按格雷码顺序填图。
2. 圈 1（或圈 0 求反函数）：圈最大、数量最少、可重叠。(能覆盖就行，多次被圈的反反而可能要消掉：同一个圈内的所有元素均被圈过2次及以上，则该圈可消除不圈)
3. 读圈得“与项”，再相或。

2、教学难点：

灵活运用逻辑代数基本定律和常用公式进行逻辑函数化简（尤其是冗余公式的应用）。

卡诺图中相邻项的识别（包括首尾相邻、四角相邻），避免遗漏可合并的相邻项。

理解逻辑函数化简的意义（简化电路设计、降低成本、提高可靠性）。

教学方法与过程

采用讲授法、演示法、实践法、互动法相结合，具体过程如下：

1、导入：回顾数字电子技术中逻辑电路知识，提问引出逻辑代数概念，说明其在FPGA设计中的重要性。

2、新课讲解：讲解逻辑代数基础、基本运算及复合运算；讲解逻辑代数基本定律和常用公式，结合例题演示化简技巧；讲解逻辑函数公式法化简的目的、标准和步骤；讲解卡诺图绘制方法及化简规则，演示完整化简过程。

3、实践指导：布置逻辑函数化简练习题，学生分组练习，教师巡视指导，解答疑问，重点指导化简困难的学生。

4、总结：梳理本章核心知识点，回顾化简技巧和注意事项，引导学生归纳公式法与卡诺图化简的适用场景。

5、布置作业：完成逻辑函数公式法和卡诺图化简练习题；整理卡诺图化简核心规则，说明两种化简方法的适用场景。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》

第 3 章：逻辑设计基础（二）(3 课时)

教学目标

1、知识目标：

理解组合逻辑电路和时序逻辑电路的定义、特点及区别。

掌握常见组合逻辑电路（编码器、译码器、数据选择器、加法器）的工作原理和逻辑功能。

掌握 D 触发器的工作原理、特性方程、时序图，理解触发器的触发方式（边沿触发）。

了解寄存器、移位寄存器、计数器的基本工作原理，为后续 Verilog 时序逻辑设计奠定基础。

2、能力目标：

能够准确区分组合逻辑电路与时序逻辑电路，根据电路特点判断电路类型。

能够分析常见组合逻辑电路（如 3-8 线译码器）的逻辑功能，写出真值表和逻辑表达式。

能够绘制 D 触发器的时序图，根据输入信号判断输出状态，理解边沿触发的特点。

3、素质目标：

培养学生的电路分析能力和逻辑推理能力。

引导学生结合前期知识，建立逻辑电路的系统认知，培养知识迁移能力。

4、课程思政目标：

培养系统思维：通过组合逻辑与时序逻辑电路的对比分析，引导学生树立系统思维，学会从整体上把握电路的逻辑功能，理解各部分的衔接关系。

培养责任意识：强调组合逻辑、时序逻辑电路在 FPGA 设计中的核心作用，引导学生认真掌握每一种电路的原理，为后续实训和工程实践奠定基础，树立责任意识。

重难点内容

1、核心重点：

组合逻辑电路与时序逻辑电路的定义、特点及区别。

3-8 线译码器、4 选 1 数据选择器、全加器的工作原理、真值表及逻辑表达式。

D 触发器的工作原理、特性方程、时序图及边沿触发方式。

2、教学难点：

理解时序逻辑电路的“记忆性”（输出由输入和现态决定），区分组合逻辑电路的“无记忆性”。

D 触发器时序图的绘制，尤其是边沿触发的判断（仅在时钟上升沿或下降沿触发，其他时刻输出保持不变）。

梳理常见组合逻辑电路的应用场景，理解其在 FPGA 设计中的实际用途。

教学方法与过程

采用讲授法、演示法、实践法、小组讨论法相结合，具体过程如下：

1、导入：回顾逻辑函数化简知识，提问引出组合逻辑电路概念，展示案例引导学生认识时序逻辑电路，激发探究兴趣。

2、新课讲解：讲解组合逻辑电路定义、特点、分析设计步骤，重点讲解常见组合逻辑电路的工作原理；讲解时序逻辑电路定义、特点及基本组成，对比与组合逻辑电路的区别；重点讲解 D 触发器的结构、原理、特性方程及时序图，演示仿真波形；简要介绍寄存器、移位寄存器、计数器的原理及应用。

3、实践指导：布置练习任务，学生分组分析组合逻辑电路功能、绘制 D 触发器时序图，教师巡视指导，重点指导时序图绘制困难的学生。

4、总结：梳理本章重点知识，强调时序逻辑电路在 FPGA 设计中的重要性，引导学生巩固所学内容。

5、布置作业：分析 3-8 线译码器逻辑功能并说明应用场景；绘制 D 触发器时序图并说明绘制要点；说明组合逻辑与时序逻辑电路的核心区别及应用案例。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》

第 4 章：软件安装与配置(3 课时)

