



中国科学技术大学  
University of Science and Technology of China

# 实验04

# Verilog硬件描述语言

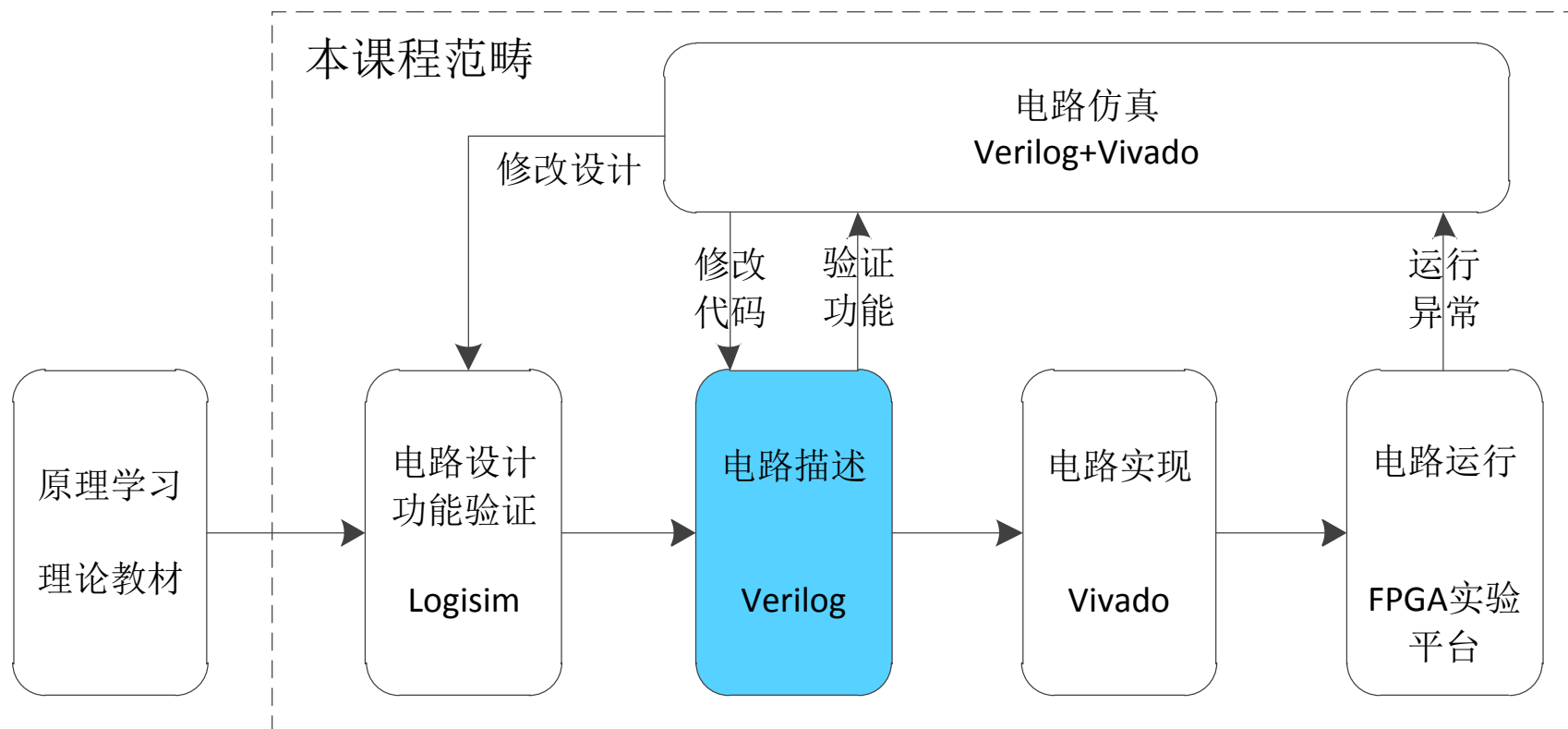
2022-10-27

# 实验安排

数字电路实验关键时间节点				
时间	实验发布（23：59前）	实验讲解（上课时间）	检查截止（21:30前）	报告提交截止（23:59前）
第四周 (09. 22)	实验01_Logisim入门			
第五周 (09. 29)	实验02_简单组合逻辑电路	实验01_Logisim入门		
第七周 (10. 13)	实验03_简单时序逻辑电路	实验02_简单组合逻辑电路	实验01_Logisim入门	
第八周 (10. 20)	实验04_Verilog硬件描述语言	实验03_简单时序逻辑电路	实验02_简单组合逻辑电路	实验01_Logisim入门
第九周 (10. 27)	实验05_使用Vivado进行仿真	实验04_Verilog硬件描述语言	实验03_简单时序逻辑电路	实验02_简单组合逻辑电路
第十周 (11. 03)	实验06_FPGA原理及Vivado综合	实验05_使用Vivado进行仿真	实验04_Verilog硬件描述语言	实验03_简单时序逻辑电路
十一周 (11. 10)	实验07_FPGA实验平台及IP核	实验06_FPGA原理及Vivado综合	实验05_使用Vivado进行仿真	实验04_Verilog硬件描述语言
十二周 (11. 17)	实验08_信号处理及有限状态机	实验07_FPGA实验平台及IP核	实验06_FPGA原理及Vivado综合	实验05_使用Vivado进行仿真
十三周 (11. 24)	实验09_竞争冒险及流水线技术	实验08_信号处理及有限状态机	实验07_FPGA实验平台及IP核	实验06_FPGA原理及Vivado综合
十四周 (12. 01)	实验10_综合实验	实验09_竞争冒险及流水线技术	实验08_信号处理及有限状态机	实验07_FPGA实验平台及IP核
十五周 (12. 08)		实验10_综合实验	实验09_竞争冒险及流水线技术	实验08_信号处理及有限状态机
十六周 (12. 15)			实验10_综合实验	实验09_竞争冒险及流水线技术
十七周 (12. 22)				实验10_综合实验

# 实验目的

- 掌握Verilog HDL常用语法
- 能够熟练阅读并理解Verilog代码
