



重慶理工大學

实验报告书

(电子版本)

实验课堂表现			实验报告成绩	实验总成绩
A ()	B ()	C ()		

实验名称: 数字 0-9 的显示

专业班级: 121070204

学 号: _____

姓 名: _____

联系电话: _____

指导老师: 王培容

实验时间: 2023/03/22

电气电子工程学院 信息处理实验室(中心)

一、 实验目的

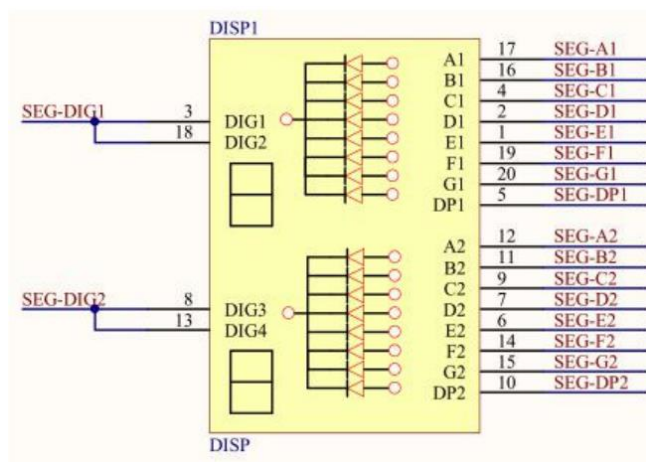
- (1) 熟悉和掌握 FPGA 开发流程和 Quartus 软件使用方法。
- (2) 编写分频器，产生 1 秒时钟。
- (3) 编写计数器，输出 0-9 的数字。
- (4) 编写译码器，显示数字 0-9。
- (5) 建立顶层原理图文件，完成在小脚丫开发板上循环显示数字 0-9。

二、 实验原理

分频器是一种常见的数字电路，用于将输入时钟信号分频成较低频率的信号。

计数器是一种非常基础和常见的电路，用于在时钟信号的控制下，对某个特定的值进行计数。在 Verilog 中，计数器可以通过组合逻辑或时序逻辑的方式来实现。

七段数码管是一种常用的数字显示设备，能够显示数字 0 到 9 以及一些字母和符号。为了控制七段数码管的显示，需要一个译码器将输入的数字转换为七段数码管的控制信号。



小脚丫 STEP MAX10 板载数码管电路图

译码器是一种基础的数字电路，用于将一个或多个输入信号转换为一个或多个输出信号。

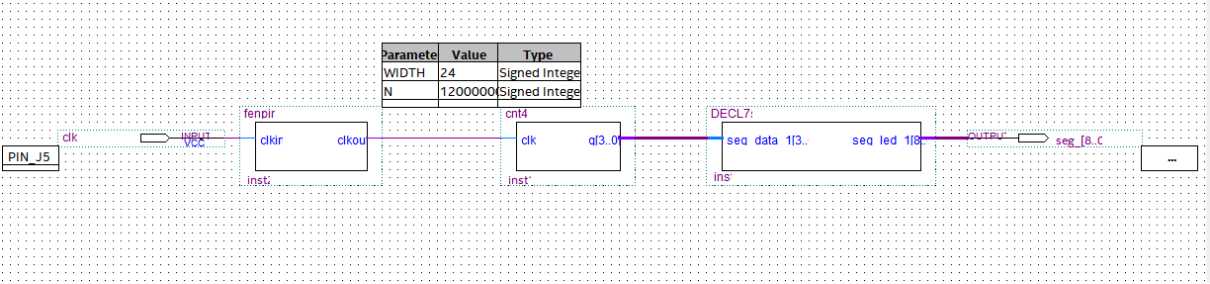
三、 实验器材

小脚丫 STEP MAX10 FPGA 开发板
Quartus II 软件

四、 实验要求及实验内容

1. 打开 Quartus 建立工程，以模块化的思路设计程序。
2. 完成分频器，用 12M 的时钟频率产生 1 秒时钟，进行编译设计，生成*.bsf 文件。
3. 完成计数器，利用之前产生的 1 秒时钟，每秒输出 0-9 数字，进行编译设计，并生成*.bsf 文件。
4. 完成译码器，对照七段数码管的译码表，将计数器产生的数字用数码管显示，进行编译设计，生成*.bsf 文件。

