

计算机系统结构实验报告 Lab04

简单的类 MIPS 单周期处理器功能部件的设计与实现: 寄存器、存储器与有符号扩展

徐阳 521021910363

2023 年 5 月 3 日

目录

1 简介	2
2 实验目的	2
3 实验原理与功能实现	2
3.1 寄存器 Registers	2
3.1.1 原理分析	2
3.1.2 代码实现	3
3.2 数据存储器 dataMemory	3
3.2.1 原理分析	3
3.2.2 代码实现	4
3.3 有符号扩展单元 signext	5
3.3.1 原理分析	5
3.3.2 代码实现	5
4 仿真验证	6
4.1 寄存器 Registers 仿真验证	6
4.2 数据存储器 dataMemory 仿真验证	6
4.3 有符号扩展单元 signext 仿真验证	7
5 总结与思考	8
6 致谢	8

1 简介

在本实验中，我学习并在 Verilog 中实现了 Mips 处理器中的几个核心部件的模块：寄存器、数据存储器、有符号扩展单元，他们分别实现了暂时存放数据与计算结果，存储较大量数据，对立即数进行有符号数拓展的功能。最后通过行为仿真验证了程序的正确性。

2 实验目的

1. 理解寄存器、数据存储器、有符号扩展单元的原理
2. 使用 Verilog 设计与实现寄存器、数据存储器、有符号扩展单元
3. 掌握激励文件的编写，使用行为仿真验证程序正确性

3 实验原理与功能实现

3.1 寄存器 Registers

3.1.1 原理分析

寄存器是指令操作的主要对象，寄存器模块的 IO 定义如图 reffigure1 所示。MIPS 中一共有 32 个 32 位的寄存器数据，用作数据的缓存，故每个 32 位数据需要 5 位二进制数来寻址。所以读写地址 readReg1, readReg2 和 writeReg 接口均为五位二进制数，读写数据 readData1, readData1, writeData 为 32 位二进制数。而 regwrite 和 clk 为写信号和时钟信号。

设计模块时，我们已经要考虑时钟因素了。在这里，我们让读取数据的操作在时钟信号稳定为高电平后进行，而写数据在时钟的下降沿进行，这样可以防止读写的数据是还没及时改变时的错误数据。

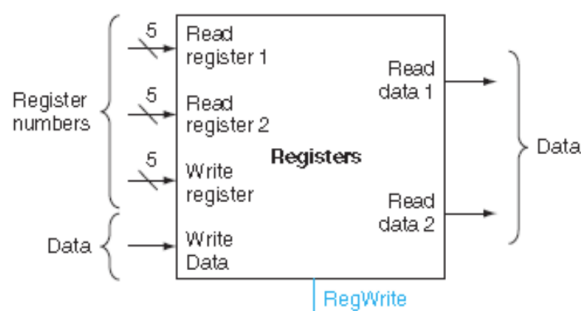


图 1: 寄存器 (Registers) 的主要接口

3.1.2 代码实现

理解了原理，代码实现就并不困难。但是需要注意，如果我们不初始化寄存器，其默认是会是 x，所以我们要在 initial 模块中初始化各寄存器值为零。

——Registers.v——

```

1 ...
2     reg [31:0] RegFile[31:0];    //32个32位寄存器
3     integer i;
4     initial begin
5         for(i = 0;i <= 31;i = i+1)
6             RegFile[i] = 0;
7     end
8
9     always @ (!clk)
10    begin
11        readData1 = RegFile[readReg1];
12        readData2 = RegFile[readReg2];
13    end
14
15    always @ (negedge clk)
16    begin
17        if(regWrite)
18            RegFile[writeReg] = writeData;
19    end
20 ...
21 }

```

3.2 数据存储器 dataMemory

3.2.1 原理分析

数据存储器 dataMemory，又称为内存，其功能是存储大量的数据，内存模块的 IO 定义如图 2 所示。其输入为读写控制信号 memRead 和 memWrite、时钟信号 clk、写数据 writeData 以及读写地址 address，输出为读数据 readData。

实际情况中的内存有页表、虚拟地址、物理地址等等复杂情况，在仿真中我们对其进行了简化，本实验中的内存为有 64 个 32 位数据的存储器，但为了贴合实际，地址仍为 32 位，但加入了地址范围判定，超出范围的地址将不予写入且读取为 0。

