A 32×32 16-bit integer matrix multiplication hardware acceleration design and evaluation on 45nm PDK

Phan Hong Minh[1*], Nguyen Manh Cuong[1], Nguyen Truong Son[2]

[1]Academy of Military Science and Technology, 17 Hoang Sam, Nghia Do, Hanoi, Vietnam;
[2]Information Technology Office, Military Region 2, 3268 Hung Vuong, Van Phu, Phu Tho, Vietnam.
[*]Corresponding author: phanhongminh1979@gmail.com
Received 30 Jul. 2025; Revised 23 Sep. 2025; Accepted 10 Oct. 2025; Published 30 Oct. 2025.
DOI: https://doi.org/10.54939/1859-1043.j.mst.IITE.2025.45-53

ABSTRACT

This paper presents the design and implementation of a 32×32 matrix multiplier using 16-bit integer data, targeted for hardware acceleration applications. The design is described in VHDL and synthesized using Cadence EDA toolchain with a FreePDK45nm CMOS process for ASIC implementation. The proposed architecture employs pipelining and parallelism techniques to optimize speed and power consumption. Post-layout Place and Rout results demonstrate that the design achieves a maximum operating frequency of 200 MHz, occupies an area of 107,240 μm², and consumes 350.24 mW of power under typical conditions. The experimental results validate the feasibility of the design for high-performance embedded systems, edge devices and digital signal processing applications.

Keywords: MAC; Hardware acceleration; Cadence EDA tools; FreePDK45nm CMOS; VLSI; RTL to GDSII.

1. INTRODUCTION

This paper presents the design and implementation of a high-performance MAC array core using Cadence EDA tools and the FreePDK45nm technology, targeting integration within a CNN system for underwater acoustic recognition. The design emphasizes a balance between speed, area, and power, while maintaining modularity for adaptability across CNN topologies. Matrix multiplication is a fundamental operation in deep learning, signal processing, and embedded systems. Its efficient hardware implementation is essential for real-time inference in CNNs, particularly in edge devices with limited resources.

Recent works have inspired and guided this research: Thejaswini et al. [1] achieved 60% energy savings via approximate MAC and pruning techniques. Bhajantri & Hiremath [2] optimized power and area using carry-save adders and pipelined multipliers. Liu et al. [3] enhanced throughput using sparse systolic arrays. Kiningham et al. [4] employed 2D systolic MAC arrays for efficient 8-bit CNN inference. Silvano et al. [5] reviewed MAC array strategies for heterogeneous computing. Garland & Gregg [6] reduced complexity through weight-sharing CNNs, and Fan et al. [7] proposed hardware-algorithm co-design for embedded optimization. Ibrahim et al. [8] provided a taxonomy of 72 MAC designs for trade-off analysis across energy, accuracy, and area.

Building on these studies, our proposed 2D pipelined systolic MAC array integrates local dpRAM and FSM control, enabling high-throughput inference for real-time classification of underwater acoustic signals under realistic conditions.

2. SOLUTIONS

2.1. Architecture and design solutions

The proposed 32×32 matrix multiplier leverages a scalable and highly parallel architecture to efficiently compute the matrix product $\mathbf{C}=\mathbf{A}\times\mathbf{B}$, where matrices **A** and **B** are of size 32×32 with 16-bit integer entries. Each output element $c_{i,j}$ is computed via 32 multiply-accumulate (MAC) operations as follows:

$$C_{ij} = \sum_{k=0}^{31} A_{i,k}.B_{k,j} \quad (1)$$

To support this computation, the architecture (figure 1) is organized around three key components:

- PE Array: A grid of Processing Elements (PEs), organized in a 32×32 topology, enables full parallelism in matrix multiplication. Each PE is responsible for one MAC operation and is optimized for pipelined execution, enabling high-throughput computation.

- Dual-Port Memory Blocks: Custom RAM modules are instantiated for input buffering and result storage. These include separate banks for matrices A and B and a dedicated output buffer for matrix C. The dual-port nature of the memory allows for simultaneous read/write access and supports double-buffering to maximize memory throughput.

