**华中科技大学**

**《数字电子技术基础》仿真实验报告**

|  |  |
| --- | --- |
| 实验名称： | 基础元件的verilog仿真 |
| 院（系）： | 电子信息与通信学院 |
| 专业班级： | 电信提高2001 |
| 姓名： | 陈芃 |
| 学号： | U202013828 |
| 时间： | 2021.12.25 |
| 指导教师： | 杨彩虹 |

2021 年 12 月 25 日

## 一、前言

①本次实验的仿真对象是《数字电子技术基础（第三版）》中所有给出了功能表的集成器件，可能遗漏一些第六版中新增的器件。

②所有器件对应的工程可在个人github仓库<https://github.com/Eynnzerr/VerilogSimfork>下载。

## 二、组合电路元件仿真

### ①8线-3线优先编码器CD4532

#### （1）源代码

module CD4532(

input EI,

input [7:0]I,

output [2:0]Y,

output GS,

output EO

);

reg [2:0]Y\_t;

parameter X1 = 8'b10000000;

parameter X2 = X1 >> 1;

parameter X3 = X2 >> 1;

parameter X4 = X3 >> 1;

parameter X5 = X4 >> 1;

parameter X6 = X5 >> 1;

parameter X7 = X6 >> 1;

assign EO = EI;

assign GS = EI & ~(I[0] | I[1] | I[2] | I[3] | I[4] | I[5] | I[6] | I[7]);

always @(\*) begin

if(I >= X1 && EI) Y\_t = 3'b111;

else if(I >= X2 && EI) Y\_t = 3'b110;

else if(I >= X3 && EI) Y\_t = 3'b101;

else if(I >= X4 && EI) Y\_t = 3'b100;

else if(I >= X5 && EI) Y\_t = 3'b011;

else if(I >= X6 && EI) Y\_t = 3'b010;

else if(I >= X7 && EI) Y\_t = 3'b001;

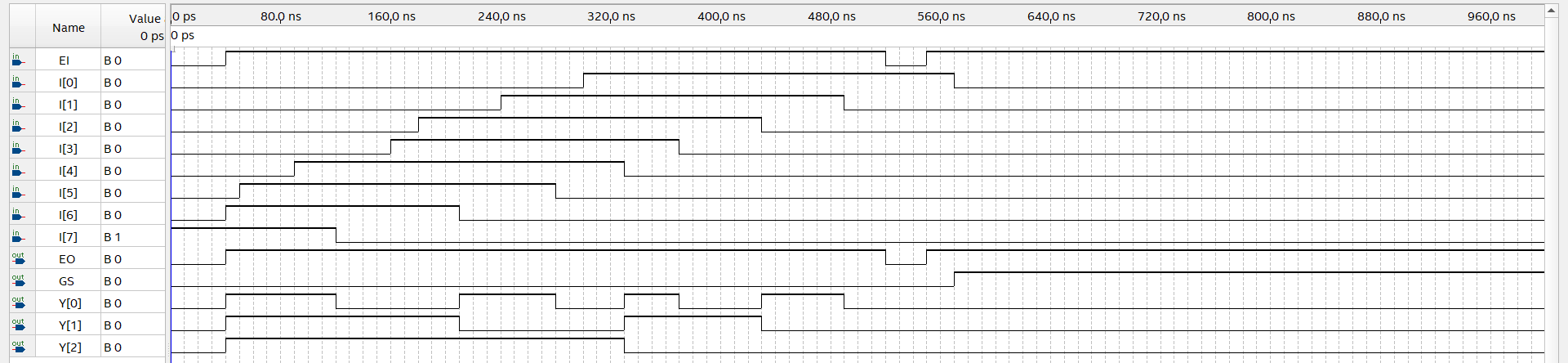
else Y\_t = 3'b000;

end

assign Y = Y\_t;

endmodule

#### （2）波形图



如图，在给定输入下，实现了编码的优先效果。

### ②3线-8线集成二进制译码器74HC138

#### （1）源代码

module decoder\_74HC138(

input [2:0] A,

input E3,E2,E1,

output reg [7:0] Y

);

always@(\*) begin

if(E3==1 && E2==0 && E1==0) begin

case(A)

