ASIC IMPLEMENTATION OF AN 8BIT RIPPLE CARRY ADDER USING 1BIT FULL ADDER

RTL to GDSII Flow Using Synopsys EDA Suite

Dhyey Hingarajiya

ABSTRACT

This project report documents the complete ASIC design flow of an **8-bit Ripple Carry Adder (RCA)** using individual 1-bit full adder modules, implemented from the Register-Transfer Level (RTL) through to GDSII. The objective is to gain hands-on experience with the digital design process, leveraging industry-standard EDA tools from Synopsys, including **VCS** for simulation, **Design Compiler (DC)** for synthesis, **IC Compiler II (ICC2)** for physical implementation, and **PrimeTime (PT)** for static timing analysis.

The Ripple Carry Adder is a fundamental arithmetic circuit where each bit addition depends on the carry generated from the previous stage. Although simple in structure, this dependency chain introduces propagation delay, which makes RCAs valuable for studying timing challenges in digital designs.

The design process begins with Verilog modelling and functional simulation to validate correctness. The verified design is synthesized in DC using standard-cell libraries under defined timing constraints. The synthesis results—such as area, cell count, timing slack, and schematic views—are analysed and documented. The synthesized netlist and constraints are then passed to ICC2 for floor planning, power planning, placement, clock tree synthesis (CTS), and routing. At each stage, timing (report_timing) and quality-of-results (report_qor) reports are generated to ensure design integrity.

Post-routing, the design is analysed in PrimeTime to verify that all timing constraints are met, with the final slack maintained between 0 and 1 ns. The report includes detailed screenshots, waveform diagrams, and logs. All design file changes, including path corrections, constraint adjustments, and naming updates, are recorded in a separate documentation section for full traceability and reproducibility.

1. INTRODUCTION

In this project, the complete ASIC design flow of an **8-bit Ripple Carry Adder (RCA)** was carried out, starting from the Verilog RTL description and ending with post-layout timing analysis. The goal was to gain practical exposure to how digital designs are taken through various stages—from initial coding to a physical layout that could theoretically be sent for fabrication. Industry-standard tools from Synopsys, including **VCS**, **Design Compiler**, **IC Compiler II**, and **PrimeTime**, were used throughout the process.

The RCA is a simple but fundamental circuit used to perform binary addition. It is built by chaining multiple 1-bit full adders, where the carry from each stage is passed on to the next. While not ideal in terms of speed due to the delay caused by carry propagation, it is still widely used in teaching and prototyping because of its easy-to-understand logic and structure. This made it a great candidate for learning the complete flow of digital design.

The design was first described in Verilog and verified using simulation in VCS. A testbench was written, and waveforms were generated to confirm the expected output. After functional correctness was ensured, the design was synthesized using Design Compiler, where constraints were applied, and reports like report_timing and report_qor were analysed. The physical design was then implemented in ICC2, including floor planning, power planning, placement, clock tree synthesis, and routing. Finally, the design was analysed in PrimeTime to ensure timing closure.

Screenshots, reports, and waveform diagrams have been included at each step, and a separate section has been added to document all changes made to file paths, constraints, and design settings. Through this project, a clear understanding of the full ASIC design pipeline was developed.

2. VERILOG RTL DESIGN

The design process began with the development of Verilog modules for the **1-bit Full Adder** and the **8-bit Ripple Carry Adder (RCA)**. These were written using structural modelling to reflect how the RCA is constructed by connecting multiple full adder blocks in series. The 1-bit full adder is responsible for computing the sum and carry-out of a single bit, and by chaining eight such adders, an 8-bit adder was built.

I. Full Adder RTL

A separate module was written for the 1-bit full adder. It takes three inputs—a, b, and cin (carry-in)—and produces two outputs—sum and cout (carry-out). The sum is calculated using the XOR of all three inputs, while the carry-out is generated using the majority function of the same three signals.

```
//File Name: full_adder_1bit.v

module full_adder_1bit (
    input a, b, cin,
    output sum, cout
);
    assign sum = a ^ b ^ cin;
    assign cout = (a & b) | (b & cin) | (a & cin);
endmodule
```

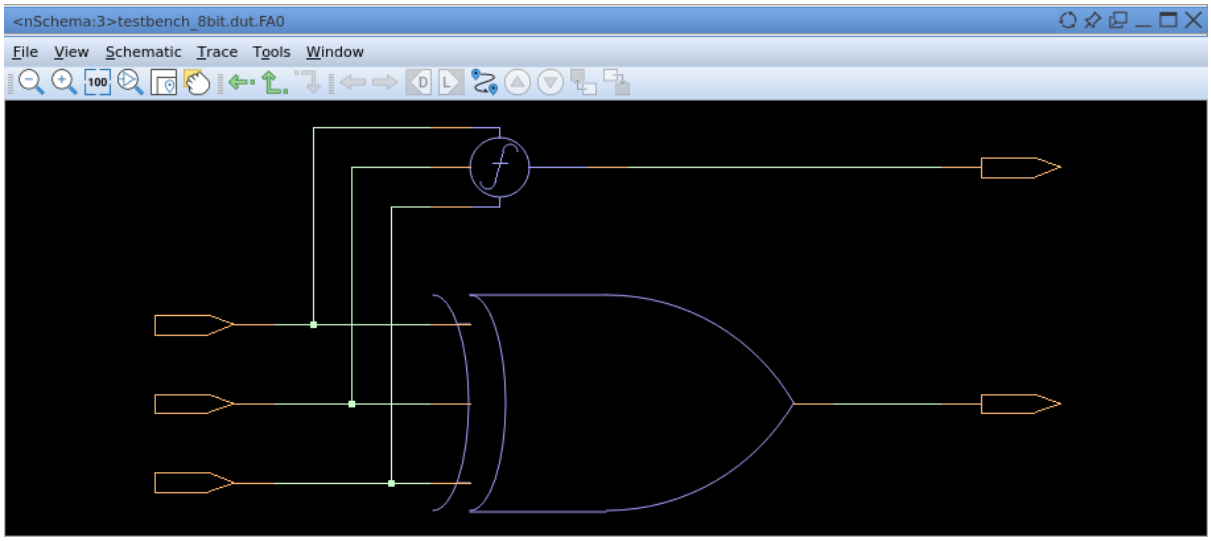

Figure 2.1: Internal Schematic of the 1-bit Full Adder Module

II. Ripple Carry Adder RTL

In the 8-bit RCA design, input operands (A, B) and the initial carry-in (C_in) are first latched using D flip-flops triggered by the rising edge of the clock. These latched signals are then passed to a chain of eight full adder modules. Each full adder computes the sum and carry for one bit, with the carry-out feeding into the next stage's carry-in.

The intermediate sum values and the final carry-out are then registered again on the rising edge of the clock to produce the outputs SUM and C_out. This sequential behaviour ensures that the adder's operation is stable and synchronized with a clock signal, which helps during timing analysis and synthesis.

```
//File Name: full_adder_8bit.v

`include "full_adder_1bit.v"

module full_adder_8bit (
    input [7:0] A,
    input [7:0] B,
    input       C_in,
    input       Clock,
    output reg [7:0] SUM,
    output reg       C_out
);

    // Internal signals
    reg [7:0] regA, regB;
    reg       regCin;
    wire [7:0] sum_wire;
    wire [7:0] carry;

    // Input latching
    always @(posedge Clock) begin
        regA   <= A;
        regB   <= B;
        regCin <= C_in;
    end

    // Output latching
    always @(posedge Clock) begin
        SUM   <= sum_wire;
        C_out <= carry[7];
    end

    // Full adder chaining (ripple carry)
    full_adder_1bit FA0 (regA[0], regB[0], regCin,      sum_wire[0],
carry[0]);
    full_adder_1bit FA1 (regA[1], regB[1], carry[0],    sum_wire[1],
carry[1]);
```

```
    full_adder_1bit FA2 (regA[2], regB[2], carry[1],    sum_wire[2],
carry[2]);
    full_adder_1bit FA3 (regA[3], regB[3], carry[2],    sum_wire[3],
carry[3]);
    full_adder_1bit FA4 (regA[4], regB[4], carry[3],    sum_wire[4],
carry[4]);
    full_adder_1bit FA5 (regA[5], regB[5], carry[4],    sum_wire[5],
carry[5]);
    full_adder_1bit FA6 (regA[6], regB[6], carry[5],    sum_wire[6],
carry[6]);
    full_adder_1bit FA7 (regA[7], regB[7], carry[6],    sum_wire[7],
carry[7]);

endmodule
```

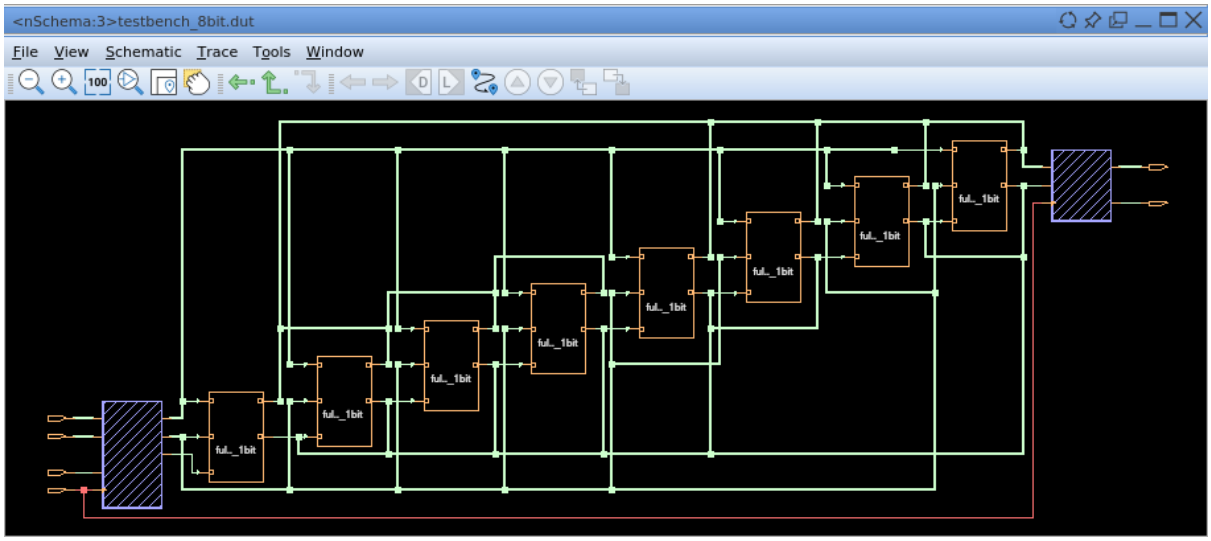

Figure 2.2: Internal Schematic of the 8-bit Ripple Carry Adder Showing Chained 1-bit Full Adder Instances

3. TESTBENCH AND FUNCTIONAL VERIFICATION

To verify the functional correctness of the 8-bit Ripple Carry Adder (RCA), a clocked testbench was developed using Verilog. The testbench was designed to provide stimulus to the DUT (Design Under Test) and observe its outputs across clock cycles. Since the RCA is a sequential design, inputs were applied on the **negative edge** and latched by the DUT on the **rising edge of the clock**, with outputs expected to stabilize by the following clock cycle.

The testbench included multiple test cases to check for various operand combinations and carry-in values. These cases were chosen to observe both regular additions and overflow conditions. The waveform was used to validate the design.

