DESIGN AND IMPLEMENTATION OF RADIX-4-8 MODIFIED BOOTH'S ENCODER USING FPGA

Soundarya K[1], Soundarya A[1], Kalavathi K P[1], Sindhu K[2]

[1]Student of Department of Electronics and Communication Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore-560082, Karnataka

[2]Assistant Professor, Department of Electronics and Communication Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore-560082, Karnataka

*Corresponding Author: soundaryagowdaa619@gmail.com

ABSTRACT

We describe a unique design methodology of approximate radix-4 Booth multipliers that can be produced at a reasonable cost and that can greatly lower the power consumption of signal processing activities that are error-resilient. The suggested method takes into account two important processing steps simultaneously by requiring the generated error directions to be opposite to each other, in contrast to previous studies that only concentrate on the approximation of either the partial product generation with encoders or the partial product reductions with compressors. In comparison to the previous designs, the suggested approximation Booth multiplier can minimize the needed processing energy under the same number of approximate bits since the internal mistakes are naturally balanced to have zero mean. This suggested multiplier design was created in Verilog HDL, simulated using Xilinx and Modelsim 6.4c.

I.INTRODUCTION

The multiplier in binary systems was transformed by the Booth multiplier, which is an essential part of digital arithmetic and computer design. This algorithmic method, which was created by Andrew Donald Booth in 1950, greatly increased the speed and efficiency of binary multiplication, making it a fundamental component of many computing systems. By adopting a sophisticated algorithm, the Booth multiplier can execute binary multiplication with just adds and bit-shifting operations. the novelty resides in the utilization of binary number patterns to minimize the amount of operations needed for multiplication, improving computational efficiency. Binary multiplication is typically performed by shifting, adding, and looping through each bit in the multiplier and multiplicand. Although this method is simple, it requires a large number of operations based on the operands' bit length, which raises the computational complexity.

An electrical circuit known as a binary multiplier is used to multiple two binary integers in digital devices, such as computers. To build it, binary adders are used. A digital multiplier can be implemented using a range of computer arithmetic methods. The majority of methods entail calculating a series of partial products and adding up all of the partial products. This procedure is adapted here for use with a base-2 (binary) numeral system; it is akin to the way long multiplication on base-10 integers is taught to elementary school students. In many applications, including digital filtering, discrete cosine transforms, and the Fourier transform, DIGITAL multipliers are among the most important arithmetic functional units. These applications' throughput is dependent on multipliers; if the multipliers are too

In digital devices like computers, an electrical circuit called a binary multiplier is used to multiply two binary integers. Binary adders are used to construct it. There are several computer arithmetic techniques that can be used to create a digital multiplier. For the most part, the methods include partial product calculations and summation of the partial products. This process is here adapted for use with a base-2 (binary) numeral system; it is similar to the way elementary school pupils are taught to perform long multiplication on base-10 integers. A number of crucial arithmetic functional units are DIGITAL multipliers, which find use in digital filtering, discrete cosine transforms, and the Fourier transform, among other applications. Multipliers are necessary to determine the throughput of these applications; if the multipliers are too

II. LITEATURE SURVEY

1. G Amar Eshwar, B Aishwarya, The design of a high-performance partial product generator for 64-bit binary multipliers using Radix-16 Booth is scheduled for 2023. Redundant Binary Partial Product Generators are used to reduce the maximum height of the partial product array produced by the radix-16 Modified Booth Encoded multiplier, while maintaining a one-row delay in the delayed partial product development block. With n=64-bit non-signed operand, the project's optimization for binary radix-16 (modified booth coded multipliers) is defined to reduce the maximum volume of the partial product columns to [n/4]. To attain best performance, hybrid designs dynamically switch between Radix-16 and other Radix Booth algorithms according on input patterns. It can be difficult to achieve precise timing in high-performance devices, and the Radix-16 Booth Partial

2. Using VHDL, Sulbha Bhongale and Rajendra Pate implemented a 32-bit Radix-4 booth multiplier in 2023. We present an implementation and design of a signed radix-4 Booth multiplier in this work. Many collation parameters have been discussed, such as the Booth radix-2, the Wallace tree multiplier, and the Modified radix-4 Booth multiplier. The radix-4 Booth 32-bit implementation is compared with an alternative multiplier. The automatic code generation made possible by the integration of the VHDL code with HLS tools facilitates design space exploration and may increase design productivity. Timing restrictions may affect the Radix-4 Booth Multiplier, particularly as clock speeds rise.

