EEC281 Homework 2

1. [20 pts] The purpose of this problem is to familiarize you with the synthesis process and to give you a rough feeling for the size of a few simple circuits in our standard cell library's technology. Copy the files from the DC tutorial (see link on the main EEC281 page) to get started. Synthesize the following circuits and report their total cell area. Do not include registers (flip-flops). Also, do not declare any wires or registers as "signed", but assume words are all 2's complement unless stated otherwise. No need to simulate, but your verilog must compile correctly (run "make check"). Also, for this problem, do not worry if designs do not meet timing (negative slack time).

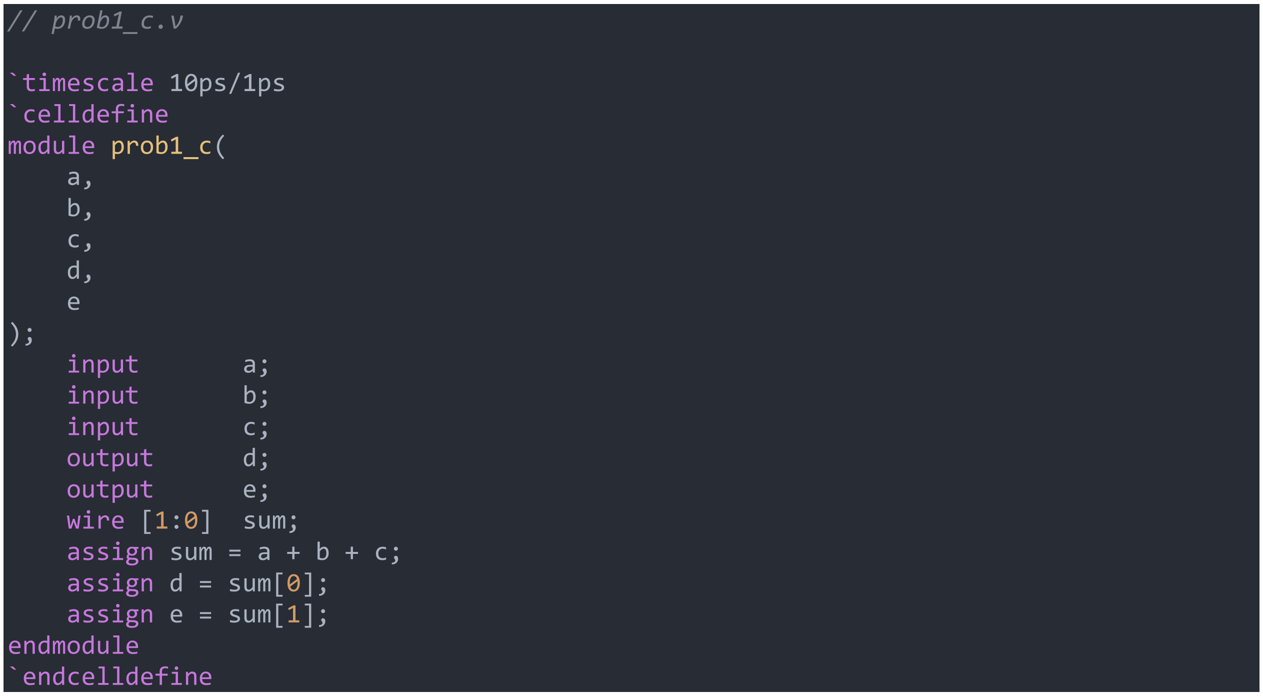
a) [2 pts] bitwise NOR of two 12-bit numbers (12-bit output)

```
// prob1_a.v

`timescale 10ps/1ps
`celldefine
module prob1_a(
    a,
    b,
    c
);
    input [11:0]    a;
    input [11:0]    b;
    output [11:0]   c;
    assign c = ~(a | b);
endmodule
`endcelldefine
```

b) [2 pts] One 3:2 adder using verilog "&" "|", "^", "~". Draw your circuit and the circuit output by DC.

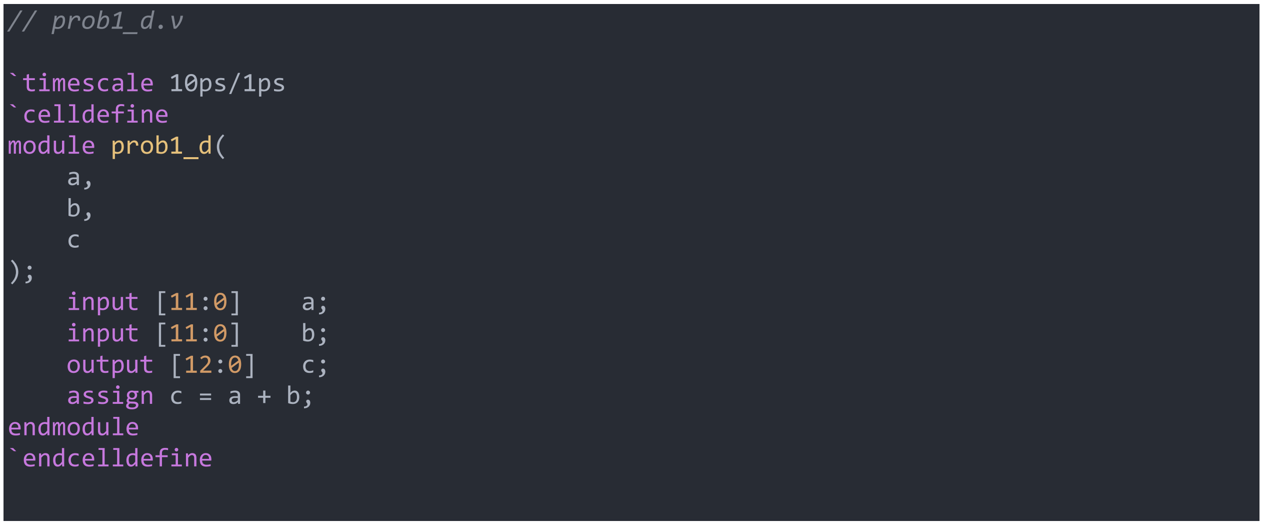
```
// prob1_b.v

`timescale 10ps/1ps
`celldefine
module prob1_b(
    a,
    b,
    c,
    d,
    e
);
    input   a;
    input   b;
    input   c;
    output  d;
    output  e;
    assign d = a ^ b ^ c;
    assign e = (a & b) | (a & c) | (b & c);
endmodule
`endcelldefine
```

c) One 3:2 adder using verilog "+".



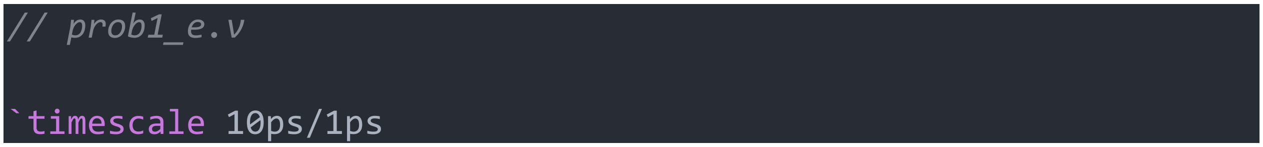
```verilog
// prob1_c.v

`timescale 10ps/1ps
`celldefine
module prob1_c(
    a,
    b,
    c,
    d,
    e
);
    input       a;
    input       b;
    input       c;
    output      d;
    output      e;
    wire [1:0]  sum;
    assign sum = a + b + c;
    assign d = sum[0];
    assign e = sum[1];
endmodule
`endcelldefine
```

d) 12-bit carry-propagate adder (13-bit output). Use "+" in verilog. Write the circuit's maximum delay in your table.

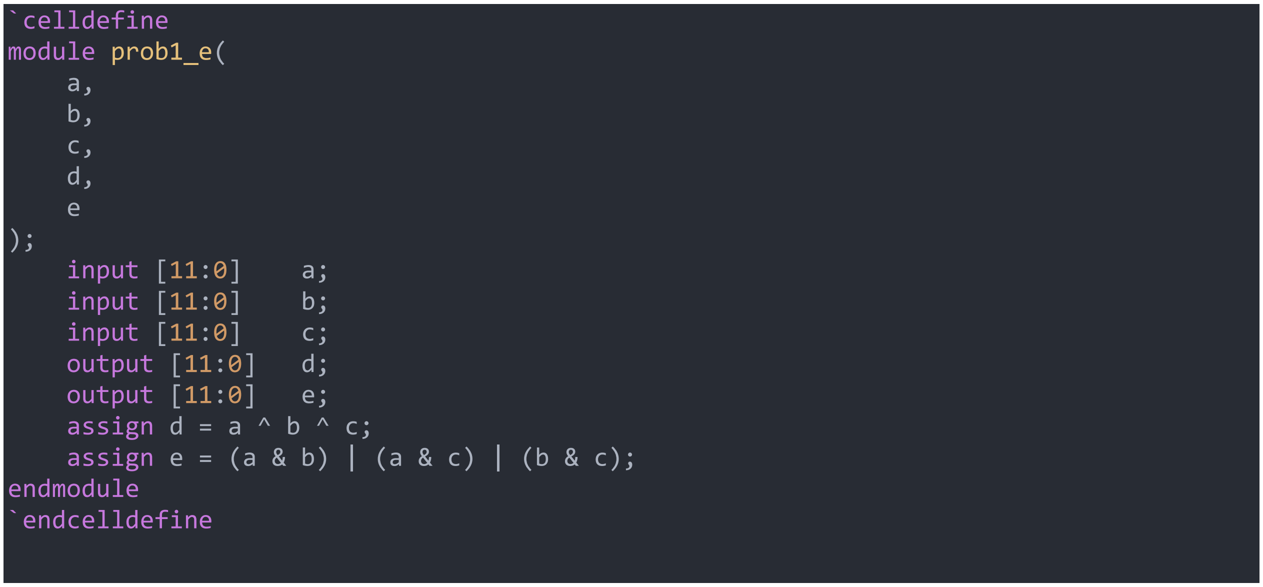


```verilog
// prob1_d.v

`timescale 10ps/1ps
`celldefine
module prob1_d(
    a,
    b,
    c
);
    input [11:0]    a;
    input [11:0]    b;
    output [12:0]   c;
    assign c = a + b;
endmodule
`endcelldefine
```

e) 12-bit carry-save adder that adds three 12-bit words into a single carry-save output. Write the circuit's maximum delay in your table.

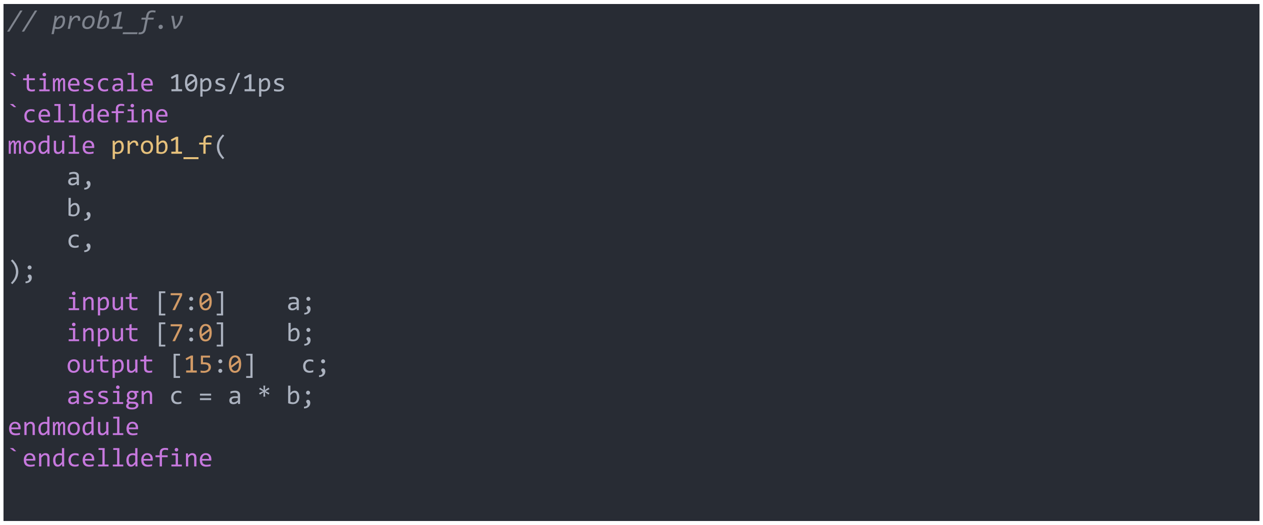


```verilog
// prob1_e.v

`timescale 10ps/1ps
```



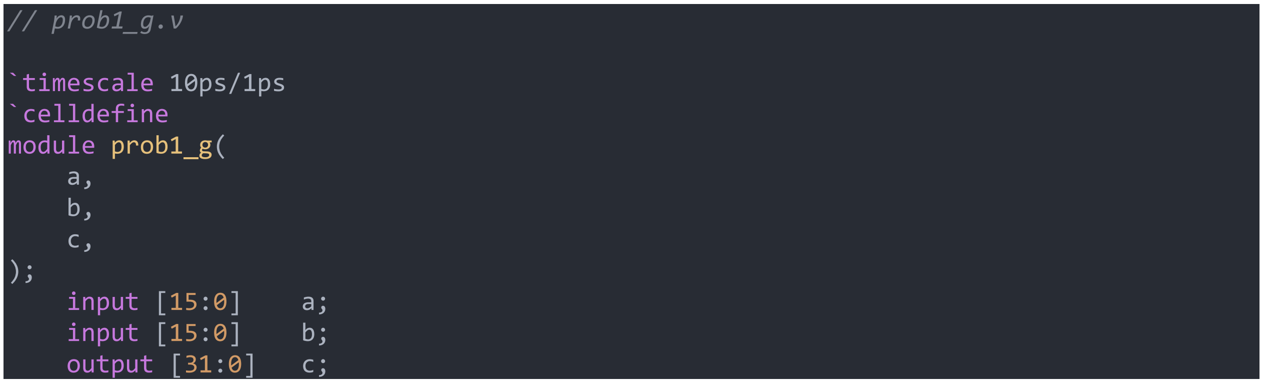
```
`celldefine
module prob1_e(
    a,
    b,
    c,
    d,
    e
);
    input [11:0]    a;
    input [11:0]    b;
    input [11:0]    c;
    output [11:0]   d;
    output [11:0]   e;
    assign d = a ^ b ^ c;
    assign e = (a & b) | (a & c) | (b & c);
endmodule
`endcelldefine
```

f) 8-bit x 8-bit unsigned multiplier (16-bit output). Use "*" in verilog.



```
// prob1_f.v

`timescale 10ps/1ps
`celldefine
module prob1_f(
    a,
    b,
    c,
);
    input [7:0]    a;
    input [7:0]    b;
    output [15:0]   c;
    assign c = a * b;
endmodule
`endcelldefine
```

g) 16-bit x 16-bit unsigned multiplier (32-bit output). Use "*" in verilog.



```
// prob1_g.v

`timescale 10ps/1ps
`celldefine
module prob1_g(
    a,
    b,
    c,
);
    input [15:0]    a;
    input [15:0]    b;
    output [31:0]   c;
    assign c = a * b;
endmodule
`endcelldefine
```

Problem	Area	Maximum dealy
a	9.576000	
b	5.054000	
c	5.054000	
d	49.476000	1.11
e	60.648000	0.18
f	285.152001	
g	1107.092006	

