**计算机系统结构试验**

**Lab04: 简单的类MIPS单周期处理器功能部件的设计与实现（二）**

姓名：N/A

摘要

在Lab04中，我进行了有关register、data memory和有符号数位扩展的实验，包括寄存器和内存的设计实现、解码指令、读写寄存器等，根据输入指令类型和操作，设置适当的输出。通过本次实验，我进一步加深了对Verilog语言的理解和运用，掌握MIPS处理器的Register File组成部分的设计和实现方法，并且能够使用仿真工具进行验证和调试，给我带来宝贵的经验和收获。

目录

[摘要 1](#_Toc166104921)

[**1.** 实验目的 2](#_Toc166104922)

[**2.** 原理分析 2](#_Toc166104923)

[2.1 Vivado工程的基本组成 2](#_Toc166104924)

[2.2 Registers模块的原理 2](#_Toc166104925)

[2.3 dataMemory模块的原理 2](#_Toc166104926)

[2.4 signext模块的原理 2](#_Toc166104927)

[**3.** 功能实现 3](#_Toc166104928)

[3.1 Registers模块的实现 3](#_Toc166104929)

[3.2 dataMemory模块的实现 3](#_Toc166104930)

[3.3 signext模块的实现 3](#_Toc166104931)

[**4.** 结果验证 4](#_Toc166104932)

[4.1 Registers模块的测试 4](#_Toc166104933)

[4.2 dataMemory模块的测试 4](#_Toc166104934)

[4.3 signext模块的测试 5](#_Toc166104935)

[**5.** 总结与反思 6](#_Toc166104936)

**1.** 实验目的

（1）理解寄存器、数据存储器、有符号扩展单元的IO定义；

（2）Registers的设计实现；

（3）Data Memory的设计实现；

（4）有符号扩展部件的实现；

（5）对功能模块进行仿真。

**2.** 原理分析

2.1 Vivado工程的基本组成

（1）Registers.v文件

（2）dataMemory.v文件

（3）signext.v文件

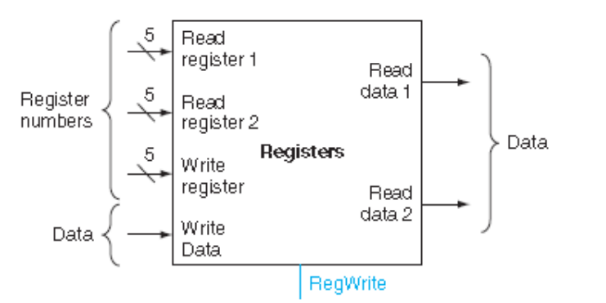
（4）Registers\_tb.v激励文件

（5）dataMemory \_tb.v激励文件

（6）signext \_tb.v激励文件

2.2 Registers模块的原理

寄存器是指令操作的主要对象，MIPS中一共有32个32位的寄存器，用作数据的缓存。寄存器模块结构如下：



2.3 dataMemory模块的原理

Data memory是用来存储运行完成的数据，或者初始化的数据。内存模块的编写与register类似，由于写数据也要考虑信号同步，因此也需要时钟。Memory模块的结构如下：

图示

描述已自动生成

2.4 signext模块的原理

Signext模块将 16 位有符号数扩展为 32 位有符号数。模块结构如下：

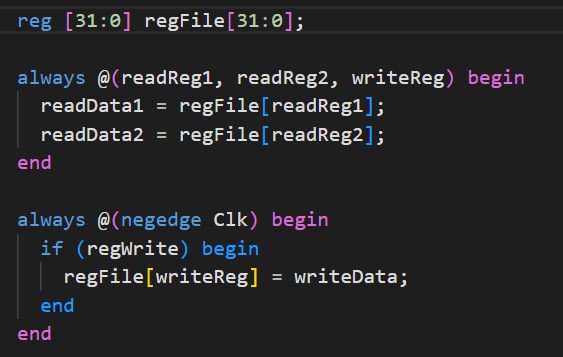
图示, 维恩图

描述已自动生成

**3.** 功能实现

3.1 Registers模块的实现

由于不确定WriteReg，WriteData，RegWrite 信号的先后次序，采用时钟的下降沿作为写操作的同步信号，防止发生错误。因此模块逻辑为：当readReg1，readReg2，writeReg其中一个发生变化时，将寄存器内值读到输出；当遇到Clk下降沿时，若regWrite使能，则将输入写入寄存器。代码如下：



