

# High-Speed and Power-Efficient MAC Unit Design Using Sequential Finite-Field Multiplier for Advanced DSP Applications

A. Venu Gopal<sup>1</sup>, P. Venkatesh<sup>2</sup>, T. Ravinder<sup>3</sup>

<sup>1</sup>Department of Electronics and Communication Engineering

<sup>1</sup>Sree Dattha Institute of Engineering and Science, Sheriguda, Hyderabad, Telangana

## Abstract

Increased processing power is constantly needed by digital signal processors (DSPs), especially when it comes to combining CPU cores into a single integrated circuit. Convolution, transformation, correlation, filtering, and other processes that mostly rely on multiplication and repetitive addition are all made possible by DSPs. In DSPs, the Multiply and Accumulate unit (MAC) is crucial, and high-performance procedures are required within this unit. DSP algorithms rely significantly on the MAC's speed performance. Bit-parallel computing techniques have historically been used to build MAC architecture, which leads to quadratic increases in hardware needs as bit accuracy increases. In contrast, bit-serial computing makes hardware size proportionate to bit accuracy by serializing a single input. Static power dissipation is another urgent issue as technology advances. This study presents a high-speed MAC unit designed for arithmetic applications that is based on the Sequential Finite-Field Multiplier (SFFM) approach. The MAC unit's adder blocks are made utilizing a high-speed pipelined adder architecture.

**Keywords:** Multiply and Accumulate unit, Digital signal processors, Sequential Finite-Field Multiplier, High-Speed Processing, Pipelined Adder.

## 1. Introduction

Designing a multiplier and accumulator using sequential finite field operations is a common task in VLSI design for many applications, including cryptography, error correction codes, and digital signal processing. The design involves implementing sequential circuits that perform arithmetic operations in a finite field. The basic approach involves representing the operands and the result as polynomials with coefficients in a finite field, such as Galois field (GF). The arithmetic operations are then performed on these polynomials using various algorithms, including Karatsuba, Toom-Cook, and Montgomery multiplication algorithms. The multiplier and accumulator design typically involves several stages, including input and output registers, finite field arithmetic logic unit (ALU), and pipeline registers for sequential processing. The pipeline registers help to minimize the critical path delay and increase the throughput of the design. The design process starts with a high-level architecture, followed by algorithm selection and optimization. The next step involves designing the finite field ALU and implementing it using digital logic gates. The circuit is then simulated using a hardware description language (HDL) and synthesized to generate a netlist. The netlist is then optimized for area, power, and timing using tools such as place and route (P&R) and static timing analysis (STA). Finally, the design is verified using functional and timing simulations. In summary, designing a multiplier and accumulator using sequential finite field operations is a complex task that involves a deep understanding of arithmetic algorithms, digital logic design, and VLSI design methodologies. The design must be optimized for area, power, and timing to meet the requirements of the target application. Applications require high-speed and efficient computation of arithmetic operations in finite fields, such as cryptography, error correction codes, and digital signal processing. Finite field operations are essential for secure data encryption and decryption, error correction in data transmission, and processing of signals in communication systems. The implementation of sequential finite field operations using VLSI technology can provide significant improvements in performance

and power efficiency compared to software-based implementations. This is because VLSI circuits can exploit the parallelism and pipelining inherent in the arithmetic operations to achieve high throughput and low latency. VLSI circuits can be customized to meet the specific requirements of the application, such as area, power, and performance. The motivation for designing a multiplier and accumulator using sequential finite field operations in VLSI stems from the need for high-speed and efficient computation of arithmetic operations in finite fields, combined with the advantages of VLSI technology in achieving high performance and low power consumption.

