

# lab4\_5

---

实验题目:lab\_4 & lab\_5

实验内容:学习verilog中testbench文件的设计和使用;学习verilog中时序逻辑单元锁存器,触发器等的veirlog实现

姓名学号:PB17030889 吉志远

日期:2018.11.2

lab4\_5  
    hints  
lab4  
    lab4\_1\_1  
    lab4\_2\_1  
    lab4\_3\_1  
lab5  
    lab5\_2\_2  
实验总结

## hints

comments:

如果在综合的时候有warning: generator latch 可能是由于条件语句不完备而造成的比如

solution:

尽量在verilog中避免latch的使用

```
//D latch
always @(D or Enable)
    if (Enable == 1)
        Q=balabala;
        Qbar=balabala;
end
//无else
//类似的还有switch没有defalut等
```

## lab4

### lab4\_1\_1

1. 仿真代码

```
`timescale 1ns / 1ps
```

```

////////////////////////////////////
// Module Name: add_two_values_task_tb
////////////////////////////////////

module add_two_values_task_tb (
);

    reg [3:0] ain, bin;
    wire cout;
    wire [3:0] sum;
    integer k;

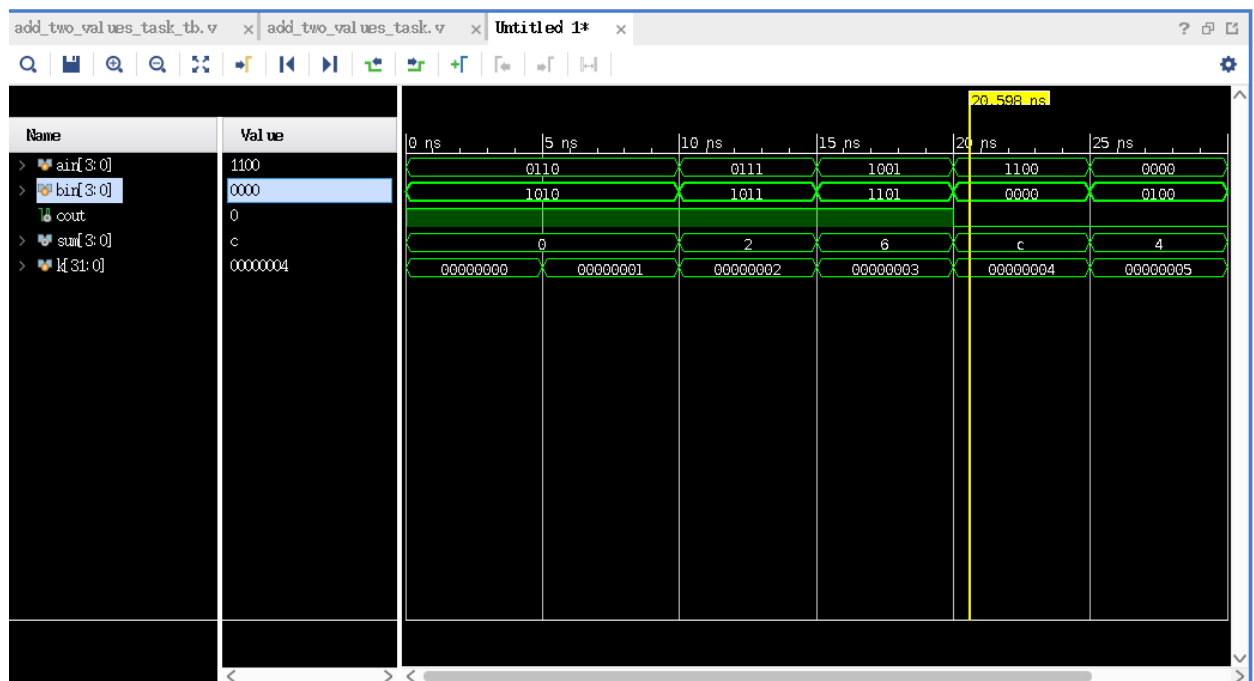
    add_two_values_task DUT (.ain(ain), .bin(bin), .cout(cout), .sum(sum));

    initial
    begin
        ain = 4'h6; bin = 4'ha;
        $display("ain=%b, bin=%b, cout=%b, sum=%b at time=%t",ain, bin, cout, sum,
$time);
        for (k=0; k < 5; k=k+1)
        begin
            #5 ain = ain + k; bin = bin + k;
            $display("ain=%b, bin=%b, cout=%b, sum=%b at time=%t",ain, bin, cout, sum,
$time);
        end
        $display("Simulation Done");
    end

endmodule

```

## 2. 仿真截图



1. 仿真代码

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Module Name: add_two_values_function_tb
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module add_two_values_function_tb (
);