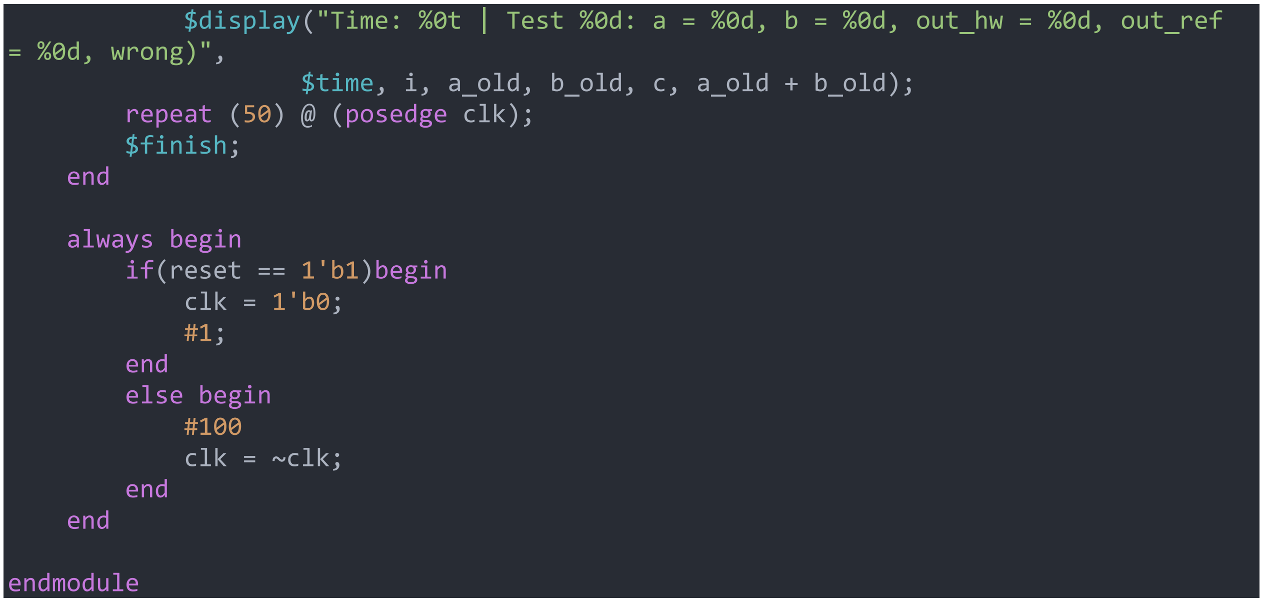
2. [20 pts] Build a ripple-carry adder with 12-bit inputs and 12-bit output using full adders from part 1(c). Register all inputs and outputs (to make synthesis timing accurate).

a) [10 pts] Write design in verilog, test with at least 15 test cases in one simulation with a unique set of inputs calculated each clock cycle. Verify using method ***(3).

```
// prob2.v

`timescale 10ps/1ps
`celldefine
module prob2(
    clk,
    a,
    b,
    c
);
    input           clk;
    input [11:0]    a;
    input [11:0]    b;
    output reg [11:0]   c;
    reg [11:0]      a_reg;
    reg [11:0]      b_reg;
    wire [11:0]     carry;
    wire [11:0]     sum;
    integer i;
    prob1_c u0(.a(a_reg[0]), .b(b_reg[0]), .c(1'b0), .d(sum[0]), .e(carry[0]));
    prob1_c u1(.a(a_reg[1]), .b(b_reg[1]), .c(carry[0]), .d(sum[1]), .e(carry[1]));
    prob1_c u2(.a(a_reg[2]), .b(b_reg[2]), .c(carry[1]), .d(sum[2]), .e(carry[2]));
    prob1_c u3(.a(a_reg[3]), .b(b_reg[3]), .c(carry[2]), .d(sum[3]), .e(carry[3]));
    prob1_c u4(.a(a_reg[4]), .b(b_reg[4]), .c(carry[3]), .d(sum[4]), .e(carry[4]));
    prob1_c u5(.a(a_reg[5]), .b(b_reg[5]), .c(carry[4]), .d(sum[5]), .e(carry[5]));
    prob1_c u6(.a(a_reg[6]), .b(b_reg[6]), .c(carry[5]), .d(sum[6]), .e(carry[6]));
    prob1_c u7(.a(a_reg[7]), .b(b_reg[7]), .c(carry[6]), .d(sum[7]), .e(carry[7]));
    prob1_c u8(.a(a_reg[8]), .b(b_reg[8]), .c(carry[7]), .d(sum[8]), .e(carry[8]));
    prob1_c u9(.a(a_reg[9]), .b(b_reg[9]), .c(carry[8]), .d(sum[9]), .e(carry[9]));
    prob1_c
u10(.a(a_reg[10]), .b(b_reg[10]), .c(carry[9]), .d(sum[10]), .e(carry[10]));
```

```
    prob1_c
u11(.a(a_reg[11]), .b(b_reg[11]), .c(carry[10]), .d(sum[11]), .e(carry[11]));

    always @(posedge clk) begin
        a_reg <= #1 a;
        b_reg <= #1 b;
        c <= #1 sum;
    end
endmodule
`endcelldefine
```

```
// prob2_tbench.v
`timescale 10ps/1ps
module prob2_tbench;
    reg         clk;
    reg         reset;
    reg [11:0]  a;
    reg [11:0]  b;
    wire [11:0] c;
    reg [11:0] a_old, b_old;
    integer i, seed;
    prob2 _prob2(.clk(clk), .a(a), .b(b), .c(c));
    initial begin
        reset = 1'b1;
        seed = 2;
        #500;
        reset = 1'b0;
        @(posedge clk); #10
        for (i = 0; i < 15; i = i + 1) begin
            a = $random(seed) % 4096;
            b = $random(seed) % 4096;
            @(posedge clk); #10
            if(i!=0)begin
                if (c == a_old + b_old)
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d,
out_ref = %0d, correct)",
                            $time, i, a_old, b_old, c, a_old + b_old);
                else
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d,
out_ref = %0d, wrong)",
                            $time, i, a_old, b_old, c, a_old + b_old);
            end
            a_old = a;
            b_old = b;
        end
        @(posedge clk); #10
        if (c == a_old + b_old)
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, correct)",
                    $time, i, a_old, b_old, c, a_old + b_old);
        else
```

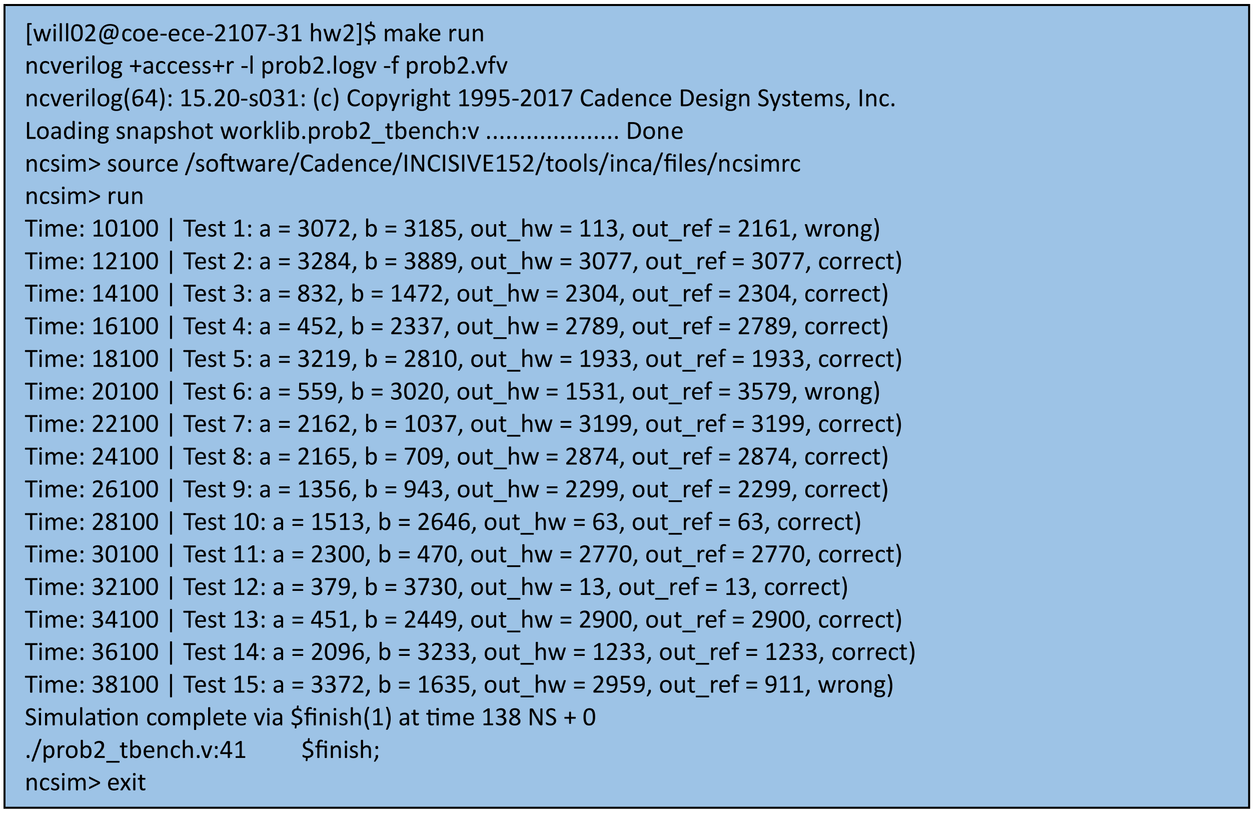


```
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, wrong)",
                    $time, i, a_old, b_old, c, a_old + b_old);
        repeat (50) @ (posedge clk);
        $finish;
    end

    always begin
        if(reset == 1'b1)begin
            clk = 1'b0;
            #1;
        end
        else begin
            #100
            clk = ~clk;
        end
    end

endmodule
```

Pass:

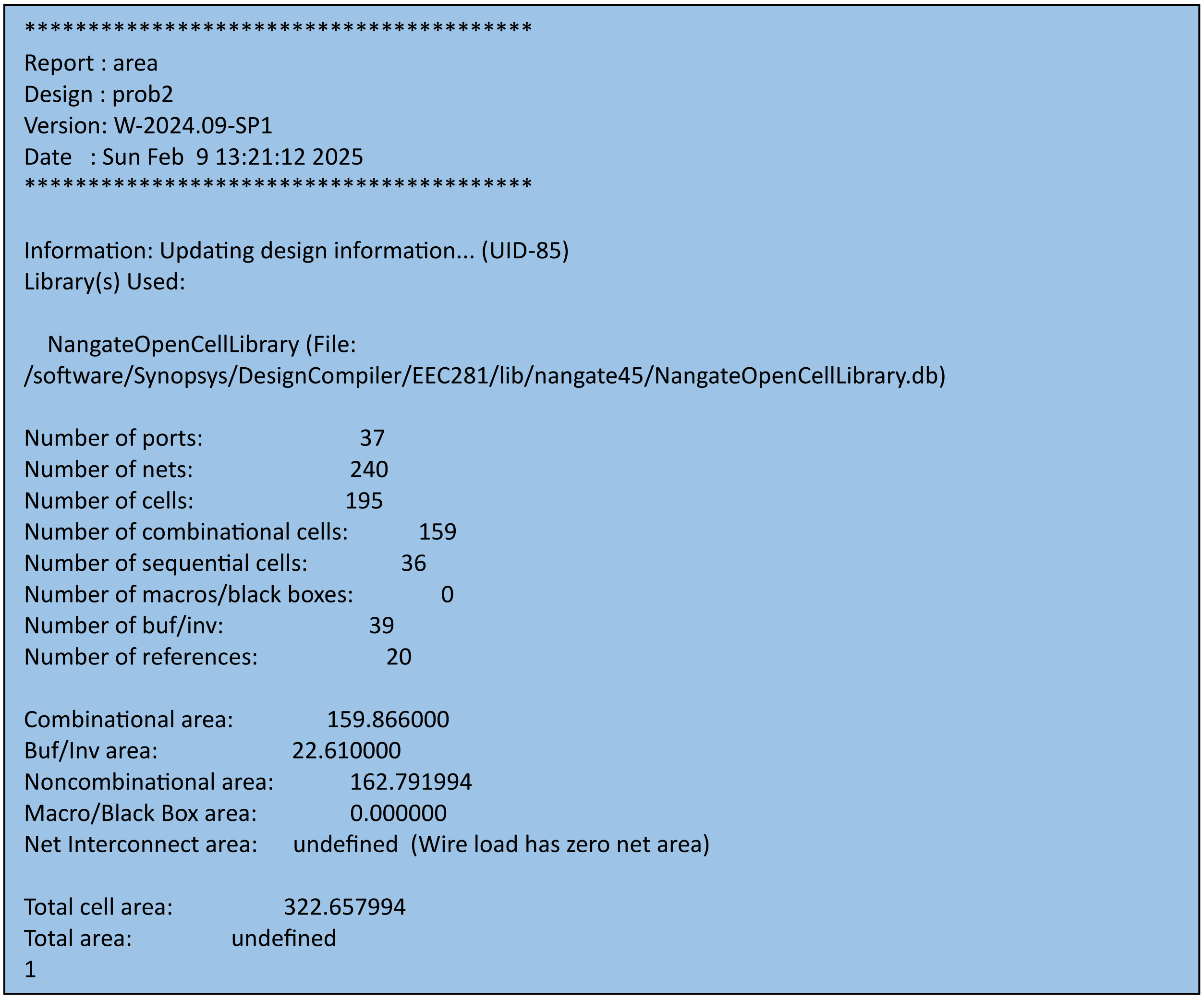
```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob2.logv -f prob2.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob2_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 10100 | Test 1: a = 3072, b = 3185, out_hw = 2161, out_ref = 2161, correct)
Time: 12100 | Test 2: a = 3284, b = 3889, out_hw = 3077, out_ref = 3077, correct)
Time: 14100 | Test 3: a = 832, b = 1472, out_hw = 2304, out_ref = 2304, correct)
Time: 16100 | Test 4: a = 452, b = 2337, out_hw = 2789, out_ref = 2789, correct)
Time: 18100 | Test 5: a = 3219, b = 2810, out_hw = 1933, out_ref = 1933, correct)
Time: 20100 | Test 6: a = 559, b = 3020, out_hw = 3579, out_ref = 3579, correct)
Time: 22100 | Test 7: a = 2162, b = 1037, out_hw = 3199, out_ref = 3199, correct)
Time: 24100 | Test 8: a = 2165, b = 709, out_hw = 2874, out_ref = 2874, correct)
Time: 26100 | Test 9: a = 1356, b = 943, out_hw = 2299, out_ref = 2299, correct)
Time: 28100 | Test 10: a = 1513, b = 2646, out_hw = 63, out_ref = 63, correct)
Time: 30100 | Test 11: a = 2300, b = 470, out_hw = 2770, out_ref = 2770, correct)
Time: 32100 | Test 12: a = 379, b = 3730, out_hw = 13, out_ref = 13, correct)
Time: 34100 | Test 13: a = 451, b = 2449, out_hw = 2900, out_ref = 2900, correct)
Time: 36100 | Test 14: a = 2096, b = 3233, out_hw = 1233, out_ref = 1233, correct)
Time: 38100 | Test 15: a = 3372, b = 1635, out_hw = 911, out_ref = 911, correct)
Simulation complete via $finish(1) at time 138 NS + 0
./prob2_tbench.v:41         $finish;
ncsim> exit
```

Modified:



```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob2.logv -f prob2.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob2_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 10100 | Test 1: a = 3072, b = 3185, out_hw = 113, out_ref = 2161, wrong)
Time: 12100 | Test 2: a = 3284, b = 3889, out_hw = 3077, out_ref = 3077, correct)
Time: 14100 | Test 3: a = 832, b = 1472, out_hw = 2304, out_ref = 2304, correct)
Time: 16100 | Test 4: a = 452, b = 2337, out_hw = 2789, out_ref = 2789, correct)
Time: 18100 | Test 5: a = 3219, b = 2810, out_hw = 1933, out_ref = 1933, correct)
Time: 20100 | Test 6: a = 559, b = 3020, out_hw = 1531, out_ref = 3579, wrong)
Time: 22100 | Test 7: a = 2162, b = 1037, out_hw = 3199, out_ref = 3199, correct)
Time: 24100 | Test 8: a = 2165, b = 709, out_hw = 2874, out_ref = 2874, correct)
Time: 26100 | Test 9: a = 1356, b = 943, out_hw = 2299, out_ref = 2299, correct)
Time: 28100 | Test 10: a = 1513, b = 2646, out_hw = 63, out_ref = 63, correct)
Time: 30100 | Test 11: a = 2300, b = 470, out_hw = 2770, out_ref = 2770, correct)
Time: 32100 | Test 12: a = 379, b = 3730, out_hw = 13, out_ref = 13, correct)
Time: 34100 | Test 13: a = 451, b = 2449, out_hw = 2900, out_ref = 2900, correct)
Time: 36100 | Test 14: a = 2096, b = 3233, out_hw = 1233, out_ref = 1233, correct)
Time: 38100 | Test 15: a = 3372, b = 1635, out_hw = 2959, out_ref = 911, wrong)
Simulation complete via $finish(1) at time 138 NS + 0
./prob2_tbench.v:41        $finish;
ncsim> exit
```

b) [10 pts] Synthesize the design with a high clock frequency to find the maximum clock rate. State the maximum clock rate and corresponding area. Submit *.area and *.tim (longest path only) reports only.

