

# Design and Implementation of VLSI Architecture for Power and Area Efficient MAC Using Modified Booth Algorithm and Hybrid Adder

Thakshak M P<sup>1</sup>, Vijaya Prakash A M<sup>2</sup>

Department of ECE Bangalore Institute of Technology Bangalore, India

## Abstract

This paper presents an area and power-efficient Multiply-Accumulate (MAC) unit architecture that integrates a modified Booth multiplier with a 16-bit hybrid adder and pipeline methodology, along with a Conditional gating technique. The modified Booth multiplier minimizes the number of partial products, optimizing power and area efficiency. The hybrid adder, combining a 4-bit carry select adder and a 28-bit carry look-ahead adder, efficiently accumulates partial products and performs addition operations. The pipeline methodology is used to further boost performance by dividing the multiplication and accumulation process into stages, enabling parallel execution and higher throughput. The Conditional gating technique significantly reduces power consumption by dynamically disabling inactive pipeline stages and functional units. The MAC unit is designed to handle 16-bit input operands and produces a 32-bit output product. This architecture achieves an optimal balance between area, power consumption, and performance, making it ideal for resource-constrained digital signal processing and computing applications.

**Index terms:** Power-efficient MAC unit, Modified Booth multiplier, Hybrid adder, Pipeline methodology, Conditional gating technique

## INTRODUCTION

In the realm of digital signal process systems and general-purpose processors, fast multipliers are integral components. The emergence of media processing has underscored the importance of speed in multiplication operations. Historically, multiplication was performed using a series of addition, subtraction, and shift operations, treating it as repeated additions. However, this method is slow and has largely been superseded by algorithms leveraging positional representation.

Modern multipliers decompose the task into two main parts: generating partial products & accumulating these products. This involves evaluating partial products & accumulating the shifted partial products through successive column-wise addition in shifted PP matrix. To handle signed & unsigned numbers both efficiently, two's complement representation is widely used.

Within the framework of VLSI design, power dissipation is a critical concern. Adhering to Moore's law and producing consumer electronics with longer battery life and less weight necessitates low-power designs. Thus, fast multiplier is crucial not only for performance but also for power efficiency. The procedure for multiplication is now optimized using positional representation and partial product decomposition, involves aligning multiplicand bits in the shifted partial product matrix and adding them to form the product bit.

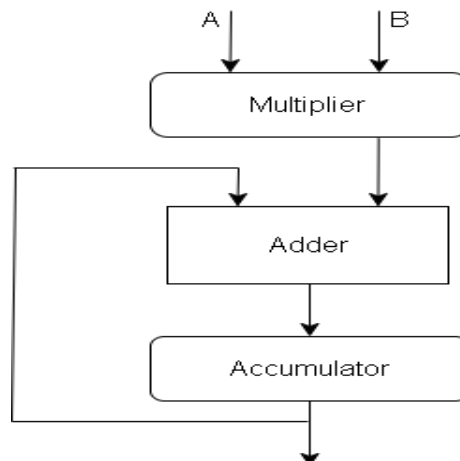
For applications in dsp and multimedia, where accuracy and speed are paramount, fixed-width multipliers are often employed. These multipliers accept two n-bit inputs but only produce the n most significant bits product, significantly reducing hardware complexity and power consumption. However, this approach may lead to truncation errors. To mitigate these errors, compensation biases are generated and added to the retained adder cells, while minimizing the count of adder cells used.

Several methods are proposed to enhance fixed-width multipliers by reducing hardware requirements and improving accuracy. These range from adding a constant bias to adaptive schemes that adjust the

compensation value based on input data. Techniques involving regression analysis and Statistics has also been employed, though they sometimes suffer from truncation errors.

A more advanced technique generates compensation bias adaptively based on input data, significantly reducing truncation errors but increasing circuit complexity. This method achieves better performance by lowering the maximum absolute error and mean square error, though it may still have larger mean errors. Other approaches improve accuracy with simpler self-compensation techniques, reducing gate numbers and average error.

Modified Booth encoding (radix-4) is commonly used in multipliers to halve the partial products number. This introduction proposes an efficient Modified Booth multiplier which decreases the partial products number, thereby minimizing power and area consumed while maintaining high speed and accuracy.



**Fig.1 Basic MAC unit**

The radix-4 encoder outputs are utilized to generate bit-paired multipliers of 4th order, effectively decreasing the partial products number and enhancing overall performance.

The paper organization is as follows. In Section II, the fundamentals of MAC is presented.

Section III Proposed Modified Booths multiplier is presented. Section IV covers the Hybrid adder.

