# 数字逻辑与处理器基础实验Verilog作业

无42 陈誉博 2014011058

1. 由卡诺图化简结果可知，y=s’\*a+ s\*b。

若不考虑冒险，则代码如下：

module MUX2x1(a,b,s,y);

input a,b,s;

output y;

not #20 fs(F,s);

and #20 u0(T0,F,a),

u1(T1,s,b);

//u2(T2,a,b);

or #20 u3(y,T0,T1/\*,T2\*/);

endmodule;

module test\_MUX2x1;

reg a,b,s;

wire y;

MUX2x1 mu1(.a(a),.b(b),.s(s),.y(y));

initial begin

a<=1;

b<=1;

s<=0;

end

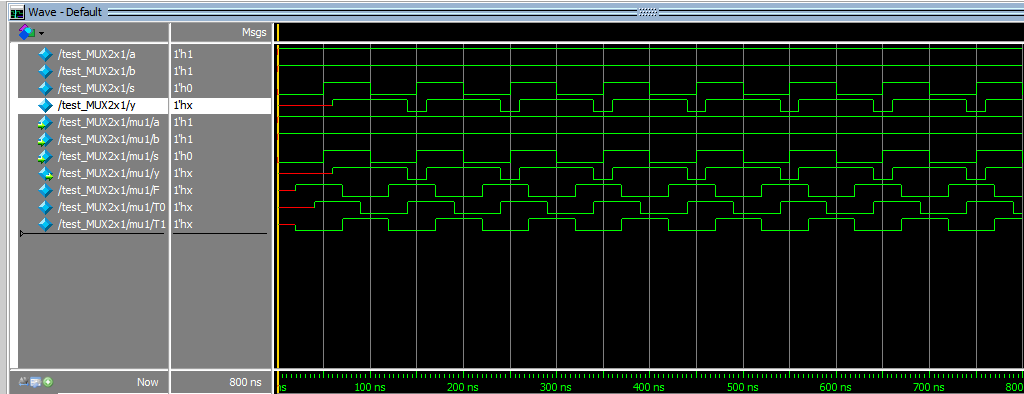
initial begin

forever #50 s<=~s;

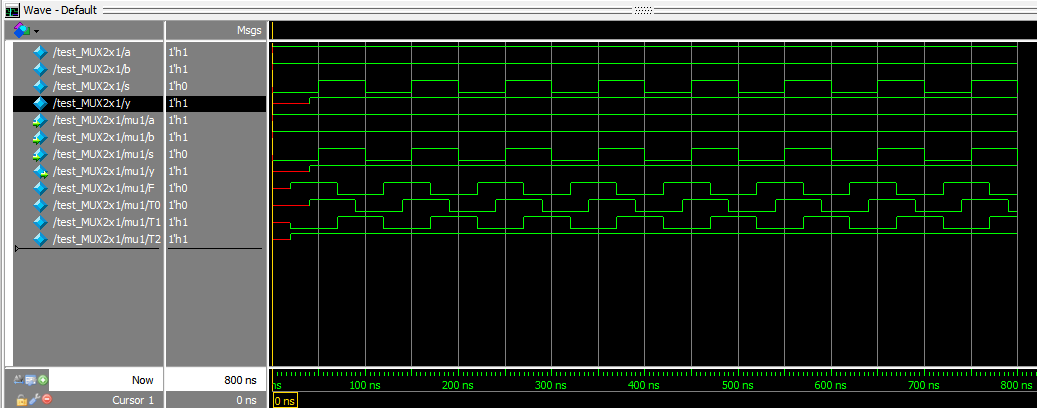
end

endmodule;

此时仿真波形如下：



由波形可以看出，当a=b=1时，输出结果应该为1，但是由于冒险的存在使得输出波形产生了毛刺。所以考虑消除冒险：令y= y=s’\*a+ s\*b+ a\*b。此时仿真波形如下：



由波形可以看出，消除冒险之后输出波形中的毛刺已经消除。

更改测试条件，测试代码如下：

module test\_MUX2x1;

reg a,b,s;

wire y;