教学目标

1、知识目标：

掌握 Quartus II 13.1 软件的安装步骤、界面布局及基本操作。

掌握 ModelSim-Altera 软件的安装方法及与 Quartus II 的关联配置。

掌握 Notepad++的安装及配置，能够用于 Verilog 代码的编写和编辑。

掌握 USB-Blaster 下载线、串口芯片的驱动安装方法，确保下载调试正常。

2、能力目标：

能够独立完成 Quartus II、ModelSim-Altera、Notepad++软件的安装与配置。

能够完成 USB-Blaster 下载线、串口芯片的驱动安装，排查简单的驱动安装错误。

能够熟悉 Quartus II 软件的基本界面，掌握新建项目的基本操作。

3、素质目标：

培养学生的动手操作能力和问题解决能力。

引导学生耐心细致地完成软件安装与配置，培养严谨的操作习惯。

4、课程思政目标：

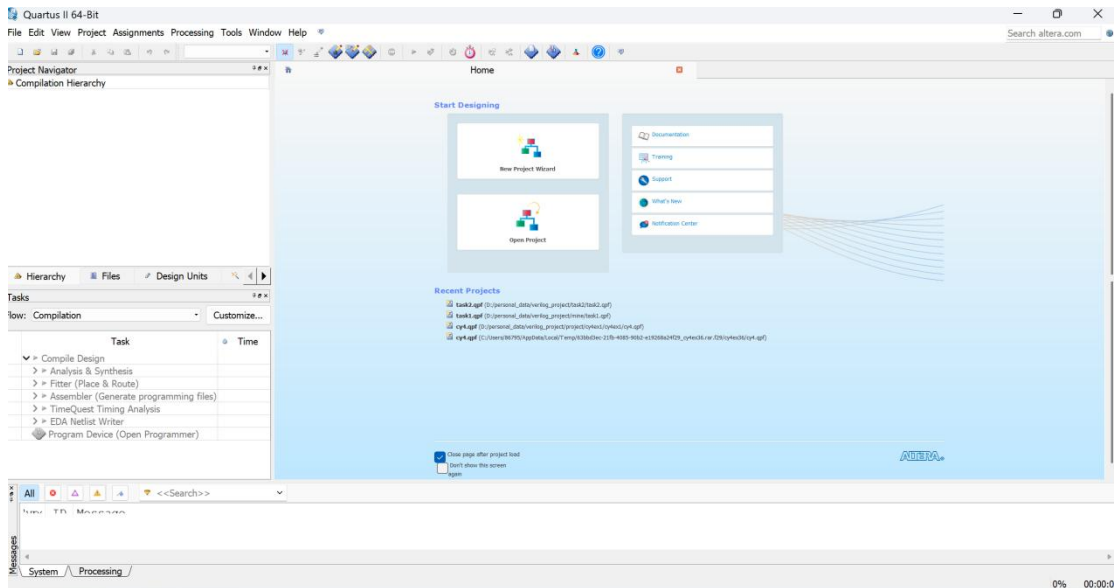
培养工匠精神：通过软件安装与配置的细致操作，引导学生树立精益求精的工匠精神，注重每一个操作步骤，避免因疏忽导致安装失败，培养严谨细致的工作态度。

培养协作意识：鼓励学生在安装过程中互相帮助，分享安装经验，排查错误，培养团队协作意识和互助精神。

重难点内容

1、核心重点：

Quartus II 13.1 软件的安装步骤、许可证配置及基本界面操作。



ModelSim-Altera 与 Quartus II 的关联配置方法，确保能够调用 ModelSim 进行仿真。

USB-Blaster 下载线驱动的安装步骤及常见错误排查。

2、教学难点：

Quartus II 许可证的配置，解决许可证无法识别的问题。

ModelSim-Altera 与 Quartus II 的关联配置，确保仿真工具能够正常调用。

USB-Blaster 驱动安装失败的排查（如系统兼容性、驱动文件缺失等）。

教学方法与过程

采用讲授法、演示法、实践法相结合，具体过程如下：

1、导入：回顾 FPGA 开发流程，强调软件工具的重要性，提问引出本章主题，说明软件安装与配置的必要性。

2、新课讲解与演示：依次演示 Quartus II、ModelSim-Altera、Notepad++ 软件的安装与配置步骤，重点讲解许可证配置、软件关联方法；演示 USB-Blaster 下载线、串口芯片驱动的安装，讲解常见错误及排查方法。

3、实践指导：学生在实验室独立完成软件及驱动安装配置，教师巡视指导，及时解决安装过程中出现的问题，指导学生测试软件安装效果。

4、总结：梳理各软件安装配置的重点和注意事项，提醒学生记录安装过程中的问题及解决方案。

5、布置作业：完成所有软件及驱动安装，截图留存安装成功界面；记录安装配置过程中的问题及解决方案，整理成简短文档。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、

各软件安装包、实验室计算机。

第 5 章：基于 Quartus-II 的与门电路设计与仿真验证(3 课时)

教学目标

1、知识目标

认知 Quartus-II 软件基础架构与 FPGA 开发完整流程，了解工程文件基本规范。

掌握二输入与门核心逻辑规则：全 1 出 1、有 0 出 0。

初步认知 Verilog 基础模块结构、端口定义及 Testbench 仿真激励的基本作用。

2、能力目标

独立完成工程创建、代码编写、编译调试、仿真配置全流程操作。

能够读懂基础 Verilog 代码与仿真激励文件，自主运行仿真并观察波形。

可根据波形结果验证与门逻辑正确性，具备基础的实验报错排查能力。

3、素质目标

养成英文路径、规范命名、文件分类的标准化工程操作习惯。

培养严谨细致、分步实操、独立排查问题的工程实践素养。

树立循序渐进的学习思维，夯实后续复杂电路设计的实操基础。

4、课程思政目标

通过标准化工程实操，践行精益求精的工匠精神与职业规范意识。

了解 FPGA 芯片在电子产业的核心作用，树立自主创新、科技报国的职业理念。

培养攻坚克难、耐心钻研的科学探究精神。

重难点内容

1、重点

Quartus-II 标准开发流程：工程创建、源码编辑、编译、仿真验证全流程操作。

工程文件命名、存储路径的标准化规范，规避基础操作报错。

ModelSim 仿真配置、激励文件编写及仿真波形的读取与验证方法。

2、难点

零基础学生快速理解 Verilog 模块、端口、模块实例化的基础概念。

熟练掌握多层次仿真配置步骤，规避配置遗漏导致的仿真失败问题。

建立波形信号与数字逻辑的对应关系，学会分析仿真结果。

读懂编译报错、警告信息，精准定位并解决基础工程问题。

教学过程与方法

- 1、课堂导入：简要回顾软件安装知识点，明确课程定位：本次实验不深入讲解 Verilog 语法，核心是熟悉 FPGA 完整开发仿真流程。让学生掌握代码编写、工程编译、仿真调试的操作路径，为后续语法学习和实操调试奠定基础。
- 2、简单理论讲解：简述 FPGA 开发核心流程：工程搭建-代码设计-编译-仿真验证。讲解与门基础逻辑规则，不做深度理论拓展。通俗介绍 Verilog 模块结构、仿真激励文件的作用，仅做认知普及，不要求深度掌握。
- 3、分步实操演示：结合任务书的说明，分步演示：工程创建与参数配置、规范新建并保存源码文件、工程编译与报错识别、Testbench 模板修改与激励编写、仿真参数配置、波形观测与结果验证。同时说明常规警告的合理性，消除学生操作顾虑。
- 4、学生独立实操：学生自主完成全套实验流程，重点熟练操作步骤、留存关键实操截图。教师巡回指导，重点解决路径错误、命名不规范、仿真配置遗漏、报错不会处理等共性问题，以熟练流程、规范操作为核心考核标准。
- 5、课堂总结与答疑：梳理 FPGA 开发全流程核心步骤，再次强调本次实验“重流程、轻语法”的学习重点，明确后续语法学习可沿用本次实操流程进行调试。集中解答学生实操共性问题，巩固操作认知。

