**实验六 寄存器堆实验**

**一、实验目的**

1、深入了解寄存器堆的原理。

2、学习使用Verilog HDL设计实现寄存器堆。

**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。

2、设计由16个32位寄存器构成的寄存器堆，该寄存器堆是双端口输出。

**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6.1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6.2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。

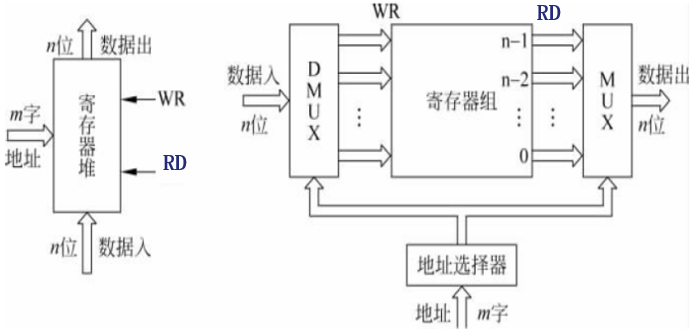


图6.1寄存器堆的逻辑结构

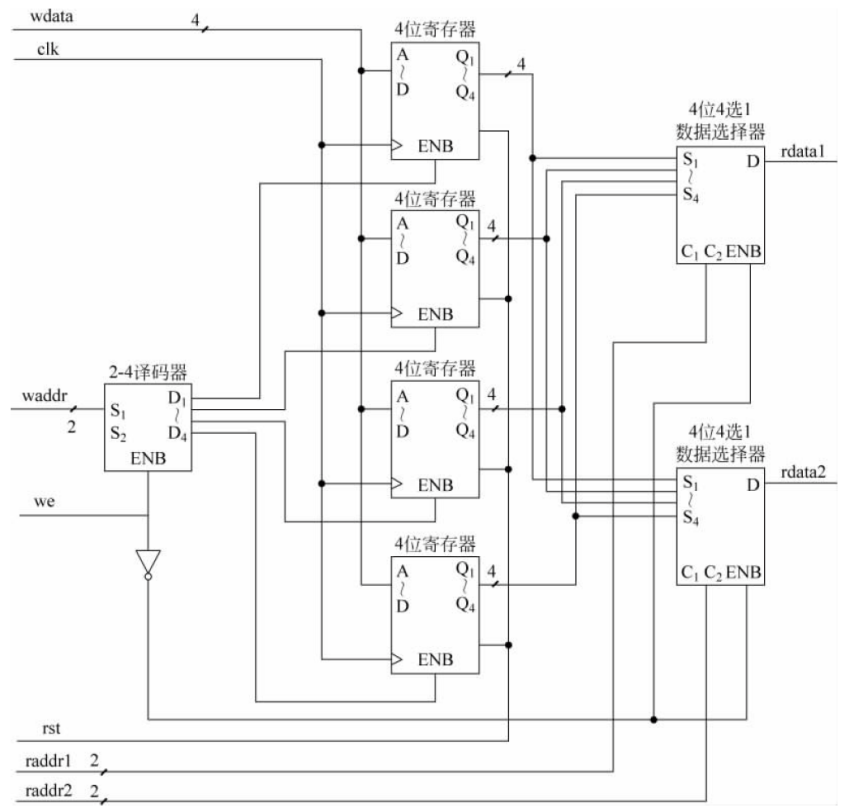


图6.2由4个4位寄存器构成的寄存器堆原理图

图6.3给出了本实验所要建模的寄存器堆的功能框图。

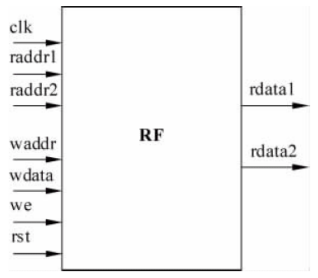


图6.3本实验所要建模的寄存器堆功能框图

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6.4所示。在图6.4中，小圆圈代表非门，这是简化表示。带使能端的四选一数据选择器的电路如图6.5所示。

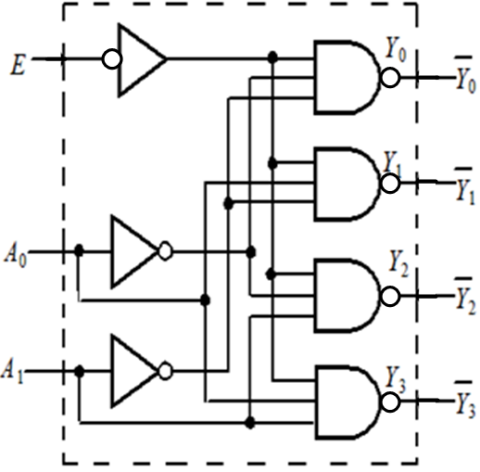
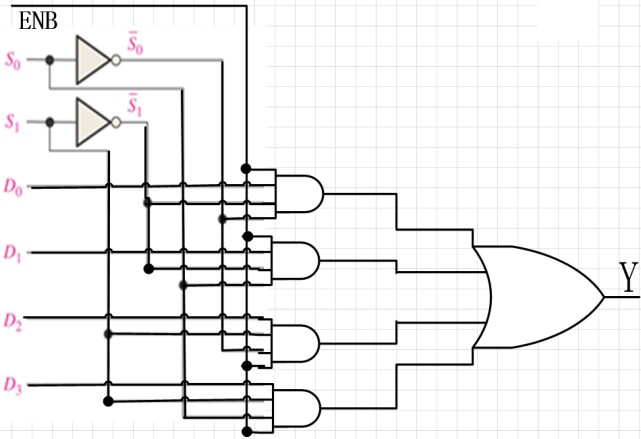
 

图6.4 图6.5

**五、实验步骤**

**（一）设计含有16个4位寄存器的寄存器堆，可以同时有两个4位的输出rdata1和rdata2（实验室完成和进行仿真验收）**

1、**新建Vivado 工程**

**参照实验五的步骤创建新工程。**

参考上图6.2，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16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，也可以用多个2-4译码器）**：

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（2组，每组4个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、用Vivado中进行仿真测试各模块。**

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。观察波形是否正确。

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。

**（二）设计16个32位的寄存器堆（课后完成，不验收，不需要写入实验报告）**

1、**新建Vivado 工程**

**参照实验五的步骤创建新工程。**

参考上图6.2，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。编程顶层文件，实现一个16\*32位的寄存器堆。在16\*32位的寄存器堆中，首先要考虑清楚译码器需要几位输入，才能唯一选中16个寄存器中的一个。然后要考虑输出端的数据选择器需要几位输入，才能从16个寄存器中选中一个输出。

填写下面各个模块的verilog代码。此部分也可以采用行为级描述方式进行verilog代码编写。

**译码器模块：**

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：**

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：**

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、用Vivado中进行仿真测试各模块。**

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。

**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。

**六、实验思考：**

1、如果采用行为级描述方式进行电路设计，对应的verilog代码应该是怎样的？

2、如果进行下载的话，EGO1开发板上的引脚应该怎样绑定，输出可以送到哪里进行显示？