

同济大学计算机系

数字逻辑课程实验报告



学 号 1652286

姓 名 李辉

专 业 计算机

授课老师 张冬冬

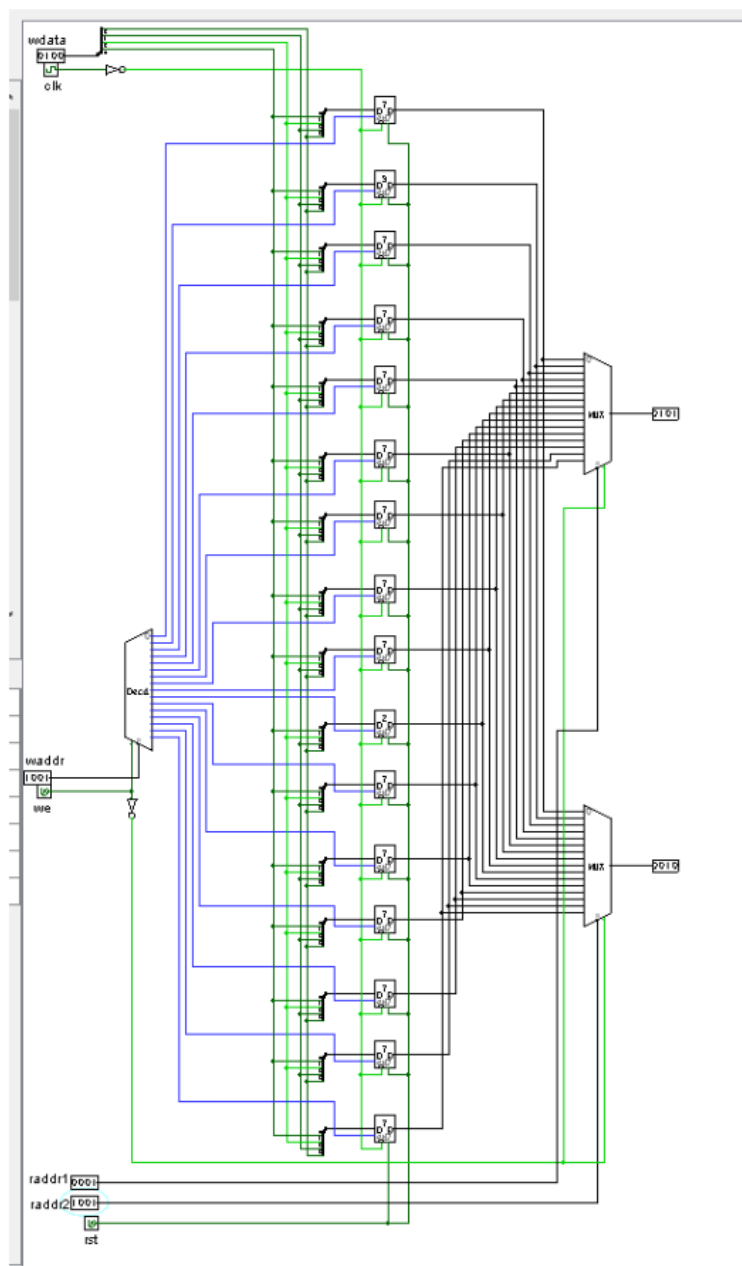
一、实验内容

1、实现同步模 8 计数器（计数器需要通过 logisim 验证逻辑，并通过实例化分频器、七段译码显示管、JK 触发器后下板验证，使数字译码管通过数码管显示）；

二、硬件逻辑图

（实验步骤中要求用 logisim 画图的实验，在该部分给出 logisim 原理图，否则该部分在实验报告中不用写）

6.8_3 16 个 4 位寄存器组成的寄存器堆



三、模块建模

（该部分要求对实验中建模的所有模块进行功能描述，并列出具体的模块建模的 verilog 代码）

(1)6.8_1 ram 半导体随机读写存储器（单线）

```
`timescale 1ns / 1ns
module ram(
    input clk,//存储器时钟信号，上升沿时向 ram 内部写入数据
    input ena,//存储器有效信号，高电平时存储器才能运行，否则输出 z
    input wena,//存储器读写有效信号，高电平时为写有效，低电平时为读有效
        //与 ena 同时有效时才能对存储器进行读写
    input [4:0] addr,//输入地址，指定数据读写的地址
    input [31:0] data_in,//存储器写入的数据，在 clk 上升沿时被写入
    output [31:0] data_out//存储器读出的数据
);
    reg [31:0] out;
    reg [31:0] mem[31:0];//存储器 1024bit
    always@(*)
    begin
        if(~ena)//ena=0
            out={32{1'bz}};
        else if(~wena)//wena=0，只读
            out=mem[addr];
        end
    always@(posedge clk)
    begin
        if(wena==1&&ena==1)
            mem[addr]=data_in;
        end
        assign data_out=out;
    endmodule
```

(2)6.8_2 ram2 半导体随机读写存储器（双线）

```
`timescale 1ns / 1ns
module ram2(
    input clk,
    input ena,
    input wena,
    input [4:0] addr,
    inout [31:0] data
);
    reg [31:0] out;
```

```

reg [31:0]mem[0:31];
always@(posedge clk)
begin
if(ena&&wena)
begin
mem[addr]=data;
end
end
always@(*)
begin
if(~ena)
begin
out=32'bz;
end
else
begin
if(~wena)
begin
out=mem[addr];
end
end
end
assign data=(wena==0)?out:{ 32{ 1'bz } };
endmodule

```

(3)6.8_3 Regfiles 寄存器堆(32 位)

顶层模块

```

`timescale 1ns / 1ns
module Regfiles(
    input clk,
    input rst,
    input we,
