RTL-GDS II IMPLEMENTATION OF HIERARCHAL ALU DESIGN

ANNAYYAGARI SRUJAN REDDY

TRAINING AND INTERSHIP PROGRAM – <u>VLSI EXPERT</u>

1. Introduction

This project focuses on the **complete RTL-to-GDSII implementation** of a **hierarchical ALU (Arithmetic Logic Unit)** design using Synopsys industry tools. The ALU performs a combination of arithmetic and logic operations and is implemented using hierarchical design principles.

The flow follows the ASIC implementation cycle:

- RTL Design using **Verilog.**
- Functional Simulation using **VCS & Verdi**
- Logic Synthesis via **Design Compiler**
- Physical Design through **ICC2**
- Final layout and **GDSII generation**

This work was conducted under the guidance of "VLSI Expert Institute", where extensive hands-on experience was gained in tool flows and digital design implementation.

2. RTL Design

2.1 Hierarchical Architecture

The ALU is designed using a hierarchical approach. The top module `hybrid_alu` instantiates a reusable `full_adder` block for implementing arithmetic operations such as addition, subtraction, increment, and decrement.

2.2 Functional Blocks

- **Arithmetic Unit**: Performs A+B, A–B, A+1, A–1 using instances of `full_adder`.
- **Logic Unit**: Computes A AND B, A OR B, A XOR B, NOT A, shift operations.
- **Control Unit**: Selects output based on 3-bit `ALU_Sel` signal.
- **Clocked Design**: All outputs are clock-synchronized, making it synthesis-friendly.

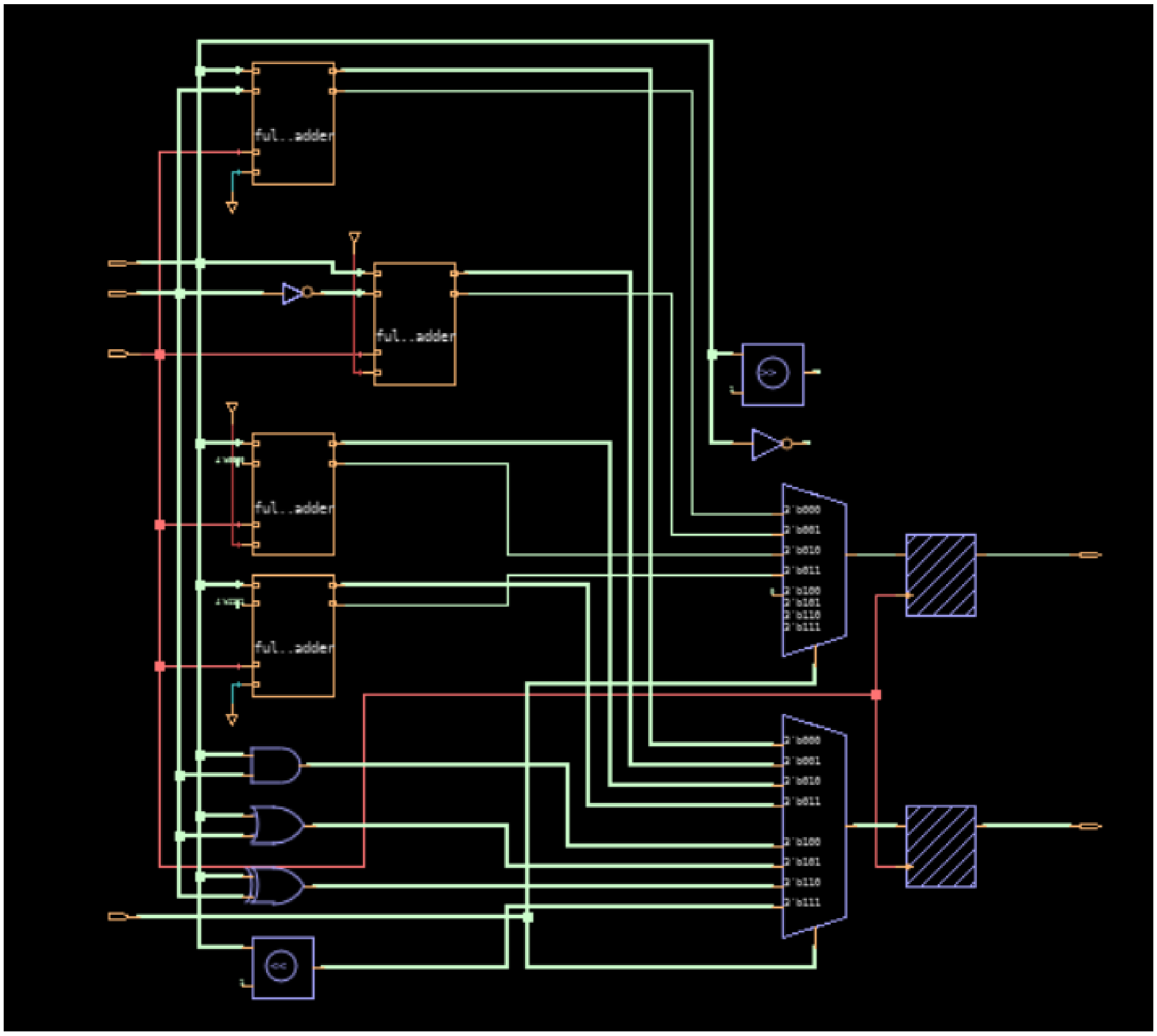
2.3 RTL CODE :

Top_module :

```
module hybrid_alu (
 input [3:0] A, B,
   input Clock,
   input [2:0] ALU_Sel,  // Selects which output to drive
   output reg [3:0] Result,
   output reg C_out
);

   wire [3:0] sum_add, sum_sub, sum_inc, sum_dec;
   wire c_add, c_sub, c_inc, c_dec;

   //  Soft macro: A + B
   full_adder FA_add (
      .A(A),
      .B(B),
      .Clock(Clock),
      .C_in(1'b0),
      .SUM(sum_add),
      .C_out(c_add)
   );

   // Soft macro: A - B = A + (~B + 1)
   full_adder FA_sub (
      .A(A),
      .B(~B),
      .Clock(Clock),
      .C_in(1'b1),
      .SUM(sum_sub),
      .C_out(c_sub)
   );

   // Soft macro: A + 1
   full_adder FA_inc (
      .A(A),
      .B(4'b0000),
      .Clock(Clock),
      .C_in(1'b1),
      .SUM(sum_inc),
      .C_out(c_inc)
   );
 // Soft macro: A - 1
   full_adder FA_dec (
      .A(A),
      .B(4'b1111), // -1 in 2's comp
      .Clock(Clock),
      .C_in(1'b0),
      .SUM(sum_dec),
      .C_out(c_dec)
   );

   // Standard cell–based logic
   wire [3:0] and_op = A & B;
   wire [3:0] or_op  = A | B;
   wire [3:0] xor_op = A ^ B;
   wire [3:0] not_a  = ~A;
   wire [3:0] shl    = A << 1;  // Shift left
   wire [3:0] shr    = A >> 1;  // Shift right

   // Output selection
   always @ (posedge Clock) begin
      case (ALU_Sel)
         3'b000: begin Result <= sum_add; C_out <= c_add; end     // A + B
         3'b001: begin Result <= sum_sub; C_out <= c_sub; end     // A - B
         3'b010: begin Result <= sum_inc; C_out <= c_inc; end     // A + 1
         3'b011: begin Result <= sum_dec; C_out <= c_dec; end     // A - 1
         3'b100: begin Result <= and_op;  C_out <= 0; end         // A AND B
         3'b101: begin Result <= or_op;   C_out <= 0; end         // A OR B
         3'b110: begin Result <= xor_op;  C_out <= 0; end         // A XOR B
         3'b111: begin Result <= shl;     C_out <= 0; end         // Shift A left
         // You can add more like NOT A or SHR if needed
      endcase
   end

endmodule
```

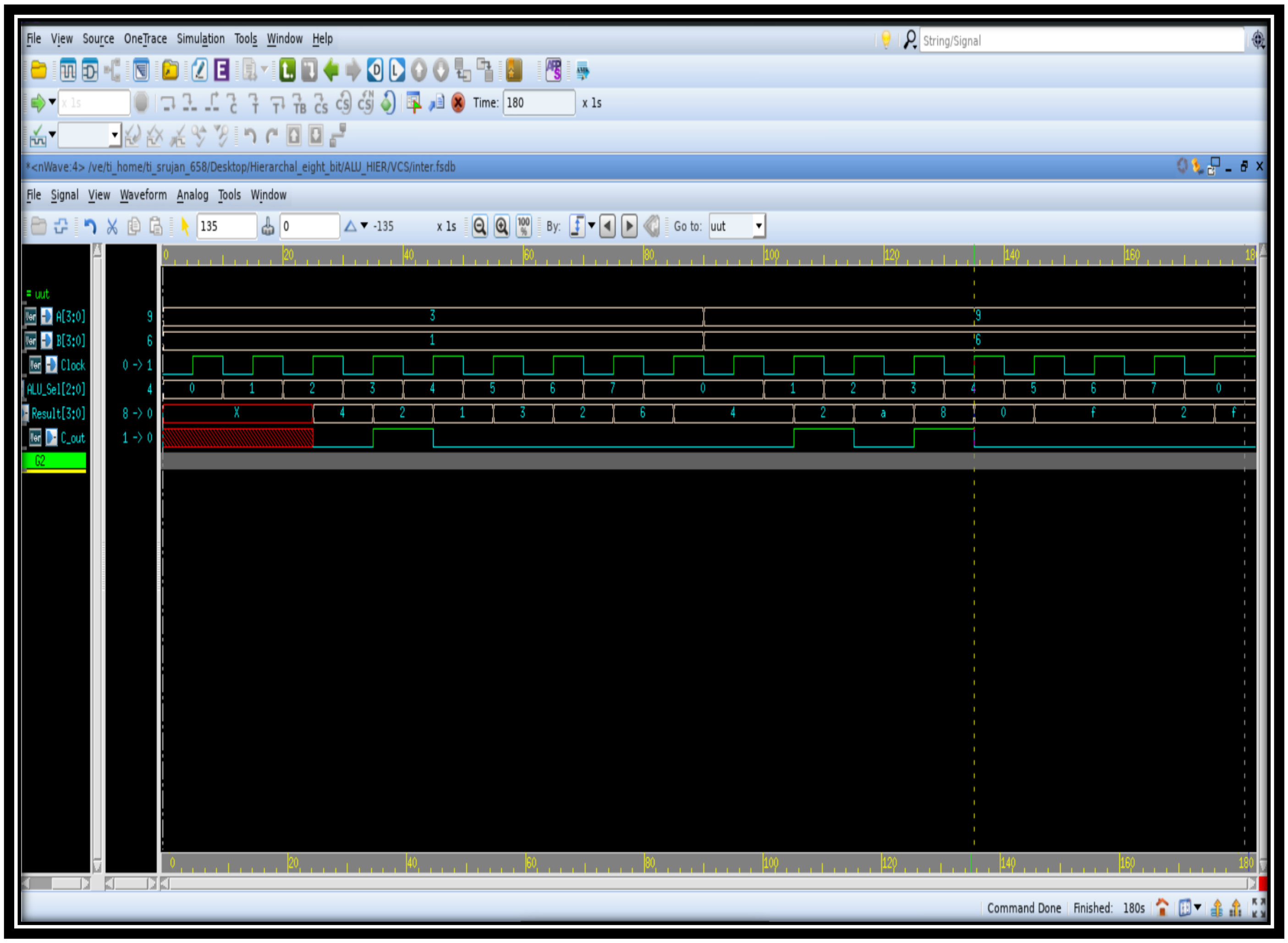
2.4 Simulation Using VCS and VERDI

Schematic :

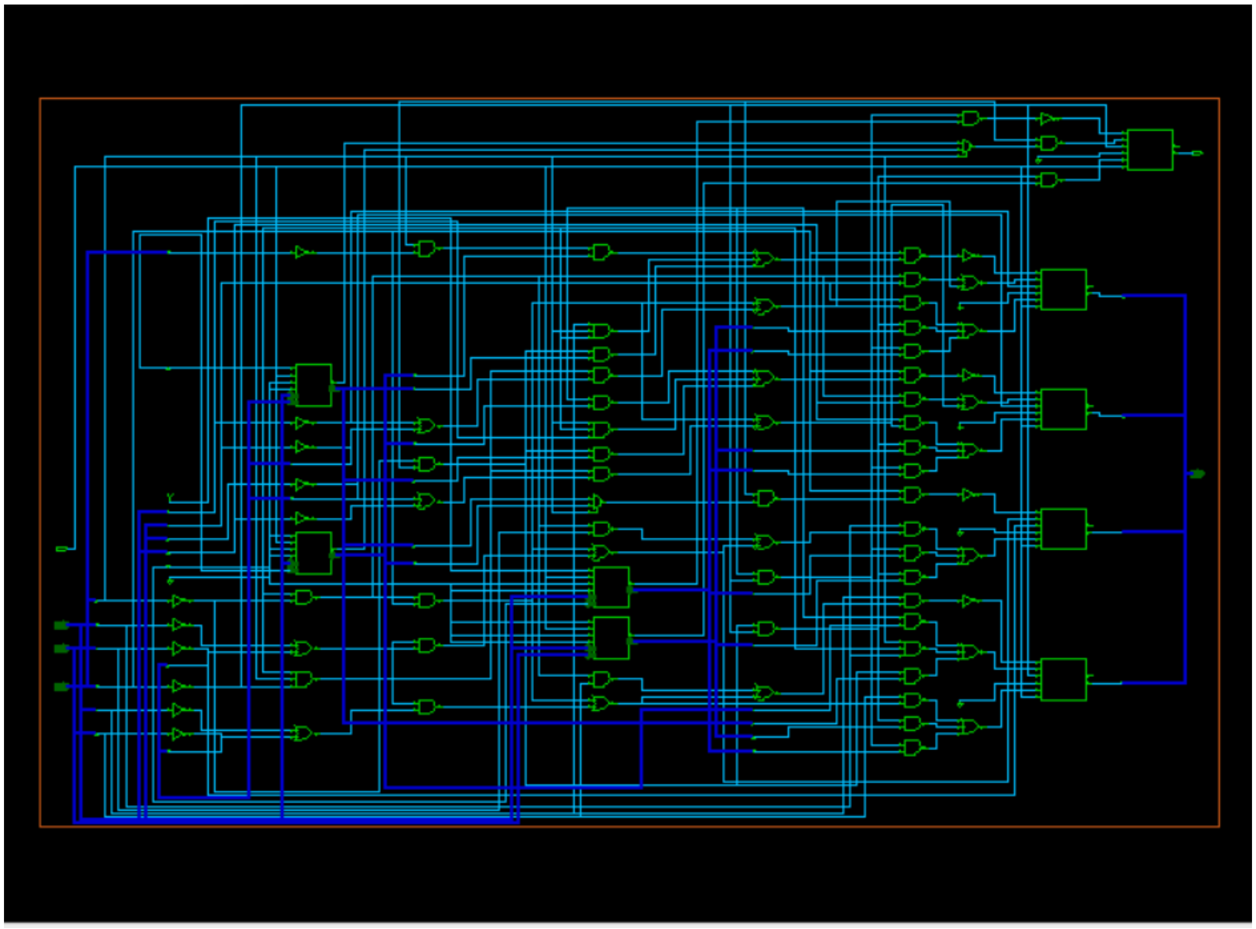


Schematic of Top module hybrid_alu

Waveform Of Simulation :


File View Source OneTrace Simulation Tools Window Help
Time: 180 x 1s
<nWave:4> /ve/ti_home/ti_srujan_658/Desktop/Hierarchal_eight_bit/ALU_HIER/VCS/inter.fsdb
File Signal View Waveform Analog Tools Window
135
0
-135
x 1s
By:
Go to: uut
uut
A[3:0] 9
B[3:0] 6
Clock 0 -> 1
ALU_Sel[2:0] 4
Result[3:0] 8 -> 0
C_out 1 -> 0
G2
Command Done Finished: 180s

3. Synthesis :



This is the snapshot of Synthesis that I have done for the alu design .

The below is the qor report of synthesis:

```
Timing Path Group 'clk'
  -----------------------------------
  Levels of Logic:              7.00
  Critical Path Length:         0.64
  Critical Path Slack:          1.13
  Critical Path Clk Period:     3.40
  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:                90
  Buf/Inv Cell Count:             16
  Buf Cell Count:                  0
  Inv Cell Count:                 16
  CT Buf/Inv Cell Count:           0
  Combinational Cell Count:       81
  Sequential Cell Count:           9
  Macro Count:                     4
  -----------------------------------
  Area
  -----------------------------------
  Combinational Area:         159.602438
  Noncombinational Area:       44.475198
  Buf/Inv Area:               20.331520
  Total Buffer Area:           0.00
  Total Inverter Area:        20.33
  Macro/Black Box Area:       822.841919
  Net Area:                   42.980249
  -----------------------------------
  Cell Area:                1026.919555
  Design Area:              1069.899804


  Design Rules
  -----------------------------------
  Total Number of Nets:          125
  Nets With Violations:            0
  Max Trans Violations:            0
  Max Cap Violations:              0
  -----------------------------------
```

4. Physical Design Flow Documentation for Hybrid Hierarchical ALU

Objective

The purpose of this physical design script is to implement a hierarchical **Hybrid ALU** by integrating pre-placed full adders (`FA_add`, `FA_sub`, `FA_inc`, `FA_dec`) using Synopsys ICC2 tool. The script defines power planning, placement, clock tree synthesis (CTS), routing, and timing analysis.

1. **Library and Design Initialization :**

   Commands :

   create_lib -ref_libs "/data/pdk/pdk32nm/SAED32_EDK/lib/stdcell_rvt/ndm/saed32rvt_c.ndm ../../FULL_ADDER/icc2/FA_LIB/lib.ndm " ALU_LIB

   read_verilog ../DC/MAPPED_HYBRID_ALU.v

   link_block

   start_gui

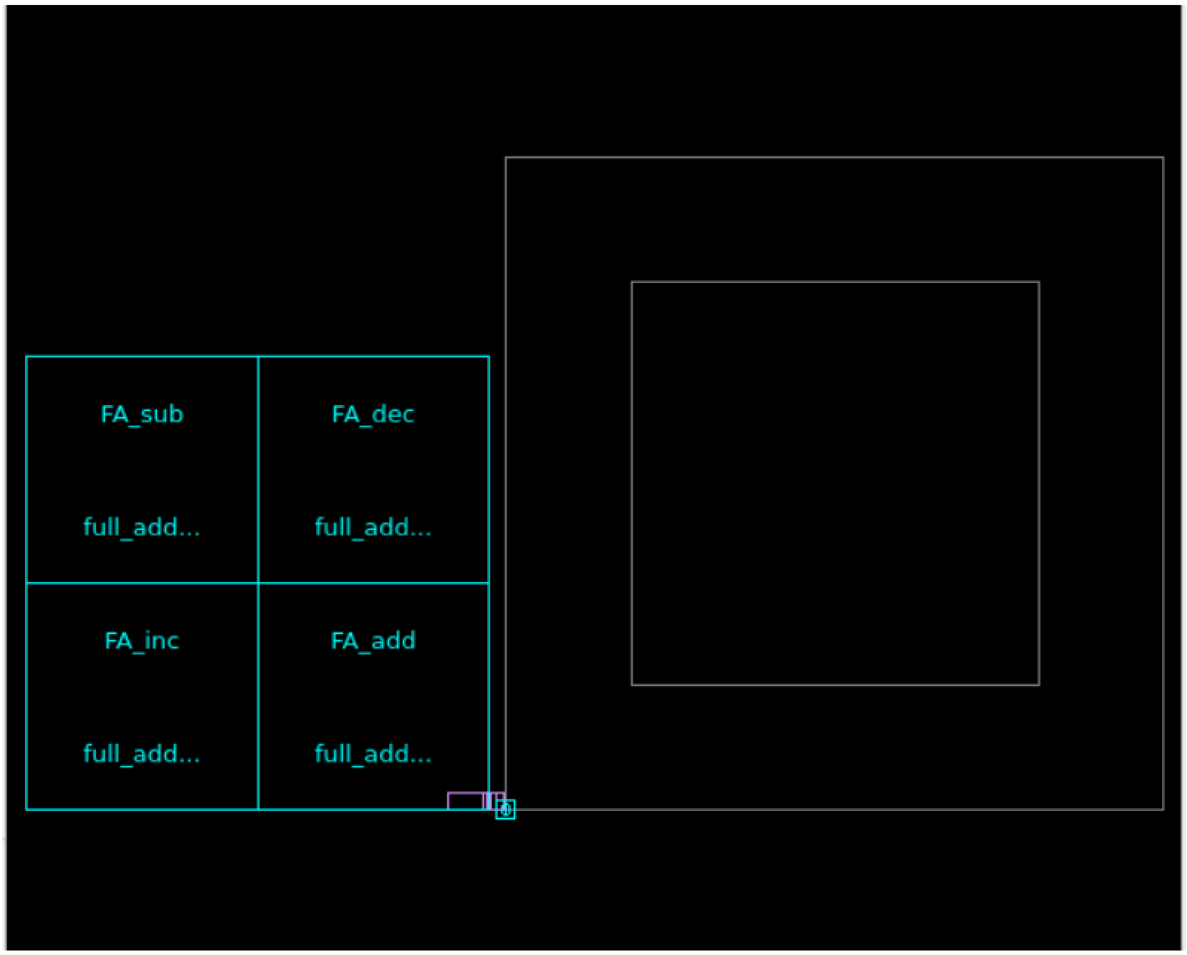
- The **standard cell libraries** (SAED32nm and custom FA_LIB) are loaded to form the basis of design cells.

