实验报告

实验题目：Lab 4　　日期：2018 年 11 月 2 日

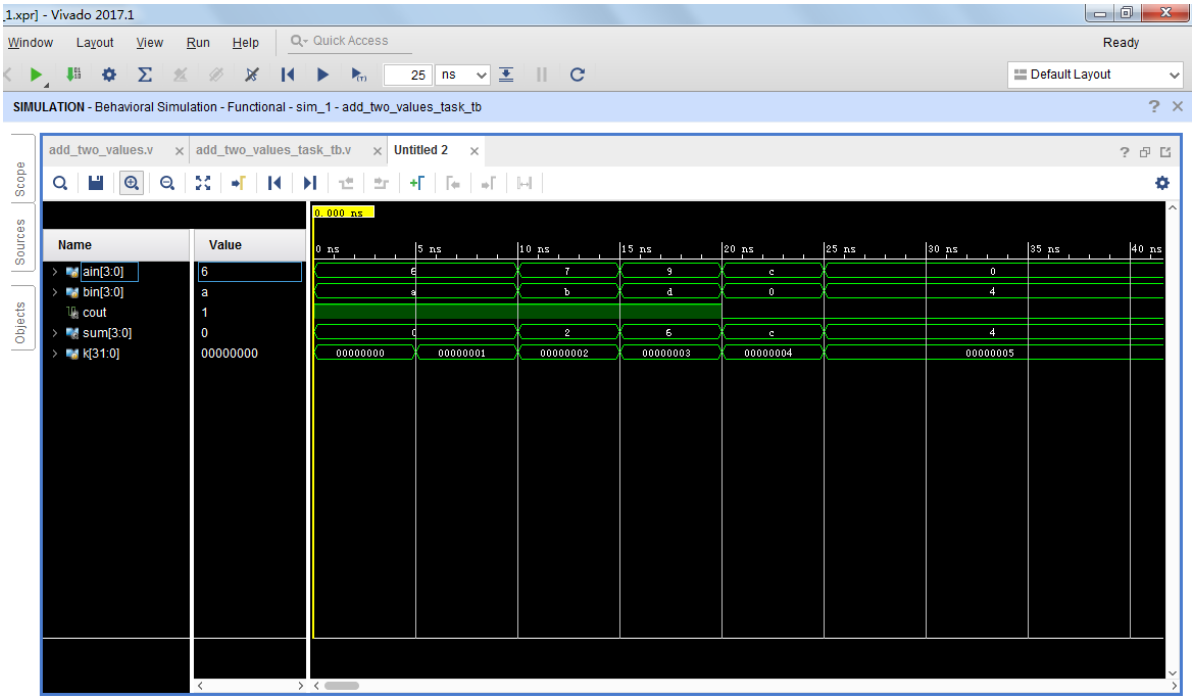
姓名:刘紫檀　　学号:PB17000232　　成绩:_____

实验目的：

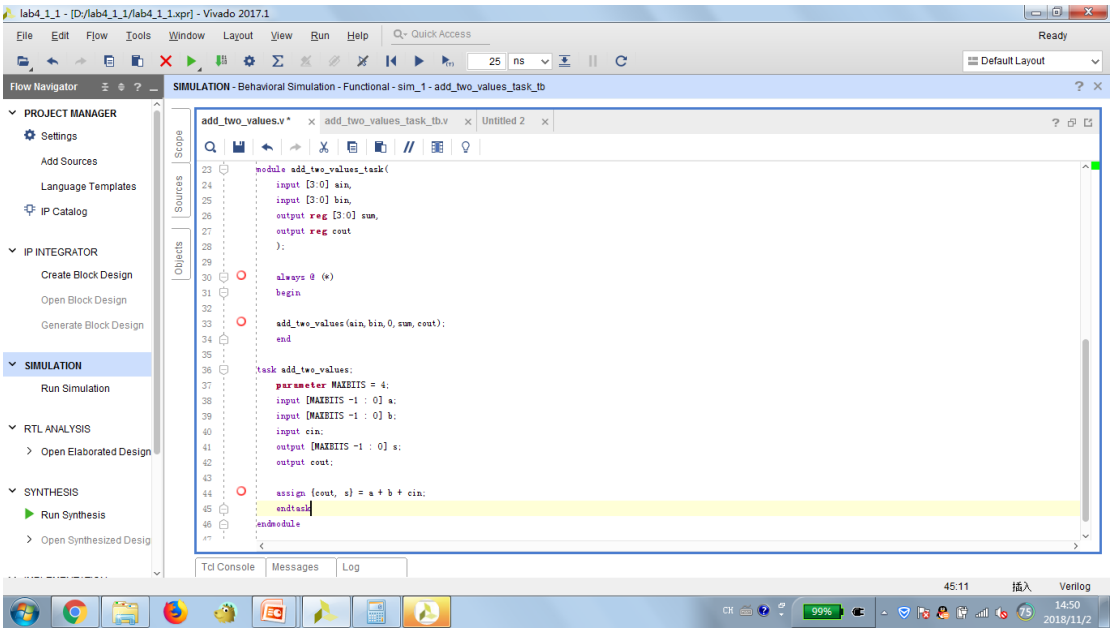
Lab4：1-1、2-1、3-2 仿真代码及仿真截图

Lab5:2-2 对 RS 锁存器、D 锁存器和 D 触发器（上升沿、下降沿）进行仿真，仿真代码和仿真截图

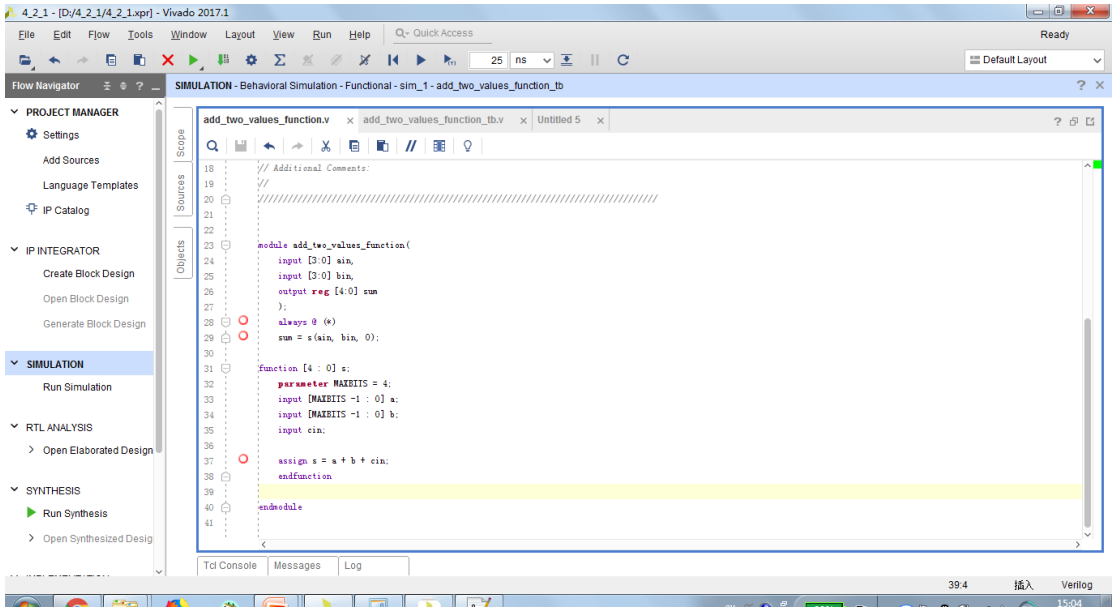
截图：



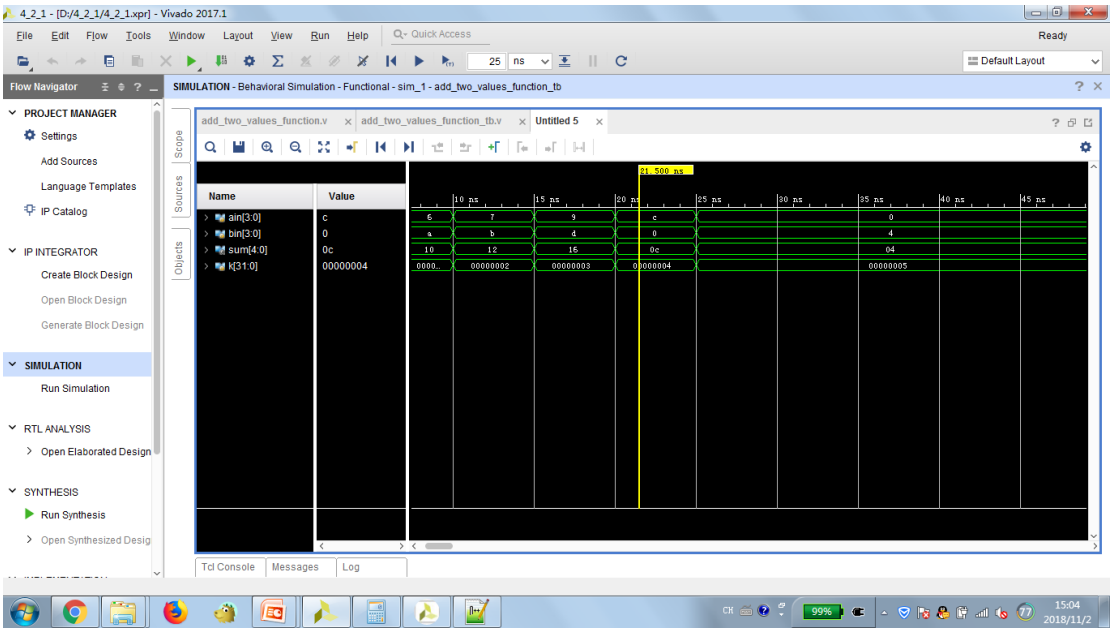
4-1-1 仿真


lab4_1_1 - [D:/lab4_1_1/lab4_1_1.xpr] - Vivado 2017.1
File Edit Flow Tools Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - add_two_values_task_tb
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Desig
add_two_values.v *
add_two_values_task_tb.v
Untitled 2
module add_two_values_task(
input [3:0] ain,
input [3:0] bin,
output reg [3:0] sum,
output reg cout
);
always @ (*)
begin
add_two_values(ain,bin,0,sum,cout);
end
task add_two_values;
parameter MAXBITS = 4;
input [MAXBITS -1 : 0] a;
input [MAXBITS -1 : 0] b;
input cin;
output [MAXBITS -1 : 0] s;
output cout;
assign {cout, s} = a + b + cin;
endtask
endmodule
Tcl Console
Messages
Log
45:11
插入
Verilog