Section V presents the results and performance comparison of the architecture with existing techniques. Section VI concludes the paper.

## FUNDAMENTALS OF MAC UNIT

The multiplication process is divided into two main stages: generating partial products and then collecting and summing them. The multiplier-accumulator unit consists of three primary blocks: the Booth encoder, the partial product summation and accumulation unit, and the final adder. Partial products can be generated using various multiplication algorithms, including bit-serial, serial-parallel, or full parallel techniques. The Booth or modified Booth algorithms are normally used for this purpose. In an  $n$ -bit multiplier, if  $n$  equals the summands number, the Baugh-Wooley algorithm is employed. If the no. of summands is less or equal  $n$ , the Booth algorithms can be used, and similarly for algorithm of modified Booth.

Partial product additions is performed using a carry-ripple adder in serial-parallel multipliers and carry-save adder in parallel multipliers. Once the partial products are reduced to a sum and carry, the final adder generates the multiplication result. The result has a bit-width equal to sum of the bit-widths of the multiplier and the multiplicand. Consequently, the data path width is doubled, leading to significant delay. The final adder produces a double-precision result that must be added with the contents of accumulator, which is also double-precision.

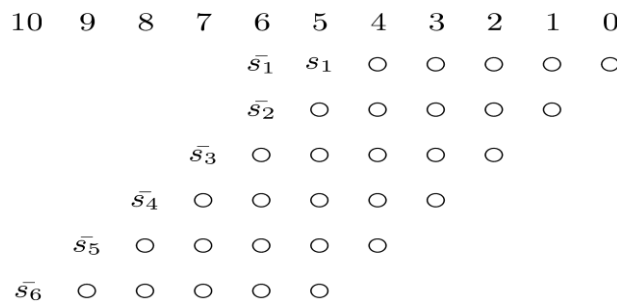
The Basic operation of MAC unit be

- a) Partial Product Generation
- b) Partial Product Reduction

- c) Partial Product Additions
- d) Partial Product Generation

Ppg is the initial stage in the multiplication process, where multiple intermediate products, or partial products, are created based on multiplier and multiplicand. These pp's are subsequently summed to get the final product. The speed and efficiency of ppg's significantly impact the effective performance of the overall multiplier.

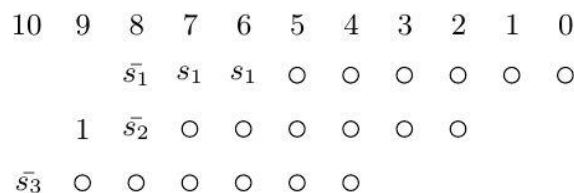
In bit-serial multiplication, bits are processed one at a time. This method requires fewer hardware resources but is slower due to its sequential processing of bits. Or In full parallel multiplication, all multiplier bits and multiplicand are processed simultaneously. This method is the fastest but requires the most hardware resources. Or Serial-Parallel approach strikes a balance between bit-serial and full parallel techniques. Some bits are processed in parallel, which speeds up the computation compared to bit-serial methods while still using fewer resources than full parallel techniques.



**Fig.2 PPG**

**b) Partial Product Reductions**

Partial product reduction is a critical stage in the multiplication process that involves summing the multiple partial products generated in the first stage. Efficient reduction of these partial products is essential for achieving high-speed multiplication with minimal hardware complexity and power consumption.



**Fig.3 PPR**

**c) Partial Product Additions**

PPA is the stage in the multiplication process where the generated PP's are combined to produce the end product. This step is critical in determining the speed and efficiency of the multiplication operation. Various techniques and adder architectures are utilized to sum the partial product effectively, balancing the trade-offs between speed, area, & power consumption.

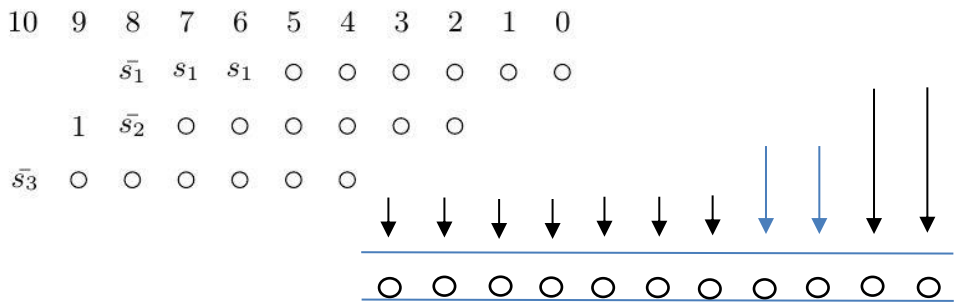


Fig.4 PPA

**PROPOSED MAC UNIT**

The Proposed design has the Hybrid multiplier , Hybrid adder will be designed in sequential –parallel way.