3'b000: Y=8'b0111\_1111;

3'b001: Y=8'b1011\_1111;

3'b010: Y=8'b1101\_1111;

3'b011: Y=8'b1110\_1111;

3'b100: Y=8'b1111\_0111;

3'b101: Y=8'b1111\_1011;

3'b110: Y=8'b1111\_1101;

3'b111: Y=8'b1111\_1110;

endcase

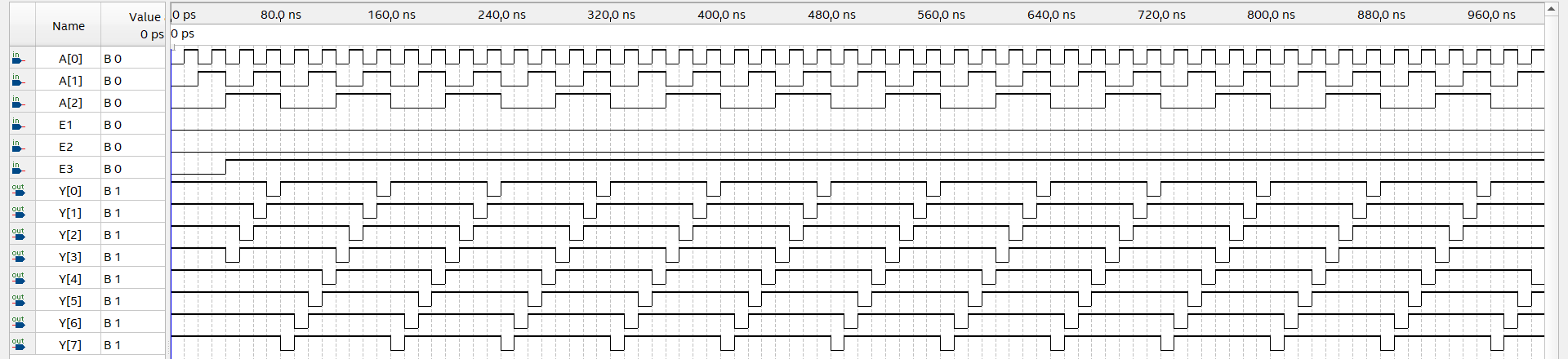
end

else Y=8'b1111\_1111;

end

endmodule

#### （2）波形图



如图，给定输入从000到111，输出正常，译码工作正常。

### ③七段显示译码器74HC4511

#### （1）源代码

module decoder\_74HC4511(

input LE,BL,LT,

input [3:0] D,

output reg [6:0] L

);

wire [2:0] E;

assign E = {LE,BL,LT};

always @ (\*)

begin

if(LE==0 && BL==1 && LT==1)

begin

case(D)

//0-9

4'b0000:L=7'b111\_1110;

4'b0001:L=7'b011\_0000;

4'b0010:L=7'b110\_1101;

4'b0011:L=7'b111\_1001;

4'b0100:L=7'b011\_0011;

4'b0101:L=7'b101\_1011;

4'b0110:L=7'b001\_1111;

4'b0111:L=7'b111\_0000;

4'b1000:L=7'b111\_1111;

4'b1001:L=7'b111\_1011;

//无效状态

4'b1010:L=7'b000\_0000;

4'b1011:L=7'b000\_0000;

4'b1100:L=7'b000\_0000;

4'b1101:L=7'b000\_0000;

4'b1110:L=7'b000\_0000;

4'b1111:L=7'b000\_0000;

endcase

end

else

begin

casex(E)

