

10111110

## Verilog HDL语言规范

主讲:何宾

Email: hebin@mail.buct.edu.cn

2014.06

1, 0100111110001

## Verilog HDL门级和开关级描述

这部分介绍Verilog HDL语言提供的内置门级和开关级电路建模原语,以及一个硬件设计中如何使用这些原语。

- □ Verilog HDL预定义了14个逻辑门和12个开关,用于提供门级和开关级电路建模工具。使用门级和开关级建模的优势包括:
  - □ 在真实门电路和模型之间,门提供了接近于一对一的映射。
  - □ 没有连续的分配,这等效于双向的传输门。

#### 对一个门或者开关的例化声明应该包含下面的规范:

- □ 关键字,用于命名开关或者原语的类型
- □ 一个可选的驱动强度
- □ 一个可选的传输延迟
- □ 一个可选的标识符,用于命名门或者例化开关
- □ 一个可选的例化数组的范围
- □ 终端连接列表

#### 下表给出了Verilog HDL提供的内建门和开关的列表

n输入门	n输出门	三态门	pull	MOS开关	双向开关
and	buf	bufif0	pulldown	cmos	rtran
nand	not	bufif1	pullup	nmos	rtranif0
nor		notif0		pmos	rtranif0
or		notif1		rcmos	tran
xnor				rnmos	tranifo
xor				rpmos	tranif1

#### 驱动强度描述

□指定了门例化输出终端逻辑值的强度。

#### 下表给出了可以使用驱动强度描述的门类型

and	nand	buf	not	pulldown
or	nor	bufif0	notif0	pullup
xor	xnor	bufif1	notif1	

用于一个门例化的驱动强度说明,除了pullup和pulldown以外,应该有strength1说明和strength0说明。

- ◆ strength1说明指定了逻辑1的信号强度;
- ◆ strength0说明指定了逻辑0的信号强度。

驱动强度跟在门类型关键字后,在延迟说明的前面。

- □ strength0说明可以在strength1说明之后,也可以在其之前。
- □ 在圆括号内,通过逗号,将strength0说明和strength1说明隔开。
- ◆ pullup门只有strength1说明; strength0说明是可选的。
- ◆ pulldown门只有strength0说明; strength1说明是可选的。

注:

■ strength1描述包含下面的关键字:

supply1 strong1 pull1 weak1

■ strength0描述包含下面的关键字:

supply0 strong0 pull0 weak0

■ 将strength1指定为highz1,将引起门或者开关输出一个逻辑值z,而不是1;将strength0指定为highz0,将引起门或者开关输出一个逻辑值z,而不是0。强度说明(highz0, highz1)和 (highz1, highz0)是无效的。

下面给出了一个集电极开路nor门的Verilog HDL描述的例子。

nor (highz1,strong0) n1(out1,in1,in2);

在这个例子中, nor逻辑门输出z, 而不是1。

在一个声明中,可选的延迟说明指定了贯穿门和开关的传播延迟。

- □ 如果在声明中,没有指定门和开关延迟描述,则没有传播延迟。
- □ 根据门的类型,一个延迟说明最多包含三个延迟值。
- □ pullup和pulldown例化声明,将不包含延迟说明。

#### 原语例化标识符

- □ 可以为一个门或者开关例化指定一个可选的名字。
- □ 如果声明了多个例化作为一组例化,则需要使用一个标识符来命名例化。

范围说明,当要求重复例化的时候,这些例化之间是不同的。通过向量索引的连接来区分它们。

- □ 为了指定一个例化数组,例化的名字后面应该跟着范围,使用 两个常数表达式指定范围,左侧索引(lhi)和右侧索引(rhi)。
- □ 它们通过 "[]" 符号中的 ":"符号隔开。范围[lhi: rhi], 表示abs(lhi-rhi)+1宽度的例化数组。

注:一个例化数组的范围应该是连续的。一个例化标识符只关联一个范围,用于声明例化数组。

#### 下面的声明是非法的:

```
nand #2 t_nand[0:3] ( ... ), t_nand[4:7] ( ... );
```

#### 例化数组声明的Verilog HDL描述例子

```
nand #2 t_nand[0:7]( ... );
nand #2 x_nand[0:3] ( ... ), y_nand[4:7] ( ... );
```

终端列表描述了门或者开关如何连接模型的剩余部分。

- □门和开关的类型限定了表达式。
- □ 连接列表通过()括起来,()号内的终端通过","符号进行分割。
- □ 输出或者双向终端总是出现在连接列表的开始,后面跟着输入。

nand #2 nand\_array[1:4]( ... );

声明了四个例化,作为nand\_array[1]、nand\_array[2]、nand\_array[3]和nand\_array[4]标识符进行引用。

#### 两个等效门例化Verilog HDL描述的例子1

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

```
module driver_equiv (in, out, en);
input [3:0] in;
output [3:0] out;
input en;
bufif0 ar3 (out[3], in[3], en); // 每一个单独的声明
bufif0 ar2 (out[2], in[2], en);
bufif0 ar1 (out[1], in[1], en);
bufif0 ar0 (out[0], in[0], en);
endmodule
```

#### 两个等效门例化Verilog HDL描述的例子2

```
module busdriver (busin, bushigh, buslow, enh, enl); input [15:0] busin; output [7:0] bushigh, buslow; input enh, enl; driver busar3 (busin[15:12], bushigh[7:4], enh); driver busar2 (busin[11:8], bushigh[3:0], enh); driver busar1 (busin[7:4], buslow[7:4], enl); driver busar0 (busin[3:0], buslow[3:0], enl); endmodule
```

```
module busdriver_equiv (busin, bushigh, buslow, enh, enl);
input [15:0] busin;
output [7:0] bushigh, buslow;
input enh, enl;
driver busar[3:0] (.out({bushigh, buslow}), .in(busin),
.en({enh, enh, enl, enl}));
endmodule
```

