A Novel Realization of Multiplier Design for Signal Processing

M. Dharani[1], M. Bharathi[2], M. Tharun Reddy[3], P. Venkata Sahithya Lakshmi[4], Kanna Samskruthi[5], Chengamma Chitteti[6]

[1]Associate Professor, Department of ECE, School of Engineering and Technology, Mohan Babu University, Tirupati, Chittoor District, Andhra Pradesh, India. E-mail: dharani.m@vidyaniketh an.edu, dharani405@gmail.com

[2]Assistant Professor, Department of ECE, School of Engineering and Technology, Mohan Babu University, Tirupati, Chittoor District, Andhra Pradesh, India. E-mail: bharathi.m@vidyaniketh an.edu, bharathi891@gmail.com

[3]B.Tech Student, Department of ECE, School of Engineering and Technology, Mohan Babu University, Tirupati, Chittoor District, Andhra Pradesh, India. E-mail: tharunmallela143@gma il.com

[4]B.Tech Student, Department of ECE, School of Engineering and Technology, Mohan Babu University, Tirupati, Chittoor District, Andhra Pradesh, India. E-mail: Sahithya3525@gmail.com

[5]B.Tech Student, Department of ECE, School of Engineering and Technology, Mohan Babu University, Tirupati, Chittoor District, Andhra Pradesh, India. E-mail: samskruthikanna@gmail.com

[6]Assistant Professor, Department of IT, School of Computing, Mohan Babu University, Tirupati, Chittoor District, Andhra Pradesh, India. E-mail: Chengamma.c@vidyanikethan.edu

Abstract

The multiplication operation is a need for all Digital Signal Processing (DSP) applications. It includes addition and shift operations. Many people have developed a wide range of concepts for computation systems with different design objectives in terms of power, area, and speed. Digital signal processor (DSP), Fast Fourier Transform (FFT), and Multiply and Accumulate Unit are examples of applications based on its regular structure (MAC). This research recommends two solutions, specifically for DSP systems, to boost speed and cut power usage. Four 2x2 LUT multipliers are utilized in proposed -1 Multiplier to show it on a 4x4 multiplier. The two multipliers that are being suggested are designed with XILINX VIVADO software and are written in Verilog HDL. Additionally, the suggested multiplier's latency, area, and power usage are compared to those of traditional multipliers. Additionally, the simulation findings suggest that the work utilizes less power than standard approaches, achieving only 3.751W in proposed-1 and 2.824W in proposed-2 as opposed to the traditional multiplier 4.804W, and has demonstrated less latency.

Keywords: Multiply and Accumulate Unit (MAC), Digital Signal Processing (DSP), Fast Fourier Transform (FFT), Verilog HDL, LUT (Look Up Table).

DOI: 10.47750/pnr.2022.13.04.133

Introduction & Literature Survey

Multipliers are used in a variety of applications, including DSPs [3], automotive embedded systems, and sensor networks. These are the primary component of the majority of digital circuits like ALU and MAC [1,9] generally employed in convolution-related tasks[4]. Figure 1 shows a block diagram of a 4-bit multiplier that multiplies two binary integers. A binary integer has four bits, and two binary integers multiplied together yield an 8-bit value. Small size, quick speed, and low power consumption are requirements that any digital circuit must fulfil in order to operate successfully. It is crucial to design an effective multiplier in order to boost speed and reduce circuit area consumption. It is crucial to design an effective multiplier in order to boost speed and save circuit area, especially for biomedical applications [6,7,8,] which machine learning algorithms[10]may analyse. An effective multiplier of this kind is provided in this paper. Four 2 bit LUT-based multipliers are used in the multiplier's architecture to boost speed while reducing power consumption.

