

Original Article

Design of Delay Efficient Multiplier Using Parallel Prefix Adders

C V P Supradeepthi¹, B Veena², M Hima Bindu³

^{1,2,3}Assistant Professor, ECE, IARE, India.

Received Date: 21 June 2023

Revised Date: 10 July 2023

Accepted Date: 21 July 2023

Abstract: As technology advances, the demand for speedy and efficient real-time digital signal processing applications has grown. Every application requires multiplication as one of the main arithmetic operations. To boost their speed, a vast number of multiplier designs have been devised. A new technique for designing High-Speed multipliers is proposed in this work. With four 8X8 approximate multipliers, three parallel prefix adders [PPA], and one OR gate, this proposed 16X16 approximate multiplier construction is offered. The insertion delay of the parallel prefix adder is longer, resulting in a faster increase in the superior for the count. The 8X8 multiplier was built using the approximation tree compressor [ATC] and the carry maskable adder [CMA]. In comparison to the traditional Wallace Tree Multiplier, the proposed multiplier has a shorter delay. In the Xilinx ISE 14.7 design suite, all multiplier structures are created in Verilog. In terms of area (number of LUTs) and delay, the proposed designs are compared to typical multiplier designs (ns).

Keywords: Approximate Multiplier, Parallel prefix adders, Brentkung adder, Han Carlson adder, Ladner Fishcer adder, Kogge Stone adder.

I. INTRODUCTION

Multiplication is an integral feature of mathematical computations. Multiplication-primarily based operations consisting of Multiply and gather and Inner product is one of the most commonly used Computation intensive arithmetic functions in today's digital signal processing (DSP) algorithms, Convolution, dynamic Fourier rework (FFT), filtering, and microprocessor mathematics and precise judgement unit. Most DSP algorithms' execution time is governed by multiplication there can be a need of an immoderate velocity multiplier. Presently, multiplication time continues to be the dominant component in figuring out the education cycle time of a DSP chip. The decision for excessive pace processing has been growing due to increasing computer and sign processing programs. In many real-time sign and photograph processing programmes, higher throughput mathematics operations are required to provide the best performance [2]. Multiplication is one of the most important mathematical processes in such algorithms, and the creation of fast multiplier circuits has long been a hot topic. For many applications, reducing the amount of time taken away and the amount of electricity consumed is critical [1].

Today's political cryptographic signature synthesis and a range of applications rely heavily on multipliers. As technology progresses, many academics are attempting to design multipliers that meet both the design goals of speed and low power consumption, as well as format regularity, resulting in a diminution in space or 1 Where they are incorporated in one multiplier, they are suitable for a multitude of fast velocities, power management, compact VLSI implementations. The "add and shift" set of rules is a frequent multiplication algorithm[3]. The range of partial products to be introduced in parallel multipliers is the major parameter that influences the multiplier's performance. Of the most well approaches for decreasing the amount of two's complement to be supplied is the modified booth set of rules. in order to boost speed To limit the number of possible sequential degrees, the Wallace Tree algorithm might be utilised. Furthermore, we may show the benefit of both techniques in one multiplier by merging the modified sales space algorithm with the Wallace Tree methodology [4]. Advanced technology researchers, attempted by way of in digital signal processing applications, multipliers are used to boost the rate while lowering the power consumption. The core parameter is used to determine how often the multiplier multiplicity of the following partial products are awaiting addition performs. The multiplier is governed by whether the program is serial or parallel [1].



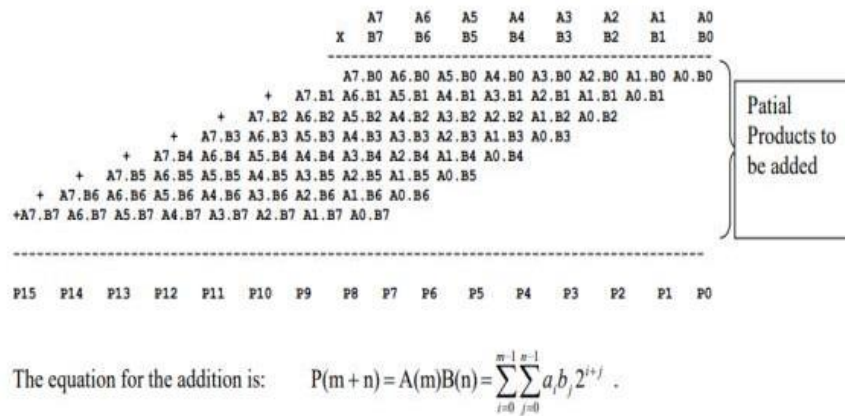


Figure 1: 8-bit Binary Numbers Multiplication

II. PROPOSED METHODOLOGY

One of the most essential design requirements for practically all digital structures, especially portable structures such as smart phones, capsules, and specialist devices, is to keep the thickness to a minimum. It is evident that achieving this minimization with the least amount of performance (speed) penalty is better. For understanding many multimedia programs, digital signature processing (DSP) blocks are a crucial complement to these portable devices. Because of this, we can use approximations to improve rate/power efficiency. This is due to human beings' limited perceptual abilities when viewing images or videos. The accuracy of mathematics operations isn't important to the device's functionality in other fields, such as photo and video processing apps [1].