#### 思考题:给出下面Verilog HDL描述所给出的门级设计结构

```
module dffn (q, d, clk);
parameter bits = 1;
input [bits-1:0] d;
output [bits-1:0] q;
input clk;
DFF dff[bits-1:0] (q, d, clk);
endmodule
```

```
module MxN_pipeline (in, out, clk);
parameter M = 3, N = 4;
input [M-1:0] in;
output [M-1:0] out;
input clk;
wire [M*(N-1):1] t;
dffn #(M) p[1:N] ({out, t}, {t, in}, clk);
endmodule
```

#### 逻辑门的例化,使用下面关键字:

and nand nor or xor xnor

#### 下表给出了多个逻辑输入门的真值表

#### 注:

这六种类型的逻辑门,有多个输入,但只有一个输出。终端列表的第一个终端将连接到门的输出;其它终端将连接到输入。

and	0	1	X	Z	or	0	1	X	Z
0	0	0	0	0	0	0	1	X	X
1	0	1	X	X	1	1	1	1	1
X	0	X	X	X	X	X	1	X	X
Z	0	X	X	X	Z	X	1	X	X
nand	0	1	X	Z	nor	0	1	X	Z
0	1	1	1	1	0	1	0	X	X
1	1	0	X	X	1	0	0	0	0
X	1	X	X	X	X	X	0	X	X
Z	1	X	X	X	Z	X	0	X	X
xor	0	1	X	Z	xnor	0	1	X	Z
0	0	1	X	X	0	1	0	X	X
1	1	0	X	X	1	0	1	X	X
X	X	X	X	X	X	X	X	X	X
Z	X	X	X	X	Z	X	X	X	X

门级逻辑设计描述中可使用具体的门例化语句。简单 的门实例语句的语法格式如下:

gate\_type[instance\_name](term1, term2, . . . ,termN);

#### 其中:

- gate\_ type为前面所列出门的关键字。
- instance\_name为可选的例化标识符。
- term1,..,termN用于表示与门的输入/输出端口相连的网络。

#### 延迟描述应该是0个、1个或者两个延迟。

- 如果描述中包含两个延迟,则:
  - 第一个延迟确定输出上升延迟;
  - 第二个延迟确定输出下降延迟。
  - > 两个延迟中较小的一个应用于输出跳变为x的延迟。
- 如果只有一个延迟,将应用到上升和下降延迟。
- 如果没有指定则没有传播延迟。

#### 两输入与门Verilog HDL描述的例子

and a1(out,in1,in2);

#### 其中:

- □ 该与门例化标识符为a1。
- □ 输出为out。
- □ 带有两个终端输入in1和in2。

#### 多输出逻辑门的关键字:

□ buf

not

其延迟特性同逻辑门的延迟描述。

#### 下表给出了输出门的真值表。

bı	uf	not			
输入	输出	输入	输出		
0	0	0	1		
1	1	1	0		
X	X	X	X		
Z	X	Z	X		

这些门只有一个输入,一个/多个输出。输出门的实例语

句的基本语法格式如下:

multiple\_output\_gate\_type [instance\_name](out1,out2,...,outn, inputA);

#### 其中:

- □ multiple\_output\_gate\_type为输出门的关键字。
- □ instance\_name为可选的例化标识符。
- □ out1,out2,...,outN,inputA为输出/输入端口。只有inputA是输入端口,其余的所有端口为输出端口。

#### 多输出门的Verilog HDL描述

buf b1(out1,out2,in);

该门实例语句中,in是缓冲门的输入,多输出门b1有两个输出:out1和out2。

- 三态逻辑门的例化,使用下面的关键字:
- □ bufif0
- bufif1
- □ notif1
- Notif0

其延迟特性同逻辑门的延迟描述。这些门除了逻辑1和逻辑0外,还可以输出z。

这些门用于对三态驱动器建模。这些门有一个数据输出、 一个数据输入和一个控制输入。下表给出了三态逻辑门的 真值表。

#### 注:

- 符号L,表示结果为0或z。
- 符号H,表示结果为1或z。
- 跳变到H或者L的延迟与跳变到x的延迟是一样的。

bufif0	控制				bufif1	控制					
		0	1	X	Z			0	1	X	Z
	0	0	Z	L	L		0	Z	0	L	L
数	1	1	Z	H	H	数	1	Z	1	H	H
据	X	X	Z	X	X	据	X	Z	X	X	X
	Z	X	Z	X	X		Z	Z	X	X	X
notif0	控制				notif1	控制					
		0	1	X	Z			0	1	X	Z
	0	1	Z	Н	H		0	Z	1	Н	Н
数	1	0	Z	L	L	数	1	Z	0	L	L
据	X	X	Z	X	X	据	X	Z	X	X	X
	Z	X	Z	X	X		Z	Z	X	X	X

#### 三态门例化语句的基本语法格式如下:

tristate\_gate [instance\_name] (outputA,inputB,control);

#### 其中:

- tristate\_gate为三态门的关键字。
- instance\_name为可选的例化标识符。
- output A是输出端口。
- input B是数据输入端口。
- control是控制输入端口。根据控制输入的值,可将输出驱动到高阻状态,即值z。

#### 三态门的延迟描述应该是0个、1个、2个或者3个延迟。

- 如果描述中包含三个延迟,则第一个延迟确定输出上升延迟,第二个延迟确定输出下降延迟,第三个延迟确定跳变到x的延迟。
- 如果描述中包含两个延迟,则第一个延迟确定输出上升延迟,第二个延迟确定输出下降延迟,两个延迟中较小的一个延迟用于确定跳变到x和z的延迟。
- 如果只有一个延迟,将应用到所有的输出跳变延迟。
- 如果没有指定延迟,则门没有传播延迟。

#### 三态门bufif1的Verilog HDL描述

bufif1 BF1 (Dbus,MemData,Strobe );

