gdut

\_\_ \_计算机 \_\_学院 \_\_ **计算机科学与技术** 专业 \_\_ **1** 班

学号\_\_3119004751\_\_\_\_\_\_姓名 叶嘉轩\_\_\_\_\_ 教师评定\_\_丁磊\_\_\_\_\_\_\_

实验题目\_ 基于Libero的数字逻辑设计仿真及验证实验\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| 序号 | 实验内容 | 完成情况 |
| 6 | 用EDA设计仿真基本门电路并烧录验证 | 已完成 |
| 7 | 用EDA设计仿真组合逻辑电路74HC148、74HC138、74HC153、74HC85、74HC283、74HC4511并烧录验证 | 已完成 |
| 8 | 用EDA设计仿真时序逻辑电路74HC74、74HC112、74HC194并烧录验证 | 已完成 |
| 9 | 综合实验（至少做三道综合题），使用Smartdesign工具来设计 | 已完成 |
| 10 | 大考核：在规定时间内完成老师布置的题目并给老师现场检查 | 已完成 |

**实验报告**

## 基本门电路

一、实验目的

1、了解基于Verilog的基本门电路的设计及其验证。

2、熟悉利用EDA工具进行设计及仿真的流程。

3、学习针对实际门电路芯片74HC00、74HC02、74HC04、74HC08、74HC32、74HC86进行VerilogHDL设计的方法。

4、熟悉实验箱的使用和程序下载（烧录）及测试的方法。

二、实验环境及仪器

1、Libero仿真软件。

2、数字逻辑与系统设计实验箱。

3、Actel A3P060 FPGA芯片及Flash Pro5烧录器。

三、实验内容

1、掌握Libero软件的使用方法。

2、进行针对74系列基本门电路的设计，并完成相应的仿真实验。

3、参考教材中相应章节的设计代码、测试平台代码（可自行编程），完成74HC00、74HC02、74HC04、74HC08、74HC32、74HC86相应的设计、综合及仿真。

4、提交针对74HC00、74HC02、74HC04、74HC08、74HC32、74HC86的综合结果，以及相应的仿真结果。

5、**将各芯片的综合后仿真结果拍照或截图发送至老师的微信**。

四、实验结果和数据处理

1、所有模块及测试平台代码清单

//74HC00代码-与非

module HC00(A,B,Y);

input [3:0]A,B;

output [3:0]Y;

assign Y=~(A&B);//与非

endmodule

//74HC00测试平台代码

// testbench.v

`timescale 1ns/1ns

module testbench();

reg [3:0]a,b;

wire [3:0]y;

HC00 u1(a,b,y);

initial

begin

a=4'b0000;b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

#10 a=4'b1111;b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

end

endmodule

//74HC02代码-或非

module HC02(A,B,Y);

input [4:1]A,B;

output [4:1]Y;

assign Y=~(A|B);//或非

endmodule

//74HC02测试平台代码

`timescale 1ns/1ns

module test02();

reg [4:1]a,b;

wire [4:1]y;

HC02 u2(a,b,y);

initial

begin

a=4'b0000; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

#10

a=4'b1111; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

end

endmodule

//74HC04代码-非

module HC04(A,Y);

input [6:1]A;

output [6:1]Y;

assign Y=~A;//非

endmodule

//74HC04测试平台代码

`timescale 1ns/1ns

module test04();

reg [6:1]a;

wire [6:1]y;

HC04 u4(a,y);

initial

begin

a=6'b000001;

#10 a=a<<1;

#10 a=a<<1;

#10 a=a<<1;

#10 a=a<<1;

#10 a=a<<1;

end

endmodule

//74HC08代码-与

module HC08(A,B,Y);

input [4:1]A,B;

output [4:1]Y;

assign Y=A&B;//与

endmodule

//74HC08测试平台代码

`timescale 1ns/1ns

module test08();

reg [4:1]a,b;

wire [4:1]y;

HC08 u8(a,b,y);

initial

begin

a=4'b0000; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

#10

a=4'b1111; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

end

endmodule

//74HC32代码-或

module HC32(A,B,Y);

input [4:1]A,B;

output [4:1]Y;

assign Y=A|B;//或

endmodule

//74HC32测试平台代码

`timescale 1ns/1ns

module test32();

reg [4:1]a,b;

wire [4:1]y;

HC32 u32(a,b,y);

initial

begin

a=4'b0000; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

#10

a=4'b1111; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

end

endmodule

//74HC86代码-异或

module HC86(A,B,Y);

input [4:1]A,B;

output [4:1]Y;

assign Y=A&(~B)|(~A&B);//异或

endmodule

//74HC86测试平台代码

`timescale 1ns/1ns

module test86();

reg[4:1]a,b;

wire [4:1]y;

HC86 u86(a,b,y)

initial

begin

a=4'b0000; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

#10 b=b<<1;

#10

a=4'b1111; b=4'b0001;

#10 b=b<<1;

#10 b=b<<1;

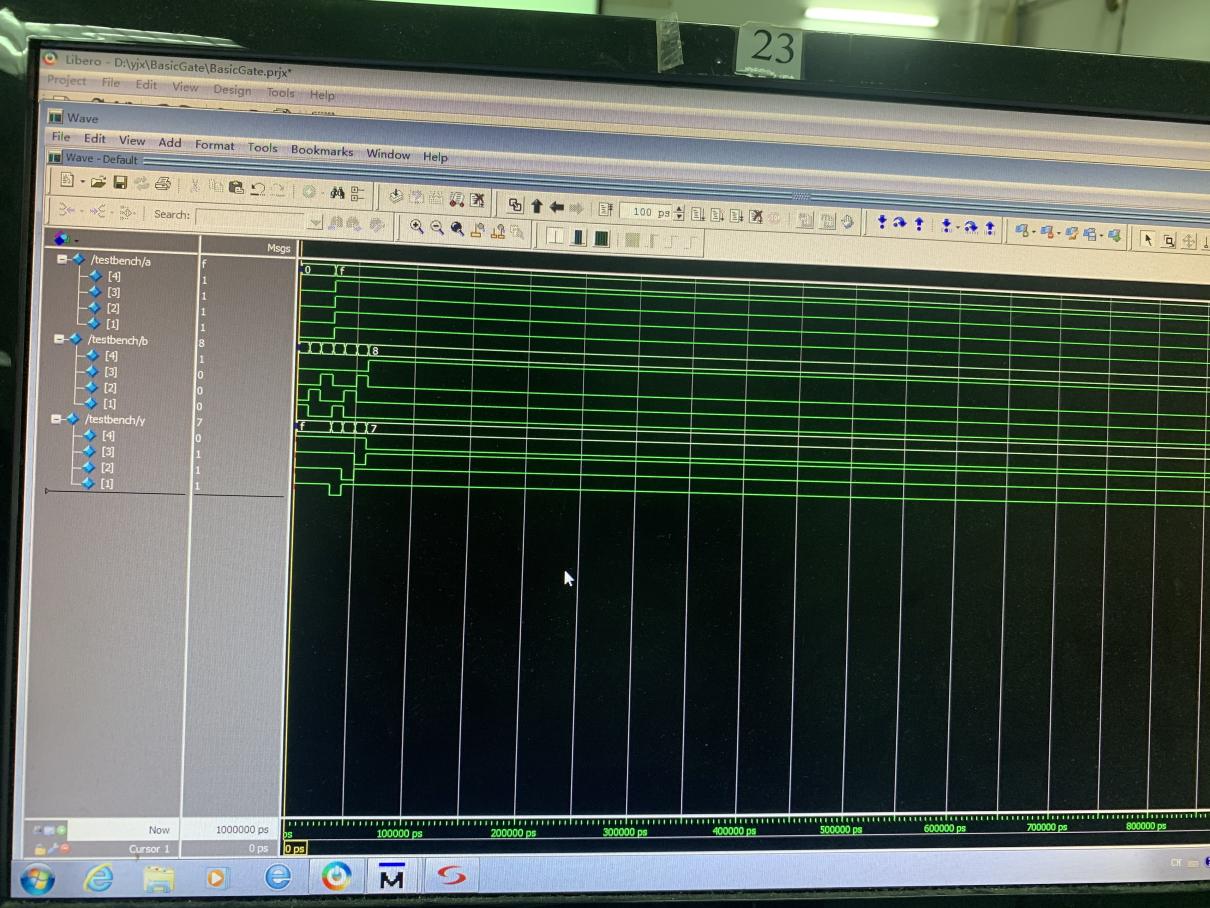
#10 b=b<<1;

end

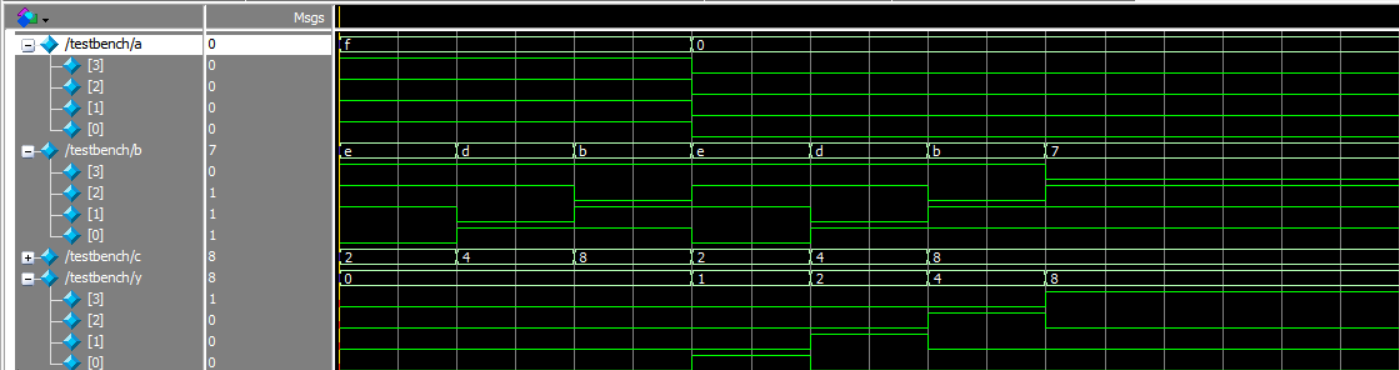
endmodule

1. 第一次仿真结果（**截图，注明对应的门电路**）。（将波形窗口背景设为**白色**，调整窗口至合适大小，使波形能完整显示，对窗口**截图**。后面实验中的仿真使用相同方法处理）

