Advanced Vlsi Project

Bitcoin

Stage 5 - CTS

שמות המגישים	אור שאול	עמרי אורן	סטאניסלאב קוגן	נועה פרקש

Workarea path for Stage5: /project/advvlsi/users/orshaul/ws/bitcoin/stage5

3.1 Warmup Questions

[#P3.1_Q1]

Clock Tree Synthesis (CTS) is a crucial step in the physical design process of Very Large Scale Integration (VLSI) circuits. The main goal of CTS is to design and implement a clock distribution network that ensures the clock signal reaches all sequential elements (like flip-flops) with minimal skew and balanced delay.

Classic CTS

In classic CTS, the primary focus is on minimizing the clock skew, which is the difference in arrival times of the clock signal at various points in the circuit. This is important because skew can lead to timing issues, such as race conditions or setup and hold time violations. The clock tree is typically designed as a balanced binary tree, where buffers or inverters are inserted at strategic points to ensure that the clock signal reaches each leaf (flip-flop) simultaneously or with minimal delay difference.

CCD (Clock-Concurrent Design)

Clock-Concurrent Design (CCD) is an advanced CTS technique that integrates clock tree design with other design tasks, such as placement, routing, and optimization. The CCD approach aims to optimize the clock network in conjunction with the rest of the design, considering factors like power consumption, signal integrity, and overall timing closure. Unlike classic CTS, which treats clock tree synthesis as a separate step, CCD allows for more holistic and concurrent optimization, potentially leading to better overall performance and lower power consumption.

[#P3.1_Q2]

Clock buffers play a crucial role in the Clock Tree Synthesis (CTS) stage of VLSI design, where they are used to distribute the clock signal efficiently throughout the chip. Here are the primary roles and optimization considerations for clock buffers in CTS:

Roles of Clock Buffers in CTS

1. **Signal Strengthening:** Clock buffers strengthen the clock signal, ensuring it can drive multiple loads (like flip-flops and latches) across different parts of the chip without degradation. This is especially important in large designs where the clock signal needs to travel long distances.
2. **Skew Minimization:** One of the key objectives in CTS is to minimize clock skew—the difference in arrival times of the clock signal at different points in the circuit. Properly placed clock buffers help synchronize the arrival times of the clock signal, reducing skew and ensuring that all parts of the chip operate in harmony.
3. **Delay Management:** Clock buffers are used to manage and balance the delay of the clock signal. By carefully choosing the type, size, and placement of clock buffers, designers can control the delay to meet timing requirements and avoid timing violations.
4. **Load Balancing:** Clock buffers help in balancing the load seen by the clock tree. This is important for reducing power consumption and avoiding signal integrity issues, as uneven loading can lead to distortions in the clock signal.

Optimization of Clock Buffers

To optimize clock buffers in the CTS stage, the following strategies are typically employed:

1. **Buffer Sizing and Insertion:** The size and number of clock buffers are carefully chosen to balance the load and control the delay. Larger buffers can drive more load but consume more power and may introduce additional delay, so a trade-off must be made.
2. **Location Optimization:** The placement of clock buffers is crucial for minimizing skew and delay. Tools used in CTS perform algorithms that place buffers at strategic points in the clock tree to ensure balanced signal distribution.
3. **Power and Area Considerations:** Clock buffers consume power, so their number and size are optimized to meet power budgets. Additionally, minimizing the area occupied by clock buffers is important in densely packed designs.
4. **Clock Gating:** To reduce dynamic power consumption, clock gating can be implemented. This involves controlling clock buffers so that they only drive the clock signal when needed, thereby reducing unnecessary switching activity.

[#P3.1_Q3]

Skew – Global and Local

Skew: Difference in clock arrival time at two different registers.

Global Skew: Global skew refers to the biggest latency difference between 2 synchronous devices (like 2 flip-flops) across the entire chip or system. It is critical in determining the overall timing and synchronization of the circuit. Large global skew can lead to timing errors, such as setup or hold time violations, across different parts of the circuit.