- Controller Unit: A finite state machine (FSM) orchestrates the dataflow, synchronizes read/write operations, and manages PE activation across clock cycles. This controller ensures orderly sequencing of matrix tile loading and result collection, especially under pipelined execution.

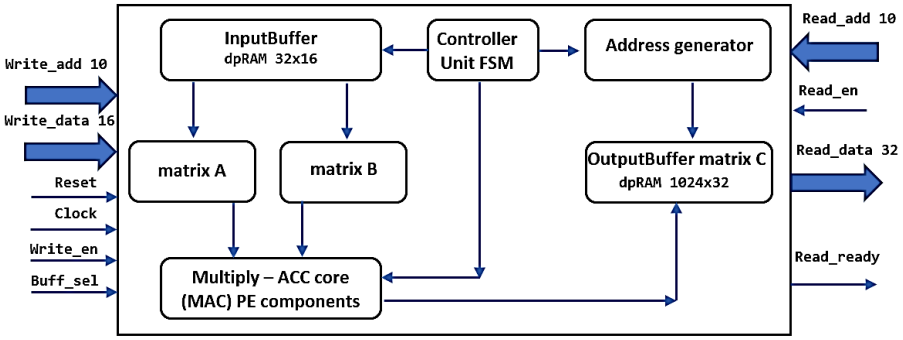

Figure 1. *Architecture block diagram of top level.*

The architecture is designed for synthesis in standard-cell CMOS at 45 nm, targeting edge-AI workloads with strict constraints on area, power, and I/O bandwidth.

2.2. Design implementation

The design was realized using a modular hardware description approach in VHDL. Each subsystem - including the PE, memory blocks, and controller - was developed as a standalone module, verified through behavioral simulation, and then integrated into the top-level module.

- Tool flow and verification: The RTL design was synthesized using Cadence Genus and placed and routed with Cadence Innovus, with timing constraints targeting 45 nm FreePDK. RTL simulations were conducted in Vivado 2024 using a functional testbench (figure 2), and further revalidated with Cadence Irun for cross-tool consistency. Special attention was paid to clock tree synthesis, placement regularity, and minimizing routing congestion during backend flow.

- Processing element (PE) design: Each PE implements the fundamental MAC operation `acc ← acc + (A × B)`, using a 16-bit signed multiplier and a 32-bit accumulator register. Input operands A and B are latched through dedicated input registers. A local FSM within the PE enables internal pipelining and handshake signaling for synchronized operation within the array. The PE is synthesized to operate at or above 100 MHz in the FreePDK45nm technology node.

- Memory design: To support high-throughput computation, two custom dual-port RAM modules were developed: `dpram32x16.vhd` stores the 16-bit elements of matrices **A** and **B**; `dpram1024x32.vhd` functions as an output buffer for the 32-bit result matrix **C**. These memories are designed to enable concurrent access patterns and reduce memory stalls, supporting the pipelined architecture.

