International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

DESIGN AND VERIFICATION OF 32-BIT VEDIC MULTIPLIER

K.G.VENKATA KRISHNA[1], A.SUPRIYA[2], R.AKHIL[3], CH.PAVAN KUMAR[4], D.KARIMULLA[5]

[1]Assistant Professor, Dept of ECE, Krishna University College of Engg & Tech, Machilipatnam, Krishna-521004, AP, India
[2,3,4,5] UG Students,Dept of ECE, Krishna University College of Engg & Tech, Machilipatnam, Krishna-521004, AP, India

---

Abstract - The Vedic sutras of classical Vedic mathematics, the proposed research project describes the modified binary Vedic multiplier. It offers adjustments to the Vedic multiplier that has been prototyped. There has been an improvement in both device use and time delay with the updated binary Vedic multiplier, which is preferred. In Verilog HDL, the suggested method was created and put into practice. The Xilinx Vivado tool is used for HDL simulation and synthesis. Simulation results are also displayed. The simulation was performed for a 32-bit multiplication operation. Greater sizes can be used with this adjusted multiplication method. Comparisons are made between the results of this multiplication method and current Vedic multiplier methods.

Key Words: Vedic-multiplier, Ripple carry adder, Carry save adder, VerilogHDL

1.INTRODUCTION

Research scholar Bharati Krishna provided a more succinct explanation of Vedic mathematics, stating that it is comprised of 64 sutras and 13 sub-sutras. This makes solving mathematical equations for computations easier.Prehistoric Vedic Mathematics was made more advanced by Swami Bharati Krishna Tirthaji Maharaj, a Shankaracharya from Goverdhan Peath. It makes sense and is both straightforward and consistent. As a result of Vedic mathematics becoming more and more common, both in India and the rest of the globe, it is a highly popular study topic. Digital signal processing must carry out tasks such as frequency transformations (DFT, FFT, and DCT) and frequency domain filtering (FIR, IIR). Multiplications are an essential hardware component for these tasks. For this reason, choosing how to display the entire structure depends in large part on how the multiplier is presented. This is due to the fact that the multiplier is the framework's slowest and most laborious element. Therefore, a notable test for the framework architects is the improvement of the multiplier speed and area. The application of ancient Vedic mathematical methods can successfully overwhelm this exam.

1.1 Need of Multiplier Architecture

The data route is the central component of any microprocessor, digital signal processor (DSP), and application-specific integrated circuit (ASIC) used in data processing. When die area, power dissipation, and especially operating speed are of concern, it is frequently the most important circuit component. Arithmetic units like multipliers, adders, and comparators form the foundation of both data-path and addressing units. Lastly, the binary addition is the fundamental operation included in the majority of mathematical components. In addition to doing basic operations like incrementing and magnitude comparison based on binary addition, adders may also perform simpler operations like adding two integers. Thus, the most significant arithmetic operation is binary addition.

2. Literature Review

2.1 S. Lad and V. S. Bendre, "Design and Comparison of Multiplier using Vedic Sutras," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-5

Fast processing units are necessary for many real-time applications in modern computerized era. The basic building elements of these units are ALU and MAC, which are necessary for quick and effective execution. Digital signal processors use multipliers as its primary component.To keep things accurate. To boost the rate of execution, registers, multipliers, and adders must be changed in order to improve the performance of the ALU and MAC. The growing limits on delay call for the design of quicker multipliers to be implemented in CPUs. It is crucial to create quicker multipliers in order to increase multiplication speed.

2.2 Balpande VishwasV, Abhishek B.Pande, Meeta J.Walke, Bhavna D.Choudhari and Kiran R. Bagade. "Design and Implementation of 64 Bit Processor on FPGA." (2015).