I. Testbench Code

```
//File Name: testbench_8bit.v

`timescale 1ns/1ns
`include "full_adder_8bit.v"

module testbench_8bit;

    reg [7:0] A, B;
    reg       C_in, Clock;
    wire [7:0] SUM;
    wire       C_out;

    // Instantiate DUT
    full_adder_8bit dut (.A(A), .B(B), .C_in(C_in), .Clock(Clock),
.SUM(SUM), .C_out(C_out));

    // Clock generation
    always #5 Clock = ~Clock;

    // Stimulus
    initial begin
        $fsdbDumpvars;

        // Reset the Inputs
        Clock = 0;
        A = 0;
        B = 0;
        C_in = 0;

        // Wait for stable startup
        @(posedge Clock);

         // TEST CASE 1
        @(negedge Clock);
        A = 8'b00000001; B = 8'b00000001; C_in = 0;
        @(posedge Clock); // inputs latched
```

```
        @(posedge Clock); // wait one more cycle for outputs
        #1; // small delay to stabilize
        $display("TC1: A=%b B=%b C_in=%b => SUM=%b C_out=%b", A, B, C_in,
SUM, C_out);

        // TEST CASE 2
        @(negedge Clock);
        A = 8'b00000110; B = 8'b00001010; C_in = 1;
        @(posedge Clock);
        @(posedge Clock);
        #1;
        $display("TC2: A=%b B=%b C_in=%b => SUM=%b C_out=%b", A, B, C_in,
SUM, C_out);

        // TEST CASE 3
        @(negedge Clock);
        A = 8'b00001000; B = 8'b00001111; C_in = 1;
        @(posedge Clock);
        @(posedge Clock);
        #1;
        $display("TC3: A=%b B=%b C_in=%b => SUM=%b C_out=%b", A, B, C_in,
SUM, C_out);

        #50 $finish;
    end

endmodule
```

II. Simulation and Output Waveform

The simulation was carried out using VCS. The following command was used for compilation and execution:

```
vcs -full64 full_adder_1bit.v -debug_all+all -lca -kdb
vcs -full64 full_adder_8bit.v -debug_all+all -lca -kdb
vcs -full64 testbench_8bit.v -debug_all+all -lca -kdb

./simv Verdi
verdi -ssf novas.fsdb -nologo
```

III. Results and Observations

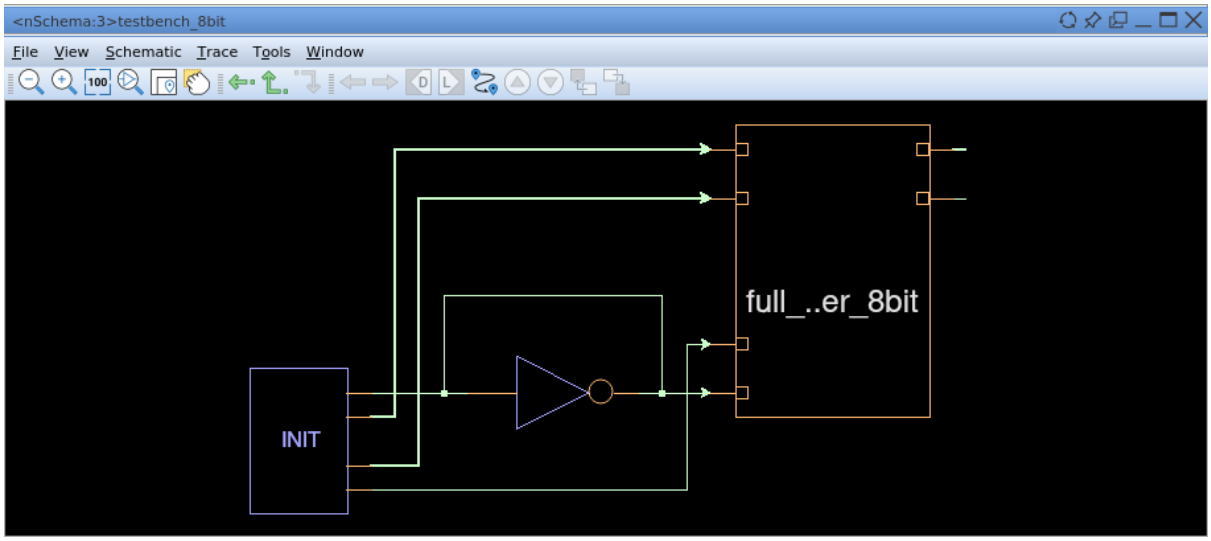

Figure 3.1 Schematic Representation of Testbench Driving the 8-bit Ripple Carry Adder

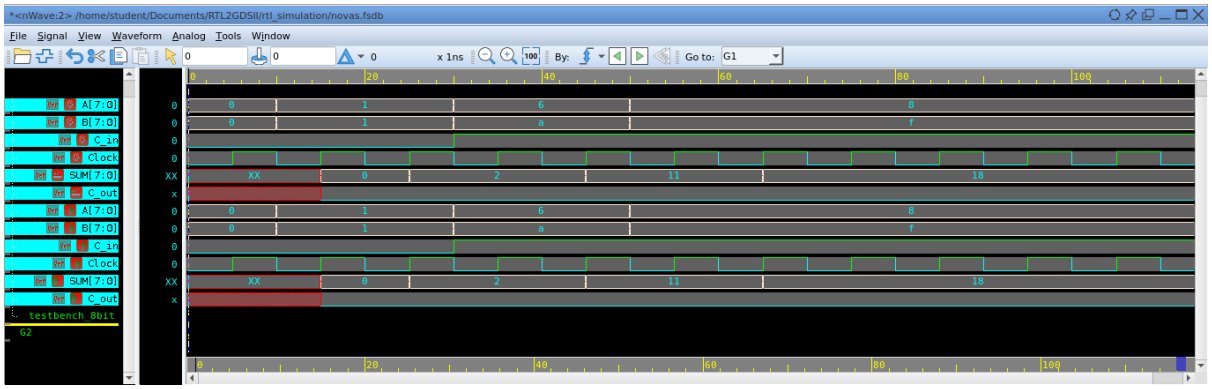

Figure 3.2: Waveform Showing Output Response of 8-bit Ripple Carry Adder for Sample Test Cases

- Inputs were properly latched on the rising edge of the clock after being applied on the negative edge.
- Outputs (SUM, C_out) were updated in the following cycle, as expected in a clocked design.
- All test cases produced correct results. The output matched the expected binary sum of the operands along with correct carry-out.
- The waveform clearly showed the pipeline-like behaviour with a one-cycle latency between input and output.

4. CONSTRAINTS FILE

The Synopsys Design Constraints (SDC) file specifies the timing and design requirements for the synthesis and implementation of the 8-bit Ripple Carry Adder design. The constraints guide the synthesis and place-and-route tools to meet timing goals and optimize the design accordingly.

The following constraints were applied:

Clock Definition: The clock 'Clock' was defined with a 2 ns period, corresponding to a 500 MHz clock frequency for the design.

Input and Output Timing: Maximum input delay and output delay relative to the clock were set to 0.5 ns to reflect realistic signal arrival and required output timing margins.

Transition Times: Input signal transition times were limited to 0.5 ns, while internal and clock path transitions were constrained for better timing performance.

Clock Uncertainty: Setup and hold clock uncertainties were set to 0.3 ns and 0.1 ns respectively to account for clock jitter and variation.

Max Transition and Fanout: The maximum allowed transition time for signals and the maximum fanout for logic gates were restricted to control signal integrity and loading.

```
//File Name: full_adder.sdc

create_clock -period 2 [get_ports Clock]

set_input_delay -max 0.5 -clock Clock [all_inputs]
set_input_transition 0.5 [all_inputs]

set_output_delay -max 0.5 -clock Clock [all_outputs]

set_clock_uncertainty -setup 0.300 [get_clocks Clock]
set_clock_uncertainty -hold 0.100 [get_clocks Clock]

set_max_transition 0.50 [current_design]
set_max_transition -clock_path 0.250 [get_clocks Clock]

set_max_fanout 5 [current_design]
```

This constraints setup ensures that the design tools consider realistic signal and clock behaviour, thus improving the reliability and timing closure of the synthesized design.

5. SYNTHESIS USING DESIGN COMPILER (DC)

I. Synthesis Script

```
#File Name: run_dc.tcl

source -echo -verbose ./rm_setup/dc_setup.tcl
set RTL_SOURCE_FILES ./../rtl/full_adder_1bit.v
set RTL_SOURCE_FILES ./../rtl/full_adder_8bit.v

define_design_lib WORK -path ./WORK

set_dont_use [get_lib_cells */FADD*]
set_dont_use [get_lib_cells */HADD*]
set_dont_use [get_lib_cells */AO*]
set_dont_use [get_lib_cells */OA*]
#set_dont_use [get_lib_cells */NAND*]
set_dont_use [get_lib_cells */XOR*]
#set_dont_use [get_lib_cells */NOR*]
set_dont_use [get_lib_cells */XNOR*]
set_dont_use [get_lib_cells */MUX*]

analyze -format verilog ${RTL_SOURCE_FILES}
elaborate ${DESIGN_NAME}
current_design


read_sdc ./../CONSTRAINTS/full_adder.sdc

#compile

compile_ultra
#report_timing
write -format verilog -hierarchy -output
${RESULTS_DIR}/${DCRM_FINAL_VERILOG_OUTPUT_FILE}
```

Design Library: A working design library called WORK is defined to hold the design database.

Cell Usage Restrictions: Certain complex cells like fast adders (FADD*, HADD*), specific logic gates (AO*, OA*), XORs, XNORs, and multiplexers are restricted with set_dont_use commands. This guides the compiler to synthesize the design using simpler, more controlled cells.

Analyze & Elaborate: The RTL code is analysed to check syntax and semantics, then elaborated to build an internal representation of the design.

Constraints: The SDC file full_adder.sdc is read to apply timing constraints such as clock definitions and input/output delays.

Compilation: The design is compiled with compile_ultra, which performs aggressive optimization for timing and area.

Output: The synthesized gate-level netlist is saved as a Verilog file named full_adder_8bit.mapped.v with hierarchy preserved for further design steps.

II. Results

Figure 5.1: Block Schematic

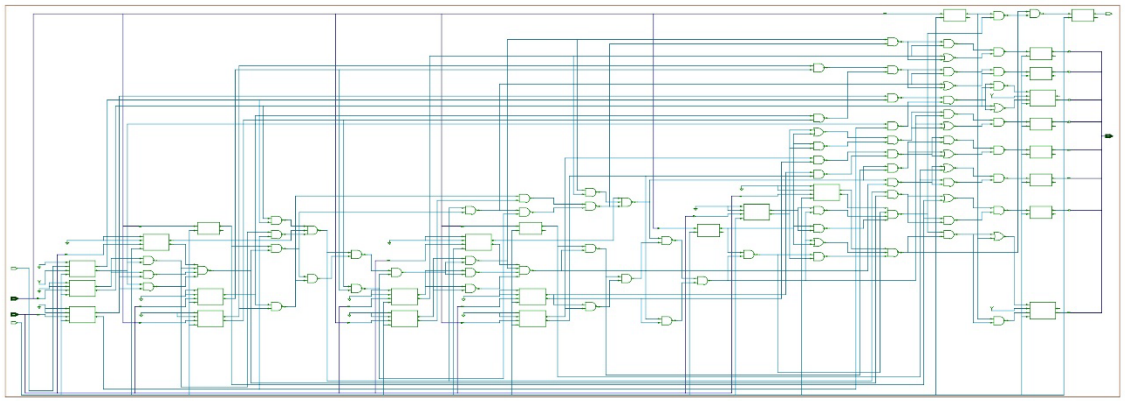

Figure 5.2 Gate Level Netlist Schematic

QOR Report

```
****************************************
Report : qor
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 10:25:59 2025
****************************************


  Timing Path Group 'Clock'
  -----------------------------------
  Levels of Logic:              15.00
  Critical Path Length:          1.55
  Critical Path Slack:           0.10
  Critical Path Clk Period:      2.00
  Total Negative Slack:          0.00
  No. of Violating Paths:        0.00
  Worst Hold Violation:          0.00
  Total Hold Violation:          0.00
  No. of Hold Violations:        0.00
  -----------------------------------


  Cell Count
  -----------------------------------
  Hierarchical Cell Count:          0
  Hierarchical Port Count:          0
  Leaf Cell Count:                103
  Buf/Inv Cell Count:               0
  Buf Cell Count:                   0
  Inv Cell Count:                   0
  CT Buf/Inv Cell Count:            0
  Combinational Cell Count:        77
  Sequential Cell Count:           26
  Macro Count:                      0
  -----------------------------------


  Area
  -----------------------------------
  Combinational Area:      129.359296
  Noncombinational Area:   179.425668
  Buf/Inv Area:              0.000000
  Total Buffer Area:             0.00
  Total Inverter Area:           0.00
  Macro/Black Box Area:      0.000000
  Net Area:                 47.358729
  -----------------------------------
  Cell Area:               308.784964
  Design Area:             356.143694
```

```
  Design Rules
  ----------------------------------
  Total Number of Nets:           150
  Nets With Violations:             0
  Max Trans Violations:             0
  Max Cap Violations:               0
  ----------------------------------


  Hostname: ict-chipin.sot.pdpu.ac.in

  Compile CPU Statistics
  ----------------------------------------
  Resource Sharing:                   0.00
  Logic Optimization:                 0.13
  Mapping Optimization:               0.20
  ----------------------------------------
  Overall Compile Time:               2.66
  Overall Compile Wall Clock Time:    2.91

  --------------------------------------------------------------------

  Design  WNS: 0.00  TNS: 0.00  Number of Violating Paths: 0


  Design (Hold)  WNS: 0.00  TNS: 0.00  Number of Violating Paths: 0

  --------------------------------------------------------------------
```