## 2. Literature Survey

Sahu, Ajay Kumar (2021)[1], this work investigated existing work and techniques used by several authors to minimize the power consumption in the design of MAC Unit. This review can provide an insight to the beginners in the VLSI Arithmetic Circuit Design to gain more idea on Low power MAC Unit Design. Rishi Kiran, E. Swathi Vangala, and J. V. R. Ravindra, In this work, PERAM deals with reversible array multiplier. As it consists of rudimentary reversible gates like CCNOT and CNOT, analysis will be uncomplicated. The proposed design methodology is implemented and verified in Cadence® virtuoso of 45 nm technology showing the improvement of 77.76% in terms of power, 71.39% in terms of power delay product. PERAM shows a great variation in the power and power delay product. Gunasekaran, K., et al (2022)[2], this work proposed a reversible logic design of a 4-bit MAC structure using Peres gates as reversible logic blocks. In addition, a variety of parameters, along with those of conventional computing, perform a relative test between classical style and quantum logic operation. Priyadarshini, K. Mariya, et al (2021) [3], In this work investigation, Pre-Accumulator and Post-Multipliers (PAPM) are proposed which accelerate the speed of processor. 4-bit multiplier using Carry Save Adder (CSA) is built with 6 Transistors-Adder and sutras of Vedic mathematics is constructed. Accumulator of multiplier and accumulator are designed with Two Level Edge Triggering Flip-Flops (TLET-FF) to increase bandwidth of the memory. The proposed architecture of Multiply Accumulate (MAC) circuit consumes very less power when compared to existing high speed MACs. Inayat, Kashif, and Jaeyong Chung (2020) [4], this demonstrated the hybrid accumulator with partial CPA factoring in “Gemmini,” an open-source practical systolic array accelerator and factoring technique does not change the functionality of the base design. Puttam, HS Krishnaprasad, P. Sivadurga Rao, and N. V. G. Prasad. (2012)[6], In this work, we proposed a new architecture of multiplier -and- accumulator (MAC) for high-speed arithmetic and low power. Multiplication occurs frequently in finite impulse response filters, fast Fourier transforms, discrete cosine transforms, convolution, and other important DSP and multimedia kernels. Sathish, M. V., and Mrs Sailaja (2011)[7], in this work a new architecture of multiplier-and accumulator (MAC) for high-speed arithmetic. By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved. Since the accumulator that has the largest delay in MAC was merged into CSA, the overall performance was elevated. The proposing method CSA tree uses 1’s-complement-based radix-2 modified Booth’s algorithm (MBA) and has the modified array for the sign extension in order to increase the bit density of the operands. Teja, Ravi, et al [224-230][8], In this work, we proposed a new architecture of multiplier-and-accumulator (MAC) for high-speed arithmetic. By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved. Since the accumulator that has the largest delay in MAC was merged into CSA, the overall performance was elevated. The CSA propagates the carries to the least significant bits of the partial products and generates the least significant bits in advance to decrease the number of the input bits of the final adder. Khubnani, Rashi, Tarunika Sharma, and Chitirala Subramanyam et al [2022][9], in this work Vedic Multiplier is a key tool in rapidly growing technology especially in the immense domain of Image processing, Digital Signal Processing, real-time signal. Multipliers are important block in