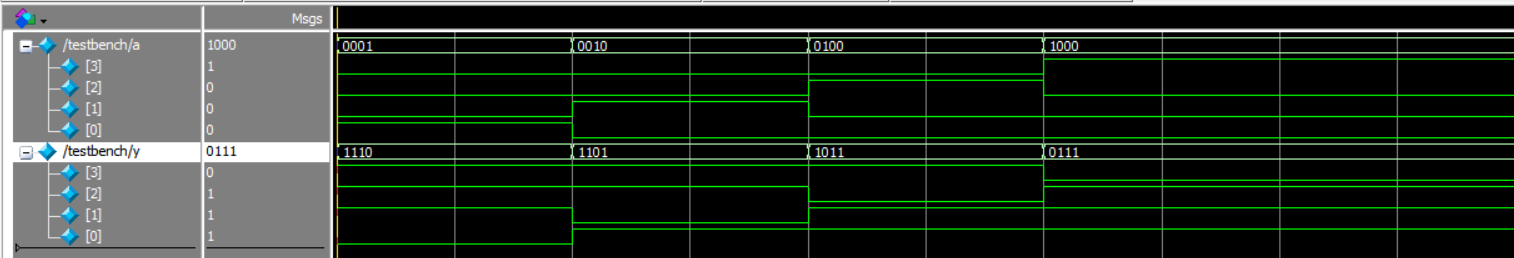
74HC00



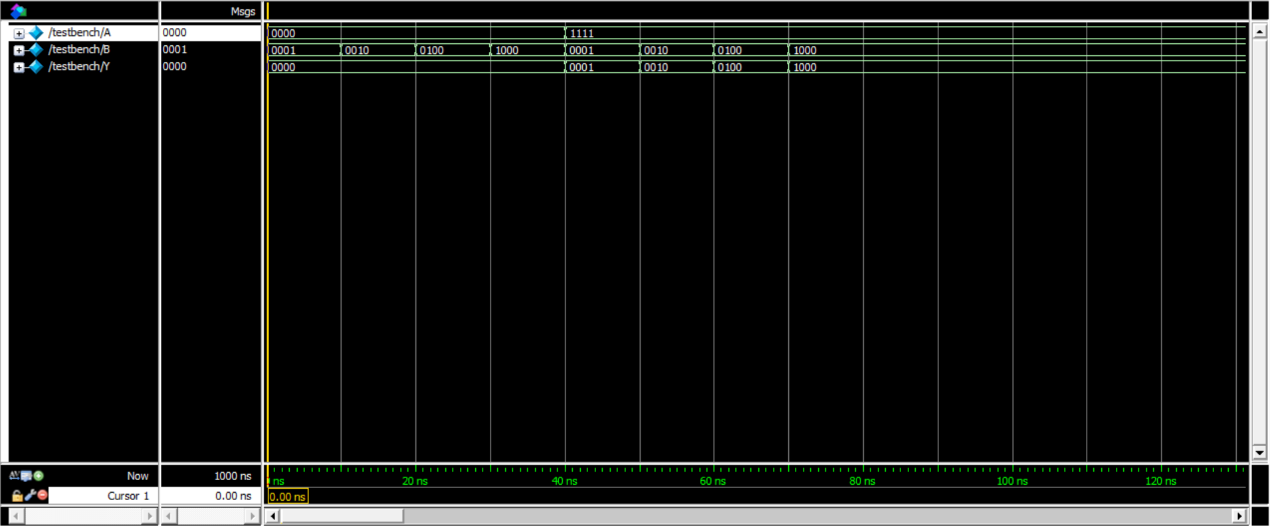
74HC02



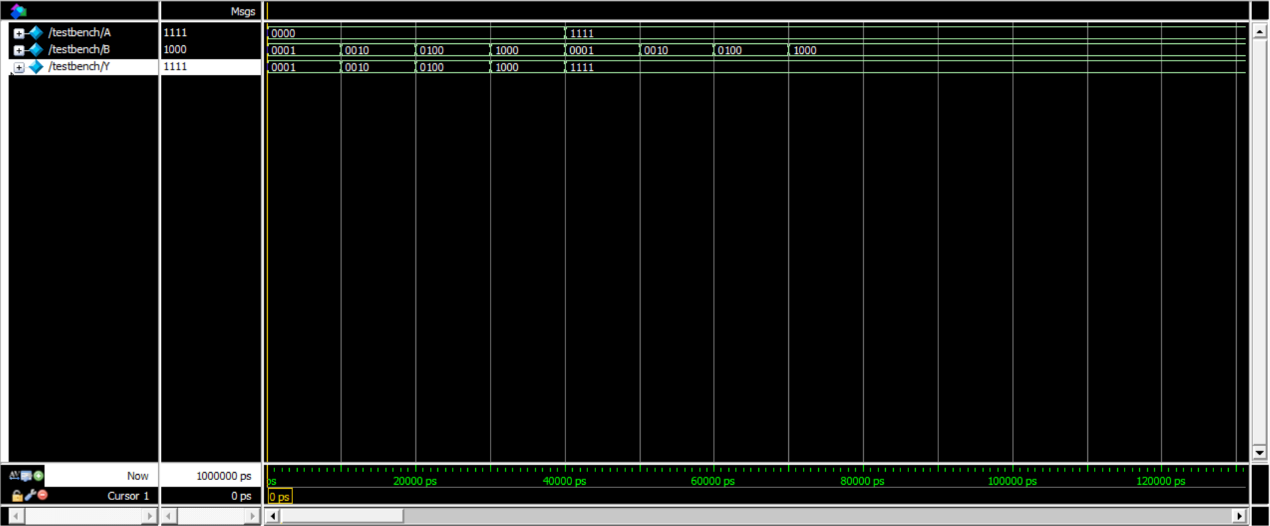
74HC04



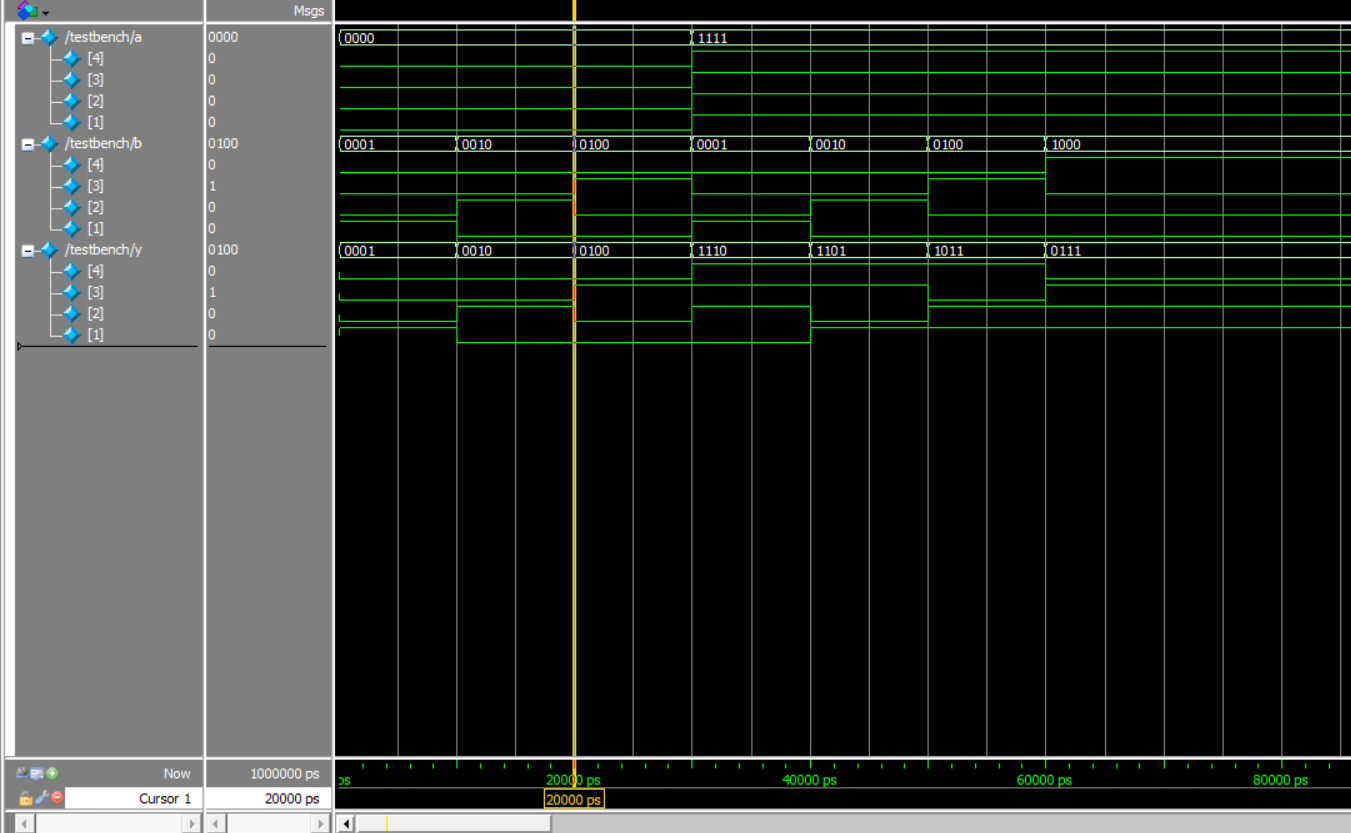
74HC08



74HC32

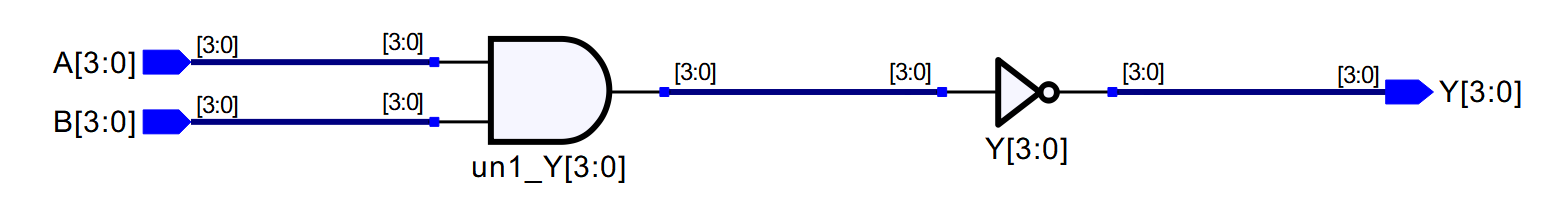


74HC86

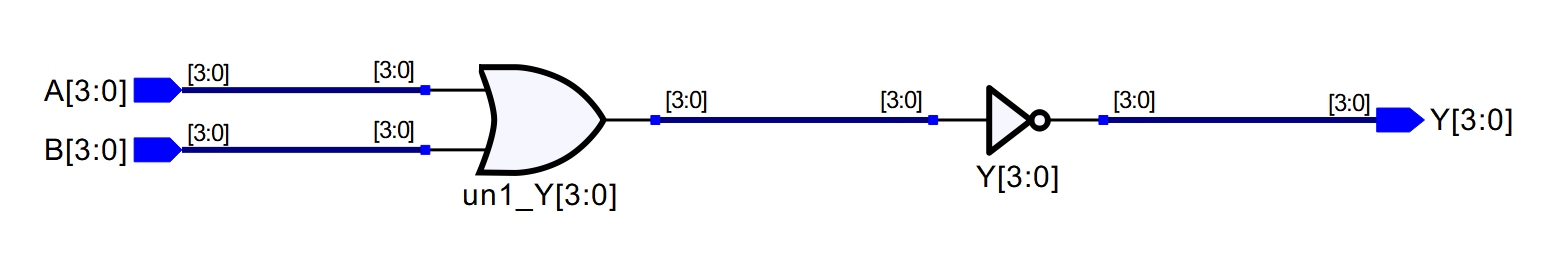


1. 综合结果（**截图，注明对应的门电路**）。（将相关窗口调至合适大小，使RTL图能完整显示，对窗口截图，后面实验中的综合使用相同方法处理）

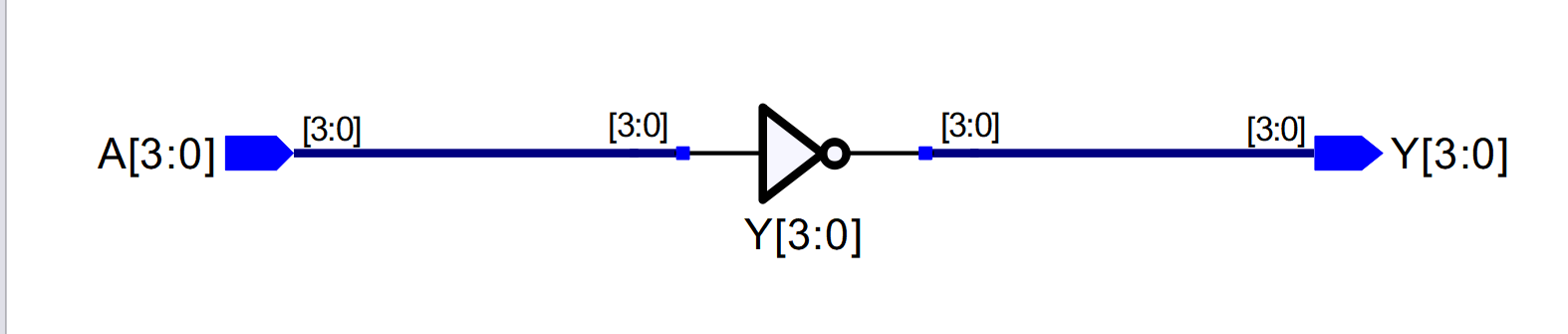
74HC00



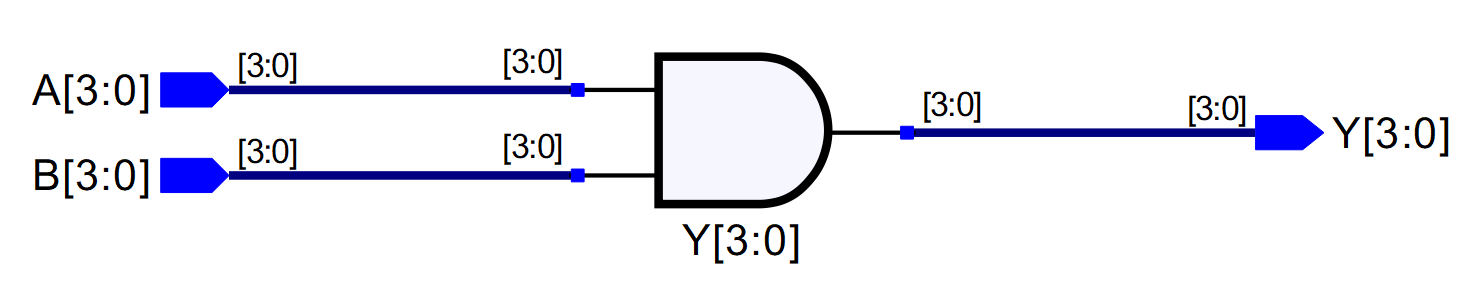
74HC02



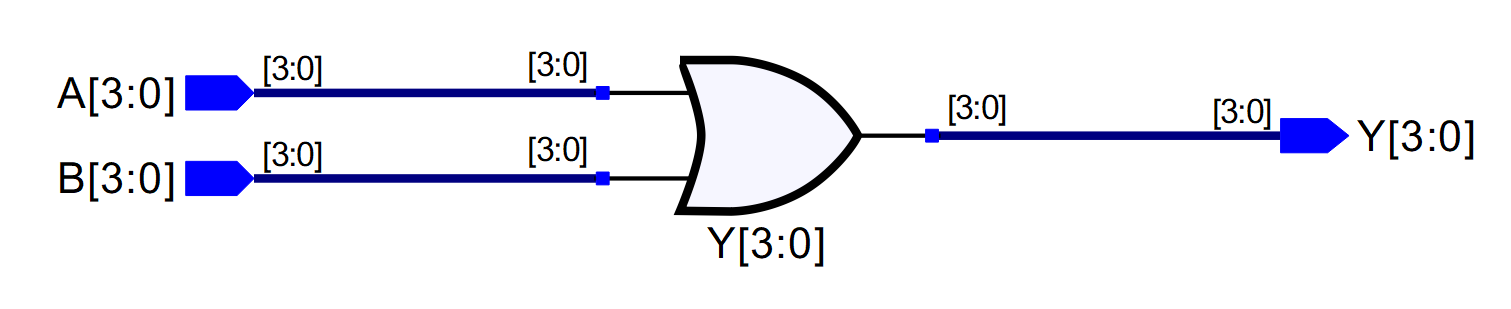
74HC04



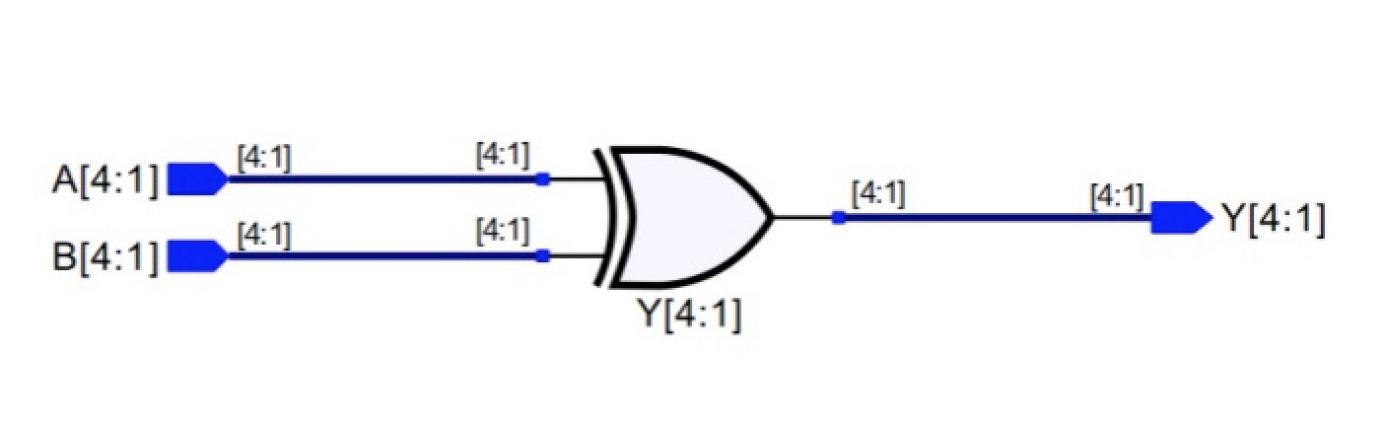
74HC08



74HC32

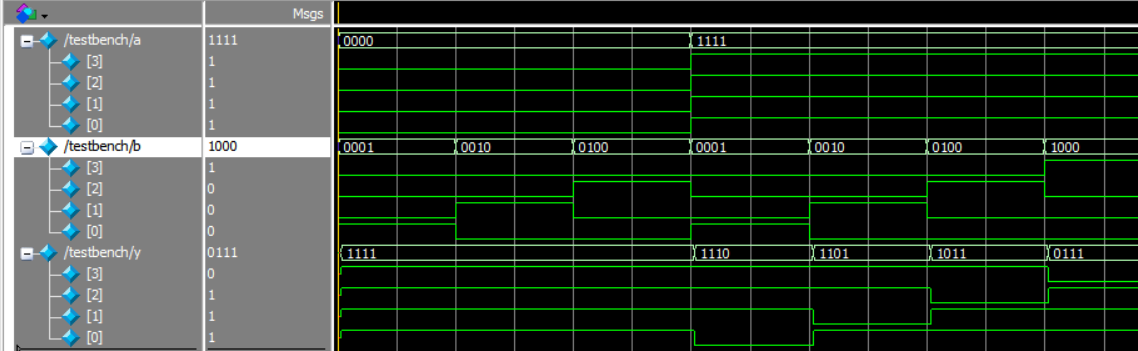


74HC86

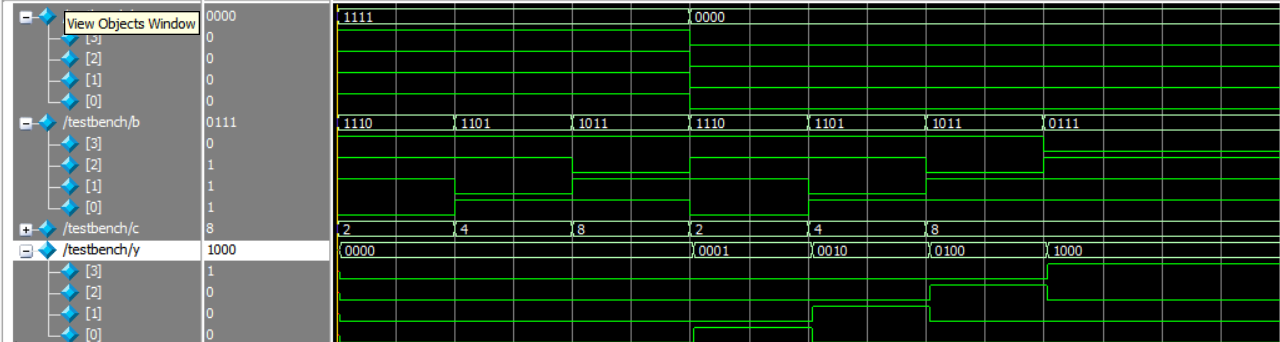


4、第二次仿真结果（综合后）（**截图，注明对应的门电路**）。回答输出信号是否有延迟，延迟时间约为多少？

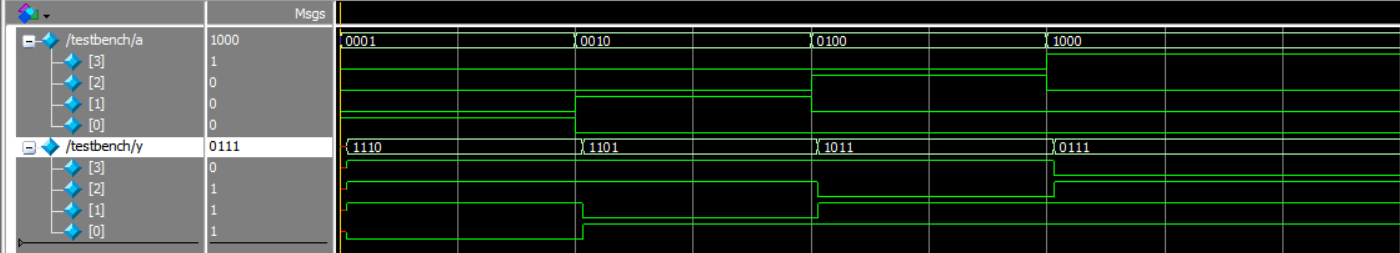
74HC00，有延迟，大概200ps



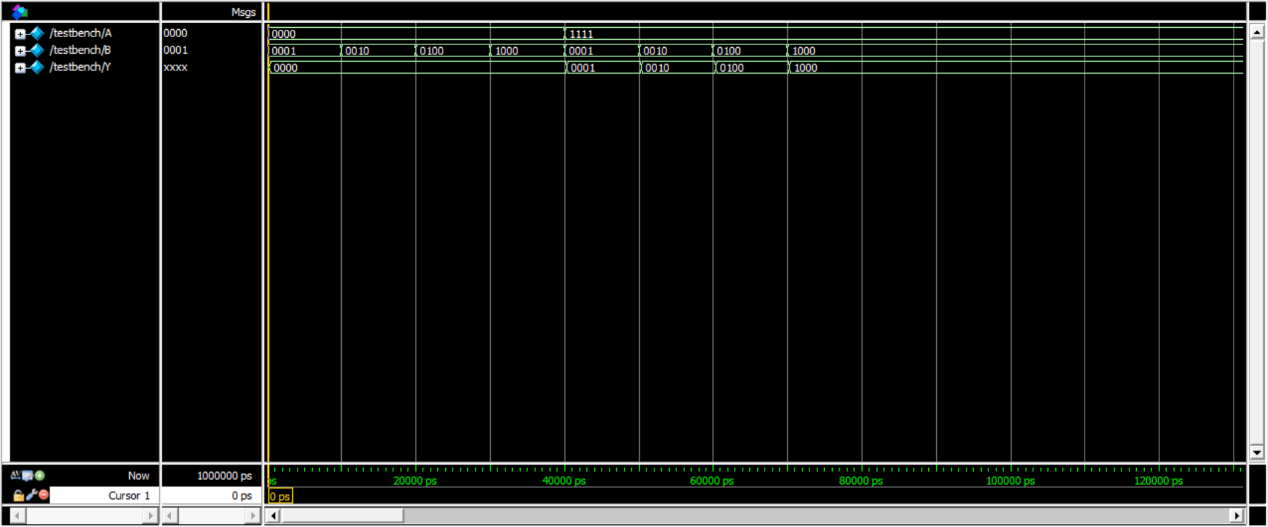
74HC02，有延迟，大概200ps



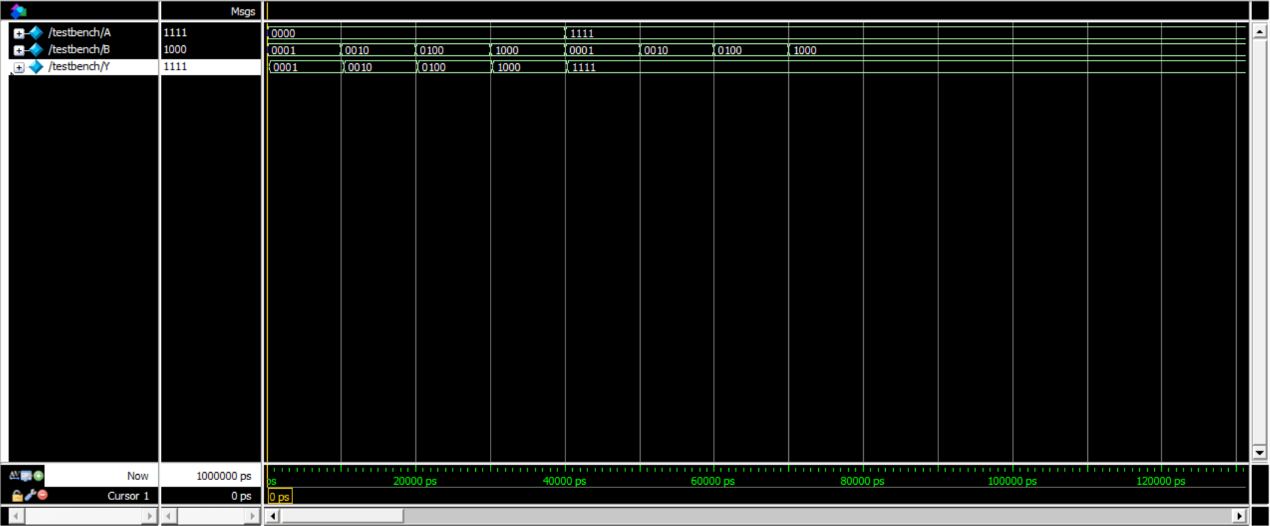
74HC04，有延迟，大概200ps



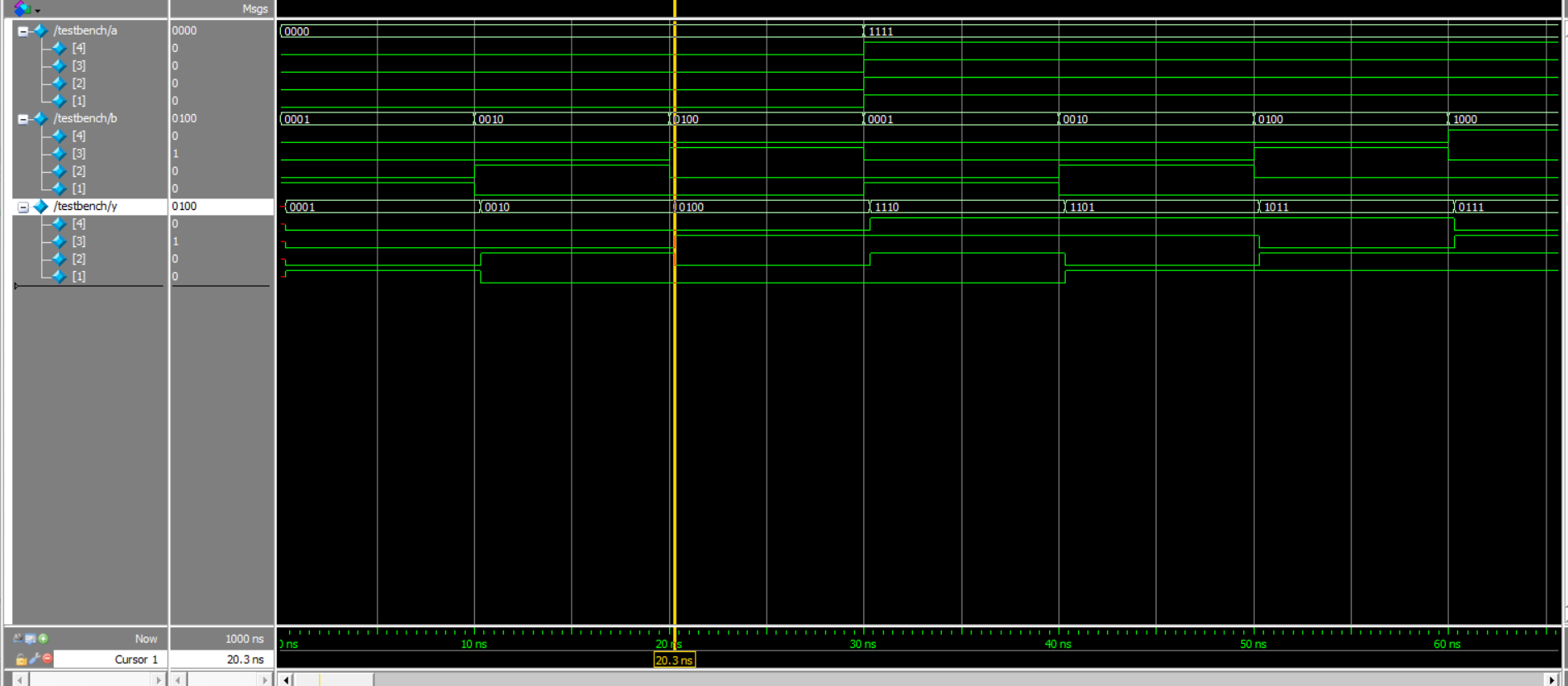
74HC08，有延迟，大概200ps



74HC32，有延迟，大概200ps

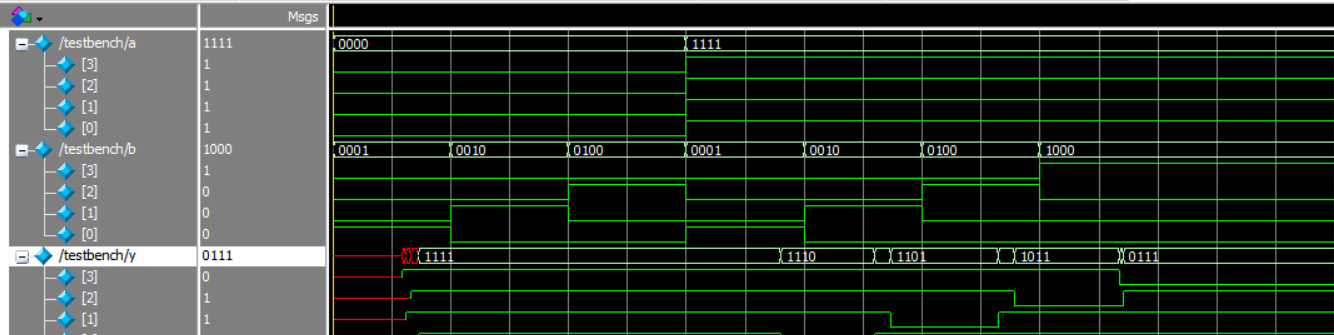


74HC86，有延迟，大概200ps

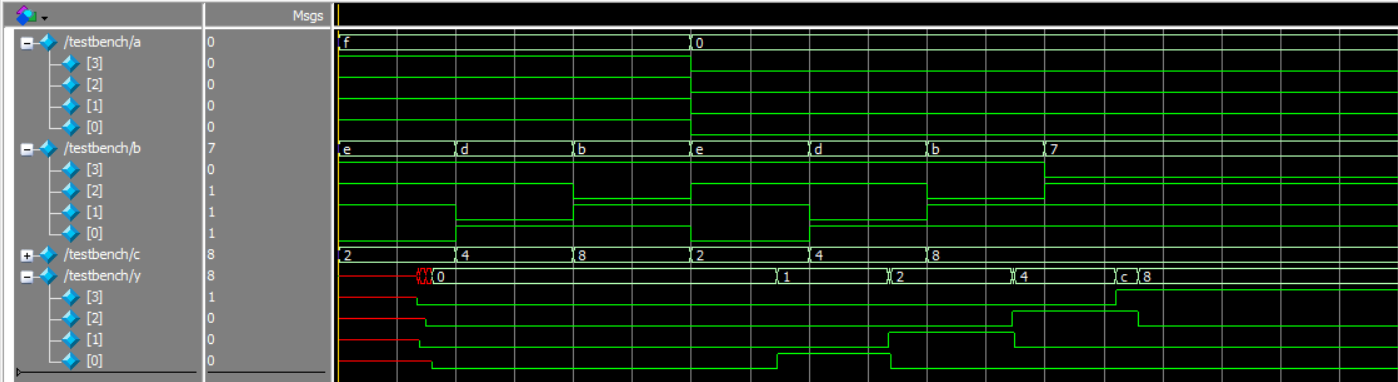


1. 第三次仿真结果（布局布线后）（**截图，注明对应的门电路**）。回答输出信号是否有延迟，延迟时间约为多少？分析是否有出现竞争冒险。

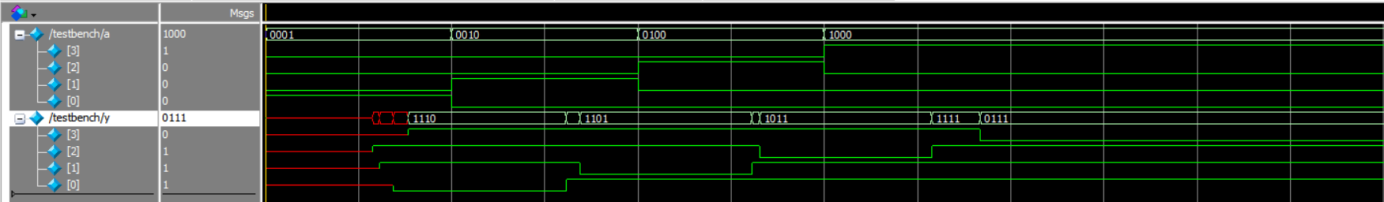
74HC00，有延迟大概7840ps，无竞争冒险



74HC02，有延迟，大概7220ps，无竞争冒险



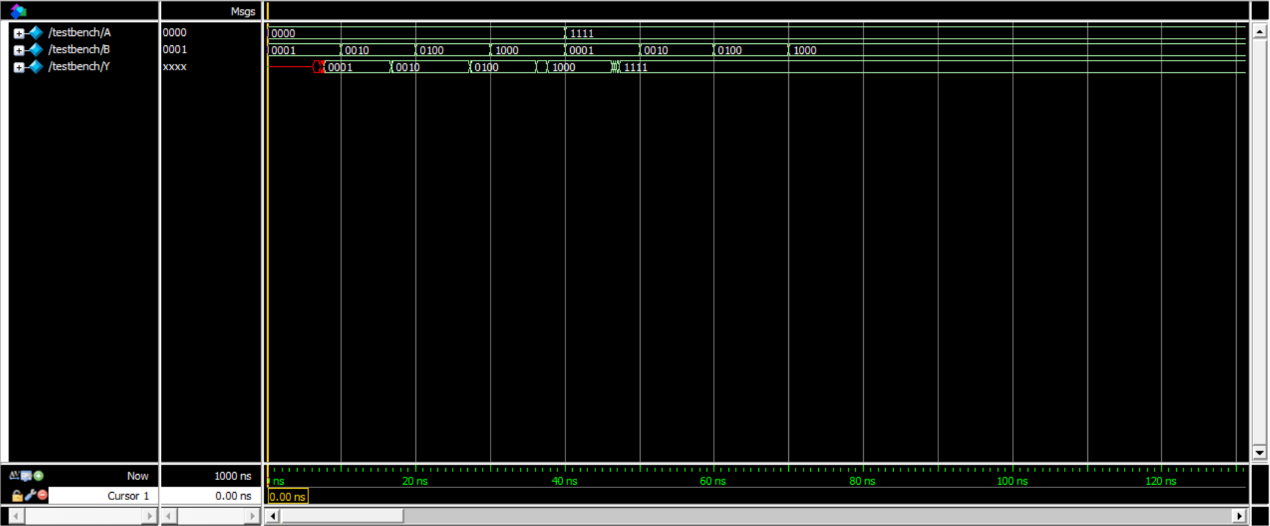
74HC04，有延迟，大概6870ps，无竞争冒险



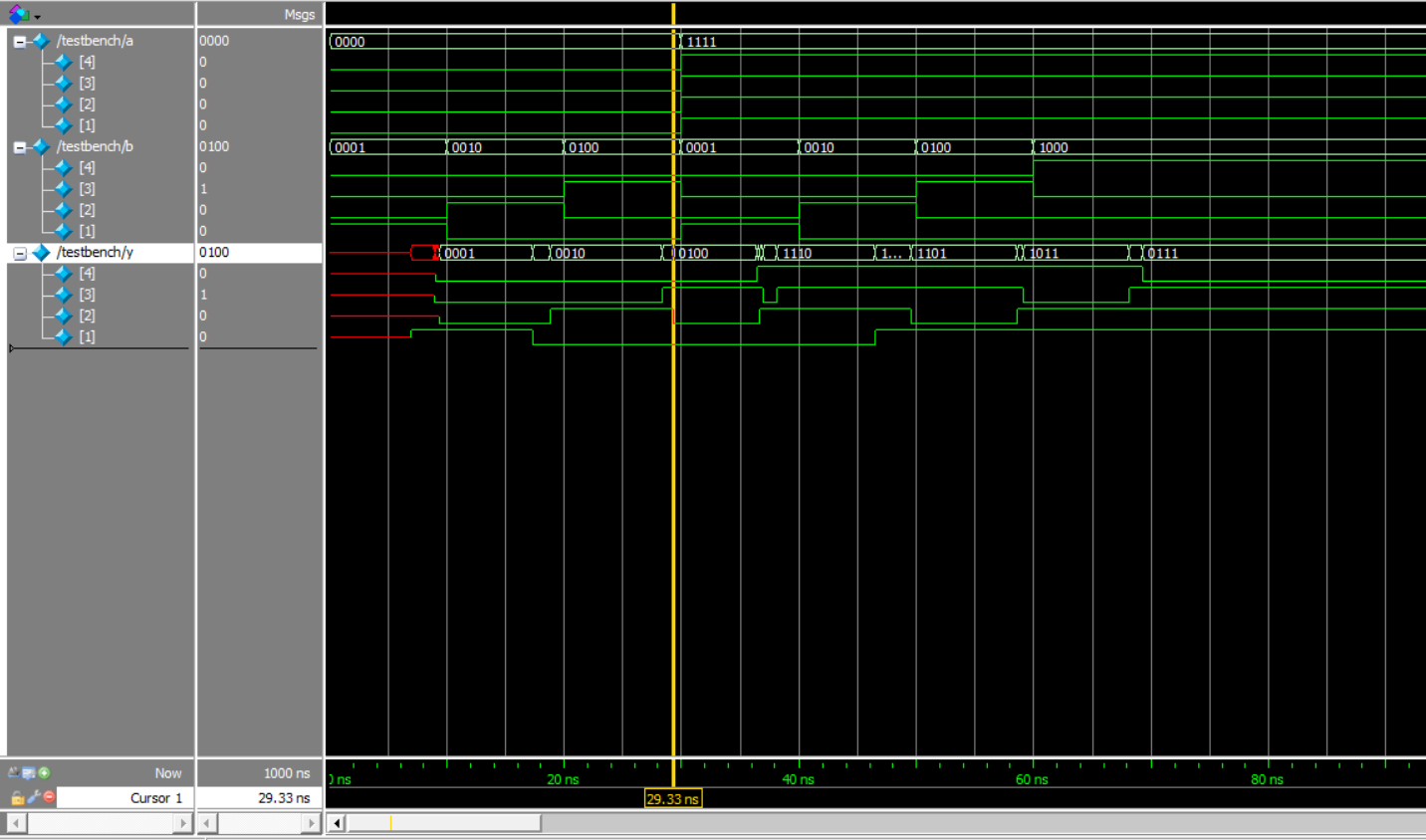
74HC08，有延迟，大概7520ps，无竞争冒险



74HC32，有延迟，大概7220ps，无竞争冒险



74HC86，有延迟，大概9320ps，无竞争冒险

组合逻辑电路

一、实验目的

1、了解基于Verilog的组合逻辑电路的设计及其验证。

2、熟悉利用EDA工具进行设计及仿真的流程。

3、熟悉实验箱的使用和程序下载（烧录）及测试的方法。

4、学习针对实际组合逻辑电路芯片74HC148、74HC138、74HC153、74HC85、74HC283、74HC4511进行VerilogHDL设计的方法。

二、实验环境及仪器

1、Libero仿真软件。

2、数字逻辑与系统设计实验箱。

3、Actel A3P060 FPGA芯片及Flash Pro5烧录器。

三、实验内容

1、掌握Libero软件的使用方法。

2、进行针对74系列基本组合逻辑电路的设计，并完成相应的仿真实验。

3、参考教材中相应章节的设计代码、测试平台代码（可自行编程），完成74HC148、74HC138、74HC153、74HC85、74HC283、74HC4511相应的设计、综合及仿真。

4、74HC85测试平台的测试数据要求：进行比较的A、B两数，分别为本人学号的末两位，如“89”，则A数为“1000”，B数为“1001”。若两数相等，需考虑级联输入（级联输入的各种取值情况均需包括）；若两数不等，则需增加一对取值情况，验证A、B相等时的比较结果。

5、74HC4511设计成扩展型的，即能显示数字0~9、字母a~f（此部分暂时不用烧录）。

6、提交针对74HC148、74HC138、74HC153、74HC85、74HC283、74HC4511的综合结果，以及相应的仿真结果。

7、**将引脚分配图及烧录成功界面拍照或截图发送至老师的微信**。

四、实验结果和数据处理

1、所有模块及测试平台代码清单

//74HC148代码

module HC148(EI,In,Out,EO,GS);

input [7:0]In;

input EI;

output reg EO,GS;

output reg [2:0]Out;

interger I;

always@(EI or In)

if(EI)

begin

Out=3’b111;EO=1;GS=1;

end

else

if(In==8’b11111111)

begin

Out=3’b111;EO=0;GS=1;

end

else

begin

for(I=0;I<8;I=I+1)

begin

if(~In[I])

begin

Out=~I;

EO=1;

GS=0;

end

end

end

endmodule

//74HC148测试平台代码

`timescale 1ns/1ns

module test148();

reg ei;

reg [7:0]turn;

wire [7:0] in=~turn;

wire [2:0] out;

wire eo,gs;

HC148 u(ei,in,out,eo,gs);

initial

begin

ei=1;turn=8’b1;

repeat(8)

#10 turn=turn<<1;

ei=0;turn=8’b1;

repeat(8)

#10 turn=turn<<1;

end

endmodule

//74HC138代码

module HC138(A,En,Y);

input [2:0]A;

input [3:1]En;

output [7:0]Y;

reg [7:0]Y;

wire [2:0]A;

integer I;

always @ (A or En)

begin

if(En[1]==1||En[2]==1||En[3]==0)

Y=8'b11111111;

else

for(I=0;I<=7;I=I+1)

if(A==I)

Y[I]=0;

else

Y[I]=1;

end

endmodule

