**实验七 寄存器堆的设计和实现**

**一、实验目的**

1、深入了解寄存器堆的结构和工作原理。

2、使用Verilog HDL语言来设计和实现寄存器堆结构，进行仿真和下载验证。

**二、实验设备**

1、装有vivado的计算机 1台

2、EGO1开发板 1块

**三、实验任务**

1、设计由16个4位寄存器构成的寄存器堆，该寄存器堆是双端口输出。进行功能仿真

2、设计由16个4位寄存器构成的寄存器堆，引脚绑定后进行下载，在EGO1开发板上进行数据验证。

**四、实验原理**

1、寄存器堆(regfiles)

一个寄存器是由n个触发器或锁存器按并行方式输入且并行方式输出连接而成。它只能记忆一个字,一个字的长度等于n个比特。当需要记忆多个字时，一个寄存器就不够用了,在这种情况下,需要使用由多个寄存器组成的寄存器堆。

图7.1所示为寄存器堆的逻辑结构与原理示意图,它由寄存器组﹑地址译码器、多路选择器MUX及多路分配器DMUX等部分组成。向寄存器写数据或读数据,必须先给出寄存器的地址编号。写数据时,控制信号WR有效,待写入的数据经DMUX送到地址给定的某个寄存器。读数据时,控制信号RD有效,由地址给定的某个寄存器的数据内容经多路开关MUX送出。由于读写工作是分时进行的,所以寄存器组在逻辑上能满足写数据或读数据的需要。

图7.2给出了由4个4位寄存器组成的具有两个数据输出端口的寄存器堆原理图,它可以同时从寄存器堆中取出两个数据﹐和加法器一起构成一个简单的运算通路。其主要由1个2-4译码器(ENB为使能端﹐S1,S2为两位编码输入端,D1～D4为译码输出端)、4个4位寄存器(ENB为使能端﹐A～D为数据输入端,Q1～Q4为数据输出端)和2个4位4选1数据选择器(ENB为使能端,S1～S4为4路数据输入端,C1 ,C2为选择控制端,D为数据输出端)组成。读数据时,读写控制信号we为高电平,由地址 raddr1和 raddr2指定的两个寄存器的数据分别送到rdata1和rdata2;写数据时,待存入的数据放到输入端wdata,并给出写地址 waddr,当读写控制信号we为低电平时, waddr指定的寄存器在时钟上升沿将数据写入到该寄存器。