- The **synthesized Verilog netlist** (`MAPPED_HYBRID_ALU.v`) is read and linked to create the initial database.

2. Floorplan and Soft Macro Placement :

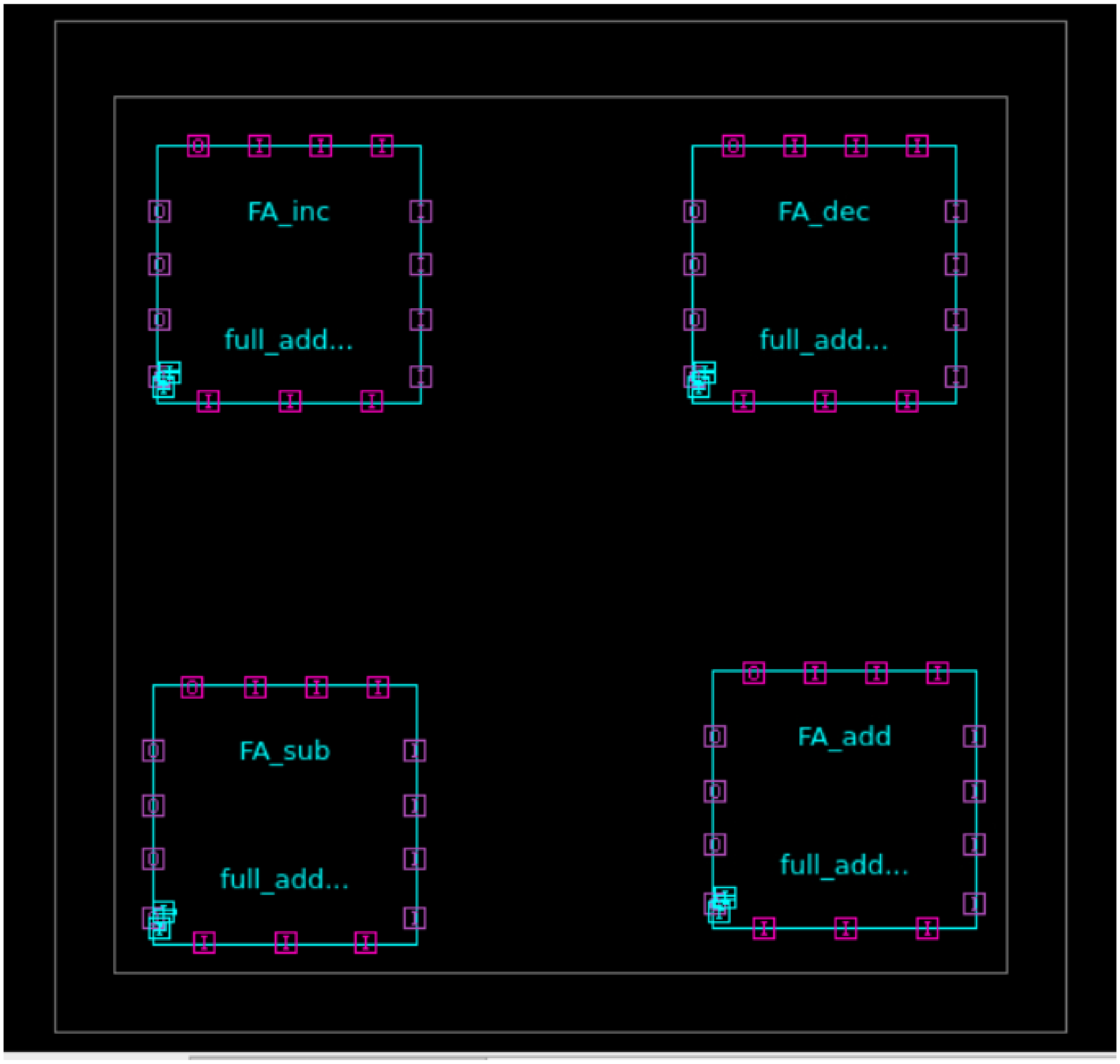
```
read_tech_lef ../../FULL_ADDER/icc2/full_adder.lef

initialize_floorplan -boundary {{0 0} {0 75 } {75 75} {75 0}} -core_offset 5
```

- Reads the **technology LEF** to understand cell geometries.
- The **floorplan** is initialized with core and boundary dimensions. Soft macros are manually placed , by analyzing the **flylines** and assigned **keepout margins** to prevent cell overlaps and routing congestion.

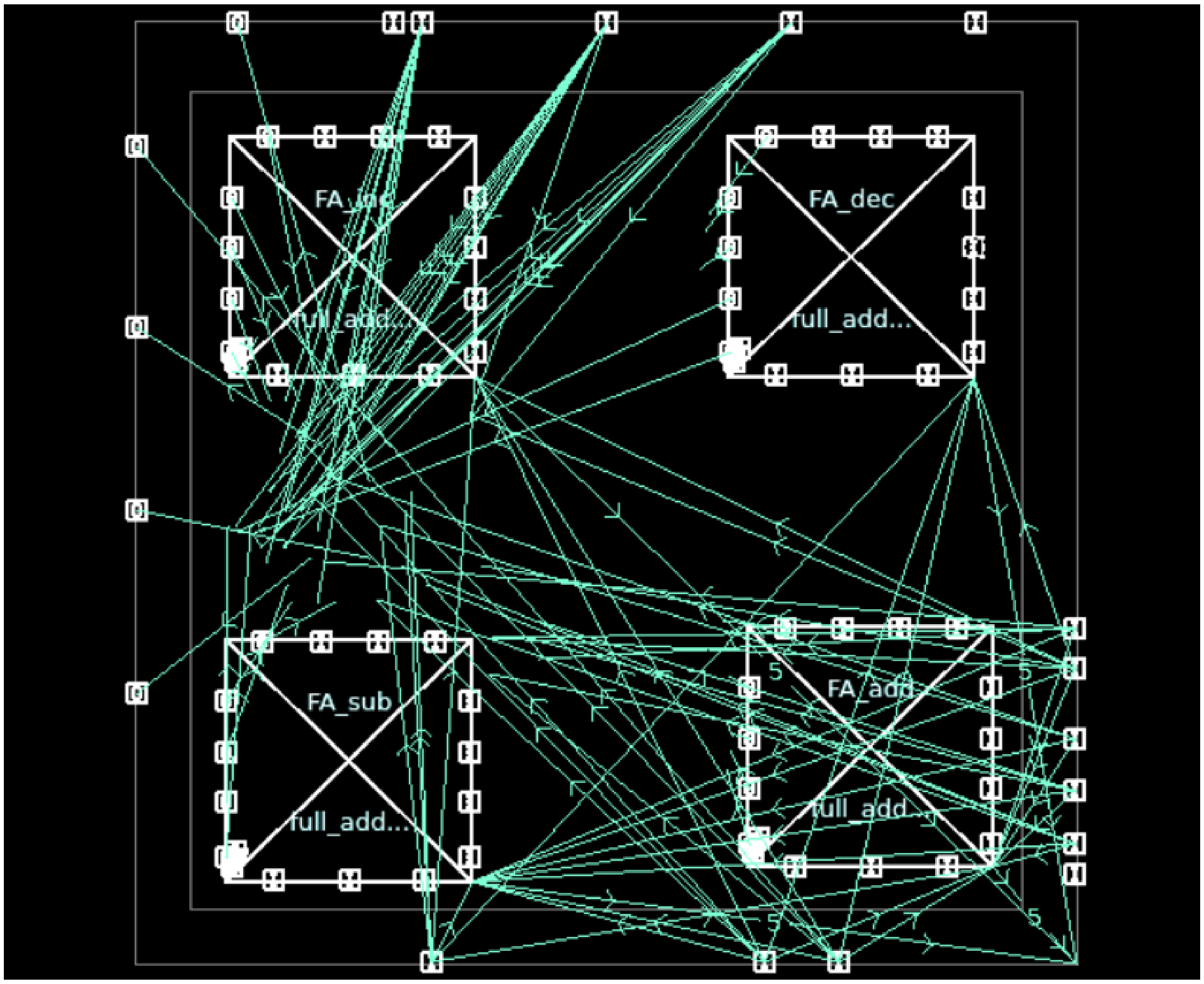


Floorplan initialization

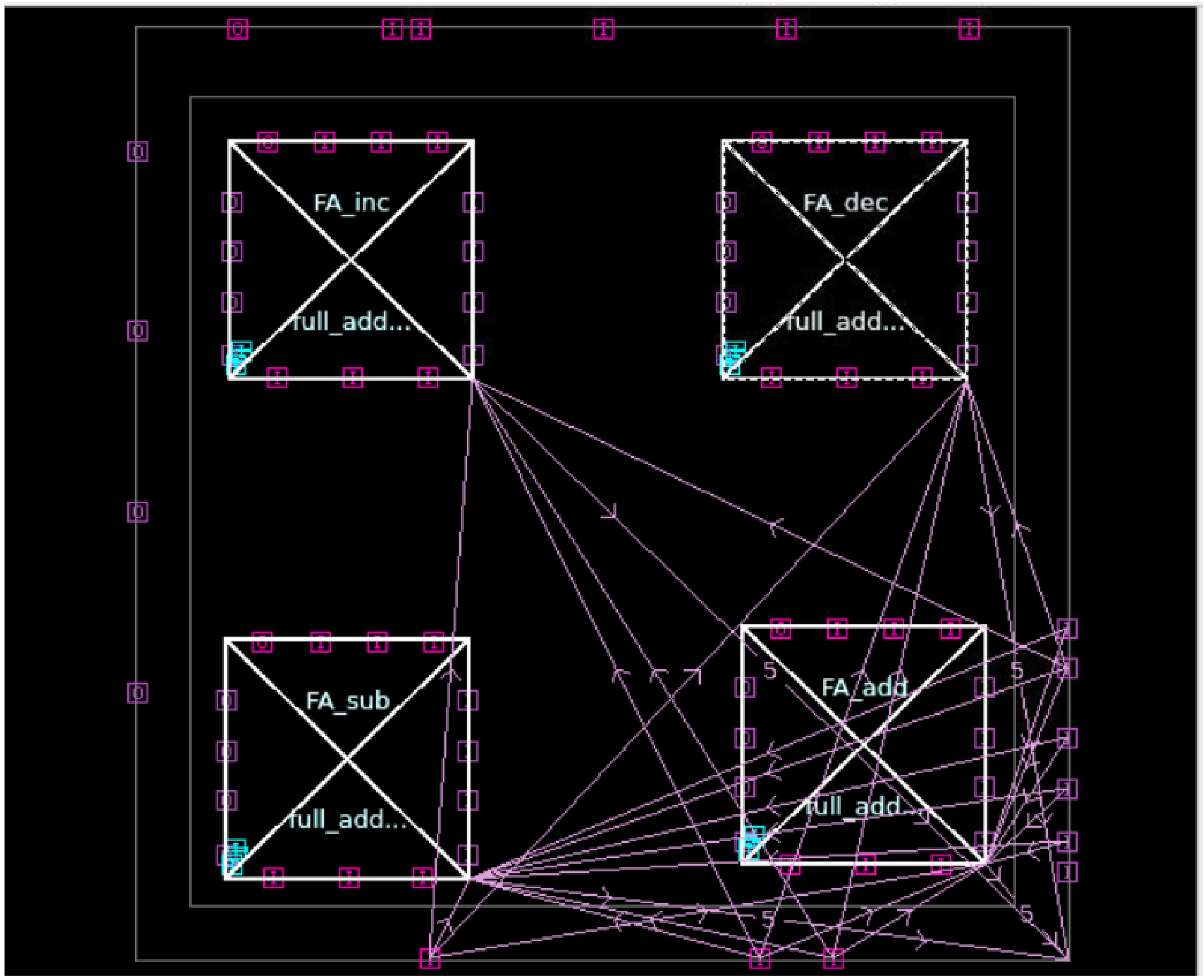
// place four soft macros manually by carefully analyzing the flylines of macros to ports .



MACRO placement



Fly lines of macros to ports and standard-cells

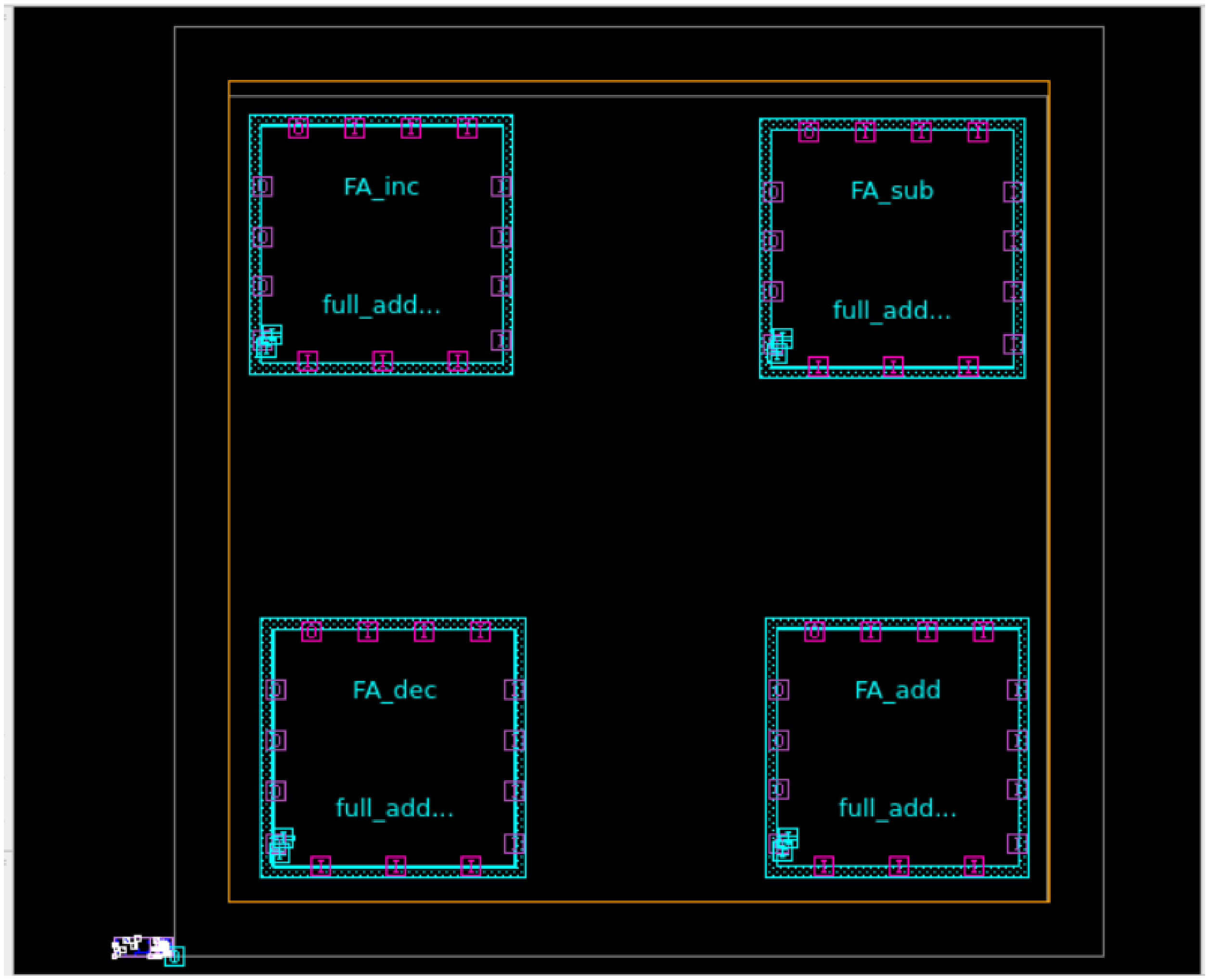


Macro to ports fly-line connections

3. Power Planning :

Commands :

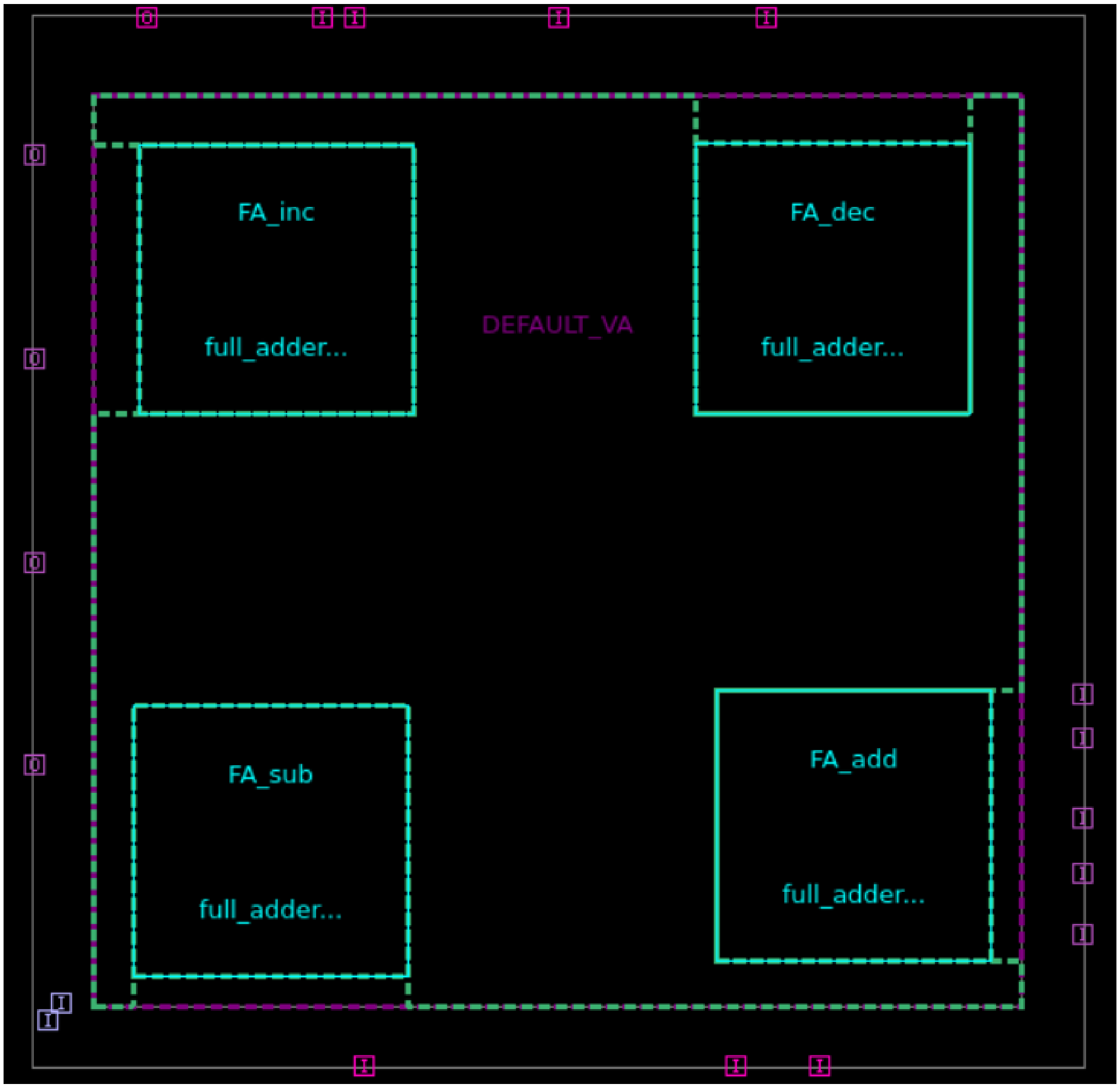
```
create_keepout_margin   -outer   {1   1   1   1}   -type   hard   FA_dec
create_keepout_margin   -outer   {1   1   1   1}   -type   hard   FA_sub
create_keepout_margin   -outer   {1   1   1   1}   -type   hard   FA_inc
create_keepout_margin   -outer   {1   1   1   1}   -type   hard   FA_add
```



