

# 中山大学数据科学与计算机学院本科生实验报告

## (2017 学年春季学期)

课程名称：计算机组成原理实验

任课教师：郭雪梅

助教：李声涛、王绍菊

年级&班级	15 级 18 班	专业(方向)	软件工程（移动信息工程）
学号	15352408	姓名	张稼伟
电话	13531810182	Email	<a href="mailto:709075442@qq.com">709075442@qq.com</a>
开始日期	2017.5.12	完成日期	2017.5.12

### 一、实验题目

设计一个寄存器堆

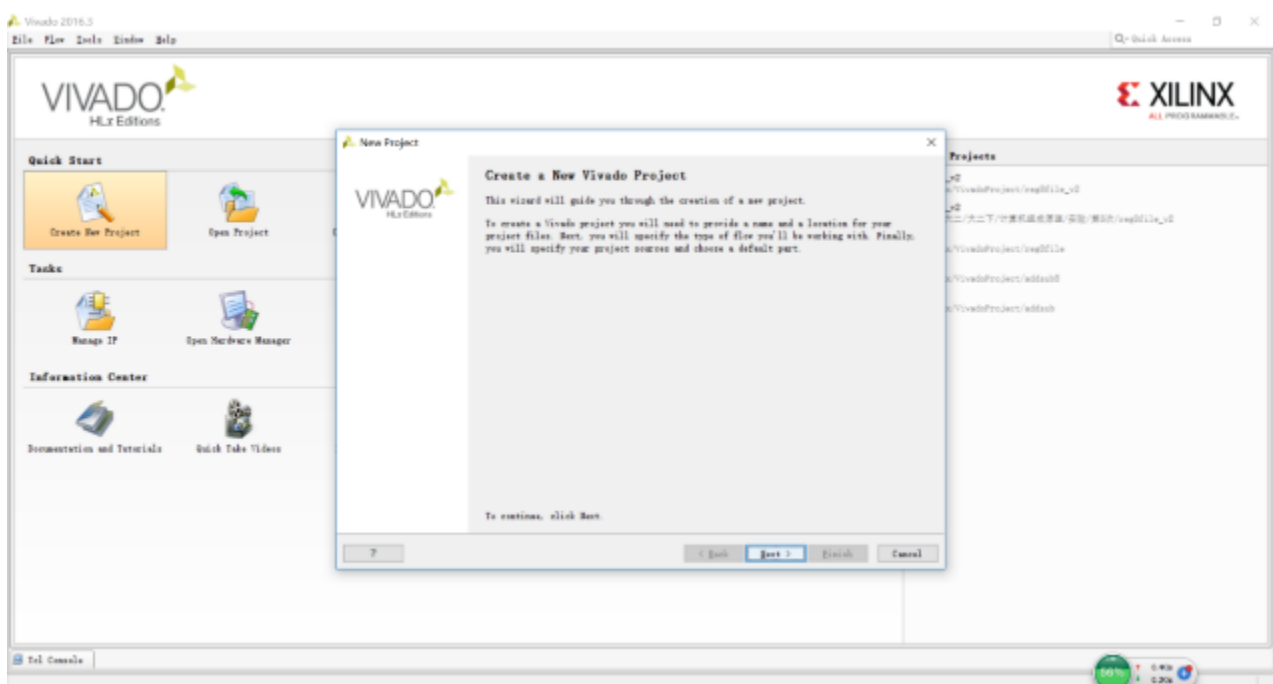
### 二、实验目的

设计一个包含 8 个 8 位寄存器的寄存器堆，每个寄存器可存储 8 位数据，设计一个带异步清零和 wen 使能端的 D 触发器，通过该触发器设计一个寄存器，再通过设计的寄存器设计一个寄存器堆。