教学资源：《基于 Quartus-II 的与门电路设计与仿真验证》实训任务书、《勇敢的芯伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、verilog_HDL 讲义。

第 6 章：Verilog 语法（一）(3 课时)

教学目标

1、知识目标：

了解 Verilog HDL 的基本特点、应用场景及与 VHDL 的区别。

掌握 Verilog 的时标定义方法，理解时间单位和时间精度的含义。

掌握 Verilog 模块的基本结构（模块声明、端口声明、内部信号声明、功能描述）。

掌握 Verilog 的注释方法、常量和变量类型，能够正确声明端口方向和端口类型。

2、能力目标：

能够正确使用时标定义命令，设置仿真的时间单位和时间精度。

能够独立编写 Verilog 模块框架，包含输入、输出端口及内部信号声明，添加规范注释。

能够正确声明常量和变量，区分 reg 型与 wire 型变量的用法。

3、素质目标：

培养学生的代码编写能力和规范意识。

引导学生熟悉 Verilog 语法规则，培养严谨的代码编写习惯。

4、课程思政目标：

培养规范意识：强调 Verilog 代码的编写规范（注释、命名、格式），引导学生树立规范编程的意识，培养严谨务实的职业素养。

培养自主学习能力：鼓励学生主动查阅 Verilog 语法手册，熟悉更多语法细节，培养自主学习和终身学习的意识。

重难点内容

1、核心重点：

Verilog 模块的基本结构（module 声明、端口列表、endmodule、内部信号声明、功能描述）。

Verilog 的注释方法、常量和变量类型（reg 型、wire 型）。

端口方向（input、output、inout）和端口类型的声明方法。

2、教学难点：

理解时标定义中时间单位和时间精度的关系，正确设置时标参数。

区分 reg 型与 wire 型变量的核心区别及应用场景（wire 型用于连续赋值，reg 型用于过程赋值）。

编写规范的 Verilog 模块框架，确保端口声明、内部信号声明无误。

教学方法与过程

采用讲授法、演示法、实践法相结合，具体过程如下：

1、导入：回顾 FPGA 开发流程中的代码编写环节，提问引出 Verilog HDL 概念，说明其在 FPGA 设计中的核心作用。

2、新课讲解与演示：讲解 Verilog 的基本特点、与 VHDL 的区别；讲解时标定义方法及仿真中的作用；演示 Verilog 模块基本结构的编写过程，强调代码格式和命名规范；讲解注释方法、常量与变量类型，重点区分 reg 型与 wire 型变量的用法，演示端口声明方法。

3、实践指导：布置练习任务，学生编写简单 Verilog 模块框架，教师巡视指导，纠正代码错误和不规范之处，重点指导变量区分和使用。

4、总结：梳理本章核心知识点，强调代码编写的规范性，提醒学生区分 reg 型与 wire 型变量的用法。

5、布置作业：编写指定端口和内部信号的 Verilog 模块框架；用不同进制表示同一常量；简要说明 reg 型与 wire 型的区别及应用场景。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、verilog_HDL 讲义。

第 7 章：Verilog 语法（二）(3 课时)

教学目标

1、知识目标：

掌握 Verilog 的赋值语句（assign 连续赋值）的用法，理解连续赋值的特点。

掌握 always 过程块的定义和用法，理解阻塞赋值（=）和非阻塞赋值（<=）的区别及应用场景。

能够正确运用赋值语句和 always 过程块描述简单的逻辑功能。

避免赋值语句使用过程中的常见错误，规范代码编写格式。

2、能力目标：

能够使用 assign 连续赋值语句描述简单的组合逻辑（如与门、或门）。

能够使用 always 过程块描述组合逻辑和时序逻辑，正确区分阻塞赋值和非阻塞赋值的用法。

能够排查赋值语句使用过程中的常见错误（如赋值对象与变量类型不匹配）。

3、素质目标：

培养学生的代码编写能力和逻辑思维能力。

引导学生规范使用赋值语句和 always 过程块，培养严谨的代码编写习惯。

4、课程思政目标：

培养严谨态度：通过赋值语句和 always 过程块的练习，引导学生认真核对每一行代码，避免因赋值方式错误、敏感信号遗漏导致逻辑错误，树立严谨的治学态度。

培养问题解决能力：引导学生排查赋值语句使用过程中的常见错误，培养分析问题、解决问题的能力，增强克服困难的信心。

重难点内容

1、核心重点：

assign 连续赋值语句的语法格式和适用场景（描述组合逻辑，赋值对象为 wire 型变量）。

always 过程块的语法格式、敏感信号列表的写法（组合逻辑、时序逻辑的敏感信号区别）。

阻塞赋值（=）和非阻塞赋值（<=）的区别及应用场景（阻塞赋值用于组合逻辑，非阻塞赋值用于时序逻辑）。

2、教学难点：

理解 **always** 过程块的执行机制，尤其是敏感信号列表的作用（触发过程块执行）。

区分阻塞赋值和非阻塞赋值的执行差异（阻塞赋值立即执行、顺序执行；非阻塞赋值延迟执行、并行执行），避免混用导致逻辑错误。

正确编写敏感信号列表（组合逻辑用*表示，时序逻辑用时钟和复位信号）。

教学方法与过程

采用讲授法、演示法、实践法、对比法相结合，具体过程如下：

1、导入：回顾 Verilog 模块结构和变量类型知识，提问引出赋值语句和 **always** 过程块的概念，导入本章新课。

2、新课讲解与演示：讲解 **assign** 连续赋值语句的语法、适用场景，演示描述基本逻辑门的代码；讲解 **always** 过程块的语法、敏感信号列表写法，演示描述组合逻辑和时序逻辑的代码；讲解阻塞与非阻塞赋值的语法和执行差异，结合仿真波形演示效果，明确应用场景，讲解常见错误及解决方案。

