同济大学计算机系

计算机组成原理实验报告

学　　号 2351753

姓　　名 黄保翔

专　　业 计算机科学与技术

授课老师 张冬冬

一、实验内容

实现 32 位无符号除法器和 32 位带符号除法器

二、模块建模

1、32 位无符号除法器

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 位商和 32 位余数。本实验分别实现 32 位有符号和无符号除法器，结果为 32 位商 quotient 和 32 位余数 remainder， 分别存放在 CPU 的专用寄存器 LO 和 HI 中。除法器时钟信号下降沿时检查 start 信号，有效 时开始执行，执行除法指令时，busy 标志位置 1。在执行除法指令时，任何情况下不产生算 术异常，当除数为 0 时，运算结果未知，对除法器除数为 0 和溢出情况的发生通过汇编指令 中其他指令进行检查和处理。

代码:

```
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,      //高电平有效
    output  [31:0]  q,          //商
    output  [31:0]  r,          //余数
    output reg      busy        //除法器忙标志位
);
    reg     [4:0]   count;
    reg     [31:0]  reg_q;
    reg     [31:0]  reg_r;
    reg     [31:0]  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}); //加、减法器
    assign r = r_sign? reg_r + reg_b : reg_r;
    assign q = reg_q;

    always @(posedge clock or posedge reset)
    begin
        if (reset == 1)
        begin
            count   <=  5'b0;
            busy    <=  0;
        end
        else
```

```
        begin
            if (start&&(!busy))
            begin //初始化
                reg_r   <= 32'b0;
                r_sign  <= 0;
                reg_q   <= dividend;
                reg_b   <= divisor;
                count   <= 5'b0;
                busy    <= 1'b1;
            end
            else if (busy)
            begin //循环操作
                reg_r   <= sub_add[31:0];   //部分余数
                r_sign  <= sub_add[32];
                reg_q   <= {reg_q[30:0],~sub_add[32]};
                count   <= count + 5'b1;    //计数器加一
                if (count == 5'b11111) begin
                    busy <= 0;
                end
            end
        end
    end
endmodule
```

2、32 位有符号除法器

有符号除法器和无符号除法器的思路相同，核心代码不变，将输入的补码先进行符号判断，转为绝对值进行计算，得出的商和余数在根据被除数和除数的值进行符号判断，商等于被除数和余数符号的异或，余数的符号等于被除数的符号

```
module DIV(
    input   [31:0]  dividend,   //被除□?
    input   [31:0]  divisor,    //除数
    input           start,      //启动除法运算
    input           clock,
    input           reset,      //高电平有□?
    output  [31:0]  q,          //□?
    output  [31:0]  r,          //余数
    output reg      busy        //除法器忙标志□?
);
    reg     [4:0]   count;
    reg     [31:0]  reg_q;
    reg     [31:0]  reg_r;
    reg     [31:0]  reg_b;
```

```
    reg           r_sign;
    wire [31:0] reg_temp_r;
    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}); //加□?□减法器
    assign reg_temp_r = r_sign? reg_r + reg_b : reg_r;
    assign r = dividend[31] ? -reg_temp_r : reg_temp_r;
    assign q = dividend[31] ^ divisor[31] ? -reg_q : reg_q;

    always @(posedge clock or posedge reset)
    begin
        if (reset == 1)
        begin
            count   <=  5'b0;
            busy    <=  0;
        end
        else
        begin
            if (start&&(!busy))
            begin //初始□?
                reg_r   <= 32'b0;
                r_sign  <= 0;
                reg_q   <= dividend[31] ? -dividend : dividend; //符号
扩展
                reg_b   <= divisor[31] ? -divisor : divisor; //符号扩展
                count   <= 5'b0;
                busy    <= 1'b1;
            end
            else if (busy)
            begin //循环操作
                reg_r   <= sub_add[31:0];   //部分余数
                r_sign  <= sub_add[32];
                reg_q   <= {reg_q[30:0],~sub_add[32]};
                count   <= count + 5'b1;    //计数器加□?
                if (count == 5'b11111) begin
                    busy <= 0;
                end
            end
        end
    end
endmodule
```

三、测试模块建模

1、无符号除法器

```
module DIVU_tb();

    // 定义模块输入输出信号
    reg [31:0] dividend;
    reg [31:0] divisor;
    reg start;
    reg clock;
    reg reset;
    wire [31:0] q;
    wire [31:0] r;
    wire busy;

    // 实例化除法器模块
    DIVU uut (
        .dividend(dividend),
        .divisor(divisor),
        .start(start),
        .clock(clock),
        .reset(reset),
        .q(q),
        .r(r),
        .busy(busy)
    );

    // 生成时钟信号
    initial begin
        clock = 0;
        forever #2 clock = ~clock; // 10ns 周期的时钟信□?
    end

    // 测试过程
    initial begin
        clock = 0;
        reset = 1;
        start = 0;
        #10;
        reset = 0;
        dividend = 32'h0000000A; // 10
        divisor = 32'h00000002;  // 2
        start = 1;   // 结果应为 5
        #10;
        start = 0;
```

```
        #150;

        reset = 1;
        #10;
        reset = 0;
        dividend = 32'h00000000; // 0
        divisor = 32'hffffffff;  // 4294967295
        start = 1;   // 结果应为 0
        #10;
        start = 0;
        #150;

        reset = 1;
        #10;
        reset = 0;
        dividend = 32'h00000014; // 20
        divisor = 32'h00000003;  // 3
        start = 1;   // 结果应为 1□?0.5
        #10;
        start = 0;
        #150;

        reset = 1;
        #10;
        reset = 0;
        dividend = 32'haaaaaaaa; // 2863311530
        divisor = 32'h55555555;  // 1431655765
        start = 1;   // 结果应为 2
        #10;
        start = 0;
        #150;

        reset = 1;
        #10;
        reset = 0;
        dividend = 32'h55555555; // 1431655765
        divisor = 32'h7fffffff;  // 2147483647
        start = 1;   // 结果应该□?0
        #10;
        start = 0;
        #150;

        reset = 1;
        #10;
```

```
        reset = 0;
    end

endmodule
```

