FPGA第二次作业

第21组：**门钰涵(主要)**、谷世昌、张浩远、郭羿呈

18050100196

1. 大致思路

主要是通过总线控制给寄存器写值，通过控制FSMC\_NOE发命令R1和R2数据交换，总线是片选/写信号/地址/数据。

1. 代码
2. 源代码

module CPU\_BUS (

RESET,

FSMC\_ADD,

FSMC\_nCS,

FSMC\_NOE,

FSMC\_NWE,

FSMC\_DATAIN,

CLK\_IN,

R1,

R2,

R3

);

input RESET; wire RESET;

input [3:0] FSMC\_ADD; wire[3:0] FSMC\_ADD;

input FSMC\_nCS; wire FSMC\_nCS;

input [1:0] FSMC\_NOE; wire[1:0] FSMC\_NOE;

input FSMC\_NWE; wire FSMC\_NWE;

input [7:0] FSMC\_DATAIN; wire [7:0] FSMC\_DATAIN;

input CLK\_IN; wire CLK\_IN;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

wire [7:0] DATA\_R1/\* synthesis keep="1" \*/;

wire [7:0] DATA\_R2/\* synthesis keep="1" \*/;

wire [7:0] DATA\_R3/\* synthesis keep="1" \*/;

wire [7:0] DATAOUT;

output [7:0] R1; reg[7:0] R1/\* synthesis keep="1" \*/;

output [7:0] R2; reg[7:0] R2/\* synthesis keep="1" \*/;

output [7:0] R3; reg[7:0] R3/\* synthesis keep="1" \*/;

reg [3:0] Num;

//总线写入数据接口

RegCPUData Reg\_1(RESET,CLK\_IN,FSMC\_nCS,FSMC\_NWE,4'h0,FSMC\_ADD, FSMC\_DATAIN,DATA\_R1);

RegCPUData Reg\_2(RESET,CLK\_IN,FSMC\_nCS,FSMC\_NWE,4'h1,FSMC\_ADD, FSMC\_DATAIN,DATA\_R2);

RegCPUData Reg\_3(RESET,CLK\_IN,FSMC\_nCS,FSMC\_NWE,4'h2,FSMC\_ADD, FSMC\_DATAIN,DATA\_R3);

always @(posedge CLK\_IN or negedge RESET)

begin

if(!RESET)

begin

R1<=0;

R2<=0;

R3<=0;

Num<=0;

end

else

begin

//将R3作为中间暂存，把R1和R2数据交换

if(FSMC\_NOE==2'b01)

begin

if(Num==0) begin R3<=R2; Num<=1; end

else if(Num==1) begin R2<=R1; Num<=2; end

else begin R1<=R3; Num<=3; end

end

else

begin

//得到的数据写入三个寄存器

R1<=DATA\_R1;

R2<=DATA\_R2;

R3<=DATA\_R3;

end

end

end

endmodule

module RegCPUData(Reset,Clk,CS,WR,SET\_ADD,ADD,DATAIN,DATAOUT);

input Reset; wire Reset;

input Clk; wire Clk;

input CS; wire CS;

input WR; wire WR;

input [3:0] SET\_ADD; wire [3:0] SET\_ADD;

input [3:0] ADD; wire [3:0] ADD;

input [7:0] DATAIN; wire [7:0] DATAIN;

output [7:0] DATAOUT; reg [7:0] DATAOUT;

always @(posedge Clk or negedge Reset)

begin

if(!Reset)

DATAOUT<=8'd0; //1 input 0 output

else if (!CS && !WR && (SET\_ADD == ADD))

DATAOUT<=DATAIN;

end

endmodule

1. 仿真代码

`timescale 1ns / 1ps

module CPU\_BUS\_tb;

//Input

reg RESET;

reg [3:0] FSMC\_ADD;

reg FSMC\_nCS;

reg [1:0] FSMC\_NOE;

reg FSMC\_NWE;

reg [7:0] FSMC\_DATAIN;

reg CLK\_IN;

//实例化被测试单元

CPU\_BUS uut (

.RESET(RESET),

.FSMC\_ADD(FSMC\_ADD),

.FSMC\_nCS(FSMC\_nCS),

.FSMC\_NOE(FSMC\_NOE),

.FSMC\_NWE(FSMC\_NWE),

.FSMC\_DATAIN(FSMC\_DATAIN),

.CLK\_IN(CLK\_IN)

);

initial

begin

//初始化input

RESET = 0;

FSMC\_ADD = 0;

FSMC\_nCS = 0;

FSMC\_NOE = 0;

FSMC\_NWE = 0;

FSMC\_DATAIN = 0;

CLK\_IN = 0;