This assignment involves utilizing Verilog HDL to model the components of a 64-bit RISC CPU and create it. The Harvard architecture serves as the basis for the CPU. The very basic instruction set used here provides insight into the sort of hardware that should be able to correctly execute the set of instructions. Beyond the basic sequential and combinational processor building blocks, such registers and adders, more intricate blocks, like ALUs and memory, have been built and modeled. In this research, the

ALU has been thoroughly structurally modeled, beginning with half adders. Finally, a semi-custom layout for ALU was created. While basic blocks like adders have been represented using a structural approach, complex blocks like memory have been modeled using a behavioral approach.

2.3 Seung Pyo Jung, Jingzhe Xu, Donghoon Lee, Ju Sung Park, Kang-joo Kim and Koon-shik Cho, "Design & verification of 64 bit RISC processor," 2008 International SoC Design Conference, Busan, 2008, pp. III-13-III-14

This article presents the design and verification process for a 64-bit RISC processor. The Harvard design of the proposed processor includes internal debug logic, a 5-stage pipeline for instruction execution, and a 24-bit address. The FPGA-based processor successfully executes the SOLA algorithm and the ADPCM vocoder. Personal digital assistants (PDAs) and portable multimedia players (PMPs) are not unique human inventions. Thus, SOC level ASICs (Application Specific Integrated Circuits) are used to create compact and low power processors. The 8051 and ARM7 processors are the most widely used SOC level ASIC processors. Millions of gates use the ARM7 on a SOC level ASIC dimensions. Furthermore, basic systems that require modest size employ the 8051 CPU. However, the basic system's bit size is increasing from 8 to 64.

3. EXISTING METHOD

The three main processes in the multiplication process are as follows: There are several types of multipliers, from simple to sophisticated.

1. Generation of partial products
2. Reduction of partial products
3. The last addition

The "add and shift" algorithm is a widely used multiplication technique. The following illustrates the multiplication method for an N-bit multiplier. the 32-bit product, which is produced by multiplying two 32-bit values. The add shift method is the foundation of the multiplier circuit. To discover the product of two binary values, a binary multiplier can be utilized in digital electronics as an electrical circuit, such as in computers. The binary multiplier multiplies the multiplicand by each bit of the multiplier starting with the least significant bit, using a carbon duplicate of the standard multiplication procedure. By using two halfadder (HA) modules, a two-bit binary multiplier may be implemented. Digital multipliers can be applied to a variety of computer arithmetic computations. Many of these methods involve calculating a series of partial products and adding up the partial products that are produced. A 2x2 binary multiplier is shown in Fig. 1.

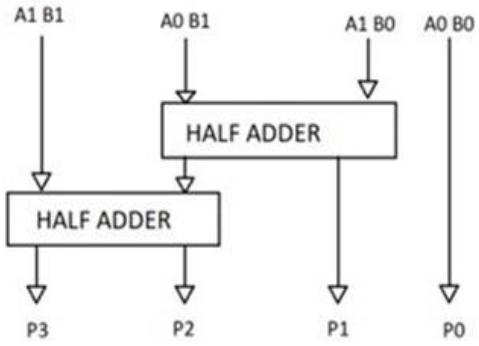

Fig -1: Two-by-two Binary Multiplier

3.1 Ripple Carry Adder (RCA):

To provide the results of an addition operation on an n-bit binary sequence, a multiplier number of full adders are placed. It continues until it reaches the final stage, using the carry output of the previous adder as the input for the subsequent Full adder stage. The four bit (RCA) Ripple Carry Adder is shown in Fig. 2.

