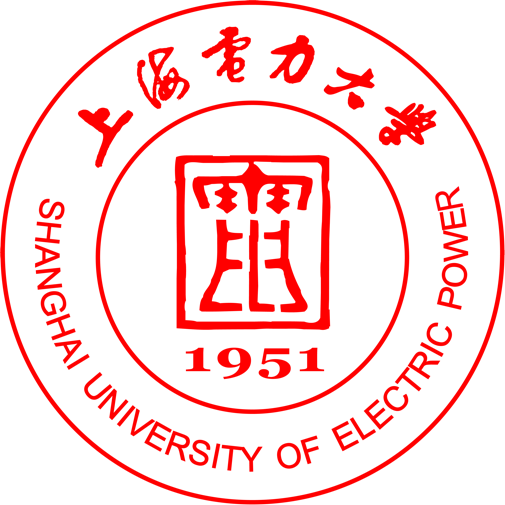
**上 海 电 力 大 学**

《数字集成电路设计与分析》实验报告



**实验题目：**  实验二 实现全加器功能代码、测试代码

**专 业：**

**班 级**  **学号**

**姓 名**

**时 间**  2023-09-28

1. 实验目的

2、完成二选一功能块的行为和结构描述，以及测试程序的编写

3、熟练掌握VI编辑器，并用VCS调试验证设计程序的正确性

1. 实验任务及要求

用VI编辑器完成二选一的源程序、测试程序的编写，并用VCS仿真验证设计的正确性

1. 实验内容及步骤

1、实验的源代码

module full\_adder (

  // module head: verillog-2001 format

  input  wire a\_in,

  input  wire b\_in,

  input  wire c\_in,     // carry in

  output wire sum\_out,

  output wire c\_out     // carry out

);

  // method 1 Gate Level describe方法1

  assign sum\_out = a\_in ^ b\_in ^ c\_in;//得到和

  assign c\_out   = (a\_in & b\_in) | (b\_in & c\_in) | (a\_in & c\_in);//得到进位

endmodule

2、实验的测试代码

module full\_adder\_tb;

  // drive the input port with the reg type

  reg ain, bin, cin;

  // sample the output port with the wire type

  wire sumout, cout;

//例化DUT

  full\_adder u\_full\_adder(

    // task 1. how to create an instance

    // module head: verillog-2001 format

    /\*input  wire \*/ .a\_in   (ain),

    /\*input  wire \*/ .b\_in   (bin),

    /\*input  wire \*/ .c\_in   (cin),     // carry in

    /\*output wire \*/ .sum\_out(sumout),

    /\*output wire \*/ .c\_out  (cout)   // carry out

  );

  // behavior of the adder can be synthesizable

  // "assign" means connectivity

  // assign {c\_out, sum\_out} = a\_in + b\_in + c\_in;

  //task 2. clock and reset generator

  parameter CLK\_PERIOD = 20;

  reg clk, reset\_n; // reset\_n : active low

  initial begin  //initial块，时钟激励波形生成

    clk = 0;

    forever begin

      #(CLK\_PERIOD/2) clk = ~clk;

    end

  end

  initial begin //initial块 ，复位激励波形生成

    reset\_n = 0;

    #100

    reset\_n = 1;

  end

 //task 3. drive the stimulus and capture the response

  //here is a testcase

  initial begin  //过程块2，顺序给其余输入激励（非 clock、reset）加上信号数值

    #110 ain = 0; bin = 0;  cin = 0;  // 00

    #20   ain = 0; bin = 1;  cin = 0;  // 01

    #20   ain = 1; bin = 0;  cin = 0;  // 01

    #20   ain = 1; bin = 1;  cin = 0;  // 10

    #20   ain = 0; bin = 0;  cin = 1;  // 01

    #20   ain = 0; bin = 1;  cin = 1;  // 10

    #20   ain = 1; bin = 0;  cin = 1;  // 10

    #20   ain = 1; bin = 1;  cin = 1;  // 11

    //#20  ain = 1; bin = 1;  cin = 0;  // 10

    #50  $finish;  // here is a system task which can stop the simulation

  end

  // task 4. check the result    //检查结果

  always @ (posedge clk) begin  //always过程块

    if (!reset\_n) begin

      $display("%t:%m: resetting ...",$time); // counter 5 clock

    end

    else begin

      $display("%t:%m: resetting finish!", $time); // the 6th clock

    end

  end

  initial begin   //initial 过程块2，顺序检查，如果结果不正确，则利用display系统函数打印出ERROR

    #115 if ({cout,sumout} != 2'b00) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b01) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b01) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b10) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b01) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b10) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b10) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b11) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

