FFT Design Report

1. Introduction

Fast Fourier Transform (FFT) is an efficient algorithm to compute the Discrete Fourier Transform (DFT) and its inverse. The FFT algorithm reduces the computational complexity of DFT from O(N^2) to O(N log N), making it highly efficient for signal processing applications.

2. RTL Design

https://github.com/PRANJAVERMA/FFT/tree/main

fft_8pt.v & fft_8pt_Tb.v

Tools Used: Vivado

Description:

The RTL design for the FFT module includes the following submodules:

- Butterfly Processing Unit

- Twiddle Factor Multiplication

- Bit-Reversal Logic

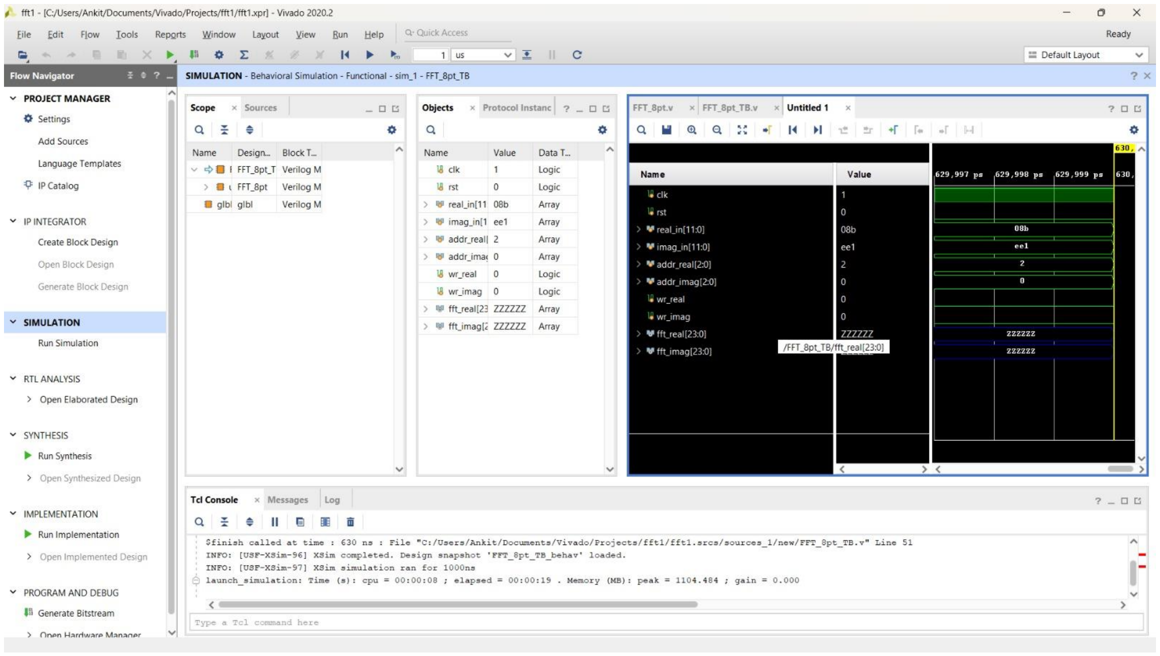
- Control FSM for stage and data flow management

Constructs Observed:

- Arithmetic operations (adders, multipliers) inferred from fixed-point logic

- FSM inferred from case statements and if-else structures

- Registers and logic inferred from always_ff and always_comb



fft1 - [C:/Users/Ankit/Documents/Vivado/Projects/fft1/fft1.xpr] - Vivado 2020.2
SIMULATION - Behavioral Simulation - Functional - sim_1 - FFT_8pt_TB
Flow Navigator
PROJECT MANAGER
IP INTEGRATOR
SIMULATION
Run Simulation
RTL ANALYSIS
SYNTHESIS
IMPLEMENTATION
PROGRAM AND DEBUG

Name	Value	Data Type
clk	1	Logic
rst	0	Logic
real_in[11:0]	08b	Array
imag_in[11:0]	ee1	Array
addr_real[2:0]	2	Array
addr_imag[2:0]	0	Array
wr_real	0	Logic
wr_imag	0	Logic
fft_real[23:0]	ZZZZZZ	Array
fft_imag[23:0]	ZZZZZZ	Array