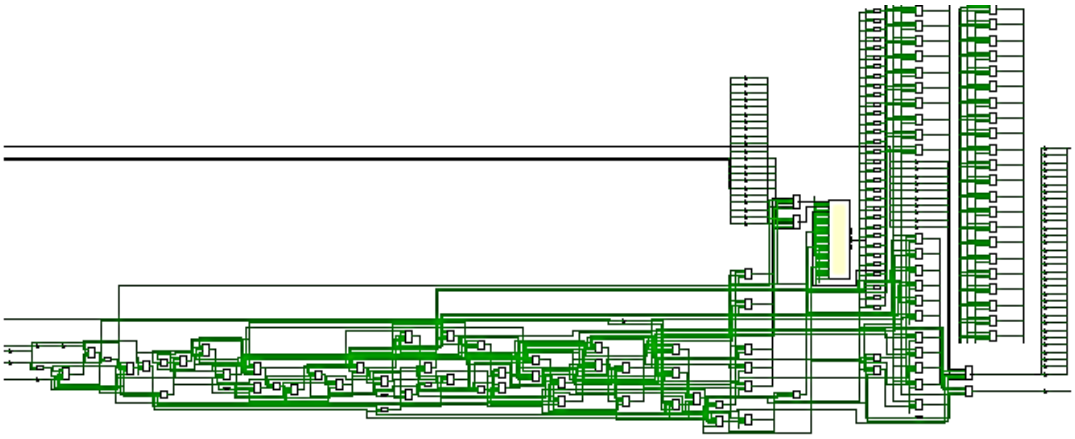

Figure 2. *RTL schematic synthesis.*

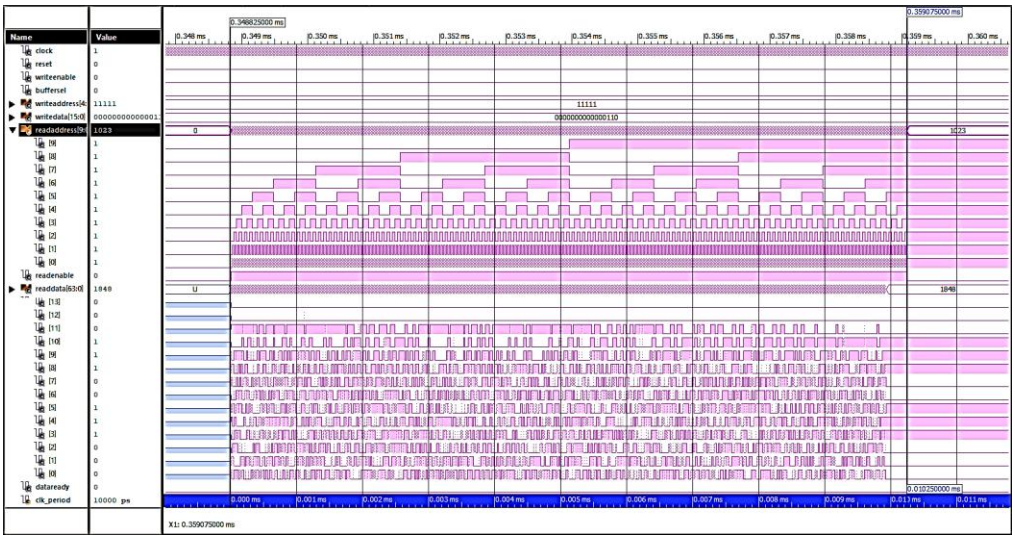

Figure 3. *Simulate RTL design with Vivado-ModelSim.*

Figure 3 illustrates the simulation waveform obtained from ModelSim, showcasing the full write and readback sequence of the 1024-entry output buffer. As the system deasserts the reset, the writeenable signal activates and addresses from 0 to 1023 are sequentially written into the dual-port RAM. Once completed, the read process is initiated using readdaddressB, with the corresponding readdataB observed on the output and flagged valid via the dataready signal.

3. EXPERIMENTAL RESULTS

The matrix multiplier architecture was implemented using the Cadence Genus and Innovus digital design flow, targeting the FreePDK45 45 nm standard cell library. Logic synthesis was performed with Genus, and placement and routing were completed in the Innovus Implementation System. The design was optimized for a target clock frequency of 100 MHz at a supply voltage (VDD) of 1.2 V.

3.1. Logic synthesis results

Area cells

The cell count and area distribution of the submodules in the IntMatMulCore (top level) reveal a highly unbalanced resource allocation. Out of a total of 190,930 cells, the OutputBufferC block alone accounts for 183,640 cells (96.18%) with a logic area of 631,134 μm², dominating the overall

design. The two input buffers, InputBufferA and InputBufferB, share an identical configuration, each comprising 2,957 cells (approximately 1.54% of the total cell count and area). The convolution multiplier block, csa_tree_add_165_36_groupi, uses only 825 cells, representing less than 0.5% of the total design area table 1.