//74HC138测试平台代码

`timescale 1ns/10ps

module test138();

reg [2:0]a;

reg [3:1]en;

wire [7:0]y;

HC138 u1(a,en,y);

initial

begin

en[1]=1;en[2]=0;en[3]=1;

#10 en[1]=0;en[2]=1;

#10 en[2]=0;en[3]=0;

#10 en[3]=1;a=3'b000;

#10 a=3'b001;

#10 a=3'b010;

#10 a=3'b011;

#10 a=3'b100;

#10 a=3'b101;

#10 a=3'b110;

#10 a=3'b111;

end

endmodule

//74HC153代码

module HC153(D0,D1,D2,D3,Sel0,Sel1,Result);

input D0,D1,D2,D3;

input Sel0,Sel1;

output Result;

reg Result;

always @ (D0 or D1 or D2 or D3 or Sel1 or Sel0)

begin

case ({Sel1,Sel0})

0:Result=D0;

1:Result=D1;

2:Result=D2;

3:Result=D3;

default:Result=1'bx;

endcase

end

endmodule

//74HC153测试平台代码

`timescale 1ns/10ps

module test153();

reg D0,D1,D2,D3,Sel1,Sel0;

wire Result;

HC153 u1(D0,D1,D2,D3,Sel0,Sel1,Result);

initial

begin

D0=0;D1=0;D2=0;D3=0;Sel1=0;Sel0=0;

#100 D0=1;D1=0;D2=0;D3=1;

#100 Sel1=0;Sel0=1;

#100 Sel1=1;Sel0=0;

#100 Sel1=1;Sel0=1;

#100;

end

endmodule

//74HC85代码

module HC85(A,B,I,Q);

input [3:0]A,B;

input [2:0]I;

output reg [2:0]Q;

always@(A,B)

begin

if(A>B) Q=3’b100;

else if(A<B) Q=3’b001;

else if(I[1]==1) Q=3’b010;

else if(I==3’b101) Q=3’b000;

else if(I==0) Q=3’b101;

else Q=I;

end

endmodule

//74HC85测试平台代码

`timescale 1ns/10ps

module test85();

reg [3:0] a,b;

reg [2:0]i;

wire [2:0]q;

HC85 u1(a,b,I,q);

initial

begin

i=0;

repeat(4)

#10 i=$random;

end

initial

begin

a=3;b=1;

#10 a=1;b=4;

#10 a=0;b=0;

#10 a=6;b=1;

#10 a=5;b=9;

end

endmodule

//74HC283代码

module HC283(DataA,DataB,Cin,Sum,Cout);

input [3:0]DataA,DataB;

input Cin;

output [3:0]Sum;

output Cout;

reg [3:0]Sum;

reg Cout;

always @ (DataA or DataB or Cin)

begin

{Cout,Sum}=DataA+DataB+Cin;

end

endmodule

//74HC283测试平台代码

`timescale 1ns/10ps

module test283();

reg [3:0]ina,inb;

reg cin;

wire [3:0]sum;

wire cout;

HC283 u1(ina,inb,cin,sum,cout);

initial

begin

ina=0;

repeat(20)

#20 ina=$random;

end

initial

begin

inb=0;

repeat(10)

#40 inb=$random;

end

initial

begin

cin=0;

#200 cin=1;

end

endmodule

//74HC4511代码

module HC4511(A,Seg,LT\_N,BI\_N,LE);

input LT\_N,BI\_N,LE;

input [3:0]A;

output [7:0]Seg;

reg [7:0]SM\_8S;

assign Seg=SM\_8S;

always @ (A or LT\_N or BI\_N or LE)

begin

if(!LT\_N)

SM\_8S=8'b11111111;

else if(!BI\_N)

SM\_8S=8'b00000000;

else if(LE)

SM\_8S=SM\_8S;

else

case(A)

4'd0:SM\_8S=8'b00111111;

4'd1:SM\_8S=8'b00000110;

4'd2:SM\_8S=8'b01011011;

4'd3:SM\_8S=8'b01001111;

4'd4:SM\_8S=8'b01100110;

4'd5:SM\_8S=8'b01101101;

4'd6:SM\_8S=8'b01111101;

4'd7:SM\_8S=8'b00000111;

4'd8:SM\_8S=8'b01111111;

4'd9:SM\_8S=8'b01101111;

4'd10:SM\_8S=8'b01110111;

4'd11:SM\_8S=8'b01111100;

4'd12:SM\_8S=8'b00111001;

4'd13:SM\_8S=8'b01011100;

4'd14:SM\_8S=8'b01111001;

4'd15:SM\_8S=8'b01110001;

default:;

endcase

end

endmodule

//74HC4511测试平台代码

`timescale 1ns/10ps

module test4511;

reg [3:0]a;

reg lt\_n,bi\_n,le;

wire [7:0]seg;

HC4511 u4511(a,seg,lt\_n,bi\_n,le);

initial

begin

a=0;lt\_n=1;bi\_n=1;le=0;

#30 a=4’b0001;

#30 a=4’b1000;

#30 a=4’b0111;

#30 a=4’b1010;

#30 a=4’b0101;s

#30 le=1;

#30 bi\_n=0;

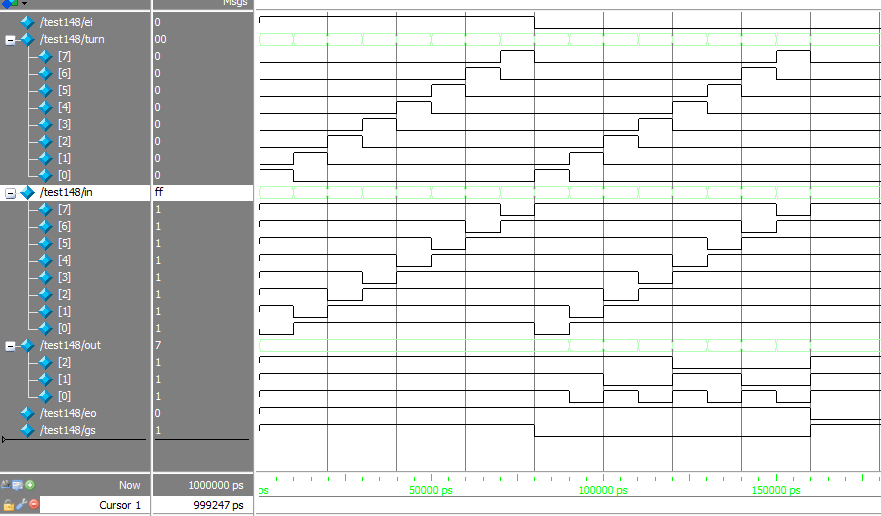
#30 lt\_n=0;