Approximation can be performed in a variety of ways, including allowing some timing violations (e.g., highlight approximation processes (e.g., changing the Boolean feature of a circuit), or a combination of them. In the category of distinctive matching set, a multitude of approximated mathematics building components, such as adders and multipliers, were provided at different layout stages. Truncation is frequently utilized in fixed-width multiplier designs to simplify the multiplier's hardware complexity. The quantization error generated by the clipped element is then corrected with a constant or variable correction period. The multiplier approach focuses on accumulating partial goods that are crucial for present consumption [2]. To alleviate design complexity, partial commodities are formed by using corrupted array multipliers and trimming the tiniest parts of input. Some adder circuits with incomplete product accumulation are saved by the suggested multiplier.

Approximate computing has sparked a lot of interest as a promising paradigm for error-tolerant software that can minimise power consumption, delay, and area with a few degrees of inaccuracy. To begin, an incomplete adder is used to create an 8-bit approximate multiplier, which is then used to create a 16-bit approximate multiplier. To make a sixteen-bit approximate multiplier, we used four 8-bit approximate multipliers, three parallel prefix adders, and one OR gate. The parallel prefix adder can activate increment in the superior for the count in less time with a shorter insertion delay. The current technique employs half adders and complete adders for the addition of partial products. In the proposed technique, however, incomplete adders are used alongside half adders and complete adders. The suggested Wallace Tree Multiplier has a smaller put-off than the regular Wallace Tree Multiplier[5]. These incomplete adders assist in the truncation of partial products, which allows for the approximation of partial products, which speeds up the process and so reduces the delay.

Using the Xilinx ISE Suite 14.7 software package, the requested work is designed in Verilog. Different strategies, such as tolerating some time violations and function approximation techniques, or a combination of them, can be used to approximate something. Approximate multiplier designs mainly use three approximation approaches:

- Approximation in generating the partial products.
- Applying truncation in the partial product tree[2].