#### 在该门例化语句中:

- □ 当Strobe为0时, bufif1驱动输出Dbus为高阻。
- □ 否则当Strobe为1时,将MemData传输到Dbus。

# Verilog HDL门级和开关级描述---MOS开关

#### MOS开关的例化,使用下面的关键字:

- cmos
- nmos
- pmos
- rcmos
- **♦** rnmos
- rpmos

## Verilog HDL门级和开关级描述 --MOS开关

#### 其中:

- pmos表示p型金属氧化物半导体场效应(PMOS)晶体管。
- nmos表示n型金属氧化物半导体场效应(PMOS)晶体管。
- 当导通时,pmos和nmos晶体管的源级和漏级之间阻抗相对较低。
- rpmos表示电阻型PMOS晶体管。
- **rnmos表示电阻型NMOS晶体管。**

与pmos和nmos相比,在导通的时候,rpmos和rnmos在源级和漏级之间的阻抗明显要高。静态MOS网络之间的负载器件是rpmos和rnmos晶体管的一个例子。

这四个开关对数据来说是单向通道,类似于bufif门。

#### MOS开关的延迟描述应该是0个、1个、2个或者3个延迟。

- 如果描述中包含三个延迟,则第一个延迟确定输出上升延迟,第二个延迟确定输出下降延迟,第三个延迟确定跳变到z的延迟。
- 如果描述中包含两个延迟,则第一个延迟确定输出上升延迟,第二个延迟确定输出下降延迟,两个延迟中较小的一个延迟用于确定跳变到x和z的延迟。
- 如果只有一个延迟,将应用到所有的输出延迟。
- 如果没有指定延迟,则开关没有传播延迟。

### 下表给出了MOS开关的真值表。

pmos			控制			nmos	控制						
rpmos		0	1	X	Z	rnmos		0	1	X	Z		
	0	0	Z	L	L		0	Z	0	L	L		
数 据	1	1	Z	H	H	数	1	Z	1	H	H		
据	X	X	Z	X	X	据	X	Z	X	X	X		
	Z	Z	Z	Z	Z		Z	Z	Z	Z	Z		

#### 注:

- 符号L,表示结果为0或z。
- 符号H,表示结果为1或z。
- 跳变到H或者L的延迟与跳变到x的延迟是一样的。

这四个开关有一个数据输出、一个数据输入和一个控制输

入。MOS开关例化语句的基本语法格式如下:

mos\_switch\_type[instance\_name] (outputA,inputB,control);

#### 其中:

- mos\_switch\_type为MOS开关的关键字。
- instance\_name为可选的例化标识符。
- outputA是数据输出端口。
- inputB是数据输入端口。
- control是控制输入端口。

### 例:开关pmos Verilog HDL描述的例子

pmos p1 (out, data, control);

#### 在该门例化语句中:

- out为数据输出。
- data为数据输入。
- control为控制输入。
- p1为例化标识符。

#### 双向传输开关的例化,使用下面的关键字:

- tran
- tranif1
- tranif0
- rtran
- rtranif1
- rtranif0

#### 双向传输开关没有延迟信号传输。

- ◆ 当tranif0、tranif1、rtranif0和rtranif1关闭时,将阻塞信号。
- ◆ 当它们打开时,信号将通过双向传输开关。
- ◆ 不能关闭tran和rtran器件,它们总是能通过信号。

对于tranif0、tranif1、rtranif0和rtranif1来说, 其延迟描述应该是0个、1个或者2个延迟。

- 如果描述中包含两个延迟,则第一个延迟确定打开开关延迟,第二个延迟确定输出关闭开关延迟,两个延迟中较小的一个确定跳变到x和z的延迟。
- 如果描述中包含一个延迟,则用于确定打开和关闭开关延迟。
- 如果没有指定延迟,双向开关没有打开和关闭延迟。

注:对于tran和rtran器件,不接受延迟描述。

#### 双向传输开关例化语句的基本语法格式如下:

pass\_switch\_type[instance\_name] (inout1,inout2,control);

#### 其中:

- pass\_switch\_type为双向开关的关键字。
- instance\_name为可选的例化标识符。
- inout1和inout2是连接信号的两个双向端口。
- control是控制输入。

注:所有6个器件的双向终端,只能连接到标量网络或者向量网络的位选择。

### cmos开关的例化,使用下面的关键字:

- cmos
- rcmos

cmos开关可以看作是pmos开关和nmos开关的组合。

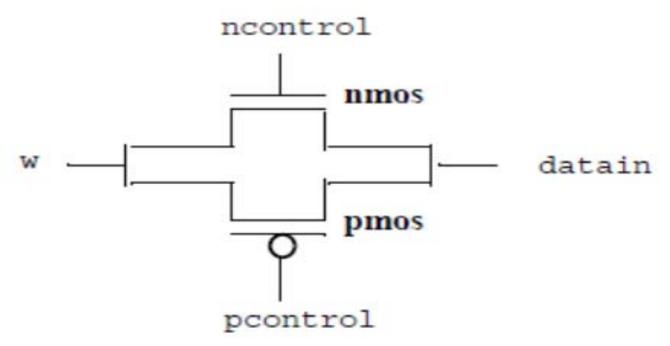
rcmos开关可以看作是rpmos开关和rnmos开关的组合。

#### cmos开关的延迟描述应该是0个、1个、2个或者3个延迟。

- 如果描述中包含三个延迟,则第一个延迟确定输出上升延迟,第
  - 二个延迟确定输出下降延迟,第三个延迟确定跳变到z的延迟,
  - 三个延迟中最小的延迟与跳变到x的延迟。跳变到H或者L的延迟和跳变到x的延迟是相同的。

- 如果描述中包含两个延迟,则第一个延迟确定输出上升延迟,第二个延迟确定输出下降延迟,两个延迟中较小的一个延迟用于确定跳变到x和z的延迟。
- 如果只有一个延迟,将应用到所有的输出跳变延迟。
- 如果没有指定延迟,则开关没有传播延迟。

