ESE5730 Chip Design

Computing: Digital 8-bit 6x6 Matrix Multiplier Engine

Team 8

Shun Yao

Yue Zhang

TA: Claire Kendell

File Path:
/home/home2/team8chips2025/TSMC_180_work/Matrix_engine_backup_final
/Final_Top_Cell_Improved

DEPARTMENT OF ELECTRICAL AND SYSTEM ENGINEERING

UNIVERSITY OF PENNSYLVANIA

I. Layout Design Flow

We divided the circuit layout into seven different subsystems just like what we did for schematic: Input/Output Memory bank, SIPO/PISO register, Computation unit, and Finite State Machine. Each subsystem consists of many different basic units. For example, the memory bank consists of Pre-charge, 6-T SRAM, Write driver, Sense Amplifier, and WL decoder. Layout was implemented from the smallest unit to the subsystem to the top cell.

Post layout optimization includes adding buffer to drive high loaded signal and make the track thicker/shorter.

II. System Block Diagram, Schematics and Layout

Top-Cell:

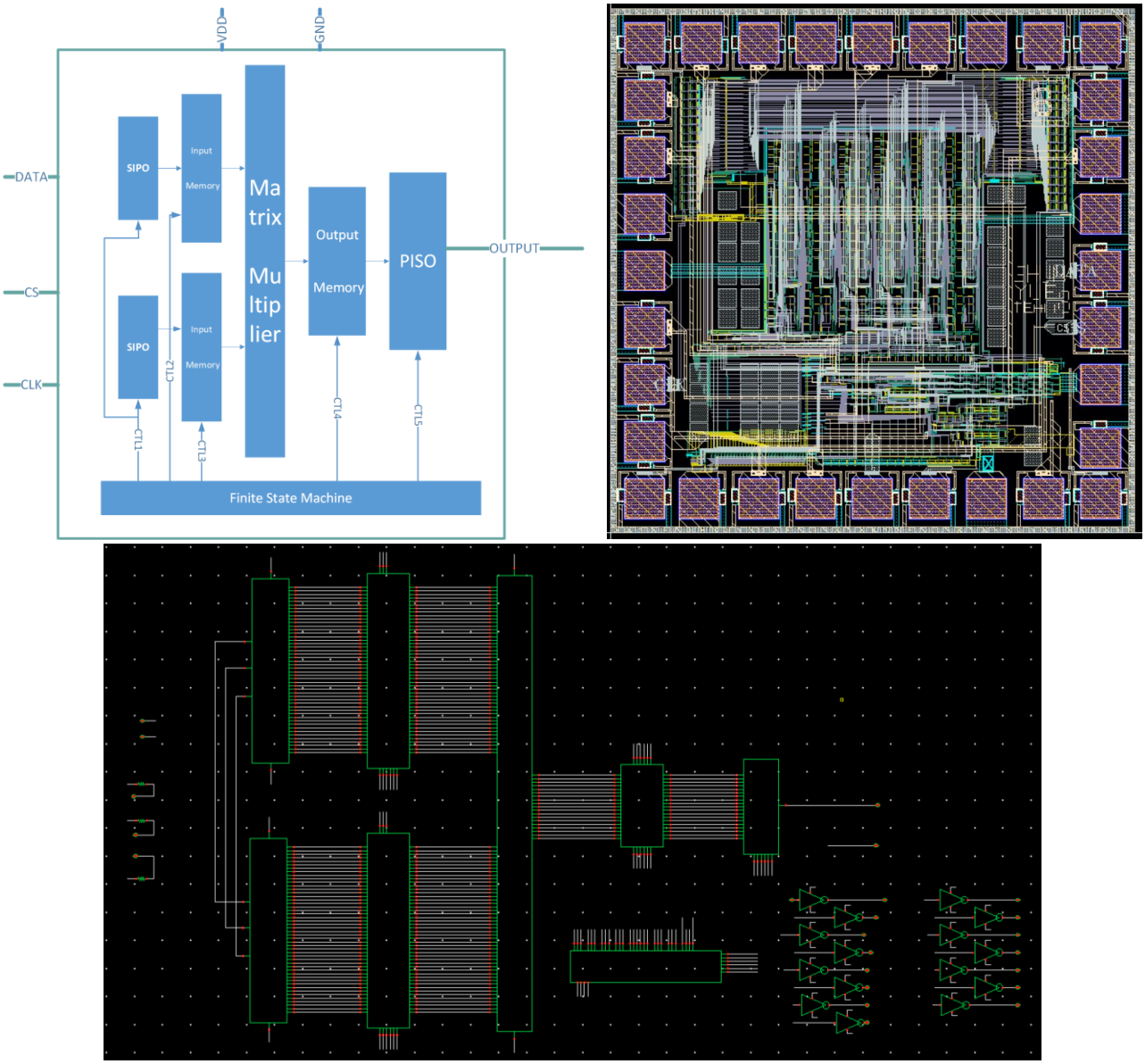

Figure 1. Top – Cell

SIPO:

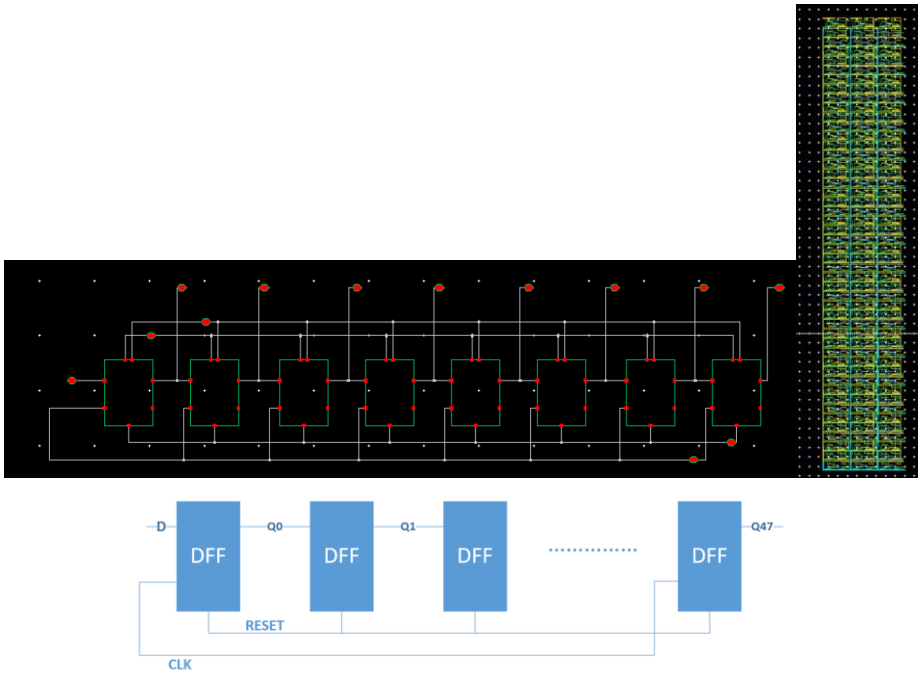

Figure 2. SIPO Shift Register (left: Detailed Schematic, right: Full Schematic)

Input Memory Array:

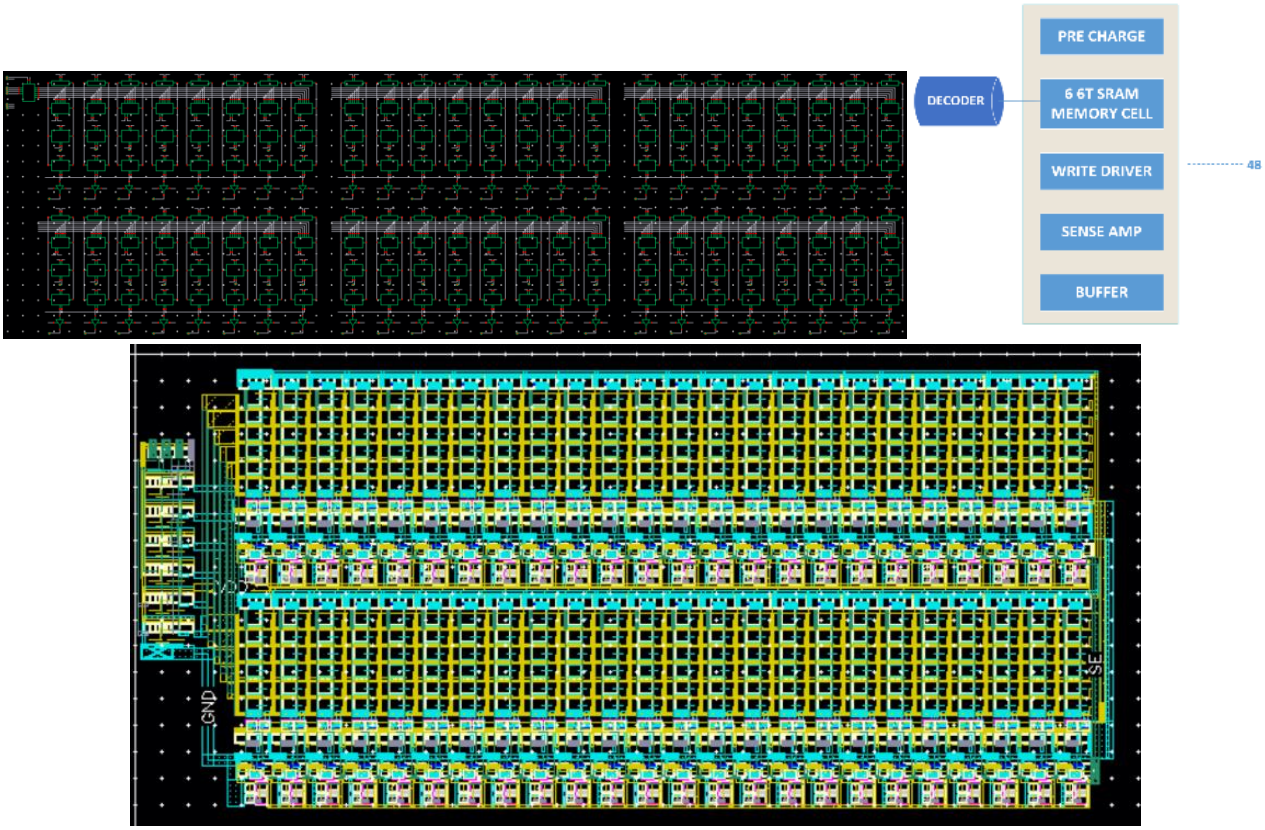

Figure 3. Input 8 bit 6x6 Memory Array

Matrix Multiplier:

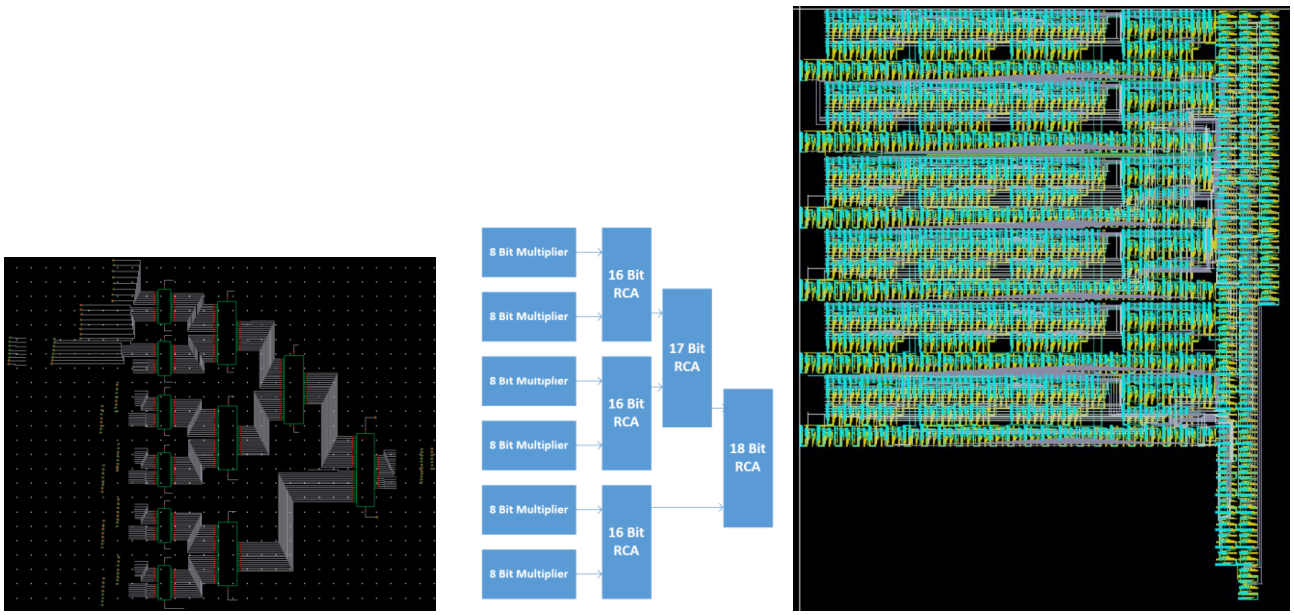

Figure 4. Matrix Multiplier

Output Memory Array:

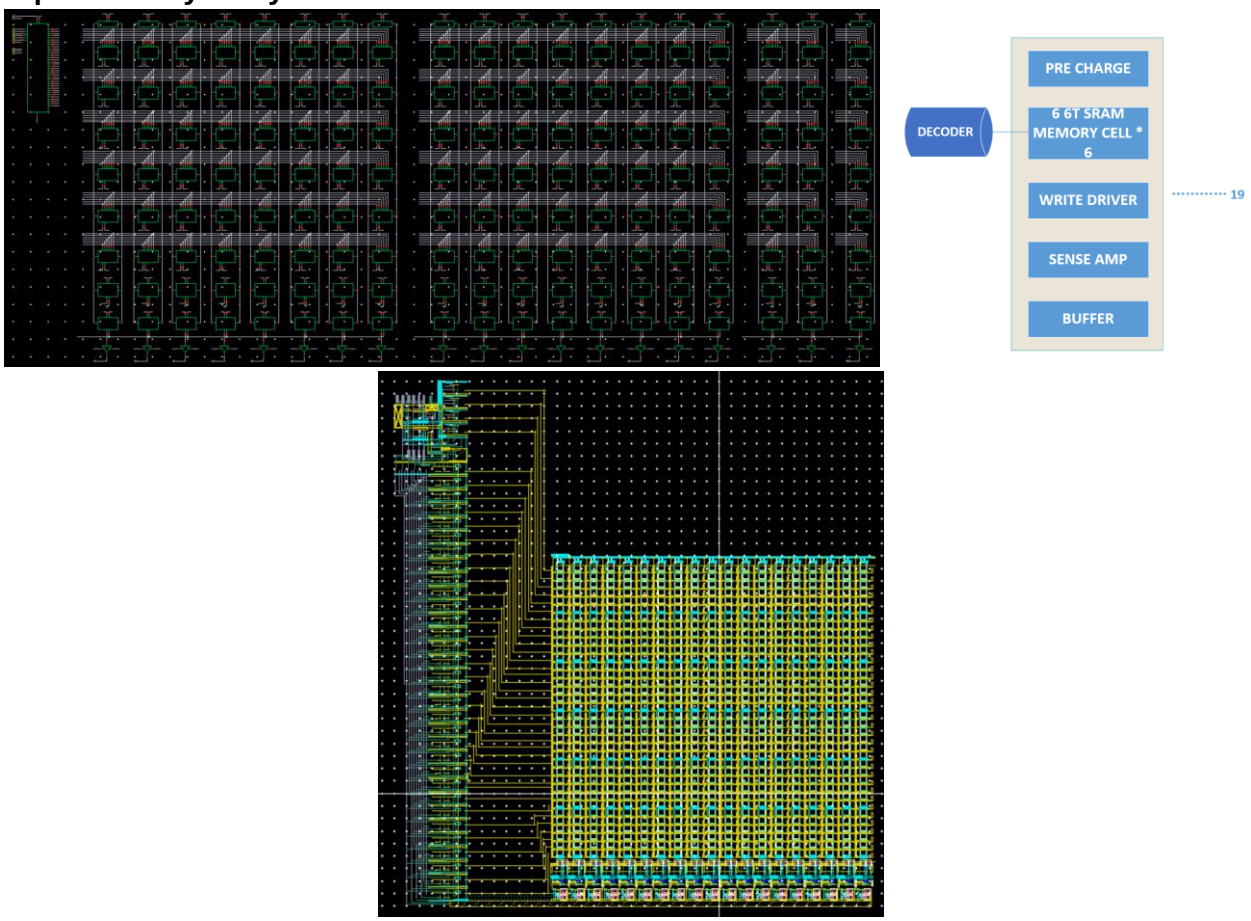

Figure 5. Output 19bit 6x6 Memory Array

PISO:

Figure 6. PISO Shift Register

Finite State Machine:

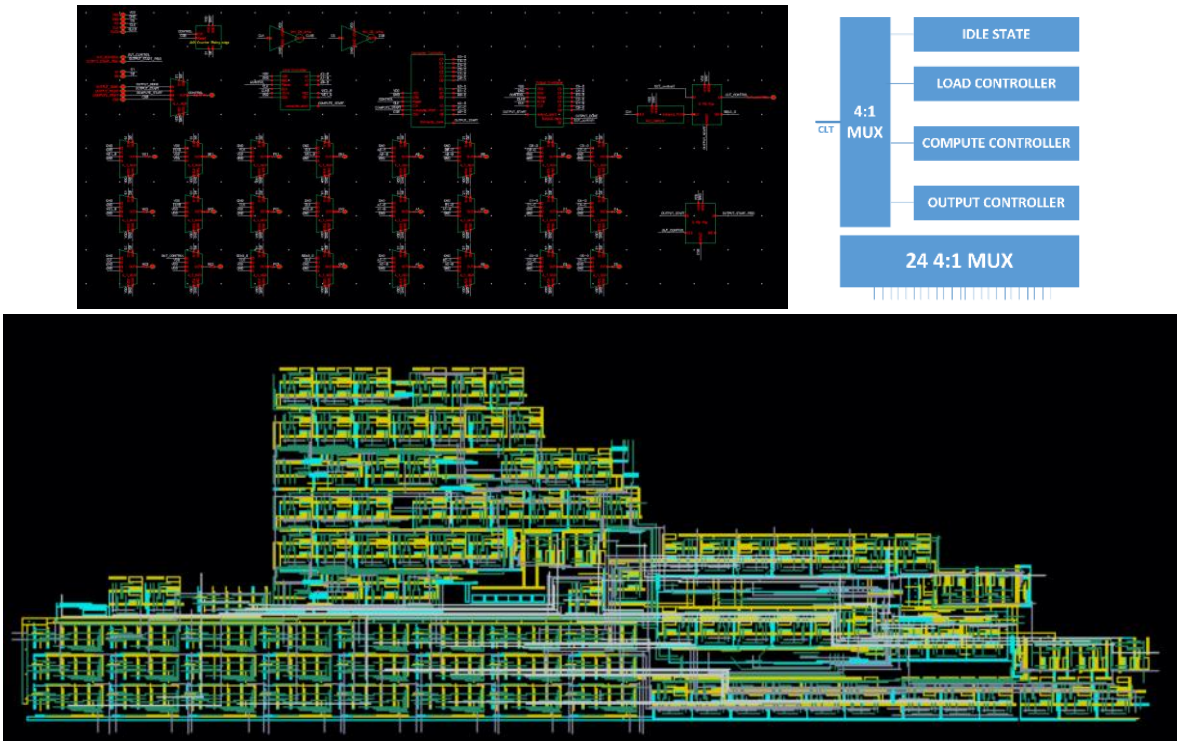

Figure 7. FSM Schematic

III. Optimization from Design Review 2

The post layout simulation result was not correct at the first simulation run. It turns out that the big capacitance introduced by pads is generating significant delay on our control signal in the finite state machine.

To fix this issue, we first clean up some of the FSM design logic to reduce the delay introduced by multi-stages, figure below has shown an example: we replaced AND+INV with a NAND gate.

Additionally, the ‘control’ signal in FSM travel through a long narrow track on the layout, which introducing certain delay. One buffer with large transistor size (W/L=10u/180n) was placed in FSM to drive the signal.

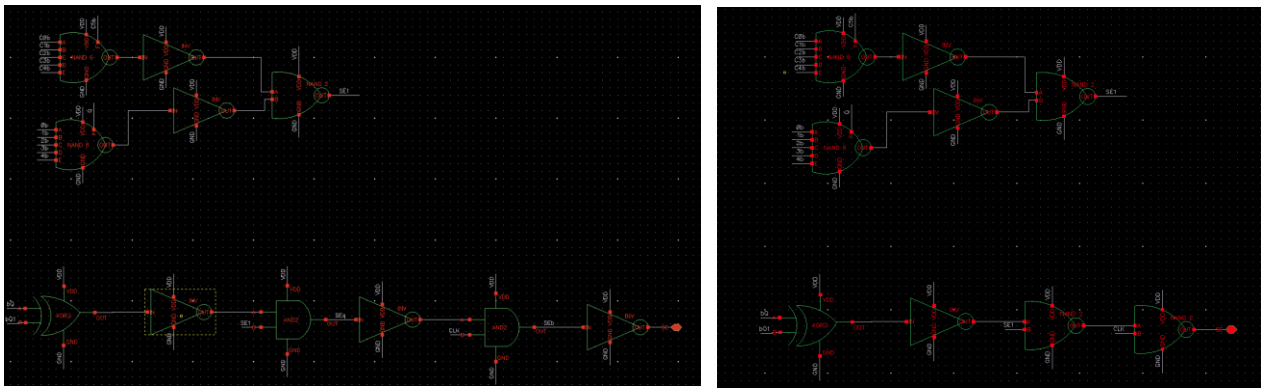

Figure 8. Design Optimization (left: before optimization; right: after optimization)

In addition, to address issues related to metal current-carrying capacity and manufacturing constraints, we widened several long metal interconnects across the full chip, including the connections to the I/O pads. These changes were made to enhance overall timing and reliability, and no schematic-level functional changes were required beyond buffering and interconnect adjustments.

IV. Circuit Block Layouts

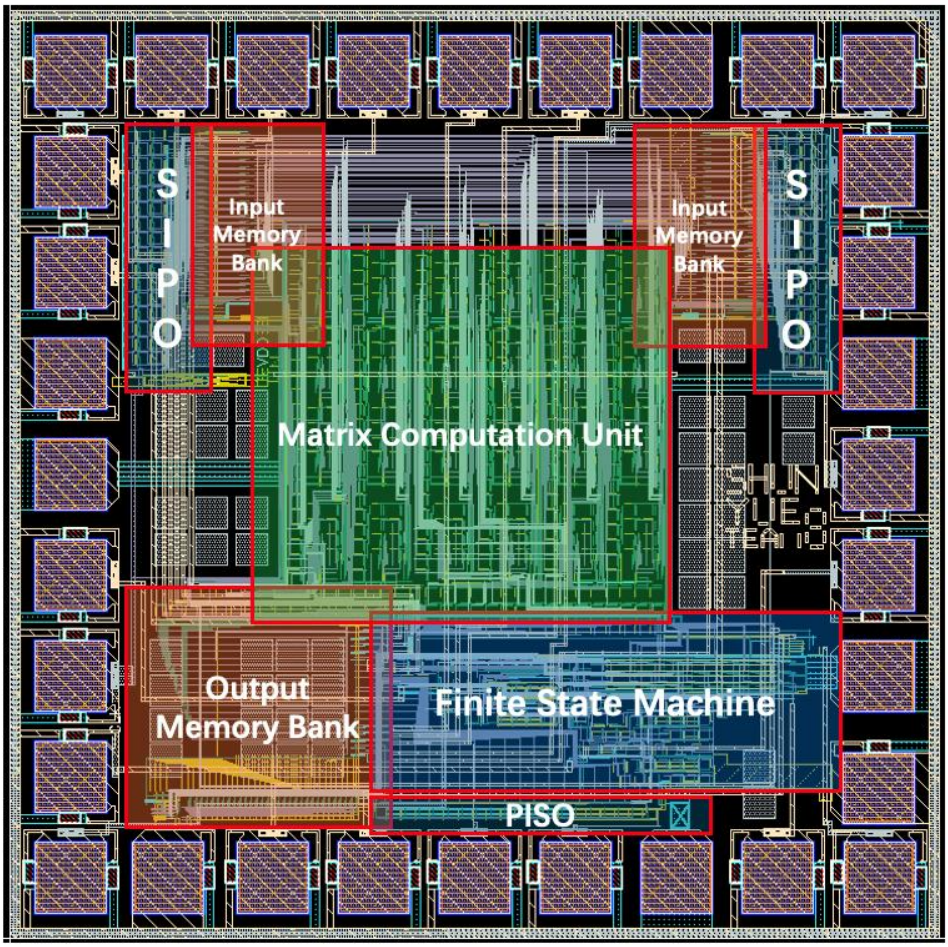

Figure 9. Top - Cell Pad Pinout

V. Full Chip Pre-Layout and Post-Layout Simulation

Transient simulations have run for top cell schematic. Take one calculation circle as an example, six 0011 0101 times six 1111 1111 correspondingly, and adding the six multiplication results together, the final result is supposed to be 19bit 0010 0111 1001 1000 010 as shown in the Figure10 below (First Computation Result).

The system is firstly in an IDLE state waiting for the start signal ‘CS’; when signal ‘CS’ pulls down from 1 to 0, the system goes to LOAD state loading data to MatrixA and MatrixB; when the load is done, the finite state machine would push the system to COMPUTE state, the system will read data from MatrixA and B, compute it, and write it to MatrixC in this state; when computation is done, FSM will drive the system to OUTPUT state, which will read data from MatrixC and load it to PISO register and the result will output in serial as shown in figuree10: (Post_layout Result will be present in critical design review)

Shun contributes to the schematic optimization and layout; Yue works on the layout improvements.

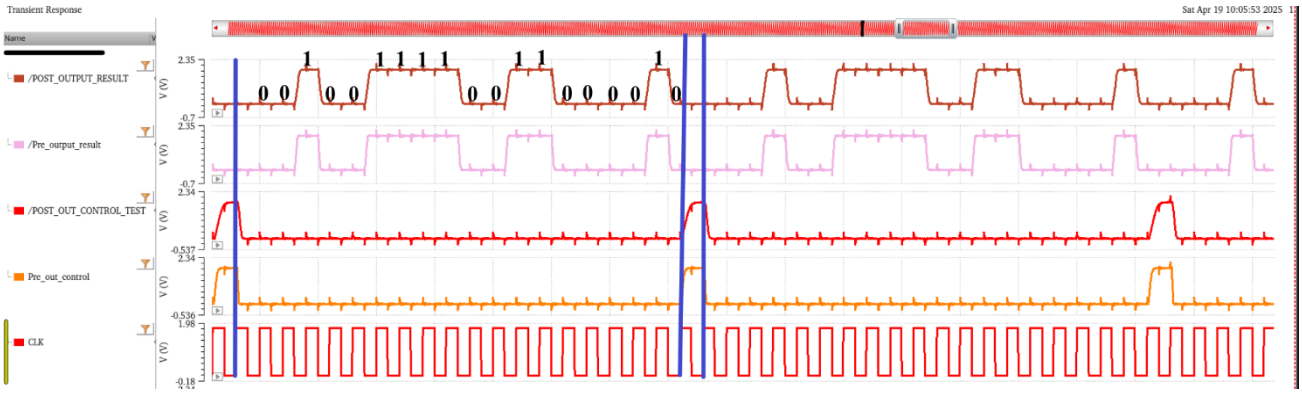

Figure 10. Top – Cell Simulation Result (with 2nH inductors)

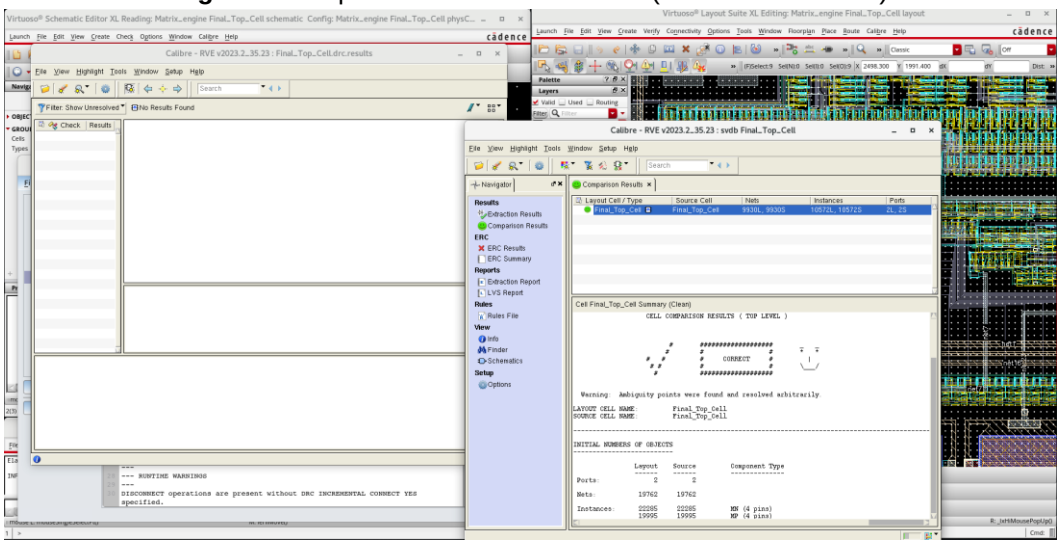

Figure 11. Top – Cell DRC/LVS Result (Included ant.drc test)

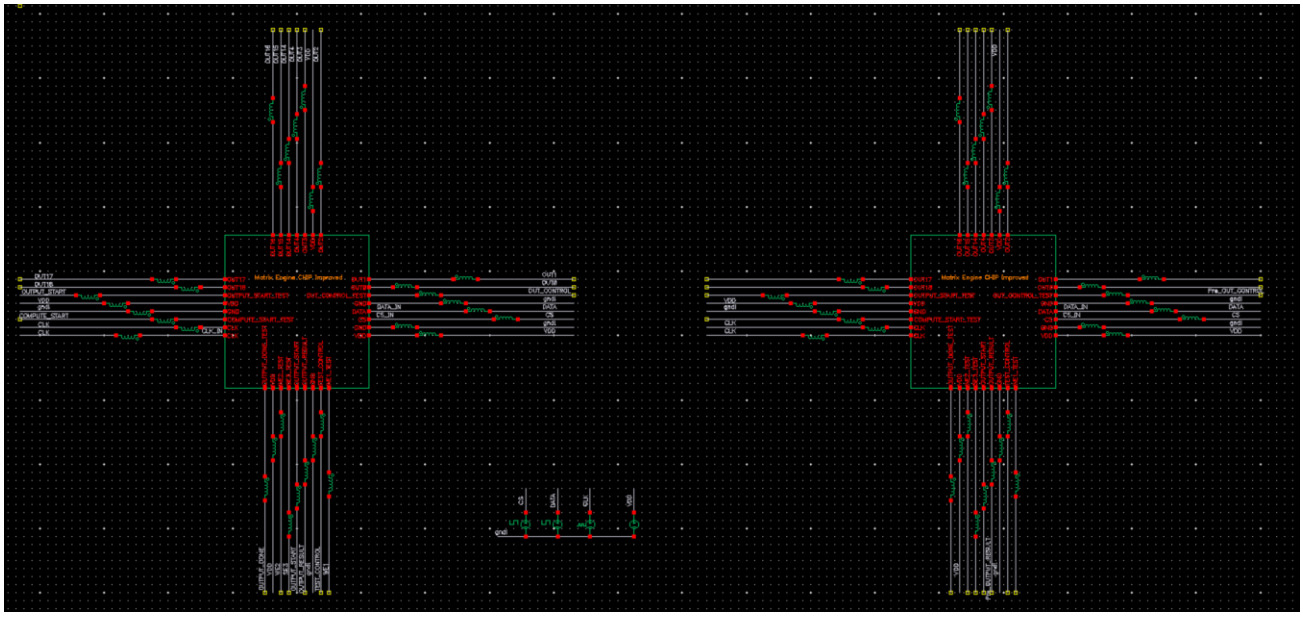

Figure 11. Top – Cell Simulation Schematic

Schematic and Layout File Path (Top Cell):

/home/home2/team8chips2025/TSMC_180_work/Matrix_engine_backup_final/Final_Top_Cell_Improved

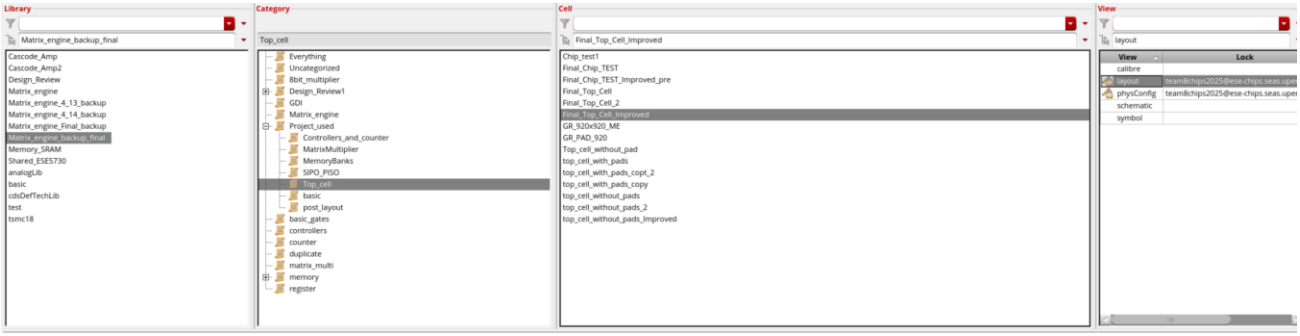

Figure 12. File Path

VI. Table of Performance

Table 1. Performance and Comparison

Full Chip	VCC	CDR Speed (CLK period)	CDR Power (CLK period)
Pre-Layout Simulation	1.8V	8ns (125MHz)	9mA*1.8V
Post-Layout Simulation		18ns (55MHz)	20mA*1.8V

VII. Updated Measurement Plan

Serial Communication:

ESP32 (80MHz) (MCU) communicate with our chip through SPI with MOSI, SCLK and CS, corresponding to DATA, CLK and CS in our chip. When CS is switched from high to low, the SPI communication channel is opened. To load the data from MCU to SIPO (in chip), an enable signal (CSB) triggers the FSM change state from idle to load data. The calculation output from SIPO can be captured by logic analyzer/Oscilloscope for following verification.

Functional verification:

1. *Basic Arithmetic Correctness Test:* Send matrix data to the input of the chip through the MCU and compare the calculation results with theoretical calculation values to verify whether the matrix multiplication is correct using logic analyzer.

Performance Evaluation:

1. *Maximum Operating Frequency Test:* Gradually increase the clock frequency to test the stability of the chip at different frequencies and determine its maximum reliable operating frequency.
2. *Power Consumption Measurement:* Use a power measurement tool to measure operating power, compared to simulated power consumption.

Typical Application Scenarios

1. *Artificial Intelligence & Machine Learning:* Acceleration of small matrix operations in Convolutional Neural Networks (CNNs) to improve AI computational efficiency.
2. *Digital Signal Processing (DSP):* For image processing, edge detection, filtering, and other scenarios to increase processing speed.
3. *Embedded Computing & Low Power Applications:* Fast matrix computation in IoT devices to improve energy efficiency and reduce computing resources.

No Changes Requires.