Stage_1-Synthesis

Advvlsi_02

Workarea path: /project/advvlsi/users/adamsalameh/ws/bitcoin/stage1

Adam Salameh adamsalameh@mail.tau.ac.il 207534538

Nawaf Gadban nawafgadban@mail.tau.ac.il 314702705

Anan Break ananbreak@mail.tau.ac.il 322663170

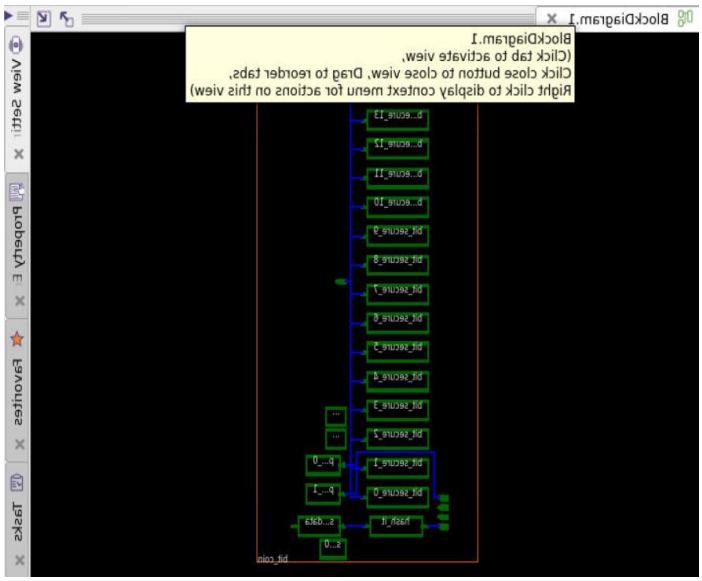
Sara Abo Salah saraabosalah23@mail.tau.ac.il 208298562

Lama Metanis Lamametanis@mail.tau.ac.il 314857152

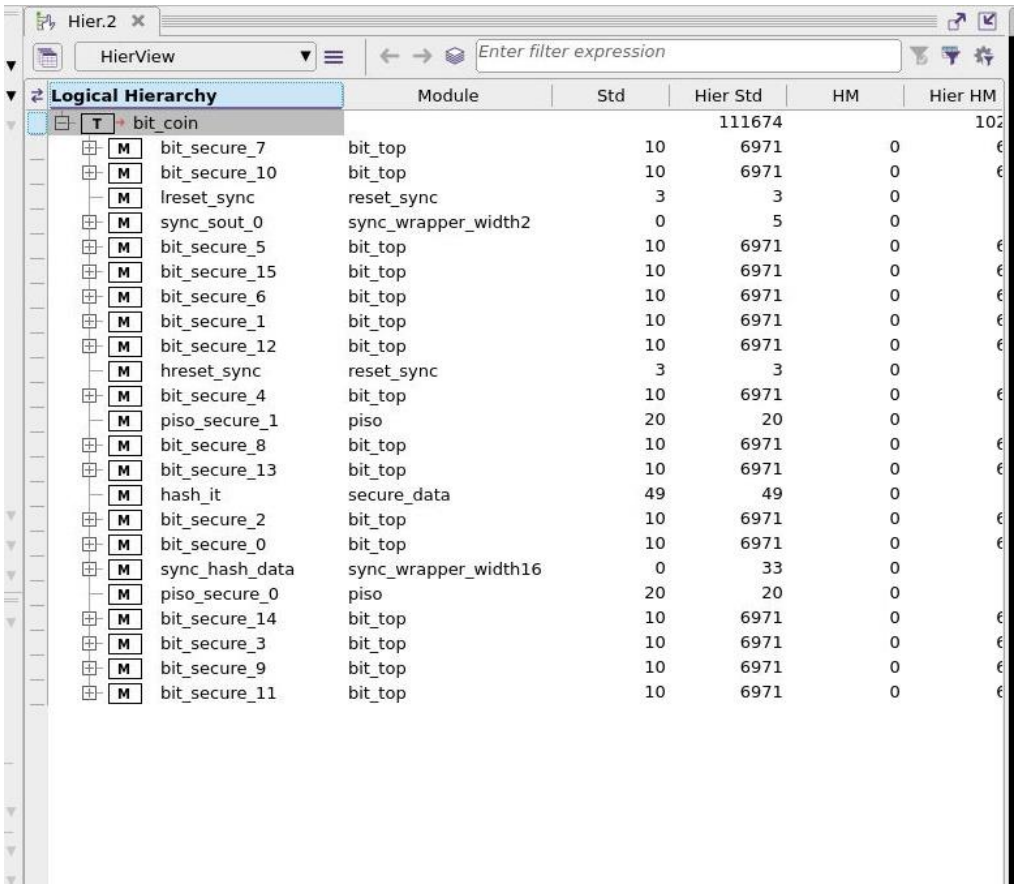
[#P3.1_Q1]

As specified in the forum we should take a screenshot of:

Bit_coin block diagram

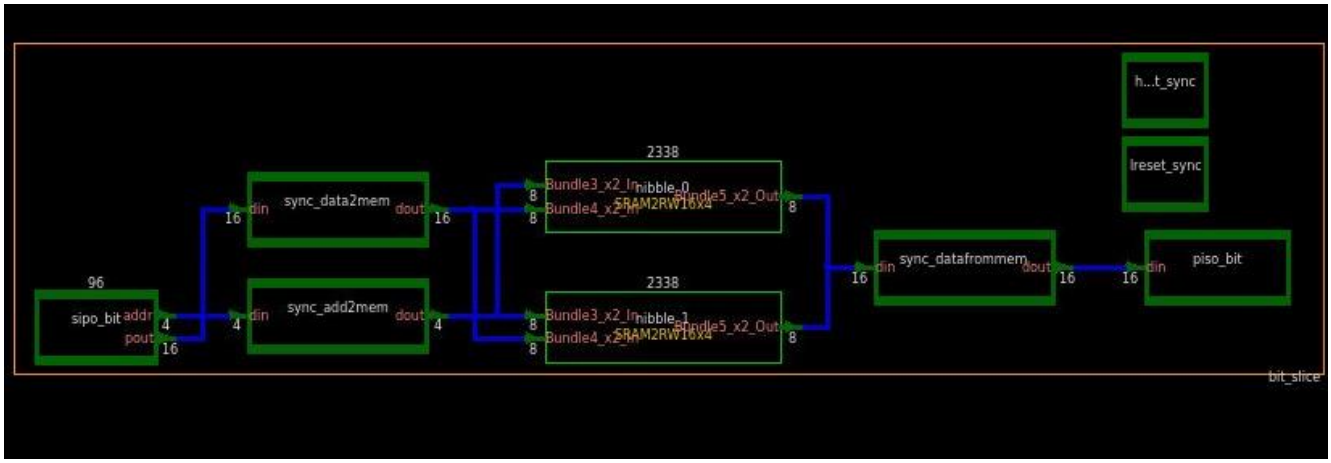


contents of the bitcoin:



Logical Hierarchy	Module	Std	Hier Std	HM	Hier HM
T bit_coin			111674		102
M bit_secure_7	bit_top	10	6971	0	6
M bit_secure_10	bit_top	10	6971	0	6
M lreset_sync	reset_sync	3	3	0	
M sync_sout_0	sync_wrapper_width2	0	5	0	
M bit_secure_5	bit_top	10	6971	0	6
M bit_secure_15	bit_top	10	6971	0	6
M bit_secure_6	bit_top	10	6971	0	6
M bit_secure_1	bit_top	10	6971	0	6
M bit_secure_12	bit_top	10	6971	0	6
M hreset_sync	reset_sync	3	3	0	
M bit_secure_4	bit_top	10	6971	0	6
M piso_secure_1	piso	20	20	0	
M bit_secure_8	bit_top	10	6971	0	6
M bit_secure_13	bit_top	10	6971	0	6
M hash_it	secure_data	49	49	0	
M bit_secure_2	bit_top	10	6971	0	6
M bit_secure_0	bit_top	10	6971	0	6
M sync_hash_data	sync_wrapper_width16	0	33	0	
M piso_secure_0	piso	20	20	0	
M bit_secure_14	bit_top	10	6971	0	6
M bit_secure_3	bit_top	10	6971	0	6
M bit_secure_9	bit_top	10	6971	0	6
M bit_secure_11	bit_top	10	6971	0	6

Bit_slice:



Warmup Questions

[#P3.2_Q1]

1.1.Write at least 5 different constraint examples, write who supplies them and their domain. For example, we have the clocks' periods constraint from the timing domain. And the architecture team is responsible for providing them.

Power Consumption Constraint (Supplied by Electrical Engineering Team, Domain: Hardware Design)

Description: This constraint specifies the maximum power consumption allowed for a device or system. It ensures that the hardware components operate within the power limits defined by the design specifications. The constraint is crucial for optimizing power usage, ensuring thermal management, and extending battery life in portable devices.

Memory Usage Constraint (Supplied by Software Engineering Team, Domain: Software Development)

Description: This constraint determines the maximum amount of memory that can be utilized by a software application or system. It helps optimize resource allocation and prevents excessive memory usage, which can lead to performance degradation or system crashes. The constraint is particularly important for memory-constrained environments like embedded systems or mobile devices.

Response Time Constraint (Supplied by Performance Engineering Team, Domain: System Performance)

Description: This constraint defines the maximum acceptable response time for a system or application. It ensures that the system responds to user requests or input within a specific timeframe. Meeting this constraint is crucial for providing a satisfactory user experience, especially in real-time

applications or interactive systems where delays can significantly impact usability.

Security Constraint (Supplied by Information Security Team, Domain: Security)

Description: This constraint outlines the security requirements and measures that need to be implemented in a system or application. It includes aspects such as data encryption, access control, authentication, and vulnerability management. The constraint aims to protect the system and its data from unauthorized access, data breaches, and other security threats.

Compliance Constraint (Supplied by Regulatory Compliance Team, Domain: Legal and Regulatory)

Description: This constraint encompasses the legal and regulatory requirements that must be met by a product or system. It ensures compliance with industry standards, government regulations, privacy laws, and other relevant guidelines. The constraint helps mitigate legal risks, avoid penalties, and maintain the reputation and trustworthiness of the organization.

1.2.Name and explain 3 techniques that we (and maybe by extension the tool) can implement during the synthesis stage to meet low power constraints?

Power Gating:

Power gating is a technique used to reduce power consumption in digital circuits by selectively shutting down power to unused or idle circuit blocks. During synthesis, the tool can implement power gating by identifying and isolating the inactive portions of the design and adding power gating cells to cut off the power supply to those blocks. This effectively reduces static power consumption, as power is only supplied to the active circuitry, conserving power when not in use.

Voltage Scaling:

Voltage scaling is a technique that involves reducing the operating voltage of a circuit to decrease power consumption. During synthesis, the tool can implement voltage scaling by analyzing the design and identifying areas where the voltage levels can be lowered without compromising functionality or timing constraints. By reducing the voltage supplied to certain circuit blocks, power consumption can be significantly reduced, as power consumption is proportional to the square of the voltage.

Clock Gating:

Clock gating is a technique used to conserve power by selectively disabling clock signals to circuit elements that are not currently in use or idle. During synthesis, the tool can implement clock gating by analyzing the design and identifying sections where the clock signals can be disabled temporarily when not required. Clock gating can significantly reduce power consumption by reducing unnecessary switching activity in the circuits, resulting in lower dynamic power consumption.

By implementing these techniques during the synthesis stage, the tool can optimize the design for low power consumption, ensuring that the power constraints are met while maintaining the functionality and performance of the system.

1.3. What are the tradeoffs with the solutions that you offered?

Power Gating:

Tradeoffs:

a) Increased design complexity: Power gating introduces additional complexity to the design, as it requires the insertion of power gating cells and control circuitry. This can make the design more challenging to implement, verify, and debug.

b) Increased design area: Power gating cells and associated control circuitry occupy additional area on the chip, which can increase the overall chip size and manufacturing costs.

c) Increased design latency: When transitioning from a power-gated state to an active state, there can be a latency as power needs to be restored to the gated circuit blocks. This latency can impact the overall system performance and responsiveness.

Voltage Scaling:

Tradeoffs:

a) Impact on timing and performance: Reducing the operating voltage of a circuit can affect its performance and timing. Lower voltages can lead to slower switching times and increased propagation delays, which may require additional design optimizations to meet timing constraints.

b) Increased susceptibility to noise: Lower voltages make circuits more susceptible to noise and voltage fluctuations. This can introduce reliability concerns and may require additional design considerations, such as noise mitigation techniques or increased guard bands.

c) Impact on circuit robustness: Lower voltages can reduce the noise margin and impact the robustness of the circuit against process variations and environmental conditions. Careful analysis and design techniques may be needed to ensure reliable operation under varying voltage levels.