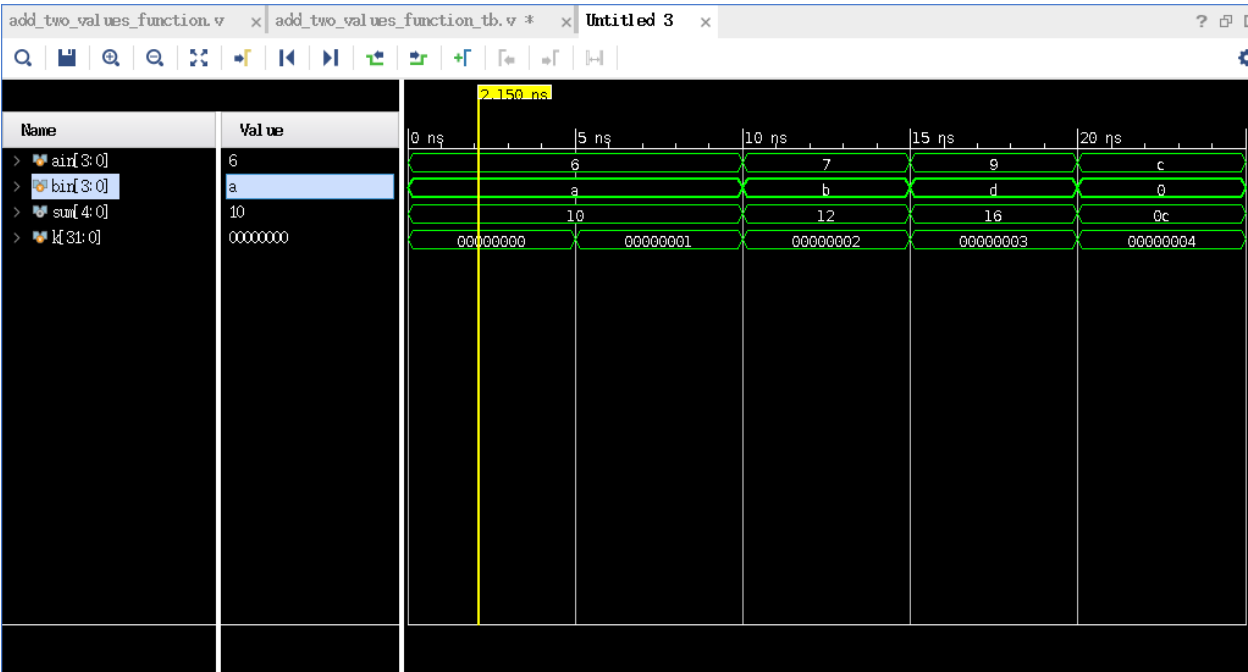
    reg [3:0] ain, bin;
    wire [4:0] sum;
    integer k;

    add_two_values_function DUT (.ain(ain), .bin(bin), .sum(sum));

    initial
    begin
        ain = 4'h6; bin = 4'ha;
        $display("ain=%b, bin=%b, sum=%b at time=%t",ain, bin, sum, $time);
        for (k=0; k < 5; k=k+1)
        begin
            #5 ain = ain + k; bin = bin + k;
            $display("ain=%b, bin=%b, sum=%b at time=%t",ain, bin, sum, $time);
        end
        $display("Simulation Done");
    end

endmodule
```

2. 仿真截图

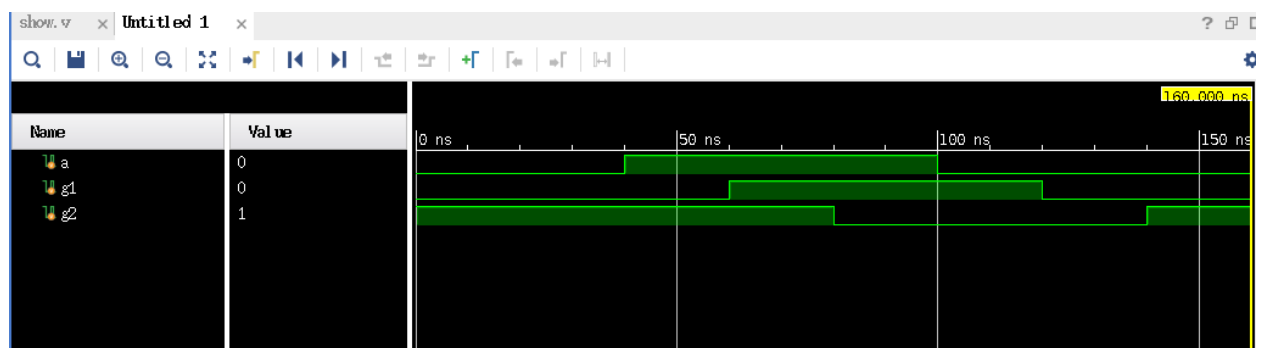


## lab4\_3\_1

### 1. 仿真代码

