

2016 年《综合课程设计 2-1》课程实验报告（第二类实验）

**实验题三：在 FPGA 芯片内用硬件描述语言
描述一个具有清除端（CLRn）的 8 位通用寄存器**

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	- 3 -
	3. 电路连接示意图	- 4 -
	4. 实验结果	- 4 -
四、	实验程序（.VHD 文件）	- 7 -
五、	实验总结	- 8 -

一、 实验内容

在 FPGA 芯片内用硬件描述语言描述一个具有清除端（CLR_N）的 8 位通用寄存器，该寄存器在 CLR_N 为低电平时清 0（输出变全 0），在 CLR_N 为高电平的情况下，接受脉冲输入端（CLK）上升沿来时进行数据写入，并立即出现在输出端。并利用用户实验板上的资源测试该逻辑（清 0 和数据写入）。

二、 实验原理

使用 VHDL 语言描述 8 位通用寄存器的功能，首先明确需要四个端口，有两个 8 位的输入和输出端口，还需要一个时钟信号 clock 来在时钟的上升沿来临时触发工作将输入端口的信号写入输出端口。按照题目要求还需要一个清零端 Clean，当其有效时（低电平）输出端口立即清零，并且此时不受时钟上升沿的限制。

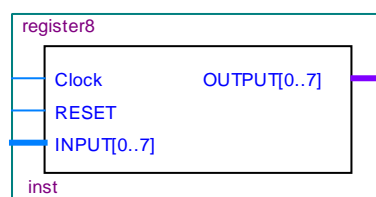
所以按照以上分析，初步设计有四个端口，一个 8 位输出端口，一个 8 位输入端口，一个时钟信号，一个清零端。在 VHDL 程序的内部逻辑上，需要设计一个 PROCESS 进程，此进程的敏感信号是时钟信号（Clock）和清零端（Clean），当清零端改变时，立即触发该进程检测清零端时是否是低电平，若是则将输出端信号立即清零。当时钟信号上升沿来临时，也会触发该进程在清零端无效的情况下将输入端信号写入输出端。

整个程序的逻辑比较简单，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

编码器有四个端口，分别是一个 8 位的输入端口，一个 8 位的输出端口，一个时钟信号 Clock，一个清零端 Clean。其外部模块图如下：



vhdl 语言设计端口如下：

```

ENTITY register8 IS
  PORT(
    Clock:IN STD_LOGIC;
    RESET:IN STD_LOGIC ;
    INPUT:IN STD_LOGIC_VECTOR(7 TO 0);
    OUTPUT:OUT STD_LOGIC_VECTOR(7 TO 0)
  );
END register8;

```

其内部逻辑输入输出关系如下：

INPUT			OUTPUT
Clean	Clock	Input[0..7]	Output[0..7]
0	X	[zzzzzzzz]	[00000000]
1	上升沿	[zzzzzzzz]	Input[0..7]
1	下降沿	[zzzzzzzz]	[xxxxxxxx]

