International Journal of Advanced Trends in Engineering and Management

(ISSN 2583-7052)

pp. 202-208, April

Low Power VLSI Implementation of a Hybrid Multiplier Using Modified Compressor Structures

S.Narmatha[1], C.Abinaya[2*]

[1]PG, Department of Electronics and Communication Engineering, Mother Terasa College of Engineering and Technology, Pudukkottai-01.

[2]PG, Department of Electronics and Communication Engineering, Mother Terasa College of Engineering and Technology, Pudukkottai-01.

[1]Corresponding Author E-mail: narmiselvan95@gmail.com

ABSTRACT: Multi-digit multiplication plays a vital role in modern digital systems, supporting applications such as numerical computation, chaos-based arithmetic, and hardware validation processes. With the increasing demand for high-speed data processing in areas like image processing and secure communication systems, there is a strong need for arithmetic units that achieve both high performance and low power consumption. This is particularly important in cryptographic applications where chaos-based techniques are employed for enhanced security. This work presents the hardware design and VLSI implementation of a multi-digit multiplier, focusing on optimizing area, speed, and energy efficiency. Novel compressor-based multiplier architecture is proposed to improve partial product reduction and minimize power dissipation. The design is evaluated in terms of performance metrics to ensure its suitability for advanced digital applications. Furthermore, the proposed architecture is adaptable for FPGA realization, providing flexibility in prototyping and practical deployment. The results offer significant guidance for system designers in selecting efficient multiplier structures for high-performance applications.

1. Introduction

A multiplier constitutes a fundamental computational unit in digital signal processing (DSP) systems, where it significantly influences overall system performance. It is extensively utilized in key DSP applications such as digital filtering, digital communication systems, and spectral analysis, where repeated arithmetic operations are required. With the growing demand for portable and battery-powered electronic devices, power consumption has become a critical design constraint in modern DSP architectures. Due to their inherent computational complexity and frequent operation at high clock frequencies, multipliers contribute substantially to system delay and energy consumption.

Therefore, optimizing multiplier design in terms of speed and power efficiency is essential to meet performance requirements. Reducing propagation delay not only enhances processing speed but also improves the overall throughput of the system. Consequently, efficient multiplier architectures are necessary to achieve a balance between high performance and low power dissipation in advanced DSP applications. Multiplication is a computationally intensive operation that significantly affects the overall system performance.

In many digital systems, the execution speed of multiplication largely determines the efficiency of complex computations. Since multiplication involves multiple sequential and parallel arithmetic operations, it often introduces higher latency compared to simpler operations such as addition.

Consider two unsigned binary numbers X and Y, having bit widths of M and N, respectively. To clearly describe the multiplication process, both operands can be represented in their binary forms, where each number is expressed as a sum of weighted binary digits. This representation provides a structured approach for understanding partial product generation and accumulation, which forms the basis of most digital multiplier architectures.

$$\sum_{i=0}^{M} X=\sum Xi \qquad (1)$$

$$\sum_{j=0}^{N} Y=\sum Yj \qquad (2)$$

With Xi, Yj ∈ {0,1}. The multplication operation is

$$Z=\sum_{k=0} X_K \; K=M+N-1(3) \qquad (3)$$

$$=\sum_{i=0}^{M}\left(\sum X_i\, Y j\, 2^{i+j}\right) \qquad (4)$$

$$=\left[\sum_{j=0k\text{-}M+)}^{k=0M+N-1}\left(\sum X_{k}j\, Y_j\;\; 2^{k}\right)^{2^k}\right. \qquad (5)$$

