CSE 4207 CT 4 Assignment
Roll No: 1903113

Assignment Problem:
Category: C
Word Size = 6
ALU Operations = ADD, OR_XOR_AND

Solution:
Video:

Have you uploaded the video?	YES
Check List 1: Have you explained the design using testbenches to prove that your circuit is working correctly and giving correct results?	YES
Check List 2: Have you shown full synthesis results showing all the required info including RTL Synthesis, RTL Floorplan, RTL Power Analysis, GDS and Heatmap (for details, see assignment template doc)?	YES
NB: Failing to upload video will cause heavy point penalty (5-6 Marks)	
NB: Failing to add any required info will cause point penalty (1-2 Marks)	

HDL Code:

Check List: Have you added all the modules written in verilog including ALU, ALU_OP1, ALU_OP2, CONTROLLER, TOP, TOP_TESTBENCH, ALU_TESTBENCH, CONTROLLER_TESTBENCH?	YES
NB: Failing to add any required info will cause point penalty (1-2 Marks)	

Google Drive link for the files:

19CT4Assignment_codes_1903113

alu.v

```
module ALU
(
    input wire [5:0] A, B,
    input wire [1:0] OP,
    output wire [5:0] R,
    output wire alu_flag
);

    reg [5:0] result;
    wire [5:0] R_ADD, XOR;
```

```
    wire CF_ADD;

    // Instantiate modules
    ALU_ADD_SUB_Nbit  ADD_SUB1(A,B,R_ADD,CF_ADD);

    OR_XOR_AND_Nbit OR_XOR_AND1 (A,B,XOR);

    always @(*)
    begin
        case (OP)
            2'b00: // Addition
                begin
                    result = R_ADD;
                end
            2'b01: // OR_XOR_AND
                begin
                    result = XOR;
                end
            default:
                begin
                    result = 6'b000000;
                end
        endcase
    end
    assign R = result;
    assign alu_flag = (result > 6'b0);
endmodule
```

alu_op1.v

```
module OR_XOR_AND_Nbit
(
    input wire [5:0] A,B,
    output wire [5:0] R
);

    assign R = ((A|B)^(A&B));

endmodule
```

alu_op2.v

```
module ALU_ADD_SUB_Nbit
(
    input wire [5:0] A,B,
    output wire [5:0] R,
    output wire CF
```

```
);

    assign {CF, R} = A + B;


endmodule
```

controller.v

```
module controller
(
    input wire clk, reset, start,
    input wire [5:0] a, b,
    input wire [1:0] op,

    input wire [5:0] alu_out,
    input wire alu_flag,
    output wire [5:0] alu_in1, alu_in2,
    output wire [1:0] alu_op,

    output reg [5:0] result,
    output reg flag
);

reg [2:0] pstate, nstate;

parameter [2:0] START = 3'b000,
                ONE = 3'b001,
                TWO = 3'b010,
                THREE = 3'b011,
                FINISH = 3'b100;

//Memory
always @(posedge clk, posedge reset)
begin : PSR
    if (reset)
    begin
        pstate <= START;
    end
    else
    begin
        pstate <= nstate;
    end
end

//Next State Logic
always @(*)
```

```
begin: NSOL
    // Monitor output
    if(pstate == START)
        $monitor("pstate = START -> clk = %b, reset = %b,
start = %b, a = %d, b = %d, op = %b, result = %d, flag =
%b\n************************************\n", clk, reset,
start, a, b, op, result, flag);
    else if(pstate == ONE)
        $monitor("pstate = ONE -> clk = %b, reset = %b, start
= %b, a = %d, b = %d, op = %b, result = %d, flag =
%b\n************************************\n", clk, reset,
start, a, b, op, result, flag);
    else if(pstate == TWO)
        $monitor("pstate = TWO -> clk = %b, reset = %b, start
= %b, a = %d, b = %d, op = %b, result = %d, flag =
%b\n************************************\n", clk, reset,
start, a, b, op, result, flag);
    else if(pstate == THREE)
        $monitor("pstate = THREE -> clk = %b, reset = %b,
start = %b, a = %d, b = %d, op = %b, result = %d, flag =
%b\n************************************\n", clk, reset,
start, a, b, op, result, flag);
    else if(pstate == FINISH)
        $monitor("pstate = FINISH -> clk = %b, reset = %b,
start = %b, a = %d, b = %d, op = %b, result = %d, flag =
%b\n************************************\n", clk, reset,
start, a, b, op, result, flag);

    nstate = pstate;

    // Next State Logic and Output Logic
    begin: NSL
    case (pstate)
        START:
        begin
            if (start)
                nstate = ONE;
        end

        ONE:
        begin
            nstate = TWO;
        end

        TWO:
        begin
            nstate = THREE;
```

```
        end

        THREE:
        begin
            nstate = FINISH;
        end

        FINISH:
        begin
            nstate = FINISH;
        end

        default:
            nstate = START;
    endcase
end
    result = 6'b000000;
    flag = 1'b0;
begin: OL
    case (pstate)
        START:
        begin
            result = 6'b0;
            flag = 0;
        end

        ONE:
        begin
            result = alu_out;
            flag = alu_flag;
        end

        TWO:
        begin
            result = alu_out;
            flag = alu_flag;
        end

        THREE:
        begin
            result = alu_out;
            flag = alu_flag;
        end

        FINISH:
        begin
            result = alu_out;
```

```
            flag = alu_flag;
        end
        default:
            nstate = START;
    endcase
end
end

assign alu_in1 = a;
assign alu_in2 = b;
assign alu_op = op;

endmodule
```

