IIIT-B Chip Design Studio Weekly Report

Week: 3

Team Name/Project: C I R C U I T C R A F T E R S

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

1. Updates

Current Progress:

- Implemented a **2×2 systolic array architecture** using Processing Elements (PEs) based on multiply–accumulate (MAC) operations.
- Enhanced the baseline systolic design by introducing **sparsity-aware control logic**.
- Implemented **per-PE enable gating**, allowing MAC operations to be skipped when zero-valued operands are detected.
- Verified functional correctness through behavioral simulation using sparse matrix inputs.
- Confirmed that the sparse design produces identical outputs compared to the dense baseline while reducing internal computation.

Challenges Faced:

- Direct power estimation using switching-activity files (SAIF/VCD) was limited due to tool constraints in Vivado XSIM.
- Absolute power comparison was not reliable for a small 2×2 design.
- Differentiating functional correctness from computational efficiency required careful validation.

Next Steps:

- Extend the design to larger systolic arrays.
- Introduce structured sparsity patterns and evaluate scalability.
- Explore more detailed power estimation techniques or ASIC-oriented flows.

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: Sparsity-Aware Systolic Array Architecture

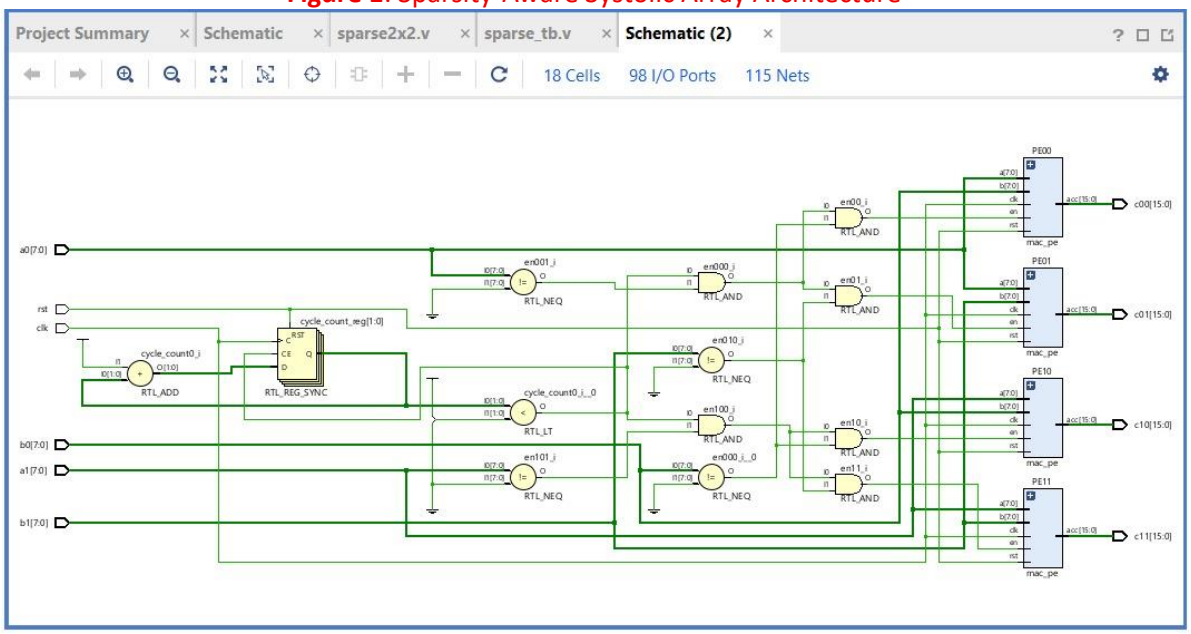

Figure 1 shows the RTL schematic of the sparsity-aware 2×2 systolic array. The processing elements (PEs) remain identical to the dense baseline and consist of MAC units. Sparsity is handled through additional control logic, where zero-detection comparators and cycle-counter-based gating generate per-PE enable signals. When either operand is zero, the corresponding PE is disabled, eliminating redundant MAC operations while preserving functional correctness and reducing internal switching activity.

