

实验平台说明

特征	说明
操作系统版本和系统类型	Windows 7 64 bit enterprise edition
EDA 软件名称和版本	Quartus II 13.1 64 bit
DE2-115 开发板编号	1904615S

自查清单

特征	Part 1/Lab1	Part ?/Lab?	Part ?/Lab?	Part ?/Lab?
原理图	√			
Verilog 代码	√			
Flow Summary	√			
RTL 图	×			
状态机图	×			
工艺图	√			
仿真图	×			
SignalTapII 图	×			
硬件运行图	√			

第一部分 Part 1/Lab1

一、实验目的

- 1.掌握建立工程，VerilogHDL 开发的基本方法。
- 2.掌握编译，下载的基本方法。

二、实验内容

- 1.实验首次向开发板中编译，下载程序。
- 2.实现使用按键控制一个计数灯的计数速度。
- 3.理解锁相环原理及该工程原理。

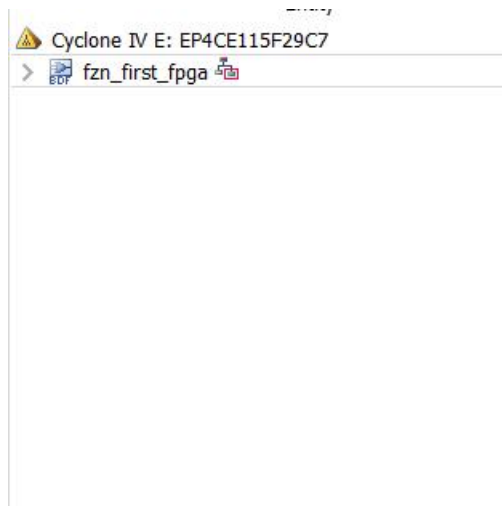
三、实验原理

- 1.输入 CLOCK_5 为一时钟输入 50MHZ，经过十分频锁相环分为 5MHZ。
- 2.5MHZ 通过 simple count 计数器为 counter[31,,0]赋值，该数据从 0x0000 到 0xffff。
- 3.通过复用寄存器使 counter[26,,23]或 counter[24,,21]输出至 output。并由按键控制前后者。
- 4.当按键按下的时候；为后者 counter[24,,21]， $2^{21}/5M = 0.4194s$ ，即计数间隔为 0.4194s；
- 5.当按键未按下的时候；为后者 counter[26,,23]， $2^{23}/5M = 1.677s$ ，即计数间隔为 1.677s；可以显著观察到试验效果

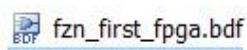
四、实验步骤

1) 建立工程，并添加文件。

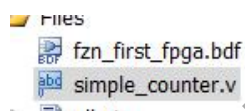
- 1.建立工程，选择常用文件夹，取名为 fzn_first_fpga。



2.添加 Block Diagram/Schematic 文件，取名为 fzn_first_fpga;



3.添加 Verilog HDL 文件，取名为 simple_counter.v



2) 开发

1.按要求键入

//It has a single clock input and a 32-bit output port

```
module simple_counter (
```

```
CLOCK_50,
```

```
counter_out
```

```
);
```

```
input CLOCK_50 ;
```

```
14
```

```
output [31:0] counter_out;
```

```
reg [31:0] counter_out;
```

