Advanced VLSI

Bitcoin

Stage 4 : Placement

Ahmad foqara 322389982

Mohammad foqara 212193916

Neama hebi 212100705

Ward aboyonis 212021869

Work path: /project/advvlsi/users/ mohammadf1 /ws/bitcoin

[#P3.1_Q1] What is the difference between Placement and Floor planning in physical design?

type	floorplan	placement
goal	Determine the overall structure and layout of the chip.	Optimize and precis placement of the standard cells within the position the decided in the floorplan, to Minimize wire length, reduce congestion, and meet timing requirements
Output	floorplan that provides a general sketch of where each large block.	detailed map of where each standard cell is located on the chip
Order of Execution	Floorplan come first	Placement comes after floorplan
Level of Detail	High level: Operates at a higher, more abstract level, arranging blocks and I/O pads.	Low-level: Works on a granular level, arranging individual cells within blocks.
Scope	Estimates global routing paths between blocks and determines core and chip aspect ratios.	Assigns specific locations to cells within blocks and establishes local routing paths.

[#P3.1_Q2] What are some of the common techniques used in Placement optimization?

Common Techniques in Placement Optimization	Description
Min-Cut Placement	This technique involves recursively dividing a circuit into smaller sub-circuits (or partitions) while minimizing the number of interconnections (cuts) between them. The goal is to reduce the wire length and improve the overall placement quality.
Quadratic Placement	This technique models the placement problem using a quadratic objective function, where the goal is to minimize the sum of squared distances between connected components. The analogy with springs is appropriate, as it reflects the idea of minimizing the "tension" in the system.
Simulated Annealing	A probabilistic technique inspired by the annealing process in metallurgy. It starts with an initial solution and explores the solution space by making random changes. It probabilistically accepts changes that worsen the solution to escape local optima.

Genetic Algorithms	Inspired by natural selection, this technique uses processes like mutation and crossover to explore the solution space. Each potential placement is treated as an individual, and those with better fitness are selected to create offspring in subsequent generations.
Partitioning Algorithms	These algorithms divide the circuit into smaller partitions to minimize interconnect length. Examples include the Kernighan-Lin and Fiduccia-Mattheyses algorithms, which optimize placement by minimizing the number of connections crossing partition boundaries.
Force-Directed Placement	Cells are modeled as objects under attractive (wire connections) and repulsive (overlap reduction) forces. The algorithm iteratively refines placement until forces are balanced, minimizing both overlap and wire length.
Analytical Placement	Formulates the placement problem as a mathematical optimization problem. The objective function typically includes wirelength and a density penalty to prevent overlaps. Techniques like gradient descent are used to approximate and solve the non-convex problem.
Machine Learning-Based Techniques	Recent approaches using machine learning, such as reinforcement learning and neural networks, predict optimal cell placements by learning from previous successful placements. These methods aim to enhance traditional techniques with data-driven predictions.

[#P3.1_Q3] What is simulated annealing in the context of the placement stage? explain how the algorithm works, what are the hyperparameters included?

A silicon chip's circuit components can be arranged optimally or almost optimally via simulated annealing.

The algorithm:

1. Begin with the chip's initial cell or component configuration, which could be random or determined by a heuristic.

2. Modify the existing arrangement slightly, for example, by moving two cells about. These changes result in the placement of new candidates.

3. Use a cost function to determine the new placement's cost. Usually, this function assesses factors like congestion and wire length.

4. Choose if you want to accept the new position:
a) Accept the new location if it results in a lower cost.
b) Accept the new location with a certain likelihood if it results in higher costs. A temperature parameter that drops with time determines probability.
5. Reduce the temperature parameter gradually in accordance with a cooling schedule. The algorithm is more inclined to accept less ideal placements in order to

investigate more options at higher temperatures.
6. Repeat

Hyperparameters:
1. Initial Temperature: the temperature's initial value. By tolerating inferior placements with a higher likelihood, a higher initial temperature enables the algorithm to investigate a greater variety of options.

2. Cooling Schedule: outlines the gradual drop in temperature. Typical schedules include logarithmic cooling, in which the temperature drops in accordance with a logarithmic function, or geometric cooling, in which the temperature is multiplied by a factor smaller than 1.

3. Cooling Rate: how quickly the temperature drops. Although it takes longer to compute, a slower cooling rate for more in-depth investigation.

4. Move Generation: the process of creating new positions for candidates. This can involve methods such as completing a cluster move, moving a cell to a nearby position, or switching two cells.

