

## Verilog 第 6 次实验报告

|      |               |    |         |                 |     |
|------|---------------|----|---------|-----------------|-----|
| 实验名称 | 32 位二进制数的乘法器  |    |         |                 |     |
| 学生姓名 | 邢清画           | 学号 | 2211999 | 指导老师            | 董前琨 |
| 实验地点 | 津南实验楼 A 区 308 |    | 实验时间    | 2023 年 12 月 6 日 |     |

### 一、实验项目名称

### 二、完成一个 32 位二进制数的乘法器

### 三、实验目的

实现一个 32 位二进制数的乘法器。通过这个实验，将学习如何使用 Verilog 编程语言中的循环语句来执行乘法运算。具体来说，这需要掌握如何利用 for 循环遍历二进制数的每一位，进行位运算和累加操作，从而实现两个二进制数的乘法。这个实验不仅强化了对循环结构的理解，也加深了对位运算在数字逻辑设计中的应用知识。

### 四、必修或选修

选修

### 五、实验平台

Vivado

### 六、实验内容及步骤

#### 实验内容：

利用循环语句，完成一个 32 位二进制数的乘法器

代码如下：

```
module mul32(  
    input [32:1]a,  
    input [32:1]b,  
    output reg[64:1]outcom );  
    integer i;  
    always@(a or b)  
    begin  
        outcom=0;  
        for(i=1;i<=32;i=i+1)  
        begin  
            if(b[i])  
            begin  
                outcom=outcom+(a<<(i-1));  
            end  
        end  
    end
```

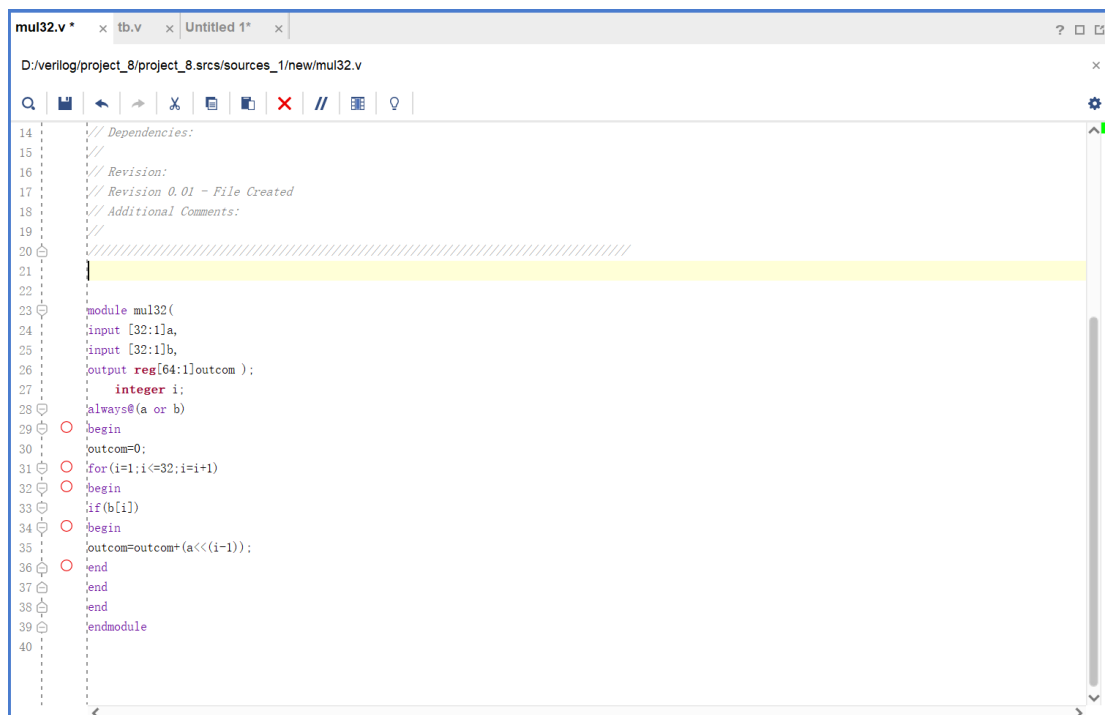
```

end
end
end
endmodule

module tb();
reg [32:1]a;
reg [32:1]b;
wire [64:1]outcom;
mul32 utt(a,b,outcom);
initial begin
a=32'b0000_0000_0000_0000_0000_0000_0000;
b=32'b0000_0000_0000_0000_0000_0000_0000;
end
always#33a=$random%33'b1_0000_0000_0000_0000_0000_0000_0000;
always#33b=$random%33'b1_0000_0000_0000_0000_0000_0000_0000;
endmodule