Clock Gating:

Tradeoffs:

a) Increased design complexity: Clock gating requires additional logic to control the clock signals, resulting in increased design complexity and potential design challenges. It may require additional effort for clock domain synchronization and analysis to ensure proper functionality.

b) Potential impact on clock tree: Clock gating can disrupt the balanced distribution of clock signals in the design, leading to skew and other timing issues. Proper analysis and design techniques are necessary to mitigate these effects and maintain proper clock distribution.

c) Increased design and verification effort: The insertion of clock gating logic requires careful consideration and analysis to ensure correct functionality and avoid any potential glitches or hazards. Verification efforts need to be increased to validate the correct operation of clock gating cells.

In summary, while these low power techniques offer significant power savings, they come with tradeoffs that need to be carefully evaluated and managed during the design process. The specific tradeoffs and their impacts depend on the design requirements, constraints, and the complexity of the system being designed.

[#P3.2_Q2]

2.1 What are False paths?

In digital design and computer architecture, false paths are paths within a circuit that do not need to meet timing requirements. They are paths that do not affect the overall functionality of the circuit and are intentionally ignored during the timing analysis process.

Timing analysis is performed to ensure that signals in a digital circuit arrive at their intended destinations within specified time constraints. However, in complex circuits, there may be paths that are not critical to the circuit's operation or performance. These paths could be due to unused or nonessential components, bypassed logic, or other design considerations.

False paths are identified during the timing analysis process and are marked as noncritical. By designating them as false paths, the timing analysis tools do not consider them when calculating timing constraints and determining whether the circuit meets the required timing specifications. This allows designers to focus on the critical paths that need to meet timing requirements, optimizing them for performance.

By ignoring false paths during timing analysis, designers can simplify the design process and reduce the complexity of timing closure. This helps in achieving better performance, reducing design iterations, and facilitating faster and more efficient development of digital circuits.

2.2. When should we use them?

False paths are used in digital design to designate noncritical paths within a circuit that can be ignored during timing analysis. They are typically employed in the following situations:

Unused components: Paths associated with components that are not utilized in the circuit's intended operation can be designated as false paths.

Bypassed logic: Paths that are bypassed by specific conditions or logic within the circuit can be treated as false paths.

Optimization: Paths with relaxed timing requirements or minimal impact on circuit performance can be marked as false paths to focus optimization efforts on critical paths.

Complexity reduction: Marking false paths simplifies timing analysis, reducing the complexity of timing closure in complex designs.

It's crucial to use false paths judiciously, ensuring they do not affect circuit functionality or performance. Thorough verification and validation are necessary to avoid incorrectly marking false paths, which could lead to unexpected issues in the final circuit implementation.

3. [#P3.2_Q3] Corners, modes, and scenarios:

3.1. Look at bitcoin_stage_1.tcl and locate the part related to the corners, modes, and scenarios (MCMM). Which scenario \ scenarios are active?

```
# mcmm_setup:
# Remove all MCMM related info
remove_corners   -all
remove_modes     -all
remove_scenarios -all

# Create Corners
create_corner Fast
create_corner Typical
create_corner Slow

# Set parasitics parameters
set_parasitics_parameters -early_spec Cmin -late_spec  Cmin -corners {Fast}
set_parasitics_parameters -early_spec Cnom -late_spec  Cnom -corners {Typical}
set_parasitics_parameters -early_spec Cmax -late_spec  Cmax -corners {Slow}

# Create Mode
create_mode  FUNC
current_mode FUNC

# Create Scenarios
create_scenario -mode FUNC -corner Fast    -name FUNC_Fast
create_scenario -mode FUNC -corner Typical -name FUNC_Typical
create_scenario -mode FUNC -corner Slow    -name FUNC_Slow
```

Based on the provided setup, the active scenarios are:

1. FUNC_Fast: This scenario is active for the "FUNC" mode and the "Fast" corner.

2. FUNC_Typical: This scenario is active for the "FUNC" mode and the "Typical" corner.

3. FUNC_Slow: This scenario is active for the "FUNC" mode and the "Slow" corner.

These active scenarios represent specific combinations of mode and corner that have been created and set up in the script. Each scenario represents a different operating condition or test scenario for the design.

3.2.What kind of corners, modes, and scenarios would we test in a typical design and why?

In a typical design, a variety of corners, modes, and scenarios are tested to ensure the robustness, reliability, and functionality of the design across different operating conditions. Some common examples include:

Process Corners: Testing under optimistic (fast), average (typical), and pessimistic (slow) process variations to evaluate performance across manufacturing variations.

Voltage Corners: Testing under different voltage levels such as nominal, low-power, and high-performance to assess functionality across voltage ranges.

Temperature Corners: Testing under different temperature extremes (low, nominal, high) to ensure the design can operate reliably across various environmental conditions.

Power Modes: Testing under different power modes, including normal, low-power, and high-performance, to evaluate power consumption and performance trade-offs.

Functional Scenarios: Testing with various input stimuli, stress testing, and specific use case scenarios to validate the design's functionality and handling of real-world scenarios.

Timing Scenarios: Testing under worst-case timing paths and critical race conditions to verify timing requirements and avoid timing violations.

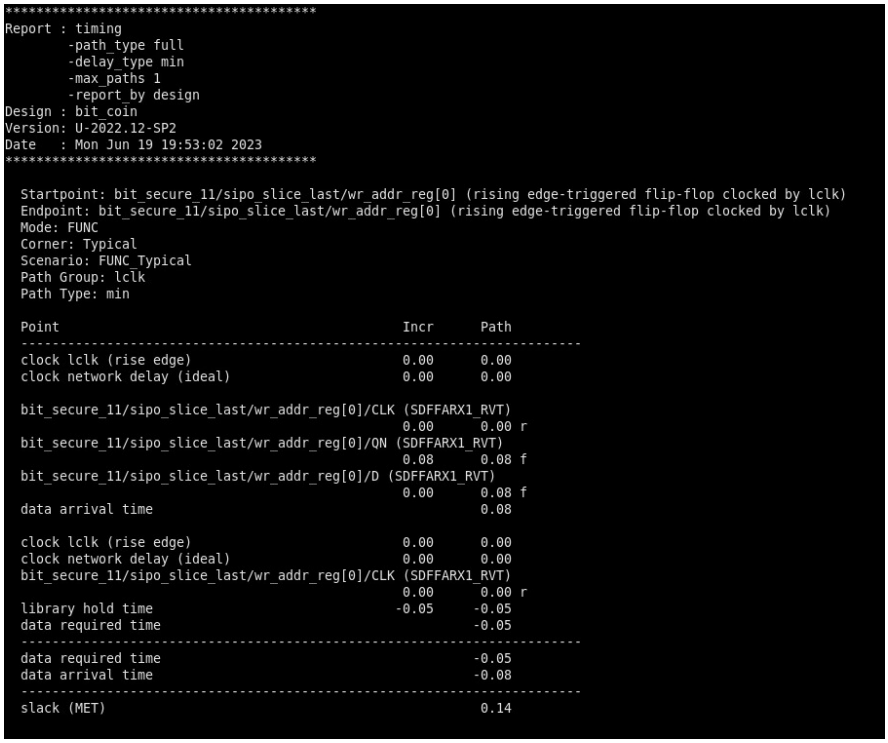
4. [#P3.2_Q4] What are Multibit Registers?