3'bxx0:L=7'b111\_1111; //灯测

3'bx01:L=7'b000\_0000; //灭灯

3'b111:L<=L; //锁存

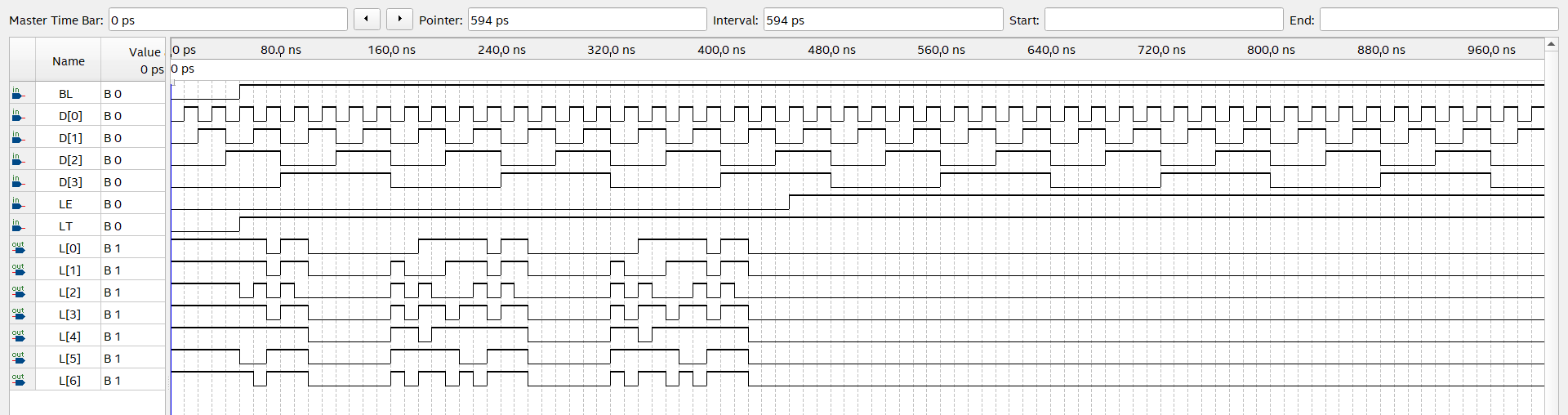
endcase

end

end

endmodule

#### （2）波形图



### ④数据选择器74HC151

#### （1）源代码

module selector\_74HC151(

input E,

input [2:0] S,

input [7:0] D,

output reg Y,

output Y1

);

always @ (\*)

begin

if(E) Y=0;

else begin

case(S)

3'b000:Y=D[0];

3'b001:Y=D[1];

3'b010:Y=D[2];

3'b011:Y=D[3];

3'b100:Y=D[4];

3'b101:Y=D[5];

3'b110:Y=D[6];

3'b111:Y=D[7];

endcase

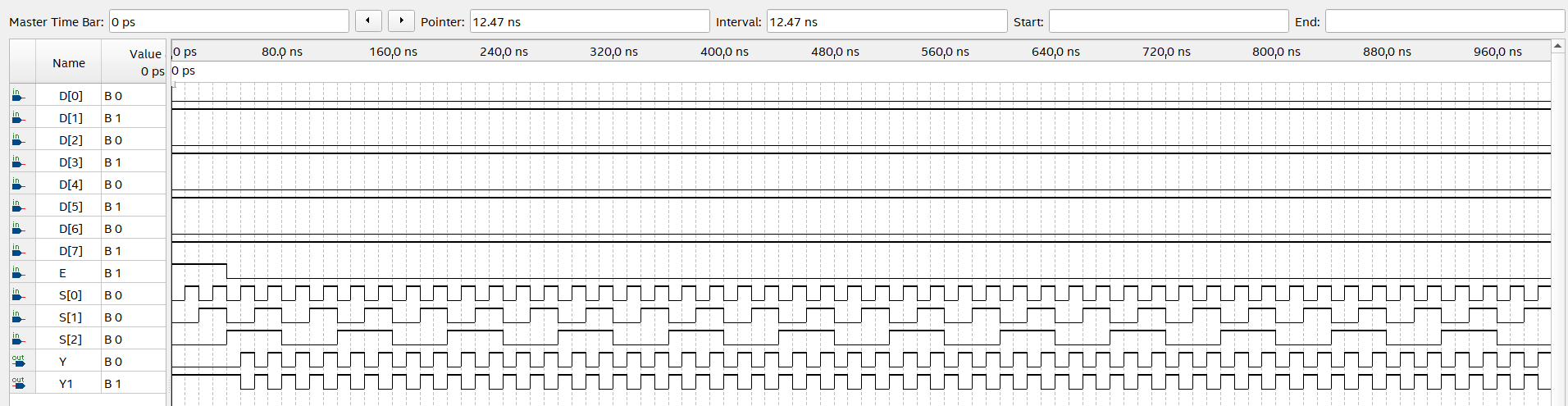
end

end

assign Y1 = ~Y;

