DESIGN AND IMPLEMENT CARRY LOOK-AHEAD ADDER BASED 32-BIT VEDIC MULTIPLIER

Dr. Shobha S. Nikam, Assistant Professor, Department of Electronics and Telecommunication, AISSMS Institute of Information Technology, Pune.
Pratik Thorat, Roshan Sharma, Prathamesh Yadav, AISSMS Institute of Information Technology, Pune.

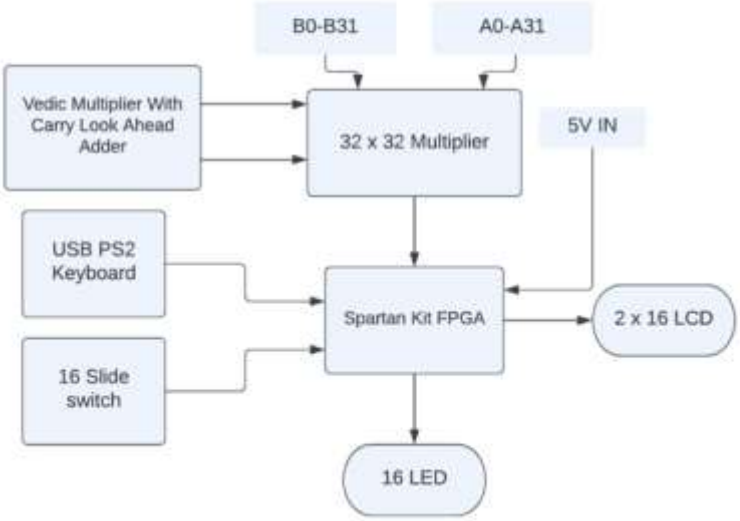
ABSTRACT

The design and implementation of high-speed multipliers be critical in enhancing the performance of digital systems, comprising processors, digital signal processors, and application-specific integrated circuits. This research focus on the development of a 32-bit multi-venue utilizing a ‘Carry Look-Ahead Adder’ (CLA) to address the innate delays associate with traditional multipliers. Traditional multipliers such as array multipliers and Booth multipliers are often limited by their sequential additional processes, causing significant delays. The Cary Look-Ahead Adder, known for its ability to reduce carrying propagations delays through parallel processes, offers a promising solution. This paper presents a comprehensive design methodology for integrate the CLA into the multiplicative process, which includes generating partial products, implementing the CLA for fast additional, reducing partial products through a tree structures, and final summation to produce the 32-bit resulting. Various optimization techniques, such as pipelines and hardware-efficient design, are employment to further enhance the performance. The experimental evaluation involves synthesizing the proposed multiplier design on a Field Programmable Gate Array (FPGA) platform and compare its performance metrics -- speed, No. of LUTs, and memory requirement against traditional multipliers. The results demonstrates that the CLA-base multiplier achieves superior computational speeds, making it available options for applications requiring high-speed arithmetic operations. The significance of this research lies in its potential to improve the ancients of digital systems by providing a faster, more efficient multiplier design. The findings contribute to the ongoing efforts to optimize arithmetic operations in digital circuits, thereby supporting advancements in various technological fields. These more detailed abstract provides a clear overview of the motivations, methodologies, and significance of the researches. It sets the stages for the rest of the papers by summarizing the key aspects of the designs and its impacts.

Keywords: VLSI, VHDL, Vedic Multiplier, FPGA

Introduction

The evolution of digital systems have necessitated the continuous improvement of arithmetic operations, particularly multiplications. Multipliers are the essential building blocks of digital system wherein multiplications is fundamental in a large number of computational tasks such as Digital signal processing (DSP), cryptographic applications, scientific computing etc. Arithmetic operations such as multiplication are one of the major components which determine the efficiency of these systems. The multiplier architectures such as array multipliers and Booth multipliers exhibit enormous delays in computation owing to their fixed addition operation. They generate partial products and sequential summation in multiplication process of an array multiplier, which leads to a delay caused by the carry propagation at each adder stage. The issue with the sequential addition process is somewhat mitigated using Booth multipliers to diminish the number of partial products, but still exists. All of these limitations impose the need for pursuing other approaches for realizing multipliers in faster and more efficient way. The Advanced adding circuits like Carry Look Ahead Adder (CLA) can also be one of the promising solutions which can be integrated along with multiplier design. CLA is best known for its capability of high-speed addition by precomputing carry signal and hence going ahead parallel with the processing of carry bits thus minimizing the time to be taken in carrying down. This is an interesting feature of CLA which makes it a better candidate to speed up the multipliers. This research targets the design and implementation of a 32-bit multiplier using CLA to overcome the performance limitations in the conventional multiplier architectures.



Related Work Review

Various digital system design has been researched and optimized in this context because multipliers offer prominent usage in different computational activities. In this section, the major advancements and current approaches in multiplier design are discussed which include various existing methods to formulate most optimum method for increasing speed, efficiency and performance.

Traditional Multiplier Design:

Array Multiplier: It is one of the simplest and most used design. This works by producing a matrix of partial products, from which can be found the sum easily. Though they are simple to understand and build, array multipliers have large delays because each serial addition step requires the previous bit's carry. Combined with no parallel blocks of adders, causes a ripple effect to take place.

Booth's multiplier: It is implementation of array multiplier with reduced number of partial products. The encoding of multiplier helps it in accommodating the positive and negative numbers efficiently. However, even with the advantage of this effect, since Booth multipliers again employ these sequential addition processes there is inherently a source of delay due to carry propagation.

Advanced Adder Based Multipliers:

In order to overcome the problems of conventional multipliers, researches have been proposed utilisation of advanced adder structures as algebraic circuits such as Carry Save Adder (CSA) and nonlinear logic circuit like Carry Look-Ahead Adder (CLA) in multiplier implementations.

CSA is an architecture which reduces the carry propagation delay and will save carry and add in subsequent stages. It plays a major role in tree-based multipliers such as wallace tree, Dadda-tree etc. Thanks to precomputing carry signals, this is precisely where the CLA excels and offers a substantial speed advantage: It is able to do this by simultaneously producing and feeding carry information back on each stage, shortening the delays related to carrying.

New and Upcoming Innovations and a few Optimization Techniques:

Subsequent research works have contributed to improving the design of the multipliers on introducing pipelined approaches, parallelism and resource efficient designs. Pipelining is breaking the multiplication into small steps (i.e. divide in N number of stages) such that during cycle 2 while result of cycle one (output of first stage) is being computed. This scheme is especially helpful for large-bit-width multipliers as the complexity and delay impact on addition operations are substantial.

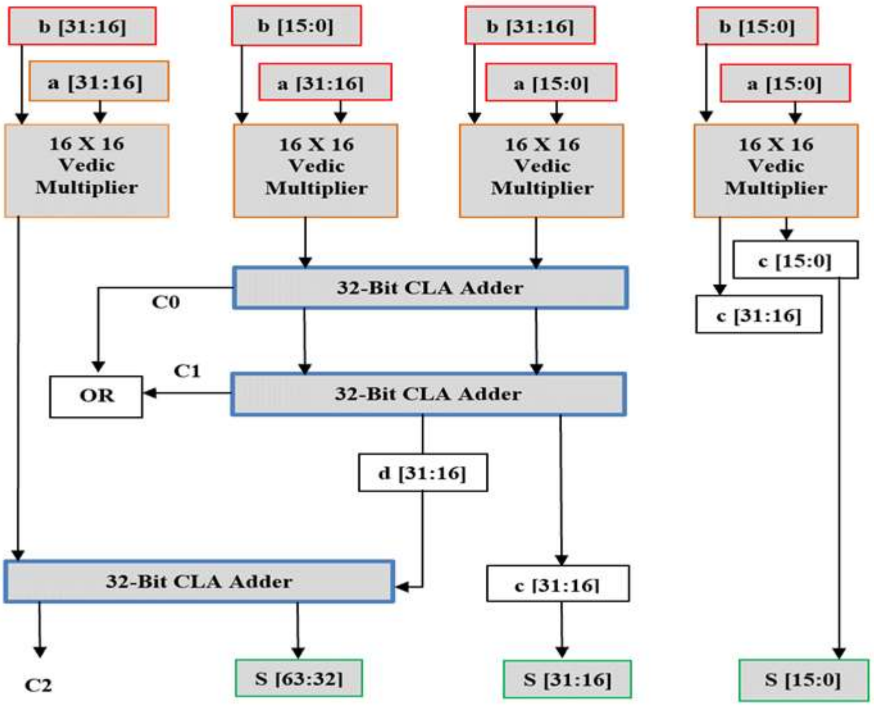
We can observe that parallel processing uses the natural way of paralleling the adder circuits as in CLA (as number of bits available for addition along two i/p remain unchanged) to do multiple additions at one go. This reduces the computation time to a large extent and helps in increasing the speed of multiplier.

Problem Statement

"Design and Implement Carry Look-ahead adder based 32-bit Vedic Multiplier."

Design Methodology

The design of a 32-bit multiplier using a Carry Look-Ahead Adder (CLA) involves a series of methodical steps aim at speed and efficiency. The methodology can be break down into four main stages: partial product generation, CLA that implementation for fastest adding, partial product reduction, and final summation.

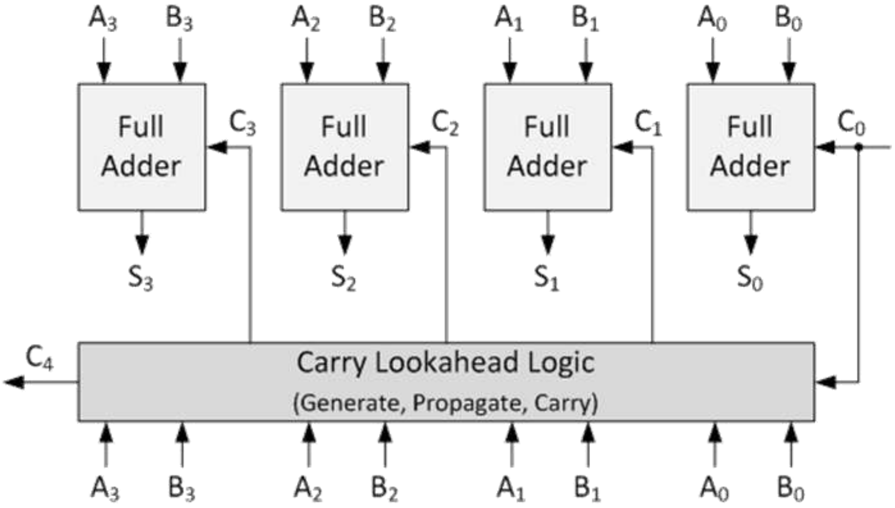


Partial Product Generation: The first step involves the generator partial products from the multiplicand and multiplier. Each bit of the multiplier is ANDed with each bit of a multiplicand produce a set of few partial products. For a 32-bit multiplier, these result in 32 sets of few 32-bit partial products.

Carry Look-Ahead Adder Implementation: The core of the design are the implementations of the CLA to handle the adding of partial products. The CLA improves the speed of adding by precomputing carry signals using generate (G) and propagate (P) functions. The generate function determine if a carry is generating by particular bit position, while the propagate function check if the carry will be passing to next position. By precomputing these signals, the CLA can determining the carry for each bit position in parallel, reducing the adding time significantly.

Partial Product Reduction: Once the partial products generating, they are organizing in a structure manner, often using a tree structure like Wallace. These structures reducing the few partial products in logarithmic time, leveraging the faster adding capabilities of CLA. Each level of the tree performs a series of adding use CLA, progressively reduce the partial products.

Final Summation: The final step involving summing the reducing partial products to produce the finally 32-bit result. The remaining partial products, significantly fewer in the number, add use CLA to ensuring the finally adding in fast as could possible.



Performance Parameters

Computational Speed:

The computational speed is one of the most important performance parameters for any multiplier design. This can be considered the delay or latency from our input to our output it is usually given as how many clock cycles we need to complete a multiplication**.** Primary objective of including CLA in multiplier is to reduce the delay happening due to carry handling. Carry propagation creates large heat and it consumes more no. of times (delay). It's a big bottleneck for Adder implementations like Ripple CAs to measure the computational speed of the suggest 32-bit CLA based multiplier, we analyse the propagation delay during different stages of multiplication like in generation of partial products, reduction of partial products and also final summation.

Optimization Techniques:

The optimization techniques play a vital in the improved performance of these two ways of 32-bit multiplications with Carry Look-Ahead Adder escaped multiply. Developing techniques that maximize speed, minimizing power consumption and optimizing resource use.