always @ (posedge CLOCK_50) // on positive clock edge

begin

counter_out <= #1 counter_out + 1; // increment counter

end

endmodule

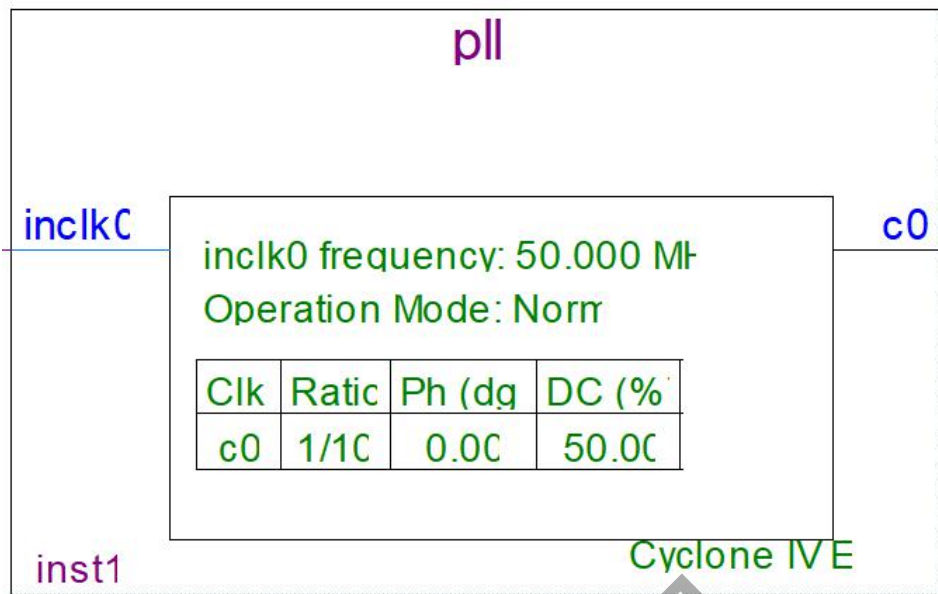
代

码

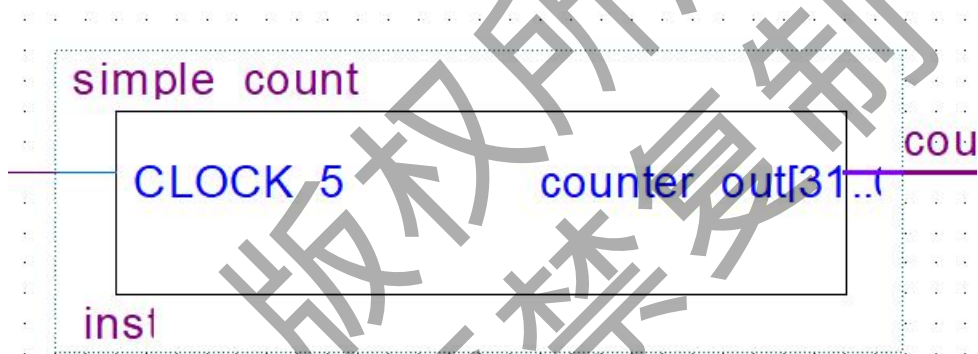
```
module simple_counter (
    CLOCK_50,
    counter_out
);
    input    CLOCK_50 ;
    output [31:0] counter_out;
    reg [31:0] counter_out;
    always @ (posedge CLOCK_50) // on positive clock edge
begin
    counter_out <= #1 counter_out + 1; // increment counter
end
endmodule // end of module counter
```

2.在 bdf 文件中，双击将 simple_counter 所代表的元件输入至 fzn_first_fpga.bdf 中。

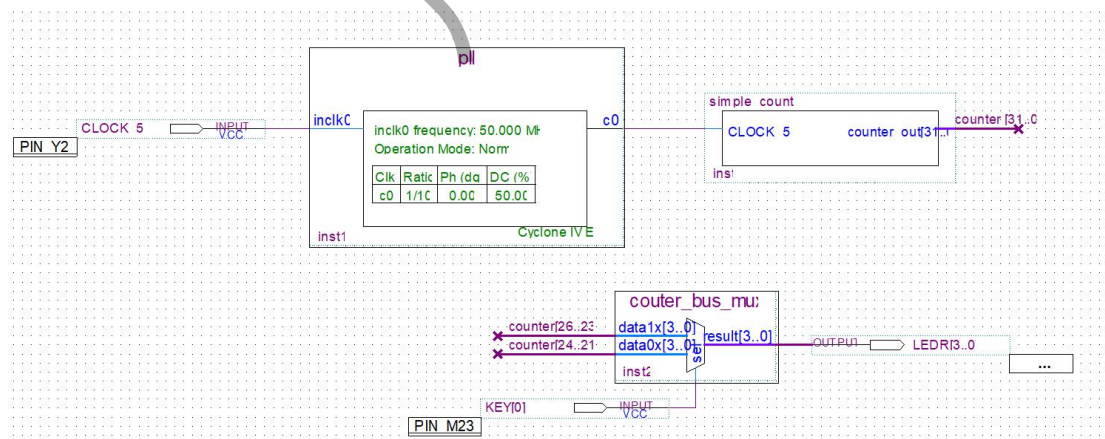
3.使用 Quatus 的 PLL 宏功能添加一个名为 ALTPLL 的预构建 LPM 宏功能。