endmodule

#### （2）波形图



### ⑤集成数值比较器74HC85

#### （1）源代码

module comparator\_74HC85(

input [2:0]I,

input [3:0] A,B,

output reg [2:0] L

);

always@(\*)

begin

if(I==3'b100) L=3'b100;

else if(I==3'b010) L=3'b010;

else

begin

if(A[3]>B[3]) L=3'b100;

else if(A[3]<B[3]) L=3'b010;

else

begin

if(A[2]>B[2]) L=3'b100;

else if(A[2]<B[2]) L=3'b010;

else

begin

if(A[1]>B[1]) L=3'b100;

else if(A[1]<B[1]) L=3'b010;

else

begin

if(A[0]>B[0]) L=3'b100;

else if(A[0]<B[0]) L=3'b010;

else L=3'b001;

end

end

end

end

end

endmodule

#### （2）波形图

略

## 三、时序电路元件仿真

### ①4位双向移位寄存器74HC194

#### （1）源代码

module shift74HC194(

input S1,S0,Dsl,Dsr,CP,CR,

input [3:0]D,

output reg[3:0]Q

);

parameter ZERO = 4'b0000; //清零

parameter case1 = 2'b00, case2 = 2'b01, case3 = 2'b10, case4 = 2'b11;

always @ (posedge CP, negedge CR)

if(~CR) Q <= ZERO; //异步置零

else

case({S1,S0})

case1: Q <= Q; //保持

case2: Q <= {Q[2:0],Dsr}; //右移（低位向高位移位）

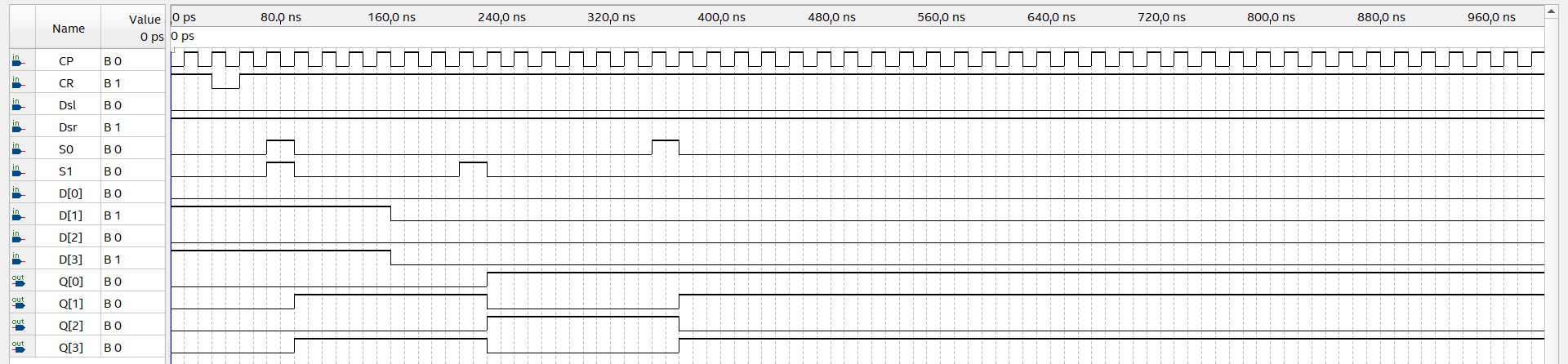
case3: Q <= {Dsl,Q[3:1]}; //左移（高位向低位移位）

case4: Q <= D; //并行置数

endcase

endmodule

#### （2）波形图



### ②集成计数器74LVC161

#### （1）源代码

module counter74LVC161(

input CET,CEP,PE,CP,CR,

input [3:0]D,

output TC,

output reg[3:0]Q

);

parameter CARRY = 4'b1111; //进位

parameter ZERO = 4'b0000; //清零

parameter BIT = 1'b1; //计数

wire CE; //中间变量

assign CE = CET & CEP;

assign TC = CET & PE & (Q==CARRY); //产生进位信号

always @ (posedge CP, negedge CR)