This technique is known as Pipelining. The pipelining divides the multiplication process into few stages and all stages will go cooking simultaneously in different phases of pipeline.

Parallel Processing: The inherent parallelism of the CLA is Used To perform multiple addition operations concurrently.

Hardware Efficient Design: Last but not least is the hardware design to achieve a reasonable trade-off between speed, power consumption and silicon area. Here, we do this by techniques such as gate-level optimization, efficient circuit layout and low-power design strategies.

Hardware Requirements:

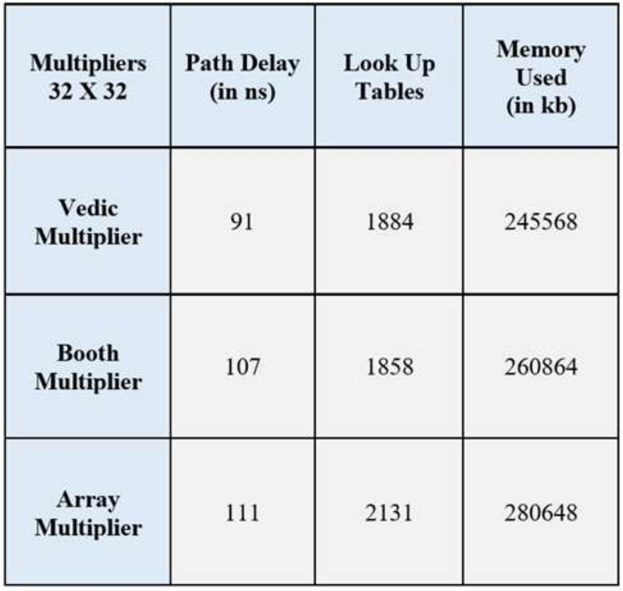
The hardware prerequisites needed to construct the 32-bit multiplier with CLA are sufficiently enough Field-Programmable Gate Array (Xilinx Spartan 3) resources i.e. logic elements, lookup tables (LUTs), and flip-flops for implementation of the complicated CLA & partial product reduction tree. Also, for managing intermediate data and enabling pipelining sufficient memory and storage are required. In addition, power consumption and silicon area are considered important parameters with optimization done to reduce them as much as possible without compromising the dynamic of computational speed and efficiency. "It is critical for the design to be scalable and compatible with current digital systems" for practical application.

Experimental Evaluation

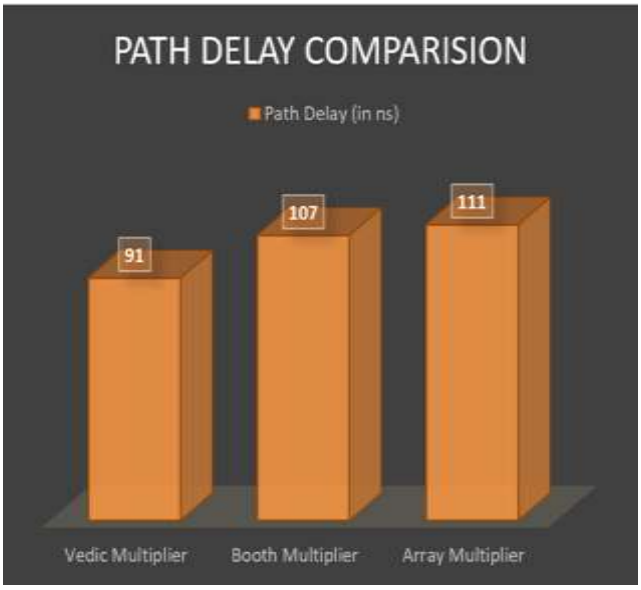
From our experiments we got a result where Vedic multiplier performs good in processing delays around 91 ns, when compared to an Array multiplier and Booth multiplier. Memory needed for Vedic multiplier is also lower compared to Array multiplier and Booth multiplier.

Conclusion

This project successful design 32-bits multiplier using Carry Look-Ahead Adder, majorly improving multiplying speed by minimize carrying propagation delay. Using CLA, combining with optimization techniques like pipelines and hardware-efficient designing, resulting in high-performance, lower-power multiplier fitting for advance digital system. Experimental evaluation confirmed that multipliers enhance the speed and efficiency versus a traditional design. These approached demonstrate a promising solution for high-speed arithmetic operations, creating a path for future research and development in digital circuit optimization.