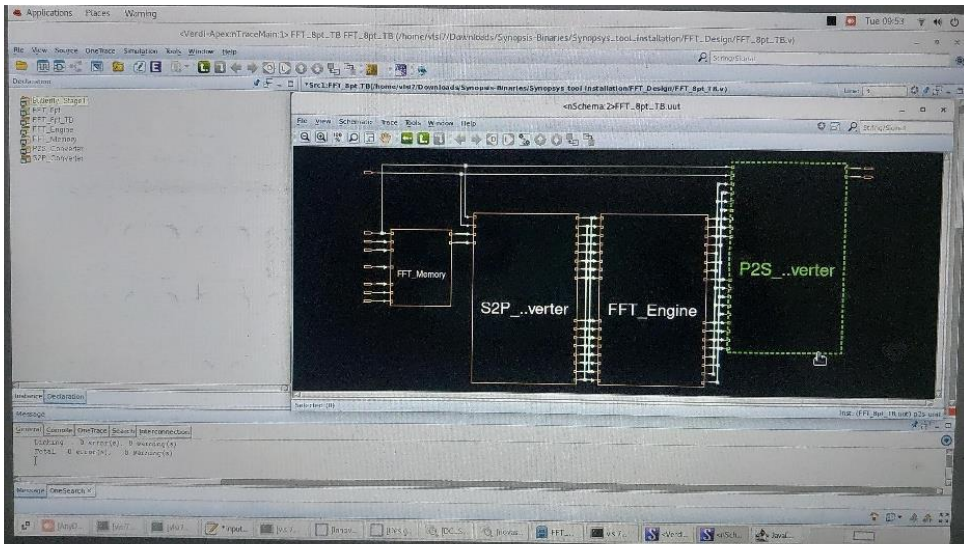
```
$finish called at time : 630 ns : File "C:/Users/Ankit/Documents/Vivado/Projects/fft1/fft1.srcs/sources_1/new/FFT_8pt_TB.v" Line 51
INFO: [USF-XSim-96] XSim completed. Design snapshot 'FFT_8pt_TB_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:08 ; elapsed = 00:00:19 . Memory (MB): peak = 1104.484 ; gain = 0.000
```

3. Pre-Synthesis Simulation

Tool Used: Synopsys VCS, Verdi , nclaunch

Description:

The pre-synthesis simulation validates the functional correctness of the FFT RTL design, ensuring all logic paths and data flows are accurately represented before synthesis.

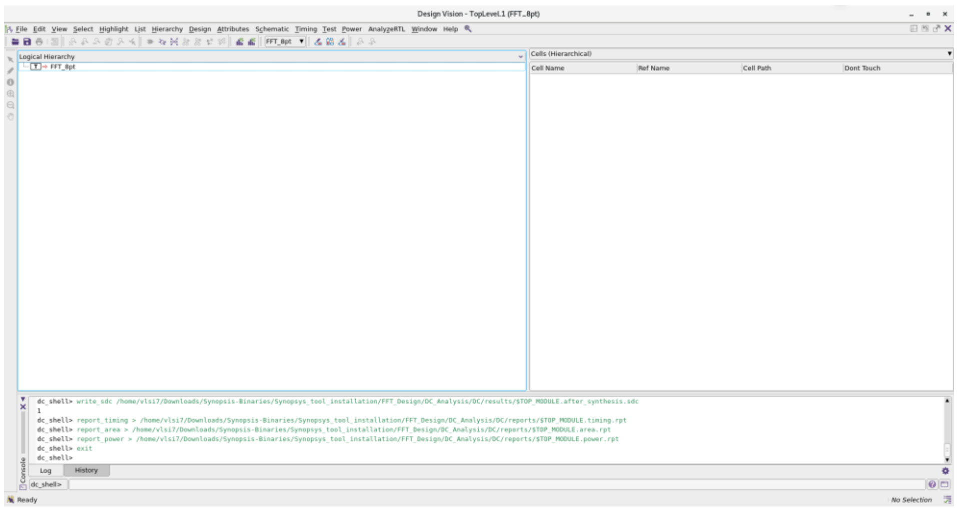


4. Synthesis

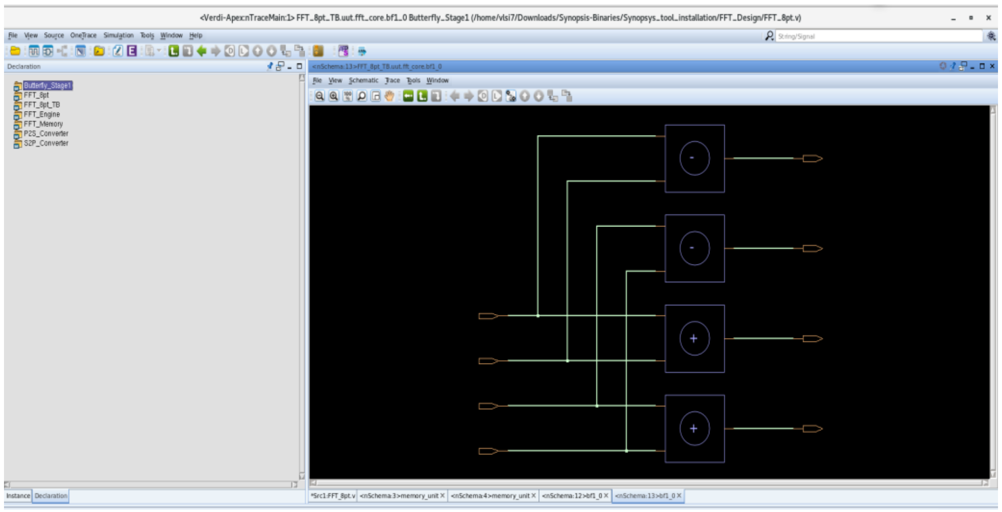
Tool Used: Synopsys Design Compiler

Description:

The RTL is synthesized into a gate-level netlist, applying timing constraints and optimizations. Key parameters include clock constraints, area optimization, and critical path analysis.



```
dc_shell> write_sdc /home/vlsi7/Downloads/Synopsis-Binaries/Synopsys_tool_installation/FFT_Design/DC_Analysis/DC/results/$TOP_MODULE.after_synthesis.sdc
1
dc_shell> report_timing > /home/vlsi7/Downloads/Synopsis-Binaries/Synopsys_tool_installation/FFT_Design/DC_Analysis/DC/reports/$TOP_MODULE.timing.rpt
dc_shell> report_area > /home/vlsi7/Downloads/Synopsis-Binaries/Synopsys_tool_installation/FFT_Design/DC_Analysis/DC/reports/$TOP_MODULE.area.rpt
dc_shell> report_power > /home/vlsi7/Downloads/Synopsis-Binaries/Synopsys_tool_installation/FFT_Design/DC_Analysis/DC/reports/$TOP_MODULE.power.rpt
dc_shell> exit
dc_shell>
```