**Modified booth multiplier**

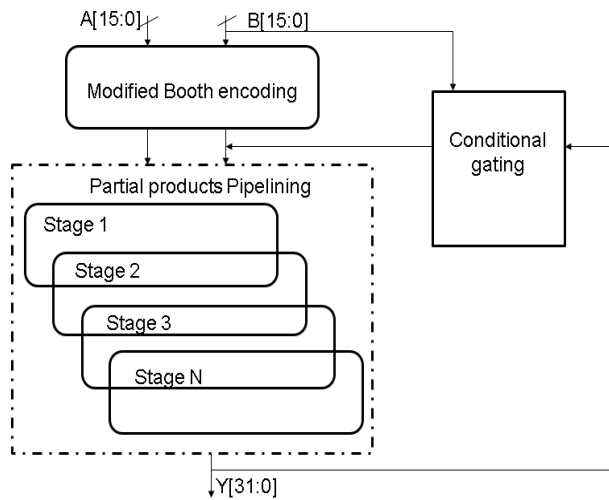


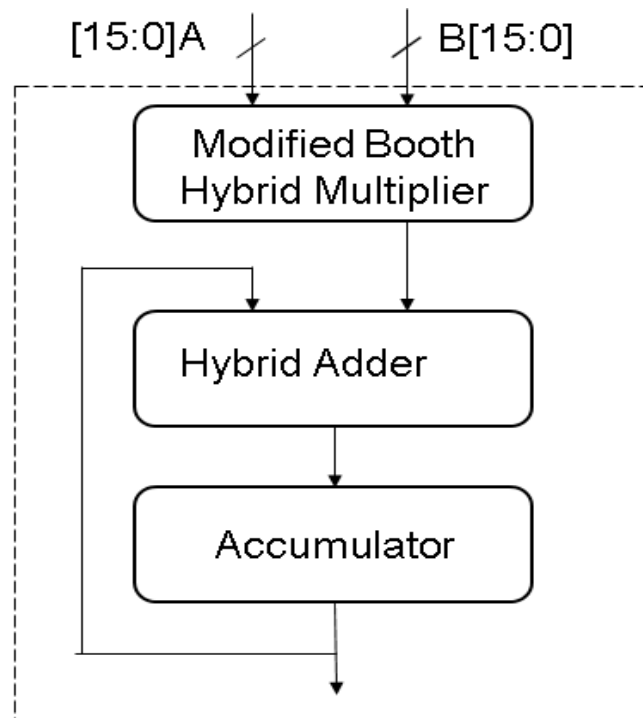
Fig.5 Modified Booth Multiplier

The image provides a high-level overview of a pipelined multiplier design utilizing Modified Booth encoding. This design is a sophisticated approach to multiplication, aiming to enhance speed and efficiency. At the core of this system are several key components: the inputs, Modified Booth encoding, partial products pipelining, conditional gating, and the final output.

The inputs to this system are the 16-bit multiplicand (A[15:0]) and the 16-bit multiplier (B[15:0]). These inputs are processed by the Modified Booth encoding block, which is designed to optimize the partial products number needed for multiplication. Booth encoding achieves this by grouping bits of the multiplier and encoding them in a way that reduces the number of total operations required.

Following the encoding process, the partial products pipelining block are fed with encoded outputs. This block consists of multiple stages, each responsible for generating and summing partial products. The pipelining structure ensures that each stage operates concurrently, allowing for partial products to be generated and accumulated in parallel. This parallel processing significantly enhances the overall speed of the multiplication process.

Conditional gating is another crucial component of this design. This block controls the flow of data based on specific conditions, effectively managing power consumption. By enabling or disabling parts of the circuit as needed, conditional gating ensures that power is only used where necessary, optimizing the system efficiently.



**Fig.6 Coventional MAC**

The final component is the output, represented as  $Y[31:0]$ , which is the 32-bit product result of the multiplication. After passing through all the pipeline stages, the cumulative sum of the PP's is the output of final result.