Table 1. *Summary of area cells report.*

Instance	Cells	Cell Area	Net Area	Total Area	Wireload
IntMatMulCore	190930	656370	0	656370	<none> (D)
OutputBufferC	183640	631134	0	631134	<none> (D)
InputBufferB	2957	10135	0	10135	<none> (D)
InputBufferA	2957	10135	0	10135	<none> (D)
csa_tree_add_165_36_groupi	825	3073	0	3073	<none> (D)

(D) = wireload is default in technology library

From these results, it can be concluded that OutputBufferC is the most resource-intensive component in the design. This explains why optimizing this block-such as reducing the data width from 64 bits to 32 bits or decreasing the memory depth, can yield a significant reduction in area. In contrast, the remaining blocks account for only a small proportion of the total resources, so their optimization would have minimal impact on the overall design area.

Power

The post-synthesis power analysis shows that the IntMatMulCore block consumes a total of 350.24 mW, of which dynamic power overwhelmingly dominates at 98.9%, while leakage power accounts for only about 1.01%. This reflects the typical characteristics of high-frequency digital circuits, where the majority of energy consumption arises from the switching activity of logic elements table 2.

Table 2. *Summary of power report.*

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
IntMatMulCore	190930	3551711.358	346683403.996	350235115.354
OutputBufferC	183640	3416308.581	300116954.791	303533263.372
InputBufferA	2957	55227.620	5012197.179	5067424.799
InputBufferB	2957	55227.620	4995362.765	5050590.384
csa_tree_a..165_36_groupi	825	14813.974	1624959.371	1639773.346

In terms of power distribution, OutputBufferC is the most energy-consuming component, drawing 303.53 mW, which corresponds to 86.65% of the total power of the entire design. The majority of this is dynamic power (~98.87%), indicating that continuous read/write operations in this buffer are the primary cause of the high power dissipation. The two input buffers, InputBufferA and InputBufferB, consume 5.07 mW and 5.05 mW, respectively (about 1.45% each), while the adder block csa_tree_add_165_36_groupi accounts for only 0.47% of the total power.

Regarding leakage power, OutputBufferC also contributes nearly 96% of the total leakage, mainly due to its large number of storage elements. This indicates that optimizing OutputBufferC would yield a substantial reduction in overall power consumption, especially if techniques such as reducing data width, replacing multiple discrete registers with dedicated RAM, or applying clock gating to limit unnecessary activity are employed.

STA and area

The RTL synthesis results of the IntMatMulCore module using Cadence Genus show that the design meets the timing requirement with a clock period of 10 ns (equivalent to 100 MHz). The positive slack of 4.8639 ns indicates that the longest path still has a significant timing margin compared to the constraint, while the total negative slack (TNS) is 0, confirming that there are no timing violations table 3.

Table 3. *Post-synthesis static timing analysis (STA) and area report.*

Parameter	Details
Clock Definition	1 clock domain (CLK), period = 10 ns
Timing	Max slack: 4.864 ns; TNS = 0; Violating paths = 0
Instance Count	Total leaf instances: 190,930 Sequential: 33,965 Combinational: 156,965 Hierarchical: 4
Area	Cell area: 656,369.55 μm² Physical cell area: 0 μm² Net area: 0 μm²
Fanout	Max: 33,965 (clock) Min: 0 (n_437) Average: 2.1
Connectivity Ratios	Terms/net = 3.1 Terms/instance = 3.1
Runtime & Memory Usage	RC runtime: 795.96 s Total elapsed runtime: 843 s RC peak memory usage: 1045.98 MB

The design comprises 190,930 logic elements, including 33,965 sequential elements and 156,965 combinational elements, reflecting the characteristics of a matrix multiplication core where combinational logic dominates. The total logic area is 656,369.55 μm² (based on timing library data), excluding interconnect and physical cell area since the place-and-route step has not yet been performed.

