

# 同济大学计算机系

## 计算机组成原理实验报告



学 号 2152809

姓 名 曾崇然

专 业 计算机科学与技术

授课老师 张冬冬老师

## 一、实验内容

了解乘法器的实现原理，实现 32 位无符号乘法器和 32 位带符号乘法器

## 二、模块建模

### 1. 无符号 32 位乘法器

功能描述：将两个 32 位无符号数相乘得到一个 64 位无符号数的

verilog 代码：

```
module MULTU(  
    input clk,  
    input reset,  
    input [31:0] a,  
    input [31:0] b,  
    output reg [63:0] z  
);  
reg [31:0] part1;  
reg [31:0] part2;  
reg [31:0] A;  
reg [31:0] B;  
reg [4:0] i;  
reg flag;  
  
initial  
begin  
    i=0;  
    part1=0;  
    part2=0;  
    A=0;  
    B=0;  
    flag=0;  
end//初始化  
  
always @ (a,b)  
begin  
    if(flag==0)  
    begin  
        A=a;  
        B=b;  
        flag=1;  
    end  
end//是否可进行下一次计算
```

```
always @ (clk)  
begin  
    if(flag==1)  
    begin  
        if(reset==1)  
        begin  
            part1=0;  
            part2=0;  
            i=0;  
            z=0;  
            A=0;  
            B=0;  
            flag=0;  
        end  
        else  
        begin  
            if(B[i]==1)  
            part1=part1+A;  
            part2[i]=part1[0];  
            part1=part1>>1;  
            if(i==31)  
            begin  
                z[63:32]=part1[31:0];  
                z[31:0]=part2[31:0];  
                part1=0;  
                part2=0;  
                i=0;  
                flag=0;  
            end  
            else  
                i=i+1;  
        end  
    end  
end  
endmodule
```

## 2. 带符号 32 位乘法器

功能描述：将两个带符号 32 位数相乘得到一个带符号 64 位数

verilog 代码：

```
module MULT(  
    input clk,  
    input reset,  
    input [31:0] a,  
    input [31:0] b,  
    output reg [63:0] z  
);  
    reg [32:0] A;  
    reg [31:0] B;  
    reg [32:0] part1;  
    reg [30:0] part2;  
    reg [4:0] i;  
    reg flag;  
    initial  
    begin  
        part1=0;  
        part2=0;  
        i=0;  
        flag=0;  
    end  
  
    always @ (a,b)  
    begin  
        if(flag==0)  
        begin  
            A[32]=a[31];  
            A[31:0]=a[31:0];  
            B=b;  
            flag=1;  
        end  
    end  
  
    always @ (clk)  
    begin  
        if(flag==1)  
        begin  
            if(reset==1)  
            begin  
                part1=0;  
                part2=0;  
                i=0;  
                flag=0;  
            end  
            else  
            begin  
                if(B[i]==1)  
                part1=part1+A;  
                part2[i]=part1[0];  
                part1[31:0]=part1[32:1];  
                part1[32]=part1[31];  
                if(i==30)  
                begin  
                    z[63:31]=part1[32:0];  
                    z[30:0]=part2[30:0];  
                    if(B[31]==1)  
                    z[63:31]=z[63:31]-A;  
                    i=0;  
                    part1=0;  
                    part2=0;  
                    flag=0;  
                end  
            end  
            else  
                i=i+1;  
        end  
    end  
end  
  
endmodule
```

## 三、测试模块建模

32 位无符号乘法器：

```
module MULTU_tb()  
    reg clk;
```

```

reg reset;
reg [31:0] a;
reg [31:0] b;
wire [63:0] z;
MULTU MULTU_inst(clk,reset,a,b,z);
initial
begin
    clk=0;
    a=3;
    b=3;
    reset=0;
    #200 a=521;
    b=3;
    #200
    a=300;
    b=521;
end
always
    #10 clk=~clk;
endmodule

```

32 位带符号乘法器:

```

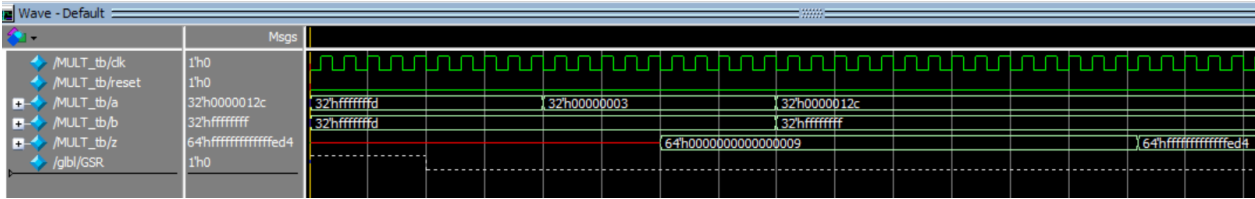
module MULT_tb();
reg clk;
reg reset;
reg [31:0] a;
reg [31:0] b;
wire [63:0] z;
MULT MULT_inst(clk,reset,a,b,z);
initial
begin
    clk=0;
    a=-3;
    b=-3;
    reset=0;
    #200 a=3;
    b=-3;
    #200
    a=300;
    b=-1;
end
always
    #10 clk=~clk;
endmodule

```

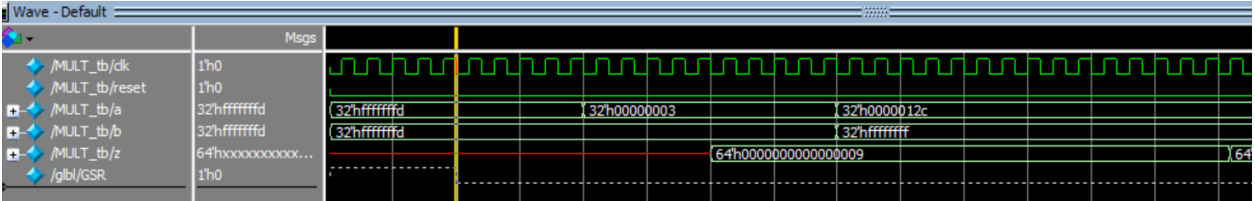
# 四、实验结果

modolsim 截图：

32 位无符号加法器：



32 位有符号加法器：



（在上一轮计算完成之前输入的乘数不被接受，除非遇到 reset 信号）