



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

# 数字电路实验

## Lab1 Verilog 语法

2024-9-26

# 实验介绍



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

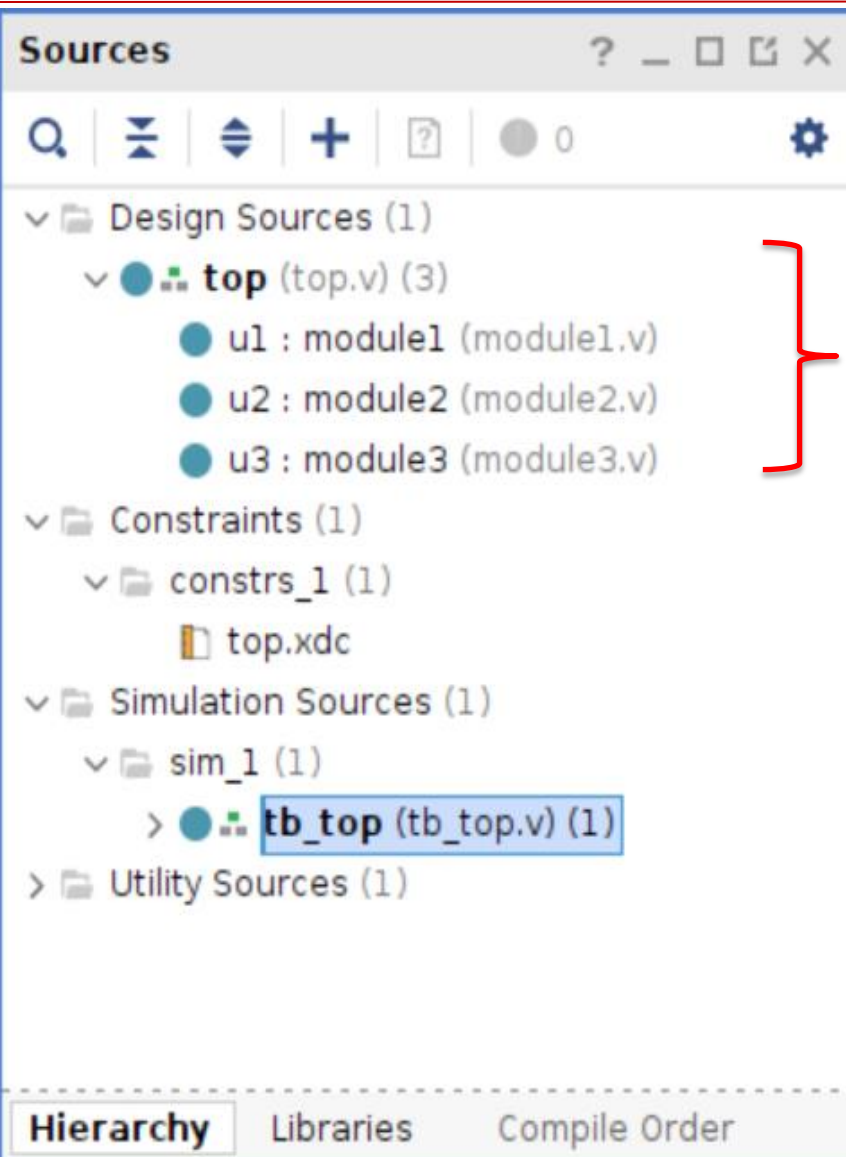
- 在本次实验中，我们主要学习 **Verilog 语言** 的基本知识，并进行简单的 **Verilog 编程** 工作。



# Vivado工程结构



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



设计文件：由多个模块组成

约束文件：管脚分配及时序约束

仿真文件：Lab2

## ■ 模块定义

```
module module_name  
#(parameter_list 参数声明)  
(  
    端口声明  
);  
    变量声明;  
    逻辑功能描述, 包括  
    模块实例化、assign 语句  
    、always 语句  
endmodule
```