3Increasing the Power Efficiency of Radix-4 Booth Multiplier with Pre-Encoded Mechanism, G. Anantha Lakshmi, G.S.P. Prem Kumar, 2022. This research proposes a Booth pre-encoded technique to lower the Booth multiplier's power consumption. By stopping the Booth encoders and decoders from operating when not needed, the suggested design can effectively lower the power of the Booth multiplier wasted on superfluous tasks. Our approach performs better than the conventional Booth multiplier, especially when the control signals are created at the pipeline input register before to the multiplier. the development of adaptive power optimization models through the incorporation of deep learning techniques. Workload-dependent power efficiency improvements could exist.

4. The optimized Radix 4 and Radix 8 booth algorithms were implemented using FPGA in 2021 by Barma Venkata RamaLakshmi and Fazal Noor Basha. Multipliers are essential to every system's operation. The main disadvantage is that it uses more space and power. Numerous methods and techniques exist to improve performance while reducing power and area usage. The primary goal of any multiplication method is to reduce the partial product summation. The booth method is one of the most used and useful algorithms. One possible avenue to consider is customizing the design for nascent technologies such as improved semiconductors or quantum computing. absence of a comparison between the Radix-4 and Radix-8 techniques.

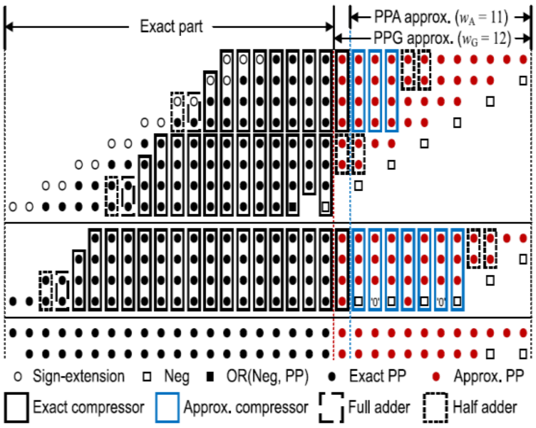
5. 2020 saw Yen-Jen Chang, Yu-Cheng Cheng, and Shao-chi Liao Radix-4 Booth Multiplier with Pre-Encoded Mechanism at Low Power. This research proposes a new radix-4 Booth pre-encoded mechanism to lower the Booth multiplier's power consumption. By stopping the Booth encoders and decoders from operating when not needed, the suggested design can effectively lower the power of the Booth multiplier wasted on superfluous tasks. The potential advantages of quantum algorithms and their relevance to power-

efficient multiplication using principles of quantum computing.

6. Suksmandhira Hari murti, Hans Herdian, and Trio Adiono designed a compact modified Radix-4 8-bit booth multiplier in 2020. The traditional serial booth multiplier requires a 2n bit adder to complete the operation and produces the full partial product (2n+1). As a result, the circuit's area will be inefficient. By using a modified booth technique, the amount of partial product calculations is cut in half compared to the serial-parallel multiplier implementation, all the while maintaining an acceptable level of area. We suggest a concept that modifies the radix-4 modified booth multiplier's architecture in order to maximize the available space. It is appropriate for secure communication networks since security measures are integrated to guard against side-channel assaults and unauthorized access. a large variety of operand sizes, possibly leading to

