



Verilog HDL语言设计规范

主讲：何宾

Email: hebin@mail.buct.edu.cn

2014.06

Verilog HDL描述方式

模块内具体逻辑行为的描述方式又称为建模方式。根据设计的要求，每个模块内部具体的逻辑行为描述方式可以分为四个不同的抽象级别。

- 对于外部来说，看不到逻辑行为的具体实现方式。因此，模块的内部具体逻辑行为描述相对于外部其它模块来说是不可见的。
- 改变一个模块内部逻辑行为的描述方式，并不会影响该模块与其它模块的连接关系。

Verilog HDL描述方式

Verilog HDL提供了下面四种方式描述具体的逻辑行为：

- 行为级描述
- 数据流描述
- 结构级描述
- 开关级描述

Verilog HDL描述方式

--行为级描述

Verilog HDL的行为级描述是最能体现电子设计自动化风格的硬件描述方式

- 它既可以描述简单的逻辑门，也可以描述复杂的数字系统乃至微处理器。
- 既可以描述组合逻辑电路，也可以描述时序逻辑电路。因此，它是Verilog HDL最高抽象级别的描述方式。
- 可以按照要求的设计算法来实现一个模块，而不用关心该模块具体硬件实现的细节。
- 这种抽象级别描述方式非常类似C编程。

Verilog HDL描述方式

--行为级描述

行为级描述只能用于对设计进行仿真，而不能用于对设计进行综合。

这是因为：行为级描述中包含了不可综合的描述语句。不可综合的语句只能用于仿真，不可用于综合。

□ 逻辑行为的描述是通过行为描述语句来实现的。可使用下述过程语句结构描述行为功能。

➤ **initial**语句

该语句只执行一次，通常用于初始化变量。

➤ **always**语句

该语句循环执行若干次。

Verilog HDL描述方式

--行为级描述

在这两种语句中，只有寄存器类型的数据能够被复制。在被赋新值前，寄存器型数据保持原有值不变。

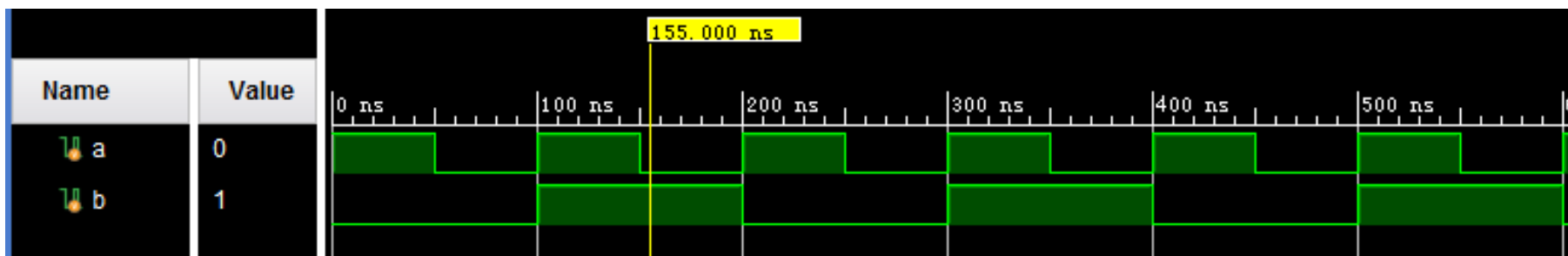
所有的initial语句和always语句在零时刻并行执行。

Verilog HDL描述方式

--行为级描述的例子

```
module behave;  
  reg a, b;  
  initial begin  
    a = 'b1;  
    b = 'b0;  
  end
```

```
    always begin  
      #50 a = ~a;  
    end  
    always begin  
      #100 b = ~b;  
    end  
endmodule
```