示例: MUX2

```
module mux2//模块名: mux2  
#(parameter MSB = 31, //参数声明: 数据最高有效位  
    LSB = 0    //数据最低有效位  
)  
(output [MSB : LSB] y, //端口声明: 输出数据  
    input [MSB : LSB] a, b,    //两路输入数据  
    input s    //数据选择控制  
);  
assign y = s? b : a;    //逻辑功能描述  
endmodule
```

## ■ 模块实例化

`module_name #(parameter_map) instance_name (port_map);`

### ■ 端口映射方式：基于位置或者基于名字，不可混合使用

#### ■ 位置映射：按模块中端口定义的顺序传递--**不推荐**

例如：模块定义为 `module M(A, B, C);`

`M M1(Y, Z, X);` //顺序很重要,

#### ■ 名字映射：`.PortName (value)` --**推荐**

`M M1(.B(Z), .C(X), .A(Y));` // 顺序无关

### ■ 参数的映射方式与端口映射方式类似

## ■ Verilog 的三种描述层次

### ■ 结构化描述

调用基本逻辑元件（逻辑门），描述它们之间的连接关系建立逻辑电路。

### ■ 数据流描述

根据变量之间的逻辑关系，采用连续赋值语句 `assign` 描述电路功能。类似传统意义的逻辑表达式。

### ■ 行为级描述

对系统的高抽象级别描述，注重整个系统的行为特性：在特定输入条件下产生何种输出。

## ■ Verilog 与 C 语言类似的地方：

- 区分大小写、相同的注释格式、分号结尾

## ■ Verilog 与 C 语言不同的地方：

- 数值类型、变量类型；
- 运算符；
- 赋值语句；
- 分支语句；
- 模块与函数；
- .....

## ■ Verilog 数值种类

- 0: 表示低电平或 False。
- 1: 表示高电平或 True。
- z: 表示高阻，电路断路就会显示为高阻。
- x: 表示电平不确定。

z或者x属于仿真阶段常见的波形异常类错误。

信号为z的错误原因:

1) RTL里声明为wire型的变量从未被赋值。2) 模块调用的信号未连接导致信号悬空。

信号为x的错误原因:

1) RTL里声明为reg型的变量从未被赋值。2) RTL里信号多驱动。

## ■ 整数数值表示法

<位宽>'<进制><数值>

例如: 2'b01, 4'd10, 8'b0010\_1011, 6'h3f



## ■ Verilog 变量类型

### ■ wire- 组合逻辑电路

线网数据类型，对应着真实的导线。用于声明在 assign 语句内赋值的变量。Verilog 的默认数据类型。

### ■ reg - 组合或时序逻辑电路

(广义) 寄存器数据类型，有时对应真实的导线，有时对应寄存器。用于声明在 always 语句内赋值的变量。

#### ➤ 组合逻辑电路

任意时刻，电路的输出仅仅取决于此时刻的输入，与其原本的状态无关（无记忆性）。

#### ➤ 时序逻辑电路

任意时刻，电路的输出不仅取决于此时刻的输入，而且还和其原来的状态有关（有记忆性）。

# Verilog 语法



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

## Verilog 运算符

优 先 级 别	
<div>! ~ * / % + - &lt;&lt; &gt;&gt; &lt; &lt;= &gt; &gt;= == != === !== &amp; &amp; ^ ~   &amp;&amp;    ?:</div>	<div>高 优 先 级 别</div> <div>↓</div> <div>低 优 先 级 别</div>

从高到低依次为：

逻辑非、按位非

乘法、除法、取模

加法、减法

逻辑左移、逻辑右移

不等关系运算符

相等关系运算符

按位与

按位异或、按位同或

按位或

逻辑与

逻辑或

条件运算符

# Verilog 语法



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

## ■ Verilog 三种赋值语句，两种过程块

### ■ assign 连续赋值语句

assign 变量 = 表达式;