5. 建立顶层模块，将分频器、计数器和译码器组合在一起并分配管脚。



顶层模块

信号	引脚	信号	引脚
clk	J5	seg[4]	G2
seg[8]	E2	seg[3]	J2
seg[7]	L1	seg[2]	K2
seg[6]	G5	seg[1]	D2
seg[5]	F2	seg[0]	E1

管脚分配

- 6. 如果仿真无误，构建并输出编程文件，烧写至 小脚丫 FPGA 的 Flash 之中。
- 7. 观察输出结果。

五、 实验数据处理及结果分析

数据处理：

- 1. 小脚丫 FPGA 的数码管上顺利循环显示 0-9 的数字。

结果分析：

- 1. 通过顶层模块的组合，我们可以将多个子模块组合在一起，形成更加复杂的数字电路。

六、 实验结论

- 1. 通过分频器可以得到任意低于自带时钟频率（12M）的时钟，本次实验设计的是偶数分频，相对简单容易理解。
- 2. 计数器可根据时钟脉冲完成按时技术，本次实验设计的是 0-9 的计数器。
- 3. 译码器可由输入信号控制对应的 LED 点亮，本次实验平台小脚丫 STEP-Max10 上有两位共阴极数码管，要注意的是共阴极和共阳极的数码管译码表是不同的。
- 4. 通过本次实验，我们熟悉了 FPGA 开发流程和 Quartus II 软件的使用方法，并掌握了数字电路设计的基本原理。在实验中，我们成功地设计了分频器、计数器和译码器，并将它们组合成了一个数字时钟。同时，还学习了如何在顶层模块中组合子模块。

七、 实验中所遇到的困难及解决办法（至少 2 点）

八、

困难：

- 1. 编写程序时，容易漏掉配对的 end。
- 2. 在编辑*.bdf 文件连线时，易混淆总线和普通线。
- 3. 在连线时，发现有一些线在连接时肉眼认为连上了但实际上没有连上。

解决办法:

1. 养成良好的代码习惯, 注意代码的高亮提醒。
2. 在模块的引脚处, 若引出的线较粗则应该使用总线。
3. 注意到当普通线/总线与模块连接上时, 模块引脚处会出现小方框。

九、 程序源码

分频器模块:

```
1  module Divider(clkin,clkout);
2      input clkin;
3      output clkout;
4      parameter WIDTH = 24;
5      parameter N=12000000;
6      reg [WIDTH-1:0] cntp;
7      reg clkout;
8      always @(posedge clkin)
9      begin
10         if(cntp==(N-1))
11             cntp<=0;
12         else cntp<=cntp+1;
13     end
14     always @(posedge clkin)
15     begin
16         if(cntp<(N>>1))
17             clkout<=0;
18         else
19             clkout<=1;
20     end
21 endmodule
```

计数器模块:

```
1  module cnt_4(clk,q);
2      input clk;
3      output [3:0]q;
4      reg [3:0]q;
5      always@(posedge clk)
6      begin
7          if(q<9)
8              q<=q+1;
9          else
10             q<=0;
11         end
12 endmodule
```

译码器模块:

```
1  module DECL7S(seg_data_1,seg_led_1);
2  input  [3:0] seg_data_1;
3  output [8:0] seg_led_1;
4  reg [8:0] seg [15:0];
5  initial
6  begin
7      seg[0] = 9'h3f;
8      seg[1] = 9'h06;
9      seg[2] = 9'h5b;
10     seg[3] = 9'h4f;
11     seg[4] = 9'h66;
12     seg[5] = 9'h6b;
13     seg[6] = 9'h7b;
14     seg[7] = 9'h07;
15     seg[8] = 9'h7f;
16     seg[9] = 9'h6f;
17     seg[10] = 9'h77;
18     seg[11] = 9'h7c;
19     seg[12] = 9'h39;
20     seg[13] = 9'h5e;
21     seg[14] = 9'h79;
22     seg[15] = 9'h71;
23 end
24 assign seg_led_1 = seg[seg_data_1];
25 endmodule
```