    #20  if ({cout,sumout} != 2'b10) $display("Error: {cout,sumout}=%b,ain=%b, bin=%b, cin=%b",{cout,sumout}, ain, bin, cin);

  end

  // task 5. dump waveform with the compile option -debug\_all  //转储波形文件

  initial begin

    $vcdpluson;  // vcdplus.vpd格式的波形文件

  end

endmodule

3、VCS仿真结果

（用截图的方法把VCS仿真结果拷贝下来）



图1 ./simv -l sim.log

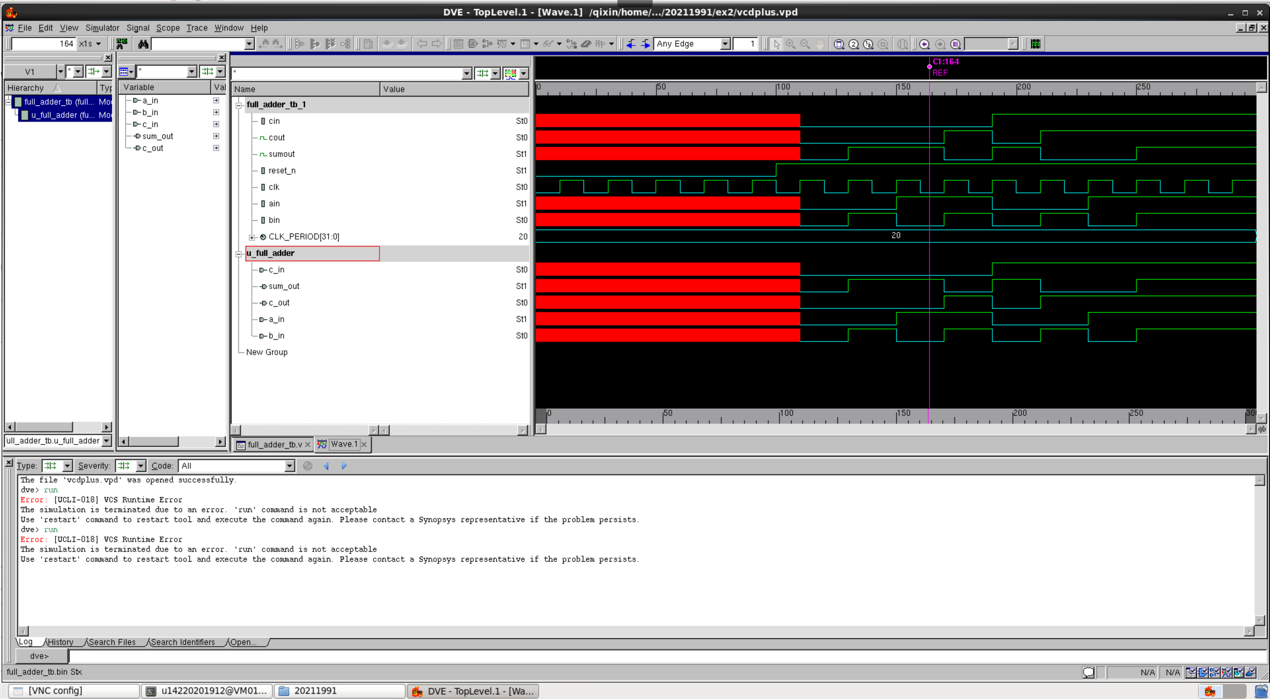


图2 dve -vpd vcdplus.vpd &vd

1. 实验总结

vcs -debug\_all full\_adder.v full\_adder\_tb.v -l com.log

./simv -l sim.log

dve -vpd vcdplus.vpd &vd