Local Skew: Local skew is the biggest latency difference between 2 synchronous devices (like 2 flip-flops), **that talk to each other.** Local skew is typically controlled by precise routing and buffering within specific areas of the design.

Jitter

Jitter refers to the variation in the timing of the clock signal's edge transitions. It is essentially the uncertainty in the arrival time of a clock edge, measured over a period, meaning the variation in the clock period over a long sequence of clock cycles.

Jitter can cause timing errors in synchronous systems, where circuits depend on precise timing for data sampling and processing. It can degrade the performance and reliability of the system. Jitter can arise from various sources, including noise in the power supply, electromagnetic interference, and variations in the manufacturing process.

PLL (Phase-Locked Loop)

A Phase-Locked Loop (PLL) is a control system that generates a clock signal whose phase is locked to the phase of an input reference signal. It is used to synchronize a clock signal in one part of a system with a clock signal in another part.

Components:

1. **Phase Detector:** Compares the phase of the input signal with the output of the VCO (Voltage-Controlled Oscillator).
2. **Low Pass Filter:** Removes high-frequency components from the phase detector output, providing a smooth control voltage.
3. **Voltage-Controlled Oscillator (VCO):** Generates an output signal whose frequency is controlled by the input voltage from the low pass filter.
4. **Feedback Path:** The output of the VCO is fed back to the phase detector to form a closed loop.

PLLs are widely used in clock generation and synchronization, frequency synthesis, and modulation/demodulation in communication systems. They help in reducing jitter and maintaining clock signal integrity across different parts of a circuit or system.

[#P3.1_Q4]

H-Tree Clock Distribution Network

Description: The H-tree is a hierarchical clock distribution network used in VLSI design to distribute the clock signal uniformly across a chip. The network resembles the letter "H" and is recursively repeated, splitting into smaller H-shaped structures at each level. The root of the H-tree starts at the clock source and branches out to various endpoints, typically clock buffers or flip-flops, at the leaves.

Pros:

1. **Low Skew:** The H-tree design inherently ensures that all paths from the clock source to the sinks are of equal or nearly equal length, which helps to minimize clock skew. This uniformity ensures that the clock signal reaches all parts of the chip almost simultaneously.
2. **Smallest Routing Capacitance:** The H-tree structure can lead to minimized wire length for a given set of clock sinks, resulting in reduced overall routing capacitance.
3. **Low Power Consumption:** With minimized routing capacitance, the H-tree also tends to have lower power consumption since less capacitance results in lower dynamic power dissipation.

Cons:

1. **Poor Floorplan Flexibility:** The rigid and symmetrical nature of the H-tree can be less adaptable to varying chip layouts or non-uniform distribution of clock sinks. This can lead to suboptimal use of space and difficulty in integrating the clock network with other elements of the design, potentially causing inefficiencies in floorplanning.

2. **Limited Scalability with Very Large Designs:** As the size of the chip increases, the complexity of the H-tree grows significantly. This can lead to increased design time and difficulty in managing the clock network, especially in very large or highly complex circuits.

3. **Maintenance of Symmetry:** While symmetry in an H-tree clock distribution network helps minimize skew, maintaining this symmetry can be challenging, especially in larger circuits or designs with an irregular distribution of clock sinks or varying block sizes. Deviations from the ideal symmetric structure can introduce additional skew and complicate routing, potentially impacting the overall performance and reliability of the clock distribution network.

[#P3.1_Q5]

In the Clock Tree Synthesis (CTS) stage of VLSI design, shielding and spacing are techniques used to improve signal integrity, minimize interference, and ensure reliable clock signal distribution. Here's an overview of each technique:

Shielding

Description: Shielding involves placing grounded wires or additional metal layers adjacent to the clock signal lines. This technique is used to protect the clock signal from external noise and to prevent it from inducing noise on neighbouring signals.

