

寄存器堆部件实验

实验介绍

本实验将实现 CPU 中的寄存器堆（regfiles）部件，以便于以后 CPU 的设计

实验目标

1. 使用 logicsim 搭建拥有 16 个寄存器的 regfile 结构
2. 使用 verilog 实现 32 个寄存器的 regfile

实验原理

接口定义，可以直接复制到文件中

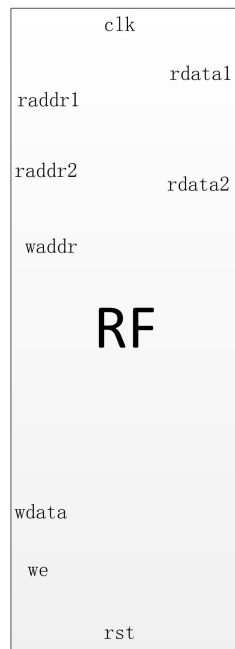
（请务必按照接口定义编写代码，在将来的实验中也是如此，模块名也请按照给出的定义命名）

1. regfiles 模块功能及接口定义

```
module regfiles(
    input clk,                //寄存器组时钟信号，下降沿写入数据（注意：pc 为上升沿，
                             //此为下降沿）
    input rst,                //reset 信号，reset 有效时全部寄存器置零
    input we,                 //写有效信号，we 有效时寄存器才能被写入
    input [4:0] raddr1,        //所需读取的寄存器的地址
    input [4:0] raddr2,        //所需读取的寄存器的地址
    input [4:0] waddr,         //写寄存器的地址
    input [31:0] wdata,        //写寄存器数据
    output [31:0] rdata1,      //raddr1 所对应寄存器的数据，只要有 raddr1 的输入即输出
                             //相应数据
    output [31:0] rdata2      //raddr2 所对应寄存器的数据，只要有 raddr2 的输入即输出
                             //相应数据

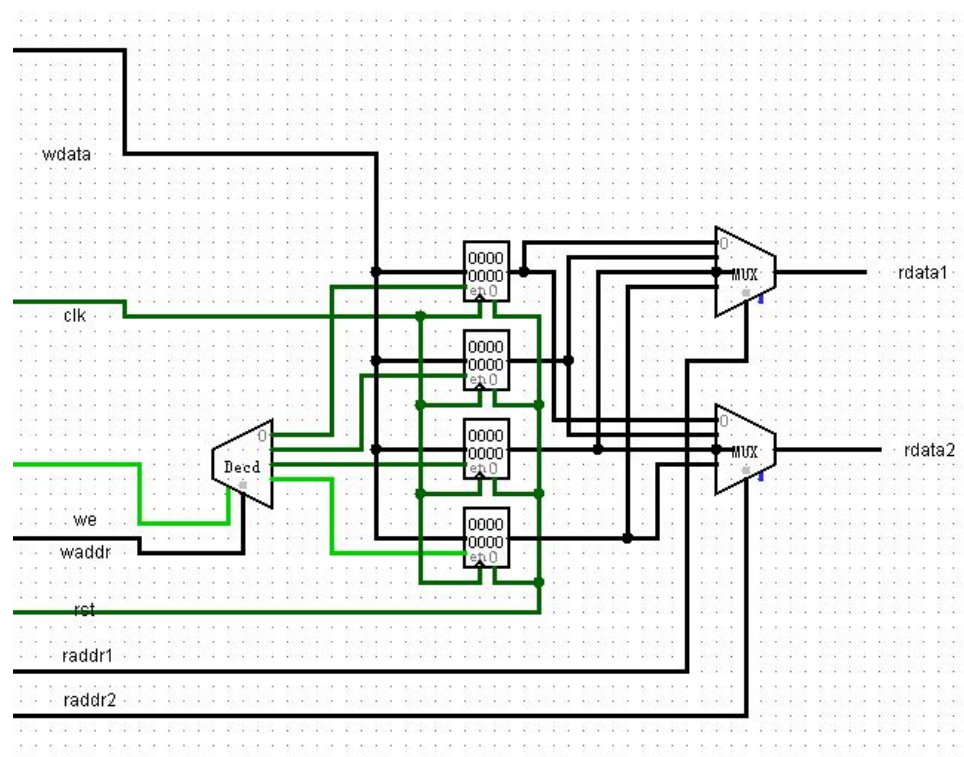
);
```

Ps:\$0 寄存器常为零且不可写入，可以使用接地实现



下图为拥有四个寄存器的 regfile 原理图，使用以前实验中的译码器，寄存器，以及选择器实现（需要根据具体情况作出更改），请根据此原理图，制作一个 16 个寄存器的 regfiles

- we 信号连入一个 2-4 译码器的使能端，使用 waddr 确定 we 信号被输入到哪个寄存器中
- clk 信号连接入每个寄存器，控制寄存器的读写
- rest 信号连接入每个寄存器
- 所有寄存器的输出连接到两个 4 选 1 选择器，使用两个输入 raddr1, raddr2 控制输入的值
- Logicsim 中有很多实用的部件，诸如 wdata 这样的多位输入可以使用多位的 pin 来实现，we rst 这样的 1 位输入可以使用 pin 来实现



提示: 可以使用 `reg` 数组来实现, 但希望大家使用寄存器, 译码器, 选择器的组合来实现(评分会考虑大家的实现方式)

实验步骤

1. 新建 logicsim 文件
2. 使用 logicsim 搭建 regfile
3. 新建 ISE 工程
4. 编写各个模块
5. 用 modelsim 仿真测试各模块