MUX2x1 mu1(.a(a),.b(b),.s(s),.y(y));

initial begin

a<=0;

b<=0;

s<=0;

end

initial begin

forever #50 a<=~a;

end

initial begin

forever #100 b<=~b;

end

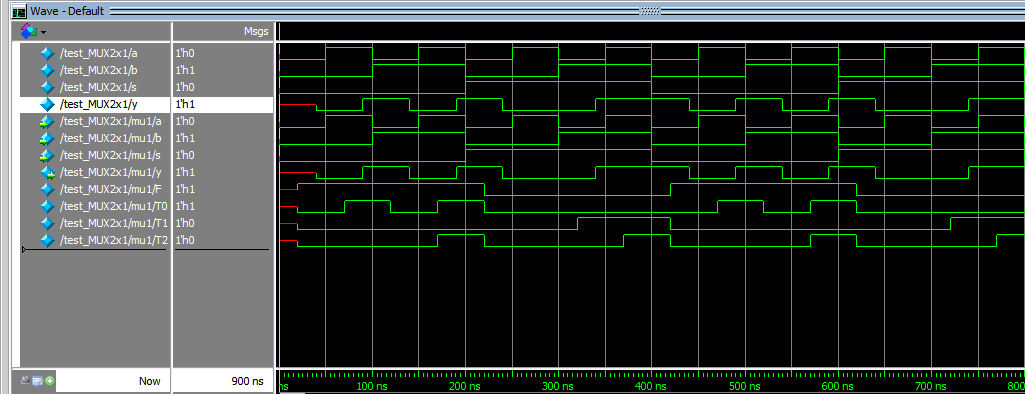
initial begin

forever #200 s<=~s;

end

endmodule;

测试结果的仿真波形如下：



由此2\*1多路选择器可以设计出4\*1的多路选择器。

module MUX2x1(a,b,s,y);

input a,b,s;

output y;

not #5 fs(F,s);

and #5 u0(T0,F,a),

u1(T1,s,b),

u2(T2,a,b);

or #5 u3(y,T0,T1,T2);

endmodule;

module MUX4x1(input[3:0]c,input[1:0]s,output z);

MUX2x1 mu1(.a(c[0]),.b(c[1]),.s(s[0]),.y(T1));

MUX2x1 mu2(.a(c[2]),.b(c[3]),.s(s[0]),.y(T2));

MUX2x1 mu3(.a(T1),.b(T2),.s(s[1]),.y(z));

endmodule;

module test\_MUX4x1;

reg [3:0] c;reg [1:0] s;

wire z;

MUX4x1 MU1(.c(c),.s(s),.z(z));

initial begin

c<=0;

s<=0;

end

initial begin

forever #10 c[0]<=~c[0];

end

initial begin

forever #20 c[1]<=~c[1];

end

initial begin

forever #40 c[2]<=~c[2];

end

initial begin

forever #80 c[3]<=~c[3];

end

initial begin

forever #160 s[0]<=~s[0];

end

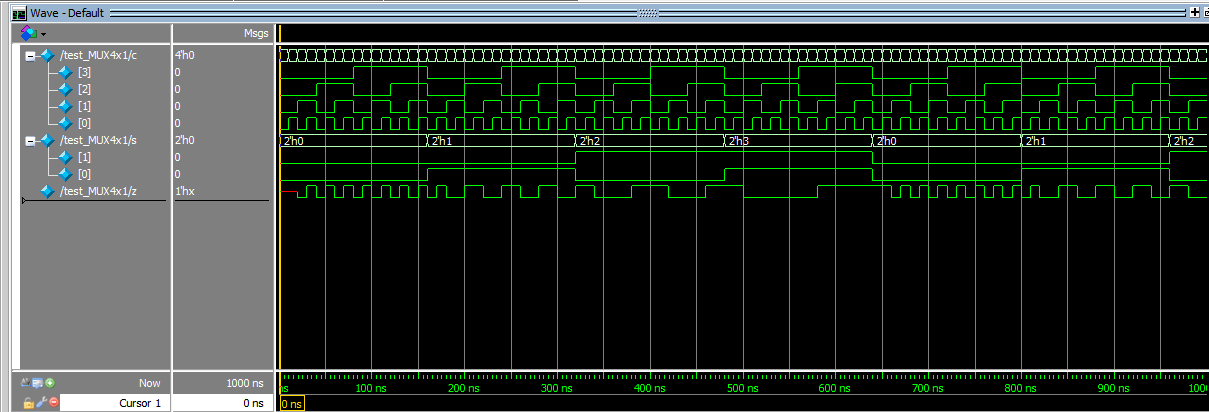
initial begin

forever #320 s[1]<=~s[1];