Keep-out Margin Creation around the Soft Macro

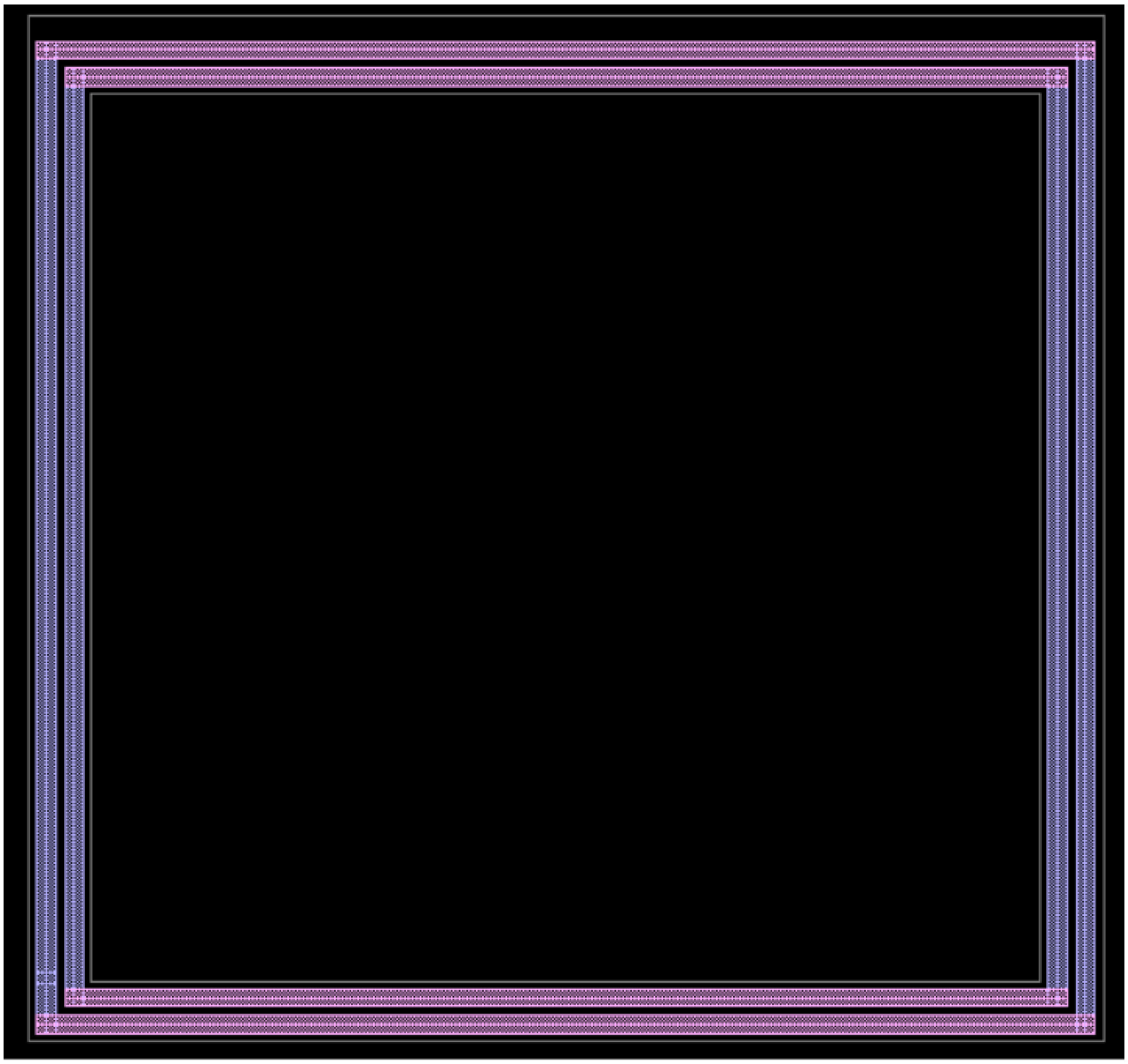
```
set_attribute -objects [get_cells "FA_add FA_sub FA_dec FA_inc"] -name physical_status -
value placed
```

```
create_port -direction in VDD
create_port -direction in VSS
create_net -power VDD
create_net -ground VSS
create_pg_region pg_dec -block FA_dec
create_pg_region pg_add -block FA_add
create_pg_region pg_sub -block FA_sub
create_pg_region pg_inc -block FA_inc
create_pg_region -core ALU_CORE -exclude_regions "pg_add pg_dec pg_sub pg_inc "
place_pins –self
```

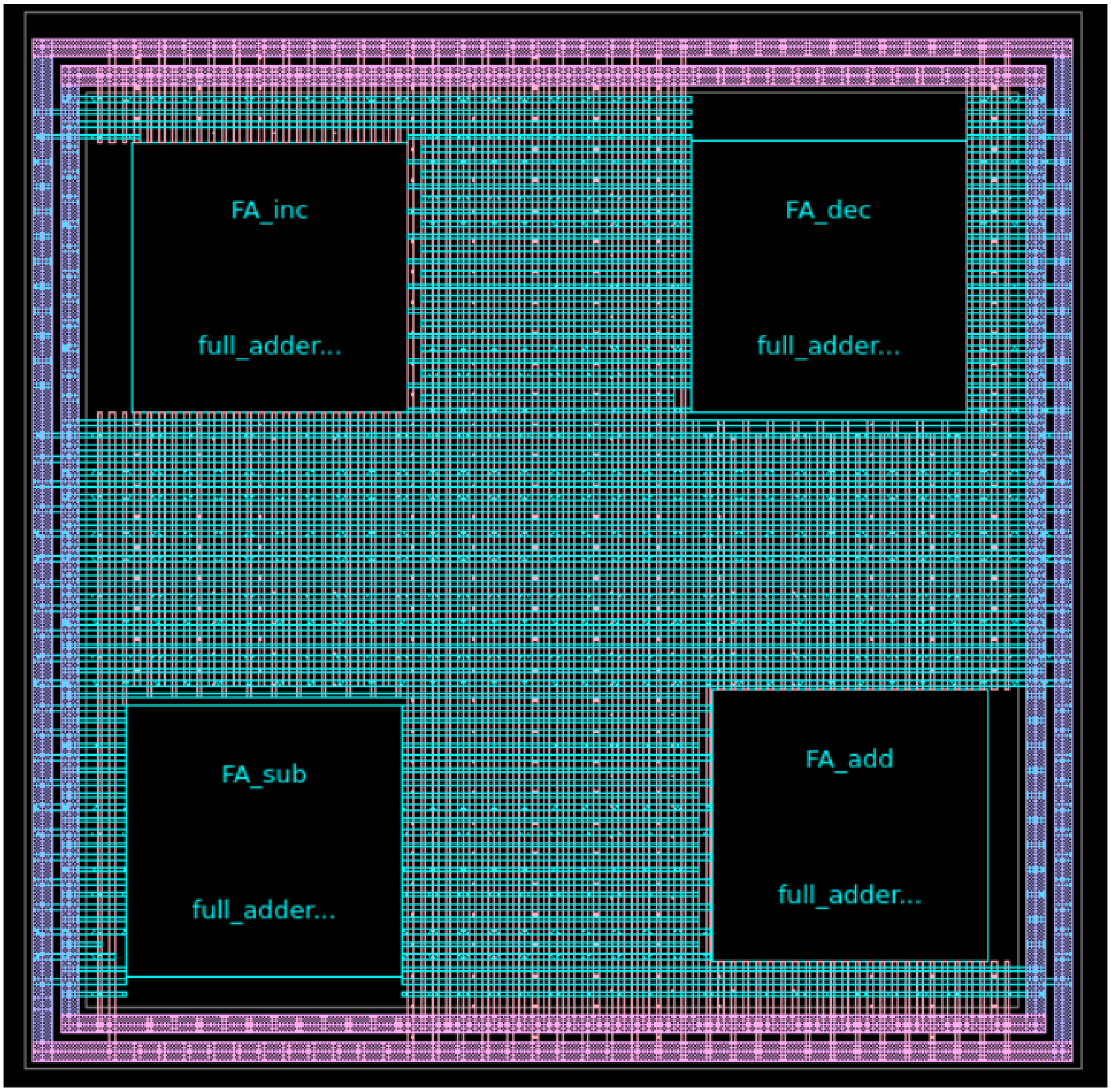


PG-Region Creation and pin-placement

```
connect_pg_net -create_nets_only

create_pg_ring_pattern core_ring_pattern -horizontal_layer M9 -horizontal_width 1.5 -
horizontal_spacing 0.7 -vertical_layer M8 -vertical_width 1.5 -vertical_spacing 0.7


set_pg_strategy core_power_ring -core -pattern {{name:core_ring_pattern} {nets:{VDD VSS}}
{offset: {0.65 0.65}}}
compile_pg -strategies core_power_ring
```



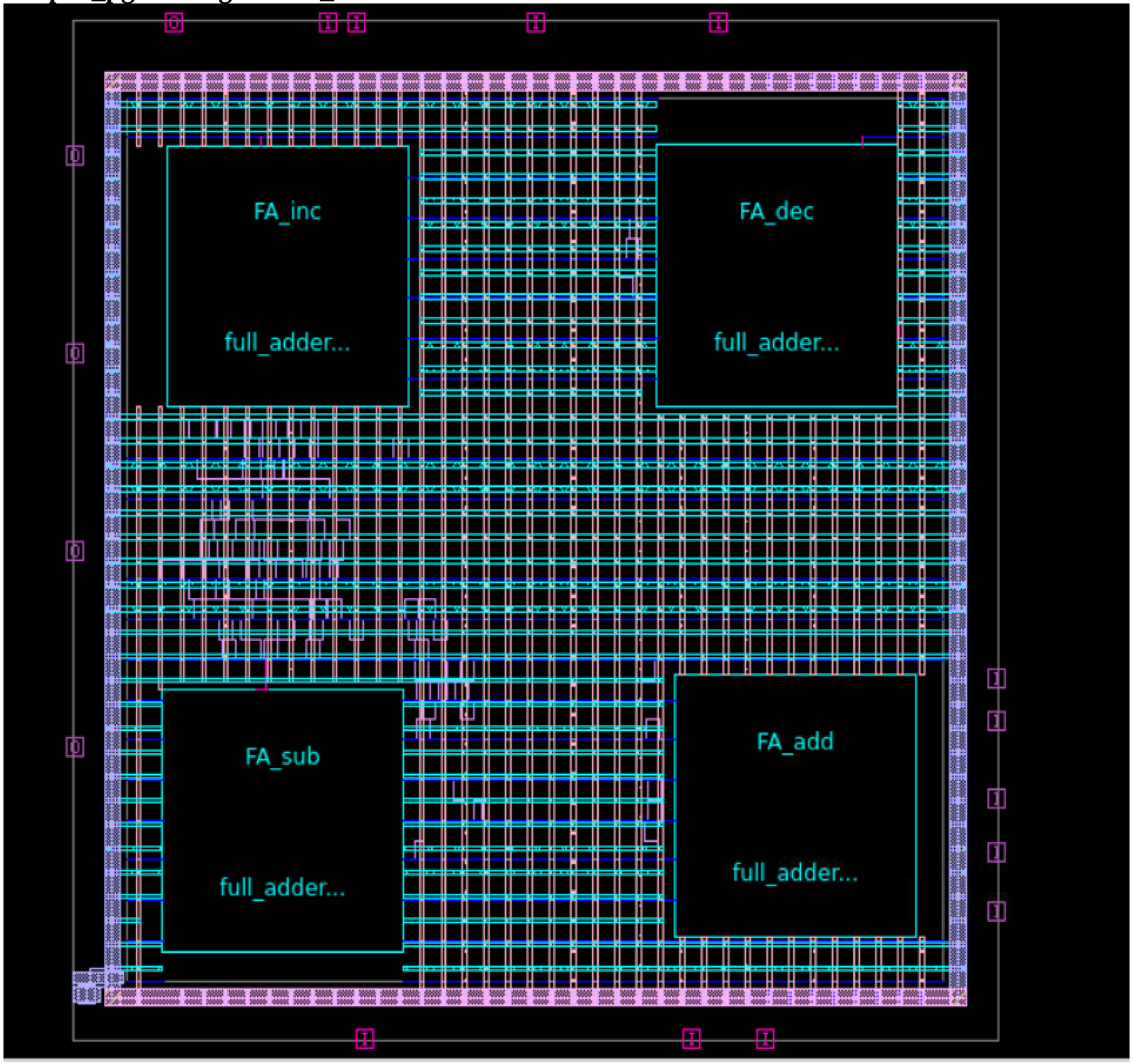
Core power ring


FA_inc
full_adder...
FA_dec
full_adder...
FA_sub
full_adder...
FA_add
full_adder...

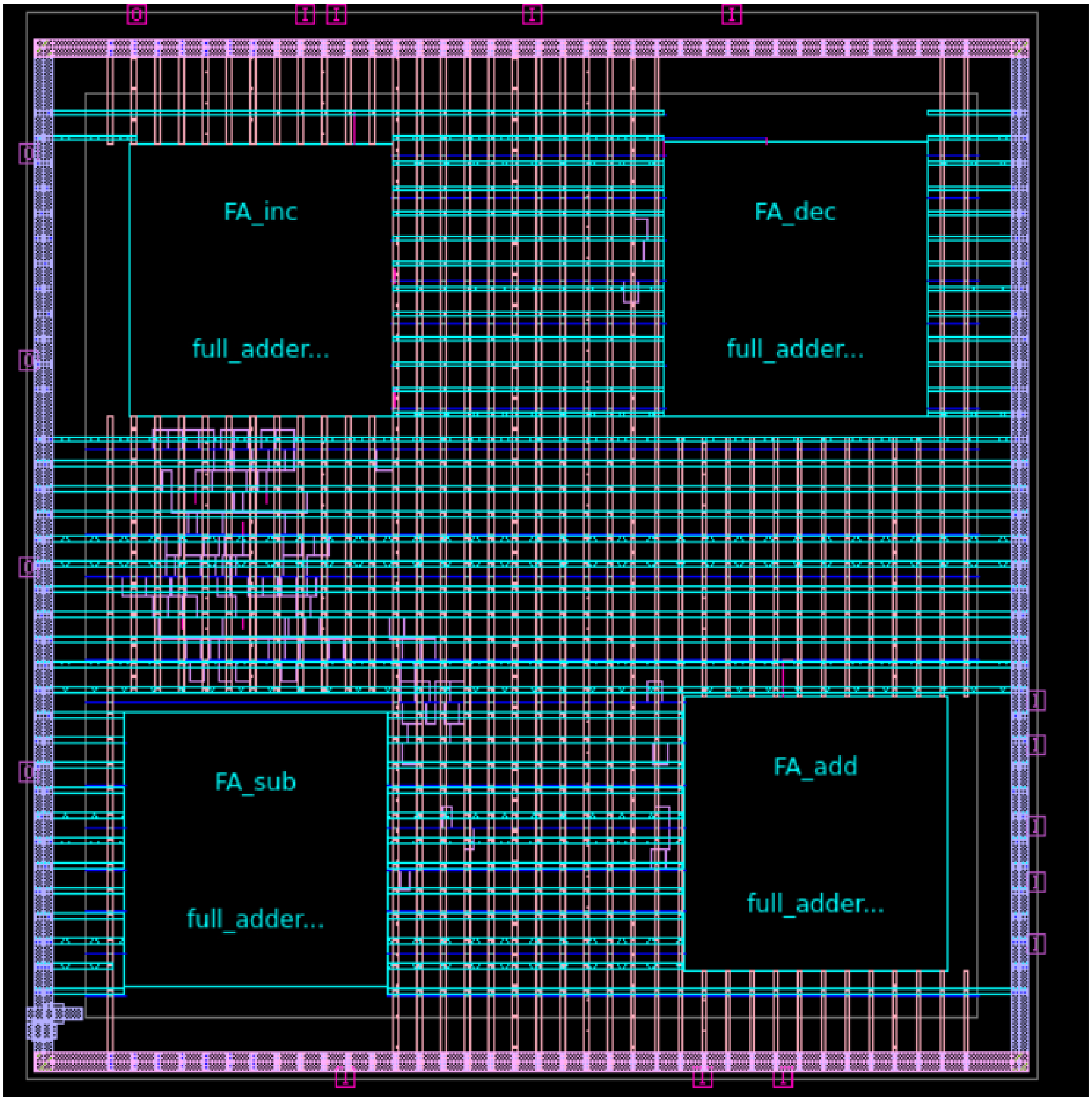
CORE_MESH

Individual meshes of VDD and VSS :

```
create_pg_mesh_pattern mesh -layers {{{vertical_layer: M6} {width: 0.4} {spacing:
interleaving} {pitch: 2} {offset: 1}} {{horizontal_layer: M7} {width: 0.4} {spacing:
interleaving} {pitch: 2} {offset: 1}}}
set_pg_strategy core_mesh -pattern {{name: mesh} {nets: VDD VSS}} -pg_regions
ALU_CORE -pattern { {pattern:mesh} {nets: VDD VSS}} -extension {{{side: 1234}
{direction: T B R L} {stop: innermost_ring}}}
compile_pg -strategies core_mesh
```