Multibit registers, also known as multibit flip-flops, are sequential logic elements used in digital circuits to store and manipulate multiple bits of data simultaneously. They are constructed by combining multiple single-bit registers, such as flip-flops, into a single unit. Multibit registers allow independent storage and manipulation of each bit of data. They are used in various digital systems and processors for tasks such as data storage, arithmetic operations, and state machine implementations. Multibit registers are essential components that enable the handling of larger data widths in modern computing systems.

2.3.[#P3.3_Q1] What is the slack for the SETUP and HOLD constraint? Attach a print screen

These reports are typically generated during the synthesis or mapping process, and they analyze the timing constraints and the circuit's behavior against those constraints.

HOLD:



```
****************************************
Report : timing
        -path_type full
        -delay_type min
        -max_paths 1
        -report_by design
Design : bit_coin
Version: U-2022.12-SP2
Date   : Mon Jun 19 19:53:02 2023
****************************************

  Startpoint: bit_secure_11/sipo_slice_last/wr_addr_reg[0] (rising edge-triggered flip-flop clocked by lclk)
  Endpoint: bit_secure_11/sipo_slice_last/wr_addr_reg[0] (rising edge-triggered flip-flop clocked by lclk)
  Mode: FUNC
  Corner: Typical
  Scenario: FUNC_Typical
  Path Group: lclk
  Path Type: min

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

  bit_secure_11/sipo_slice_last/wr_addr_reg[0]/CLK (SDFFARX1_RVT)
                                                 0.00       0.00 r
  bit_secure_11/sipo_slice_last/wr_addr_reg[0]/QN (SDFFARX1_RVT)
                                                 0.08       0.08 f
  bit_secure_11/sipo_slice_last/wr_addr_reg[0]/D (SDFFARX1_RVT)
                                                 0.00       0.08 f
  data arrival time                                         0.08

  clock lclk (rise edge)                         0.00       0.00
  clock network delay (ideal)                    0.00       0.00
  bit_secure_11/sipo_slice_last/wr_addr_reg[0]/CLK (SDFFARX1_RVT)
                                                 0.00       0.00 r
  library hold time                             -0.05      -0.05
  data required time                                       -0.05
  -----------------------------------------------------------------------
  data required time                                       -0.05
  data arrival time                                        -0.08
  -----------------------------------------------------------------------
  slack (MET)                                               0.14
```

SETUP:

```
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -report_by design
Design : bit_coin
Version: U-2022.12-SP2
Date   : Mon Jun 19 19:53:02 2023
****************************************

  Startpoint: bit_secure_2/lreset_sync/reset_sync_reg (rising edge-triggered flip-flop clocked by lclk)
  Endpoint: bit_secure_2/slice_9/sipo_bit/rd_addr_reg[1] (recovery check against rising-edge clock clocked by lclk)
  Mode: FUNC
  Corner: Typical
  Scenario: FUNC_Typical
  Path Group: lclk
  Path Type: max

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

  bit_secure_2/lreset_sync/reset_sync_reg/CLK (DFFARX1_RVT)
                                                 0.00      0.00 r
  bit_secure_2/lreset_sync/reset_sync_reg/Q (DFFARX1_RVT)
                                               5316.49   5316.49 r
  bit_secure_2/slice_9/sipo_bit/rd_addr_reg[1]/SETB (SDFFASX1_RVT)
                                                 0.00    5316.49 r
  data arrival time                                      5316.49

  clock lclk (rise edge)                         1.00      1.00
  clock network delay (ideal)                    0.00      1.00
  bit_secure_2/slice_9/sipo_bit/rd_addr_reg[1]/CLK (SDFFASX1_RVT)
                                                 0.00      1.00 r
  library setup time                         -2206.15  -2205.15
  data required time                                   -2205.15
  -----------------------------------------------------------------------------
  data required time                                   -2205.15
  data arrival time                                    -5316.49
  -----------------------------------------------------------------------------
  slack (VIOLATED)                                     -7521.64
```

2.4.[#P3.3_Q2] Read the reports and find the starting and ending point of the critical path. If needed use the “-through” option to specify the specific path between those cells. Fill the code for the command: (the through is optional, you can delete the flag if you don’t use it).

Setup Timing Report: (violated)

Startpoint: bit_secure_2/lreset_sync/reset_sync_reg (rising edge-triggered flip-flop clocked by lclk)

Endpoint: bit_secure_2/slice_9/sipo_bit/rd_addr_reg[1] (recovery check against rising-edge clock clocked by lclk)

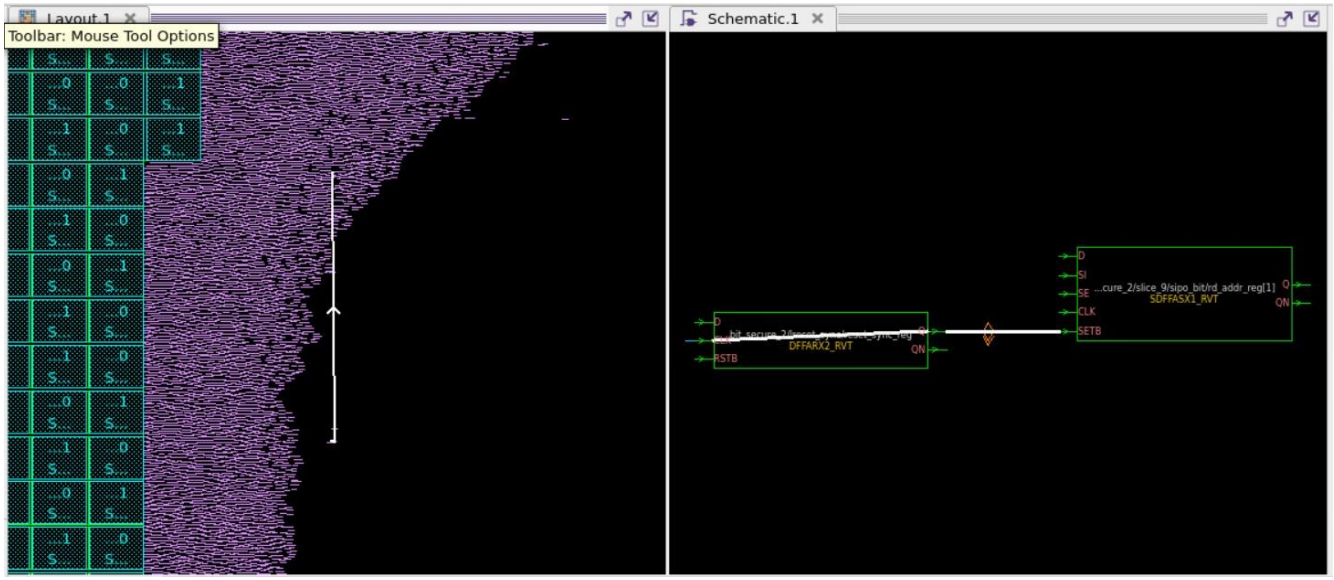
Hold Timing Report:

Startpoint: bit_secure_11/sipo_slice_last/wr_addr_reg[0] (rising edge-triggered flip-flop clocked by lclk)

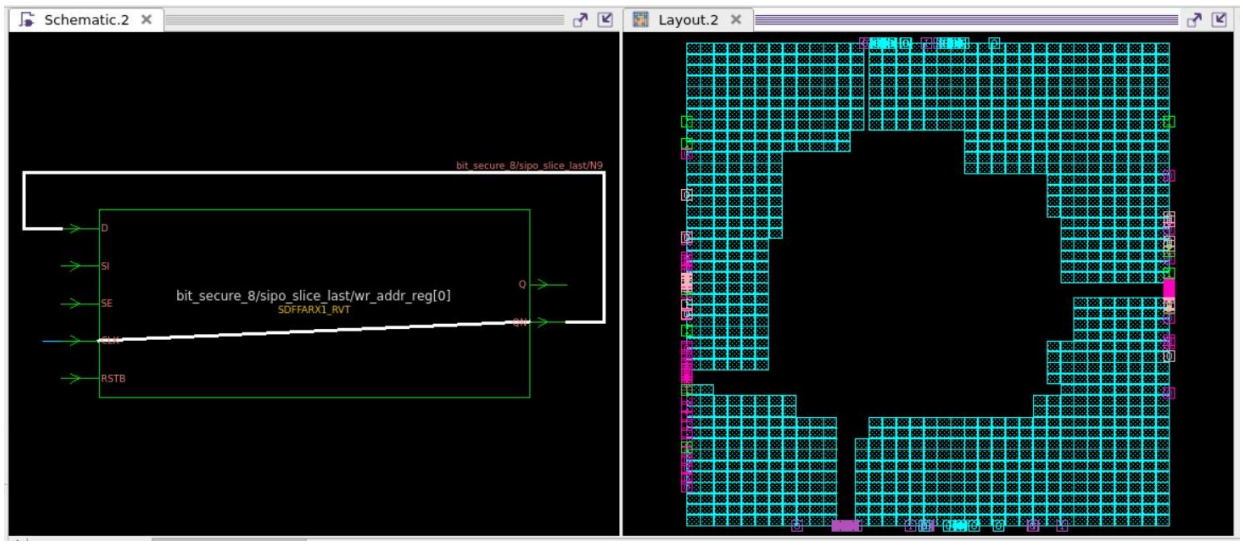
Endpoint: bit_secure_11/sipo_slice_last/wr_addr_reg[0] (rising edge-triggered flip-flop

2.5.[#P3.3_Q3] While we still in the context of the path (done by section d) select the “Schematic View“ and take a snapshot of the critical paths for SETUP and HOLD:

Setup path:

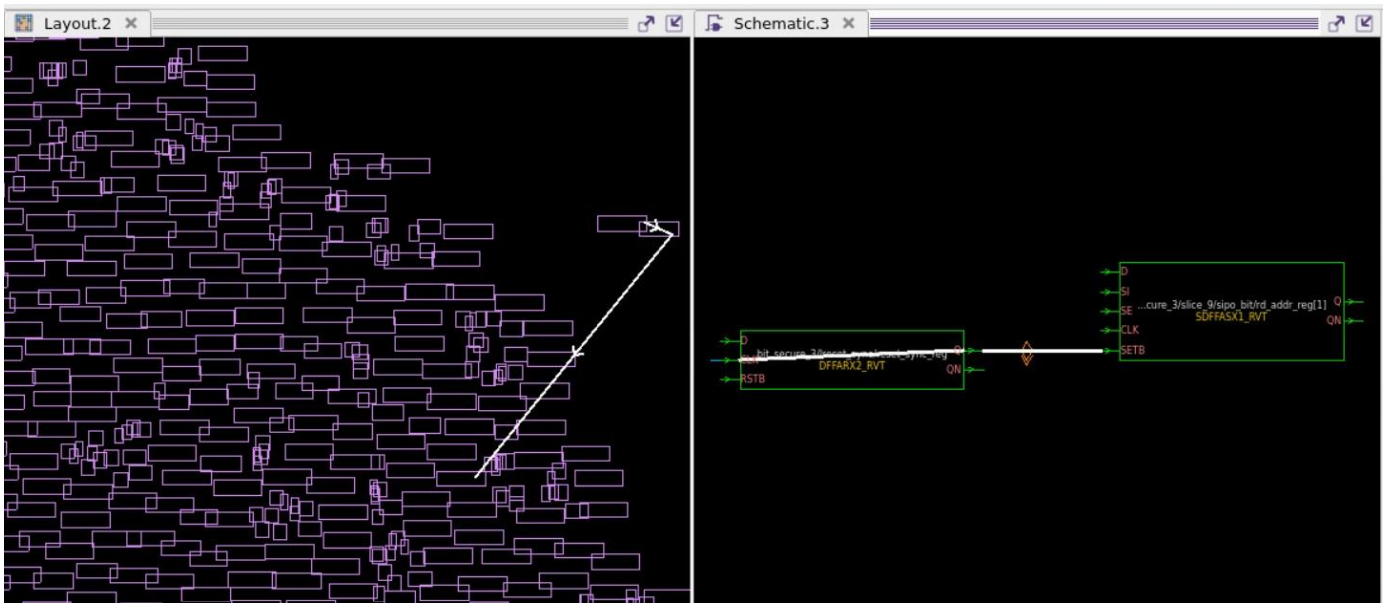


Hold path:



.[#P3.3_Q4]] Find an additional timing path with a major negative slack. To do that, use the report_timing command with the following flag: "-max_path". Try to find a pattern to those paths. Hint: Maybe you can find a common factor that can cause the problematic slack. Use this pattern to fill the code for the "set_false_path" command.