end

endmodule

仿真波形如下：



1. 电路代码如下：

module Decoder3x8(input[2:0]a,output[7:0]d);

and u0(d[0],~a[2],~a[1],~a[0]),

u1(d[1],~a[2],~a[1],a[0]),

u2(d[2],~a[2],a[1],~a[0]),

u3(d[3],~a[2],a[1],a[0]),

u4(d[4],a[2],~a[1],~a[0]),

u5(d[5],a[2],~a[1],a[0]),

u6(d[6],a[2],a[1],~a[0]),

u7(d[7],a[2],a[1],a[0]);

endmodule

测试代码如下：

module test\_Decoder;

reg [2:0] a;

wire [7:0] d;

Decoder3x8 d1(.a(a),.d(d));

initial begin

a<=0;

end

initial begin

forever #25 a[0]=~a[0];

end

initial begin

forever #50 a[1]=~a[1];

end

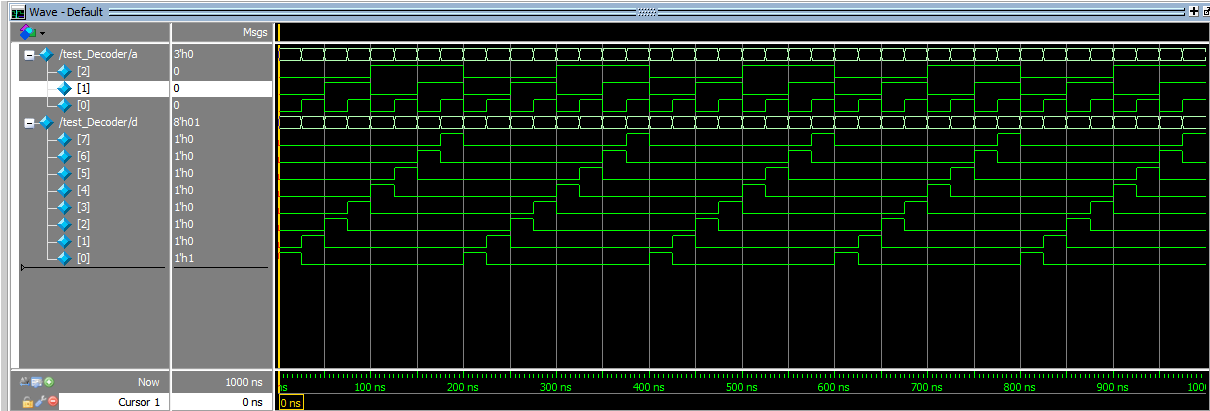
initial begin

forever #100 a[2]=~a[2];

end

endmodule

仿真波形如下：



1. D触发器的电路代码：

module Dtrigger(D,clk,r,s,q,qf);

input D,clk,r,s;

output q,qf;

nand #2 u0(T0,s,T3,T1),

u1(T1,T0,clk,r),

u2(T2,T1,clk,T3),

u3(T3,T2,r,D),

u4(q,s,T1,qf),

u5(qf,q,r,T2);

endmodule

测试代码：

module test\_D;

reg D,clk,r,s;

wire q,qf;

Dtrigger trigger1(.D(D),.clk(clk),.r(r),.s(s),.q(q),.qf(qf));

initial begin

D<=0;

clk<=0;

r<=1;

s<=1;

end

initial begin

forever #53 D=~D;

end

initial begin

forever #30 clk=~clk;

end

initial begin

#112 r<=0;