Timing Report

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 10:26:31 2025
****************************************

Operating Conditions: ff1p16v125c   Library: saed32rvt_ff1p16v125c
Wire Load Model Mode: enclosed

  Startpoint: regCin_reg (rising edge-triggered flip-flop clocked by Clock)
  Endpoint: SUM_reg[7] (rising edge-triggered flip-flop clocked by Clock)
  Path Group: Clock
  Path Type: max
```

```
Des/Clust/Port     Wire Load Model       Library
------------------------------------------------
full_adder_8bit    8000                  saed32rvt_ff1p16v125c

Point                                    Incr       Path
-----------------------------------------------------------
clock Clock (rise edge)                  0.00       0.00
clock network delay (ideal)              0.00       0.00
regCin_reg/CLK (DFFSSRX1_RVT)            0.00       0.00 r
regCin_reg/QN (DFFSSRX1_RVT)             0.04       0.04 r
U85/Y (NAND2X0_RVT)                      0.13       0.18 f
U86/Y (NAND2X0_RVT)                      0.07       0.25 r
U87/Y (NAND3X0_RVT)                      0.04       0.29 f
U94/Y (NAND2X0_RVT)                      0.15       0.43 r
U96/Y (AND2X1_RVT)                       0.03       0.46 r
U97/Y (NAND2X0_RVT)                      0.07       0.53 f
U98/Y (NAND2X0_RVT)                      0.15       0.67 r
U103/Y (NAND3X0_RVT)                     0.04       0.71 f
U110/Y (NAND2X0_RVT)                     0.14       0.86 r
U112/Y (AND2X1_RVT)                      0.04       0.90 r
U113/Y (NAND2X0_RVT)                     0.40       1.29 f
U118/Y (NAND2X0_RVT)                     0.07       1.36 r
U119/Y (NAND3X0_RVT)                     0.03       1.39 f
U120/Y (NAND2X0_RVT)                     0.07       1.46 r
U67/Y (OR2X1_RVT)                        0.08       1.54 r
SUM_reg[7]/D (DFFSSRX1_RVT)              0.01       1.55 r
data arrival time                                   1.55

clock Clock (rise edge)                  2.00       2.00
clock network delay (ideal)              0.00       2.00
clock uncertainty                       -0.30       1.70
SUM_reg[7]/CLK (DFFSSRX1_RVT)            0.00       1.70 r
library setup time                      -0.05       1.65
data required time                                  1.65
-----------------------------------------------------------
data required time                                  1.65
data arrival time                                  -1.55
-----------------------------------------------------------
slack (MET)                                         0.10
```

6. PHYSICAL DESIGN – FLOORPLANNING USING ICCII

I. Floor planning Script

After synthesis, the design was imported into Synopsys IC Compiler II (ICC2) for the physical design process. The first step in this flow was floor planning, which defines the physical layout boundaries, IO pin placement, and initial placement of standard cells. The following script was used to perform the floor planning:

```
set PDK_PATH ./../ref

create_lib -ref_lib $PDK_PATH/lib/ndm/saed32rvt_c.ndm FULL_ADDER_8BIT_LIB

read_verilog {./../DC/results/full_adder_8bit.mapped.v} -library
FULL_ADDER_8BIT_LIB -design full_adder_8bit_v -top full_adder_8bit

#open the lib and block after saving
#open_lib FULL_ADDER_8BIT_LIB
#open_block FULL_ADD_8Bit

#--------------------------------------------------------------------#

# FloorPlan settings

# Set floorplan
initialize_floorplan -side_length {20 20} -core_offset 2 -
coincident_boundary true
set ports [get_ports]

# Separate ports based on function
set input_portsa [get_ports *A*]
set input_portsb [get_ports *B*]
set output_ports [get_ports *SUM*]
set Cout [get_ports *out*]
set Cin [get_ports *in*]
set clk_ports [get_ports *Clock*]

# Assign constraints to sides:
# Side 1: inputs
# Side 2: outputs
# Side 3: clock
set_individual_pin_constraints -ports $input_portsa  -sides 1
set_individual_pin_constraints -ports $input_portsb  -sides 1
set_individual_pin_constraints -ports $Cin  -sides 1
set_individual_pin_constraints -ports $Cout  -sides 2
set_individual_pin_constraints -ports $output_ports -sides 2
set_individual_pin_constraints -ports $clk_ports    -sides 3
```

```
# Place pins respecting constraints
place_pins -self

# Place standard cells
create_placement -floorplan -effort high
save_block
save_lib

save_block -as FULL_ADD_8BIT_Floorplan
save_lib

#####################End of Floorplan Scenarios#######################
```

The design was read into ICC2 using the post-synthesis mapped Verilog file.

A new design library named FULL_ADDER_8BIT_LIB was created using the 32nm SAED PDK.

The core and die area were defined using initialize_floorplan, with a side length of 20 units and a 2-unit core offset on all sides.

Ports were grouped and placed on specific sides: **Side 1** for all input ports (A, B, Cin), **Side 2** for output ports (SUM and Cout), **Side 3** for the Clock signal.

Pins were auto-placed using the defined constraints, and standard cells were placed using create_placement with high effort for better QoR.

The floorplan was saved, and a named checkpoint FULL_ADD_8BIT_Floorplan was created.

II. Results

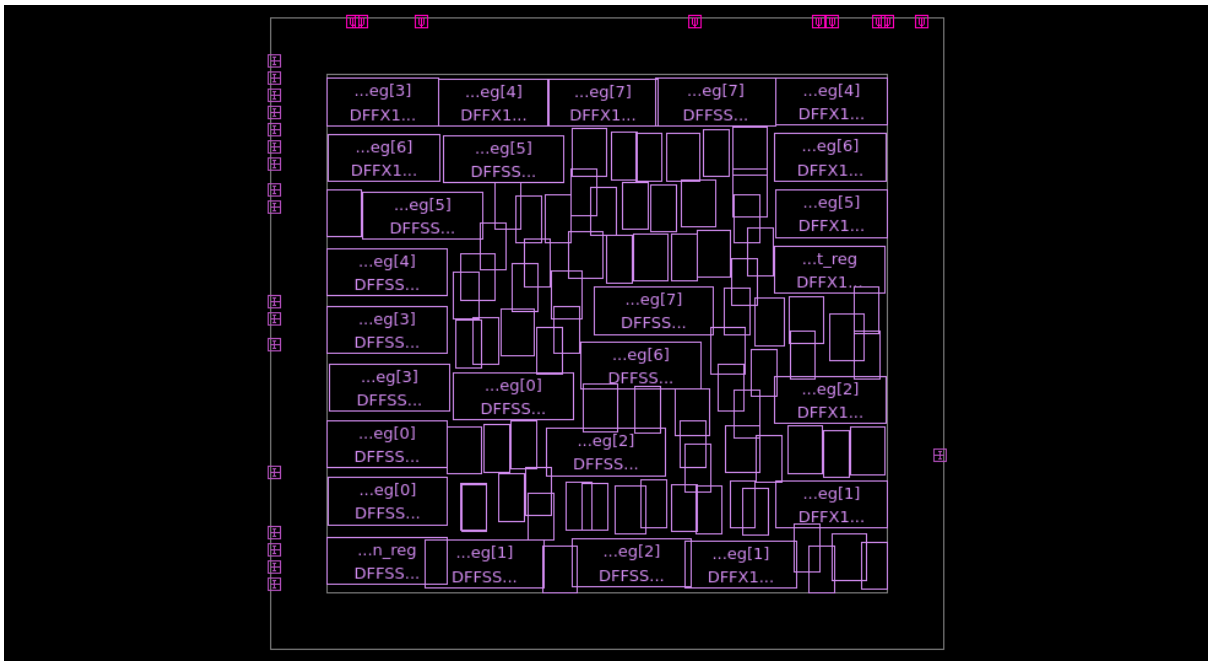

Figure 6.1 Initial Floor planning: Core Area and IO Pin Placement defined

QOR Report

```
****************************************
Report : qor
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:05:29 2025
****************************************
Warning: no valid parasitic for default corner(LATE) (NEX-018)
Warning: no valid parasitic for default corner(EARLY) (NEX-018)
Warning: no valid parasitic for (all corners) corner((both specs)) (NEX-018)
Information: The stitching and editing of coupling caps is turned OFF for design
'FULL_ADDER_8BIT_LIB:full_adder_8bit_v.design'. (TIM-125)
Warning: The extractor can not be initialized for block 'full_adder_8bit_v'. (TIM-103)
Warning: The extractor can not be initialized for design 'full_adder_8bit'. Zero
interconnect delay is used in timing analysis. (TIM-102)
Warning: Batch extraction is skipped for block "full_adder_8bit"
****************************************************
Timer Settings:
Delay Calculation Style:                  auto
Signal Integrity Analysis:                disabled
Timing Window Analysis:                   disabled
Advanced Waveform Propagation:            disabled
Variation Type:                           fixed_derate
Clock Reconvergence Pessimism Removal:    disabled
Advanced Receiver Model:                  disabled
LLE:                                      disabled
ML Acceleration:                          off
Smart Arc Optimization:                   disabled
****************************************************


Cell Count
---------------------------------------
Hierarchical Cell Count:              0
Hierarchical Port Count:              0
Leaf Cell Count:                    103
Buf/Inv Cell Count:                   0
Buf Cell Count:                       0
Inv Cell Count:                       0
Combinational Cell Count:            77
   Single-bit Isolation Cell Count:                     0
   Multi-bit Isolation Cell Count:                      0
   Isolation Cell Banking Ratio:                        0.00%
   Single-bit Level Shifter Cell Count:                 0
   Multi-bit Level Shifter Cell Count:                  0
   Level Shifter Cell Banking Ratio:                    0.00%
   Single-bit ELS Cell Count:                           0
   Multi-bit ELS Cell Count:                            0
   ELS Cell Banking Ratio:                              0.00%
Sequential Cell Count:               26
```

```
   Integrated Clock-Gating Cell Count:                    0
   Sequential Macro Cell Count:                           0
   Single-bit Sequential Cell Count:                      26
   Multi-bit Sequential Cell Count:                       0
   Sequential Cell Banking Ratio:                         0.00%
   BitsPerflop:                                           1.00
Macro Count:                          0
----------------------------------------


Area
----------------------------------------
Combinational Area:             129.36
Noncombinational Area:          179.43
Buf/Inv Area:                     0.00
Total Buffer Area:                0.00
Total Inverter Area:              0.00
Macro/Black Box Area:             0.00
Net Area:                            0
Net XLength:                    486.05
Net YLength:                    407.51
----------------------------------------
Cell Area (netlist):                           308.78
Cell Area (netlist and physical only):         308.79
Net Length:                     893.55