The process of this design can be explained as

1. **PPG- Bit Paring by booth Encoding**
2. **PPR- booth multiplication with pipeline**
3. **Processing**
4. **PPA- CSA+CLA(4+28)**

### 1. PPG- Bit Paring by booth Encoding

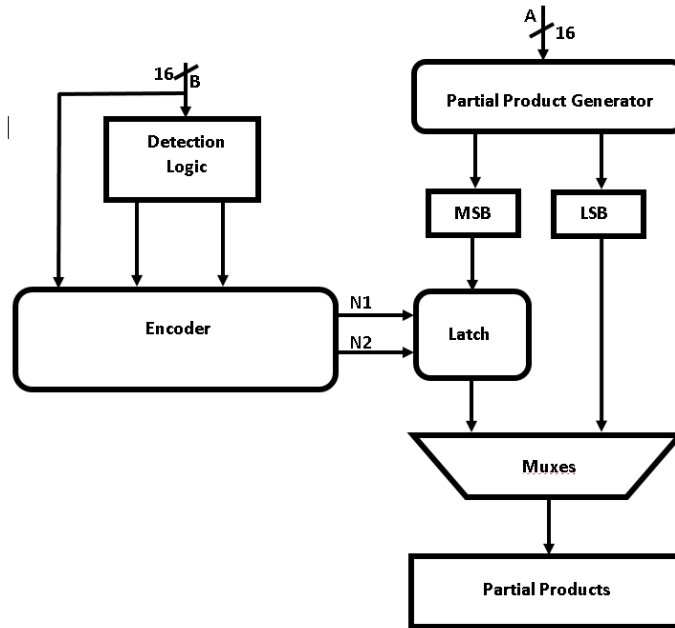
Method of Booth encoding has been used to simplify and speed up the multiplication of binary numbers by decreasing the partial products number. Radix-4 Booth encoding, in particular, examines pairs of (or triplets when considering the bit extension) the multiplier bits to generate fewer partial products.

In radix-4 Booth encoding, the multiplier bits are grouped as sets of three, starting from the least significant bit (LSB). Each group of three bits is then utilized for partial product determination. The possible groups and their corresponding operations are:

- 000 or 111: 0
- 001 or 010: +1 \* Multiplicand
- 011: +2 \* Multiplicand
- 100: -2 \* Multiplicand
- 101 or 110: -1 \* Multiplicand

This encoding decreases the partial products number by about half compared to the straightforward binary multiplication

**Fig.7 Modified Bit Pair Encoding**



Booth multiplication optimizes multiplication circuits by recoding the numbers involved, decreasing the partial products number by half through radix-4 Booth recoding. Instead of processing every column in multiplier individually, the technique only considers every second column, multiplying by  $\pm 1$ ,  $\pm 2$ , or  $0$  to achieve the same results. This method groups multiplier bits in blocks of three starting from the LSB, with each block overlapping the previous one by one bit. The initial block uses just two bits of the multiplier. This approach, illustrated in Figure 7, efficiently reduces unnecessary computations and transient signals, thereby lowering power consumption. To further enhance efficiency, a SPST-equipped modified-Booth encoder controlled by a detection unit decides whether computations are redundant, as depicted in Figure 7. Latches freeze inputs to specific multiplexers based on terms such as zero partial products, minimizing transition power dissipation.

**2. PPR- booth multiplication with pipeline processing**

The image provides a high-level overview of a pipelined multiplier design utilizing Modified Booth encoding. This design is a sophisticated approach to multiplication, aiming to enhance speed and efficiency. At the core of this system are several key components: the inputs, Modified Booth encoding, partial products pipelining, conditional gating, and the final output.

The inputs to this system are the 16-bit multiplicand ( $A[15:0]$ ) and the 16-bit multiplier ( $B[15:0]$ ). These inputs are processed by the Modified Booth encoding block, which is designed to minimize the no. of partial products needed for multiplication. Booth encoding achieves this by grouping bits of the multiplier and encoding them in a path that decreases the number of total operations required.

Following the encoding process, the paired encoded outputs are sent into the partial products pipelining block. This block consists of multiple stages, each responsible for generating and summing partial products. The pipelining structure ensures that each stage operates concurrently, allowing for partial products to be generated and accumulated in parallel. This parallel processing significantly enhances the overall speed of the multiplication process.

Conditional gating is another crucial component of this design. This block controls the flow of data based on specific conditions, effectively managing power consumption. By enabling or disabling parts of the circuit as needed, conditional gating ensures that power is only used where necessary, optimizing the efficiency of the system.

The final component is the output, represented as  $Y[31:0]$ , which is the 32-bit product result of the multiplication. After passing through all the pipeline stages, the cumulative sum of PP's is output of final result.

