

**实验题四：在 FPGA 芯片内用硬件描述语言构建
一个信号发生器，分别产生周期为 2 秒、1
秒、0.5 秒和 0.25 秒的方波**

姓名：王科

学号：1310583

专业：计算机科学与技术

完成日期：2016/3/27

目录

一、	实验内容	- 2 -
二、	实验原理	- 2 -
三、	实验步骤	- 2 -
	1. 设计译码器的端口、内部逻辑程序	- 2 -
	2. 观察 FPGA 电路板，将输入输出端口进行引脚绑定	- 4 -
	3. 电路连接示意图	- 5 -
	4. 实验结果	- 5 -
四、	实验程序（.VHD 文件）	- 8 -
五、	实验总结	- 10 -

一、 实验内容

在 FPGA 芯片内用硬件描述语言构建一个信号发生器，分别产生周期为 2 秒、1 秒、0.5 秒和 0.25 秒的方波，使用实验箱或 FPGA 板上的任意时钟输出作为输入，将这些输出信号接 LED 显示模块，以便观察其周期。

二、 实验原理

实现一个能产生固定周期的方波，主要依靠使用一个固定频率的时钟信号来进行计数。通过计算产生多少次脉冲过后刚好能够实现要求输出的方波频率，然后使用一个变量，每隔这么多脉冲计数过后改变电平，实现输出特定的方波。这就是分频的基本原理。

实验时实验箱上有 1.843MHz, 184.32KHz, 18.432KHz 三种频率的时钟信号，我使用了 184.32KHz 的时钟信号，即每个时钟周期是 5.425×10^{-6} 秒。若要产生周期是 2 秒、1 秒、0.5 秒和 0.25 秒的方波，则需要时钟周期计数分别是 368640、184320、92160、46080。使用四个信号变量分别计数，每当 184320、92160、46080、23040 时（每半个周期内电平跳变一次，所以数值是整个周期的一半），改变端口的输出电平。这样就可以实现周期为 2 秒、1 秒、0.5 秒和 0.25 秒的方波。

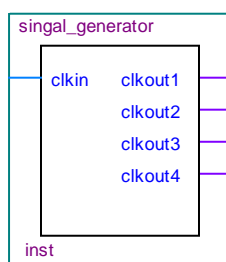
设计信号产生器的端口有 5 个，一个输入端口是频率是 184.32KHz 的时钟信号，另外四个输出端口是输出 2 秒、1 秒、0.5 秒和 0.25 秒的方波信号。

按照以上分析，设计实验的步骤如下：

三、 实验步骤

1. 设计译码器的端口、内部逻辑程序

编码器有五个端口，一个输入端口是频率是 184.32KHz 的时钟信号，另外四个输出端口是输出 2 秒、1 秒、0.5 秒和 0.25 秒的方波信号。其外部模块图如下：



vhdl 语言设计端口如下：

```
ENTITY singal_generator IS
  PORT(
    clk: IN STD_LOGIC;      -- 184.32KHz
    clkout1: OUT STD_LOGIC;  --0.25s
    clkout2: OUT STD_LOGIC;  --0.5s
    clkout3: OUT STD_LOGIC;  --1s
    clkout4: OUT STD_LOGIC   --2s
  );
END singal_generator;
```

其内部逻辑输入输出关系如下：

INPUT	OUTPUT			
clk	clkout1	clkout2	clkout3	clkout4
184.32KHz	4Hz	2Hz	1Hz	0.5Hz

由此设计 VHDL 结构体程序如下：

```
ARCHITECTURE content OF singal_generator IS

  SIGNAL counter1: integer range 0 to 23040;
  SIGNAL counter2: integer range 0 to 46080;
  SIGNAL counter3: integer range 0 to 92160;
  SIGNAL counter4: integer range 0 to 184320;
  SIGNAL Clk1:Std_Logic;
  SIGNAL Clk2:Std_Logic;
  SIGNAL Clk3:Std_Logic;
  SIGNAL Clk4:Std_Logic;

BEGIN

  PROCESS(clk)
    BEGIN
```

```

IF( counter1=23040) THEN
    counter1 <= 0;
    Clk1 <= NOT Clk1;
ELSIF rising_edge(clkin) THEN
    counter1 <= counter1+1;
END IF;

IF( counter2=46080) THEN
    counter2 <= 0;
    Clk2 <= NOT Clk2;
ELSIF rising_edge(clkin) THEN
    counter2 <= counter2+1;
END IF;

IF( counter3=92160) THEN
    counter3 <= 0;
    Clk3 <= NOT Clk3;
ELSIF rising_edge(clkin) THEN
    counter3 <= counter3+1;
END IF;

IF( counter4=184320) THEN
    counter4 <= 0;
    Clk4 <= NOT Clk4;
ELSIF rising_edge(clkin) THEN
    counter4 <= counter4+1;
END IF;

END PROCESS;

clkout1 <= Clk1;
clkout2 <= Clk2;
clkout3 <= Clk3;
clkout4 <= Clk4;

END content;