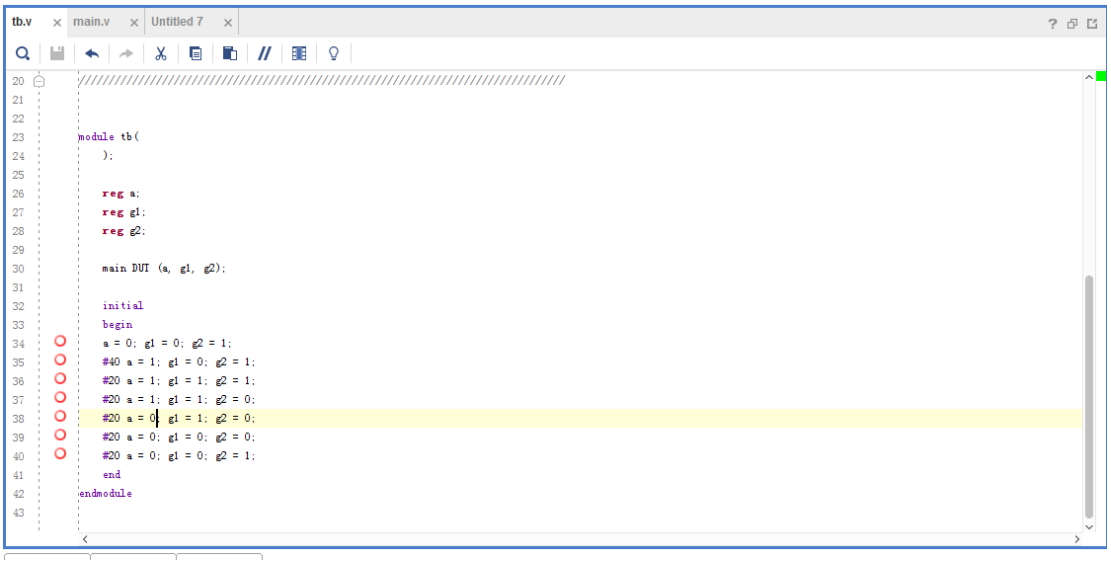
4-1-1 代码


4_2_1 - [D:/4_2_1/4_2_1.xpr] - Vivado 2017.1
File Edit Flow Tools Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - add_two_values_function_tb
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Desig
add_two_values_function.v
add_two_values_function_tb.v
Untitled 5
// Additional Comments:
//
//
module add_two_values_function(
input [3:0] ain,
input [3:0] bin,
output reg [4:0] sum
);
always @ (*)
sum = s(ain, bin, 0);
function [4 : 0] s;
parameter MAXBITS = 4;
input [MAXBITS -1 : 0] a;
input [MAXBITS -1 : 0] b;
input cin;
assign s = a + b + cin;
endfunction
endmodule
Tcl Console
Messages
Log
39:4
插入
Verilog

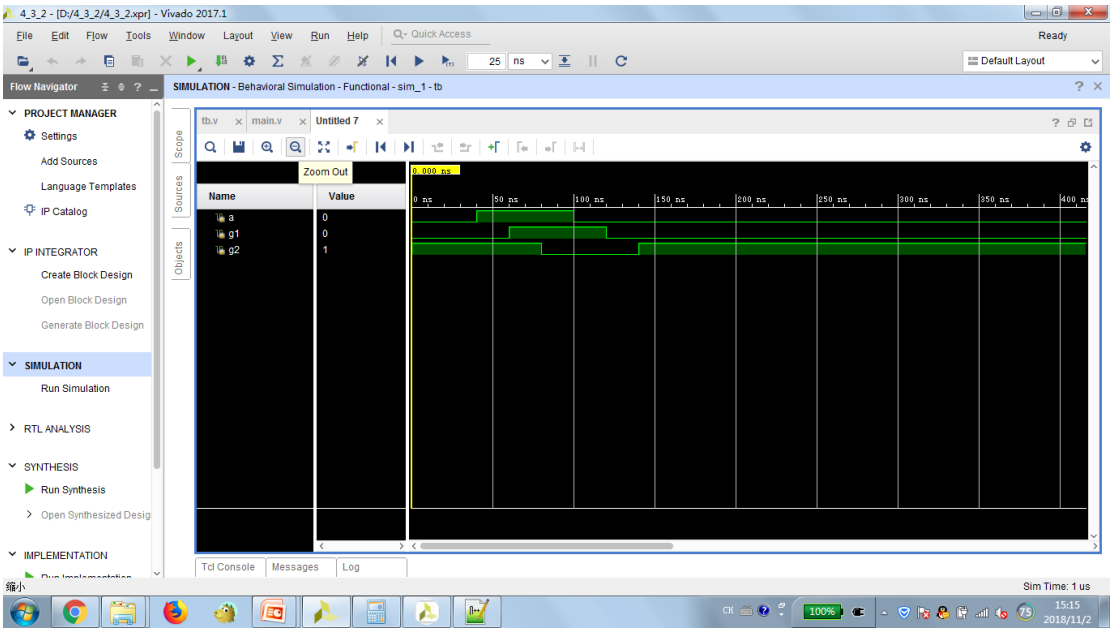
4-2-1 代码


4_2_1 - [D:/4_2_1/4_2_1.xpr] - Vivado 2017.1
File Edit Flow Tools Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - add_two_values_function_tb
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Desig
add_two_values_function.v
add_two_values_function_tb.v
Untitled 5
Name
Value
ain[3:0]
bin[3:0]
sum[4:0]
k[31:0]
Tcl Console
Messages
Log

4-2-1 仿真


tb.v
main.v
Untitled 7
module tb(
);
reg a;
reg g1;
reg g2;
main DUT (a, g1, g2);
initial
begin
a = 0; g1 = 0; g2 = 1;
#40 a = 1; g1 = 0; g2 = 1;
#20 a = 1; g1 = 1; g2 = 1;
#20 a = 1; g1 = 1; g2 = 0;
#20 a = 0; g1 = 1; g2 = 0;
#20 a = 0; g1 = 0; g2 = 0;
#20 a = 0; g1 = 0; g2 = 1;
end
endmodule

