International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM07321
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

RTL to GDS-II Implementation of Systolic Array-Based Matrix Multiplication and Reduction on Elastic CGRAS

Harshith U
Department of Electronics and Communication Engineering BMS College of Engineering
Bengaluru, India harshith.ec22@bmsce.ac.in

Veena M.B.
Department of Electronics and Communication Engineering
BMS College of Engineering
Bengaluru, India veenamb.ece@bmsce.ac.in

***Abstract*—**This paper presents the RTL-to-GDSII implementa-tion of a systolic array-based matrix multiplication and reduction architecture on an Elastic Coarse-Grained Reconfigurable Archi-tecture (CGRA). The proposed design focuses on achieving scal-able parallel computation using a regular systolic dataflow struc-ture integrated with reconfigurable processing elements. The ar-chitecture enables efficient matrix computation through pipelined multiply-accumulate operations while also supporting reduction functionality for high-throughput applications. The complete hardware architecture is modeled using Verilog/SystemVerilog and verified through functional simulation. A modular RTL design methodology is adopted to improve scalability and sim-plify hardware integration. The design is further implemented using a complete ASIC physical design flow including synthesis, floorplanning, power planning, placement, clock tree synthesis, routing, and physical verification. The implementation is carried out using SCL 180 nm standard-cell technology. Power rings, stripes, clock distribution, and routing optimization techniques are incorporated to achieve a physically realizable layout. Post-layout analysis validates the timing and routing feasibility of the proposed architecture. The regular structure of the systolic array and Elastic CGRA framework makes the design suitable for high-performance parallel processing applications such as signal processing, machine learning acceleration, and scientific computing.

Index Terms—Systolic Array, Elastic CGRA, RTL-to-GDSII, Matrix Multiplication, Physical Design, ASIC, VLSI, Parallel Processing

I. INTRODUCTION

Matrix multiplication is one of the most important opera-tions used in modern digital systems and high-performance computing applications. It is widely utilized in digital sig-nal processing, image processing, machine learning, scien-tific computation, and hardware accelerators. Since matrix multiplication involves a large number of repetitive arith-metic operations, conventional sequential architectures often experience higher latency and lower throughput. To improve computational efficiency, parallel hardware architectures are preferred for implementing such operations.

Systolic arrays provide an efficient solution for parallel matrix computation due to their regular structure and pipelined data flow. A systolic array consists of multiple processing ele-ments interconnected in a grid-like fashion where data moves rhythmically between neighboring elements. Each processing element performs multiply-accumulate operations simultane-ously, thereby enabling high-speed computation with efficient hardware utilization. The regular arrangement of processing elements also simplifies VLSI implementation and makes systolic arrays suitable for ASIC-based physical design.

In this work, a 3×3 systolic array architecture is pro-posed for matrix multiplication and reduction operations. The design consists of pipelined multiply-accumulate processing elements capable of performing parallel arithmetic computa-tions with reduced computational delay. In addition to matrix multiplication, reduction functionality is incorporated to sup-port accumulation-based operations required in computation-intensive applications. The architecture is modeled using Ver-ilog/SystemVerilog and verified through functional simulation. Unlike conventional works focused only on behavioral modeling or FPGA-level implementation, this paper presents a complete RTL-to-GDSII ASIC design flow for the proposed systolic array architecture. The implementation includes logic synthesis, floorplanning, power planning, placement, clock tree synthesis, routing, and physical verification using SCL 180 nm standard-cell technology. The proposed work demonstrates the practical hardware realization of a systolic-array-based com-putation engine while emphasizing efficient physical design and implementation methodology.

II. LITERATURE SURVEY

Matrix multiplication using systolic array architectures has gained significant importance due to its parallel processing capability and efficient hardware utilization. Systolic arrays are widely used in machine learning and signal processing applications because they provide high throughput with regular interconnections.