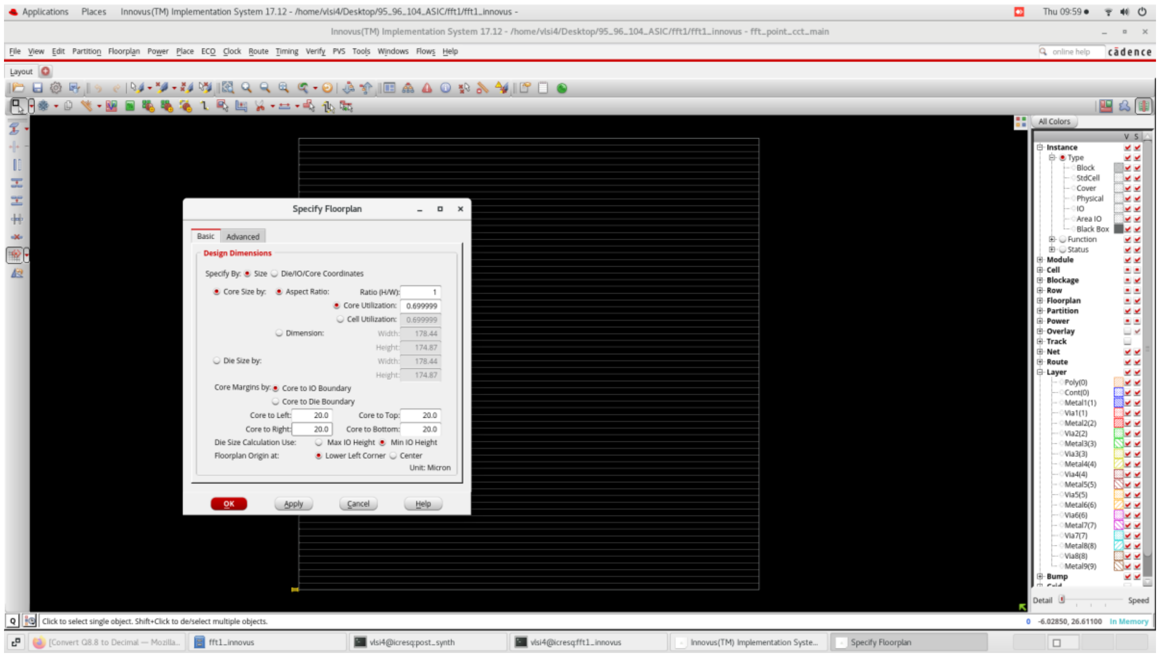

<Verdi-Apex:nTraceMain:1> FFT_8pt_TB.uut.fft_core.bf1_0 Butterfly_Stage1 (/home/vlsi7/Downloads/Synopsis-Binaries/Synopsys_tool_installation/FFT_Design/FFT_8pt.v)
Declaration
Butterfly_Stage1
FFT_8pt
FFT_8pt_TB
FFT_Engine
FFT_Memory
P2S_Converter
S2P_Converter
-
-
+
+
Instance
Declaration

5. Floorplanning

Tool Used: Cadence Innovus

Description:

The initial placement of macros, I/O pads, and standard cells is defined, optimizing the core area utilization and routing paths.



Innovus(TM) Implementation System 17.12 - /home/vlsi4/Desktop/95_96_104_ASIC/fft1/fft1_innovus - fft_point_cct_main

Specify Floorplan

Basic | Advanced

Design Dimensions

Specify By: [✓ Size] Die/IO/Core Coordinates

[✓ Core Size by:] [✓ Aspect Ratio:] Ratio (H/W): 1

[✓ Core Utilization:] 0.699999

Cell Utilization: 0.699999

Dimension: Width: 178.44 Height: 174.87

Die Size by: Width: 178.44 Height: 174.87

Core Margins by: [✓ Core to IO Boundary] Core to Die Boundary

Core to Left: 20.0 Core to Top: 20.0

Core to Right: 20.0 Core to Bottom: 20.0

Die Size Calculation Use: Max IO Height [✓ Min IO Height]

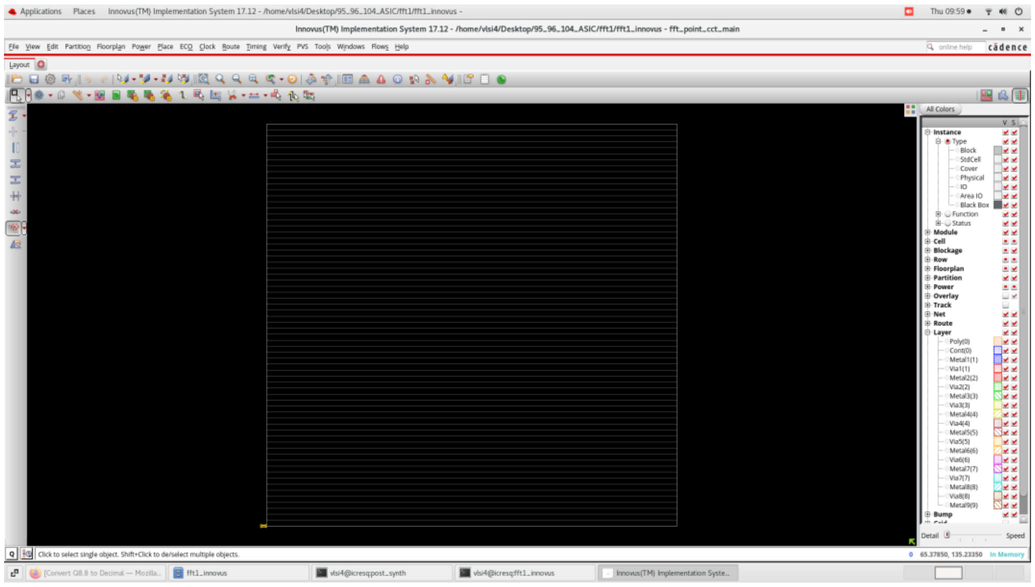
Floorplan Origin at: [✓ Lower Left Corner] Center

Unit: Micron

OK Apply Cancel Help

Click to select single object. Shift+Click to de/select multiple objects.

-6.02850, 26.61100 In Memory



Innovus(TM) Implementation System 17.12 - /home/vlsi4/Desktop/95_96_104_ASIC/fft1/fft1_innovus - fft_point_cct_main

Click to select single object. Shift+Click to de/select multiple objects.

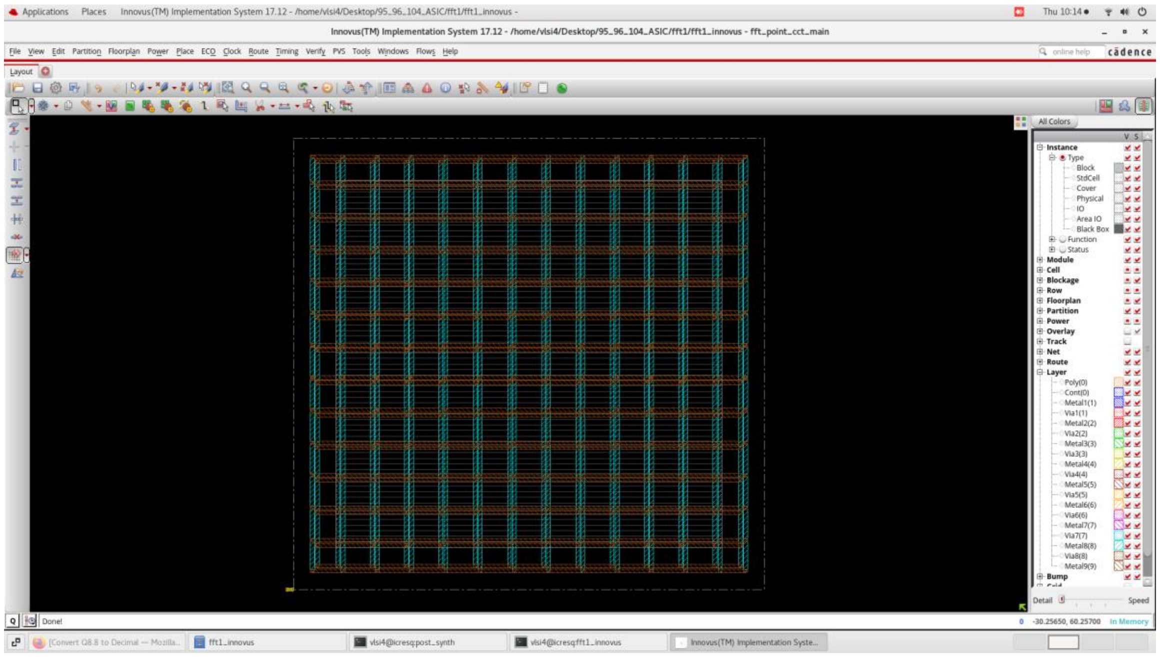
65.37850, 135.23350 In Memory

6. Power Planning

Tool Used: Innovus

Description:

Power rings and stripes are defined to ensure even power distribution across the design, preventing IR drops and electromigration.