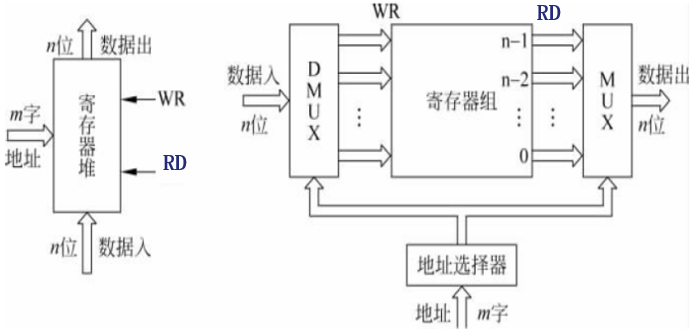


图7.1寄存器堆的逻辑结构

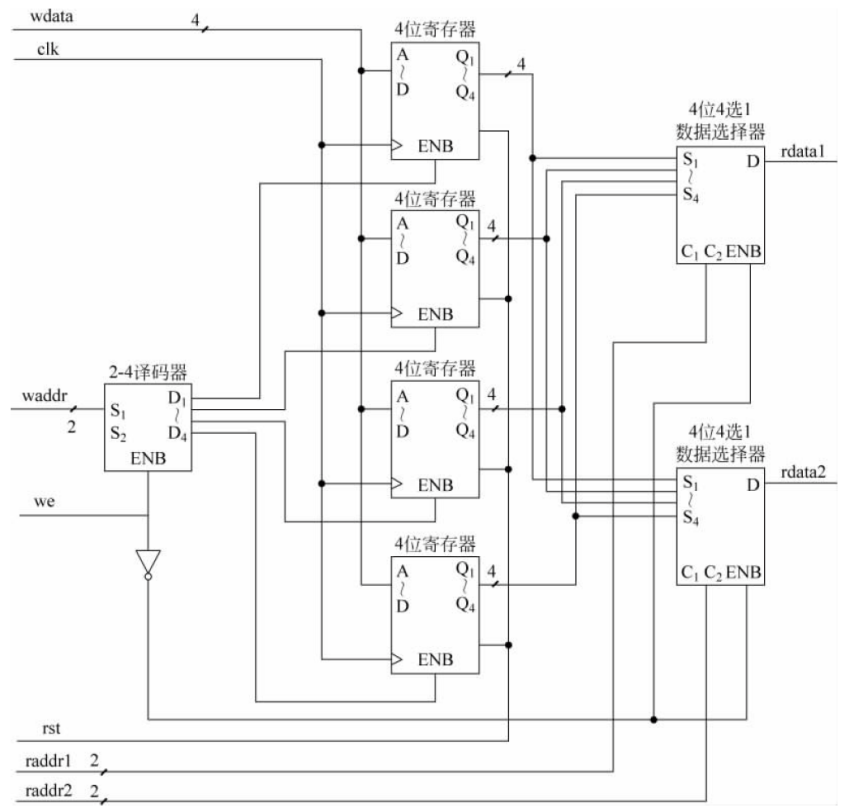


图7.2 4个4位寄存器构成的寄存器堆原理图

图7.3给出了本实验所要建模的寄存器堆的功能框图。

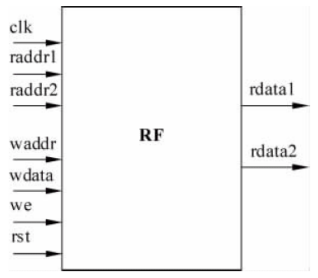


图7.3本实验所要建模的寄存器堆功能框图

由4个4位寄存器构成的寄存器堆，接口定义类似如下:

module Regfiles(

input clk, //寄存器组时钟信号，下降沿写入数据

input rst, //异步复位信号,高电平时全部寄存器置零

input we, //寄存器读写有效信号,低电平时允许寄存器写入数据，

//高电平时允许寄存器读出数据

input[1:0] raddr1, //所需读取的寄存器的地址

input[1:0] raddr2, //所需读取的寄存器的地址

input[1:0] waddr, //写寄存器的地址

input[3: 0] wdata, //写寄存器数据,数据在clk下降沿时被写入

output[3: 0] rdata1, //raddr1所对应寄存器的输出数据

output[3: 0] rdata2 //raddr2所对应寄存器的输出数据

);

注意: 实验图中要用到的译码器、寄存器﹐以及选择器的设计，都要求采用结构化的方法进行描述和实现，顶层模块通过子模块的实例化来设计和实现。

2-4译码器的电路如图7.4所示。在图7.4中，小圆圈代表非门，这是简化表示。图7.2中出现的带使能端的四选一数据选择器的电路如图7.5所示。

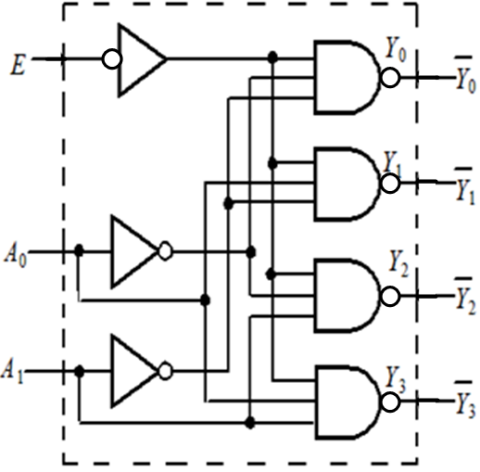
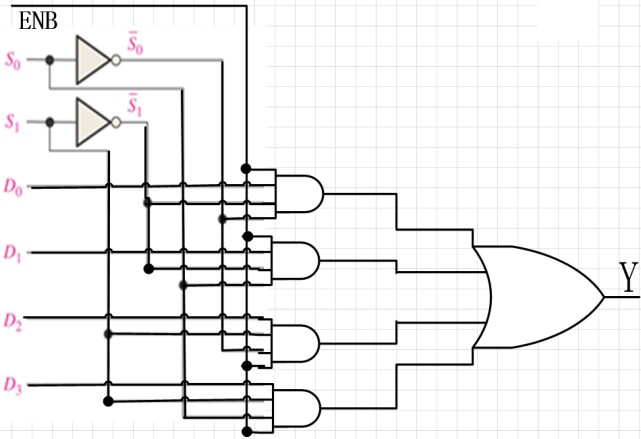
 

图7.4 2-4译码器结构 图7.5 带使能端的四选一数据选择器结构

**五、实验步骤**

**（一）设计含有16个4位寄存器的寄存器堆，该寄存器堆同时有两个地址输入raddr1和raddr2，对应的两个输出为rdata1和rdata2。**

1、**新建Vivado 工程**

**参照实验五的步骤创建新工程。**

参考上图7.2，由4个4位寄存器组成的寄存器堆电路原理图，编写译码器、寄存器﹐以及选择器等各个模块的模块文件。编写顶层文件，实现一个16\*4位的寄存器堆。可以同时有两个4位的输出rdata1和rdata2。

由两个4位寄存器构成的寄存器堆，接口定义类似如下:

module Regfiles(

input clk, //寄存器组时钟信号，下降沿写入数据

input rst, //异步复位信号,高电平时全部寄存器置零

input we, //寄存器读写有效信号,低电平时允许寄存器写入数据，

//高电平时允许寄存器读出数据

input[3:0] raddr1, //所需读取的寄存器的地址

input[3:0] raddr2, //所需读取的寄存器的地址

input[3:0] waddr, //写寄存器的地址

input[3: 0] wdata, //写寄存器数据,数据在clk下降沿时被写入

output[3: 0] rdata1, //raddr1所对应寄存器的输出数据

output[3: 0] rdata2 //raddr2所对应寄存器的输出数据

);