由此设计 VHDL 结构体程序如下：

```

ARCHITECTURE content OF register8 IS

BEGIN
  PROCESS(RESET , Clock) --sensitive signal
  BEGIN

    IF  RESET = '0' THEN
      OUTPUT <= "00000000";
    ELSIF rising_edge (Clock) THEN
      OUTPUT <= INPUT;
    END IF;
  END PROCESS;

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
Clock	Input	PIN_91	3	B3_N0	3.3-V LVTTTL (default)			24mA (default)
INPUT[0]	Input	PIN_144	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
INPUT[1]	Input	PIN_143	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
INPUT[2]	Input	PIN_141	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
INPUT[3]	Input	PIN_139	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
INPUT[4]	Input	PIN_137	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
INPUT[5]	Input	PIN_136	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
INPUT[6]	Input	PIN_135	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
INPUT[7]	Input	PIN_134	2	B2_N1	3.3-V LVTTTL (default)		INPUT[0..7]	24mA (default)
OUTPUT[0]	Output	PIN_133	2	B2_N1	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
OUTPUT[1]	Output	PIN_132	2	B2_N1	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
OUTPUT[2]	Output	PIN_129	2	B2_N1	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
OUTPUT[3]	Output	PIN_126	2	B2_N0	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
OUTPUT[4]	Output	PIN_125	2	B2_N0	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
OUTPUT[5]	Output	PIN_122	2	B2_N0	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
OUTPUT[6]	Output	PIN_121	2	B2_N0	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
OUTPUT[7]	Output	PIN_120	2	B2_N0	3.3-V LVTTTL (default)		OUTPUT[0..7]	24mA (default)
RESET	Input	PIN_101	3	B3_N0	3.3-V LVTTTL (default)			24mA (default)

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：

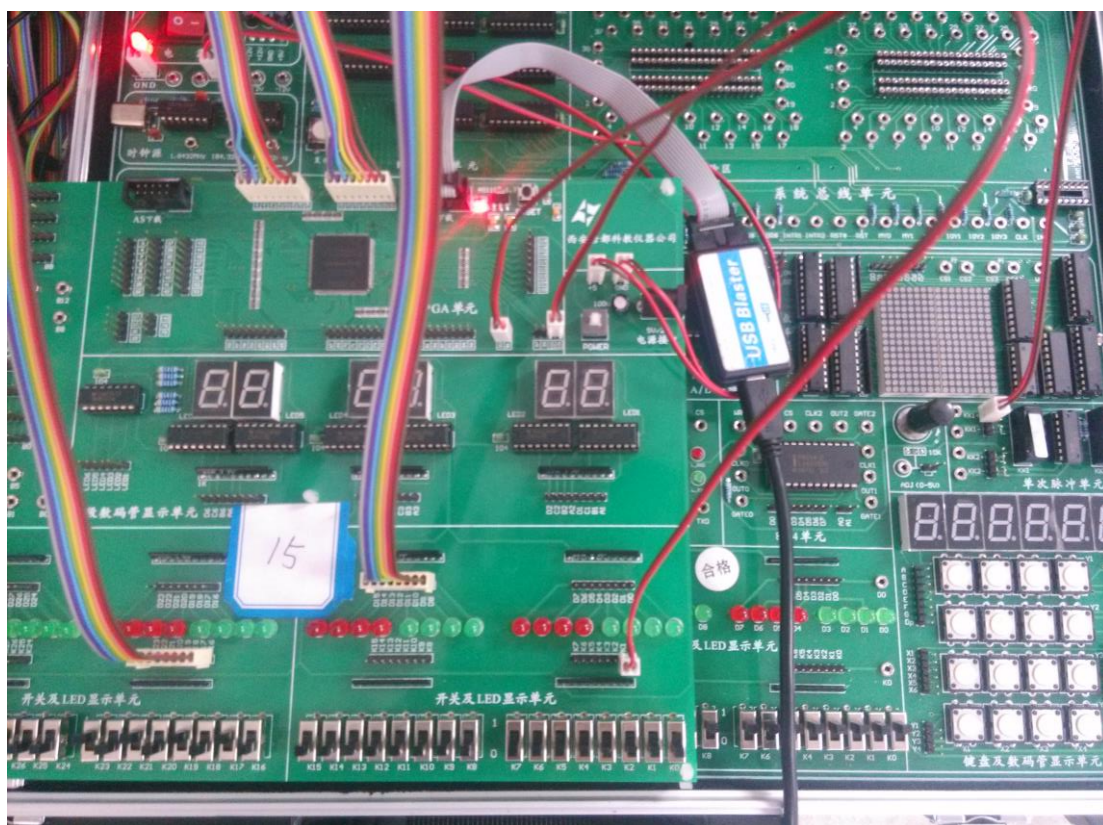
- 将 FPGA 板上的 PIN_144~PIN_134 (INPUT[0]~INPUT[7]) 接到 FPGA 电路板的控制开关上；
- 将 FPGA 板上的 PIN_133~PIN_120 (OUTPUT[0]~OUTPUT[7]) 接到 FPGA 电路板的 LED 灯泡上。
- 将 FPGA 板上的 PIN_91 (时钟信号 Clock) 接到实验箱上的脉冲开关上。
- 将 FPGA 板上的 PIN_101 (清零信号 RESET) 接到 FPGA 电路板的控制开关上。

对应端口连接表			
说明	引脚	说明	引脚
FPGA 板的输入端 (INPUT[0]~INPUT[7])	PIN_144~PIN_134	FPGA 电路板控制开关	K0~K7
FPGA 板的输出端口 (OUTPUT[0]~OUTPUT[7])	PIN_133~PIN_120	FPGA 电路板的 LED 灯泡	L0~L7
FPGA 板上的时钟信号 Clock	PIN_91	实验箱上的脉冲开关	KK0
FPGA 板上的清零信号 RESET	PIN_101	FPGA 电路板的控制开关	K8

4. 实验结果

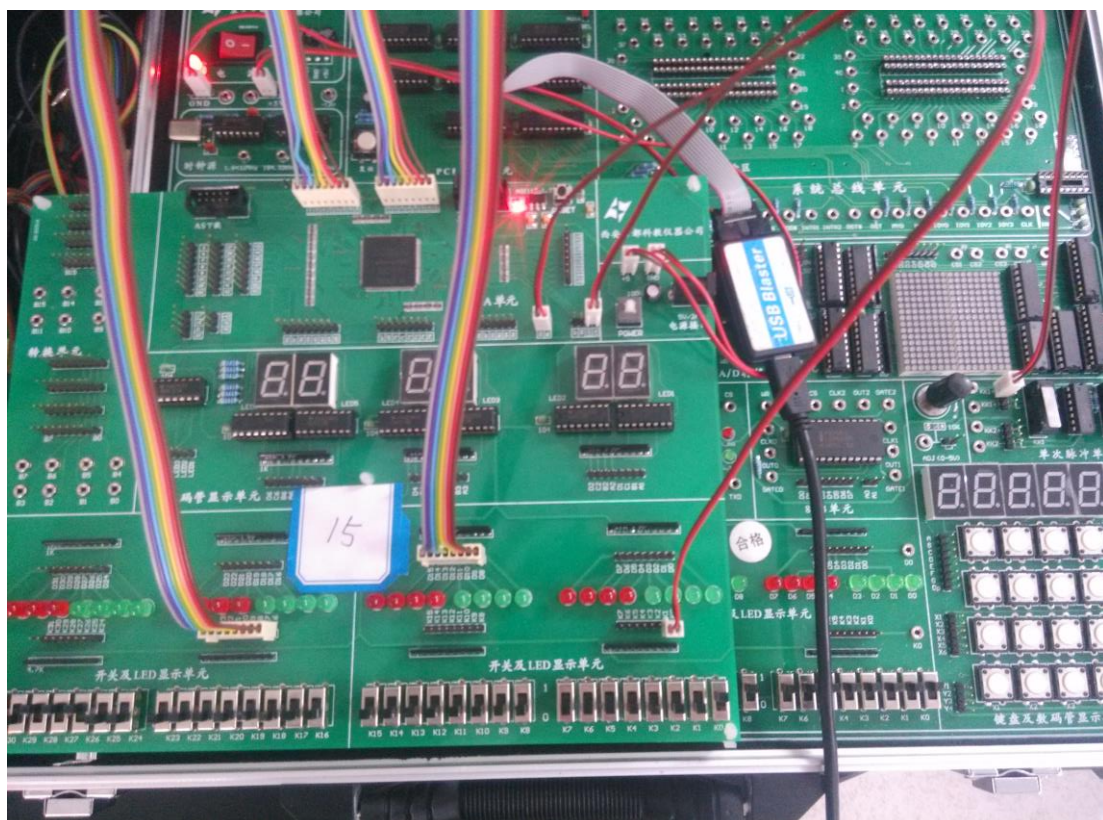
实验时，将清零端 RESET 置为低电平时，无论输入端的信号是什么，也不用等到