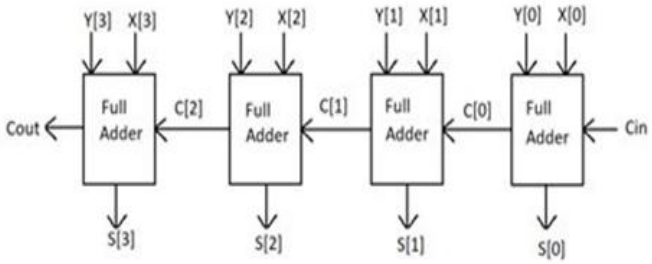

Fig -2: 4-Bit Ripple Carry Adder

Vedic mathematics is the mode that the Vedic multiplier uses. This method will result in a rise in performance while utilizing less hardware components. The Vedic multiplier uses the sutra Urdhva Tiryakbhyam, which signifies both horizontally and vertically. The resultant bears The output of these two adders is sent into a second RCA. Concatenate the bits if their sizes are not identical. The final RCA's size is cut in half, and the outputs of the parallel adder are sent into an OR gate for the 32-bit Modified Vedic multiplier.

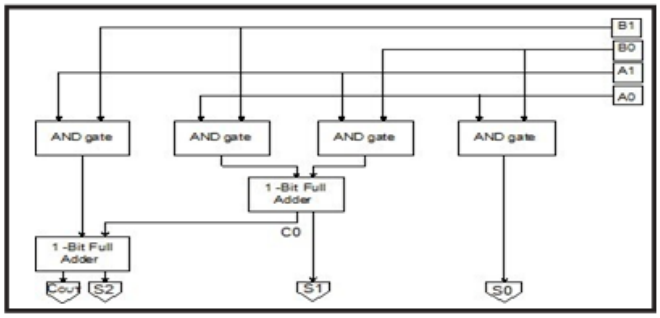

Fig -3: 2-Bit Vedic Multiplier

4. PROPOSED SYSTEM

4.1 VEDIC MULTIPLIER

Mathematical Vedas A collection of sixteen Sutras (aphorisms) and thirteen Sub-Sutras (corollaries) from the Atharva Veda is known as the Swami Bharati Krishna Tirtha. He created methods and techniques for elaborating on the concepts found in the aphorisms and their corollaries, which he called Vedic Mathematics. Vedic Mathematics generates partial products in parallel, which accelerates processing. In this study, we provide a format that allows these intermediate products to accumulate quickly. In digital signal processing, multiplication is a necessary operation for many applications. This paper presents an extremely fast Vedic multiplier that utilizes Urdhva Tiryakbhyam, a multiplication sutra from Vedic Math, to achieve efficiency in speed. The processing capability of a multiplier increases when the input and output data bus widths are increased because of its regular construction. After multiplying the numbers on each end of the line, the outcome is added to the carry from before. All of the results are added to the previous carry when several lines are processed in a single step. The remaining digits serve as the carry for the following stage, and the least significant digit of the number so produced serves as one of the outcome digits. The carry is first assumed to be zero. The 2x2, 4x4, and 8x8 bit Vedic multiplier units' hardware designs are based on "Urdhva- Tiryakbhyam" (Vertically and Crosswise).A concurrent execution strategy is used in Vedic Multiplier for the production and adding of fractional products.

4.1 Vedic Multiplier for 2×2 bit Module

The following explains the technique for two, two-bit values, A and B, where A = a1a0 and B = b1b0, as seen in the following figure. The procedure first considers the multiplication of LSBs that result in the LSB of the final product (vertical). The next higher digit of the multiplier is then multiplied by the LSB of the multiplicand, and this is followed by the addition of the result, the product of the LSB of the multiplier, and the next higher bit of the multiplicand (crosswise). The second bit of the final product is provided by the summation, and the carry is added to the part product that is obtained by multiplying the most significant bits, leading to the sum and carry terms.

