超大型積體電路測試

VLSI testing

Homework II

國立清華大學

系所:電子所碩一

中文姓名:李聖謙、賴家均

英文姓名:Sheng-Qian-Li、Jia-Jun-Lai

學號:111063517、111063578

授課老師:黃錫瑜

分工

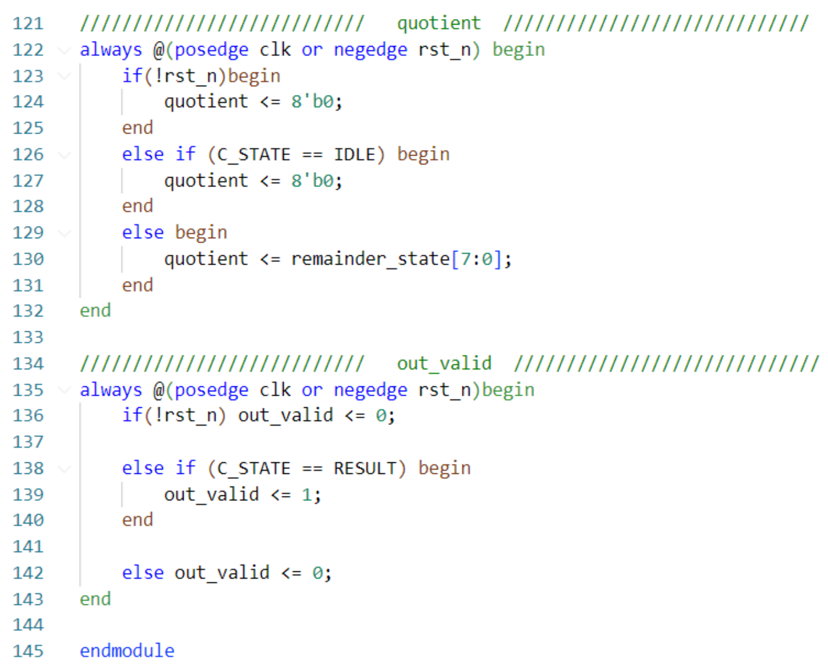
111063517 李聖謙: RTL Code、Test bench、Compiler、DFT、ATPG、Report
111063578 賴家均: Algorithm、RTL Code、Test bench、Compiler

(a) (20%) Write the **RTL code** in Verilog or VHDL that takes in two 8-bit positive integers, A[7:0] and B[7:0], and produces its quotient Q[7:0] and remainder R[7:0].

RTL Code

```verilog
module divide_8bit ( clk, rst_n, in_valid, dividend, divisor, remainder, quotient,  out_valid);

input clk;
input rst_n;
input in_valid;
input [7:0] dividend;       //被除數
input [7:0] divisor;        //除數
output reg out_valid;
output reg [7:0] remainder;
output reg [7:0] quotient;

reg[2:0]  C_STATE , next_state;
reg [7:0] counter;
reg [15:0] remainder_state, divisor_state;

parameter IDLE = 2'b00;
parameter SHIFT = 2'b01;
parameter SUBSTRACT = 2'b10;
parameter RESULT = 2'b11;

always@(posedge clk or negedge rst_n)
begin
    if (!rst_n) begin
        C_STATE <= IDLE;
    end
    else begin
        C_STATE <= next_state;
    end
end

always @(*) begin
    case(C_STATE)

        IDLE : begin
            if(in_valid) next_state = SHIFT;
            else next_state = IDLE;
        end

        SHIFT : if(counter < 8)begin
                    next_state = SUBSTRACT;
                end
                else begin
                    next_state = RESULT;
                end

        SUBSTRACT : if(counter < 8)begin
                    next_state = SHIFT;
                end
                else begin
                    next_state = RESULT;
                end
        RESULT : next_state = IDLE;

        default next_state = C_STATE;
    endcase
end

```

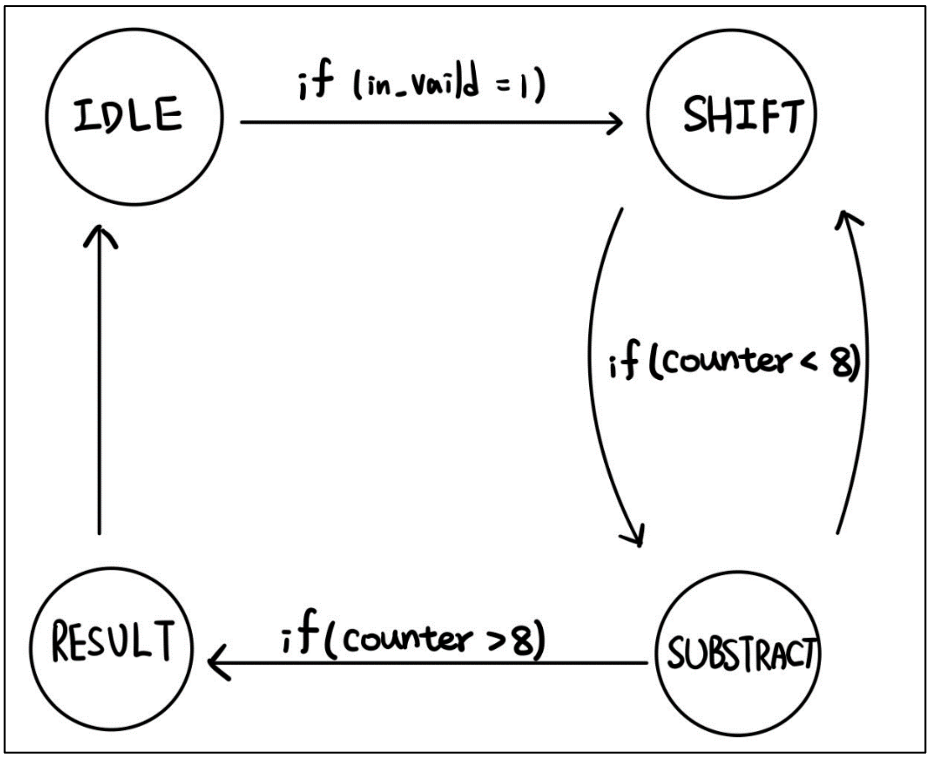
```
//////////////    remainder_state & divisor_state   //////////////
always @(posedge clk or negedge rst_n) begin
    if(!rst_n)begin
        remainder_state <= 16'b0;
        divisor_state <= 16'b0;
    end
    else if(in_valid)begin
        remainder_state <= {8'b00000000, dividend};
        divisor_state <= {divisor, 8'b00000000};
    end
    else if (C_STATE == IDLE) begin
        remainder_state <= {8'b00000000, dividend};
        divisor_state <= {divisor, 8'b00000000};
    end
    else if (C_STATE == SHIFT) begin
        remainder_state <= remainder_state << 1 ;
    end
    else if (C_STATE == SUBSTRACT) begin
        if(remainder_state >= divisor_state)
            remainder_state <= remainder_state - divisor_state + 1;
        else begin
            remainder_state <= remainder_state;
        end
    end
    else begin
        remainder_state <= 16'b0;
        divisor_state <= 16'b0;
    end
end

//////////////////////////    counter  ////////////////////////////
always @(posedge clk or negedge rst_n) begin
    if(!rst_n)begin
        counter <= 0;
    end
    else if(C_STATE == IDLE) begin
        counter <= 0;
    end
    else if(C_STATE == SHIFT)begin
        counter <= counter + 1;
    end
    else if(C_STATE == SUBSTRACT)begin
        counter <= counter;
    end
    else if(C_STATE == RESULT)begin
        counter <= 0;
    end
    else counter <= 0;
end

//////////////////////////    remainder  ////////////////////////////
always @(posedge clk or negedge rst_n) begin
    if(!rst_n)begin
        remainder <=8'b0;
    end
    else if (C_STATE == IDLE) begin
        remainder <=8'b0;
    end
    else begin
        remainder <= remainder_state[15:8];
    end
end
```



```
/////////////////////////   quotient  ///////////////////////////
always @(posedge clk or negedge rst_n) begin
    if(!rst_n)begin
        quotient <= 8'b0;
    end
    else if (C_STATE == IDLE) begin
        quotient <= 8'b0;
    end
    else begin
        quotient <= remainder_state[7:0];
    end
end

/////////////////////////   out_valid  ///////////////////////////
always @(posedge clk or negedge rst_n)begin
    if(!rst_n) out_valid <= 0;

    else if (C_STATE == RESULT) begin
        out_valid <= 1;
    end

    else out_valid <= 0;
end

endmodule
```