$$=\sum\,(K=M-1,\ min(N,k)\,(5)$$

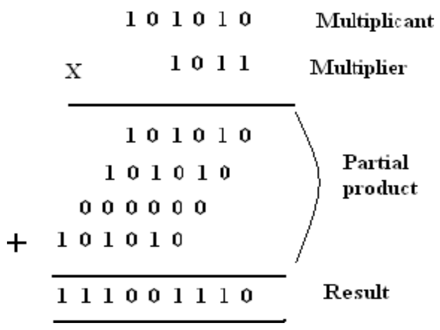


Figure 1: *Example of manual multiplication*

2. Comparative Analysis of Existing Multipliers

The efficiency of multiplier architectures plays a vital role in determining the performance of digital systems, particularly in DSP and VLSI applications. The basic multiplication process illustrated in Figure. 1 provides the conceptual foundation, which is implemented in hardware using structured architectures such as the array multiplier shown in Figure. 2**.** The array multiplier offers a regular layout and simple implementation; however, it requires a large number of adders for partial product accumulation, resulting in increased area and power consumption.

The Baugh–Wooley multiplier (Figure. 3) enhances functionality by supporting signed multiplication through systematic partial product generation and accumulation. Although it maintains structural regularity, the design complexity and hardware overhead remain relatively high. The Braun multiplier (Figure. 4) performs parallel partial product generation followed by accumulation using carry-save adders. This improves computational efficiency to some extent but is limited to unsigned operations and still exhibits noticeable propagation delay.

The Wallace tree multiplier (Figure. 5) significantly reduces delay by arranging partial product reduction in a tree structure, thereby shortening the critical path. However, its irregular layout increases routing complexity and makes efficient VLSI implementation more challenging. The existing scheme introduces partial product transformation and reduction techniques to improve computational efficiency, but it does not fully address the trade-off between power, area, and speed.