本设计保存在本书配套资料\eda_verilog\behav_des目录下

Verilog HDL描述方式

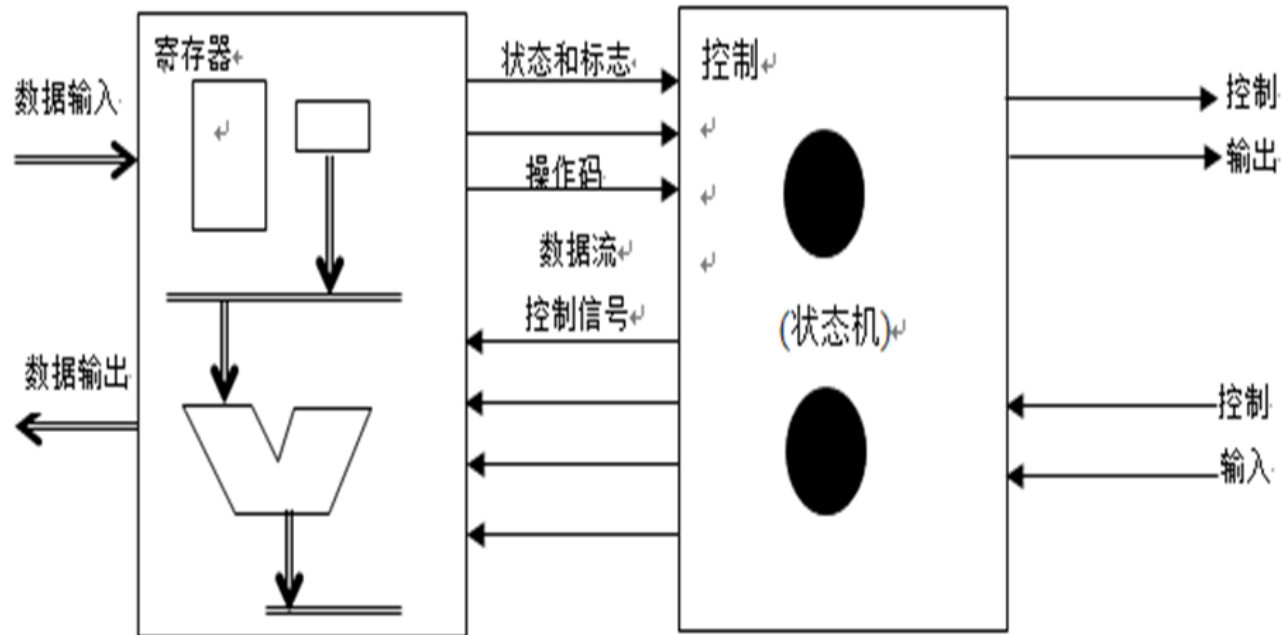
--数据流描述方式

数据流描述方式，也称寄存器传输级（ Register Transfer , RTL ）描述方式。

- 所谓的数据流描述可以这样理解，即：在一个复杂的数字系统中，应该包含有数据流和控制流。
- 控制流用于控制数据的“流向”。
 - ✓ 即，数据将要到达的地方。

Verilog HDL描述方式

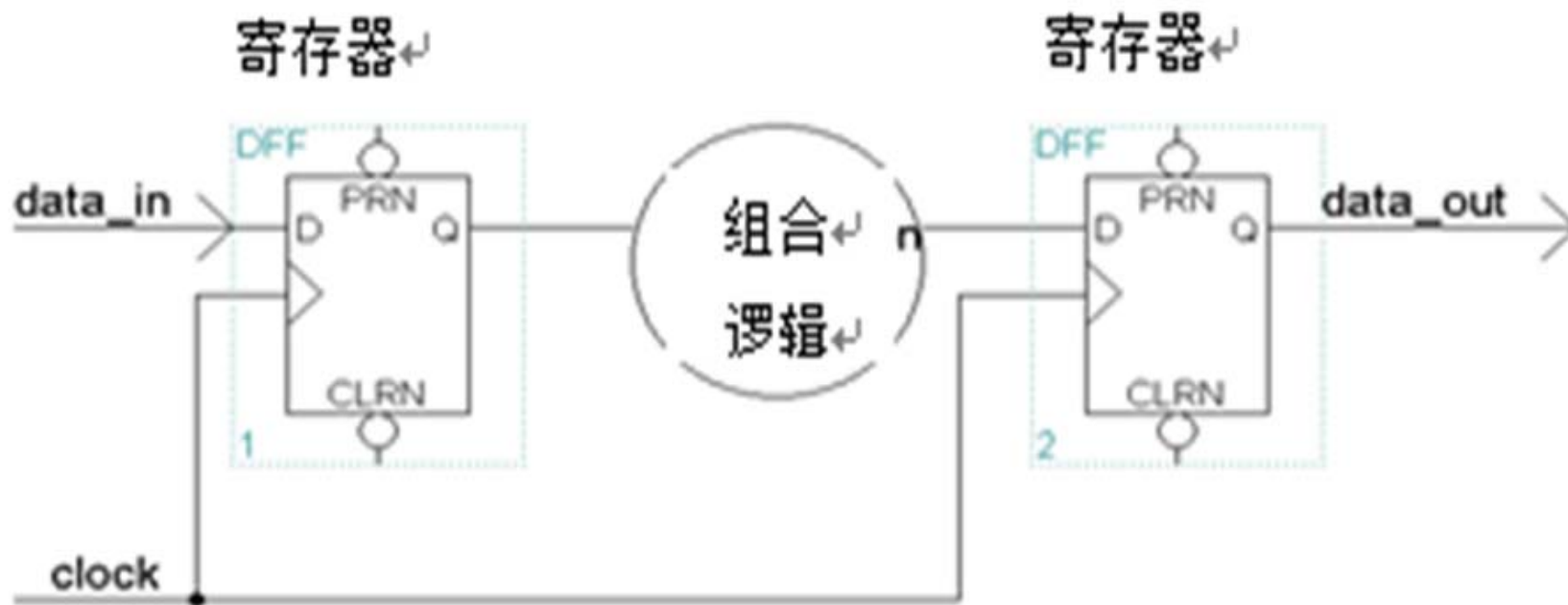
--数据流描述方式



Verilog HDL描述方式

--数据流描述方式

从寄存器传输级的角度，可以这样理解，即：在寄存器之间插入组合逻辑电路。



Verilog HDL描述方式

--数据流描述方式

在一个复杂的数字系统中，任何数据从输入到输出，都需要经过寄存器，寄存器用于重定时和记忆。

□ 这样，就能保证数据从输入到输出满足时序收敛条件，不会出现竞争冒险和亚稳定状态。

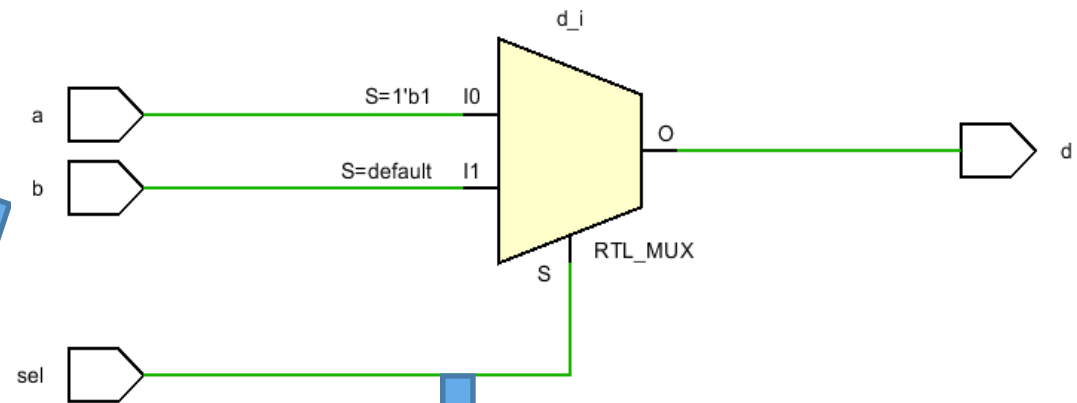