3、实践指导：布置练习任务，学生用 **assign** 语句描述基本逻辑门，用 **always** 过程块描述组合逻辑和时序逻辑，教师巡视指导，纠正错误。

4、总结：梳理本章核心知识点，强调赋值语句和过程块的用法，以及常见错误注意事项。

5、布置作业：编写指定逻辑功能的 Verilog 代码；说明阻塞与非阻塞赋值的执行差异及注意事项；排查并修正错误代码。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、verilog_HDL 讲义。

第 8 章：Verilog 语法（三）(3 课时)

教学目标

1、知识目标：

掌握 if-else 条件语句的用法，理解 if-else 与锁存器陷阱的关系，学会避免锁存器生成。

掌握 case 语句（case、casez、casex）的用法，理解 case 语句与 if-else 语句的区别及适用场景。

掌握常用循环语句（for、while、repeat、forever）的用法，明确各循环语句的适用场景。

能够运用条件语句和循环语句描述复杂的逻辑功能。

2、能力目标：

能够使用 if-else 条件语句描述多分支逻辑功能，避免锁存器生成。

能够使用 case 语句描述译码、选择等逻辑功能，根据逻辑需求选择合适的条件语句。

能够使用循环语句（主要是 for 循环）实现重复逻辑（如移位、计数），区分可综合与不可综合的循环语句。

3、素质目标：

培养学生的逻辑思维能力和代码优化能力。

引导学生根据逻辑需求选择合适的语句，培养严谨的代码编写习惯和优化意识。

4、课程思政目标：

培养创新思维：通过条件语句和循环语句的灵活运用，引导学生思考如何优化代码逻辑，简化代码结构，培养创新思维和优化意识。

培养工匠精神：强调避免锁存器生成的重要性，引导学生认真核对每一条条件语句，注重细节，培养精益求精的工匠精神。

重难点内容

1、核心重点：

if-else 条件语句的语法格式、执行流程及避免锁存器生成的方法（确保所有条件下都有赋值）。

case 语句（**case**、**casez**、**casex**）的语法格式、执行流程及适用场景。

for 循环语句的用法（可综合），实现重复逻辑功能。

2、教学难点：

理解锁存器的生成条件，掌握避免锁存器生成的方法（**if-else** 语句不遗漏 **else** 分支，**case** 语句添加 **default** 分支）。

区分 **case** 语句与 **if-else** 语句的适用场景（**case** 语句适合多分支、等值判断；**if-else** 语句适合优先级判断）。

区分可综合与不可综合的循环语句（**for** 循环可综合，**while**、**repeat**、**forever** 循环多用于仿真）。

教学方法与过程

采用讲授法、演示法、实践法、案例分析法相结合，具体过程如下：

1、导入：回顾赋值语句和 **always** 过程块知识，提问引出条件语句和循环语句的概念，导入本章新课。

2、新课讲解与演示：讲解 **if-else** 语句的语法、执行流程，演示描述选择器的代码，讲解锁存器生成条件、危害及避免方法；讲解 **case** 语句的语法、特点，演示描述选择器的代码，对比与 **if-else** 语句的适用场景；讲解各类循环语句的语法，重点演示 **for** 循环实现数据移位的代码，说明各循环语句的适用场景。

3、实践指导：布置练习任务，学生用 **if-else** 和 **case** 语句描述选择器，用 **for** 循环实现数据移位，教师巡视指导，纠正错误。

4、总结：梳理本章核心知识点，强调条件语句、循环语句的用法及优化技巧，引导学生巩固所学内容。

5、布置作业：编写指定逻辑功能的 Verilog 代码；用 **for** 循环实现数据移位并添加注释；说明各循环语句的适用场景及可综合特性。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、verilog_HDL 讲义。

第 9 章：Verilog 语法（四）(3 课时)

教学目标

1、知识目标：

掌握 Verilog 中各类运算符（算术、逻辑、位、关系、条件等）的用法和优先级。

掌握参数化设计的方法（parameter 关键字），理解参数化设计的优势。

掌握模块实例化的方法（门级实例化、行为级实例化），理解模块调用的原理。

了解 Verilog 语法中的常用系统任务和函数，能够运用简单系统任务进行仿真调试。

2、能力目标：

能够熟练运用各类运算符编写 Verilog 表达式，遵循运算符优先级避免逻辑错误。

能够使用 parameter 关键字进行参数化设计，实现模块的复用和灵活修改。

能够正确进行模块实例化，实现复杂逻辑功能的模块化设计。

能够运用简单系统任务进行仿真调试，查看信号波形和变量值。

3、素质目标：

培养学生的代码编写能力和模块化设计思维。

引导学生规范运用语法知识，培养代码复用和优化意识，提升编程效率。

4、课程思政目标：

培养系统思维：通过模块化设计和模块实例化的学习，引导学生树立系统思维，学会将复杂任务拆解为简单模块，理解模块间的衔接关系，提升系统设计能力。

培养创新意识：通过参数化设计的学习，引导学生思考如何优化模块设计，实现代码复用，培养创新意识和高效编程的职业素养。

重难点内容

1、核心重点：

各类运算符的用法和优先级，能够正确编写 Verilog 表达式。

parameter 参数化设计的方法和优势，实现模块参数的灵活配置。

模块实例化的方法（门级、行为级），正确进行模块调用和信号连接。

2、教学难点：

各类运算符的优先级区分，避免因优先级错误导致的逻辑功能异常。

模块实例化中的信号连接（位置关联、名称关联），避免信号连接错误。

理解参数化设计的核心思想，灵活运用 `parameter` 实现模块复用和修改。

教学方法与过程

采用讲授法、演示法、实践法相结合，具体过程如下：

- 1、导入：回顾前期 Verilog 语法知识，提问引出运算符、参数化设计和模块实例化的必要性，导入本章新课。
- 2、新课讲解与演示：分类讲解各类运算符的用法和优先级，结合例题演示表达式编写，强调优先级注意事项；讲解 `parameter` 参数化设计的语法和优势，演示参数化模块的编写和调用；讲解模块实例化的两种方法，演示门级实例化（基本逻辑门）和行为级实例化（自定义模块）的代码，说明信号连接方法；简要介绍常用系统任务，演示仿真调试中的应用。
- 3、实践指导：布置练习任务，学生编写含各类运算符的表达式、参数化模块及模块实例化代码，教师巡视指导，纠正错误。
- 4、总结：梳理本章核心知识点，强调运算符优先级、参数化设计和模块实例化的重点，引导学生巩固所学内容。
- 5、布置作业：编写含多种运算符的 Verilog 代码；设计一个参数化模块并进行实例化；简要说明模块实例化的两种方法及区别。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、`verilog_HDL` 讲义。