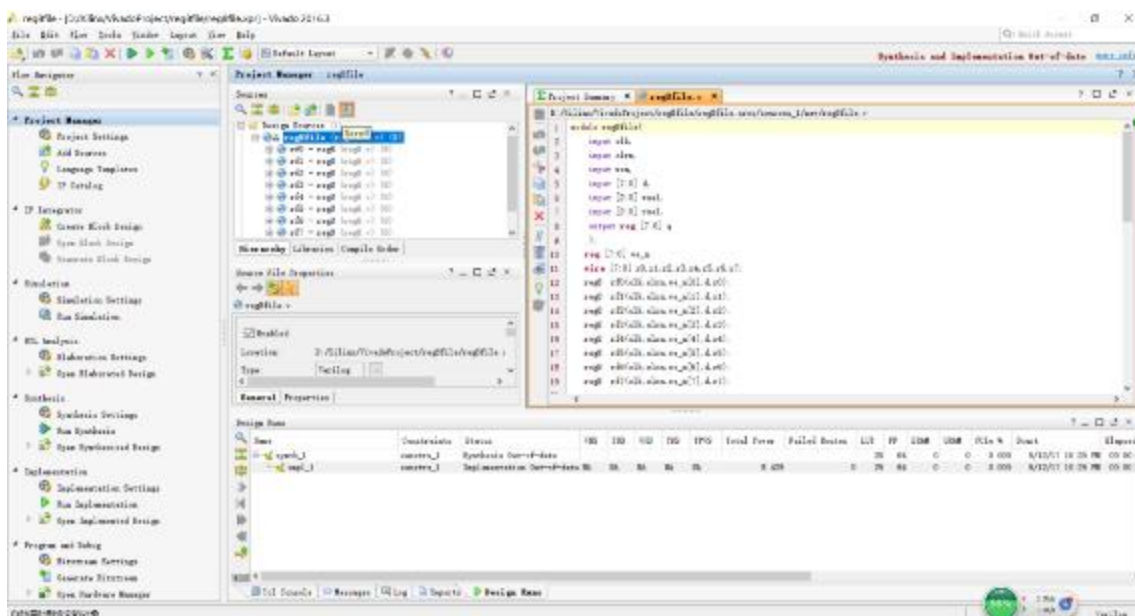
### 三、实验内容

#### 1. 实验步骤

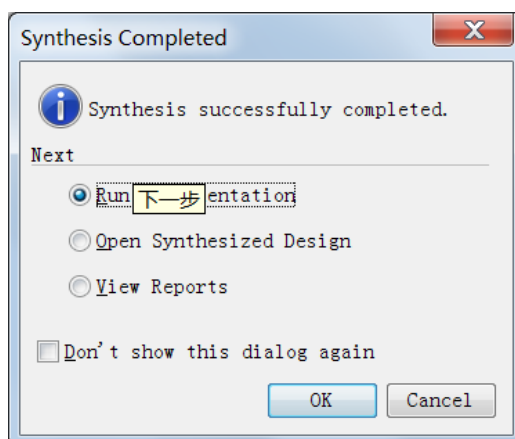
(1) 打开 Vivado，按照上一次实验的教程创建好 RTL 项目，同时创建 reg8file.v、dfffe.v、reg8.v 三个源文件。