变量wire类型；综合生成组合逻辑；输出立即可见。

### ■ always 过程“阻塞”赋值语句

always @(\*)begin

变量 = 表达式; end

组合过程块：变量reg类型；综合生成组合逻辑；输出立即可见。

### ■ always 过程“非阻塞”赋值语句

always @(posedge clk or posedge rst)begin

变量 <= 表达式; end

时序过程块：变量reg类型；综合生成组合逻辑+寄存器；输出在下一个时钟边沿发生变化。

## ■ Verilog 阻塞赋值和非阻塞赋值

### ■ 阻塞赋值 =

在同一个always中，所有语句之间是一种**串行（顺序）关系**。一条阻塞赋值语句如果没有执行结束，那么该语句后面的语句就不能被执行，即被“阻塞”。

### ■ 非阻塞赋值 <=

在同一个always中，所有语句之间是一种**并行关系**。非阻塞赋值在时钟边沿到来之前，执行赋值语句右边；时钟边沿到来时，所有的赋值语句**同时**赋值到赋值语句的左边，且一个时钟只执行一次，属于并行执行语句。

## ■ Verilog 分支语句-用于always块中

带有优先级的条件分支: if/else 语句

```
always @(*)begin
    if (condition1)begin
        .....
    end
    else if (cobddition2)begin
        .....
    end
    else if (cobddition3)begin
        .....
    end
    else begin
        .....
    end
end
```

if else 成对出现, 避免形成死锁。

无优先级的条件分支: case 语句

```
always @(*)begin
    case(in)
        <case1>:begin .....end(语句>1)
        <case2>: .....
        default : .....
    endcase
end
```

## ■ Verilog约束文件

```
set_property -dict {PACKAGE_PIN V18 IOSTANDARD LVCMOS33} [get_ports clk_100M]  
create_clock -period 10.000 -name sys_clk_pin -waveform {0.000 5.000} -add [get_ports  
clk_100M]---时序约束
```

### ## LEDs

```
set_property -dict {PACKAGE_PIN B15 IOSTANDARD LVCMOS33} [get_ports {led[0]}]  
set_property -dict {PACKAGE_PIN B16 IOSTANDARD LVCMOS33} [get_ports {led[1]}]  
set_property -dict {PACKAGE_PIN C13 IOSTANDARD LVCMOS33} [get_ports {led[2]}]  
set_property -dict {PACKAGE_PIN B13 IOSTANDARD LVCMOS33} [get_ports {led[3]}]  
set_property -dict { PACKAGE_PIN A15  IOSTANDARD LVCMOS33 } [get_ports {led[4]}]  
set_property -dict { PACKAGE_PIN A16  IOSTANDARD LVCMOS33 } [get_ports {led[5]}]  
set_property -dict { PACKAGE_PIN A13  IOSTANDARD LVCMOS33 } [get_ports {led[6]}]  
set_property -dict { PACKAGE_PIN A14  IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
```

# Verilog 代码示例



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

```
1  module MUX2(  
2      input      a, b,  
3      input      sel,  
4      output reg  out  
5  );  
6  always @(*) begin  
7      if (!sel)  
8          out = a;  
9      else  
10         out = b;  
11  end  
12  endmodule
```

**1~5 行：模块声明**

**2：输入端口 a、b，wire 型**

**3：输入端口 sel，wire 型**

**4：输出端口 out，reg 型**

**6-11 行：过程赋值 always 语句块**

**7：if 语句与判断条件**

**8：真分支，阻塞赋值**

**9：else 语句**

**10：假分支，阻塞赋值**

**12 行：模块结束**

**这段代码实现了什么功能？**

# Verilog 代码示例



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

## 一些编程小提示:

- if 后面要有 else;
- case 不要忘记 default;
- 不要忘记 endcase 和 endmodule;
- 变量名要有意义;
- 模块中assign和always块都是并行执行的, 不要在多个并行执行体中对同一变量赋值;
- 不要尝试直接用其他信号边沿:  
always@ (posedge en)





# 实验任务



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

- [必做] 学习 Verilog 语言的基础语法知识;
- [必做] 按要求完成 Lab1 的实验练习题;
- [选做] 配置本地的 Verilog 开发环境。

---

**谢谢！**