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;

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。

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;

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。

multiplier mymul(ta,tb,tans);

initial begin

ta = 32'b0;

tb = 32'b0;

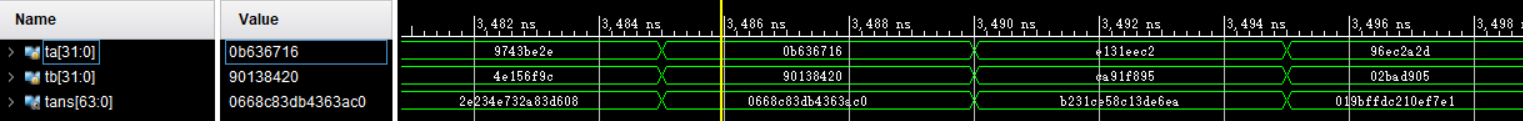
end

always #5 ta = $random;

always #5 tb = $random;

endmodule

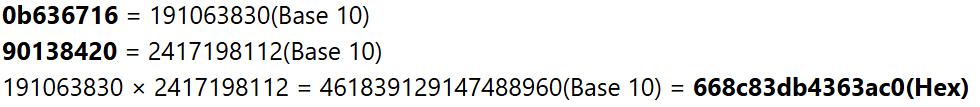
仿真测试结果：



举例分析：

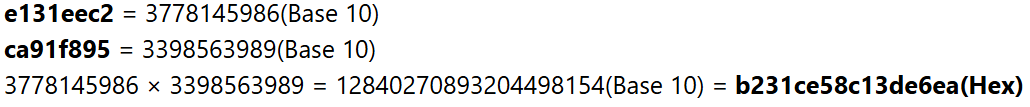
3485ns处，ta=0x0b636716，tb=0x90138420，tsum=0x0668c83db4363ac0。

通过运行16进制计算器，发现答案正确，以下是计算过程。



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。



因此，验证成功！