如下图所示,给出了cmos的符号。cmos和rcmos有一个数据输入datain,一个数据输出,两个控制输入pcontrol和ncontrol。



#### CMOS开关例化语句的基本语法格式如下:

cmos\_switch\_type[instance\_name] (w,datain,ncontrol,pcontrol);

#### 其中:

- cmos\_switch\_type为cmos开关的关键字。
- instance\_name为可选的例化标识符。
- w为数据输出。
- datain为数据输入
- ncontrol是nmos的控制输入。
- pcontrol是pmos的控制输入。

#### 表达式

cmos(w,datain,ncontrol,pcontrol)

#### 等效于:

nmos (w, datain, ncontrol);

pmos (w, datain, pcontrol);

# Verilog HDL门级和开关级描述---pull门

### 例化上拉和下拉源使用下面关键字:

- pullup
- Pulldown

# Verilog HDL门级和开关级描述 --pull门

上拉源pullup将终端列表中的网络置为1。下拉源 pulldown将终端列表中的网络置为0。

- ◆ 在没有强度说明的情况下,放置在网络上的这些信号源,将为 pull强度。
- ◆如果在pullup源上有strength1说明或者在pulldown上有 strength0说明,信号应该有强度说明。
- ◆ 将忽略在pullup上的strength0说明或者在pulldown上的 strength1说明。

# Verilog HDL门级和开关级描述 --pull门

- ◆ 这些源没有延迟描述。
- ◆ 这类门没有输入只有输出。门实例的端口表只包含1个输出。

例:pullup门 Verilog HDL描述的例子

pullup (strong1) p1 (neta), p2 (netb);

在该例化语句中,p1例化驱动neta,p2例化驱动netb,并且为strong1强度。

通过允许标量网络值有宽范围的未知值和不同级的强度或者强度的组合,Verilog HDL为信号竞争、双向传输门、电阻MOS器件、动态MOS,电荷共享和其它依赖于技术的网络配置提供了精确的模型。

通过多层次的逻辑强度建模,解决了当信号组合进入 已知或者未知值的时候,改善硬件行为的精度。

### 提供两个元件用于描述强度:

■ 网络值的strength0,通过下面表示:

supply0 strong0 pull0 weak0 highz0

■ 网络值的strength1,通过下面表示:

supply1 strong1 pull1 weak1 highz1

#### 注:

(highz0, highz1)和(highz1, highz0)的组合是非法的。

### 下表给出了用于网络信号值的驱动层次。

	强度名字	强度极
	supply0	7
atnon atle0	strong0	6
strength0	pull0	5
	large0	4
	weak0	3
	medium0	2
	small0	1
	highz0	0

	highz1	0
strength1	small1	1
	small1 medium1 weak1 arge1 pull1 strong1	2
	weak1	3
	arge1	4
	pull1	5
	strong1	6
	supply1	7

### 从表中可以看出:

■ 有四个驱动强度:

supply strong pull weak

带有强度的信号,将从门的输出进行传播,并且连续的分配输出。

■ 有三个电荷存储强度:

large medium small

带有电荷存储强度的信号将出现在trireg网络类型中。

#### 注:

- 如果一个网络的信号值是已知的,则它的强度级是strength0或者 strength1。
- 如果一个网络的信号值是未知的,则它的强度层次在所有的 strength0或者strength1中。
- ●如果一个网络的信号值是z,则它的强度级是标量部分0划分的一个。

除了一个信号值外,一个网络应该有明确的强度级或者由多个层次构成的模糊强度级。当对信号进行组合时, 根据下面的规则,将确定最终信号的强度和值。

□明确强度的组合信号

这里针对的是每个有确定值和单个强度级的信号的组合。如果在一个连线网络配置中,有两个或者多个不同强度的组合,

- 较强强度的信号将支配所有较弱的驱动器,并确定最终的结果。
- 两个或多个具有相似值的信号组合,将使得相同的值带有更大的强度。
- 强度和值相同的信号组合,将产生相同的信号。

对于带有不同值和相同强度信号的组合,可能有三种结果:

- 两个结果出现在连线逻辑中。
- ■第三个结果产生在非连线逻辑中。

如下图所示,括号内的数字表示信号的相对强度。 pull1和strong0的结果是strong0,即:两个信号中较 强的。

```
Pul(5)
St0(6)
St0(6)
Sul(7)
Lal(4)
Sul(7)
```

□ 不确定强度:源和组合

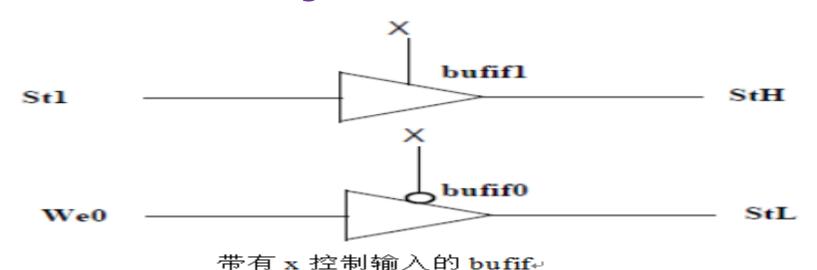
对于不确定强度信号,给出了下面的分类:

- ◆ 带有确定值和多个强度级的信号。
- ◆ 带有x的信号,其信号强度由strength1部分和strength0部分范围的子分类构成。
- ◆ 带有L的信号,其信号强度由高阻和strength0部分的强度级构成。
- ◆ 带有H的信号,其信号强度由高阻和strength1部分的强度级构成。