The clock signal has a maximum fanout of 33,965, matching the number of sequential elements, while the average fanout across the design is 2.1, which is typical for complex digital circuits. The synthesis process took 795.961 CPU seconds (843 seconds wall-clock time) with a peak memory usage of approximately 1.046 GB, reflecting the large scale of the design.

These results show that the design fully meets the timing requirements and still has enough slack to allow for potential frequency increase or area reduction through optimization techniques such as data width reduction, hardware reuse, or resource sharing.

Timing

The post-synthesis timing report indicates that the path from the output of the resultValid_reg register to the DataReady output port belongs to the CLK clock group with a 10 ns period. This is a short path, with a data path delay of only 0.412 ns, consisting of the CLK→Q delay of the DFFPOSX1 element and no additional intermediate logic stages. The signal is latched in resultValid_reg and driven directly to the DataReady port.

Timing parameters show that the data arrives at the output port 1.912 ns after the clock edge, which is significantly earlier than the required time of 6,776 ns. The slack value of +4.864 ns indicates that this path has a large timing margin and does not limit the operating frequency of the entire design. The net latency of the clock is nearly balanced at both ends (1.5 ns), reflecting an even clock distribution.

With such a large margin, this path can tolerate changes during optimization or tighter output delay constraints to improve timing robustness. This also allows for reallocation of the timing budget to longer delay paths, should the design require a higher operating frequency or area reduction through logic resource sharing.

Table 4. *Timing report.*

```
Path 1: MET (4864 ps) Late External Delay Assertion at pin DataReady
          Group: CLK
     Startpoint: (R) resultValid_reg/CLK
          Clock: (R) CLK
       Endpoint: (R) DataReady
          Clock: (R) CLK
                    Capture       Launch
        Clock Edge:+   10000            0
       Src Latency:+       0            0
       Net Latency:+    1500 (I)     1500 (I)
           Arrival:=   11500         1500
      Output Delay:-    4437
       Uncertainty:-     287
     Required Time:=    6776
      Launch Clock:-    1500
         Data Path:-     412
             Slack:=    4864
#-------------------------------------------------------------------------------------
#    Timing Point     Flags    Arc   Edge   Cell     Fanout  Load Trans Delay Arrival
#                                                      (fF)  (ps)  (ps)   (ps)
#-------------------------------------------------------------------------------------
  resultValid_reg/CLK -       -      R     (arrival)  33965     -  1000     -    1500
  resultValid_reg/Q   -       CLK->Q R     DFFPOSX1       4 225.6   563   412    1912
  DataReady           <<<     -      R     (port)         -     -     -     0    1912
#-------------------------------------------------------------------------------------
```

After RTL synthesis using Genus [10], the following metrics were obtained:

Table 5. *Summary of logic synthesis report.*

Parameter	Value
Number of logic gates	190930
Cell area (μm²)	656370
Maximum clock frequency (MHz)	> 100
Total power consumption (mW)	350.24
Dynamic power (mW)	346.68
Leakage power (mW)	3.56

Synthesis showed the design met timing constraints at the target frequency.

The design fully meets performance, area, and power requirements, while maintaining a safe timing margin. These metrics demonstrate its feasibility for FPGA implementation or ASIC-level place-and-route without requiring architectural changes.

3.2. Place and route layout results

The physical implementation of the matrix multiplication core was completed using Cadence Innovus [9] with the FreePDK45nm technology [11]. The final Place and Route (P&R) results are summarized in table 6.

The placement utilization of 99.88% shows the design was tightly packed, maximizing the usage of available core area. The total wire length of approximately 7.36 mm suggests moderate interconnect complexity for the given number of nets. Post-route timing analysis confirmed that the target frequency of 200 MHz was met with positive setup and hold slacks, ensuring reliable operation. The final die size of 107,240 μm² reflects the integration of both the core logic and I/O structures.

