**实验报告**

实验题目： 架构向导和IP目录 日期： 2018 年 11 月 16 日

姓名: 罗晏宸 学号: PB17000297 成绩:\_\_\_\_\_\_\_\_\_\_\_\_

**实验目的：**

1. 学习FPGA架构与IP核例化的基础知识与使用流程
2. 通过正确的配置和实例化，有效的使用专业和高级的架构
3. 使用IP目录工具了解并配置时钟与计数器架构，认识其黑盒特性与各属性
4. 掌握七段数码管动态扫描显示原理和具体实现
5. 使用例化的时钟与计数器以及七段数码管显示的技术，实现实际功能

**实验内容**（截图、照片与代码）

Lab8\_1\_1

代码lab8\_1\_1.v

`timescale 1ns / 1ps

module lab8\_1\_1(input CLK100MHZ,output rst\_n,output pulse);

    wire clk;

clk\_wiz\_0(clk,rst\_n,CLK100MHZ);

Counter(clk,rst\_n,pulse);

endmodule

代码Counter.v

`timescale 1ns / 1ps

module Counter(input clk,input rst\_n,output pulse);

     reg [23:0] cnt;

always @(posedge clk or negedge rst\_n)

begin

   if(~rst\_n)

       cnt <= 'd0;

   else if(cnt >= 'd4999999)

       cnt <= 'd0;

   else

       cnt <= cnt + 'd1;

end

assign pulse = (cnt >= 'd2499999) ? 1'b1 : 1'b0;

endmodule



照片 1-lab8\_1\_1下载A



照片 2-lab8\_1\_1下载B

“照片 1-lab8\_1\_1下载A”与“照片 2-lab8\_1\_1下载B”的说明：本工程使用实例化时钟IP核Clock Wizard实现一个时钟周期为一秒的高电平脉冲信号；照片中的LED[0]每秒内交替闪烁，实现脉冲信号的物理显示。

Lab8\_1\_2

代码lab8\_1\_2.v

`timescale 1ns / 1ps

module lab8\_1\_2(

input [3:0] v,

input CLK100MHZ,

output [7:0] an,

output [6:0] seg

);

wire [3:0] m;

wire [3:0] temp = 4'b0001;

wire clk\_out,pulse,z;

wire [6:0] seg0,seg1;

clk\_wiz\_0(clk\_out,CLK100MHZ);

lab8\_1\_1(clk\_out,pulse);

lab2\_2\_1\_partA (v,z,m);

bcdto7segment\_dataflow (m,seg0);

bcdto7segment\_dataflow (temp,seg1);

assign seg = (pulse) ? seg0 : seg1;

assign an[0] = (pulse) ? 0 : 1;

assign an[1] = (!pulse & (z==1)) ? 0 : 1;

assign an[2] = 1;

assign an[3] = 1;

assign an[4] = 1;

assign an[5] = 1;

assign an[6] = 1;

assign an[7] = 1;

endmodule

代码lab8\_1\_1.v

`timescale 1ns / 1ps

module lab8\_1\_1(input CLK100MHZ,output pulse);

    wire clk;

clk\_wiz\_0(clk,CLK100MHZ);

Counter(clk,pulse);

endmodule

代码Counter.v

`timescale 1ns / 1ps

module Counter(input clk,output pulse);

     reg [13:0] cnt;

always @(posedge clk)

begin

        if(cnt >= 'd999)

       cnt <= 'd0;

   else

       cnt <= cnt + 'd1;

end

assign pulse = (cnt >= 'd499) ? 1'b1 : 1'b0;

endmodule

代码lab2\_2\_1\_partA.v

`timescale 1ns / 1ps

module lab2\_2\_1\_partA(

input [3:0] v,

output z,

output [3:0] m

);

wire [3:0] CircuitAOutput;

assign z = (v[3] & v[2]) | (v[3] & v[1]);

assign CircuitAOutput[3] = 0;

assign CircuitAOutput[2] = v[1] & v[2];

assign CircuitAOutput[1] = ~v[1];

assign CircuitAOutput[0] = v[0];

assign m[3] = (~z & v[3]) | (z & CircuitAOutput[3]);

assign m[2] = (~z & v[2]) | (z & CircuitAOutput[2]);

assign m[1] = (~z & v[1]) | (z & CircuitAOutput[1]);

assign m[0] = (~z & v[0]) | (z & CircuitAOutput[0]);

endmodule

代码bcdto7segment\_dataflow.v

`timescale 1ns / 1ps

module bcdto7segment\_dataflow(

input [3:0] x,

output [6:0] seg

);