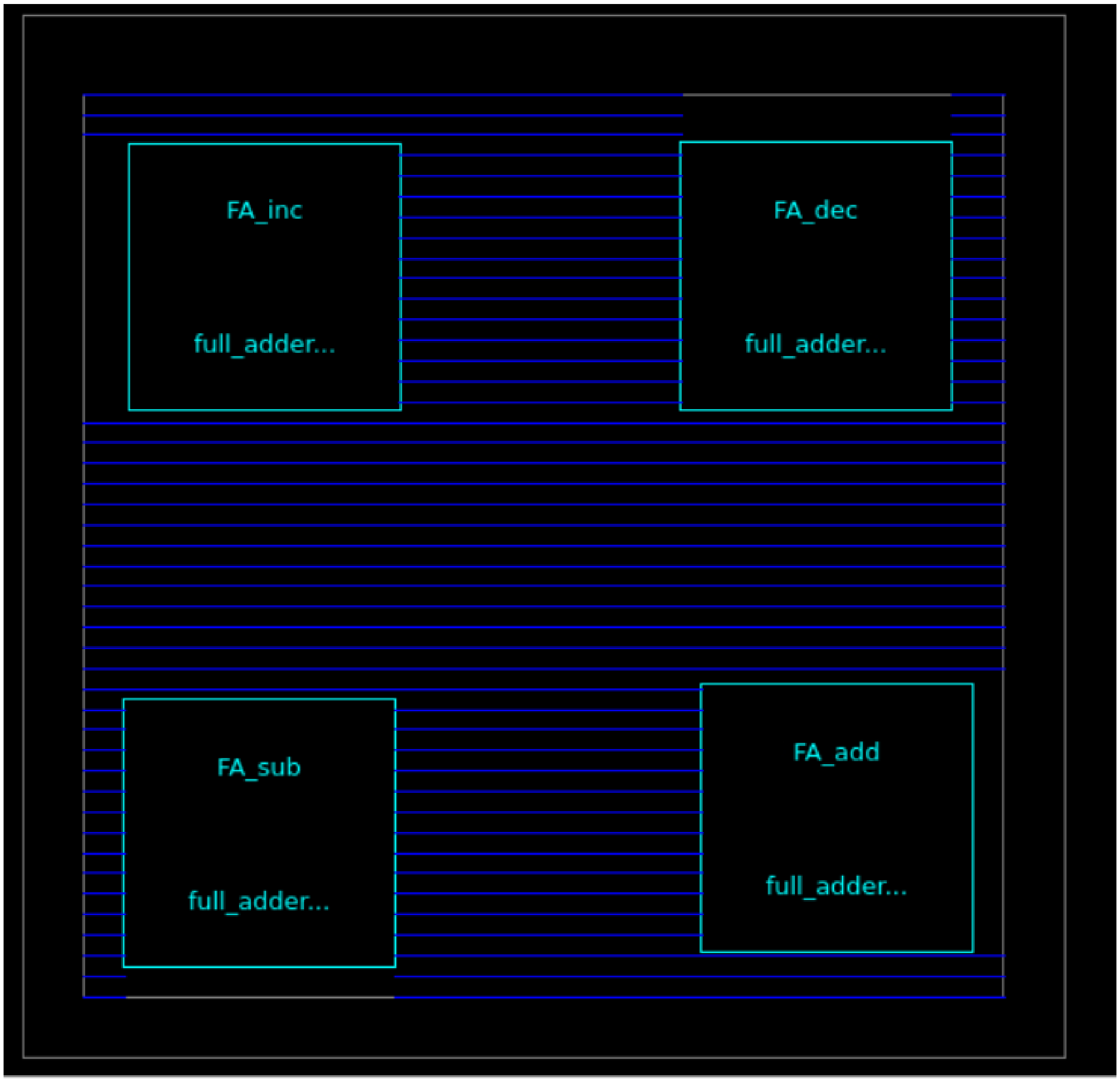


VSS MESH

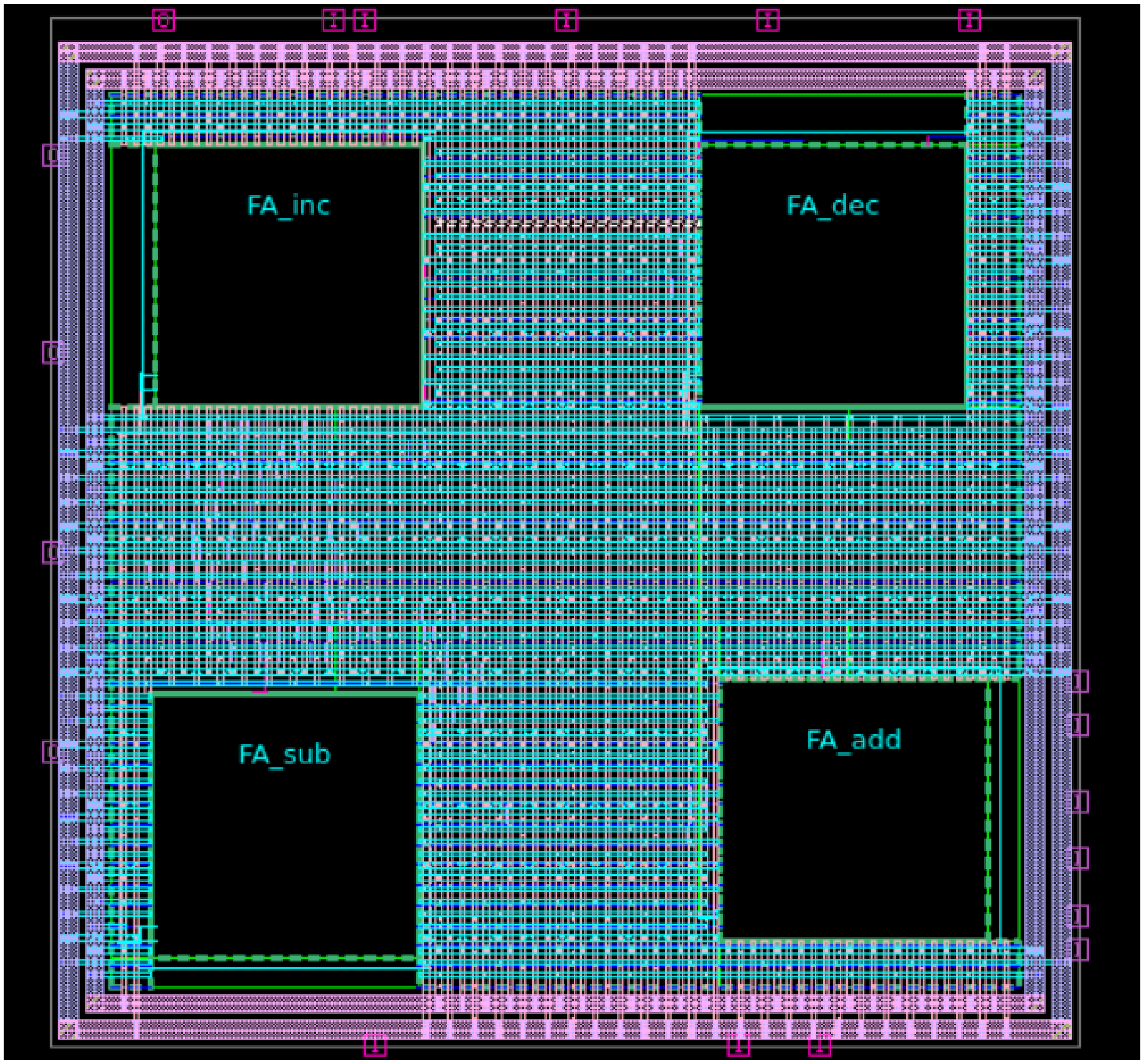

FA_inc
full_adder...
FA_dec
full_adder...
FA_sub
full_adder...
FA_add
full_adder...

VDD MESH

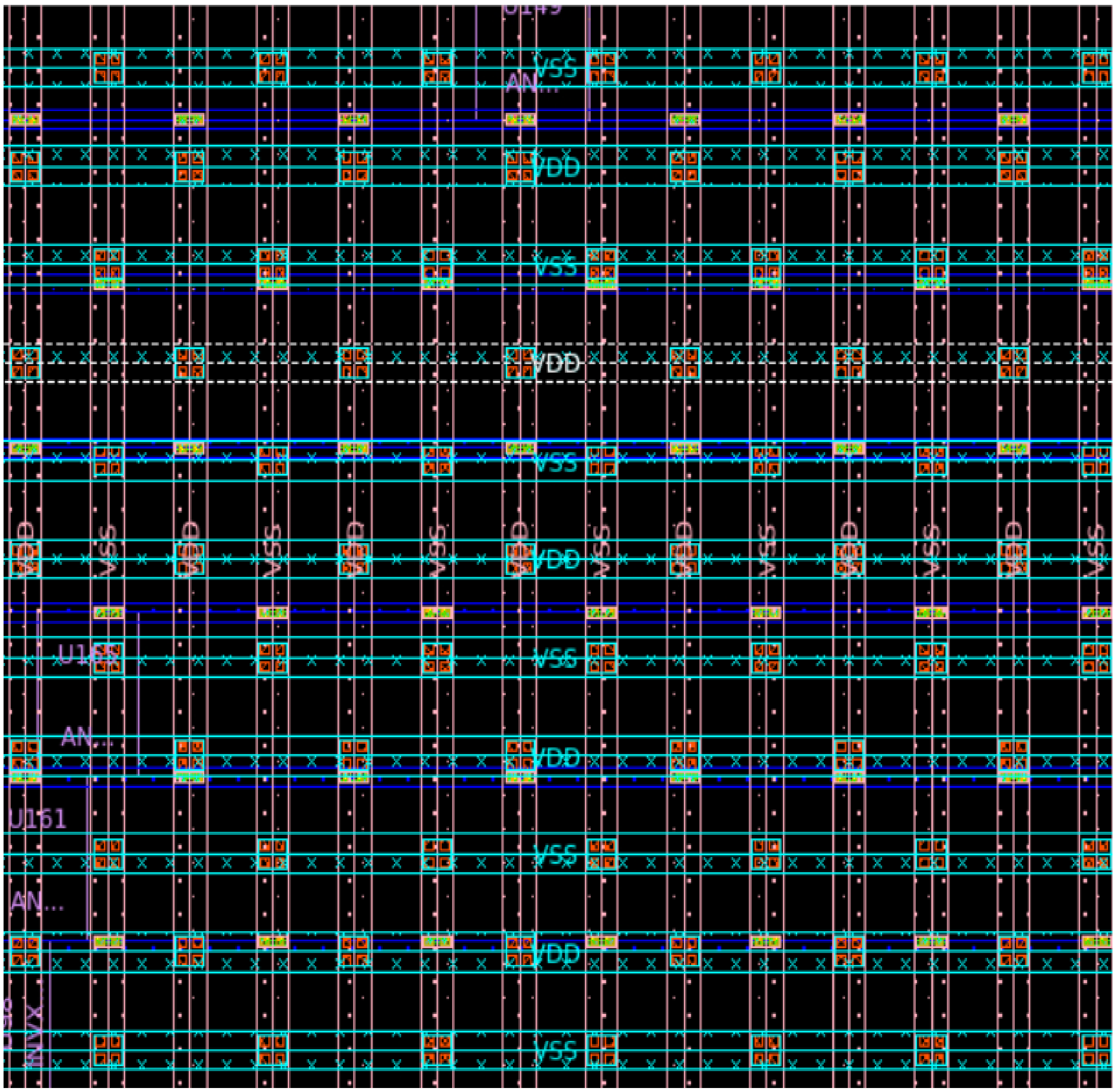
```
create_pg_std_cell_conn_pattern std_cell_rail -layers M1 -rail_width 0.19
set_pg_strategy rail_strategy -pg_regions ALU_CORE -pattern {{name: std_cell_rail} {nets:
VDD VSS}}
compile_pg -strategies rail_strategy
```



Standatd cell rails



Final Power Grid

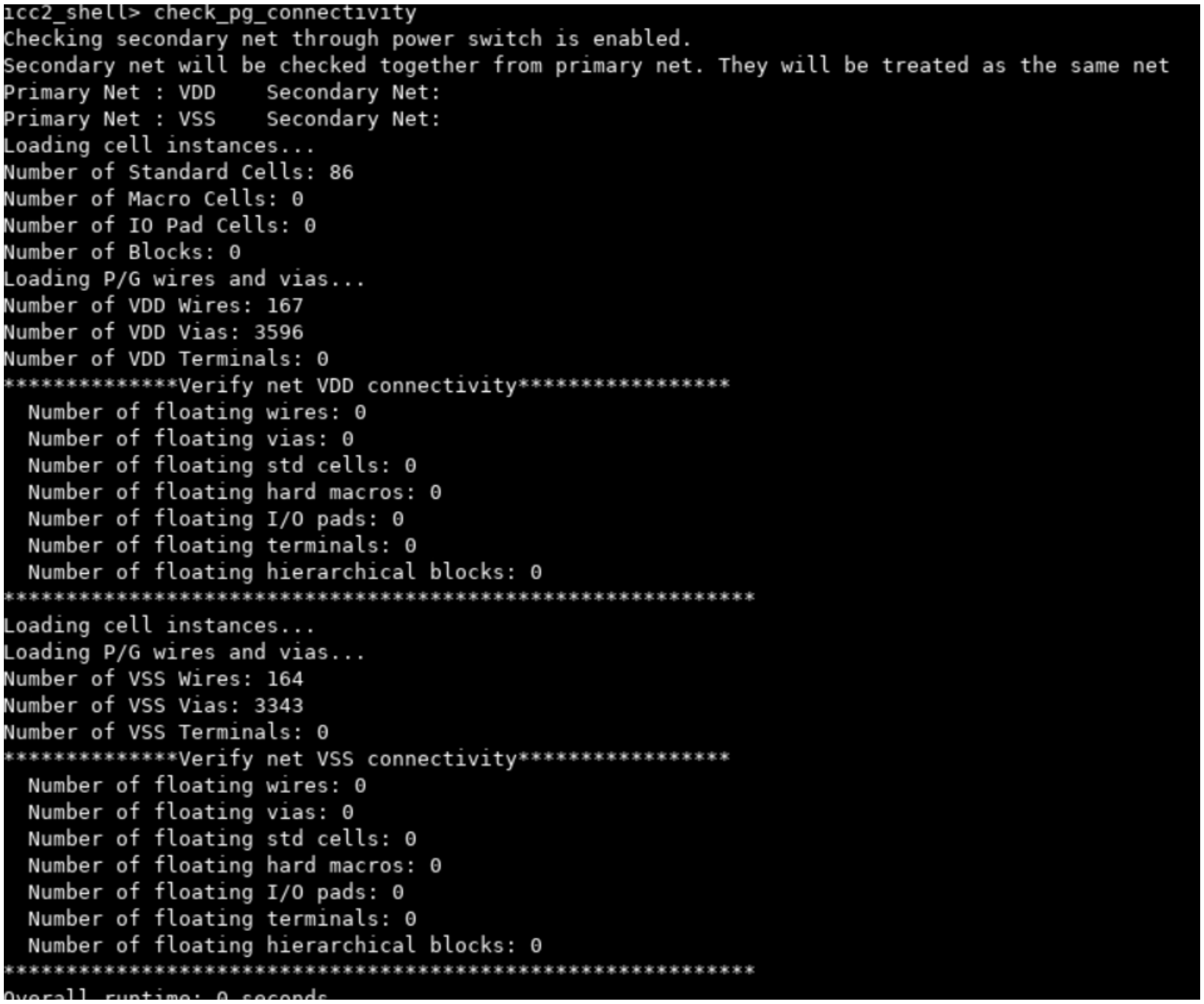


All the vias of mesh and standard cell rails are dropped

- **Power (VDD)** and **Ground (VSS)** ports and nets are created.
- Defines **power grid (PG) regions** for individual soft macros and the ALU core.
- A **ring-based power grid** is created around the core and compiled.
- A **mesh** pattern is applied over the ALU core for consistent power distribution.
- Standard cell rails are defined using metal layer M1 and compiled.

4. Power Grid Checks :

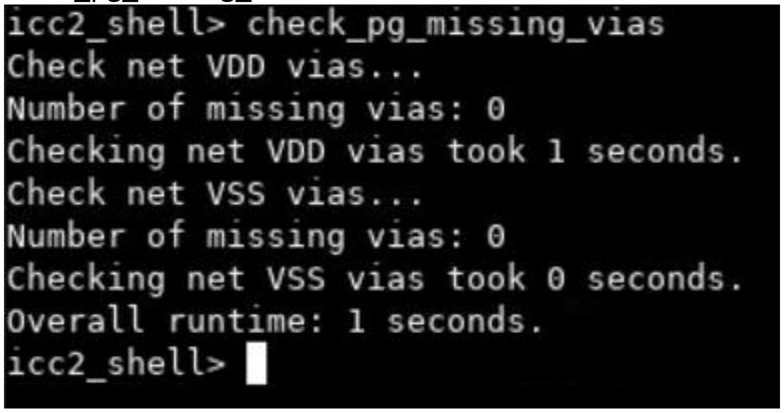
check_pg_connectivity



```
icc2_shell> check_pg_connectivity
Checking secondary net through power switch is enabled.
Secondary net will be checked together from primary net. They will be treated as the same net
Primary Net : VDD    Secondary Net:
Primary Net : VSS    Secondary Net:
Loading cell instances...
Number of Standard Cells: 86
Number of Macro Cells: 0
Number of IO Pad Cells: 0
Number of Blocks: 0
Loading P/G wires and vias...
Number of VDD Wires: 167
Number of VDD Vias: 3596
Number of VDD Terminals: 0
**************Verify net VDD connectivity*****************
  Number of floating wires: 0
  Number of floating vias: 0
  Number of floating std cells: 0
  Number of floating hard macros: 0
  Number of floating I/O pads: 0
  Number of floating terminals: 0
  Number of floating hierarchical blocks: 0
************************************************************
Loading cell instances...
Loading P/G wires and vias...
Number of VSS Wires: 164
Number of VSS Vias: 3343
Number of VSS Terminals: 0
**************Verify net VSS connectivity*****************
  Number of floating wires: 0
  Number of floating vias: 0
  Number of floating std cells: 0
  Number of floating hard macros: 0
  Number of floating I/O pads: 0
  Number of floating terminals: 0
  Number of floating hierarchical blocks: 0
************************************************************
Overall runtime: 0 seconds
```

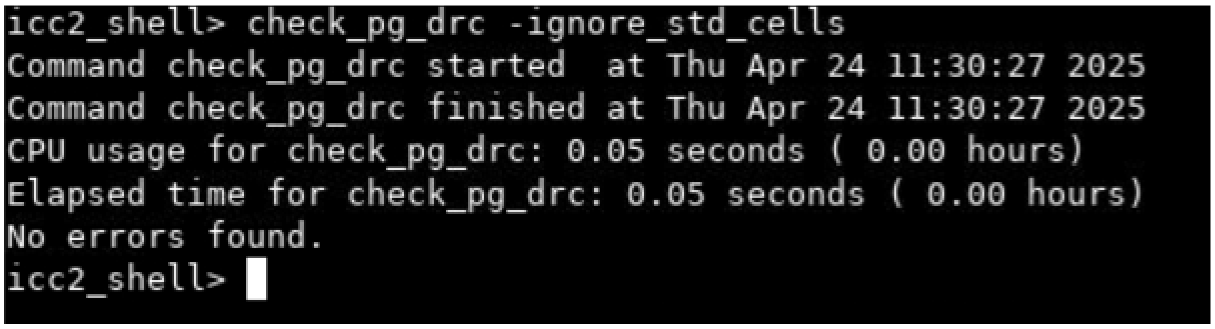
As we can see there are no floating standard cells , no floating hierarchal blocks , no floating terminals of VDD and VSS and no floating vias of VDD and VSS .

check_pg_missing_vias



```
icc2_shell> check_pg_missing_vias
Check net VDD vias...
Number of missing vias: 0
Checking net VDD vias took 1 seconds.
Check net VSS vias...
Number of missing vias: 0
Checking net VSS vias took 0 seconds.
Overall runtime: 1 seconds.
icc2_shell>
```

check_pg_drc



```
icc2_shell> check_pg_drc -ignore_std_cells
Command check_pg_drc started  at Thu Apr 24 11:30:27 2025
Command check_pg_drc finished at Thu Apr 24 11:30:27 2025
CPU usage for check_pg_drc: 0.05 seconds ( 0.00 hours)
Elapsed time for check_pg_drc: 0.05 seconds ( 0.00 hours)
No errors found.
icc2_shell>
```

As we can see the design is free from design rule violations and there are no missing vias in the power-plan stage.

- Ensures **power connectivity**, absence of **missing vias**, and no **design rule violations** in the PG network.

- Reports generated help validate proper power distribution.

5. Constraints and Scenario Setup

- Reads **timing constraints** from the `.sdc` file.
- Defines a **functional mode** and a **slow process corner** to simulate worst-case timing.
- Creates a **timing scenario** and links it with parasitic information (`.tluplus`) for accurate delay estimation.

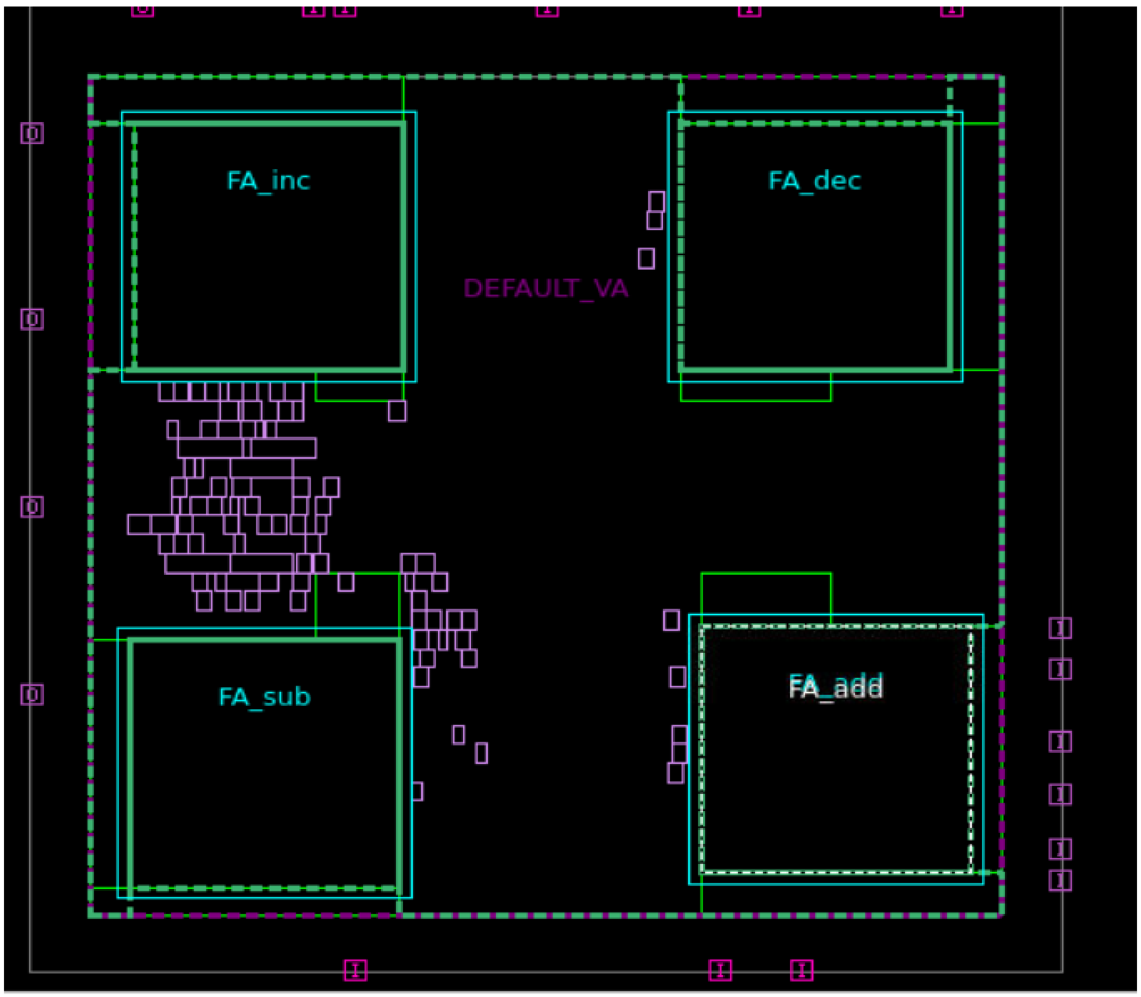
check_design -checks pre_placement_stage
check_design -checks design_mismatch

```
read_sdc ../../ALU_HIER/DC/Hybrid_alu.sdc


set_app_options -name place.coarse.continue_on_missing_scandef -value true

set mode1 "func"

set corner1 "slow"

set scenario1 "${mode1}_${corner1}"

create_mode $mode1

create_corner $corner1

create_scenario -name $scenario1 -mode $mode1 -corner $corner1

set parasitics "p1"

read_parasitic_tech -tlup
"/data/pdk/pdk32nm/SAED32_EDK/tech/star_rcxt/saed32nm_1p9m_Cmax.tluplus" -layermap
"/data/pdk/pdk32nm/SAED32_EDK/tech/star_rcxt/saed32nm_tf_itf_tluplus.map" -name p1

set_parasitic_parameters -late_spec $parasitics -early_spec $parasitics
```

