第5章　门电平模型化

本章讲述Verilog HDL为门级电路建模的能力，包括可以使用的内置基本门和如何使用它们来进行硬件描述。

5.1　内置基本门

Verilog HDL中提供下列内置基本门：

1) 多输入门：

and, nand,or, nor,xor,xnor

2) 多输出门：

buf, not

3) 三态门：

bufif0, bufif1, notif0,notif1

4) 上拉、下拉电阻：

pullup, pulldown

5) MOS开关：

cmos, nmos, pmos, rcmos, rnmos, rpmos

6) 双向开关：

tran,tranif0, tranif1, rtran, rtranif0, rtranif1

门级逻辑设计描述中可使用具体的门实例语句。下面是简单的门实例语句的格式。

```
gate_type[instance_name] (term1, term2, . . . ,termN);
```

注意，*instance_name*是可选的；*gate_type*为前面列出的某种门类型。各 *term*用于表示与门的输入/输出端口相连的线网或寄存器。

同一门类型的多个实例能够在一个结构形式中定义。语法如下：

```
gate_type
       [instance_name1] (term11, term12, . . .,term1N),
       [instance_name2] (term21, term22, . . .,term2N),
       . . .
       [instance_nameM] (termM1, termM2, . . .,termMN);
```

5.2　多输入门

内置的多输入门如下：

```
and   nand   nor   or   xor   xnor
```

这些逻辑门只有单个输出， 1个或多个输入。多输入门实例语句的语法如下：

```
multiple_input_gate_type
  [instance_name] (OutputA, Input1, Input2, . . .,InputN);
```

第一个端口是输出，其它端口是输入。如图5-1所示。

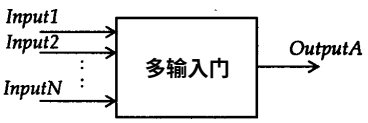

图5-1　多输入门

下面是几个具体实例。图5-2为对应的逻辑图。

```
and A1(Out1, In1, In2);

and RBX (Sty, Rib, Bro, Qit, Fix);

xor (Bar, Bud[0],Bud[1], Bud[2]),
    (Car, Cut[0], Cut[1]),
    (Sar, Sut[2], Sut[1], Sut[0], Sut[3]);
```

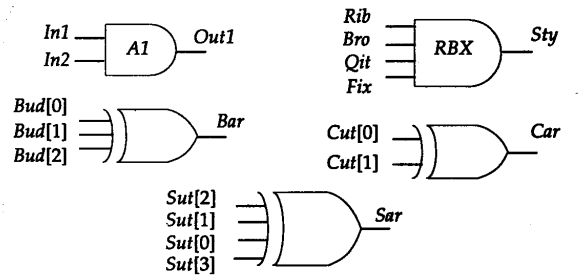

图5-2 多输入门实例

第一个门实例语句是单元名为*A1*、输出为*Out1*、并带有两个输入*In1*和*In2*的两输入与门。第二个门实例语句是四输入与门，单元名为*RBX*，输出为*Sty*，4个输入为*Rib*、*Bro*、*Qit*和*Fix*。第三个门实例语句是异或门的具体实例，没有单元名。它的输出是 *Bar*，三个输入分别为*Bud*[0]、*Bud*[1]和*Bud*[2]。同时，这一个实例语句中还有两个相同类型的单元。

下面是这些门的真值表。注意在输入端的 z与对x的处理方式相同；多输入门的输出决不能是z。

nand	0	1	x	z
0	1	1	1	1
1	1	0	x	x
x	1	x	x	x
z	1	x	x	x

and	0	1	x	z
0	0	0	0	0
1	0	1	x	x
x	0	x	x	x
z	0	x	x	x

or	0	1	x	z
0	0	1	x	x
1	1	1	1	1
x	x	1	x	x
z	x	1	x	x

nor	0	1	x	z
0	1	0	x	x
1	0	0	0	0
x	x	0	x	x
z	x	0	x	x

xor	0	1	x	z
0	0	1	x	x
1	1	0	x	x
x	x	x	x	x
z	x	x	x	x

xnor	0	1	x	z
0	1	0	x	x
1	0	1	x	x
x	x	x	x	x
z	x	x	x	x

5.3 多输出门

多输出门有：

```
buf  not
```

这些门都只有单个输入，一个或多个输出。如图 5-3所示。这些门的实例语句的基本语法如下：

```
multiple_output_gate_type
        [instance_name] (Out1, Out2, . . . OutN ,InputA);
```

