同济大学计算机系
计算机组成原理实验报告
32 位除法器实验

1.实验介绍

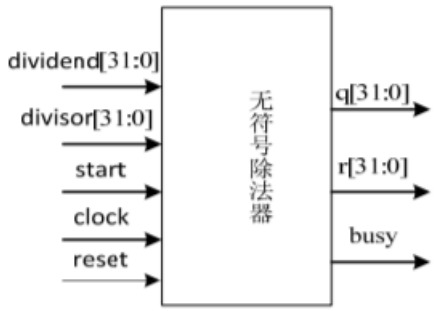
通过本次试验，了解除法器的实现原理，并学习如何实现一个除法器，本实验将实现32位无符号除法器和32位带符号除法器。

2.实验目标

- 了解32位带符号、无符号除法器的实现原理
- 使用Verilog实现32位无符号除法器和带符号除法器

3.实验原理

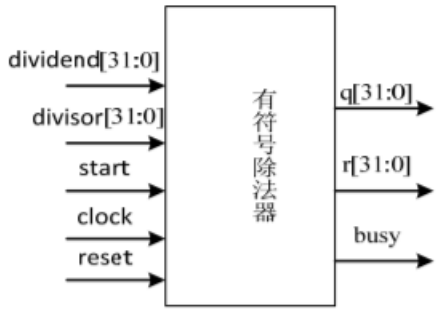
1）无符号除法器功能为：将两个32位无符号数相除，得到一个32位的商和32位的余数。



接口定义

```
module DIV(
    input [31:0]dividend,        //被除数
    input [31:0]divisor,         //除数
    input start,                 //启动除法运算
    input clock,
    input reset,
    output [31:0]q,              //商
    output [31:0]r,              //余数
    output busy                  //除法器忙标志位
    );
```

2）带符号除法器功能为：将两个32带无符号数相除，得到一个32位的商和32位的余数。



接口定义

```
module DIVU(
    input [31:0]dividend,        //被除数
    input [31:0]divisor,         //除数
    input start,                 //启动除法运算
    input clock,
    input reset,
    output [31:0]q,              //商
    output [31:0]r,              //余数
    output busy                  //除法器忙标志位
    );
```

4.实验步骤

1.新建 Vivado 工程

2.编写各个模块

3.用 ModelSim 仿真测试各模块

5.模块建模

1）DIVU

功能描述：

作为 32 位无符号除法器使用，输入[31:0]dividend、[31:0]divisor、start、clock、reset、输出[31:0]q、[31:0]r、busy。其中[31:0]dividend 为被除数，[31:0]divisor 为除数、start 为除法启动信号、clock 为时钟信号，reset 为复位信号，[31:0]q 为商，[31:0]r 为余数，busy 为除法器忙标志位。

Verilog 代码：

```
module DIVU(
    input [31:00] dividend,
    input [31:00] divisor,
    input start,
```

```
input clock,
input reset,
output [31:00] q,
output [31:00] r,
output reg busy
);
reg [5:0] count;
reg [31:00] reg_q;
reg [31:00] reg_r;
reg [31:00] reg_b;
reg r_sign;
wire[32:0]sub_add=r_sign?({reg_r,q[31]}+{1'b0,reg_b}):
({reg_r,q[31]}-{1'b0,reg_b});
assign r=r_sign?reg_r+reg_b:reg_r;
assign q=reg_q;
always @(posedge clock or posedge reset)
begin
    if(reset)
    begin
        count<=0;
        busy<=0;
    end
    else
    begin
        if(start)
        begin
            reg_r<=32'b0;
            r_sign<=0;
            reg_q<=dividend;
            reg_b<=divisor;
            count<=0;
            busy<=1;
        end
        else if(busy)
        begin
            reg_r<=sub_add[31:0];
            r_sign<=sub_add[32];
            reg_q<={reg_q[30:0],~sub_add[32]};
            count<=count+3'b1;
```

```
                if(count==6'd31)
                    busy<=0;
            end
        end
    end
endmodule
```

2）MULT

功能描述：

作为32位带符号除法器使用，输入[31:0]dividend、[31:0]divisor、start、clock、reset、输出[31:0]q、[31:0]r、busy。其中[31:0]dividend为被除数，[31:0]divisor为除数、start为除法启动信号、clock为时钟信号，reset为复位信号，[31:0]q为商，[31:0]r为余数，busy为除法器忙标志位。