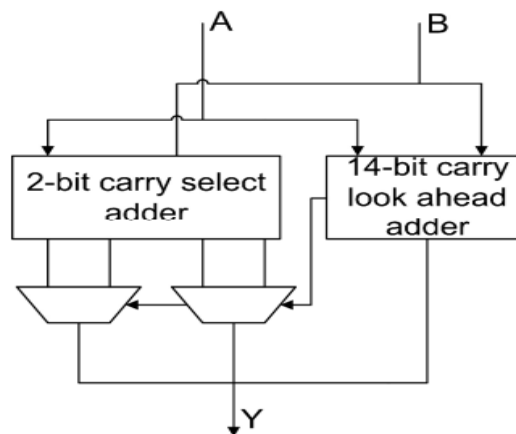
### 3. PPA- CSA+CLA(4+28)

The image depicts a hybrid adder that integrates a 4-bit Carry Select Adder with a 28-bit Carry Lookahead Adder (CLA) for enhanced efficiency. This Hybrid adder is formulated to handle two 32-bit inputs, A and B, by splitting the adding process between the CSA and the CLA. Specifically, the most significant 4 bits of the whole inputs are managed by the 4-bit CSA, while the remaining 28 bits are processed by the 28-bit CLA.

The 4-bit CSA precomputes two potential sums: one assuming a carry-in of 0 and another assuming a carry-in of 1. The actual carry-out from the 28-bit CLA determines which of these sums is selected. Meanwhile, the 28-bit CLA quickly computes sum of least significant 28 bits of A and B using precomputed propagate and generate signals, which accelerate the carry calculation compared to traditional ripple carry adders.

To correct the output Multiplexers are used to select the from the 4-bit CSA based on carry-out from the CLA. The final result, Y, is formed by combining the selected partial sum from the CSA with the sum from the CLA, effectively producing the desired 32-bit sum.

This hybrid approach leverages the fast carry computation of CLA for the majority of the bits while utilizing the flexibility of the CSA for the most significant bits, optimizing both area and speed.



**Fig.8 Bit Hybrid Adder**

### SIMULATION AND SYNTHESIS RESULTS

The developed MAC design consists of design entry, simulation and synthesis. This architecture which has been proposed is implemented in Verilog language and verified their functionality using using Cadence NCSim simulator. Once the functional verification is done, the RTL model is taken to the synthesis process using the Genus Synthesis Tool, where the optimized netlist is viewed and Area, power, timing etc is Obtained. These area ,power etc are compared with referred papers.

Our proposed MAC (Multiply-Accumulate) unit is designed for power & area efficiency. It utilizes a hybrid multiplier and hybrid adder, where the hybrid multiplier employs a modified encoding scheme. This scheme reduces the area, directly improving the speed. The design also incorporates pipeline logic in partial products synchronous functionality, further reducing power & area.

Consequently, the processed output is both area and power-efficient compared to referenced papers. The simulated and synthesised results are as follows...



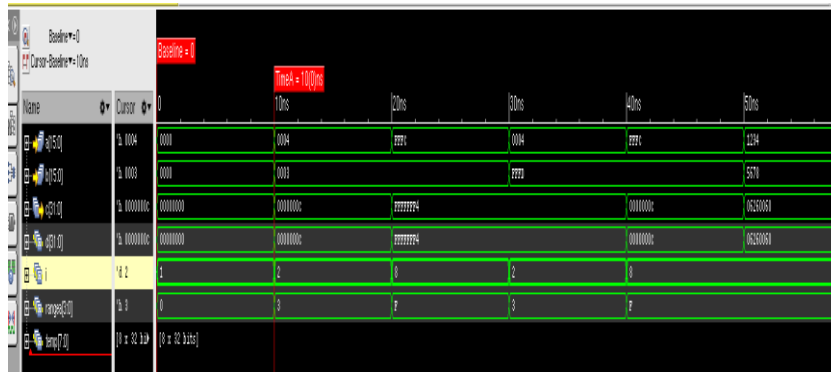


Fig.9 Simulated waveform(Multiplier)

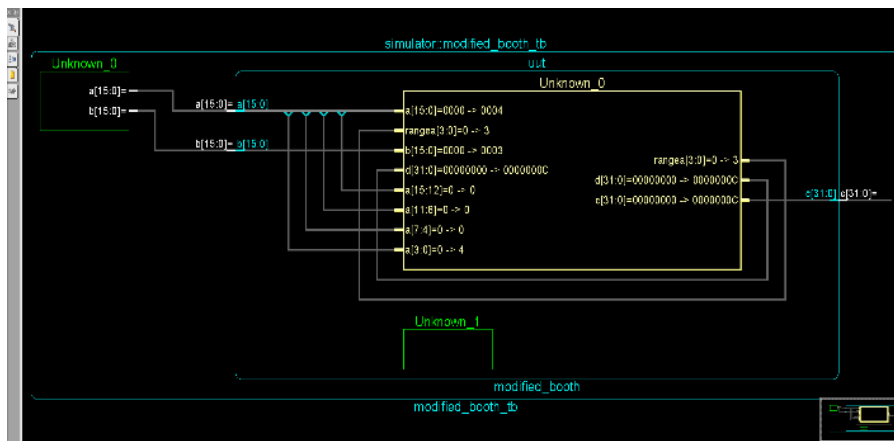


Fig.10 Simulated Schematic(Multiplier)