digital systems and play a critical role in digital designs. Along with accuracy demand for minimizing time area, power, and delay of the processor by enhancing speed is the focus point. Vedic mathematics rules and Algorithms generate partial products concurrently and save time. This paper is a review of the application and modification of Vedic multiplier in different fields and a comparison of Vedic multiplier with other multipliers for enhancing performance parameters. Ponugoti, Vamshi, et al(2022)[10] , in this paper presents a four-bit Baugh-Wooley multiplier using full swing gate diffusion input (GDI) technology. In general, addition is a crucial arithmetic operation and is heavily demanded in VLSI design. These are widely used in digital signal processing, accumulators, microprocessors and many other applications. So, the full adder performance decides the overall system performance. N. Pandu Ranga, and K. Maheswari Devi. et.al(2022)[11] , in this work MAC unit performed important operation in many of the digital signal processing (DSP) applications. In existing method the multiplier is designed using modified Wallace multiplier and the adder is done with carry save adder. Nithiya, C., et al(2022)[12]. in this test bench work the proposed approach uses the RADIX-4 and RADIX-8 models with a glitch optimization circuit to construct a low power adjustable path selective Booth Multiplier architecture. The system utilized low power techniques like clock gating and sequenced latching process. The design is simulated using Modelsim tool and validated with test and evaluation of multiplier circuit with equivalent test bench. Zhang, Jiayi, et al(2022)[13] in this work we developed Easy MAC, a flexible Chisel-based MAC generator with a canonical architectural representation. We design a compact and canonical sequence representation to express the architecture of MACs. And the MAC generator takes the compact representation as input to gain the Verilog codes. We also give a case study on developing a heuristic design space exploration (DSE) method based on this representation. The experimental result shows the effectiveness of the representation in DSE. Nithiya, C., et al(2022)[14] in this work we discussed the ideal path for selecting an arithmetic unit for IoT applications. Based on the analysis of eight types of 16-bit adders, Carry Look-ahead (CLA) adder was found to produce low power. Additionally, Multiplier and Accumulator (MAC) unit is implemented with Booth multiplier by using the low power adders in the order of preference. The design is synthesized and verified using Synopsys Design Compiler and VCS. Rajalakshmi, G., et al(2022)[15] in this work a 16-bit MAC unit based on a 4-bit Ripple Carry Adder and a 4x4-Vedic Multiplier in this paper. HSPICE Synopsys Tool is used for simulation to mimic the MAC unit. COSMOSSCOPE is used to determine the latency of all circuits in the MAC unit. For the delay and power, the MAC implementations of proposed and current architectures are compared. The suggested MAC architecture, which employs a Vedic multiplier and a ripple carry adder, considerably lowers circuit delay and excessive power consumption, resulting in increased speed and performance.

### 3. Proposed Methodology

MAC is composed of an adder, multiplier, and an accumulator. Usually adders implemented are Carry- Select or Carry-Save adders, as speed is of utmost importance in DSP. One implementation of the multiplier could be as a parallel array multiplier. The inputs for the MAC are to be fetched from memory location and fed to the multiplier block of the MAC, which will perform multiplication and give the result to adder which will accumulate the result and then will store the result into a memory location. This entire process is to be achieved in a single clock cycle. The architecture of the MAC unit which had been designed in this work consists of one 16 bit register, one 16-bit Modified Booth Multiplier, 32-bit accumulator. To multiply the values of A and B, Modified Booth multiplier is used instead of conventional multiplier because Modified Booth multiplier can increase the MAC unit design speed and reduce multiplication complexity. SPST Adder is used for the addition of partial products and a register is used for accumulation. The operation of the designed MAC unit is as in Figure 1. The product of  $A_i \times B_i$  is always fed back into the 32-bit accumulator and then added again

with the next product  $A_i \times B_i$ . This MAC unit is capable of multiplying and adding with previous product consecutively up to as many as times.