Verilog HDL描述方式

--数据流描述方式（2选1多路选择器）

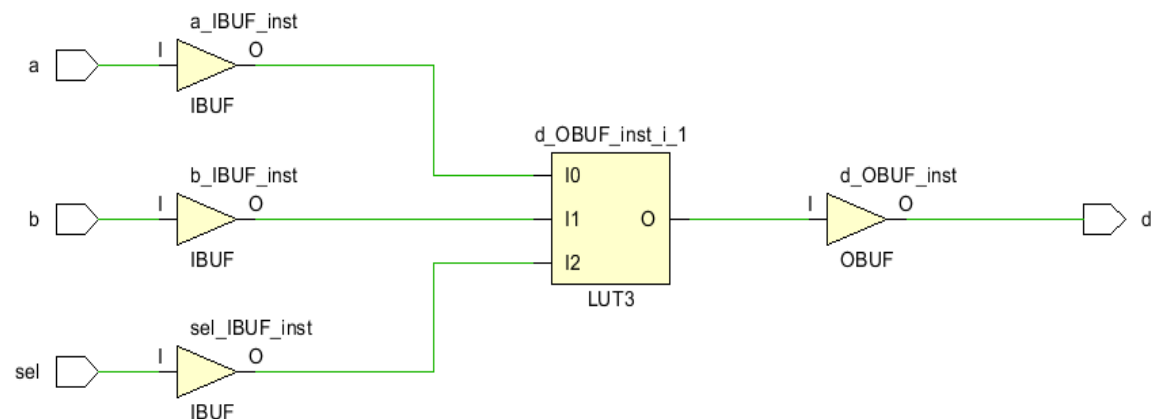
```
module mux2_1(
  input sel,
  input a,
  input b,
  output reg d
);
```

```
  always @(sel,a,b)
  begin
    if(sel)
      d=a;
    else
      d=b;
  end
endmodule
```

详细描述电路



综合后的电路



Cell Properties

d_OBUF_inst_i_1

I2	I1	I0	O=I1 & !I2 + I0 & I2
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0

Verilog HDL描述方式

--数据流描述方式

讲完数据流描述方式，经常有人就会问行为级描述方式到底和数据流描述方式的本质区别是什么？

□在行为级描述中，使用了Verilog HDL中提供的一些描述语句，这些语句无法通过Vivado综合转换为电路结构；

■例如包含对时间延迟的描述；

□而数据流/RTL描述中所使用的Verilog HDL描述语句，是可以通过Vivado综合后转换为电路结构。

Verilog HDL描述方式

--数据流描述方式

- 行为级描述，一般只用于对设计进行仿真，也就是生成对设计的测试向量，通过仿真工具来验证设计有无缺陷；
- 而数据流/RTL描述最终要转换为电路。

Verilog HDL描述方式

--结构级描述方式

结构描述就是在设计中，通过调用库中的元件或者是已经设计好的模块来完成设计实体功能的描述。

通常情况下，在使用层次化设计时，一个高层次模块会调用一个或者多个低层次模块。这种模块的调用是通过模块例化语句实现的。

Verilog HDL描述方式

--结构级描述方式

模块例化语句的基本格式如下：

```
<module_name> <list_of_variable> <module_example_name>  
( <list_of_port> ) ;
```

其中：

- `module_name`是指被调用模块指定的模块名。
- `list_of_variable`是可选项，它是由一些参数值组成的一个有序列表，将这些参数值传递给被调用模块实例内的各个参数。