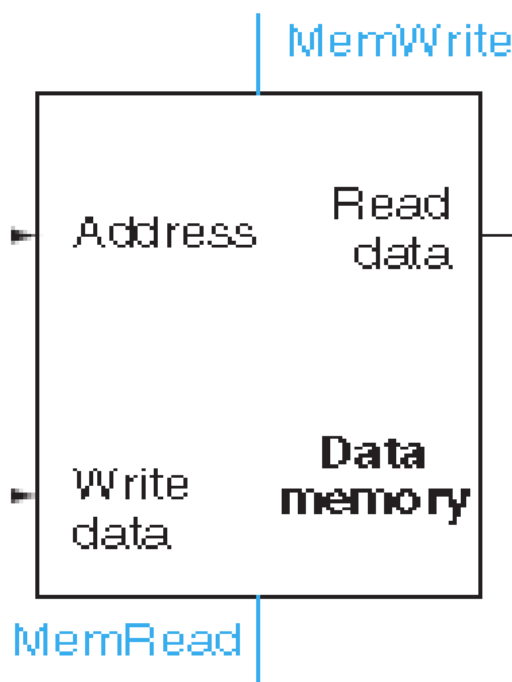


图 2: 内存 (dataMemory) 的主要接口

3.2.2 代码实现

实际的指令解析中,数据的地址是按字节寻址,地址应当除以四的,因为在当前模块还没有涉及,在这里我们进行了简化。

与寄存器类似地,我们让读取数据的操作在读信号 memRead 改变后进行,而写数据仍在时钟的下降沿进行,具体代码如下。

——dataMemory.v——

```

1 ...
2     reg [31:0] memFile[0:63];
3
4     integer i;
5     initial begin
6         readData = 0;
7         for(i = 0;i < 64;i = i+1)
8             memFile[i] = 0;
9     end
10
11     always @ (memRead or address)
12     begin
13         if (memRead)
14             begin

```

```

15         if(address <= 63)
16             readData = memFile[address];
17         else
18             readData = 0;
19     end
20 end
21
22 always @ (negedge Clk)
23 begin
24     if (memWrite && address <= 63)
25         memFile[address] = writeData;
26     end
27 ...
28 }

```

3.3 有符号扩展单元 signext

3.3.1 原理分析

有符号扩展单元 signext 的作用是将 16 位有符号立即数拓展为 32 位有符号数。原理图如图 3 所示，输入为 16 位二进制待拓展立即数 inst，输出为拓展后的 32 位二进制数 data。因为是有符号位拓展，我们可以在原数的高 16 位填充上原数据的最高位（符号位）相同的 16 个数。

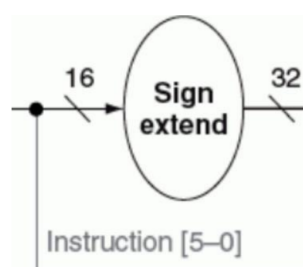


图 3: 带符号扩展单元 (signext)

3.3.2 代码实现

代码实现利用了 Verilog 中的花括号的拼接功能，具体实现代码如下。

—— signext.v ——

```

1 module signext(
2     input [15:0] inst,

```

```

3     output [31:0] data
4 );
5     assign data = { {16{inst[15]}}, inst[15 : 0] };
6 endmodule
7 }

```

4 仿真验证

4.1 寄存器 Registers 仿真验证

依实验指导书编写激励文件，进行仿真验证。在仿真中对多种情况进行了检验，仿真结果如图 4 所示。可见在没有进行数据写入时，因为我们进行了初始化，故 readData 一开始为 0。仿真结果正确，我们实现了寄存器 Registers。

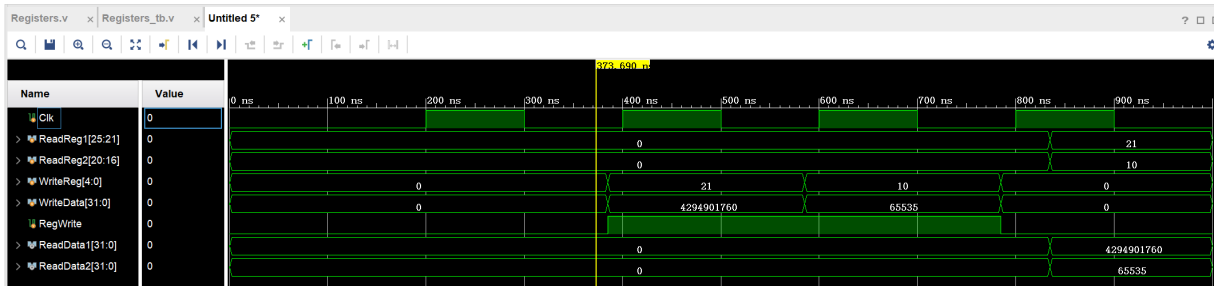


图 4: 寄存器 Registers 仿真验证

4.2 数据存储器 dataMemory 仿真验证

依实验指导书编写激励文件，进行仿真验证。在仿真中对多种情况进行了检验，对读、写、同时读写都进行了验证，仿真结果如图 5 所示。仿真结果正确，我们实现了数据存储器 dataMemory。