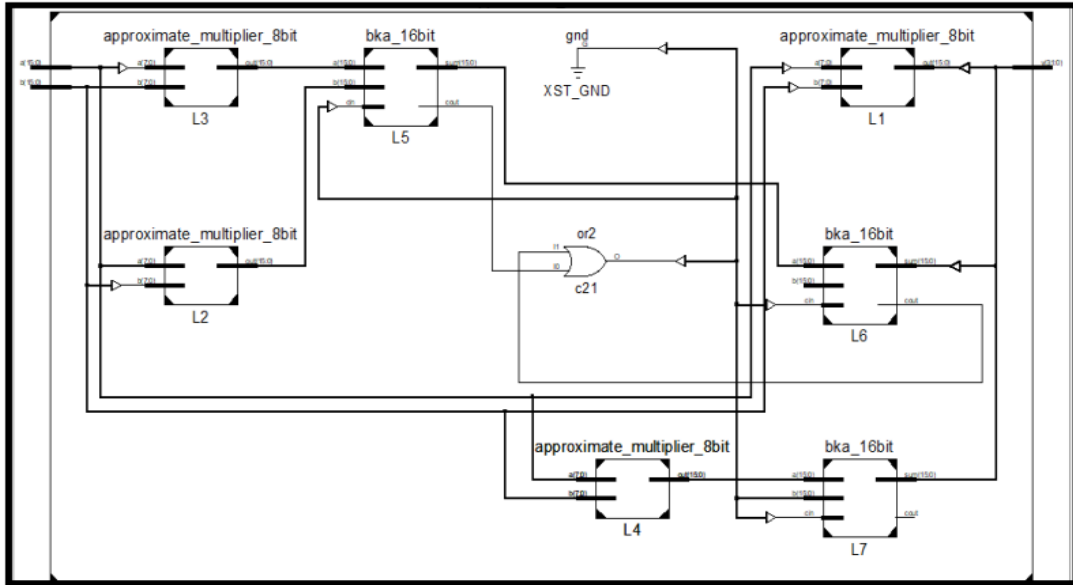


Figure 2: 16-bit Approximation Multiplier Schematic Utilising 16-bit BKA

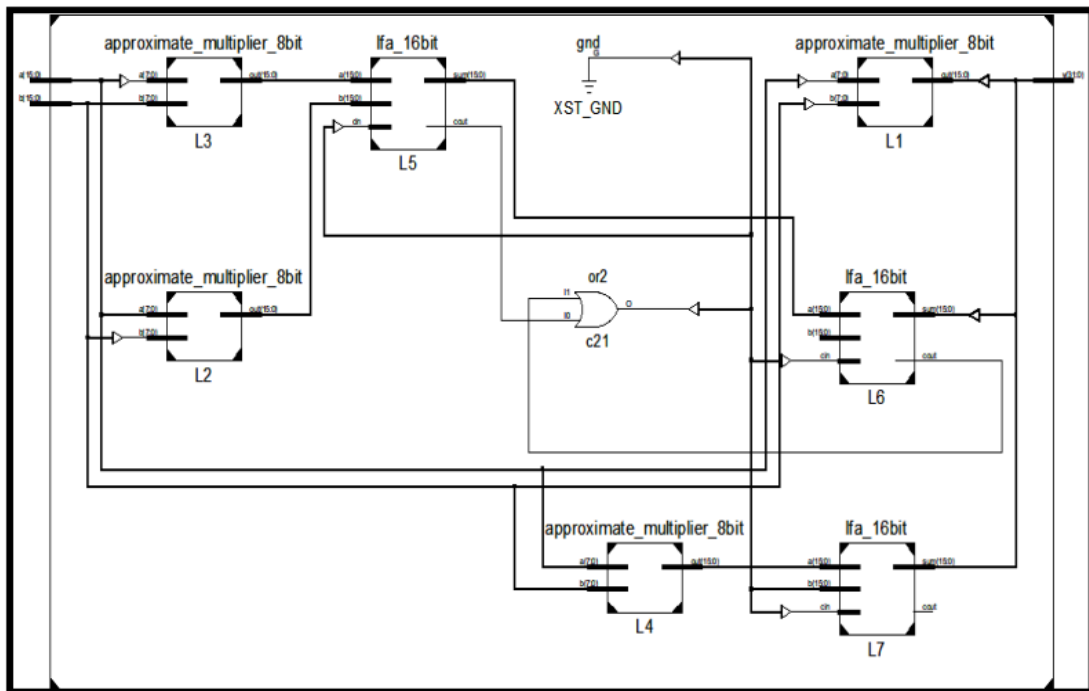


Figure 3: 16-bit Approximation Multiplier Utilising 16-bit LFA Schematic

III. RESULTS

We first looked at which parallel prefix adders are the most efficient in terms of the number of LUTs occupied and delay(ns), and then displayed their graphs.

Table 1: In Terms of Latency and Area, Four PPAs were Compared.

NAME OF THE ADDER	NO. OF LUTs	DELAY(ns)
Brent Kung adder	25	4.77
Han Carlson adder	29	5.33
Ladner fisher adder	27	4.55
Kogge stone adder	42	4.84

COMPARISON OF DIFFERENT PPAs IN TERMS OF DELAY

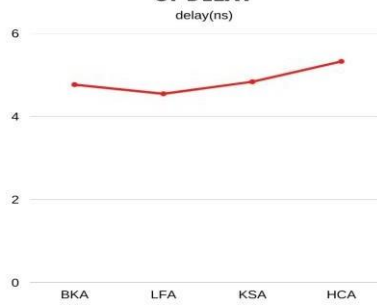


Figure 4: Graph Showing the Delay Took in Different PPAs

COMPARISON OF DIFFERENT PPAs in terms of LUTs

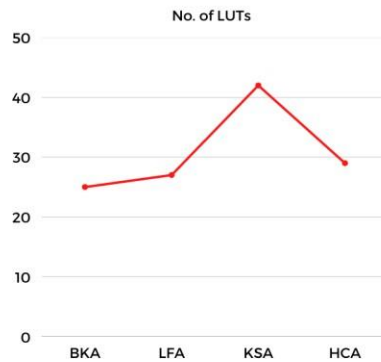


Figure 5: Graph Showing the No. of LUTs Used In Different PPAs

By seeing the above table that shows comparison of four parallel prefix adders, brent kung adder and ladnerfischer adder are most efficient in terms of both area and delay. Hence, we designed 16-bit approximate multiplier with both brentkung and ladnerfischer adder in the paper. By using this approach power consumption also decreases and complexity of circuit gets minimized.

In this study, we propose a new method for reducing the delay and area in multipliers. We have designed a 16-bit approximate multiplier using two efficient parallel prefix adders by comparing with other PPAs and have noted the number of LUTs that occupied for the construction of approximate multipliers and also delay that required to generate outputs for their respective inputs.