#20;

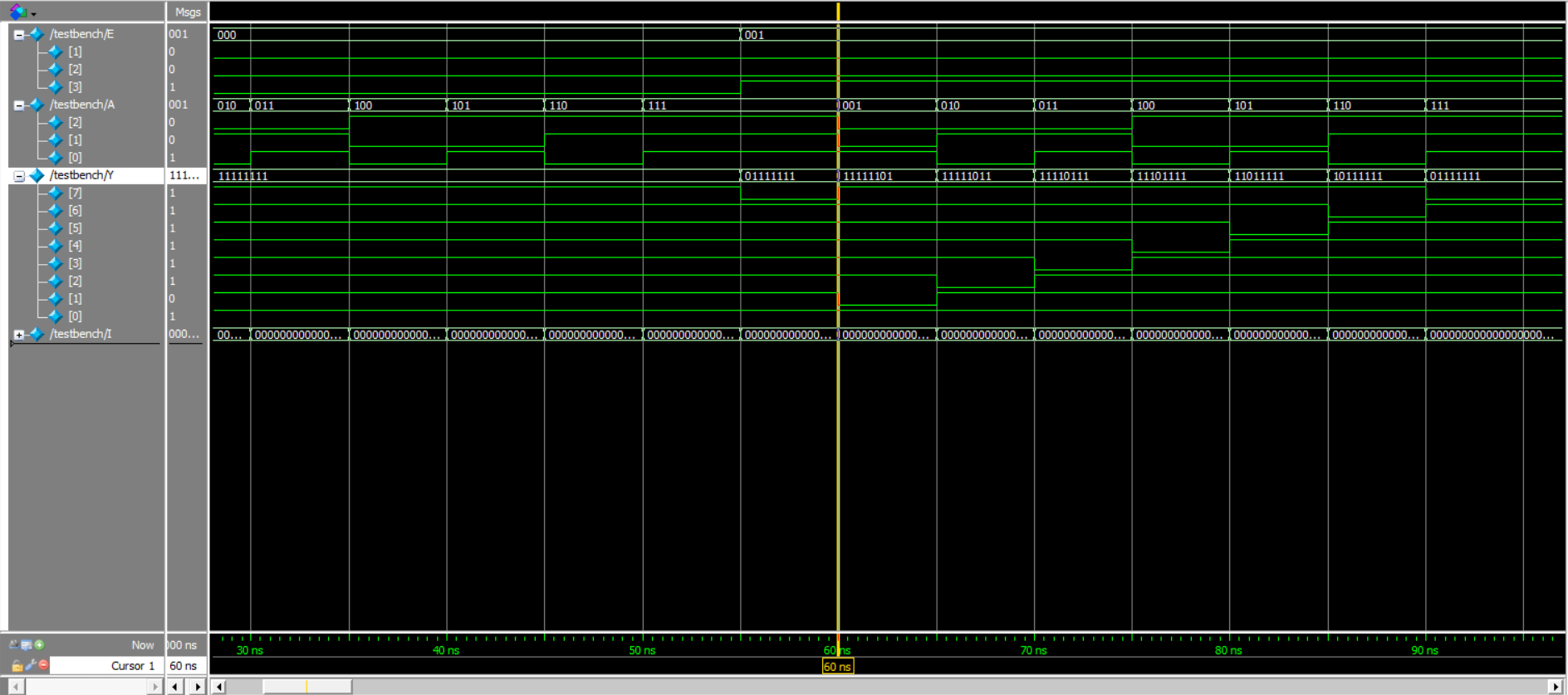
end

endmodule

2、第一次仿真结果（**截图，注明对应的模块**）



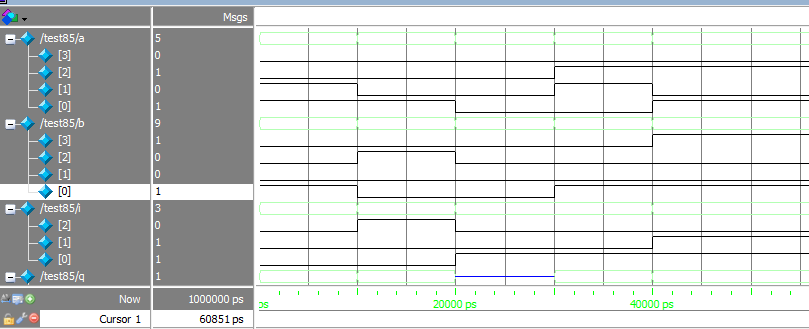
74HC148



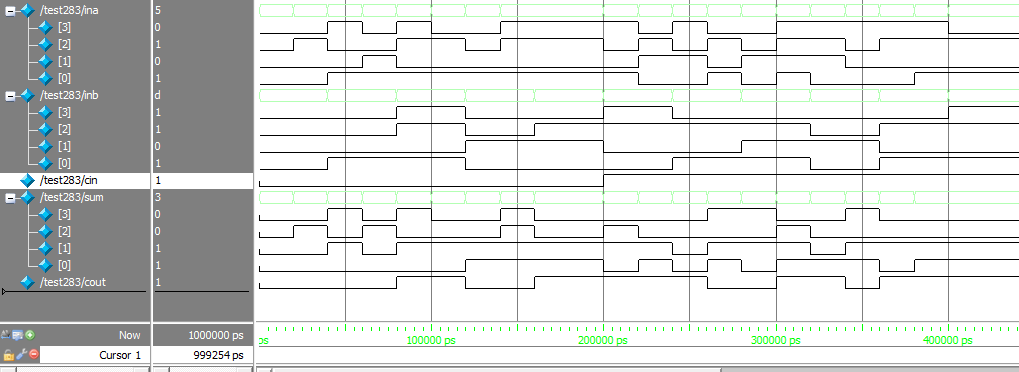
74HC138



74HC153



74HC85

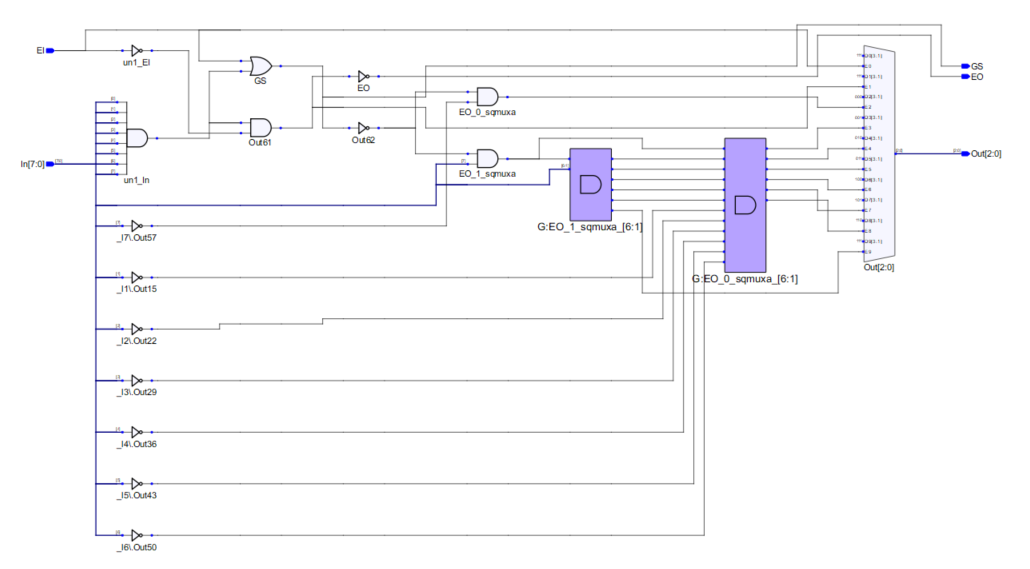


74HC283

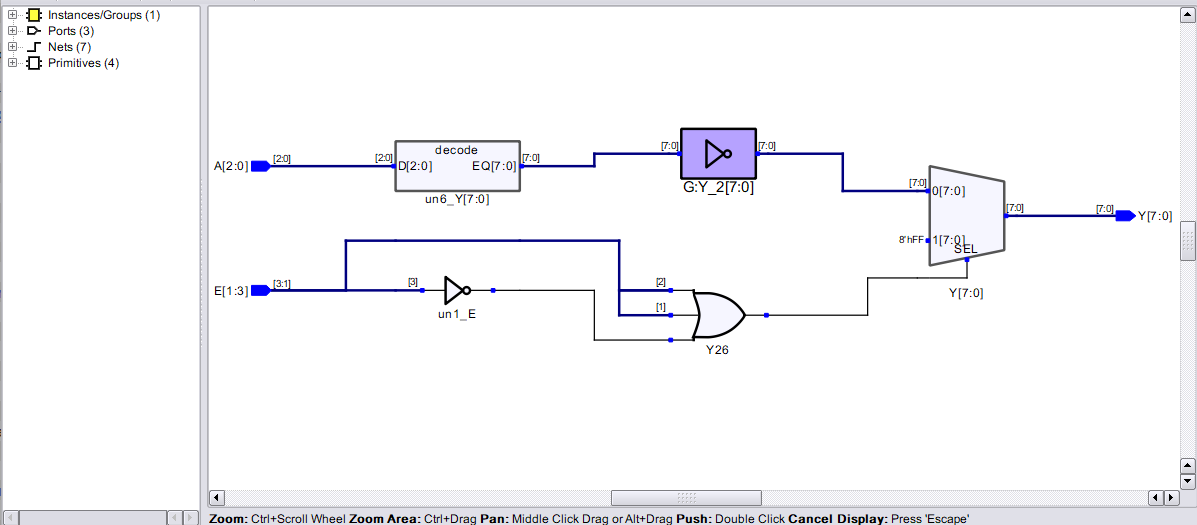


74HC4511

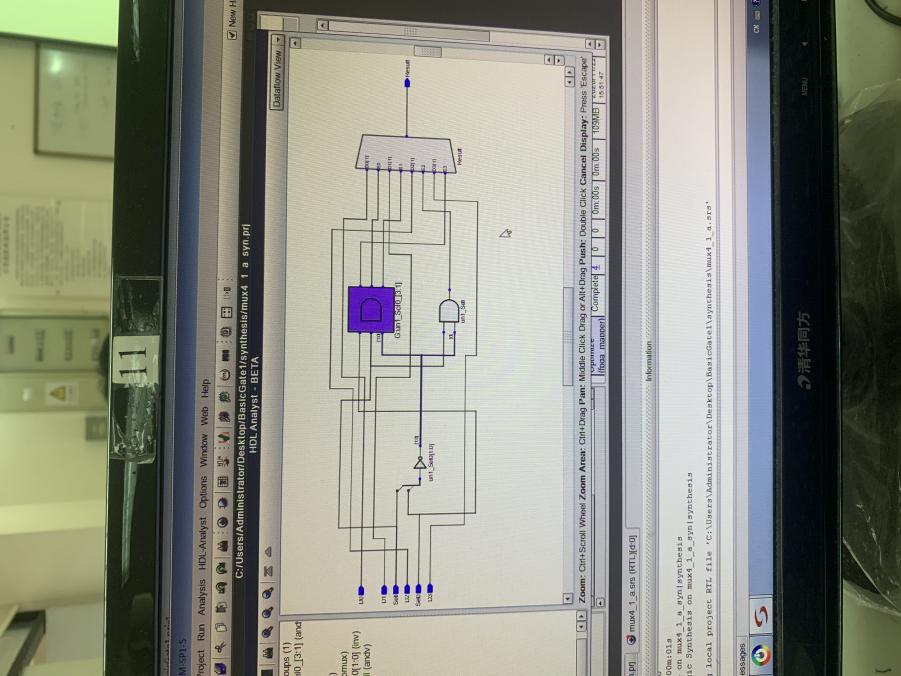
3、综合结果（**截图，注明对应的模块**）



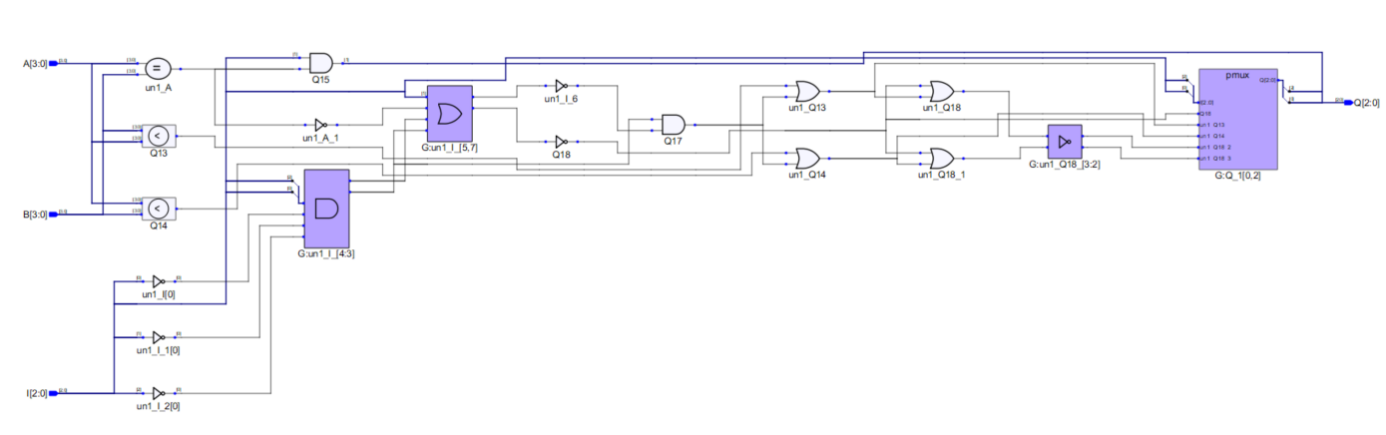
74HC148



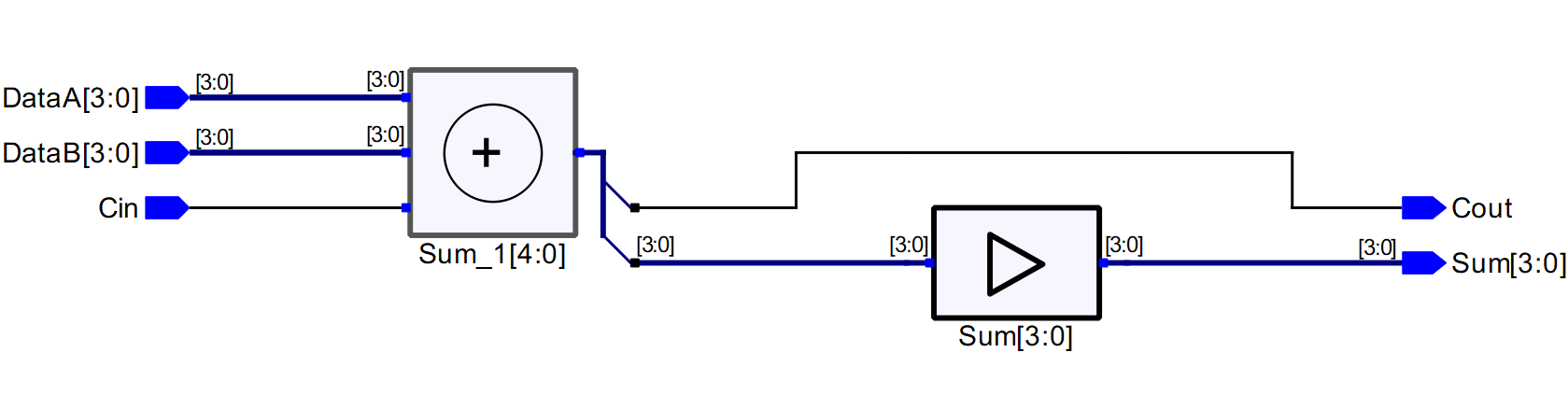
74HC138



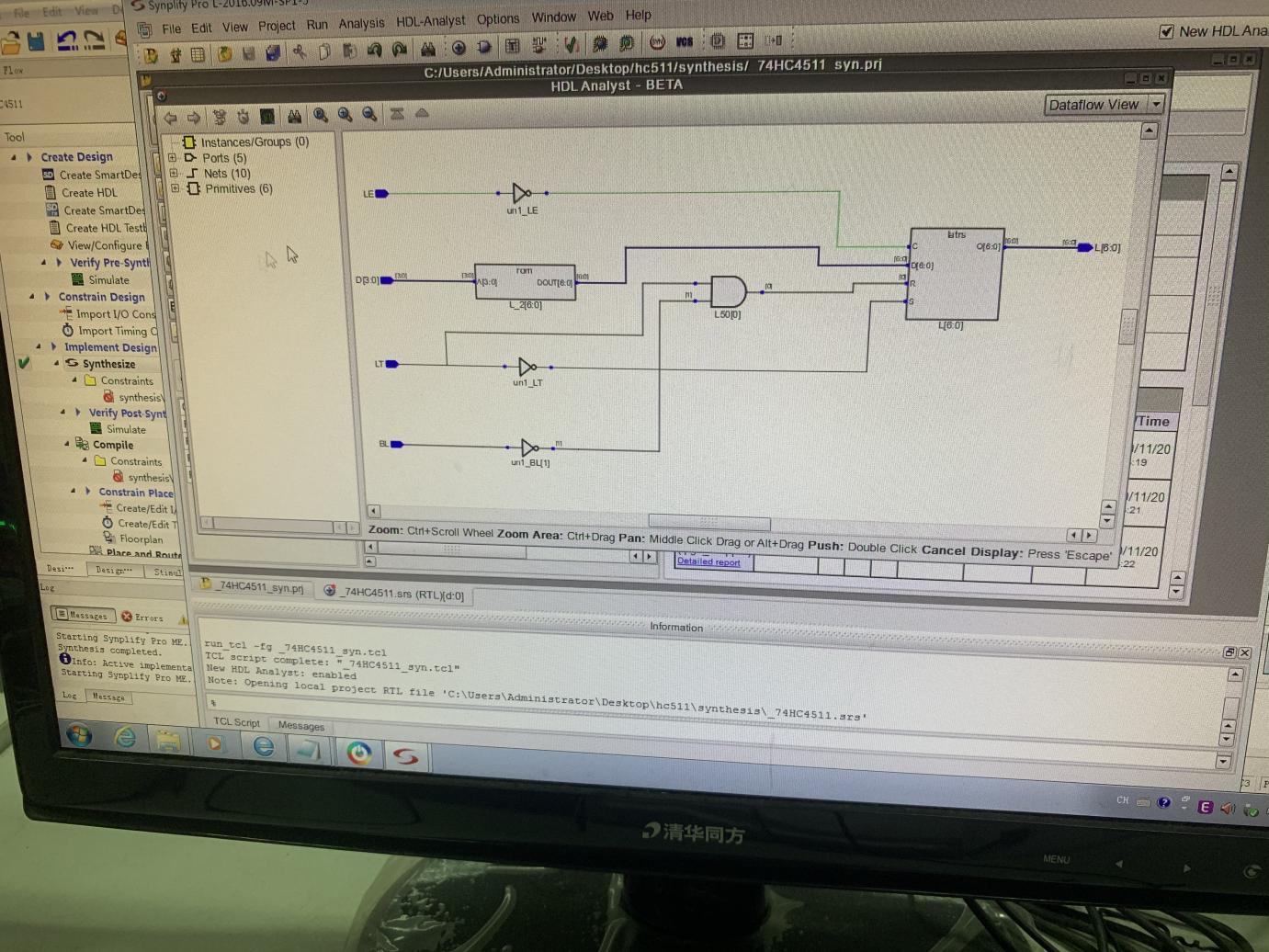
74HC153



74HC85

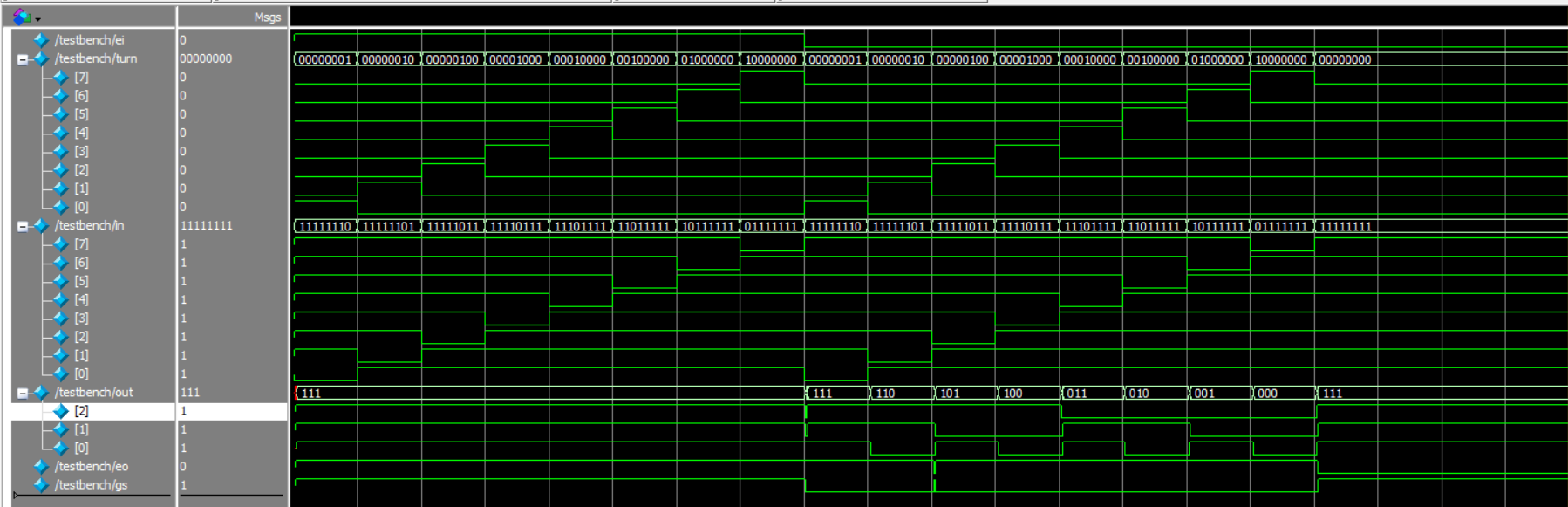


74HC283