此 RTL Code 我們使用了 Finite State Machine 實現 8-bits 的 divider，以下是 FSM 的狀態圖。

Divider - Finite State Machine



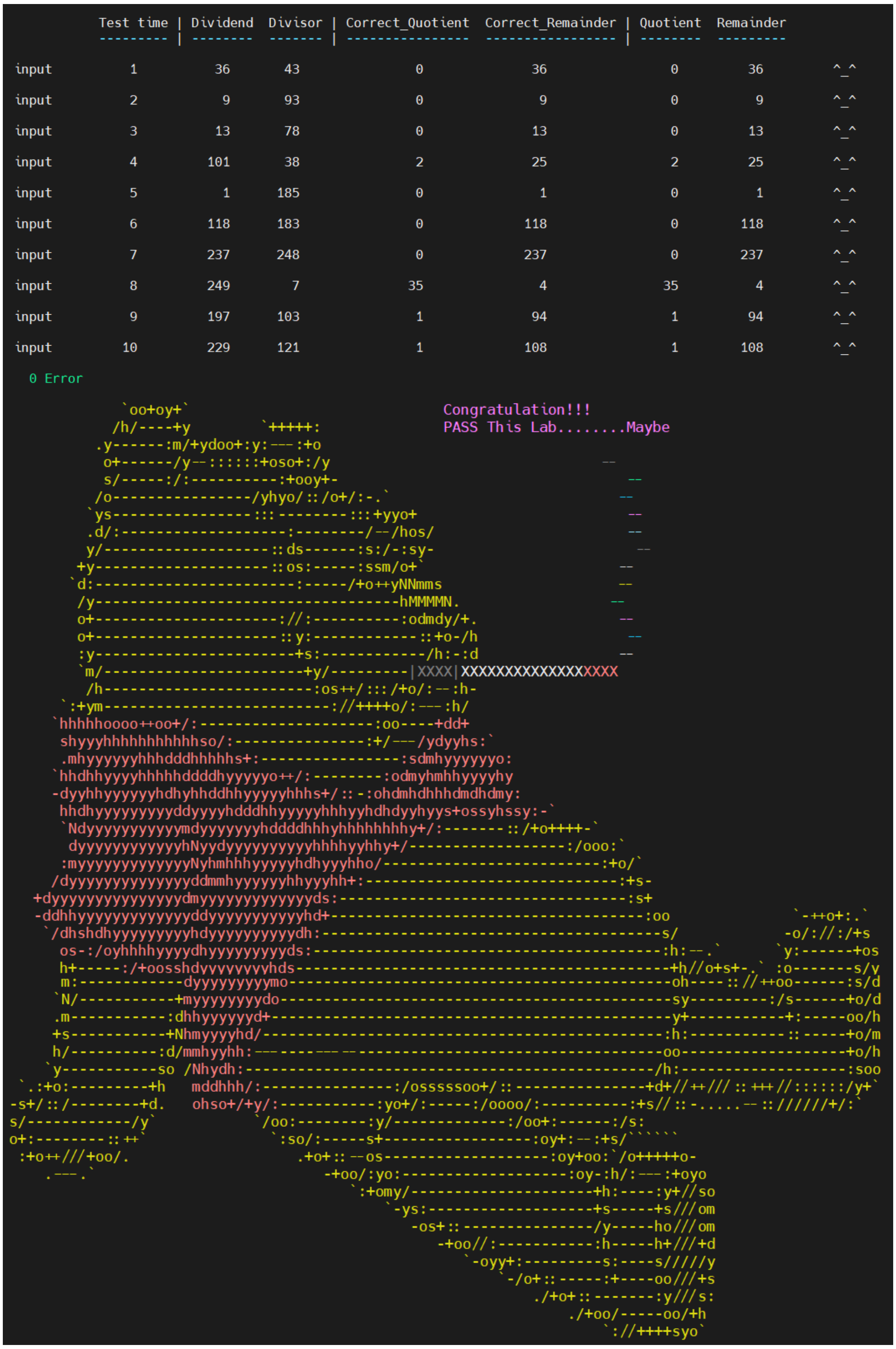
(b) (20%) **Verify the correctness of your RTL code** by a test bench. You should try it out by at least 3 pairs of input numbers.