top.v

```
module top
(
    input wire clk, reset, start,
    input wire [5:0] a, b,
    input wire [1:0] op,
    output wire [5:0] result,
    output wire flag
);

wire [5:0] alu_in1, alu_in2;
wire [1:0] alu_op;
wire [5:0] alu_out;
wire alu_flag;

controller controller1
(
    .clk(clk),
    .reset(reset),
    .start(start),
    .a(a),
    .b(b),
    .op(op),
    .alu_in1(alu_in1),
    .alu_in2(alu_in2),
    .alu_op(alu_op),
    .alu_out(alu_out),
    .alu_flag(alu_flag),
    .result(result),
    .flag(flag)
);
```

```
ALU datapath1
(
    .A(alu_in1),
    .B(alu_in2),
    .OP(alu_op),
    .R(alu_out),
    .alu_flag(alu_flag)
);

endmodule
```

top_testbench.v

```
`timescale 1ns/1ns

module top_tb;

reg clk, reset, start;
reg [5:0] a, b;
reg [1:0] op;
wire [5:0] result;
wire flag;

top top1
(
    .clk(clk),
    .reset(reset),
    .start(start),
    .a(a),
    .b(b),
    .op(op),
    .result(result),
    .flag(flag)
);

initial begin
    clk = 0;
    forever #5 clk = ~clk;  // Toggle clk every 5 time units
end

initial begin
    $dumpfile("top_test.vcd");
    $dumpvars(0, top_tb);

    clk <= 0;
    reset <= 0;
```

```
    start <= 0;
    a <= 0;
    b <= 0;
    op <= 0;

    @(negedge clk);
    reset <= 1;

    @(negedge clk);
    reset <= 0;
    start <= 1;
    a <= 5;
    b <= 3;
    op <= 2'b00; // Addition: 5 + 3 = 8 (flag = 1)
    #15;

    a <= 0;
    b <= 0;
    op <= 2'b00; // Addition: 0 + 0 = 0 (flag = 0)
    #10;

    a <= 10;
    b <= 7;
    op <= 2'b01; // OR_XOR_AND: (10 | 7) ^ (10 & 7) = 15 ^ 2
= 13 (flag = 1)
    #20;

    reset <= 1;
    #30;

    $finish();
end

endmodule
```

alu_testbench.v

```
`timescale 1ns/1ns

module alu_tb;
    reg [5:0] A, B;
    reg [1:0] OP;
    wire [5:0] R;
    wire alu_flag ;

    ALU uut (
        .A(A),
```

```
        .B(B),
        .OP(OP),
        .R(R),
        .alu_flag(alu_flag)
    );

    initial begin
        $dumpfile("alu_tb.vcd");
        $dumpvars(0, alu_tb);

        A = 6'b000100; B = 6'b000011; OP = 2'b00;
        #10;

        A = 6'b000100; B = 6'b000010; OP = 2'b01;
        #10;

        A = 6'b000111; B = 6'b000010; OP = 2'b10;
        #10;

        A = 6'b000110; B = 6'b000001; OP = 2'b11;
        #10;

        A = 6'b000111; B = 6'b000111; OP = 2'b00;
        #10;

        A = 6'b000001; B = 6'b000111; OP = 2'b01;
        #10;

        $finish;
    end

    initial begin
        $monitor("Time=%0t A=%b B=%b OP=%b -> R=%b
alu_flag=%b", $time, A, B, OP, R, alu_flag);
    end

endmodule
```