74HC4511

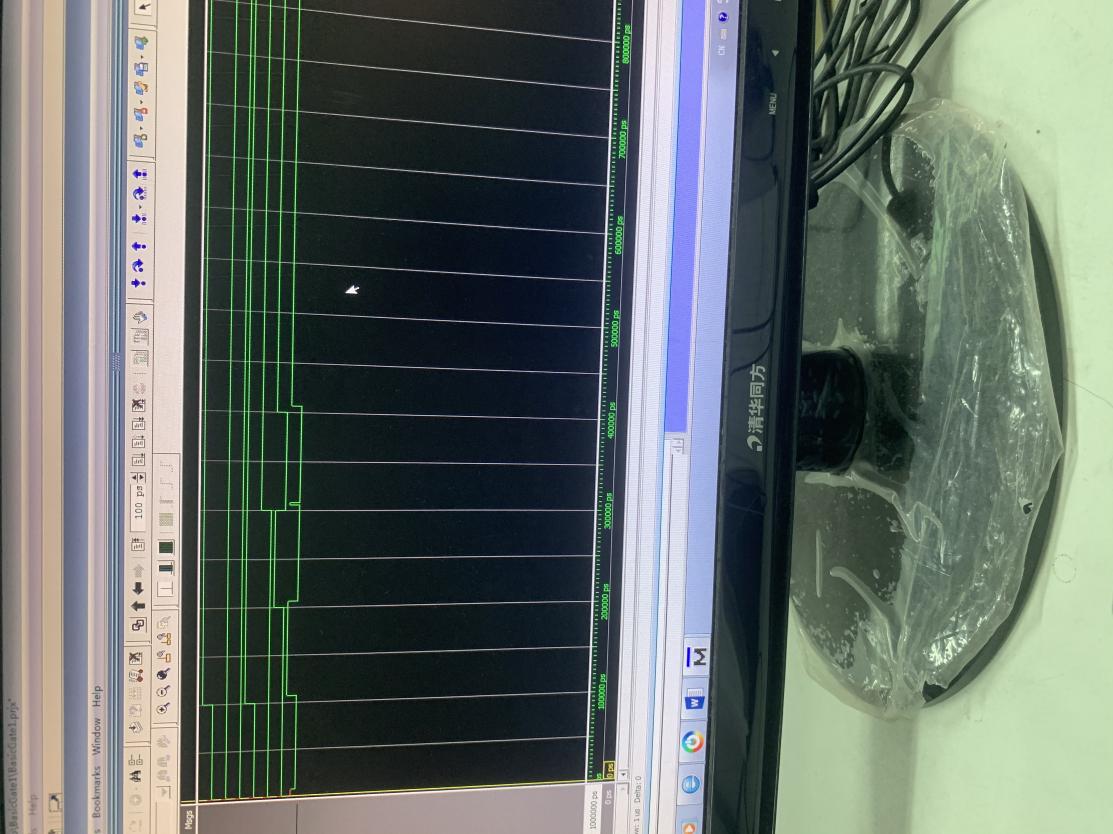
4、第二次仿真结果（综合后）（**截图，注明对应的模块**）。回答输出信号是否有延迟，延迟时间约为多少？



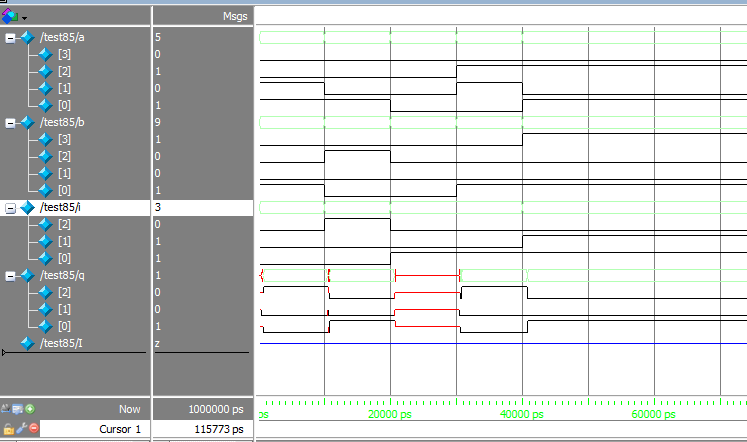
74HC148 ，信号有延迟 约400ps 出现竞争冒险



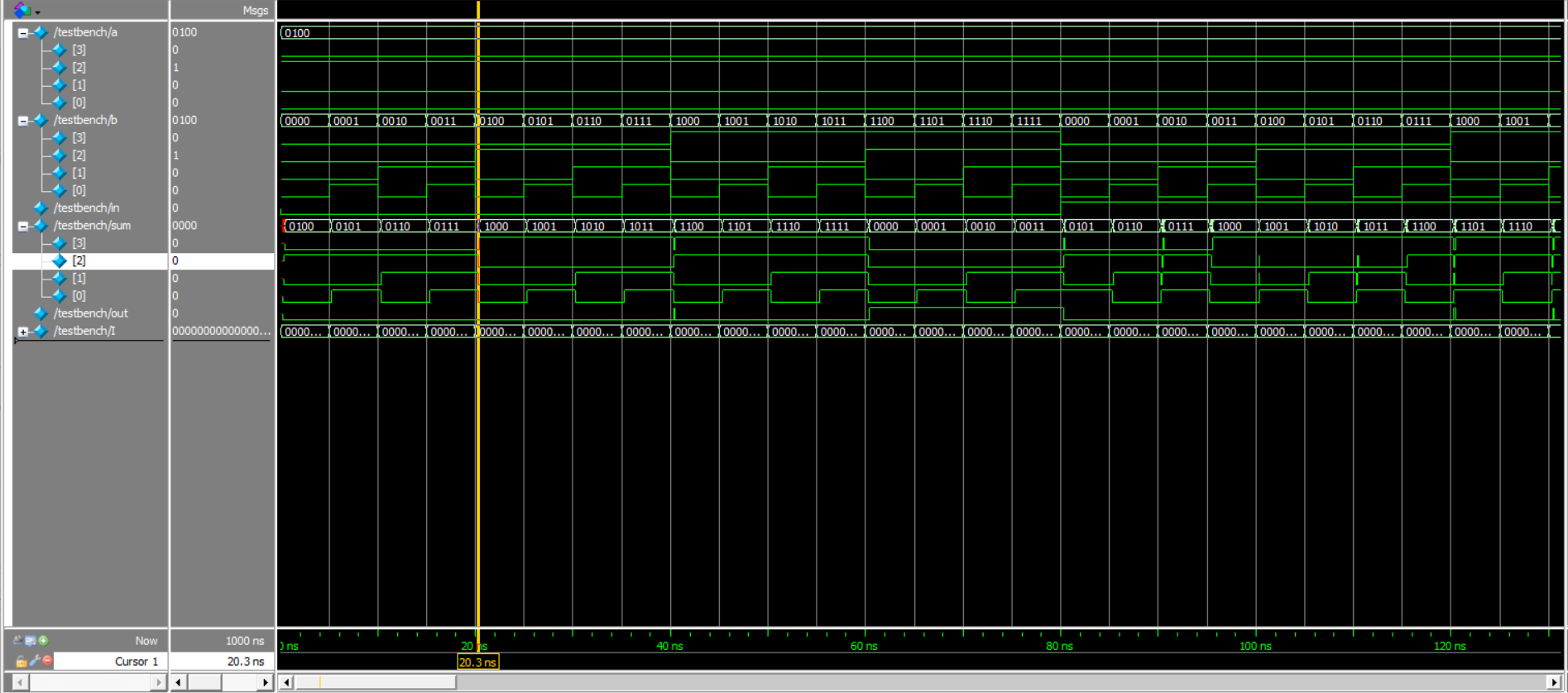
74HC138，有延时，延时大概为320ps，有竞争冒险



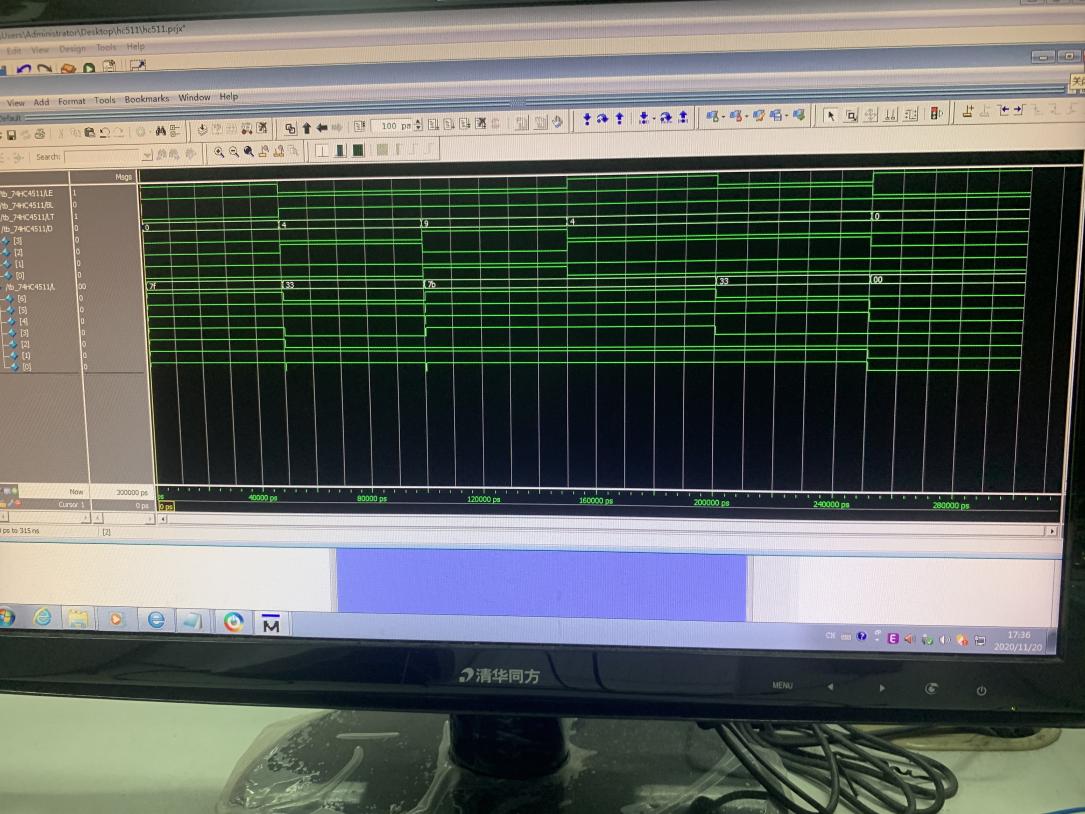
74HC153，有延时，延时约为500ps, 有竞争冒险



74HC85，信号有延迟 约400ps 出现竞争冒险

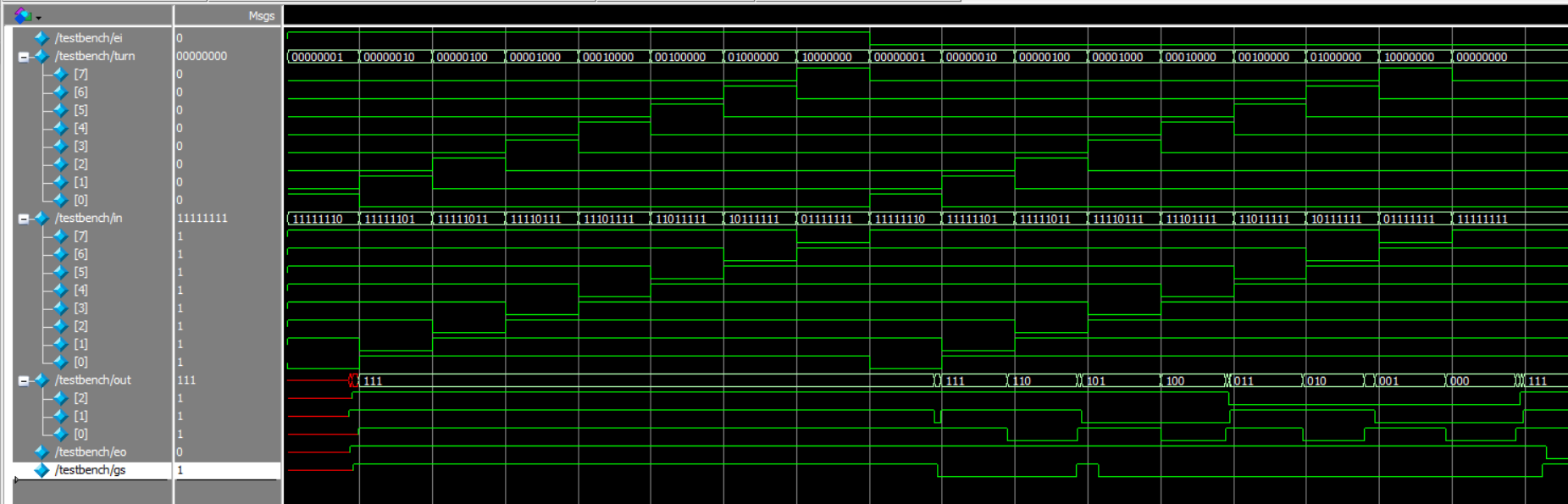


74HC283，有延时，延时大概为300ps。

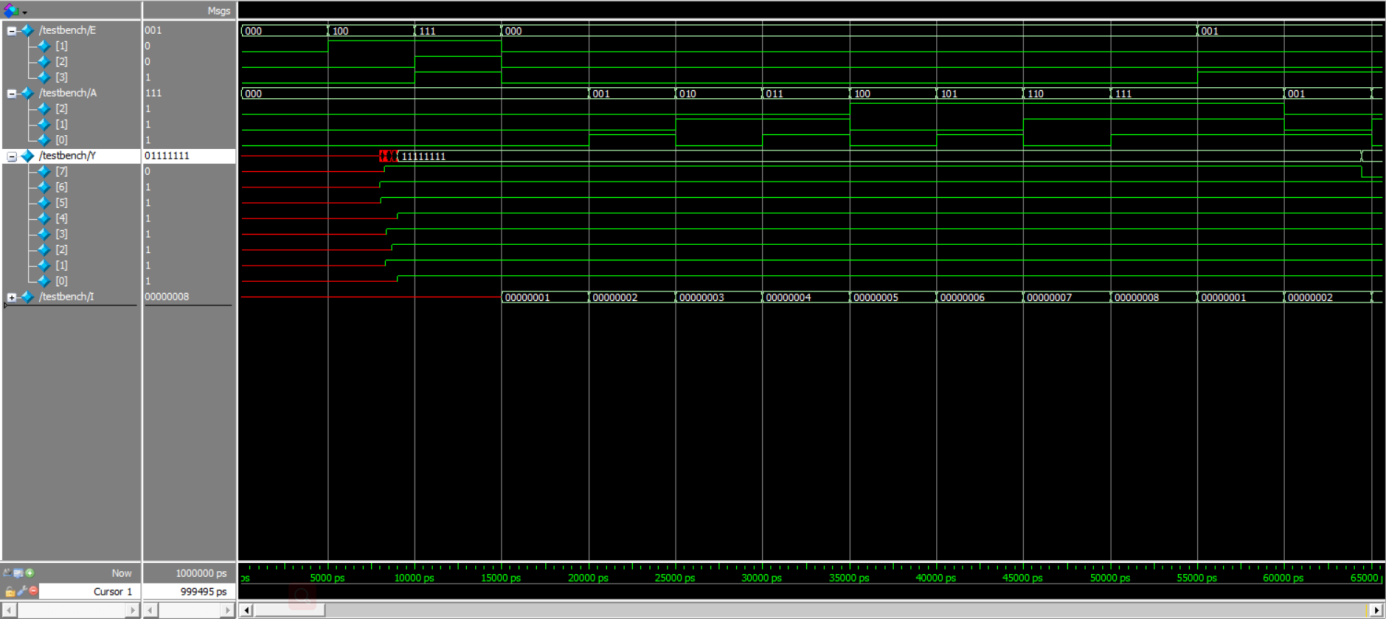


74HC4511，信号有延迟 约500ps 出现竞争冒险

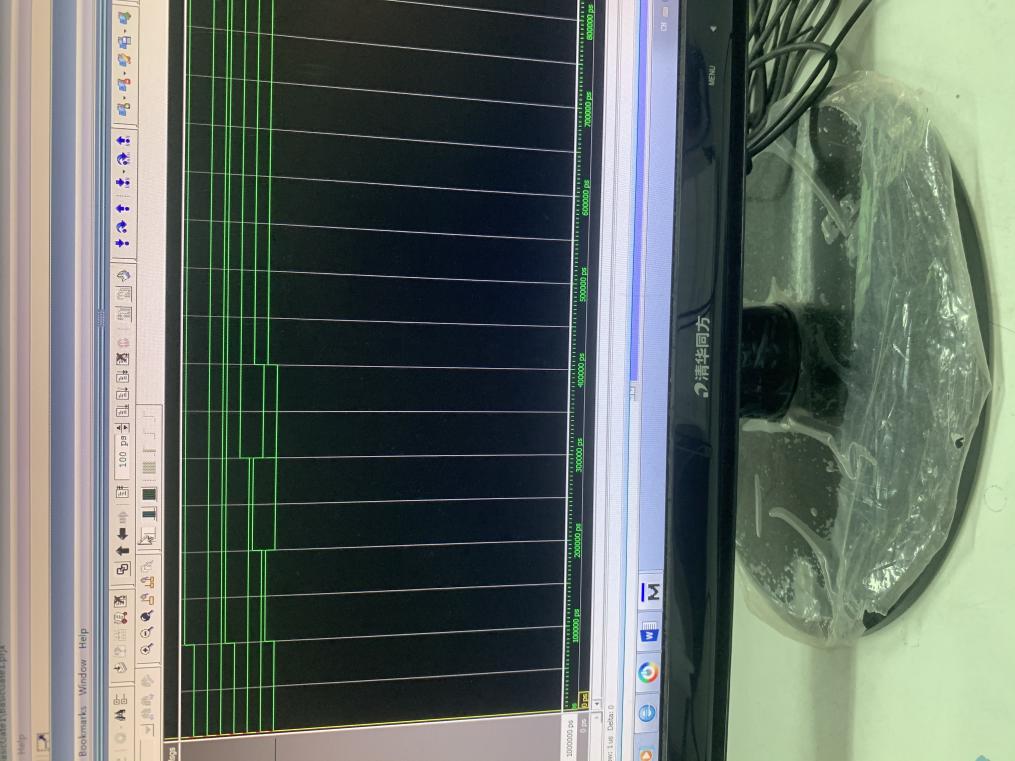
5、第三次仿真结果（布局布线后）（**截图，注明对应的模块**）。回答输出信号是否有延迟，延迟时间约为多少？分析是否有出现竞争冒险。