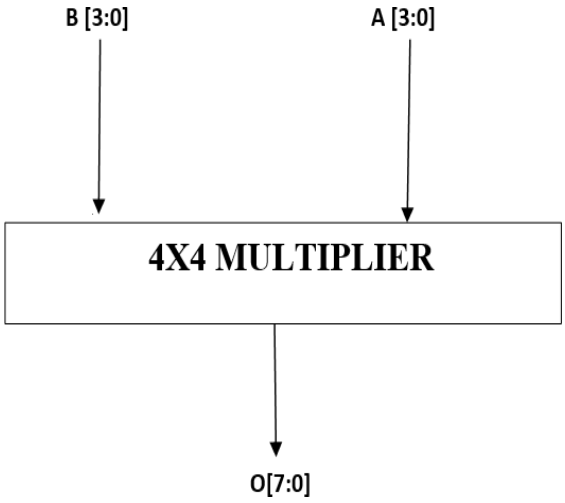

Figure 1. Block diagram of 4 Bit Multiplier

The authors of the article [5] proposed a design for a 4x4 multiplier 1 rganized the Dadda algorithm and a 10T full adder. In comparison to other multipliers, an array multiplier requires more power and has a longer propagation delay. Wallace multipliers require a huge space. It is not recommended to organize the Vedic multiplier with complex values. The author of the study suggested the Dadda multiplier as a solution to the problems with the aforementioned current multipliers. Tanner EDA and 45nm technology are used for the simulations.

The primary objective of the authors' work [2] is to acceleratethe organized2n Multiply and Accumulate unit to speed up processing. The development of MAC cores can benefit froma persuasive technique that improves the size, latency, andpower trade-off [2], [3], which has benefits for the design of distributed arithmetic-based digital signal processors (DA). The paper [16] presents the Urdhva Tiryagbhyam (vertical and across) Vedic technique of multiplication, which is distinct from common multiplication.

The most effective algorithm, regardless of the type of number being multiplied or the magnitude of the numbers, is Urdhva-Tiryagbhyam.

He author of the paper [12] defines a modified form of the binary vedic multiplier using vedic sutras from classical vedic mathematics. Both device organized2n and time delay have increased with the revised binary vedic multiplier.

To translate the input into a booth equivalent, the study's authors [13] provided a booth multiplier with a booth decoder. As a result, switching operations will occur less frequently, which will reduce the design's power [11] consumption. The adder's related rows or columns should be inactive when the input coefficient is 0. The component's switching behavior iscontrolled by the input bit coefficient.

The signed-unsigned Modified Booth Encoding is a high- speed, unique multiplier unit for signed and unsigned numbers that is described by the authors of the book [14]. (SUMBE). Half of the partial products are produced simultaneously by the Booth Encoder circuit that is being tested. By extending the operands' sign bits and producing an additional partial product, the SUMBE multiplier is made.

The Wallace multiplier was compared to the Dadda multiplier and the traditional array multiplier in the article [15] in terms of delay. The proposed sixteen-bit Wallace multiplier is constructed by the Carry Select Adder (CSLA) and Binary to Excess -1 Converter (BEC) adders. The results show that the CSLA-based Wallace multiplier has a smaller delay than the BEC-based Wallace multiplier.

The following sections of the essay are organized as follows. Section II explicitly illustrates the construction of a 2-bit LUT multiplier and a 4-bit multiplier made up of four 2-bit LUT multipliers. In Section-III, the simulation results for the suggested 4-Bit multiplier are presented. Section IV has the conclusion.

PROPOSED DESIGN

LUT-based Approach

The suggested 4 bit multiplier is built using four 2 bit multipliers, and adder elements are added at the very end to carry out results. The suggested 4-bit multiplier in question is shown in Figure -3. Reduced latency is produced by the LUT-based design approach for the 2 Bit multipliers.