controller_testbench.v

```
`timescale 1ns/1ns

module controller_tb;
    reg clk;
    reg reset;
    reg start;
```

```
    reg [5:0] a;
    reg [5:0] b;
    reg [1:0] op;
    reg [5:0] alu_out;
    reg alu_flag;
    wire [5:0] alu_in1;
    wire [5:0] alu_in2;
    wire [1:0] alu_op;
    wire [5:0] result;
    wire flag;

    controller dut (
        .clk(clk),
        .reset(reset),
        .start(start),
        .a(a),
        .b(b),
        .op(op),
        .alu_out(alu_out),
        .alu_flag(alu_flag),
        .alu_in1(alu_in1),
        .alu_in2(alu_in2),
        .alu_op(alu_op),
        .result(result),
        .flag(flag)
    );

    initial begin
        clk = 0;
        forever #5 clk = ~clk;
    end

    // Stimulus
    initial begin

        reset = 0;
        start = 0;
        a = 6'd0;
        b = 6'd0;
        op = 2'b00;
        alu_out = 6'd0;
        alu_flag = 0;


        #10;
        reset = 1;
```

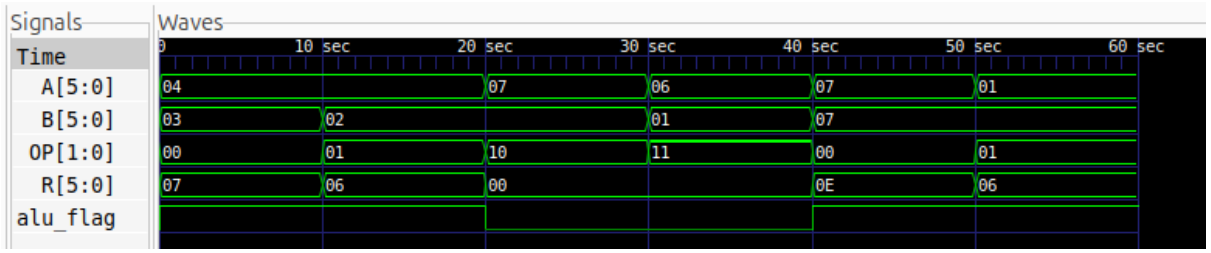
```
        #10;
        reset = 0;
        start = 1;
        a = 6'd5;
        b = 6'd3;
        op = 2'b00;
        alu_out = 6'd8;  // Addition: 5 + 3
        alu_flag = 1;


        #15;
        a = 6'd0;
        b = 6'd0;
        op = 2'b00;
        alu_out = 6'd0;  // Addition: 0 + 0
        alu_flag = 0;


        #10;
        a = 6'd10;
        b = 6'd7;
        op = 2'b01;
        alu_out = 6'd13; // OR_XOR_AND: (10 | 7) ^ (10 & 7) =
15 ^ 2 = 13
        alu_flag = 1;


        #20;
        $finish;
    end


    initial begin
        $monitor("Time=%0t | clk=%b | reset=%b | start=%b |
a=%d | b=%d | op=%b | alu_out=%d | alu_flag=%b | alu_in1=%d |
alu_in2=%d | alu_op=%b | result=%d | flag=%b",
                 $time, clk, reset, start, a, b, op, alu_out,
alu_flag, alu_in1, alu_in2, alu_op, result, flag);
    end


    initial begin
        $dumpfile("controller_tb.vcd");
        $dumpvars(0, controller_tb);
    end
```

```
endmodule
```