//等待100秒以完成全局Reset

#100;

RESET = 1;

end

always

begin

#10;CLK\_IN=0;

#10;CLK\_IN=1;

end

reg[31:0] Num;

reg[31:0] Ctr;

always @(posedge CLK\_IN or negedge RESET)

begin

if(!RESET)

begin

FSMC\_ADD <= 0;

FSMC\_nCS <= 0;

FSMC\_NOE <= 0;

FSMC\_NWE <= 0;

FSMC\_DATAIN <= 0;

Num<=0;

Ctr<=0;

end

else

begin

case (Num)

0:begin

FSMC\_ADD <= 0;//总线向R1寄存器写数据数据大小为AB

FSMC\_nCS <= 0;

FSMC\_NOE <= 0;

FSMC\_NWE <= 0;

FSMC\_DATAIN <= 8'hAB;

Num<=1;

Ctr<=0;

end

1:begin

FSMC\_ADD <= 0;

FSMC\_nCS <= 1;

Num<=2;

end

2:begin

FSMC\_ADD <= 1;//总线向R2寄存器写数据数据大小为AC

FSMC\_nCS <= 0;

FSMC\_NOE <= 0;

FSMC\_NWE <= 0;

FSMC\_DATAIN <= 8'hAC;

Num<=3;

end

3:begin

FSMC\_ADD <= 0;

FSMC\_nCS <= 1;

Num<=4;

end

4:begin

FSMC\_ADD <= 2;//总线向R3寄存器写数据数据大小为AC

FSMC\_nCS <= 0;

FSMC\_NOE <= 0;

FSMC\_NWE <= 0;

FSMC\_DATAIN <= 8'hAD;

Num<=5;

end

5:begin //发命令让R1和R2数据交换

FSMC\_ADD <= 0;

FSMC\_nCS <= 1;

Ctr<=Ctr+1;

if(Ctr>20)

Num<=6;

end

6,7,8,9:begin

FSMC\_NOE <= 2'b01;

Num<=Num+1;

end

10:begin

FSMC\_NOE <= 2'b01;