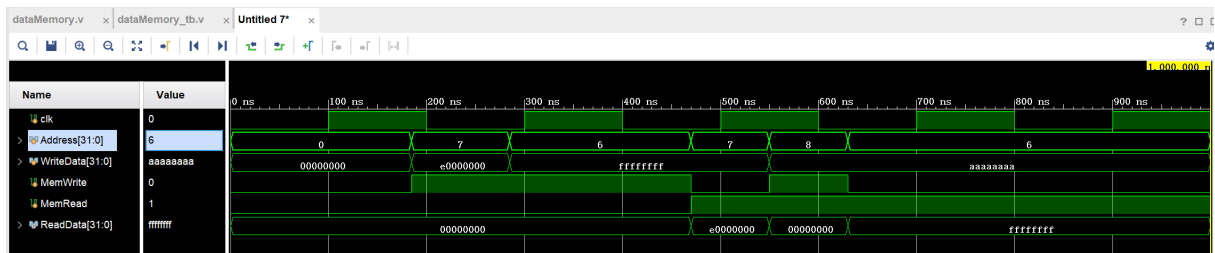


图 5: 数据存储器 dataMemory 仿真验证

4.3 有符号扩展单元 signext 仿真验证

依实验指导书编写激励文件，进行仿真验证。在仿真中对多种情况进行了检验，仿真结果如图 6所示。我们特别截取了二进制的结果如图 7所示。仿真结果正确，我们实现了有符号扩展单元 signext。

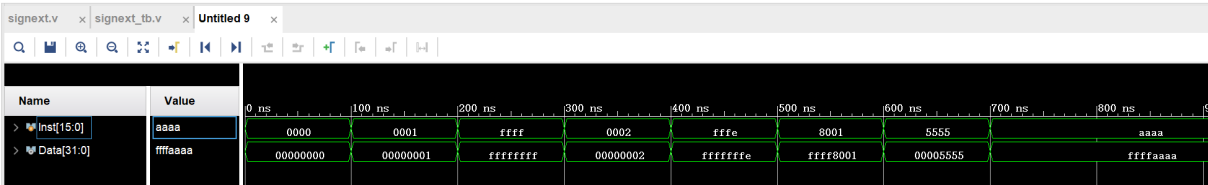


图 6: 有符号扩展单元 signext 仿真验证

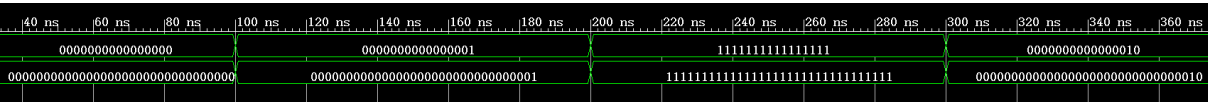


图 7: signext 仿真结果细节

5 总结与思考

本实验的目标是理解并实现寄存器 Registers、存储器 dataMemory、有符号扩展单元 signext，并分别通过编写激励文件，进行行为仿真验证了他们的正确性。

在本次实验中我们实现的模块功能比较简单，比如有符号扩展单元 signext 模块没有考虑拓展信号和有符号/无符号控制信号的作用，存储器 dataMemory 的地址也是一个地址存了 32 位的数据而不是生活实际的 1Byte，这些地方在以后都存在着改进的空间，当前实现的简单功能为之后的进阶打下了基础。

在本次实验中，我也对 always 模块中的敏感列表有了更深刻的理解，在其中写入时钟的上升/下降沿和稳定值是有着许多些微但重要的差别的。同时，我对阻塞赋值和非阻塞赋值也有了更深刻的理解。

同时在本次实验中，我们依然是从模块化出发，设计一个又一个元件，至此，我们已经理解并实现了 mips 处理器中的大部分元件的运行原理和逻辑，为整个处理器的实现做足了充分的准备。

6 致谢

感谢老师课堂上对于相关内容的教学。

感谢助教课堂上对于我的疑问的解答。

感谢老师与助教提供的实验手册对于实验的指导。