

Vedic Multiplier for High Speed Applications

Type: Conceptual Paper

Received: November 21, 2022

Published: November 28, 2022

Citation:

Nethra Perli., et al. "Vedic Multiplier for High Speed Applications". PriMera Scientific Engineering 1.3 (2022): 49-54.

Copyright:

© 2022 Nethra Perli., et al.
This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

JVR Sudhamsu Preetham, Nethra Perli*, D Chandrasekhar, Mathangi Akhila, N Arun Vignesh and Asisa Kumar Panigrahy

Department of ECE, Gokaraju Rangaraju Institute of Engineering & Technology, Hyderabad, Telangana, India

***Corresponding Author:** Nethra Perli, Department of ECE, Gokaraju Rangaraju Institute of Engineering & Technology, Hyderabad, Telangana, India.

Abstract

We live in a technologically advanced society. The use of diverse electronic gadgets is interwoven with even the most fundamental aspects of our daily lives. They increase and smoothen the pace of our life. The multiplier component controls the speed of most electronic systems with high-speed applications that employ the IEEE 754-2008 standard for single-precision FPUs. Several existing methods have been included to enhance the multiplier's speed of operation. They have, however, not demonstrated a substantial difference in speed, raising it by a maximum of 1.182 times.

As a result, we presented "Vedic Design," a novel algorithm with a distinctive architecture. When this was simulated in Vivado, it improved the multiplier's speed by 3.4478 times, resulting in a multiplier that is nearly 3.5 times more efficient. The gadget is better equipped to function as a result of the reduced computational path latency.

Keywords: Computational Path Delay; Latency; Vedic Multiplier; Vivado; Speed

Introduction

The world we currently reside in is driven by many gadgets, devices, and several technological advancements made by the human mind since the dawn of time. It's these developments that have equipped us humans to face any challenge thrown at us however daunting they may seem. We have found the innermost constituents of an atom which was considered indivisible. The human mind has always been curious to find ways to make life simpler, to make existing solutions to various problems even more flexible and universally applicable. The most important thing that we users look for in any device or system is the speed at which it works, as no one would want to use a device that would delay the work.

A similar problem of overcoming the speed of the device is dealt with in this paper using our proposed method. Every electronic device irrespective of its usage and purpose has a multiplier as one of its many essential components. Systems like digital signal processors and microprocessors require high speed multipliers to match their performance capabilities. Multiplier is the slowest component of the system, hence its performance influences the system's total performance.

The IEEE 754 standard outlines how computers should represent the Binary floating point numbers. Single precision and double precision forms are two types of representations available in the floating point numbers in binary format. In such applications, multiplication is one of the most significant arithmetic operations, hence a fast multiplier circuit is required. In many applications, power usage and time delay are crucial. Multipliers like Vedic multiplier, add-sub multipliers and Booth recoding are also described along with the IEEE 754-2008 FPU.

Algorithms for existing multipliers are presented using flow charts, followed by the proposed modified Vedic Design Algorithm. The paper concludes with the results and proof of efficiency for the proposed multiplier algorithm as well as potential future works.

Literature Survey

In the last few years, there are many methods that tried to implement such proposed algorithms with the sole purpose of decreasing the latency of an electronic device. One of the most renowned is mentioned below.

Novel Vedic and Shift-Add design for Single Precision IEEE 754-2008 FPU in High Speed Applications by Anshuman Mohapatra and Abhyarthana Bisoyi [2].

Anshuman Mohapatra and Abhyarthana Bisoyi had introduced 3 different algorithms in order to decrease the latency of an electronic device and increase the overall performance of the device. While all the three methods are minute changes of the existing Shift-Add and Vedic algorithms, all three of them are proven to be efficient in different categories.

Proposals

1. Modified Shift-Add Multiplier.
2. Proposed Vedic Multiplier - 1.
3. Proposed Vedic Multiplier - 2.

The existing multiplier algorithms are presented using flow charts, followed by the Vedic multiplier and shift add algorithms suggested in [2]. In section 5, the proposed Vedic design algorithm is discussed. The algorithms provided are not only fast but also efficient for area considerations. This provided a gateway for a number of applications considering the efficiency of a multiplier. During the process of calculations, there might be few errors which effect the result of the multiplier. To avoid that, we can use concurrent error detection and self-repairable adders [3]. The logic design of a computer can be made with the help of few general considerations [4].