Table 2: Comparison of Constructed Approximation Multipliers is made

Name of the Multiplier	No. of LUTs	Delay(ns)
Approximate multiplier using Brent kung adder	429	13.207
Approximate multiplier using Ladner Fischer adder	432	12.323

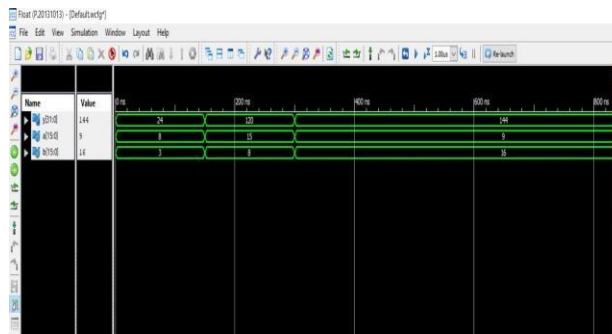


Figure 6: Using BKA Output Waveform of 16-bit Approximation Multiplier

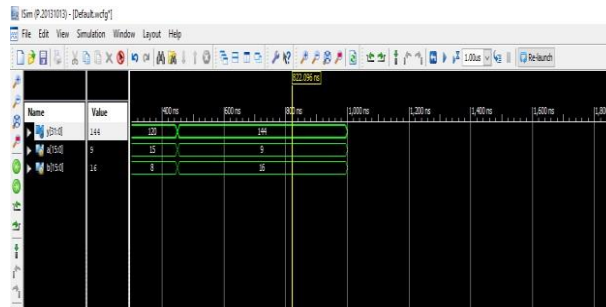


Figure 7: LFA's Output Waveform of a 16-bit Approximation Multiplier

IV. CONCLUSION

Total four parallel prefix adders structures are compared in this paper in which two are proposed approximate multipliers using two different PPAs named brent kung adder and ladnerfischer adder. In terms of delay parameter and also area parameter the difference of LUTs occupied and delay (ns). When compared to each other, the difference between the standard Wallace tree multiplier and the proposed multipliers is relatively small. In proposed multipliers, approximate multiplier using ladnerfischer adder is having least delay but its area parameter is more when compared to brent kung adder. In terms of area parameter, approximate multiplier using brent kung adder is having least area among proposed structures. So it can be concluded that the proposed multipliers are better in delay and area when compared with the conventional Wallace tree multiplier structure. Among the proposed structures, approximate multiplier using Ladner fischer adder can be used for application of circuits which require less area and speed of the circuits will also be in medium. Whereas for high speed application circuits, approximate multiplier using Kogge stone adder can be used but costing little bit more area when compared to other proposed structures. Accordance with the needs, the proposed multiplier topologies can be employed in high-speed and low-area application circuits. Designing the multiplier circuit with a larger input size may yield in a multiplier structure with lesser LUTs and, as a result, a better delay value. Depending on the needed computational precision, the proposed approximation multiplier lowered power consumption by 47.3 percent to 56.2 percent and critical path latency by 29.9 percent to 60.5 percent. The design area of the approximation multiplier was 44.6 percent smaller than the traditional Wallace tree multiplier. The proposed multiplier offered the optimal balance of power, latency, and accuracy.

V. REFERENCES

- [1] "Approximate adders for low-power digital signal processing," says the paper. Vol. 32, no. 1, pp. 124-137, IEEE Trans. Computer- Aided Design Integr. Circuits Syst., Jan. 2013. "Low-power digital signal processing using approximate adders," IEEE Trans. Computer- Aided Design Integr. Circuits Syst., vol. 2. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.- Aided Design Integr. Circuits Syst., vol. 2. In Proc. 31st Asilomar Conf. Signals, Circuits Syst., Nov. 1998, pp. 1178-1182.
- [2] "Data-dependent truncation strategy for parallel multipliers" is discussed. E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation technique for parallel multipliers," in Proc. 31st Asilomar Conf. Signals, Circuits, and Systems, Nov. 1998, pp. 11-13.
- [3] "Design of low-error fixed-width modified booth multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522-531, May 2004. K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified hardware implementation," IEEE Trans. Very Wide Spread Integr. (VLSI) Syst
- [4] "Design of a fixed-width modified booth multiplier with reduced error," May 2004, vol. 12, no. 5, pp. 522-531, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. "Design of low-error fixed-width modified hardware implementation," K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, IEEE Trans. Very Wide Spread Integr. (VLSI) Syst. 4, IEEE Trans. Very Wide Spread Integr. (VLSI) Syst. IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850-862, April 2010.
- [5] "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850-862, April 2010. "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," by H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas. IEEE Transactions
- [6] "Design and analysis of approximation compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984-994, April 2015. A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate multipliers for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984-994, Apr. 2015.
- [7] "Partial product perforation for design-efficient approximation multiplication circuits," says the paper. Vol. 24, no. 10, pp. 3105-3117, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Oct. 2016. "Architecture approximate multiplication circuits spanning partial product wound," by G.Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi. IEEE Transactions.