    input [4:0] raddr1,
    input [4:0] raddr2,
    input [4:0] waddr,
    input [31:0] wdata,
    output [31:0] rdata1,
    output [31:0] rdata2
);
    wire [31:0]datA;
    wire [1023:0]meM;
    reg [31:0]data;

```

```

reg [1023:0]mem;
always@(*)
begin
data=datA;
mem=meM;
end
decoder5_32 decoder(.iData(waddr),.iEna(we),.oData(datA));
generate
    genvar i;
    for(i=0;i<32;i=i+1)
        begin:queues
            pcreg Reg(
                .clk(clk),.rst(rst),.ena(data[i]),.data_in(wdata),.data_out(meM[32*i+31:32*i])
            );
        end
    endgenerate
selector32_1 sel1(
    .ena(~we),.iS(raddr1),.iC(mem),.oZ(rdata1)
);
selector32_1 sel2(
    .ena(~we),.iS(raddr2),.iC(mem),.oZ(rdata2)
);
endmodule

*****
Decoder5_32 5-32 译码器
*****

`timescale 1ns / 1ns
module decoder5_32(//五位地址译码 32 位，用
    input [4:0] iData,
    input iEna,//高电平有效
    output [31:0] oData
);
    reg [31:0] odata;
    always @(*)
    begin
        if(iEna==1)
        begin
            odata=0;
            odata[iData]=1;
        end
    end
    assign oData=odata;
Endmodule

```

pcreg pc 寄存器

```
`timescale 1ns / 1ns
module pcreg(
    input clk,    //1 位输入，寄存器时钟信号，上升沿时为 PC 寄存器赋值
    input rst,    //1 位输入，异步重置信号，高电平时将 PC 寄存器清零 //注：当 ena 信号无效时，rst 也可以重置寄存器
    input ena,    //1 位输入,有效信号高电平时 PC 寄存器读入 data_in //的值，否则保持原有输出
    input [31:0] data_in, //32 位输入，输入数据将被存入寄存器内部
    output [31:0] data_out //32 位输出，工作时始终输出 PC //寄存器内部存储的值
);
wire Q2;
generate
    genvar i;
    for(i=0;i<32;i=i+1)//允许产生 module 和 primitive 的多个实例化
    begin:thirty_two
        Asynchronous_D_FF every(clk&ena,data_in[i],~rst,data_out[i],Q2);
    end
endgenerate
endmoduleendmodule
```