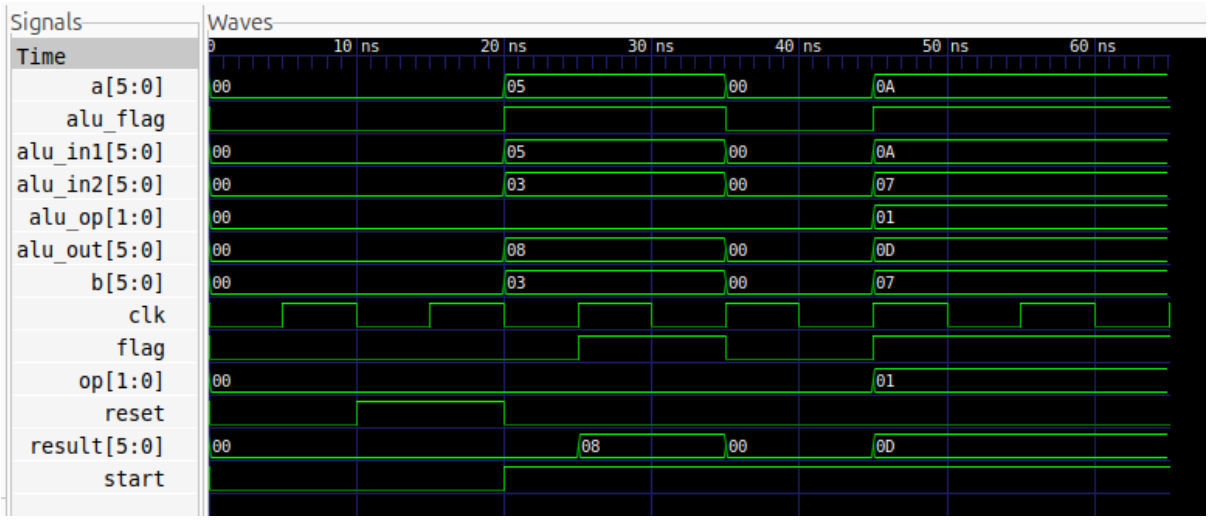
RTL Timing Diagram:

Check List: Have you added all the timing diagrams of ALU_TESTBENCH, CONTROLLER_TESTBENCH, TOP_TESTBENCH?	YES
NB: Failing to add any required info will cause point penalty (1-2 Marks)	

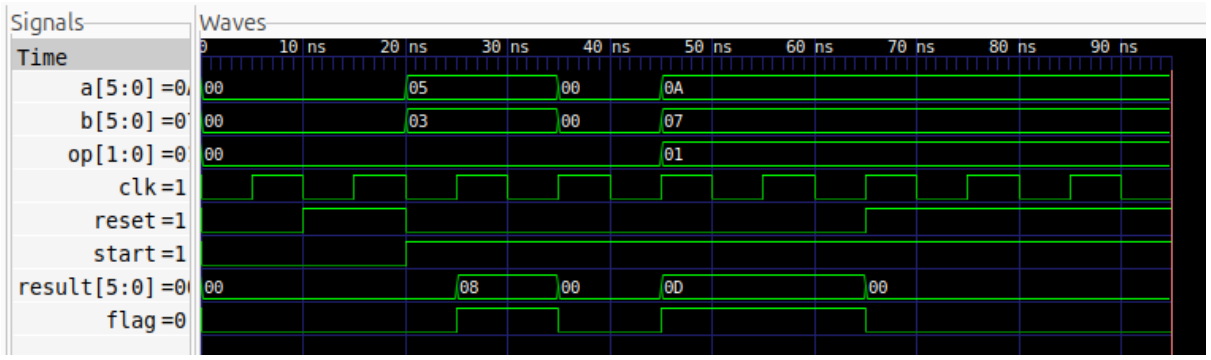
`alu_tb.v`



`controller_tb.v`



`top_tb.v`



RTL Synthesis (130nm Skywater PDK with OpenLane toolchain):

Check List: Have you added *RTL synthesis summary*, *RTL synthesized design figure* and *Standard cell usage in synthesized design*?	YES

NB: Failing to add any required info will cause point penalty (1-2 Marks)

RTL synthesis summary **(Following table is showing what info need to be shown /Just copy paste these info from terminal here)**

post_dff.rpt:

Metric	Count
Number of Wires	103
Number of wire bits	123
Number of public wires	10
Number of public wire bits	30
Number of ports	8
Number of port bits	24
Number of memories	0
Number of memory bits	0
Number of processes	0
Number of cells	106