```

2. 观察 FPGA 电路板，将输入输出端口进行引脚绑定

根据 FPGA 电路板上空闲的剩余引脚，在这次实验中个输入输出端对应的引脚号如下表：

	Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard	Reserved	Group	Current Strength
1	clk	Input	PIN_91	3	83_N0	3.3-V LVTTTL (default)			24mA (default)
2	clkout1	Output	PIN_97	3	83_N0	3.3-V LVTTTL (default)			24mA (default)
3	clkout2	Output	PIN_99	3	83_N0	3.3-V LVTTTL (default)			24mA (default)
4	clkout3	Output	PIN_100	3	83_N0	3.3-V LVTTTL (default)			24mA (default)
5	clkout4	Output	PIN_101	3	83_N0	3.3-V LVTTTL (default)			24mA (default)
6	<<new node>>								

编译下载到 FPGA 电路板上，然后连接实验箱的电路。

3. 电路连接示意图

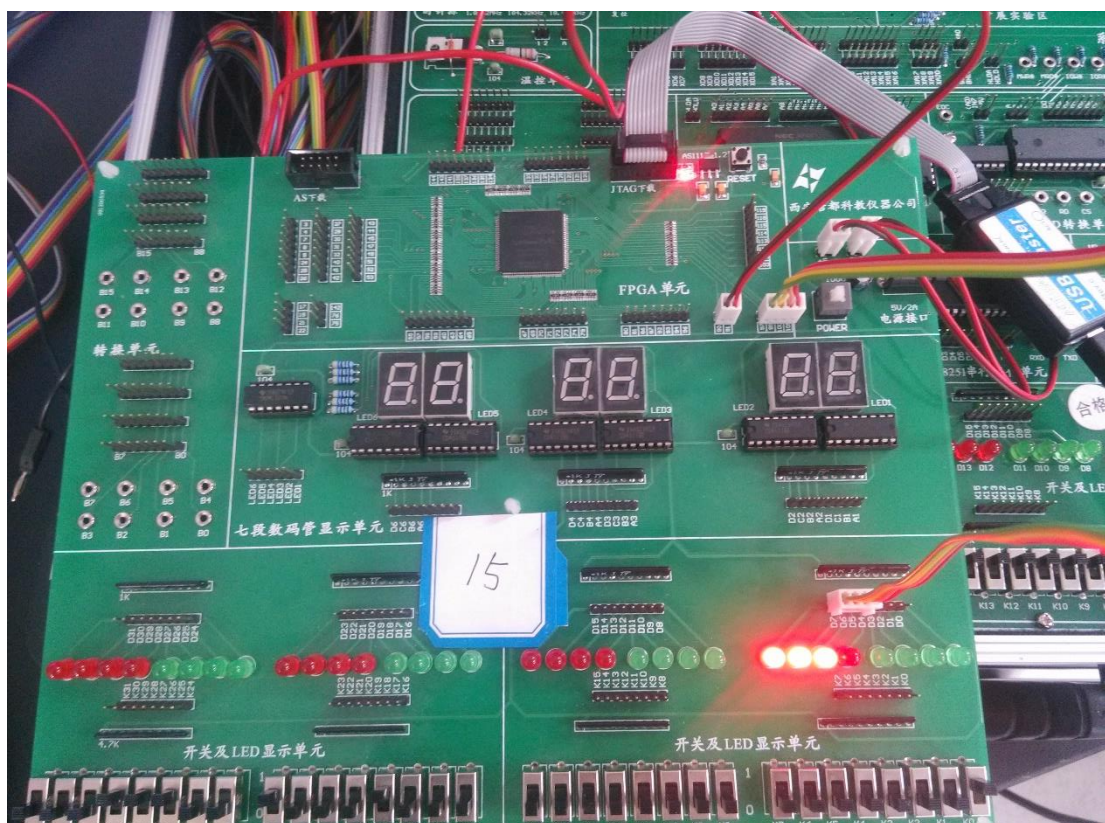
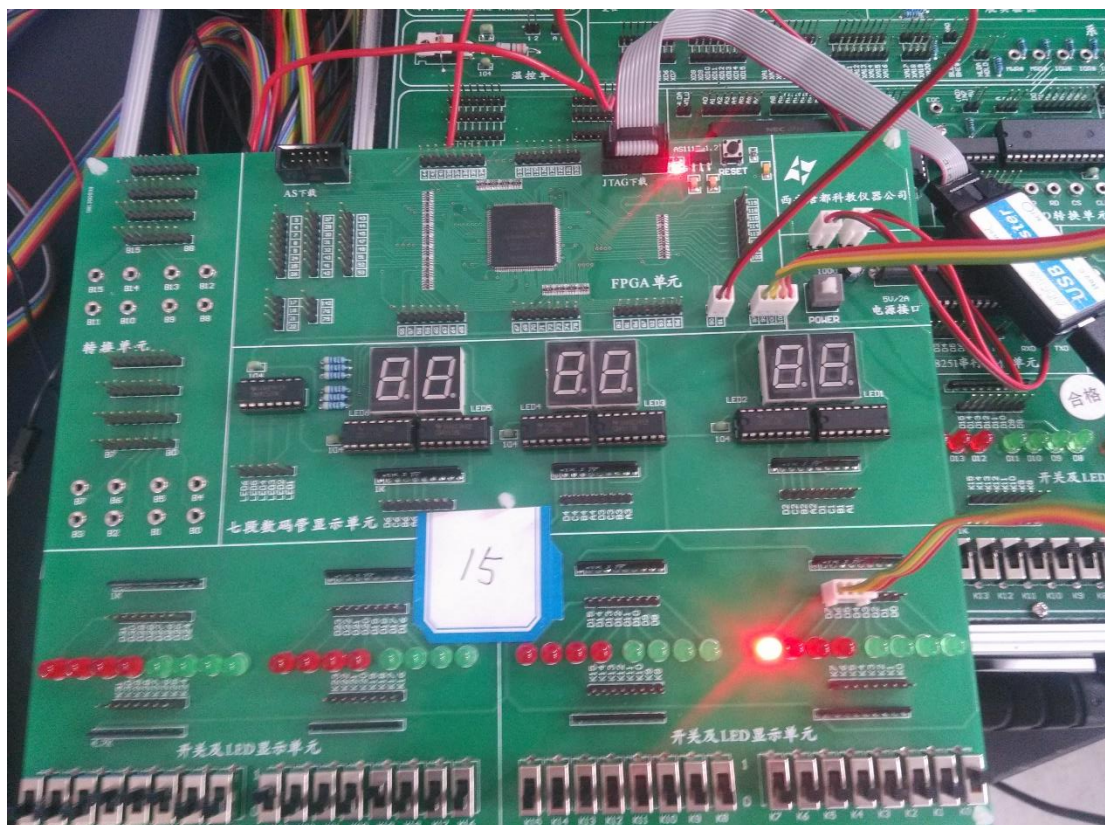
连接 FPGA 电路板上电路：

