

SHANGHAI JIAO TONG UNIVERSITY



计算机系统结构实验报告

实验名称: 简单的类MIPS单周期处理器部件实现-寄存器与存储器

姓 名： 洪瑄锐

学 号： 517030910227

班 级： F1703302

手 机： 15248246044

邮 箱： [1204378645@qq.com](mailto:1204378645@qq.com)

目录

[1. 实验目的 3](#_Toc8840376)

[2. 实验原理与代码实现 3](#_Toc8840377)

[2.1 寄存器Register的实现 3](#_Toc8840378)

[2.1.1主要思想 3](#_Toc8840379)

[2.1.2 Register模块代码 4](#_Toc8840380)

[2.1.3 仿真测试 4](#_Toc8840381)

[2.2内存单元模块Memory的实现 6](#_Toc8840382)

[2.2.1主要思想 6](#_Toc8840383)

[2.2.2 Memory模块代码 7](#_Toc8840384)

[2.2.3 仿真测试 7](#_Toc8840385)

[2.3 带符号扩展的实现 9](#_Toc8840386)

[2.3.1 主要思想 9](#_Toc8840387)

[2.3.2 带符号扩展模块代码 9](#_Toc8840388)

[2.3.3 仿真测试 9](#_Toc8840389)

[3. 感想 10](#_Toc8840390)

[4. 参考文献 10](#_Toc8840391)

# 1. 实验目的

1. 理解CPU寄存器与存储器；
2. egister的实现；
3. Data Memory的实现；
4. 有符号扩展的实现；
5. 使用行为仿真。

# 2. 实验原理与代码实现

## 2.1 寄存器Register的实现

### 2.1.1主要思想

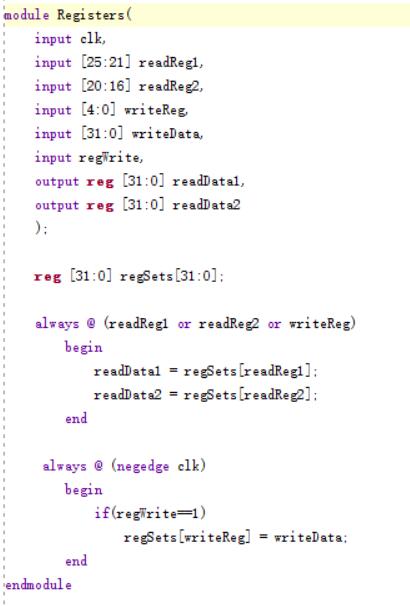
MIPS中一共有32个32位的寄存器，因此需要设置reg [31:0] regSets [31:0]，类似二维数组。

该模块共有7个输入，分别为readReg1（所读寄存器1编号），readReg2（所读寄存器2编号），writeReg（所写寄存器编号），writeData（将要向寄存器内写入的数值），regWrite（是否要向寄存器中写入）。

共有2个输出，readData1（读寄存器1的数值），readData2（读寄存器2的数值）。

读寄存器时只要给出readReg1和readReg2即可，写寄存器时需要clk下跳沿、允许写入信号regWrite，所写寄存器编号writeReg，所写值writeData同时存在。

