

Lab3 RAM存储器的使用

实验目的

- 学习使用ISE的Ip核
- 学习使用Xilinx FPGA的RAM资源
 - 例化一个简单双端口RAM
 - 使用coe文件对RAM进行初始化

实验内容

利用coe文件把RAM的前两个单元初始化为2

coe文件截图如下：

```
1 MEMORY_INITIALIZATION_RADIX=10;  
2 MEMORY_INITIALIZATION_VECTOR=2,2;
```

初始化或者复位之后的开始阶段，从RAM[0]和RAM[1]中读出数放到REG_FILE[0]和REG_FILR[1]中，但是这里比较坑的是要注意的是从RAM中读数据出来会有一个周期的延时，所以初始化需要三个周期才能将两个数写入到REG_FILE中。

REG_FILE模块和ALU模块与前两次实验相同，不再叙述。

本次实验难点在控制模块control，控制模块输出REG_FILE和RAM的读写地址和使能信号

top模块例化上面的四个模块，此外，在top模块中设有两个选择器，选择写入REG_FILE中的数据值是来自RAM还是ALU，前三周期来自RAM,之后来自ALU，以及根据control的输出设置RAM的写使能信号。

代码和实验结果截图

control模块

```
21 module control(  
22     input clk,  
23     input rst_n,  
24     output reg [5:0] RAddr1,  
25     output reg [5:0] RAddr2,  
26     output reg [5:0] WAddr,  
27     output reg ctr,  
28     output reg [5:0] init_control  
29 );  
30  
31 reg [5:0] Addr=0;  
32 reg c_control=0;  
33 reg [5:0] init=0;  
34  
35 always@(posedge clk,negedge rst_n)  
36 begin  
37     if(~rst_n)  
38         init<=0;  
39     else if (init<3) begin  
40         | init<=init+1; //when init accumulate to 3 ,then init keep as 3 until rese  
41     end
```

```
43  
44 always@(*)  
45 begin  
46     if(init==3)begin  
47         RAddr1 = Addr; //now Addr==0 , after this time , alu read from REG_FILE and c  
48         RAddr2 = Addr+1;  
49         WAddr = Addr+2;  
50         ctr = c_control;  
51     end  
52     else begin  
53         RAddr1 = init;  
54         RAddr2 = Addr;  
55         WAddr = init-1;  
56         ctr = 1;  
57     end  
58 end  
59  
60 always@(*)begin  
61     | init_control = init;  
62 end  
63
```

```

64 always@(posedge clk,negedge rst_n)begin
65     if(~rst_n)
66         Addr<=0;
67     else if (init<3) begin
68         Addr<=0;
69     end
70     else if (Addr<=5'd60&&c_control==1) begin
71         Addr<=Addr+1;
72     end
73 end
74
75 always@(posedge clk,negedge rst_n)
76 begin
77     if(~rst_n)
78         c_control<=0;
79     else if(init==1)
80         c_control<=1;
81     else if (Addr==5'd61) begin
82         c_control<=0;
83     end
84 end
85
86 endmodule

```

top模块

```

21 module top(
22     input clk,
23     input rst_n,
24     output reg [31:0] tmp
25 );
26
27 wire [31:0] alu_out,RE_OUT1,RE_OUT2,RAM_OUT;
28 wire [5:0] WAddr,RAddr1,RAddr2;
29 wire ctr;
30 wire [5:0] where;
31 //reg [31:0] tmp=0;
32 reg we=0;
33 ALU myALU(RE_OUT1,RE_OUT2,5'h01,alu_out);
34 REG_FILE myREG_FILE(clk,rst_n,RAddr1,RE_OUT1,RAddr2,RE_OUT2,WAddr,tmp,ctr);
35 RAM myRAM(clk,1,we,WAddr,alu_out,clk,1,RAddr1,RAM_OUT);
36 control mycontrol(clk,rst_n,RAddr1,RAddr2,WAddr,ctr,where);
37

```

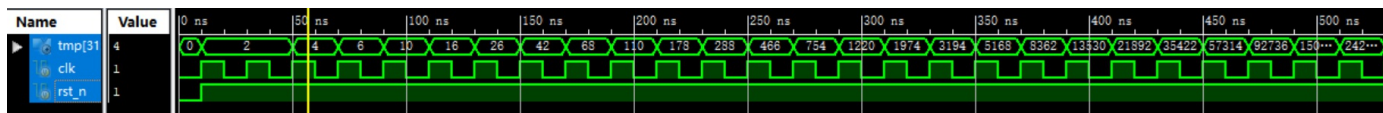
```

38 always@(*)
39 begin
40     if(where<3)
41         tmp = RAM_OUT;
42     else begin
43         tmp = alu_out;
44     end
45 end
46
47 always@(*)
48 begin
49     if(where<3)
50         we = 0;
51     else if(RAddr1 < 61)
52         we = 1;
53 end
54 endmodule

```

仿真结果截图

波形



REG_FILE内容

RAM内容men