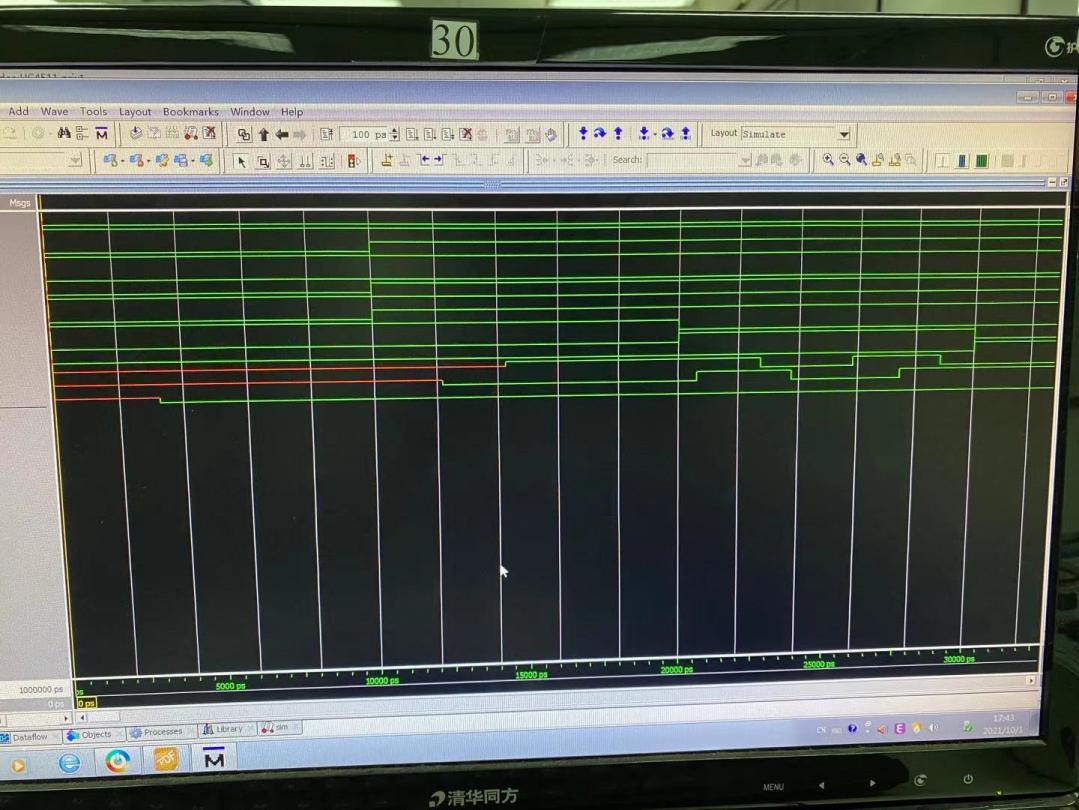
74HC148，信号有延迟 约1300ps 无竞争冒险



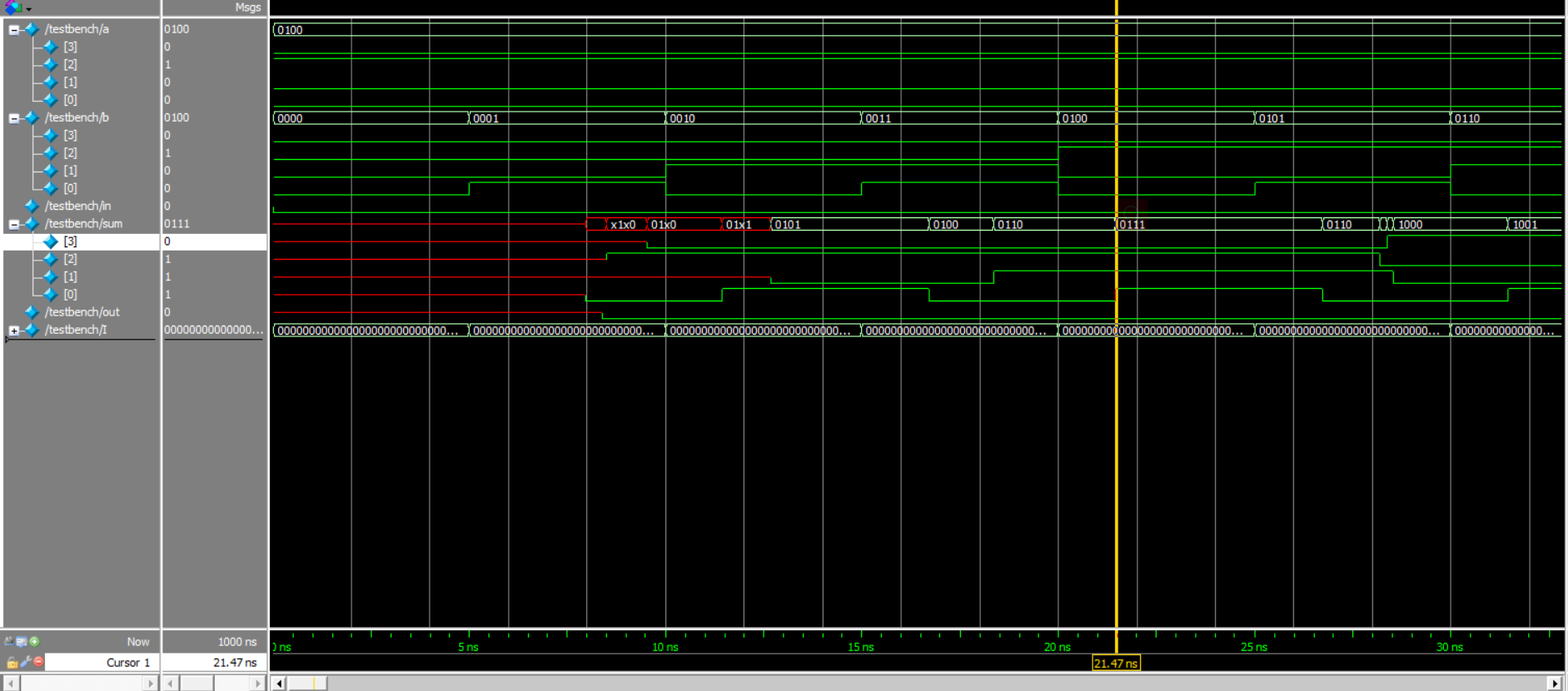
74HC138，有延时，延时大概为4453ps，无竞争冒险



74HC153有延时，延时约10000ps, 有竞争冒险



74HC85信号有延迟 约550ps 无竞争冒险



74HC283有延时，大概为470ps，无竞争冒险。



74HC4511信号有延迟 约10400ps 无竞争冒险时序逻辑电路

一、实验目的

1、了解基于Verilog的时序逻辑电路的设计及其验证。

2、熟悉利用EDA工具进行设计及仿真的流程。

3、熟悉实验箱的使用和程序下载（烧录）及测试的方法。

4、学习针对实际时序逻辑电路芯片74HC74、74HC112、74HC194、74HC161进行VerilogHDL设计的方法。

二、实验环境及仪器

1、Libero仿真软件。

2、数字逻辑与系统设计实验箱。

3、Actel A3P060 FPGA芯片及Flash Pro5烧录器。

三、实验内容

1、熟练掌握Libero软件的使用方法。

2、进行针对74系列时序逻辑电路的设计，并完成相应的仿真实验。

3、参考教材中相应章节的设计代码、测试平台代码（可自行编程），完成74HC74、74HC112、74HC161、74HC194相应的设计、综合及仿真。

4、提交针对74HC74、74HC112、74HC161、74HC194的综合结果，以及相应的仿真结果。

5、**将引脚分配图及烧录成功界面拍照或截图发送至老师的微信**。

四、实验结果和数据处理

1、所有模块及测试平台代码清单

//74HC74代码

module HC74(SD,RD,Clk,D,Q,NQ);

input [1:0]SD,RD,Clk,D;

output [1:0]Q,NQ;

reg [1:0]Q,NQ;

always @ (posedge Clk[0] or negedge SD[0] or negedge RD[0])

begin

if(!SD[0])

begin

if(RD[0])

begin

Q[0]=1;

NQ[0]=0;

end

else

begin

Q[0]=1;

NQ[0]=1;

end

end

else

begin

if(!RD[0])

begin

Q[0]=0;

NQ[0]=1;

end

else

begin

Q[0]=D[0];

NQ[0]=~D[0];

end

end

end

always @ (posedge Clk[1] or negedge SD[1] or negedge RD[1])

begin

if(!SD[1])

begin

if(RD[1])

begin

Q[1]=1;

NQ[1]=0;

end

else

begin

Q[1]=1;

NQ[1]=1;

end

end

else

begin

if(!RD[1])

begin

Q[1]=0;

NQ[1]=1;

end

else

begin

Q[1]=D[1];

NQ[1]=~D[1];

end

end

end

endmodule

//74HC74测试平台代码

`timescale 1ns/1ns

module test74();

reg [1:0]SD,RD,Clk,D;

wire [1:0]Q,NQ;

HC74 u1(SD,RD,Clk,D,Q,NQ);

initial

begin

Clk=0;

#400 $finish;

end

parameter clock\_period=20;

always #(clock\_period/2)Clk=~Clk;

initial

begin

SD=2'b01;RD=2'b10;D=2'b01;

#10 SD=2'b10;RD=2'b01;

#10 SD=2'b00;RD=2'b00;

#10 SD=2'b11;RD=2'b11;D=2'b01;

#10 SD=2'b11;RD=2'b11;D=2'b10;

end

endmodule

//74HC112代码

module HC112(SD,RD,Clk,J,K,Q,NQ);

input [1:0]SD,RD,Clk,J,K;

output [1:0]Q,NQ;

reg [1:0]Q,NQ;

always@(negedge SD[0] or negedge RD[0] or negedge Clk[0])

begin

if(SD[0]==0&&RD[0]==1)

begin

Q[0]=1;

NQ[0]=0;

end

else if(SD[0]==1&&RD[0]==0)

begin

Q[0]=0;

NQ[0]=1;

end

else if(SD[0]==0&&RD[0]==0)

begin

Q[0]=1;

NQ[0]=0;

end

else

begin

if(J[0]==0&&K[0]==0)

begin

Q[0]=Q[0];

NQ[0]=NQ[0];

end

else if(J[0]==0&&K[0]==1)

begin

Q[0]=0;

NQ[0]=1;

end

else if(J[0]==1&&K[0]==0)

begin

Q[0]=1;

NQ[0]=0;

end

else

begin

Q[0]=~Q[0];

NQ[0]=~NQ[0];

end

end

end

always@(negedge SD[1] or negedge RD[1] or negedge Clk[1])

begin

if(SD[1]==0&&RD[1]==1)

begin

Q[1]=1;

NQ[1]=0;

end

else if(SD[1]==1&&RD[1]==0)

begin

Q[1]=0;

NQ[1]=1;

end

else if(SD[1]==0&&RD[1]==0)

begin

Q[1]=1;

NQ[1]=0;

end

else

begin

if(J[1]==0&&K[1]==0)

begin

Q[1]=Q[1];

NQ[1]=NQ[1];

end

else if(J[1]==0&&K[1]==1)

begin

Q[1]=0;

NQ[1]=1;

end

else if(J[1]==1&&K[1]==0)

begin

Q[1]=1;

NQ[1]=0;

end

else

begin

Q[1]=~Q[1];

NQ[1]=~NQ[1];

end

end

end

endmodule

//74HC112测试平台代码

`timescale 1ns/10ps

module test112();

reg [1:0]SD,RD,Clk,J,K;

wire [1:0]Q,NQ;

HC112 u1(SD,RD,Clk,J,K,Q,NQ);

initial

begin

Clk=2'b01;

#400 $finish;

end

parameter clock\_period=20;

always #(clock\_period/2) Clk=~Clk;

initial

begin

SD=2'b01;RD=2'b10;J=2'b01;K=2'b01;

#10 SD=2'b10;RD=2'b01;

#10 SD=2'b00;RD=2'b00;

#10 SD=2'b11;RD=2'b11;K=2'b10;

#10 SD=2'b11;RD=2'b11;J=2'b10;K=2'b01;

#10 SD=2'b11;RD=2'b11;K=2'b10;

end

endmodule

//74HC161代码

module HC161(CP,CEP,CET,MRN,PEN,Dn,Qn,TC);

input CP;

input CEP,CET;

output [3:0]Qn;

input MRN,PEN;

input [3:0]Dn;

output TC;

reg [3:0]qaux;

reg TC;

always @ (posedge CP)

begin

if(!MRN)

qaux<=4'b0000;

else if(!PEN)

qaux<=Dn;

else if(CEP&CET)

qaux<=qaux+1;

else

qaux<=qaux;

end

always @ (posedge CP)

begin

if(qaux==4'b1110)

TC=1'b1;

else

TC=1'b0;

end

assign Qn=qaux;

endmodule

//74HC161测试平台代码

`timescale 1ns/10ps

module test161();

reg MRN,CP,CEP,CET,PEN;

reg [3:0]Dn;

wire [3:0]Qn;

wire TC;

HC161 u1(CP,CEP,CET,MRN,PEN,Dn,Qn,TC);

integer i;

initial

begin

CP=0;

#400 $finish;

end

parameter clock\_period=10;

always #(clock\_period/2)CP=~CP;

initial

begin

MRN=1;PEN=0;CEP=1;CET=1;Dn=4'b0000;

#20 CEP=1;CET=1;PEN=1;MRN=1;

end

initial

begin

#200 PEN=0;

for(i=15;i>=0;i=i-1)

begin

#5 Dn=i;

end

end

endmodule

//74HC194代码

module HC194(MRN,Sn,CP,DSR,DSL,Dn,Qn);

input MRN,CP,DSR,DSL;

input [1:0]Sn;

input [3:0]Dn;

output [3:0]Qn;

reg [3:0]Qn;

always@(posedge CP or negedge MRN)

begin

if(MRN==0) Qn<=4'b0000;

else if(Sn==2'b00) Qn<=Qn;

else if(Sn==2'b01&&DSR==0)

begin

Qn[3]<=Qn[2];

Qn[2]<=Qn[1];

Qn[1]<=Qn[0];

Qn[0]<=0;

end

else if(Sn==2'b01&&DSR==1)

begin

Qn[3]<=Qn[2];

Qn[2]<=Qn[1];

Qn[1]<=Qn[0];

Qn[0]<=1;

end

else if(Sn==2'b10&&DSL==0)

begin

Qn[0]<=Qn[1];

Qn[1]<=Qn[2];

Qn[2]<=Qn[3];

Qn[3]<=0;

end

else if(Sn==2'b10&&DSL==1)

begin

Qn[0]<=Qn[1];

Qn[1]<=Qn[2];

Qn[2]<=Qn[3];

Qn[3]<=1;

end

else if(Sn==2'b11)

begin

Qn<=Dn;

end

end

endmodule

//74HC194测试平台代码

`timescale 1ns/10ps

module test194();

reg MRN,CP,DSR,DSL;

reg [1:0]Sn;

reg [3:0]Dn;

wire [3:0]Qn;

HC194 u1(MRN,Sn,CP,DSR,DSL,Dn,Qn);

initial

begin

CP=0;

#400 $finish;

end

parameter clock\_period=10;

always #(clock\_period/2) CP=~CP;

initial

begin

MRN=0;

#20 MRN=1;Sn=4'b11;Dn=4'b1010;

#20 Sn=2'b01;DSR=0;

#20 MRN=1;Sn=4'b11;Dn=4'b1010;

#20 Sn=2'b01;DSR=1;

#20 MRN=1;Sn=4'b11;Dn=4'b1010;

#20 Sn=2'b10;DSL=0;

#20 MRN=1;Sn=4'b11;Dn=4'b1010;

#20 Sn=2'b10;DSL=1;

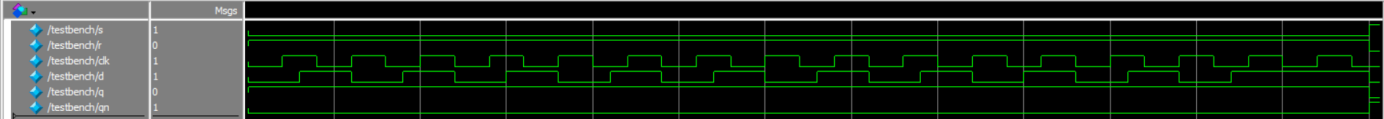
#20 Sn=2'b00;