Srichandan *et al.* proposed a systolic-array-based matrix multiplication architecture on elastic CGRAs for machine learning acceleration [1]. Their work focused on improving matrix multiplication performance using optimized dataflow and architectural modifications for scalable computation.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM07321
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Libano *et al.* presented an efficient systolic-array-based matrix multiplication implementation on FPGA platforms with fault detection techniques [2]. The work highlighted the advantages of systolic architectures in achieving high-speed computation with efficient DSP utilization.

From the literature, it is observed that most existing works mainly focus on FPGA acceleration and machine learning applications. In this work, a 3×3 systolic array for matrix multiplication is implemented and analyzed through complete RTL-to-GDSII ASIC physical design flow using Verilog HDL and Cadence tools.

III. Proposed Architecture

A. Processing Element

The Processing Element (PE) is the basic computational unit of the proposed systolic array architecture. Each PE performs a Multiply-and-Accumulate (MAC) operation, where the incoming data elements from matrix *A* and matrix *B* are multiplied and accumulated to generate partial results. The matrix multiplication operation is mathematically represented as

$$C_{ij} = \sum_{k=0}^{N-1} A_{ik} \times B_{kj} \qquad (1)$$

where *A* and *B* are the input matrices and *C* is the resultant matrix.

In the proposed 3×3 systolic array, data from matrix *A* propagates horizontally across the rows, while data from matrix *B* propagates vertically across the columns. Each PE receives one element from both matrices, performs multiplica-tion, and adds the result to the accumulated partial sum stored internally. The computed partial sums are forwarded through the array in a pipelined manner until the final matrix result is obtained.

The PE consists mainly of a multiplier, accumulator register, and control logic. The multiplier generates the product of the input operands, while the accumulator stores interme-diate results during successive clock cycles. The operation is controlled using enable and reset signals to synchronize computation across all PEs. Due to the regular interconnection and local communication between neighboring PEs, the archi-tecture achieves efficient parallel computation with reduced routing complexity and improved throughput.

B. Systolic Array Architecture

The proposed architecture consists of a 3×3 systolic array designed for matrix multiplication operations. The systolic array is formed by arranging multiple Processing Elements (PEs) in a regular grid structure, where each PE performs local multiply-and-accumulate computations. The architecture enables parallel processing and pipelined data movement, thereby improving computational efficiency and throughput.

For matrix multiplication, the elements of matrix *A* are propagated horizontally across the rows of the array, while the elements of matrix *B* are propagated vertically across the

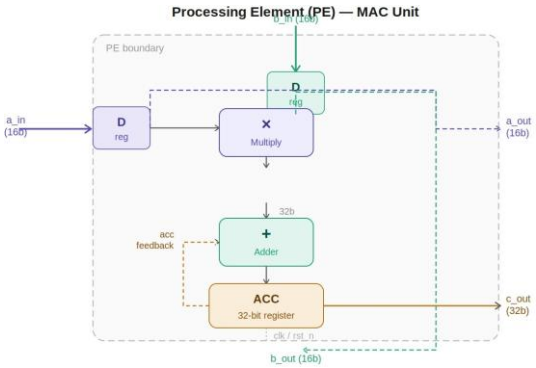

Fig. 1. Processing Element Architecture

columns. Each PE receives data from its neighboring PEs, performs multiplication, and accumulates the partial result.

The systolic architecture operates synchronously using a global clock signal. Data flows rhythmically through the array in successive clock cycles, allowing all PEs to operate simul-taneously. Due to the regular interconnection pattern and lo-calized communication between neighboring cells, the design achieves reduced routing complexity and efficient hardware utilization.

The proposed 3×3 systolic array is implemented using Verilog HDL and verified through simulation. The architecture is further synthesized and implemented using ASIC physical design flow to analyze placement, routing, and overall chip-level implementation.