Test bench

```verilog
`timescale 1ns/1ps
`define CYCLE_TIME 2

module testbench;

parameter period = `CYCLE_TIME;

reg clk;
reg rst_n;
reg in_valid;
reg [7:0] dividend;            //被除數
reg [7:0] divisor;     //除數
reg [7:0] correct_quotient;
reg [7:0] correct_remainder;
wire [7:0] quotient;  //[商]
wire [7:0]remainder;  //餘數
wire out_valid;
integer i;
integer err_cnt ;
divide_8bit u1(
    .clk(clk),
    .rst_n(rst_n),
    .in_valid(in_valid),
    .out_valid(out_valid),
    .dividend(dividend),           //被除數
    .divisor(divisor),     //除數
    .quotient(quotient),  //[商]
    .remainder(remainder) //餘數
);

//clock
initial clk = 1'b0;
always #(period/2.0) clk = ~clk;
//---------------------------------------

//TASKS
initial begin
    reset_task;
    $display("\n\t\t   Test time | Dividend  Divisor | Correct_Quotient  Correct_Remainder | Quotient  Remainder ");
    $display("\t\t   --------- | --------  ------- | ----------------  ----------------- | --------  --------- ");
    for(i=0; i<10; i=i+1)
    begin
        input_1;
        check_task;
    end
    if (err_cnt !== 0) fail_task;
    else pass_task;
    end

//--------------------------------------------
//reset
task reset_task;
    begin
        clk = 0;
        rst_n = 1'b1 ;
        dividend = 0 ;
        divisor = 0  ;
        in_valid = 0 ;
        err_cnt = 0;
        #(period/2.0) rst_n = 0 ;
        #(period/2.0) rst_n = 1 ;
    end
endtask
//-----------------------------------------

//input_1
task input_1;

    begin
        in_valid = 1 ;
        dividend = {$random}%256 ;
        divisor  = {$random}%255 + 1;
        correct_quotient = dividend/divisor ;
        correct_remainder = dividend % divisor;
        $write("\n\tinput");
        $write("%d \t %d \t  %d \t\t  %d \t\t  %d",i+1,dividend,divisor,correct_quotient,correct_remainder);
        repeat(1) @(negedge clk);
        in_valid = 0 ;
        dividend = 8'bx ;
        divisor  = 8'bx ;
    end

endtask

task check_task; begin
    #(17*period) @(negedge clk);
    if(out_valid)begin
    $write("\t\t   %d \t      %d",quotient,remainder);
        if((correct_quotient !== quotient) || (correct_remainder !== remainder))
        begin
            $display("        >.<") ;
            err_cnt = err_cnt + 1;
        end
        else begin
            $display("         ^_^") ;
        end
    end
    repeat(1) @(negedge clk);
end endtask
```

```
task pass_task; begin
    $display("\n\033[1;32m%d Error  \n",err_cnt);
    $display("\033[1;33m                 `oo+oy+`                            \033[1;35m Congratulation!!! \033[1;0m                                      ");
    $display("\033[1;33m                /h/----+y        `+++++:             \033[1;35m PASS This Lab........Maybe \033[1;0m                             ");
    $display("\033[1;33m              .y------:m/+ydoo+:y:---:+o                                                                                       ");
    $display("\033[1;33m               o+------/y--::::::+oso+:/y                              \033[1;30m  --                                           ");
    $display("\033[1;33m               s/-----:/:----------:+ooy+-                              \033[1;32m    --                                        ");
    $display("\033[1;33m              /o---------------/yhyo/::/o+/:-.`                         \033[1;34m   --                                         ");
    $display("\033[1;33m             `ys---------------:::--------:::+yyo+                       \033[1;35m   --                                        ");
    $display("\033[1;33m             .d/:------------------:--------/--/hos/                     \033[1;36m   --                                        ");
    $display("\033[1;33m             y/------------------::ds------:s:/-:sy-                     \033[1;90m    --                                       ");
    $display("\033[1;33m            +y-------------------::os:-----:ssm/o+`                      \033[1;37m  --                                         ");
    $display("\033[1;33m           `d:----------------------:-----/+o++yNNmms                    \033[1;38m  --                                         ");
    $display("\033[1;33m            /y----------------------------------hMMMMN.                  \033[1;92m --                                          ");
    $display("\033[1;33m            o+--------------------://:----------:odmdy/+.                 \033[1;95m --                                         ");
    $display("\033[1;33m            o+--------------------::y:-----------::+o-/h                   \033[1;94m --                                        ");
    $display("\033[1;33m            :y---------------------+s:--------\033[1;33m----/h:-:d                  \033[1;39m --                               ");
    $display("\033[1;33m            `m/---------------------+y/-----\033[1;33m---\033[1;30m|XXXX|\033[1;39mXXXXXXXXXXXXXX\033[1;91mXXXX                    ");
    $display("\033[1;33m             /h-----------------------:os++/:::/+o/:--:h-                                                                      ");
    $display("\033[1;33m          `:+ym-------------------------://++++o/:---:h/                                                                       ");
    $display("\033[1;31m         `hhhhhoooo++oo+/:\033[1;33m-------------------:oo----\033[1;31m+dd+                                                       ");
    $display("\033[1;31m          shyyyhhhhhhhhhhhso/:\033[1;33m---------------:+/---\033[1;31m/ydyyhs:`                                                 ");
    $display("\033[1;31m          .mhyyyyyyhhhdddhhhhhs+:\033[1;33m---------------\033[1;31m:sdmhyyyyyyo:                                                ");
    $display("\033[1;31m         `hhdhhyyyyhhhhhddddhyyyyyo++/:\033[1;33m--------\033[1;31m:odmyhmhhyyyyhy                                               ");
    $display("\033[1;31m         -dyyhhyyyyyyhdhyhhddhhyyyyyhhhs+/::\033[1;33m-\033[1;31m:ohdmhdhhhdmdhdmy:                                              ");
    $display("\033[1;31m          hhdhyyyyyyyyyddyyyyhdddhhyyyyyhhhyyhdhdyyhyys+ossyhssy:-`                                                            ");
    $display("\033[1;31m          `Ndyyyyyyyyyyymdyyyyyyyhddddhhhyhhhhhhhhy+/:\033[1;33m-------::/+o++++-`                                               ");
    $display("\033[1;31m           dyyyyyyyyyyyyhNyydyyyyyyyyyyhhhhyyhhy+/\033[1;33m-----------------:/ooo:`                                            ");
    $display("\033[1;31m          :myyyyyyyyyyyyyNyhmhhhyyyyyhdhyyyhho/\033[1;33m------------------------:+o/`                                           ");
    $display("\033[1;31m         /dyyyyyyyyyyyyyyddmmhyyyyyyhhyyyhh+:\033[1;33m----------------------------:+s-                                         ");
    $display("\033[1;31m       +dyyyyyyyyyyyyyyydmyyyyyyyyyyyyyyds:\033[1;33m--------------------------------:s+                                         ");
    $display("\033[1;31m       -ddhhyyyyyyyyyyyyyddyyyyyyyyyyyhd+\033[1;33m-----------------------------------:oo              `-++o+:.`             ");
    $display("\033[1;31m        `/dhshdhyyyyyyyyyhdyyyyyyyyyyydh:\033[1;33m--------------------------------------s/             -o/://:/+s             ");
    $display("\033[1;31m          os-:/oyhhhhyyyydhyyyyyyyyyds:\033[1;33m---------------------------------------:h:--.`       `y:------+os            ");
    $display("\033[1;33m          h+-----\033[1;31m:/+oosshdyyyyyyyyhds\033[1;33m--------------------------------------------+h//o+s+-.` :o-------s/y  ");
    $display("\033[1;33m          m:------------\033[1;31mdyyyyyyyyyymo\033[1;33m-------------------------------------------oh----:://++oo------:s/d  ");
    $display("\033[1;33m         `N/-----------+\033[1;31mmyyyyyyyyydo\033[1;33m-------------------------------------------sy---------:/s------+o/d  ");
    $display("\033[1;33m         .m-----------:d\033[1;31mhhyyyyyyd+\033[1;33m--------------------------------------------y+-----------+:-----oo/h  ");
    $display("\033[1;33m         +s-----------+N\033[1;31mhmyyyyhd/\033[1;33m--------------------------------------------:h:-----------::-----+o/m  ");
    $display("\033[1;33m         h/----------:d/\033[1;31mmmhyyhh:\033[1;33m---------------------------------------------oo-------------------+o/h  ");
    $display("\033[1;33m        `y-----------so /\033[1;31mNhydh:\033[1;33m----------------------------------------------/h:-------------------:soo  ");
    $display("\033[1;33m     `.:+o:---------+h   \033[1;31mmddhhh/:\033[1;33m--------------:/osssssoo+/::---------------+d+//++///::+++//::::::/y+`  ");
    $display("\033[1;33m    -s+/::/--------+d.   \033[1;31mohso+/+y/:\033[1;33m----------:yo+/:-----:/oooo/:----------:+s//::-.....--::///////+/:`    ");
    $display("\033[1;33m    s/------------/y`           `/oo:--------:y/------------:/oo+:------:/s:                                                   ");
    $display("\033[1;33m    o+:--------::++`              `:so/:-----s+----------------:oy+:--:+s/``````                                              ");
    $display("\033[1;33m     :+o++///+oo/.                  .+o+::--os------------------:oy+oo:`/o+++++o-                                              ");
    $display("\033[1;33m        .---.`                         -+oo/:yo:------------------:oy-:h/:---:+oyo                                             ");
    $display("\033[1;33m                                          `:+omy/--------------------+h:----:y+//so                                            ");
    $display("\033[1;33m                                              `-ys:------------------+s-----+s///om                                            ");
    $display("\033[1;33m                                                 -os+::---------------/y-----ho///om                                           ");
    $display("\033[1;33m                                                    -+oo//:-----------:h-----h+///+d                                           ");
    $display("\033[1;33m                                                       `-oyy+:---------s:----s/////y                                           ");
    $display("\033[1;33m                                                           `-/o+::-----:+----oo///+s                                           ");
    $display("\033[1;33m                                                               ./+o+::-------:y///s:                                           ");
    $display("\033[1;33m                                                                   ./+oo/-----oo/+h                                            ");
    $display("\033[1;33m                                                                       `://++++syo`                                            ");
    $display("\033[1;0m");
    repeat(5) @(negedge clk);
    $finish;
end
endtask

task fail_task; begin
    $display("\nFAIL!! There were %d errors in all.\n", err_cnt);
    $display("                                                                             ");
    $display("                                                 ./+oo+/.                    ");
    $display("                                                /s:-----+s`                  ");
    $display("                                                y/-------:y                  ");
    $display("                                           `.-:/od+/------y`                 ");
    $display("                              `:///+++ooooooo+//::::-----:/y+:`              ");
    $display("                             -m+:::::::---------------------::o+.            ");
    $display("                            `hod-------------------------------:o+           ");
    $display("                      ./++/:s/-o/-------------------------------/s///::.     ");
    $display("                     /s::-://--:-------------------------------:oo/::::o+    ");
    $display("                   -+ho++++//hh:-------------------------------:s:-------+/  ");
    $display("                 -s+shdh+::+hm+-------------------------------+/--------:s   ");
    $display("                -s:hMMMMNy---+y/-------------------------------:---------//  ");
    $display("                y:/NMMMMMN:---:s-/o:------------------------------------+`   ");
    $display("                h--sdmmdy/-------:hyssoo++:---------------------------:/`    ");
    $display("                h---::::----------+oo+/::/+o:---------------------:+++s-`    ");
    $display("                s:---------------/s+///-----------------------------o`       ");
    $display("          ``..../s-----------------::-----------------------------o          ");
    $display("       -/oyhyyyyyym:----------------://////:-------------------------:/      ");
    $display("      /dyssyyyssssyh:-------------/o+/::::/+o/------------------------+`     ");
    $display("    -+o/---:/oyyssshd/-----------+o:---------:oo---------------------:/.     ");
    $display("  `++---------:/sysssddy+:-------/+------------s/------------------://`      ");
    $display(" .s:---------:+ooyysyyddoo++os-:s-------------/y----------------:++.         ");
    $display(" s:------------/yyhssyshy:---/:o:-------------:dsoo++//:::::-::+syh`         ");
    $display("`h-------------shysssyyms+oyo:--------------/hyyyyyyyyyyyysyhyyyy`           ");
    $display("`h-------------:yyssssyyhhyy+----------------+dyyyysssssssyyyhs+/.           ");
    $display(" s:-------------/yyssssyhy:-----------------shyyyyhyyssssyyh.                ");
    $display(" .s-------------+sooosyyo-----------------/ysssssyyyyssssyo                  ");
    $display("  /+------------------:++-----------------:yssssssssssssssy-                 ");
    $display("  `s+-------------------------------------:sysssssssssssssyo                 ");
    $display("`+yhdo-------------------:/-------------:sysssssssssssssyy.                  ");
    $display("+yysyhh:-------------------+o-----------/ysysssssssssssssy/                  ");
    $display(" /hhysyds:------------------y----------/+yyssssssssssssssyh`                 ");
    $display(" .h-+yysyds:---------------:s----------:--/yssssssssssssssym:                ");
    $display(" y/---oyyyyhyo:-----------:o:-------------:yssssssssyyyssyyd-                ");
    $display("`h------+syyyyhhsoo+///+osh--------------:ysssyysyyyyysssssyd:              ");
    $display("/s--------:+syyyyyyyyyyyyyyhso/:-------::+oyyyyhyyyyssssssssyy+-             ");
    $display("+s-----------:/osyyyssssssyyyyhyyyyyyyyydhyyyyyysssssssssyys/`               ");
    $display("+s---------------:/osyyyysssssssssssssssyyhyysssssssyyyyso/y`                ");
    $display("/s-------------------:/+ossyyyyyysssssssssyyyyyyyysso+:----:+                ");
    $display(".h-------------------------:::/++oooooooo+++/:::----------o`                 ");
    repeat(5) @(negedge clk);
    $finish;
end endtask

initial begin
    $sdf_annotate("gcd_syn.sdf", u1);
    $fsdbDumpfile("./lai_rtl.fsdb");
    $fsdbDumpvars;
end

endmodule
```

Test bench result

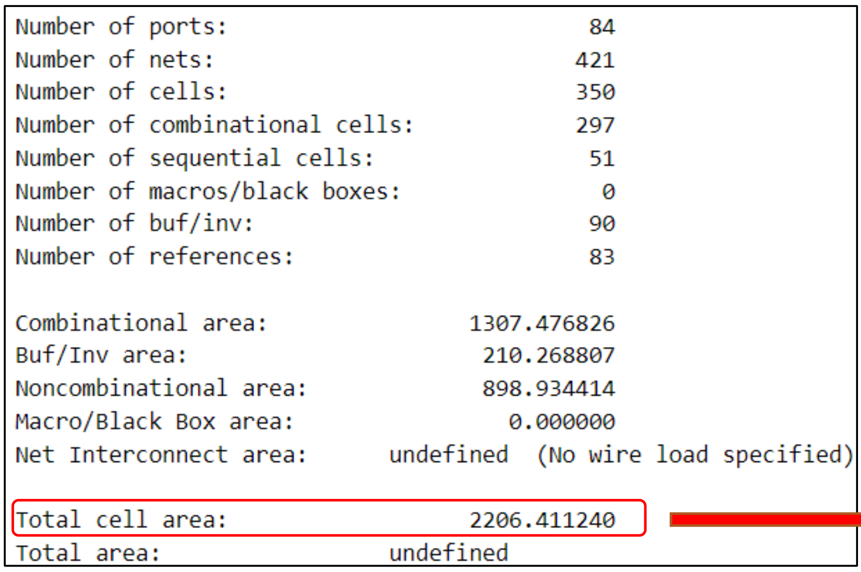


```
            Test time | Dividend  Divisor | Correct_Quotient  Correct_Remainder | Quotient  Remainder
            --------- | --------  ------- | ----------------  ----------------- | --------  ---------

input               1        36       43                  0                  36          0         36        ^_^

input               2         9       93                  0                   9          0          9        ^_^

input               3        13       78                  0                  13          0         13        ^_^

input               4       101       38                  2                  25          2         25        ^_^

input               5         1      185                  0                   1          0          1        ^_^

input               6       118      183                  0                 118          0        118        ^_^

input               7       237      248                  0                 237          0        237        ^_^

input               8       249        7                 35                   4         35          4        ^_^

input               9       197      103                  1                  94          1         94        ^_^

input              10       229      121                  1                 108          1        108        ^_^

  0 Error

                                                  Congratulation!!!
                                                  PASS This Lab........Maybe
```

測試 10 筆 input 接通過。

(c) (20%) **Use a synthesis script to convert your RTL code into a gate-level netlist.** Report the final **gate count**, the **maximum operating speed** (in MHz) and the **estimated power dissipation** in (mW) using Design Compiler.

Non scan chain – Gate level netlist

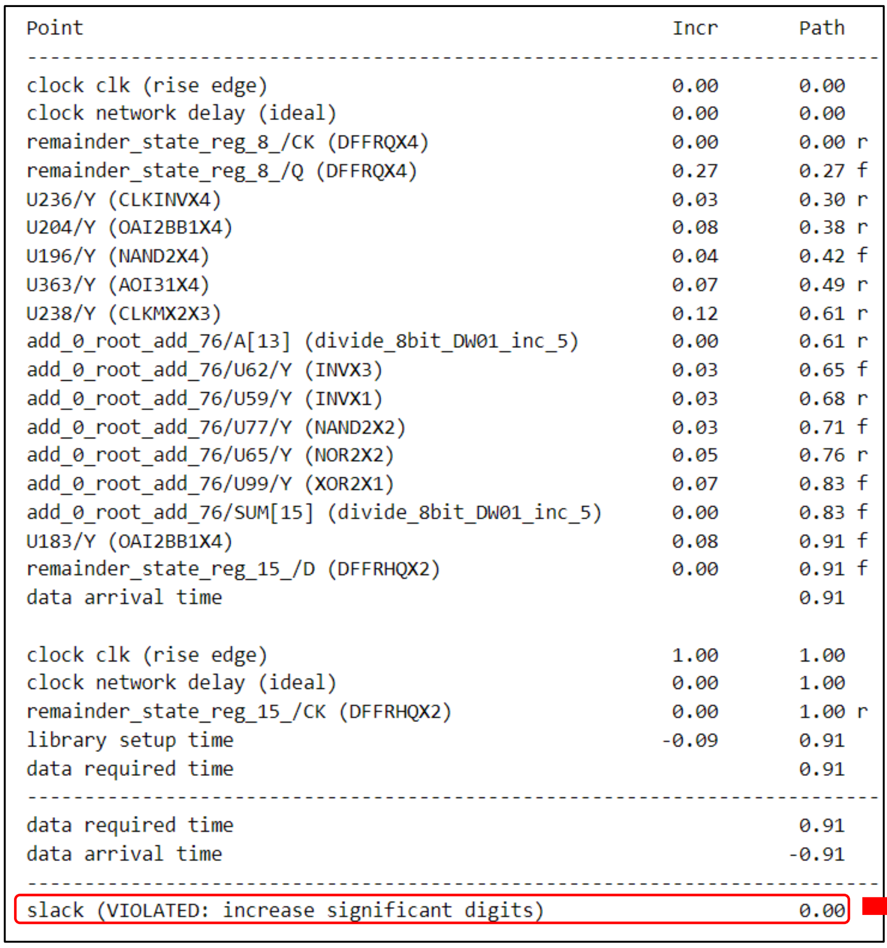


```
Number of ports:                           84
Number of nets:                           421
Number of cells:                          350
Number of combinational cells:            297
Number of sequential cells:                51
Number of macros/black boxes:               0
Number of buf/inv:                         90
Number of references:                      83

Combinational area:               1307.476826
Buf/Inv area:                      210.268807
Noncombinational area:             898.934414
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  2206.411240
Total area:                 undefined
```

