IIIT-B Chip Design Studio Weekly Report

Week : 2

Team Name/Project: CIRCUIT CRAFTERS

Sparse Systolic Array for AI Acceleration and Matrix Computation Based Chip Design

1. Updates

Current Progress:

During Week-2, the single Processing Element (PE) designed in Week-1 was extended into a **dense 2×2 systolic array architecture**. Four identical MAC-based PEs were instantiated and interconnected to validate systolic dataflow for matrix multiplication. A cycle-controlled enable mechanism was introduced to limit accumulation to the required number of computation cycles. Behavioral simulation was performed using a known 2×2 matrix multiplication example, and correct outputs were verified.

Challenges Faced:

- Understanding systolic dataflow and temporal sequencing of matrix inputs.
- Preventing repeated accumulation beyond the required computation cycles.
- Interpreting intermediate accumulator values (partial sums) in simulation waveforms.

Next Steps:

- Need to introduce sparsity-aware control by generating the enable signal using zero-detection logic.
- Should compare dense versus sparse execution behaviour.
- Must perform switching-activity-based power analysis to evaluate sparsity benefits.

2. GitHub Link: Provide the GitHub repository link for the project, if any:

Repository: (Not created yet)

3. Project Idea

The objective of this project is to design a sparsity-aware systolic array architecture optimized for matrix multiplication in AI acceleration. Since modern AI workloads often use sparse matrices, conventional systolic arrays waste energy and computation on zero-valued operands. Our approach introduces zero-skipping mechanisms and compressed data representations to reduce redundant MAC operations, improve energy efficiency, and enhance throughput. The system targets AI accelerators, DSP applications, and low-power edge devices.

4. Schematic/Simulations

- **Schematics:**

Figure 1: RTL Schematic of 2×2 Dense Systolic Array

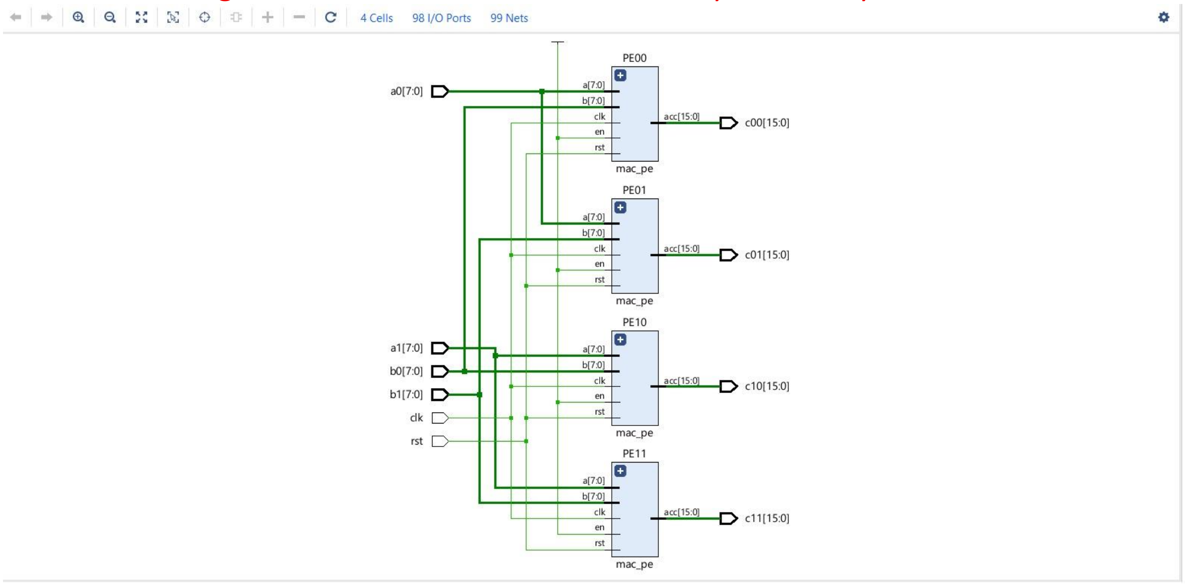


Figure 1 shows the RTL schematic of the implemented 2×2 dense systolic array. The design consists of four identical MAC-based Processing Elements (PEs), where each PE computes one output matrix element $C_{i,j}$. Matrix A inputs are streamed horizontally, matrix B inputs are streamed vertically, and partial sums are accumulated locally within each PE. A common clock and reset are shared across all PEs, validating correct systolic dataflow and multi-PE operation.

