

组成原理课程第 三 次实验报告

实验名称：寄存器堆实现

学号： 2012679 姓名： 王娇妹 班次： 张金老师

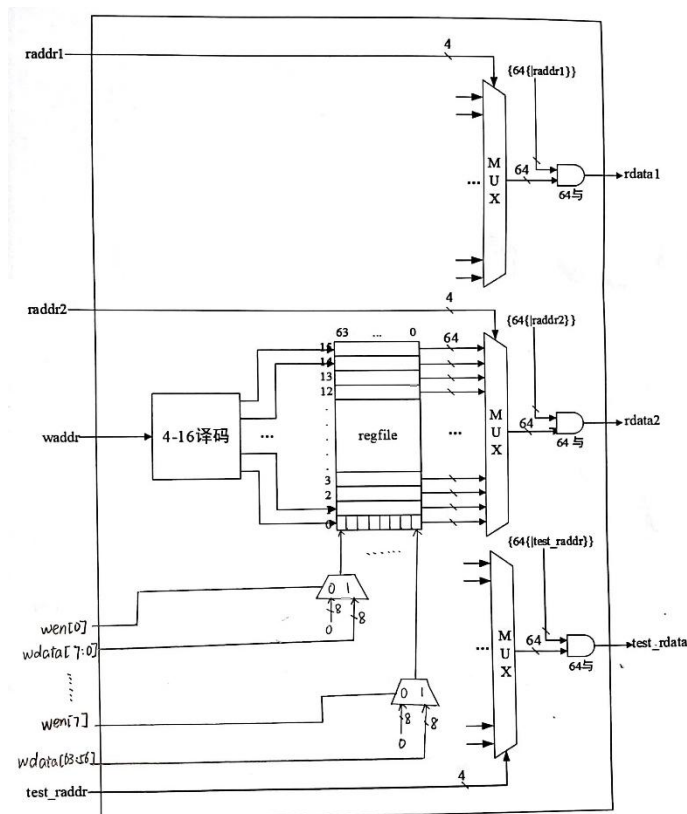
一、实验目的

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. 为后续设计 cpu 的实验打下基础。

二、实验内容说明

1. 在原有寄存器堆的基础上，将位宽由 32 位扩展为 64 位，将容量缩减为 16 个。
2. 在寄存器写入时，对 wen 信号进行优化，不能一次写入 64 位数
3. 完成仿真文件，进行波形仿真
4. 修改模块图。

三、实验原理图



将寄存器堆的位宽扩展为 64 位，容量缩减为 16 个，那么读写的地址 waddr、raddr 只需要 4 位，通过一个 4-16 译码器判断寄存器位置。regfile 需要改为 64 位，wen 是 8 位。

根据 wen 每一位的数据，将对应位置的 wdata 写入寄存器。

若 wen[0]为 0，rf[waddr][7:0]全为 0，若 wen[0]为 1，wdata[7:0]写入 rf[waddr][7:0]……
若 wen[7]为 0，rf[waddr][63:56]全为 0，若 wen[7]为 1，wdata[63:56]写入 rf[waddr][63:56]。

四、实验步骤

1、regfile.v 文件的修改

将寄存器堆的位宽扩展为 64 位，容量缩减为 16 个，reg [15:0] rf[63:0]。

读写的地址 waddr、raddr 只需要 4 位，如[3 :0] raddr1、[3 :0] waddr。

根据 wen 每一位的数据，将对应位置的 wdata 写入寄存器，每次写入一个字节，wen 是 8 位，[7 :0] wen。例如，若 wen[0]为 0，rf[waddr][7:0]全为 0，wen[0]为 1，wdata[7:0] 写入 rf[waddr][7:0]。

```
module
regfile(
    input          clk,
    input          [7 :0]  wen,//wen 改为 8 位，每一位对应一个字节
    input          [3 :0] raddr1, //改为 4 位
    input          [3 :0] raddr2, //改为 4 位
    input          [3 :0] waddr, //改为 4 位
    input          [63:0] wdata,//改为 64 位
    output reg [63:0] rdata1,//64 位
    output reg [63:0] rdata2,//64 位
    input          [3 :0] test_addr, //改为 4 位
    output reg [63:0] test_data//64 位
);
reg [15:0] rf[63:0];//容量 16，位宽 64
//初始化 16 个寄存器，全为 0
integer i = 0;
initial
    for(i = 0;i < 16;i = i + 1)
        rf[i]<= 0;
//传入数据
always @(posedge clk)
begin
    if (wen[0]) //****,***1,0-7 位传入
    begin
        rf[waddr] [7:0]<= wdata[7:0];
    end
    .....
    if (wen[7]) //1***,****,56:63 位传入
    begin
        rf[waddr][63:56] <= wdata[63:56];
    end
end
end
```

2、testbench.v 文件的修改

```
module tb;
// Inputs
reg clk;
reg [7 :0]  wen; //wen 改为 8 位，
reg [3 :0] raddr1; //改为 4 位
```