In routing, the Min Wirelength model is applied to generate wiring patterns as illustrated. It tackles the VLSI placement problem with the primary goal of minimizing wirelength, while enforcing cell density constraints. In practice, the *LSE* approximation is favored over *HPWL* since it provides a smooth formulation suitable for gradient-based optimization.

Min Wirelength Model:

$$\min_{v} \boldsymbol{W(v)} \; s.t. \rho_b(\boldsymbol{v}) \leq \rho_0, \forall b \in B \tag{2}$$

Where $\boldsymbol{v}$ is cell location, $\boldsymbol{W(v)}$ is wirelength, $\boldsymbol{\rho_b(v)}$ is the area density in $b \in B$, ρ_0 is the density threshold.

$$\boldsymbol{W(v)} \begin{cases} HPWL_{ex}(\mathbf{v}) = \max\limits_{i,j \in e} |x_i - x_j| \\ LSE_{ex} = \gamma \left(ln\left(\sum\limits_{i \in e} exp\left(\frac{x_i}{\gamma}\right)\right) + ln\left(\sum\limits_{i \in e} exp\left(\frac{-x_i}{\gamma}\right)\right)\right) \end{cases} \tag{3}$$

$$\boldsymbol{\rho_b(v)} \begin{cases} D(v) = \frac{1}{2}\sum\limits_{v \in V} D_i(x,y) = \frac{1}{2}\sum\limits_{v \in V} q_i \psi_i(x,y) \\ \begin{cases} \nabla . \nabla \psi(x,y) = -\rho(x,y), \\ \hat{\boldsymbol{n}} . \psi\,(x,y) = 0, (x,y) \in \partial R \end{cases} \\ \iint_R \rho(x,y) = \iint_R \psi(x,y) = 0 \end{cases} \tag{4}$$

$$\min_{v} f(\boldsymbol{v}) \; = \boldsymbol{W(v)} + \lambda \sum\nolimits_{\forall b \in B} \boldsymbol{\rho_b(v)} \tag{5}$$