7Bipul Boroa, Y.B. Nithin Kumar, and K. Manikanta Reddy, IN 2019 Radix-8 Booth Multiplier Approximation for High-Speed and Low Power Applications. An emerging method in circuit design is approximate computation, which lowers energy usage while maintaining an acceptable accuracy deterioration. In order to investigate the benefits of approximation, this work proposes three approximate radix-8 Booth multipliers. Two proposed approximation Booth encoders are used in the construction of these multipliers in order to generate approximate partial products. A small number of the partial product matrix's least significant columns (AC) are filled in with approximate partial products. The multiplier may adjust its precision in accordance with the demands of the application by including dynamic precision scaling, which maximizes power consumption without sacrificing tolerable accuracy. It is intricate, requiring design, verification, and

8. Design of an Effective High-Speed Radix-4 Booth Multiplier for Signed and Unsigned Numbers, Kalaiyarasi, D., and M. Saraswathi, IN 2018. The design of an effective high-speed Radix-4 Booth multiplier for both signed and unsigned numbers is presented in this study. In contrast to traditional multipliers, the Proposed Booth multiplier is a capable multiplier that consistently handles both positive and negative numbers. Multiplication is commonly accomplished by the add and shift operation, where each multiplier bit generates one or more multiplicand bits that need to be added to the partial result. In the future, the CSA will be replaced by a Koggestone parallel prefix adder at the summation stage to further reduce delay. The multiplier's adaptability may be impacted by its inability to scale to very big operands.



III. EXISTING SYSTEM

The current design minimizes the switching capacitance in order to lower the dynamic power usage. In order to reduce power consumption, the design suggests a technique known as partially guarded computing (PGC), which splits arithmetic units, such as adders and multipliers, into two sections and switches off the unused component. According to the published results, in speech-related applications, the PGC can lower power consumption in an array multiplier with area overheads. A 32-bit 2's complement adder with a dynamic-range determination (DRD) unit, a sign-extension unit, and master- and slave-stage flip-flops for each of the adder's operands are proposed in the design. For multimedia applications, this design tends to lower the power dissipation of traditional adders. ALGORITHM CURRENT IN THE SYSTEM: Partial guarded computation (PGC)

IV. PROPOSED SYSTEM

In order to create an approximate radix-4 Booth multiplier that is more cost-effective than the current designs, we first created a modified Booth encoder that produced unidirectional errors that were more aggressive while using less hardware. The error originating from the encoder is then carefully compensated for by an approximate 4-2 compressor that has the opposite error direction. This produces a zero-mean and balanced error distribution without the need for intricate error recovery circuits.

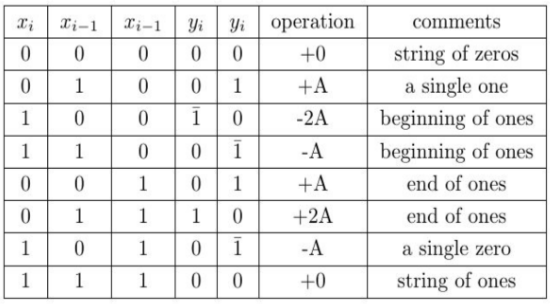
Figure: The dot diagram of our 16×16approximate multipliers

In order to reduce hardware costs, the error directions are precisely defined, resulting in the first single-gate result. The dot diagram of our 16 x 16 approximation multipliers is displayed in Fig. below. The colored dots and rectangular boxes correspond to the produced PP elements and corresponding compressors, respectively. The approximate PP elements are indicated by red in the clear figure.