Identification of the Problem

A multiplier should be fast in order to support any high speed applications as the execution of any process depends on the latency of the electronic device. Multiplier is usually the unit that takes up the greatest space and time. As a consequence, optimising the multiplier's area and performance is a crucial design consideration for any digital signal processor. This section describes the different types of multiplier algorithms that are widely used in DSP devices for faster applications. Booth's algorithm is presented for demonstrating its relevance with Booth's original algorithm for multiplication processes [5]. In this part, the paper will focus on the remaining traditional multipliers and also a novel multiplier algorithm.

FPU in IEEE 754 standard 2008

A 32 bit binary number is used in the process. It can be identified as a single precision representation. In the 32 bit input given to the multiplier, 23 belong to mantissa, 8 are for exponent and the remaining 1 is marked for sign bit. This paper consists of implementations of the existing multiplier algorithms. From the simulations obtained, novel techniques are obtained by making few small yet significant changes that supports in optimized design of the MAC units of the electronic device.

Booth Recoding

The concept of shifting instead of adding is the basic principle used by the booth recoding algorithm. This approach proved faster execution and smaller area consumption compared to other multipliers. It reduces the number of individual product values that must be combined for the output [12-14].

Vedic Multiplication

The process of Vedic Multiplication [6] was discovered that it had a parallel production of incomplete products that consumed less space [7]. P.Rai used Vedic mathematics to build a 32-bit floating-point multiplier for the IEEE 754 standard [8]. In 2016, a solution for reducing power consumption by 53% from capabilities and limitations was explored [9]. In DSP processors, MAC unit will be benefited to a great extent [10, 11].

Algorithms

Shift Add

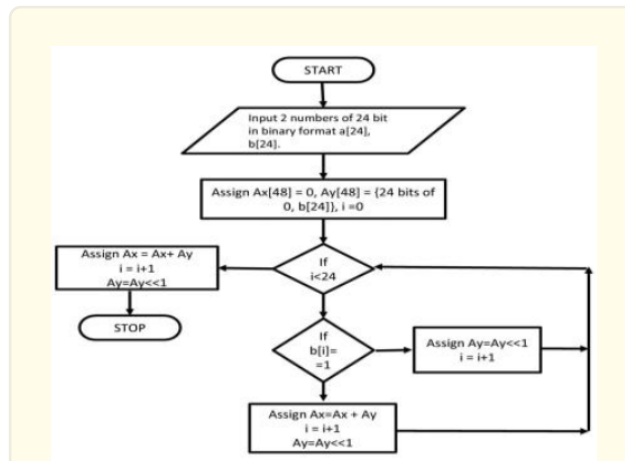


Figure 1: Existing Shift-Add algorithm flowchart.

Vedic Design

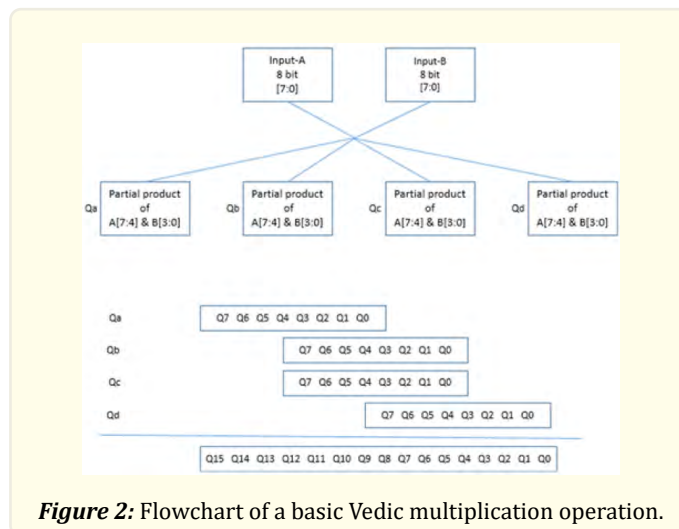
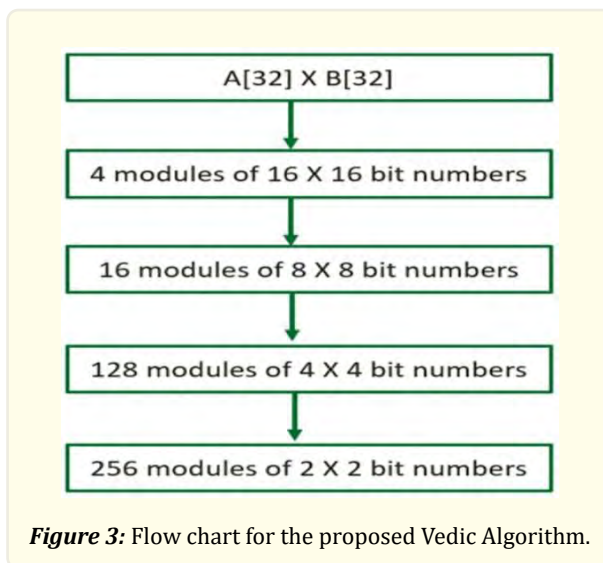


Figure 2: Flowchart of a basic Vedic multiplication operation.