```

## 七、关键问题讨论

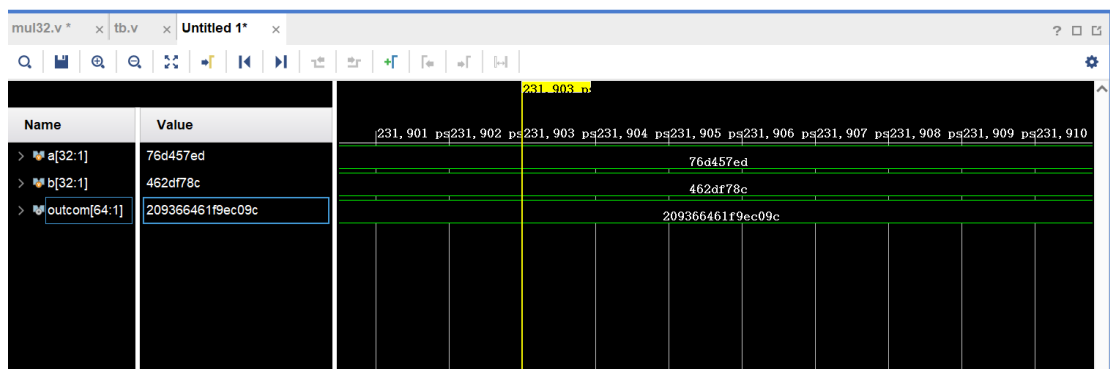
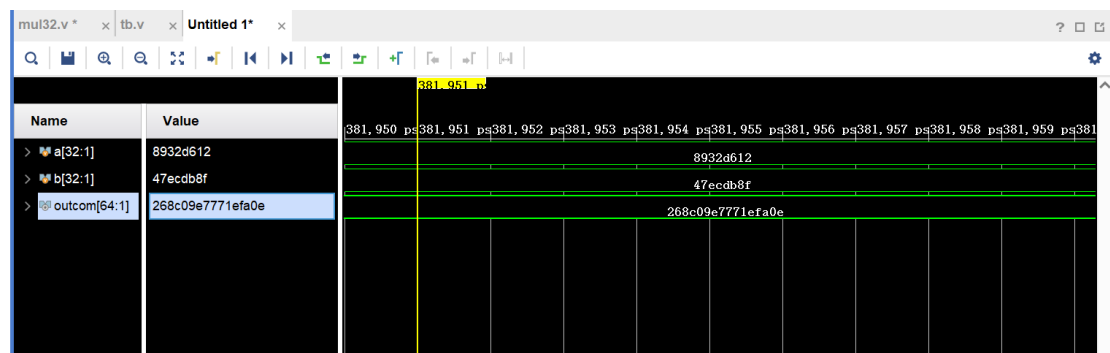


```

mul32.v * x tb.v x Untitled 1* x
D:/verilog/project_8/project_8.srcs/sim_1/new/tb.v
7 // Design Name:
8 // Module Name: tb
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module tb();
24 reg [32:1]a;
25 reg [32:1]b;
26 wire [64:1]outcom;
27 mul32 uut(a,b,outcom);
28 initial begin
29     a=32'b0000_0000_0000_0000_0000_0000_0000;
30     b=32'b0000_0000_0000_0000_0000_0000_0000;
31 end
32 always#33a=$random%33'b1_0000_0000_0000_0000_0000_0000;
33 always#33b=$random%33'b1_0000_0000_0000_0000_0000_0000;
34 endmodule
35

```

上图为代码，下图为波形图



如上波形图所示，以十六进制显示：在这个模块中

图 1：a= 8932d612,b= 47ecdb8f,a\*b= 268c09e7771efa0e, 结果正确。

图 2：a= 76d457ed,b= 462df78c,a\*b= 209366461f9ec09c, 结果正确

## 七、总结

通过本次实验，练习使用了循环语句，加深了对语法的理解和记忆，更好的掌握了 always 和 for 的用法，在 always 块语句中更好的理解组合逻辑的基本原理，实现了逐位相乘和累加的乘法运算。实验不仅强化了对循环结构的理解，也加深了对位运算在数字逻辑设计中的应用知识。