最后的端口是输入端口，其余的所有端口为输出端口。

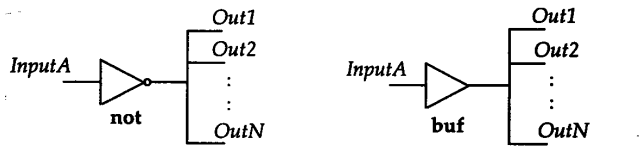

图5-3 多输出门

例如：

```
buf B1  (Fan [0], Fan [1], Fan [2], Fan [3], Clk) ;
not N1  (PhA, PhB, Ready) ;
```

在第一个门实例语句中，*Clk*是缓冲门的输入。门*B1*有4个输出：*Fan*[0]到*Fan*[3]。在第二个门实例语句中，*Ready*是非门的唯一输入端口。门*N1*有两个输出：*PhA*和*PhB*。

这些门的真值表如下：

buf	0	1	x	z
(输出)	0	1	x	x

not	0	1	x	z
(输出)	1	0	x	x

5.4 三态门

三态门有：

```
bufif0 bufif1 notif0 notif1
```

这些门用于对三态驱动器建模。这些门有一个输出、一个数据输入和一个控制输入。三态门实例语句的基本语法如下：

```
tristate_gate[instance_name] (OutputA, InputB,Control);
```

第一个端口*OutputA*是输出端口，第二个端口*InputB*是数据输入，*ControlC*是控制输入。参见图5-4。根据控制输入，输出可被驱动到高阻状态，即值 z。对于bufif0，若通过控制输入为1，则输出为z；否则数据被传输至输出端。对于 bufif1，若控制输入为0，则输出为z。对于notif0，如果控制输出为 1，那么输出为 z；否则输入数据值的非传输到输出端。对于 notif1，若控制输入为0；则输出为z。

例如：

```
bufif1 BF1  (Dbus, MemData, Strobe) ;
notif0 NT2 (Addr, Abus, Probe);
```

当*Strobe*为0时，bufif1门*BF1*驱动输出*Dbus*为高阻；否则*MemData*被传输至*Dbus*。在第2个实例语句中，当*Probe*为1时，*Addr*为高阻；否则*Abus*的非传输到*Addr*。

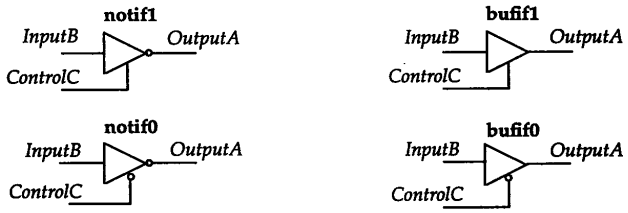

图5-4 三态门

下面是这些门的真值表。表中的某些项是可选项。例如， 0/z表明输出根据数据的信号强度和控制值既可以为0也可以为z，信号强度在第10章中讨论。

bufif0		控制			
		0	1	x	z
数据	0	0	z	0/z	0/z
	1	1	z	1/z	1/z
	x	x	z	x	x
	z	x	z	x	x

bufif1		控制			
		0	1	x	z
数据	0	z	0	0/z	0/z
	1	z	1	1/z	1/z
	x	z	x	x	x
	z	z	x	x	x

notif0		控制			
		0	1	x	z
数据	0	1	z	1/z	1/z
	1	0	z	0/z	0/z
	x	x	z	x	x
	z	x	z	x	x

notif1		控制			
		0	1	x	z
数据	0	z	1	1/z	1/z
	1	z	0	0/z	0/z
	x	z	x	x	x
	z	z	x	x	x

5.5 上拉、下拉电阻

上拉、下拉电阻有：

```
pullup        pulldown
```

这类门设备没有输入只有输出。上拉电阻将输出置为 1。下拉电阻将输出置为 0。门实例语句形式如下：

```
pull_gate[instance_name] (OutputA);
```