时钟信号上升沿来临，输出端立即清零。其结果图如下：



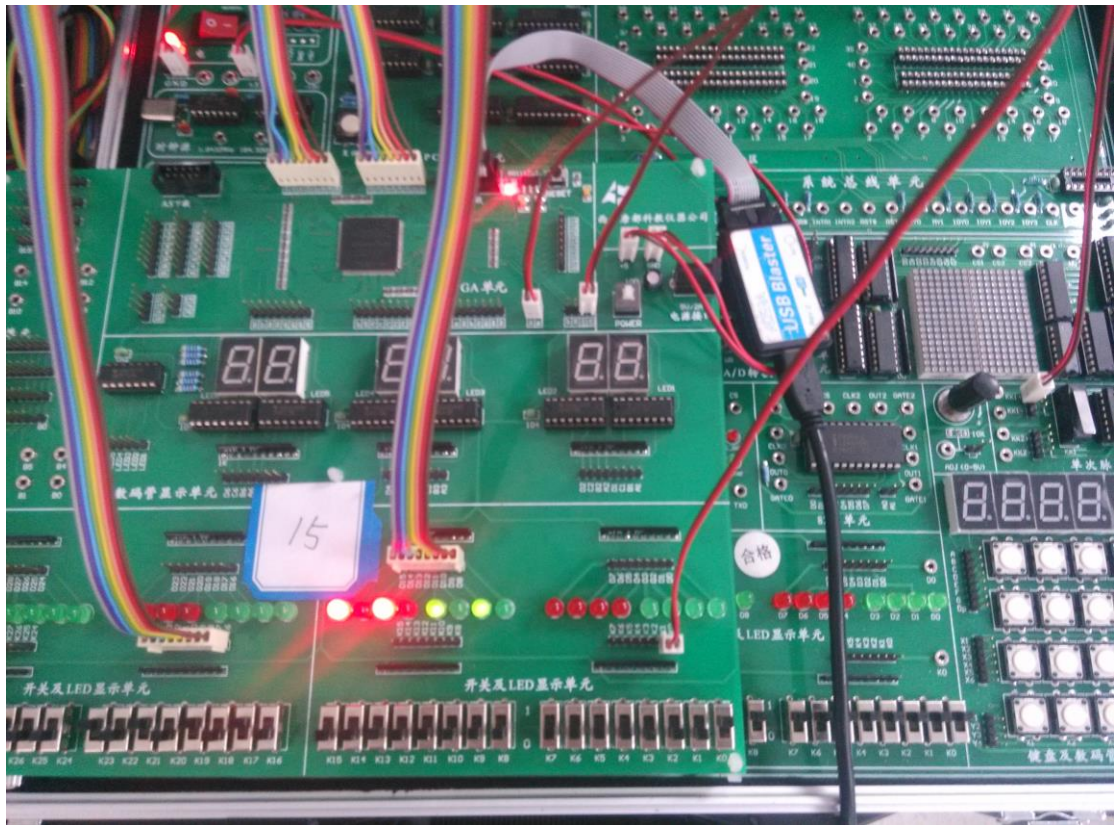
实验输出结果是 00000000，验证了清零端 RESET 的正确工作。

当把清零端 RESET 置为无效时，输入信号是 00000000 时，实验结果如下：



实验时给一个时钟信号上升沿，输出端显示 00000000，这与输入信号一致。

当把清零端 RESET 置为无效时，输入信号是 10101010 时，实验结果如下：



实验时给一个时钟信号上升沿，输出端显示 10101010，这与输入信号一致，验证了 8 位寄存器的正确工作。

四、 实验程序（.VHD 文件）

整个实验所用的程序如下（附件“register8.vhd”）:

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY register8 IS
    PORT(
        Clock:IN STD_LOGIC;
        RESET:IN STD_LOGIC ;
        INPUT:IN STD_LOGIC_VECTOR(7 TO 0);
        OUTPUT:OUT STD_LOGIC_VECTOR(7 TO 0)
    );
END register8;

ARCHITECTURE content OF register8 IS

BEGIN
    PROCESS(RESET , Clock) --sensitive signal
    BEGIN

        IF  RESET = '0' THEN
            OUTPUT <= "00000000";
        ELSIF rising_edge (Clock) THEN
            OUTPUT <= INPUT;
        END IF;
    END PROCESS;

END content;
```


五、 实验总结

这次的使用 VHDL 硬件描述语言设计一个 8 位寄存器实验宁不是很复杂。使用的逻辑也很简单，主要就是根据时钟信号和清零端来触发输出信号的变化。

即使使用的逻辑很简单，我由于没有仔细的考虑题意在这次的实验过程中还是遇到了一个小小的问题，我在一开始在进程的敏感信号里只考虑了时钟信号，而判断清零端是否有效的程序在时钟信号触发进程之内，这就导致了当清零端为低电平时而时钟信号的上升沿还未来临时，输出端是维持不变的而不是按照题目要求的那样立即清零。我后来在进程的敏感信号加入了清零信号 RESET 就解决了这个问题。

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！