Figure 2: RTL Elaborated Netlist of the 2×2 Systolic Array

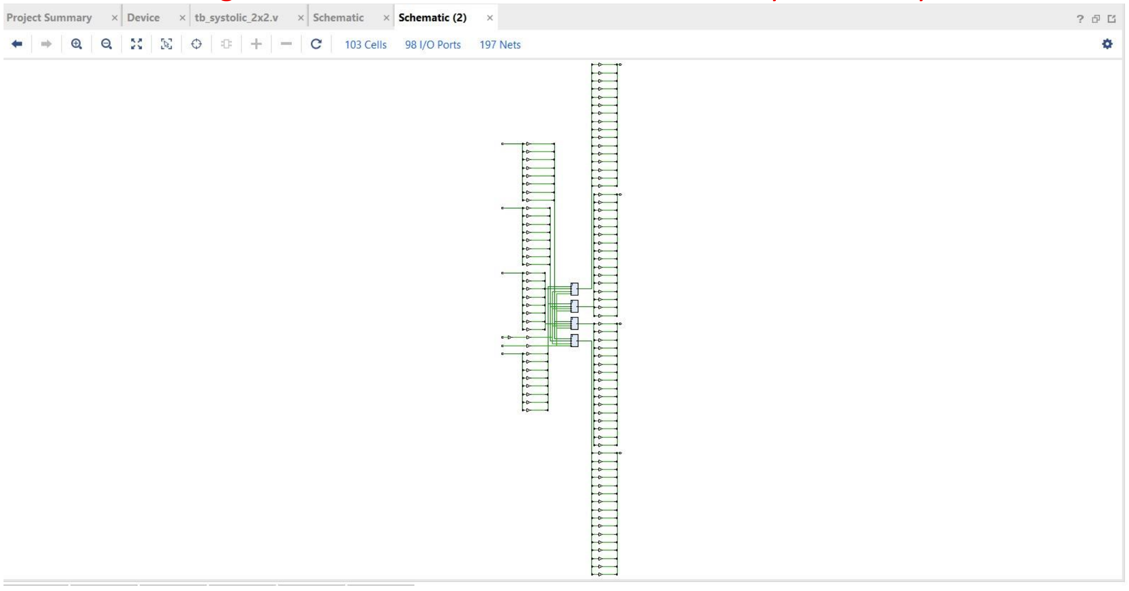
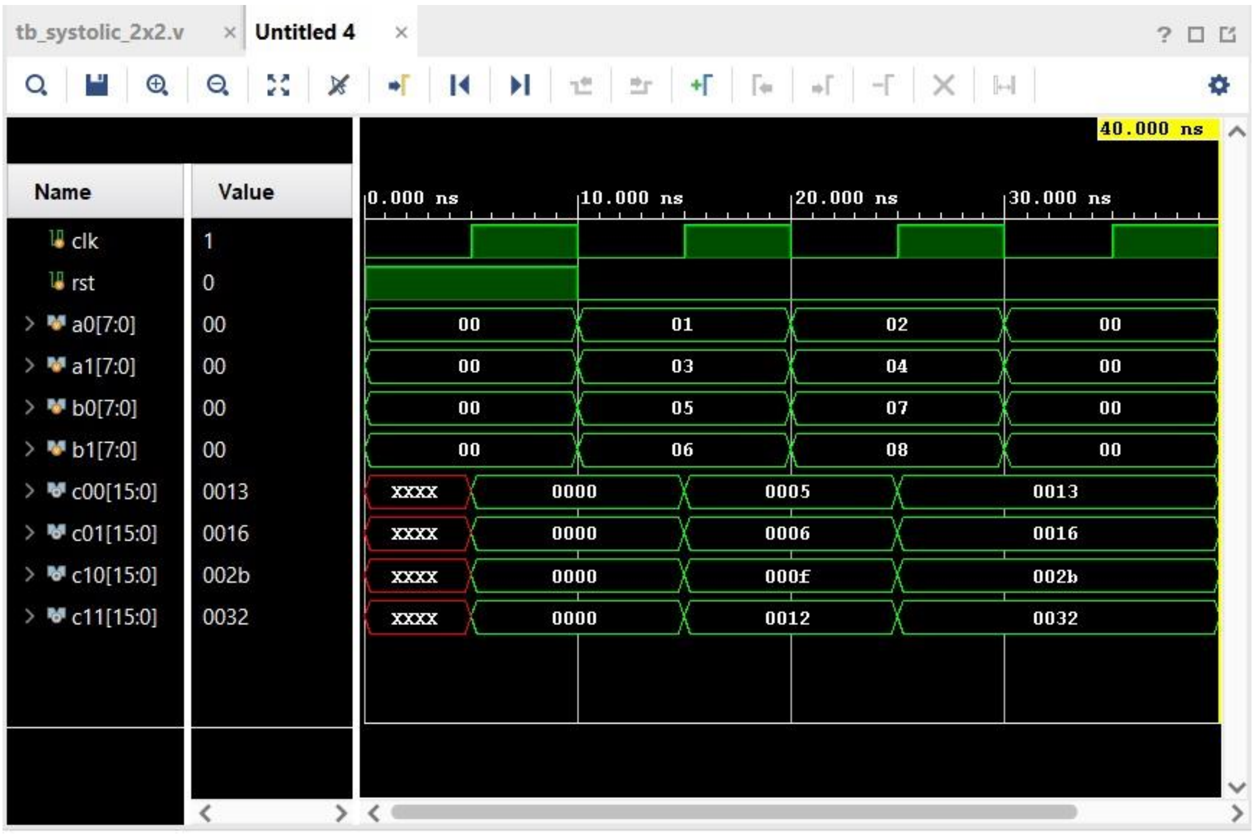