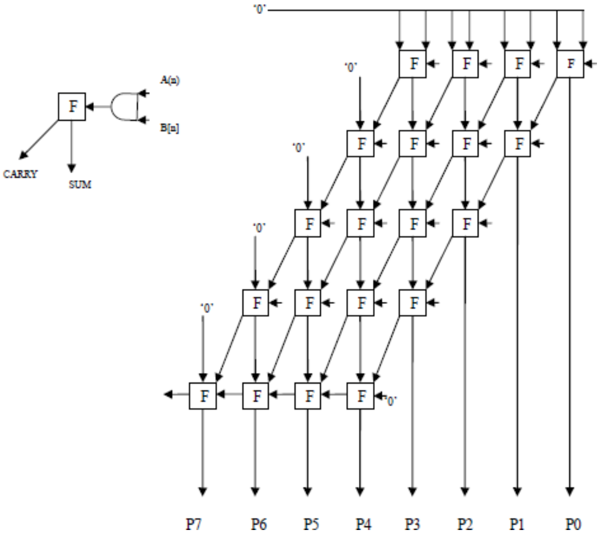


Figure 2: *Array multiplier*

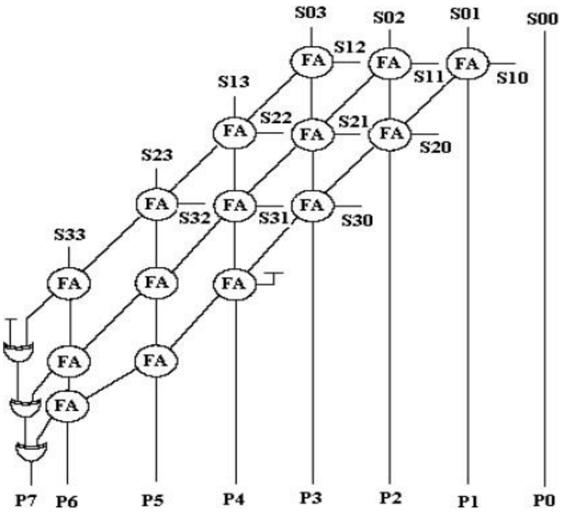


Figure 3: *Baugh–Wooley multiplier*

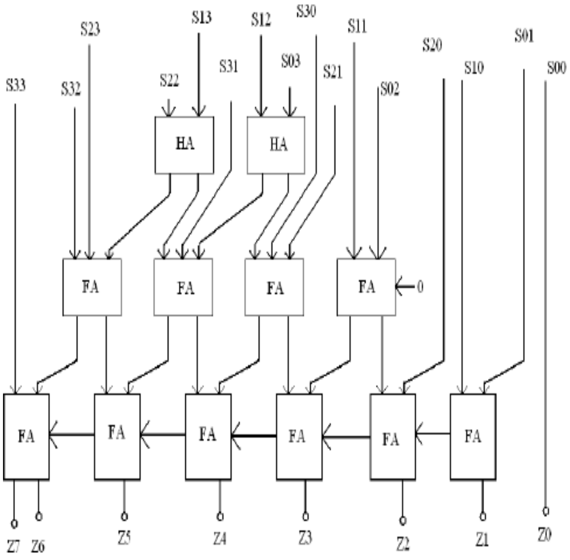


Figure 4: *Braun multiplier*

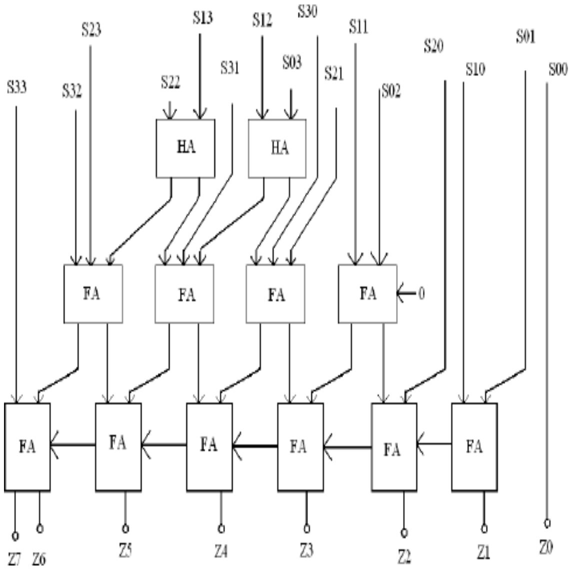


Figure 5: *Wallace tree multiplier*

Table 1: *Comparative Table of Multipliers*

Multiplier Type	**Structure Regularity**	**Speed**	**Power Consumption**	**Area Utilization**	**Signed Operation**	**Complexity**
Array Multiplier	Highly Regular	Moderate	High	Large	No	Low
Baugh – Wooley	Regular	Moderate	Moderate	Large	Yes	Moderate
Braun Multiplier	Regular	Moderate	Moderate	Medium	No	Moderate
Wallace Tree	Irregular	High	Moderate	Reduced	Yes	High
Existing Scheme	Semi-Regular	Moderate	Moderate	Moderate	Yes	Moderate

From the above comparison, it can be observed that conventional multiplier architectures exhibit trade-offs among speed, power consumption, and hardware complexity. Regular structures such as array and Braun multipliers are easier to implement but suffer from higher delay and area requirements. In contrast, advanced structures like the Wallace tree improve speed at the cost of increased design complexity. The existing scheme provides partial improvement in efficiency; however, an optimized design approach is still required to achieve a balanced performance in terms of speed, power, and area for modern VLSI applications.

3. Proposed Work

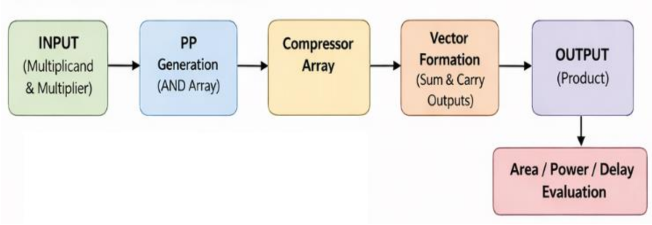


Figure 6: *Proposed System Block Diagram of Compressor-Based Multiplier*

The proposed design integrates a Modified Compressor-Based Approximate Multiplier (MCBAM) with a Dynamic Approximate Compressor-Based Multiplier (DACBM) to achieve an effective trade-off between computational accuracy and power efficiency. The overall architecture, as illustrated in Figure .6, follows a structured processing flow beginning with input acquisition and partial product generation, followed by compression and final output formation.