7. Placement

Tool Used: Innovus

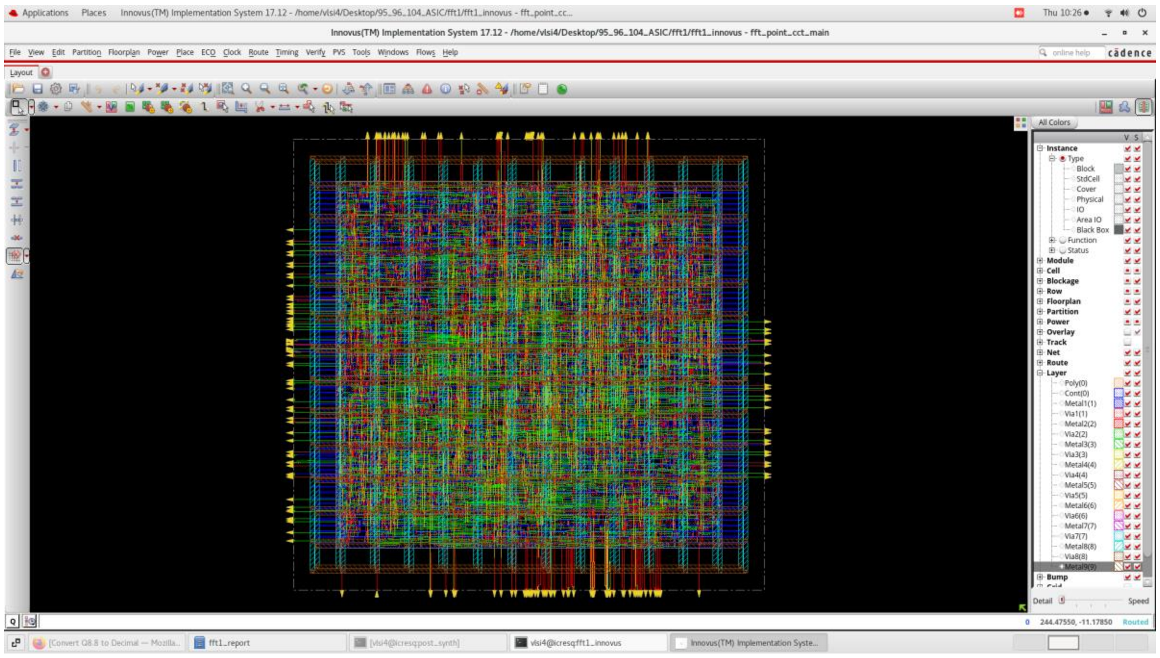
Description: Placement optimizes the locations of standard cells, macros, and I/O pads for minimal wire length and optimal timing.

Steps Involved:

1. **Global Placement:** Initial cell layout.
2. **Detailed Placement:** Overlap removal and optimization.
3. **Legalization:** Ensures cells follow design rules.

Commands:

- place_opt: Optimize placement.
- legalize: Legalize cell positions.
- check_placement: Verify placement.


Innovus(TM) Implementation System 17.12 - /home/vlsi4/Desktop/95_96_104_ASIC/fft1/fft1_innovus - fft_point_cct_main
File View Edit Partition Floorplan Power Place ECO Clock Route Timing Verify PVS Tools Windows Flows Help
cadence
Layout
All Colors
Instance
Module
Cell
Blockage
Row
Floorplan
Partition
Power
Overlay
Track
Net
Route
Layer
Bump
Detail
Speed
244.47550, -11.17850
Routed

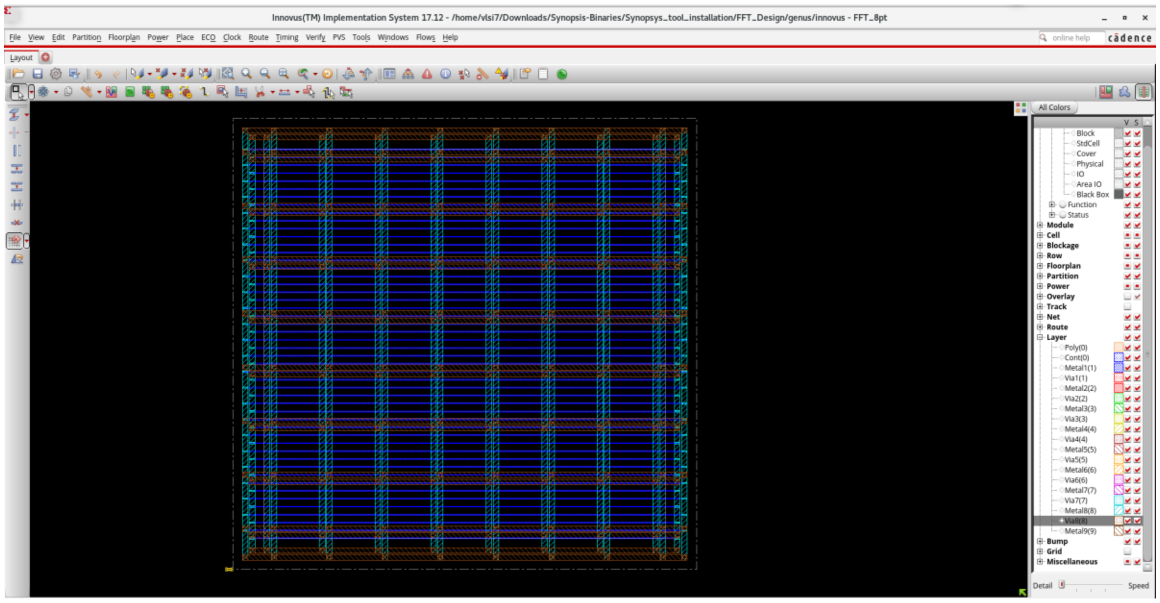
-
-

8. Routing

Tool Used: Innovus

Description:

Routing of the design is completed with special attention to clock tree synthesis and signal integrity.



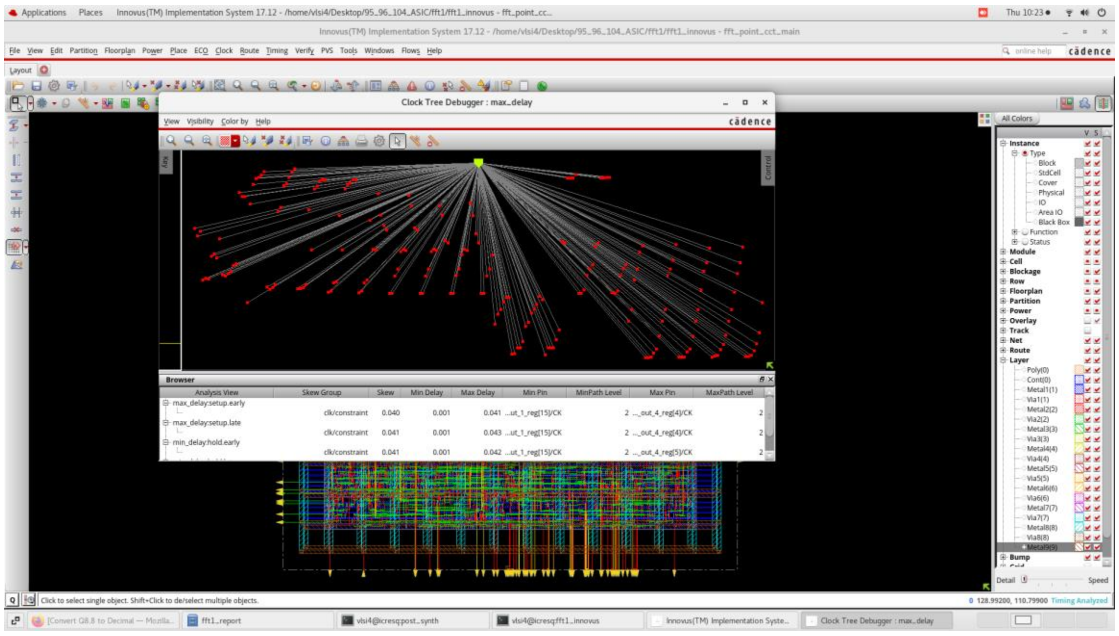
9. Clock Tree Synthesis (CTC)

Tool Used: Innovus

Description: Clock Tree Synthesis (CTS) is performed to distribute the clock signal evenly across all sequential elements, minimizing skew and ensuring synchronization.

Observed Structure: The clock tree consists of buffer stages and clock gating cells, optimized to reduce latency and maintain timing integrity.

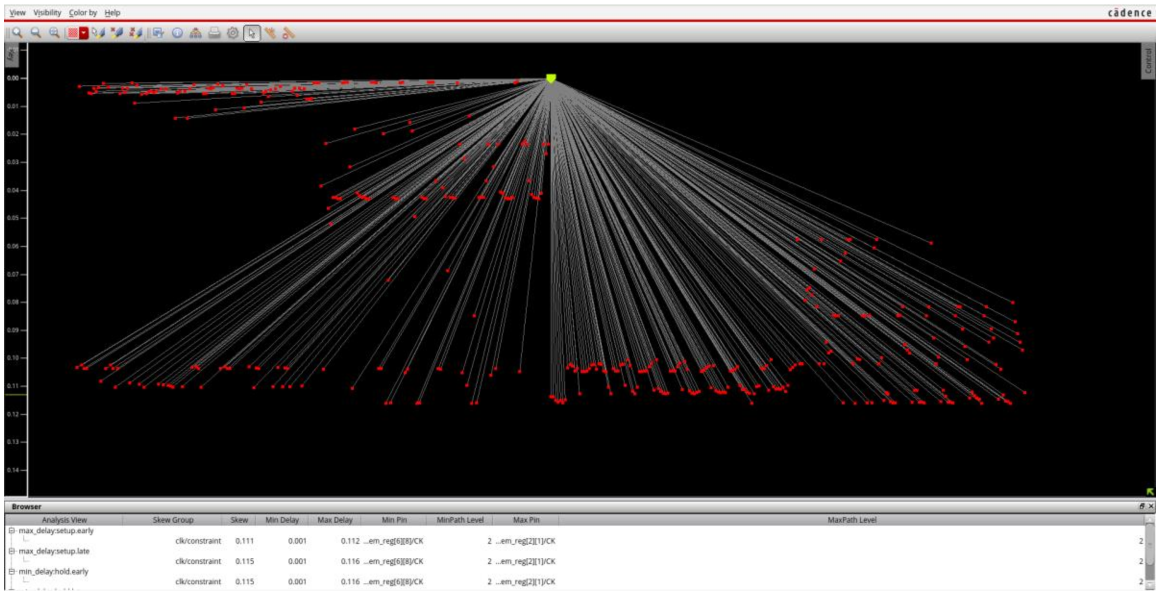
Analysis: During CTS, constraints such as clock uncertainty and latency are adjusted to optimize timing closure. Varying these parameters impacts the clock skew and timing path delay.