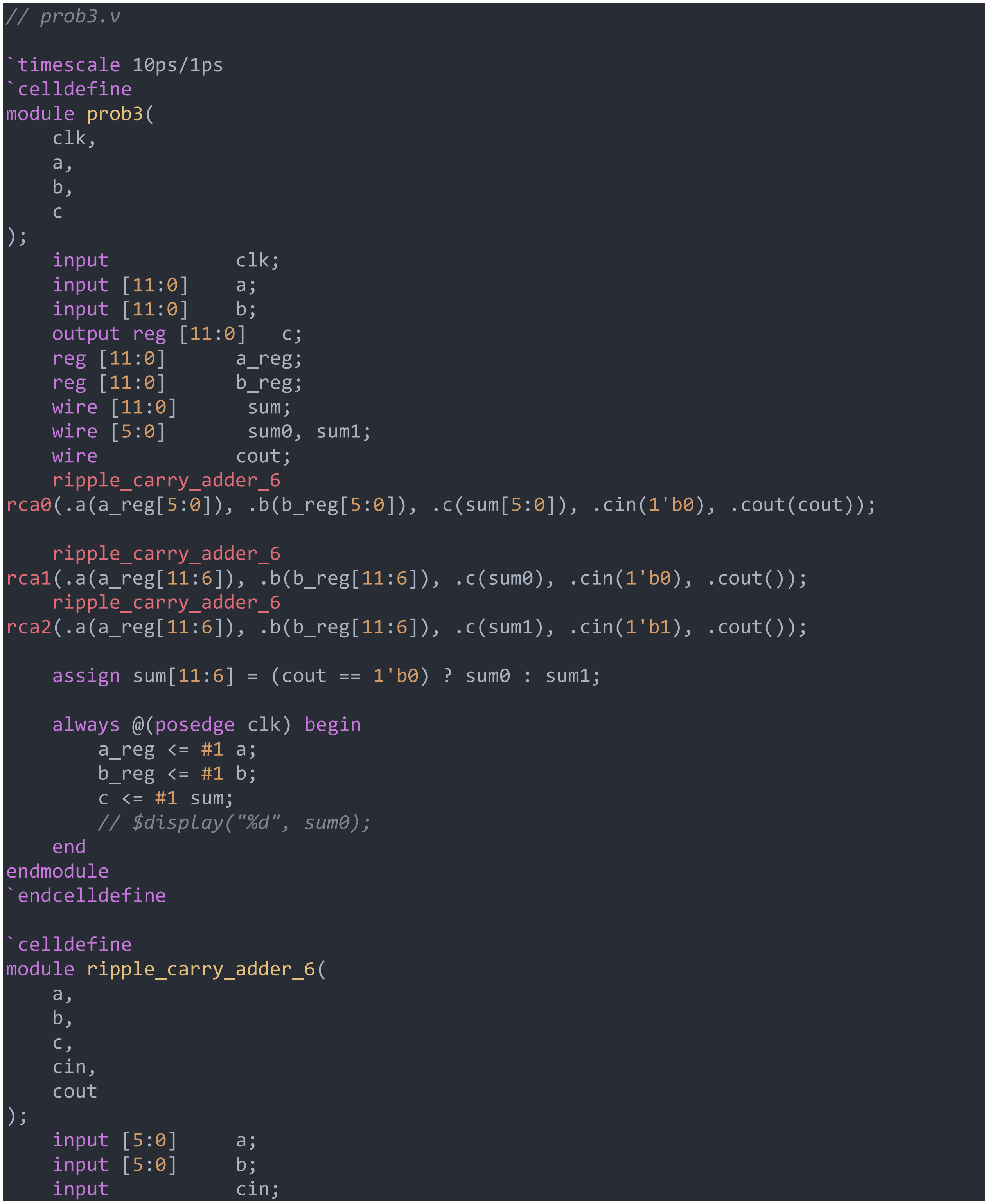


```
****************************************
Report : area
Design : prob2
Version: W-2024.09-SP1
Date   : Sun Feb  9 13:21:12 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db)

Number of ports:                           37
Number of nets:                           240
Number of cells:                          195
Number of combinational cells:            159
Number of sequential cells:                36
Number of macros/black boxes:               0
Number of buf/inv:                         39
Number of references:                      20

Combinational area:              159.866000
Buf/Inv area:                     22.610000
Noncombinational area:           162.791994
Macro/Black Box area:              0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 322.657994
Total area:                 undefined
1
```

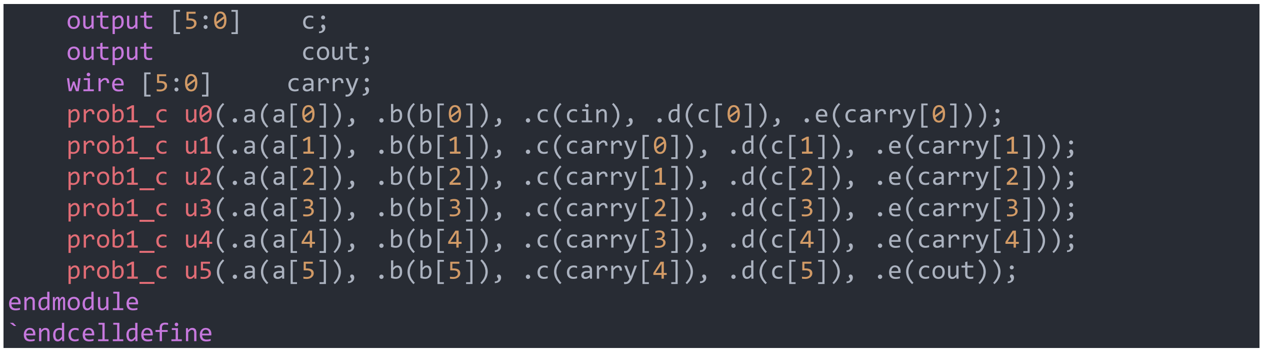
```
****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob2
Version: W-2024.09-SP1
Date   : Sun Feb  9 13:21:13 2025
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: b_reg_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: c_reg[7] (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob2              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  b_reg_reg[0]/CK (DFF_X1)                 0.00       0.00 r
  b_reg_reg[0]/Q (DFF_X1)                  0.09       0.09 f
  U76/ZN (OAI22_X1)                        0.06       0.15 r
  U63/ZN (NOR2_X1)                         0.03       0.18 f
  U98/ZN (NAND3_X1)                        0.03       0.21 r
  U40/ZN (AND2_X1)                         0.07       0.29 r
  U107/ZN (NAND3_X1)                       0.05       0.33 f
  U108/ZN (OAI211_X1)                      0.04       0.37 r
  U109/ZN (INV_X1)                         0.02       0.40 f
  U117/ZN (OAI22_X1)                       0.05       0.45 r
  U118/ZN (OAI21_X1)                       0.05       0.50 f
  U65/ZN (XNOR2_X1)                        0.06       0.56 f
  U14/ZN (OAI22_X1)                        0.05       0.61 r
  U33/ZN (XNOR2_X1)                        0.07       0.68 r
  U42/ZN (XNOR2_X1)                        0.06       0.74 r
  c_reg[7]/D (DFF_X1)                      0.01       0.75 r
  data arrival time                                   0.75

  clock clk (rise edge)                    0.10       0.10
  clock network delay (ideal)              0.00       0.10
  clock uncertainty                        0.00       0.09
  c_reg[7]/CK (DFF_X1)                     0.00       0.09 r
  library setup time                      -0.03       0.06
  data required time                                  0.06
  -----------------------------------------------------------
  data required time                                  0.06
  data arrival time                                  -0.75
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.69
```

Maximum clock rate = 1 / (0.1 – (-0.69)) = 1.266 GHz

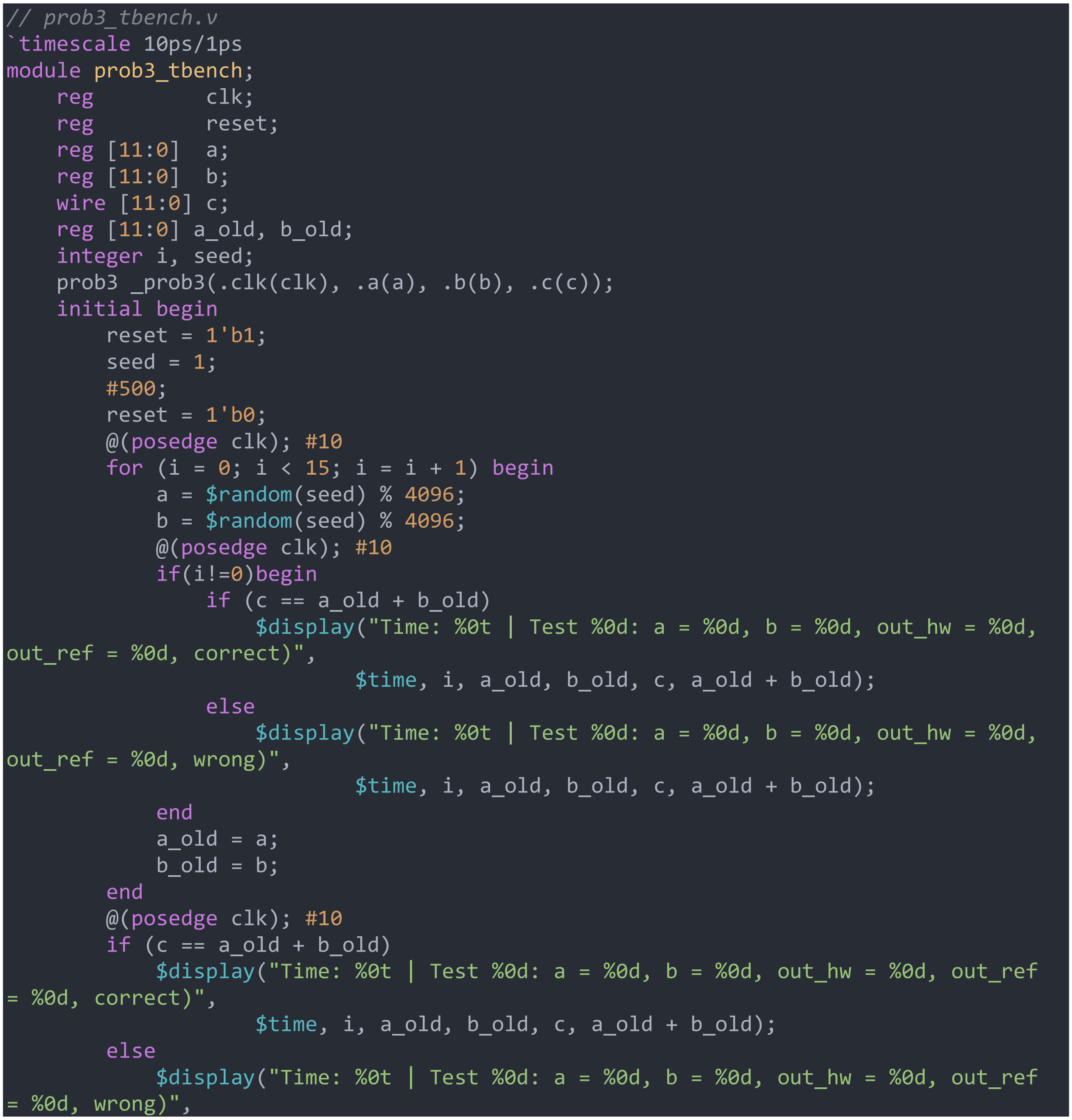
3. [30+10 pts] Repeat Problem 2(a)(b) with a carry-select adder composed of two 6-bit sections.



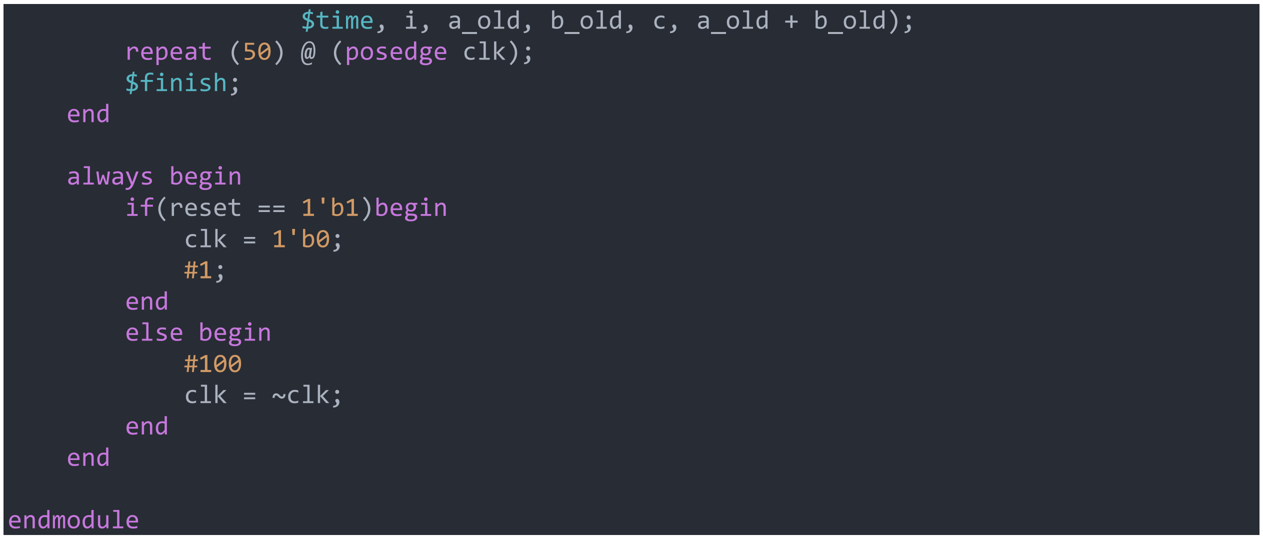
```
// prob3.v

`timescale 10ps/1ps
`celldefine
module prob3(
    clk,
    a,
    b,
    c
);
    input           clk;
    input [11:0]    a;
    input [11:0]    b;
    output reg [11:0]   c;
    reg [11:0]      a_reg;
    reg [11:0]      b_reg;
    wire [11:0]      sum;
    wire [5:0]       sum0, sum1;
    wire            cout;
    ripple_carry_adder_6
rca0(.a(a_reg[5:0]), .b(b_reg[5:0]), .c(sum[5:0]), .cin(1'b0), .cout(cout));

    ripple_carry_adder_6
rca1(.a(a_reg[11:6]), .b(b_reg[11:6]), .c(sum0), .cin(1'b0), .cout());
    ripple_carry_adder_6
rca2(.a(a_reg[11:6]), .b(b_reg[11:6]), .c(sum1), .cin(1'b1), .cout());

    assign sum[11:6] = (cout == 1'b0) ? sum0 : sum1;

    always @(posedge clk) begin
        a_reg <= #1 a;
        b_reg <= #1 b;
        c <= #1 sum;
        // $display("%d", sum0);
    end
endmodule
`endcelldefine

`celldefine
module ripple_carry_adder_6(
    a,
    b,
    c,
    cin,
    cout
);
    input [5:0]     a;
    input [5:0]     b;
    input           cin;
```



```
    output [5:0]    c;
    output          cout;
    wire [5:0]     carry;
    prob1_c u0(.a(a[0]), .b(b[0]), .c(cin), .d(c[0]), .e(carry[0]));
    prob1_c u1(.a(a[1]), .b(b[1]), .c(carry[0]), .d(c[1]), .e(carry[1]));
    prob1_c u2(.a(a[2]), .b(b[2]), .c(carry[1]), .d(c[2]), .e(carry[2]));
    prob1_c u3(.a(a[3]), .b(b[3]), .c(carry[2]), .d(c[3]), .e(carry[3]));
    prob1_c u4(.a(a[4]), .b(b[4]), .c(carry[3]), .d(c[4]), .e(carry[4]));
    prob1_c u5(.a(a[5]), .b(b[5]), .c(carry[4]), .d(c[5]), .e(cout));
endmodule
`endcelldefine
```



```
// prob3_tbench.v
`timescale 10ps/1ps
module prob3_tbench;
    reg         clk;
    reg         reset;
    reg [11:0]  a;
    reg [11:0]  b;
    wire [11:0] c;
    reg [11:0] a_old, b_old;
    integer i, seed;
    prob3 _prob3(.clk(clk), .a(a), .b(b), .c(c));
    initial begin
        reset = 1'b1;
        seed = 1;
        #500;
        reset = 1'b0;
        @(posedge clk); #10
        for (i = 0; i < 15; i = i + 1) begin
            a = $random(seed) % 4096;
            b = $random(seed) % 4096;
            @(posedge clk); #10
            if(i!=0)begin
                if (c == a_old + b_old)
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, 
out_ref = %0d, correct)",
                            $time, i, a_old, b_old, c, a_old + b_old);
                else
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, 
out_ref = %0d, wrong)",
                            $time, i, a_old, b_old, c, a_old + b_old);
            end
            a_old = a;
            b_old = b;
        end
        @(posedge clk); #10
        if (c == a_old + b_old)
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref 
= %0d, correct)",
                    $time, i, a_old, b_old, c, a_old + b_old);
        else
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref 
= %0d, wrong)",
```



```
                    $time, i, a_old, b_old, c, a_old + b_old);
        repeat (50) @ (posedge clk);
        $finish;
    end

    always begin
        if(reset == 1'b1)begin
            clk = 1'b0;
            #1;
        end
        else begin
            #100
            clk = ~clk;
        end
    end

endmodule
```