assign seg[0] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[1] = (~x[3] && x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && ~x[0]) | (x[3] && (x[2] | x[1]));

assign seg[2] = (~x[3] && ~x[2] && x[1] && ~x[0]) | (x[3] && (x[2] | x[1]));

assign seg[3] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[4] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && ~x[2] && x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0]) | (~x[3] && x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0]) | (x[3] && ~x[2] && ~x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[5] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && ~x[2] && x[1] && ~x[0]) | (~x[3] && ~x[2] && x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[6] = ((~x[3] && ~x[2] && ~x[1] && ~x[0]) | (~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

endmodule



照片 3-lab8\_1\_2下载

“照片 3-lab8\_1\_2下载”的说明：本工程使用实例化时钟IP核Clock Wizard与动态扫描技术实现BCD输入到两位七段数码管的显示；照片中switch[0], switch[2]与switch[4]为开，其余开关为关，表示输入为v=(1101)­B；右侧两位七段数码管显示为十进制两位数“13”字样，表示输出为(13)D

Lab8\_2\_1

代码lab8\_2\_1.v

`timescale 1ns / 1ps

module lab8\_2\_1(input CLK100MHZ,input rst,input enable,output [7:0] an,output [6:0] seg);

    wire pulse,tick,threshold,clk;

    wire [3:0] Q0,Q1;

    wire [6:0] seg0,seg1;

    clk\_wiz\_0(clk,CLK100MHZ);

    Counter0(clk,enable,tick);

    c\_counter\_binary\_0 (tick,1,threshold,Q0);

    c\_counter\_binary\_0 (tick,threshold,,Q1);

    Counter1(clk,pulse);

    bcdto7segment\_dataflow (Q0,seg0);

    bcdto7segment\_dataflow (Q1,seg1);

    assign seg = (pulse) ? seg0 : seg1;

    assign an[0] = (pulse) ? 0 : 1;

    assign an[1] = (!pulse && Q1) ? 0 : 1;

    assign an[2] = 1;

    assign an[3] = 1;

    assign an[4] = 1;

    assign an[5] = 1;

    assign an[6] = 1;

    assign an[7] = 1;

endmodule

代码Counter0.v

`timescale 1ns / 1ps

module Counter0(input clk,input rst,output pulse);

     reg [23:0] cnt;

always @(posedge clk or posedge rst)

begin

   if (rst)

   cnt <= 'd0;

   else

       if(cnt >= 'd4999999)

       cnt <= 'd0;

   else

       cnt <= cnt + 'd1;

end

assign pulse = (cnt == 'd4999999) ? 1'b1 : 1'b0;

endmodule

代码Counter1.v

`timescale 1ns / 1ps

module Counter1(input clk,output pulse);

     reg [23:0] cnt;

always @(posedge clk)

begin

        if(cnt >= 'd9999)

       cnt <= 'd0;

   else

       cnt <= cnt + 'd1;

end

assign pulse = (cnt >= 'd4999) ? 1'b1 : 1'b0;

endmodule

代码bcdto7segment\_dataflow.v

`timescale 1ns / 1ps

module bcdto7segment\_dataflow(

input [3:0] x,

output [6:0] seg

);

assign seg[0] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[1] = (~x[3] && x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && ~x[0]) | (x[3] && (x[2] | x[1]));

assign seg[2] = (~x[3] && ~x[2] && x[1] && ~x[0]) | (x[3] && (x[2] | x[1]));