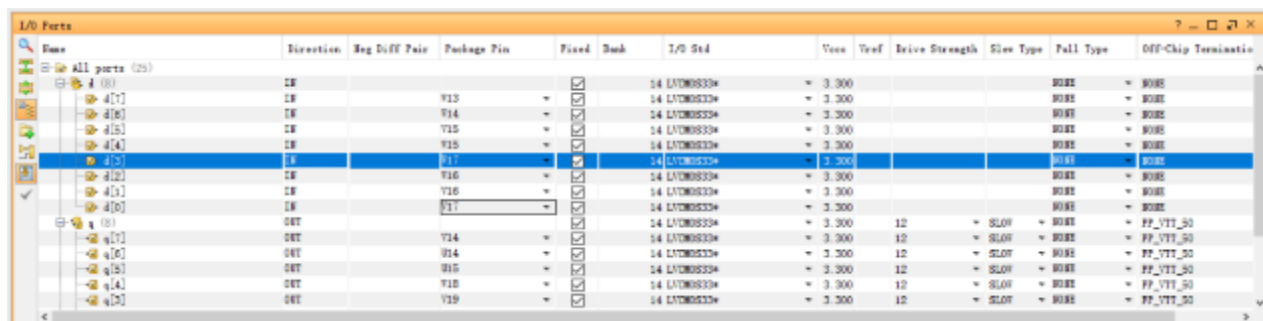
(2) 在三个文件中输入实验要求中给出的代码。保存全部内容。



(3) 点击 Run Synthesis,完成综合后，点击 ok。



(4) 在 I/O Ports 进行引脚分配。时钟 clock 连接引脚 W5，d0-d7 连接开关 sw0 到 sw7。rsl0-rsl2 连接开关 sw13 到 sw15，wsl0-wsl2 连接开关 sw10 到 sw12，q0-q7 连接 led0 到 led7，clrn 接开关 sw9，wen 接开关 sw8。I/O Std 这一列选择 LVCMOS33。保存文件。



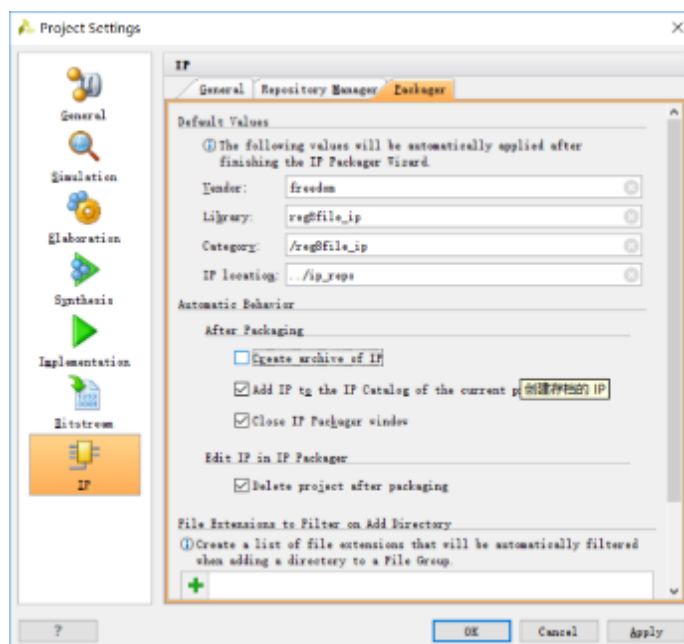
(5) 点击 Generate Bitstream 生成比特流文件。

(6) 点击 Hardware Manager 下的 Open Target,添加板子。

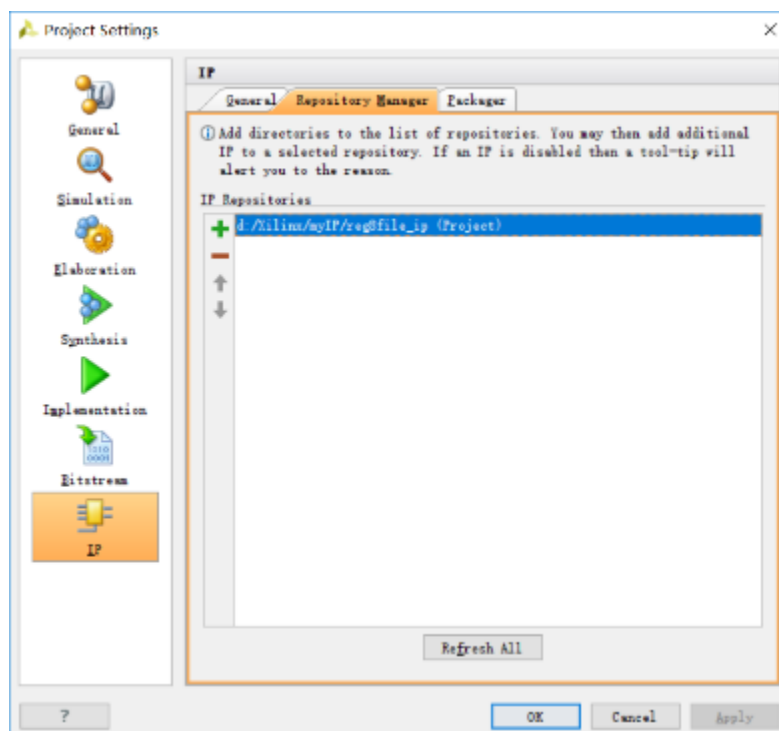
- (7) 点击 Hardware Manager 下的 Program Device, 选择板子, 烧录 bit 文件。
- (8) 测试板子, 没有问题。