Verilog HDL描述方式

--结构级描述方式

- `module_example_name` 是所生成的模块实例所命名的一个名字，它是被调用模块实例的唯一标识。
- `list_of_port` 是由外部信号信号组成的一个有序列表，这些外部信号端口表示与模块实例各个端口的连接。所以，<端口连接表>指明了模块实例端口与外部电路的连接情况。

注：在Verilog HDL中提供了两种方法用于端口信号的连接。

可以按照端口列表的顺序进行端口的映射，也可以通过端口的名字进行映射。

Verilog HDL描述方式

--结构级描述方式

模块例化规则

- 在某一模块内，可以多次调用同一模块。但是，每次调用生成的模块实例名不能重复。
- 实例名和模块名的区别是：
 - 模块名表示不同的模块，即用来区分电路单元的不同种类。
 - 而实例名则表示不同的模块实例，用来区分电路系统中的不同硬件电路单元。

结构级描述方式的例子

--采用位置关联（1）

// 低层次模块，一个与非门触发电路的模块描述

```
module ffnand (q, qbar, preset, clear);
```

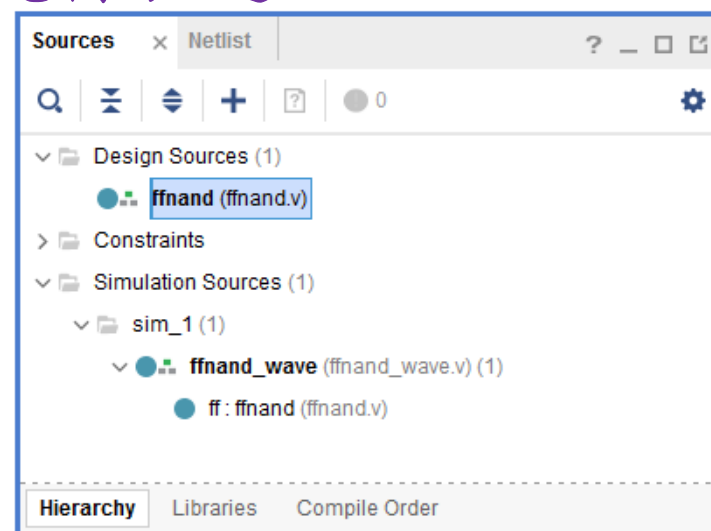
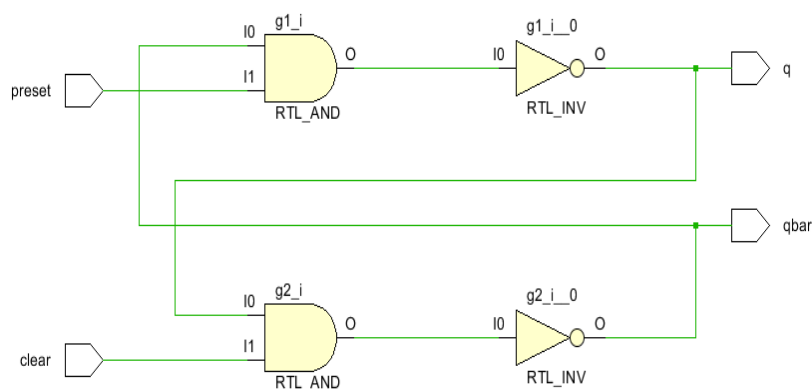
```
output q, qbar;           //声明两个电路输出网络
```

```
input preset, clear;      //声明两个电路输入网络
```

```
nand g1 (q, qbar, preset), //声明nand门，以及它们的互连
```

```
    g2 (qbar, q, clear); //声明nand门，以及它们的互连
```

```
endmodule
```