#20 r<=1;

end

initial begin

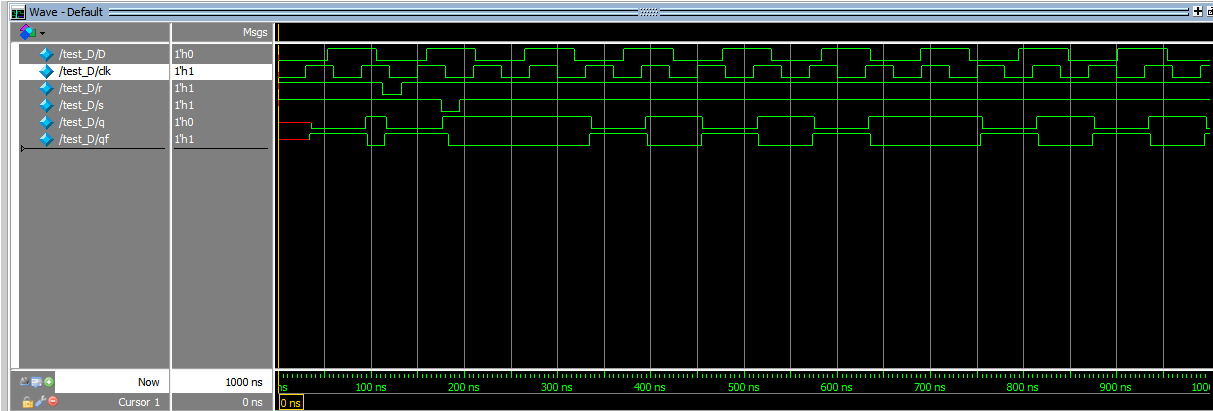
#175 s<=0;

#20 s<=1;

end

endmodule

仿真波形：



二路选择器代码采用第1题的设计（延时设置为2ns，与或非门延时相同）

寄存器的电路代码：

module Register(input ResetB,input Clk,input Load,input [3:0]D,output Q3);

wire Q0,Q1,Q2,QF0,QF1,QF2,QF3;

Dtrigger D0(.D(D[0]),.clk(Clk),.r(ResetB),.s(1),.q(Q0),.qf(QF0));

MUX2x1 mu1(.a(Q0),.b(D[1]),.s(Load),.y(T1));

Dtrigger D1(.D(T1),.clk(Clk),.r(ResetB),.s(1),.q(Q1),.qf(QF1));

MUX2x1 mu2(.a(Q1),.b(D[2]),.s(Load),.y(T2));

Dtrigger D2(.D(T2),.clk(Clk),.r(ResetB),.s(1),.q(Q2),.qf(QF2));

MUX2x1 mu3(.a(Q2),.b(D[3]),.s(Load),.y(T3));

Dtrigger D3(.D(T3),.clk(Clk),.r(ResetB),.s(1),.q(Q3),.qf(QF3));

Endmodule

寄存器的测试代码：

1. **Load=1时，=D3**

module test\_Register;

reg Reset,clk,load;reg [3:0] d;

wire q3;

Register r1(.ResetB(Reset),.Clk(clk),.Load(load),.D(d),.Q3(q3));

initial begin

Reset<=1;

clk<=0;

load<=1;

d<=0;

end

initial begin

forever #33 clk<=~clk;

end

//initial begin

// forever #47 load<=~load;

//end

initial begin

#51 Reset=~Reset;

#22 Reset=~Reset;

end

initial begin

forever #50 d[0]<=~d[0];

end

initial begin

forever #100 d[1]<=~d[1];

end

initial begin

forever #200 d[2]<=~d[2];