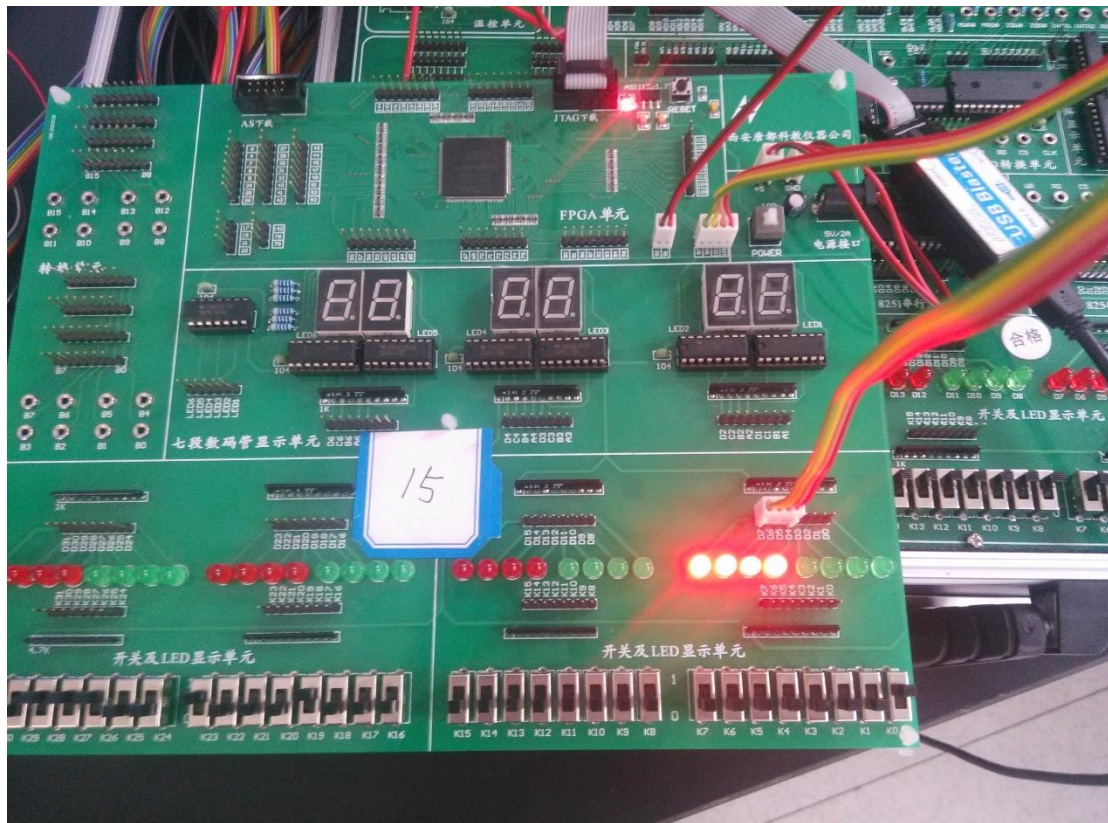
- 将 FPGA 板上的 PIN_91(clkin)接到实验箱上的 184.32KHz 时钟信号上；
- 将 FPGA 板上的 PIN_97~PIN_101(clkout1~clkout4)接到 FPGA 电路板的 LED 灯泡上。