V. Module Descriptions

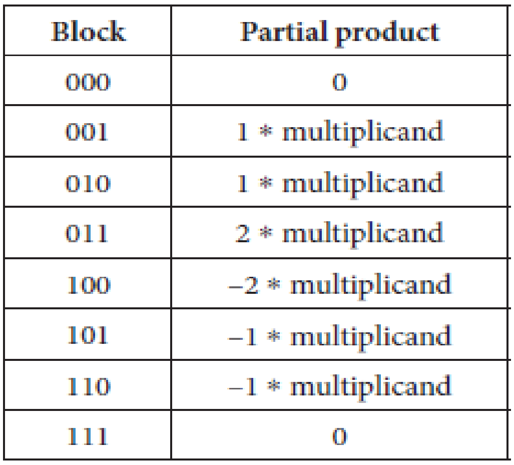
RADIX-4 BOOTH ENCODING:

Booth's Algorithm for Radix-4: The Radix-4 Booth algorithm improves upon the drawbacks of the Radix-2 algorithm. Rather than examining two bits, three bits are inspected here. While xi-2 serves as the reference bit, the bits xi and xi-1 are recoded into yi and yi-1. The value of the variable i is taken from the set {1,3,5….}. The following equation makes recoding the multiplier simple: yi yi-1 = xi-1 xi-2 - 2xi Table 2 displays the strategy for recoding the multiplier in the Booth's Radix-4 algorithm. All of the shortcomings of the Radix-2 recoding algorithm are effectively addressed by the Radix-4 technique. Three add/sub operations are carried out in total during this multiplication process. As a result, the Radix-4 algorithm requires n/2 add/sub operations overall. Every action involves two bits being



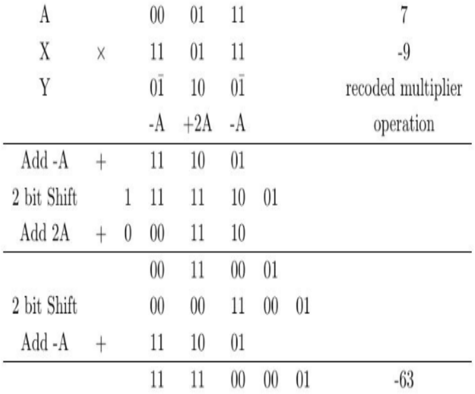
x_i	x_{i-1}	x_{i-1}	y_i	y_i	operation	comments
0	0	0	0	0	+0	string of zeros
0	1	0	0	1	+A	a single one
1	0	0	$\bar{1}$	0	-2A	beginning of ones
1	1	0	0	$\bar{1}$	-A	beginning of ones
0	0	1	0	1	+A	end of ones
0	1	1	1	0	+2A	end of ones
1	0	1	0	$\bar{1}$	-A	a single zero
1	1	1	0	0	+0	string of ones

Radix 4 Booth Encoding



Block	Partial product
000	0
001	1 * multiplicand
010	1 * multiplicand
011	2 * multiplicand
100	−2 * multiplicand
101	−1 * multiplicand
110	−1 * multiplicand
111	0

RADIX 4 Example:

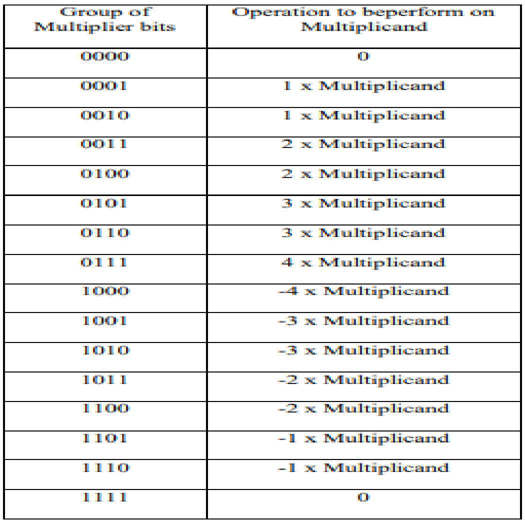


A			00	01	11			7
X	×		11	01	11			-9
Y			$0\bar{1}$	10	$0\bar{1}$			recoded multiplier
			-A	+2A	-A			operation
Add -A	+		11	10	01			
2 bit Shift		1	11	11	10	01		
Add 2A	+	0	00	11	10			
			00	11	00	01		
2 bit Shift			00	00	11	00	01	
Add -A	+		11	10	01			
			11	11	00	00	01	-63

RADIX-8 BOOTH'S ENCODER:

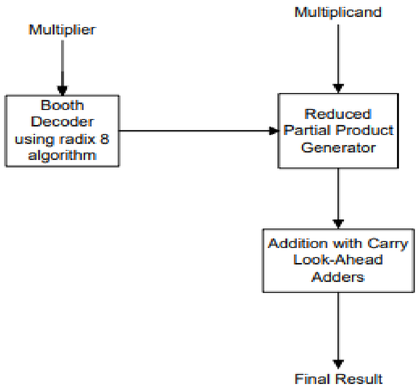
It conducts the eight various sorts of operations on the multiplicand—+M, +2M, +3M, +4M, –4M, –3M, –2M, and –M—this booth multiplier is referred to as radix-8. With the exception of 3M, all of the multiples are easily obtained by complimenting and shifting. It is not possible to generate the 3M (3× multiplicand), also known as a hard multiple, using complementation and simple shifting alone. It can be generated in 4M–M or M+2M formats. M+2M is the production company behind this project. For instance, in 8×8-bit multiplication, the 8 partial product rows are produced by a simple multiplier; however, they are reduced to 3 by a radix-8 booth multiplier. It implies that the partial product rows are decreased by the radix-8 booth multiplier.

Radix 8 booth Encoding



Group of Multiplier bits	Operation to beperform on Multiplicand
0000	0
0001	1 x Multiplicand
0010	1 x Multiplicand
0011	2 x Multiplicand
0100	2 x Multiplicand
0101	3 x Multiplicand
0110	3 x Multiplicand
0111	4 x Multiplicand
1000	-4 x Multiplicand
1001	-3 x Multiplicand
1010	-3 x Multiplicand
1011	-2 x Multiplicand
1100	-2 x Multiplicand
1101	-1 x Multiplicand
1110	-1 x Multiplicand
1111	0

PROPOSED DIAGRAM Radix-8:


Multiplier
Multiplicand
Booth Decoder using radix 8 algorithm
Reduced Partial Product Generator
Addition with Carry Look-Ahead Adders
Final Result

Verilog Structure: Implement a module that recognizes Radix-specific bit patterns in the multiplier and encodes them using appropriate logic gates.

VI. OBJECTIVE:

1.The goal of the standard Booth multiplier and the suggested Booth multiplier using compressor techniques is to maximize area and minimize delay while performing multiplication operations effectively.

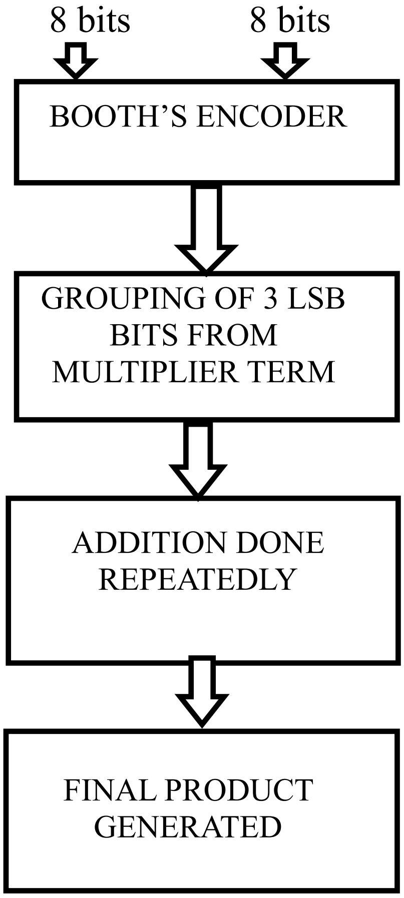
2. The conventional Booth multiplier accomplishes this by using the Booth algorithm to minimize the quantity of partial products produced, which are subsequently added to sequentially. Due to its sequential structure and requirement for a complete set of partial products, it might, nevertheless, experience greater area and delay problems.

3. By using compressor circuits to compress and aggregate partial products more effectively, the suggested Booth multiplier, on the other hand, minimizes area and latency.