end

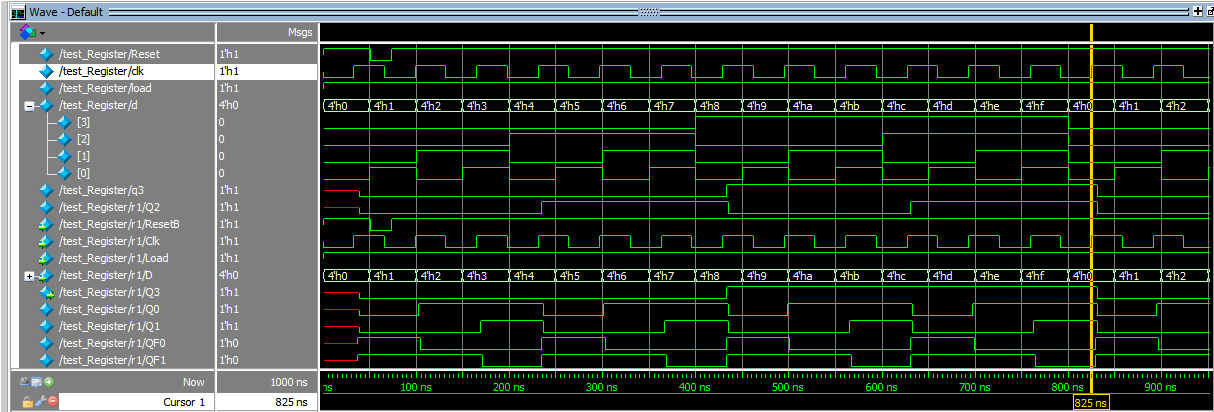
initial begin

forever #400 d[3]<=~d[3];

end

endmodule

仿真波形：



1. **Load=0时，=Q2**

测试代码：

module test\_Register;

reg Reset,clk,load;reg [3:0] d;

wire q3;

Register r1(.ResetB(Reset),.Clk(clk),.Load(load),.D(d),.Q3(q3));

initial begin

Reset<=1;

clk<=0;

load<=0;

d<=0;

end

initial begin

forever #33 clk<=~clk;

end

//initial begin

// forever #47 load<=~load;

//end

initial begin

#51 Reset=~Reset;

#22 Reset=~Reset;

end

initial begin

forever #50 d[0]<=~d[0];

end

initial begin

forever #100 d[1]<=~d[1];

end

initial begin

forever #200 d[2]<=~d[2];

end

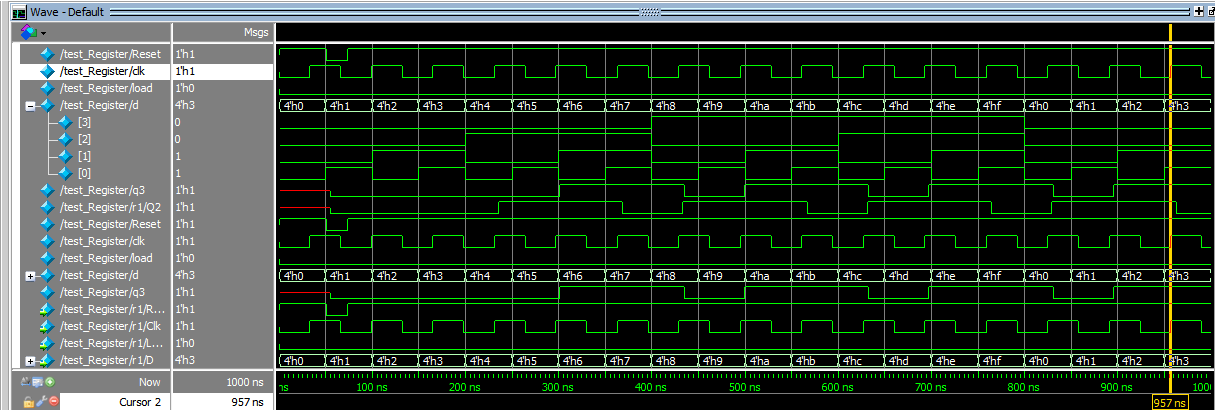
initial begin

forever #400 d[3]<=~d[3];

end

endmodule

仿真波形：



1. **Load为周期信号**

测试代码：

module test\_Register;

reg Reset,clk,load;reg [3:0] d;

wire q3;

Register r1(.ResetB(Reset),.Clk(clk),.Load(load),.D(d),.Q3(q3));

initial begin

Reset<=1;

clk<=0;

load<=0;

d<=0;

end

initial begin

forever #33 clk<=~clk;

end

initial begin

forever #47 load<=~load;

end

initial begin

#51 Reset=~Reset;

#22 Reset=~Reset;

end

initial begin

forever #50 d[0]<=~d[0];

end

initial begin

forever #100 d[1]<=~d[1];

end

initial begin

forever #200 d[2]<=~d[2];

end

initial begin

forever #400 d[3]<=~d[3];

end

endmodule

仿真波形：