Figure 2: Flattened RTL schematic of the sparsity-aware 2×2 systolic array after synthesis elaboration

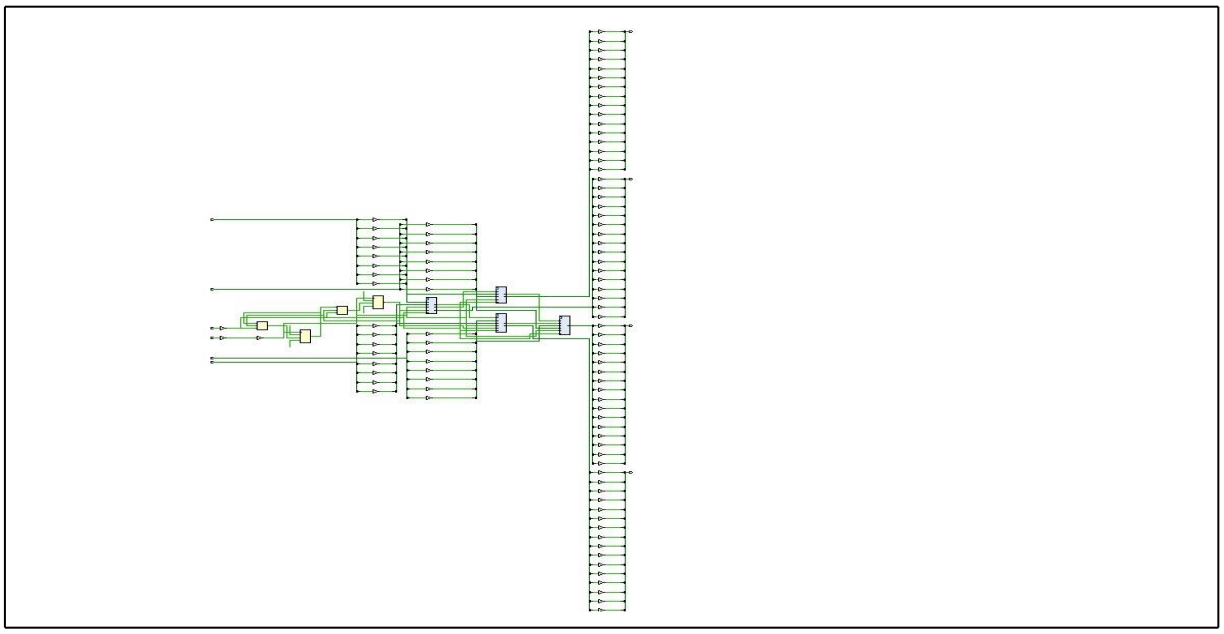

Figure 2 shows the flattened RTL schematic generated after synthesis elaboration in Vivado. The design includes datapath logic for the systolic array along with additional control logic for sparsity-aware execution. The expanded view confirms correct integration of zero-detection, enable gating, and accumulation control without modifying the core MAC datapath.

- **Simulation Results:**

Figure 3: Simulation waveform of sparsity-aware 2×2 systolic array under sparse input conditions