- 能够设计较复杂的数字功能电路
- 能够将Verilog代码与实际硬件相对应



# 实验环境

---

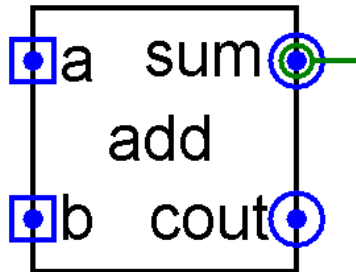
- PC一台：能流畅的连接校园网
- 远程虚拟机平台：[vlab.ustc.edu.cn](http://vlab.ustc.edu.cn)
- FPGA在线实验平台：[fpgaol.ustc.edu.cn](http://fpgaol.ustc.edu.cn)
- Verilog在线测评系统：[hdlbits.01xz.net](http://hdlbits.01xz.net)
- Logisim仿真工具

# 实验原理

## ■ Verilog HDL关键字

- Verilog语法中定义了120多个关键字，但常用的实际上很少，作为初学者只需要使用其中的十几个，便可以完成绝大多数的电路设计
- module/endmodule、input、output、wire、reg、assign、always、initial、begin/end、posedge、negedge、if、else、case/endcase

## ■ Verilog代码结构



```
module add(  
    input a, b,  
    output sum, cout);  
    assign {cout, sum} = a + b;  
endmodule
```

# 实验原理

## ■ Verilog数据及类型

- Verilog中，有四种基本的值：0,1,x,z
- Verilog中有两种常用的数据类型：wire,reg
- Verilog中常量有三种：整数、实数、字符串
- Verilog中整数有两种书写方式：十进制格式、基数格式
- 十进制格式：0,1,10,32 , -15
- 基数格式写44：
  - 6' b10\_1100 ( 二进制 )
  - 6' o54 ( 八进制 )
  - 6' d44 ( 十进制 )
  - 6' h2C ( 十六进制 )

# 实验原理

## ■ Verilog操作符

运算符分类	所含运算符
算术运算符	+、-、*、/、%
位运算符	~、&、 、^、~^或^~
缩位运算符	&、~&、 、~ 、^、~^或^~
逻辑运算符	!、&&、
关系运算符	<、>、<=、>=
相等与全等运算符	=、=、!=、=、=、=、!=、=、=
逻辑移位运算符	<<、>>
连接运算符	{}
条件运算符	? :

