ECE 260B Final Project Report

Group: Edge States

Anjana Manoj, Ashwin Manoj, Devansh Gupta, Lingzhan Xu, Mark Sui, Riyansh Chaturvedi

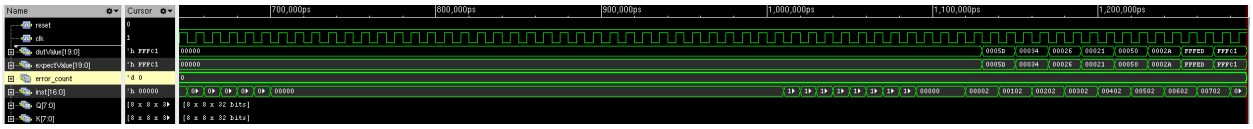
Introduction

This report presents the design, synthesis, and optimization of a custom hardware accelerator for the Attention Mechanism in Transformer models. The primary objective of this project is to implement an efficient vector processor architecture at the Register Transfer Level (RTL) to execute Query (Q) and Key (K) matrix multiplications, followed by a hardware-based output normalization stage. Through a multi-step design space exploration, the architecture is incrementally scaled from a baseline single-core matrix multiplier to a hierarchically synthesized dual-core system, leveraging asynchronous protocols for cross-domain data synchronization. Finally, the design is aggressively optimized for Power, Performance, and Area (PPA) through advanced microarchitectural techniques to maximize operating frequency and minimize dynamic power consumption.

Step 1. Single Core RTL,Synthesis & PNR

Implementation

Implements a parameterized multiply-accumulate (MAC) hardware core that integrates query, key, and partial sum SRAMs with a MAC array and an output FIFO to calculate and store vector operations.



Performance Analysis

Area (mm^2)	Total Power(mW)	Switching Power(mW)	WNS(ns)
0.317	181.742	1.708	-0.391

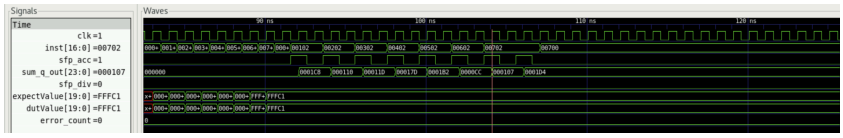
Step 2. Output Normalization

Implementation

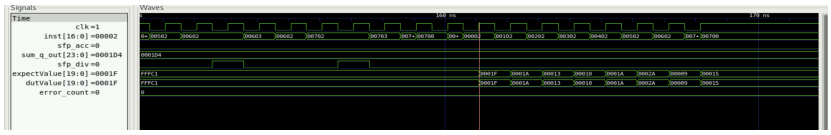
- Regional Summation: Upon the sfp_acc pulse, 8 values from pmem_out are converted to absolute values and summed to create a regional sum_q.

- Dual Buffering: The sum_q is written to two FIFOs: one for local core processing (sum_this_core) and one for the partner core (sum_out).Similarly, the sfp_row receives the neighbor's sum via sum_in (tied to 0 in Single_Core case), calculating a combined sum_2core total.
- Scaling: The system uses the upper bits of this total to perform a shift-based scaling operation prior to division.
- Normalization: When sfp_div is pulsed, the 8 original absolute values are divided by the sum_2core and driven to the sfp_out port.
- Output Selection: A multiplexer selects between the raw MAC array output and the processed sfp_out for the final write-back to the pmem SRAM.

Accumulation Step



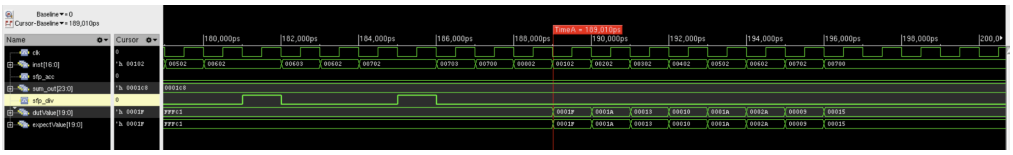
Normalization Step



Step 3. *Hierarchical Synthesis of core.*

Implementation

The SRAM PNR is implemented first and then used as a macro for the core design, finally merging the gds files into core gds.



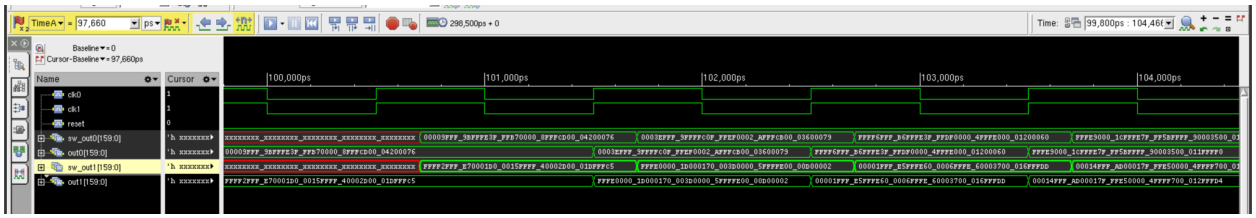
Performance Analysis

Area (mm^2)	Total Power(mW)	WNS(ns)	DRC Violation	LVS Violations
3.568	182.29	-1.025 (Setup)	2	0

Step 4. Hierarchical Synthesis of dual core.

Implementation

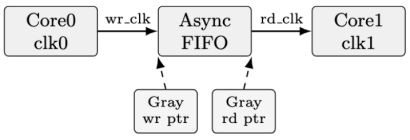
The fullchip implementation is an asynchronously-clocked dual-core Systolic Attention Transformer designed to perform dot-product array operations and normalization across independent clock domains. Each core encapsulates its own MAC array, SRAM banks, output FIFOs, and a dedicated Systolic Filter Processor (SFP) block. To bridge the independent clock domains (clk0 and clk1), the top-level fullchip module instantiates asynchronous, multi-depth cross-core FIFOs that safely handle Clock Domain Crossing (CDC) data transmission. During the normalization phase, each core independently calculates its local partial sums and pushes them through the CDC FIFOs to the neighboring core; the SFP blocks then pop these synchronized values (sum_in) to compute a global shared denominator (sum_2core) and dynamically divide their local outputs before writing the final normalized results back to PSUM memory.



Step 5. Optimization

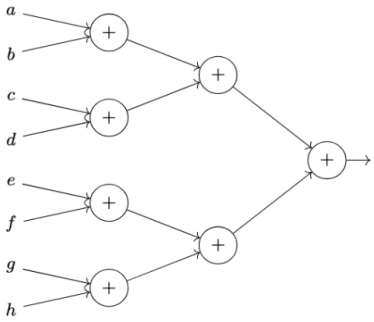
Implementation

1. Redesigned the cross-core SFP sum-sharing path by adding a dedicated asynchronous FIFO in **fifo_async_depth16.v** and a separate read clock as sum_rd_clk in **sfp_row.v**. Decouples two clock domains and uses Gray-coded pointer to synchronization to ensure data transfer across two cores.



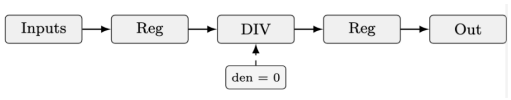
Cross-core asynchronous FIFO for CDC-safe

2. Changing the absolute-value accumulation in sfp_row.v to a balanced adder tree by converting the cumulative depth from linear to logarithmic.will help reduce the critical path and combinational delay.



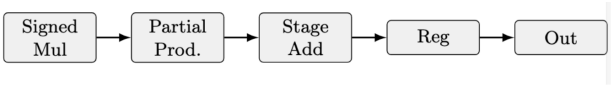
8-4-2-1 Adder Tree

3. Added the reset synchronizers for each clock domain in **fullchip.v** it will make the dual-clock operation more stable. The reset is asynchronously asserted but synchronously released in each domain to avoid metastability during reset deassertion.
4. Improved the normalization path by registering the inputs before division and handling the zero-denominator case in **sfp_row.v.** The pipeline register we added can break the long combinational path, while **den=0** to check for prevent invalid divide operations.



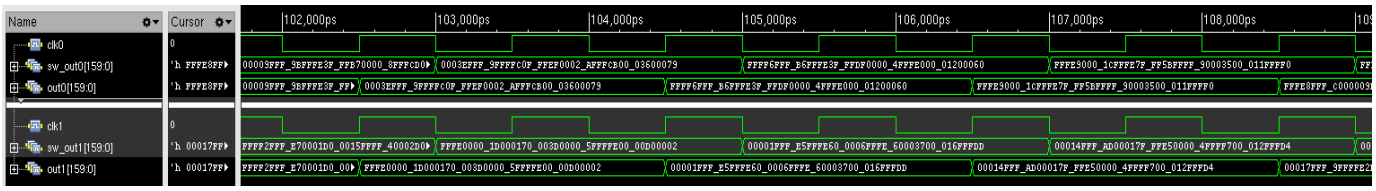
Pipelined Normalization with den=0

5. Rewritten the **mac_8in.v** using signed partial products and staged additions, by the structured reduction and optional pipelining improve synthesis quality and reduce timing pressure on the MAC unit.



Signed partial products + staged reduction

6. Updated the testbench **fullchip_tb.v** with one pipeline stage to the SFP divide/writeback path. Because of the extra stage, the testbench writeback timing was delayed by one cycle. Functional simulation all passed for both dot-product and SFP normalization



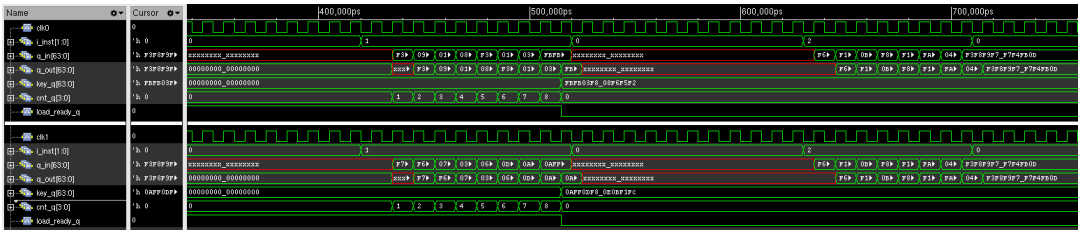

```
VG: elapsed time: 791.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 1
  Overlap     : 0
End Summary

  Verification Complete : 1 Viols.  0 Wrngs.

**********End: VERIFY GEOMETRY**********
 *** verify geometry (CPU: 0:02:53  MEM: 518.8M)
```

Step 6 Hierarchical Sparsity-Aware Attention

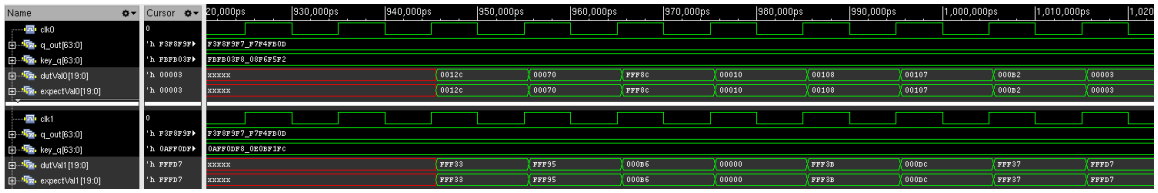
Implementation:

Row-level sparsity is dynamically applied by asserting a `skip_row_flag` whenever a row's pre-normalization sum falls below a Python-calculated 30% threshold. The hardware leverages this flag for both **gating** (saving dynamic power by freezing combinational divider toggling) and **skipping** (saving clock cycles by bypassing the memory write-back state). Verification is achieved through a Python co-simulation script that generates random Q/K vectors, enabling the Verilog testbench to validate exact accuracy for dense rows while successfully identifying and ignoring the bypassed memory of skipped rows.

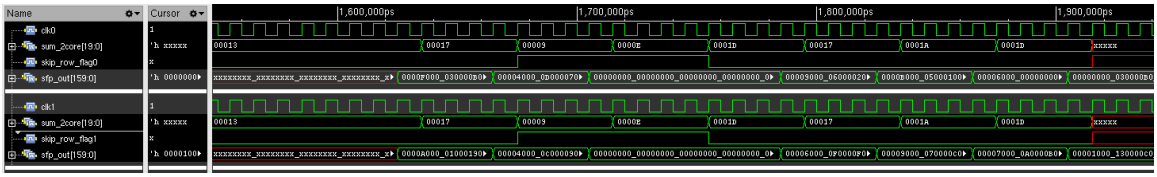
- Loading keys and processing query in the systolic array (showing waveforms for column 1 of both the cores):



- Read data from pmem and comparing with expected output:



- Verifying skip row when sum is less than threshold (threshold = 0x13) for both cores:



Conclusion

Parameter	Dual Core (Step 4)	Dual Core Optimized (Step 5)	Dual Core with Sparsity (Step 6)
Area (mm^2)	7.16	6.925	6.85
Setup WNS (ns)	-4.219	-0.866	-1.06
Power (mw)	260.98	443.4	334.86

The optimizations in Step 5 effectively tackled severe timing bottlenecks, drastically improving Setup WNS from -4.219ns to -0.866ns, though at the cost of significantly higher power consumption (443.4mW). Introducing dynamic row-level sparsity in Step 6 successfully recovered much of this power penalty, dropping consumption to 334.86mW via logic gating. Ultimately, the final sparsity-aware design strikes the best balance, retaining the critical timing improvements of the structural overhaul while maintaining strong power efficiency.

Deliverables

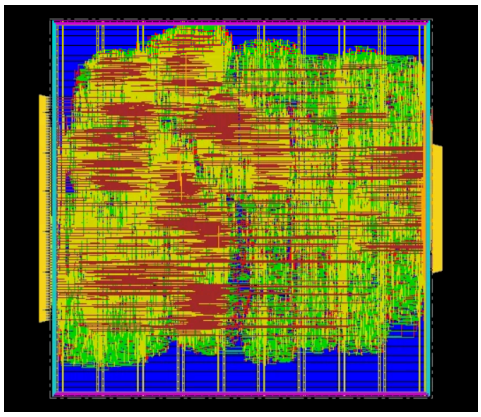
Step	Path	Files
Step 1	/home/linux/ieng6/ECE260B_WI26_A00/richaturvedi/ECE_260B_Project_Files/Step1/Deliverables	1. RTL Verilog 2. PNR .enc File 3. GLS Waveform vcd
Step 2	/home/linux/ieng6/ECE260B_WI26_A00/richaturvedi/ECE_260B_Project_Files/Step2/Deliverables	1. RTL Verilog 2. Behavioural Waveform vcd
Step 3	/home/linux/ieng6/ECE260B_WI26_A00/richaturvedi/ECE_260B_Project_Files/Step3/Deliverables	1. RTL Verilog 2. PNR .enc File 3. GLS Waveform vcd
Step 4	/home/linux/ieng6/ECE260B_WI26_A00/richaturvedi/ECE_260B_Project_Files/Step4/Deliverables	1. RTL Verilog 2. PNR .enc File 3. GLS Waveform vcd
Step 5	/home/linux/ieng6/ECE260B_WI26_A00/richaturvedi/ECE_260B_Project_Files/Step5/Deliverables	1. RTL Verilog 2. PNR .enc File 3. GLS Waveform vcd
Step 6	/home/linux/ieng6/ECE260B_WI26_A00/richaturvedi/ECE_260B_Project_Files/Step6/Deliverables	1. RTL Verilog 2. PNR .enc File 3. GLS Waveform vcd

Submission Directory

/home/linux/ieng6/ECE260B_WI26_A00/richaturvedi/ECE_260B_Project_Files/

Appendix

Step 1
PNR



DRC violations

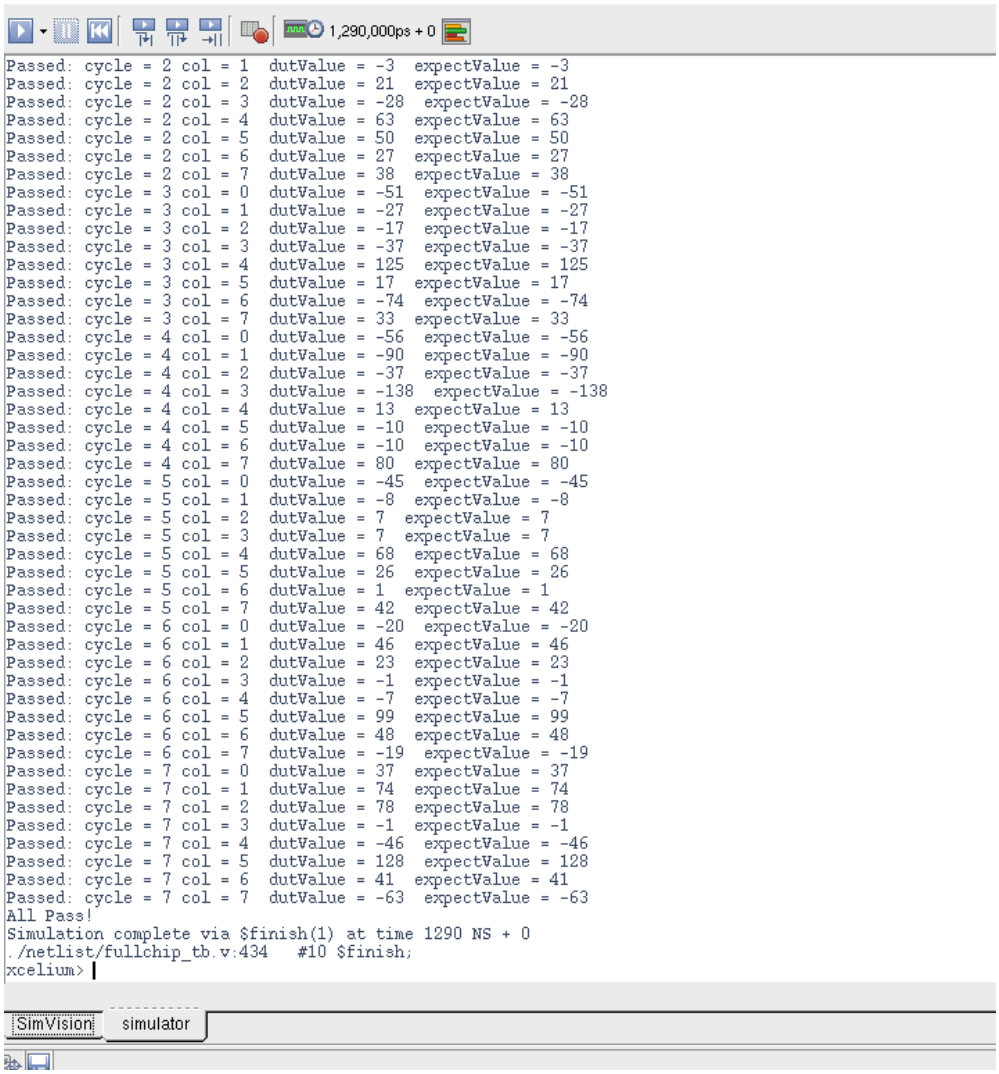
```
innovus 46> verify_drc
 *** Starting Verify DRC (MEM: 2053.5) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
  VERIFY DRC ..... Using new threading
  VERIFY DRC ..... Sub-Area : 1 of 4
  VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.
  VERIFY DRC ..... Sub-Area : 2 of 4
  VERIFY DRC ..... Sub-Area : 2 complete 0 Viols.
  VERIFY DRC ..... Sub-Area : 3 of 4
  VERIFY DRC ..... Sub-Area : 3 complete 0 Viols.
  VERIFY DRC ..... Sub-Area : 4 of 4
  VERIFY DRC ..... Sub-Area : 4 complete 0 Viols.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:00:14.8  ELAPSED TIME: 15.00  MEM: 0.0M
```

LVS violations

```
******** Start: VERIFY CONNECTIVITY ********
Start Time: Tue Mar 10 22:40:35 2026

Design Name: fullchip
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (565.0000, 561.8000)
Error Limit = 1000; Warning Limit = 50
Check all nets
**** 22:40:35 **** Processed 5000 nets.
**** 22:40:36 **** Processed 10000 nets.
**** 22:40:36 **** Processed 15000 nets.
**** 22:40:36 **** Processed 20000 nets.
**** 22:40:37 **** Processed 25000 nets.
**** 22:40:37 **** Processed 30000 nets.
**** 22:40:37 **** Processed 35000 nets.
**** 22:40:37 **** Processed 40000 nets.

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Tue Mar 10 22:40:38 2026
Time Elapsed: 0:00:03.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:02.5  MEM: -0.535M)
```

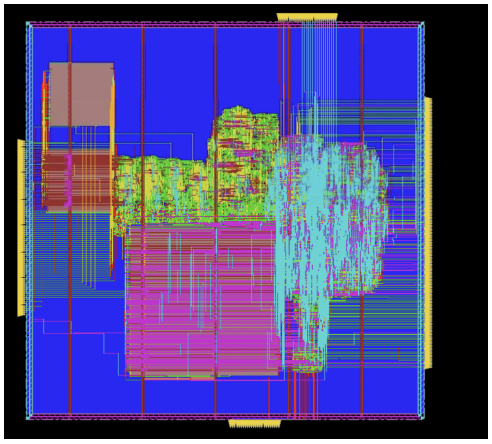
GLS verification

```
xcelium>
xcelium> source /software/ECE/XCELIUM-2009/tools/xcelium/files/xmsimrc
xcelium> run
##### Q data txt reading #####
##### K data txt reading #####
##### Estimated multiplication result #####
prd @cycle 0: 0005dfffecffff90005efffd0fffd3fffc1fffaa
prd @cycle 1: 00034fffebfffb700018ffffbfffdfffffd7fffe9
prd @cycle 2: 000260001b000320003fffffe400015fffffffc9
prd @cycle 3: 00021fffb6000110007dfffdbfffefffffe5fffcd
prd @cycle 4: 00050ffff6ffff60000dfff76fffdbfffa6fffc8
prd @cycle 5: 0002a000010001a0004400007000007ffff8fffd3
prd @cycle 6: fffed0003000063ffff9fffff000170002efffec
prd @cycle 7: fffc10002900080fffd2fffff0004e0004a00025
##### Qmem writing  #####
##### Kmem writing #####
##### K data loading to processor #####
##### execute #####
##### move ofifo to pmem #####
##### pmem read-back & compare #####
Passed: cycle = 0 col = 0  dutValue = -86  expectValue = -86
Passed: cycle = 0 col = 1  dutValue = -63  expectValue = -63
Passed: cycle = 0 col = 2  dutValue = -45  expectValue = -45
Passed: cycle = 0 col = 3  dutValue = -48  expectValue = -48
Passed: cycle = 0 col = 4  dutValue = 94  expectValue = 94
Passed: cycle = 0 col = 5  dutValue = -7  expectValue = -7
Passed: cycle = 0 col = 6  dutValue = -20  expectValue = -20
Passed: cycle = 0 col = 7  dutValue = 93  expectValue = 93
Passed: cycle = 1 col = 0  dutValue = -23  expectValue = -23
Passed: cycle = 1 col = 1  dutValue = -41  expectValue = -41
Passed: cycle = 1 col = 2  dutValue = -33  expectValue = -33
Passed: cycle = 1 col = 3  dutValue = -5  expectValue = -5
Passed: cycle = 1 col = 4  dutValue = 24  expectValue = 24
Passed: cycle = 1 col = 5  dutValue = -73  expectValue = -73
Passed: cycle = 1 col = 6  dutValue = -21  expectValue = -21
Passed: cycle = 1 col = 7  dutValue = 52  expectValue = 52
Passed: cycle = 2 col = 0  dutValue = -55  expectValue = -55
Passed: cycle = 2 col = 1  dutValue = -3  expectValue = -3
Passed: cycle = 2 col = 2  dutValue = 21  expectValue = 21
Passed: cycle = 2 col = 3  dutValue = -28  expectValue = -28
Passed: cycle = 2 col = 4  dutValue = 63  expectValue = 63
Passed: cycle = 2 col = 5  dutValue = 50  expectValue = 50
Passed: cycle = 2 col = 6  dutValue = 27  expectValue = 27
Passed: cycle = 2 col = 7  dutValue = 38  expectValue = 38
Passed: cycle = 3 col = 0  dutValue = -51  expectValue = -51
Passed: cycle = 3 col = 1  dutValue = -27  expectValue = -27
Passed: cycle = 3 col = 2  dutValue = -17  expectValue = -17
Passed: cycle = 3 col = 3  dutValue = -37  expectValue = -37
Passed: cycle = 3 col = 4  dutValue = 125  expectValue = 125
Passed: cycle = 3 col = 5  dutValue = 17  expectValue = 17
Passed: cycle = 3 col = 6  dutValue = -74  expectValue = -74
```


1,290,000ps + 0
Passed: cycle = 2 col = 1 dutValue = -3 expectValue = -3
Passed: cycle = 2 col = 2 dutValue = 21 expectValue = 21
Passed: cycle = 2 col = 3 dutValue = -28 expectValue = -28
Passed: cycle = 2 col = 4 dutValue = 63 expectValue = 63
Passed: cycle = 2 col = 5 dutValue = 50 expectValue = 50
Passed: cycle = 2 col = 6 dutValue = 27 expectValue = 27
Passed: cycle = 2 col = 7 dutValue = 38 expectValue = 38
Passed: cycle = 3 col = 0 dutValue = -51 expectValue = -51
Passed: cycle = 3 col = 1 dutValue = -27 expectValue = -27
Passed: cycle = 3 col = 2 dutValue = -17 expectValue = -17
Passed: cycle = 3 col = 3 dutValue = -37 expectValue = -37
Passed: cycle = 3 col = 4 dutValue = 125 expectValue = 125
Passed: cycle = 3 col = 5 dutValue = 17 expectValue = 17
Passed: cycle = 3 col = 6 dutValue = -74 expectValue = -74
Passed: cycle = 3 col = 7 dutValue = 33 expectValue = 33
Passed: cycle = 4 col = 0 dutValue = -56 expectValue = -56
Passed: cycle = 4 col = 1 dutValue = -90 expectValue = -90
Passed: cycle = 4 col = 2 dutValue = -37 expectValue = -37
Passed: cycle = 4 col = 3 dutValue = -138 expectValue = -138
Passed: cycle = 4 col = 4 dutValue = 13 expectValue = 13
Passed: cycle = 4 col = 5 dutValue = -10 expectValue = -10
Passed: cycle = 4 col = 6 dutValue = -10 expectValue = -10
Passed: cycle = 4 col = 7 dutValue = 80 expectValue = 80
Passed: cycle = 5 col = 0 dutValue = -45 expectValue = -45
Passed: cycle = 5 col = 1 dutValue = -8 expectValue = -8
Passed: cycle = 5 col = 2 dutValue = 7 expectValue = 7
Passed: cycle = 5 col = 3 dutValue = 7 expectValue = 7
Passed: cycle = 5 col = 4 dutValue = 68 expectValue = 68
Passed: cycle = 5 col = 5 dutValue = 26 expectValue = 26
Passed: cycle = 5 col = 6 dutValue = 1 expectValue = 1
Passed: cycle = 5 col = 7 dutValue = 42 expectValue = 42
Passed: cycle = 6 col = 0 dutValue = -20 expectValue = -20
Passed: cycle = 6 col = 1 dutValue = 46 expectValue = 46
Passed: cycle = 6 col = 2 dutValue = 23 expectValue = 23
Passed: cycle = 6 col = 3 dutValue = -1 expectValue = -1
Passed: cycle = 6 col = 4 dutValue = -7 expectValue = -7
Passed: cycle = 6 col = 5 dutValue = 99 expectValue = 99
Passed: cycle = 6 col = 6 dutValue = 48 expectValue = 48
Passed: cycle = 6 col = 7 dutValue = -19 expectValue = -19
Passed: cycle = 7 col = 0 dutValue = 37 expectValue = 37
Passed: cycle = 7 col = 1 dutValue = 74 expectValue = 74
Passed: cycle = 7 col = 2 dutValue = 78 expectValue = 78
Passed: cycle = 7 col = 3 dutValue = -1 expectValue = -1
Passed: cycle = 7 col = 4 dutValue = -46 expectValue = -46
Passed: cycle = 7 col = 5 dutValue = 128 expectValue = 128
Passed: cycle = 7 col = 6 dutValue = 41 expectValue = 41
Passed: cycle = 7 col = 7 dutValue = -63 expectValue = -63
All Pass!
Simulation complete via $finish(1) at time 1290 NS + 0
./netlist/fullchip_tb.v:434 #10 $finish;
xcelium>
SimVision
simulator