Verilog代码：

```
module DIV(
    input [31:0] dividend,
    input [31:0] divisor,
    input start,
    input clock,
    input reset,
    output [31:0] q,
    output [31:0] r,
    output reg busy
    );
    reg [5:0] count;
    reg [31:00] reg_q;
    reg [31:00] reg_r;
    reg [31:00] reg_b;
    wire [31:00] reg_r2;
    reg r_sign,sign;
    wire [32:0] sub_add=r_sign?({reg_r,reg_q[31]}+{1'b0,reg_b}):
    ({reg_r,reg_q[31]}-{1'b0,reg_b});
    assign reg_r2=r_sign?reg_r+reg_b:reg_r;
    assign r=dividend[31]?(~reg_r2+1):reg_r2;
    assign q=(divisor[31]^dividend[31])?(~reg_q+1):reg_q;
    always @(posedge clock or posedge reset)
    begin
        if(reset)
        begin
```

```
            count<=0;
            busy<=0;
        end
        else
        begin
            if(start)
            begin
                reg_r<=32'b0;
                r_sign<=0;
                if(dividend[31]==1)
                    reg_q<=~dividend+1;
                else
                    reg_q<=dividend;
                if(divisor[31]==1)
                    reg_b<=~divisor+1;
                else
                    reg_b<=divisor;
                count<=0;
                busy<=1;
            end
            else if(busy)
            begin
                reg_r<=sub_add[31:0];
                r_sign<=sub_add[32];
                reg_q<={reg_q[30:0],~sub_add[32]};
                count<=count+1;
                if(count==31)busy<=0;
            end
        end
    end
endmodule
```

6.测试模块建模

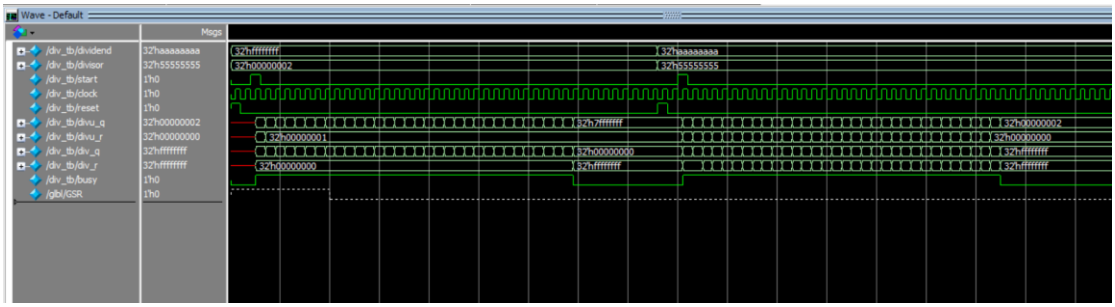
```
`timescale 1ns / 1ps
module div_tb;
    reg [31:0] dividend;
    reg [31:0] divisor;
    reg start;
```

```
    reg clock;
    reg reset;
    wire [31:0] divu_q;
    wire [31:0] divu_r;
    wire [31:0] div_q;
    wire [31:0] div_r;
    wire busy;
    DIVU
U1(.dividend(dividend),.divisor(divisor),.start(start),.clock(clock),.reset(res
et),.q(divu_q),.r(divu_r),.busy(busy));
    DIV
U2(.dividend(dividend),.divisor(divisor),.start(start),.clock(clock),.reset(res
et),.q(div_q),.r(div_r),.busy(busy));
    initial
    begin
        reset = 1;
        clock = 0;
        start = 0;
        dividend = 32'b11111111_11111111_11111111_11111111;
        divisor = 32'b10;
        # 10
        reset = 0;
        # 10
        start = 1;
        # 10
        start = 0;
        # 400
        reset = 1;
        dividend = 32'b10101010_10101010_10101010_10101010;
        divisor = 32'b01010101_01010101_01010101_01010101;
        # 10
        reset = 0;
        # 10
        start = 1;
        # 10
        start = 0;
    end
    always
    #5 clock <= ~clock;
```

```
endmodule
```

7. 实验结果

modelsim 仿真波形图：



可以看出：

1) dividend = 32'b11111111_11111111_11111111_11111111

   divisor = 32'b10

   divu_q = 32'h7fffffff

   divu_r = 32'h00000001

   div_q = 32'h00000000

   div_r = 32'hffffffff

2) dividend = 32'b10101010_10101010_10101010_10101010

   divisor = 32'b01010101_01010101_01010101_01010101

   divu_q = 32'h00000002

   divu_r = 32'h00000000

   div_q = 32'hffffffff

   div_r = 32'hffffffff

无论从最终结果来看还是从除法的中间结果来看，都可得出上述测试结果无误，因此 32 位除法器设计完成。