Purpose and Benefits:

1. **Noise Reduction:** Shielding helps in minimizing crosstalk noise, which can occur when a high-frequency clock signal induces unwanted signals in nearby wires. This is particularly important in high-speed designs where signal integrity is crucial.
2. **Signal Integrity:** By surrounding clock lines with grounded shields, the likelihood of signal degradation due to electromagnetic interference (EMI) is reduced, ensuring that the clock signal remains clean and stable.
3. **Reduced Skew and Jitter:** Shielding can also help in reducing clock skew and jitter by maintaining a more consistent and controlled environment for the clock signals.

Spacing

Description: Spacing refers to the practice of maintaining adequate distance between clock signal wires and other signal wires in the design. This technique is used to reduce the effects of crosstalk and to minimize the interference between signals.

Purpose and Benefits:

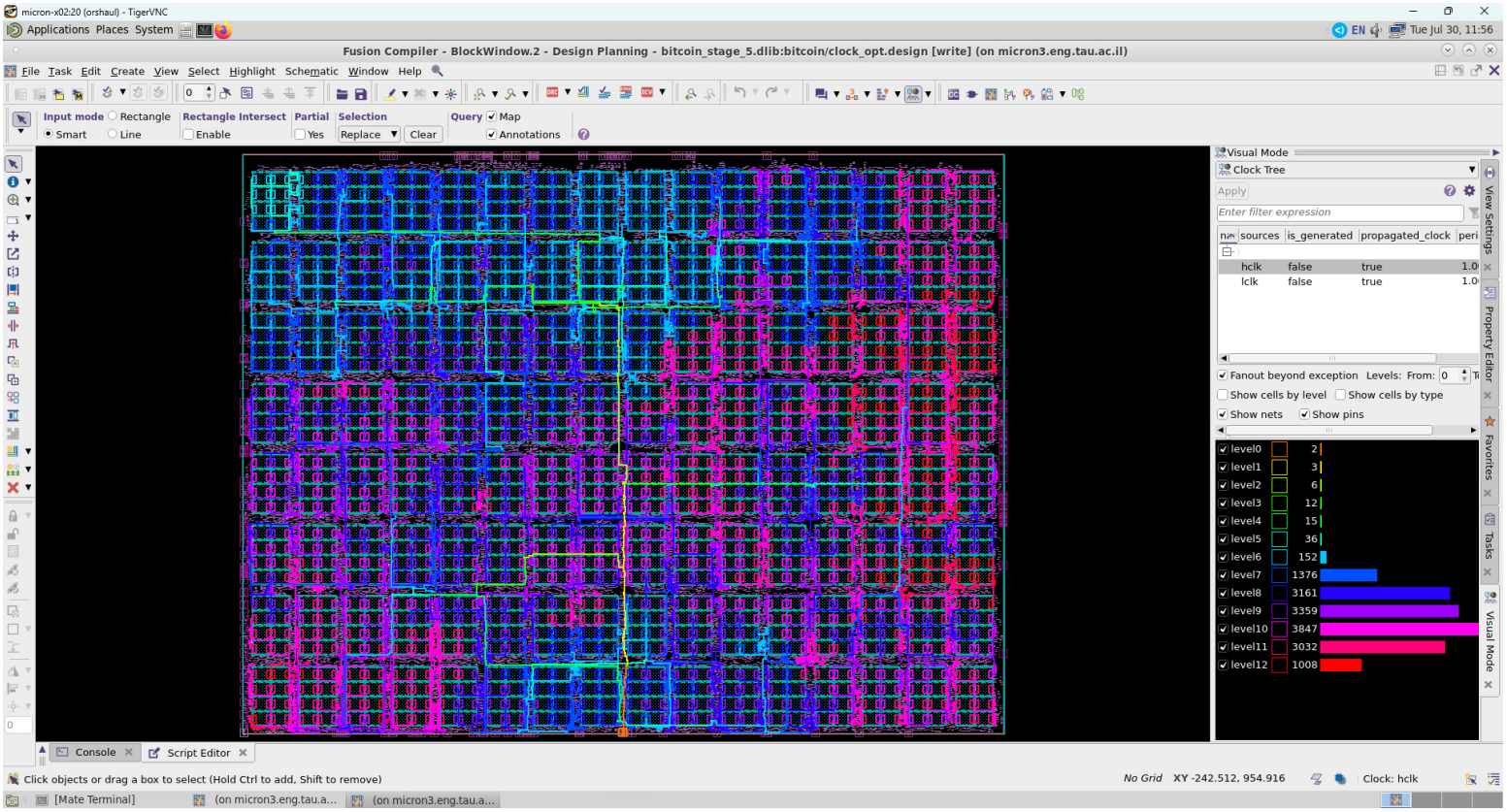
1. **Crosstalk Mitigation:** By increasing the distance between clock lines and other signals, spacing reduces the likelihood of crosstalk, where signals on adjacent wires can interfere with each other. This is critical in ensuring the integrity of the clock signal and the overall timing performance of the circuit.
2. **Reduced Capacitance Coupling:** Proper spacing reduces capacitive coupling between adjacent wires, which can affect signal speed and cause timing issues.
3. **Power and Delay Management:** Adequate spacing can help manage the delay introduced by signal interactions and can also contribute to reducing power consumption by minimizing unwanted signal activity.