Asynchronous_D_FF pcreg 的底层 module

```
`timescale 1ns / 1ns
module Asynchronous_D_FF(
    input CLK,
    input D,
    input RST_n,
    output reg Q1,
    output reg Q2
);
always@(negedge CLK or negedge RST_n)
begin
    if(RST_n==0)
    begin
        Q1<=0;
        Q2<=1;
    end
    else
    begin
        Q1<=D;
    end
end
end
```

```

        Q2<=~D;
    end
end
endmodule    endgenerate
endmoduleendmodule

*****

Selector32_1  32 选一数据选择器（32 位）
*****

`timescale 1ns / 1ns
module selector32_1(//32 选一（32 位宽）
    input ena,//使能端，高电平有效
    input [4:0] iS,//选择信号
    input [1023:0] iC,//输入信号 1024 位
    output reg [31:0] oZ//选择输出的信号
);
    integer i,j;
    always@(*)
    begin
        if(ena==0)
            oZ=32'bx;
        else
            begin
                i=iS*32;
                j=0;
                while(j<=31)
                    begin
                        oZ[j]=iC[i];
                        i=i+1;
                        j=j+1;
                    end
            end
        end
    end
endmoduleee

```

四、测试模块建模

（要求列写各建模模块的 test bench 模块代码）

(1)6.8_1 ram 半导体随机读写存储器（单线）

```

`timescale 1ns / 1ns
module ram_tb();
    reg clk;
    reg ena;
    reg wena;

```

```
reg [4:0]addr;
reg [31:0]data_in;
wire [31:0]data_out;
ram uut(clk,ena,wena,addr,data_in,data_out);
```

```
always
begin
#5 clk=~clk;
end
```

```
integer i;
initial
begin
clk=0;
ena=1;
wena=0;//读测试
addr=0;//从 0 开始取
for(i=0;i<=31;i=i+1)
begin
#6 addr=addr+1;
end
```

```
wena=1;//写测试
addr=0;
data_in={ 32{ 1'b1 } };
```

```
#60
for(i=0;i<=31;i=i+1)
begin
#6
data_in=data_in-1;
addr=addr+1;
end
//下面把写进去的读出来
#6
wena=0;
addr=0;
for(i=0;i<=31;i=i+1)
begin
#6
addr=addr+1;//逐个读出 data_out
end
//下面进行 ena 测试
#6
```



```

ena=0;
#100 ena=1;
end
endmodule

```

(2)6.8_2 ram2 半导体随机读写存储器（双线）

```

`timescale 1ns / 1ns
module ram_tb();
reg clk;
reg ena;
reg wena;
reg [4:0]addr;
wire [31:0]data;
reg [31:0] data_in;
wire [31:0] data_out;
assign data=(wena==1)?data_in:{32{1'bz}};
assign data_out=(wena==0)?data:{32{1'bz}};
ram2 uut(clk,ena,wena,addr,data);
always
begin
#5 clk=~clk;
end

```

```

integer i;
initial
begin
clk=0;
ena=1;
wena=0;//读测试
addr=0;//从 0 开始取
data_in=0;
for(i=0;i<=31;i=i+1)
begin
#6 addr=addr+1;
end

```

```

wena=1;//写测试
addr=0;
data_in={32{1'b1}};

```

```

#60
for(i=0;i<=31;i=i+1)
begin
#6
data_in=data_in-1;

```

```

addr=addr+1;
end
//下面把写进去的读出来
#6
wena=0;
addr=0;
for(i=0;i<=31;i=i+1)
begin
#6
addr=addr+1;//逐个读出 data_out
end
//下面进行 ena 测试
#6
ena=0;
#100 ena=1;
end
endmodule

```

(3)6.8_3 Regfiles 寄存器堆(32 位)

```

`timescale 1ns / 1ns
module Regfiles_tb();
reg clk;
reg rst;
reg we;
reg [4:0] raddr1;
reg [4:0] raddr2;
reg [4:0] waddr;
reg [31:0] wdata;
wire [31:0] rdata1;
wire [31:0] rdata2;

Regfiles uut(
.clk(clk),.rst(rst),.we(we),.raddr1(raddr1),
.raddr2(raddr2),.waddr(waddr),.wdata(wdata),
.rdata1(rdata1),.rdata2(rdata2)
);
always
begin
#5 clk=~clk;
end
initial
begin
clk=1;
raddr1=5'b00000;