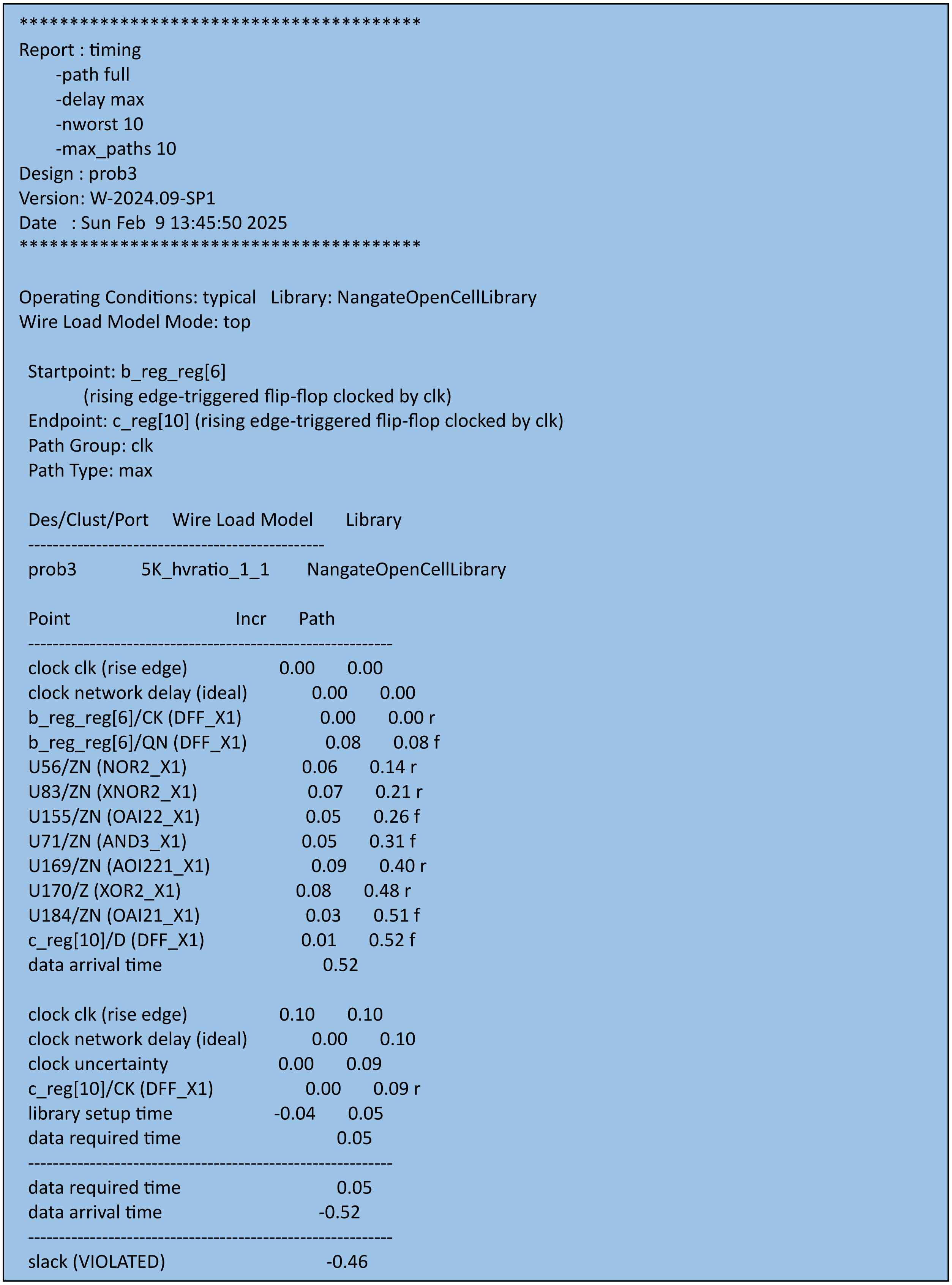
Pass:

```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob3.logv -f prob3.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob3_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 10100 | Test 1: a = 3584, b = 1080, out_hw = 568, out_ref = 568, correct)
Time: 12100 | Test 2: a = 2438, b = 3164, out_hw = 1506, out_ref = 1506, correct)
Time: 14100 | Test 3: a = 1998, b = 1991, out_hw = 3989, out_ref = 3989, correct)
Time: 16100 | Test 4: a = 3270, b = 755, out_hw = 4025, out_ref = 4025, correct)
Time: 18100 | Test 5: a = 1987, b = 2143, out_hw = 34, out_ref = 34, correct)
Time: 20100 | Test 6: a = 1351, b = 137, out_hw = 1488, out_ref = 1488, correct)
Time: 22100 | Test 7: a = 3198, b = 2117, out_hw = 1219, out_ref = 1219, correct)
Time: 24100 | Test 8: a = 93, b = 401, out_hw = 494, out_ref = 494, correct)
Time: 26100 | Test 9: a = 878, b = 655, out_hw = 1533, out_ref = 1533, correct)
Time: 28100 | Test 10: a = 3388, b = 713, out_hw = 5, out_ref = 5, correct)
Time: 30100 | Test 11: a = 1507, b = 732, out_hw = 2239, out_ref = 2239, correct)
Time: 32100 | Test 12: a = 556, b = 95, out_hw = 651, out_ref = 651, correct)
Time: 34100 | Test 13: a = 17, b = 1489, out_hw = 1506, out_ref = 1506, correct)
Time: 36100 | Test 14: a = 3840, b = 1233, out_hw = 977, out_ref = 977, correct)
Time: 38100 | Test 15: a = 511, b = 561, out_hw = 1072, out_ref = 1072, correct)
Simulation complete via $finish(1) at time 138 NS + 0
./prob3_tbench.v:41         $finish;
ncsim> exit
```

Modified:

```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob3.logv -f prob3.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob3_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 10100 | Test 1: a = 3584, b = 1080, out_hw = 568, out_ref = 568, correct)
Time: 12100 | Test 2: a = 2438, b = 3164, out_hw = 1506, out_ref = 1506, correct)
Time: 14100 | Test 3: a = 1998, b = 1991, out_hw = 3989, out_ref = 3989, correct)
Time: 16100 | Test 4: a = 3270, b = 755, out_hw = 4025, out_ref = 4025, correct)
Time: 18100 | Test 5: a = 1987, b = 2143, out_hw = 34, out_ref = 34, correct)
Time: 20100 | Test 6: a = 1351, b = 137, out_hw = 1488, out_ref = 1488, correct)
Time: 22100 | Test 7: a = 3198, b = 2117, out_hw = 1155, out_ref = 1219, wrong)
Time: 24100 | Test 8: a = 93, b = 401, out_hw = 494, out_ref = 494, correct)
Time: 26100 | Test 9: a = 878, b = 655, out_hw = 1533, out_ref = 1533, correct)
Time: 28100 | Test 10: a = 3388, b = 713, out_hw = 4037, out_ref = 5, wrong)
Time: 30100 | Test 11: a = 1507, b = 732, out_hw = 2239, out_ref = 2239, correct)
Time: 32100 | Test 12: a = 556, b = 95, out_hw = 587, out_ref = 651, wrong)
Time: 34100 | Test 13: a = 17, b = 1489, out_hw = 1506, out_ref = 1506, correct)
Time: 36100 | Test 14: a = 3840, b = 1233, out_hw = 977, out_ref = 977, correct)
Time: 38100 | Test 15: a = 511, b = 561, out_hw = 1008, out_ref = 1072, wrong)
Simulation complete via $finish(1) at time 138 NS + 0
./prob3_tbench.v:41         $finish;
ncsim> exit
```

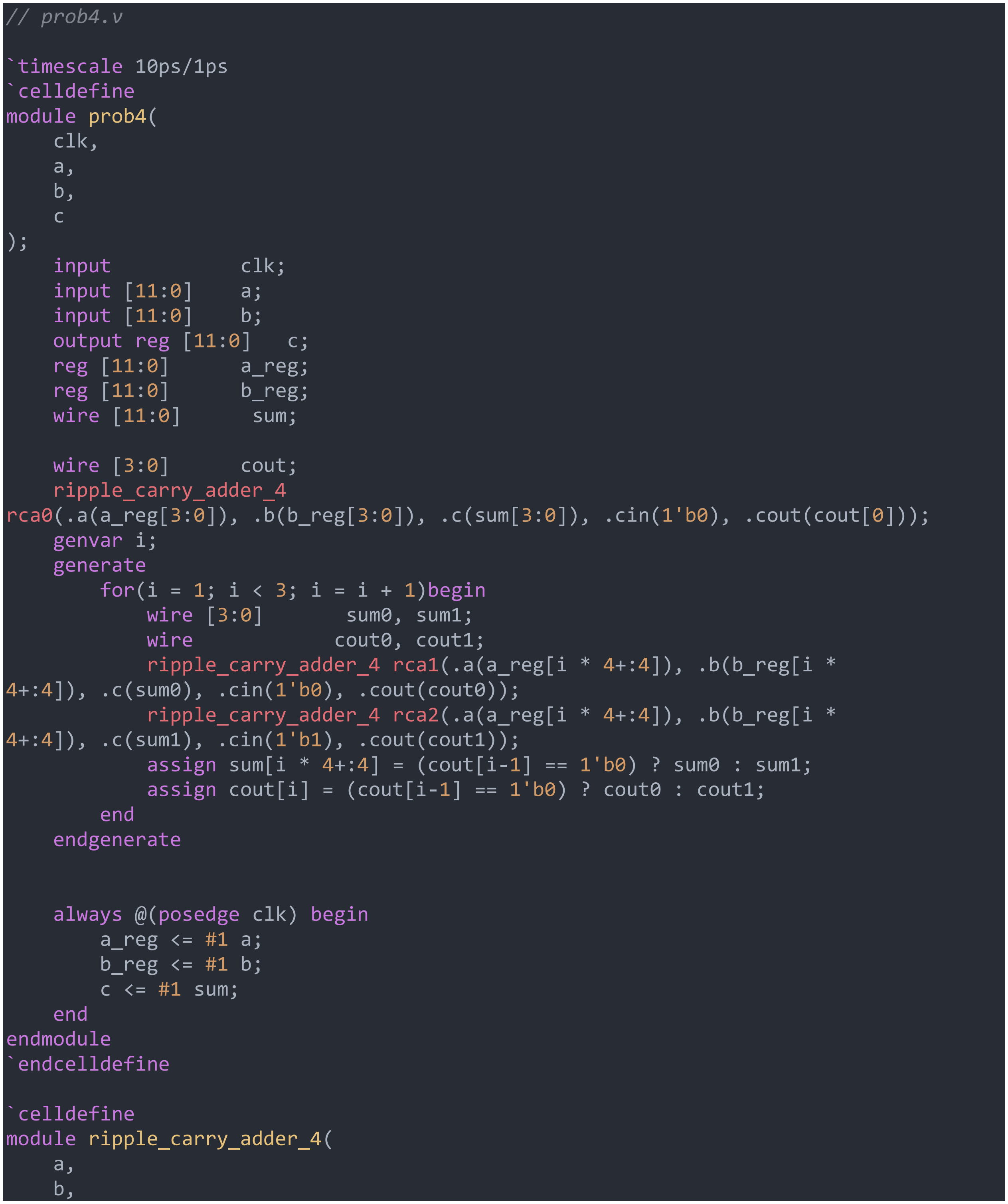
```
****************************************
Report : area
Design : prob3
Version: W-2024.09-SP1
Date   : Sun Feb  9 13:45:49 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db)

Number of ports:                           37
Number of nets:                           281
Number of cells:                          236
Number of combinational cells:            200
Number of sequential cells:                36
Number of macros/black boxes:               0
Number of buf/inv:                         47
Number of references:                      26

Combinational area:                207.746000
Buf/Inv area:                       27.930000
Noncombinational area:             166.249994
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                   373.995994
Total area:                 undefined
1
```



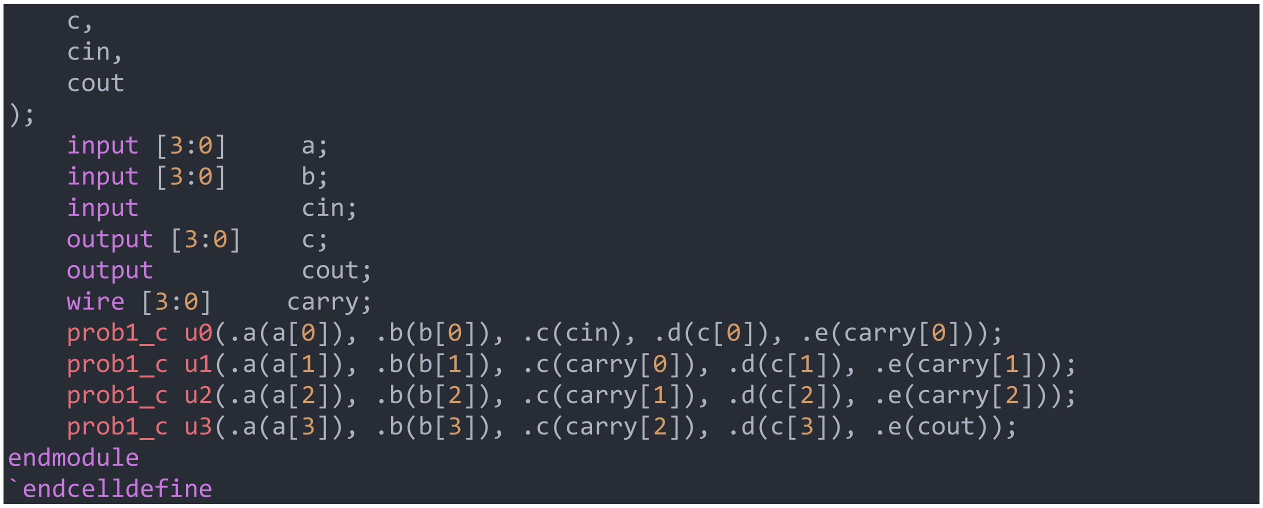
```
****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob3
Version: W-2024.09-SP1
Date   : Sun Feb  9 13:45:50 2025
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: b_reg_reg[6]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: c_reg[10] (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob3              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  b_reg_reg[6]/CK (DFF_X1)                 0.00       0.00 r
  b_reg_reg[6]/QN (DFF_X1)                 0.08       0.08 f
  U56/ZN (NOR2_X1)                         0.06       0.14 r
  U83/ZN (XNOR2_X1)                        0.07       0.21 r
  U155/ZN (OAI22_X1)                       0.05       0.26 f
  U71/ZN (AND3_X1)                         0.05       0.31 f
  U169/ZN (AOI221_X1)                      0.09       0.40 r
  U170/Z (XOR2_X1)                         0.08       0.48 r
  U184/ZN (OAI21_X1)                       0.03       0.51 f
  c_reg[10]/D (DFF_X1)                     0.01       0.52 f
  data arrival time                                   0.52

  clock clk (rise edge)                    0.10       0.10
  clock network delay (ideal)              0.00       0.10
  clock uncertainty                        0.00       0.09
  c_reg[10]/CK (DFF_X1)                    0.00       0.09 r
  library setup time                      -0.04       0.05
  data required time                                  0.05
  -----------------------------------------------------------
  data required time                                  0.05
  data arrival time                                  -0.52
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.46
```