Proposed Algorithm

A significant change is made: The final output of a 32X32 bit multiplication is broken down to 2X2 bit multiplication and the gate level implementation is shifted to 2X2 bit inputs. Mantissa multiplication is done with 32 bits here. As a result, we split 32-bit multiplication into its first portion, i.e., 16 bits. Decomposing the 32 X 32-bit multiplication into four numbers of 16X16 subunits is done here. Following the Vedic technique of 3-bit multiplication, this is further broken into 8X8 modules, then to 4X4, followed by gate level implementation at 2X2 bit level multiplication. The gate level implementation of the 2X2 module is used in the proposed algorithm.

The flowchart of the process of implementation for the proposed algorithm is shown in Figure 3.



Result Analysis

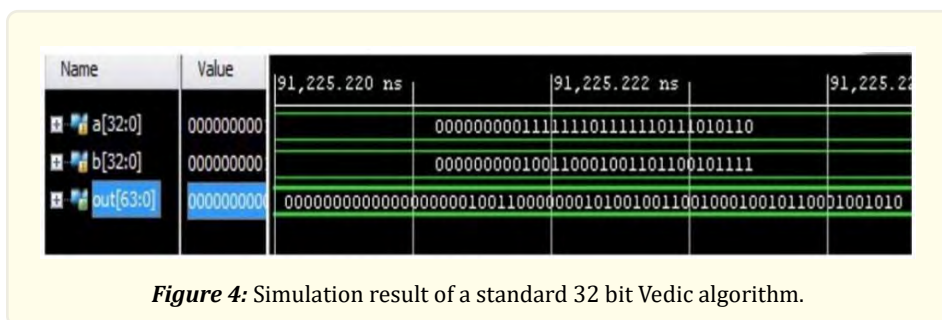
The proposed Vedic Multiplier is used to multiply the inputs in the Vivado software.

Considering the inputs,

Input-A: 111111101111110111010110.

Input-B: 100110001001101100101111.

1. Simulation of 32x32 result using conventional Vedic multiplier [7].



2. Simulation of 32x32 result using proposed Vedic multiplier.

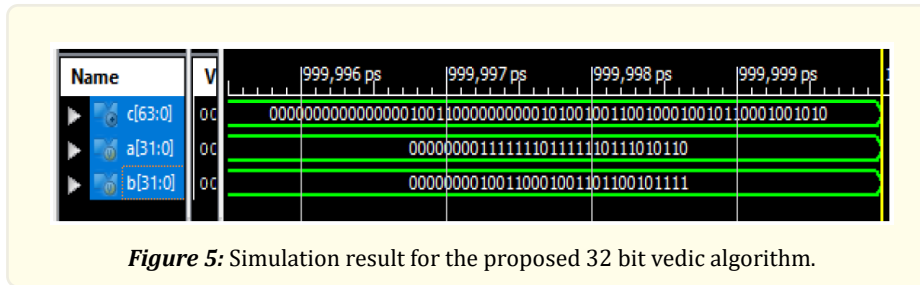


Figure 5: Simulation result for the proposed 32 bit vedic algorithm.

By comparing multiple algorithms for the same inputs and tabulating the computational path delays, the values are as shown in the table.

Available Algorithms	Maximum Expected Latency (in ns)
Existing Vedic algorithm [7]	47.10
Standard Shift and Add algorithm	43.91
Modified Vedic Multiplier	13.602

Table 1: Comparison of maximum path delays in different algorithms.

Conclusion

As shown in the above simulation results, the simulation for a 32 bit multiplier by using the conventional Vedic algorithm is using 47.10 ns of time to execute a 32 bit multiplication. This is the computational path delay. Whereas the proposed Vedic algorithm is using only 13.602 ns (mentioned in fig.6) which is approximated to 3.5 times better than the conventional one. This simulation has proven to be effective in case of time. This reduces a lot of pressure on the hardware while solving huge and complex calculations. In order to obtain an efficient arithmetic FPU, we can design a fused Add/Sub unit [15].

