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

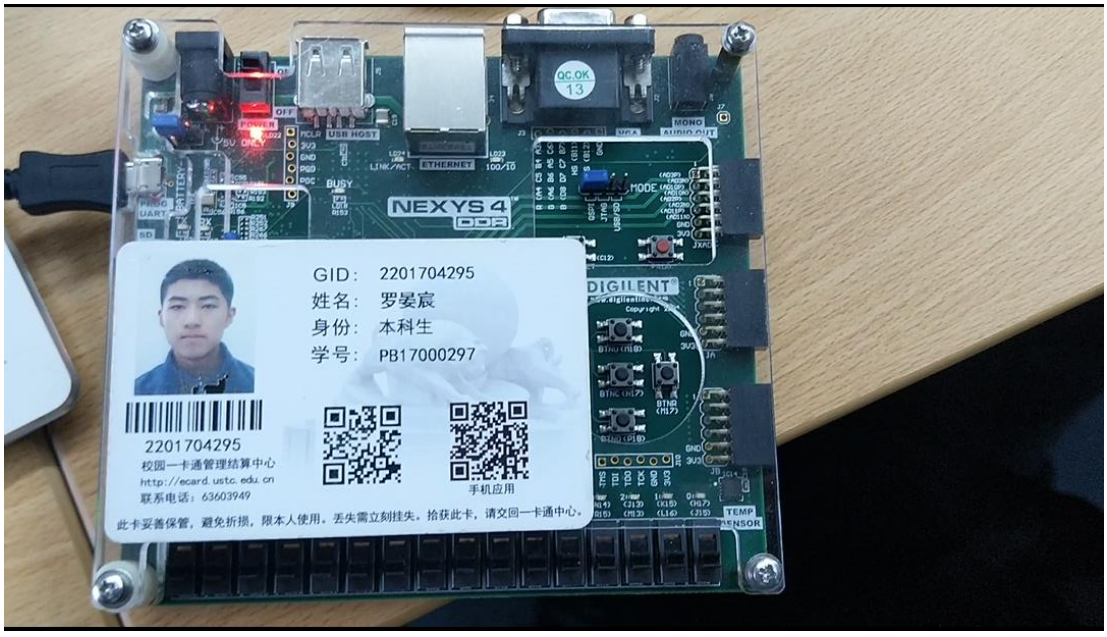
代码 lab8_1_1.v

```
`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

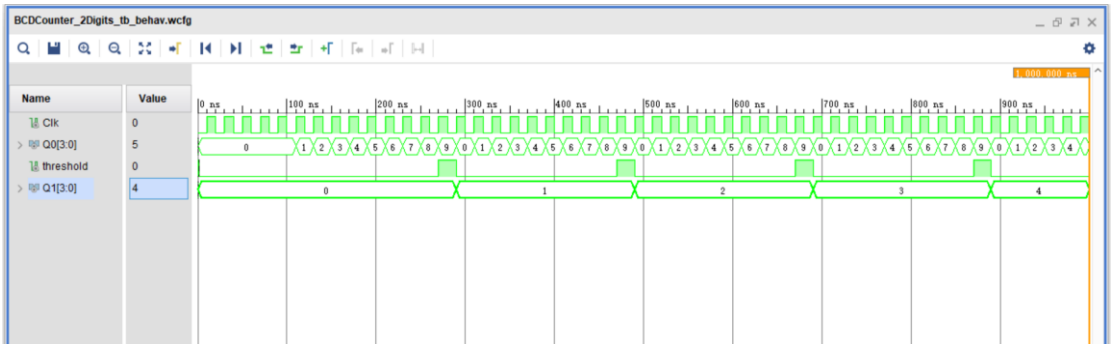
```
`timescale 1ns / 1ps
module bcdto7segment_dataflow(
    input [3:0] x,
    output [6:0] seg
    );
    assign seg[0] = ((~x[3] && ~x[2] && ~x[1] && x[0]) | (~x[3] && x[2]
&& ~x[1] && ~x[0])) && ~(x[3] && (x[2] | x[1]));
    assign seg[1] = (~x[3] && x[2] && ~x[1] && x[0]) | (~x[3] && x[2] &&
x[1] && ~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

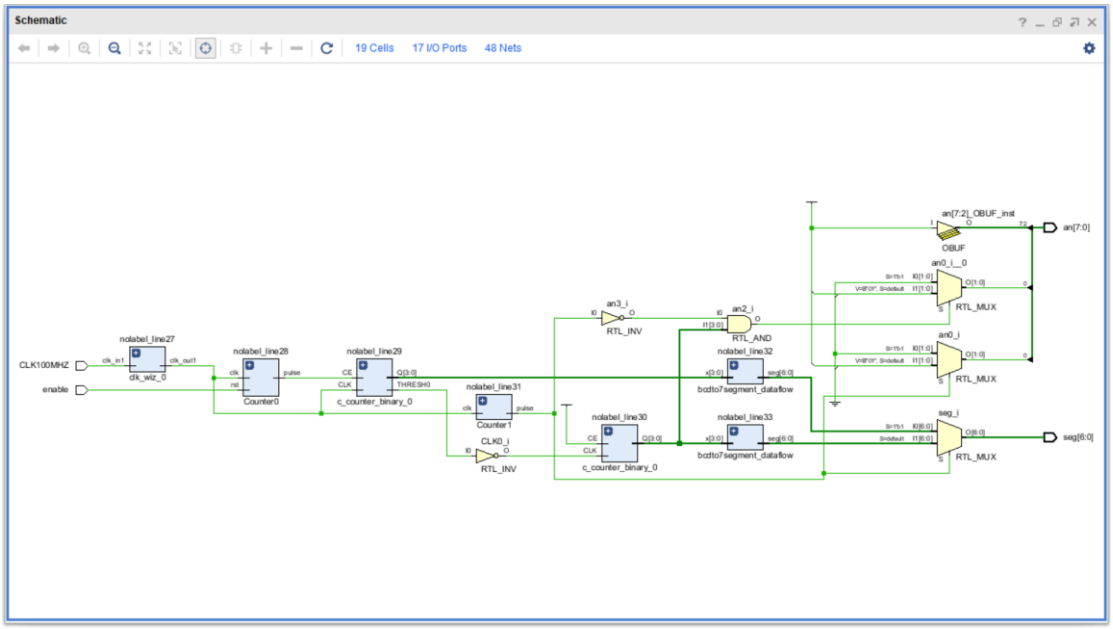
1. 对两位数的 BCD 计数器进行仿真，以确保功能正确



截图 1-课后题 1 仿真

2. 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

2) 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
```

3) 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。