Clock Tree Debugger : max_delay
View Visibility Color by Help
cadence
Browser

Analysis View	Skew Group	Skew	Min Delay	Max Delay	Min Pin	MinPath Level	Max Pin	MaxPath Level
max_delay:setup.early	clk/constraint	0.040	0.001	0.041	...ut_1_reg[15]/CK	2	..._out_4_reg[4]/CK	2
max_delay:setup.late	clk/constraint	0.041	0.001	0.043	...ut_1_reg[15]/CK	2	..._out_4_reg[4]/CK	2
min_delay:hold.early	clk/constraint	0.041	0.001	0.042	...ut_1_reg[15]/CK	2	..._out_4_reg[5]/CK	2

Click to select single object. Shift+Click to de/select multiple objects.



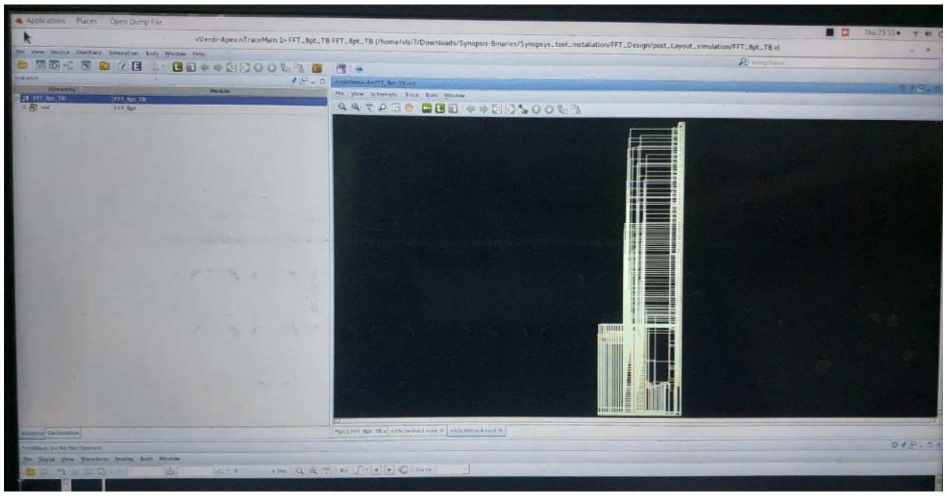
Analysis View	Skew Group	Skew	Min Delay	Max Delay	Min Pin	MinPath Level	Max Pin	MaxPath Level
max_delay:setup.early	clk/constraint	0.111	0.001	0.112	...em_reg[6][8]/CK	2	...em_reg[2][1]/CK	2
max_delay:setup.late	clk/constraint	0.115	0.001	0.116	...em_reg[6][8]/CK	2	...em_reg[2][1]/CK	2
min_delay:hold.early	clk/constraint	0.115	0.001	0.116	...em_reg[6][8]/CK	2	...em_reg[2][1]/CK	2

10. Post-Synthesis Simulation

Tool Used: NCLaunch

Description:

Simulation is performed on the synthesized netlist to verify timing correctness and functional integrity.



dc-shell schematic