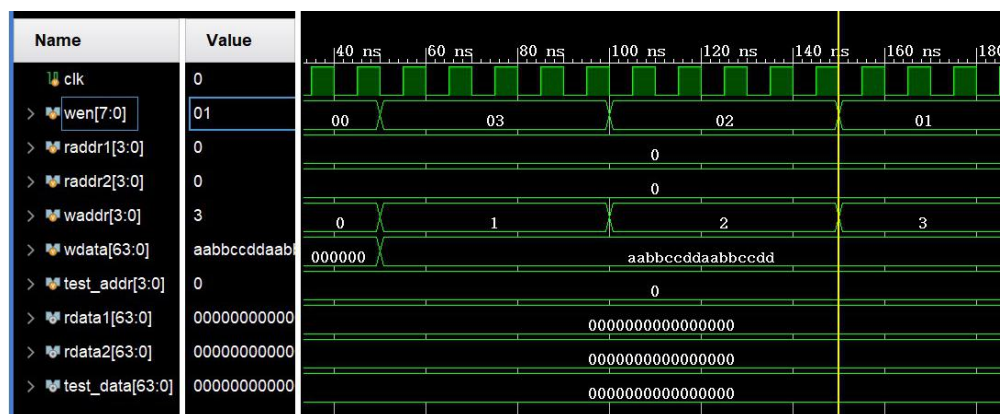
```

reg [3 :0] raddr2;
reg [3 :0] waddr;
reg [63:0] wdata;//64 位
reg [3 :0] test_addr;
// Outputs
wire [63:0] rdata1;//64
wire [63:0] rdata2;//64
wire [63:0] test_data;//64
// Instantiate the Unit Under Test (UUT)
regfile rf(
.clk(clk),
.wen(wen),
.raddr1(raddr1),
.raddr2(raddr2),
.waddr(waddr),
.wdata(wdata),
.rdata1(rdata1),
.rdata2(rdata2),
.test_addr(test_addr),
.test_data(test_data)
);
initial begin
    #50;
    waddr = 4'b0001;//写入 1 号寄存器
    wdata = 64'haabbccddaabbccdd;
    wen = 8'b00000011;
    #50;
    waddr = 4'b0010; //写入 2 号寄存器
    wdata = 64'haabbccddaabbccdd;
    wen = 8'b00000010;
    #50;
    waddr = 4'b0011; //写入 3 号寄存器
    wdata = 64'haabbccddaabbccdd;
    wen = 8'b00000001;
    #50; //读 1 号寄存器
    raddr1 = 4'h1;
    #50; //读 2 号寄存器
    raddr2 = 4'h2;
    #50; //读 3 号寄存器
    test_addr = 4'h3;
end
always #5 clk = ~clk;
endmodule

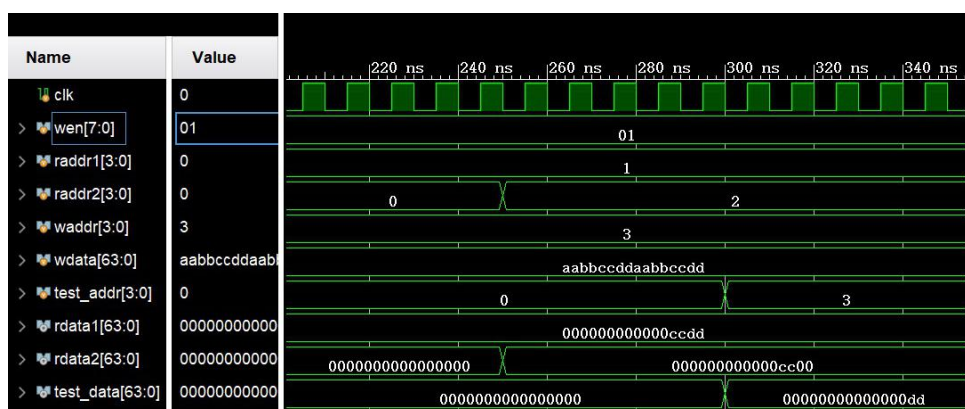
```

五、实验结果分析

写:



读:



写入数据时:

将 aabbccddaabbccdd 写入 1、2、3 号寄存器。

写入 1 号寄存器时, wen = 00000011, 存入的数据应为 000000000000ccdd;

写入 2 号寄存器时, wen = 00000010, 存入的数据应为 000000000000cc00;

写入 3 号寄存器时, wen = 00000001, 存入的数据应为 00000000000000dd。

读取数据时:

读 1 号寄存器, 为 000000000000ccdd;

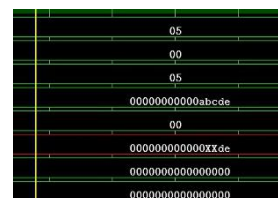
读 2 号寄存器, 为 000000000000cc00;

读 3 号寄存器, 为 00000000000000dd。

经过测试, 结果正确。

六、总结感想

在实验过程中, 我在仿真测试时的读取数据过程中遇到了问题, 输出如右图所示, 经过检查, 读数有问题是因为没有初始化 rf, 再添加了初始化 rf 所有数都是 0 的初始化步骤后, 能够正常读取数据。



通过此次实验, 我学习了寄存器堆的原理和设计方法, 了解了 MIPS 指令结构和源操作数/目的操作数的概念。