#20 MRN=0;

end

endmodule

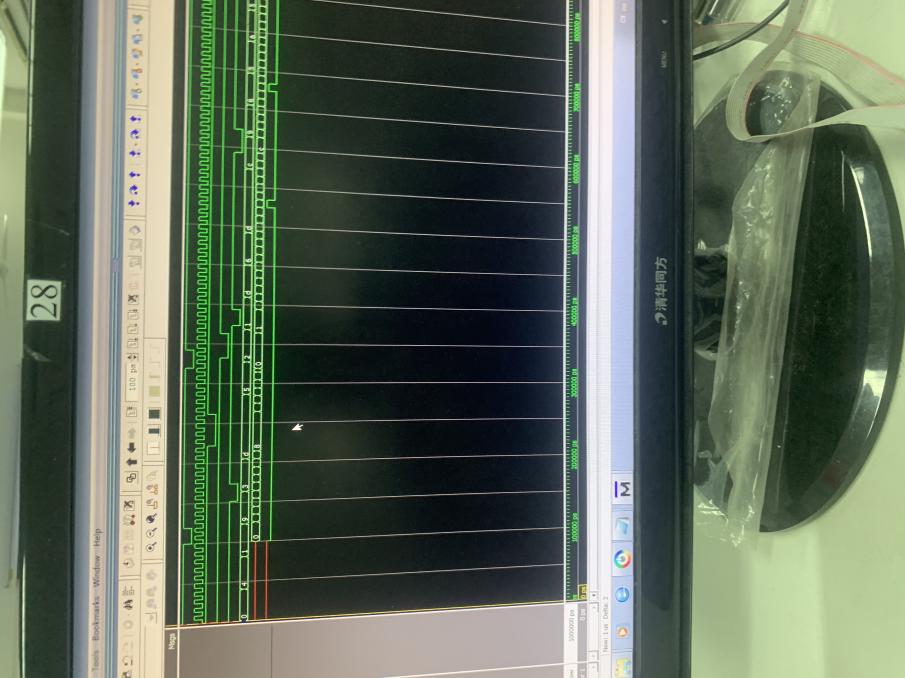
2、第一次仿真结果（**截图，注明对应的模块**）



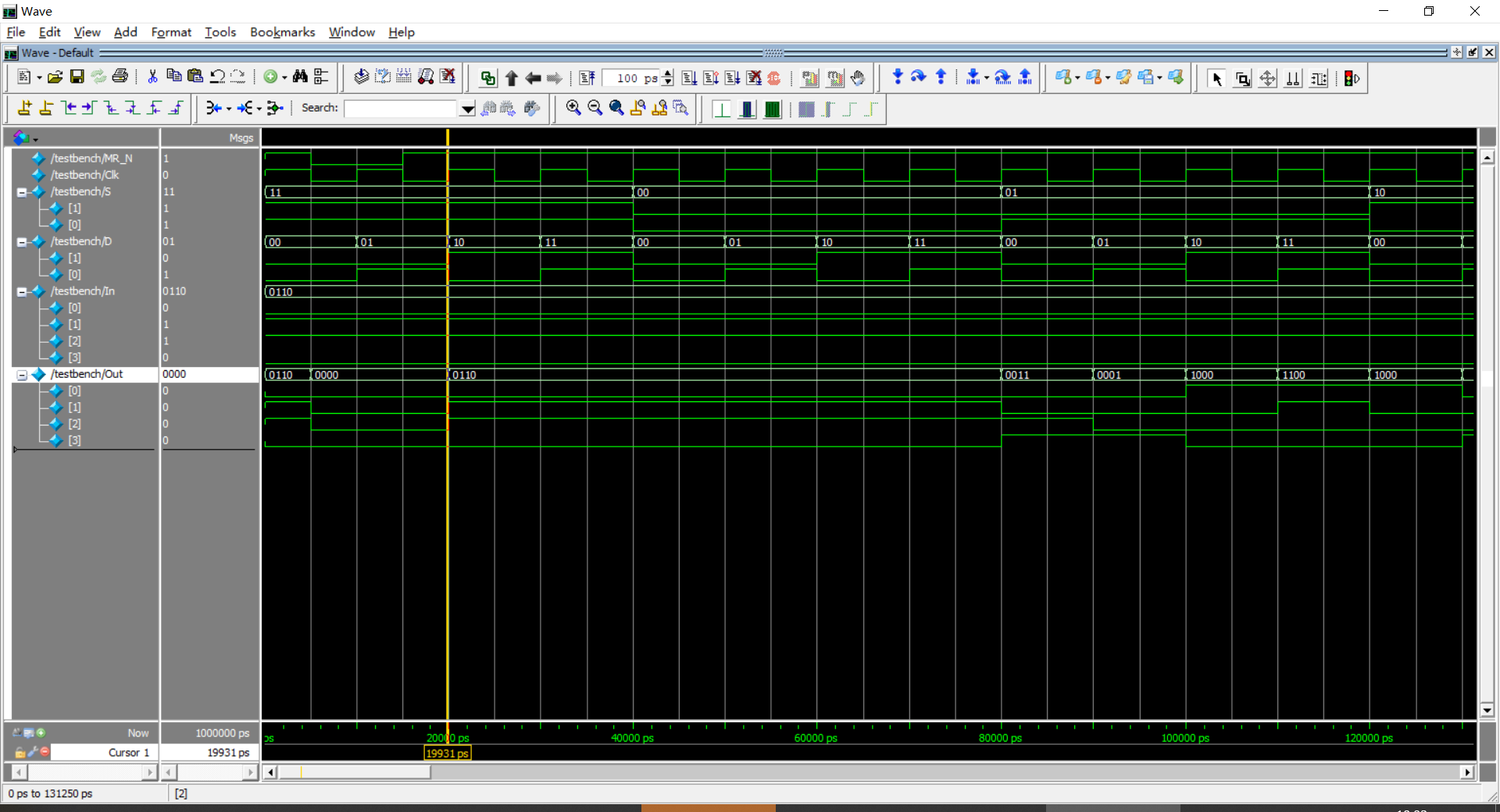
74HC74



74HC112

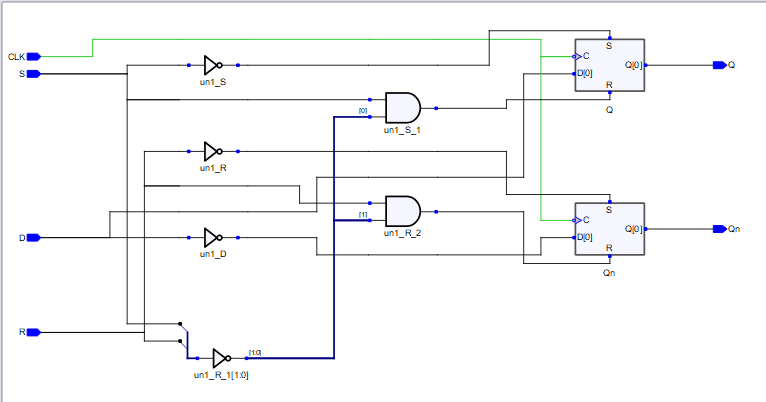


74HC161

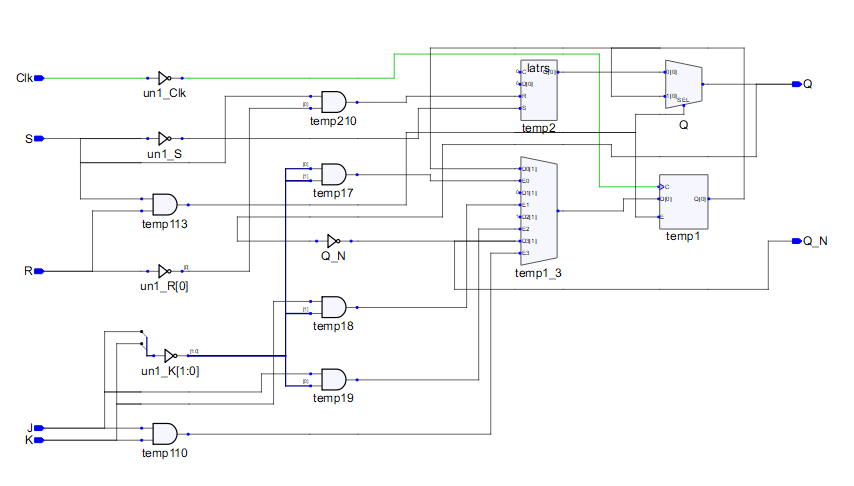


74HC194

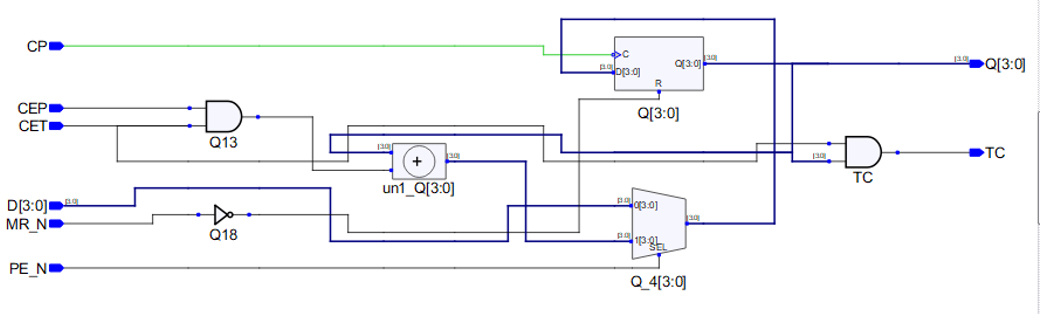
3、综合结果（**截图，注明对应的模块**）



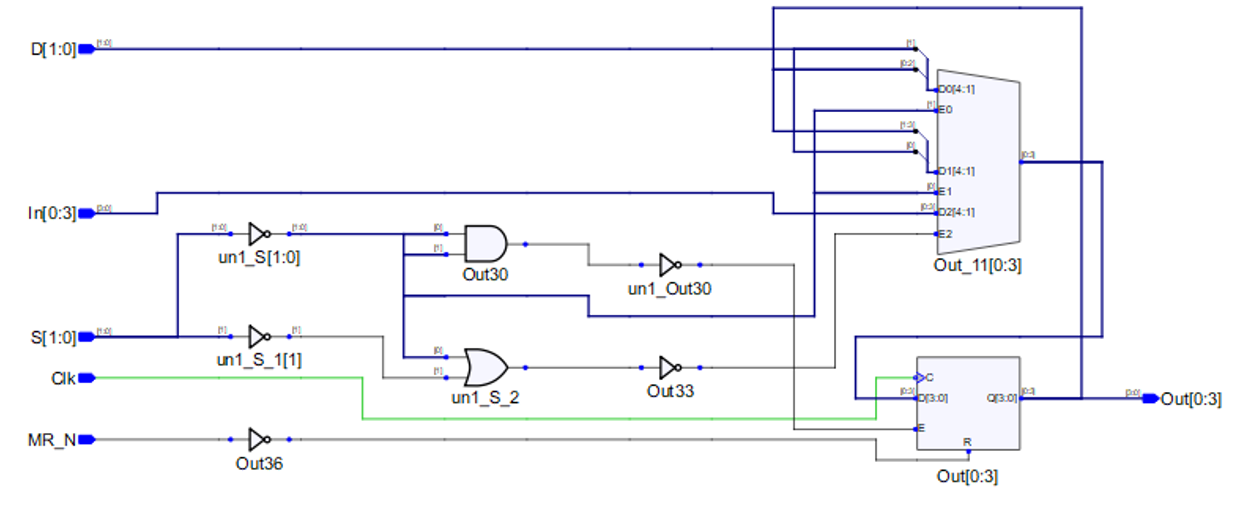
74HC74



74HC112

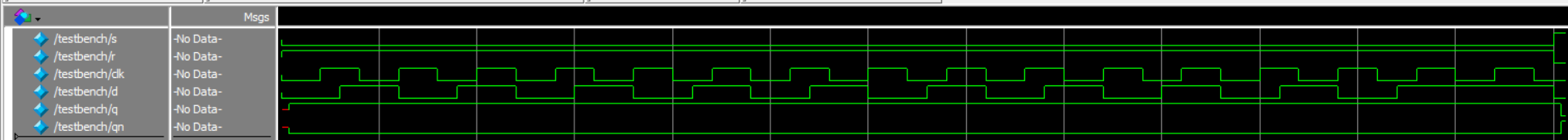


74HC161



74HC194

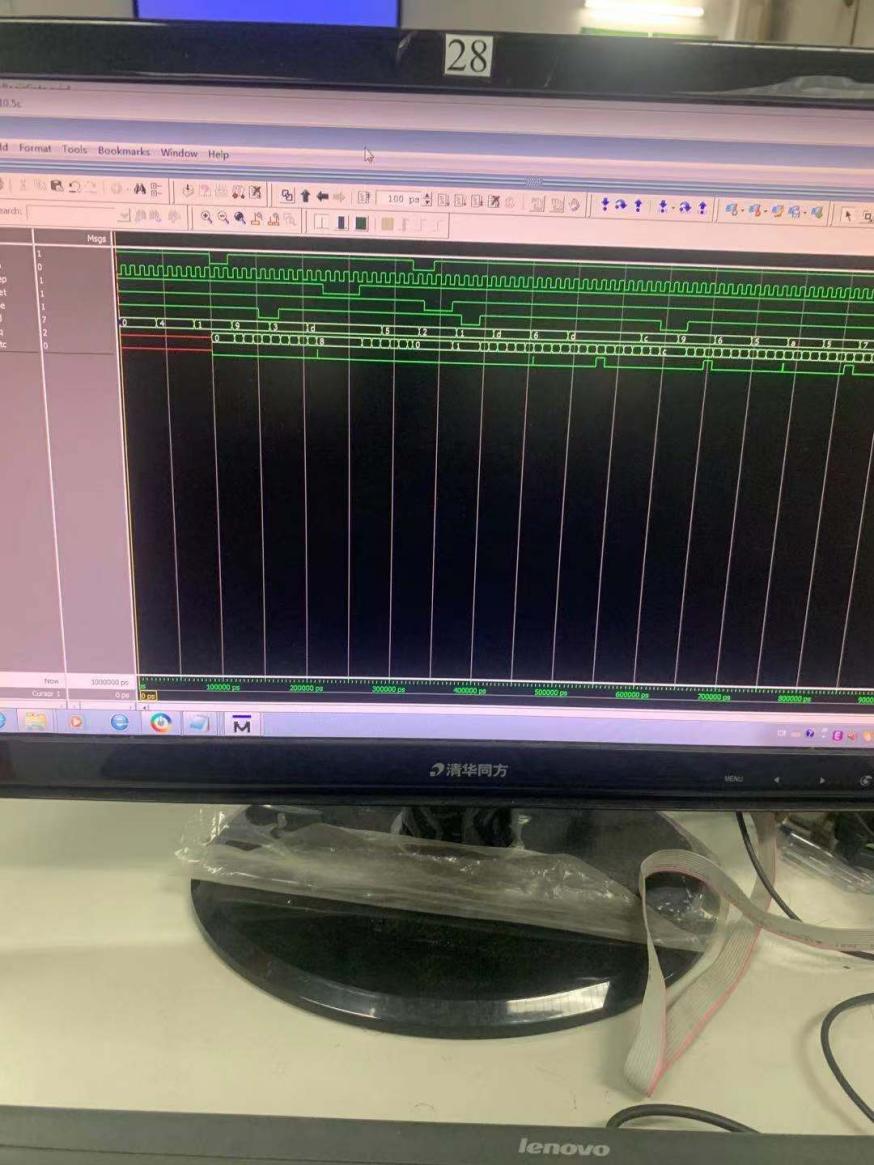
4、第二次仿真结果（综合后）（**截图，注明对应的模块**）。



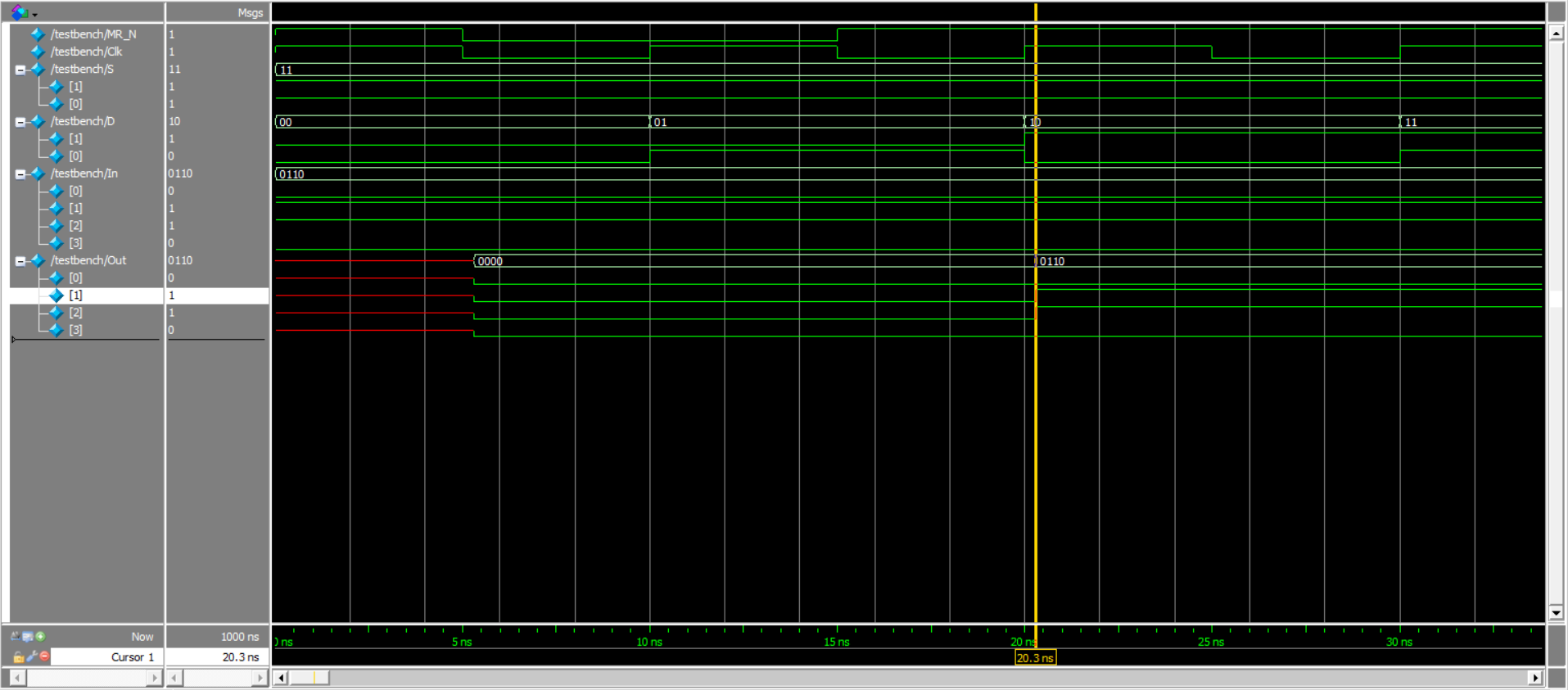
74HC74，有延时，延时大概为1000ps



74HC112有延时，延时约为400ps

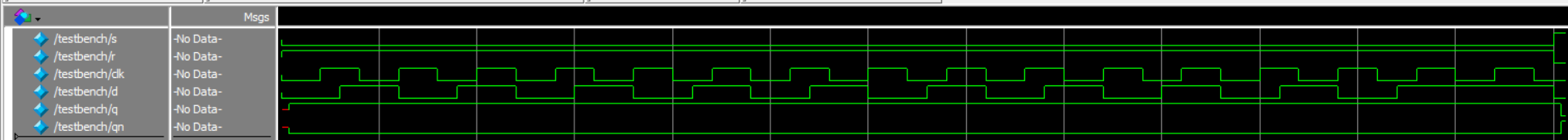


74HC161信号有延迟 约400ps

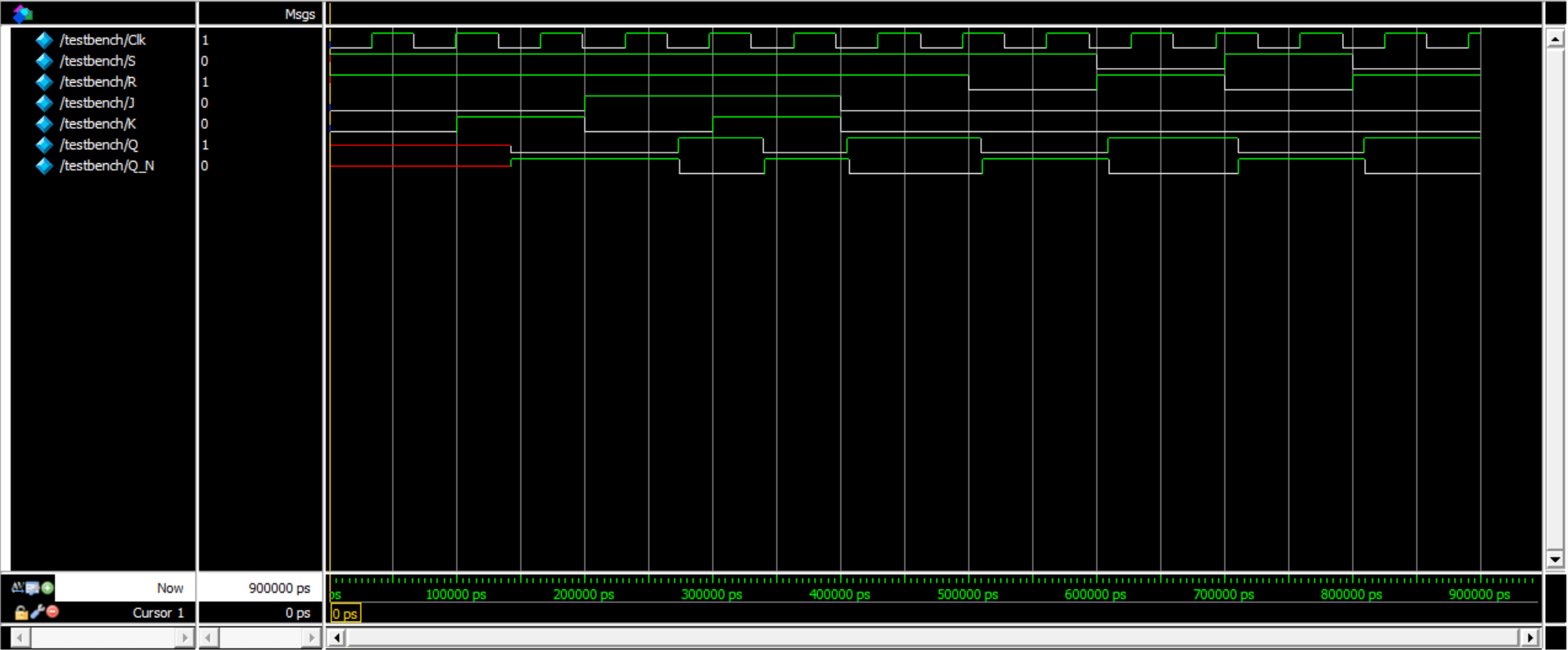


74HC194有延时，延时时间大概为300ps

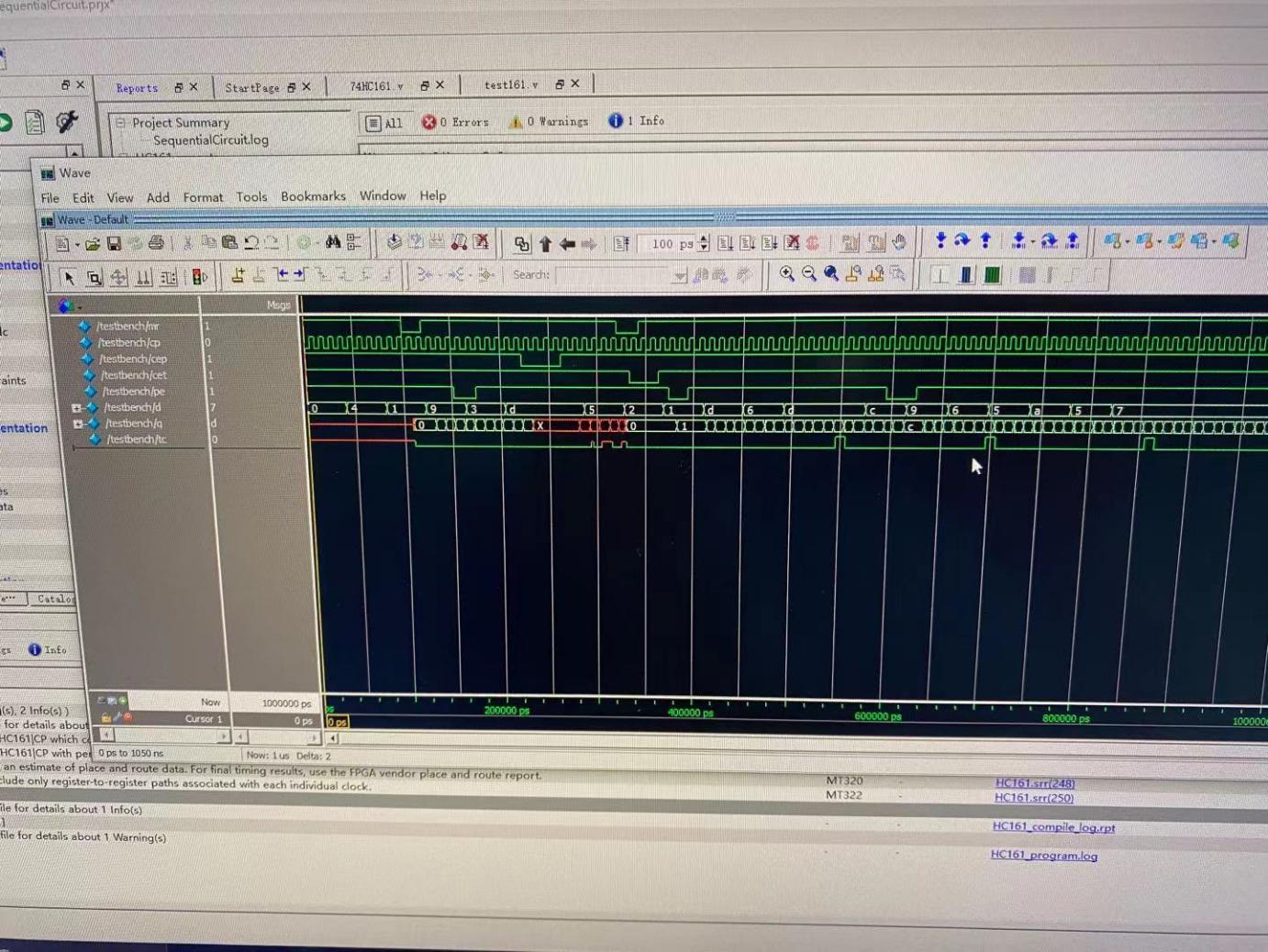
5、第三次仿真结果（布局布线后）（**截图，注明对应的模块**）。



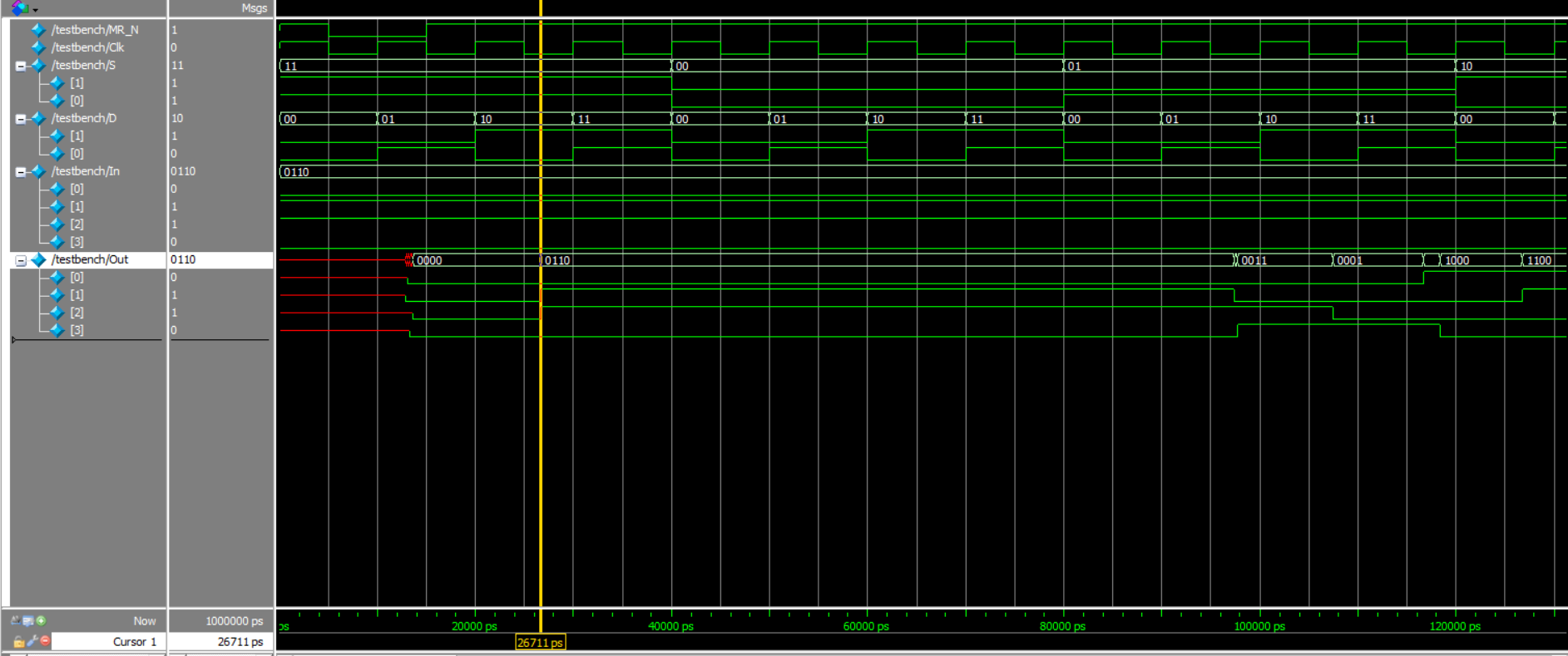
74HC74，有延时，延时大概为1000ps。无竞争冒险。



74HC112，有延时，延时约为800ps，无竞争冒险



74HC161信号有延迟 约440ps 无竞争冒险。



74HC194有延时，延时时间大概为680ps。没有竞争冒险

## 综合实验

一、实验目的

1、进一步熟悉利用EDA工具进行设计及仿真的流程。

2、熟悉利用EDA工具中的图形化设计界面进行综合设计。

3、熟悉芯片烧录的流程及步骤。

4、掌握分析问题、解决问题的综合能力，通过EDA工具设计出能解决实际问题的电路。

二、实验环境

1、Libero仿真软件。

2、数字逻辑与系统设计实验箱。

3、Actel A3P060 FPGA芯片及Flash Pro5烧录器。

三、实验内容

按老师要求在以下题目中选做，并按要求完成实验报告。

综合实验1——编码器扩展实验：当按下小于等于9的按键后，显示数码管显示数字，当按下大于9的按键后，显示数码管不显示数字。若同时按下几个按键，优先级别的顺序是9到0。本实验需要两个编码器74HC148、一个数码显示译码器74HC4511、一个共阴极8段显示数码管LN3461Ax和一个数值比较器74HC85。利用Libero SmartDesign图形化设计工具，采用图文混合设计方法进行设计。

综合实验2——译码器扩展实验：设计一个电路，通过改变输入，令显示数码管的4个数位轮流显示数字。本实验需要一个3-8译码器74HC138、一个数码显示译码器74HC4511、一个共阴极8段显示数码管LN3461Ax。将译码器74HC138的输入显示在数码管LN3461Ax上，并利用译码器74HC138的输出控制数码显示译码器74HC4511的工作（、或LE中任一个）。

综合实验3——有符号比较器实验：设计一个电路，比较两个8位有符号数的大小，判定是否满足大于等于的关系。方法：直接利用Libero工具提供的比较器IP核，实现一个有符号比较器。

综合实验4——4位有符号二进制加法器。

综合实验5——二——十进制码转换电路：设计一个能实现8位二进制码转换为12位8421BCD码的电路。

综合实验6——利用状态机实现一个简单自动售货机控制电路（顶层结构框图如图7-23）。该电路有两个投币口（1元和5角），商品2元一件，不设找零。In[0]表示投入5角，In[1]表示投入1元，D\_out表示是否提供货品。分别用Mealy和Moore型实现。

综合实验7——与寄存器结合的有限状态机：将寄存器逻辑（利用时钟信号同步进行赋值）与Mealy或Moore状态机组合起来，可以得出两种解决方案。

综合实验8——跑马灯：设计要求：共8个LED灯连成一排，用以下3种模式来显示，模式选择使用两个按键进行控制。