对应端口连接表			
说明	引脚	说明	引脚
FPGA 板的输入端 (clkin)	PIN_91	实验箱上的时钟 信号	184.32KHz
FPGA 板的输出端口 (clkout1~clkout4)	PIN_91~PIN_101	FPGA 电路板的 LED 灯泡	L0~L3

4. 实验结果

实验时，输出 2 秒、1 秒、0.5 秒和 0.25 秒的方波信号如下：





实验时给四个小灯泡的周期分别是 0.25 秒、0.5 秒、1 秒和 2 秒，即能产生 0.25 秒、0.5 秒、1 秒和 2 秒的时钟周期，验证了程序的正确性。

四、 实验程序（.VHD 文件）

整个实验所用的程序如下（附件“singal_generator.vhd”）:

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_unsigned.all;
Use ieee.std_logic_arith.all;

ENTITY singal_generator IS
PORT(
    clkin: IN STD_LOGIC;      -- 184.32KHz
    clkout1: OUT STD_LOGIC;    --0.25s
    clkout2: OUT STD_LOGIC;    --0.5s
    clkout3: OUT STD_LOGIC;    --1s
    clkout4: OUT STD_LOGIC    --2s
);
END singal_generator;

ARCHITECTURE content OF singal_generator IS

    SIGNAL counter1: integer range 0 to 23040;
    SIGNAL counter2: integer range 0 to 46080;
    SIGNAL counter3: integer range 0 to 92160;
    SIGNAL counter4: integer range 0 to 184320;
    SIGNAL Clk1:Std_Logic;
    SIGNAL Clk2:Std_Logic;
    SIGNAL Clk3:Std_Logic;
    SIGNAL Clk4:Std_Logic;

BEGIN

    PROCESS(clkin)
    BEGIN

        IF( counter1=23040) THEN
            counter1 <= 0;
            Clk1 <= NOT Clk1;
        ELSIF rising_edge(clkin) THEN
            counter1 <= counter1+1;
        END IF;

        IF( counter2=46080) THEN
```

```

        counter2 <= 0;
        Clk2 <= NOT Clk2;
    ELSIF rising_edge(clkin) THEN
        counter2 <= counter2+1;
    END IF;

    IF( counter3=92160) THEN
        counter3 <= 0;
        Clk3 <= NOT Clk3;
    ELSIF rising_edge(clkin) THEN
        counter3 <= counter3+1;
    END IF;

    IF( counter4=184320) THEN
        counter4 <= 0;
        Clk4 <= NOT Clk4;
    ELSIF rising_edge(clkin) THEN
        counter4 <= counter4+1;
    END IF;

END PROCESS;

clkout1 <= Clk1;
clkout2 <= Clk2;
clkout3 <= Clk3;
clkout4 <= Clk4;

END content;

```

五、 实验总结

这次的使用 VHDL 硬件描述语言设计信号发生器，是第二类实验里的最后一个实验，相对于之前的三个实验来说逻辑较为复杂，而参考示例分频器文件之后，我也顺利的写出了这次的程序。在这次的实验遇到的问题主要是如何将四个计数器和输出信号联系起来。一开始想到的是只是用一个进程，使用时钟信号作为敏感信号，当计数到特定的数值时立即改变输出信号的电平，不过发现在实际中没有成功，可能是因为我在这个进程里一次判断了四次时钟信号上升沿来为四个计数器计数，导致时间上有延迟。后来引入了四个中间变量，使用了多个进程，而以前的那个进程只需要给四个计数器计数即可，当达到特定的数值时归零重新计数。另外引入其他进程判断中间变量是否变化，若有变化及时更新输出电平。

通过这次的实验，我感更加体会到了做实验需要细心，对于 VHDL 硬件描述语言有了更深的认识，对于其各种描述语句的逻辑有了更深的理解，也对于 VHDL 语言的使用熟练了一些。

最后感谢老师们不厌其烦的为我解答疑惑并帮助我调试 bug，我才能顺利的完成了这次实验！