```
`timescale 1ns / 1ps
module show(
);
  reg a,g1,g2;
  initial begin
    a=0; g1=0; g2=1;
    #40 a=1;
    #20 g1=1;
    #20 g2=0;
    #20 a=0;
    #20 g1=0;
    #20 g2=1;
  end
endmodule
```

### 2. 仿真截图



## lab5

### lab5\_2\_2

### 1. 仿真代码

```
`timescale 1ns / 1ps
module testbench(

);
  reg clk,D,s;
  wire Qa,Qb,Qc,Qd;

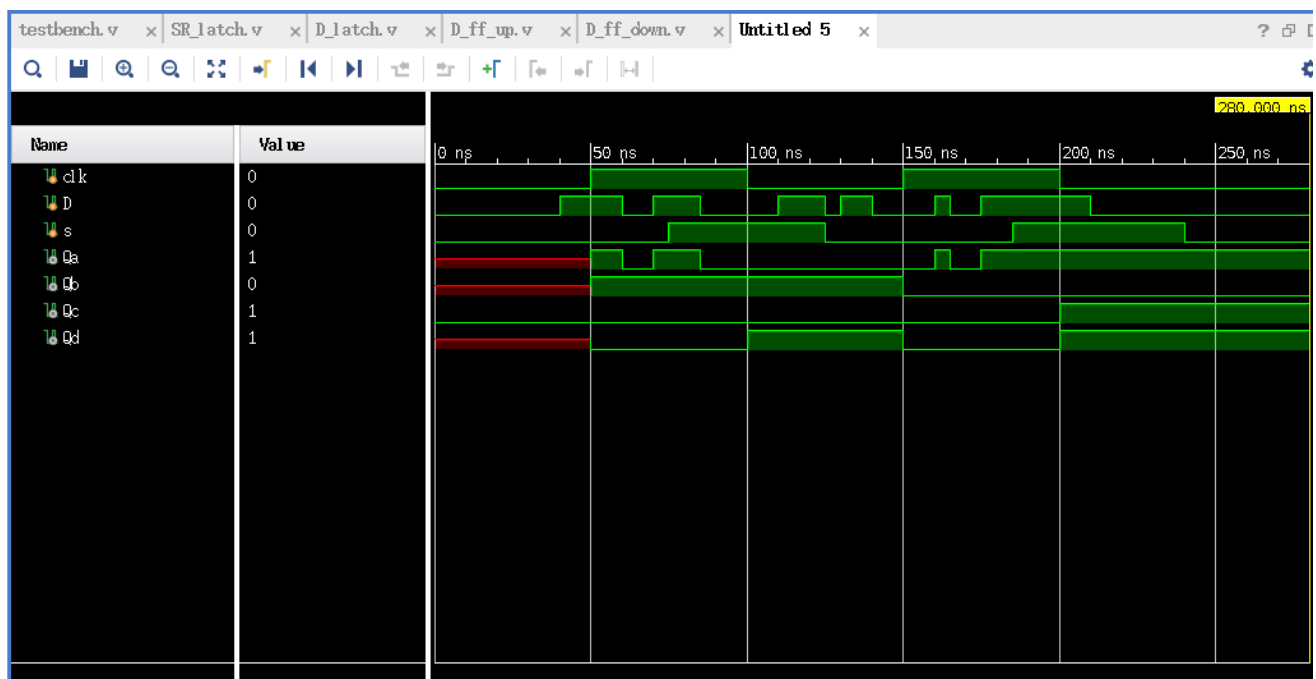
  SR_latch SR (clk,s,Qd);
  D_latch D1a(D,clk,Qa);
  D_ff_up DFU(D,clk,Qb);
  D_ff_down DFD(D,clk,Qc);
  always #50 clk = ~clk;
```

```

initial begin
    clk=0;
    D=0;
    s=0;
    #40 D=1;
    #10 D=0;
    #10 D=1;
    #5 s=1;
    #10 D=0;
    #10 D=1;
    #15 D=0;s=0;
    #5 D=1;
    #10 D=0;
    #10 D=1;
    #5 D=0;
    #10 D=1;
    #10 s=1;
    #25 D=0;
    #10 s=0;
end
endmodule

```

## 2. 仿真截图



## 实验总结

通过本次试验,了解了verilog中posedge和negedge的用法,通过这两个触发信号来实现对触发器锁存器的模拟,并学习了如何通过延时赋值等操作来编写test文件,从而进行仿真,减少烧录到硬件才能发现bug的情况