Design Rules
----------------------------------------
Total Number of Nets:             138
Nets with Violations:               0
Max Trans Violations:               0
Max Cap Violations:                 0
----------------------------------------
```

Timing Report

```
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -report_by design
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:05:36 2025
****************************************
No paths.
```

7. PHYSICAL DESIGN – POWER PLANNING USING ICCII

I. Power Planning Script

After floor planning, power planning was carried out in Synopsys ICC2 to ensure that power and ground (VDD and VSS) are distributed uniformly across the chip. A Power Delivery Network (PDN) was built in multiple stages using rings, mesh, and standard cell rails. The script below was used to generate the full PDN for the FULL_ADDER_8BIT block:

```
#########################################################################
## Power-planning - To build Power Delivery Network (PDN) ##
#########################################################################
#
#########################################################################
Step 1: to_create power/ground nets and to_connect power/ground nets :-
#########################################################################

#to create power nets
create_net -power {VDD}
create_net -ground {VSS}

## to connect power/ground_nets
create_io_pin -net VDD -layer M1 -location {0 0}
create_io_pin -net VSS -layer M1 -location {0 5}

connect_pg_net -all_blocks -automatic

#########################################################################
### step 2: to create power and ground ring patterns
#########################################################################

create_pg_ring_pattern core_ring_pattern -horizontal_layer M7 -
horizontal_width .4 -horizontal_spacing .3 -vertical_layer M8 -
vertical_width .4 -vertical_spacing .3

set_pg_strategy core_power_ring -core -pattern {{name :
core_ring_pattern}{nets : {VDD VSS}}{offset : {.5 .5}}}

compile_pg -strategies core_power_ring

#########################################################################
### step 3: to create pg mesh pattern
#########################################################################

create_pg_mesh_pattern mesh -layers { {{vertical_layer: M6}{width: .34}
{spacing: interleaving}{pitch: 5} {offset: .5}} {{horizontal_layer:
M7}{width: .38} {spacing: interleaving} {pitch: 5} {offset: .5}}
{{vertical_layer: M8}{width: .38} {spacing: interleaving}{pitch: 5}
{offset: .5}}}
```

```
set_pg_strategy core_mesh -pattern { {pattern:mesh} {nets: VDD VSS}} -
core -extension {stop: innermost_ring}

compile_pg -strategies core_mesh

#########################################################################
###step 4 : to create std cell power rail pattern
#########################################################################

create_pg_std_cell_conn_pattern std_cell_rail -layers {M1} -rail_width
0.06

set_pg_strategy rail_strat -core -pattern {{name: std_cell_rail} {nets:
VDD VSS} }

compile_pg -strategies rail_strat

connect_pg_net -all_blocks -automatic
#compile_pg

#########################################################################
### step 5 : To save block and Save lib
#########################################################################

save_block
save_lib

save_block -as FULL_ADD_8BIT_PowerPlan
save_lib
```

Power (VDD) and ground (VSS) nets were created and connected using automatic connection techniques.

Core rings were added around the die using metals M7 and M8 to enclose the design with robust power rails.

A **power mesh** was created using interleaved vertical and horizontal lines on M6, M7, and M8 for uniform distribution inside the core area.

The **standard cell power rails** were connected using narrow horizontal stripes on M1, allowing each cell in the design to receive power.

At each stage, strategies were compiled and applied to ensure connectivity and clean PDN generation.

The final power plan was saved under the name FULL_ADD_8BIT_PowerPlan.

II. Results

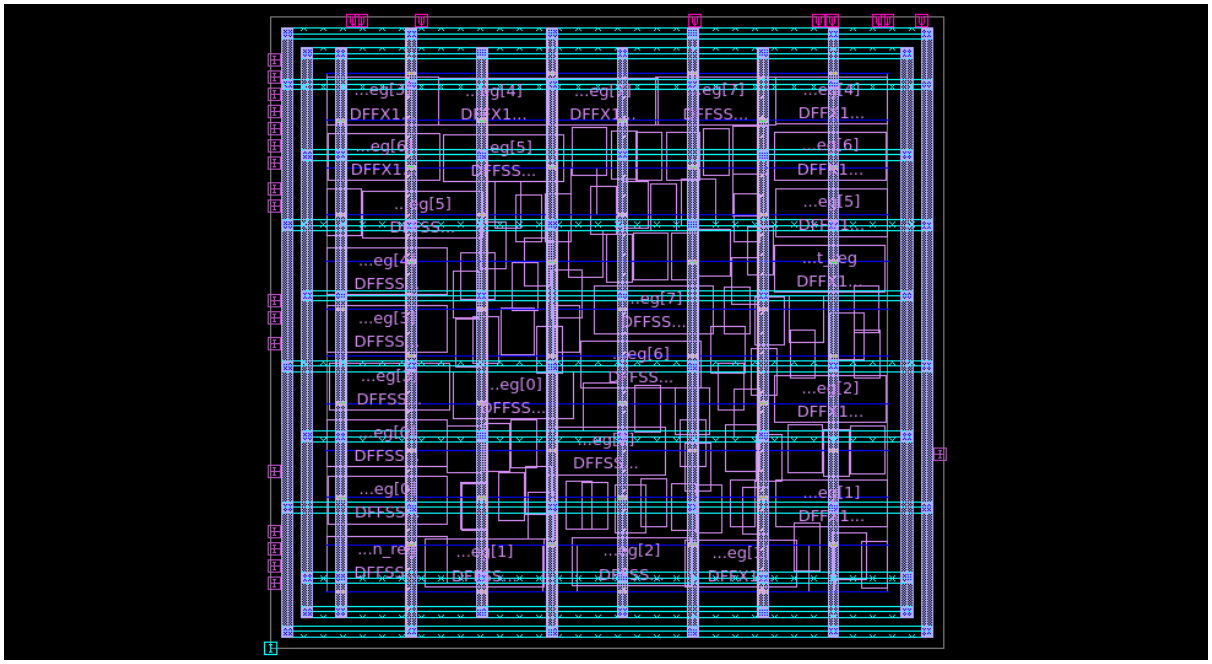

Figure 7.1: Power Planning : Power Pins and Ring, Mesh and Rail Patterns defined

QOR Report

```
****************************************
Report : qor
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:09:48 2025
****************************************

Cell Count
----------------------------------------
Hierarchical Cell Count:              0
Hierarchical Port Count:              0
Leaf Cell Count:                    103
Buf/Inv Cell Count:                   0
Buf Cell Count:                       0
Inv Cell Count:                       0
Combinational Cell Count:            77
   Single-bit Isolation Cell Count:                    0
   Multi-bit Isolation Cell Count:                     0
   Isolation Cell Banking Ratio:                       0.00%
   Single-bit Level Shifter Cell Count:                0
   Multi-bit Level Shifter Cell Count:                 0
   Level Shifter Cell Banking Ratio:                   0.00%
   Single-bit ELS Cell Count:                          0
   Multi-bit ELS Cell Count:                           0
   ELS Cell Banking Ratio:                             0.00%
Sequential Cell Count:               26
   Integrated Clock-Gating Cell Count:                 0
   Sequential Macro Cell Count:                        0
```

```
   Single-bit Sequential Cell Count:                      26
   Multi-bit Sequential Cell Count:                       0
   Sequential Cell Banking Ratio:                         0.00%
   BitsPerflop:                                           1.00
Macro Count:                          0
----------------------------------------


Area
----------------------------------------
Combinational Area:             129.36
Noncombinational Area:          179.43
Buf/Inv Area:                     0.00
Total Buffer Area:                0.00
Total Inverter Area:              0.00
Macro/Black Box Area:             0.00
Net Area:                            0
Net XLength:                    455.42
Net YLength:                    366.89
----------------------------------------
Cell Area (netlist):                           308.78
Cell Area (netlist and physical only):         308.79
Net Length:                     822.31


Design Rules
----------------------------------------
Total Number of Nets:               138
Nets with Violations:                 0
Max Trans Violations:                 0
Max Cap Violations:                   0
----------------------------------------
```

Timing Report

```
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -report_by design
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:10:39 2025
****************************************
No paths.
```

8. PHYSICAL DESIGN – CELL PLACEMENT USING ICCII

I. Cell Placement Script

Placement is the process where all logic cells (standard cells) are physically positioned within the core area, as defined during floor planning. The goal is to minimize area, wirelength, congestion, and satisfy timing constraints before Clock Tree Synthesis (CTS). Before placement, it's important to define the design *mode*, *corner*, and *scenario*, and read parasitic information for accurate estimation.

```
####mode for placement

set mode1 "func"
set corner1 "nom"
set scenario1 "${mode1}::${corner1}"
remove_modes -all; remove_corners -all; remove_scenarios -all

create_mode $mode1
create_corner $corner1
create_scenario -name func::nom -mode func -corner nom
current_mode func
current_scenario func::nom

source ./../CONSTRAINTS/full_adder.sdc

set_dont_use [get_lib_cells */FADD*]
set_dont_use [get_lib_cells */HADD*]
set_dont_use [get_lib_cells */AO*]
set_dont_use [get_lib_cells */OA*]
set_dont_use [get_lib_cells */NAND*]
set_dont_use [get_lib_cells */XOR*]
set_dont_use [get_lib_cells */NOR*]
set_dont_use [get_lib_cells */XNOR*]
set_dont_use [get_lib_cells */MUX*]

current_corner nom
current_scenario func::nom

set parasitic1 "p1"
set tluplus_file$parasitic1
"$PDK_PATH/tech/star_rcxt/saed32nm_1p9m_Cmax.tluplus"
set layer_map_file$parasitic1
"$PDK_PATH/tech/star_rcxt/saed32nm_tf_itf_tluplus.map"

set parasitic2 "p2"
set tluplus_file$parasitic2
"$PDK_PATH/tech/star_rcxt/saed32nm_1p9m_Cmin.tluplus"
set layer_map_file$parasitic2
"$PDK_PATH/tech/star_rcxt/saed32nm_tf_itf_tluplus.map"
```

```
read_parasitic_tech -tlup $tluplus_filep1 -layermap $layer_map_filep1 -
name p1
read_parasitic_tech -tlup $tluplus_filep2 -layermap $layer_map_filep2 -
name p2

set_parasitic_parameters -late_spec $parasitic1 -early_spec $parasitic2
set_app_options -name place.coarse.continue_on_missing_scandef -value
true

place_pins -self
place_opt
legalize_placement

save_block
save_lib

save_block -as FULL_ADD_8BIT_Placement
save_lib
```

Mode / Corner / Scenario: These define the operating condition under which placement and timing analysis are performed (e.g., functional mode, nominal process corner).

Parasitic Tech Files: Required for accurate wire delay estimation using TLU+ files from the PDK.

set_dont_use: Tells the tool to avoid using specific logic gates to enforce your own full adder design.

place_opt: Performs initial placement and optimization for timing and congestion.

legalize_placement: Adjusts any overlapping cells into legal positions on the cell rows.

II. Results

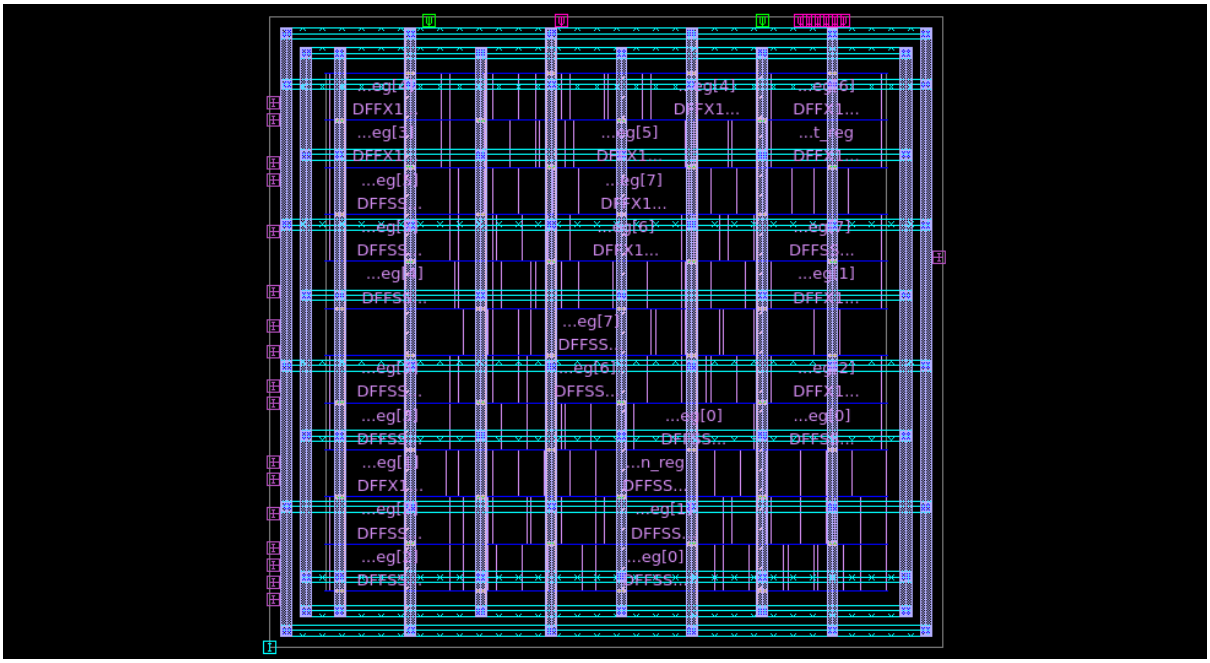

Figure 8.1: Cell Placement: All cells legalized avoiding overlaps and spacing violations

QOR Report