end

endcase

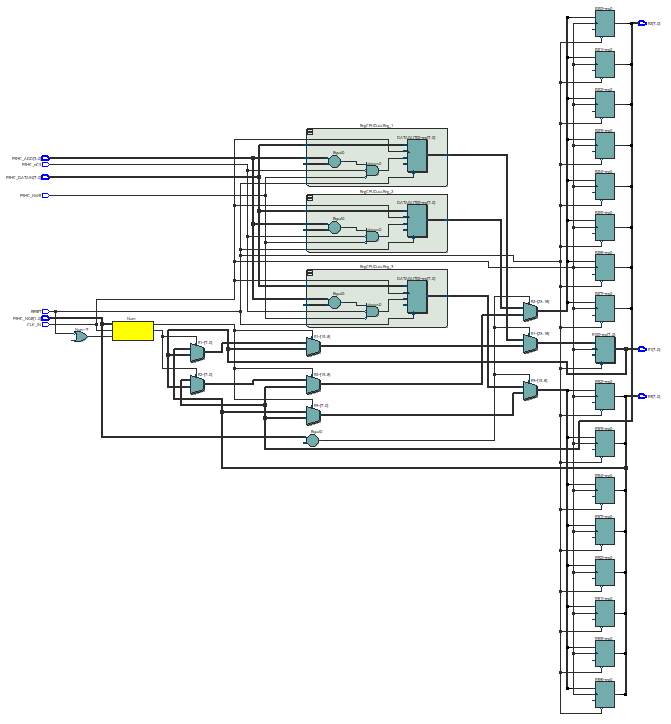
end

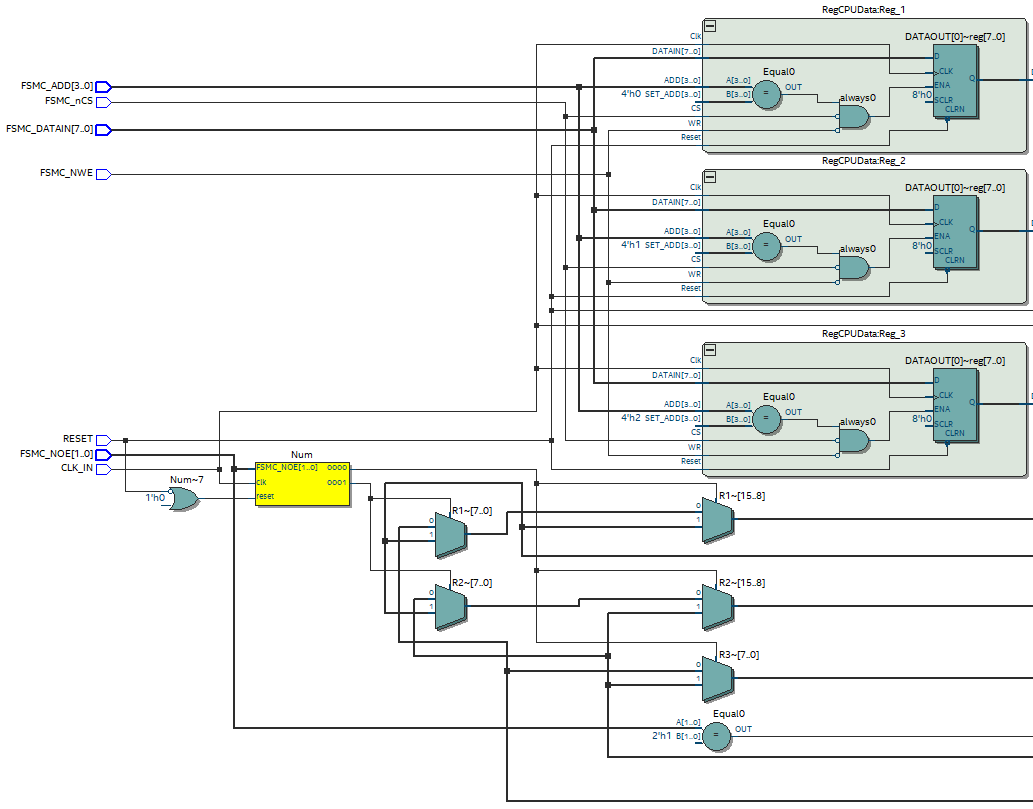
end

endmodule

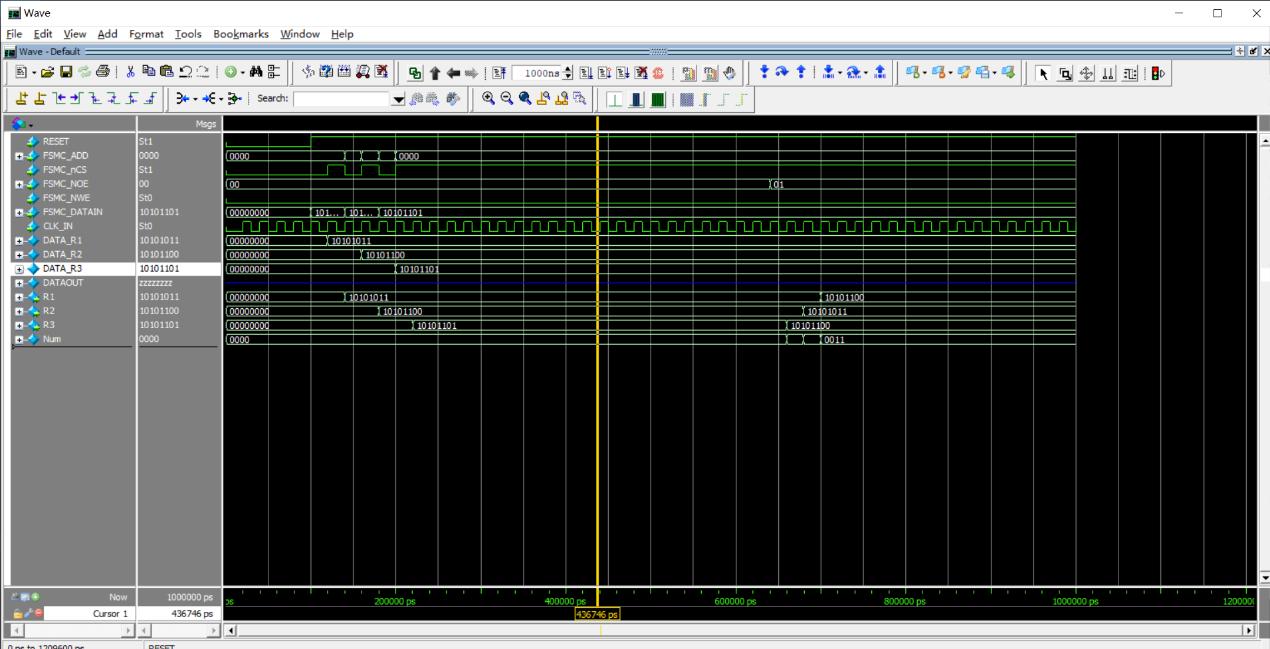
1. RTL图

后图为前图的局部放大。





1. 仿真图





红框1处为写入的数据，红框2处为R3作为暂存，来交换R1和R2的数据。