Shielding and spacing are essential techniques in the CTS stage to ensure the reliable distribution of the clock signal. Shielding helps protect the clock signal from noise and interference, while spacing minimizes crosstalk and other signal integrity issues. Both techniques are crucial for maintaining the performance and reliability of high-speed digital circuits.

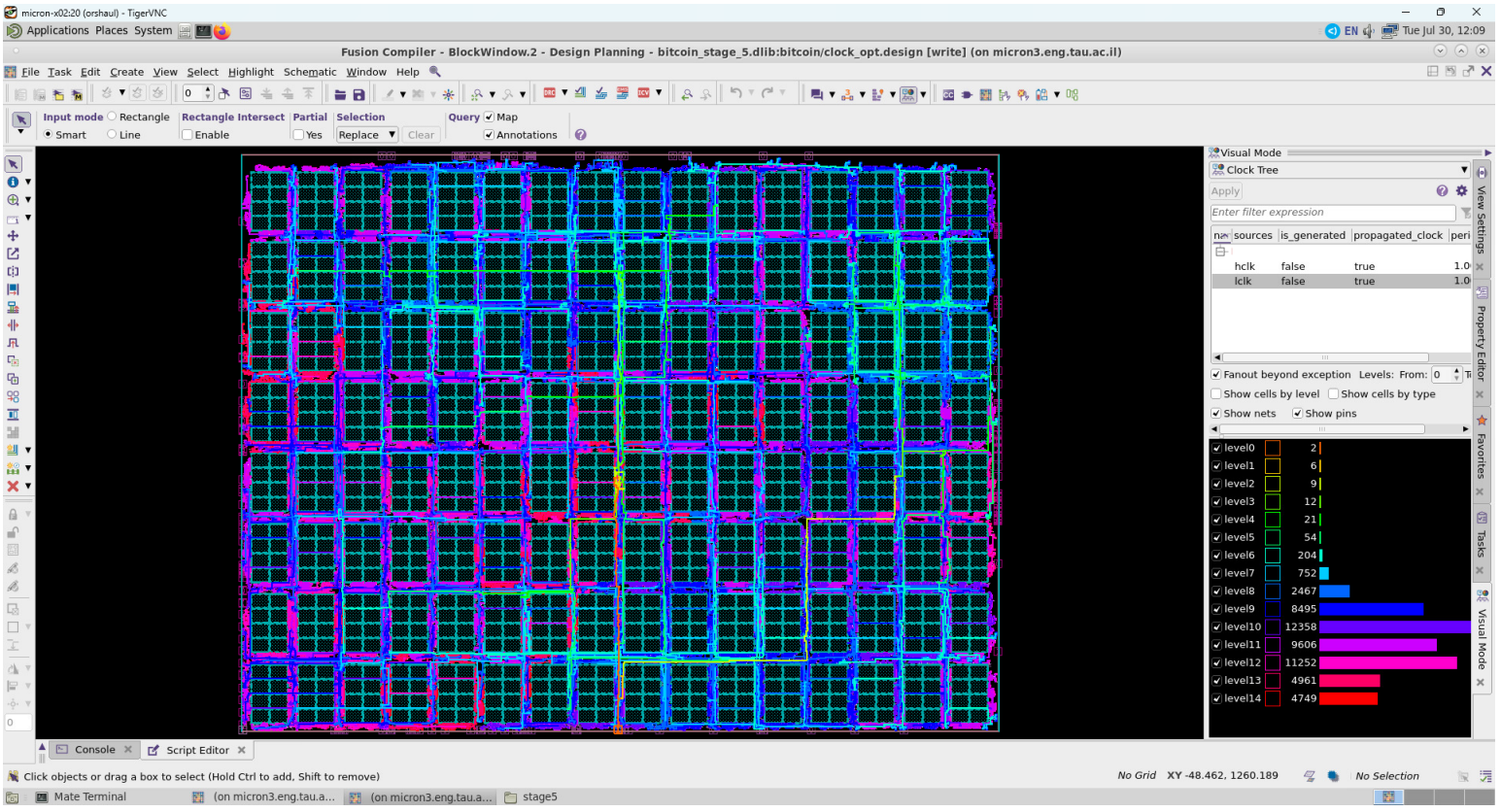
[#P3.2_Q1]