```
****************************************
Report : qor
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:12:13 2025
****************************************


Scenario           'func::nom'
Timing Path Group  '**in2reg_default**'
----------------------------------------
Levels of Logic:                     0
Critical Path Length:             0.50
Critical Path Slack:              1.03
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
----------------------------------------

Scenario           'func::nom'
Timing Path Group  '**reg2out_default**'
----------------------------------------
Levels of Logic:                     0
Critical Path Length:             0.08
Critical Path Slack:              1.12
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
----------------------------------------
```

```
Scenario           'func::nom'
Timing Path Group  'Clock'
----------------------------------------
Levels of Logic:                    16
Critical Path Length:             0.71
Critical Path Slack:              0.89
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
Worst Hold Violation:            -0.00
Total Hold Violation:            -0.00
No. of Hold Violations:              1
----------------------------------------


Cell Count
----------------------------------------
Hierarchical Cell Count:             0
Hierarchical Port Count:             0
Leaf Cell Count:                   103
Buf/Inv Cell Count:                  0
Buf Cell Count:                      0
Inv Cell Count:                      0
Combinational Cell Count:           77
   Single-bit Isolation Cell Count:                    0
   Multi-bit Isolation Cell Count:                     0
   Isolation Cell Banking Ratio:                       0.00%
   Single-bit Level Shifter Cell Count:                0
   Multi-bit Level Shifter Cell Count:                 0
   Level Shifter Cell Banking Ratio:                   0.00%
   Single-bit ELS Cell Count:                          0
   Multi-bit ELS Cell Count:                           0
   ELS Cell Banking Ratio:                             0.00%
Sequential Cell Count:              26
   Integrated Clock-Gating Cell Count:                 0
   Sequential Macro Cell Count:                        0
   Single-bit Sequential Cell Count:                   26
   Multi-bit Sequential Cell Count:                    0
   Sequential Cell Banking Ratio:                      0.00%
   BitsPerflop:                                        1.00
Macro Count:                         0
----------------------------------------


Area
----------------------------------------
Combinational Area:             129.36
Noncombinational Area:          179.43
Buf/Inv Area:                     0.00
Total Buffer Area:                0.00
Total Inverter Area:              0.00
```

```
Macro/Black Box Area:              0.00
Net Area:                             0
Net XLength:                     394.09
Net YLength:                     340.37
----------------------------------------
Cell Area (netlist):                           308.78
Cell Area (netlist and physical only):         308.79
Net Length:                      734.46


Design Rules
----------------------------------------
Total Number of Nets:               136
Nets with Violations:                 0
Max Trans Violations:                 0
Max Cap Violations:                   0
----------------------------------------
```

Timing Report

```
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -report_by design
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:12:35 2025
****************************************

  Startpoint: regB_reg[0] (rising edge-triggered flip-flop clocked by Clock)
  Endpoint: SUM_reg[7] (rising edge-triggered flip-flop clocked by Clock)
  Mode: func
  Corner: nom
  Scenario: func::nom
  Path Group: Clock
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------------------
  clock Clock (rise edge)                  0.00       0.00
  clock network delay (ideal)              0.00       0.00

  regB_reg[0]/CLK (DFFSSRX1_RVT)           0.00       0.00 r
  regB_reg[0]/Q (DFFSSRX1_RVT)             0.09       0.09 f
  U85/Y (NAND2X0_RVT)                      0.03       0.13 r
  U90/Y (NAND4X0_RVT)                      0.04       0.17 f
  U91/Y (NAND2X0_RVT)                      0.05       0.22 r
```

```
U93/Y (AND2X1_RVT)                                0.03       0.24 r
U94/Y (NAND2X0_RVT)                               0.02       0.26 f
U96/Y (AND2X1_RVT)                                0.04       0.30 f
U97/Y (NAND2X0_RVT)                               0.04       0.34 r
U106/Y (NAND4X0_RVT)                              0.05       0.39 f
U107/Y (NAND2X0_RVT)                              0.05       0.44 r
U109/Y (AND2X1_RVT)                               0.03       0.47 r
U110/Y (NAND2X0_RVT)                              0.02       0.48 f
U112/Y (AND2X1_RVT)                               0.05       0.53 f
U113/Y (NAND2X0_RVT)                              0.04       0.57 r
U116/Y (NAND4X0_RVT)                              0.04       0.62 f
U120/Y (NAND2X0_RVT)                              0.05       0.67 r
U67/Y (OR2X1_RVT)                                 0.04       0.71 r
SUM_reg[7]/D (DFFSSRX1_RVT)                       0.00       0.71 r
data arrival time                                            0.71

clock Clock (rise edge)                           2.00       2.00
clock network delay (ideal)                       0.00       2.00
SUM_reg[7]/CLK (DFFSSRX1_RVT)                     0.00       2.00 r
clock uncertainty                                -0.30       1.70
library setup time                               -0.10       1.60
data required time                                           1.60
-----------------------------------------------------------------------
data required time                                           1.60
data arrival time                                           -0.71
-----------------------------------------------------------------------
slack (MET)                                                  0.89
```

9. PHYSICAL DESIGN – CLOCKTREE SYNTHESIS USING ICCII

I. Clock Tree Synthesis Script

After placement, the next step is Clock Tree Synthesis (CTS), which builds a balanced distribution network for the clock signal across all sequential elements. This ensures minimal clock skew and meets timing requirements. The CTS process here used CCD (Clock Concurrent Design) methodology.

Before CTS, a sanity check was performed to validate the pre-clock state of the design. The CTS was then divided into three stages:

```
#######################################################
#pre-clock sanity checks
#######################################################

check_design -checks pre_clock_tree_stage


#####################################################
##        CTS using CCD  commands
#####################################################
##
##stage 1:
##

synthesize_clock_tree
##stage 2:
#
set_app_options -name cts.optimize.enable_local_skew -value true
set_app_options -name cts.compile.enable_local_skew -value true
set_app_options -name cts.compile.enable_global_route -value false
set_app_options -name clock_opt.flow.enable_ccd -value true

clock_opt -to build_clock
#stage 3:
###
clock_opt -from route_clock -to route_clock
clock_opt

#############################################################
#  to save the block
#############################################################

save_block
save_lib

save_block -as FULL_ADD_8BIT_Clock
save_lib
```

check_design -checks pre_clock_tree_stage: Verifies the design is ready for CTS by checking for DRC violations or missing clock definitions.

synthesize_clock_tree: Prepares and identifies clock sinks and insertion points.

set_app_options: Enables local skew optimization and CCD (Clock Concurrent Design) mode for enhanced clock tree balance.

clock_opt -to build_clock: Constructs the actual clock tree structure.

clock_opt -from route_clock -to route_clock: Refines clock tree routing to reduce skew and improve timing.

II. Results

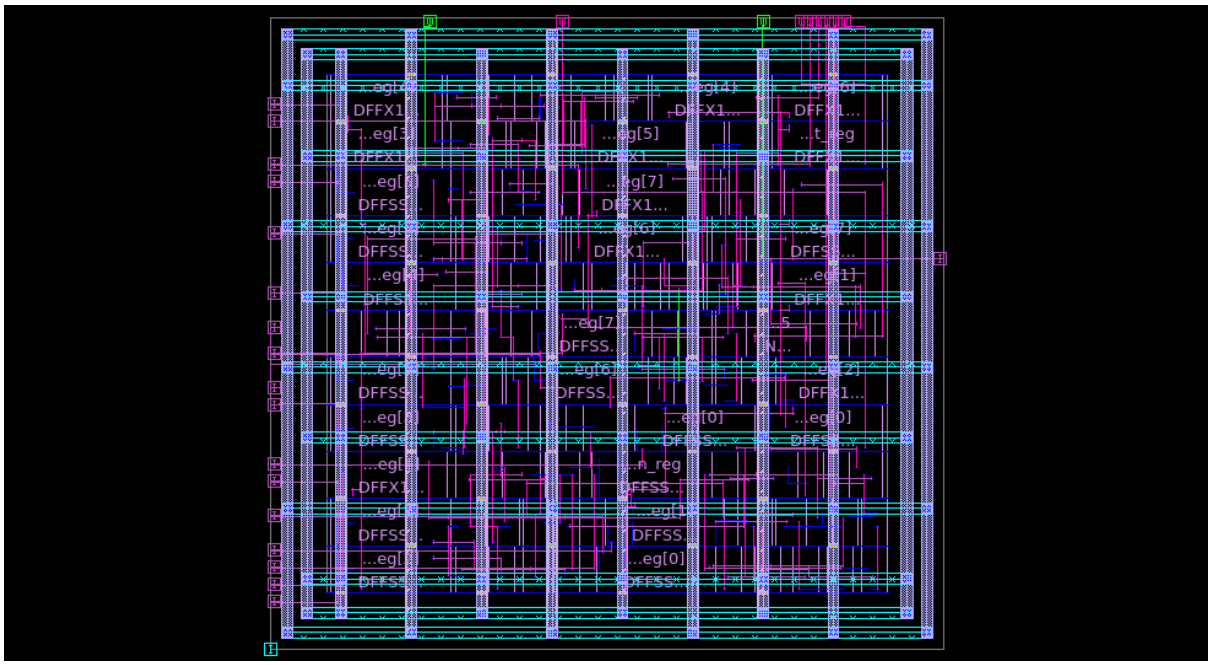

Figure 9.1 Clock Tree Synthesis: Clock Path defined and optimized

QOR Report

```
****************************************
Report : qor
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:15:26 2025
****************************************


Scenario           'func::nom'
Timing Path Group  '**in2reg_default**'
----------------------------------------
Levels of Logic:                     0
Critical Path Length:             0.50
Critical Path Slack:              1.04
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
```

```
----------------------------------------

Scenario           'func::nom'
Timing Path Group  '**reg2out_default**'
----------------------------------------
Levels of Logic:                     0
Critical Path Length:             0.09
Critical Path Slack:              1.08
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
----------------------------------------

Scenario           'func::nom'
Timing Path Group  'Clock'
----------------------------------------
Levels of Logic:                    17
Critical Path Length:             0.69
Critical Path Slack:              0.92
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
Worst Hold Violation:             0.00
Total Hold Violation:             0.00
No. of Hold Violations:              0
----------------------------------------