[#P3.3_Q1] Which type of timing violation can be more critical – setup or hold?

Because setup time violations might result in metastability—a situation in which the flip-flop records erroneous or undefinable data—and can create widespread logic faults in the circuit, they are typically regarded as more serious than hold time violations. When timing margins are limited in high-speed or high-frequency devices, this is especially problematic. Although they can nevertheless result in improper data latching and timing mistakes, hold time violations—which happen when data changes too quickly after the clock edge—are frequently less serious than setup time violations. To guarantee dependable functioning, both kinds of breaches must be carefully handled in digital circuit design. In conclusion, setup breaches are more serious since they have the potential to result in improper circuit operation.

The circuit may latch the incorrect data if the data is not received in time for the clock signal, which could cause problems for the entire system. Particularly in rapid circuits, it is crucial to make sure that data comes early enough (setup time) to ensure proper operation.

[#P3.3_Q2] Suggest 3 ways to fix a setup violation.

1) Extend the Clock Period: Data can stabilize before the clock edge if the clock is slowed down.
2) Optimize the Data Path: One way to help fulfill the setup time criteria is to reduce the amount of time it takes for data to move from one area of the circuit to the flip-flop.
3) Adding a register to the troublesome path reduces the amount of logic that must be processed in a single clock cycle by segmenting the lengthy path into smaller parts.

[#P3.3_Q3] How can a hold violation be created in a digital design.

When data changes too quickly after the clock signal, it becomes unstable and is not kept for the necessary amount of time, which is known as a hold violation. Here are several explanations on why this could happen:
1) Fast Data Path: The data may change before it is held for a sufficient amount of time if the path between two flip-flops is too quick or the delay is less than the required hold time.
2) Slow or Uneven Clock Edge: The timing of data capture may be impacted if the clock signal transitions slowly or unevenly. As a result, the data may alter excessively quickly in relation to the clock edge.

3) Clock Skew: Some flip-flops may receive the clock edge earlier than others if the clock signal reaches different areas of the circuit at different times. This may result in an early data capture and a hold violation.

[#P3.3_Q4] Suggest 3 ways to fix a hold violation.

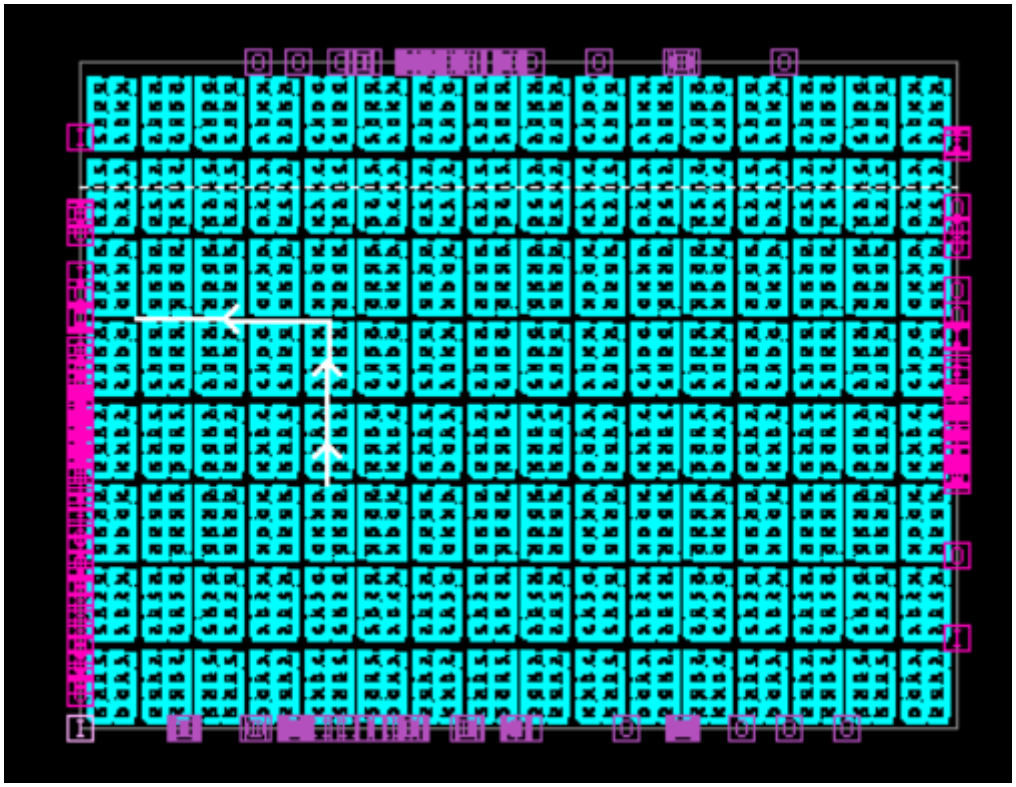
1) To guarantee that the data is maintained steadily for a longer amount of time and lower the possibility of a hold violation, extend the flip-flop's hold time.

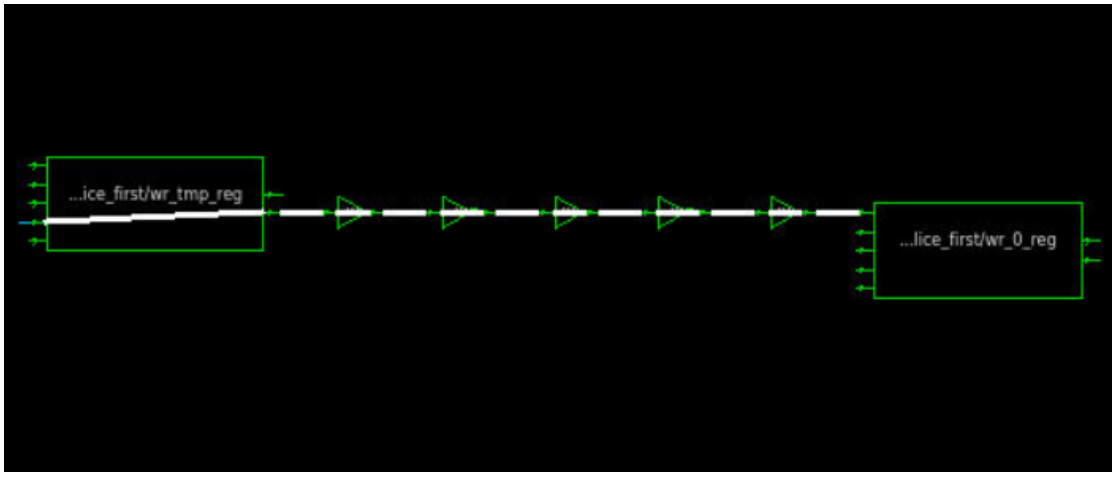
2) Data can change more slowly and remain stable for the necessary hold period by slowing down the data route between flip-flops.

3) Clock skew can be prevented by making sure the clock signal is more constant throughout the circuit.

[#P3.3_Q4] Analyze the timing reports extracted and find 2 problematic paths. For each path:
Path_1 (setup):

```
Startpoint: bit_secure_0/sipo_slice_first/wr_tmp_reg (rising edge-triggered flip-flop clocked by lclk)
Endpoint: bit_secure_6/sipo_slice_first/wr_0_reg (rising edge-triggered flip-flop 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_0/sipo_slice_first/wr_tmp_reg/CLK (SDFFASX1_RVT)
                                               0.00      0.00 r
bit_secure_0/sipo_slice_first/wr_tmp_reg/QN (SDFFASX1_RVT)
                                               0.13      0.13 r
place_optHFSBUF_400_89235/Y (NBUFFX8_RVT)      0.08      0.21 r
HFSINV_3087_47138/Y (INVX0_RVT)                0.05      0.26 f
place_optHFSBUF_465_93191/Y (NBUFFX16_RVT)     0.07      0.33 f
HFSINV_375_47133/Y (INVX0_RVT)                 0.11      0.43 r
place_optHFSBUF_47_87670/Y (NBUFFX8_RVT)       0.08      0.51 r
bit_secure_6/sipo_slice_first/wr_0_reg/D (SDFFARX1_RVT)
                                               0.02      0.53 r
data arrival time                                        0.53

clock lclk (rise edge)                         0.60      0.60
clock network delay (ideal)                    0.00      0.60
bit_secure_6/sipo_slice_first/wr_0_reg/CLK (SDFFARX1_RVT)
                                               0.00      0.60 r
library setup time                            -0.12      0.48
data required time                                       0.48
-----------------------------------------------------------------------
data required time                                       0.48
data arrival time                                       -0.53
-----------------------------------------------------------------------
slack (VIOLATED)                                        -0.05
```