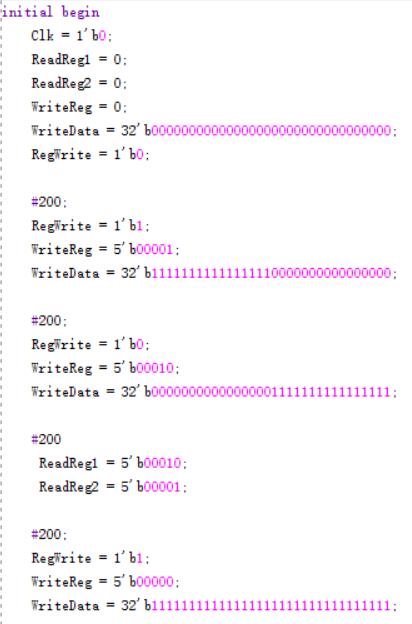
### 2.1.2 Register模块代码

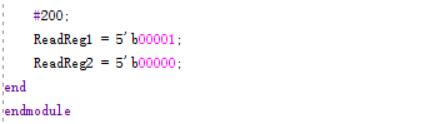


### 2.1.3 仿真测试

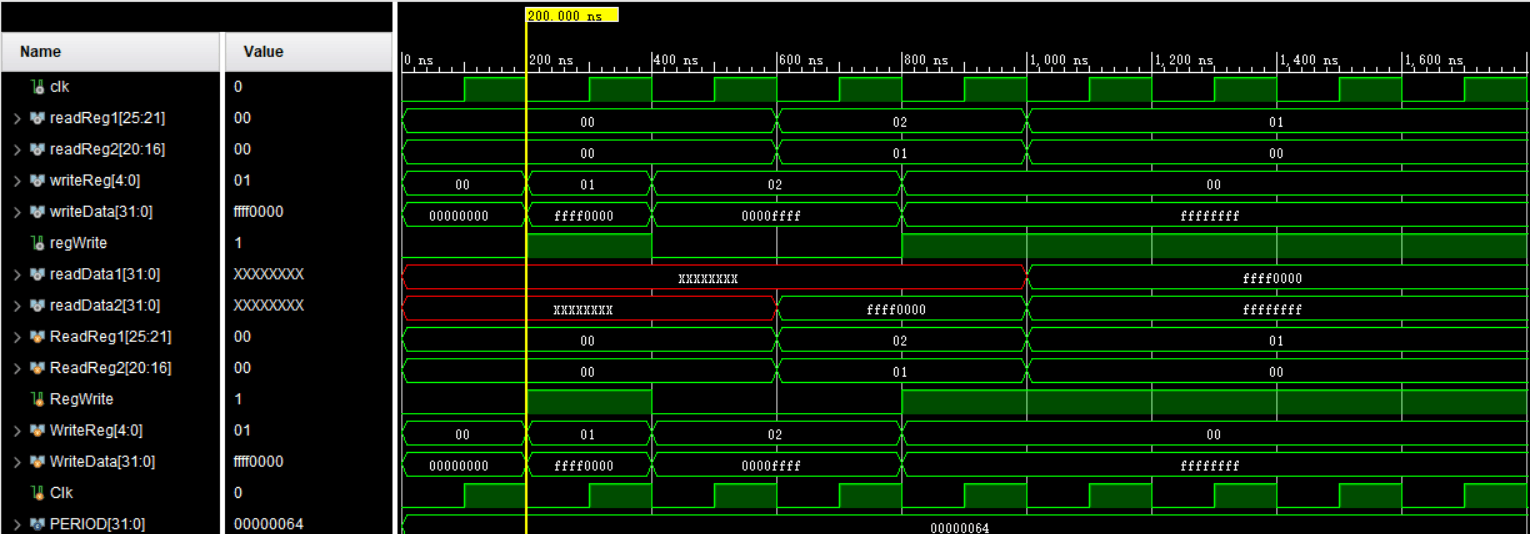
通过给regWrite、WriteData、WriteReg、readReg1、readReg2赋值观察readData1和readData2是否符合正确输出。

**部分仿真代码如下所示：**





**仿真结果：**

****

向0号寄存器写入，但regWrite为0即不允许写入，向1号寄存器写入，并且regWrite为1即允许写入，readData1读入0号寄存器，readData2读入1号寄存器，可以看出上图readData1仍为未知数x，readData1读入ffff0000。再向0号寄存器写入ffffffff且writeReg为1，使readData1读入1号寄存器，readData2读入0号寄存器，则如上图所示readData1为ffff0000，readData2为ffffffff。仿真结果正确。

## 2.2内存单元模块Memory的实现

### 2.2.1主要思想

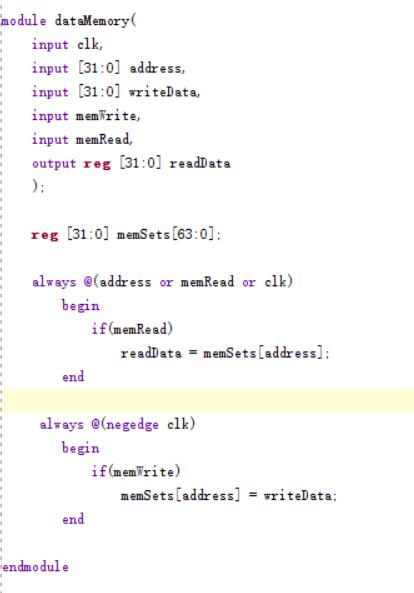
内存单元模块与register类似，由于写数据也要考虑信号同步，因此也需要时钟。