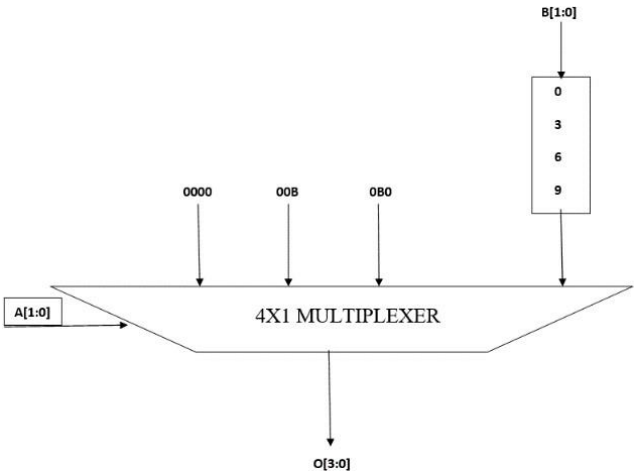

Figure 2. Design of 2 Bit LUT Based Multiplier

The 2 bit LUT-based design, shown in Figure 2, shows how the output depends on the input. The multiplexer selects the first input, which is 0000 all zeros, if the input A is 00. Or else the multiplexer's second input, 000B, is chosen if the input A is 1, and its third input, 00B0, is picked if the input A is 10. Finally, if the value of input A is 11, the value of input B will determine the output.

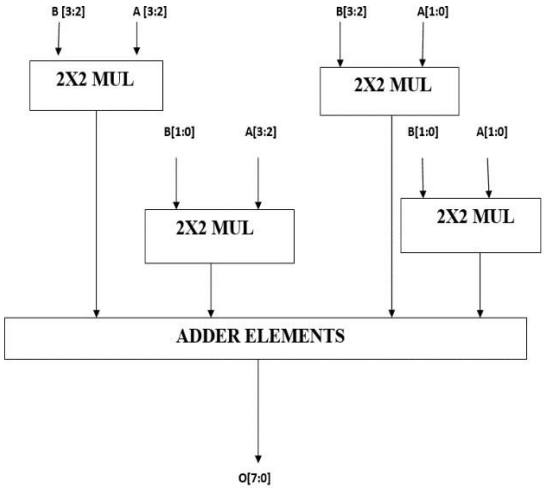

Figure 3. Design of 4 Bit Multiplier using four 2 bit LUT based Multipliers

Figure 3 depicts the creation of a 4 bit multiplier with four2 bit LUT-based multipliers. A four bit multiplier has two inputs, each of which has four bits. These four bits are divided

in half, and the four halves are multiplied collectively using two-bit lut-based multipliers. The output of these four multipliers was created by sending their results to full addersand half adders, respectively.

Second Proposed Method

Temp1 is the product of multiplying B by A[0] (Temp1 = B x A[0]). Temp2 is the product of multiplying B by A[1], hence Temp2 = B x A[1]. Temp3 is the product of multiplying B by A[2]; therefore, Temp3 = B x A[2]. Temp1 = B x A[3], and Temp4 is the product of multiplying B and A[3].

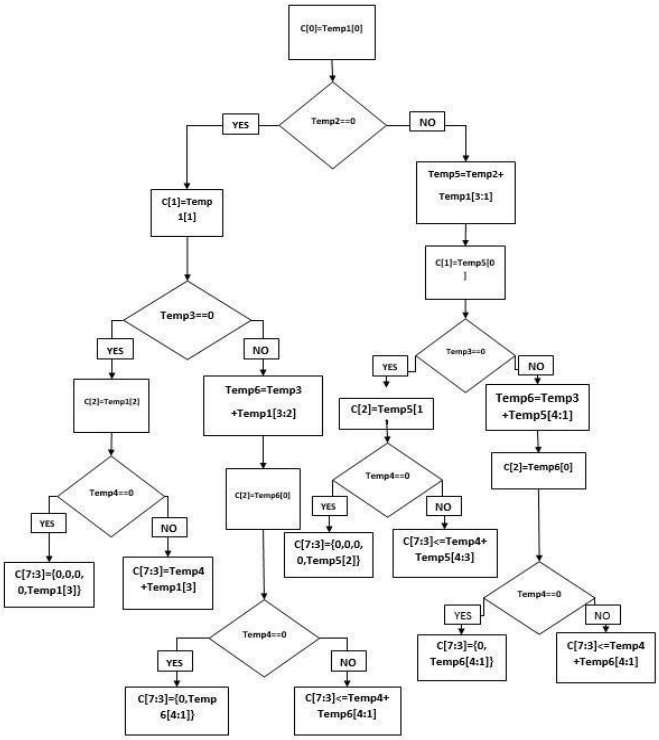

Figure 4. Flow Chart of Method-2 4Bit Multiplier

The computations will be done based on the partial products. Output LSB C[0] is initially given the value temp1[0]. Stage 1 calls for the addition to be made between temperatures 1 and 2. If the outcome of temp2 is zero, we proceed directly to the next level of data propagation by assigning temp1[1] to C[1]. If the outcome of temp2 is not zero, temp2 and temp1 will be added, and the result will be saved in the temporary variable temp5, as illustrated in Figure 4. The output C[2] is designated as the LSB of the temp6 variable.