Print screen of the clock tree distribution

For hclk:



For lclk:



[#P3.2_Q2]

```
fc_shell> report_clock_timing -clock [get_clocks lclk] -type skew
****************************************
Report : clock timing
        -type skew
        -nworst 1
        -setup
Design : bit_coin
Version: V-2023.12-SP3
Date   : Tue Jul 30 12:36:41 2024
****************************************
Scenario FUNC_Fast is not configured for setup or hold analysis
Scenario FUNC_Slow is not configured for setup or hold analysis

  Mode: FUNC
  Clock: lclk

  Clock Pin                                          Latency      Skew             Corner
-------------------------------------------------------------------------------------------------
  bit_secure_10/slice_30/piso_bit/temp_reg[10]/CLK      0.70             rp-+    Typical
  bit_secure_10/slice_30/piso_bit/temp_reg[11]/CLK      0.41      0.29    rp-+    Typical

-------------------------------------------------------------------------------------------------
1
fc_shell>
```

Log History

fc_shell>

Console Script Editor

The skew of lclk is 0.29 [nsec].

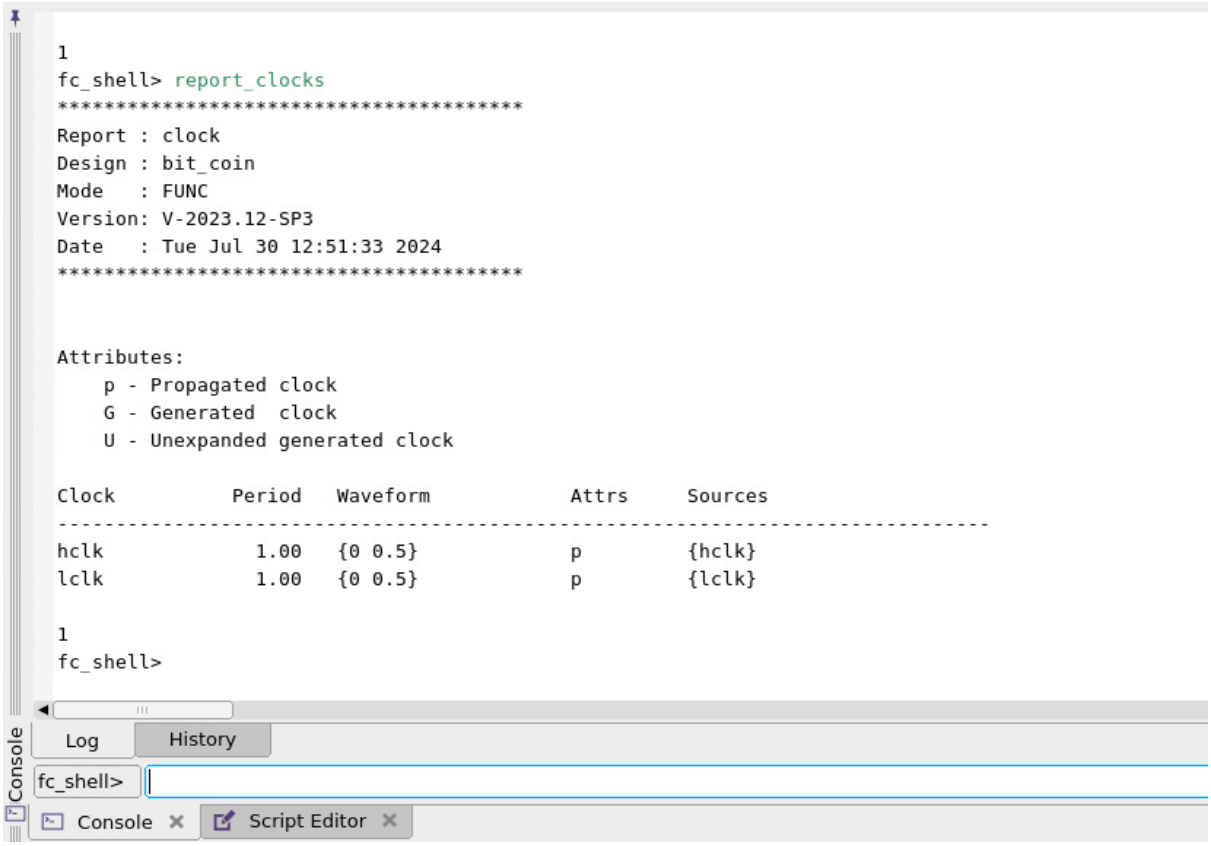
```
fc_shell> report_clock_timing -clock [get_clocks hclk] -type skew
****************************************
Report : clock timing
        -type skew
        -nworst 1
        -setup
Design : bit_coin
Version: V-2023.12-SP3
Date   : Tue Jul 30 12:46:02 2024
****************************************
Scenario FUNC_Fast is not configured for setup or hold analysis
Scenario FUNC_Slow is not configured for setup or hold analysis

  Mode: FUNC
  Clock: hclk

  Clock Pin                                            Latency      Skew             Corner
-------------------------------------------------------------------------------------------------
  bit_secure_3/slice_0/sync_data2mem/dut_sync/sync_out_reg[7]/CLK
                                                          0.46             rp-+    Typical
  bit_secure_3/slice_0/nibble_0/CE2                       0.24      0.22    rp-+    Typical

-------------------------------------------------------------------------------------------------
1
```

Log History

fc_shell>

Console Script Editor

The skew of hclk is 0.22 [nsec].