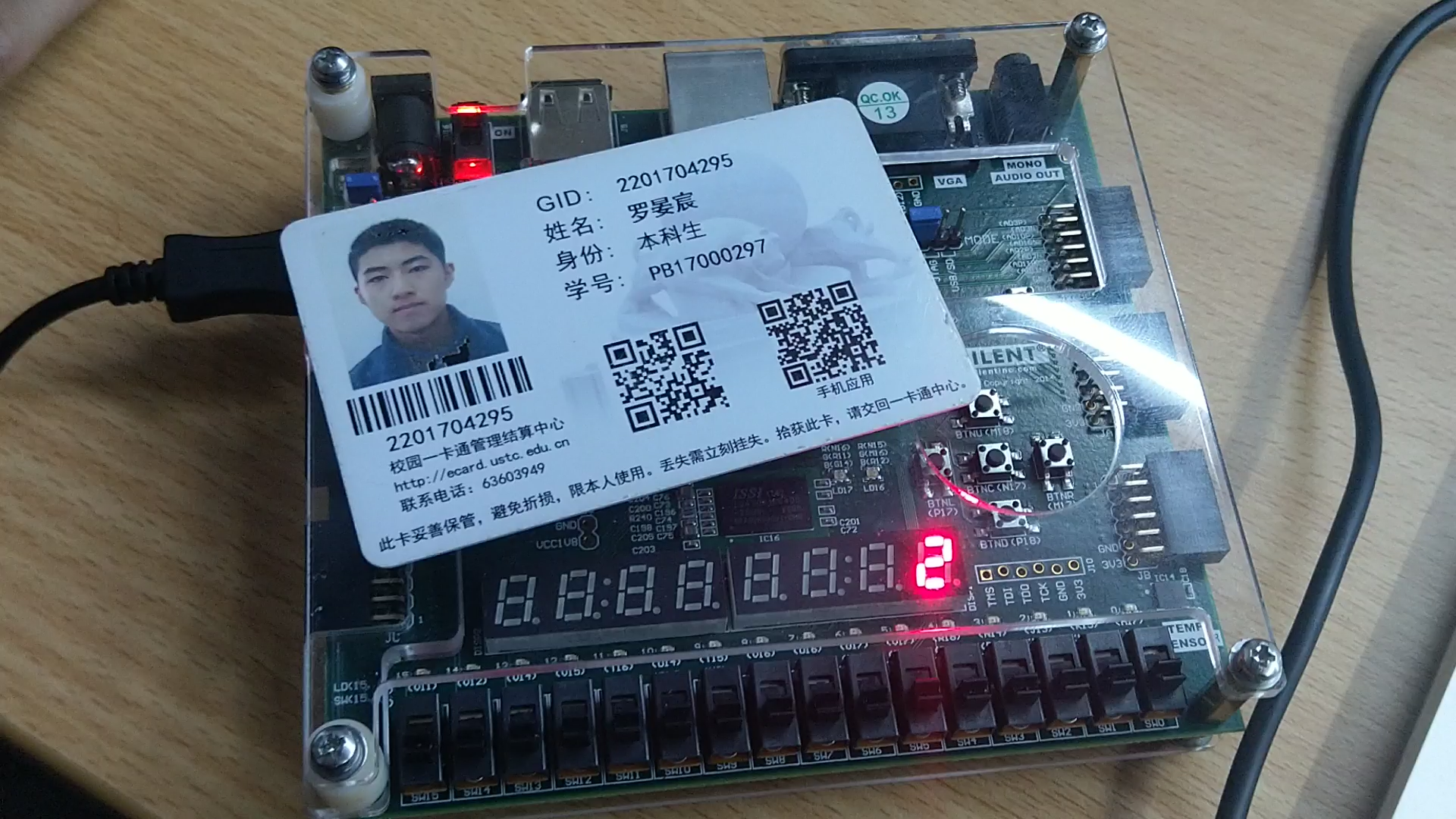
assign seg[3] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[4] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && ~x[2] && x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0]) | (~x[3] && x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0]) | (x[3] && ~x[2] && ~x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[5] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && ~x[2] && x[1] && ~x[0]) | (~x[3] && ~x[2] && x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[6] = ((~x[3] && ~x[2] && ~x[1] && ~x[0]) | (~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