Multipliers 32 X 32	Path Delay (in ns)	Look Up Tables	Memory Used (in kb)
Vedic Multiplier	91	1884	245568
Booth Multiplier	107	1858	260864
Array Multiplier	111	2131	280648



References

[1] BalaMurugesh, M. J. Jayasree and S. Nagaraj. Process of the International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, “Modify High Speed 32-bit Vedic Multiplier Design an Implement.”

[2] Vijay Kumar Reddy, "Modified High Speed 32-bit Vedic Multiplier Design and Implementation". 2. R. Seshadri, S. Ramakrishnan, and V. Vijayalakshmi Proceedings of the 2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Communication System (ICEVENT): "Design and implementation of 32 bit unsigned multiplier using CLAA and CSLA".

[3] Suresh Kumar Soniya International Journal of Emerging Trends Technology in Computer Science (IJETTCS),"A Review of Different Type of Multipliers and Multiplier Accumulator Unit."

[4] "Design of High Speed Low Power 32-bit Multiplier using Reversible Logic," published in the International Journal of VLSI System Design and Communication System, vol. 02, 2014, by R. Vasim Akram, Mohammad Rahmatullah Khan, and B. Raj Kumar.

[5] International Journal of Advance Research in Electrical Electronics and Instrumentation Engineering, V. Murlidharan, N. Satish Kumar, "Design and implementation of low power and high speed multiplier using quaternary carry look-ahead adder." 2020 June.

[6] S. Mohan Das, M. Mahaboob Basha, and N. Fahmina Afreen "Delay-Free, Power Efficient, Area-Delay CSLA-Based 32-Bit Array Multiplier Design and Implementation," 2nd IEEE International

Conference on Recent Trends in Electronics Information Communication Technology (RTEICT), May 19–20, 2017.

[7] Afreen Begum, B. Amala, A. Vimala, V. Vidhya Aparna, and Siva.S. Sinthura 2nd International Conference on Trends in Electronics and Informatics (ICOEI) 2018: "Imple- menation and Analysis of different 32-bit multipliers on aspects of Power, Speed, and Area".

[8] ASHLY JACOB, SERIN SHIBU, ANU THOMAS, and SWATHI SUDHAKARAN "A Comparison of Vedic Multiplier with Conventional Array and Wallace Tree Multiplier".

[9] "Low Power High Performance 4 bit Vedic-multiplier in 32nm," Mestry, S. T., Sankpal, S. V., and Golatkar, D. N., 6th International Conference for Convergence in Technology (I2CT), 2021.

[10] The work "Implementation of 32-Bit Complex Floating Point Multiplier Using Vedic Multiplier, Array Multiplier and Combined integer and floating point Multiplier (CIFM)" was presented at the IEEE International Conference for Innovation in Technology (INOCON), 2020 by Rathod, R., Ramesh, P., Zele, P. S., Y, A. K.

[11] Sachin Mittal and Sarita Singh "Optimal delay and area for multiplier and accumulator units by 32-bit sequential multiplier using VHDL design and implementation," International Journal of Engineering Trends and Technology, Volume 3, Issue 5, 2012.

[12] "Design of 16 Bit Vedic Multiplier using Carry Look Ahead Adder" Divya, K. Surya, K, Uma Narayani, R. Vidhiya, B. Sathya, G. National Conference on Emerging Trends in Electrical, Electronics, and Computer Engineering (ETEEC2018) April 2018.

[13] M. Z. Kurian, M. C. Chandrashekhar, and Sumanashree C. V. International Journal of VLSI System Design and Communication Systems, Volume 4, Issue No. 4, April 2016, "Design, Implementation and comparison of 8-Bit Vedic Multiplier Using Multiplexers and Logic Gates".

[14] Ram Ghalla, Sudha Rani D, Balasaikesava R, and Balasindhuri K "Using Vedic Mathematic Sutras to Design a VLSI Architecture for a Delay-Effective 32-Bit Multiplier," IEEE International Conference on Recent Advances in Electronics, Information and Communication Technology, May 20–21, 2016.