填写下面各个模块和顶层模块的verilog代码。

**译码器模块（可以采用一个4-16译码器，或者用多个2-4译码器来实现）**：

module decode(…… )

请在此处补充对应的verilog代码，要求在实验前完成。进实验室时会检查。验收时会围绕核心代码进行提问。

endmodule

**寄存器模块（16个4位的寄存器）：**

module regfile(…… )

请在此处补充对应的verilog代码，要求在实验前完成。进实验室时会检查。验收时会围绕核心代码进行提问。

endmodule

**选择器模块（可以采用多个4选1的数据选择器，或者16选1的数据选择器）：**

module select(…… )

请在此处补充对应的verilog代码，要求在实验前完成。进实验室时会检查。验收时会围绕核心代码进行提问。

Endmodule

在顶层模块top中，采用元件例化的方法，将上面3个模块的子模块例化后，进行引脚连接。要注意，顶层模块top中的输入引脚和输出引脚的定义和描述。

**顶层模块：**

module top(…… )

请在此处补充对应的verilog代码，要求在实验前完成。进实验室时会检查。验收时会围绕核心代码进行提问。

endmodule

**2、在Vivado中进行仿真，测试各模块工作状态情况。**

建立了顶层模块后，如果需要对顶层模块进行测试的话，就需要给顶层模块添加相应的输入，观察顶层模块的输出所具有的特点。例如：对地址为0000到1111单元依次赋值1111~0000，然后在读控制信号的作用下，读出地址为0011和1010这两个4位寄存器的值。观察地址信号和从该地址中读出的寄存器值是否正确。

激励输入模块：

module toptest(…… )

请在此处补充对应的verilog代码，要求在实验前完成。进实验室时会检查。验收时会围绕核心代码进行提问。

endmodule

进行仿真，验证仿真后的波形图是否正确。

在下面粘贴你的仿真截图，并分析结果的正确性。也可以将两个schematic原理图截图放入实验报告中。

**3、在EGO1开发板上进行引脚约束，然后进行电路生成和下载，通过按键输入对应的数值到对应的寄存器堆中，将寄存器的读出值送到发光的LED灯上。**

如果将输入输出信号与EGO1开发板上的外设进行绑定，对应绑定关系如下表7.1。下载后，在开发板上进行数据验证。对地址为0000到1111单元依次赋值1111~0000，然后在读控制信号的作用下，读出地址为0011和1010这两个4位寄存器的值。观察输出信号对应的LED灯的显示是否正确。

表7.1 输入输出信号与开发板上外设对应关系

|  |  |  |
| --- | --- | --- |
| 信号 | 引脚 | 说明 |
| raddr1[3:0] | sw3~sw0 | 用4个开关的值对应于读地址1 |
| raddr2[3:0] | sw7~sw4 | 用另外4个开关的值对应于读地址2 |
| waddr[3:0] | sw11~sw8 | 用4个开关的值对应于写地址 |
| wdata[3:0] | sw15~sw12 | 用另外4个开关的值对应于要写的数据 |
| rdata1[3:0] | led[3:0] | 用4个发光管的值对应于读出的数据1（也可以用多个七段数码管进行显示） |
| rdata2[3:0] | led[7:4] | 用另外4个发光管的值对应于读出的数据2（也可以用多个七段数码管进行显示） |
| we | s1 | 用通用按键s1（按下为1）对应于读写控制信号we |

下载后，拨动开关，观察LED灯亮灭情况，填写下表：

|  |  |  |
| --- | --- | --- |
| 地址 | LED灯对应的数值 | 说明 |
| 0011 |  |  |
| 0110 |  |  |
| 1000 |  |  |
| 1010 |  |  |

**六、实验验收要求和思考**

1、可以使用实验室的电脑进行验收。也可以自己带电脑到实验室，在自己的电脑上完成实验，然后在自己的电脑上进行验收。验收时，要求能简单介绍一下系统能实现的主要功能，核心代码的含义和实验中引脚的分配等等。老师会简单提问，如果回答问题较差，会酌情扣分。所以要求对代码进行提前预习和准备。

2、实验七的验收规则为：只验收到实验步骤2，80分。验收到步骤3，100分。如果当次实验课没有验收完的，可以在其他班的实验课时进行补验收。或者下次进行实验八时补验收实验七。

3、如果采用行为级描述方式进行16个32位的寄存器堆电路设计，对应的verilog代码应该是怎样的？

4、下载后，进行寄存器存储值读出时，寄存器存储的值还可以送到EGO1开发板上的哪里进行显示？如果读出的存储值送到七段显示器上进行显示，verilog代码应该怎么写？