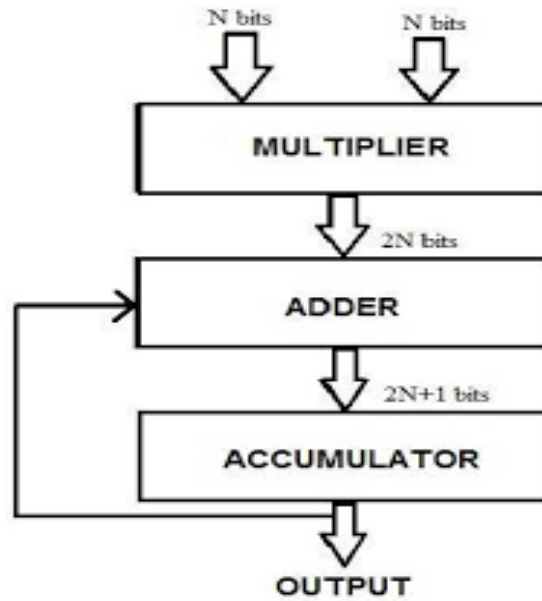


Fig.1. Multiplier and Accumulator Architecture

### 3.2 Sequential Finite Field Adder

Adder-subtractor tree is a useful sub-circuit that often finds applications in the parallel implementation of distributed arithmetic-based FIR Filters, matrix multiplication circuits or pipelined binary tree multipliers. Each ADD/SUB block shown in Figure 2 can be independently configured as a two's complement adder or subtractor, as they have independent mode select lines. Each of the blocks can perform a 24-bit two's complement addition/subtraction, with a total of five stages of ADD/SUB blocks. The circuit has also been redesigned by inserting scan FFs at the site of the pipeline registers with no hardware overhead and performance deterioration. The slice configuration of a pipelined adder/subtractor tree without the scan FFs have been shown in Figure 3.

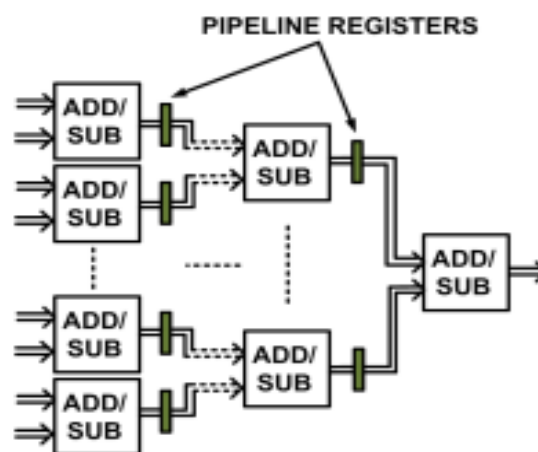


Figure 2. Block diagram of pipelined adder subtractor tree

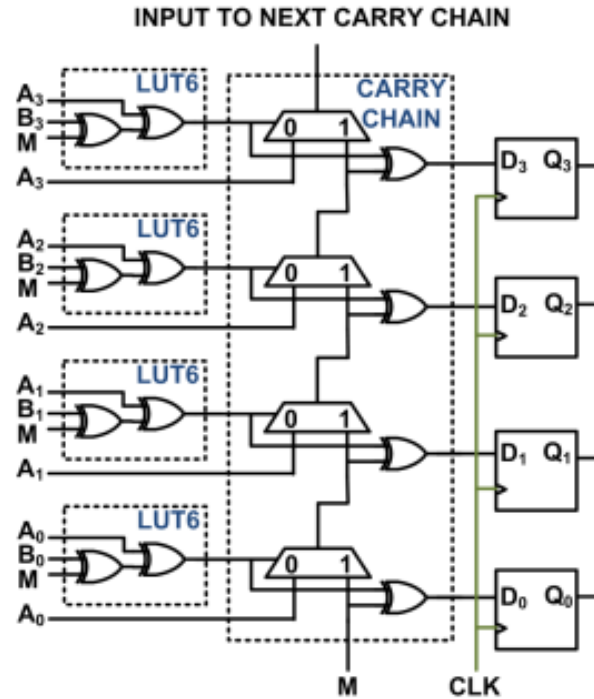


Figure 3. Original pipelined adder subtractor design

The sum bit can be computed by EX-ORing the LUT and carry chain multiplexer output as  $S_i = A_i \oplus (B_i \oplus M) \oplus C_i$ . The carry output of each multiplexer stage can be computed as:

$$\begin{aligned}
 C_{i+1} &= A_i(B_i \oplus M) + (A_i \oplus (B_i \oplus M))C_i \\
 &= A_i(A_i(B_i \oplus M) + \overline{A_i}(B_i \oplus M)) \\
 &\quad + (A_i \oplus B_i \oplus M)C_i \\
 &= A_i(A_i \odot (B_i \oplus M)) + (A_i \oplus B_i \oplus M)C_i
 \end{aligned}$$

**Sequential Finite Field Multiplier:** Sequential Finite Field multiplier is the simplest structure of parallel multiplier. This multiplier using the standard adds and shift operation based on 'add and shift' algorithms to perform a multiplication operation. The structure of 4-bit array multiplier is presented in fig. 6. The partial products generator consists of n number of 'AND' gates to multiply the multiplicand with each bit of the multiplier and then these partial products are shifted depending on their order and this summation operation can be performed by using full adder and a half adder. In 4x4 Sequential Finite Field Multiplier, 4x4 AND gates used to generate partial products and 4x (4-2) full adders, and 4 half adders used to generate.