If the temp3 variable's result in stage 2 is 0, the result is propagated to stage 3 by assigning the LSB to output C[2] without being added. Otherwise, based on stage 1 operation, addition is to be done between variable temp3 and temp1 or temp5, and the result is saved in variable temp6 by assigningthe LSB of temp6 to output C[2]. In step 3, if the temp4 variable's result is zero, the temp6 result will be immediatelyassigned to the remaining bits of the output, or

The suggested 2-multiplier is intended to minimise computations, i.e. additions based on incomplete products. We have two inputs, A and B, each of which has four bits, ina 4x4 multiplier. To obtain the partial products, A and B are multiplied initially. Temporary variables temp1, temp2, temp3, and temp4 are used to hold these incomplete items. C[7:3], otherwise addition between temp4 and temp6 will be done, and the result will be placed in output C[7:3]. We have N-1 Stages for a N Bit multipler.

Simulation of Results of Proposed Designs

Four 2 bit LUT-based multipliers were combined to formthe 4 bit multiplier shown in figure 3. Two 4 bit inputs make up a 4 bit multiplier. These four bits are divided into two equal parts, and the two-bit lut-based multipliers are used to multiply the four parts by one another. Full adders and half adders received the output of the four multipliers in order to calculate the results.

Figure 5. Simulation result of 2 X 2 LUT Based Multiplier

The simulation output for the 2x2 LUT multiplier is shown in Figure 5. For instance, if inputs a[1:0] and b[1:0] are both 11, input b will choose the fourth value, which is 9, and input a will choose the fourth input of the multiplexer, resulting in an output of 9, as shown in the simulation result. Results of the 4x4 multiplier simulation are shown in Figure 6.

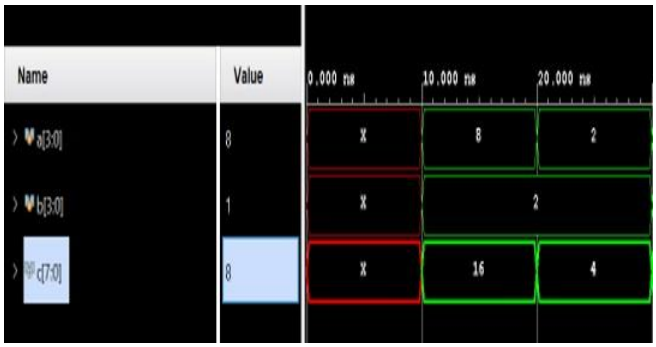

Figure 6. Simulation result of 4 X 4 Propose-1 LUT based Multiplier

The simulation output of the second proposed 4X4 multiplier is shown in Figure 7. If input A is 0101 and input

B is 1111, the partial products temp4, temp3, temp2, and temp1 will be 0000, 1111,0000, and 1111, respectively. The determined by temp2 results; if temp2 results are zero, temp[1] will be assigned directly to output C[1], if temp3 results are non-zero, the addition between temp3 and temp1[3:2] will be performed, and the LSB of the result will be assigned to C[2], and if temp4 results are zero, the previous addition results will be directly assigned to output C[7:3]. Now, the output C will be 01001011—75 in decimal—which is the result.

Figure 7. Simulation result of 4 X 4 Propose-2 Multiplier

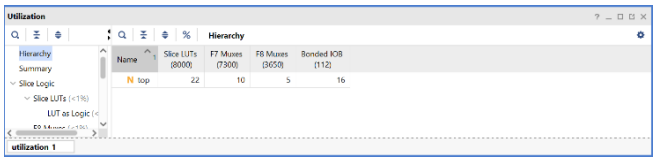

Figure 8. Device Utilization of proposed-1 LUT based multiplier

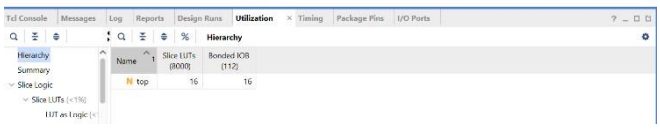

Figure 9. Device utilization of conventional multiplier

Figure 8 displays the number of slices and LUTs that the device selected for the specified multiplier is 22 slices used. Figure 9 illustrates the 16 slices and LUTs used by the device selected for the traditional multiplier.

