```
2211290 姚知言 Verilog 作业 5
参照循环语句中的例子,完成一个32位二进制数的乘法。
multiplier.v 设计:
`timescale 1ns / 1ps
module multiplier(
    input [31:0] ina,inb,
    output [63:0] ans
    );
    reg [63:0] out,a;
    reg [31:0] b;
    reg [5:0] i;
    assign ans = out;
    always@(ina or inb) begin
         out = 64'd0;
         a = {32'b0,ina};
         b = inb;
         i = 6'd32;
         while(i) begin
             if(b[0]) begin
                  out = out + a;
             end
             a = (a << 1);
             b = (b >> 1);
             i = i - 1;
         end
    end
endmodule
testbench.v 设计:
`timescale 1ns / 1ps
module testbench();
reg [31:0] ta,tb;
wire [63:0] tans;
multiplier mymul(ta,tb,tans);
initial begin
    ta = 32'b0;
    tb = 32'b0;
end
always #5 ta = $random;
always #5 tb = $random;
endmodule
仿真测试结果:
```

## 设计思路:

输入为两个乘数,输出为乘法结果。

由于是 32 位乘 32 位,结果应设计 64 位以免 溢出。

使用阻塞赋值方式,循环 32 次(使用 i 作为 循环控制变量),每次将 a 左移一位, b 右移 一位, 若此时 b[0]为 1, 则需要把 a 的结果累 加到 out 中。

通过 assign ans = out 把寄存器结果赋值给输 出。

## 设计思路:

初始化为0,每5个时钟周期将新的随机32 位整数赋值给 ta 和 tb,将 ta 和 tb 作为 multiplier 的输入,tans 作为输出。 观察结果。

Name	Value	3,482 ns 3,484 ns	3,486 ns 3,488 ns	3,490 ns 3,492 ns 3,494 n	s 3,496 ns 3,498 :
> 🕶 ta[31:0]	0b636716	9743be2e	0ъ636716	e131eec2	96ec2a2d
> 🐝 tb[31:0]	90138420	4e156f9c	90138420	ch91f895	O2bad905
> 🦋 tans[63:0]	0668c83db4363ac0	2e234e732a83d608	0668c83db4363c0	b231c=58c13de6ea	019bffdc210ef7e1

## 举例分析:

3485ns 处,ta=0x0b636716,tb=0x90138420,tsum=0x0668c83db4363ac0。

通过运行 16 进制计算器,发现答案正确,以下是计算过程。

**0b636716** = 191063830(Base 10)

**90138420** = 2417198112(Base 10)

191063830 × 2417198112 = 461839129147488960(Base 10) = **668c83db4363ac0(Hex)** 

3490ns 处,ta=0xe131eec2,tb=0xca91f895,tsum=0xb231ce58c13de6ea。

通过运行 16 进制计算器,发现答案同样正确,以下是计算过程。

**e131eec2** = 3778145986(Base 10)

ca91f895 = 3398563989(Base 10)

3778145986 × 3398563989 = 12840270893204498154(Base 10) = **b231ce58c13de6ea(Hex)** 

因此,验证成功!