① 模式1：先点亮奇数灯，即1、3、5、7灯亮，然后偶数灯，即2、4、6、8灯亮，依次循环，灯亮的时间按时钟信号的二分频设计。

② 模式2：按照1、2、3、4、5、6、7、8的顺序依次点亮所有灯；然后再按1、2、3、4、5、6、7、8的顺序依次熄灭所有灯，间隔时间按时钟信号的八分频设计。

③ 模式3：按照1/8、2/7、3/6、4/5的顺序依次点亮所有灯，每次同时点亮两个灯；然后再按1/8、2/7、3/6、4/5的顺序熄灭相应灯，每次同时熄灭两个灯，灯亮的时间按时钟信号的四分频设计。

④ 模式4：自行设计。

综合实验9——四位数码管扫描显示电路的设计

设计要求：共4个数码管，连成一排，要求可以显示其中任意一个数码管。具体要求如下：

① 依次选通4个数码管，并让每个数码管显示相应的值，其结果由相应输入决定（通过扩展板按键输入）。

② 要求能在实验箱上演示出数码管的动态显示过程。必须使得4个选通信号DIG1、DIG2、DIG3、DIG4轮流被单独选通，同时，在段信号输入口加上希望在对应数码管上显示的数据，这样随着选通信号的变化，才能实现扫描显示的目的。

综合实验10——交通灯控制器

设计要求：实现一个常见的十字路口交通灯控制功能。一个十字路口的交通灯一般分为两个方向，每个方向具有红灯、绿灯和黄灯三种。实现一个常见的十字路口交通灯控制功能，具体要求如下：

① 十字路口包含A、B两个方向的车道。A方向放行一分钟（绿55秒，黄5秒），同时B方向禁行（红60秒）；然后A方向禁行1分钟（红60秒），同时B方向放行（绿55秒，黄5秒），示意图如图5-56所示。依此类推，循环往复。

② 实现正常的倒计时功能，用两组数码管作为A和B两个方向的倒计时显示。

③ 系统时钟1KHz。

综合实验11——键盘扫描器和编码器

设计要求：

① 检测是否有按键按下；

② 确定按键的位置；

③ 生成唯一按键编码；

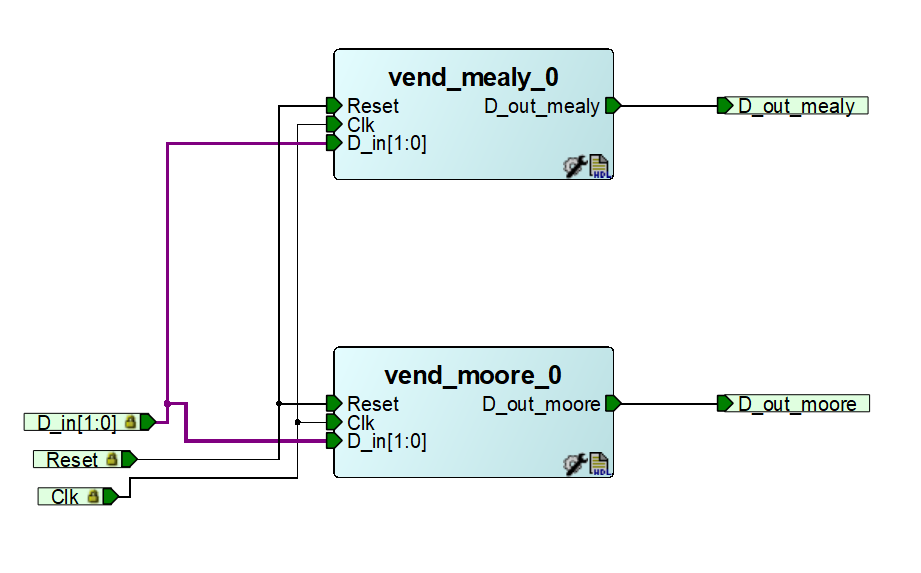
④ 在数码管上显示相应的按键内容。

四、实验结果和数据处理

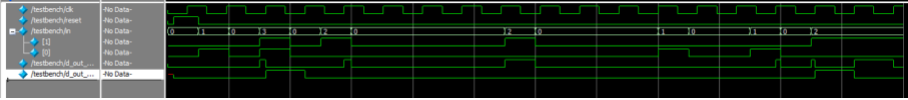
每个实验均要记录以下内容。

自动售货机

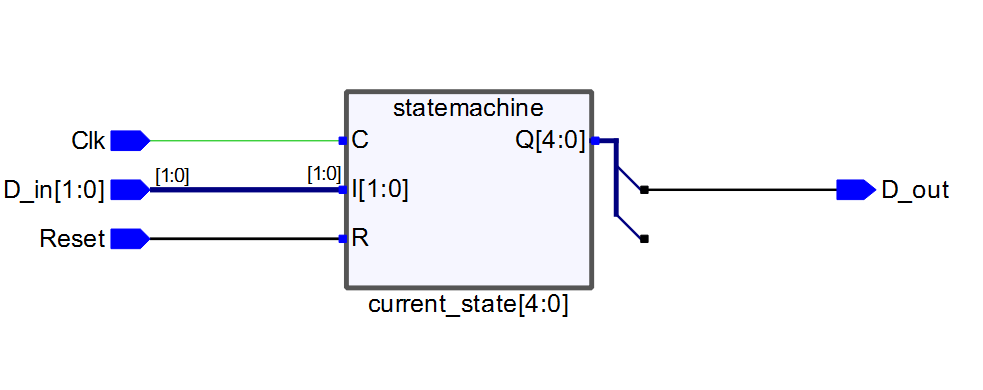
1. SmartDesign的连线图



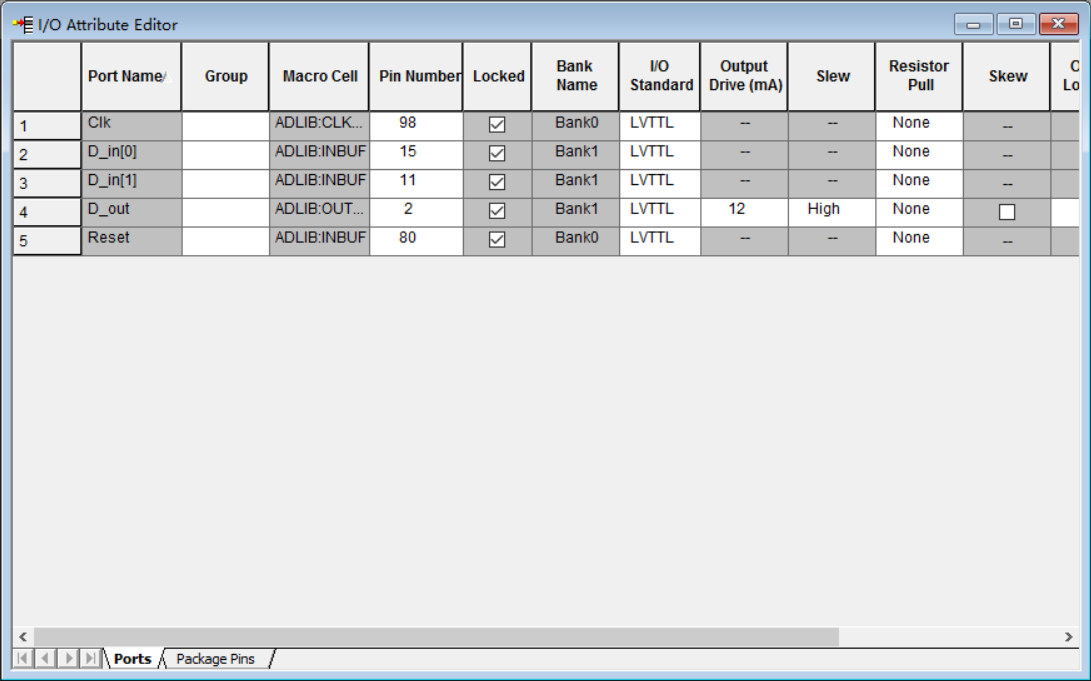
1. 功能仿真波形图



1. 综合结果RTL图



1. 引脚分配I/O Attribute Editor截图



## 附录：子箱26位开关及LED灯引脚分配表

由于子箱中提供输入输出的26位开关及LED灯，已在内部与FPGA芯片相关引脚连接，所以在引脚分配时必须按照下表进行。若不需要用这26位开关及LED灯，则无需遵照。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **26开关** | **开关座编号** | **物理引脚** | **26灯** | **物理引脚** |
| KEY1 | U2 | 15 | LED1 | 2 |
| KEY2 | 11 | LED2 | 3 |
| KEY3 | 10 | LED3 | 4 |
| KEY4 | 8 | LED4 | 46 |
| KEY5 | 7 | LED5 | 45 |
| KEY6 | 6 | LED6 | 44 |
| KEY7 | 5 | LED7 | 43 |
| KEY8 | 57 | LED8 | 42 |
| KEY1 | U3 | 58 | LED9 | 41 |
| KEY2 | 59 | LED10 | 40 |
| KEY3 | 60 | LED11 | 36 |
| KEY4 | 61 | LED12 | 35 |
| KEY5 | 62 | LED13 | 34 |
| KEY6 | 63 | LED14 | 33 |
| KEY7 | 64 | LED15 | 32 |
| KEY8 | 65 | LED16 | 31 |
| KEY1 | U6 | 69 | LED17 | 30 |
| KEY2 | 70 | LED18 | 29 |
| KEY3 | 71 | LED19 | 28 |
| KEY4 | 72 | LED20 | 27 |
| KEY5 | 73 | LED21 | 26 |
| KEY6 | 76 | LED22 | 23 |
| KEY7 | 77 | LED23 | 22 |
| KEY8 | 78 | LED24 | 21 |
| KEY1 | U7 | 79 | LED25 | 20 |
| KEY2 | 80 | LED26 | 19 |