该设计保存在本书配套资源\eda_verilog\arh_des目录下

结构级描述方式的例子

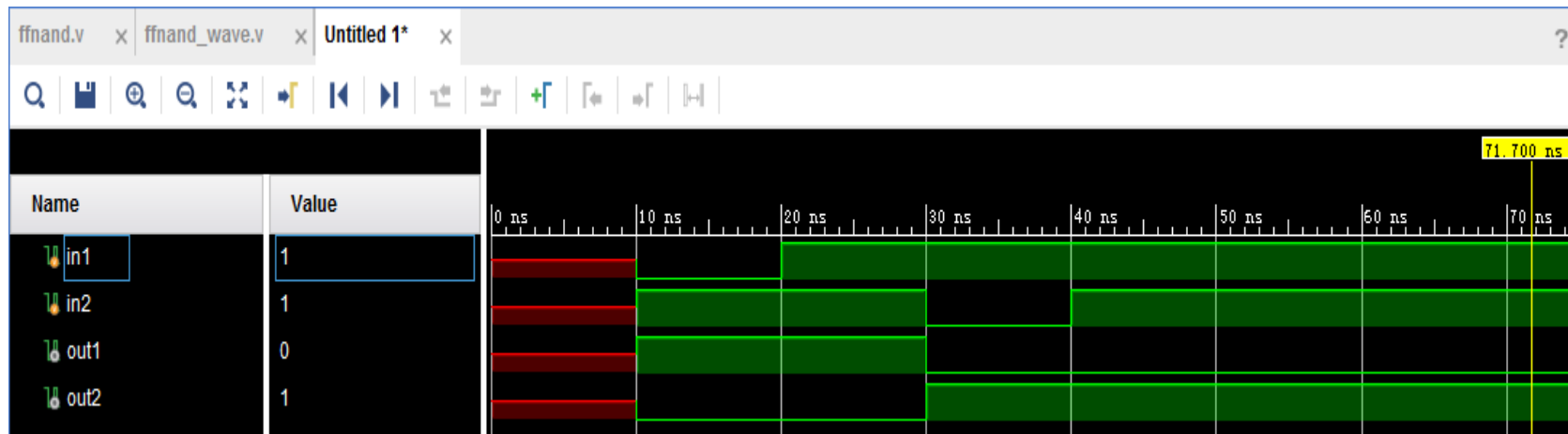
--测试文件（1）

```
module ffnand_wave;
wire out1, out2;      //来自电路的输出
reg in1, in2;          //到输出的驱动变量
parameter d = 10;
ffnand ff(out1, out2, in1, in2); //例化电路ffnand,并将其命名为"ff"
//定义用于仿真电路的波形
initial begin
    #d in1 = 0; in2 = 1;
    #d in1 = 1;
    #d in2 = 0;
    #d in2 = 1;
end
endmodule
```

该设计保存在本书配套资源\eda_verilog\arh_des目录下

结构级描述方式的例子

--测试结果（1）



思考与练习：请画出生成的ffnand的结构。并分析其测试原理。

结构级描述方式的例子

--测试文件（1）

```
module ffnand_wave;
reg in1, in2;    //驱动电路的两个变量
parameter d = 10;
//例化ffnand模块两次
//ff1的qbar没有连接, ff2 的q没有连接
    ffnand ff1(out1, , in1, in2),
    ff2(.qbar(out2), .clear(in2), .preset(in1), .q());
//定义仿真电路的波形
initial begin
    #d in1 = 0; in2 = 1;
    #d in1 = 1;
    #d in2 = 0;
    #d in2 = 1;
end
endmodule
```


Verilog HDL描述方式

--开关级描述方式

从本质上来说，开关级属于结构化描述方式，但是其描述更接近于底层的门级和开关级电路。

- 突出说明开关级描述方式，是为了说明Verilog HDL对底层强大的描述功能。
- 所调用的底层模块是由Verilog HDL库本身提供的，这也是Verilog HDL的优势，就是对底层描述的能力比较强。