Cell Count
----------------------------------------
Hierarchical Cell Count:             0
Hierarchical Port Count:             0
Leaf Cell Count:                   106
Buf/Inv Cell Count:                  6
Buf Cell Count:                      3
Inv Cell Count:                      3
Combinational Cell Count:           80
   Single-bit Isolation Cell Count:                     0
   Multi-bit Isolation Cell Count:                      0
   Isolation Cell Banking Ratio:                        0.00%
   Single-bit Level Shifter Cell Count:                 0
   Multi-bit Level Shifter Cell Count:                  0
   Level Shifter Cell Banking Ratio:                    0.00%
   Single-bit ELS Cell Count:                           0
   Multi-bit ELS Cell Count:                            0
   ELS Cell Banking Ratio:                              0.00%
Sequential Cell Count:              26
   Integrated Clock-Gating Cell Count:                  0
   Sequential Macro Cell Count:                         0
   Single-bit Sequential Cell Count:                    26
   Multi-bit Sequential Cell Count:                     0
```

```
   Sequential Cell Banking Ratio:                          0.00%
   BitsPerflop:                                            1.00
Macro Count:                          0
----------------------------------------


Area
----------------------------------------
Combinational Area:              137.75
Noncombinational Area:           179.43
Buf/Inv Area:                     12.20
Total Buffer Area:                 8.39
Total Inverter Area:               3.81
Macro/Black Box Area:              0.00
Net Area:                             0
Net XLength:                     442.33
Net YLength:                     372.90
----------------------------------------
Cell Area (netlist):                           317.17
Cell Area (netlist and physical only):         317.17
Net Length:                      815.22


Design Rules
----------------------------------------
Total Number of Nets:               138
Nets with Violations:                 0
Max Trans Violations:                 0
Max Cap Violations:                   0
----------------------------------------
```

Timing Report

```
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -report_by design
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:15:46 2025
****************************************

  Startpoint: regB_reg[0] (rising edge-triggered flip-flop clocked by Clock)
  Endpoint: SUM_reg[7] (rising edge-triggered flip-flop clocked by Clock)
  Mode: func
  Corner: nom
  Scenario: func::nom
```

```
Path Group: Clock
Path Type: max

Point                                              Incr       Path
-----------------------------------------------------------------------
clock Clock (rise edge)                           0.00       0.00
clock network delay (propagated)                  0.07       0.07

regB_reg[0]/CLK (DFFSSRX1_RVT)                    0.00       0.07 r
regB_reg[0]/Q (DFFSSRX1_RVT)                      0.08       0.16 r
U85/Y (NAND2X0_RVT)                               0.03       0.19 f
U86/Y (NAND2X0_RVT)                               0.03       0.22 r
U87/Y (NAND3X0_RVT)                               0.05       0.27 f
ctmTdsLR_2_710/Y (AND3X1_RVT)                     0.05       0.32 f
ctmTdsLR_1_709/Y (INVX0_RVT)                      0.02       0.34 r
U96/Y (AND2X1_RVT)                                0.03       0.36 r
U97/Y (NAND2X0_RVT)                               0.04       0.40 f
U98/Y (NAND2X0_RVT)                               0.03       0.43 r
U103/Y (NAND3X0_RVT)                              0.05       0.48 f
ctmTdsLR_2_712/Y (AND3X1_RVT)                     0.05       0.54 f
ctmTdsLR_1_711/Y (INVX0_RVT)                      0.02       0.55 r
U112/Y (AND2X1_RVT)                               0.04       0.59 r
U113/Y (NAND2X0_RVT)                              0.03       0.62 f
U118/Y (NAND2X0_RVT)                              0.03       0.65 r
U119/Y (NAND3X0_RVT)                              0.04       0.68 f
U120/Y (NAND2X0_RVT)                              0.05       0.73 r
U67/Y (OR2X1_RVT)                                 0.04       0.77 r
SUM_reg[7]/D (DFFSSRX1_RVT)                       0.00       0.77 r
data arrival time                                            0.77

clock Clock (rise edge)                           2.00       2.00
clock network delay (propagated)                  0.07       2.07
SUM_reg[7]/CLK (DFFSSRX1_RVT)                     0.00       2.07 r
clock uncertainty                                -0.30       1.77
library setup time                               -0.08       1.69
data required time                                           1.69
-----------------------------------------------------------------------
data required time                                           1.69
data arrival time                                           -0.77
-----------------------------------------------------------------------
slack (MET)                                                  0.92
```

10. PHYSICAL DESIGN – ROUTING USING ICCII

I. Routing Script

After Clock Tree Synthesis (CTS), the next step in the digital physical design flow is Routing, which establishes the physical interconnections between the placed standard cells according to the netlist.

The routing process is typically divided into three main stages:

- Global Routing: Finds approximate routes to guide detailed routing.
- Track Assignment: Maps nets to routing tracks.
- Detail Routing: Performs the actual metal layer connections while resolving design rule violations.

```
#########################################################
# set app options
#########################################################
#
#global route
set_app_options -name route.global.timing_driven -value true
set_app_options -name route.global.crosstalk_driven -value false


#track assignment
set_app_options -name route.track.timing_driven -value true
set_app_options -name route.track.crosstalk_driven -value true
#
#

#####################################
#Atomic commands for route_auto
#####################################
route_global
#save_block route_global_database

route_track
#save_block route_track_database

route_detail
#save_block route_detail_database

#route_auto
route_opt

#######################################
# OUTPUTS
#
#
```

```
write_verilog ./results/full_adder_8bit.routed.v
write_sdc -output ./results/full_adder_8bit.routed.sdc
write_parasitics -format spef -output
./results/full_adder_8bit${scenario1}.spef

save_block
save_lib

save_block -as FULL_ADD_8BIT_Route
save_lib
#
```

set_app_options: Ensures routing is timing-driven and considers crosstalk during track assignment.

route_global: Plans rough routes across the chip to avoid congestion.

route_track: Maps those global routes to actual metal tracks.

route_detail: Connects the nets precisely, adhering to DRC (Design Rule Checks).

route_opt: Optional stage to improve congestion, timing, and fix violations post-routing.

write_verilog, write_sdc, write_parasitics: Outputs routed netlist, updated SDC, and parasitic data for timing analysis.

II. Results

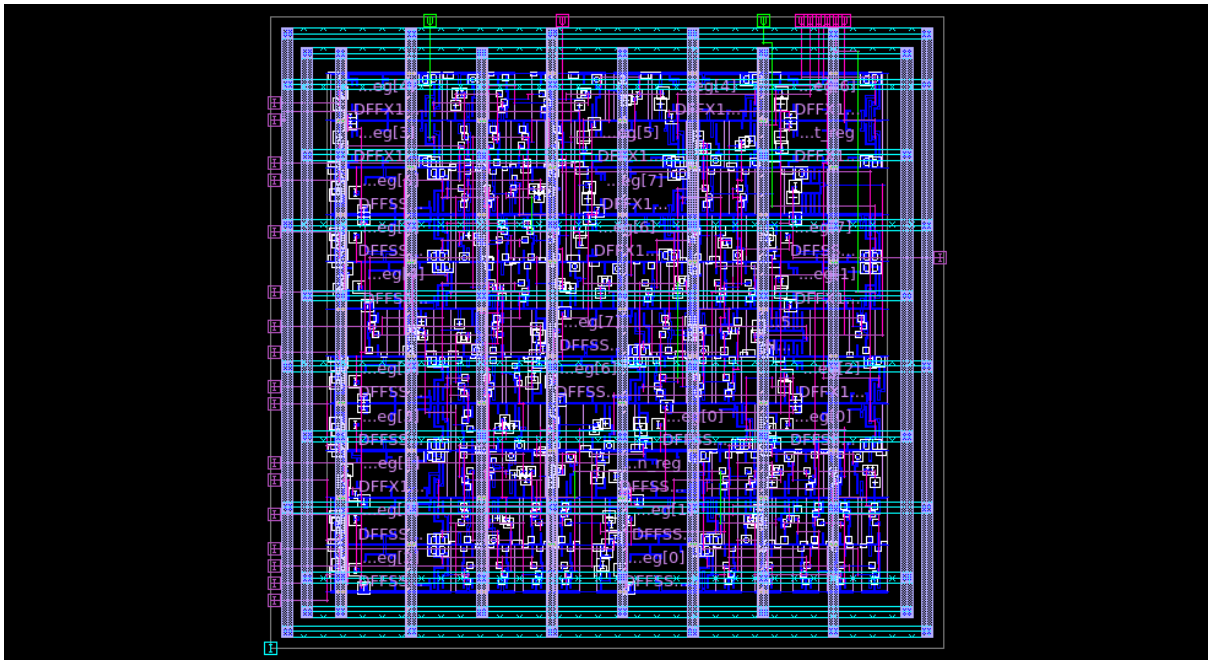

Figure 10.1: Routing: Final Cell Placement with all the paths

QOR Report

```
****************************************
Report : qor
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:19:27 2025
****************************************


Scenario           'func::nom'
Timing Path Group  '**in2reg_default**'
----------------------------------------
Levels of Logic:                     0
Critical Path Length:             0.50
Critical Path Slack:              1.04
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
----------------------------------------

Scenario           'func::nom'
Timing Path Group  '**reg2out_default**'
----------------------------------------
Levels of Logic:                     0
Critical Path Length:             0.09
Critical Path Slack:              1.08
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
----------------------------------------

Scenario           'func::nom'
Timing Path Group  'Clock'
----------------------------------------
Levels of Logic:                    17
Critical Path Length:             0.70
Critical Path Slack:              0.91
Critical Path Clk Period:         2.00
Total Negative Slack:             0.00
No. of Violating Paths:              0
Worst Hold Violation:             0.00
Total Hold Violation:             0.00
No. of Hold Violations:              0
----------------------------------------


Cell Count
----------------------------------------
Hierarchical Cell Count:             0
Hierarchical Port Count:             0
```

```
Leaf Cell Count:                    105
Buf/Inv Cell Count:                   5
Buf Cell Count:                       2
Inv Cell Count:                       3
Combinational Cell Count:            79
   Single-bit Isolation Cell Count:                      0
   Multi-bit Isolation Cell Count:                       0
   Isolation Cell Banking Ratio:                         0.00%
   Single-bit Level Shifter Cell Count:                  0
   Multi-bit Level Shifter Cell Count:                   0
   Level Shifter Cell Banking Ratio:                     0.00%
   Single-bit ELS Cell Count:                            0
   Multi-bit ELS Cell Count:                             0
   ELS Cell Banking Ratio:                               0.00%
Sequential Cell Count:               26
   Integrated Clock-Gating Cell Count:                   0
   Sequential Macro Cell Count:                          0
   Single-bit Sequential Cell Count:                     26
   Multi-bit Sequential Cell Count:                      0
   Sequential Cell Banking Ratio:                        0.00%
   BitsPerflop:                                          1.00
Macro Count:                         0
---------------------------------------


Area
---------------------------------------
Combinational Area:             135.71
Noncombinational Area:          179.43
Buf/Inv Area:                    10.17
Total Buffer Area:                6.35
Total Inverter Area:              3.81
Macro/Black Box Area:             0.00
Net Area:                            0
Net XLength:                    435.89
Net YLength:                    392.58
---------------------------------------
Cell Area (netlist):                          315.14
Cell Area (netlist and physical only):        315.14
Net Length:                     828.47


Design Rules
---------------------------------------
Total Number of Nets:              138
Nets with Violations:                0
Max Trans Violations:                0
Max Cap Violations:                  0
---------------------------------------
```

Timing Report

```
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -report_by design
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:19:04 2025
****************************************

  Startpoint: regB_reg[0] (rising edge-triggered flip-flop clocked by Clock)
  Endpoint: SUM_reg[7] (rising edge-triggered flip-flop clocked by Clock)
  Mode: func
  Corner: nom
  Scenario: func::nom
  Path Group: Clock
  Path Type: max