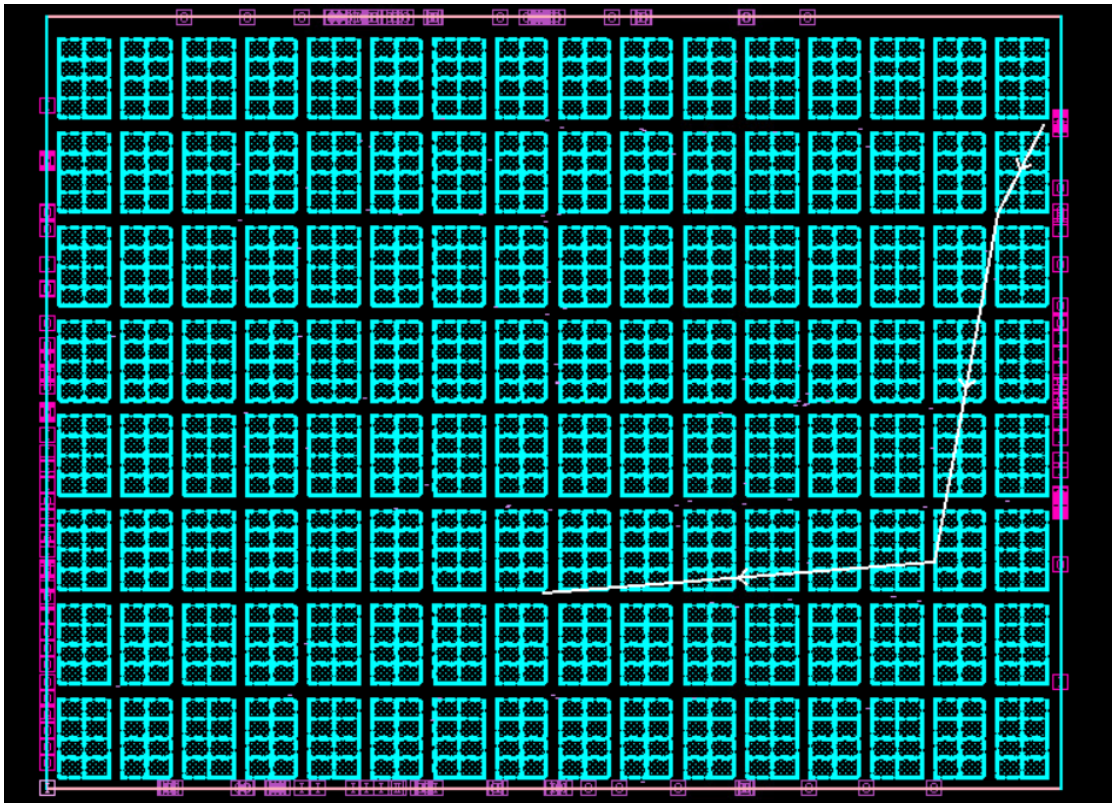


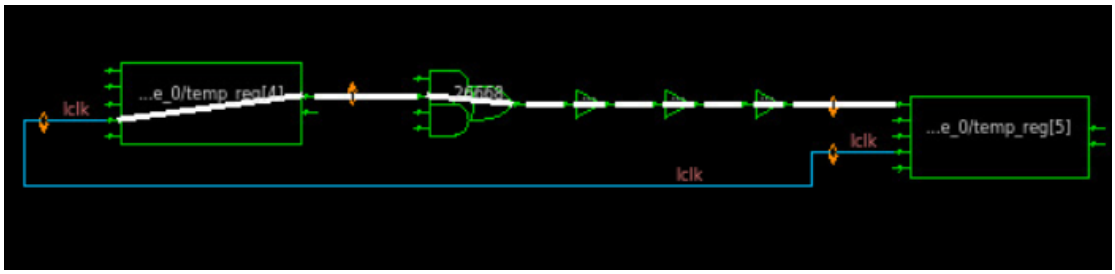
According to the timing report, there are five cells in the path (excluding the start and finish registers), the cell with the largest delay is HFSINV_375_47133/Y, with a delay of 0.11[ns], and we have a negative slack (-0.05). We may infer from the timing report that the slack was caused by the data arriving later than the specified arrival time due to the delay of all the cells in the path.

1) We can reduce the clock rate to address the issue, giving the data more time to traverse the path and preventing negative slack.
2) The data will arrive before the requisite arrival time if we add a register to the path to break it into two timing tracks, but we will now need to wait two clock cycles to obtain the data.

