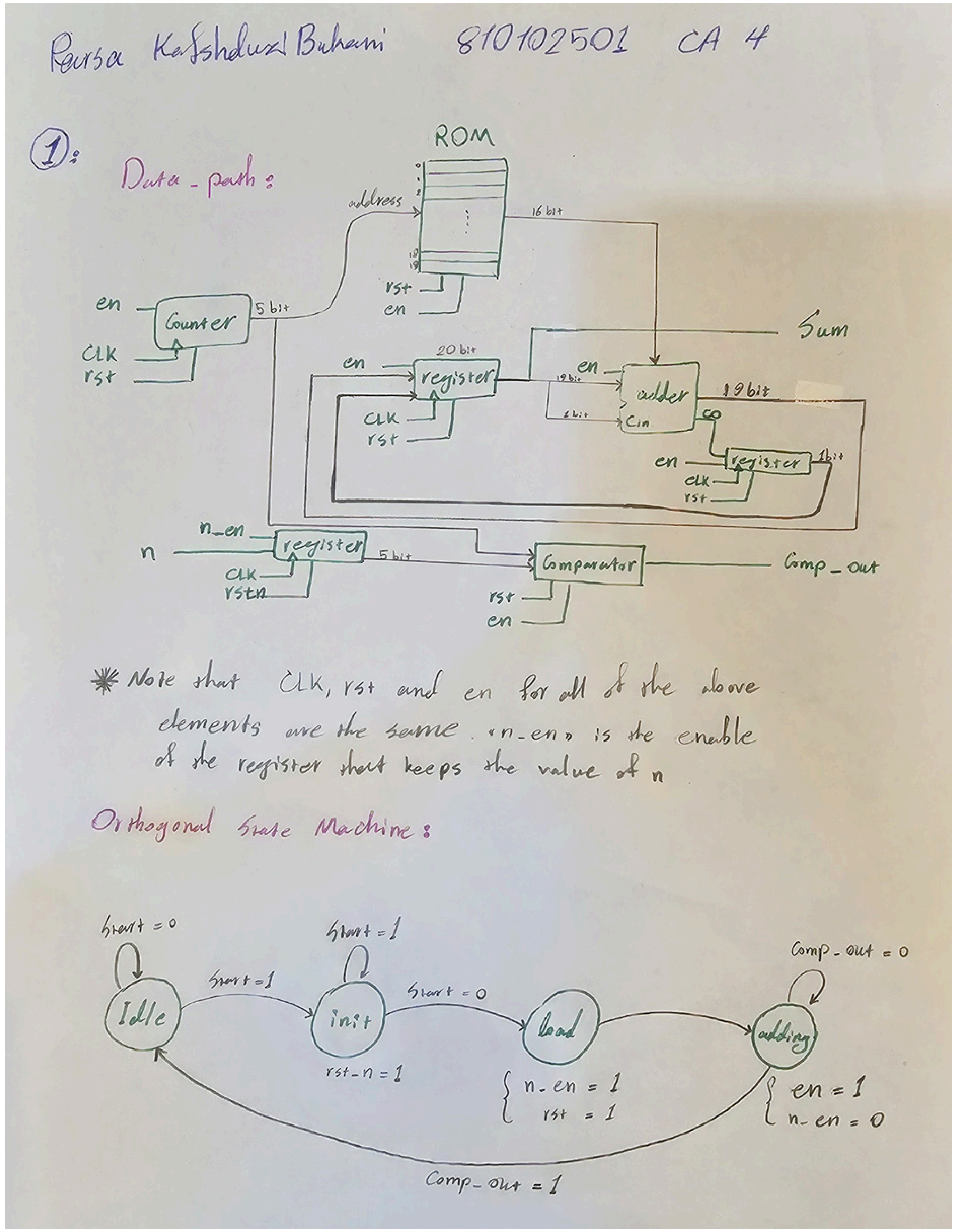
Parsa KafshduziBukani

PROJECT REPORT

PRESENTATION

CA4

Part A:



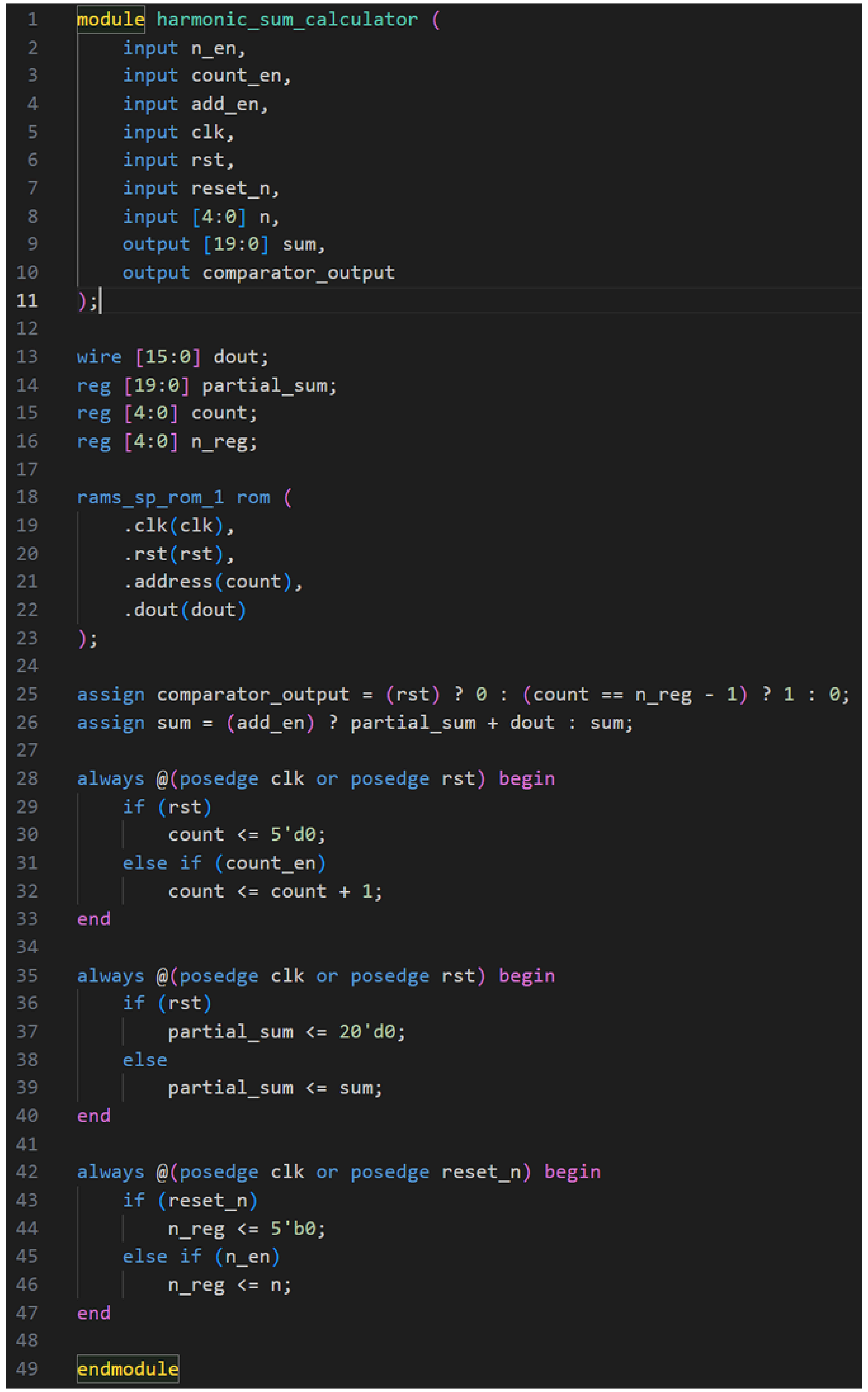
Paria Kafshduzi Bahani 810102501 CA 4

①: Data_path:

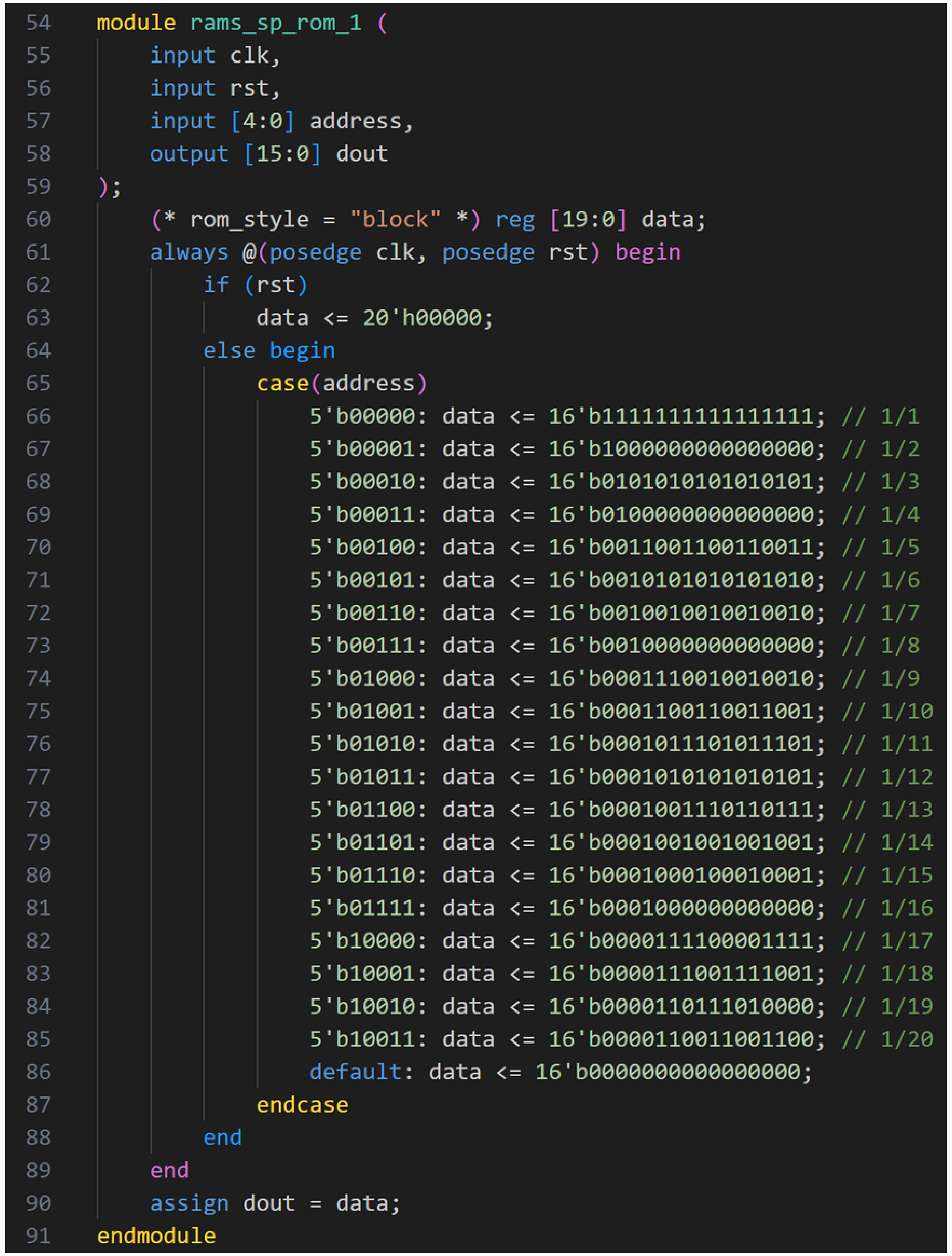
✳ Note that CLK, rst and en for all of the above elements are the same. «n_en» is the enable of the register that keeps the value of n

Orthogonal State Machine:

Part B (Data Path):

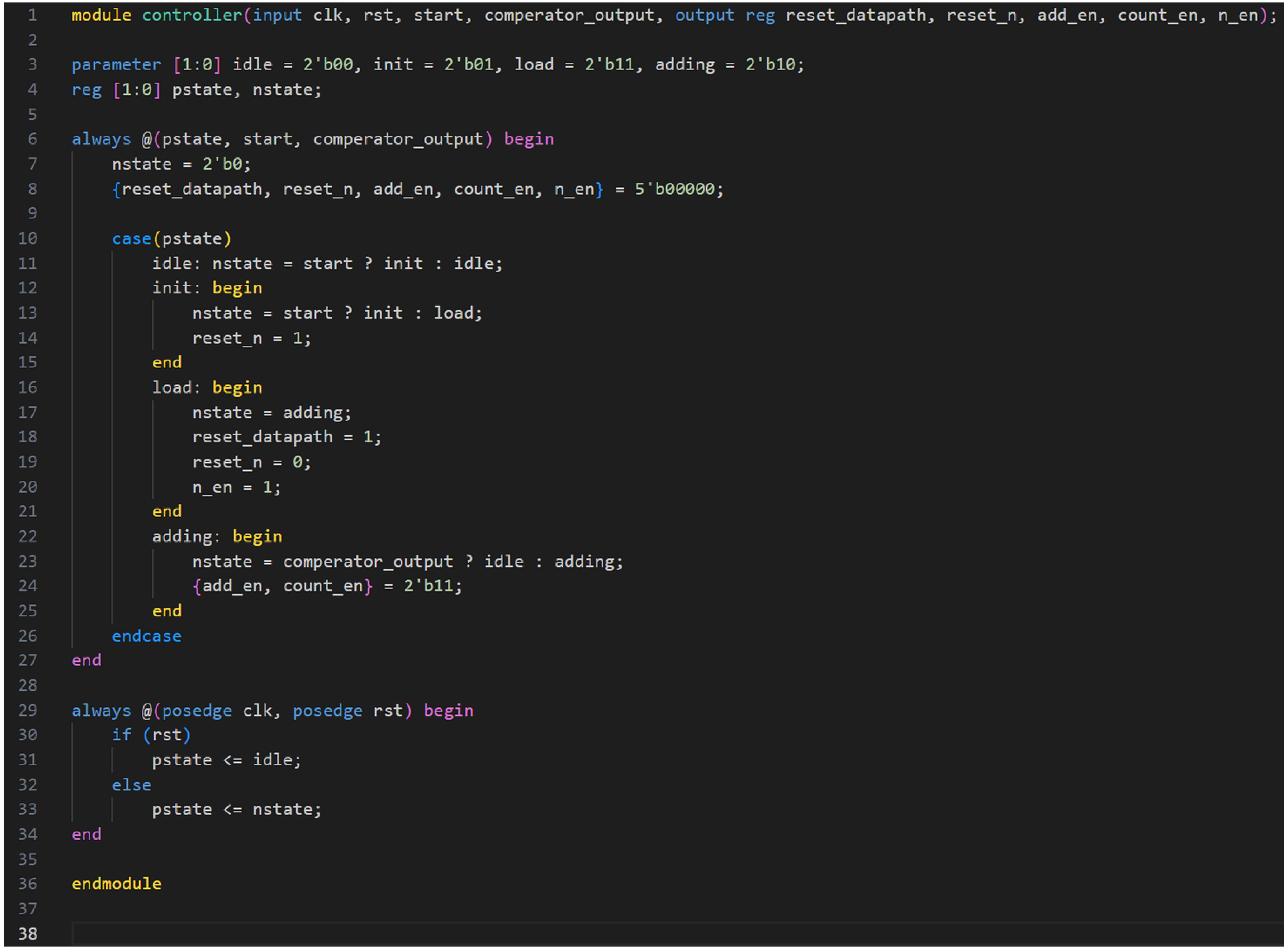


```verilog
module harmonic_sum_calculator (
    input n_en,
    input count_en,
    input add_en,
    input clk,
    input rst,
    input reset_n,
    input [4:0] n,
    output [19:0] sum,
    output comparator_output
);

wire [15:0] dout;
reg [19:0] partial_sum;
reg [4:0] count;
reg [4:0] n_reg;

rams_sp_rom_1 rom (
    .clk(clk),
    .rst(rst),
    .address(count),
    .dout(dout)
);

assign comparator_output = (rst) ? 0 : (count == n_reg - 1) ? 1 : 0;
assign sum = (add_en) ? partial_sum + dout : sum;

always @(posedge clk or posedge rst) begin
    if (rst)
        count <= 5'd0;
    else if (count_en)
        count <= count + 1;
end

always @(posedge clk or posedge rst) begin
    if (rst)
        partial_sum <= 20'd0;
    else
        partial_sum <= sum;
end

always @(posedge clk or posedge reset_n) begin
    if (reset_n)
        n_reg <= 5'b0;
    else if (n_en)
        n_reg <= n;
end

endmodule
```



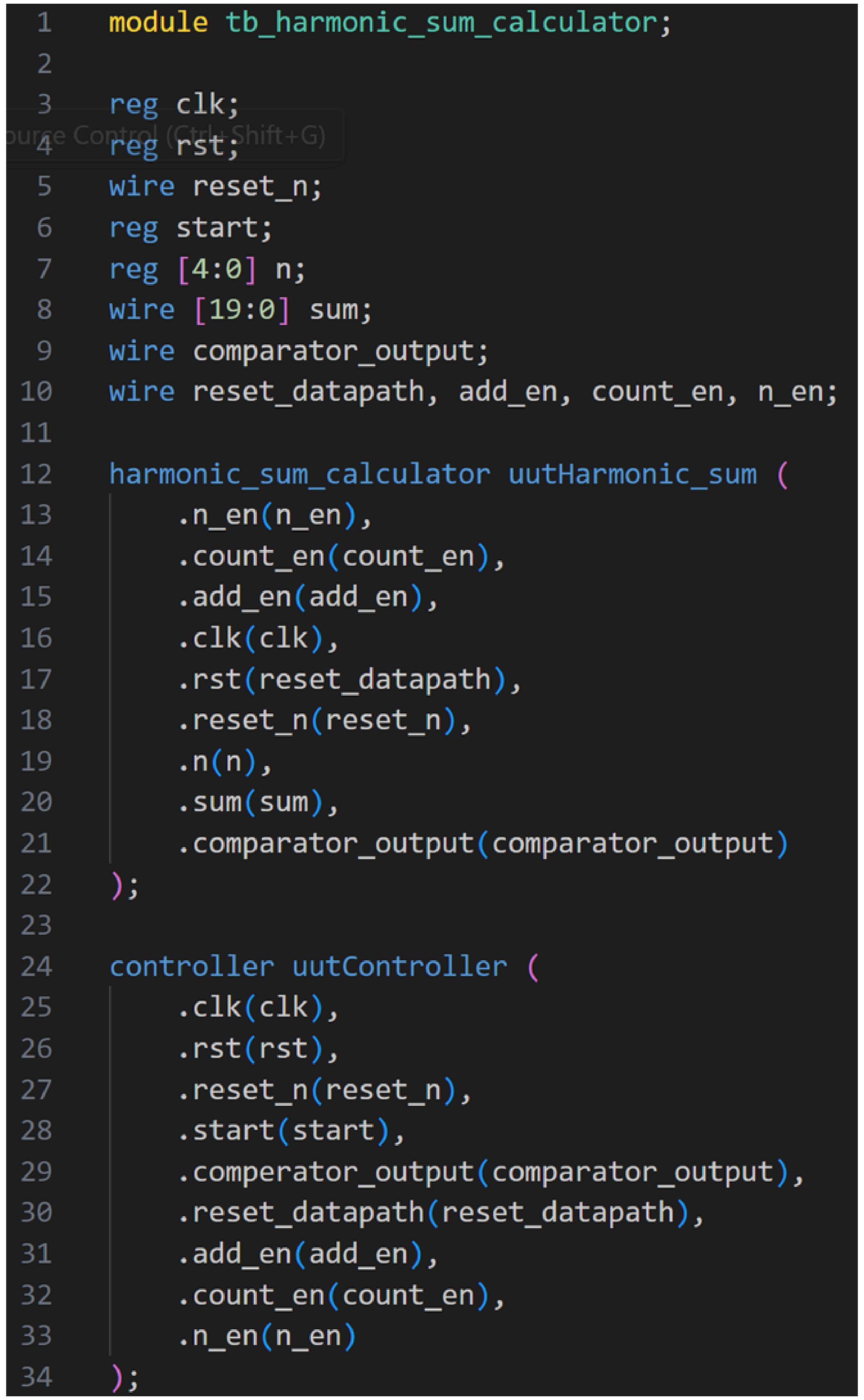
```verilog
module rams_sp_rom_1 (
    input clk,
    input rst,
    input [4:0] address,
    output [15:0] dout
);
    (* rom_style = "block" *) reg [19:0] data;
    always @(posedge clk, posedge rst) begin
        if (rst)
            data <= 20'h00000;
        else begin
            case(address)
                5'b00000: data <= 16'b1111111111111111; // 1/1
                5'b00001: data <= 16'b1000000000000000; // 1/2
                5'b00010: data <= 16'b0101010101010101; // 1/3
                5'b00011: data <= 16'b0100000000000000; // 1/4
                5'b00100: data <= 16'b0011001100110011; // 1/5
                5'b00101: data <= 16'b0010101010101010; // 1/6
                5'b00110: data <= 16'b0010010010010010; // 1/7
                5'b00111: data <= 16'b0010000000000000; // 1/8
                5'b01000: data <= 16'b0001110010010010; // 1/9
                5'b01001: data <= 16'b0001100110011001; // 1/10
                5'b01010: data <= 16'b0001011101011101; // 1/11
                5'b01011: data <= 16'b0001010101010101; // 1/12
                5'b01100: data <= 16'b0001001110110111; // 1/13
                5'b01101: data <= 16'b0001001001001001; // 1/14
                5'b01110: data <= 16'b0001000100010001; // 1/15
                5'b01111: data <= 16'b0001000000000000; // 1/16
                5'b10000: data <= 16'b0000111100001111; // 1/17
                5'b10001: data <= 16'b0000111001111001; // 1/18
                5'b10010: data <= 16'b0000110111010000; // 1/19
                5'b10011: data <= 16'b0000110011001100; // 1/20
                default: data <= 16'b0000000000000000;
            endcase
        end
    end
    assign dout = data;
endmodule
```

Part B(Controller):

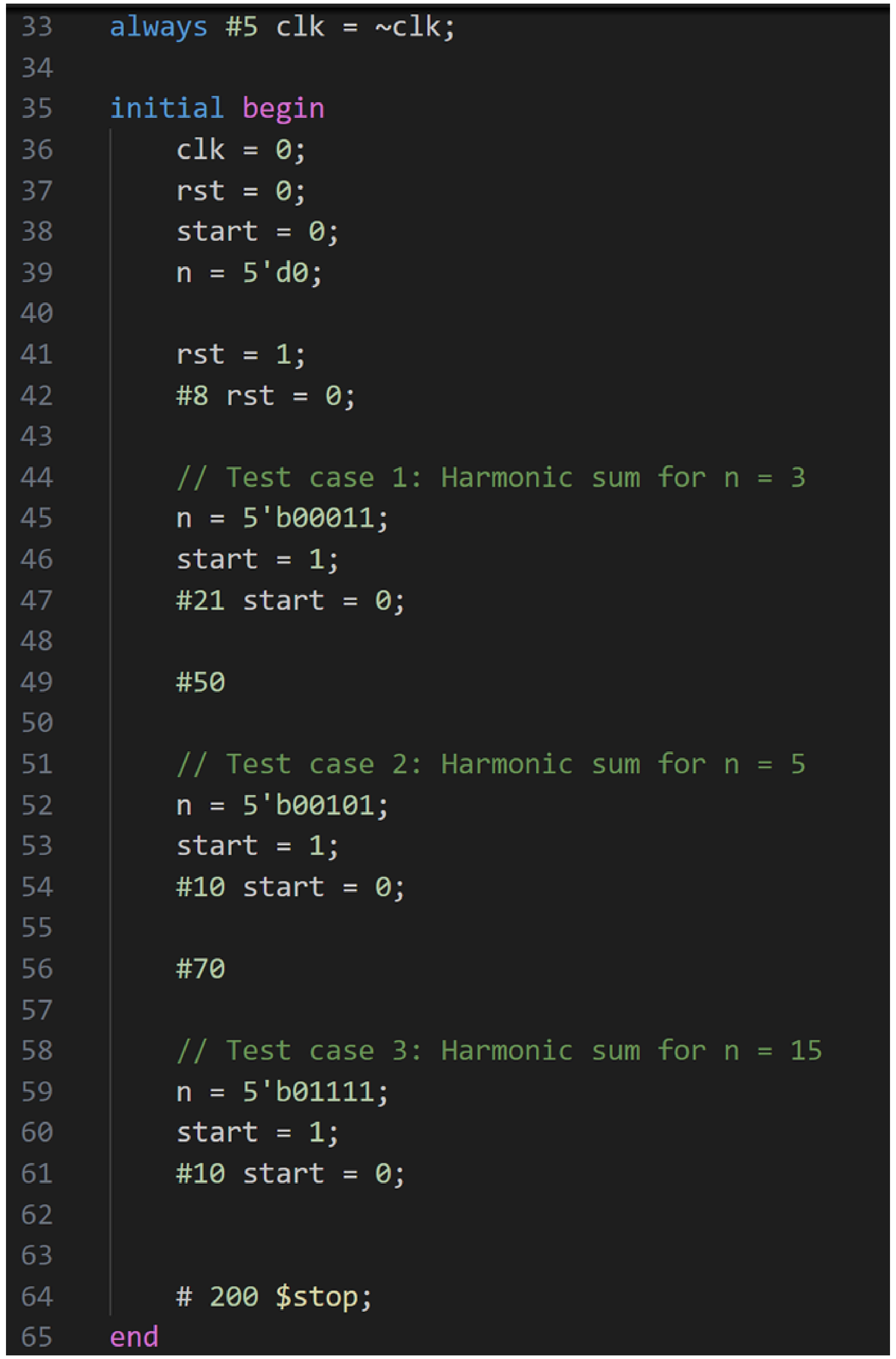


```
module controller(input clk, rst, start, comperator_output, output reg reset_datapath, reset_n, add_en, count_en, n_en);

parameter [1:0] idle = 2'b00, init = 2'b01, load = 2'b11, adding = 2'b10;
reg [1:0] pstate, nstate;

always @(pstate, start, comperator_output) begin
    nstate = 2'b0;
    {reset_datapath, reset_n, add_en, count_en, n_en} = 5'b00000;

    case(pstate)
        idle: nstate = start ? init : idle;
        init: begin
            nstate = start ? init : load;
            reset_n = 1;
        end
        load: begin
            nstate = adding;
            reset_datapath = 1;
            reset_n = 0;
            n_en = 1;
        end
        adding: begin
            nstate = comperator_output ? idle : adding;
            {add_en, count_en} = 2'b11;
        end
    endcase
end

always @(posedge clk, posedge rst) begin
    if (rst)
        pstate <= idle;
    else
        pstate <= nstate;
end

endmodule

```

Part B(Test Bench):

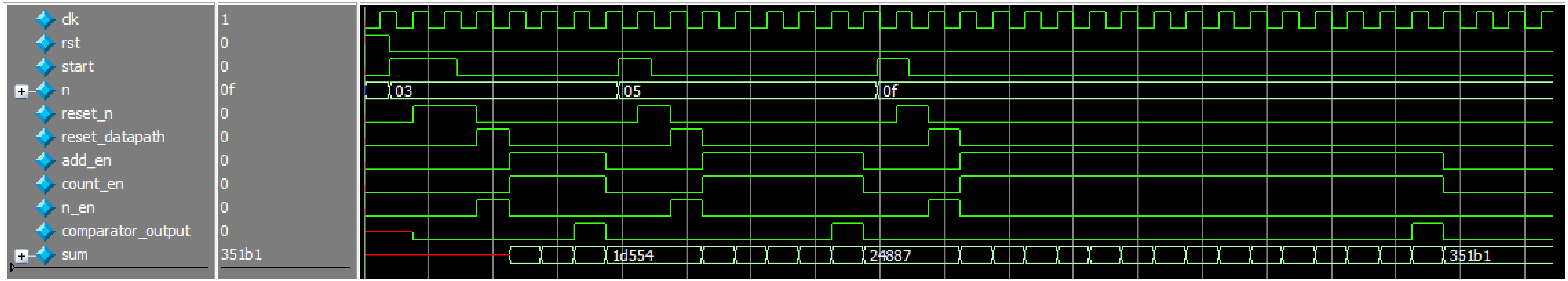


```verilog
module tb_harmonic_sum_calculator;

reg clk;
reg rst;
wire reset_n;
reg start;
reg [4:0] n;
wire [19:0] sum;
wire comparator_output;
wire reset_datapath, add_en, count_en, n_en;

harmonic_sum_calculator uutHarmonic_sum (
    .n_en(n_en),
    .count_en(count_en),
    .add_en(add_en),
    .clk(clk),
    .rst(reset_datapath),
    .reset_n(reset_n),
    .n(n),
    .sum(sum),
    .comparator_output(comparator_output)
);

controller uutController (
    .clk(clk),
    .rst(rst),
    .reset_n(reset_n),
    .start(start),
    .comperator_output(comparator_output),
    .reset_datapath(reset_datapath),
    .add_en(add_en),
    .count_en(count_en),
    .n_en(n_en)
);
```