```

```
raddr2=5'b11111;
rst=0;
//写测试
#6
we=1;
waddr=5'b11001;
wdata={{16{1'b1}}, {8{1'b0}}, {8{1'b1}}};
#10
waddr=5'b00111;
wdata={{8{1'b1}}, {8{1'b0}}, {8{1'b1}}, {8{1'b0}}};
//读测试

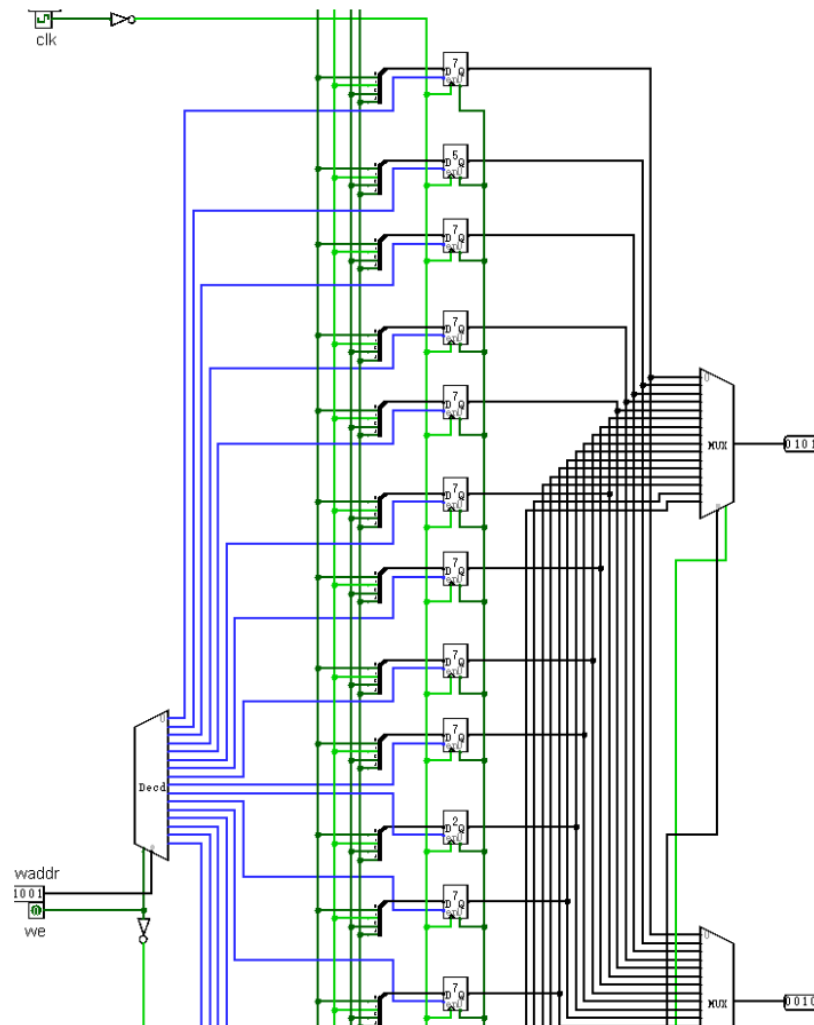
#28
we=0;
raddr1=5'b11001;
raddr2=5'b00111;
//rst 测试
#12
rst=1;
raddr1=5'b11001;
raddr2=5'b00111;
end
endmodule
```