  Point                                              Incr       Path
  -----------------------------------------------------------------------------
  clock Clock (rise edge)                            0.00       0.00
  clock network delay (propagated)                   0.07       0.07

  regB_reg[0]/CLK (DFFSSRX1_RVT)                     0.00       0.07 r
  regB_reg[0]/Q (DFFSSRX1_RVT)                       0.08       0.16 r
  U85/Y (NAND2X0_RVT)                                0.03       0.19 f
  U86/Y (NAND2X0_RVT)                                0.03       0.22 r
  U87/Y (NAND3X0_RVT)                                0.05       0.27 f
  ctmTdsLR_2_710/Y (AND3X1_RVT)                      0.06       0.32 f
  ctmTdsLR_1_709/Y (INVX0_RVT)                       0.02       0.34 r
  U96/Y (AND2X1_RVT)                                 0.03       0.37 r
  U97/Y (NAND2X0_RVT)                                0.04       0.41 f
  U98/Y (NAND2X0_RVT)                                0.03       0.44 r
  U103/Y (NAND3X0_RVT)                               0.05       0.49 f
  ctmTdsLR_2_712/Y (AND3X1_RVT)                      0.06       0.54 f
  ctmTdsLR_1_711/Y (INVX0_RVT)                       0.02       0.56 r
  U112/Y (AND2X1_RVT)                                0.04       0.60 r
  U113/Y (NAND2X0_RVT)                               0.03       0.63 f
  U118/Y (NAND2X0_RVT)                               0.03       0.66 r
  U119/Y (NAND3X0_RVT)                               0.04       0.69 f
  U120/Y (NAND2X0_RVT)                               0.05       0.74 r
  U67/Y (OR2X1_RVT)                                  0.04       0.78 r
  SUM_reg[7]/D (DFFSSRX1_RVT)                        0.00       0.78 r
  data arrival time                                             0.78

  clock Clock (rise edge)                            2.00       2.00
  clock network delay (propagated)                   0.07       2.07
  SUM_reg[7]/CLK (DFFSSRX1_RVT)                      0.00       2.07 r
```

```
clock uncertainty                          -0.30      1.77
library setup time                         -0.08      1.69
data required time                                    1.69
-------------------------------------------------------------------------
data required time                                    1.69
data arrival time                                    -0.78
-------------------------------------------------------------------------
slack (MET)                                           0.91
```

11. STATIC TIMING ANALYSIS USING PRIMETIME

After the final routed design is completed in ICC2, **PrimeTime** is used for **static timing analysis (STA)**. It verifies that the design meets its timing requirements under different parasitic and corner conditions, using the final routed netlist, parasitic data, and timing constraints (SDC).

I. PT Scripts

For Setup Analysis

```
set report_default_significant_digits 6 ;
set link_path "./../ref/lib/stdcell_rvt/saed32rvt_ff1p16v125c.db"

read_verilog "./../ICCII/results/full_adder_8bit.routed.v"
link_design
current_design full_adder_8bit


read_sdc ./../CONSTRAINTS/full_adder.sdc

read_parasitics
"./../ICCII/results/full_adder_func::nom.spef.p1_125.spef"

update_timing -full

report_timing
report_design

check_timing -verbose > ./reports/check_timing/check_timing.p1_report
report_global_timing > ./reports/timing/report_global_timing.p1_report
report_clock -skew -attribute > ./reports/clock/report_clock.p1_report
report_analysis_coverage >
./reports/analysis_coverage/report_analysis_coverage.p1_report
report_timing -slack_lesser_than 0.0 -delay min_max -nosplit -input -
net  > ./reports/timing/report_timing.p1_report
```

For Hold Analysis

```
set report_default_significant_digits 6 ;
set link_path "./../ref/lib/stdcell_rvt/saed32rvt_ff1p16v125c.db"

read_verilog "./../ICCII/results/full_adder_8bit.routed.v"
link_design
current_design full_adder_8bit


read_sdc ./../CONSTRAINTS/full_adder.sdc

read_parasitics
"./../ICCII/results/full_adder_func::nom.spef.p2_125.spef"
```

```
update_timing -full

report_timing
report_design

check_timing -verbose > ./reports/check_timing/check_timing.p2_report
report_global_timing > ./reports/timing/report_global_timing.p2_report
report_clock -skew -attribute > ./reports/clock/report_clock.p2_report
report_analysis_coverage >
./reports/analysis_coverage/report_analysis_coverage.p2_report
report_timing -slack_lesser_than 0.0 -delay min_max -nosplit -input -
net  > ./reports/timing/report_timing.p2_report
```

II. Results

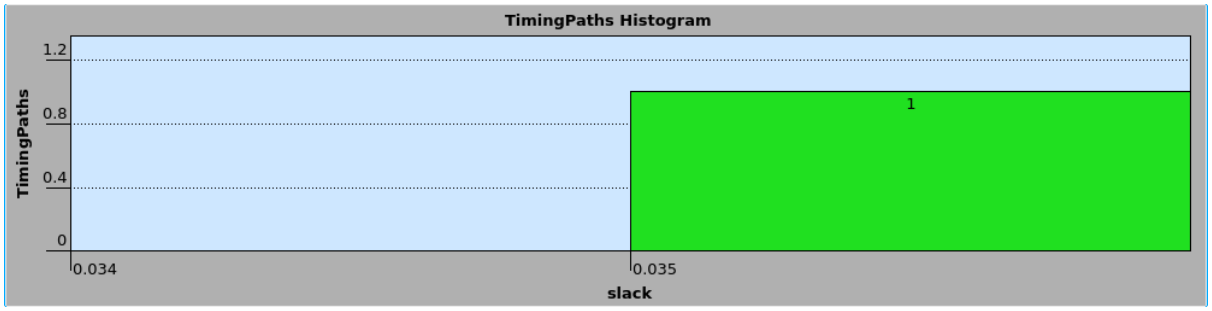

Figure 11.1 Timing Paths Histogram showing Positive Slack (0.035)

Timing Report

```
****************************************
Report : timing
    -path_type full
    -delay_type max
    -max_paths 1
    -sort_by slack
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:29:44 2025
****************************************


  Startpoint: regCin_reg (rising edge-triggered flip-flop clocked by Clock)
  Endpoint: SUM_reg[7] (rising edge-triggered flip-flop clocked by Clock)
  Path Group: Clock
  Path Type: max

  Point                                                   Incr       Path
  -----------------------------------------------------------------------------
  clock Clock (rise edge)                                 0.000000   0.000000
  clock network delay (ideal)                             0.000000   0.000000
  regCin_reg/CLK (DFFSSRX1_RVT)                           0.000000   0.000000 r
```

```
  regCin_reg/QN (DFFSSRX1_RVT)                          0.044293   0.044293 r
  U85/Y (NAND2X0_RVT)                                   0.134848   0.179141 f
  U86/Y (NAND2X0_RVT)                                   0.069157   0.248298 r
  U87/Y (NAND3X0_RVT)                                   0.037456   0.285754 f
  ctmTdsLR_2_710/Y (AND3X1_RVT)                         0.155896   0.441650 f
  ctmTdsLR_1_709/Y (INVX0_RVT)                          0.020493   0.462143 r
  U96/Y (AND2X1_RVT)                                    0.031121   0.493265 r
  U97/Y (NAND2X0_RVT)                                   0.065170   0.558435 f
  U98/Y (NAND2X0_RVT)                                   0.145091   0.703525 r
  U103/Y (NAND3X0_RVT)                                  0.039460   0.742985 f
  ctmTdsLR_2_712/Y (AND3X1_RVT)                         0.155820   0.898805 f
  ctmTdsLR_1_711/Y (INVX0_RVT)                          0.020487   0.919292 r
  U112/Y (AND2X1_RVT)                                   0.035711   0.955003 r
  U113/Y (NAND2X0_RVT)                                  0.395495   1.350498 f
  U118/Y (NAND2X0_RVT)                                  0.069272   1.419770 r
  U119/Y (NAND3X0_RVT)                                  0.032035   1.451806 f
  U120/Y (NAND2X0_RVT)                                  0.070932   1.522738 r
  U67/Y (OR2X1_RVT)                                     0.076558   1.599296 r
  SUM_reg[7]/D (DFFSSRX1_RVT)                           0.011994   1.611290 r
  data arrival time                                                1.611290

  clock Clock (rise edge)                               2.000000   2.000000
  clock network delay (ideal)                           0.000000   2.000000
  SUM_reg[7]/CLK (DFFSSRX1_RVT)                                    2.000000 r
  clock reconvergence pessimism                         0.000000   2.000000
  clock uncertainty                                    -0.300000   1.700000
  library setup time                                   -0.053678   1.646322
  data required time                                               1.646322
  ---------------------------------------------------------------------------
  data required time                                               1.646322
  data arrival time                                               -1.611290
  ---------------------------------------------------------------------------
  slack (MET)                                                      0.035032
```

QOR Report

```
****************************************
Report : qor
Design : full_adder_8bit
Version: W-2024.09
Date   : Thu Jun  5 11:30:42 2025
****************************************

  Timing Path Group 'Clock' (max_delay/setup)
  -----------------------------------------
  Levels of Logic:                         18
  Critical Path Length:                1.611290
  Critical Path Slack:                 0.035032
  Total Negative Slack:                0.000000
  No. of Violating Paths:                     0
```

```
--------------------------------------------

Timing Path Group 'Clock' (min_delay/hold)
--------------------------------------------
Levels of Logic:                          3
Critical Path Length:              0.131979
Critical Path Slack:               0.028003
Total Negative Slack:              0.000000
No. of Violating Paths:                   0
--------------------------------------------

Area
--------------------------------------------
Net Interconnect area:            45.962540
Total cell area:                 315.138489
Design Area:                     361.101013
--------------------------------------------


Cell & Pin Count
--------------------------------------------
Pin Count:                              378
Hierarchical Cell Count:                  0
Hierarchical Port Count:                  0
Leaf Cell Count:                        105
--------------------------------------------