**Multiplier Area Report**

Instance	Module	Cell Count	Cell Area	Net Area	Total Area
<u>modified_booth</u>		705	0.000	1169.660	1169.660

**Multiplier Power Report**

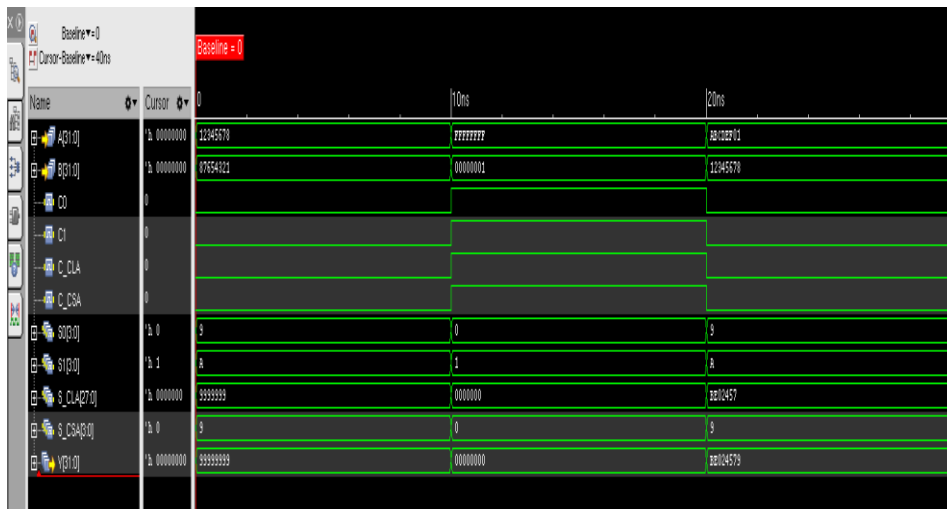
Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	1.25439e-07	1.11229e-04	2.03980e-04	3.15335e-04	100.00%
<u>bbox</u>	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	1.25439e-07	1.11229e-04	2.03980e-04	3.15335e-04	100.00%
Percentage	0.04%	35.27%	64.69%	100.00%	100.00%



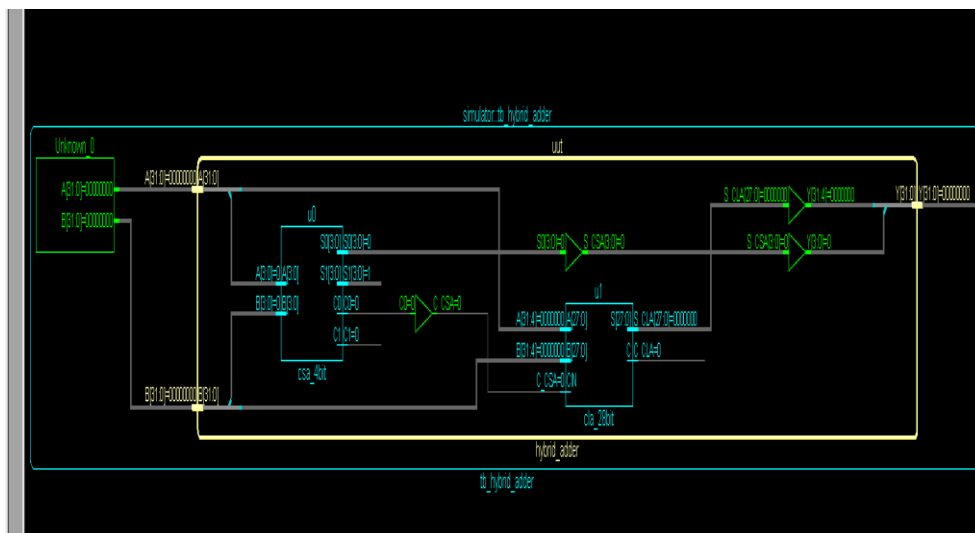
**Area & Power Comparision(Multiplier)**

Design/Criteria	Proposed Design	Conventional Design
Area	1169	1800
Power	0.315 mw	0.468 mw

The hybrid adder also plays a vital role in reducing power and area. By designing the CSA (Carry Save Adder) with fewer bits and CLA (Carry Lookahead Adder) with the remaining bits, the area is significantly reduced, which indirectly reduces power consumption as well. Thus, the combined adder is an important component of the design. The hybrid adder's simulated and synthesized results are as follows...



**Fig.11 Simulated waveform(Hybrid Adder)**



**Fig.12 Simulated Schematic(Hybrid Adder)**

### Hybrid Adder Area Report

Instance	Module	Cell Count	Cell Area	Net Area	Total Area
hybrid_adder_16bit		50	346.660	0.000	346.660

### Hybrid Adder Power Report

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	2.17394e-06	7.90739e-06	8.75646e-06	1.88378e-05	100.00%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	2.17394e-06	7.90739e-06	8.75646e-06	1.88378e-05	100.00%
Percentage	11.54%	41.98%	46.48%	100.00%	100.00%

### Area & Power Comparison(Hybrid Adder)

Design/Criteria	Proposed Design	Conventional Design
Area	346	470
Power	0.0188mw	0.217mw

Finally, the MAC unit combines a Booth multiplier and an adder. The power consumption is primarily reduced in the adder, while the area is mostly reduced in the multiplier. The final simulated and synthesized outputs are shown below

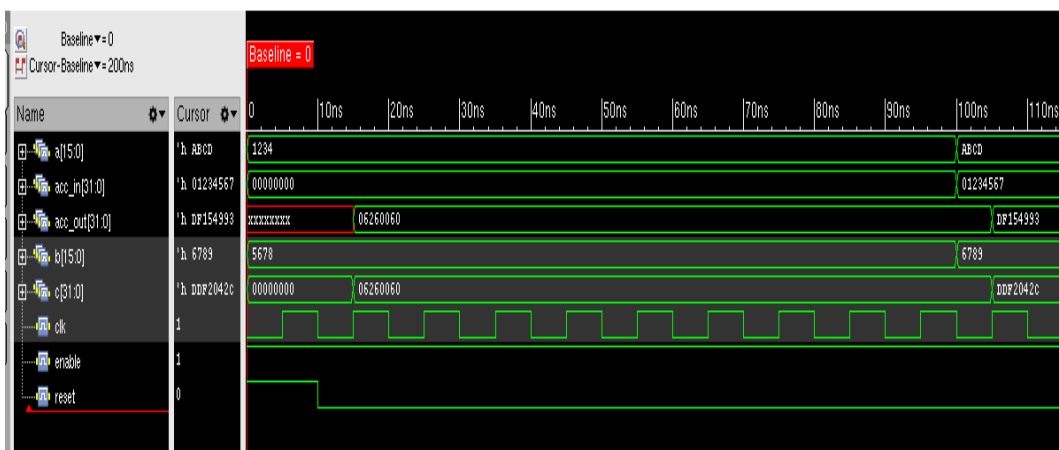


Fig.12 Simulated Waveform(MAC)

## MAC Area Report

Instance	Module	Cell Count	Cell Area	Net Area	Total Area
<u>modified_booth_mac</u>		890	0.000	1441.178	1441.178

## MAC Power Report

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	6.21180e-03	5.41650e-05	3.97859e-06	5.82057e-05	14.01%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	1.45572e-07	1.25648e-04	2.31489e-04	3.57282e-04	85.99%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	2.07690e-07	1.79813e-04	2.35468e-04	4.15488e-04	100.00%
Percentage	0.05%	43.28%	56.67%	100.00%	100.00%

## CONCLUSION

The proposed MAC (Multiply-Accumulate) unit is highly efficient in consideration of both power and area. The design effectively combines a Booth multiplier and a hybrid adder, each optimized for their respective strengths. The hybrid multiplier employs a modified encoding scheme, reducing the area and improving speed, while the hybrid adder, designed with a combination of CSA and CLA, plays a crucial role in minimizing power consumption & further reducing the area.

The use of pipeline logic in the partial products' synchronous functionality contributes significantly to reduce the power and area. The entire architecture, coded in Verilog and verified using Cadence NCSim simulator, demonstrates that the proposed MAC unit outperforms traditional designs as referenced in other papers. Synthesis using the Genus Synthesis Tool has provided optimized netlists that confirm these improvements. Overall, this proposed MAC unit's architecture demonstrates a superior balance of power & area efficiency, validated by both simulated and synthesized results, making it a valuable advancement in the MAC unit designs.