门实例的端口表只包含1个输出。例如：

```
pullup PUP (Pwr);
```

此上拉电阻实例名为PUP，输出Pwr置为高电平1。

5.6 MOS开关

MOS开关有：

```
cmos pmos nmos rcmos rpmos rnmos
```

这类门用来为单向开关建模。即数据从输入流向输出，并且可以通过设置合适的控制输入关闭数据流。

pmos(p类型MOS管)、nmos(n类型MOS管)，rnmos(r代表电阻)和rpmos开关有一个输出、一个输入和一个控制输入。实例的基本语法如下：

```
gate_type[instance_name] (OutputA, InputB, Control);
```

第一个端口为输出，第二个端口是输入，第三个端口是控制输入端。如果 nmos和rnmos开关的控制输入为0，pmos和rpmos开关的控制为1，那么开关关闭，即输出为 z；如果控制是1，输入数据传输至输出；如图 5-5所示。与nmos和pmos相比，rnmos和rpmos在输入引线和输出引线之间存在高阻抗(电阻)。因此当数据从输入传输至输出时，对于 rpmos和rmos，存在数据信号强度衰减。信号强度将在第10章进行讲解。

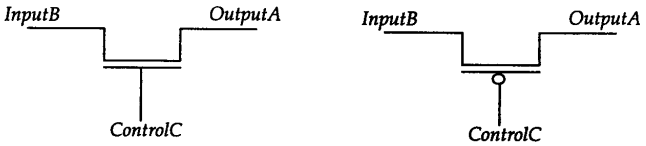

图5-5 nmos和pmos开关

例如：

```
pmos P1 (BigBus, SmallBus, GateControl);
rnmos RN1 (ControlBit, ReadyBit, Hold);
```

第一个实例为一个实例名为 *P1* 的pmos开关。开关的输入为 *SmallBus*，输出为 *BigBus*，控制信号为 *GateControl*。

这些开关的真值表如下所示。表中的某些项是可选项。例如， 1/z表明，根据输入和控制信号的强度，输出既可以为1，也可以为z。

pmos rpmos		控制			
		0	1	x	z
数据	0	0	z	0/z	0/z
	1	1	z	1/z	1/z
	x	x	z	x	x
	z	z	z	z	z

nmos rnmos		控制			
		0	1	x	z
数据	0	z	0	0/z	0/z
	1	z	1	1/z	1/z
	x	z	x	x	x
	z	z	z	z	z

cmos(mos求补)和rcmos(cmos的高阻态版本)开关有一个数据输出，一个数据输入和两个控制输入。这两个开关实例语句的语法形式如下：

```
(r)cmos [instance_name]
        (OutputA, InputB, NControl, PControl);
```

第一个端口为输出端口，第二个端口为输入端口，第三个端口为n通道控制输入，第四个端口为是 P通道控制输入。cmos(rcmos)开关行为与带有公共输入、输出的 pmos（rpmos）和nmos(rnmos)开关组合十分相似。参见图5-6。