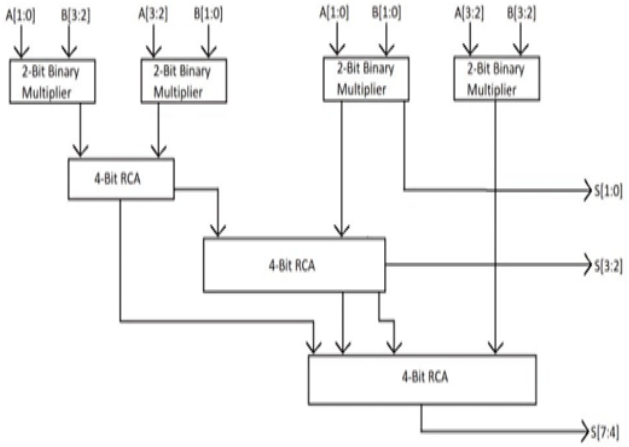

Fig -4: Traditional Multiplier for 4x4 bit module

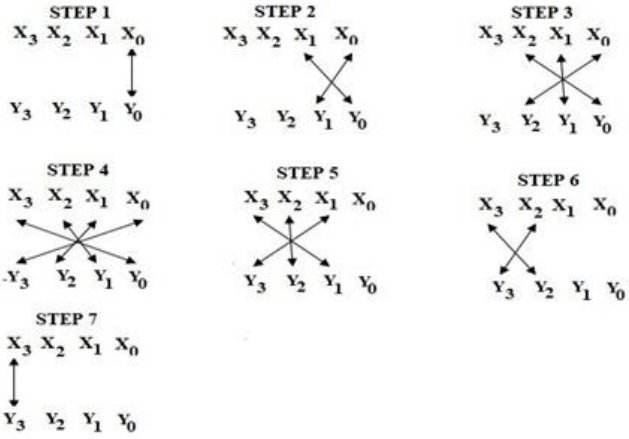

Fig -5: Algorithm for UT Sutra

4.2 CARRY SAVE ADDER (CSA)

A carry-save adder's primary characteristic is that it divides the addition of the carry—also known as the "propagate" term—and the total. To do this, it generates two partial sums (called "sum" and "carry-out") first, ignoring the carry-in from the previous step. After that, a last addition is made in order to calculate the true total and execute.

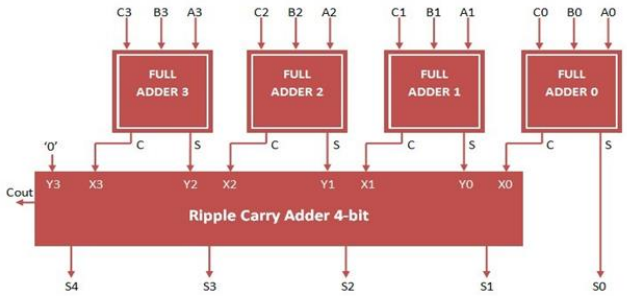

Fig -6: 4-Bit Carry save adder(CSA)