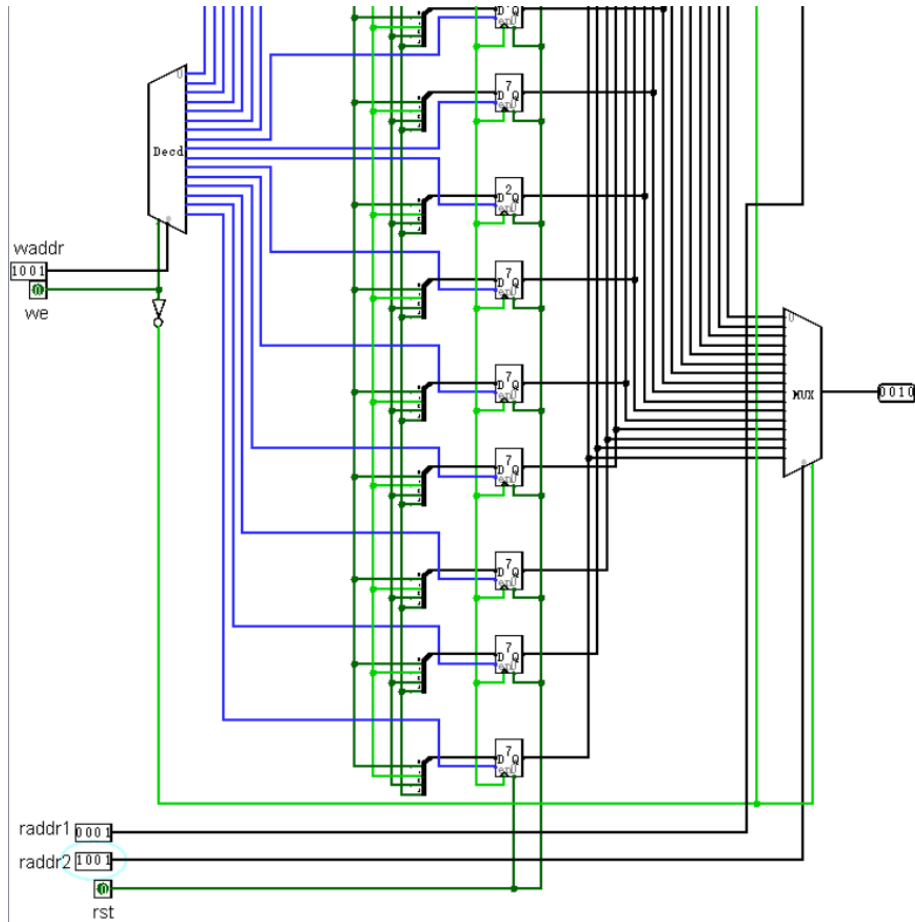
五、实验结果

（该部分可截图说明，要求 logisim 逻辑验证图、modelsim 仿真波形图、以及下板后的实验结果贴图（实验步骤中没有下板要求的实验，不需要下板贴图））

logisim 逻辑验证图

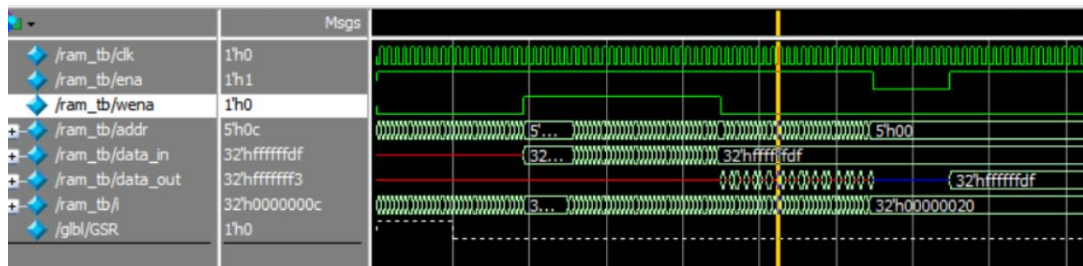
6.8_3 Regfiles 16 个 4 位寄存器组成的寄存器堆



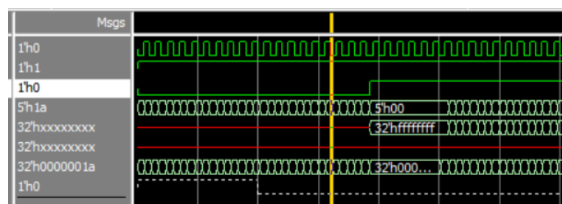


Modelsim 仿真波形图

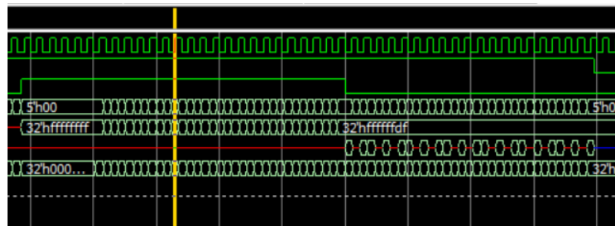