if(~CR) Q <= ZERO; //异步清零信号有效，直接清零

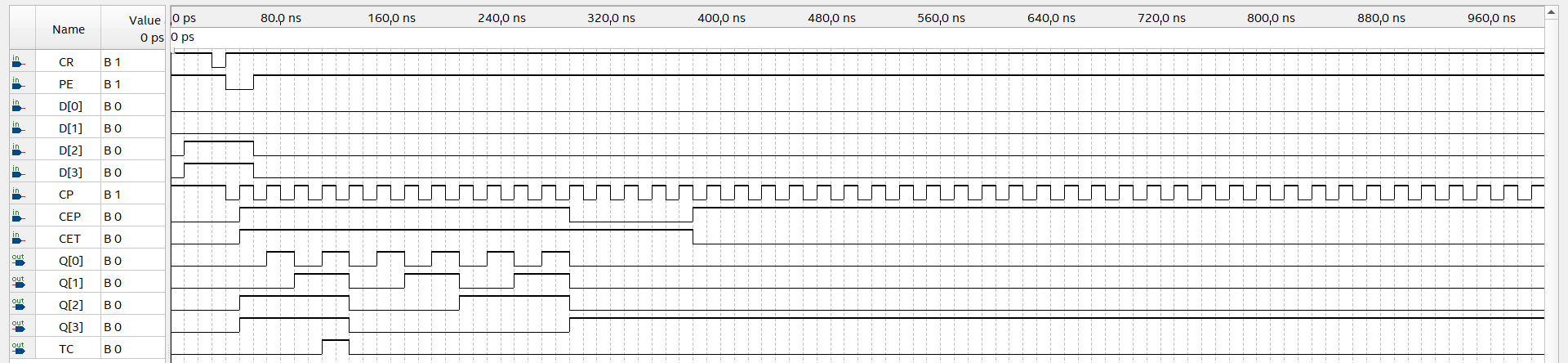
else if(~PE) Q <= D; //同步置数信号有效，实现预置

else if(CE) Q <= Q + BIT; //CET,CEP均为高电平，正常计数

else Q <= Q; //否则保持

endmodule

#### （2）波形图



在给定输入下，正常计数。

### ③寄存器74HC374

#### （1）源代码

module reg74HC374(

input OE,CP,

input [7:0] D,

output reg [7:0] Q

);

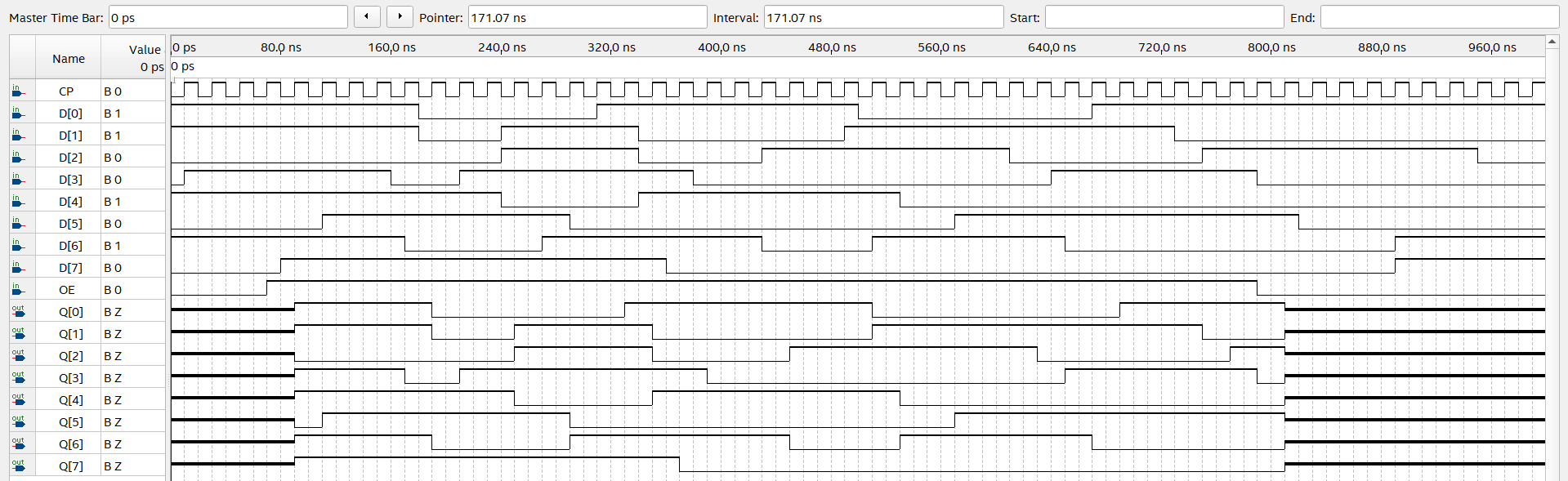
always @ (posedge CP)