4-3-2 代码



4-3-2 仿真

```
`timescale 1ns / 1ps

module latches(
    input Clock,
    input D,
    input S,
    output Qa,
    output Qb,
    output Qc,
    output Qd
    );

    D_latch_behavior p(D, Clock, Qa);
    D_ff_pbehavior q(D, Clock, Qb);
    D_ff_nbehavior r (D, Clock, Qc);
    RS_latch m(D, Clock, Qd);



endmodule

module D_latch_behavior(
    input D,
    input Enable,
    output reg Q,
    output reg Qbar);
    always @ (*)
```

```
    if (Enable)
    begin
        Q <= D;
        Qbar <= ~D;
    end
 endmodule

 module D_ff_pbehavior(
    input D,
    input Clk,
    output reg Q);
    always @ (posedge Clk)
        Q <= D;
endmodule

 module D_ff_nbehavior(
    input D,
    input Clk,
    output reg Q);
    always @ (negedge Clk)
        Q <= D;
endmodule

 module RS_latch(
    input R,
    input S,
//    output reg Q);
    output Q);
    wire Qbar;
    nor (Q, R, Qbar);
    nor (Qbar, S, Q);

   /* always @ (*)
    begin
        if (R == 1)
            Q <= 0;
        if (S == 1)
            Q <= 1;
        //Haven't figure out how to simulate R=S=1, imp. defined behaviour
    end*/
endmodule
```

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

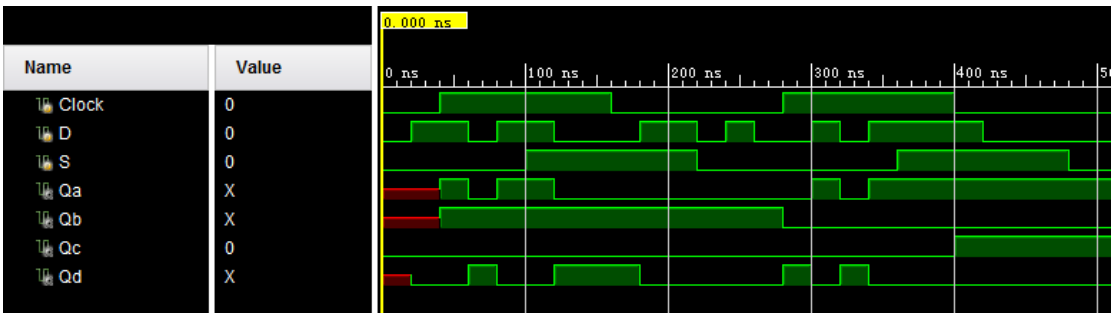
```
module tb(

    );

        reg Clock, D, S;
        wire Qa, Qb, Qc, Qd;
        latches DUT (Clock, D, S, Qa, Qb, Qc, Qd);

        initial
        begin
        Clock = 0; D = 0; S = 0;
        #20 Clock = 0; D = 1; S = 0;
        #20 Clock = 1; D = 1; S = 0;
        #20 Clock = 1; D = 0; S = 0;
        #20 Clock = 1; D = 1; S = 0;
        #20 Clock = 1; D = 1; S = 1;  //5
        #20 Clock = 1; D = 0; S = 1;
        #20 Clock = 1; D = 0; S = 1;
        #20 Clock = 0; D = 0; S = 1;
        #20 Clock = 0; D = 1; S = 1;  //9
        #20 Clock = 0; D = 1; S = 1;  //A
        #20 Clock = 0; D = 0; S = 0;
        #20 Clock = 0; D = 1; S = 0;
        #20 Clock = 0; D = 0; S = 0;
        #20 Clock = 1; D = 0; S = 0;
        #20 Clock = 1; D = 1; S = 0; //F
        #20 Clock = 1; D = 0; S = 0;
        #20 Clock = 1; D = 1; S = 0;
        #20 Clock = 1; D = 1; S = 1;
        #20 Clock = 1; D = 1; S = 1; //J
        #20 Clock = 0; D = 1; S = 1;
        #20 Clock = 0; D = 0; S = 1;
        #20 Clock = 0; D = 0; S = 1;
        #20 Clock = 0; D = 0; S = 1;
        #20 Clock = 0; D = 0; S = 0; //0
        end
endmodule
```

5-2-2 代码和仿真代码



5-2-2 仿真截图

实验总结：

本次实验我掌握了 Testbench 的设计方法，也学习了 function 和 task 的使用。