### 下面开始封装 IP 和进行仿真

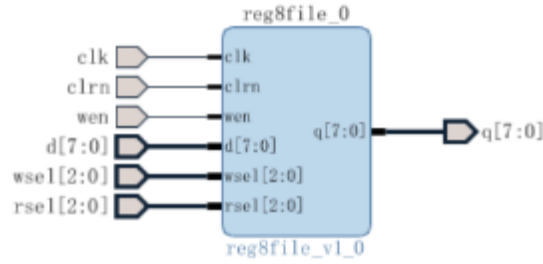
- (9) 点击 Project Setting, 选 IP 下的 Packager, 编辑封装信息。



- (10) 点击 package ip, 封装 ip。
- (11) 新建一个项目来 reg8file\_v2 使用这个 ip
- (12) 在 Project Settings 里选 IP 下的 Repository Manager, 添加自己编写 IP 的位置



- (13) 点击 Create Block Design, 添加自己封装的 ip, 连接端口。保存文件。



（14）在 Add Sources 里新建仿真文件 reg8file\_sim。输入仿真代码。

仿真代码如下：

```
`timescale 1ns / 1ps
```

```
module reg8file_sim(
```

```
    );
    reg clk;
    reg clrn;
    reg wen;
    reg[7:0] d;
    wire[7:0] q;
    reg[2:0] wsel;
    reg[2:0] rsel;
    reg8file_ip reg_sim(
        .clk(clk),
        .clrn(clrn),
        .wen(wen),
        .d(d),
        .wsel(wsel),
        .rsel(rsel),
        .q(q)
    );
```

```
    initial begin
```