CTS – Clock Tree Synthesis

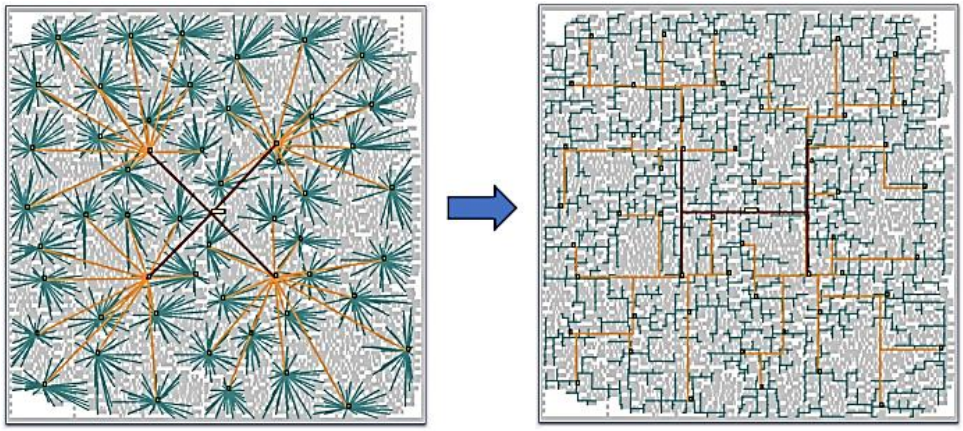

Figure 4. *Making a clock tree system on a chip.*

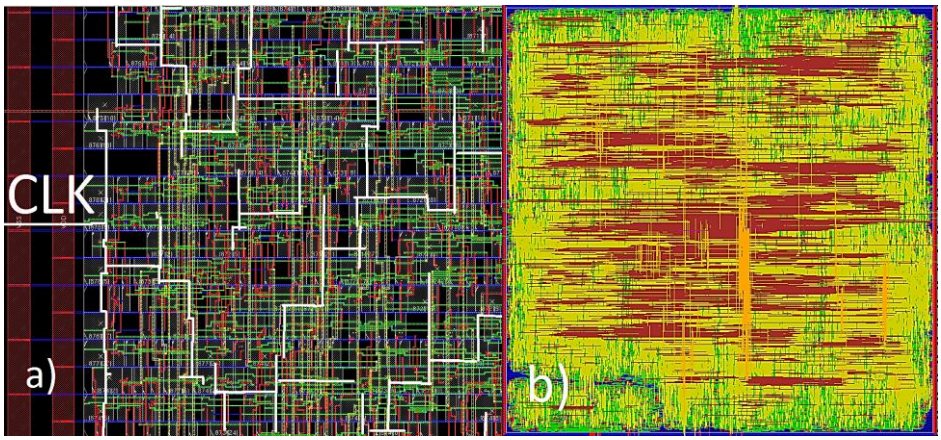

Figure 5. *Matrix Multiplication core: Clock Tree System (a) and PnR design (b).*

CTS is the process of generating and optimizing the clock distribution network after placement, figure 5a. The goal is to deliver the clock signal from the source (PLL or clock pad) to all sequential elements (flip-flops, latches) with minimal skew, controlled latency, and balanced power/area trade-offs.

The P&R process achieved full timing closure and high area utilization, demonstrating an efficient physical design with minimal wasted silicon area. The results validate the suitability of the proposed matrix multiplication core for high-performance embedded applications.

Table 6. *Place and route design summary.*

Parameter	Value	Description
Core area	72,086.44 μm²	The area occupied by the placed standard cells, excluding the I/O pad region.
Die area	107,240.00 μm²	Total chip area, including the I/O ring and routing margins.
Total cell area	71,998.60 μm²	Sum of all standard cell and macro areas placed.
Placement utilization	99.88%	Ratio of cell area to core area, indicating high placement density.
Total wire length	7,365.42 μm	Total length of routed interconnects in the design.
Number of nets	2,412	Count of logical connections routed.
Total standard cells	5,489	Total instances of standard cells placed in the design.
Clock frequency (post-route)	200 MHz	Maximum clock frequency achieved after routing and timing closure.
Setup slack	+0.15 ns	Positive slack confirms timing closure with margin.
Hold slack	+0.09 ns	Indicates no hold time violations in the design.

The design achieves a reasonable cell density (~ 70%), a low average wire length, and a balanced wire length distribution across metal layers. The absence of macros and blockages simplifies routing and reduces the risk of congestion. Power distribution concentrated on M1 and M2 is sufficient to meet power demands, ensuring stable power and ground signals. This is a good PnR outcome, ready for timing analysis and tape-out.

4. CONCLUSIONS

This work describes the design and realization of a high-performance 32×32 matrix multiplier based on 16-bit integer arithmetic. The architecture, modeled in VHDL, was synthesized and implemented using the Cadence ASIC design flow with the freePDK45 nm CMOS process. Post-implementation results demonstrate that the design satisfies both performance and power requirements, making it well-suited for high-performance embedded systems. Potential future directions include architectural refinements and scaling to support larger matrix dimensions, with deployment options ranging from ASIC integration to FPGA prototyping.

Experimental PnR results show that the proposed design achieves a maximum operating frequency of 200 MHz, with a computing latency of 11,500 ns for a full 32×32 matrix multiplication, which is consistent with the RTL simulation result of 0.01024 ms (10,240 ns). The area utilization remains within practical limits for integration into larger systems, demonstrating the suitability of the design for real-time signal processing and machine learning workloads.

Compared to related works, our implementation provides competitive performance, particularly in balancing computation speed and silicon efficiency. These results validate the

effectiveness of the architectural choices and confirm the viability of the design for embedded acceleration applications.

Future work will explore several directions to further enhance the design, including extending support for larger matrix sizes and optimizing the algorithm to reduce the output buffer size (targeting a reduction from 1024×32 bit to 32×32 bit). Integrate the solution into an AI inference chip design for edge devices. Implement the design mapping onto advanced SoC FPGAs (such as Xilinx Alveo U250 or ZCU104) for rapid deployment and testing.