Step 3

PNR



DRC and LVS Violations

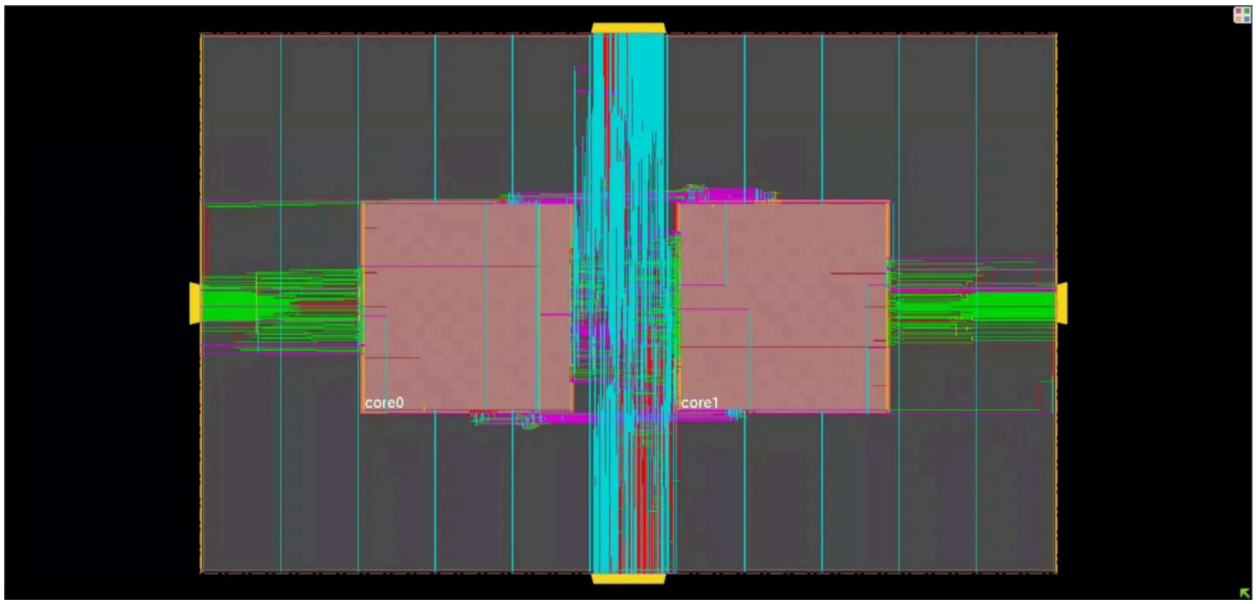
```
SHORT: Regular Via of Net core_instance/sfp_row_inst/n23392 & Blockage of Cell core_instance/sfp_row_inst/U18425  ( M1 )
Bounds : ( 1248.450, 683.960 ) ( 1248.550, 684.380 )


SHORT: Regular Via of Net core_instance/sfp_row_inst/n21121 & Blockage of Cell core_instance/sfp_row_inst/U27701  ( M1 )
Bounds : ( 1595.850, 918.070 ) ( 1595.950, 918.490 )



Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 2
  Overlap     : 0
End Summary

  Total Violations : 2 Viols.
~
~
~
~
~
```

```
#####################################################################
Verify Connectivity Report is created on Fri Mar 20 20:45:22 2026



Begin Summary
    Found no problems or warnings.
End Summary
~
~
~
~
~
```

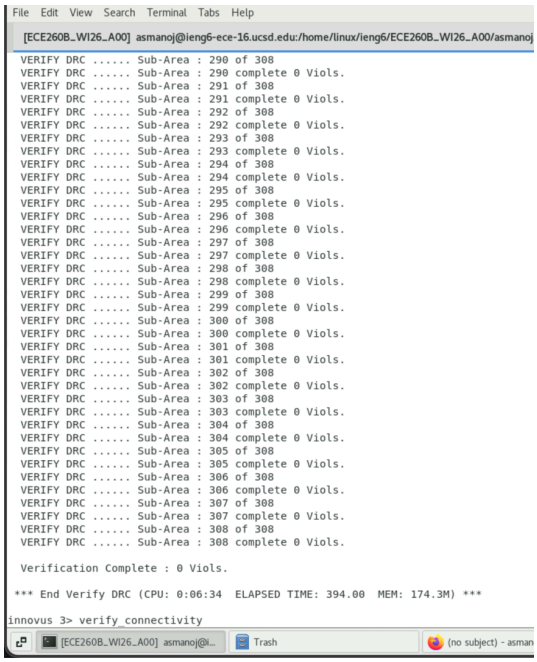
GLS

```
  col[1]: hw_sfp=13  sw_ref=13
  col[2]: hw_sfp=8  sw_ref=8
  col[3]: hw_sfp=18  sw_ref=18
  col[4]: hw_sfp=62  sw_ref=62
  col[5]: hw_sfp=8  sw_ref=8
  col[6]: hw_sfp=37  sw_ref=37
  col[7]: hw_sfp=16  sw_ref=16
sfp @cycle 4: 0001a0000300003000040002e0000c0001e00012
  col[0]: hw_sfp=18  sw_ref=18
  col[1]: hw_sfp=30  sw_ref=30
  col[2]: hw_sfp=12  sw_ref=12
  col[3]: hw_sfp=46  sw_ref=46
  col[4]: hw_sfp=4  sw_ref=4
  col[5]: hw_sfp=3  sw_ref=3
  col[6]: hw_sfp=3  sw_ref=3
  col[7]: hw_sfp=26  sw_ref=26
sfp @cycle 5: 0002a000010001a0004400007000070000800002d
  col[0]: hw_sfp=45  sw_ref=45
  col[1]: hw_sfp=8  sw_ref=8
  col[2]: hw_sfp=7  sw_ref=7
  col[3]: hw_sfp=7  sw_ref=7
  col[4]: hw_sfp=68  sw_ref=68
  col[5]: hw_sfp=26  sw_ref=26
  col[6]: hw_sfp=1  sw_ref=1
  col[7]: hw_sfp=42  sw_ref=42
sfp @cycle 6: 0000900018000310000300000000000b000170000a
  col[0]: hw_sfp=10  sw_ref=10
  col[1]: hw_sfp=23  sw_ref=23
  col[2]: hw_sfp=11  sw_ref=11
  col[3]: hw_sfp=0  sw_ref=0
  col[4]: hw_sfp=3  sw_ref=3
  col[5]: hw_sfp=49  sw_ref=49
  col[6]: hw_sfp=24  sw_ref=24
  col[7]: hw_sfp=9  sw_ref=9
sfp @cycle 7: 000150000d0002a0000f000000001a000180000c
  col[0]: hw_sfp=12  sw_ref=12
  col[1]: hw_sfp=24  sw_ref=24
  col[2]: hw_sfp=26  sw_ref=26
  col[3]: hw_sfp=0  sw_ref=0
  col[4]: hw_sfp=15  sw_ref=15
  col[5]: hw_sfp=42  sw_ref=42
  col[6]: hw_sfp=13  sw_ref=13
  col[7]: hw_sfp=21  sw_ref=21
Simulation complete via $finish(1) at time 206010 PS + 0
./netlist/fullchip_tb.v:655   #10 $finish;\r
xcelium>
```