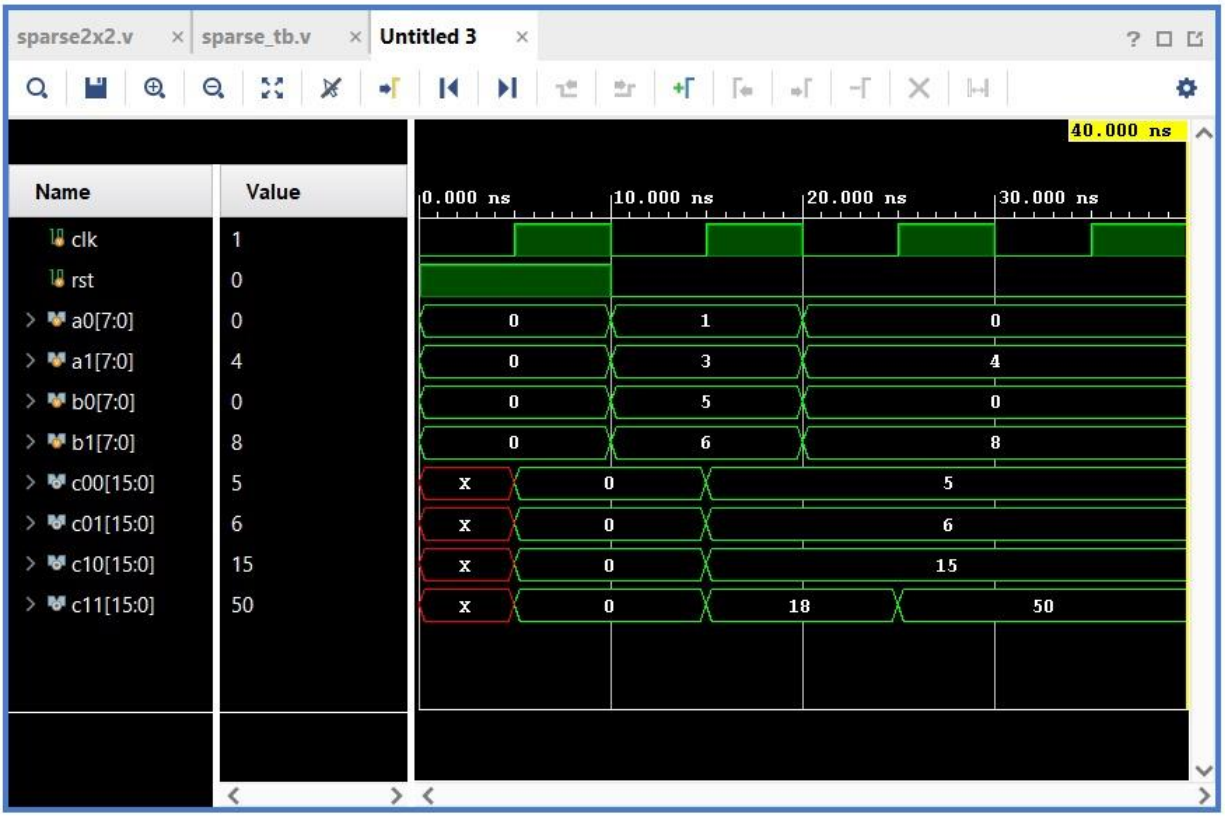

Figure 3 shows the simulation waveform of the sparsity-aware 2×2 systolic array using sparse input matrices. Zero-valued operands are intentionally introduced to evaluate sparse execution behavior. The output values match the expected matrix multiplication results, confirming functional correctness. During cycles where one or both operands are zero, accumulation does not occur, indicating that redundant MAC operations are skipped through enable gating. This validates sparsity-aware computation without affecting correctness.

5. Analysis

- **Key Findings:**

  - The sparse systolic design maintains functional correctness while reducing redundant computation.
  - For a 2×2 matrix multiplication:

    1. Dense baseline executes **8 MAC operations**
    2. Sparse design executes **5 MAC operations**

  - This represents a **37.5% reduction in MAC activity** for the same workload.

- **Insights or Learnings:**

  - Output waveforms alone do not reflect computational efficiency; internal enable gating provides better insight into activity reduction.
  - MAC operation count serves as a reliable proxy for dynamic power comparison, as each MAC involves multiplier, adder, and register switching.
  - Architectural optimizations at the control level can yield meaningful efficiency gains without altering the datapath.

- **Improvements or Modifications Needed:**

  - Scaling the design to larger arrays will better highlight the benefits of sparsity.
  - More advanced power analysis methods can be explored for quantitative energy estimation.
  - Include integration with ASIC-level simulation tools.

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.