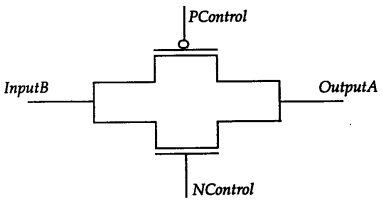

图5-6 (r)cmos开关

5.7 双向开关

双向开关有：

```
tran rtran tranif0 rtranif0 tranif1 rtranif1
```

这些开关是双向的，即数据可以双向流动，并且当数据在开关中传播时没有延时。后 4个开关能够通过设置合适的控制信号来关闭。 tran和rtran开关不能被关闭。

tran或rtran(tran 的高阻态版本)开关实例语句的语法如下：

```
(r)tran [instance_name] (SignalA, SignalB);
```

端口表只有两个端口，并且无条件地双向流动，即从 *SignalA*向*SignalB*，反之亦然。

其它双向开关的实例语句的语法如下：

```
gate_type[instance_name] (SignalA, SignalB, Control);
```

前两个端口是双向端口，即数据从 *SignalA*流向*SignalB*，反之亦然。第三个端口是控制信号。如果对 tranif0和tranif0，*ControlC*是1；对tranif1和rtranif1，*ControlC*是0；那么禁止双向数据流动。对于 rtran、rtranif0和rtranif1，当信号通过开关传输时，信号强度减弱。

5.8 门时延

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。门时延可以在门自身实例语句中定义。带有时延定义的门实例语句的语法如下：

```
gate_type [delay][instance_name](terminal_list);
```

时延规定了门时延，即从门的任意输入到输出的传输时延。当没有强调门时延时，缺省的时延值为0。

门时延由三类时延值组成：

1) 上升时延

2) 下降时延

3) 截止时延

门时延定义可以包含 0个、1个、2个或3个时延值。下表为不同个数时延值说明条件下，各种具体的时延取值情形。

	无时延	1个时延(d)	2个时延(d1, d2)	3个时延 (dA, dB, dC)
上升	0	d	d1	dA
下降	0	d	d2	dB
to_x	0	d	min① (d1, d2)	min (dA, dB, dC)
截止	0	d	min (d1, d2)	dC

① min 是minimum 的缩写词。

注意转换到x的时延(to_x)不但被显式地定义，还可以通过其它定义的值决定。

下面是一些具体实例。注意 Verilog HDL模型中的所有时延都以单位时间表示。单位时间与实际时间的关联可以通过 `timescale编译器指令实现。在下面的实例中，

```
not N1 (Qbar, Q);
```

因为没有定义时延，门时延为 0。下面的门实例中，

```
nand #6 (Out, In1, In2);
```

所有时延均为6，即上升时延和下降时延都是6。因为输出决不会是高阻态，截止时延不适用于与非门。转换到x的时延也是6。

```
and #(3,5) (Out, In1, In2, In3);
```

在这个实例中，上升时延被定义为3，下降时延为5，转换到x的时延是3和5中间的最小值，即3。在下面的实例中，

```
notif1 #(2,8,6) (Dout, Din1, Din2);
```

上升时延为2，下降时延为8，截止时延为6，转换到x的时延是2、8和6中的最小值，即2。

对多输入门（例如与门和非门）和多输出门（缓冲门和非门）总共只能够定义2个时延(因为输出决不会是z)。三态门共有3个时延，并且上拉、下拉电阻实例门不能有任何时延。

min:typ:max时延形式

门延迟也可采用*min:typ:max*形式定义。形式如下：

```
minimum: typical: maximum
```

最小值、典型值和最大值必须是常数表达式。下面是在实例中使用这种形式的实例。

```
nand #(2:3:4, 5:6:7) Pout, Pin1, Pin2;
```

选择使用哪种时延通常作为模拟运行中的一个选项。例如，如果执行最大时延模拟，与非门单元使用上升时延4和下降时延7。

程序块也能够定义门时延。程序块的定义和说明在第10章中讨论。

5.9 实例数组

当需要重复性的实例时，在实例描述语句中能够有选择地定义范围说明(范围说明也能够在模块实例语句中使用)。这种情况的门描述语句的语法如下：

```
gate_type [delay]instance_name[leftbound:rightbound]
        (list_of_terminal_names);
```