第 10 章：Testbench 编写与仿真验证(3 课时)

教学目标

1、知识目标：

理解 Testbench 的作用和基本结构，掌握 Testbench 的编写规范。

掌握激励信号（时钟、复位、输入数据等）的生成方法。

学会使用 ModelSim 软件进行仿真验证，查看仿真波形，分析仿真结果。

掌握仿真调试的基本方法，能够排查简单的代码逻辑错误。

2、能力目标：

能够独立编写简单的 Testbench，生成所需的激励信号，对设计模块进行仿真验证。

能够使用 ModelSim 软件加载 Testbench，运行仿真，查看和分析仿真波形。

能够根据仿真结果排查代码中的简单逻辑错误，优化设计模块。

3、素质目标：

培养学生的仿真调试能力和问题排查能力。

引导学生树立“设计-仿真-调试”的工程思维，培养严谨的工程实践态度。

4、课程思政目标：

培养严谨态度：通过仿真验证和调试的学习，引导学生认识到仿真在 FPGA 设计中的重要性，认真对待每一次仿真，细致分析仿真结果，树立严谨的工程态度。

培养坚韧品质：引导学生在调试过程中勇于面对错误，耐心排查问题，培养坚韧不拔、持之以恒的品质。

重难点内容

1、核心重点：

Testbench 的基本结构和编写规范，激励信号（时钟、复位）的生成方法。

ModelSim 软件的仿真操作流程（加载文件、运行仿真、查看波形）。

根据仿真波形分析设计模块的逻辑功能，排查简单错误。

2、教学难点：

复杂激励信号的生成方法（如随机输入数据）。

仿真波形的分析技巧，准确判断逻辑错误的位置和原因。

Testbench 的优化方法，提高仿真效率和准确性。

教学方法与过程

采用讲授法、演示法、实践法相结合，具体过程如下：

1、导入：回顾 Verilog 设计模块的编写方法，提问引出“如何验证设计模块的逻辑功能是否正确”，导入 Testbench 编写与仿真验证的主题。

2、新课讲解与演示：讲解 Testbench 的作用和基本结构，演示 Testbench 的编写规范；讲解时钟、复位、输入数据等激励信号的生成方法，结合代码演示实现过程；演示 ModelSim 软件的仿真操作流程，包括加载设计文件和 Testbench、设置仿真时长、运行仿真、添加波形、查看和分析波形；讲解仿真调试的基本方法，结合错误案例演示排查过程。

3、实践指导：布置练习任务，学生为前期编写的设计模块（如 D 触发器、选择器）编写 Testbench，进行仿真验证，教师巡视指导，解答疑问，指导学生排查错误。

4、总结：梳理 Testbench 编写规范、仿真操作流程和调试方法，强调仿真验证在 FPGA 设计中的核心作用。

5、布置作业：为前期编写的四选一选择器编写 Testbench，生成合理激励信号；使用 ModelSim 进行仿真，查看波形并分析仿真结果；记录仿真过程中遇到的问题及解决方案。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、verilog_HDL 讲义。

第 11 章：时钟二分频 FPGA 设计和仿真验证(3 课时)

教学目标

1、知识目标：

掌握时钟二分频的工作原理和实现方法（基于 D 触发器、计数器）。

掌握 FPGA 完整开发流程（新建工程、编写代码、编译、仿真验证、下载调试）。

了解 Quartus II 软件中编译报错的排查方法，掌握下载调试的基本操作。

2、能力目标：

能够独立完成时钟二分频模块的 Verilog 代码编写和 Testbench 编写。

能够使用 Quartus II 进行工程创建、代码编译，排查简单的编译报错。

能够完成仿真验证和下载调试，实现时钟二分频功能，查看实际运行效果。

3、素质目标：

培养学生的工程实践能力和综合应用能力。

引导学生熟悉 FPGA 完整开发流程，培养严谨的工程实践习惯和问题解决能力。

4、课程思政目标：

培养工程素养：通过完整的 FPGA 开发流程实训，引导学生树立“理论联系实际”的工程素养，注重每一个开发环节，培养严谨务实的工作态度。

培养协作能力：鼓励学生在实训过程中互相交流经验，共同排查问题，培养团队协作能力和互助精神。

重难点内容

1、核心重点：

时钟二分频的工作原理和 Verilog 代码实现方法。

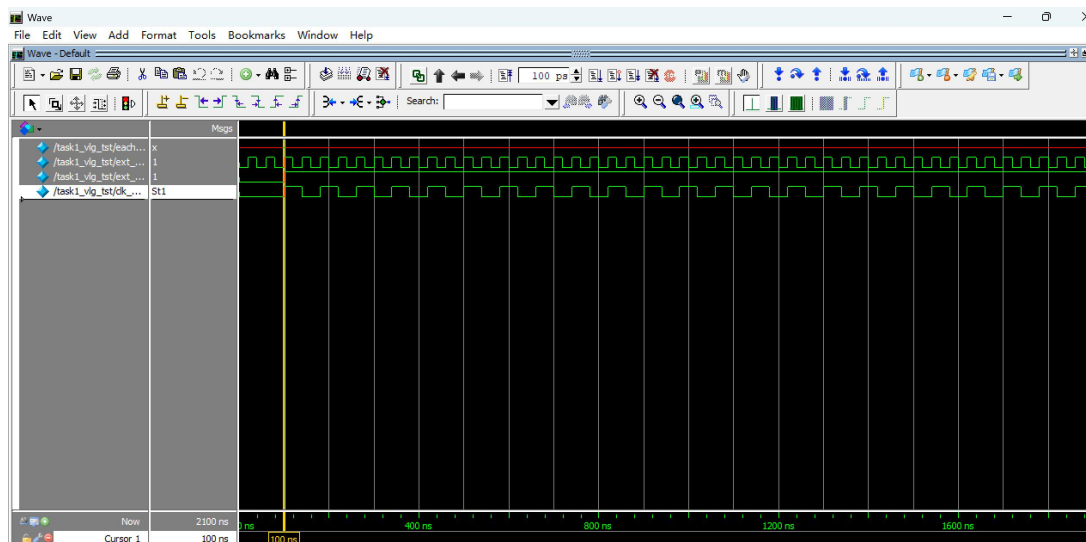
```
source_code/task1.v
1 //对外部输入时钟做二分频
2 module task1(
3     input ext_clk_25m, //外部输入25MHz时钟信号
4     input ext_rst_n, //外部输入复位信号，低电平有效
5     output reg clk_12m5 //二分频时钟信号
6 );
7 always @(posedge ext_clk_25m or negedge ext_rst_n)
8     if(!ext_rst_n) clk_12m5 <= 1'b0;
9     else clk_12m5 <= ~clk_12m5;
10 endmodule
11
```

