# 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位无符号书相乘得到一个64位无符号数的
verilog 代码:
                                               always @ (clk)
module MULTU(
                                               begin
input clk,
                                                 if(flag==1)
input reset,
                                                 begin
input [31:0] a,
                                                   if(reset==1)
input [31:0] b,
                                                   begin
output reg [63:0] z
                                                     part1=0;
                                                     part2=0;
);
reg [31:0] part1;
                                                     i=0;
reg [31:0] part2;
                                                     z=0;
                                                     A=0;
reg [31:0] A;
reg [31:0] B;
                                                     B=0;
reg [4:0] i;
                                                     flag=0;
reg flag;
                                                   end
                                                   else
initial
                                                   begin
begin
                                                     if(B[i]==1)
  i=0;
                                                        part1=part1+A;
  part1=0;
                                                     part2[i]=part1[0];
  part2=0;
                                                     part1=part1>>1;
  A=0;
                                                     if(i==31)
  B=0;
                                                     begin
  flag=0;
                                                        z[63:32]=part1[31:0];
end//初始化
                                                        z[31:0]=part2[31:0];
                                                        part1=0;
always @ (a,b)
                                                        part2=0;
begin
                                                        i=0;
  if(flag==0)
                                                        flag=0;
  begin
                                                     end
    A=a;
                                                     else
    B=b;
                                                        i=i+1;
    flag=1;
                                                   end
  end
                                                 end
end//是否可进行下一次计算
                                               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(
                                                   if(flag==1)
input clk,
                                                   begin
input reset,
                                                     if(reset==1)
input [31:0] a,
                                                     begin
input [31:0] b,
                                                        part1=0;
output reg [63:0] z
                                                        part2=0;
);
                                                        i=0;
reg [32:0] A;
                                                        flag=0;
reg [31:0] B;
                                                     end
reg [32:0] part1;
                                                     else
reg [30:0] part2;
                                                     begin
reg [4:0] i;
                                                        if(B[i]==1)
reg flag;
                                                          part1=part1+A;
initial
                                                        part2[i]=part1[0];
begin
                                                        part1[31:0]=part1[32:1];
  part1=0;
                                                        part1[32]=part1[31];
                                                        if(i==30)
  part2=0;
  i=0;
                                                        begin
  flag=0;
                                                          z[63:31]=part1[32:0];
end
                                                          z[30:0]=part2[30:0];
                                                          if(B[31]==1)
always @ (a,b)
                                                             z[63:31]=z[63:31]-A;
begin
                                                          i=0;
  if(flag==0)
                                                          part1=0;
  begin
                                                          part2=0;
    A[32]=a[31];
                                                          flag=0;
    A[31:0]=a[31:0];
                                                        end
    B=b;
                                                        else
    flag=1;
                                                          i=i+1;
  end
                                                     end
end
                                                   end
                                                 end
always @ (clk)
                                                 endmodule
begin
```

## 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
```

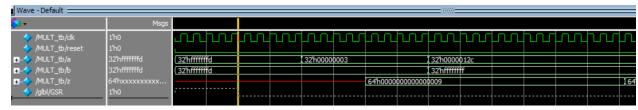
## 四、实验结果

### modolsim 截图:

32 位无符号加法器:

Wave - Default :																	
<b>\$</b> 1 <b>₹</b>	Msgs																
/ /MULT_tb/dk / /MULT_tb/reset	1'h0 1'h0	w	w	w	w	M	TT.		w	<b></b>	w	<b></b>	w	M.	r	w	w
	32'h0000012c 32'hffffffff	32'hffffff 32'hfffffff				32'h0000	0003			32'h00000 32'hffffff							
<b>II</b> /MULT_tb/z	64'hfffffffffffed4						64	64'h0000	_	_						∑64'hffff	ffffffffed4
→ /glbl/GSR	1'h0																

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)