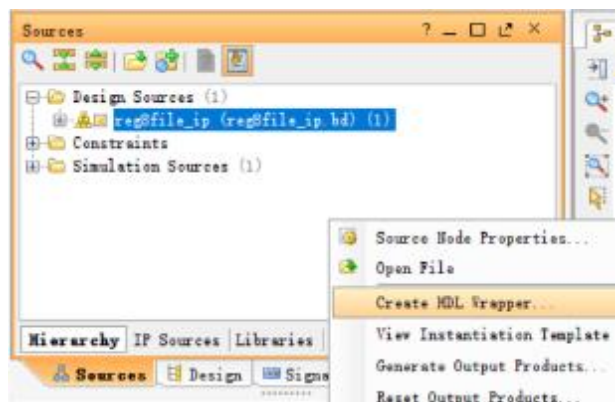
```
#0 begin clrn=0;clk=0;wen=0;wsel=3'b001;rsel=3'b001;d=8'b00000001;end //清零端开着，不会亮
#5 begin clrn=1;end //清零端关闭 也没变化，因为 clk 不在上升沿
#5 begin clk=1;end //上升沿
#10 begin clk=0;wsel=3'b010;rsel=3'b010;d=8'b00000010;end //二号寄存器存 2
#10 begin clk=1;end
#10 begin clk=0;d=8'b00000011;end //二号寄存器存 3
#10 begin clk=1;end
#10 begin clk=0;wen=1;d=8'b0000100;end //关闭使能，二号寄存器值没变
#10 begin clk=1;end
#10 begin clk=0;wsel=3'b011;rsel=3'b011;d=8'b0000110;end //使能未开，三号寄存器存 2 失败
#10 begin clk=1;end
#10 begin clk=0;wen=0;end //使能打开，二号寄存器成功存 3
#10 begin clk=1;end
```

```

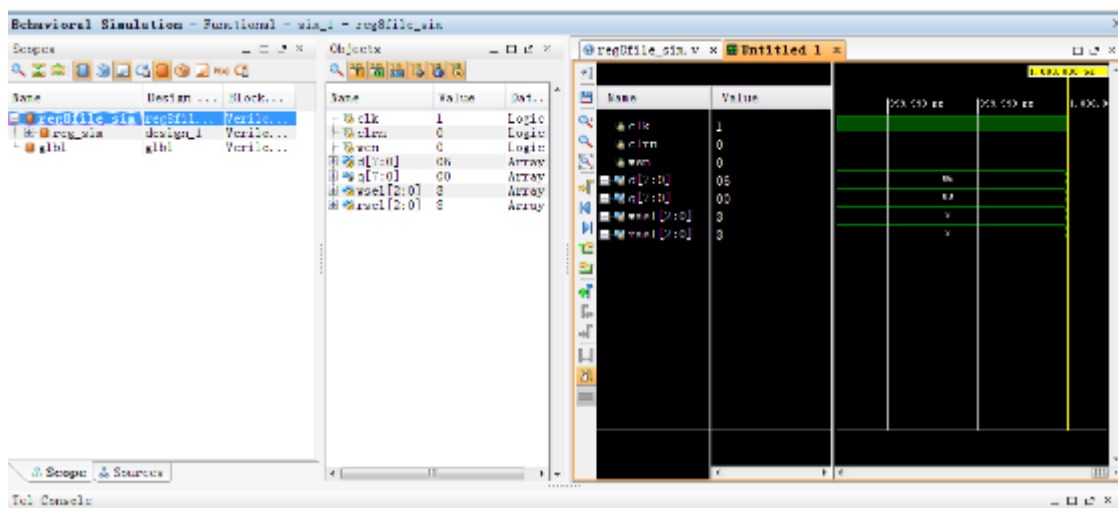
#10 begin clk=0;clrn=0;end//清零
#10 begin clk=1;end
end
endmodule

```

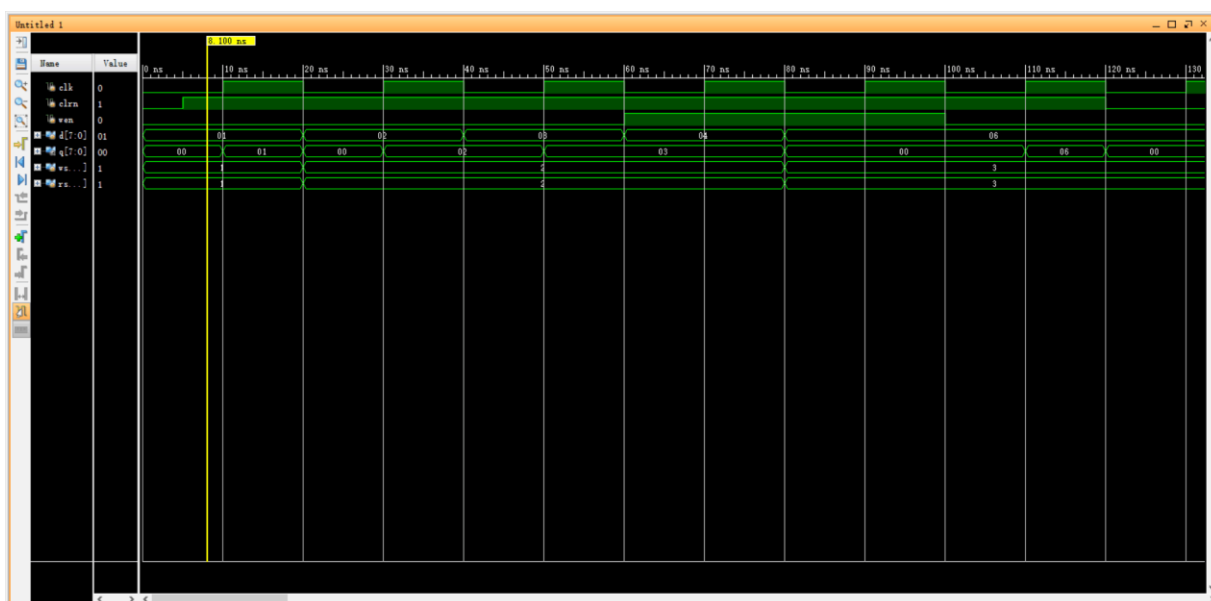
(15) 保存了仿真代码文件之后，在 source 栏下右键选择 creat HDL wrapper