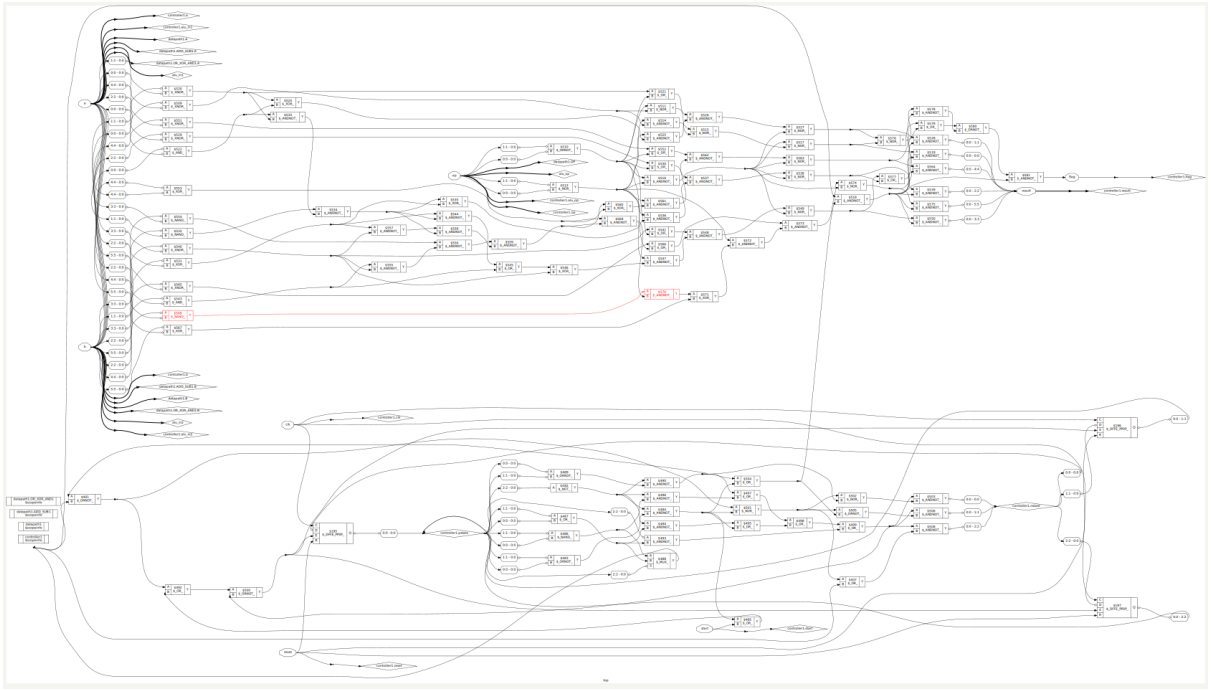
stat.rpt:

Metric	Count
Number of Wires	58
Number of wire bits	74
Number of public wires	11
Number of public wire bits	27
Number of ports	8
Number of port bits	24
Number of memories	0
Number of memory bits	0
Number of processes	0
Number of cells	57

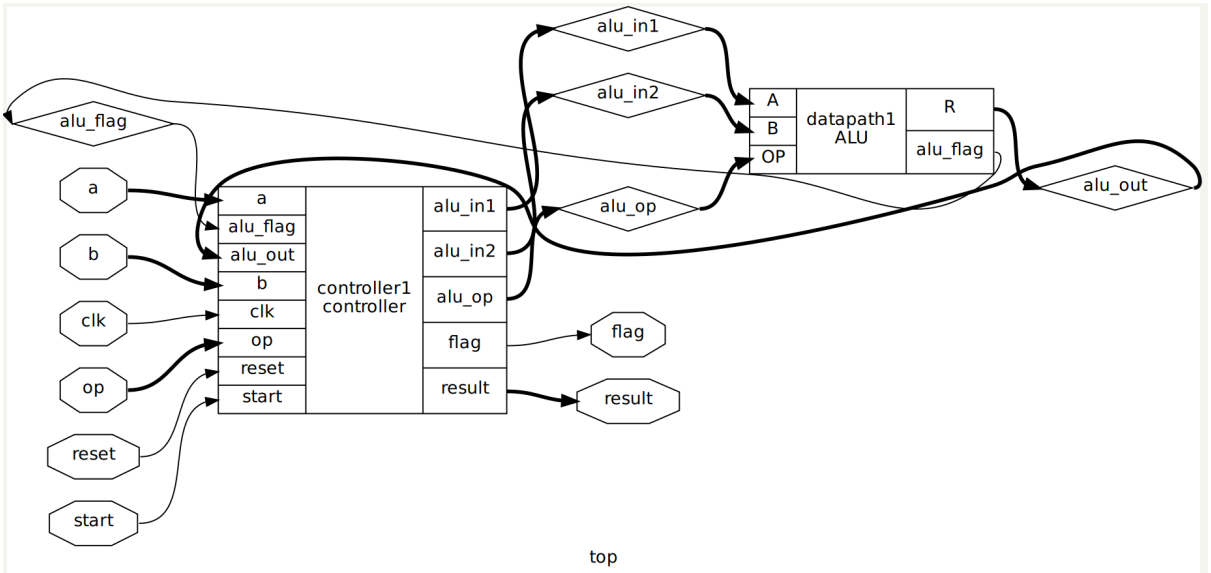
RTL synthesized design figure **(Following figure is showing what info need to be shown /Just copy paste these figures)**