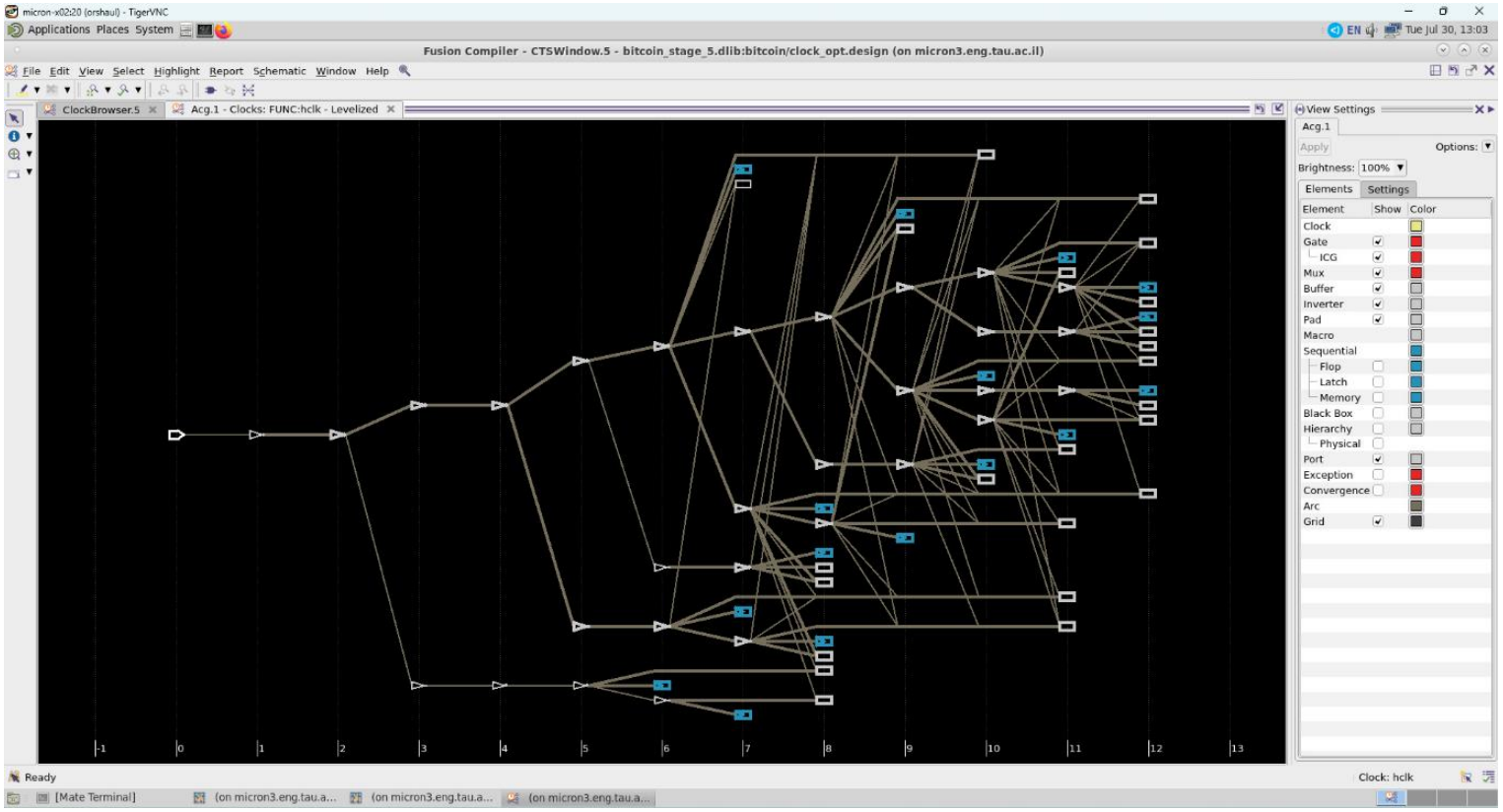
We can see that the period time T = 1 [nsec].

Therefore, in my opinion, the skew of lclk and hclk is quite high because it means their skew is 22% and 29% from the period time T and it can lead to timing problems, especially in high-frequency systems where every nanosecond counts.

[#P3.2_Q3]

Clock Tree Levelized Graph

Print screen of the hclk clock tree:



print screen of the lclk clock tree: