实验四报告

一、实验名称

简单的类 MIPS 单周期处理器实现-寄存器与内存

二、实验目的

1. 理解 CPU 的寄存器与内存
2. 使用 Verilog 语言设计存储器件
3. 使用 ISim 进行行为仿真

三、实验范围

1. ISE 的使用
2. register 的实现
3. data memory 的实现
4. 有符号扩展的实现

四、主要的设计思想和测试仿真

1. 寄存器单元模块 Register

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。

主要的操作就是读寄存器和写寄存器。这里采用时钟的下降沿作为写操作的同步信号，防止发生错误。

always @ (readReg1 or readReg2) //向寄存器读数据

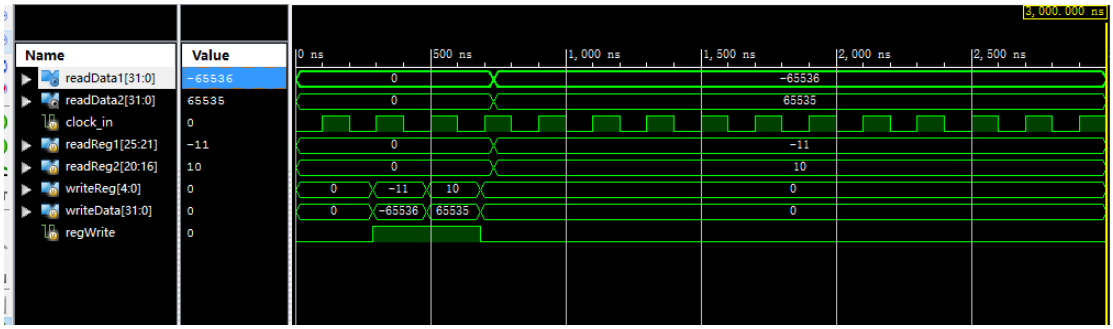
```
    begin
        readData1 = regFile[readReg1];
        readData2 = regFile[readReg2];
    end

always @ (negedge clock_in) //向寄存器写数据
    begin
        if(regWrite)
            regFile[writeReg] = writeData;
    end
```

register 的仿真代码：

```
        #285;
        regWrite = 1'b1;
        writeReg = 5'b10101;
        writeData = 32'b11111111111111110000000000000000;

        #200;
        writeReg = 5'b01010;
        writeData = 32'b00000000000000001111111111111111;

        #200;
```

```
        regWrite = 1'b0;
        writeReg = 5'b00000;
        writeData = 32'b00000000000000000000000000000000;

        #50;
        readReg1 = 5'b10101;
        readReg2 = 5'b01010;
```

仿真波形如下：



2. 内存单元模块 memory

内存模块与 register 模块类似，由于写数据也要考虑信号同步，因此也要 clock_in。

关于内存的操作也是两个，即读内存和写内存。

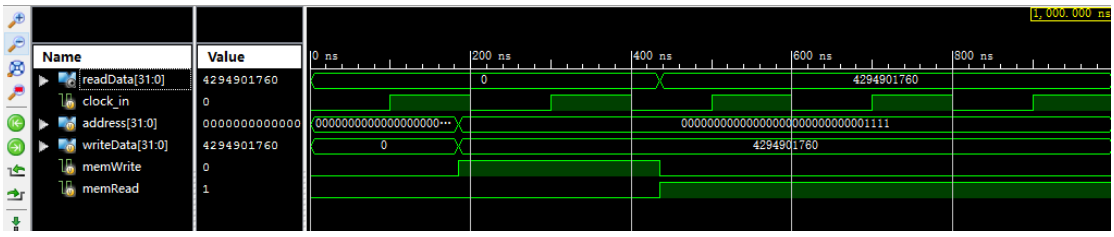
```
always @ (memRead or address) //向存储器中读数据
    begin
        readData = memFile[address];
    end
```

```
always @ (negedge clock_in) //向存储器中写数据
    begin
        if(memWrite)
            memFile[address] = writeData;
    end
```

内存的仿真代码：

```
        #185;
        memWrite = 1'b1;
        address = 32'b00000000000000000000000000001111;
        writeData = 32'b11111111111111110000000000000000;

        #250;
        memRead = 1'b1;
        memWrite = 1'b0;
```

仿真波形如下：



3. 带符号扩展

将 16 位带符号数扩展为 32 位带符号数。

主要想法就是判断 16 位数的最高位，如果是 1，则前面全扩展成 1，如果是 0，则全扩展成 0.

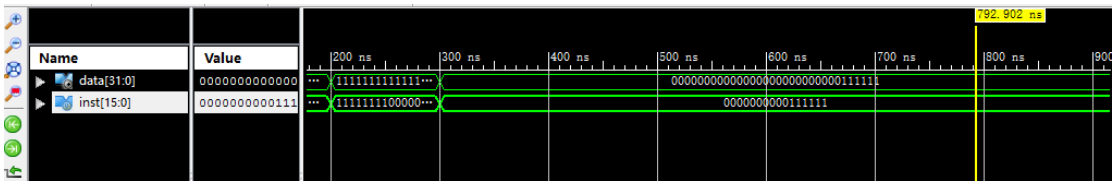
关键代码：

```
always @(inst)
    begin
        data = inst;
        if(data[15])
            data[31:16] = 16'b1111111111111111;
        else
            data[31:16] = 16'b0000000000000000;
    end
```

仿真测试代码：

```
        #100;
        #100 inst = 16'b1111111100000000;
        #100 inst = 16'b0000000000111111;
```

仿真波形如下：



五、心得体会

本次实验较为简单，主要是增加了时钟这么一个概念，在写寄存器，写内存时都要涉及到时钟。而在带符号扩展中，主要就是判断最高位即可。

通过这两次的实验，逐渐掌握了一些小模块的写法，这对于后面整个处理器的设计无疑是有益的。