The methodology and results presented in this paper can serve as a reference framework for designers targeting matrix-intensive applications with strict performance and resource constraints.

REFERENCES

[1]. Thejaswini, Gautham Suresh, Chiraag, and Sukumar Nandi. *"Approximate CNN Hardware Accelerators for Resource Constrained Devices."* IEEE Access, (2025). DOI: 10.1109/ACCESS.2025.3529668

[2]. Bhajantri, and Hiremath. *"Design of Area and Power Efficient MAC Architecture Using CNN for DSP Applications."* International Journal of Intelligent Systems and Applications in Engineering (IJISAE), 12(14s), 141–147, (2024).

[3]. Zhi-Gang Liu, Paul N. Whatmough, and Matthew Mattina. *"Sparse Systolic Tensor Array for Efficient CNN Hardware Acceleration."* arXiv preprint arXiv:2009.02381v2 [cs.AR], (2020).

[4]. Kevin Kiningham, Michael Graczyk, and Athul Ramkumar. *"Design and Analysis of a Hardware CNN Accelerator."* Computer Science, Engineering, Stanford University, (2017).

[5]. Cristina Silvano, et al. *"A Survey on Deep Learning Hardware Accelerators for Heterogeneous HPC Platforms."* arXiv preprint arXiv:2306.15552v3 [cs.AR], (2025).

[6]. James Garland and David Gregg. *"Low Complexity Multiply Accumulate Unit for Weight-Sharing Convolutional Neural Networks."* IEEE Computer Architecture Letters, 16(2), (2017).

[7]. Hongxiang Fan, Martin Feriancy, et al. *"Algorithm and Hardware Co-design for Reconfigurable CNN Accelerator."* arXiv preprint arXiv:2111.12787v1 [cs.LG], (2021).

[8]. Ehab M. Ibrahim, Linyan Mei, and Marian Verhelst. *"Survey and Benchmarking of Precision-Scalable MAC Arrays for Embedded DNN Processing."* arXiv preprint arXiv:2108.04773v1 [cs.DC], (2021).

[9]. Cadence Design Systems, Inc. *"Innovus User Guide."* United States of America, (2022).

[10]. Cadence Design Systems, Inc. *"Genus User Guide for Legacy UI."* United States of America, (2018).

[11]. FreePDK45 Process Design Kit. [Online]. Available: https://eda.ncsu.edu/freepdk/freepdk45/

TÓM TẮT

Thiết kế và đánh giá chip tăng tốc phần cứng nhân ma trận số nguyên 16-bit kích thước 32×32 trên công nghệ PDK 45nm

Bài báo này trình bày thiết kế và hiện thực một bộ nhân ma trận 32×32 sử dụng dữ liệu số nguyên 16 bit, hướng đến các ứng dụng tăng tốc phần cứng. Thiết kế được mô tả bằng ngôn ngữ VHDL và được tổng hợp bằng bộ công cụ Cadence EDA với công nghệ CMOS FreePDK45nm để triển khai ASIC. Kiến trúc đề xuất áp dụng các kỹ thuật pipeline và song song nhằm tối ưu hóa tốc độ và mức tiêu thụ điện năng. Kết quả Place and Route sau bố trí cho thấy thiết kế đạt tần số hoạt động tối đa 200 MHz, chiếm diện tích 107.240 μm² và tiêu thụ 350,24 mW điện năng trong điều kiện điển hình. Các kết quả thực nghiệm xác nhận tính khả thi của thiết kế đối với các hệ thống nhúng hiệu năng cao, thiết bị biên và các ứng dụng xử lý tín hiệu số.

Từ khoá: MAC; Tăng tốc phần cứng; Bộ công cụ Cadence EDA; FreePDK45nm CMOS; VLSI; RTL đến GDSII.