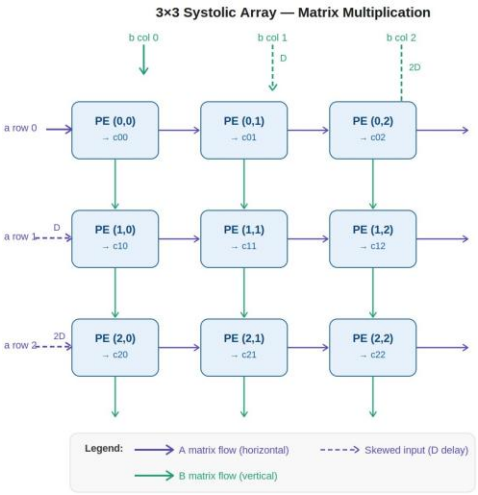

Fig. 2. Systolic Array Architecture

C. Elastic CGRA Integration

The proposed systolic array architecture can be integrated with Coarse-Grained Reconfigurable Array (CGRA) plat-forms to improve flexibility and computational efficiency. CGRAs consist of interconnected processing elements and

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM07321
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

programmable routing resources that support parallel execution of compute-intensive applications such as matrix multiplication and signal processing.

In the proposed work, the systolic array structure is designed in a modular manner, which enables easy mapping onto CGRA-based architectures. The regular communication pat-tern between neighboring Processing Elements (PEs) matches well with the routing organization of CGRAs, thereby reducing communication overhead and improving dataflow efficiency.

The integration of systolic arrays with CGRA architectures enables scalable computation and better hardware utilization for parallel processing applications. Since the architecture is highly regular and pipelined, it supports efficient implementation in recon g high throu

end

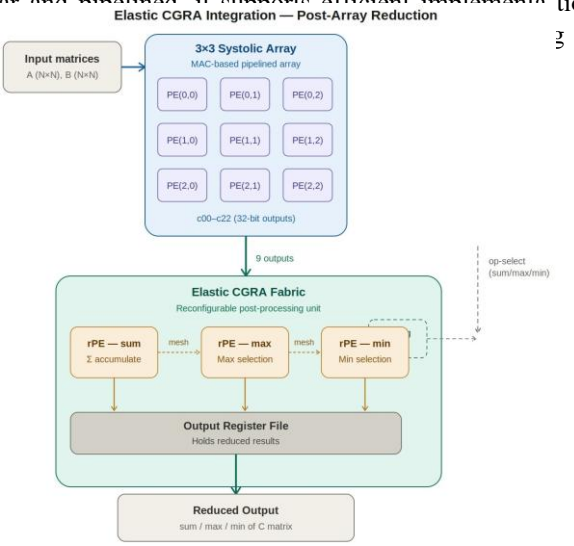

Fig. 3. Elastic CGRA Architecture

IV. RTL Design and Verification

A. RTL Design and Verification

The proposed 3×3 systolic array architecture was designed using Verilog HDL at the Register Transfer Level (RTL). The design consists of Processing Elements (PEs), control logic, accumulator registers, and interconnection modules required for matrix multiplication operations. The modular RTL design approach improves scalability and simplifies hardware implementation.

Each Processing Element performs Multiply-and-Accumulate (MAC) operations synchronously using clock-driven sequential logic. Control signals such as reset, enable, and start signals are used to coordinate the overall dataflow and computation across the systolic array. The RTL implementation was developed to ensure proper pipelined data propagation between neighboring PEs.

Functional verification of the design was carried out using a dedicated testbench environment. Different matrix input combinations were applied to validate the correctness of matrix multiplication results. Simulation results confirmed proper operation of the systolic array, including accurate accumulation of partial sums and generation of final output values.

The RTL simulation and verification were performed using Cadence Xcelium simulator before proceeding to synthesis and physical design stages.