**4.Results and Discussion**

The simulation results will done by using in Vivado ISE. The timing, power and synthesis reports listed below.

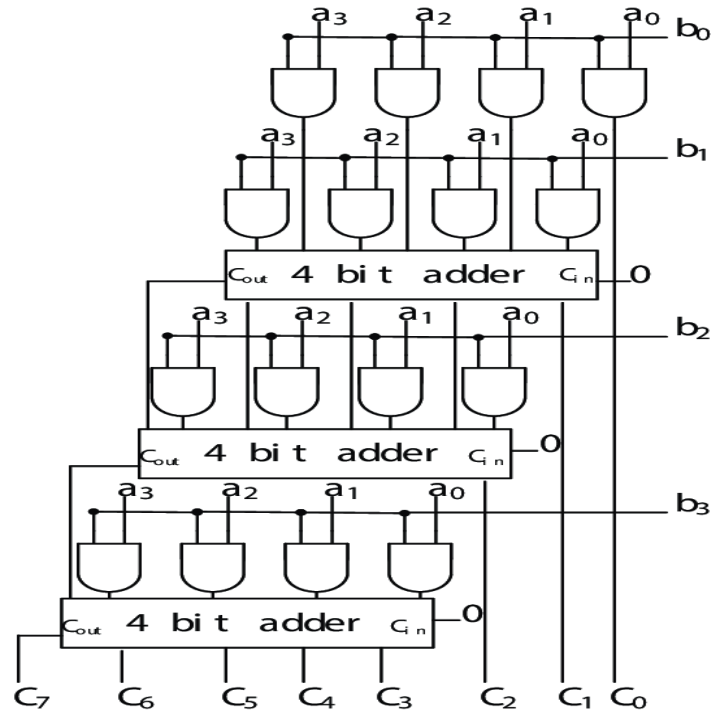


Figure 4. 4x4 Sequential Finite Field Multiplier

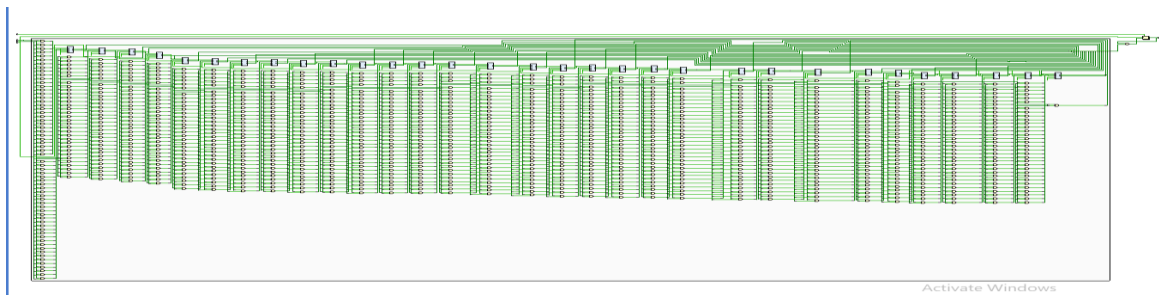


Figure 5. RTL schematic

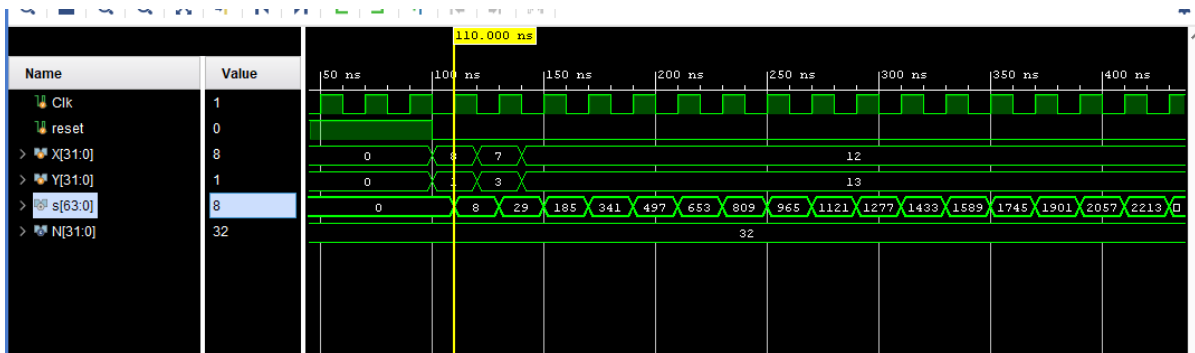


Figure 6. Simulation Output

Utilization			
Post-Synthesis   Post-Implementation			
Graph   Table			
Resource	Utilization	Available	Utilization %
LUT	3665	41000	8.94
FF	64	82000	0.08
IO	130	300	43.33
BUFG	1	32	3.13

Figure 7. Area Summary.

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	3	1	2	s_reg[2]/C	s_reg[2]/D	0.298	0.179	0.119
