

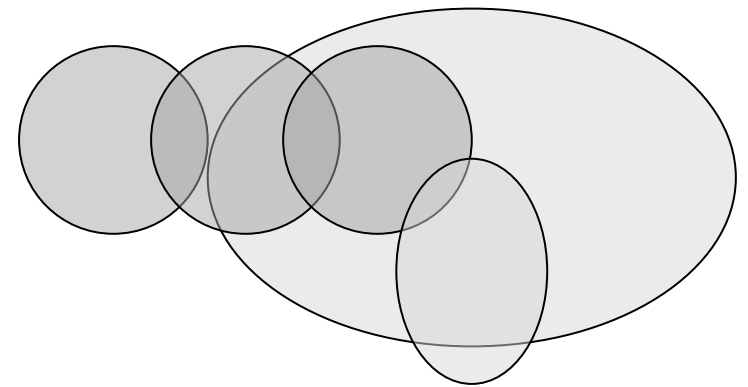
从数字逻辑 到计算机组成原理

吕昱峰

2020.2.10

Why this?

- 数字逻辑是组成原理和体系结构的前导课
 - 仿真、综合工具使用不熟 (Vivado)
 - 面向硬件电路的设计思路缺乏 (Verilog)
-
- 要会造CPU，而不是只会背课本
 - 17级推广过程中遇到的问题
 - 对硬件设计的恐惧



You need.....

- 复习之前学的数字电路知识
- 学会用Vivado和编辑器
- 看PPT，避坑

数字逻辑知识索引

- 数值表示和数制，重点是二进制数、十六进制数。
- 数值的原码表示和补码表示，有符号数、无符号数，溢出。
- 基本逻辑门：与、或、非。
- 布尔代数：逻辑表达式，真值表，逻辑运算的常用运算律。

数字逻辑知识索引

- $n-2^n$ 变量译码器，如“3-8译码器”等。
- 不同位宽的数据选择器，如“1位宽4选1”、“8位宽32选1”等。
- 一位全加器，串行进位多位全加器。
- 锁存器与触发器。
- 触发器时序分析，触发器的Setup延迟、Hold延迟和Clock-to-Q延迟的含义及原理
- 典型状态机设计。
- 计数器设计，如二进制计数器、模- n 计数器。
- 移位寄存器设计。

数字逻辑知识索引

- 组合逻辑电路和时序逻辑电路的原理。
- 只读存储器ROM基本原理。
- 随机存储器RAM基本原理。
- 动态存储器DRAM基本原理。
- 现场可编程门阵列FPGA基本原理。

避坑开始。。。

- Verilog
- Vivado
- 一些常见的误解

Verilog语言

- 怎么学
- 面向硬件电路的设计思路
- 可综合代码（限制版）
- 代码风格建议（要求）
- Demo

怎么学？

- 采用RTL (Register Transfer Level, 寄存器传输级) 设计
- 使用限制的可综合子集
- 把数字逻辑课自己实现的和参考代码对比思考 (后续设计以参考代码为标准)

面向硬件电路的设计思路

- 时刻谨记这不是在写代码，而是在设计电路（并行）
- 先进行电路结构设计，再进行Verilog代码编写
- “自顶向下、模块划分、逐层细化”的设计步骤
- 不要“写一点，试一下，改一点，再试一下”

可综合代码 (Verilog-1995)

- 模块声明 module, endmodule
- 端口声明 input, output, inout
- 线网数据类型 wire
- 变量数据类型 reg, integer
- 参数常量 (parameter constants)
- 整型数 (literal integer numbers)
- 模块实例化
- 连续赋值语句 assign
- always结构化过程语句 always
- begin ... end块
- 阻塞赋值 (=) 和非阻塞赋值 (<=)
- 条件判断语句 if, if...else, case
- for循环

可综合代码 (Verilog-2001)

- 敏感列表中用逗号分隔
- 组合逻辑敏感列表用@*
- 将端口和数据类型集成在一起的声明方式
- ANSI C 形式的端口声明
- 连续赋值中的隐藏线网
- 多维数组
- 数组位选或部分选择运算符+:
和 -:
- sized parameters
- 以名字相关传递的参数
(parameter)
- 本地参数 (local parameter)
- 编译制导`ifndef, `elsif, `undef
- 常数函数 (constant functions)
- generate表达式

代码风格建议（要求）

- 数据通路上的组合逻辑用assign语句写，禁止用always语句写
- 用always语句写组合逻辑只允许出现在生成状态机next state的时候，该语句中只允许出现阻塞赋值（=）
- 写时序逻辑的always语句中只允许出现非阻塞赋值（<=）。
- 寄存器堆封装成单独的模块，以实例化方式使用。
- case语句任何情况下都要有default分支。
- 模块实例化时的参数和端口只允许用名字相关的方式进行赋值和连接。

代码风格建议（要求）

- 数据通路的组合逻辑中，1比特的逻辑运算用&、|、~、^这类位运算符；控制信号的组合逻辑中，1比特的逻辑运算用&&、||、!这三个逻辑运算符。
- 如果你对于表达式中运算符的优先级没有确切把握的话，要么立刻去找一本参考书明确它，要么按照你所需要的优先级层次加上括号；我们推荐前者，因为括号太多影响代码的可阅读性。
- 不要在RTL代码中加入用于仿真的延迟信息；时序是否满足的问题通过综合阶段的STA保证，不是通过仿真来保证。
- 禁用//synopsys parallel case、//synopsys full case。

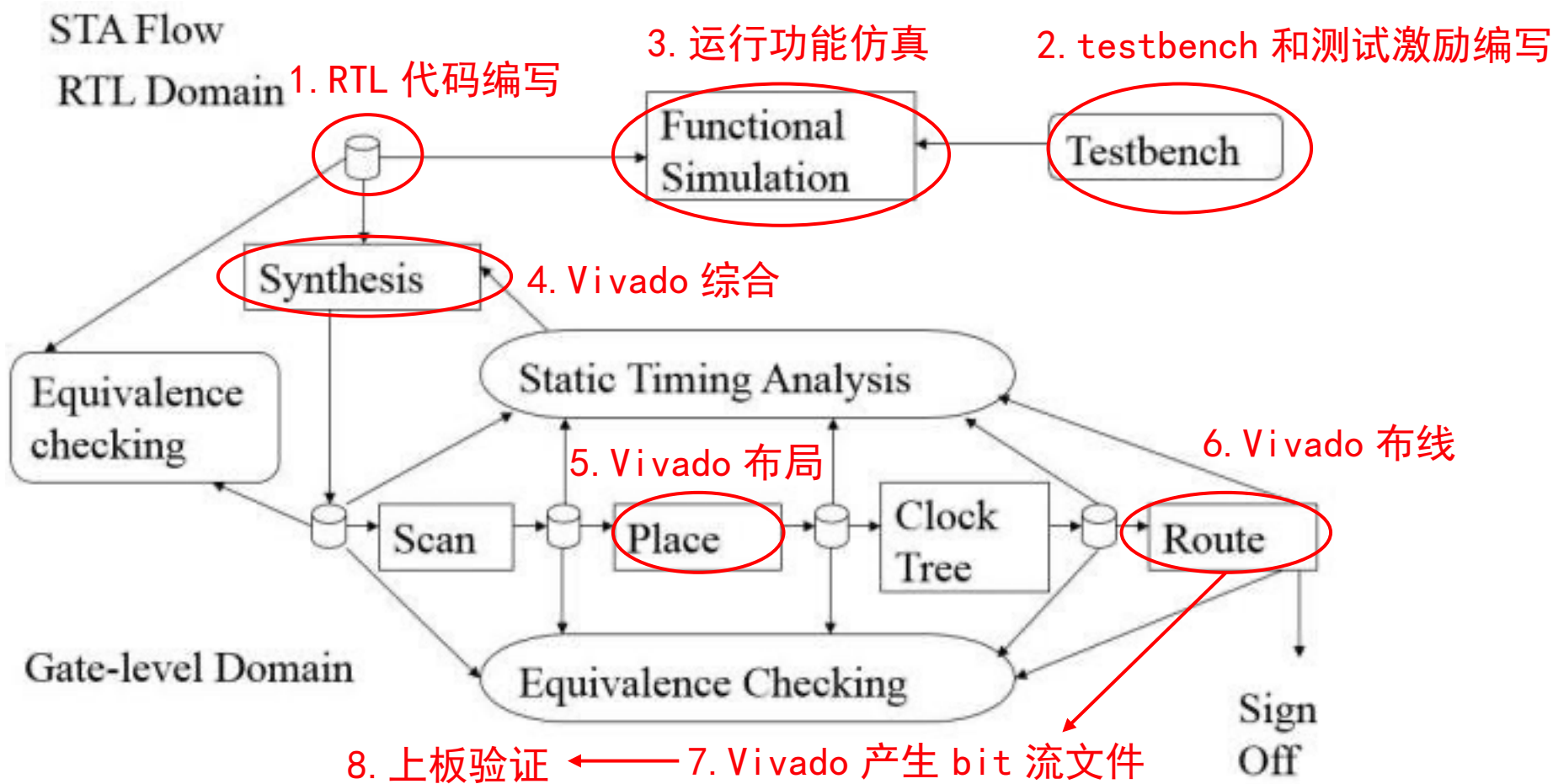
Demo(见压缩包)

- 模块声明和实例化 (01)
- 基础逻辑门 (02)
- 译码器与编码器 (03、04)
- 多路选择器 (05、06)
- 寄存器 (12、13、14)
- 寄存器堆 (15)

Vivado

- 基本安装与使用（见附件）
- 开发流程中的一些问题

数字电路设计与 FPGA 开发对照图





- Sources单独放在rtl文件夹，不要copy into project
- 不要打包IP！
- 禁止Block Design！！
- 行为仿真！！！！
- 仿真不正确的情况下不要继续后面的步骤！

其他内容

- Verilog设计中经常遇到的问题，参《A09_CPU仿真调试说明_v1.00》
- Vivado自带编辑器太难用，推荐使用第三方编辑器，如Sublime, VS Code等，修改方法参考《1.修改Vivado第三方编辑器及Verliog语法环境》

推荐两本教材之外的书

- **数字设计和计算机体系结构**：一本数字逻辑和计组结合的实践指导书，作者是MIPSfpga项目的发起人，我们的实验以该书为基础进行设计。
- **自己动手写CPU**：如果想要快速实现一个CPU，看这本；如果想要SOLO，看这本

写在最后

- 硬件电路设计没有那么难，最难的是如何打破自己的畏难情绪。
其实人人写个CPU也不是什么难事。
- 疫情期间，唯有直播了，但还是希望这点内容能有用。
- 最后的最后，武汉加油！中国加油！