B. RTL Snippet

```
always_ff @(posedge clk or negedge rst_n) begin
if (!rst_n) begin acc   <=32'd0;
a_out <= 16'd0; b_out <= 16'd0;
end else begin
acc<= acc + (a_in * b_in); a_out <= a_in;
b_out <= b_in;
        end
```

V. Physical Design Flow

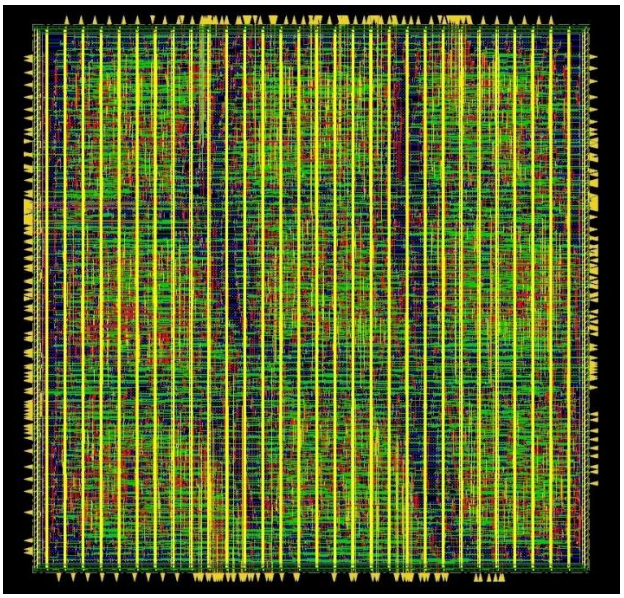

Fig. 4. GDS-II Layout

A. Physical Design Flow

After successful RTL verification, the proposed systolic ar-ray architecture was synthesized and implemented using ASIC physical design flow. The synthesized netlist was generated using Cadence Genus synthesis tool targeting SCL 180 nm standard cell technology. Timing constraints and design rules were applied during synthesis to optimize area, timing, and power characteristics of the design. The generated gate-level

netlist was then imported into Cadence Innovus for physical implementation.

The physical design flow included floorplanning, power planning, placement, clock tree synthesis (CTS), routing, and design rule verification. Power rings and power stripes were created to ensure proper power distribution across the chip. Standard cells were placed and optimized for routing efficiency and timing closure. Clock tree synthesis was performed to minimize clock skew and latency, followed by detailed routing to establish all signal connections. Finally, the layout was analyzed for design rule correctness and overall implementation quality before generating the final GDSII layout.

VI. Results and Analysis

The proposed systolic array architecture for matrix multiplication was successfully implemented and verified using Verilog HDL. Functional simulation results confirmed the correct operation of the Processing Elements (PEs), data propagation, and matrix multiplication functionality. The design produced accurate output matrices for different input test cases, validating the correctness of the Multiply-and-Accumulate (MAC) operations within the systolic array.

The verified RTL design was synthesized and implemented using Cadence Genus and Innovus tools targeting SCL 180 nm technology. Physical design stages such as floorplanning, power planning, placement, clock tree synthesis, and routing were completed successfully. The final layout demonstrated proper standard cell placement, routing connectivity, and power distribution across the design. The regular structure of the systolic array resulted in simplified routing and efficient hardware implementation suitable for parallel computation applications.

A. Implementation Results

TABLE I
Implementation Results

Parameter	Value
Technology Node	SCL 180nm
Operating Frequency	20 MHz
Total Area	453,287.40 μm^2
Total Power	14.3 mW
Timing Slack	+15.77 ns
No. of Pipeline Stages	9

B. Comparison with Existing Work

The comparison presented in Table II is based on observa-tions from existing systolic-array-based matrix multiplication architectures reported in the literature, including [1] and [2]. Most previous works mainly focus on FPGA implementation and algorithm-level optimization, whereas the proposed work demonstrates complete RTL-to-GDSII ASIC implementation using Cadence design tools.