There will be 2 figures in 2 .dot files in the synthesis folder. Show them both.

xdot primitive_techmap.dot :



xdot hierarchy.dot :



Standard cell usage in synthesized design **(Following table is showing what info need to be shown /Just copy paste these info from terminal here)**

Cell name	Count
sky130_fd_sc_hd__a211o_2	1
sky130_fd_sc_hd__a21o_2	1
sky130_fd_sc_hd__a21oi_2	2
sky130_fd_sc_hd__a22o_2	2
sky130_fd_sc_hd__a311o_2	1
sky130_fd_sc_hd__a31o_2	2
sky130_fd_sc_hd__a32o_2	1
sky130_fd_sc_hd__a41o_2	1
sky130_fd_sc_hd__and2_2	3
sky130_fd_sc_hd__and3_2	4
sky130_fd_sc_hd__and3b_2	2
sky130_fd_sc_hd__and4_2	1
sky130_fd_sc_hd__dfrtp_2	3
sky130_fd_sc_hd__inv_2	8
sky130_fd_sc_hd__mux2_1	1
sky130_fd_sc_hd__nand2_2	4
sky130_fd_sc_hd__nand4_2	1
sky130_fd_sc_hd__nor2_2	2
sky130_fd_sc_hd__o211a_2	1
sky130_fd_sc_hd__o21a_2	1
sky130_fd_sc_hd__o21ai_2	1
sky130_fd_sc_hd__o31ai_2	1
sky130_fd_sc_hd__or2_2	5
sky130_fd_sc_hd__or4_2	1
sky130_fd_sc_hd__or4bb_2	1
sky130_fd_sc_hd__xnor2_2	4
sky130_fd_sc_hd__xor2_2	2

RTL Floorplan (130nm Skywater PDK with OpenLane toolchain)

Check List: Have you added RTL Floorplan info?	YES
NB: Failing to add any required info will cause point penalty (1-2 Marks)	

(Following table is showing what info need to be shown /Just copy paste these info from terminal here)

General RTL Floorplan Information		
Parameter	**Value**	**Unit**
Database Units(DBU)	**?**	**--**
Site Size	**(2.72, 0.46)**	**µm**
Die Area (BBox)	**(0.0 0.0) to (48.14 58.86)**	**µm**
Core Area(BBox)	**(5.52 10.88) to (42.32 46.24)**	**µm**