Figure 2 shows the RTL elaborated netlist of the 2×2 systolic array generated by AMD Vivado. This view represents the internal logical connectivity of the design after RTL elaboration, including registers, combinational logic, and signal interconnections. The figure confirms correct instantiation of four processing elements and validates hierarchical integration prior to synthesis

- **Simulation Results:**

Figure 3: Behavioral Simulation Waveform of 2×2 Systolic Array



Name	Value
clk	1
rst	0
a0[7:0]	00
a1[7:0]	00
b0[7:0]	00
b1[7:0]	00
c00[15:0]	0013
c01[15:0]	0016
c10[15:0]	002b
c11[15:0]	0032

```
a0[7:0]    00    01    02    00
a1[7:0]    00    03    04    00
b0[7:0]    00    05    07    00
b1[7:0]    00    06    08    00
c00[15:0]  XXXX  0000  0005  0013
c01[15:0]  XXXX  0000  0006  0016
c10[15:0]  XXXX  0000  000f  002b
c11[15:0]  XXXX  0000  0012  0032
```

Figure 3 shows the behavioral simulation waveform of the implemented 2×2 systolic array. Matrix inputs are streamed over two clock cycles, corresponding to the matrix dimension. Intermediate values observed after the first cycle represent partial sums, while the final accumulated values stabilize after the second cycle. The output values match the expected results of the matrix multiplication, confirming correct systolic dataflow and multi-PE operation.

Example: For example, the output c00 accumulates to 0x0013 (19), matching the expected value of $1 \times 5 + 2 \times 7$.

5. Analysis

- **Key Findings:**
  - The 2×2 systolic array correctly computes matrix multiplication results.
  - Intermediate values observed in the waveform represent valid partial sums.
  - Cycle-based enable control successfully prevents repeated accumulation.

- **Insights or Learnings:**
  - Systolic arrays rely on **temporal sequencing of inputs**, not static matrix representation.
  - Each processing element accumulates partial products over multiple cycles.
  - Separating data path (MAC) and control (enable) simplifies future sparsity integration.

- **Improvements or Modifications Needed:**
  - Replace cycle-based enable with sparsity-aware zero-detection logic.
  - Extend the design to support sparse matrix formats.
  - Perform comparative power analysis between dense and sparse operation.

TEAM DETAILS:

Team Name: Circuit Crafters

1.Jeswin S - sec23ec089@sairamtap.edu.in (sec23ec089@iiitb.net)
2.Moneswaran P - sec23ec225@sairamtap.edu.in (sec23ec225@iiitb.net)
3.Mokshith M - sec23ec192@sairamtap.edu.in (sec23ec192@iiitb.net)

College Name: Sri Sai Ram Engineering College, Chennai.