Maximum clock rate = 1 / (0.1 – (-0.46)) = 1.786 GHz

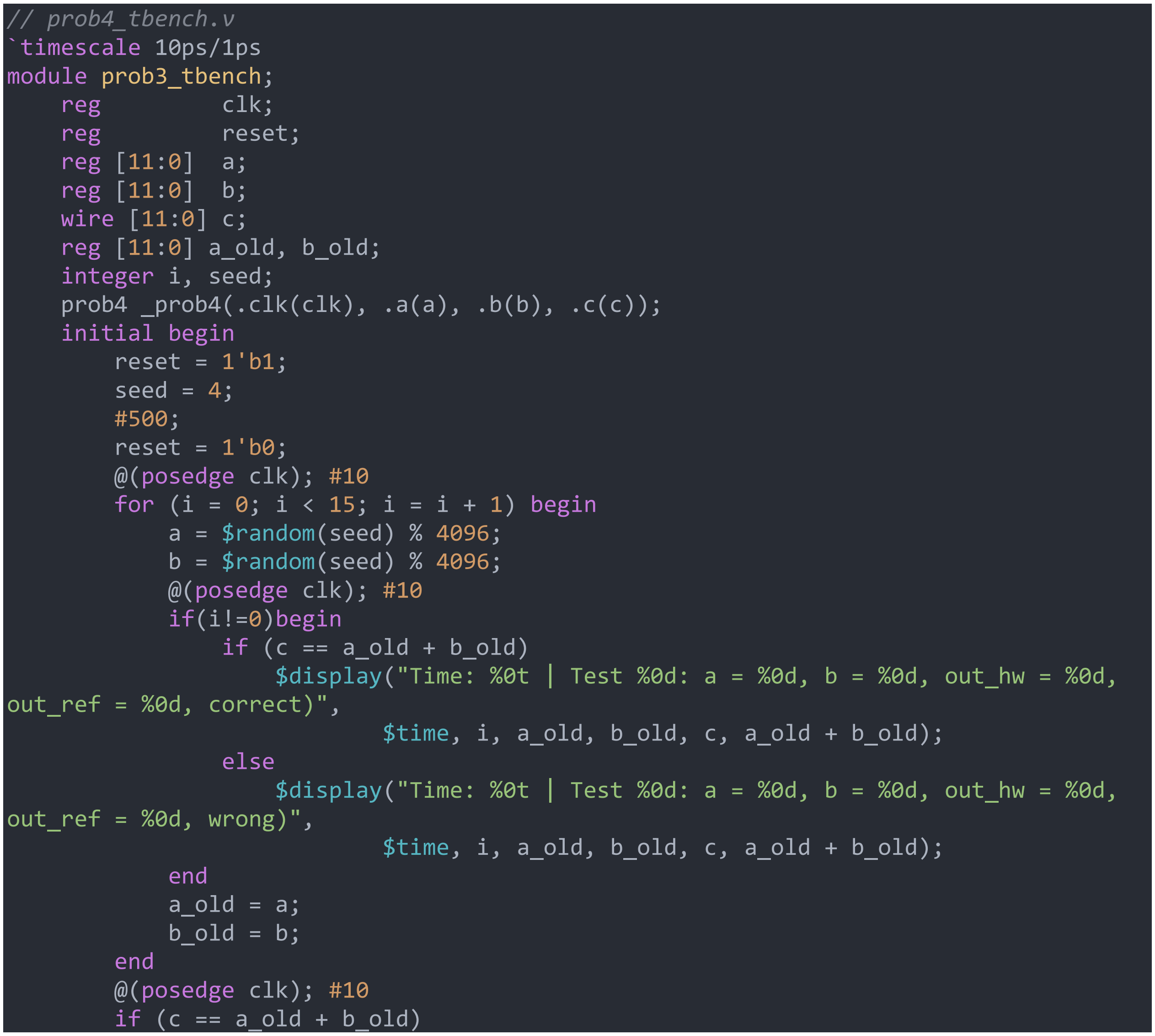
4. [30+10+10 pts] Repeat Problem 2(a)(b) with a carry-select adder composed of three sections whose widths are chosen to minimize delay.



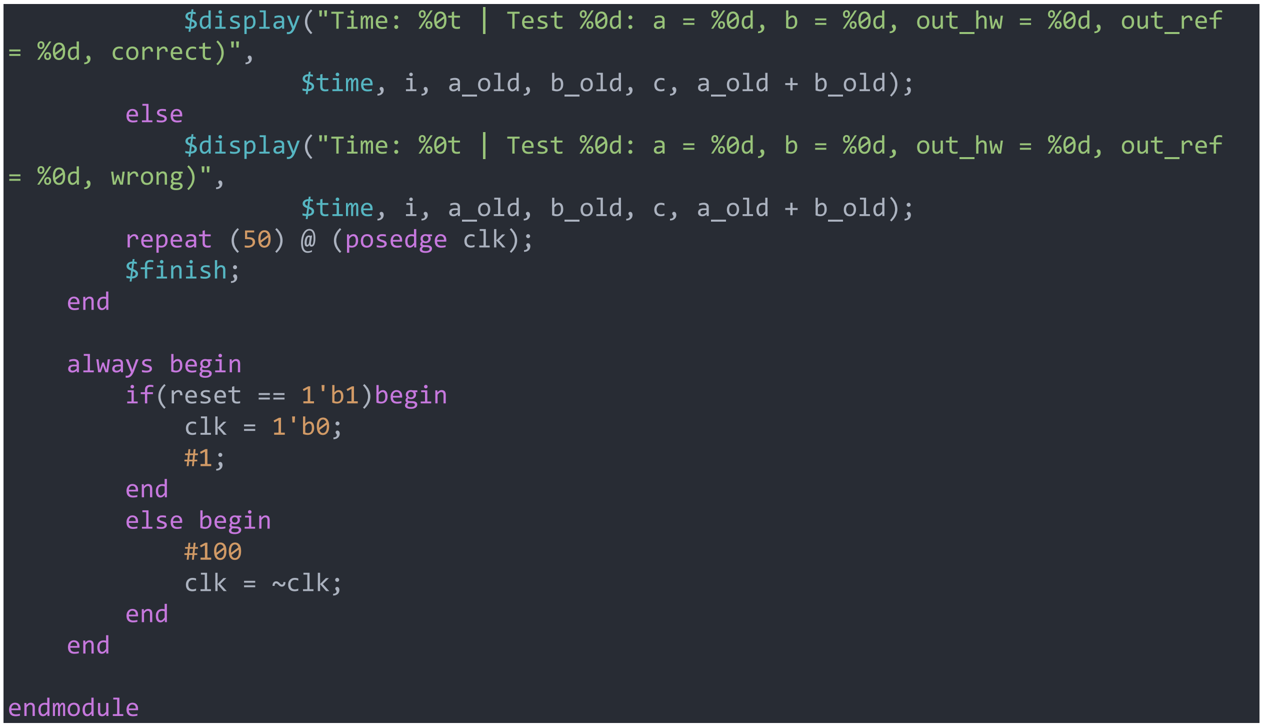
```
// prob4.v

`timescale 10ps/1ps
`celldefine
module prob4(
    clk,
    a,
    b,
    c
);
    input           clk;
    input [11:0]    a;
    input [11:0]    b;
    output reg [11:0]   c;
    reg [11:0]      a_reg;
    reg [11:0]      b_reg;
    wire [11:0]      sum;

    wire [3:0]      cout;
    ripple_carry_adder_4
rca0(.a(a_reg[3:0]), .b(b_reg[3:0]), .c(sum[3:0]), .cin(1'b0), .cout(cout[0]));
    genvar i;
    generate
        for(i = 1; i < 3; i = i + 1)begin
            wire [3:0]       sum0, sum1;
            wire            cout0, cout1;
            ripple_carry_adder_4 rca1(.a(a_reg[i * 4+:4]), .b(b_reg[i *
4+:4]), .c(sum0), .cin(1'b0), .cout(cout0));
            ripple_carry_adder_4 rca2(.a(a_reg[i * 4+:4]), .b(b_reg[i *
4+:4]), .c(sum1), .cin(1'b1), .cout(cout1));
            assign sum[i * 4+:4] = (cout[i-1] == 1'b0) ? sum0 : sum1;
            assign cout[i] = (cout[i-1] == 1'b0) ? cout0 : cout1;
        end
    endgenerate


    always @(posedge clk) begin
        a_reg <= #1 a;
        b_reg <= #1 b;
        c <= #1 sum;
    end
endmodule
`endcelldefine

`celldefine
module ripple_carry_adder_4(
    a,
    b,
```



```
    c,
    cin,
    cout
);
    input [3:0]     a;
    input [3:0]     b;
    input           cin;
    output [3:0]    c;
    output          cout;
    wire [3:0]     carry;
    prob1_c u0(.a(a[0]), .b(b[0]), .c(cin), .d(c[0]), .e(carry[0]));
    prob1_c u1(.a(a[1]), .b(b[1]), .c(carry[0]), .d(c[1]), .e(carry[1]));
    prob1_c u2(.a(a[2]), .b(b[2]), .c(carry[1]), .d(c[2]), .e(carry[2]));
    prob1_c u3(.a(a[3]), .b(b[3]), .c(carry[2]), .d(c[3]), .e(cout));
endmodule
`endcelldefine
```



```
// prob4_tbench.v
`timescale 10ps/1ps
module prob3_tbench;
    reg         clk;
    reg         reset;
    reg [11:0]  a;
    reg [11:0]  b;
    wire [11:0] c;
    reg [11:0] a_old, b_old;
    integer i, seed;
    prob4 _prob4(.clk(clk), .a(a), .b(b), .c(c));
    initial begin
        reset = 1'b1;
        seed = 4;
        #500;
        reset = 1'b0;
        @(posedge clk); #10
        for (i = 0; i < 15; i = i + 1) begin
            a = $random(seed) % 4096;
            b = $random(seed) % 4096;
            @(posedge clk); #10
            if(i!=0)begin
                if (c == a_old + b_old)
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d,
out_ref = %0d, correct)",
                            $time, i, a_old, b_old, c, a_old + b_old);
                else
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d,
out_ref = %0d, wrong)",
                            $time, i, a_old, b_old, c, a_old + b_old);
            end
            a_old = a;
            b_old = b;
        end
        @(posedge clk); #10
        if (c == a_old + b_old)
```

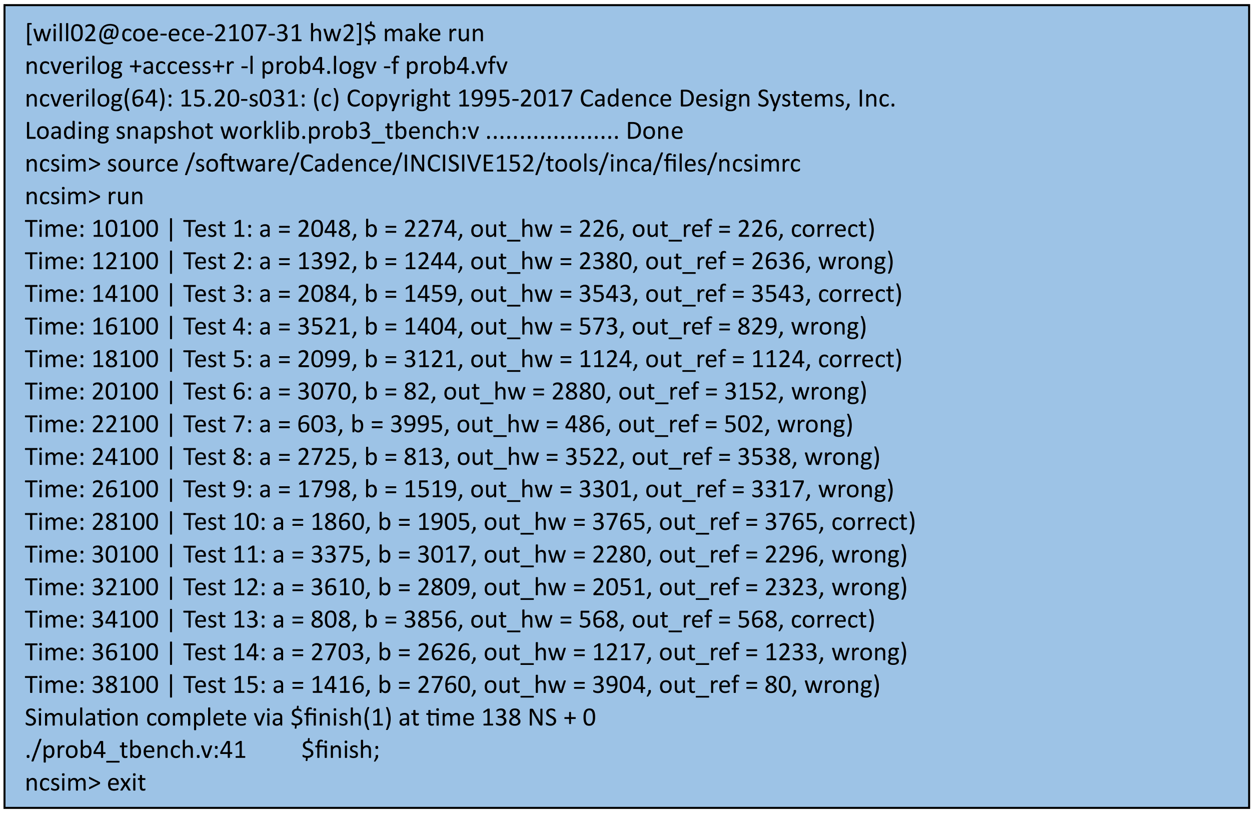


```
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, correct)",
                    $time, i, a_old, b_old, c, a_old + b_old);
        else
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, wrong)",
                    $time, i, a_old, b_old, c, a_old + b_old);
        repeat (50) @ (posedge clk);
        $finish;
    end

    always begin
        if(reset == 1'b1)begin
            clk = 1'b0;
            #1;
        end
        else begin
            #100
            clk = ~clk;
        end
    end

endmodule
```