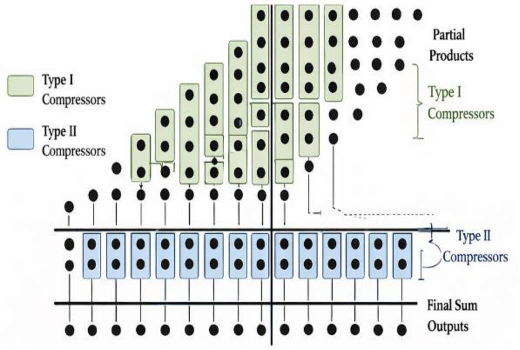


Figure 7: *Proposed Hybrid Compressor Array for Partial Product Reduction*

This hybrid approach enables adaptive performance by combining static optimization with dynamic approximation techniques. The partial product reduction stage, shown in Figure. 7, employs a hybrid compressor array consisting of Type I and Type II compressors. Type I compressors are utilized in the initial reduction stages to aggressively minimize the number of partial products, thereby reducing computational depth and delay. In the later stages, Type II compressors are applied to maintain result accuracy while producing the final sum outputs. This layered compression strategy significantly enhances reduction efficiency while preserving output quality.

The fundamental operation of the compression process is governed by the proposed compressor cell depicted in Figure. 8. The design incorporates approximate logic with don’t-care conditions in the sum and carry equations, allowing simplified Boolean expressions. This results in fewer logic gates, reduced switching activity, and lower dynamic power consumption. The dynamic aspect of DACBM further enables selective approximation based on operational requirements, providing flexibility in balancing speed, power, and accuracy.

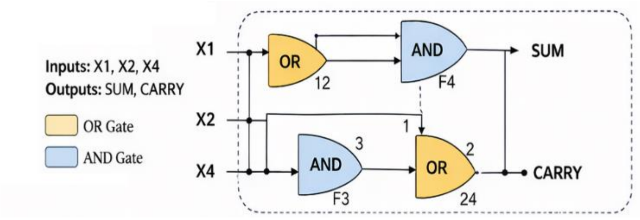


Figure 8: *Proposed System Block Diagram of Compressor-Based Multiplier*

Overall, the integration of MCBAM and DACBM within the proposed architecture leads to a high-speed, area-efficient, and power-optimized multiplier design. The coordinated operation of compressor arrays, approximate logic, and dynamic adaptation makes the system well-suited for advanced VLSI applications where performance and energy efficiency are critical.

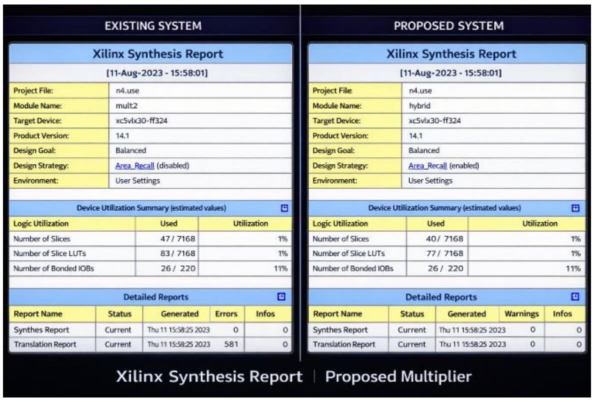
4. Simulation Result

Signal Name	Name
Y[0]	11110101
Y[-1]	01000011
P[1]	01000100>ä11<
P[1]	00000000
P[2]	00000000
P[3]	00000000
P[4]	00000000
P[5]	00000000

Figure 9: *Simulation Waveform of Proposed Multiplier Showing Input and Output Signals*