endmodule

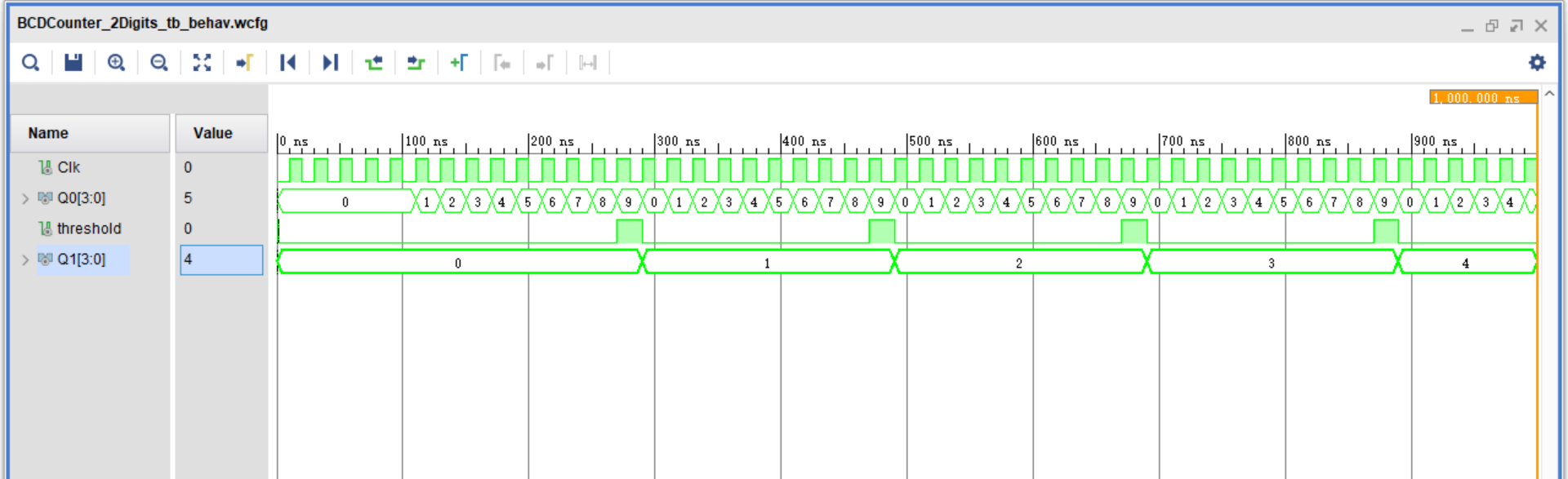


照片 4-lab8\_2\_1下载

“照片 4-lab8\_2\_1下载”的说明：本工程使用例化时钟与计数器IP核，并使用动态扫描技术实现每秒加1，计数范围为0-99的计数器，并将十进制计数结果以两位七段数码管形式输出。图中是计数器在重置为0后再次计数为2的状态。