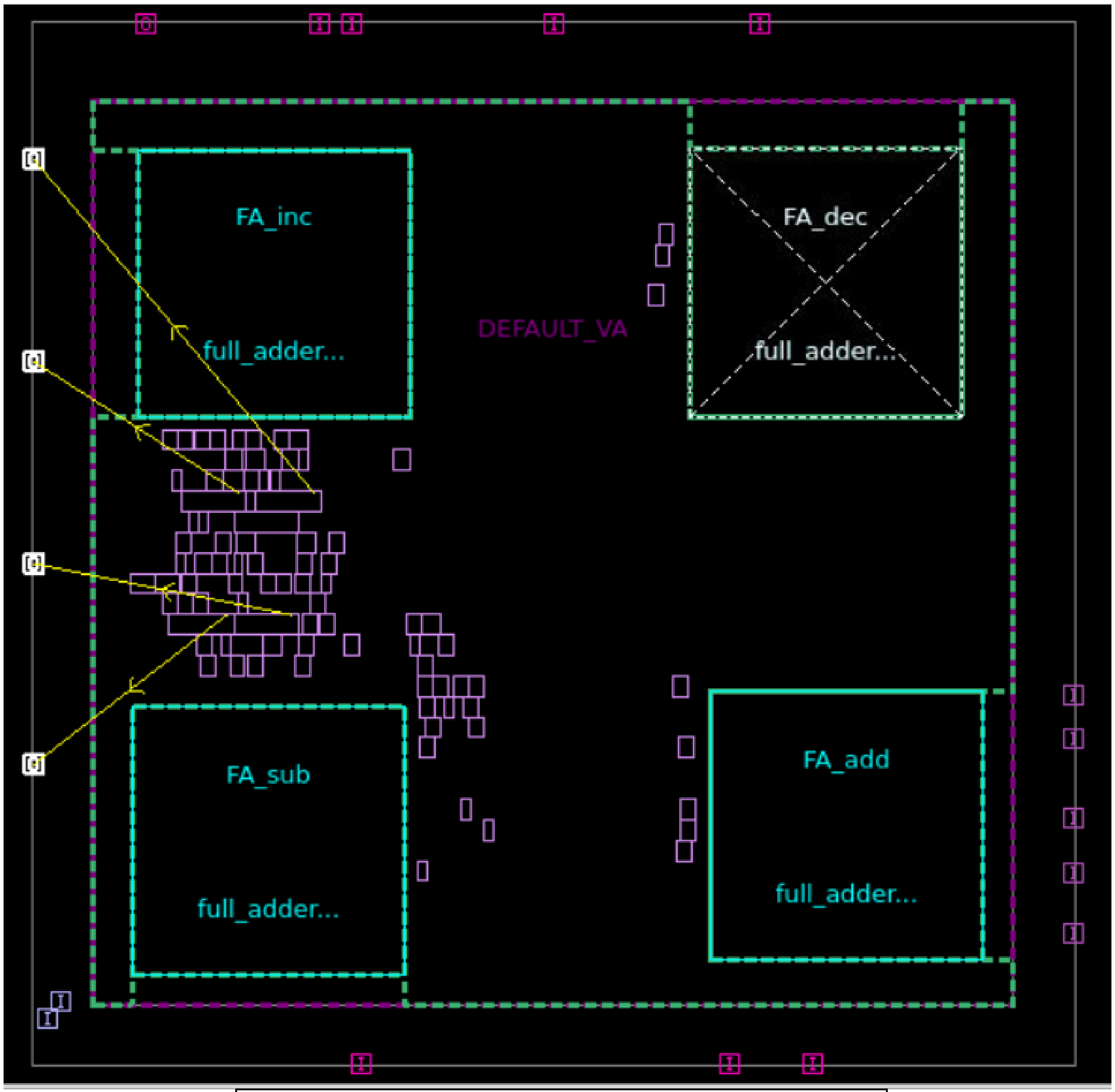
6. Placement and Optimization

```
create_placement -effort high

set_scenario_status func_slow -hold true -setup true -leakage_power true -max_capacitance true –
min_capacitance true -dynamic_power true -max_transition true -active true

place_opt

legalize_placement
```

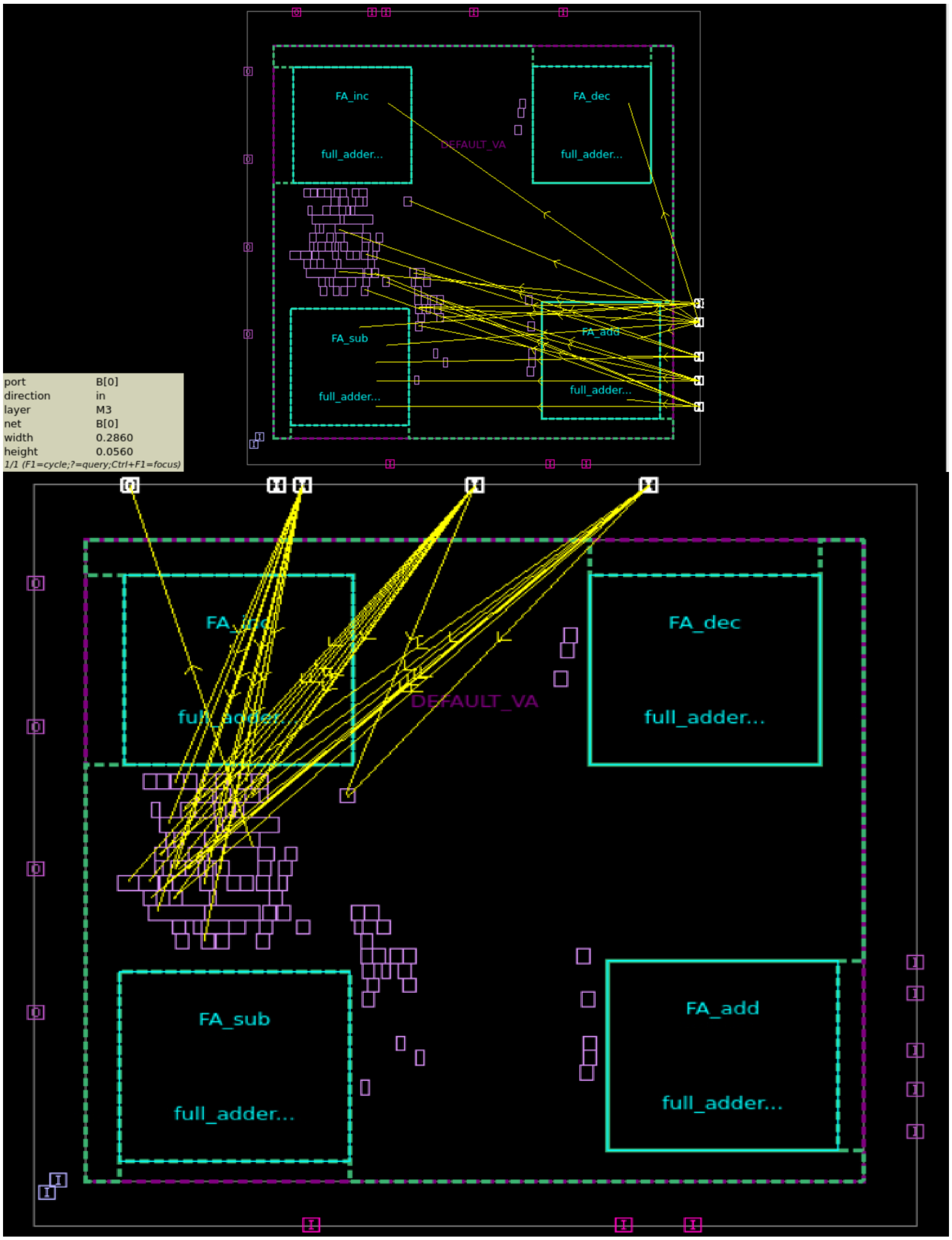
- The initial placement is done with high effort, followed by **placement optimization** to reduce congestion and timing violations.
- **Legalization** ensures all cells are aligned to placement rows and not overlapping.



Placement of standard cells

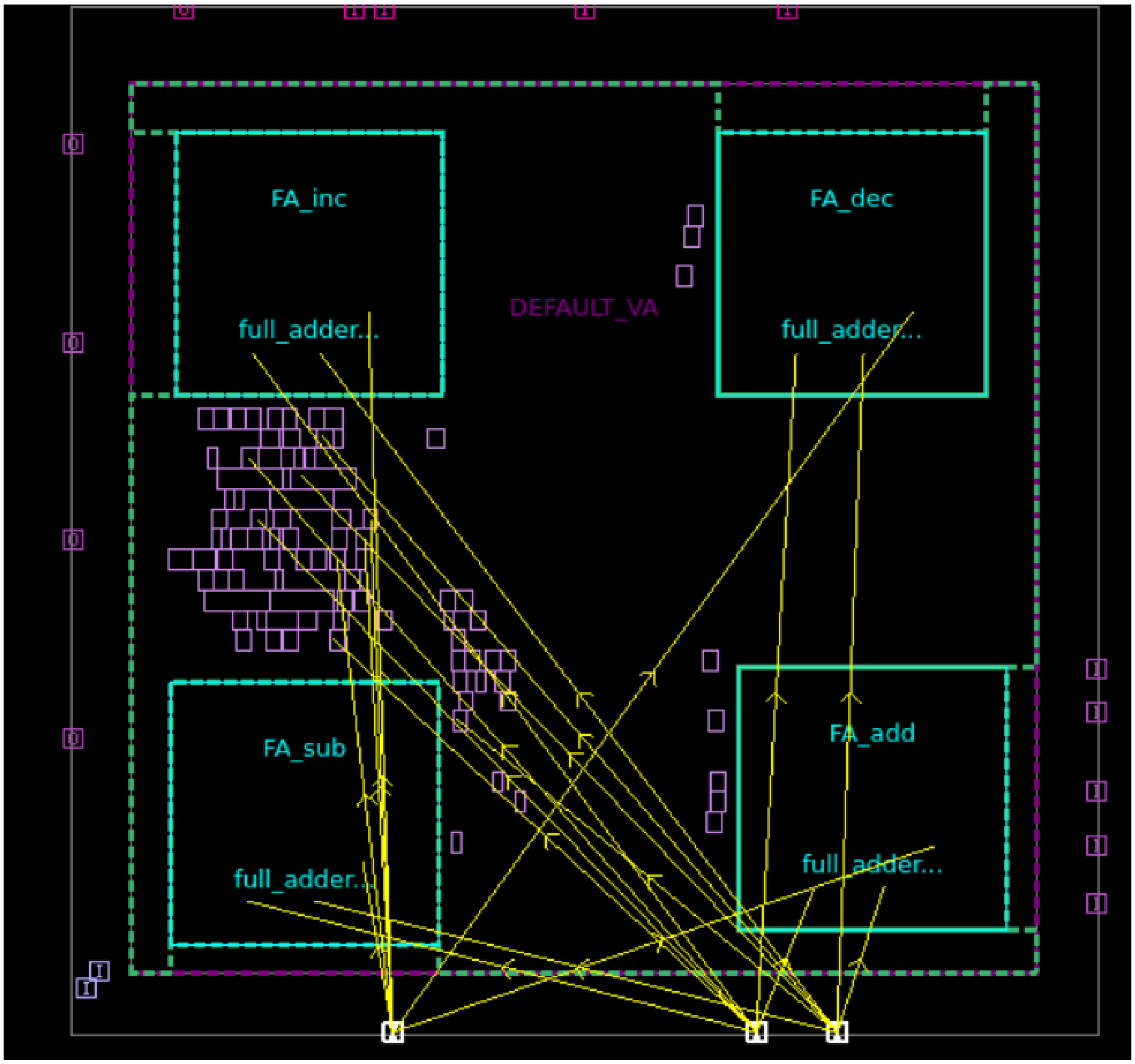


Output ports flyline connections

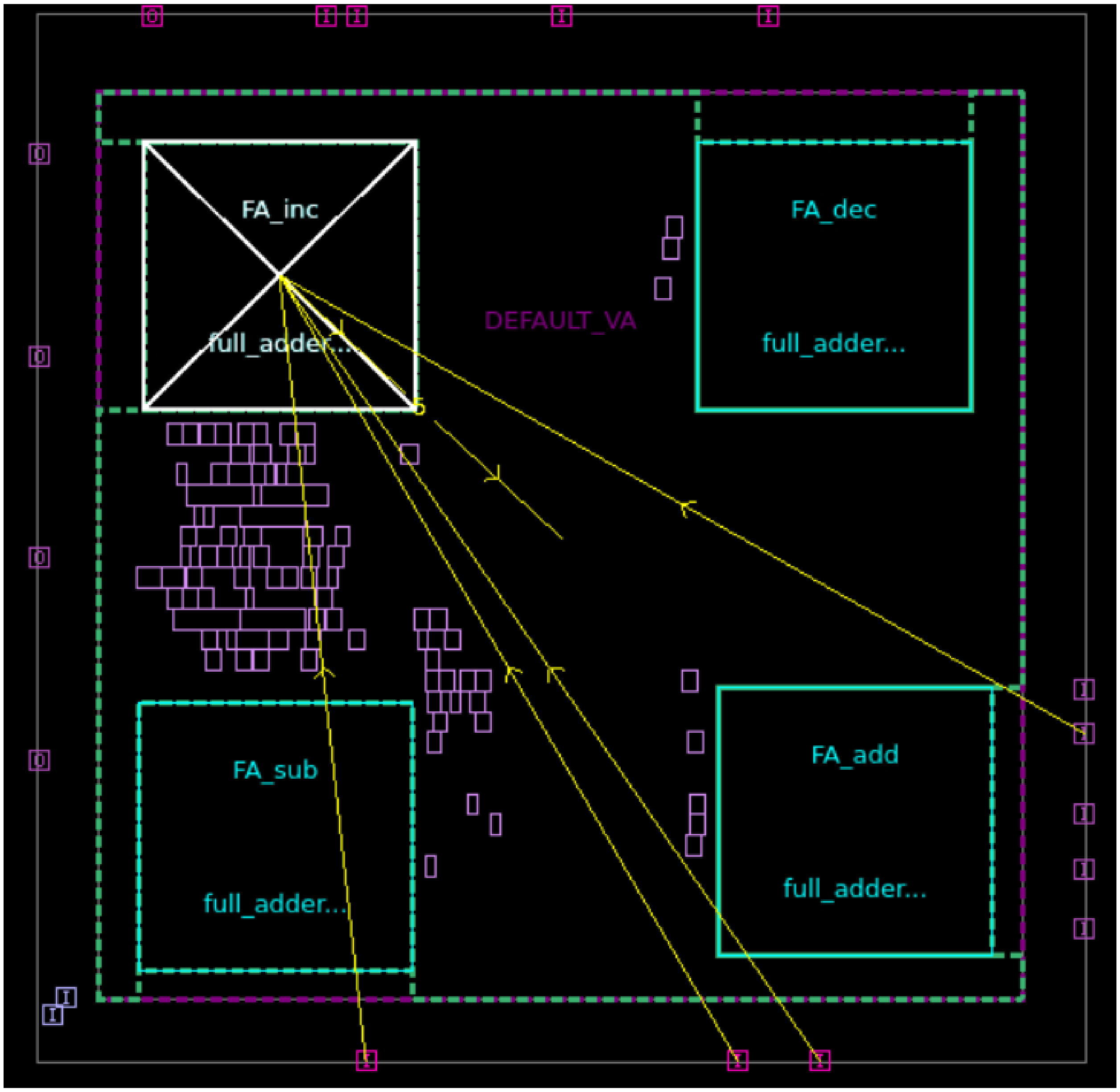


```
port        B[0]
direction   in
layer       M3
net         B[0]
width       0.2860
height      0.0560
1/1 (F1=cycle;?=query;Ctrl+F1=focus)
```