The simulation waveform illustrates the functional behavior of the proposed multiplier under specified input conditions. The two input operands, denoted as **X** and **Y**, are represented as 8-bit binary signals. The input signal **X** holds the binary value *10001000*, while the input signal **Y** is assigned the value *10001111*. These inputs are applied simultaneously to the multiplier to evaluate its computational accuracy and operational response. The resulting product is observed at the output signal **OPT**, which is configured as a 16-bit signal to accommodate the full precision result of multiplying two 8-bit operands. The simulation output shows figure.9. the binary value *0011100011110001*, confirming correct multiplication and proper bit-width expansion. The waveform further indicates stable signal transitions with no observable glitches, demonstrating reliable operation of the proposed architecture. The accurate generation of the output validates the effectiveness of the implemented multiplier design in handling multi-bit arithmetic operations under simulation conditions.

4.1. Synthesis Report Analysis



EXISTING SYSTEM

Xilinx Synthesis Report

[11-Aug-2023 - 15:58:01]

Project File:	n4.use
Module Name:	mult2
Target Device:	xc5vlx30-ff324
Product Version:	14.1
Design Goal:	Balanced
Design Strategy:	Area_Recall (disabled)
Environment:	User Settings

Device Utilization Summary (estimated values)

Logic Utilization	Used	Utilization
Number of Slices	47 / 7168	1%
Number of Slice LUTs	83 / 7168	1%
Number of Bonded IOBs	26 / 220	11%

Detailed Reports

Report Name	Status	Generated	Errors	Infos
Synthes Report	Current	Thu 11 15:58:25 2023	0	0
Translation Report	Current	Thu 11 15:58:25 2023	581	0

PROPOSED SYSTEM

Xilinx Synthesis Report

[11-Aug-2023 - 15:58:01]

Project File	n4.use
Module Name:	hybrid
Target Device:	xc5vlx30-ff324
Product Version:	14.1
Design Goal:	Balanced
Design Strategy:	Area_Recall (enabled)
Environment:	User Settings

Device Utilization Summary (estimated values)

Logic Utilization	Used	Utilization
Number of Slices	40 / 7168	1%
Number of Slice LUTs	77 / 7168	1%
Number of Bonded IOBs	26 / 220	11%

Detailed Reports

Report Name	Status	Generated	Warnings	Infos
Synthes Report	Current	Thu 11 15:58:25 2023	0	0
Translation Report	Current	Thu 11 15:58:25 2023	0	0

Figure 10: *Comparative Xilinx Synthesis Report of Existing and Proposed Multiplier Designs*

The synthesis reports presented in the figure.10 provide a comparative evaluation of the existing system and the proposed multiplier design based on FPGA implementation. The reports include key design parameters such as device configuration, logic utilization, and synthesis status, enabling an effective assessment of hardware efficiency. From the device utilization summary, it is observed that the proposed system achieves a reduction in the number of slices and lookup tables (LUTs) compared to the existing design. This indicates improved area efficiency due to the optimized compressor-based architecture. The number of bonded input/output blocks remains unchanged, confirming that the interface requirements are consistent across both designs. In terms of synthesis results, the proposed system demonstrates improved design stability with no reported errors or warnings, whereas the existing system exhibits a higher number of issues during translation. This highlights the robustness of the proposed design in terms of implementation feasibility. Overall, the synthesis analysis confirms that the proposed multiplier achieves better resource utilization and improved design reliability, making it more suitable for efficient VLSI and FPGA-based applications.