Gate count = $\frac{2206.41}{2.8224} = 781.75$

Area= 2206.41 μm^2

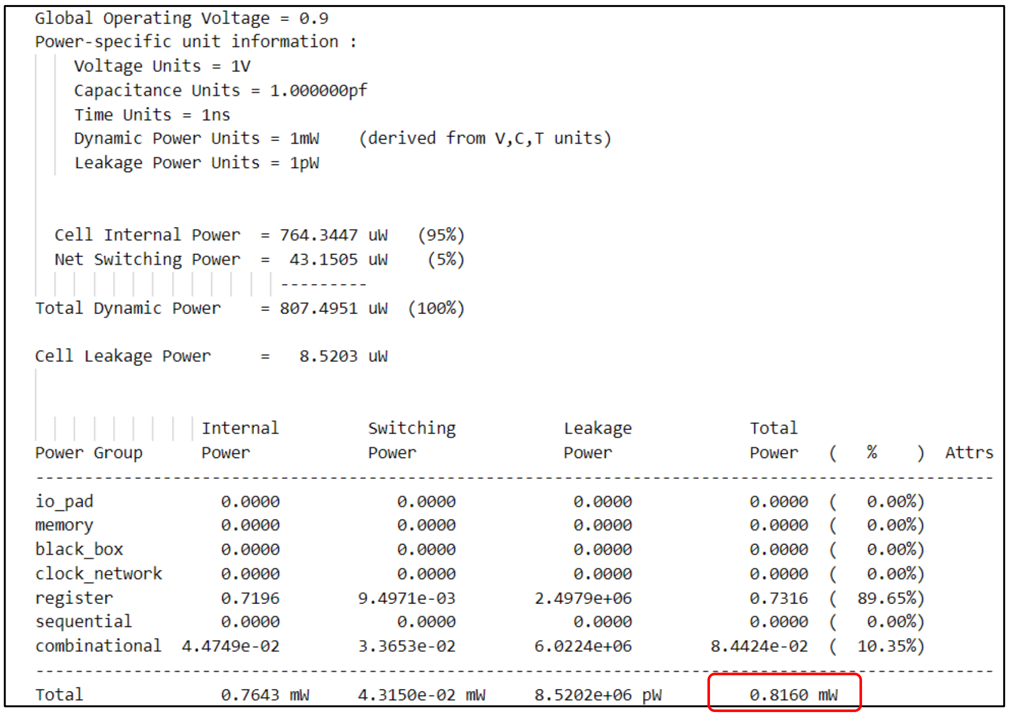
Non scan chain



```
Point                                                    Incr       Path
-------------------------------------------------------------------------
clock clk (rise edge)                                    0.00       0.00
clock network delay (ideal)                              0.00       0.00
remainder_state_reg_8_/CK (DFFRQX4)                      0.00       0.00 r
remainder_state_reg_8_/Q (DFFRQX4)                       0.27       0.27 f
U236/Y (CLKINVX4)                                        0.03       0.30 r
U204/Y (OAI2BB1X4)                                       0.08       0.38 r
U196/Y (NAND2X4)                                         0.04       0.42 f
U363/Y (AOI31X4)                                         0.07       0.49 r
U238/Y (CLKMX2X3)                                        0.12       0.61 r
add_0_root_add_76/A[13] (divide_8bit_DW01_inc_5)         0.00       0.61 r
add_0_root_add_76/U62/Y (INVX3)                          0.03       0.65 f
add_0_root_add_76/U59/Y (INVX1)                          0.03       0.68 r
add_0_root_add_76/U77/Y (NAND2X2)                        0.03       0.71 f
add_0_root_add_76/U65/Y (NOR2X2)                         0.05       0.76 r
add_0_root_add_76/U99/Y (XOR2X1)                         0.07       0.83 f
add_0_root_add_76/SUM[15] (divide_8bit_DW01_inc_5)       0.00       0.83 f
U183/Y (OAI2BB1X4)                                       0.08       0.91 f
remainder_state_reg_15_/D (DFFRHQX2)                     0.00       0.91 f
data arrival time                                                   0.91

clock clk (rise edge)                                    1.00       1.00
clock network delay (ideal)                              0.00       1.00
remainder_state_reg_15_/CK (DFFRHQX2)                    0.00       1.00 r
library setup time                                      -0.09       0.91
data required time                                                  0.91
-------------------------------------------------------------------------
data required time                                                  0.91
data arrival time                                                  -0.91
-------------------------------------------------------------------------
slack (VIOLATED: increase significant digits)                       0.00
```

maximum operating speed = $\frac{1000}{1+|0|} = 1000$ MHz

Power dissipation

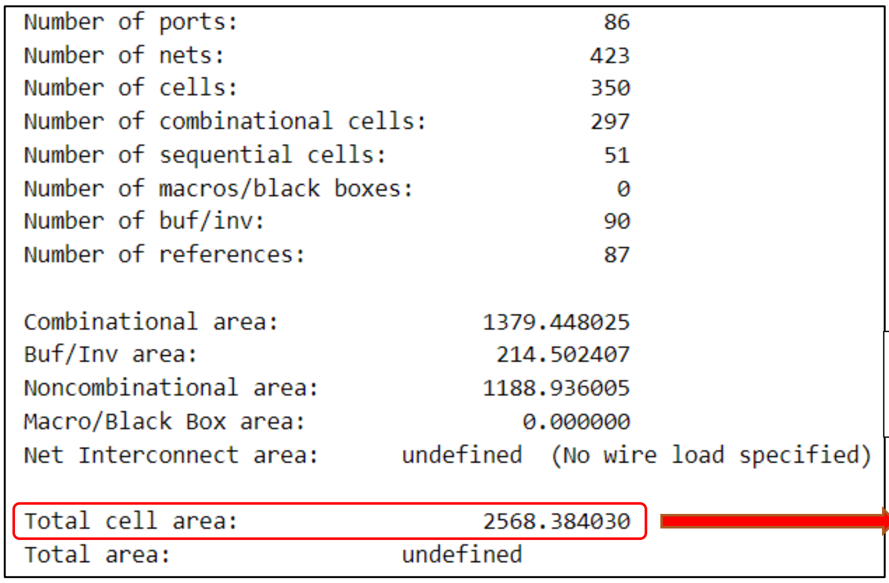


```
Global Operating Voltage = 0.9
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  = 764.3447 uW   (95%)
  Net Switching Power  =  43.1505 uW    (5%)
                         ---------
Total Dynamic Power    = 807.4951 uW  (100%)

Cell Leakage Power     =   8.5203 uW


                Internal         Switching           Leakage            Total
Power Group     Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad            0.0000           0.0000            0.0000            0.0000  (   0.00%)
memory            0.0000           0.0000            0.0000            0.0000  (   0.00%)
black_box         0.0000           0.0000            0.0000            0.0000  (   0.00%)
clock_network     0.0000           0.0000            0.0000            0.0000  (   0.00%)
register          0.7196       9.4971e-03        2.4979e+06            0.7316  (  89.65%)
sequential        0.0000           0.0000            0.0000            0.0000  (   0.00%)
combinational 4.4749e-02       3.3653e-02        6.0224e+06        8.4424e-02  (  10.35%)
--------------------------------------------------------------------------------------------------
Total             0.7643 mW    4.3150e-02 mW     8.5202e+06 pW         0.8160 mW
```

power dissipation = 0.8160 mW

(d) (20%) Add the scan chain into your gate-level netlist obtained by part(c), report the resulting gate count, the maximum operating speed (in MHz) of your circuit. Compare to the non-scan version, and report the **area overhead percentage** and **performance penalty** due to scan chain insertion.