## References

1. B.Hemalatha, Dr.Hari Shanker Srivastava, V.Vinay Kumar, "Design of MAC Unit For DSP Applications using Verilog HDL", International journal of Research in Advent Technology,2019.
2. C M Selvi, A modular technique of Booth encoding and Vedic multiplier for low-area and high-speed applications "Scientific reports ,nature.com,2023 "
3. Divya Govekar, Design and Implementation of High Speed Modified Booth Multiplier using Hybrid Adder " proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC)
4. Sukhmeet Kaur , Suman and Manpreet Singh Manna "Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2) " Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 3, Number 6 (2013), pp. 683-690 © Research India Publications.
5. Aamir Bin Hamid, Nadeem Tariq Beigh, Ritu Singh "Radix-4 Modified Booth's Multiplier Using Verilog Rtl" Journal of Emerging Technologies and Innovative Research (JETIR) -2018.
6. R.Prathiba "Design of High Performance and Low Power Multiplier using Modified Booth Encode ", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016.
7. S. Radhakrishnan, Rakesh Kumar Karn, Mubarak Ali Meerasha , and T. Nirmalraj "Design of Low Power

- and High Speed MAC based FIR Filter using Hybrid Adder and Modified Booth Multiplier”, 2020 5th IEEE International Conference on Emerging Electronics (ICEE).
8. S.S.Sreeja , N.Vidhya ,J.Vinitha E.Manoranjitham, “High Speed Alu Architecture With Mac Unit For Iot Processor”, International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682 Vol.6, No.3, March 2017.
  9. Gopal Raut · Jogesh Mukala · Vishal Sharma · Santosh Kumar Vishvakarma, “Designing a Performance-Centric MAC Unit with Pipelined Architecture for DNN Accelerators”, Circuits, Systems, and Signal Processing (2023) 42:6089–6115.
  10. Samanthapudi Swathi, R.Devi, D.Bhavani, PSSN Mowlika, V.Bhavani, ” Performance Analysis of 16bit-Mac Unit Using Vedic and Booth Multiplier”International Journal of Creative Research thoughts,2023.
  11. Gennaro Di Meo, Gerardo Saggese, Antonio G.M. Strollo, Senior Member IEEE, and Davide De Caro, Senior Member IEEE, ”Approximate MAC unit using Static Segmentation”, Ieee Transactions On Emerging Topics In Computing, Manuscript Id,2023.
  12. Neha Rajput , Nidhi Sharma,Surya Deo Choudhary, “Design A High-Speed Low Power Mac Unit For The Dsp Applications Using Verilog”, 2nd International
  13. Conference On “Advancement In Electronics & Communication Engineering (Aece 2022) July 14-15, 2022.
  14. N. Manoj Kumar, G. Saravanan, D. Shyam Ganesh and S. Kanimozhi, “An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics”, BOHR International Journal of Intelligent Instrumentation and Computing, Volume. 1, 2021.
  15. Nitin Krishna V, “Performance Analysis of MAC Unit using Booth, Wallace Tree, Array and Vedic Multipliers”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 9 Issue 09, September-2020.
  16. Akella Srinivasa Krishna Vamsi and Ramesh S R, “An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics”, International Conference on Communication and Signal Processing, April 4-6, 2019.
  17. MAHMOUD MASADEH , (Student Member, IEEE), OSMAN HASAN, (Senior Member, IEEE), AND SOFIÈNE TAHAR , (Senior Member, IEEE), “Input-Conscious Approximate Multiply-Accumulate (MAC) Unit For Energy-Efficiency”, IEEE Access, 2019.
  18. Sana Zeba Bakshi & Prof. M. Nasiruddin, “Optimization Of Mac Unit Using Full Pipelined Accumulator: A Review”, © 2018 Jetir October, Volume 5, Issue 10, 2018.
  19. Roshani Pawar, Dr. S. S. Shriramwar, “Review on Multiply-Accumulate Unit”, Roshani Pawar. Int. Journal of Engineering Research and Application , ISSN : 2248-9622, Vol. 7, Issue 6, (Part -4) June 2017.
  20. Naluvala Ashwini , T.Krishnarajuna Rao , Dr. D Subba Rao, “Low Power Multiply Accumulate Unit (MAC) for DSP Applications”, International Journal of Research Studies in Science, Engineering and Technology Volume 2, Issue 8, August 2015.
  21. P.Jagadeesh, Mr.S.Ravi, Dr. Kittur Harish Mallikarjun, “Design of High Performance 64 bit MAC Unit”, International Conference on Circuits, Power and Computing Technologies, 2013.