多个配置可以产生不确定强度的信号。如下图所示, 当两个相同强度和相反值的信号组合在一起时,导致x的值,其强度包含所有信号的强度和所有更小的强度级。

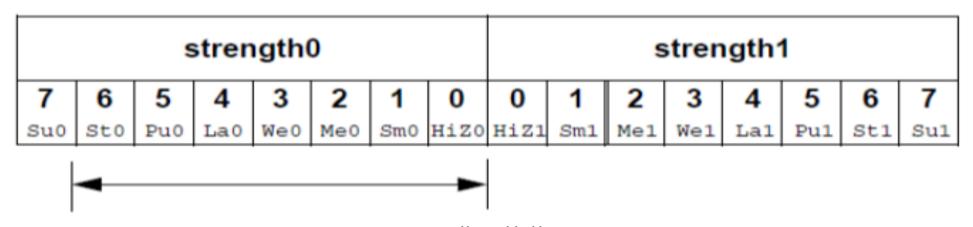
We1									1	WeX					
We0 -															
	strength0								strength1						
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Mel	Wel	La1	Pu1	St1	Su1
				<b>—</b>							-				

- 一个不确定信号的强度可以是一个范围内可能的值。如下 图给出了从三态驱动器的输出强度,其控制端的输入未知。
- ◆ bufif1的输出是Strong H , 如下图a所示 , 给出了取值的范围。
- ◆ bufif0的输出是Strong L,如下图b所示,给出了取值的范围。

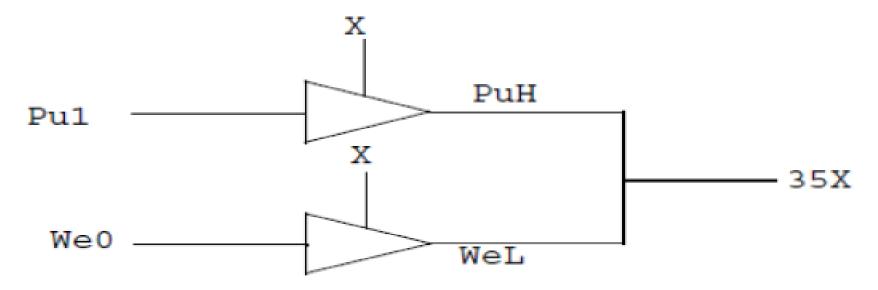


strength0							strength1								
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1
								1							

图a强H范围的值↓



两个不确定强度的信号的组合将导致一个不确定强度的信号。最终的信号将是一个强度级范围,其中包含了它元件内信号的强度级。如下图所示,不确定强度的信号组合,将产生一个范围。



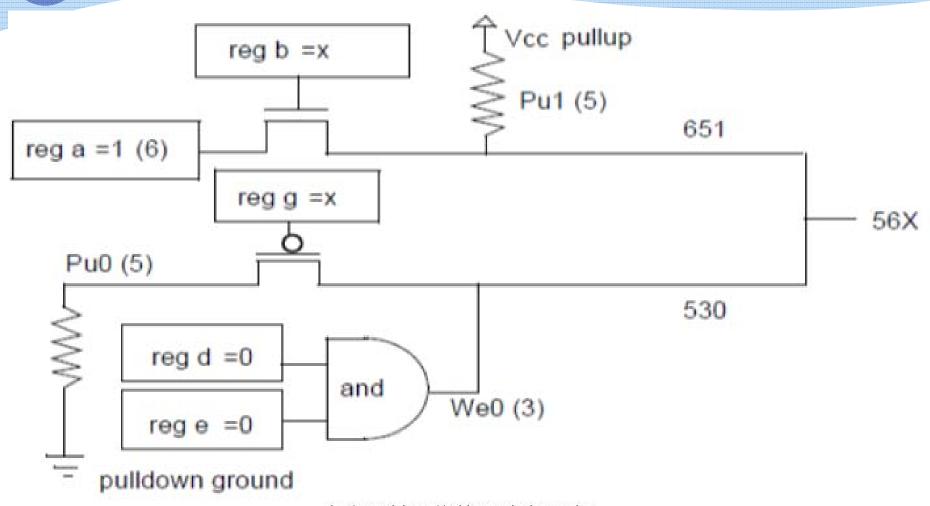
#### 如下图所示,包含信号的极限值和极限值之间的取值。

	strength0							strength1							
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	Wel	La1	Pu1	St1	Su1
				•									-		

不确定信号的强度范围↓

因为其范围包含1和0,所以结果是一个值x。x前面的数字35,是两位的连接。第一个3,对应于用于结果的最高的strength0级;第二个5,对应于用于结果的最高的strength1级。

如下图所示,开关网络产生相同值,包括一个强度范围,比如:来自上面和下面配置。一个寄存器的上半部分组合,由一个未指定值的寄存器所控制的门,一个上拉电阻产生一个值为1的信号。



来自开关网络的不确定强度₽

如下图所示,强度范围(651)。一个pulldown的下半部分组合。由一个未指定值的寄存器控制的门和一个and门生成一个值为零的信号。

		5	stren	gth(	)					s	tren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1
						个定义	义值的	两个强	1度范	围艹			4	-	

#### 如下图所示,强度范围(530)。

		9	strer	gth	0					S	tren	gth	I		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1
		4		-											

一个定义值的三个强度范围₹

当把上部分和下部分的信号进行组合时,结果是一个不确定的值。如下图所示,范围为(56x),由两个信号的极值确定。

		5	strer	igth(	)					S	stren	gth′	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1
		4			<b>ተ</b> ታበ /	± 65 29		III						-	