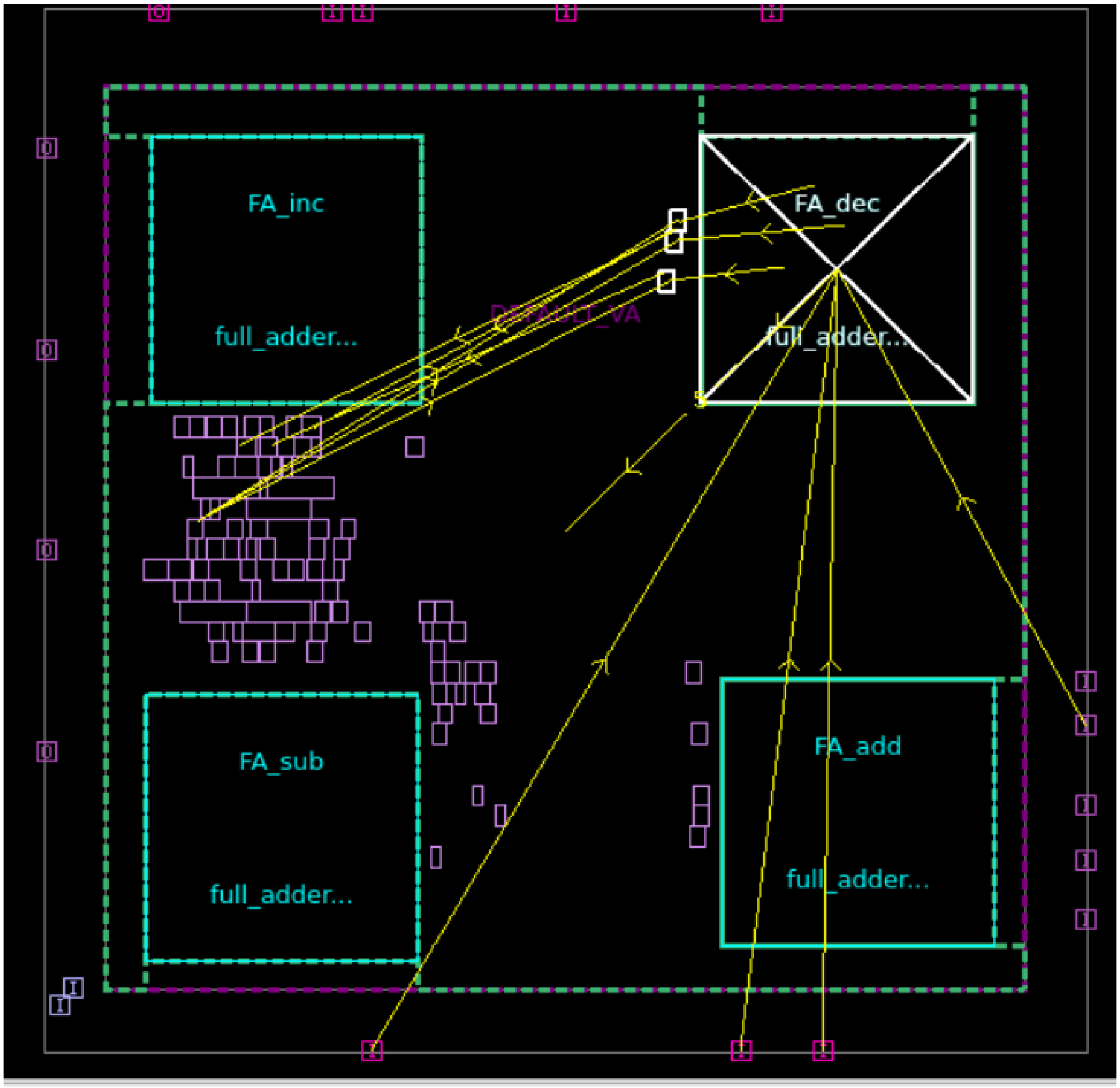
flyline connections of ports on side 2



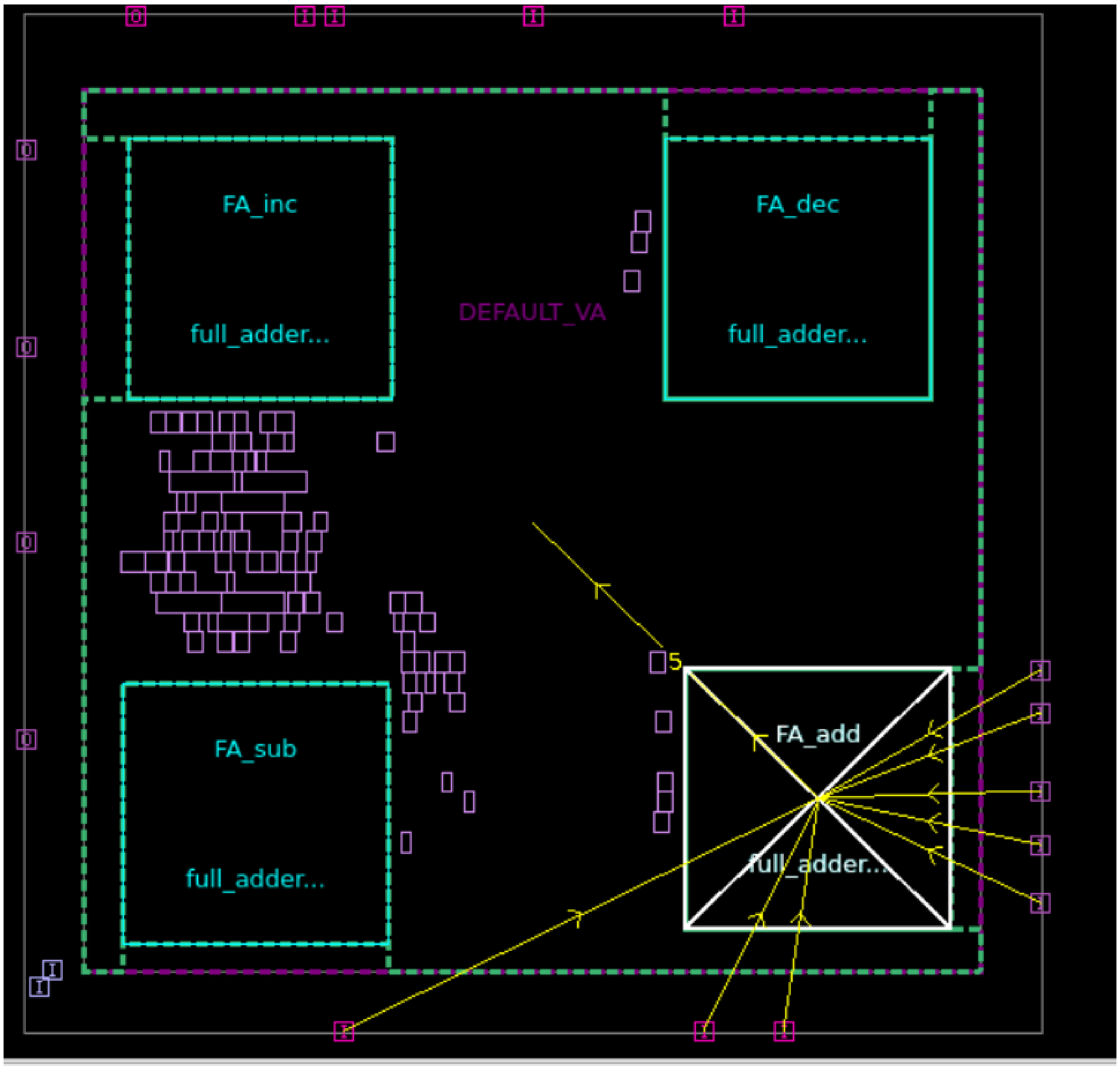
flyline connections of ports on side 4



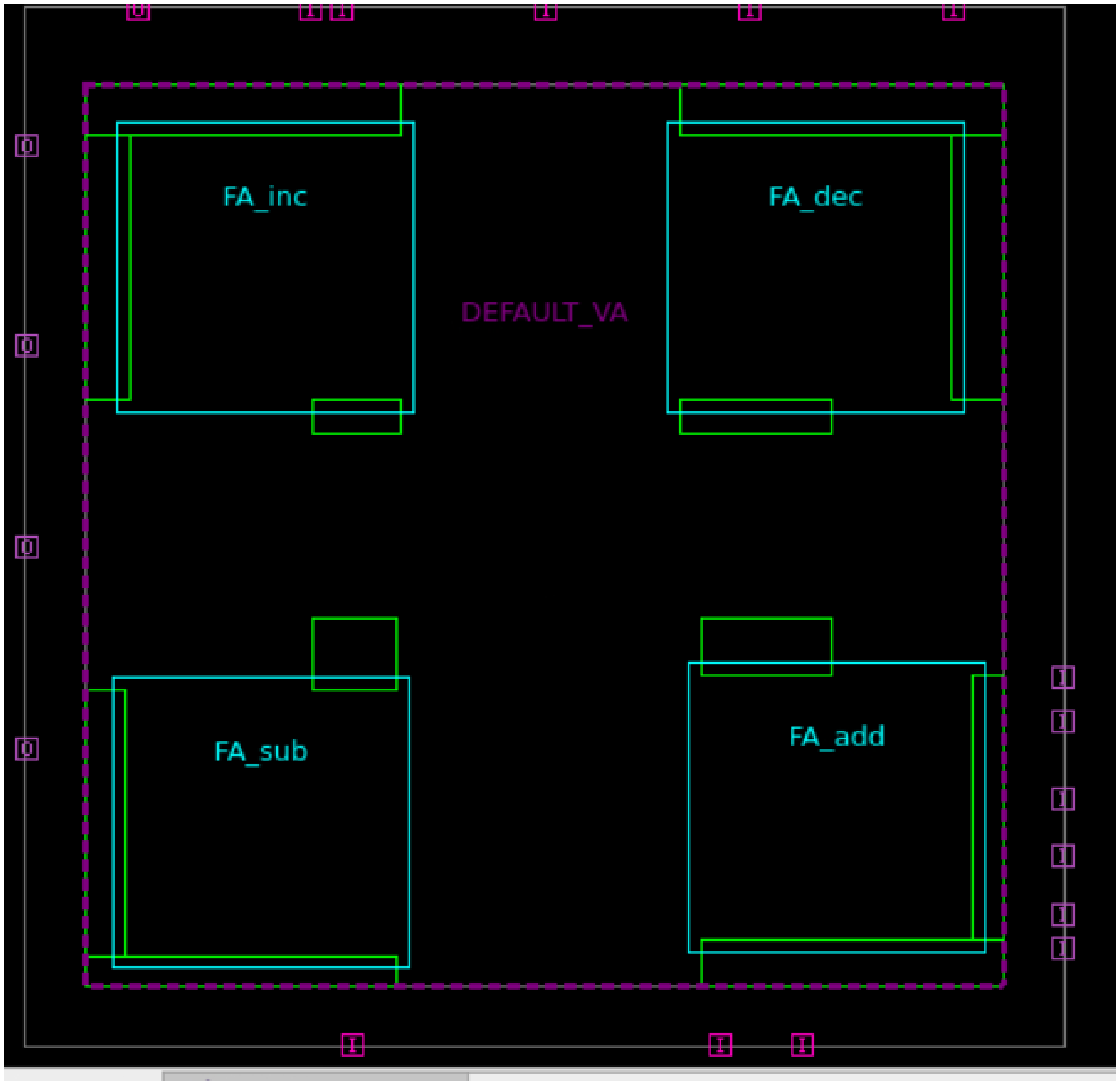
Soft Macro FA_inc (incrementing using full adder block) flyline connections



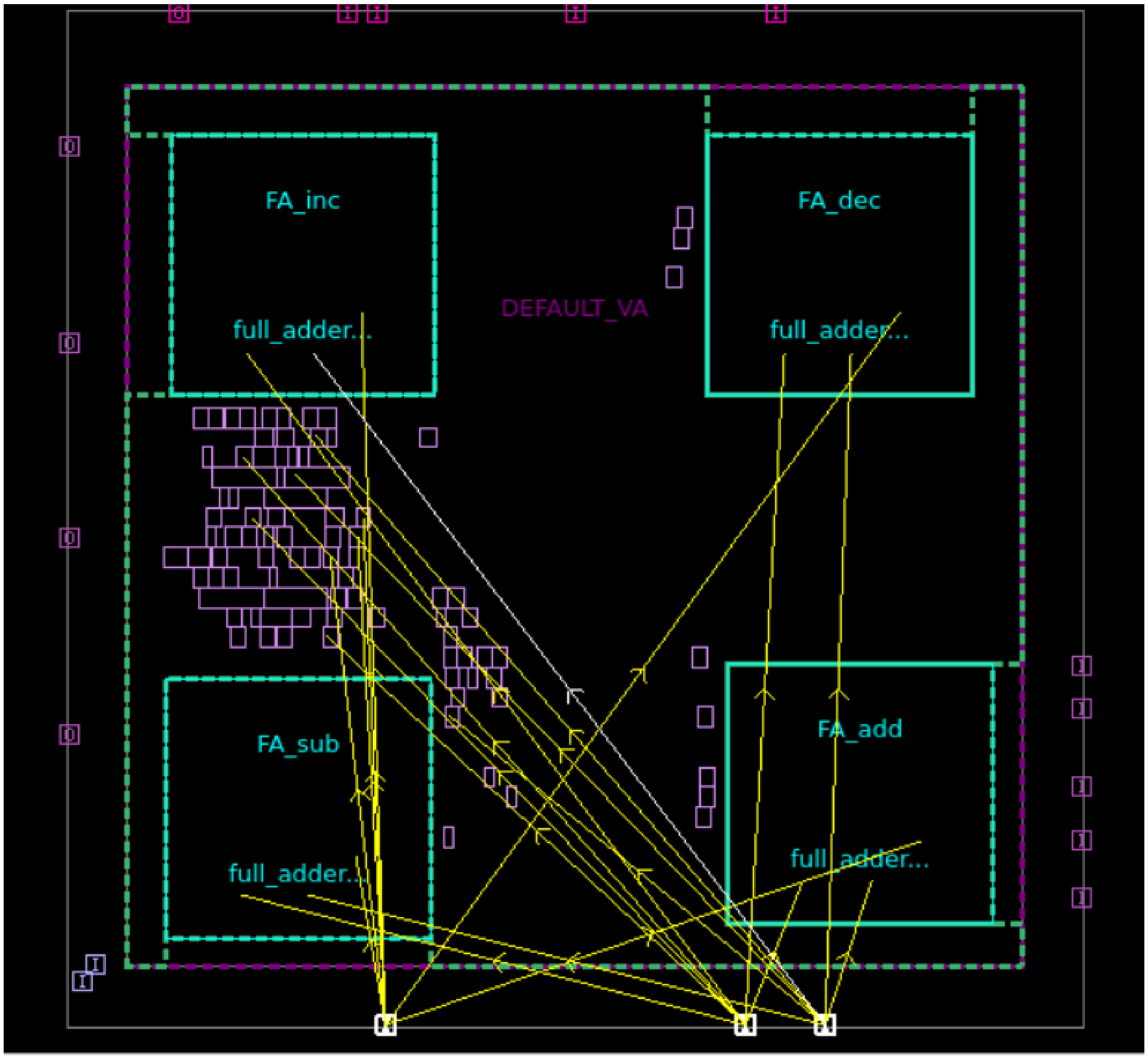
Soft Macro FA_dec (decrementing using full adder block) flyline connections



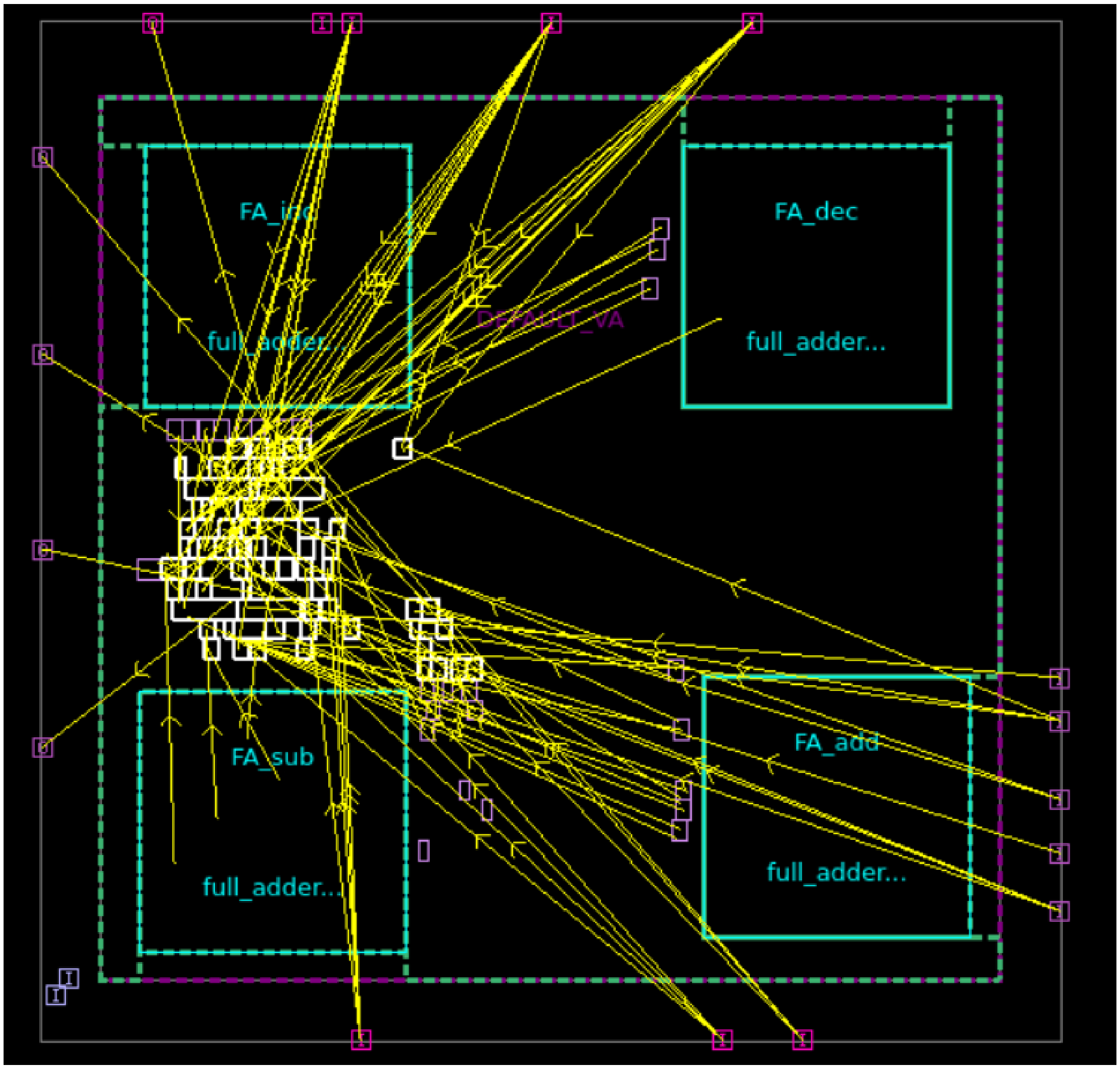
Soft Macro FA_add (adder logic using full adder block) flyline connections



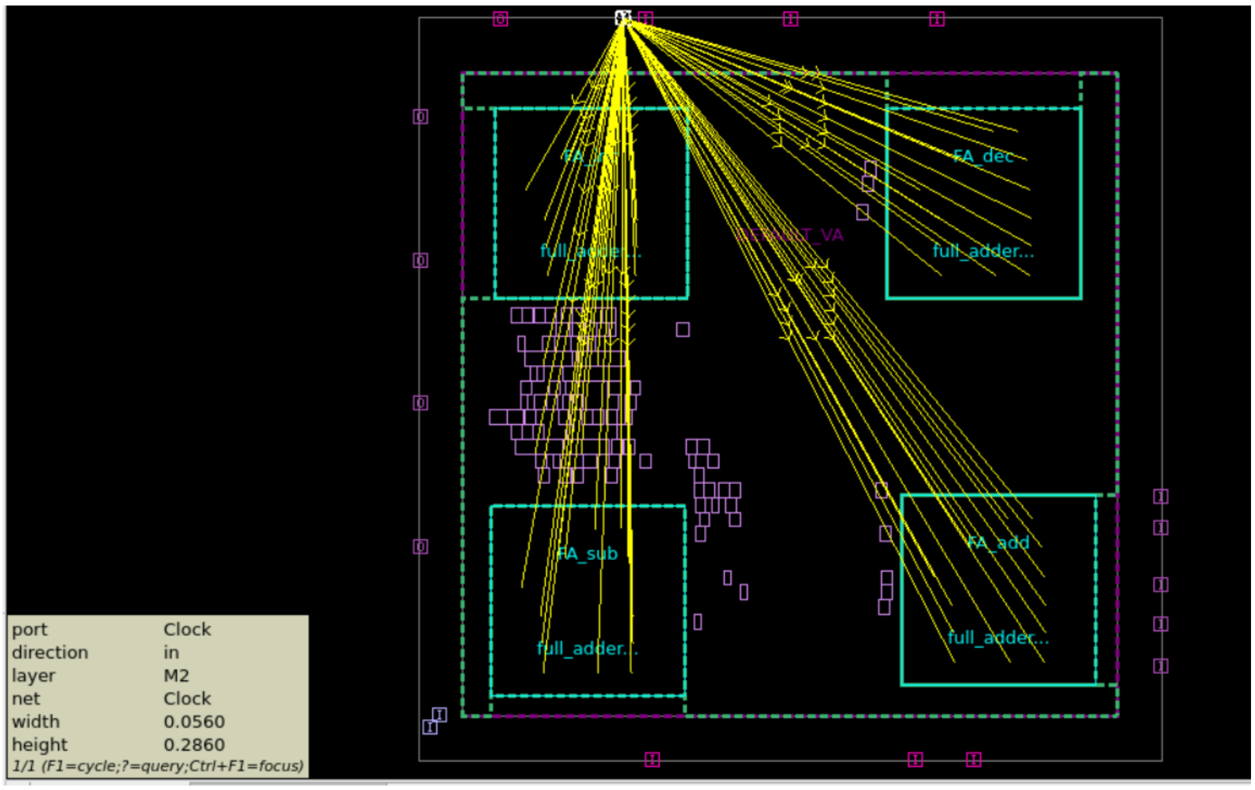
Placing Soft Blockage around the macros .



Standard cell connections to macros and ports after place_opt



Flylines connectios of few standard cells



port	Clock
direction	in
layer	M2
net	Clock
width	0.0560
height	0.2860

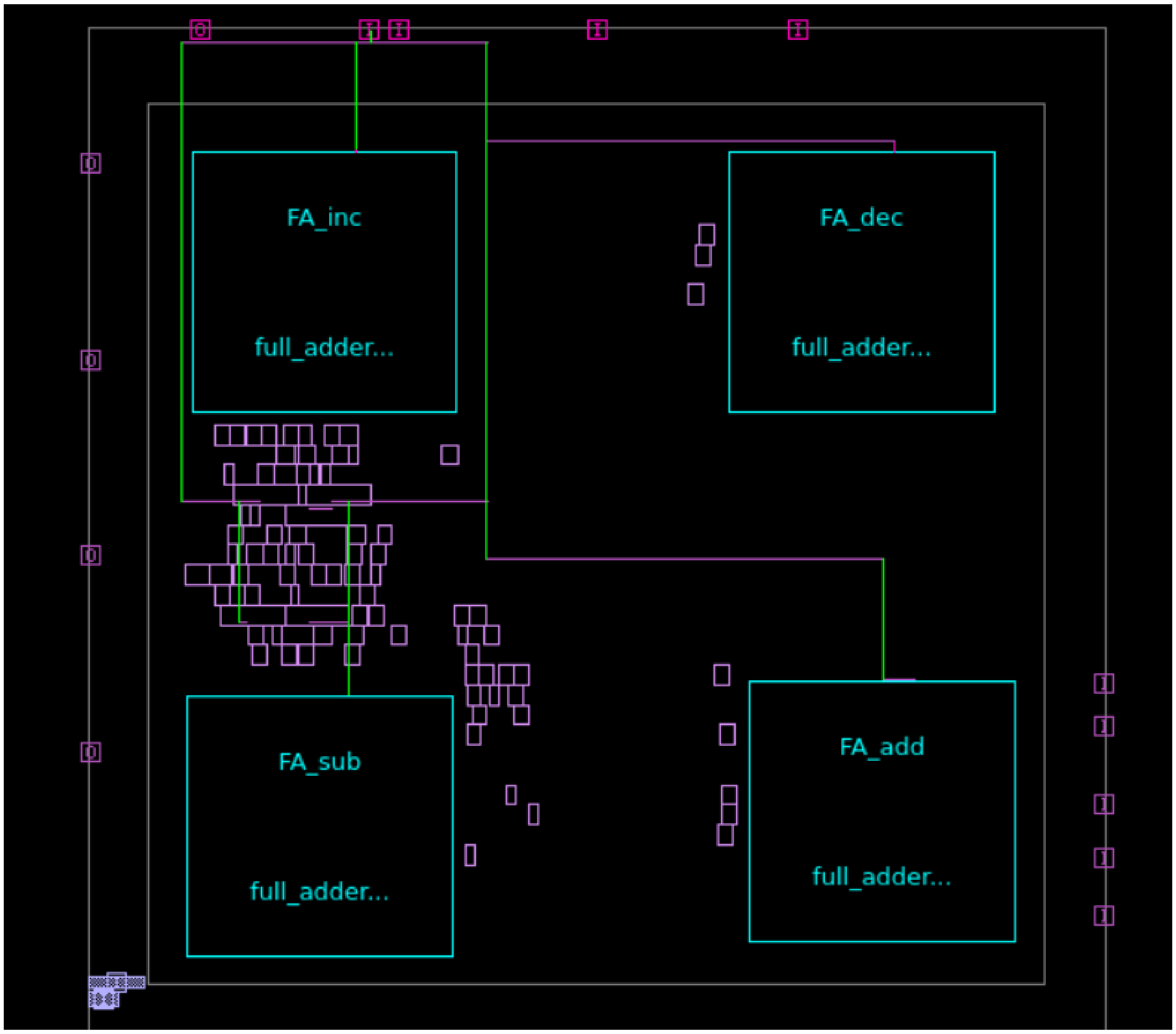
1/1 (F1=cycle;?=query;Ctrl+F1=focus)

Clock port to macro connections flylines before clock tree stage.