FPGA 完整开发流程的每一个环节（新建工程、编译、仿真、下载）的操作方法。

Quartus II 编译报错和下载调试中常见问题的排查方法。

2、教学难点：

时钟二分频代码的逻辑设计，确保分频后的时钟稳定、无毛刺。



Quartus II 编译报错的解读和排查（如语法错误、端口错误）。

下载调试中硬件连接和配置错误的排查，确保程序正常下载和运行。

教学方法与过程

采用讲授法、演示法、实践法相结合，以实践操作为主，具体过程如下：

1、导入：回顾时钟信号在 FPGA 设计中的作用，提问引出时钟分频的需求，说明时钟二分频的应用场景，导入本次实训主题。

2、新课讲解与演示：讲解时钟二分频的工作原理，演示两种实现方法（D 触发器、计数器）的 Verilog 代码编写；演示 FPGA 完整开发流程，包括 Quartus II 新建工程、导入代码、编译、排查报错，ModelSim 仿真验证，以及程序下载调试的完整操作；讲解常见问题及排查方法。

3、实践指导：学生分组完成时钟二分频设计的完整开发流程，教师巡视指导，及时解决学生在工程创建、编译、仿真、下载过程中遇到的问题，重点指导代码编写和报错排查。

4、总结：梳理时钟二分频的实现方法和 FPGA 完整开发流程，总结实训过程中的常见问题及解决方案，引导学生归纳实训要点。

5、布置作业：完成时钟二分频设计的完整开发流程，截图留存各环节成功界面；编写实训报告，总结实训过程、收获及遇到的问题和解决方案；尝试实现时钟四分频设计。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、实训任务书。

第 12 章：FPGA 蜂鸣器 PWM 发声设计和板级验证 (6 课时)

教学目标

1、知识目标：

掌握 PWM（脉冲宽度调制）的工作原理和应用场景。

掌握蜂鸣器的工作原理（有源、无源）及 FPGA 驱动方法。

掌握 PWM 波形的 Verilog 代码实现方法，理解占空比对蜂鸣器音量的影响。

掌握 FPGA 程序下载固化的方法，确保设备断电后程序不丢失。

2、能力目标：

能够独立完成蜂鸣器 PWM 驱动模块的 Verilog 代码编写和 Testbench 编写。

能够进行仿真验证，确保 PWM 波形正确生成；完成程序下载、固化和板级调试，实现蜂鸣器发声控制。

能够调整 PWM 占空比，实现蜂鸣器音量调节，排查驱动过程中的常见问题。

3、素质目标：

培养学生的综合应用能力和板级调试能力。

引导学生将理论知识应用于实际硬件驱动，培养工程实践思维 and 创新能力。

4、课程思政目标：

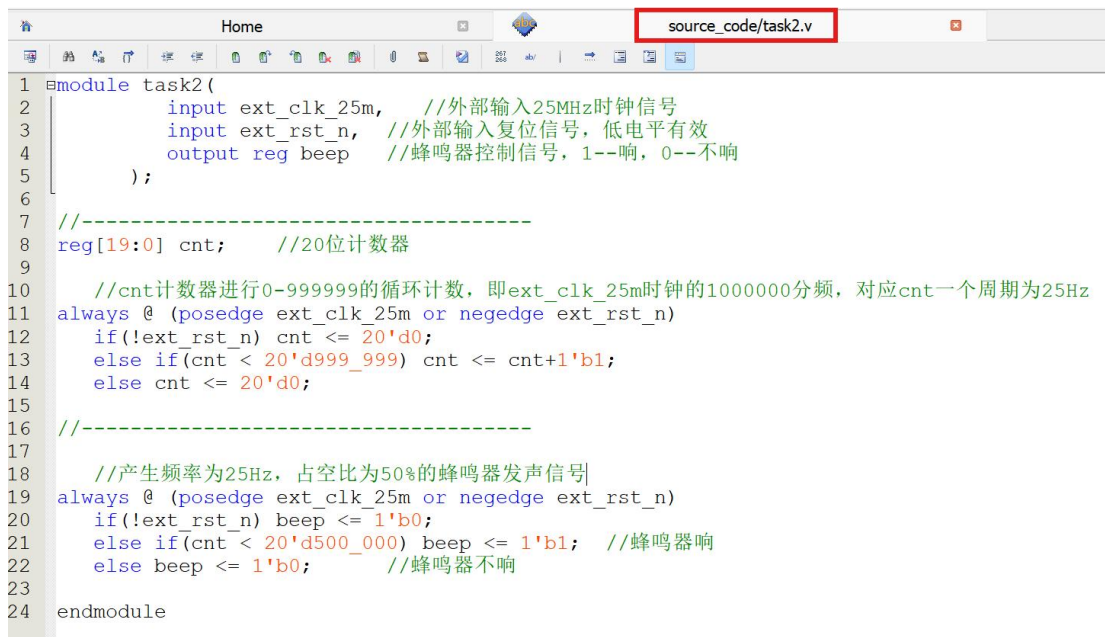
培养创新思维：通过 PWM 驱动蜂鸣器的实训，引导学生思考 PWM 技术的其他应用场景，培养创新思维和知识迁移能力。