Add scan chain – Gate level netlist

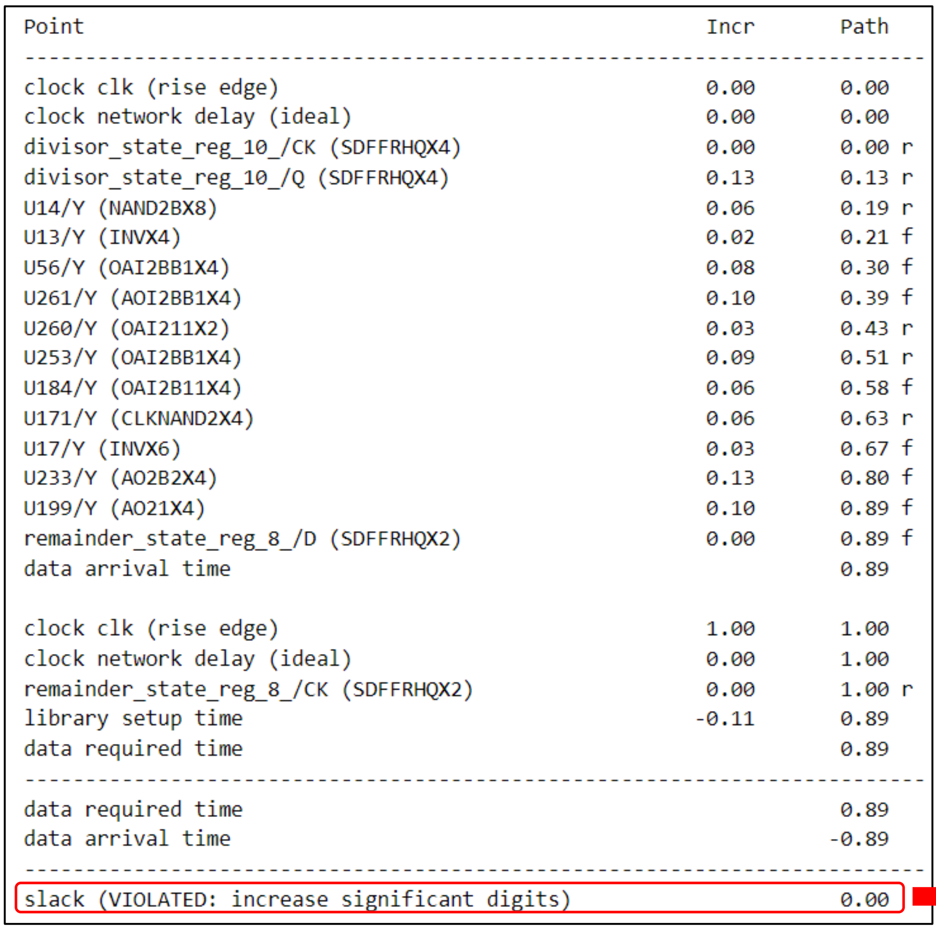


```
Number of ports:                           86
Number of nets:                           423
Number of cells:                          350
Number of combinational cells:            297
Number of sequential cells:                51
Number of macros/black boxes:               0
Number of buf/inv:                         90
Number of references:                      87

Combinational area:               1379.448025
Buf/Inv area:                      214.502407
Noncombinational area:            1188.936005
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  2568.384030
Total area:                 undefined
```

Gate count = $\frac{2568.38}{2.8224} = 909.99$

Area = 2568.38 μm^2

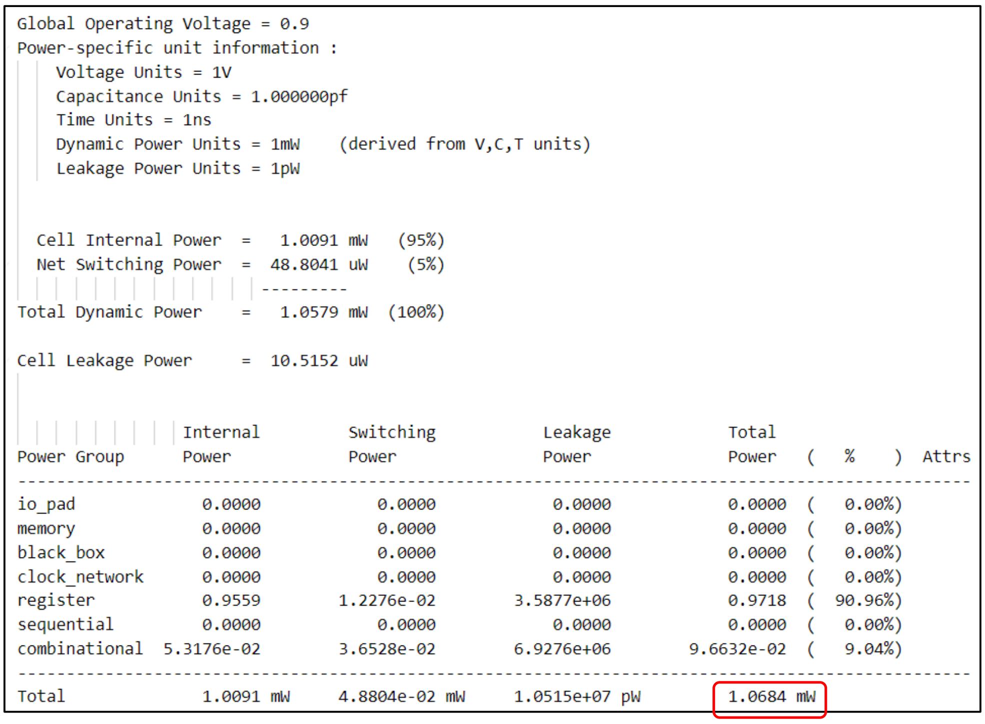
Add scan chain



```
Point                                                Incr       Path
-----------------------------------------------------------------------
clock clk (rise edge)                                0.00       0.00
clock network delay (ideal)                          0.00       0.00
divisor_state_reg_10_/CK (SDFFRHQX4)                 0.00       0.00 r
divisor_state_reg_10_/Q (SDFFRHQX4)                  0.13       0.13 r
U14/Y (NAND2BX8)                                     0.06       0.19 r
U13/Y (INVX4)                                        0.02       0.21 f
U56/Y (OAI2BB1X4)                                    0.08       0.30 f
U261/Y (AOI2BB1X4)                                   0.10       0.39 f
U260/Y (OAI211X2)                                    0.03       0.43 r
U253/Y (OAI2BB1X4)                                   0.09       0.51 r
U184/Y (OAI2B11X4)                                   0.06       0.58 f
U171/Y (CLKNAND2X4)                                  0.06       0.63 r
U17/Y (INVX6)                                        0.03       0.67 f
U233/Y (AO2B2X4)                                     0.13       0.80 f
U199/Y (AO21X4)                                      0.10       0.89 f
remainder_state_reg_8_/D (SDFFRHQX2)                 0.00       0.89 f
data arrival time                                               0.89

clock clk (rise edge)                                1.00       1.00
clock network delay (ideal)                          0.00       1.00
remainder_state_reg_8_/CK (SDFFRHQX2)                0.00       1.00 r
library setup time                                  -0.11       0.89
data required time                                              0.89
-----------------------------------------------------------------------
data required time                                              0.89
data arrival time                                              -0.89
-----------------------------------------------------------------------
slack (VIOLATED: increase significant digits)                   0.00
```