Setup:



The path we got:

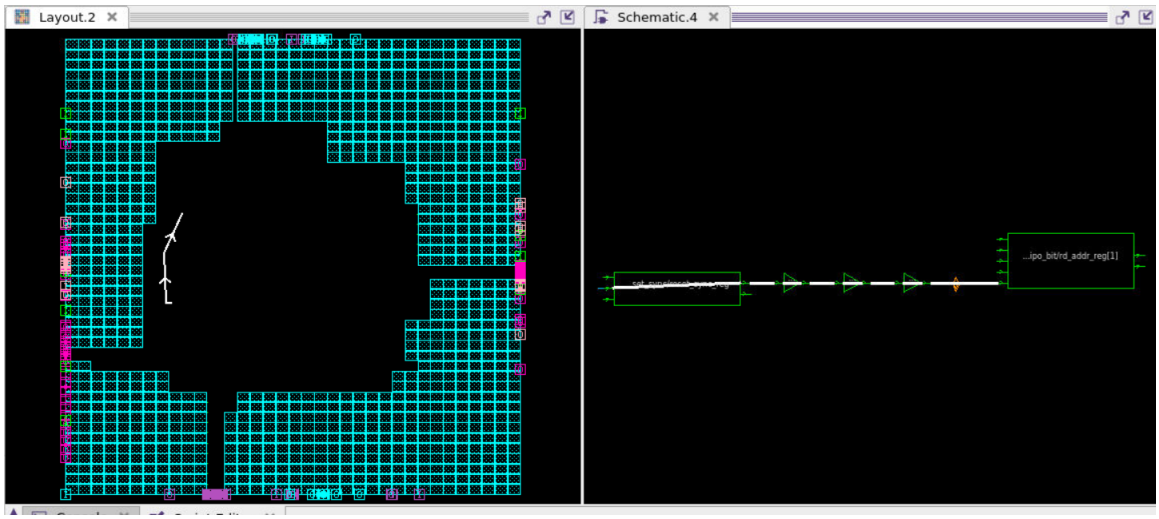
Start point: bit_secure_3/lreset_sync/reset_sync_reg

End point: bit_secure_3/slice_9/sipo_bit/rd_addr_reg[1]

we got the same negative slack -3903.94 as we can see above the reason we got the same slack is because basically both paths uses the same components(sdffasx1_rvt and dffarx2_rvt) and also t_log is the same.

2.8.[#P3.3_Q5] After you ran the “final_opto” compilation, change your selection again to the same path you chose for the SETUP path in [#P3.3_Q3].

After running the final_opto compilation we got:



As we can see now tlogic isn’t the same because we already ran the “Setup Analysis and Repair” which deals with violations and because we solved them so now we see “different path”(the same logic but changed so setup constrains could be valid)

[#P3.3_Q6] Analyze those reports and answer the following questions:

4.1.Setup timing: What is the first stage in the compile_fusion flow that fixes the SETUP violations?

In the compile_fusion process, the initial phase that usually deals with SETUP violations is referred to as the "Setup Analysis and Repair" stage. This particular stage is dedicated to detecting and resolving any instances of violations pertaining to setup timing specifications within a digital design.

4.2.Hold timing: From which stage in the general chip implementation (Synthesis, STA, Floorplan, Placement…) should we check the hold violations and why?

During the chip implementation process, the Static Timing Analysis (STA) stage is responsible for checking and addressing hold violations. This stage occurs after synthesis and placement but before final routing and signoff. Hold violations occur when the arrival time of data at a register is too close to the clock edge, making it challenging for the data to stabilize properly.

In order to address hold violations, the placement and routing of sequential elements like flip-flops are carefully considered, taking into account the hold requirements. Hold violations can also be influenced by clock skew, which refers to the difference in arrival times of the clock signal. STA takes into account clock skew and analyzes worst-case scenarios to accurately assess hold violations.

Fixing hold violations often involves inserting buffers or introducing delays in the affected paths, which can impact the overall performance of the design. By checking hold violations during the STA stage, designers can identify and implement necessary fixes while evaluating their impact on the design as a whole. STA combines information from physical placement, routing, and timing constraints to conduct a thorough analysis of hold violations before proceeding to the final stages of the chip implementation process.

4.3.Area: What can we derive out of the following information from the area report?

```
Number of ports:                        12040
Number of nets:                        100130
Number of cells:                        78843
Number of combinational cells:          19614
Number of sequential cells:             59229
Number of macros/black boxes:            1024
Number of buf/inv:                       1641
Number of references:                      29

Combinational area:                  47473.85
Buf/Inv area:                         2123.88
Noncombinational area:              489446.02
Macro/Black Box area:              2394010.91

Total cell area:                   2930930.78
```

4.3.1) Number of ports: 12040

4.3.2) The ratio between number of the combinational cells and the sequential cells: 33.1%