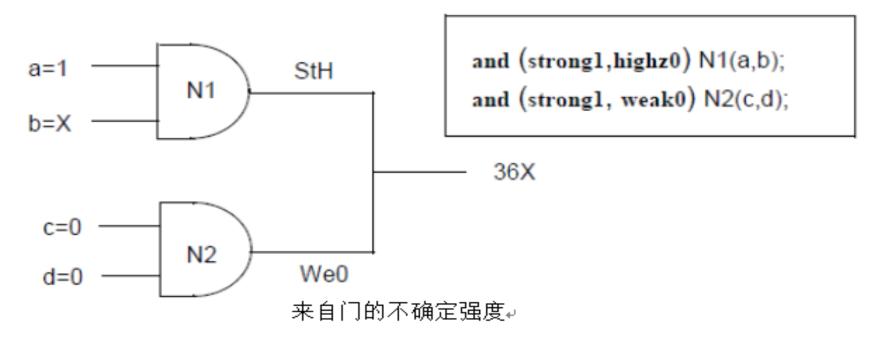
如果将下部分的配置中的pulldown用supply0代替,如下图所示,将范围改变为(Stx)。图中结果是Strongx,这是因为它是未知的,所有元件的极值都是strong。由于较低部分pmos减少了supply0信号的强度所以较低配置输出的极值是strong。在后面会讨论其模型特性。

		9	strer	gth(	)					5	tren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1
	4				·	 虽 <b>x</b> 范	 .围↓							•	,

如下图所示,逻辑门产生不确定的强度和三态驱动器。与门N1声明了highz0强度,N2声明了weak0强度。图中寄存器类型的b没有确定的值。

		5	strer	igth(	0					5	stren	gth′	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1
	4				2	——— 虽 <b>x</b> 茆	 							•	

因此,输入到上面的and门是stong x。上面的and的强度描述包括highz0。来自上面and门的信号是strong H,如下图所示,给出了强度的范围。



因为在考虑门的强度描述中,一个输出的指定强度中有一个值为0。所以,Hiz0是结果的一部分。用于0值输出的一个强度描述不同于高阻时,将使得一个门的输出值为x。

#### 如下图所示,下面and门的输出是weak0。

		s	tren	gth(	0					s	tren	gth′	1		
7	7 6 5 4 3 2 1								1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Sul
				<b>*</b>			V	/eak0+	J						

如下图所示,将信号进行组合时,结果是(36x)。



□模糊强度信号和明确信号

一个带有明确强度和值的信号和其它带有未确定强度 信号的组合,将导致几种可能的情况。

为了理解用于这种类型组合的规则集,需要分开考虑每个未确定强度信号相对于确定强度信号的强度级。

#### 当一个带有明确强度和值的信号和其它带有未确定强度信 号的组合时,存在下面的规则:

- ◆ 不确定信号的强度级大于确定信号的强度级时,将保留在结果中。
- ◆ 不确定信号的强度级小于或者等于确定信号的强度级时,将从结果中消失,服从规则c。下图说明了这个规则。

		s	trer	gth	0					s	tren	gth	1		
7	7 6 5 4 3 2 1								1	2	3	4	5	6	7
Su0	St0	Pu0	La0	WeO	MeO	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Sul
	•						-								

		stre	ngth	0					S	tren	gth	1		
7 (	6 5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0 St	to Pu	10 La0	WeO	Me0	Sm0	HiZ0	HiZ1	Sm1	Mel	We1	La1	Pu1	St1	Su1

上面两个信号组合的结果如下:

		s	tren	gth	0					s	tren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1
	<b>4</b>					虽度值	.的消》								

◆如果规则a和规则b的操作,由于信号的值相反,导致强度级的空隙,强度级的空隙也将是结果的一部分。下图说明了这个规则。

		5	stren	gth	0					s	tren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	StO	Pu0	La0	WeO	MeO	Sm0	HiZO	HiZ1	Sml	Mel	Wel	La1	Pu1	St1	Sul
					$\overline{}$									_	

		5	stren	gth	0					S	stren	gth	1		
7	6	5	4	3	2	1	0 0 1 2 3 4 5 6							7	
Su0	St0	Pu0	La0	WeO	MeO	Sm0	HiZO	HiZ1	Sml	Mel	Wel	La1	Pul	St1	Sul

上面两个信号组合的结果如下:

		s	stren	gth(	0					s	tren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	WeO	Me0	Sm0	HiZO	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1

如上图所示,应用规则a,b和c。不确定强度信号相反的值,比确定强度信号强度小的信号强度从结果中消失了。确定强度信号的强度和不确定强度信号中较大的极值,定义了结果中的一个范围。

如下图所示,应用规则a,b。不确定强度信号中,比确 定强度信号强度小的强度从结果中消失了。确定强度信号 的强度和不确定强度信号中较大的极值,定义了结果中的 一个范围。

		5	stren	gth	0					S	tren	gth′	1		
7	6	5	4	3					1	2	3	4	5	6	7
Su0	St0	Pu0	La0	WeO	MeO	Sm0	HiZ0	HiZ1	Sm1	Me1	Wel	La1	Pu1	St1	Su1
								$\blacksquare$					$\overline{}$		

			s	stren	gth	0					s	trer	gth′	1		
Г	7	6	5	4	3	2	1	0	O HiZ1	1	2	3	4	5	6	7
5	Su0	St0	Pu0	La0	WeO	MeO	Sm0	HiZ0	HiZ1	Sm1	Mel	We1	La1	Pu1	St1	Sul

上面两个信号组合的结果如下:

		s	stren	gth	0					s	tren	gth'	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	WeO	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	Wel	La1	Pu1	St1	Su1

如下图所示,应用规则a,b和c。不确定强度信号强度范围的较大的极值比确定强度信号的强度级要大。确定强度信号的强度和不确定强度信号中最大的极值,定义了结果中的一个范围。

		s	stren	gth(	0					S	stren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	WeO	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	Wel	La1	Pu1	St1	Su1
								-						-	

		5	stren	gth(	0					s	tren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	WeO	Me0	Sm0	HiZ0	HiZ1	Sm1	Mel	We1	La1	Pu1	St1	Sul
		-													

上面两个信号组合的结果如下:

		ş	stren	ngth(	0					ş	tren	gth′	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1

#### □线逻辑网络类型

当多个驱动器有相同的强度时,网络类型triand、wand、trior和wor可以解决冲突。通过将信号看作是逻辑功能的输入,这些网络类型将解决信号的值。

如下图所示,给出了两个确定信号的组合。使用"线与"逻辑产生的结果,与两个信号作为and门的输入,产生的结果相同。使用"线或"逻辑产生的结果,与两个信号作为or门的输入,产生的结果相同。在所有情况下,结果的强度和的组合信号的强度相同。

