

实验报告

2021 年 5 月 14 日

成绩: _____

姓名	*****	学号	*****	班级	*****
专业	计算机科学与技术		课程名称	计算机组成原理课程设计	
任课老师	曾虹	指导老师	曾虹	机位号	31
实验序号	5	实验名称	存储器设计实验		
实验时间	2021.5.14	实验地点	一教 225	实验设备号	31

一、实验程序源代码

```
`timescale 1ns / 1ps

module memory(clk_m,Mem_Write,
              R_Addr_A,R_Addr_B,W_Addr,Write_Reg,rst,clk,
              ALU_OP,
              Mem_Addr,M_R_Data,R_data_A,R_data_B,ALU_F);
    input clk_m,Mem_Write;

    input [4:0]R_Addr_A;//A 端口读寄存器地址

    input [4:0]R_Addr_B;//B 端口读寄存器地址

    input [4:0]W_Addr;//写寄存器地址

    input Write_Reg;
    input [2:0]ALU_OP;
    input rst,clk;
    output [31:0]M_R_Data;
    output [31:0]R_data_A;
    output [31:0]R_data_B;
    wire ZF,OF;
    output [31:0] ALU_F;
    input [7:0] Mem_Addr;

    RAM_B Data_RAM (
        .clka(clk_m), // input clka
        .wea(Mem_Write), // input [0 : 0] wea
        .addra(Mem_Addr[7:2]), // input [5 : 0] addra
        .dina(R_data_B), // input [31 : 0] dina
        .douta(M_R_Data) // output [31 : 0] douta
    )
endmodule
```

```

    );

    REGS
    REGS_1(R_data_A,R_data_B,ALU_F,R_Addr_A,R_Addr_B,W_Addr,Write_Reg,rst
    ,clk);

    ALU ALU_1(ALU_OP,R_data_A,M_R_Data,ALU_F,ZF,OF);

endmodule

module
REGS(R_Data_A,R_Data_B,ALU_F,R_Addr_A,R_Addr_B,W_Addr,Write_Reg,rst,c
lk);

    input  clk;//写入时钟信号

    input  rst;//清零信号

    input  Write_Reg;//reg

    input  [4:0]R_Addr_A;//A 端口读寄存器地址

    input  [4:0]R_Addr_B;//B 端口读寄存器地址

    input  [4:0]W_Addr;//写寄存器地址

    input  [31:0]ALU_F;//写入数据

    output [31:0]R_Data_A;//A 端口读出数据

    output [31:0]R_Data_B;//B 端口读出数据


    integer i;
    reg [31:0] REG_Files[0:31];
    initial
        for(i=0;i<32;i=i+1) REG_Files[i]<=0;
    always@(posedge clk or posedge rst)
    begin
        if(rst)
            for(i=0;i<32;i=i+1) REG_Files[i]<=0;
        else
            if(Write_Reg) REG_Files[W_Addr]<=ALU_F;
    end
    assign R_Data_A=REG_Files[R_Addr_A];
    assign R_Data_B=REG_Files[R_Addr_B];

```

```

endmodule

module ALU(ALU_OP,A,B,F,ZF,OF);
    input  [2:0] ALU_OP;
    input  [31:0] A;
    input  [31:0] B;
    output [31:0] F;
    output  ZF;
    output  OF;
    reg [31:0] F;
    reg  C,ZF,OF;
    always@(*)
    begin
        C=0;
        OF=0;
        case(ALU_OP)
            3'b000:begin F=A&B; end
            3'b001:begin F=A|B; end
            3'b010:begin F=A^B; end
            3'b011:begin F=~(A|B); end
            3'b100:begin {C,F}=A+B;OF = A[31]^B[31]^F[31]^C; end
            3'b101:begin {C,F}=A-B;OF = A[31]^B[31]^F[31]^C; end
            3'b110:begin F=A<B; end
            3'b111:begin F=B<<A; end
        endcase
        ZF = F==0;
    end
endmodule

```

二、仿真测试代码

```

`timescale 1ns / 1ps

module test2;

    // Inputs
    reg clk_m;
    reg Mem_Write;
    reg [4:0] R_Addr_A;
    reg [4:0] R_Addr_B;
    reg [4:0] W_Addr;
    reg Write_Reg;
    reg rst;
    reg clk;