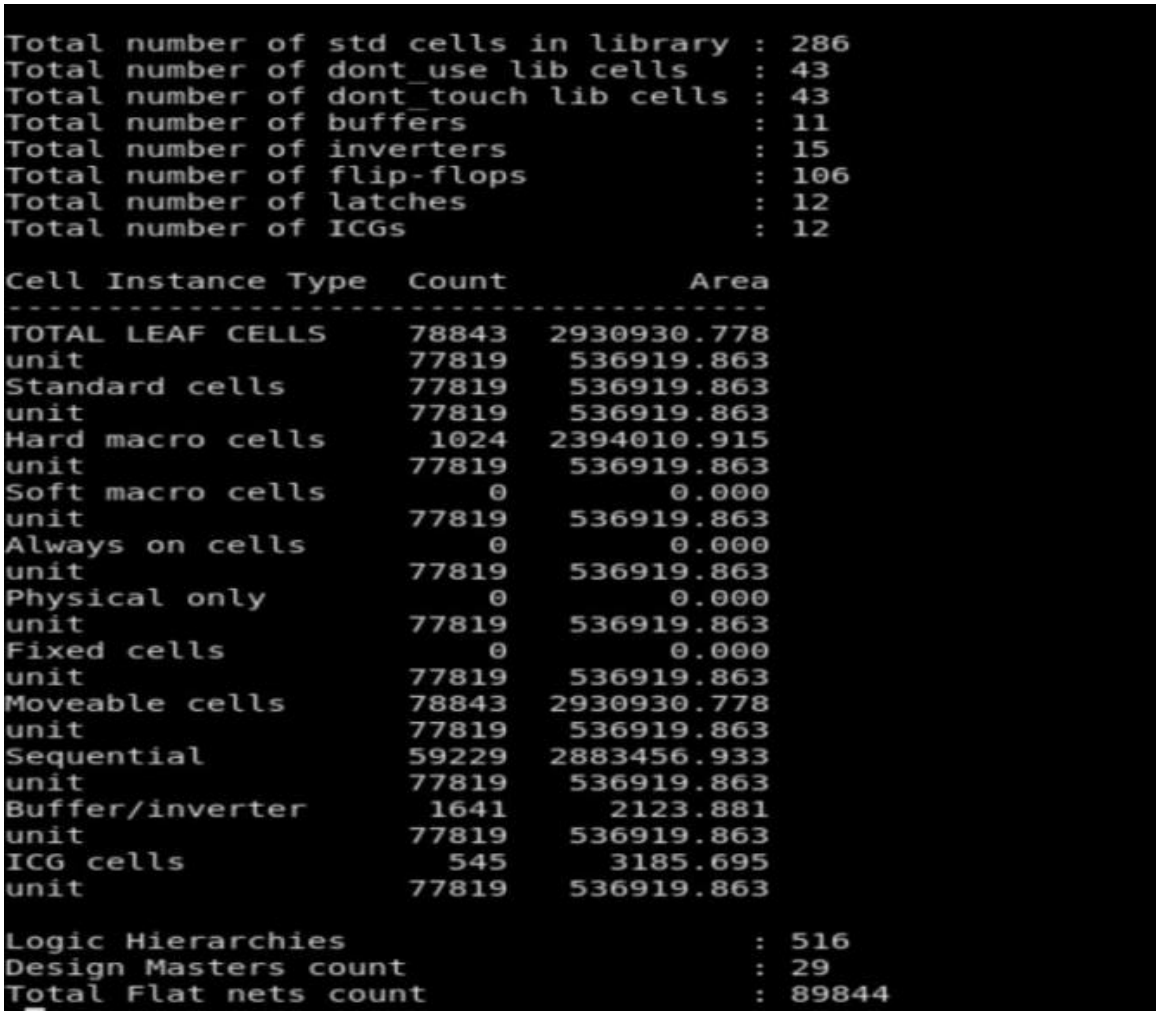
4.3.3) Number of macros: 1024

4.3.4) . Number of buffers and inverters: 1641

4.3.5) Area of cells + area of macros:

536919.87+ 2394010.91=2930930.78

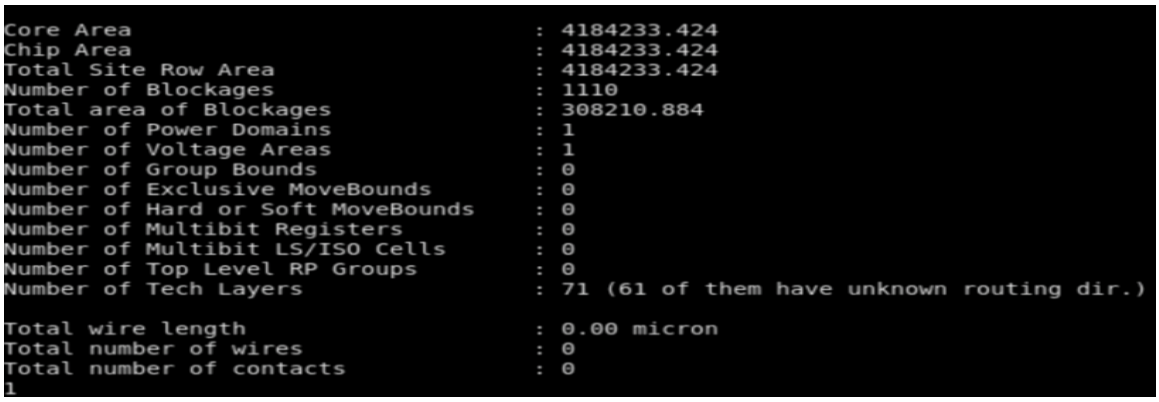
4.4.Design: What can we derive out of the following information from the design report?



```
Total number of std cells in library : 286
Total number of dont_use lib cells   : 43
Total number of dont_touch lib cells : 43
Total number of buffers              : 11
Total number of inverters            : 15
Total number of flip-flops           : 106
Total number of latches              : 12
Total number of ICGs                 : 12

Cell Instance Type  Count          Area
-----------------------------------------
TOTAL LEAF CELLS    78843  2930930.778
unit                77819   536919.863
Standard cells      77819   536919.863
unit                77819   536919.863
Hard macro cells     1024  2394010.915
unit                77819   536919.863
Soft macro cells        0        0.000
unit                77819   536919.863
Always on cells         0        0.000
unit                77819   536919.863
Physical only           0        0.000
unit                77819   536919.863
Fixed cells             0        0.000
unit                77819   536919.863
Moveable cells      78843  2930930.778
unit                77819   536919.863
Sequential          59229  2883456.933
unit                77819   536919.863
Buffer/inverter      1641     2123.881
unit                77819   536919.863
ICG cells             545     3185.695
unit                77819   536919.863

Logic Hierarchies                    : 516
Design Masters count                 : 29
Total Flat nets count                : 89844
```



```
Core Area                          : 4184233.424
Chip Area                          : 4184233.424
Total Site Row Area                : 4184233.424
Number of Blockages                : 1110
Total area of Blockages            : 308210.884
Number of Power Domains            : 1
Number of Voltage Areas            : 1
Number of Group Bounds             : 0
Number of Exclusive MoveBounds     : 0
Number of Hard or Soft MoveBounds  : 0
Number of Multibit Registers       : 0
Number of Multibit LS/ISO Cells    : 0
Number of Top Level RP Groups      : 0
Number of Tech Layers              : 71 (61 of them have unknown routing dir.)

Total wire length                  : 0.00 micron
Total number of wires              : 0
Total number of contacts           : 0
1
```

4.4.1) ICG - integrated clock gates:

number of ICGs cells is :545 and the number of ICG:12 and every one of the ICG : 545/12 &AREA:3185.695