(1)6.8_1 ram 半导体随机读写存储器（单线）



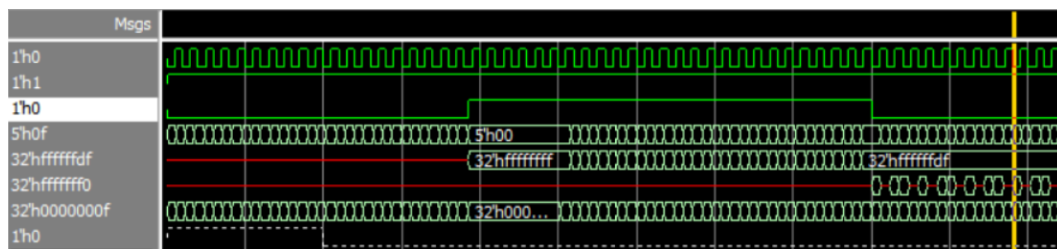
1、读测试



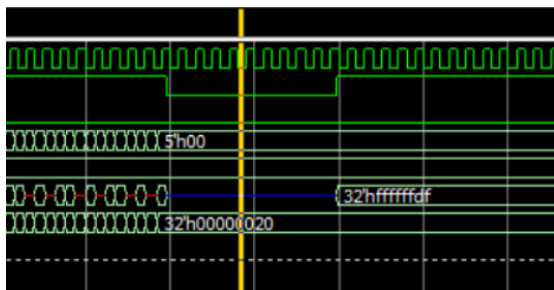
2、写测试



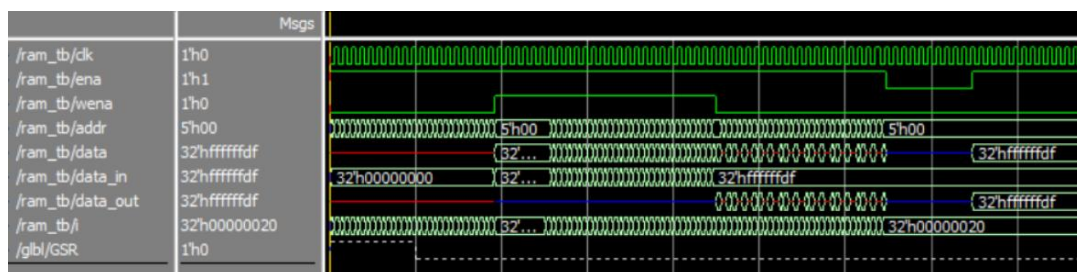
3、写测试读出



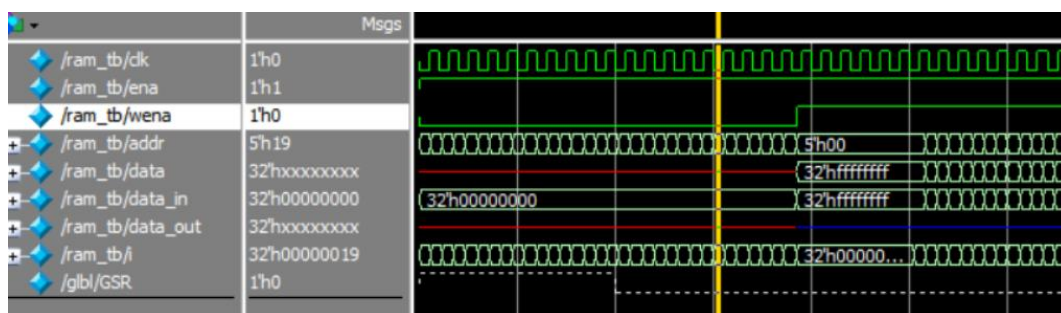
4、测试 ena 清零



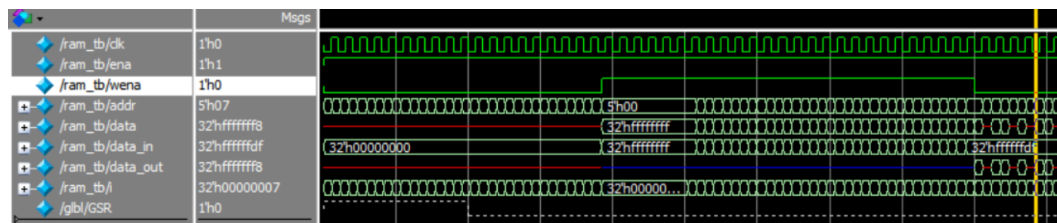