```

```

reg [2:0] ALU_OP;
reg [7:0] Mem_Addr;

// Outputs
wire [31:0] M_R_Data;
wire [31:0] R_data_A;
wire [31:0] R_data_B;
wire [31:0] ALU_F;

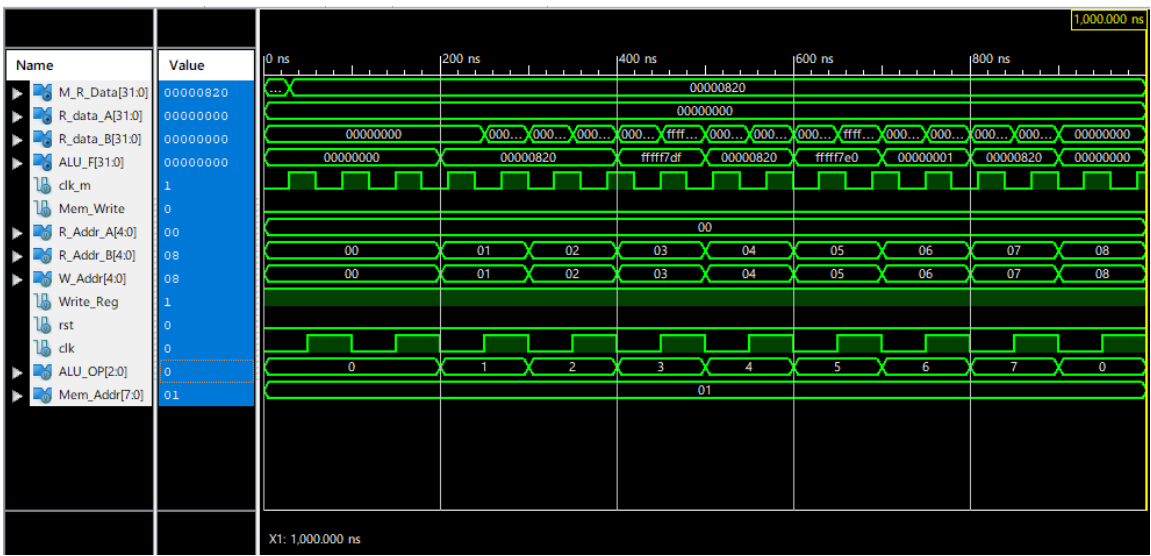
// Instantiate the Unit Under Test (UUT)
memory uut (
    .clk_m(clk_m),
    .Mem_Write(Mem_Write),
    .R_Addr_A(R_Addr_A),
    .R_Addr_B(R_Addr_B),
    .W_Addr(W_Addr),
    .Write_Reg(Write_Reg),
    .rst(rst),
    .clk(clk),
    .ALU_OP(ALU_OP),
    .Mem_Addr(Mem_Addr),
    .M_R_Data(M_R_Data),
    .R_data_A(R_data_A),
    .R_data_B(R_data_B),
    .ALU_F(ALU_F)
);
initial begin
    // Initialize Inputs
    clk_m = 0;
    Mem_Write = 0;
    R_Addr_A = 5'b000000;
    R_Addr_B = 5'b000000;
    W_Addr = 5'b000000;
    Write_Reg = 1;
    ALU_OP = 0;
    rst = 0;
    clk = 0;
    Mem_Addr = 6'b0000001;
    // Wait 100 ns for global reset to finish
    #100;

    // Add stimulus here
    repeat(8)
        begin

```

```
#100;
W_Addr = W_Addr + 1;
R_Addr_B = W_Addr;
ALU_OP = ALU_OP + 1;
end
end
always #50 clk=~clk;
always #30 clk_m=~clk_m;
endmodule
```

三、仿真波形



四、思考与探索

实现了存储器与前几次实验得到的 ALU 与 REG 模块的级联。因为对 Verilog 的不熟悉，级联的时候遇到了比较严重的问题，就是无法从寄存器堆中读取出数据，这个问题花费了较大力气才得到成功解决。这次实验使用的存储器使用 Memory IP 核直接创建，相比自己手动写，性能还是很好的，而且也很方便。