4.4.2) Latches: 12

4.4.3) Number of Power Domains:1

4.5.Power: What can we derive out of the following information from the power report?

```
Information: Activity propagation will be performed for scenario FUNC_Typical.
Information: Doing activity propagation for mode 'FUNC' and corner 'Typical' with effort level 'medium'. (POW-024)
Information: Timer-derived activity data is cached on scenario FUNC_Typical (POW-052)
Infomation: Fast mode activity propagation power.rtl_activity_annotation setup is ignored. Always use accurate mode.
Information: Turn on parallel simulation of generator nets.
Information: Running switching activity propagation in scalar mode!

 **** Information : No. of simulation cycles = 6 ****
Information: The stitching and editing of coupling caps is turned OFF for design 'bitcoin_stage_1.dlib:bit_coin.design'. (TIM-125)
Information: Design Average RC for design bit_coin  (NEX-011)
Information: r = 1.565692 ohm/um, via_r = 0.514221 ohm/cut, c = 0.100765 ff/um, cc = 0.000000 ff/um (X dir) (NEX-017)
Information: r = 1.782872 ohm/um, via_r = 0.418669 ohm/cut, c = 0.083130 ff/um, cc = 0.000000 ff/um (Y dir) (NEX-017)
Information: The RC mode used is VR for design 'bit_coin'. (NEX-022)
Information: Update timing completed net estimation for all the timing graph nets (TIM-111)
Information: Net estimation statistics: timing graph nets = 89324, routed nets = 0, across physical hierarchy nets = 0, parasitics cac
parasitics annotated nets = 0, multi-voltage nets = 0. (TIM-112)
************************************************************
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
ML Acceleration:                          off
************************************************************
Mode: FUNC
Corner: Typical
Scenario: FUNC_Typical
Voltage: 0.85
Temperature: 125.00

Voltage Unit         : 1V
```

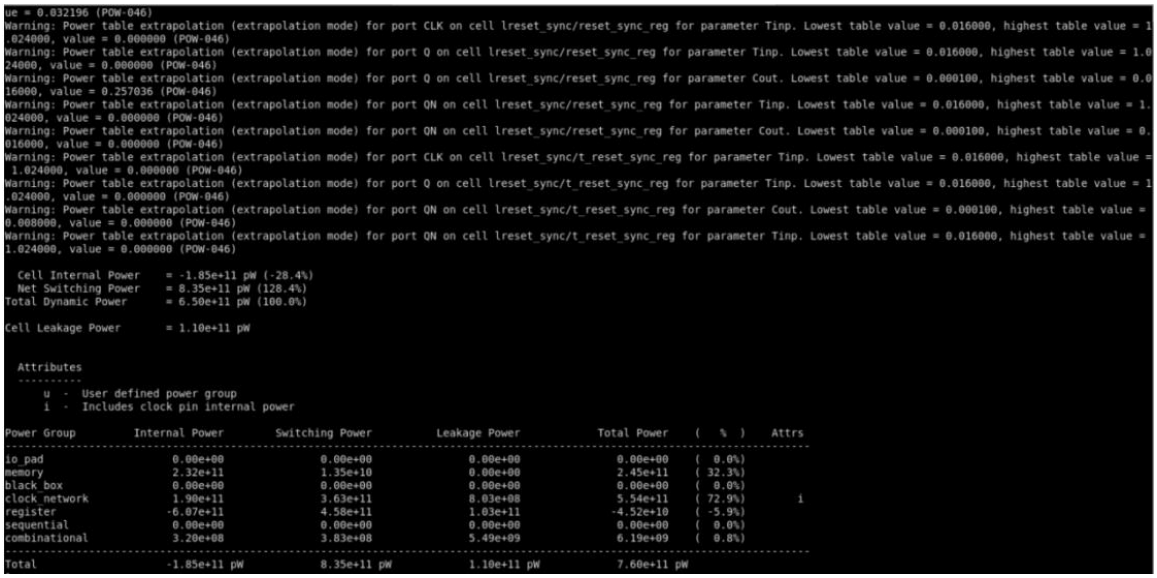
```
Mode: FUNC
Corner: Typical
Scenario: FUNC_Typical
Voltage: 0.85
Temperature: 125.00

Voltage Unit         : 1V
Capacitance Unit     : 1fF
Time Unit            : 1ns
Temperature Unit     : 1C
Dynamic Power Unit   : 1pW
Leakage Power Unit   : 1pW

Switched supply net power scaling:
scaling for leakage power

Supply nets:
VDD (power) probability 1.00 (default)
VSS (ground) probability 1.00 (default)
Warning: Power table extrapolation (extrapolation mode) for port D on cell lreset_sync/reset_sync_reg for parameter Tinp. Lowest table value = inf, highest table value = inf, val
ue = 0.032196 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port CLK on cell lreset_sync/reset_sync_reg for parameter Tinp. Lowest table value = 0.016000, highest table value = 1
.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port Q on cell lreset_sync/reset_sync_reg for parameter Tinp. Lowest table value = 0.016000, highest table value = 1.0
24000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port Q on cell lreset_sync/reset_sync_reg for parameter Cout. Lowest table value = 0.000100, highest table value = 0.0
16000, value = 0.257036 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell lreset_sync/reset_sync_reg for parameter Tinp. Lowest table value = 0.016000, highest table value = 1.
024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell lreset_sync/reset_sync_reg for parameter Cout. Lowest table value = 0.000100, highest table value = 0.
016000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port CLK on cell lreset_sync/t_reset_sync_reg for parameter Tinp. Lowest table value = 0.016000, highest table value =
 1.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port Q on cell lreset_sync/t_reset_sync_reg for parameter Tinp. Lowest table value = 0.016000, highest table value = 1
.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell lreset_sync/t_reset_sync_reg for parameter Cout. Lowest table value = 0.000100, highest table value =
0.008000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell lreset_sync/t_reset_sync_reg for parameter Tinp. Lowest table value = 0.016000, highest table value =
1.024000, value = 0.000000 (POW-046)

  Cell Internal Power    = -1.85e+11 pW (-28.4%)
  Net Switching Power    = 8.35e+11 pW (128.4%)
```



4.5.1. What is the static, dynamic and total power after all compile_fusion command?

power=Internal power + Leakage Power=1.10e+11 + 1.85e+11=1.96e+11 pw

Dynamic power= Switching power + Internal Power=6.50e+11 pw Static

Total power = Dynamic power + Static power =7.60e+11 pw

4.5.2. What does each of the power types mean?

Power consumption in electronic devices involves various components:

Leakage Power: It refers to the power consumed when a transistor or circuit is in a non-switching state, primarily due to current leakage.

Switching Power: This is the power consumed during active state transitions, which occurs as a result of charging and discharging capacitances.

Internal Power: This component represents the power consumed for internal operations within the device. It includes activities such as signal propagation, clock generation, and the functioning of internal circuitry. Internal power encompasses both static power (non-switching power) and dynamic power (power associated with state transitions).

Static Power: Static power refers to the power consumed by a circuit when it is in a steady state and not actively switching. It encompasses leakage power and other forms of power dissipation that are not related to switching.

Dynamic Power: Dynamic power is the power consumed during active switching. It comprises both the switching power associated with state transitions and the internal power consumed by the device's internal circuit operations.

In summary, leakage power and static power are terms used interchangeably to describe non-switching power, while switching power and dynamic power are related to power consumption during active state changes. Internal power represents the overall power consumed by internal circuit operations and encompasses both static and dynamic power.