5. RESULTS

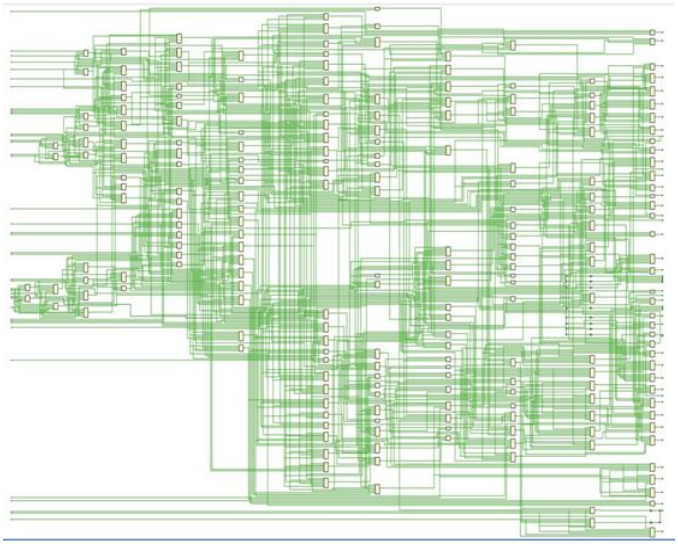

Fig -7: RTL Schematic View

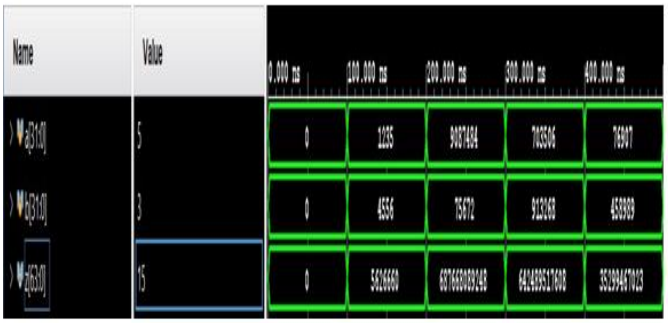

Fig -8: Simulation Output

Name	Slice LUTs (134600)	Bonded IOB (400)
modified_vedic	1912	128

Fig -9: Area

Name	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	28	91	a[0]	z[59]	21.050	6.029	15.021	∞	input port clock
Path 2	28	91	a[0]	z[57]	21.047	6.026	15.021	∞	input port clock
Path 3	28	91	a[0]	z[58]	21.047	6.026	15.021	∞	input port clock
Path 4	28	91	a[0]	z[60]	21.047	6.026	15.021	∞	input port clock
Path 5	28	91	a[0]	z[61]	21.031	6.023	15.008	∞	input port clock
Path 6	28	91	a[0]	z[62]	21.031	6.023	15.008	∞	input port clock
Path 7	28	91	a[0]	z[63]	21.020	6.023	14.997	∞	input port clock
Path 8	27	91	a[0]	z[55]	20.568	5.918	14.650	∞	input port clock
Path 9	27	91	a[0]	z[56]	20.568	5.918	14.650	∞	input port clock
Path 10	26	91	a[0]	z[51]	20.091	5.816	14.275	∞	input port clock

Fig -10: Time delay

6. CONCLUSION

An organized approach to binary multiplier circuits, derived from Vedic mathematics, has been provided in this study. The suggested approach is more efficient than the current ones in terms of time delay. By using the suggested approach, elongation for a larger bit size can be accomplished. In the proposed modified Vedic multiplier, adders with varying architectures can be employed in the Carry Save Adder (CSA) design. One of the several methods for increasing and accelerating the multiplication is to employ changed architecture. There is a disadvantage to this technique: hemorrhage occurs. Using Verilog as a modeling tool, this study ultimately provides a 32-bit Vedic multiplier that utilizes the carry save adder (CSA) logic. By using a multiplier instead of traditional multiplier units, the findings show that the multiplier implementation increases overall performance.

REFERENCES

[1] S. Akhter, "VHDL implementation of fast NxN multiplier based on Vedic mathematics," in Proc. 18th European Conference on Circuit Theory and Design, 2007, pp. 472-475

[2] S. Nagaraj, Dr.G.M. Sreerama Reddy and Dr.S. Aruna Mastani; A Comparative Study on Different Multipliers-Survey Journal of Advanced Research in Dynamical and Control Systems14739-7522018Institute of Advanced Scientific Research.

[3] M.Pushpa, S. Nagaraj, Design and Analysis of 8-bit Array, Carry Save Array, Braun, Wallace Tree and Vedic Multipliers, IEEE Sponsored International Conference On New Trends In Engineering & Technology(ICNTET 2018).

BIOGRAPHIES

K.G.VENKATA KRISHNA,
Assistant Professor,
Krishna University College of Engineering and Technology,
Krishna University,
Machilipatnam.

AKULA SUPRIYA, Student of Department of Electronics and Communication Engineering, Krishna University, Machilipatnam.

REBBA AKHIL, student of Department of Electronics and Communication Engineering, Krishna University, Machilipatnam.

CHITTIPROLU PAVAN KUMAR, Student of Department of Electronics and Communication Engineering,Krishna University, Machilipatnam.

DUDEKULA KARIMULLA, Student of Department of Electronics and Communication Engineering,Krishna University, Machilipatnam.