Core Area	1301.25	µm2
Placement Utilization	?	%

Openroad_floorplan.log:

```
Reading library file at '/foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lib/sky130_fd_sc_hd__tt_025C_1v80.lib'…
Reading technology LEF file at '/foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/techlef/sky130_fd_sc_hd__nom.tlef'…
[INFO ODB-0227] LEF file: /foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/techlef/sky130_fd_sc_hd__nom.tlef, created 14 layers, 25 vias
Reading cell LEF file at '/foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lef/sky130_fd_sc_hd.lef'…
[WARNING ODB-0220] WARNING (LEFPARS-2008): NOWIREEXTENSIONATPIN statement is obsolete in version 5.6 or later.
The NOWIREEXTENSIONATPIN statement will be ignored. See file /foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lef/sky130_fd_sc_hd.lef at line 2.

[INFO ODB-0227] LEF file: /foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lef/sky130_fd_sc_hd.lef, created 437 library cells
Reading cell LEF file at '/foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lef/sky130_ef_sc_hd.lef'…
[WARNING ODB-0220] WARNING (LEFPARS-2008): NOWIREEXTENSIONATPIN statement is obsolete in version 5.6 or later.
The NOWIREEXTENSIONATPIN statement will be ignored. See file /foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lef/sky130_ef_sc_hd.lef at line 2.

[INFO ODB-0227] LEF file: /foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lef/sky130_ef_sc_hd.lef, created 4 library cells
Reading top-level netlist at '/foss/designs/assignment/runs/RUN_2025-06-13_04-36-13/06-yosys-synthesis/top.nl.v'…
Linking design 'top' from netlist…
Reading design constraints file at '/usr/local/lib/python3.12/dist-packages/openlane/scripts/base.sdc'…
[WARNING] No CLOCK_PORT found. A dummy clock will be used.
[WARNING STA-0366] port '__VIRTUAL_CLK__' not found.
[INFO] Using clock __VIRTUAL_CLK__…
[INFO] Setting output delay to: 2.0
[INFO] Setting input delay to: 2.0
[WARNING STA-0366] port '__VIRTUAL_CLK__' not found.
[INFO] Setting load to: 0.033442
[INFO] Setting clock uncertainty to: 0.25
[INFO] Setting clock transition to: 0.14999999999999999444888487687421729788184165954589843 75
[WARNING STA-0419] transition time can not be specified for virtual clocks.
[INFO] Setting timing derate to: 5%
[WARNING STA-0450] virtual clock __VIRTUAL_CLK__ can not be propagated.
Using site height: 2.72 and site width: 0.46…
[INFO] Using relative sizing for the floorplan.
[INFO IFP-0001] Added 13 rows of 80 site unithd.
[INFO IFP-0030] Inserted 0 tiecells using sky130_fd_sc_hd__conb_1/LO.
[INFO IFP-0030] Inserted 0 tiecells using sky130_fd_sc_hd__conb_1/HI.
[INFO] Extracting DIE_AREA and CORE_AREA from the floorplan
[INFO] Floorplanned on a die area of 0.0 0.0 48.14 58.86 (µm).
[INFO] Floorplanned on a core area of 5.52 10.88 42.32 46.24 (µm).
Writing metric design__die__bbox: 0.0 0.0 48.14 58.86
Writing metric design__core__bbox: 5.52 10.88 42.32 46.24
Setting global connections for newly added cells…
[INFO] Setting global connections...
Updating metrics…
Cell type report:                       Count       Area
  Inverter                                  8      30.03
  Sequential cell                           3      78.83
  Multi-Input combinational cell           46     441.67
  Total                                    57     550.53
Writing OpenROAD database to '/foss/designs/assignment/runs/RUN_2025-06-13_04-36-13/13-openroad-floorplan/top.odb'…
Writing netlist to '/foss/designs/assignment/runs/RUN_2025-06-13_04-36-13/13-openroad-floorplan/top.nl.v'…
Writing powered netlist to '/foss/designs/assignment/runs/RUN_2025-06-13_04-36-13/13-openroad-floorplan/top.pnl.v'…
Writing layout to '/foss/designs/assignment/runs/RUN_2025-06-13_04-36-13/13-openroad-floorplan/top.def'…
Writing timing constraints to '/foss/designs/assignment/runs/RUN_2025-06-13_04-36-13/13-openroad-floorplan/top.sdc'…
```

Or_metrics_out.json:

Open | or_metrics_out.json — /foss/designs/assignment/runs/RUN_2025-06-13_04-36-13/13-openroad-floorplan | Save


```
{
        "design__die__bbox": "0.0 0.0 48.14 58.86",
        "design__core__bbox": "5.52 10.88 42.32 46.24",
        "design__io": 24,
        "design__die__area": 2833.52,
        "design__core__area": 1301.25,
        "design__instance__count": 57,
        "design__instance__area": 550.528,
        "design__instance__count__stdcell": 57,
        "design__instance__area__stdcell": 550.528,
        "design__instance__count__macros": 0,
        "design__instance__area__macros": 0,
        "design__instance__utilization": 0.423077,
        "design__instance__utilization__stdcell": 0.423077,
        "design__instance__count__class:inverter": 8,
        "design__instance__count__class:sequential_cell": 3,
        "design__instance__count__class:multi_input_combinational_cell": 46,
        "flow__warnings__count": 6,
        "flow__errors__count": 0
}
```


RTL Power Analysis (130nm Skywater PDK with OpenLane toolchain)

Check List: Have you added RTL Power Analysis info?	YES
NB: Failing to add any required info will cause point penalty (1-2 Marks)	