Path 2	∞	3	1	2	s_reg[11]/C	s_reg[11]/D	0.316	0.177	0.139
Path 3	∞	3	1	2	s_reg[15]/C	s_reg[15]/D	0.316	0.177	0.139
Path 4	∞	3	1	2	s_reg[19]/C	s_reg[19]/D	0.316	0.177	0.139
Path 5	∞	3	1	2	s_reg[23]/C	s_reg[23]/D	0.316	0.177	0.139
Path 6	∞	3	1	2	s_reg[27]/C	s_reg[27]/D	0.316	0.177	0.139
Path 7	∞	3	1	2	s_reg[31]/C	s_reg[31]/D	0.316	0.177	0.139
Path 8	∞	3	1	2	s_reg[35]/C	s_reg[35]/D	0.316	0.177	0.139
Path 9	∞	3	1	2	s_reg[39]/C	s_reg[39]/D	0.316	0.177	0.139
Path 10	∞	3	1	2	s_reg[3]/C	s_reg[3]/D	0.316	0.177	0.139

Figure 8. Delay Summary.

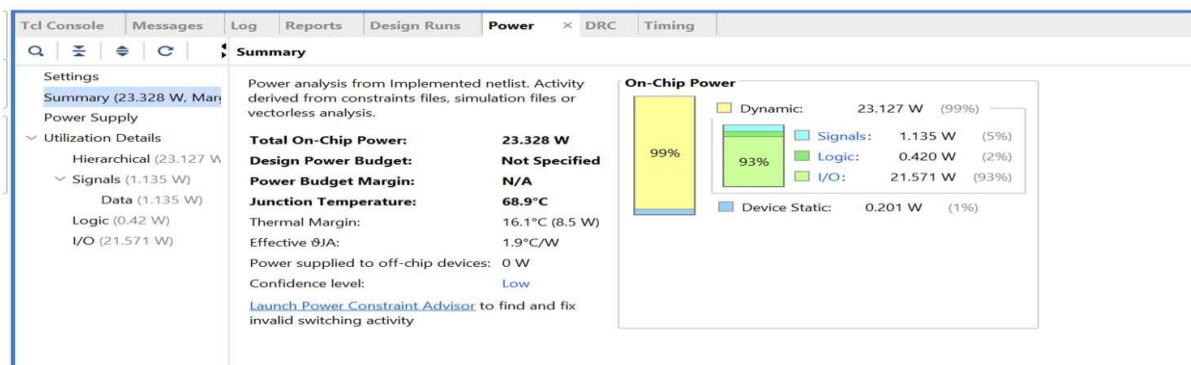


Figure 9. Power Summary.