*leftbound*和*rightbound*值是任意的两个常量表达式。左界不必大于右界，并且左、右界两者都不必限定为0。示例如下。

```
wire [3:0] Out, InA, InB;
. . .
nand Gang [3:0] (Out, InA, InB);
```

带有范围说明的实例语句与下述语句等价：

```
nand
    Gang3 (Out[3], InA[3], InB[3]),
    Gang2 (Out[2], InA[2], InB[2]),
    Gang1 (Out[1], InA[1], InB[1]),
    Gang0 (Out[0], InA[0], InB[0]);
```

注意定义实例数组时，实例名称是不可选的。

5.10 隐式线网

如果在Verilog HDL模型中一个线网没有被特别说明，那么它被缺省声明为1位线网。但是`default_nettype编译指令能够用于取代缺省线网类型。编译指令格式如下：

```
`default_nettype net_type
```

例如：

```
`default_nettype wand
```

根据此编译指令，所有后续未说明的线网都是 wand类型。

`default_nettype编译指令在模块定义外出现，并且在下一个相同编译指令或 `resetall编译指令出现前一直有效。

5.11 简单示例

下面是图5-7中4-1多路选择电路的门级描述。注意因为实例名是可选的（除用于实例数组情况外），在门实例语句中没有指定实例名。

```
module MUX4x1 (Z,D0,D1,D2,D3,S0,S1);
  output Z;
  input D0,D1,D2,D3,S0,S1;

  and (T0,D0,S0bar,S1bar),
      (T1,D1,S0bar,S1),
      (T2,D2,S0,S1bar),
      (T3,D3,S0,S1),

  not (S0bar,S0),
      (S1bar,S1);

  or (Z,T0,T1,T2,T3,);
endmodule
```

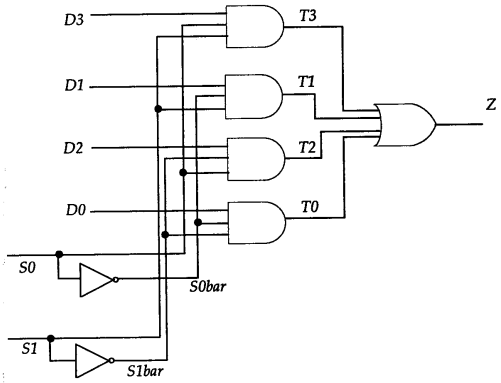

图5-7 4-1多路选择电路

如果或门实例由下列的实例代替呢？

```
or Z (Z,T0,T1,T2,T3);   //非法的Verilog HD表达式。
```

注意实例名还是Z，并且连接到实例输出的线网也是 Z。这种情况在Verilog HDL中是不允许的。在同一模块中，实例名不能与线网名相同。

5.12 2-4解码器举例

图5-8中显示的2-4解码器电路的门级描述如下：

```
module DEC2X4 (A,B,Enable,Z);
  input A,B,Enable;
  output [0:3] Z;
  wire Abar, Bbar;

  not # (1,2)
    V0 (Abar,A),
    V1(Bbar, B);

  nand # (4,3)
    N0 (Z[3], Enable, A,B),
    N1 (Z[0], Enable, Abar,Bbar),
    N2 (Z[1], Enable, Abar,B),
    N3 (Z[2], Enable, A,Bbar);
endmodule
```