3.2 dataMemory模块的实现

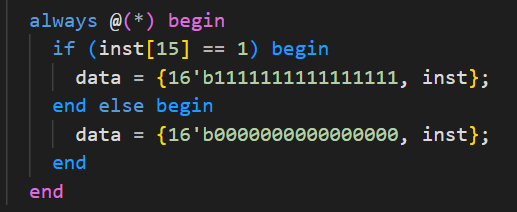
dataMemory的实现与register基本一致，但只有一个地址输入和一个数据输出。为了防止对同一个地址的读写竞争，设置为当memRead和memWrite同时使能时，memWrite被禁用。代码如下：

屏幕的截图

描述已自动生成

3.3 signext模块的实现

Signext模块将 16 位有符号数扩展为 32 位有符号数。因此当输入数值为正数（第15位为0）时位扩展16位0，当输入数值为负数（第15位为1）时位扩展16位1。代码如下：



**4.** 结果验证

4.1 Registers模块的测试

编写激励文件设置各输入初值，代码如下：

文本

描述已自动生成

测试结果如图所示：

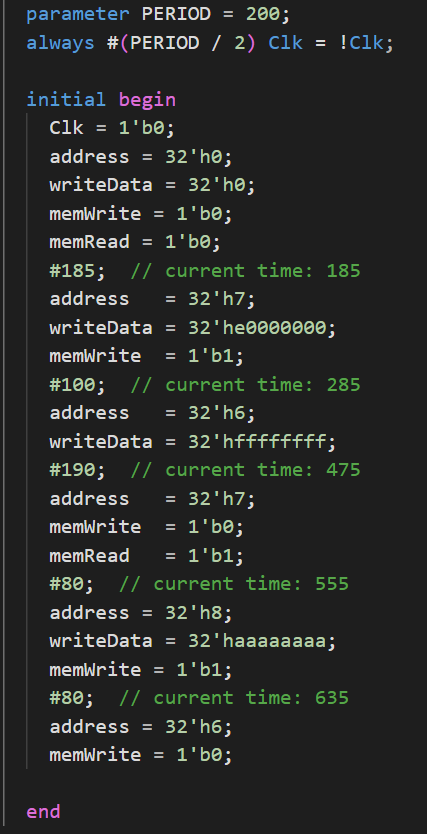
截图里有图片

描述已自动生成

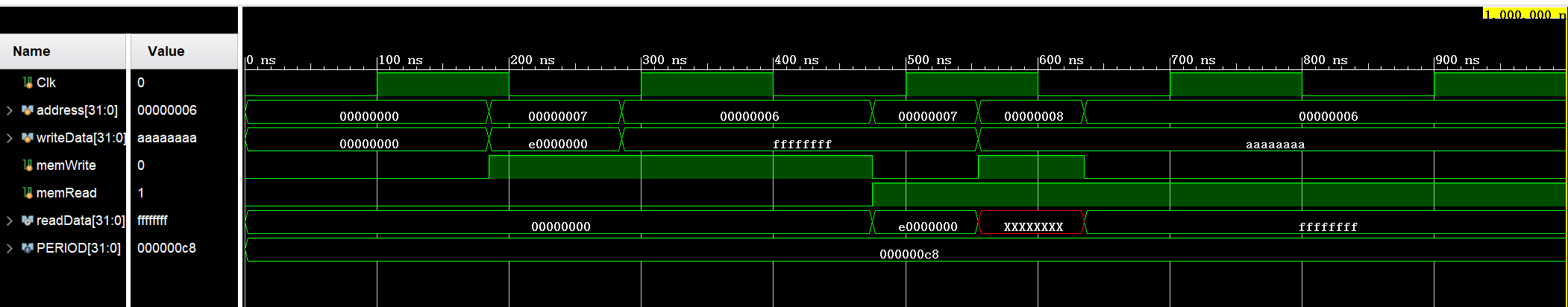
因为初始状态下模块内寄存器未被初始化，所以Registers模块在开始时输出为未定值；在写入reg15和reg0a后，读取值与写入值一致。