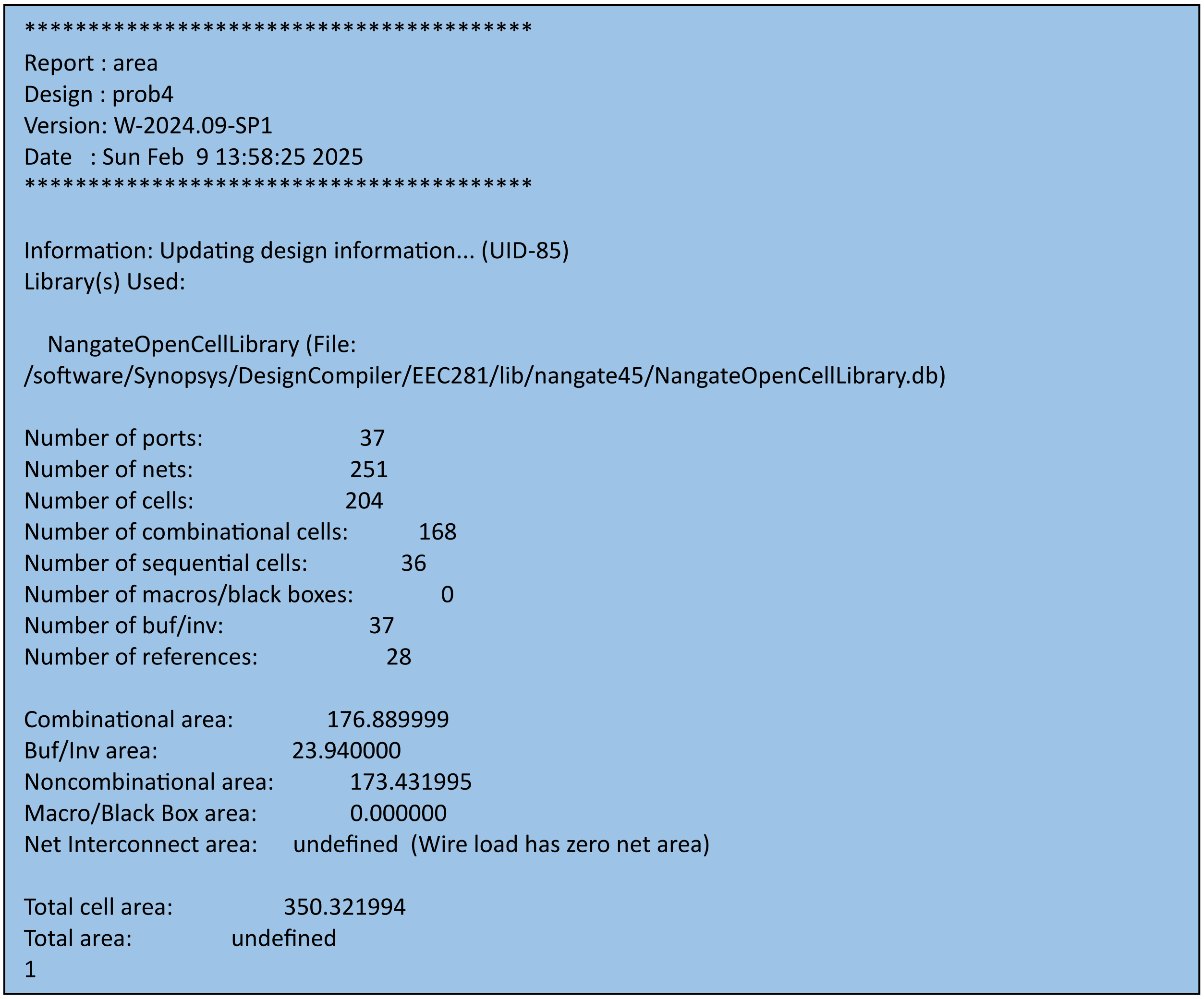
Pass:

```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob4.logv -f prob4.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob3_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 10100 | Test 1: a = 2048, b = 2274, out_hw = 226, out_ref = 226, correct)
Time: 12100 | Test 2: a = 1392, b = 1244, out_hw = 2636, out_ref = 2636, correct)
Time: 14100 | Test 3: a = 2084, b = 1459, out_hw = 3543, out_ref = 3543, correct)
Time: 16100 | Test 4: a = 3521, b = 1404, out_hw = 829, out_ref = 829, correct)
Time: 18100 | Test 5: a = 2099, b = 3121, out_hw = 1124, out_ref = 1124, correct)
Time: 20100 | Test 6: a = 3070, b = 82, out_hw = 3152, out_ref = 3152, correct)
Time: 22100 | Test 7: a = 603, b = 3995, out_hw = 502, out_ref = 502, correct)
Time: 24100 | Test 8: a = 2725, b = 813, out_hw = 3538, out_ref = 3538, correct)
Time: 26100 | Test 9: a = 1798, b = 1519, out_hw = 3317, out_ref = 3317, correct)
Time: 28100 | Test 10: a = 1860, b = 1905, out_hw = 3765, out_ref = 3765, correct)
Time: 30100 | Test 11: a = 3375, b = 3017, out_hw = 2296, out_ref = 2296, correct)
Time: 32100 | Test 12: a = 3610, b = 2809, out_hw = 2323, out_ref = 2323, correct)
Time: 34100 | Test 13: a = 808, b = 3856, out_hw = 568, out_ref = 568, correct)
Time: 36100 | Test 14: a = 2703, b = 2626, out_hw = 1233, out_ref = 1233, correct)
Time: 38100 | Test 15: a = 1416, b = 2760, out_hw = 80, out_ref = 80, correct)
Simulation complete via $finish(1) at time 138 NS + 0
./prob4_tbench.v:41         $finish;
ncsim> exit
```

Modified:



```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob4.logv -f prob4.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob3_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 10100 | Test 1: a = 2048, b = 2274, out_hw = 226, out_ref = 226, correct)
Time: 12100 | Test 2: a = 1392, b = 1244, out_hw = 2380, out_ref = 2636, wrong)
Time: 14100 | Test 3: a = 2084, b = 1459, out_hw = 3543, out_ref = 3543, correct)
Time: 16100 | Test 4: a = 3521, b = 1404, out_hw = 573, out_ref = 829, wrong)
Time: 18100 | Test 5: a = 2099, b = 3121, out_hw = 1124, out_ref = 1124, correct)
Time: 20100 | Test 6: a = 3070, b = 82, out_hw = 2880, out_ref = 3152, wrong)
Time: 22100 | Test 7: a = 603, b = 3995, out_hw = 486, out_ref = 502, wrong)
Time: 24100 | Test 8: a = 2725, b = 813, out_hw = 3522, out_ref = 3538, wrong)
Time: 26100 | Test 9: a = 1798, b = 1519, out_hw = 3301, out_ref = 3317, wrong)
Time: 28100 | Test 10: a = 1860, b = 1905, out_hw = 3765, out_ref = 3765, correct)
Time: 30100 | Test 11: a = 3375, b = 3017, out_hw = 2280, out_ref = 2296, wrong)
Time: 32100 | Test 12: a = 3610, b = 2809, out_hw = 2051, out_ref = 2323, wrong)
Time: 34100 | Test 13: a = 808, b = 3856, out_hw = 568, out_ref = 568, correct)
Time: 36100 | Test 14: a = 2703, b = 2626, out_hw = 1217, out_ref = 1233, wrong)
Time: 38100 | Test 15: a = 1416, b = 2760, out_hw = 3904, out_ref = 80, wrong)
Simulation complete via $finish(1) at time 138 NS + 0
./prob4_tbench.v:41         $finish;
ncsim> exit
```



```
****************************************
Report : area
Design : prob4
Version: W-2024.09-SP1
Date   : Sun Feb  9 13:58:25 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db)

Number of ports:                           37
Number of nets:                           251
Number of cells:                          204
Number of combinational cells:            168
Number of sequential cells:                36
Number of macros/black boxes:               0
Number of buf/inv:                         37
Number of references:                      28

Combinational area:              176.889999
Buf/Inv area:                     23.940000
Noncombinational area:           173.431995
Macro/Black Box area:              0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 350.321994
Total area:                 undefined
1
```

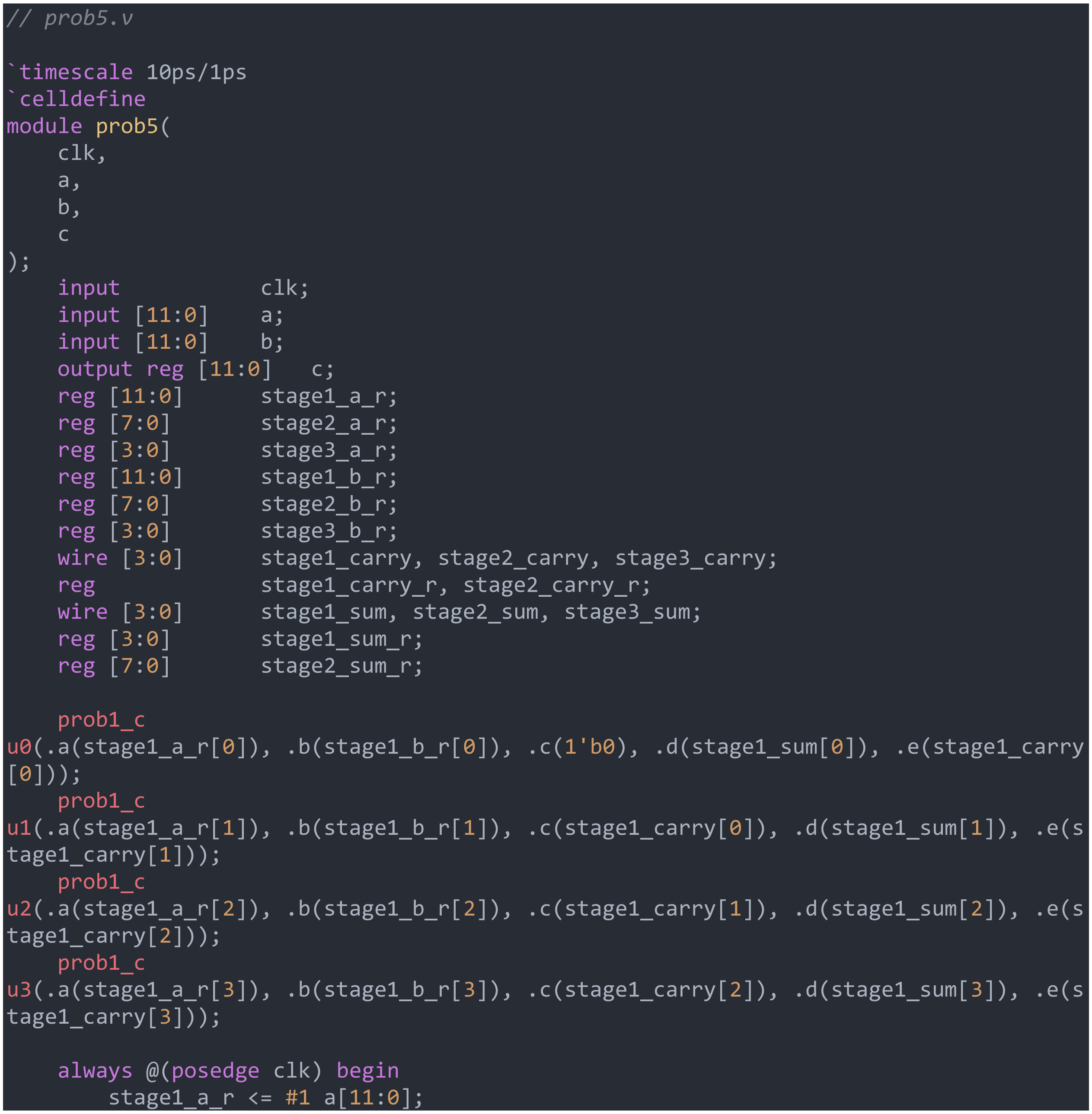
```
****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob4
Version: W-2024.09-SP1
Date   : Sun Feb  9 13:58:26 2025
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: a_reg_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: c_reg[8] (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob4              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  a_reg_reg[0]/CK (DFF_X1)                 0.00       0.00 r
  a_reg_reg[0]/Q (DFF_X1)                  0.08       0.08 f
  U93/ZN (AND3_X1)                         0.04       0.13 f
  U122/ZN (NAND3_X1)                       0.03       0.16 r
  U123/ZN (NAND2_X1)                       0.03       0.19 f
  U89/ZN (AOI211_X1)                       0.08       0.27 r
  U98/ZN (OAI21_X1)                        0.04       0.31 f
  U96/ZN (NOR3_X1)                         0.07       0.38 r
  U74/ZN (NOR2_X1)                         0.04       0.42 f
  U65/ZN (XNOR2_X1)                        0.06       0.48 f
  c_reg[8]/D (DFF_X1)                      0.01       0.49 f
  data arrival time                                   0.49

  clock clk (rise edge)                    0.10       0.10
  clock network delay (ideal)              0.00       0.10
  clock uncertainty                        0.00       0.09
  c_reg[8]/CK (DFF_X1)                     0.00       0.09 r
  library setup time                      -0.04       0.05
  data required time                                  0.05
  -----------------------------------------------------------
  data required time                                  0.05
  data arrival time                                  -0.49
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.44
```

Maximum clock rate = 1 / (0.1 – (-0.44)) = 1.852 GHz

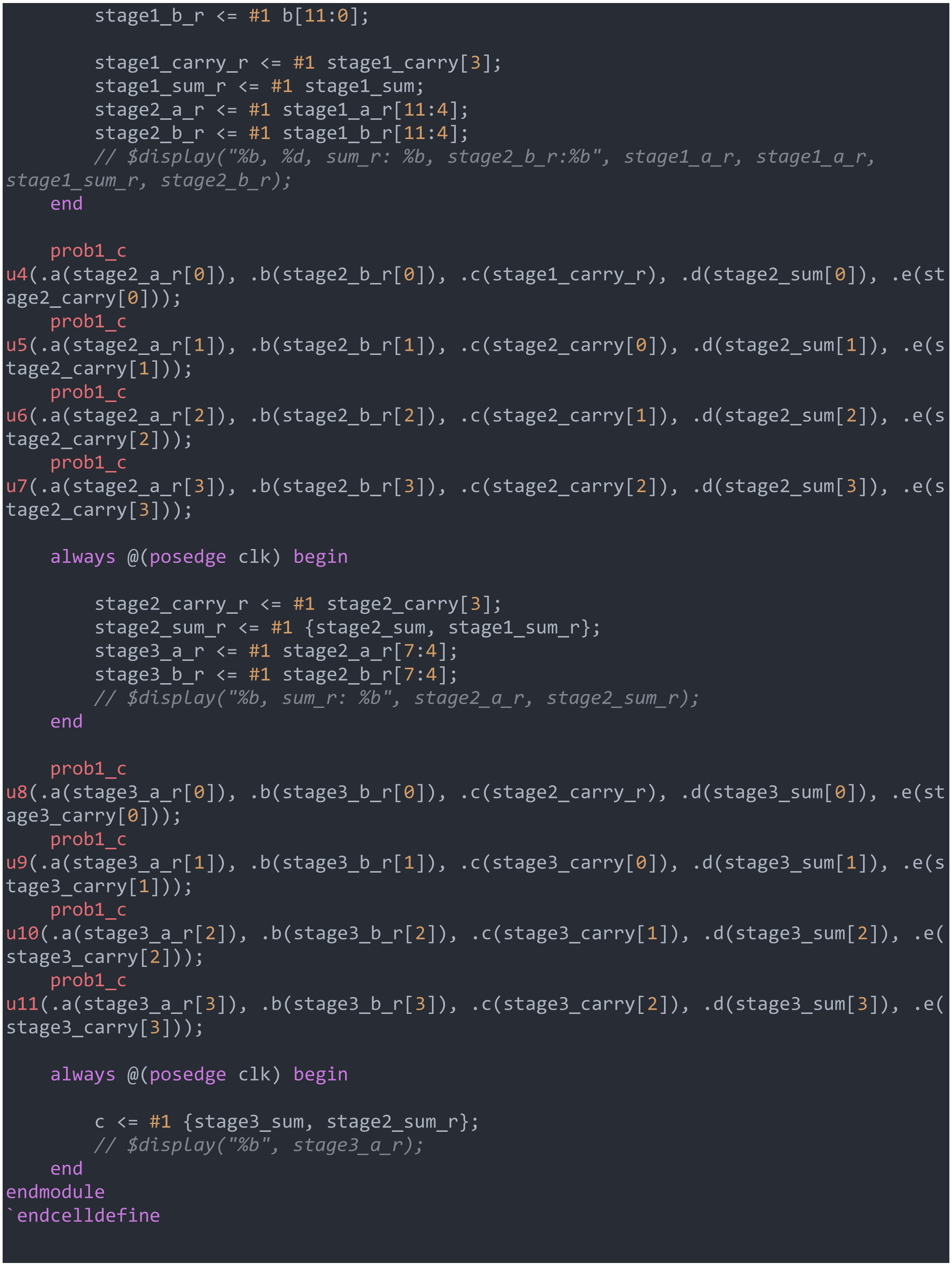
c) [10 pts] Justify the partitioning you chose.

The time complexity is O(block_size + block_cnt), and block_size * block_cnt = the number of bits. So when block_size = block_cnt = sqrt(the number of bits), we have minimum of (block_size + block_cnt). Here I choose block_size = 4, block_cnt = 3.

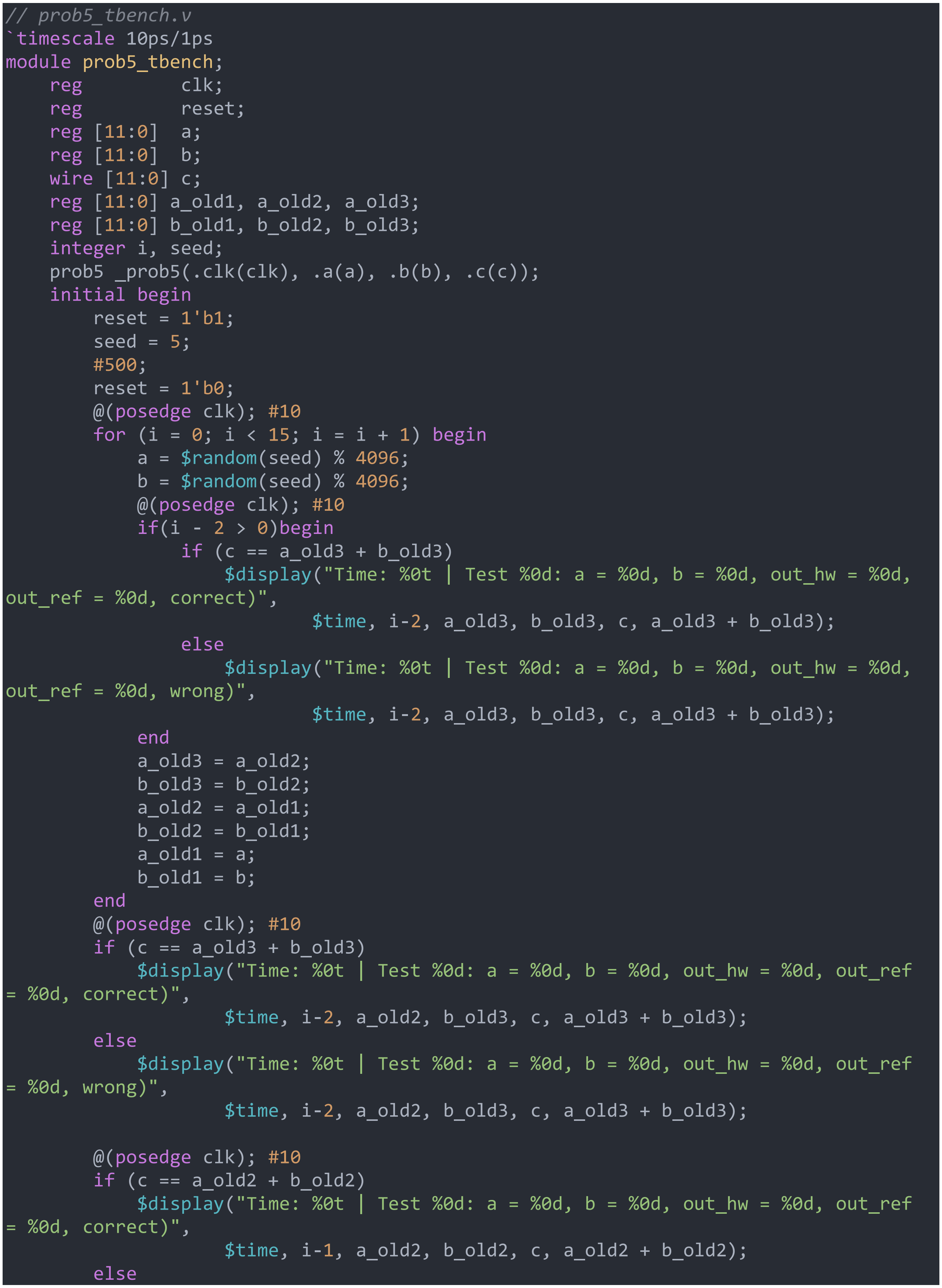
5. [50+10 pts] Repeat Problem 2(a)(b) but pipeline the ripple-carry adder into 3 pipeline stages.



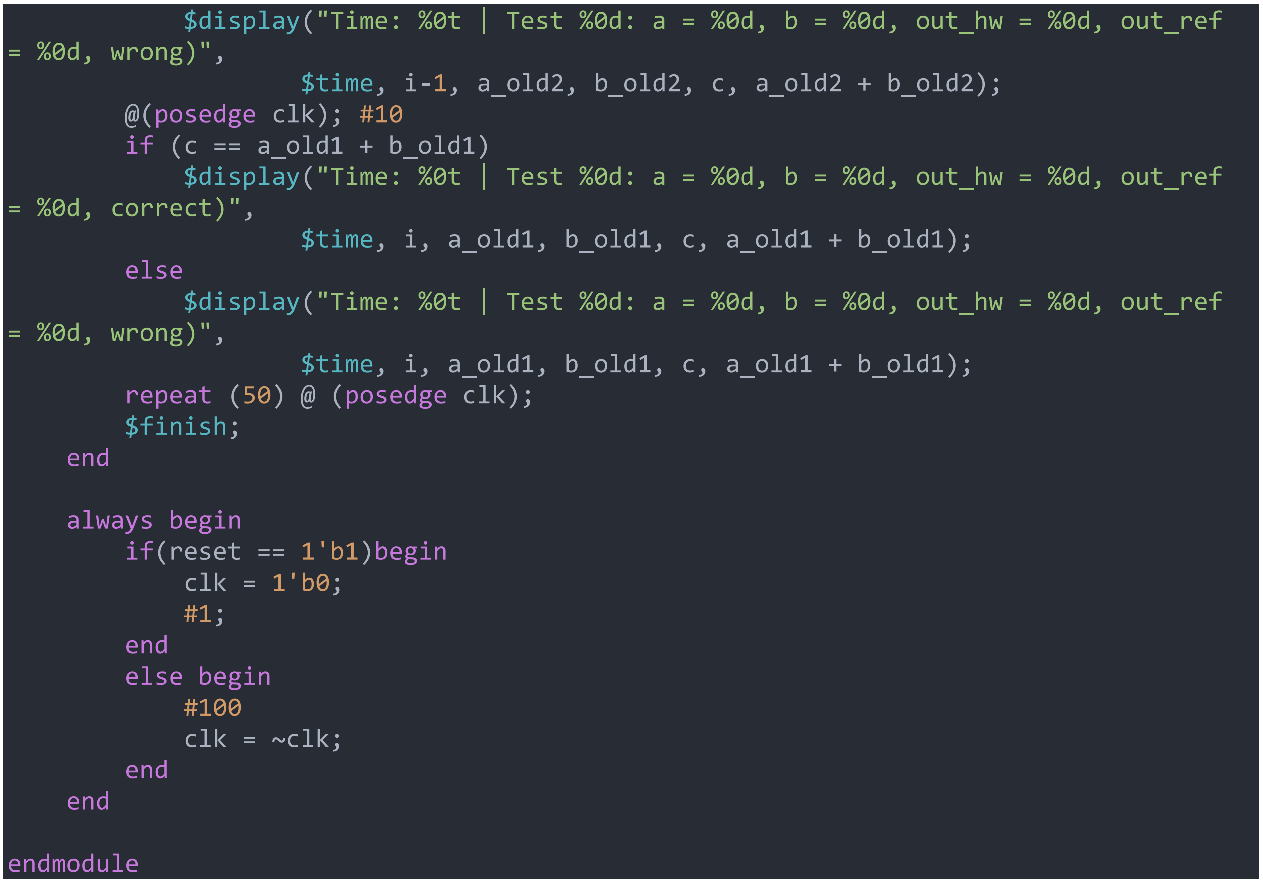
```
// prob5.v

`timescale 10ps/1ps
`celldefine
module prob5(
    clk,
    a,
    b,
    c
);
    input           clk;
    input [11:0]    a;
    input [11:0]    b;
    output reg [11:0]   c;
    reg [11:0]      stage1_a_r;
    reg [7:0]       stage2_a_r;
    reg [3:0]       stage3_a_r;
    reg [11:0]      stage1_b_r;
    reg [7:0]       stage2_b_r;
    reg [3:0]       stage3_b_r;
    wire [3:0]      stage1_carry, stage2_carry, stage3_carry;
    reg             stage1_carry_r, stage2_carry_r;
    wire [3:0]      stage1_sum, stage2_sum, stage3_sum;
    reg [3:0]       stage1_sum_r;
    reg [7:0]       stage2_sum_r;

    prob1_c
u0(.a(stage1_a_r[0]), .b(stage1_b_r[0]), .c(1'b0), .d(stage1_sum[0]), .e(stage1_carry
[0]));
    prob1_c
u1(.a(stage1_a_r[1]), .b(stage1_b_r[1]), .c(stage1_carry[0]), .d(stage1_sum[1]), .e(s
tage1_carry[1]));
    prob1_c
u2(.a(stage1_a_r[2]), .b(stage1_b_r[2]), .c(stage1_carry[1]), .d(stage1_sum[2]), .e(s
tage1_carry[2]));
    prob1_c
u3(.a(stage1_a_r[3]), .b(stage1_b_r[3]), .c(stage1_carry[2]), .d(stage1_sum[3]), .e(s
tage1_carry[3]));

    always @(posedge clk) begin
        stage1_a_r <= #1 a[11:0];
```



```
        stage1_b_r <= #1 b[11:0];

        stage1_carry_r <= #1 stage1_carry[3];
        stage1_sum_r <= #1 stage1_sum;
        stage2_a_r <= #1 stage1_a_r[11:4];
        stage2_b_r <= #1 stage1_b_r[11:4];
        // $display("%b, %d, sum_r: %b, stage2_b_r:%b", stage1_a_r, stage1_a_r, stage1_sum_r, stage2_b_r);
    end

    prob1_c u4(.a(stage2_a_r[0]), .b(stage2_b_r[0]), .c(stage1_carry_r), .d(stage2_sum[0]), .e(stage2_carry[0]));
    prob1_c u5(.a(stage2_a_r[1]), .b(stage2_b_r[1]), .c(stage2_carry[0]), .d(stage2_sum[1]), .e(stage2_carry[1]));
    prob1_c u6(.a(stage2_a_r[2]), .b(stage2_b_r[2]), .c(stage2_carry[1]), .d(stage2_sum[2]), .e(stage2_carry[2]));
    prob1_c u7(.a(stage2_a_r[3]), .b(stage2_b_r[3]), .c(stage2_carry[2]), .d(stage2_sum[3]), .e(stage2_carry[3]));

    always @(posedge clk) begin

        stage2_carry_r <= #1 stage2_carry[3];
        stage2_sum_r <= #1 {stage2_sum, stage1_sum_r};
        stage3_a_r <= #1 stage2_a_r[7:4];
        stage3_b_r <= #1 stage2_b_r[7:4];
        // $display("%b, sum_r: %b", stage2_a_r, stage2_sum_r);
    end

    prob1_c u8(.a(stage3_a_r[0]), .b(stage3_b_r[0]), .c(stage2_carry_r), .d(stage3_sum[0]), .e(stage3_carry[0]));
    prob1_c u9(.a(stage3_a_r[1]), .b(stage3_b_r[1]), .c(stage3_carry[0]), .d(stage3_sum[1]), .e(stage3_carry[1]));
    prob1_c u10(.a(stage3_a_r[2]), .b(stage3_b_r[2]), .c(stage3_carry[1]), .d(stage3_sum[2]), .e(stage3_carry[2]));
    prob1_c u11(.a(stage3_a_r[3]), .b(stage3_b_r[3]), .c(stage3_carry[2]), .d(stage3_sum[3]), .e(stage3_carry[3]));

    always @(posedge clk) begin

        c <= #1 {stage3_sum, stage2_sum_r};
        // $display("%b", stage3_a_r);
    end
endmodule
`endcelldefine
```



```
// prob5_tbench.v
`timescale 10ps/1ps
module prob5_tbench;
    reg         clk;
    reg         reset;
    reg [11:0]  a;
    reg [11:0]  b;
    wire [11:0] c;
    reg [11:0] a_old1, a_old2, a_old3;
    reg [11:0] b_old1, b_old2, b_old3;
    integer i, seed;
    prob5 _prob5(.clk(clk), .a(a), .b(b), .c(c));
    initial begin
        reset = 1'b1;
        seed = 5;
        #500;
        reset = 1'b0;
        @(posedge clk); #10
        for (i = 0; i < 15; i = i + 1) begin
            a = $random(seed) % 4096;
            b = $random(seed) % 4096;
            @(posedge clk); #10
            if(i - 2 > 0)begin
                if (c == a_old3 + b_old3)
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d,
out_ref = %0d, correct)",
                            $time, i-2, a_old3, b_old3, c, a_old3 + b_old3);
                else
                    $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d,
out_ref = %0d, wrong)",
                            $time, i-2, a_old3, b_old3, c, a_old3 + b_old3);
            end
            a_old3 = a_old2;
            b_old3 = b_old2;
            a_old2 = a_old1;
            b_old2 = b_old1;
            a_old1 = a;
            b_old1 = b;
        end
        @(posedge clk); #10
        if (c == a_old3 + b_old3)
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, correct)",
                    $time, i-2, a_old2, b_old3, c, a_old3 + b_old3);
        else
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, wrong)",
                    $time, i-2, a_old2, b_old3, c, a_old3 + b_old3);

        @(posedge clk); #10
        if (c == a_old2 + b_old2)
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, correct)",
                    $time, i-1, a_old2, b_old2, c, a_old2 + b_old2);
        else
```



```
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, wrong)",
                    $time, i-1, a_old2, b_old2, c, a_old2 + b_old2);
        @(posedge clk); #10
        if (c == a_old1 + b_old1)
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, correct)",
                    $time, i, a_old1, b_old1, c, a_old1 + b_old1);
        else
            $display("Time: %0t | Test %0d: a = %0d, b = %0d, out_hw = %0d, out_ref
= %0d, wrong)",
                    $time, i, a_old1, b_old1, c, a_old1 + b_old1);
        repeat (50) @ (posedge clk);
        $finish;
    end

    always begin
        if(reset == 1'b1)begin
            clk = 1'b0;
            #1;
        end
        else begin
            #100
            clk = ~clk;
        end
    end

endmodule
```