if(OE) Q <= D;//使能端为高电平，即~OE低电平，输出内部触发器状态

else Q <= 8'bz;//使能端为低电平，即~OE高电平，输出高阻抗

endmodule

#### （2）波形图



如图，使能端为低电平时，输出禁止，全为高阻态；使能端为高电平时，输出内部触发器存储的状态。寄存器工作正常。

### ④D锁存器74HC373

#### （1）源代码

module latch\_74HC373(

input OE,LE,

input [7:0] D,

output reg [7:0] Q

);

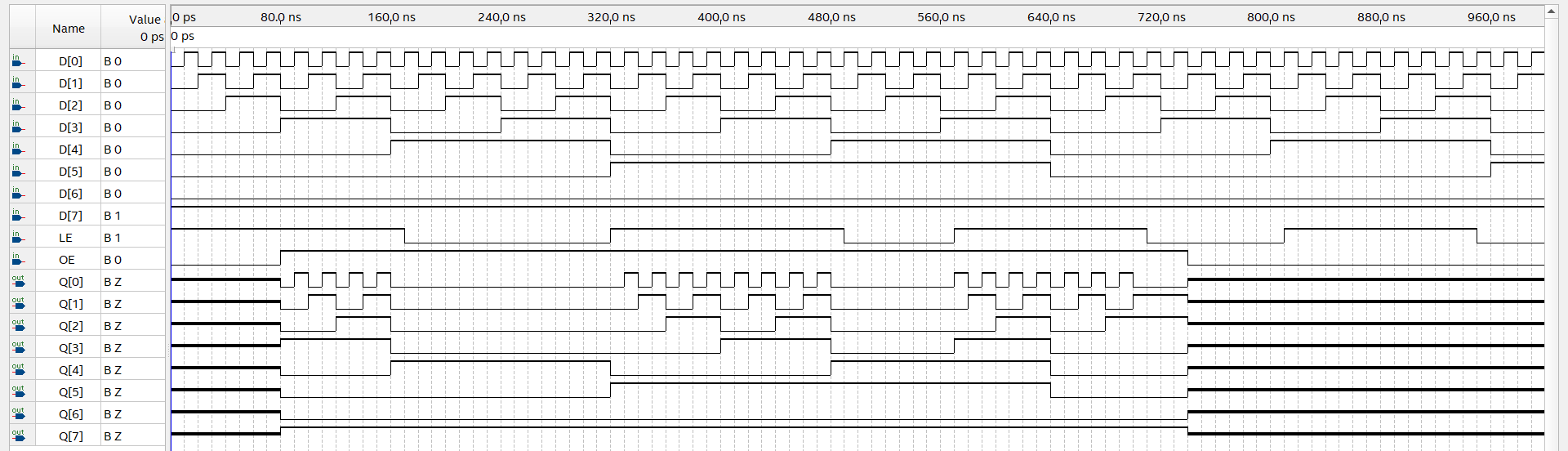
always @ (OE or LE or D)

if(!OE) Q = 8'bz;//禁止输出，低电平有效

else if(LE) Q = D;//使能（高电平）和读锁存器模式

endmodule

#### （2）波形图



如图，当OE为低电平时，锁存器禁止输出，输出全为高阻态；当OE为高电平，LE为低电平时，锁存器锁存，输出保持不变；当OE为高电平，LE为高电平时，锁存器更新，输出写入的值。D锁存器工作正常。