4.2 dataMemory模块的测试

编写激励文件设置各输入初值，代码如下：



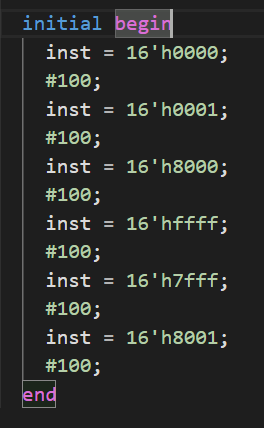
测试结果如图所示：



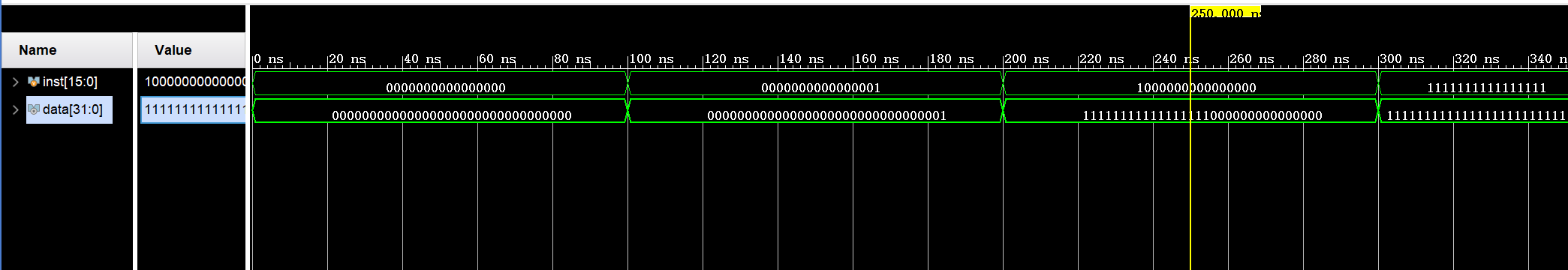
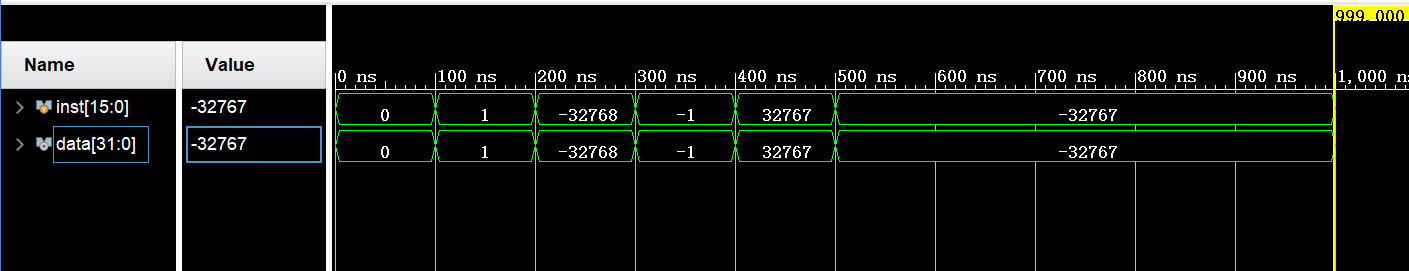
475ns-555ns时，读取到了mem[7]在185ns-285ns时写入的值0xe0000000;475ns-555ns时，发生读写竞争，写信号被禁用；635ns-1000ns时，读取到了mem[6]在285ns-475ns时写入的值0xffffffff。

4.3 signext模块的测试

编写激励文件设置各输入初值，代码如下：



测试结果如图所示：



输出与预期一致。

**5.** 总结与反思

在Lab04中，在本次实验中，我主要关注了register、data memory和符号数位扩展部件的设计和实现。通过使用Vivado开发环境，我能够更好地理解Verilog HDL的基本语法和编程技巧。通过这次实验，我掌握了使用if-else块编写分支逻辑的方法，并学习了存储部件的内部结构实现。

我要感谢课程组为我们提供的详细指导书，它为我提供了清晰的实验步骤，使我能够更好地理解和实践所学的知识。通过这次实验，我不仅巩固了Verilog的基础知识，还为的学习和设计打下了坚实的基础。