maximum operating speed = $\frac{1000}{1+|0|} = 1000$ MHz

Add scan chain - Power dissipation



```
Global Operating Voltage = 0.9
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =   1.0091 mW   (95%)
  Net Switching Power  =  48.8041 uW    (5%)
                         ---------
Total Dynamic Power    =   1.0579 mW  (100%)

Cell Leakage Power     =  10.5152 uW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000            0.0000            0.0000  (   0.00%)
memory             0.0000           0.0000            0.0000            0.0000  (   0.00%)
black_box          0.0000           0.0000            0.0000            0.0000  (   0.00%)
clock_network      0.0000           0.0000            0.0000            0.0000  (   0.00%)
register           0.9559       1.2276e-02        3.5877e+06            0.9718  (  90.96%)
sequential         0.0000           0.0000            0.0000            0.0000  (   0.00%)
combinational  5.3176e-02       3.6528e-02        6.9276e+06        9.6632e-02  (   9.04%)
--------------------------------------------------------------------------------------------------
Total              1.0091 mW    4.8804e-02 mW     1.0515e+07 pW         1.0684 mW
```

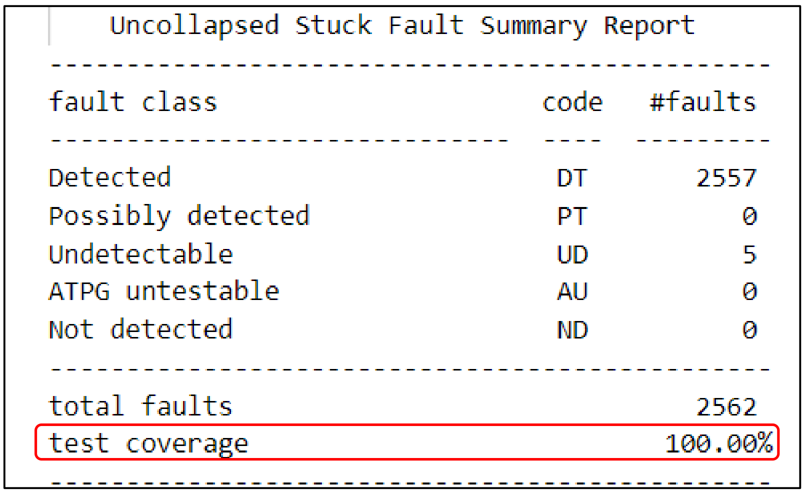
power dissipation = 1.0684 mW

Compare

	Gate count	Area	maximum operating speed (MHz)	Power (mW)
Non-scan chain	781.75	2206.41	1000	0.82
Add scan chain	909.99	2568.38	1000	1.07
Performance penalty	+16.40%		0.00%	+30.48%
Area overhead percentage		85.91%		

(e) (20%) Run **ATPG** using a commercial tool available and report the **fault coverage**.

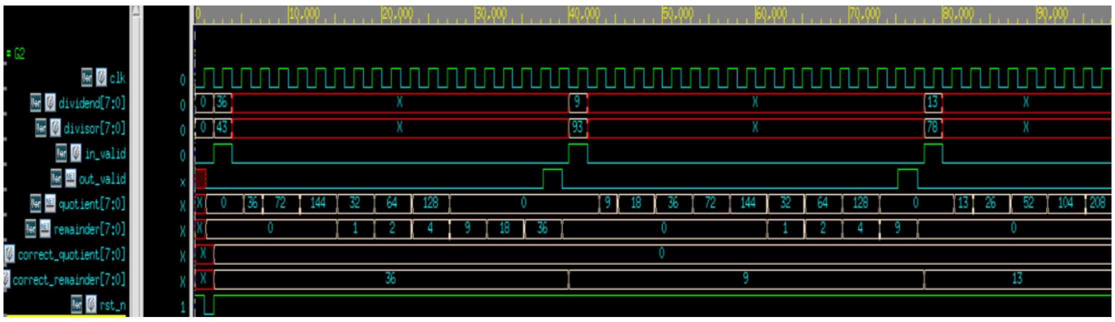
Fault report

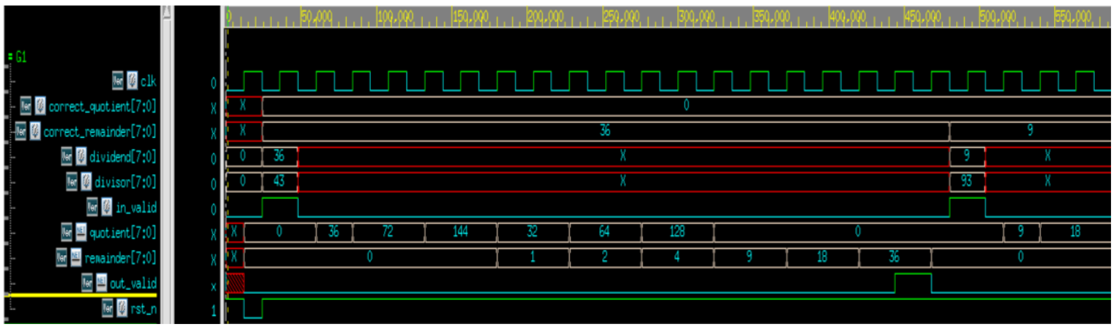


```
   Uncollapsed Stuck Fault Summary Report
-----------------------------------------------
fault class                     code   #faults
------------------------------  ----  ---------
Detected                         DT       2557
Possibly detected                PT          0
Undetectable                     UD          5
ATPG untestable                  AU          0
Not detected                     ND          0
-----------------------------------------------
total faults                              2562
test coverage                          100.00%
-----------------------------------------------
```

Fault coverage 100%

Verdi – nWave





Discussion

在寫 RTL code 的時候要注意在不同 always 裡面輸入的 reg 不可以有重複，這會導致在合成的時候出現問題，因為是第一次寫 RTL code，所以在寫的時候沒有注意到導致浪費不少時間在修改，還有在寫 code 的的時候盡量條件寫的越詳細越好，這樣在合成的時候面積可以再減少。

這次我們使用 Finite State Machine 配合移位法來寫除法器，使用了 4 個 states 分別是歸零、移位、比較大小後相減加 1 跟輸出，演算法則是先位移，再來比大小如果大於等於就相減加一，沒有的話就回位移，總共做 8 次，最後寫出來的面積是 2000 左右。透過討論後我們覺得面積還是有點大，我們認為在位移跟比較大小的那兩個 states 因該可以合併成一個可以減少面積，但是我們發現使用 4 個 states 的方法可以把 time period 調的更小讓頻率可以上升。結論是用比較少的 state 可以減少面積但是頻率較低功耗也較少，用較多的 state 面積大但頻率可以調大。