
兰州大学信息科学与工程学院

计算机组成原理课程设计实验报告

一、实验目的

1. 熟悉并掌握 MIPS 计算机中寄存器堆的原理和设计方法
2. 初步了解 MIPS 指令结构和源操作数/目的操作数的概念
3. 熟悉并运用 Verilog 语言进行电路设计
4. 为后续 CPU 的实验打下基础

二、实验器材与设备

- 2.1 装有 Xilinx vivado 的计算机一台
- 2.2 LS-CPU-EXB-002 教学系统实验箱一套

三、实验分析与设计

3.1 实验原理

3.1.1 首先我们要了解 MIPS 指令的格式结构：

32位MIPS指令集的长度均为32位，保持所有指令长度相同，但不同类型指令有不同的格式，为R型MIPS字段命名如下：

op	rs	rt	rd	shamt	funct
6位	5位	5位	5位	5位	6位
指令操作码	源寄存器1	源寄存器2	目的寄存器	位移量（移位指令）	功能码

图 1 32 位 MIPS 指令介绍

其由六个部分组成：6 位操作码指令、两个操作数的寄存器 5 位地址、一个目的寄存器的五位地址、用于跳转的 5 位偏移量和六位功能码。

而 MIPS 的指令分为三类：I 型，R 型，J 型

1. I 型指令：立即数相关的指令

op	rs	rt	constant or address
6位	5位	5位	16位
指令操作码	源寄存器	目的寄存器	立即数或地址

图 2 I 型指令介绍

2. R 型指令：常用的三地址指令

op	rs	rt	rd	shamt	funct
6位	5位	5位	5位	5位	6位
指令操作码	源寄存器	寄存器2	目的寄存器	位移量（移位指令）	功能码

图 3 R 型指令介绍

3. J 型指令：跳转相关指令

指令：j 10000

op	address
6位	26位
2	10000

指令：bne \$s0,\$s1,Exit

op	rs	rt	address
6位	5位	5位	16位
5	16	17	Exit

图 4 J 型指令介绍

由上可知在设计寄存器时，我们至少应该设计两个读端口用于根据给定的 5 位源寄存器地址读取两个操作数；当然还要加上一个写端口用于运算结果的写回。

3.2 端口设计

RADD1：第一个读地址

RADD2：第二个读地址

RDAT1：第一个读地址对应寄存器的值

RDAT2：第二个读地址对应寄存器的值

WADDR：写地址

WDATA：写地址对应寄存器需要存入的值

WEN：控制写使能

通过三个开关来控制输入：

SW18：控制 WEN 写使能

SW19-20：00 为第一个读地址；01 为第二个读地址；10 为写地址；11 为写数据

3.3 设计框图

项目整体设计：

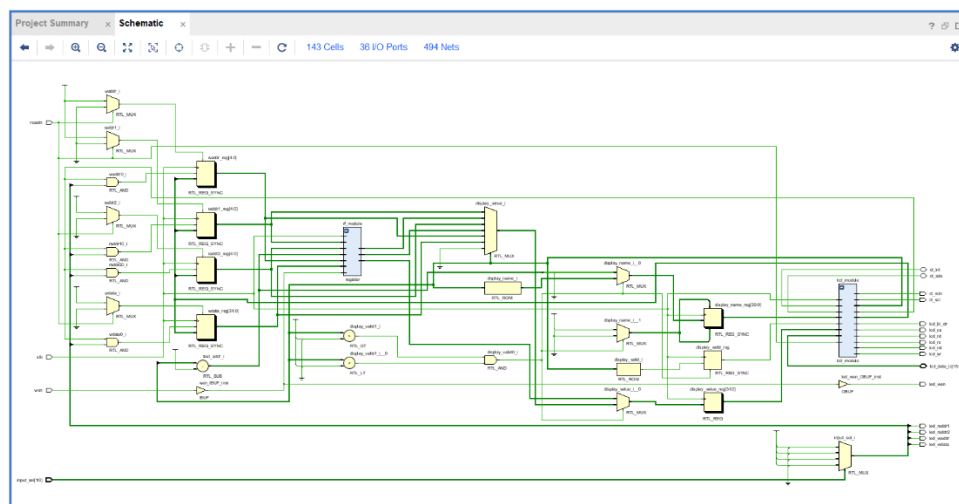


图 6 整个运行模的设计框图

寄存器设计:

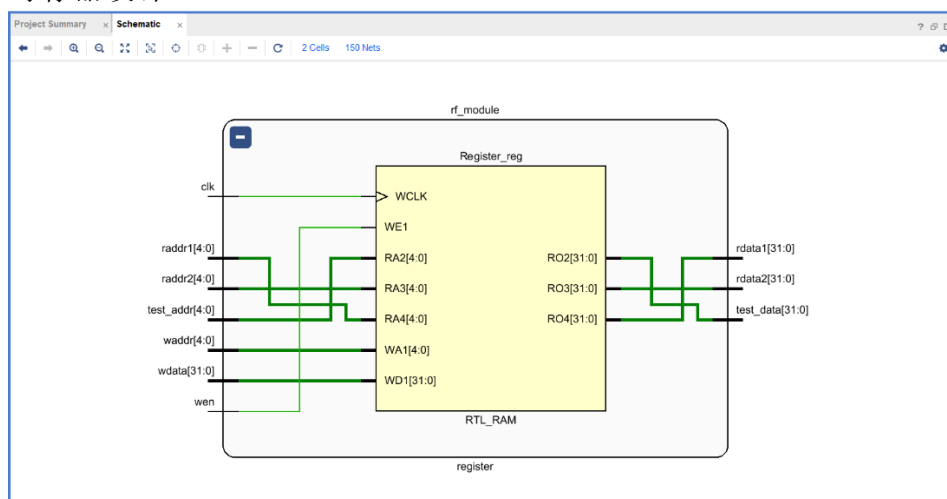


图 7 寄存器模块的设计框图

四、实验步骤

4.1 仿真与综合

具体的代码设计在问题回答中会有详细的讲解，这里不再赘述，直接进行仿真和综合，设计的测试文件内容如下：

```

#100;
waddr = 5'h07;
wdata = 32'h3C;
#100;
wen = 1'b1;
#100;
wen = 1'b0;
#100;
raddr1 = 5'h07;
#100;
raddr2 = 5'h07;
#100;
test_addr = 5'h07;
# 100
waddr = 5'h15;
wdata = 32'h22;
#100
wen = 1'b1;
#100
raddr2 = 5'h15;
# 100
wen = 1'b0;

```

图 8 测试文件的主要内容

具体步骤：(使用十六进制表示)

1. 将数据 3C 写入 7 号寄存器
2. 使能信号置为 1，将数据写入，使能信号置为 0
3. 让三个读地址(为了展示方便添加了 test_addr)都读 7 号寄存器中的值
4. 将数据 22 写入 15 号寄存器，并使用第二个读地址将其读出

Simulation:

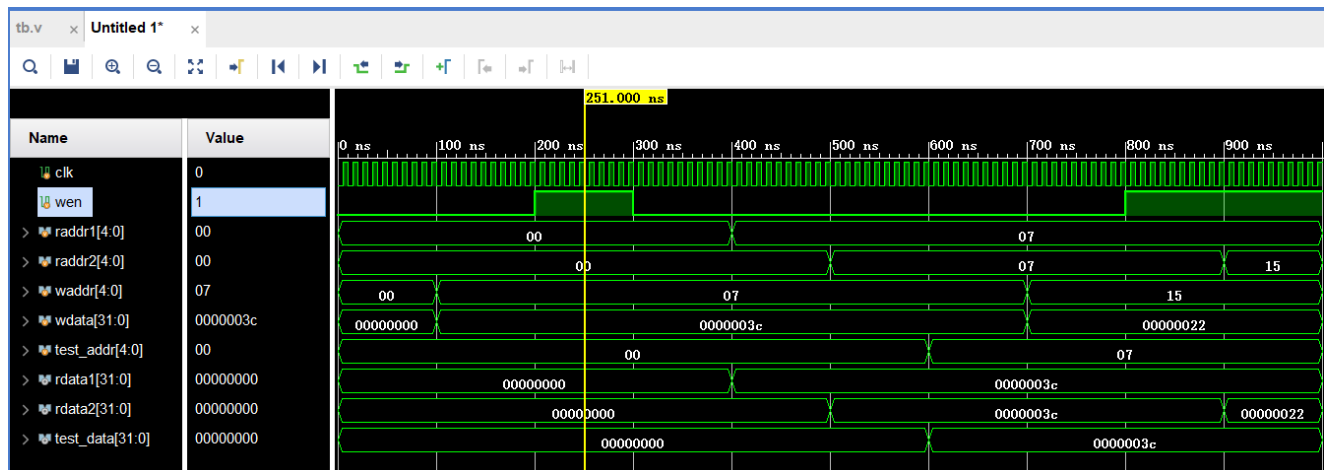


图 9 仿真测试的波形图

从仿真波形图中我们也可以验证：在 WEN 为 1 是将数据 3C 写入了 07 号寄存器，后面当 raddr1、raddr2 和 test_addr 为 07 时，rdata1 和 rdata2 以及 test_data 都显示 3C；后面将数据 22 写入 15 号寄存器也是一样，由此可知，寄存器正常工作。

4.2 上板验证与结果展示

然后我们将 bit 流文件烧录实验箱中测试其实际工作时功能是否正确

-
1. 烧录进板子后显示屏的界面:

图 10 显示屏初始界面

可以看到显示屏显示了我们规定的六个显示读取和写入的地址和数据并且显示了全部 32 个寄存器中存在的值(这也是设计 test_addr 和 test_data 的原因)

2. 将 WEN 置为 1, 并将数据 3C 写入 15 号寄存器

图 11 将 3C 写入 REG15

可以看到在 REG15 的寄存器中成功显示了写入的数据 3C, 写入模块正常

3. 通过两个读地址将 15 号寄存器的值读出

图 12 将 REG15 中的数据读出

可以看到当 RADDR1 和 RADDR2 为 15 时, 其对应的 RDAT1 和 RDAT2 都为 3C——即 REG15 中的数据值, 读入模块正常

4. 测试 WEN 使能信号, 将其置为 0, 往 REG15 中写入数据 6A

图 13 当 WEN 为 0 时无法写入寄存器

可以看到由于使能信号为 0, 所以并没有真正进行数据的写入, REG15 中的数据仍然还是 3C

经过以上的上板测试, 可以知道该寄存器的读写功能编写基本正确

五、问题回答

- 5.1 基础问题: 结合代码和仿真结果讲解寄存器堆的实现

六、收获与体会

1. 学习了 MIPS 指令的结构和寄存器的特点
2. 尝试使用 Verilog 语言设计了寄存器堆, 加深了对 Verilog 语言的理解
3. 学习了使用多个拨码来控制多个输入的获取
4. 学习了如何在显示屏中的一个格子内输出多个拼接内容