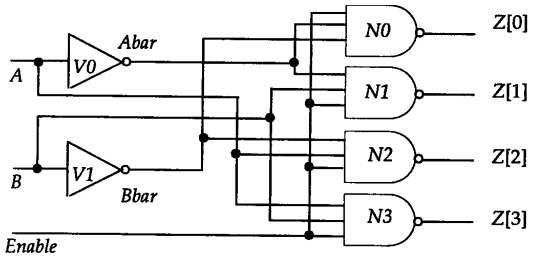

图5-8 2-4解码器电路

5.13 主从触发器举例

图5-9所示的主从D触发器的门级描述如下：

```
module MSDFF (D,C,Q,Qbar);
  input D,C;
  output Q,Qbar;

  not
    NT1 (NotD,D),
    NT2 (NotC,C),
    NT3 (NotY,Y);

  nand
    ND1 (D1,D,C),
    ND2 (D2,C,NotD),
    ND3 (Y,D1,Ybar),
    ND4 (Ybar,Y,D2),
    ND5 (Y1,Y,NotC),
    ND6 (Y2,NotY,NotC),
    ND7 (Q,Qbar,Y1),
    ND8 (Qbar,Y2,Q);
endmodule
```

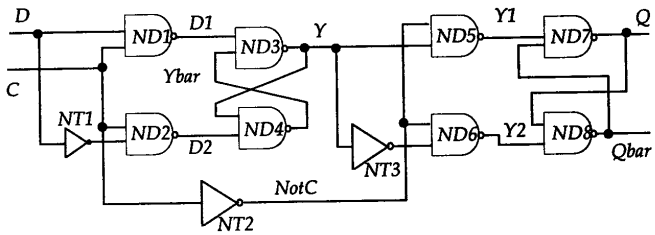

图5-9 主从触发器

5.14 奇偶电路

图5-10所示的9位奇偶发生器门级模型描述如下：

```
module Parity_9_Bit(D, Even,Odd);
  input [0:8] D;
  output Even, Odd;
```

```
  xor # (5,4)
    XE0 (E0,D[0],D[1]),
    XE1 (E1,D[2],D[3]),
    XE2 (E2,D[4],D[5]),
    XE3 (E3,D[6],D[7]),
    XF0 (F0,E0,E1),
    XF1 (F1,E2,E3),
    XH0 (H0,F0,F1),
    XEVEN (Even, D[8], H0);
  not #2
    XODD (Odd, Even);
endmodule
```

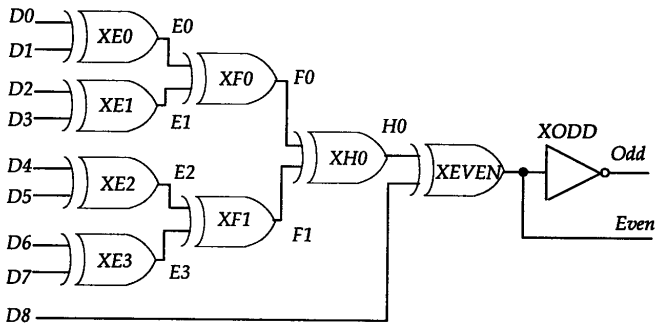

图5-10 奇偶发生器

习题

1. 用基本门描述图 5-11显示的电路模型。编写一个测试验证程序用于测试电路的输出。使用所有可能的输入值对电路进行测试。
2. 使用基本门描述如图 5-12所示的优先编码器电路模型。当所有输入为 0时，输出 *Valid*为 0，否则输出为1。并且为验证优先编码器的模型行为编写测试验证程序。

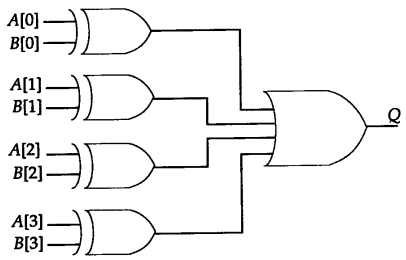

图5-11 A不等于B的逻辑

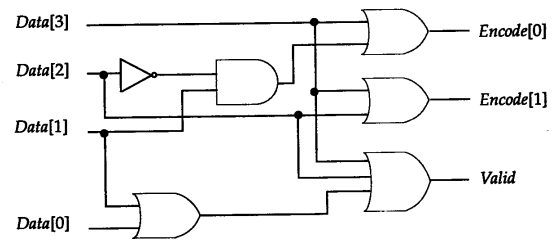

图5-12 优先编码器