Path_2(hold):

```
Date   : Sat Mar  1 00:07:06 2025
****************************************

  Startpoint: piso_secure_0/temp_reg[4] (rising edge-triggered flip-flop clocked by lclk)
  Endpoint: piso_secure_0/temp_reg[5] (rising edge-triggered flip-flop 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

  piso_secure_0/temp_reg[4]/CLK (SDFFARX1_RVT)     0.00       0.00 r
  piso_secure_0/temp_reg[4]/Q (SDFFARX1_RVT)       0.14       0.14 r
  ctmi_26668/Y (AOI22X2_RVT)                     0.11       0.26 f
  place_optHFSINV_8075_83981/Y (IBUFFX16_RVT)      0.09       0.35 r
  place_optHFSBUF_7552_83980/Y (NBUFFX4_RVT)       0.12       0.47 r
  place_optHFSINV_7515_83979/Y (IBUFFX16_RVT)      0.09       0.56 f
  place_optHFSINV_2914_83978/Y (INVX8_RVT)         0.12       0.68 r
  piso_secure_0/temp_reg[5]/D (SDFFARX1_RVT)       0.02       0.70 r
  data arrival time                                           0.70

  clock lclk (rise edge)                         0.60       0.60
  clock network delay (ideal)                    0.00       0.60
  piso_secure_0/temp_reg[5]/CLK (SDFFARX1_RVT)     0.00       0.60 r
  library setup time                            -0.12       0.48
  data required time                                        0.48
  -----------------------------------------------------------------------
  data required time                                        0.48
  data arrival time                                        -0.70
  -----------------------------------------------------------------------
  slack (VIOLATED)                                         -0.23


1
fc shell>
```

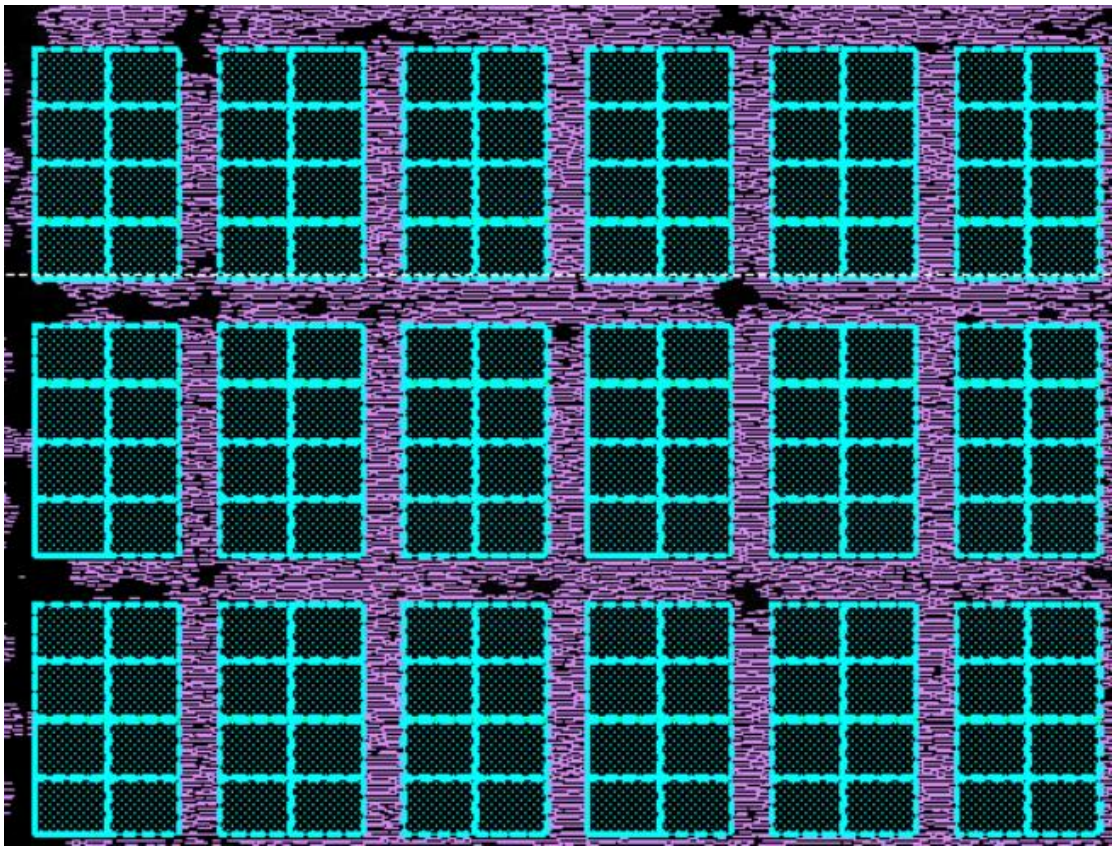


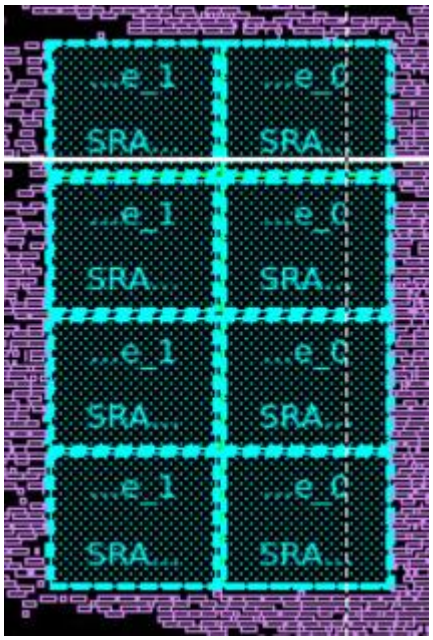


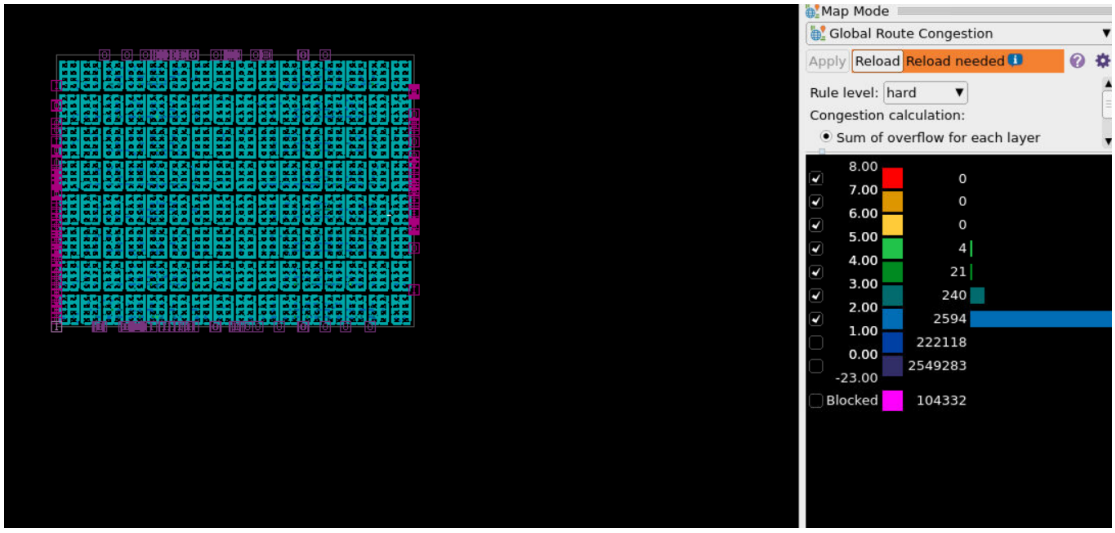
According to the timing report, we have a negative slack (-0.23), six cells on the path (excluding the start and end registers), and the two cells with the biggest delays are place_optHFSINV_2914_83978/Y and place_optHFSBUF_7552_83980/Y, both of which have delays of 0.12[ns]. We may infer from the timing report that the slack was caused by the data arriving later than the specified arrival time due to the delay of all the cells in the path.

1) We can solve the issue by lowering the clock rate, which will give the data more time to travel and prevent negative slack.
2) The data will arrive before the requisite arrival time if we add a register to the path to break it into two timing tracks, but we will now need to wait two clock cycles to obtain the data.

Appendix :






Map Mode
Global Route Congestion
Apply
Reload
Reload needed
Rule level:
hard
Congestion calculation:
Sum of overflow for each layer
8.00
0
7.00
0
6.00
0
5.00
4
4.00
21
3.00
240
2.00
2594
1.00
222118
0.00
2549283
-23.00
Blocked
104332