## ■ Verilog表达式

- Verilog中的表达式由操作数和操作符组成，可以在出现数值的任何地方使用。表达式可以是以下类型：常数、参数、线网、寄存器、位选择、部分选择、存储器单元、函数调用

*always@(posedge clk or negedge rst\_n)*

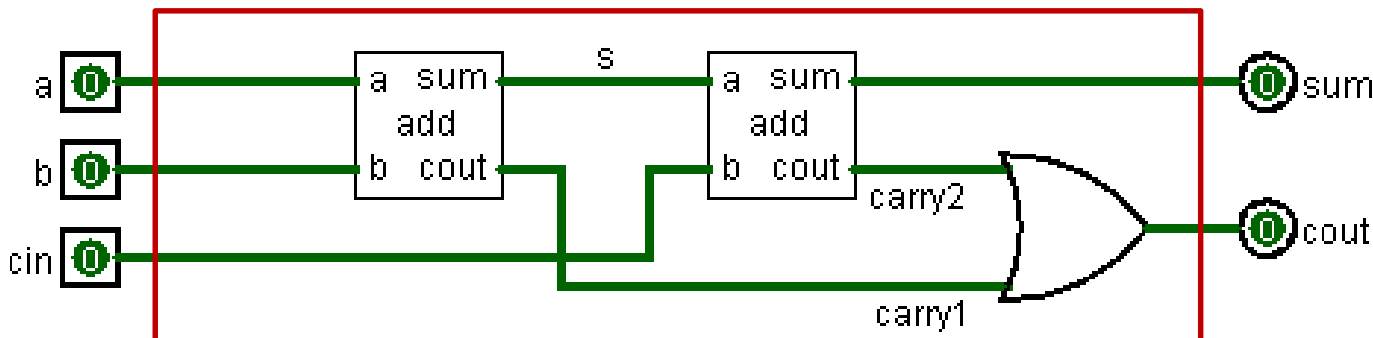
*if( cnt >= 10 )*

*assign c = a & b;*

# 实验原理

## ■ Verilog模块调用

- 一个模块能够在另一个模块中被引用，这样就建立了描述的层次，使得设计大规模复杂电路的效率大大提高。模块实例语句的形式为：模块名 实例化名（端口关联）



```
module full_add(  
    input a, b, cin,  
    output sum, cout);  
    wire s, carry1, carry2;  
    add add_inst1(a, b, s, carry1); //通过位置关联  
    add add_inst2(. a(s), . b(cin), . sum(sum), . cout(cout)); //通过名称关联
```



# 实验原理

## ■ 代码实例

```
module mux_4to1(  
    input [7:0] a,b,c,d,  
    input [1:0] sel,  
    output reg [7:0] o);  
    always @(*)  
    begin  
        case(sel)  
            2'b00: o = a;  
            2'b01: o = b;  
            2'b10: o = c;  
            2'b11: o = d;  
            default: o = 8'h0;  
        endcase  
    end  
endmodule
```

*//8bit位宽的4选1选择器*

*//always语句内赋值的信号都应定义成reg类型*  
*//always语句内实现组合逻辑*

*//组合逻辑使用“=”进行赋值*

*//实现组合逻辑时一定要有default*

# 实验原理

## ■ 代码实例

```
module cnt_1to10(  
input clk,rst_n,  
output reg [3:0] cnt);  
always @(posedge clk or negedge rst_n)//时序控制条件为时钟上升沿和复位下降沿  
begin  
    if(!rst_n)                                //复位信号优先级最高，应是第一个判断的条件  
        cnt <= 4'h1;  
    else if(cnt>=10)  
        cnt      <= 4'h1;  
    else  
        cnt <= cnt + 4'h1;  
end  
endmodule
```

# 实验内容

---

- 登录[hdlbits.01xz.net](http://hdlbits.01xz.net)练习Verilog（选做）
- 登录[verilogoj.ustc.edu.cn](http://verilogoj.ustc.edu.cn)练习Verilog（选做）
- 完成实验指导书中的所有题目，两周内提交
- 撰写实验报告，三周内提交
  - ~~实验报告中应体现实验过程中的关键环节和最终结果~~
  - 填写实验练习部分的答案（代码、截图或结果）
  - 对于本次实验的总结思考、回答实验练习中提到的问题

---

**谢谢！**