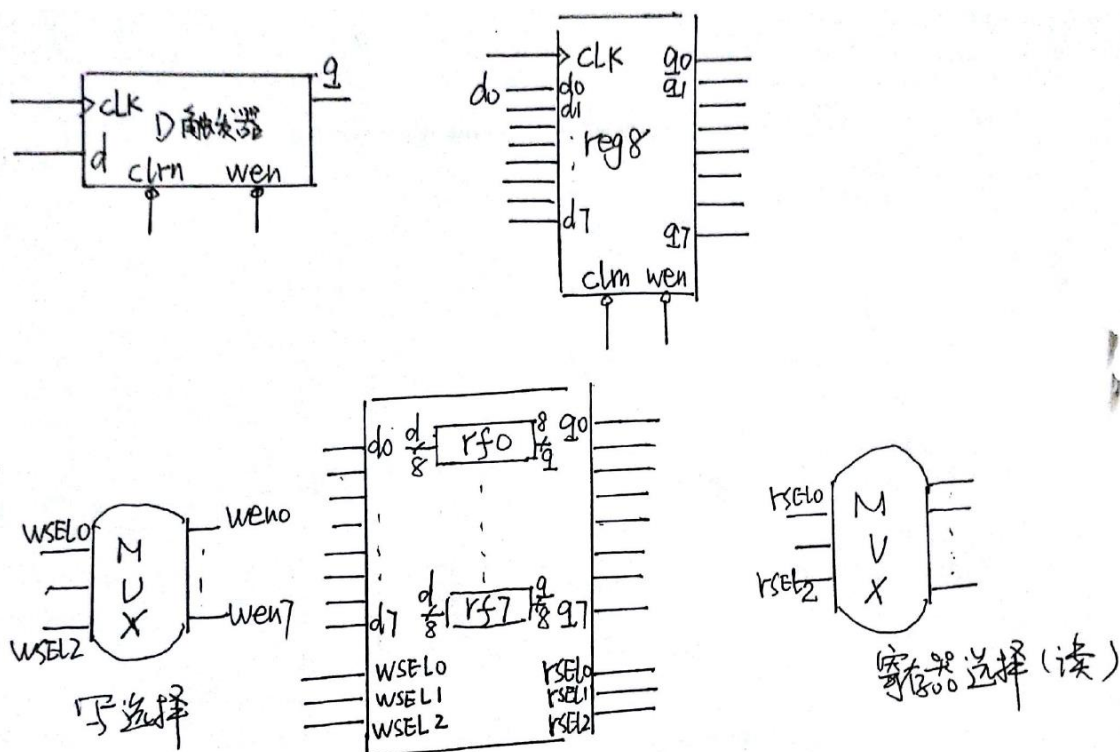
(16) 点击 Run Simulation 选择 Run Behavioral Simulation 开始仿真。