Step 4
PNR

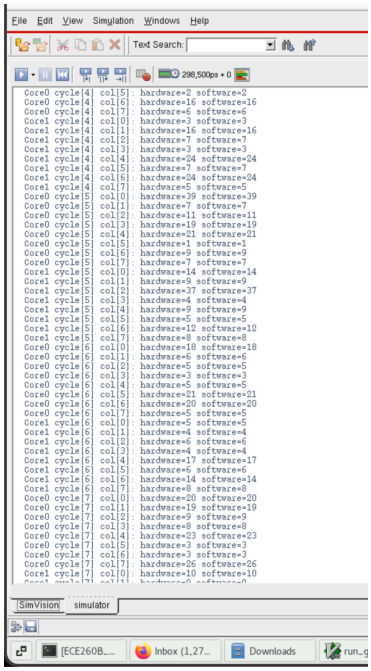


DRC Violations



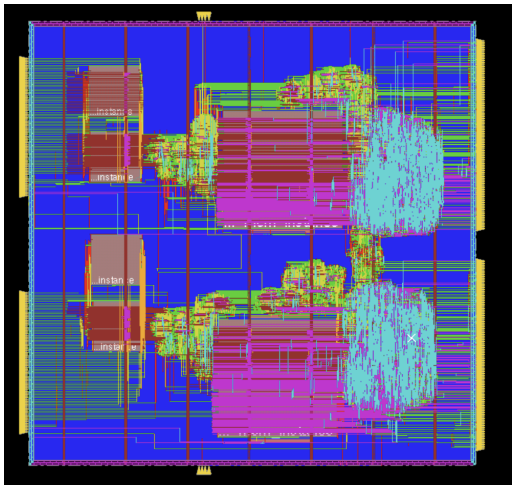
```
File  Edit  View  Search  Terminal  Tabs  Help
[ECE260B_WI26_A00] asmanoj@ieng6-ece-16.ucsd.edu:/home/linux/ieng6/ECE260B_WI26_A00/asmanoj
 VERIFY DRC ..... Sub-Area : 290 of 308
 VERIFY DRC ..... Sub-Area : 290 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 291 of 308
 VERIFY DRC ..... Sub-Area : 291 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 292 of 308
 VERIFY DRC ..... Sub-Area : 292 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 293 of 308
 VERIFY DRC ..... Sub-Area : 293 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 294 of 308
 VERIFY DRC ..... Sub-Area : 294 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 295 of 308
 VERIFY DRC ..... Sub-Area : 295 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 296 of 308
 VERIFY DRC ..... Sub-Area : 296 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 297 of 308
 VERIFY DRC ..... Sub-Area : 297 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 298 of 308
 VERIFY DRC ..... Sub-Area : 298 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 299 of 308
 VERIFY DRC ..... Sub-Area : 299 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 300 of 308
 VERIFY DRC ..... Sub-Area : 300 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 301 of 308
 VERIFY DRC ..... Sub-Area : 301 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 302 of 308
 VERIFY DRC ..... Sub-Area : 302 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 303 of 308
 VERIFY DRC ..... Sub-Area : 303 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 304 of 308
 VERIFY DRC ..... Sub-Area : 304 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 305 of 308
 VERIFY DRC ..... Sub-Area : 305 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 306 of 308
 VERIFY DRC ..... Sub-Area : 306 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 307 of 308
 VERIFY DRC ..... Sub-Area : 307 complete 0 Viols.
 VERIFY DRC ..... Sub-Area : 308 of 308
 VERIFY DRC ..... Sub-Area : 308 complete 0 Viols.

 Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:06:34  ELAPSED TIME: 394.00  MEM: 174.3M) ***

innovus 3> verify_connectivity
[ECE260B_WI26_A00] asmanoj@i...   Trash   (no subject) - asman
```

GLS



Step 5

PNR



DRC Violations

```
VG: elapsed time: 791.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 1
  Overlap     : 0
End Summary

  Verification Complete : 1 Viols.  0 Wrngs.

**********End: VERIFY GEOMETRY**********
 *** verify geometry (CPU: 0:02:53  MEM: 518.8M)
```

LVS:

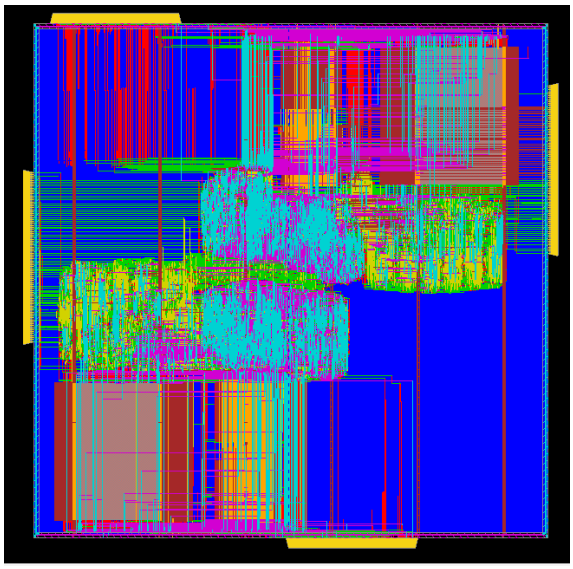
```
*** 14:09:09 *** Checking data for Net VDD ...................
*** 14:09:16 *** Building data for Net VSS
*** 14:09:16 *** Building data for Net VSS
*** 14:09:16 *** Building data for Net VSS
*** 14:09:17 *** Building data for Net VSS
*** 14:09:17 *** Building data for Net VSS
*** 14:09:17 *** Checking data for Net VSS ...................

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Sat Mar 21 14:09:24 2026
Time Elapsed: 0:00:21.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:21.7  MEM: 871.766M)
```

Hold mode	all	reg2reg	default
WNS (ns):	-0.086	-0.086	0.001
TNS (ns):	-1.040	-1.040	0.000
Violating Paths:	69	69	0
All Paths:	12372	12010	1062

Setup mode	all	reg2reg	default
WNS (ns):	-0.866	-0.866	-0.479
TNS (ns):	-302.145	-192.540	-109.604
Violating Paths:	1290	970	320
All Paths:	12984	12330	1674

Step 6

PNR



DRC violations

```
VERIFY DRC ..... Sub-Area : 57 complete 0 Viols.
VERIFY DRC ..... Sub-Area : 58 of 64
VERIFY DRC ..... Sub-Area : 58 complete 0 Viols.
VERIFY DRC ..... Sub-Area : 59 of 64
VERIFY DRC ..... Sub-Area : 59 complete 0 Viols.
VERIFY DRC ..... Sub-Area : 60 of 64
VERIFY DRC ..... Sub-Area : 60 complete 1 Viols.
VERIFY DRC ..... Sub-Area : 61 of 64
VERIFY DRC ..... Sub-Area : 61 complete 0 Viols.
VERIFY DRC ..... Sub-Area : 62 of 64
VERIFY DRC ..... Sub-Area : 62 complete 0 Viols.
VERIFY DRC ..... Sub-Area : 63 of 64
VERIFY DRC ..... Sub-Area : 63 complete 0 Viols.
VERIFY DRC ..... Sub-Area : 64 of 64
VERIFY DRC ..... Sub-Area : 64 complete 0 Viols.

Verification Complete : 18 Viols.

*** End Verify DRC (CPU: 0:03:43  ELAPSED TIME: 224.00  MEM: 400.7M) ***
```

LVS

```
*** 15:11:24 *** Building data for Net VSS
*** 15:11:24 *** Building data for Net VSS
*** 15:11:24 *** Building data for Net VSS
*** 15:11:24 *** Checking data for Net VSS ......................

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Sat Mar 21 15:11:32 2026
Time Elapsed: 0:00:22.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:20.5  MEM: 575.207M)

innovus 4>
```