Verilog HDL描述方式

--开关级描述方式

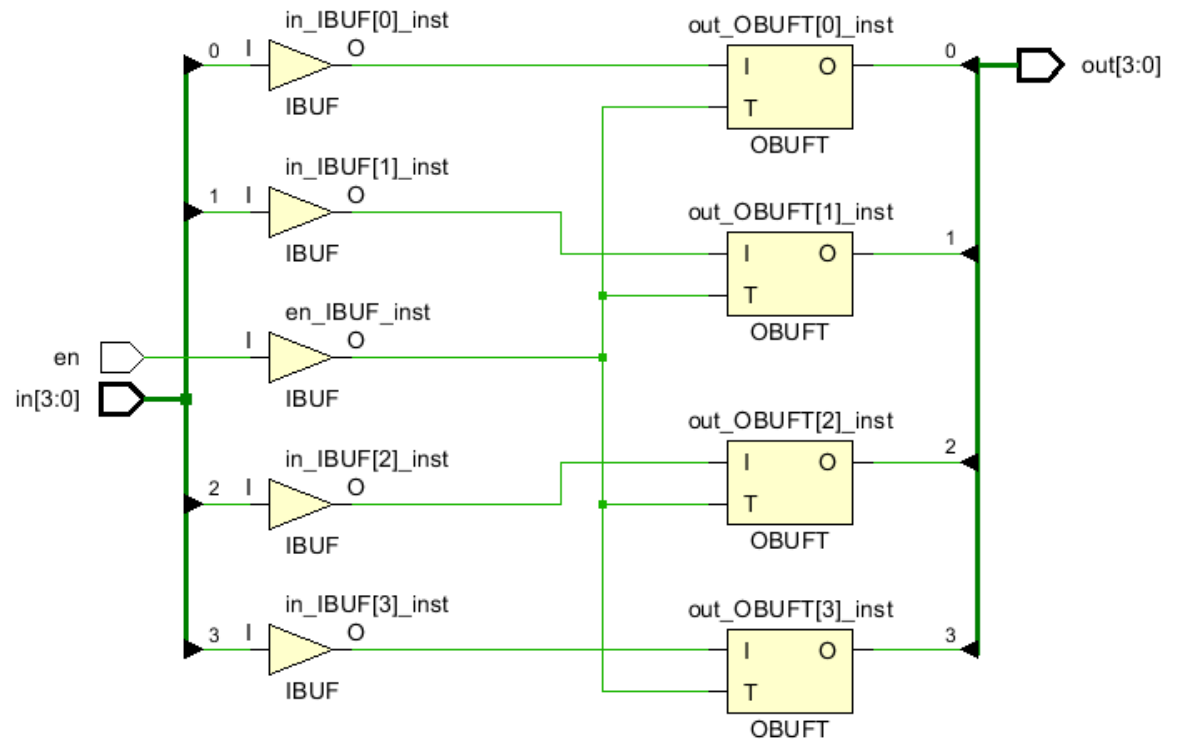
对于一个门或者开关例化来说，包含下面的描述：

- 关键字命名门或者开关原语的类型。
- 可选的驱动强度。
- 可选的传播延迟。
- 可选的标识符，命名了每个门或者开关例化的名字。
- 可选的用于例化阵列的范围。
- 终端连接列表。

Verilog HDL描述方式

--开关级描述方式的例子

```
module driver (in, out, en);  
input [3:0] in;  
output [3:0] out;  
input en;  
    // 三态缓冲器阵列  
    bufif0 ar[3:0] (out, in, en);  
endmodule
```



该设计保存在本书配套资源\eda_verilog\switch_des目录下

Verilog HDL描述方式

--开关级描述方式的例子

```
module driver_equiv (in, out, en);  
input [3:0] in;  
output [3:0] out;  
input en;  
    bufif0 ar3 (out[3], in[3], en); // 独立声明每个缓冲区  
    bufif0 ar2 (out[2], in[2], en);  
    bufif0 ar1 (out[1], in[1], en);  
    bufif0 ar0 (out[0], in[0], en);  
endmodule
```