		S	trer	gth	0					S	tren	igth′	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Me1	We1	La1	Pu1	St1	Su1

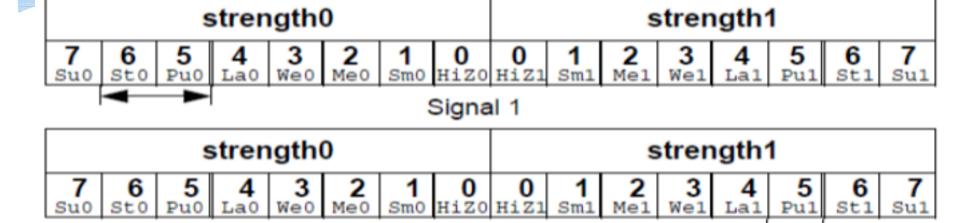
		S	stren	gth	0	_				s	tren	gth	1		
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
Su0	St0	Pu0	La0	We0	Me0	Sm0	HiZ0	HiZ1	Sm1	Mel	Wel	La1	Pu1	St1	Su1

**→** 

wired AND logic value result: 0 wired OR logic value result: 1

带有确定强度的布线逻辑

接下来的两张图将表示,当在布线逻辑中,对不确定强度信号组合时,需要考虑第一个信号的每个强度级和第二个信号的每个强度级所有组合的结果。



signal	1	signal	2	result	t
strength	value	strength	value	strength	value
5	0	5	1	5	0
6	0	5	1	6	0

Signal 2

用于 and 逻辑的强度强度级组合↓

signal	11	signal	2	result	t
strength	value	strength	value	strength	value
5	0	5	1	5	1
6	0	5	1	6	0

7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 Su0 St0 Pu0 La0 We0 Me0 Sm0 HiZ0 HiZ1 Sm1 Me1 We1 La1 Pu1 St			1	ngth'	strer	S					0	gth	strer	•		
Su0 St0 Pu0 La0 We0 Me0 Sm0 HiZ0 HiZ1 Sm1 Me1 We1 La1 Pu1 St	<b>7</b>	6	<b>5</b>	<b>4</b>	3	<b>2</b>	1	0	0	1	<b>2</b>	3	<b>4</b>	<b>5</b>	6	<b>7</b>
	Su1	St1	Pu1	La1	Wel	Me1	Sm1	HiZ1	HiZO	Sm0	Me0	We0	La0	Pu0	St0	Su0

#### Verilog HDL门级和开关级描述 --通过非电阻器件的强度降低

除了将supply强度降低到strong外, nmos、pmos和cmos开关将数据输入的强度传输到输出。

除了将supply强度降低到strong外, tran、tranif0和 tranif1开关不影响穿越双向终端的信号强度。

#### Verilog HDL门级和开关级描述 --通过电阻器件的强度降低

如下表所示, rmos、rpmos、rcmos、rtran、rtranif1和rtranif0器件, 将减少通过这些器件的信号强度。

## Verilog HDL门级和开关级描述 --通过电阻器件的强度降低

输入强度	输出强度
Supply驱动	Pull驱动
Strong驱动	Pull驱动
Pull驱动	Weak驱动
Large电容	Medium电容
Weak驱动	Medium电容
Meaium电容	Small电容
Small电容	Small电容
高阻	高阻

tri0、tri1、supply0、supply1网络类型将产生指定强度级生成信号。trireg声明能指定两个信号强度级中的一个,而不是一个默认的强度级。

#### □ triO和tri1网络强度

- ◆ tri0网络类型用于建模一个连接到一个电阻pulldown器件的 网络。当缺少一个覆盖信号源时,这样一个信号其值为0 且强度为pull。
- ◆ tri1网络类型用于建模一个连接到一个电阻pullup器件的网络。当缺少一个覆盖信号源时,这样一个信号其值为1且强度为pull。

#### □trireg强度

- ◆ trireg网络类型,用于建模电荷存储点。
- ◆ 由一个trireg网络产生的驱动强度,即在电荷存储状态(即:一个驱动器对一个网络进行充电,然后转为高阻),应该是三种状态的其中一个,即:large、medium和small。
- ◆ 一个特殊trireg网络相关的指定强度,应该在网络声明中由设计者指定。默认地,强度为medium。

- □supply0和supply1网络强度
  - ◆ supply0用于对地连接进行建模。
  - ◆ supply1网络类型用于对电源连接进行建模。
  - ◆ supply0和supply1网络类型应该有supply驱动强度。

可以使用门时延定义门从任何输入到其输出的信号传输时延。门时延可以在门例化语句中定义。

带有时延定义的门实例语句的语法格式如下:

gate\_type [delay] [instance\_name] (terminal\_list);

#### 其中:

- ◆ gate\_type为门的类型。
- ◆ instance\_name为可选的例化标识符。
- ◆ delay指定了门时延,即从门的任意输入到输出的传输时延。

用于输出的门时延最多有三类时延值,即上升时延,下降时延,高阻时延。

- ◆ 当没有强调门和网络时延时,默认的时延值为0。
- ◆ 当指定一个延迟时,这个值用于和门或者网络相关的所有传输延迟。
- ◆ 当指定两个延迟时,第一个延迟指定上升延迟,第二个延迟指定下降延迟。当信号变为高阻或者不确定时,延迟时两个延迟中较小的一个。

- ◆ 对于三个延迟描述:
  - 第一个延迟指的是跳变到值1(上升延迟)。
  - 第二个延迟指的是跳变到值0(下降延迟)。
  - 第三个延迟指的是跳变到高阻值。
- ◆ 当跳变到未确定值的时候,值是三个延迟中最小的一个。输入信号的强度不影响从一个输入到输出的传播延迟。