Design Rule Violations
--------------------------------------------
Total No. of Pins in Design:            378
Total DRC Cost:                    0.000000
--------------------------------------------
```

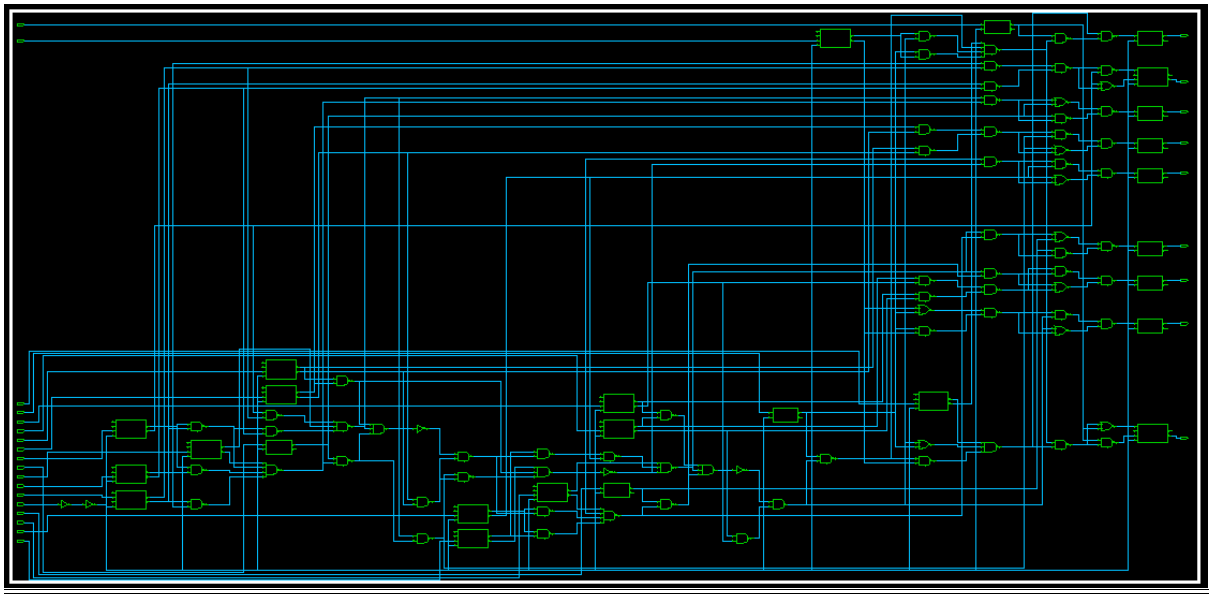

Figure 11.2: Final Internal Schematic

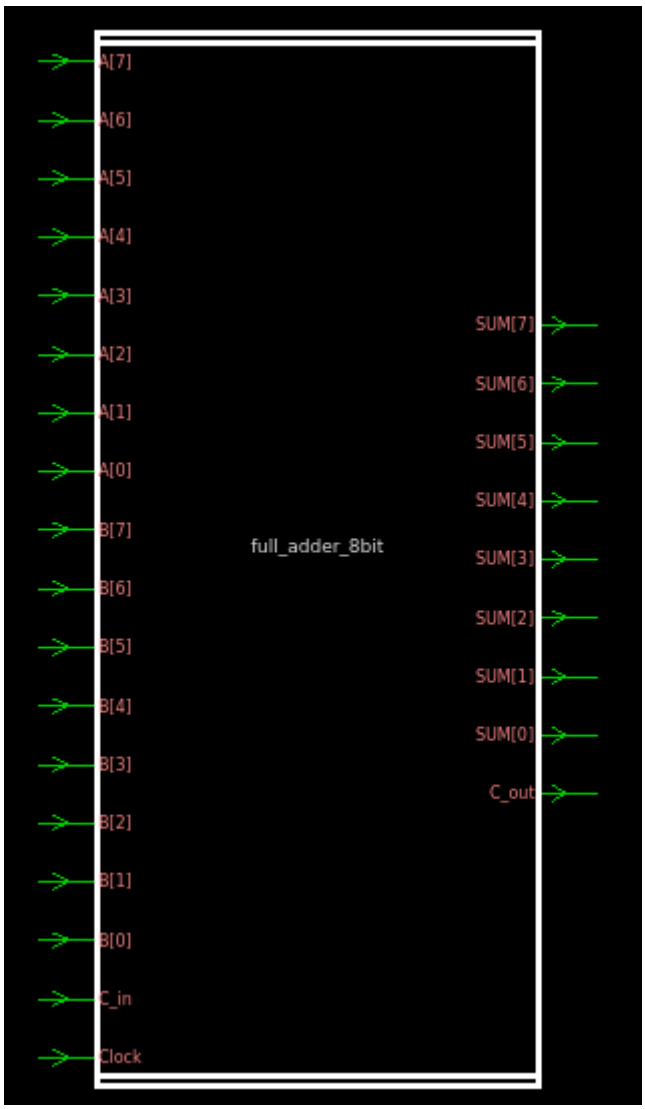

Figure 11.3: Final Block Schematic

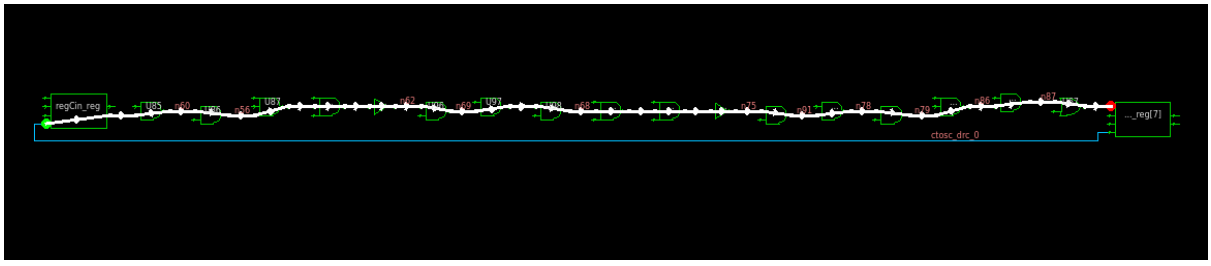

Figure 11.4: Clock Path Schematic

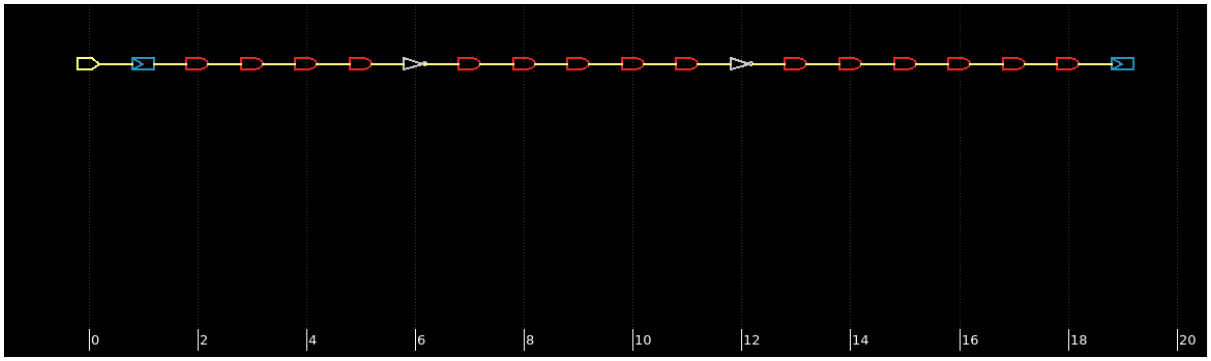

Figure 11.5: Clock Path Graph

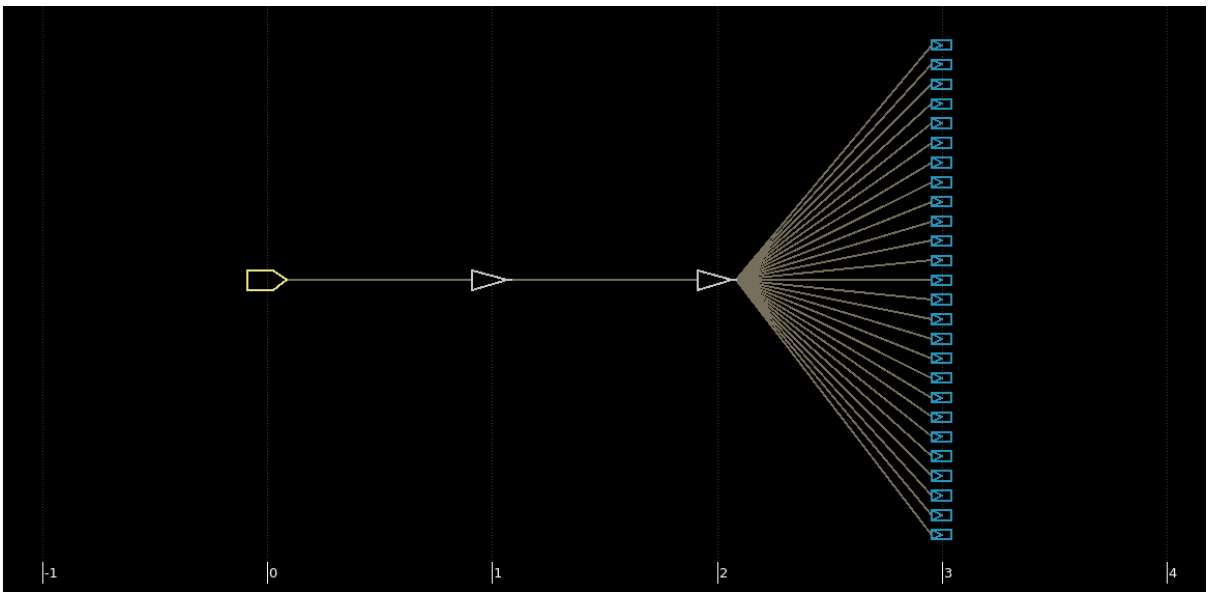

Figure 11.6: Clock Tree

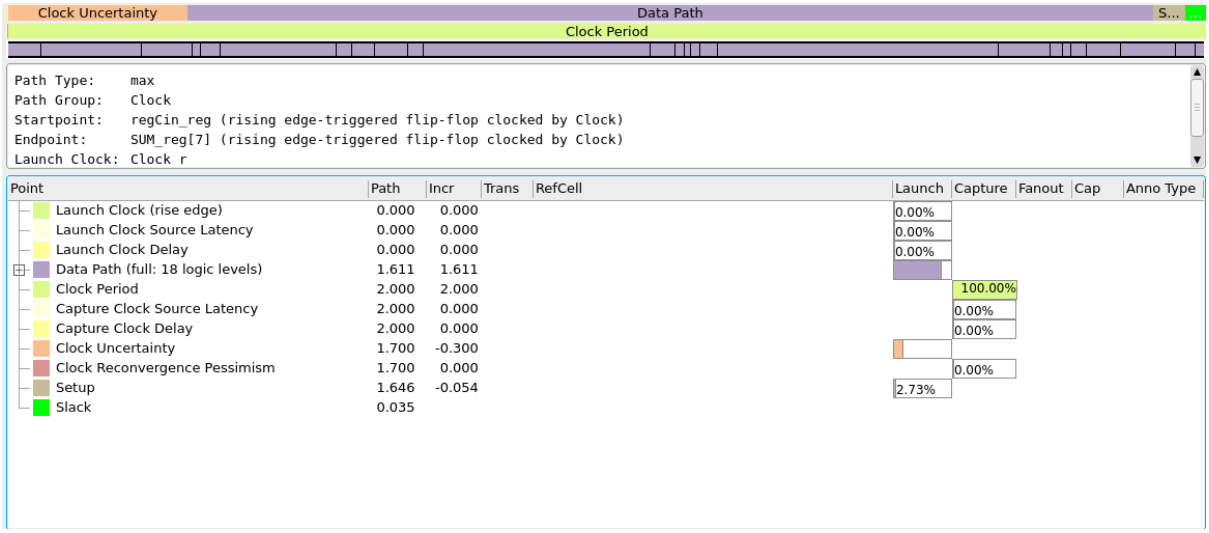

Figure 11.7: Clock Path Inspection (100% Capture)

12. LEARNINGS AND OUTCOMES

1. RTL Design & Testbench Creation

- Understood modular design by instantiating 1-bit full adders to build an 8-bit RCA.
- Learned how to simulate and verify logic functionality using VCS, observing waveforms and ensuring expected outputs.

2. Logic Synthesis (Design Compiler)

- Gained hands-on experience in synthesizing the RTL to a gate-level netlist.
- Learned how to apply timing constraints (SDC) and optimize for area, power, and timing.
- Understood the impact of cell choices, fanout, and transition limits on timing QoR.

3. Floorplanning (ICC2)

- Learned how to define a core area, place I/O pins based on functional roles (inputs, outputs, clock).
- Practiced pin planning and standard cell placement using proper aspect ratios and utilization.

4. Power Planning

- Understood the need for robust Power Delivery Network (PDN) design.
- Implemented power rings, meshes, and rails using VDD and VSS nets.
- Ensured proper connectivity across power domains.

5. Placement & Optimization

- Learned to place and legalize cells while applying timing-driven constraints.
- Applied parasitic modeling (TLU+) and optimization strategies to meet design corners.

6. Clock Tree Synthesis (CTS)

- Understood the importance of low clock skew and latency.
- Implemented CCD-based CTS and enabled local skew optimization.

7. Routing

- Completed global, track, and detailed routing with timing and crosstalk-driven settings.
- Generated final netlists and parasitic files (SPEF) for sign-off.

8. Static Timing Analysis (PrimeTime)

- Verified design timing under setup (late) and hold (early) conditions.
- Analysed timing reports, slack violations, clock skews, and coverage to ensure timing closure.

Outcomes

Successfully implemented a complete digital design flow from RTL to GDSII for an 8-bit RCA.

Met functional and timing correctness at each stage of the design.

Gained hands-on experience with the following Synopsys tools:

- VCS – Simulation
- Design Compiler (DC) – Logic Synthesis
- ICC2 – Physical Design (Floorplan, Power, Placement, CTS, Routing)
- PrimeTime (PT) – Static Timing Analysis

Achieved positive timing slack in PrimeTime, validating correct design closure.