培养责任意识：强调硬件驱动设计的严谨性，引导学生认真编写代码、调试程序，避免因设计错误导致硬件损坏，树立责任意识。

重难点内容

1、核心重点：

PWM 的工作原理和 Verilog 代码实现方法，占空比的调整方式。

The image shows a screenshot of a Verilog code editor window. The title bar indicates the file is 'source_code/task2.v'. The code defines a module named 'task2' with three ports: 'ext_clk_25m' (input), 'ext_rst_n' (input), and 'beep' (output). The code implements a 20-bit counter 'cnt' that increments on the rising edge of 'ext_clk_25m' unless 'ext_rst_n' is active low. The counter is used to generate a 25Hz square wave for the 'beep' signal, with a 50% duty cycle. The code is as follows:

```
1 module task2(  
2     input ext_clk_25m,    //外部输入25MHz时钟信号  
3     input ext_rst_n,    //外部输入复位信号，低电平有效  
4     output reg beep    //蜂鸣器控制信号，1--响，0--不响  
5 );  
6  
7 //-----  
8 reg[19:0] cnt;    //20位计数器  
9  
10 //cnt计数器进行0-999999的循环计数，即ext_clk_25m时钟的1000000分频，对应cnt一个周期为25Hz  
11 always @ (posedge ext_clk_25m or negedge ext_rst_n)  
12     if(!ext_rst_n) cnt <= 20'd0;  
13     else if(cnt < 20'd999_999) cnt <= cnt+1'b1;  
14     else cnt <= 20'd0;  
15  
16 //-----  
17  
18 //产生频率为25Hz，占空比为50%的蜂鸣器发声信号  
19 always @ (posedge ext_clk_25m or negedge ext_rst_n)  
20     if(!ext_rst_n) beep <= 1'b0;  
21     else if(cnt < 20'd500_000) beep <= 1'b1;    //蜂鸣器响  
22     else beep <= 1'b0;    //蜂鸣器不响  
23  
24 endmodule
```

蜂鸣器的工作原理及 FPGA 驱动电路的连接和代码实现。

FPGA 程序下载固化的方法和板级调试技巧。

2、教学难点：

PWM 波形的精准生成，确保占空比可调、波形稳定。

蜂鸣器驱动过程中的硬件连接错误排查，以及音量调节的优化方法。

程序下载固化失败的排查，确保设备断电后程序正常运行。

教学方法与过程

采用讲授法、演示法、实践法相结合，以实践操作为主，具体过程如下：

1、导入：回顾时钟分频实训内容，展示蜂鸣器实际应用案例（如智能设备报警、简易音乐播放），提问引出 PWM 驱动技术，说明本次实训的核心目标和实践意义，激发学生实训兴趣。

2、新课讲解与演示：讲解 PWM 脉冲宽度调制的工作原理、占空比概念及应用场景，结合波形图演示占空比对输出信号的影响；讲解有源与无源蜂鸣器的区别、工作电压及驱动原理，演示 FPGA 与蜂鸣器的硬件连接方式；演示 PWM 波形的 Verilog 代码编写（基于计数器实现占空比可调），讲解代码核心逻辑（计数器计数、占空比比较、输出控制）；演示 Testbench 编写方法，用于验证 PWM 波形正确性；演示 Quartus II 程序下载固化的完整操作，讲解固化原理及注意事项。

3、实践指导：学生分组完成实训任务，教师巡视指导，重点关注三个环节：一是硬件连接的规范性，避免接反电源导致蜂鸣器损坏；二是 PWM 代码编写与调试，指导学生修改代码中的占空比参数，观察波形变化；三是下载固化与板级调试，帮助学生排

查下载失败、蜂鸣器不发声等常见问题，引导学生调整占空比，实现音量调节。

4、实训总结：邀请 2-3 组学生展示实训成果（蜂鸣器发声、占空比调节效果），分享实训过程中的问题及解决方案；教师梳理本次实训的核心知识点和操作要点，总结常见错误（硬件接反、代码逻辑错误、固化失败）及排查技巧，强调理论与实践结合的重要性。

5、布置作业：完成蜂鸣器 PWM 驱动设计的完整流程（代码编写、仿真验证、下载固化、板级调试），截图留存 PWM 仿真波形和蜂鸣器工作效果；编写实训报告，详细记录实训目的、原理、步骤、成果及遇到的问题和解决方案；尝试修改代码，实现蜂鸣器播放简易音阶（如 do、re、mi）。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》、实训任务书。

第 13 章：FPGA 流水灯应用设计、仿真及板级验证 (6 课时)

教学目标

1、知识目标：

掌握流水灯的工作原理和显示逻辑（循环点亮、方向可控、速度可调）。

掌握 LED 灯的 FPGA 驱动方法，理解时序逻辑在流水灯控制中的应用。

掌握流水灯控制模块的 Verilog 代码编写方法，实现流水效果的灵活控制。

熟练掌握 FPGA 设计的完整流程（代码编写、仿真验证、板级调试），规范完成设计验证全流程。

2、能力目标：

能够独立完成流水灯控制模块的 Verilog 代码编写，实现指定流水效果（如左移、右移、循环点亮）。

能够编写 Testbench 对流水灯模块进行仿真验证，查看时序波形，验证逻辑正确性。

能够完成 FPGA 与 LED 灯的硬件连接，进行板级调试，排查流水灯不亮、效果异常等问题，实现稳定的流水显示。

能够修改代码参数，实现流水灯速度可调、方向可控，提升代码优化能力。

3、素质目标：

培养学生的时序逻辑设计能力和综合调试能力，强化工程实践思维。

引导学生规范完成“设计-仿真-调试”全流程，培养严谨务实的工作态度和耐心细致的做事风格。

4、课程思政目标：

培养系统思维：通过流水灯时序逻辑的设计和全流程验证，引导学生树立“全局把控、分步实施”的系统思维，理解每一个设计环节的关联性。

培养坚持品质：强调流水灯显示的稳定性和流畅性，引导学生反复调试代码、优化时序，培养精益求精、坚持不懈的品质。

重难点内容

1、核心重点：

流水灯的工作逻辑和 Verilog 代码实现（时序控制、流水方向、速度调节）。

流水灯模块的 Testbench 编写与仿真验证，确保逻辑功能正确。

FPGA 与 LED 灯的硬件连接规范，板级调试技巧及常见问题排查。

2、教学难点：

流水灯时序逻辑的精准控制，避免 LED 灯点亮紊乱、闪烁或流水速度不均。

仿真验证中时序波形的分析，准确判断逻辑错误的位置和原因。

板级调试中硬件连接错误、代码时序不匹配导致的流水效果异常排查。

教学方法与过程

采用讲授法、演示法、实践法相结合，以“设计-仿真-调试”全流程为主线，具体过程如下：

1、导入：回顾数码管显示实训的模块化设计思路，展示流水灯实际应用案例（如装饰灯、指示灯、设备状态提示），提问引导学生思考流水灯的工作逻辑和实现方法，说明本次实训的核心是掌握 FPGA 应用设计的全流程，导入本章实训主题。