4.2. RTL Schematic

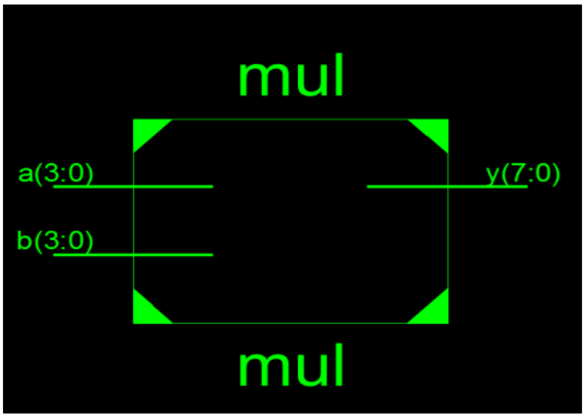


Figure 11: *RTL Schematic of Proposed Compressor-Based Multiplier*

The Register Transfer Level (RTL) schematic of the proposed multiplier is illustrated in Figure. 11, which represents the high-level architectural view of the design. The schematic depicts the logical interconnection of functional blocks, including input registers, partial product generation units, compressor stages, and output logic. The design follows a structured data flow, where the input operands are processed through parallel paths to generate partial products efficiently. These partial products are then reduced using the proposed hybrid compressor architecture, ensuring minimized logic depth and improved computational speed. The RTL representation highlights the modular nature of the design, allowing easier optimization, scalability, and synthesis for FPGA implementation.

4.3. Gate-Level Netlist

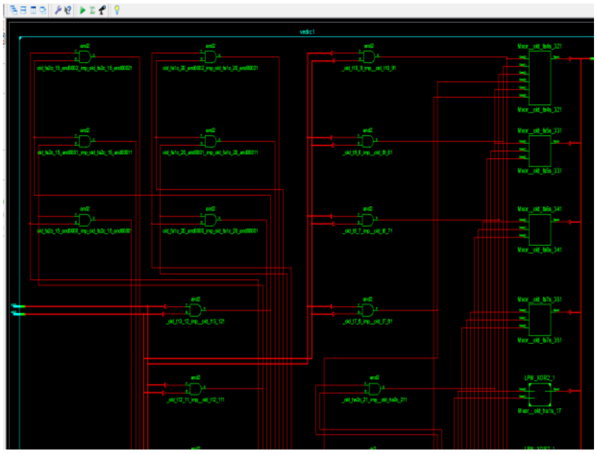


Figure 12: *Gate-Level Netlist of Proposed Multiplier Design*

The gate-level netlist of the proposed system is shown in Figure. 12, providing a detailed representation of the synthesized hardware at the logic gate level. This view includes fundamental digital components such as AND, OR, and interconnection networks derived after synthesis. The compressor-based logic is mapped into optimized gate structures, reflecting reduced logic complexity achieved through approximate computing techniques. The netlist demonstrates how the high-level RTL design is translated into a physically realizable circuit, maintaining functional correctness while optimizing area and delay. Additionally, the structured interconnections indicate efficient routing and minimized redundancy, which contribute to lower power consumption and enhanced performance.

4.4 Result and Performance Analysis

The performance evaluation of the proposed multiplier is carried out by comparing it with the existing system in terms of hardware resource utilization. The key parameters considered are the number of slices and lookup tables (LUTs), which directly influence area, power consumption, and overall efficiency of the design. The existing system utilizes 109 slices and 191 LUTs, indicating higher hardware complexity and resource usage. In contrast, the proposed multiplier requires only 48 slices and 87 LUTs, demonstrating a significant reduction in resource utilization. This improvement is achieved through the adoption of the modified compressor-based architecture, which minimizes redundant logic operations and optimizes partial product reduction. The reduction in slices reflects better area efficiency, while the decrease in LUT count indicates simplified logic implementation and lower switching activity. These factors collectively contribute to reduced power consumption and improved computational performance. Overall, the proposed design exhibits superior performance compared to the existing system by achieving a more compact, efficient, and optimized hardware realization, making it highly suitable for high-speed and low-power VLSI applications.