Table 1. Comparison Table

Parameter	Existing method	Proposed method
LUTs	4195	3665
Time Delay	0.450ns	0.361ns
Power Consumption	298mw	267mw

### 5. Conclusion

Pipelined large word length digital multipliers are difficult to design under the constraints of core cycle time (for nominal voltage), pipeline depth, power and energy consumption and area. Low level optimizations might be required to meet these constraints. In this work, we have presented a method to reduce by one the maximum height of the partial product array for 64-bit MAC has been developed. This reduction may allow more flexibility in the design of the reduction tree of the pipelined multiplier.

### References

[1] Sahu, Ajay Kumar, et al. "VLSI design techniques for low power MAC unit: A review." AIP A. Venu Gopal et al 257-264

- Conference Proceedings. Vol. 2358. No. 1. AIP Publishing LLC, 2021.
- [2] Rishi Kiran, E., Swathi Vangala, and J. V. R. Ravindra. "Peram: ultra power efficient array multiplier using reversible logic for high-performance mac." *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020*. Springer Singapore, 2021.
  - [3] Gunasekaran, K., et al. "Design Of 4-Bit Multiplier Accumulator Unit By Using Reversible Logic Gates In Peres Logic." *European Journal Of Molecular & Clinical Medicine* 7.09 (2022): 2020.
  - [4] Penchalaiah, Usthulamuri, and VG Siva Kumar. "Design and Implementation of Low Power and Area Efficient Architecture for High Performance ALU." *Parallel Processing Letters* 32.01n02 (2022): 2150017.
  - [5] S. V. G. Kumar, M. Vadivel, U. Penchalaiah, P. Ganesan and T. Somassoundaram, "Real Time Embedded System for Automobile Automation," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-6, doi: 10.1109/ICSCAN.2019.8878820.
  - [6] Puttam, HS Krishnaprasad, P. Sivadurga Rao, and N. V. G. Prasad. "Implementation of low power and high speed multiplier-accumulator using SPST adder and verilog." *International Journal of Modern Engineering Research (IJMER)* 2.5 (2012): 3390-3397.
  - [7] Sathish, M. V., and Mrs Sailaja. "VLSI architecture of parallel multiplier-accumulator based on radix-2 modified booth algorithm." *International Journal of Electrical and Electronics Engineering (IJEEE)* 1 (2011).
  - [8] Teja, Ravi, et al. "Implementation of New VLSI Architecture of Multiplier and Accumulator using Carry Save Adder." *International Journal of Applied Research & Studies (iJARS)* 1.1 (2012): 224-230.
  - [9] Khubnani, Rashi, Tarunika Sharma, and Chitirala Subramanyam. "Applications of Vedic multiplier-A Review." *Journal of Physics: Conference Series*. Vol. 2225. No. 1. IOP Publishing, 2022.
  - [10] Ponugoti, Vamshi, et al. "Design of Baugh-Wooley Multiplier Using Full Swing GDI Technique." *Soft Computing and Signal Processing: Proceedings of 4th ICSCSP 2021*. Singapore: Springer Nature Singapore, 2022. 769-779.
  - [11] N. Pandu Ranga, and K. Maheswari Devi. "Design of High Performance 64 bit MAC Unit."
  - [12] Nithiya, C., et al. "Design of Low Power Adaptive Path Changing Glitch Free Radix-4, Radix-8 Multipliers." 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, 2022.
  - [13] Zhang, Jiayi, et al. "EasyMAC: design exploration-enabled multiplier-accumulator generator using a canonical architectural representation." 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2022.
  - [14] Nithiya, C., et al. "Performance Analysis of Arithmetic Unit for IoT Applications." 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS). IEEE, 2022
  - [15] Rajalakshmi, G., et al. "Performance Analysis of Energy Efficient MAC Unit for Digital Applications." 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES). IEEE, 2022.