2、新课讲解与演示：讲解流水灯的工作原理、常见流水效果（左移、右移、循环）及设计思路；演示流水灯控制模块的 Verilog 代码编写，重点讲解时序控制逻辑（基于计数器实现速度调节）；演示 Testbench 编写方法，用于验证流水灯逻辑正确性，展示仿真波形的查看与分析技巧；演示 Quartus II 编译、ModelSim 仿真、板级下载调试的完整操作，讲解常见问题及排查方法。

3、实践指导：学生分组完成流水灯应用设计全流程，教师巡视指导，重点关注四个环节：一是代码编写的规范性，指导学生优化时序逻辑；二是 Testbench 编写与仿真，帮助学生分析波形异常问题；三是硬件连接的规范性，检查引脚连接是否正确；四是板级调试，引导学生排查流水灯不亮、效果异常等问题，尝试修改参数实现速度和方向可调。

4、总结：邀请各组展示实训成果（流水灯效果演示），分享设计思路和调试经验；教师梳理本次实训的核心知识点和全流程要点，总结代码编写、仿真验证、板级调试的常见错误及排查技巧，强调“仿真先行、调试为辅”的工程设计理念。

5、布置作业：完成流水灯应用设计全流程（代码编写、仿真验证、板级调试），截图留存仿真波形和流水灯工作效果；编写实训报告，详细记录设计思路、全流程步骤、成果及遇到的问题和解决方案；尝试修改代码，实现两种及以上流水效果切换（如左移/右移切换）。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》。

第 14 章：FPGA3-8 译码器应用设计、仿真及板级验证(6 课时)

教学目标

1、知识目标：

掌握 3-8 译码器的工作原理、真值表及逻辑功能，理解译码器的核心作用。

掌握基于 Verilog 代码实现 3-8 译码器的方法。

熟练掌握 3-8 译码器的 Testbench 编写方法，能够通过仿真验证译码逻辑正确性。

掌握 3-8 译码器的板级验证方法，实现 FPGA 与译码器、负载（LED 灯）的联动控制。

2、能力目标：

能够独立编写 3-8 译码器的 Verilog 代码，准确实现其逻辑功能。

能够编写 Testbench 对 3-8 译码器进行仿真验证，分析仿真波形，排查逻辑错误。

能够完成 FPGA、3-8 译码器、LED 灯的硬件连接，进行板级调试，实现译码器的正确译码与负载控制。

能够区分行为级与门级描述的差异，根据需求选择合适的设计方法，提升模块化设计能力。

3、素质目标：

培养学生的逻辑设计能力、仿真验证能力和板级调试能力，强化工程实践思维。

引导学生规范设计流程，培养严谨务实的治学态度和精益求精的做事风格，注重细节把控。

4、课程思政目标：

培养严谨思维：通过 3-8 译码器真值表、逻辑功能的严格验证，引导学生树立严谨的逻辑思维，认识到每一个逻辑环节的准确性对整体设计的影响。

培养应用意识：强调译码器在数字系统中的核心应用（如地址译码、指令译码），引导学生认识专业知识的实际价值，树立“学以致用”的应用意识和责任担当。

重难点内容

1、核心重点：

3-8 译码器的工作原理、真值表及逻辑功能，Verilog 代码实现。

3-8 译码器的 Testbench 编写与仿真验证，确保译码逻辑正确。

FPGA、3-8 译码器、LED 灯的硬件连接规范，板级验证方法及常见问题排查。

2、教学难点：

3-8 译码器门级描述的代码编写，准确还原其逻辑电路结构。

仿真验证中，根据真值表核对仿真波形，精准排查译码错误（如输出电平异常、译码错位）。

板级调试中，硬件连接错误（如引脚接错、电源异常）及代码逻辑不匹配导致的译码失败排查。

教学方法与过程

采用讲授法、演示法、实践法、对比法相结合，以“原理-设计-仿真-验证”为主线，具体过程如下：

1、导入：回顾前期组合逻辑电路知识，提问引出译码器的概念和应用场景（如数字系统中的地址译码），展示 3-8 译码器的实际应用案例，说明本次实训的核心是掌握译码器的设计、仿真与板级验证全流程，导入本章实训主题。

2、新课讲解与演示：讲解 3-8 译码器的工作原理、真值表、逻辑功能及核心作用，结合逻辑电路图演示其工作过程；讲解 Verilog 3-8 译码器代码编写；演示 Testbench 编写方法，用于验证译码器的逻辑正确性，展示如何根据真值表核对仿真波形；演示编译、仿真、下载调试的完整操作，讲解常见问题及排查方法。

3、实践指导：学生分组完成 3-8 译码器应用设计全流程，教师巡视指导，重点指导四个方面：一是代码编写，区分行为级与门级描述的编写要点，纠正逻辑错误；二是仿真验证，帮助学生根据真值表核对波形，排查译码异常；三是硬件连接，检查引脚连接和电源供给，避免设备损坏；四是板级调试，引导学生排查译码失败、LED 点亮异常等问题，确保译码器正常工作。

4、总结：邀请各组展示实训成果（3-8 译码器板级验证效果），分享设计方法和调试经验；教师梳理本次实训的核心知识点，总结 3-8 译码器的设计要点、仿真技巧和板级调试重点，对比行为级与门级描述的优劣，引导学生巩固所学内容，强化逻辑设计和全流程验证思维。

5、布置作业：完成 3-8 译码器的 Verilog 代码编写、仿真验证及板级调试，截图留存仿真波形和验证效果；编写实训报告，详细记录设计思路、代码逻辑、全流程步骤及遇到的问题和解决方案；尝试将 3-8 译码器与流水灯模块结合，实现基于译码器的流

水灯控制。

教学资源：《勇敢的芯 伴你玩转 Altera FPGA》、《FPGA 从入门到精通(实战篇)》。