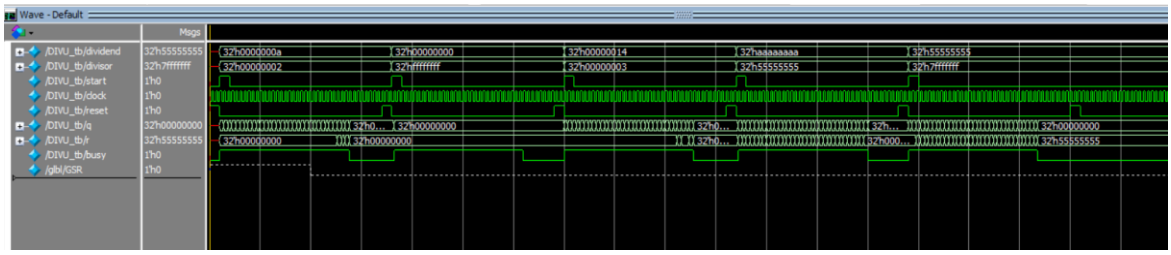
2、有符号除法器

```
module DIV_tb();
reg  [31:0]  dividend;
reg  [31:0]  divisor;
reg  start;
reg  clock;
reg  reset;
wire [31:0]  q;
wire [31:0]  r;
wire busy;

DIV uut(dividend, divisor, start, clock, reset, q, r, busy);

initial begin
    clock = 0;
    reset = 1;
    start = 0;
    #10;
    reset = 0;
    dividend = 32'h00000014; // 20
    divisor = 32'h00000003;  // 3
    start = 1;   // 结果应为 q=6, r=2
    #10;
    start = 0;
    #150;

    reset = 1;
    #10;
    reset = 0;
    dividend = 32'hffffffec; // -20
    divisor = 32'h00000003;  // 3
    start = 1;   // 结果应为 q=-6, r=-2
    #10;
    start = 0;
    #150;

    reset = 1;
```

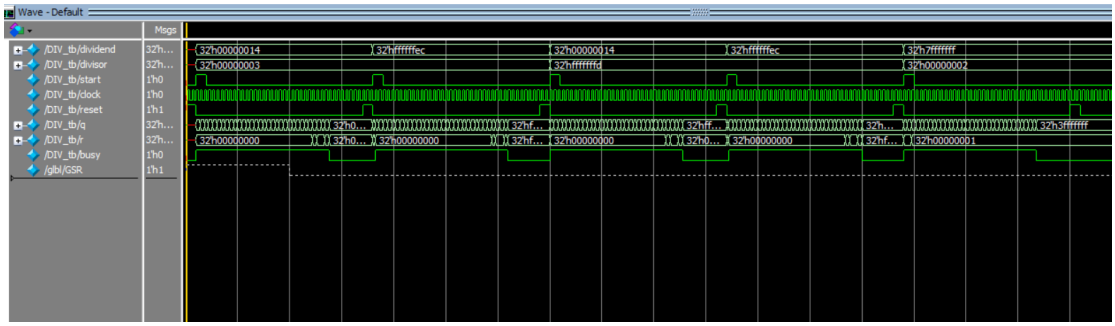
```
    #10;
    reset = 0;
    dividend = 32'h00000014; // 20
    divisor = 32'hfffffffd;  // -3
    start = 1;   // 结果应为 q=-6, r=2
    #10;
    start = 0;
    #150;

    reset = 1;
    #10;
    reset = 0;
    dividend = 32'hffffffec; // -20
    divisor = 32'hfffffffd;  // -3
    start = 1;   // 结果应为 q=6, r=-2
    #10;
    start = 0;
    #150;

    reset = 1;
    #10;
    reset = 0;
    dividend = 32'h7fffffff; // 2147483647（最大正数）
    divisor = 32'h00000002;  // 2
    start = 1;   // 结果应为 q=1073741823, r=1
    #10;
    start = 0;
    #150;

    reset = 1;
    #10;
    reset = 0;
end

always #2 clock = ~clock;
endmodule
```

四、实验结果

1、 无符号除法器


Wave - Default
Msgs
/DIVU_tb/dividend
/DIVU_tb/divisor
/DIVU_tb/start
/DIVU_tb/clock
/DIVU_tb/reset
/DIVU_tb/q
/DIVU_tb/r
/DIVU_tb/busy
/glbl/GSR

2、有符号除法器


Wave - Default
Msgs
/DIV_tb/dividend
/DIV_tb/divisor
/DIV_tb/start
/DIV_tb/clock
/DIV_tb/reset
/DIV_tb/q
/DIV_tb/r
/DIV_tb/busy
/glbl/GSR