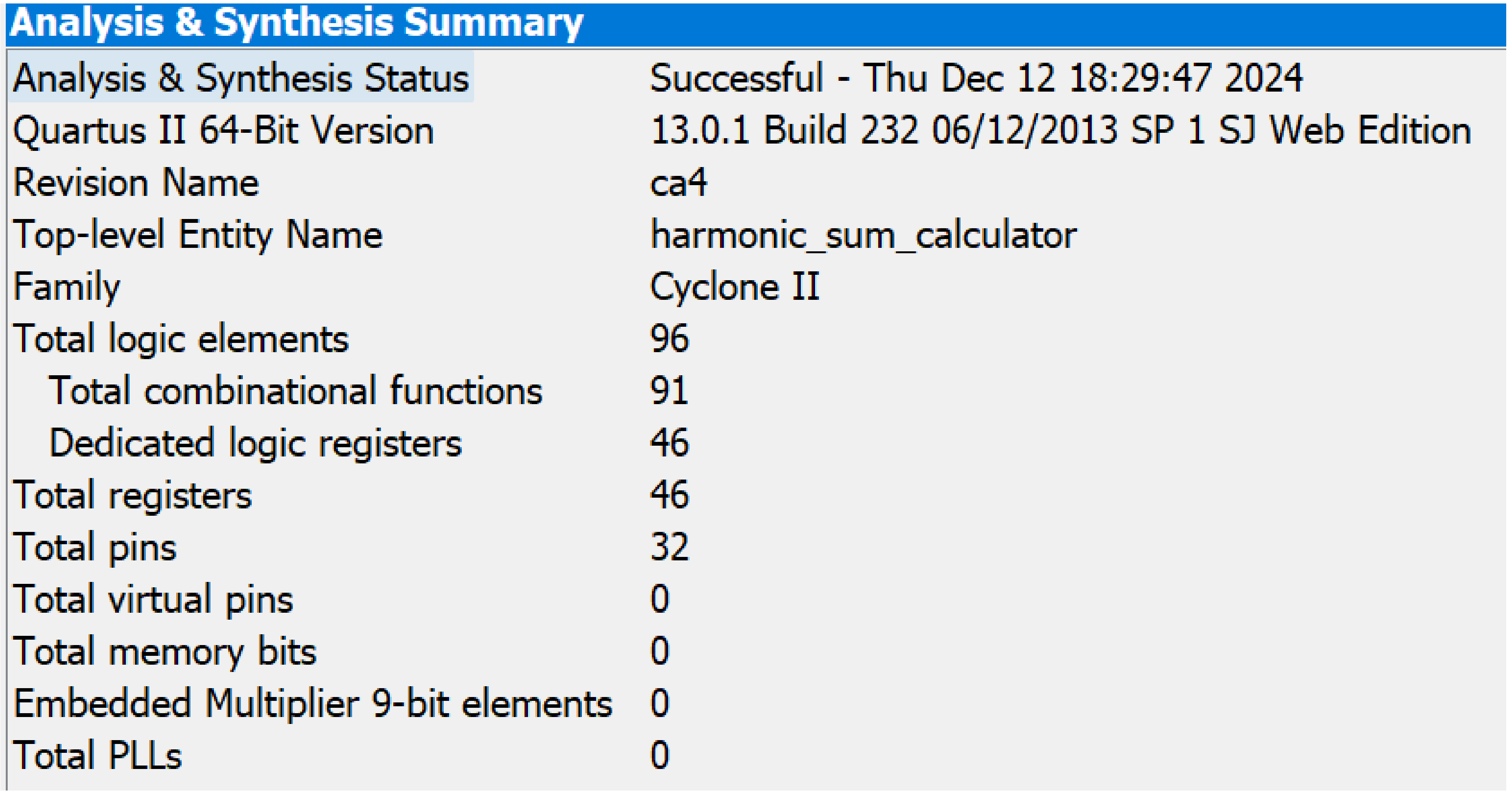
```verilog
always #5 clk = ~clk;

initial begin
    clk = 0;
    rst = 0;
    start = 0;
    n = 5'd0;

    rst = 1;
    #8 rst = 0;

    // Test case 1: Harmonic sum for n = 3
    n = 5'b00011;
    start = 1;
    #21 start = 0;

    #50

    // Test case 2: Harmonic sum for n = 5
    n = 5'b00101;
    start = 1;
    #10 start = 0;

    #70

    // Test case 3: Harmonic sum for n = 15
    n = 5'b01111;
    start = 1;
    #10 start = 0;


    # 200 $stop;
end
```

Part B(Simulation results):



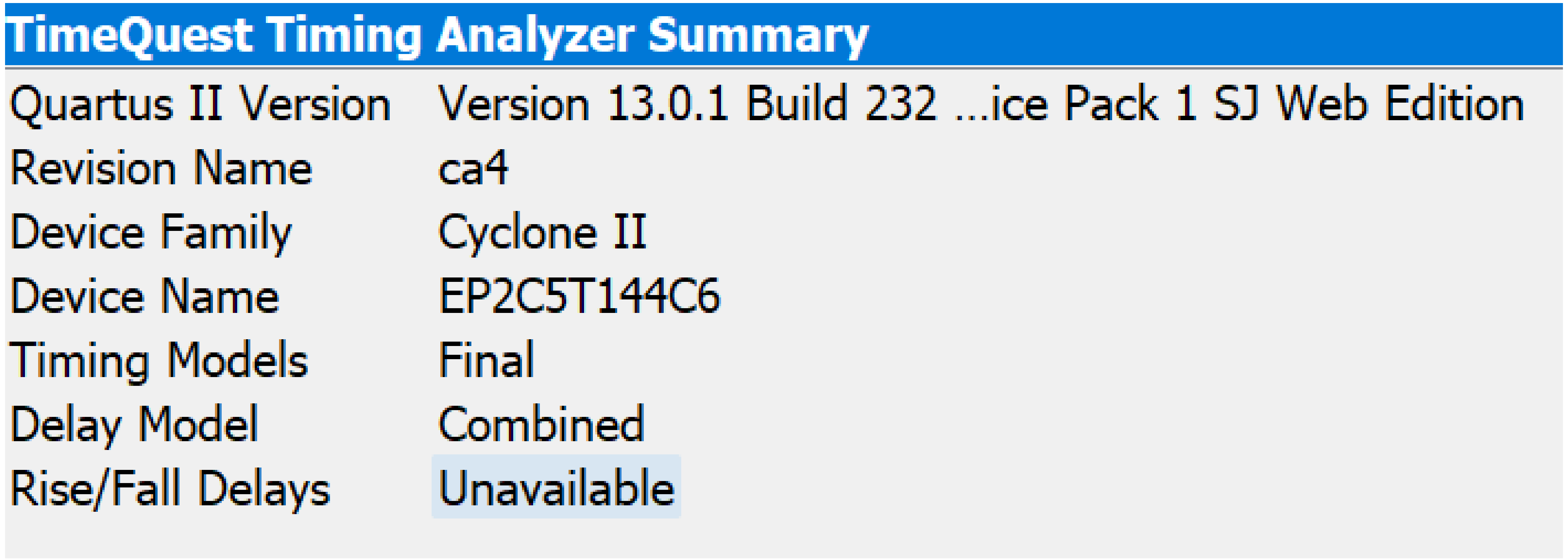
clk	1
rst	0
start	0
n	0f
reset_n	0
reset_datapath	0
add_en	0
count_en	0
n_en	0
comparator_output	0
sum	351b1

Part C:



Analysis & Synthesis Summary

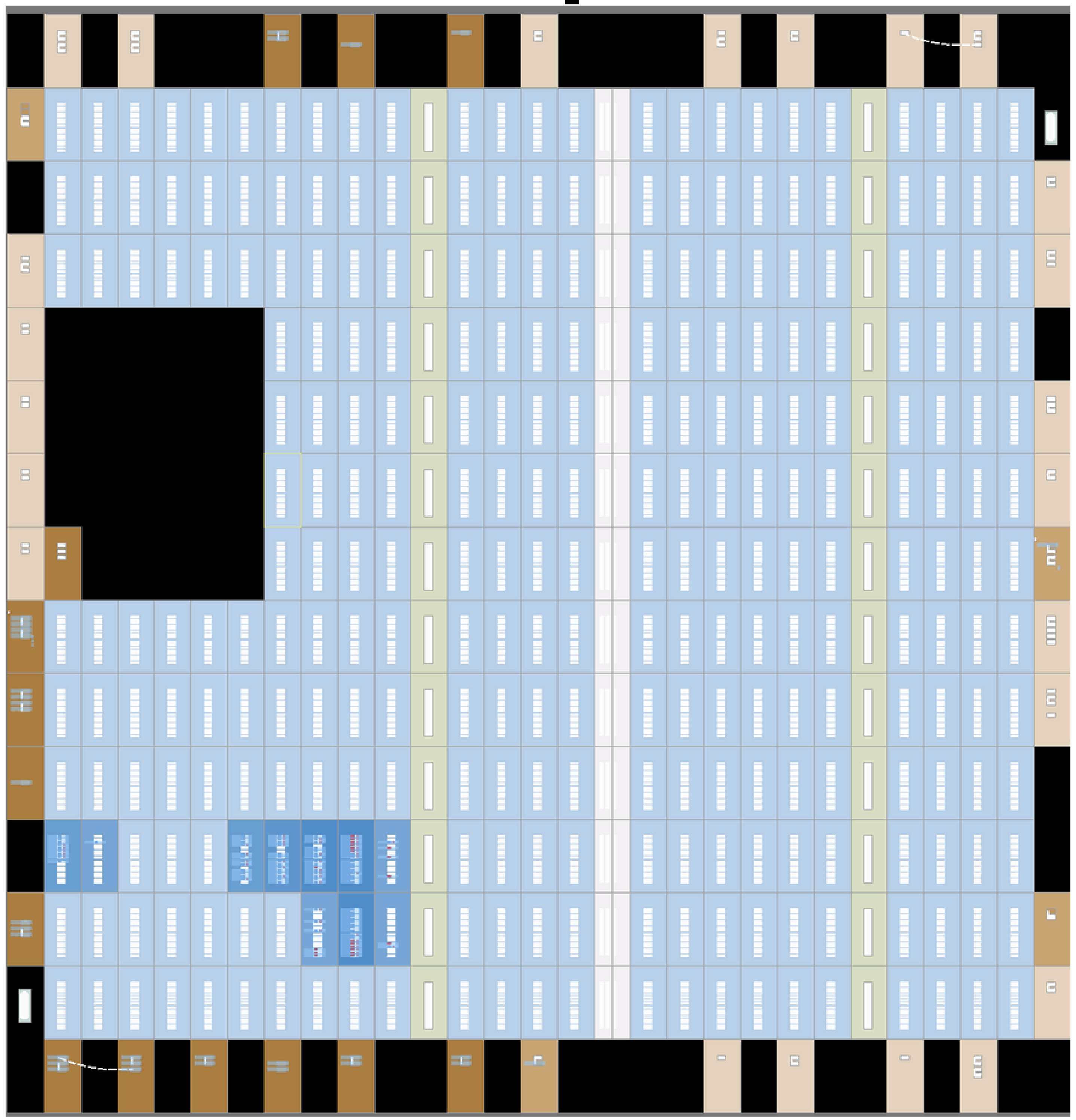
Analysis & Synthesis Status	Successful - Thu Dec 12 18:29:47 2024
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	ca4
Top-level Entity Name	harmonic_sum_calculator
Family	Cyclone II
Total logic elements	96
Total combinational functions	91
Dedicated logic registers	46
Total registers	46
Total pins	32
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0



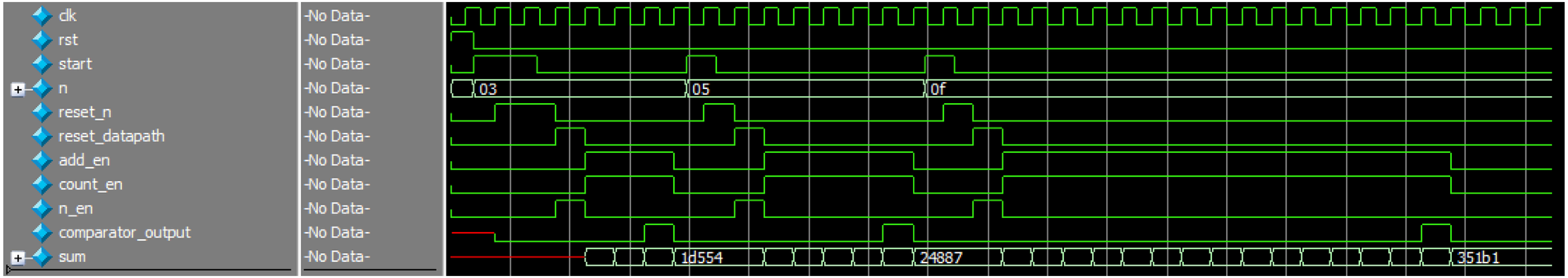
TimeQuest Timing Analyzer Summary

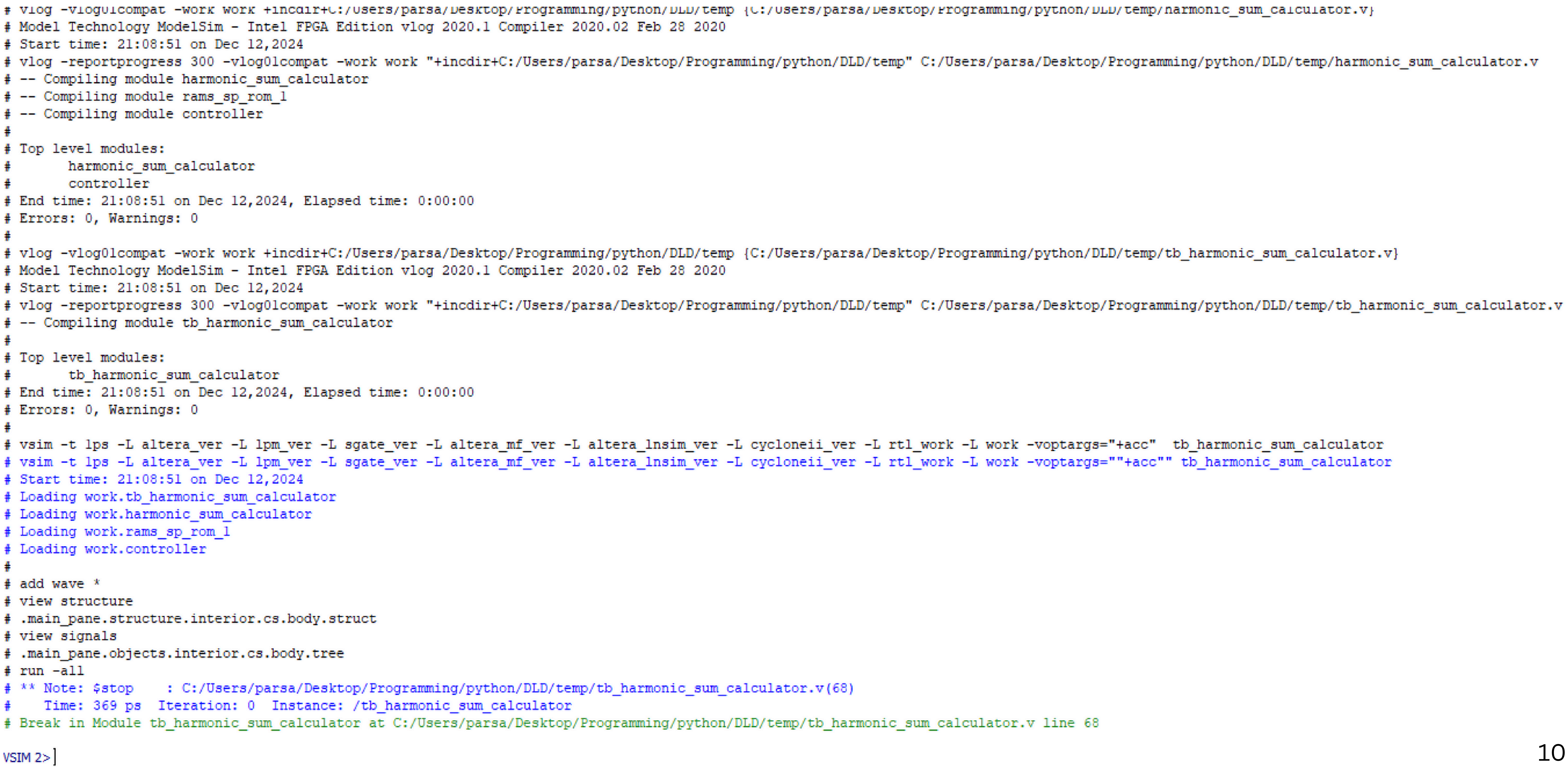
Quartus II Version	Version 13.0.1 Build 232 ...ice Pack 1 SJ Web Edition
Revision Name	ca4
Device Family	Cyclone II
Device Name	EP2C5T144C6
Timing Models	Final
Delay Model	Combined
Rise/Fall Delays	Unavailable

Floor plan:



Post Synthesis simulation





```
# vlog -vlog01compat -work work +incdir+C:/Users/parsa/Desktop/Programming/python/DLD/temp {C:/Users/parsa/Desktop/Programming/python/DLD/temp/harmonic_sum_calculator.v}
# Model Technology ModelSim - Intel FPGA Edition vlog 2020.1 Compiler 2020.02 Feb 28 2020
# Start time: 21:08:51 on Dec 12,2024
# vlog -reportprogress 300 -vlog01compat -work work "+incdir+C:/Users/parsa/Desktop/Programming/python/DLD/temp" C:/Users/parsa/Desktop/Programming/python/DLD/temp/harmonic_sum_calculator.v
# -- Compiling module harmonic_sum_calculator
# -- Compiling module rams_sp_rom_1
# -- Compiling module controller
#
# Top level modules:
#       harmonic_sum_calculator
#       controller
# End time: 21:08:51 on Dec 12,2024, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
#
# vlog -vlog01compat -work work +incdir+C:/Users/parsa/Desktop/Programming/python/DLD/temp {C:/Users/parsa/Desktop/Programming/python/DLD/temp/tb_harmonic_sum_calculator.v}
# Model Technology ModelSim - Intel FPGA Edition vlog 2020.1 Compiler 2020.02 Feb 28 2020
# Start time: 21:08:51 on Dec 12,2024
# vlog -reportprogress 300 -vlog01compat -work work "+incdir+C:/Users/parsa/Desktop/Programming/python/DLD/temp" C:/Users/parsa/Desktop/Programming/python/DLD/temp/tb_harmonic_sum_calculator.v
# -- Compiling module tb_harmonic_sum_calculator
#
# Top level modules:
#       tb_harmonic_sum_calculator
# End time: 21:08:51 on Dec 12,2024, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
#
# vsim -t 1ps -L altera_ver -L lpm_ver -L sgate_ver -L altera_mf_ver -L altera_lnsim_ver -L cycloneii_ver -L rtl_work -L work -voptargs="+acc"  tb_harmonic_sum_calculator
# vsim -t 1ps -L altera_ver -L lpm_ver -L sgate_ver -L altera_mf_ver -L altera_lnsim_ver -L cycloneii_ver -L rtl_work -L work -voptargs=""+acc"" tb_harmonic_sum_calculator
# Start time: 21:08:51 on Dec 12,2024
# Loading work.tb_harmonic_sum_calculator
# Loading work.harmonic_sum_calculator
# Loading work.rams_sp_rom_1
# Loading work.controller
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# ** Note: $stop    : C:/Users/parsa/Desktop/Programming/python/DLD/temp/tb_harmonic_sum_calculator.v(68)
#    Time: 369 ps  Iteration: 0  Instance: /tb_harmonic_sum_calculator
# Break in Module tb_harmonic_sum_calculator at C:/Users/parsa/Desktop/Programming/python/DLD/temp/tb_harmonic_sum_calculator.v line 68

VSIM 2>
```

Part D:

- ## Functional Consistency:

The functional behavior remained consistent between pre- and post-synthesis simulations, indicating a successful synthesis process.

- ## Timing Differences:

The post-synthesis timing analysis revealed critical paths and timing violations that were not present in the pre-synthesis analysis due to the idealized conditions