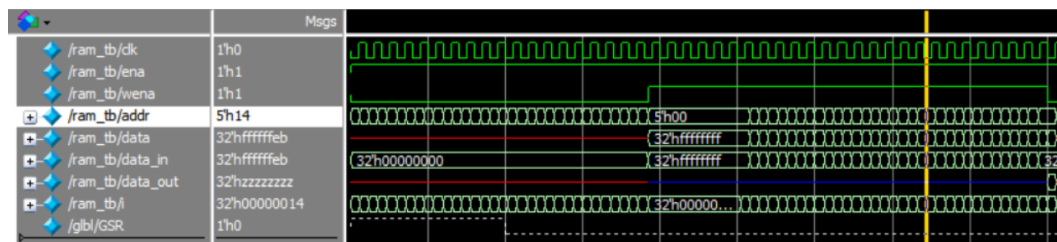
(2)6.8_2 ram2 半导体随机读写存储器（双线）



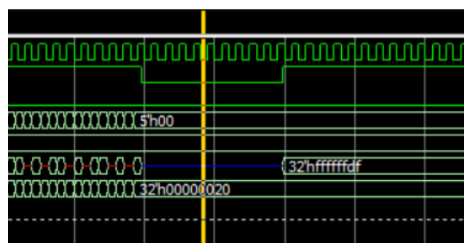
1、输出测试



2、输入测试

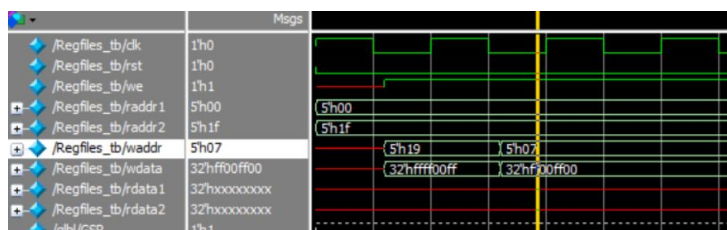
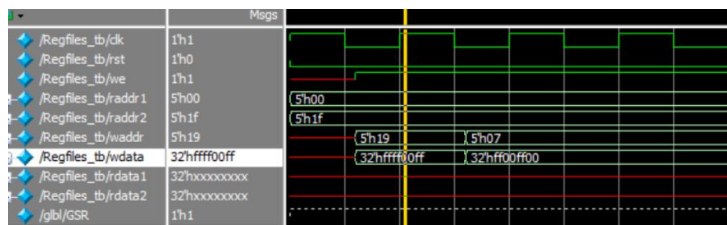


3、测试 ena 清零

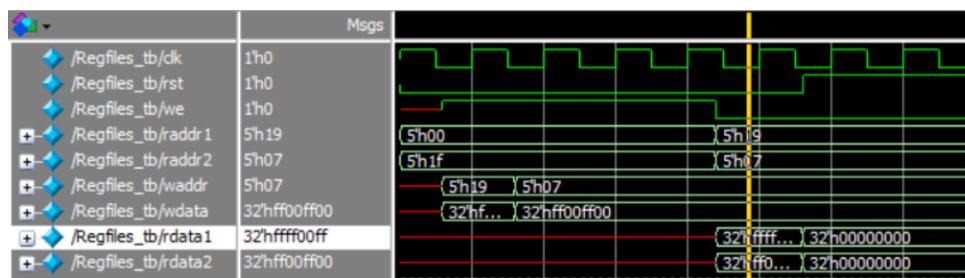


(3)6.8_3 Regfiles 寄存器堆(32 位)

1、写测试



2、读测试



3、寄存器清零测试