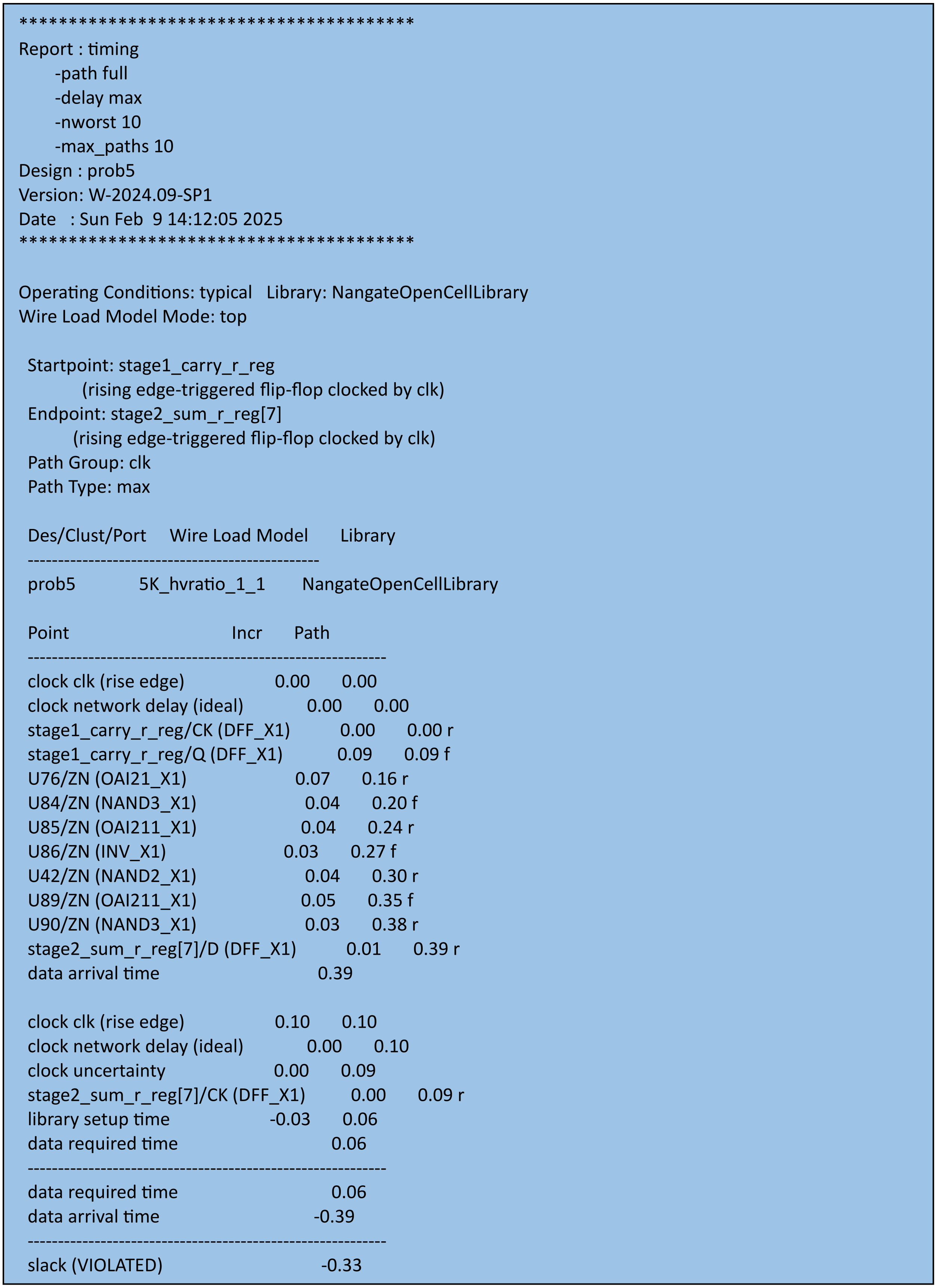
Pass:

```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob5.logv -f prob5.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob5_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 14100 | Test 1: a = 1536, b = 3867, out_hw = 1307, out_ref = 1307, correct)
Time: 16100 | Test 2: a = 2238, b = 1969, out_hw = 111, out_ref = 111, correct)
Time: 18100 | Test 3: a = 918, b = 940, out_hw = 1858, out_ref = 1858, correct)
Time: 20100 | Test 4: a = 704, b = 2985, out_hw = 3689, out_ref = 3689, correct)
Time: 22100 | Test 5: a = 3331, b = 3788, out_hw = 3023, out_ref = 3023, correct)
Time: 24100 | Test 6: a = 1766, b = 2964, out_hw = 634, out_ref = 634, correct)
Time: 26100 | Test 7: a = 80, b = 2915, out_hw = 2995, out_ref = 2995, correct)
Time: 28100 | Test 8: a = 701, b = 609, out_hw = 1310, out_ref = 1310, correct)
Time: 30100 | Test 9: a = 1763, b = 1296, out_hw = 3059, out_ref = 3059, correct)
Time: 32100 | Test 10: a = 498, b = 3838, out_hw = 240, out_ref = 240, correct)
Time: 34100 | Test 11: a = 3656, b = 2242, out_hw = 1802, out_ref = 1802, correct)
Time: 36100 | Test 12: a = 3433, b = 2348, out_hw = 1685, out_ref = 1685, correct)
Time: 38100 | Test 13: a = 1470, b = 720, out_hw = 1450, out_ref = 1450, correct)
Time: 40100 | Test 14: a = 1470, b = 19, out_hw = 1489, out_ref = 1489, correct)
Time: 42100 | Test 15: a = 182, b = 3835, out_hw = 4017, out_ref = 4017, correct)
Simulation complete via $finish(1) at time 142 NS + 0
./prob5_tbench.v:61        $finish;
ncsim> exit
```

Modified:

```
[will02@coe-ece-2107-31 hw2]$ make run
ncverilog +access+r -l prob5.logv -f prob5.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob5_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
Time: 14100 | Test 1: a = 1536, b = 3867, out_hw = X, out_ref = 1307, wrong)
Time: 16100 | Test 2: a = 2238, b = 1969, out_hw = Z, out_ref = 111, wrong)
Time: 18100 | Test 3: a = 918, b = 940, out_hw = X, out_ref = 1858, wrong)
Time: 20100 | Test 4: a = 704, b = 2985, out_hw = Z, out_ref = 3689, wrong)
Time: 22100 | Test 5: a = 3331, b = 3788, out_hw = X, out_ref = 3023, wrong)
Time: 24100 | Test 6: a = 1766, b = 2964, out_hw = Z, out_ref = 634, wrong)
Time: 26100 | Test 7: a = 80, b = 2915, out_hw = X, out_ref = 2995, wrong)
Time: 28100 | Test 8: a = 701, b = 609, out_hw = X, out_ref = 1310, wrong)
Time: 30100 | Test 9: a = 1763, b = 1296, out_hw = X, out_ref = 3059, wrong)
Time: 32100 | Test 10: a = 498, b = 3838, out_hw = Z, out_ref = 240, wrong)
Time: 34100 | Test 11: a = 3656, b = 2242, out_hw = X, out_ref = 1802, wrong)
Time: 36100 | Test 12: a = 3433, b = 2348, out_hw = X, out_ref = 1685, wrong)
Time: 38100 | Test 13: a = 1470, b = 720, out_hw = X, out_ref = 1450, wrong)
Time: 40100 | Test 14: a = 1470, b = 19, out_hw = X, out_ref = 1489, wrong)
Time: 42100 | Test 15: a = 182, b = 3835, out_hw = Z, out_ref = 4017, wrong)
Simulation complete via $finish(1) at time 142 NS + 0
./prob5_tbench.v:61        $finish;
ncsim> exit
```

```
****************************************
Report : area
Design : prob5
Version: W-2024.09-SP1
Date   : Sun Feb  9 14:12:04 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db)

Number of ports:                           37
Number of nets:                           233
Number of cells:                          187
Number of combinational cells:            113
Number of sequential cells:                74
Number of macros/black boxes:               0
Number of buf/inv:                         21
Number of references:                      20

Combinational area:               121.827999
Buf/Inv area:                      12.502000
Noncombinational area:            335.159988
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  456.987987
Total area:                 undefined
1
```



```
****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob5
Version: W-2024.09-SP1
Date   : Sun Feb  9 14:12:05 2025
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: stage1_carry_r_reg
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: stage2_sum_r_reg[7]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob5              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  stage1_carry_r_reg/CK (DFF_X1)           0.00       0.00 r
  stage1_carry_r_reg/Q (DFF_X1)            0.09       0.09 f
  U76/ZN (OAI21_X1)                        0.07       0.16 r
  U84/ZN (NAND3_X1)                        0.04       0.20 f
  U85/ZN (OAI211_X1)                       0.04       0.24 r
  U86/ZN (INV_X1)                          0.03       0.27 f
  U42/ZN (NAND2_X1)                        0.04       0.30 r
  U89/ZN (OAI211_X1)                       0.05       0.35 f
  U90/ZN (NAND3_X1)                        0.03       0.38 r
  stage2_sum_r_reg[7]/D (DFF_X1)           0.01       0.39 r
  data arrival time                                   0.39

  clock clk (rise edge)                    0.10       0.10
  clock network delay (ideal)              0.00       0.10
  clock uncertainty                        0.00       0.09
  stage2_sum_r_reg[7]/CK (DFF_X1)          0.00       0.09 r
  library setup time                      -0.03       0.06
  data required time                                  0.06
  -----------------------------------------------------------
  data required time                                  0.06
  data arrival time                                  -0.39
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.33
```

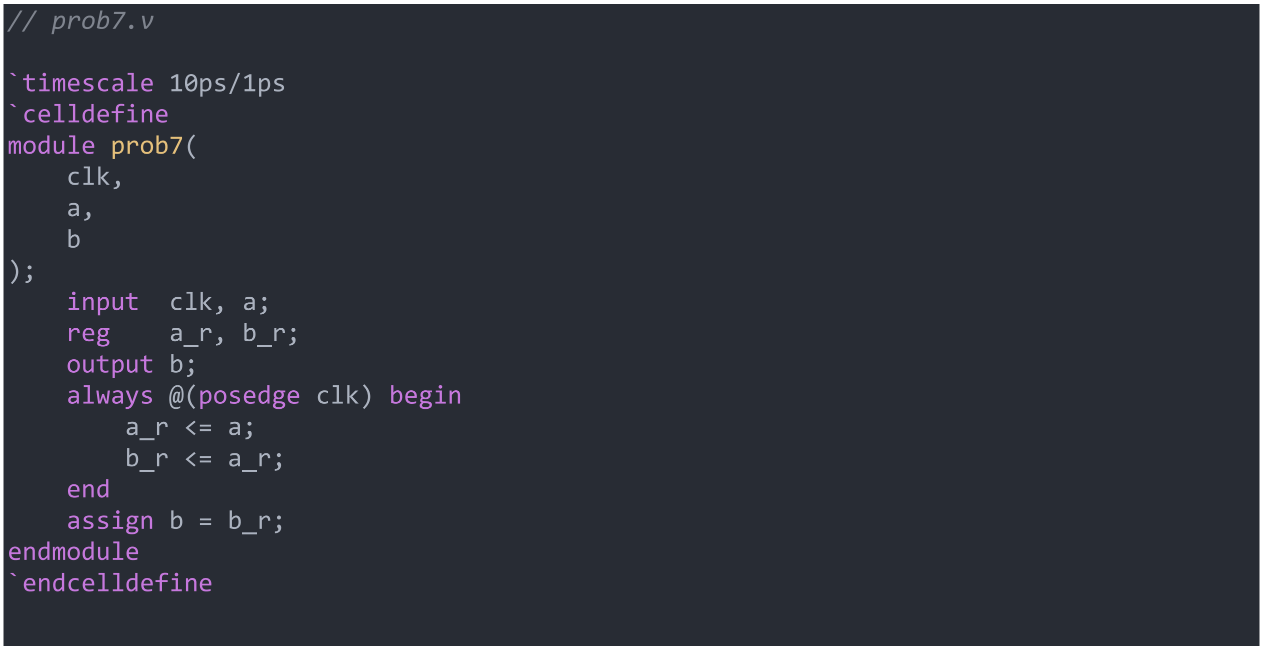
Maximum clock rate = 1 – (0.1 -(-0.33)) = 2.326 GHz

6. [5 pts] Write a single table with 1) max clock frequency and 2) area for problems 2–5.

Problem	Max clock frequency(GHz)	Area
2	1.266	322.657994
3	1.786	373.995994
4	1.852	350.321994
5	2.326	456.987987

7. [40 pts] Find the maximum possible clock frequency for a pipeline stage with no logic; that is, one register connected directly to another register.

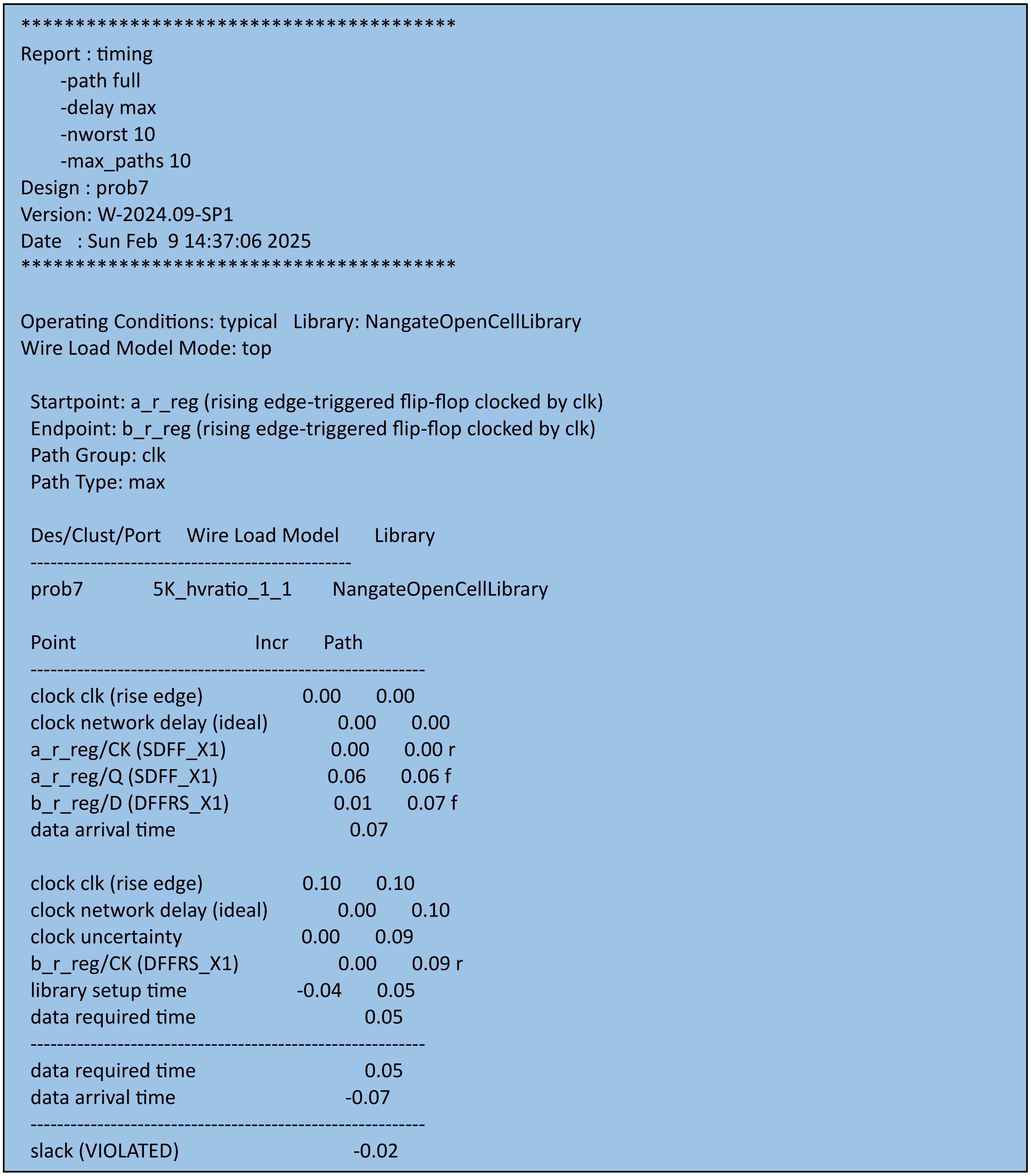
a) [25 pts] Appropriately synthesize one flip-flop connected to a second flip-flop to find the desired value. In a few sentences, state how you did this.



```
// prob7.v

`timescale 10ps/1ps
`celldefine
module prob7(
    clk,
    a,
    b
);
    input  clk, a;
    reg    a_r, b_r;
    output b;
    always @(posedge clk) begin
        a_r <= a;
        b_r <= a_r;
    end
    assign b = b_r;
endmodule
`endcelldefine
```

Input signal a, saves in a_r. a_r connect to b_r. Output b form b_r.

b) [15 pts] Report each component of the minimum clock cycle time.



```
****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob7
Version: W-2024.09-SP1
Date   : Sun Feb  9 14:37:06 2025
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: a_r_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: b_r_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob7              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                              Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)              0.00       0.00
  clock network delay (ideal)        0.00       0.00
  a_r_reg/CK (SDFF_X1)               0.00       0.00 r
  a_r_reg/Q (SDFF_X1)                0.06       0.06 f
  b_r_reg/D (DFFRS_X1)               0.01       0.07 f
  data arrival time                             0.07

  clock clk (rise edge)              0.10       0.10
  clock network delay (ideal)        0.00       0.10
  clock uncertainty                  0.00       0.09
  b_r_reg/CK (DFFRS_X1)              0.00       0.09 r
  library setup time                -0.04       0.05
  data required time                            0.05
  -----------------------------------------------------------
  data required time                            0.05
  data arrival time                            -0.07
  -----------------------------------------------------------
  slack (VIOLATED)                             -0.02
```

T_clock_to_q = 0.06 ns

T_logic_max = 0.01 ns

T_setup = 0.05 ns

T_minimum_clock = 0.12 ns