VII. PROBLEM STATEMENT:

Developing reduced compressor and encoder designs for an affordable approximate radix-4 Booth multiplier is the problem. With regard to conventional Booth multipliers, this project seeks to minimize hardware complexity while maintaining a reasonable level of performance. Using approximate computing principles to improve computational efficiency, investigating novel ways to simplify compressor and encoder circuits in the radix-4 architecture, and incorporating low-power design techniques for energy-conscious applications are the main goals. It will be crucial to balance accuracy, speed, area, and power consumption, which calls for a careful analysis of trade-offs.

VIII. FLOW CHART AND PROCEDURE FOR SOLVING MODIFIED BOOTH ENCODER



USING MODIFIED BOOTH ENCODER

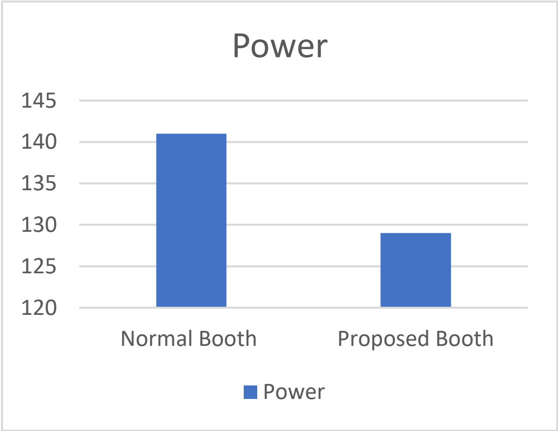
Modified to generate a maximum of n/4+1 partial products, booth 4. (For unsigned numbers) Algorithm
1. Add one zero to the LSB.
2. If n is even, pad the MSB with two zeros; if n is odd, pad it with one zero.
3. Split the multiplier into three-bit chunks that overlap.
4. Using the amended booth 2 encoding table, calculate the partial product scale factor.
5. Determine the Multiples of the Multiplicand Sixth Total Partial Product

Advantages

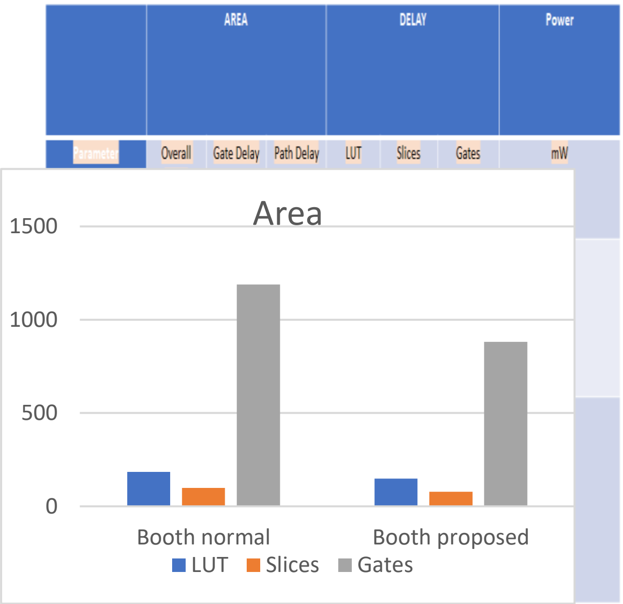
Compared to the booth's multiplier, the Modified Booth Encoder will require less space.
Equations in mathematics can be resolved with its help.
Less ports are available.
Faster since there is less delay
Reduced energy usage (measured in watts).

DISADVANTAGES

Booth encoders get larger as we go toward higher radix.
When compared to booth multiplier, it is less frequent.