7. Clock Tree Synthesis (CTS)

- Synthesizes the **clock tree** for the design, balancing delays and minimizing skew.
- Optimizes the tree to meet **timing, power, and noise constraints**.

```
set_clock_routing_rules -default_ndr -clock Clocks -min_layer M3 -max_layer M4

set_scenario_status func_slow -hold true -setup true -leakage_power true -max_capacitance true -
min_capacitance true -dynamic_power true -max_transition true -active true

synthesize_clock_trees

clock_opt
```



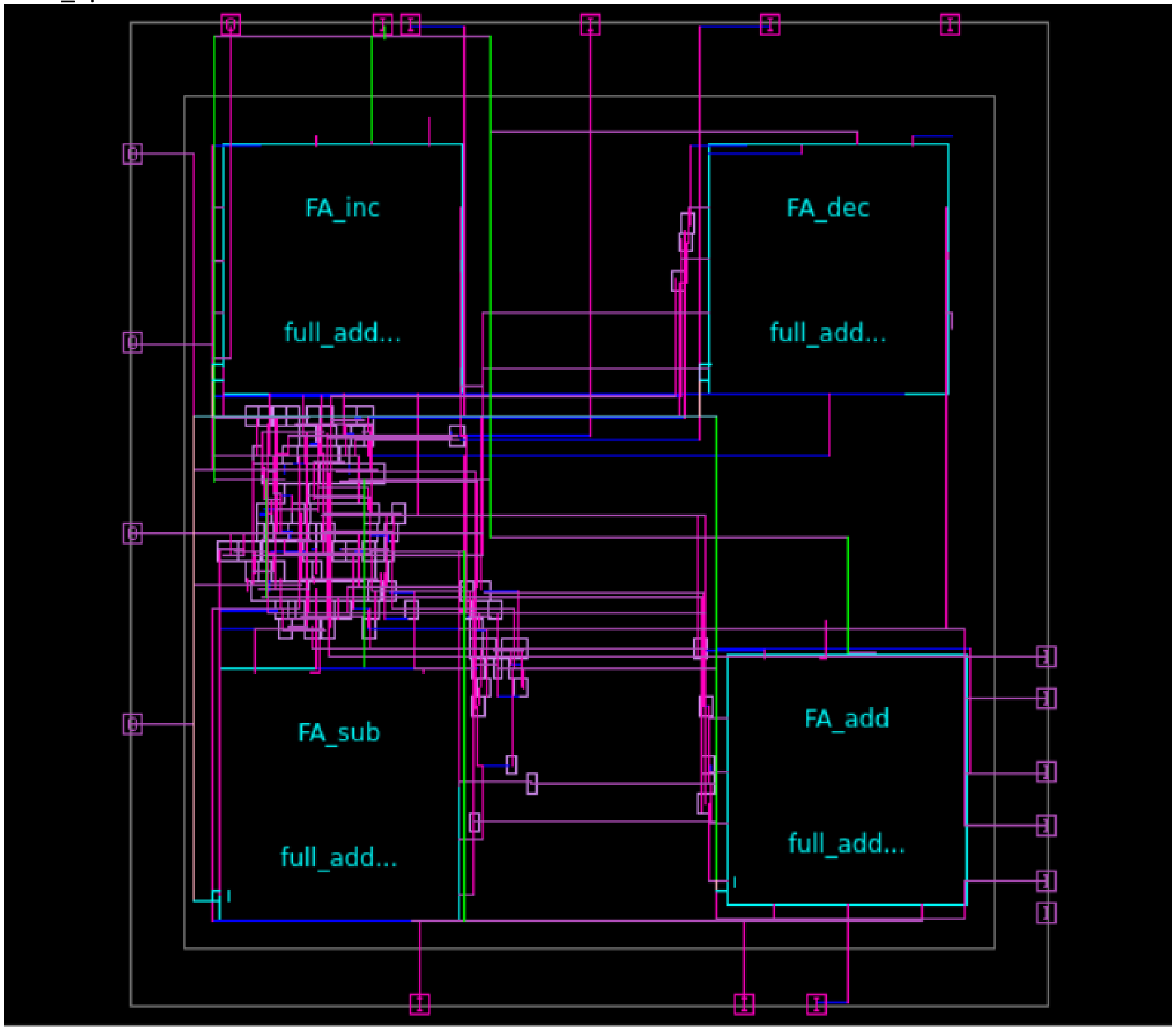
Clock Tree Synthesis

8. Routing

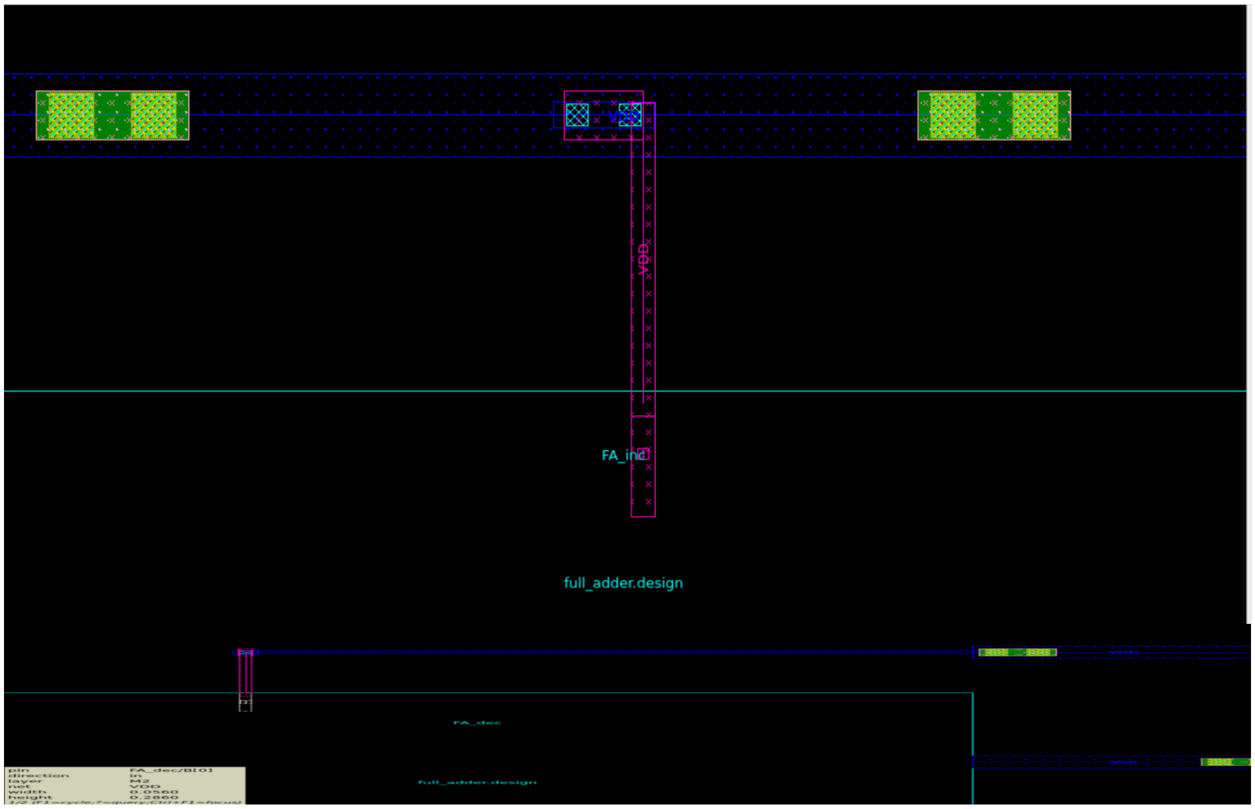
- Performs **global**, **track**, and **detailed routing** while being **timing- and crosstalk-driven**.
- Applies **antenna fixing** and uses **diodes** where required.

Commands:

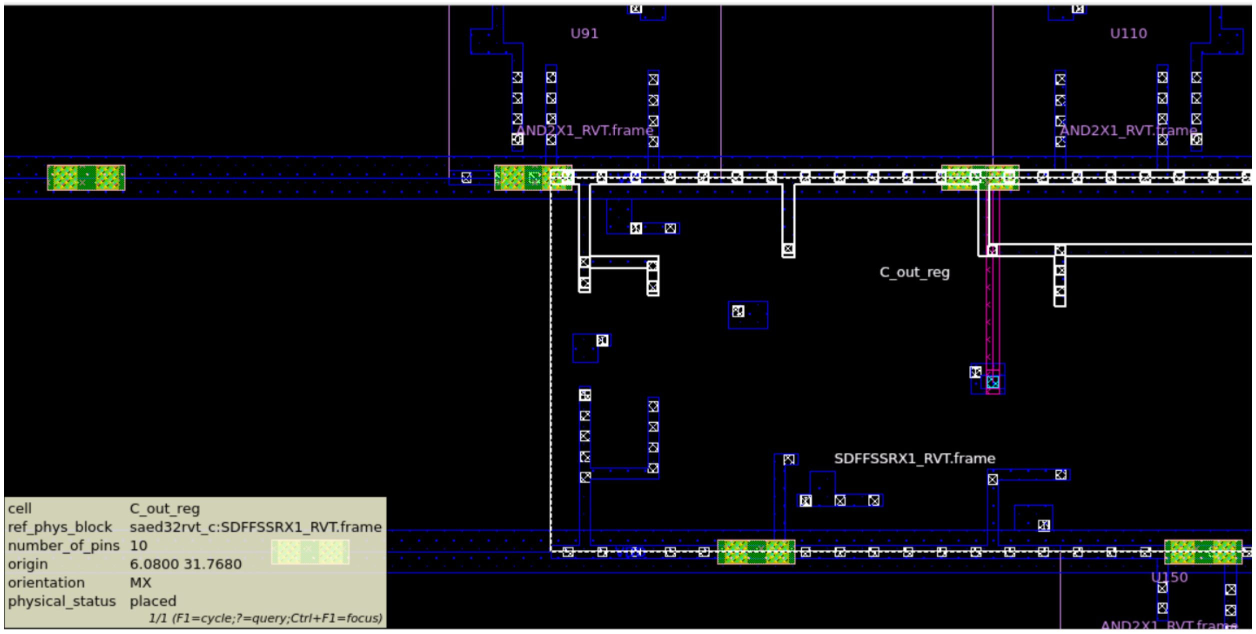
```
check_design -checks pre_route_stage
set_app_options -name route.global.timing_driven -value true
set_app_options -name route.global.crosstalk_driven -value true
set_app_options -name route.track.timing_driven -value true
set_app_options -name route.track.crosstalk_driven -value true
set_app_options -name route.detail.timing_driven -value true
set_app_options -name route.detail.antenna -value true
set_app_options -name route.detail.antenna_fixing_preference -value use_diodes
set_app_options -block [current_design ] -name route.detail.diode_libcell_names -value
{*/ANTENNA_RVT}
route_global
route_track
route_detail
set_app_options -name route_opt.flow.enable_ccd -value true
route_opt
```



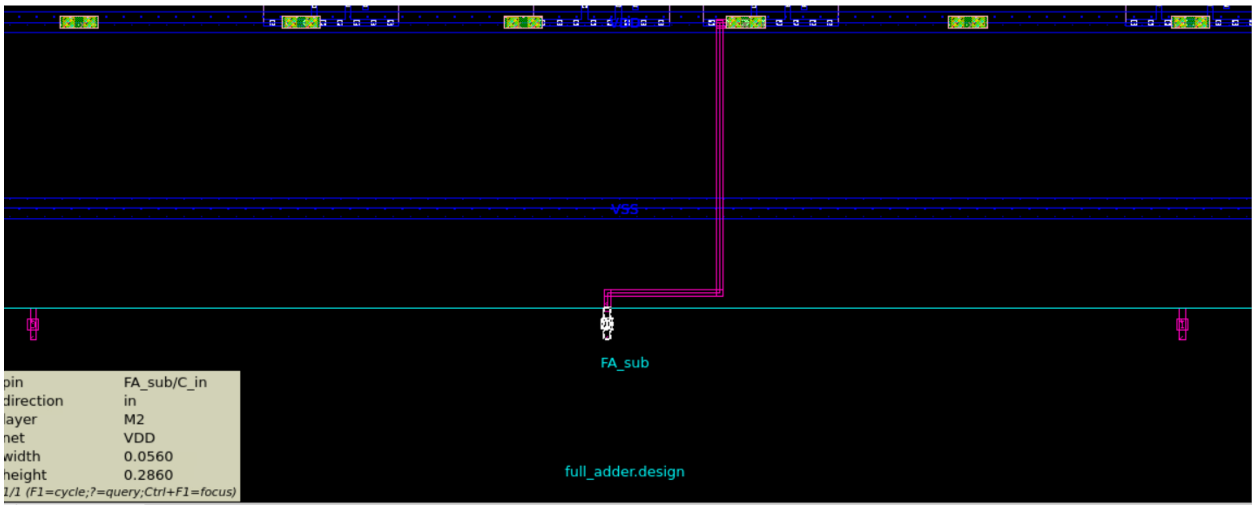
Signal and clock routing



```
pin         FA_dec/B[0]
direction   in
layer       M2
net         VDD
width       0.0560
height      0.2860
1/2 (F1=cycle;?=query;Ctrl+F1=focus)
```



```
cell              C_out_reg
ref_phys_block    saed32rvt_c:SDFFSSRX1_RVT.frame
number_of_pins    10
origin            6.0800 31.7680
orientation       MX
physical_status   placed
        1/1 (F1=cycle;?=query;Ctrl+F1=focus)
```



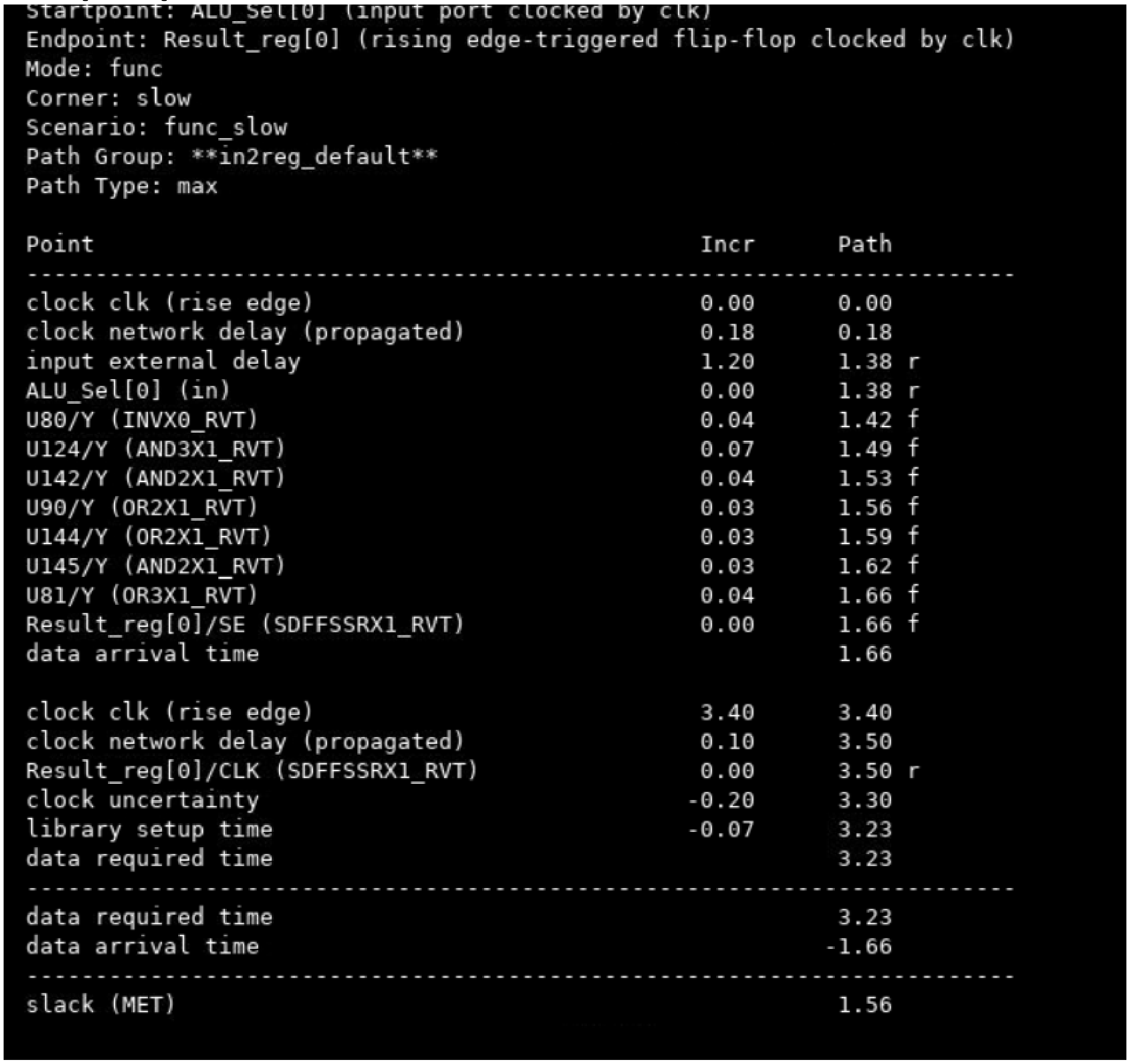
```
pin          FA_sub/C_in
direction    in
layer        M2
net          VDD
width        0.0560
height       0.2860
1/1 (F1=cycle;?=query;Ctrl+F1=focus)
```

As we can see the VDD and VSS pins of all the macros and standard cells are being connected to the standad cell rail for power.