共4个输入，address（写入或读出的内存地址），writeData（将要写入内存的数据），

memWrite（是否允许写数据信号），memRead（是否允许读数据信号）。

共1个输出，readData（所读数据）。

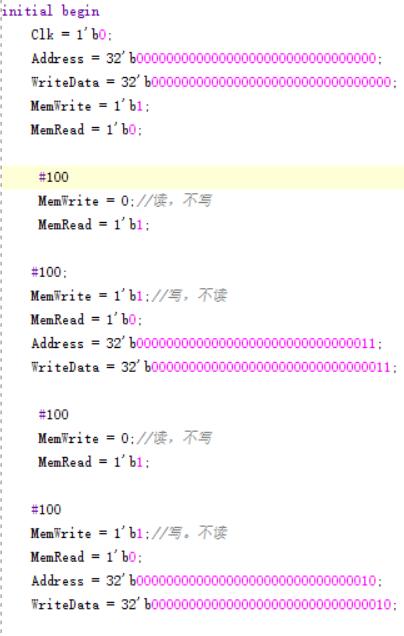
### 2.2.2 Memory模块代码

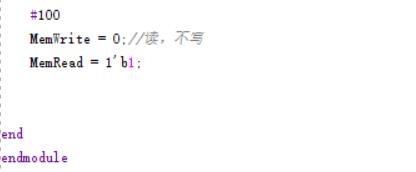


### 2.2.3 仿真测试

通过给memread，memwrite，address，writeData赋值检查readData。本次仿真采用先写再读的方式。

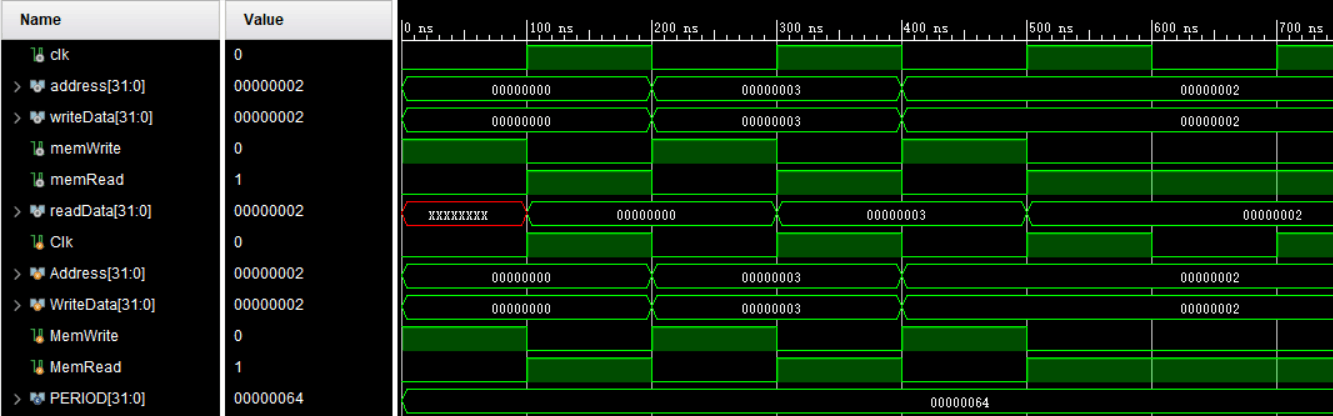
**部分代码：**





**仿真结果：**

第100nsreadData开始读值，后每200ns变化一次。分析可知仿真结果正确。

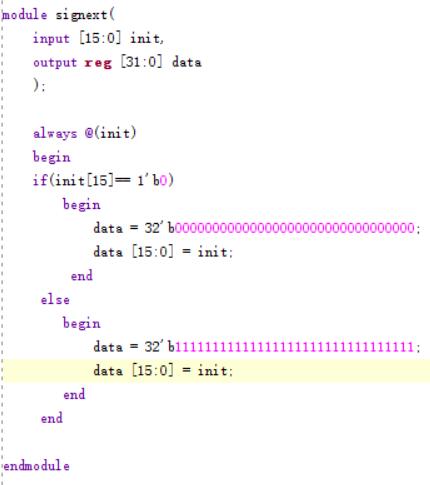


## 2.3 带符号扩展的实现

### 2.3.1 主要思想

将16位有符号数扩展为32位有符号数，采用ifelse语句，如果16位有符号数最高位为0，则在其前补16个0，否则补16个1。

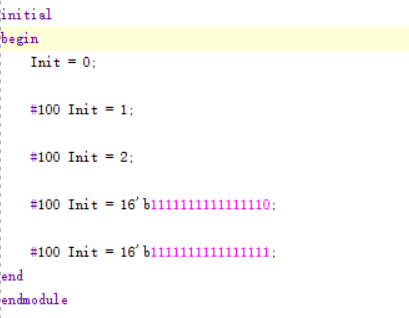
### 2.3.2 带符号扩展模块代码



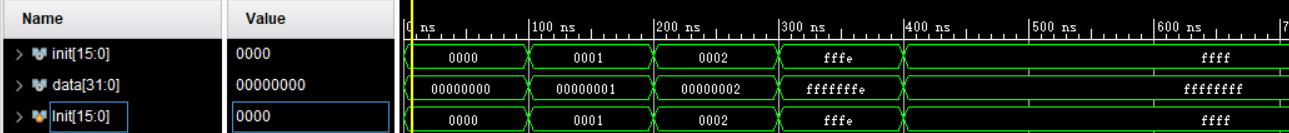
### 2.3.3 仿真测试

通过对init赋最高位不同的值得到正确的扩展数data。

**部分代码：**



**仿真结果：**



分析知仿真正确。

# 3. 感想

在本次实验中我实现了简单cpu中的寄存器模块，内存模块和带符号扩展，深入理解了寄存器和内存模块的相关信号是如何起作用，知道了其内部工作原理。另外本次实验使我对verilog的基础语法使用的更加熟练。

感谢老师和同学们在本次实验中对我的教导和帮助。

# 4. 参考文献

《cs145实验指导书lab04》