4.添加多路复用寄存器；lpm_mux。该设计将计数器总线的两种输出多路复用到 DE2-115 开发板上的四个 LED。



5，设置 input 与 output 和端口名，并连线

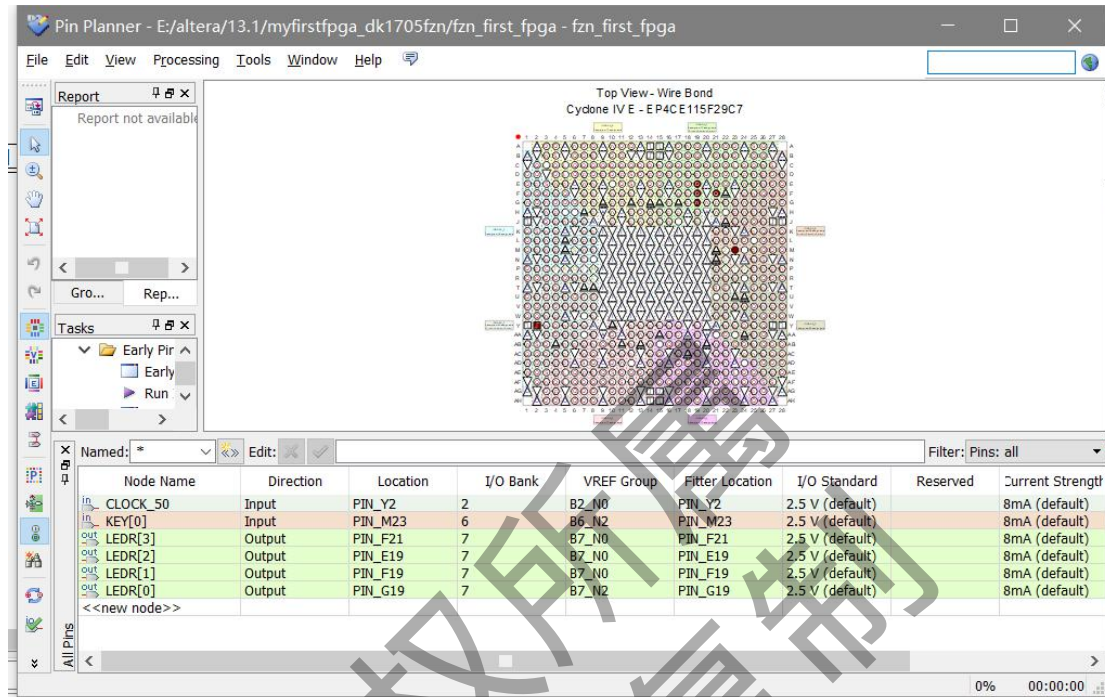


3) 仿真

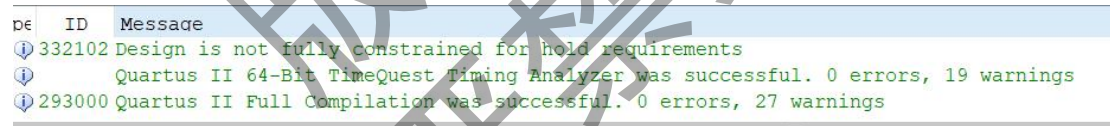
Modelsim 仿真文件库无法找到，故失败。

4) 编译与下载

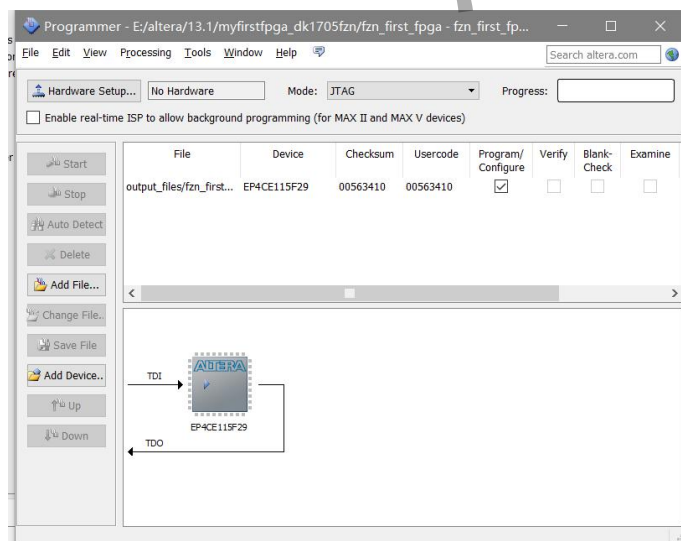
1. 将系统 KEY, LED,CLOCK 对应于板子引脚。



2.编译，0error, 27warning



3.下载



五、实验代码

```
module simple_counter (  
  
    CLOCK_50,  
    counter_out  
);  
  
input    CLOCK_50 ;  
output [31:0] counter_out;  
reg      [31:0] counter_out;  
  
always @ (posedge CLOCK_50) // on positive clock edge  
begin  
    counter_out <= #1 counter_out + 1; // increment counter  
end  
  
endmodule // end of module counter
```

六、实验体会





第一次体验到了 FPGA 编程的强大，由 C 语言要描述半天的程序可以瞬间解决。