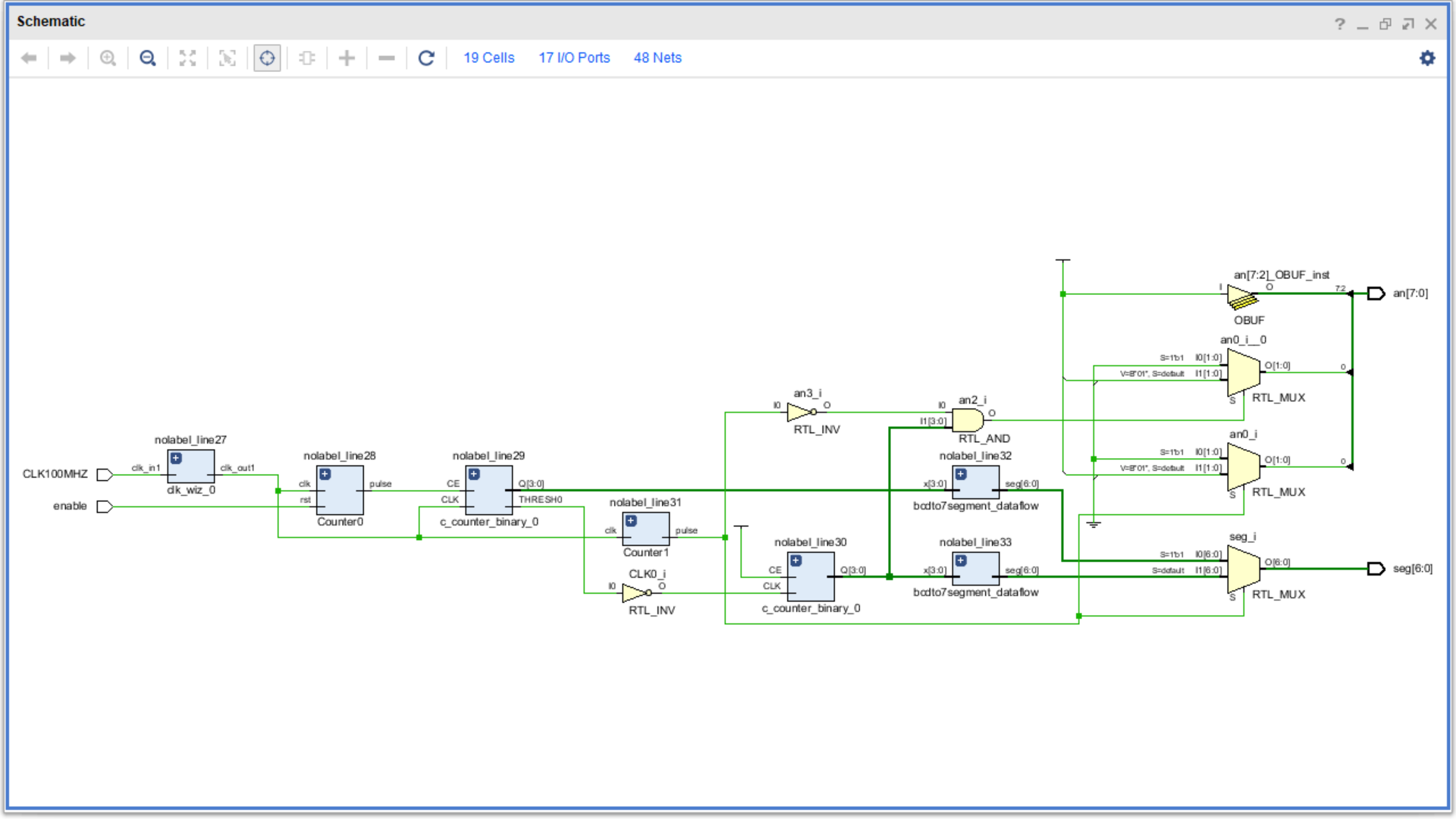
**课后题**

1. 对两位数的BCD计数器进行仿真，以确保功能正确



截图 1-课后题1仿真

1. 2\_1实验中十进制计数器使用5MHz时钟控制，用1Hz周期信号作为计数器时钟使能信号（CE）
   1. 电路图



截图 2-课后题2原理

* 1. 代码lab8\_homework\_2.v

`timescale 1ns / 1ps

module lab8\_homework\_2 (input CLK100MHZ,input enable,output [7:0] an,output [6:0] seg);

    wire pulse,tick,threshold,clk;

    wire [3:0] Q0,Q1;

    wire [6:0] seg0,seg1;

    clk\_wiz\_0(clk,CLK100MHZ);

    Counter0(clk,enable,tick);

    c\_counter\_binary\_0 (clk,tick,threshold,Q0);

    c\_counter\_binary\_0 (~threshold,1,,Q1);

    Counter1(clk,pulse);

    bcdto7segment\_dataflow (Q0,seg0);

    bcdto7segment\_dataflow (Q1,seg1);

    assign seg = (pulse) ? seg0 : seg1;

    assign an[0] = (pulse) ? 0 : 1;

    assign an[1] = (!pulse && Q1) ? 0 : 1;

    assign an[2] = 1;

    assign an[3] = 1;

    assign an[4] = 1;

    assign an[5] = 1;

    assign an[6] = 1;

    assign an[7] = 1;

endmodule

代码Counter0.v

`timescale 1ns / 1ps

module Counter0(input clk,input rst,output pulse);

     reg [23:0] cnt;

always @(posedge clk or posedge rst)

begin

   if (rst)

   cnt <= 'd0;

   else

       if(cnt >= 'd4999999)

       cnt <= 'd0;

   else

       cnt <= cnt + 'd1;

end

assign pulse = (cnt == 'd4999999) ? 1'b1 : 1'b0;

endmodule

代码Counter1.v

`timescale 1ns / 1ps

module Counter1(input clk,output pulse);

     reg [23:0] cnt;

always @(posedge clk)

begin

        if(cnt >= 'd9999)

       cnt <= 'd0;

   else

       cnt <= cnt + 'd1;

end

assign pulse = (cnt >= 'd4999) ? 1'b1 : 1'b0;

endmodule

代码bcdto7segment\_dataflow.v

`timescale 1ns / 1ps

module bcdto7segment\_dataflow(

input [3:0] x,

output [6:0] seg

);

assign seg[0] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[1] = (~x[3] && x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && ~x[0]) | (x[3] && (x[2] | x[1]));

assign seg[2] = (~x[3] && ~x[2] && x[1] && ~x[0]) | (x[3] && (x[2] | x[1]));

assign seg[3] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[4] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && ~x[2] && x[1] && x[0]) | (~x[3] && x[2] && ~x[1] && ~x[0]) | (~x[3] && x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0]) | (x[3] && ~x[2] && ~x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[5] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && ~x[2] && x[1] && ~x[0]) | (~x[3] && ~x[2] && x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

assign seg[6] = ((~x[3] && ~x[2] && ~x[1] && ~x[0]) | (~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2] && x[1] && x[0])) && ~(x[3] && (x[2] | x[1]));

endmodule

* 1. 下载



照片 5-课后题2下载

“照片 5-课后题2下载”的说明：本工程使用例化时钟与计数器IP核，并使用动态扫描技术实现每秒加1，计数范围为0-99的计数器，并将十进制计数结果以两位七段数码管形式输出。图中是计数器在若干次重置为0后再次计数为97的状态。