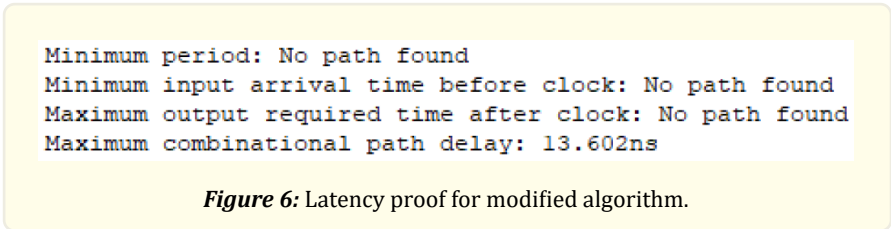


Figure 6: Latency proof for modified algorithm.

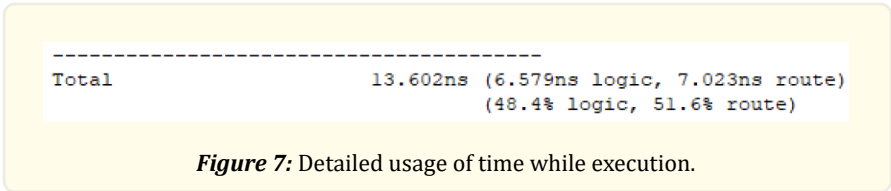


Figure 7: Detailed usage of time while execution.

We are currently using hardware co-simulation to implement the aforementioned methods in Vivado system generator. Though developing the block diagram is comparatively simpler, configuring hardware parameters such as gateways in co-simulation necessitates changes. Due to presence of multipliers in every architectural design of a DSP device, creating energy efficient design is important. Future works will include improving the multiplier algorithms to achieve even shorter propagation delays, as well as developing it in python and other developing platforms.

References

1. P Kishore., et al. "Chapter 41 A Review on Comparative Analysis of Add-Shift Multiplier and Array Multiplier Performance Parameters". Springer Science and Business Media LLC (2021).
2. Anshuman Mohapatra and Abhyarthana Bisoyi. "Design of Novel Vedic and Shift-Add multipliers for Single Precision IEEE 754-2008 Floating-point Unit Applications". High Speed Applications (2020).
3. Sarada Musala., et al. "Concurrent error detectable and self-repairable carry select adder". International Journal of Electronics (2021).
4. AD Booth and KHV Britten. General considerations in the Design of an Electronic Computer (1947).
5. AD Booth. "A signed binary multiplication technique". Quarterly Journal of Mechanics and Applied Mathematics 4.2 (1951): 236-240.
6. A Bisoyi, M Baral and MK Senapati. "Comparison of a 32-Bit Vedic Multiplier with a Conventional Binary Multiplier". IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) (2014).
7. TK Haripriya and KU Sajesh. "VHDL implementation of novel squaring circuit based on Vedic mathematics". 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (2017): 1549-1552.
8. P Rai and S Kumar. "Design of Floating-point Multiplier Using Vedic Aphorisms". International Journal of Engineering Trends and Technology 11.3 (2014): 123-126.
9. V Camus., et al. "Approximate 32-bit floating-point unit design with 53% power-area product reduction". ESSCIRC Conference 2016: 42nd European SolidState Circuits Conference (2016): 465-468.
10. DK Kahar and H Mehta. "High speed Vedic multiplier used International Vedic Conference mathematics". Intelligent Computing and Control Systems (ICICCS), Madurai (2017): 356-359.
11. AK Itawadiya., et al. "Design a DSP operations using Vedic mathematics". International Conference on Communication and Signal Processing, Melmaruvathur (2013): 897-902.
12. AS Prabhu and V Elakya. "Design of modified low power booth multiplier". 2012 International Conference on Computing, Communication and Applications (2012): 1-6.
13. K Tsoumanis., et al. "An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator". IEEE Transactions on Circuits and Systems I: Regular Papers 61.4 (2014): 1133-1143.
14. E Antelo, P Montuschi and A Nannarelli. "Improved 64-bit Radix-16 Booth Multiplier Based on Partial Product Array Height Reduction". IEEE Transactions on Circuits and Systems I: Regular Papers 64.2 (2017): 409-418.
15. H Saleh and EE Swartzlander. "A Floating point Fused Add Subtract Unit". 2008 51st Midwest Symposium on Circuits and Systems (2008).