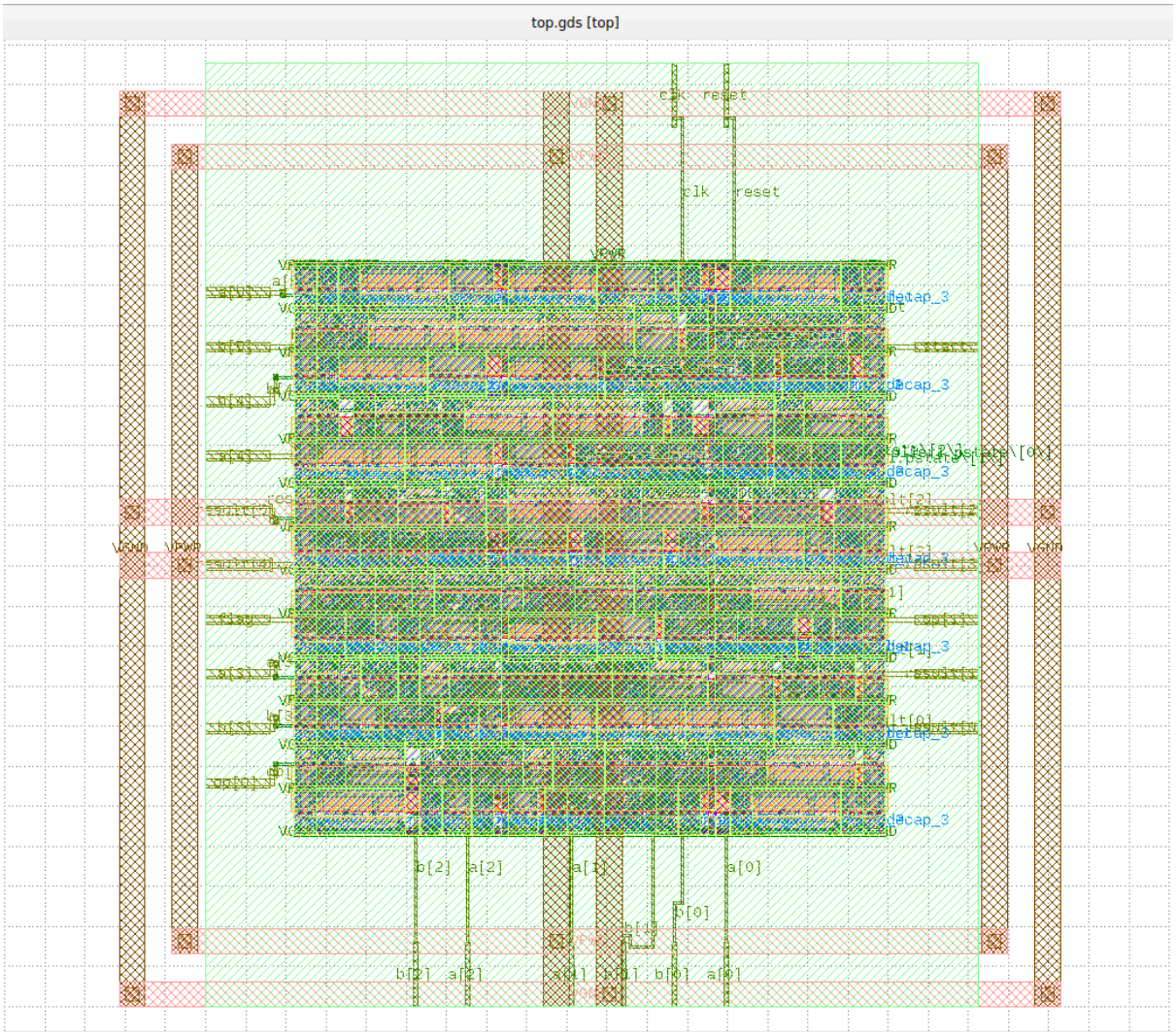
(Following table is showing what info need to be shown /Just copy paste these info from terminal here)

Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)	
Sequential	4.842016e-06	5.543969e-06	3.768916e-11	1.038602e-05	13.2%
Combinational	2.520306e-05	4.297765e-05	2.475429e-10	6.818095e-05	86.8%
Clock	0.000000e+00	0.000000e+00	2.041229e-10	2.041229e-10	0.0%
Macro	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0%
Pad	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0%
Total	3.004508e-05	4.852161e-05	4.893550e-10	7.856718e-05	100.0%
	38.2%	61.8%	0.0%		

GDS Layout (130nm Skywater PDK with OpenLane toolchain)

Check List: Have you added the GDS Layout figure?	YES
NB: Failing to add any required info will cause point penalty (1-2 Marks)	

(Following figure is showing what info need to be shown /Just copy paste these figures)



Heatmap (130nm Skywater PDK with OpenLane toolchain)

Check List: Have you added the heatmap?	YES
NB: Failing to add any required info will cause point penalty (1-2 Marks)	

(Following figure is showing what info need to be shown /Just copy paste these figures)