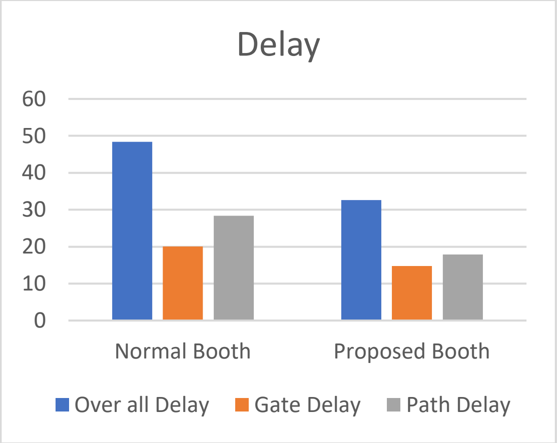
IX. COMPARISON REPORT FOR MULTIPLIER:



	AREA			DELAY			Power
Parameter	Overall	Gate Delay	Path Delay	LUT	Slices	Gates	mW

X. CONCLUSION:

This work has considered error characteristics from PPG and PPA stages to propose two approximate radix-4 Booth encoder and 4-2 compressor solutions. We force two simplified procedures to have different error directions while limiting their hardware costs with aggressive unit-gate topologies, balancing the resulting errors for expanding the range of approximation. As a result, the suggested multiplier exhibits competitive accuracy with noticeably less hardware complexity when compared to the earlier studies with biased error generators, resulting in the most alluring quality-energy trade-offs in the real-world error-tolerant applications.



XI. REFERENCE

1.B Aishwarya, G Amar Eshwar, In 2023 Design of High-performance Radix-16 Booth Partial Product Generator for 64-bit Binary Multipliers.

2. Sulbha Bhongale, Rajendra Pate, in 2023 Implementation of Radix-4 (32 bit) Booth Multiplier using VHDL.

3. G. Anantha Lakshmi, G.S.P Prem Kumar, in 2022 Increasing the Power Efficiency of Radix-4 Booth Multiplier with Pre-Encoded Mechanism

4. Barma Venkata RamaLakshmi and Fazal Noorbasha, in 2021 FPGA Implementation of Optimized Radix 4 and Radix 8 Booth Algorithm

5. Yen-Jen Chang, Yu-Cheng Cheng, Shao-chi Liao, in 2020 A Low Power Radix-4 Booth Multiplier with Pre-Encoded Mechanism.

6. Trio Adiono, Hans Herdian, Suksmandhira Harimurti, in 2020 Design of Compact Modified Radix-4 8-Bit Booth Multiplier.

7. Bipul Boroa, K. Manikanta Reddy, Y.B. Nithin Kumar, IN 2019 Approximate Radix-8 Booth Multiplier for Low Power and High-Speed Applications.

8. Kalaiyarasi. D, M Saraswathi, IN 2018 Design of an Efficient High-Speed Radix-4 Booth Multiplier for both Signed and Unsigned Numbers.

9. W. Liu, L. Qian, C. Wang, H. Jiang, J. Han and F. Lombardi, "Design of Approximate Radix-4 Booth Multipliers for Error Tolerant Computing," IEEE Transactions on Computers, Vol. 66, No. 8, pp. 1435-1441, 2017.

10. T. Adiono, H. Ega, H. Kasan, S. Fuada and S. Hari murti, "Full custom design of adaptable Montgomery modular multiplier for asymmetric RSA cryptosystem," 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Xiamen, China, pp. 910-914, Nov. 6-9, 2017.

11. A. Akbar, T. Adiono, S. Hari murti and T. A. M. Putra, "Design of Neuron Net Function using Modified Radix-4 Booth Multiplier with a Flipped Logic Parallel Prefix Adder," 2018 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, pp. 1-6, Oct. 23-24, 2018.

12. D. Chandrika Sowmini, A. Lavanya, D. Jagadeesh Sai, C. Raja and M Aditya, "Design and Analysis of Multipliers using Radix-8 Booth Encoding Technique for Low Power and Area Consumption", International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2, 2019.

13. Duncan J. M. Moss, David Boland and Philip H. W. Leong, "A Two-Speed Radix-4 Serial-Parallel Multiplier", IEEE TRANSACTION Son Very Large-Scale Integration (VLSI) Systems, vol. 27, no. 4, 2019.