#### 下表为不同个数时延值条件下,各种具体的时延取值情形。

从一个值	到另一个值	如果有,使用的延迟	
		2个时延	3个时延
0	1	d1	d1
0	X	min(d1, d2)	min(d1, d2, d3)
0	Z	min(d1, d2)	d3
1	0	d2	<b>d2</b>

从值	到值	如果有,使用的延迟	
		2个时延	3个时延
1	X	min(d1, d2)	min(d1, d2, d3)
1	Z	min(d1, d2)	d3
X	0	<b>d2</b>	<b>d2</b>
X	1	d1	<b>d1</b>
X	Z	min(d1, d2)	<b>d3</b>
Z	0	<b>d2</b>	<b>d2</b>
Z	1	d1	<b>d1</b>
Z	X	min (d1, d2)	min(d1, d2, d3)

注:

Verilog HDL模型中的所有时延都以单位时间表示。可以通过`timescale编译器指令实现单位时间与实际时间的关联。

例:指定延迟Verilog HDL描述的例子

```
and #(10) a1 (out, in1, in2);  //只有一个延迟
```

and #(10,12) a2 (out, in1, in2); //上升和下降延迟

bufif0 #(10,12,11) b3 (out, in, ctrl); // 上升、下降和截止延迟

#### 例:简单锁存器Verilog HDL描述的例子

```
module tri_latch (qout, nqout, clock, data, enable);
output qout, nqout;
input clock, data, enable;
tri qout, nqout;
not #5 n1 (ndata, data);
nand #(3,5) n2 (wa, data, clock),
n3 (wb, ndata, clock);
nand #(12,15) n4 (q, nq, wa),
n5 (nq, q, wb);
bufif1 \#(3,7,13) q_drive (qout, q, enable),
               nq_drive (nqout, nq, enable);
endmodule
```

□min:typ:max延迟

门延迟也可采用min:typ:max形式定义。其语法格式如

下:

minimum: typical: maximum

最小值、典型值和最大值必须是常数表达式。

带有min: typ: max的上升、下降和截止延迟的Verilog HDL描述的例子。

```
module iobuf (io1, io2, dir);
....
bufif0 #(5:7:9, 8:10:12, 15:18:21) b1 (io1, io2, dir);
bufif1 #(6:8:10, 5:7:9, 13:17:19) b2 (io2, io1, dir);
....
endmodule
```

在过程描述中控制延迟Veriog HDL描述的例子。

```
parameter min_hi = 97, typ_hi = 100, max_hi = 107;
reg clk;
always begin
     #(95:100:105) clk = 1;
     #(min_hi:typ_hi:max_hi) clk = 0;
end
```

□ trireg网络电荷衰减

trireg网络声明中,最多可以包含三个延迟。当一个驱动器将trireg网络驱动到1和0状态时,前两个延迟指定了跳变到1和0的逻辑状态的延迟,第三个延迟指定了电荷衰减的时间,而不是跳变到z逻辑状态的延迟。

电荷衰减时间是指当驱动一个trireg网络关闭到不再确定它所存储的电荷时的延迟。

由于trireg网络从来不会跳变到z逻辑状态,所以trireg网络不需要说明截止延迟。

当一个trireg网络的驱动器将1,0或者x逻辑状态改变为关闭时,trireg将保持前面驱动器的1,0或者x状态。z值不会从一个trireg网络的驱动器传播到一个trireg网络。

当z是trireg网络的初始状态,或者使用force语句强迫trireg网络到z状态时,trireg才能保持一个z逻辑状态。

为电荷衰减的一个延迟说明,用于建模一个非理想的 电荷存储节点,比如:一个电荷存储节点,通过它周围的 器件和连接,泄露电荷。

电荷衰减是在一个指定延迟后,引起trireg网络内保存的1或者0跳变到未确定值x的原因。当关闭trireg网络的驱动器时,开始电荷衰减过程。并且,trireg网络开始保持电荷。

在下面两种情况下,结束电荷衰减的过程。

- ◆ 到达由电荷衰减指定的延迟,并且trireg网络从1或者0跳变到x。
- ◆打开trireg网络的驱动器,将1,0或者x传播到trireg网络。

#### 在trireg网络声明中,一个三值延迟描述的形式为:

#(d1, d2, d3)

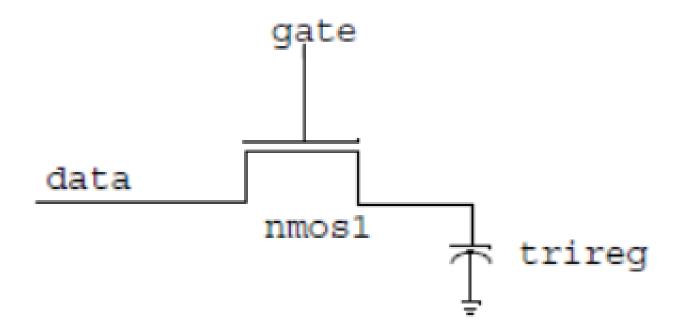
#### 其中:

- d1为上升延迟。
- d2为下降延迟。
- d3为电荷衰减时间。

三值trireg网络声明Verilog HDL描述的例子。

trireg (large) #(0,0,50) cap1;

如下图所示,给出三值trireg网络声明的Veriog HDL描述的例子。



带有电容的 trireg 网络₽

```
module capacitor;
reg data, gate;
// trireg declaration with a charge decay time of 50
 time units
trireg (large) #(0,0,50) cap1;
nmos nmos1 (cap1, data, gate); // nmos that drives
                                   the trireg
```

initial begin

\$monitor("%0d data=%v gate=%v cap1=%v", \$time, data, gate, cap1);

```
data = 1;
// Toggle the driver of the control input to the nmos
  switch
gate = 1;
#10 gate = 0;
#30 gate = 1;
#10 gate = 0;
#100 $finish;
end
endmodule
```