(17) 与仿真代码每句的注释相比较，发现波形符合预期。



## 2. 实验原理

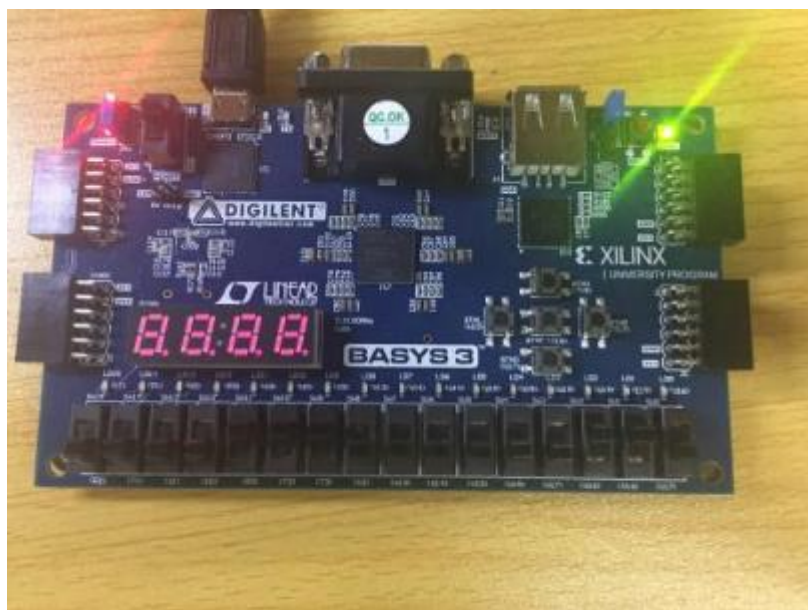


寄存器是有限存储容量的高速存储部件，它们可用来暂存指令、数据和地址。而寄存器堆是多个寄存器组成的阵列。而在verilog中的体现就是，寄存器的功能放在一个代码文件中实现，作为一个底层文件。而寄存器堆的代码文件作为一个顶层文件，在这个文件里面调用多个寄存器以构成寄存器堆的功能。

## 四、实验结果

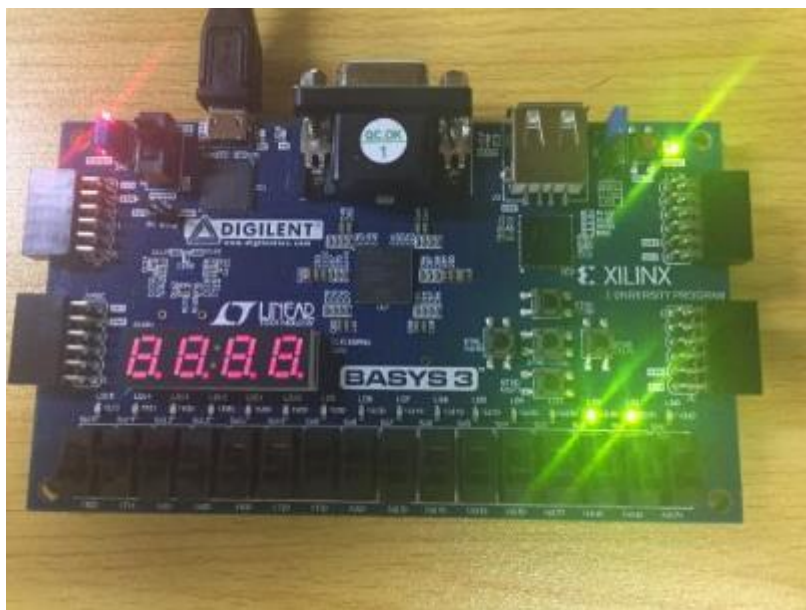
下面展现的是在3号寄存器存入数字6

将clrn(SW9)拨上去，在寄存器011(将开关SW11、SW10拨上去)，存入6(将开关SW2、SW1拨上去)。





打开使能端，将开关SW8拨上去。把写入寄存器和存入数字的开关拨下来。读寄存器011，将开关SW14、SW13拨上去，表示6的LED灯亮了。



## 五、实验感想

这一次的实验主要是熟悉之前的 vivado 基本操作：建项目、创建和编写代码文件、综合实现、烧录进板子运行。初次之外，这次的仿真代码老师没有给出，需要我们自己编写，这让我对仿真代码的编写又更加熟悉了。在写仿真代码的时候，一定要自己先设计好什么时候波形怎么改变，必要的话可以在代码旁边加上注释，这样后面看波形图的时候可以和注释代码对比着看，就不会看晕自己。