9. Post-Route Analysis

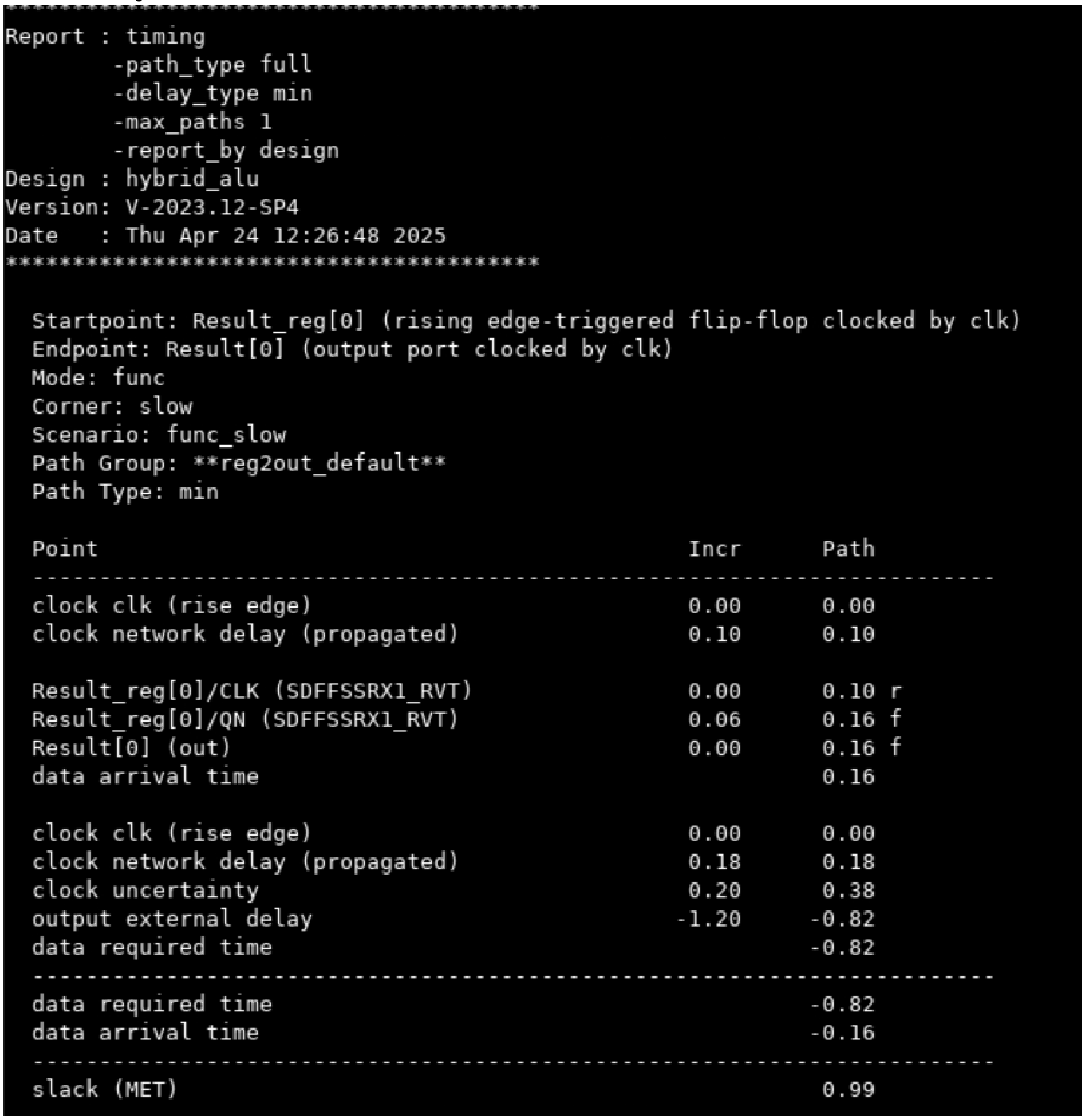
- Generates **timing reports** (both setup and hold) post-routing.
- Saves the design database, outputs **final netlist** and **LEF**, useful for backend integration or chip assembly.

Setup Report:

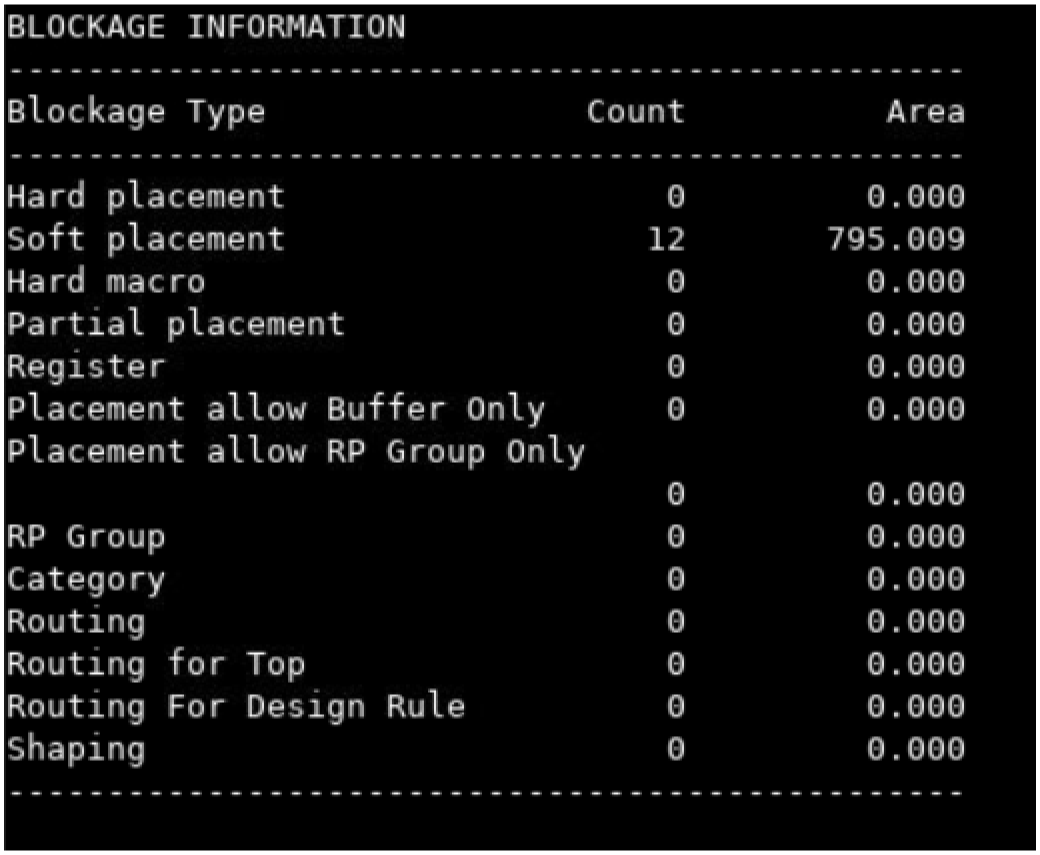


```
Startpoint: ALU_Sel[0] (input port clocked by clk)
Endpoint: Result_reg[0] (rising edge-triggered flip-flop clocked by clk)
Mode: func
Corner: slow
Scenario: func_slow
Path Group: **in2reg_default**
Path Type: max

Point                                                Incr       Path
-----------------------------------------------------------------------
clock clk (rise edge)                                0.00       0.00
clock network delay (propagated)                     0.18       0.18
input external delay                                 1.20       1.38 r
ALU_Sel[0] (in)                                      0.00       1.38 r
U80/Y (INVX0_RVT)                                    0.04       1.42 f
U124/Y (AND3X1_RVT)                                  0.07       1.49 f
U142/Y (AND2X1_RVT)                                  0.04       1.53 f
U90/Y (OR2X1_RVT)                                    0.03       1.56 f
U144/Y (OR2X1_RVT)                                   0.03       1.59 f
U145/Y (AND2X1_RVT)                                  0.03       1.62 f
U81/Y (OR3X1_RVT)                                    0.04       1.66 f
Result_reg[0]/SE (SDFFSSRX1_RVT)                     0.00       1.66 f
data arrival time                                               1.66

clock clk (rise edge)                                3.40       3.40
clock network delay (propagated)                     0.10       3.50
Result_reg[0]/CLK (SDFFSSRX1_RVT)                    0.00       3.50 r
clock uncertainty                                   -0.20       3.30
library setup time                                  -0.07       3.23
data required time                                              3.23
-----------------------------------------------------------------------
data required time                                              3.23
data arrival time                                              -1.66
-----------------------------------------------------------------------
slack (MET)                                                     1.56
```

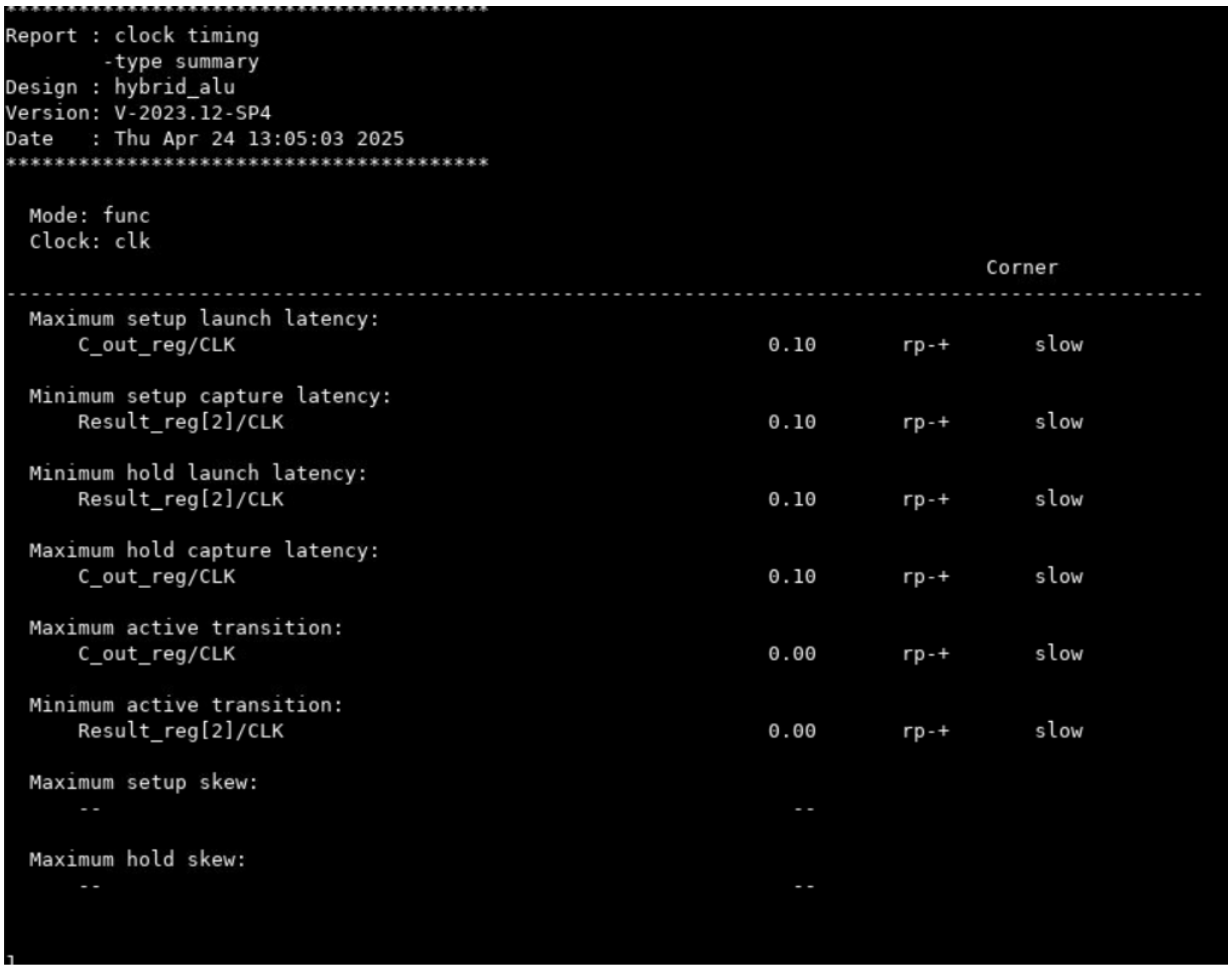
Hold Report :



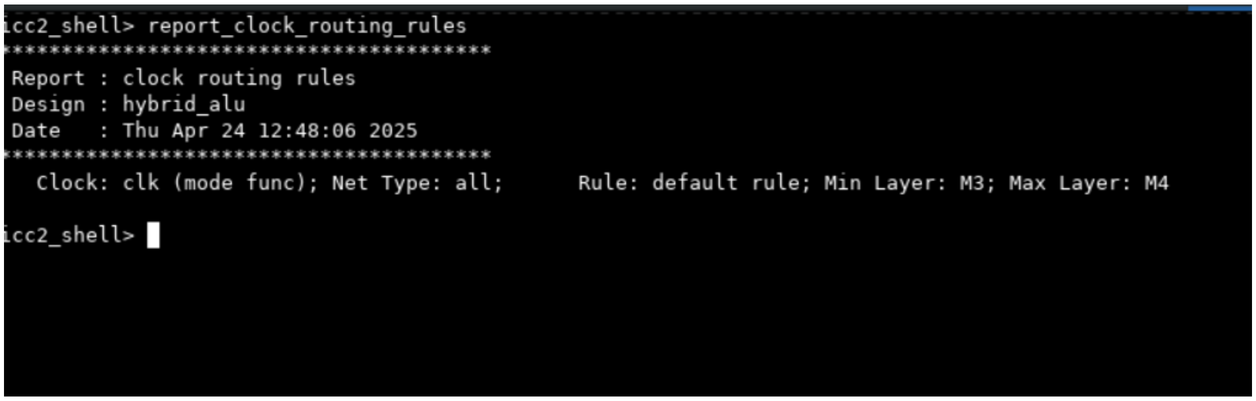
```
****************************************
Report : timing
        -path_type full
        -delay_type min
        -max_paths 1
        -report_by design
Design : hybrid_alu
Version: V-2023.12-SP4
Date   : Thu Apr 24 12:26:48 2025
****************************************

  Startpoint: Result_reg[0] (rising edge-triggered flip-flop clocked by clk)
  Endpoint: Result[0] (output port clocked by clk)
  Mode: func
  Corner: slow
  Scenario: func_slow
  Path Group: **reg2out_default**
  Path Type: min

  Point                                            Incr      Path
  ------------------------------------------------------------------------
  clock clk (rise edge)                            0.00      0.00
  clock network delay (propagated)                 0.10      0.10

  Result_reg[0]/CLK (SDFFSSRX1_RVT)                0.00      0.10 r
  Result_reg[0]/QN (SDFFSSRX1_RVT)                 0.06      0.16 f
  Result[0] (out)                                  0.00      0.16 f
  data arrival time                                          0.16

  clock clk (rise edge)                            0.00      0.00
  clock network delay (propagated)                 0.18      0.18
  clock uncertainty                                0.20      0.38
  output external delay                           -1.20     -0.82
  data required time                                        -0.82
  ------------------------------------------------------------------------
  data required time                                        -0.82
  data arrival time                                         -0.16
  ------------------------------------------------------------------------
  slack (MET)                                                0.99
```



```
BLOCKAGE INFORMATION
------------------------------------------------
Blockage Type                    Count        Area
------------------------------------------------
Hard placement                       0       0.000
Soft placement                      12     795.009
Hard macro                           0       0.000
Partial placement                    0       0.000
Register                             0       0.000
Placement allow Buffer Only          0       0.000
Placement allow RP Group Only
                                     0       0.000
RP Group                             0       0.000
Category                             0       0.000
Routing                              0       0.000
Routing for Top                      0       0.000
Routing For Design Rule              0       0.000
Shaping                              0       0.000
------------------------------------------------
```



```
****************************************
Report : clock timing
        -type summary
Design : hybrid_alu
Version: V-2023.12-SP4
Date   : Thu Apr 24 13:05:03 2025
****************************************

  Mode: func
  Clock: clk
                                                                                 Corner
-----------------------------------------------------------------------------------------------------
  Maximum setup launch latency:
      C_out_reg/CLK                                                  0.10       rp-+       slow

  Minimum setup capture latency:
      Result_reg[2]/CLK                                              0.10       rp-+       slow

  Minimum hold launch latency:
      Result_reg[2]/CLK                                              0.10       rp-+       slow

  Maximum hold capture latency:
      C_out_reg/CLK                                                  0.10       rp-+       slow

  Maximum active transition:
      C_out_reg/CLK                                                  0.00       rp-+       slow

  Minimum active transition:
      Result_reg[2]/CLK                                              0.00       rp-+       slow

  Maximum setup skew:
      --                                                               --

  Maximum hold skew:
      --                                                               --

1
```



```
icc2_shell> report_clock_routing_rules
****************************************
 Report : clock routing rules
 Design : hybrid_alu
 Date   : Thu Apr 24 12:48:06 2025
****************************************
   Clock: clk (mode func); Net Type: all;      Rule: default rule; Min Layer: M3; Max Layer: M4

icc2_shell> 
```

Clock routing rules

Files to be written post route :

write_gds ALU.gdsii

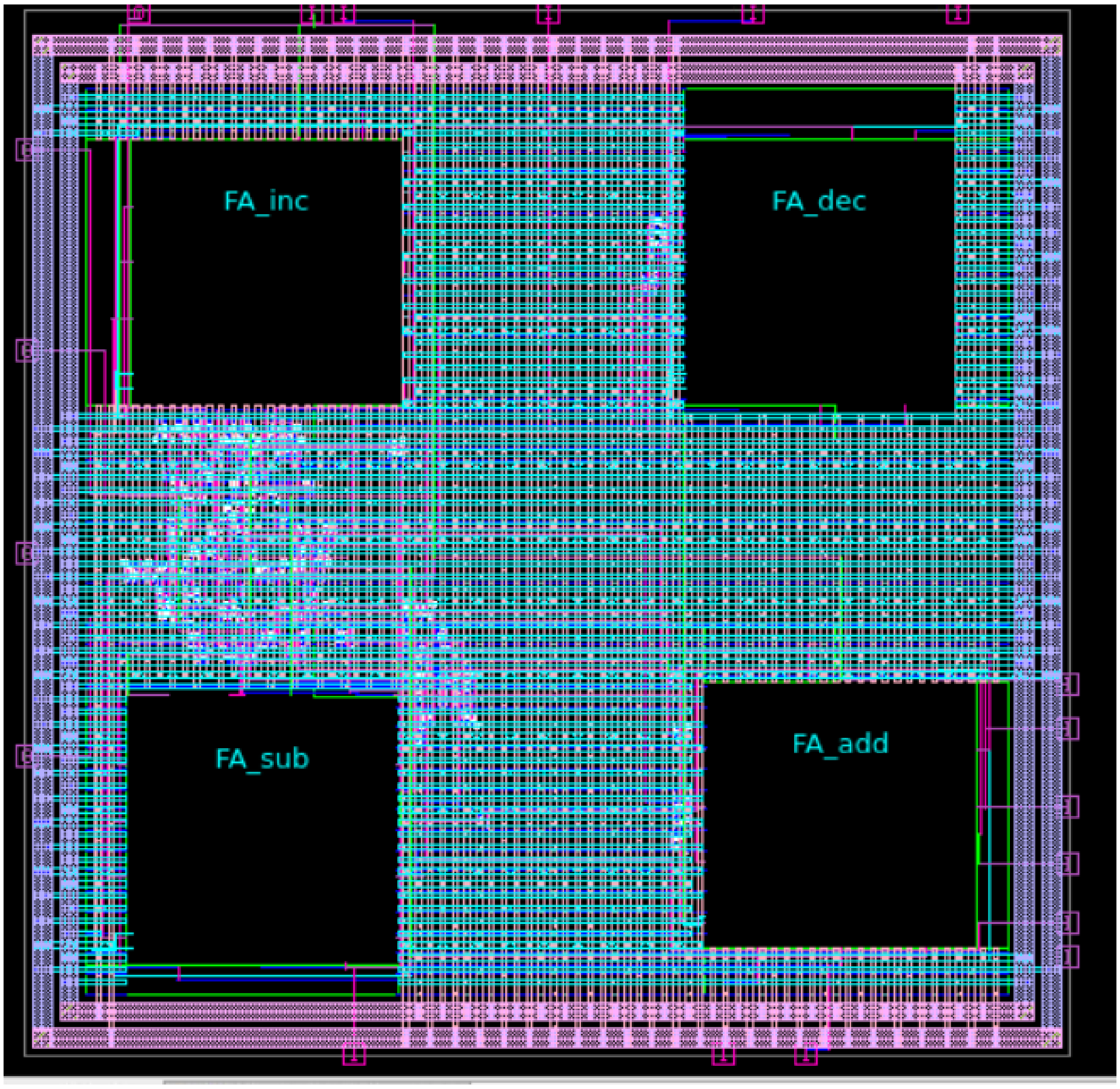
write_verilog ALU.routed.v

write_parasitics –format spef – output ALU

create_frame

write_lef ALU.lef

Final Layout of the design :



Summary and Conclusion

In this project, a hierarchical **Hybrid ALU** design was successfully implemented using the Synopsys ICC2 physical design flow. Standard cell libraries and technology files were integrated to initialize the floorplan, and soft macros representing full adders were placed manually to create a modular ALU layout. Power planning was meticulously handled using ring, mesh, and rail strategies, ensuring proper power delivery across all modules.

Clock tree synthesis and optimization were performed to ensure uniform clock distribution with minimal skew. The routing stage was completed using a timing-driven approach to enhance signal integrity and reduce crosstalk effects. Post-route checks, including DRC, LVS, and connectivity verification, were passed successfully.

Final post-route timing analysis yielded the following results:

- **Setup Slack:** +1.56 ns
- **Hold Slack:** +0.99 ns

These results indicate that the design meets all critical timing constraints with sufficient margin, proving the effectiveness of floorplanning, placement, CTS, and routing strategies applied throughout the flow.

This implementation demonstrates strong understanding and practical application of VLSI physical design principles, with successful sign-off metrics validating the design's robustness and manufacturability.