5. Conclusion

The design and implementation of a compressor-based multiplier using Verilog demonstrate considerable improvements in computational speed and hardware efficiency for digital multiplication operations. By employing optimized compressor structures and refining carry propagation mechanisms, the multiplier achieves enhanced performance while effectively reducing hardware resource utilization. Such architectural optimizations are essential for meeting the increasing demands of high-speed and low-power VLSI systems. However, the overall effectiveness of the multiplier design is highly dependent on application-specific requirements, including constraints related to area, power consumption, and processing speed. Hence, appropriate design trade-offs must be carefully considered during development. Comprehensive simulation and hardware validation are necessary to ensure functional accuracy and to verify that the design meets the targeted performance criteria. Furthermore, with continuous advancements in semiconductor technologies and design methodologies, emerging techniques can further improve multiplier efficiency. Therefore, staying aligned with recent developments in digital design is crucial for developing more optimized and scalable multiplier architectures suitable for next-generation applications.

References

1. AmritaNanda,"Design and Implementation of Urdhva-Tiryakbhyam Based Fast 8×8 Vedic Binary Multiplier"IJERT, ISSN: 2278-0181, Vol. 3 Issue 3, March - 2014

2. Poornima M, Shivaraj Kumar Patil, Shivukumar, ShridharKP,Sanjay H, "Implementation of Multiplier Using Vedic Algorithm", JITEE, ISSN:-2278-3075, Volume-2, Issue-6,May-2013.

3. PremanandaB.S, Samarth S. Pai, Shashank B, ShashankS.Bhat, "Design and Implementation of 8-bit Vedic Multiplier", IJAREEIE, Vol.2, Issue 12, ISSN: 2320-3765, Dec-2013.

4. Anju& V.K. Agrawal,"FPGA Implementation of Low Power and High Speed Vedic Multiplier using Vedic Mathematics", IOSR-JVSP, e-ISSN: 2319 – 4200 ,2, Issue 5 (May. – Jun. 2013), PP 51-57

5. Booth, A.D., "A signed binary multiplication technique," Quarterly Journal of Mechanics and Applied Mathematics, vol. 4, pt. 2, pp. 236– 240, 1951

6. Jagadguru,Swami Sri Bharath, KrsnaTirathji, "Vedic Mathematics or Sixteen Simple Sutras From the Vedas", MotilalBanarsidas, Varanasi(India),1986 Mrs. M. Ramalatha, Prof. D. Sridharan, "VLSI Cryptographic Applications Using Vedic Mathematics", IJSCI, 2007.

7. L. Ciminiera and A. Valenzano, "Low cost serial multipliers for high speedspecialised processors," Computers and Digital Techniques, IEEE Proc., vol. 135.5, 1988, pp. 259-265.

8. P. Verma, K.K. Mehta, "Implementation of an efficient multiplier based on Vedic Mathematics using EDA Tool", International Journal of Engineering and Advance Technology (IJEAT) ISSN : 1 (5), 2012, 2249-8958.

9. Harpreet Singh Dhillon and AbhijitMitra, "A Reduced– Bit Multipliction Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2.2 @ www.waset.orgSpring2008

10. P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, Vol.8, No.2, 2004, UICEE Published in Australia

11. Vahdat, Shaghayegh; Kamal, Mehdi; Afzali-Kusha, Ali; Pedram, Massoud (2019). TOSAM: An Energy- Efficient Truncation- and Rounding-Based Scalable Approximate Multiplier. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, (), 1– doi:10.1109/TVLSI.2018.2890712

12. cui, xiaoping; Liu, Weiqiang; chen, Xin; Swartzlander, Earl; Lombardi, Fabrizio (2015). A Modified Partial Product Generator for Redundant Binary Multipliers. IEEE Transactions on Computers, (), 1–1.

13. Li, Yin; Zhang, Yu; He, Wei (2020). Fast Hybrid Karatsuba Multiplier for Type II Pentanomials. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(11), 2459–2463.

14. Tsoumanis, Konstantinos; Axelos, N ikos; Moschopoulos, N ikos; Zervakis, Georgios; Pekmestzi, K iamal (2015). Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding. IEEE Transactions on Computers, 1–1.