TABLE II
Comparison with Existing Systolic Array Architectures

Feature	Previous Work	Proposed Work
Implementation	RTL / FPGA Based	Full RTL-to-GDSII
Architecture	Generic Systolic Array	3×3 Systolic Array
Computation	Sequential / Partial Parallel	Fully Parallel MAC
Processing Element	Basic MAC Unit	Optimized PE Design
Pipeline Structure	Limited	Pipelined Dataflow
Throughput	Moderate	High
Routing Structure	Irregular	Regular Interconnect
Physical Design	Not Discussed	Complete ASIC Flow
Technology	FPGA Focused	SCL 180 nm ASIC
Verification	Functional Only	RTL + Physical Verification

VII. Conclusion

In this work, a 3×3 systolic array architecture for matrix multiplication was successfully designed and implemented using Verilog HDL. The proposed architecture utilized multiple Processing Elements (PEs) operating in parallel to per-form efficient Multiply-and-Accumulate (MAC) operations. Functional verification confirmed the correctness of matrix multiplication and proper data propagation through the systolic array structure.

The verified RTL design was synthesized and implemented through complete ASIC physical design flow using Cadence Genus and Innovus tools with SCL 180 nm technology. Physical design stages including floorplanning, power planning, placement, clock tree synthesis, and routing were completed successfully. The regular and modular architecture of the systolic array resulted in efficient hardware implementation with simplified routing and improved computational throughput. The proposed design demonstrates the suitability of systolic array architectures for parallel processing and matrix-based computational applications.

VIII. Future Work

The proposed 3×3 systolic array architecture can be further extended to support larger matrix sizes such as 4×4, 8×8, and higher dimensions for improved computational capabil-ity. Additional optimization techniques can be applied to reduce area, power consumption, and latency during ASIC implementation. The architecture can also be integrated with advanced CGRA and AI accelerator platforms to support machine learning and signal processing applications. Future work may include implementation using lower technology nodes, incorporation of fault-tolerant techniques, and development of dynamically reconfigurable systolic architectures for high-performance computing systems.

Acknowledgement

The authors would like to thank the Chips to Startup (C2S) programme for providing access to VLSI EDA tools and Department of Electronics and Communication Engineering, BMS College of Engineering for continuous support.

REFERENCES

[1] S. Srichandan, S. K. Nandy, and P. P. Chakrabarti, "Systolic Array Based Matrix Multiplication and Reduction on Elastic CGRAs," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 43, no. 2, pp. 415–428, 2024.
[2] F. Libano, M. Sonza Reorda, and M. Violante, "Efficient Fault Detection for Matrix Multiplication Using Systolic Arrays on FPGA," IEEE Transactions on Nuclear Science, vol. 67, no. 7, pp. 1458–1465, 2020.
[3] H. T. Kung and C. E. Leiserson, "Systolic Arrays for VLSI," Sparse Matrix Proceedings, pp. 256–282, 1978.
[4] N. P. Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," Proceedings of the 44th Annual International Sympo-sium on Computer Architecture, pp. 1–12, 2017.
[5] Y. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 127–138, 2017.
[6] H. T. Kung, "Why Systolic Architectures?" IEEE Computer, vol. 15, no. 1, pp. 37–46, 1982.
[7] C. E. Leiserson and H. T. Kung, "Systolic Arrays for VLSI," Introduction to VLSI Systems, pp. 271–292, 1979.
[8] M. Satheesh Kumar and P. Muralidhar, "FPGA Implementation of Matrix Multiplication Using Systolic Array Architecture," International Journal of Engineering Research and Technology, vol. 4, no. 5, pp. 512–516, 2015.
[9] J. Dean, "Challenges in Building Large-Scale Machine Learning Sys-tems," Proceedings of the IEEE, vol. 103, no. 2, pp. 176–196, 2015.
[10] Y. Ma, N. Suda, Y. Cao, J. Seo, and S. Vrudhula, "Scalable and Modularized RTL Compilation of Convolutional Neural Networks onto FPGA," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 11, pp. 2556–2567, 2018.