Figure 10. Device Utilization of proposed-2 Multiplier

Figure 10 depicts the 43 slices and LUTs used by the device selected for the traditional multiplier.

LSB of output C is given the temp1's LSB. The addition between temp1 and temp2 will be

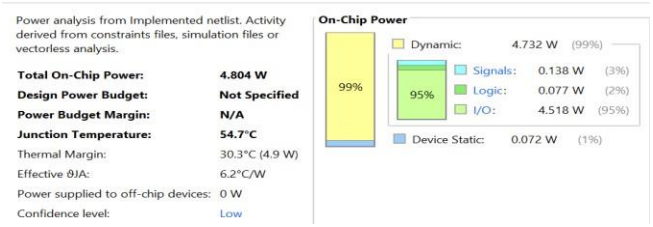

Figure 11. Power Consumption of Conventional Multiplier

Figure -11 displays the energy consumed by the typical multiplier device, which is 4.804W, with dynamic power being 4,732W and static power being 0.072W.

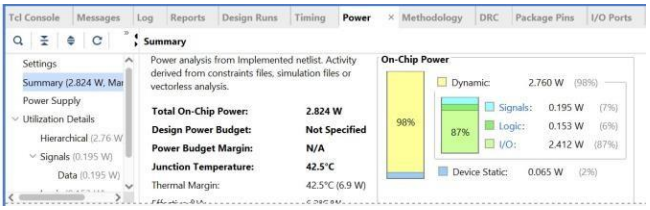

Figure 12. Power Consumption of proposed-2 Multiplier

Figure 12 illustrates the energy consumption of the selected device for the proposed-2 multiplier, which is 2.824W, of which 2.760W is dynamic power and 0.065W is static power.

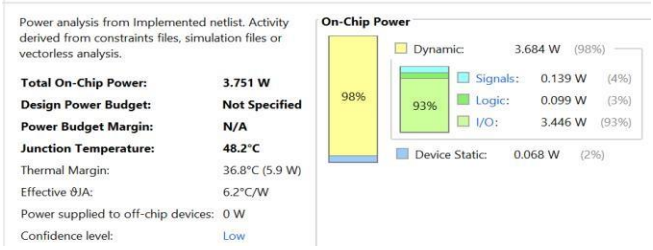

Figure 13. Power Consumption of proposed-1 LUT based Multiplier

Figure 13 displays the energy consumption of the selected device for the suggested multiplier, which is 3.751W, with 3.648W of dynamic power and 0.068W of static power.

Table 1. Comparison between conventional multiplier, proposed-1 and proposed-2 multiplier

	Conventional Multiplier	**Proposed- 1 LUT based multiplier**	**Proposed- 2 Multiplier**
No. Of Slices	16	22	43
Power Consumption	4.804W	3.751W	2.824W
Set Up Delay(ns)	8.023	7.241	8.742
Hold Delay(ns)	2.227	2.036	0.448

CONCLUSION

Two innovative designs for a 4 bit multiplier that are demonstrated in this study are implemented using the Xilinx Vivado tool. The proposed-1 Multiplier is built using four 2 Bit LUT-based multipliers, and the proposed-2 Multipler is intended to simplify computations based on partial products and hence reduce power consumption. The given 4-Bit multiplier is synthesised, constructed, and simulated using the Xilinx vivado tool. Simulated data is stored in Verilog HDL. We compared the results shown in Table 1 such as Device Utilization, Area, and Power Consumption and came to the conclusion that the proposed multiplier offers reduced latency and low power consumption. It will perform effectively if the proposed 4 bit multiplier is used. The proposed-1 Multiplier can be utilised in low power applications whereas the proposed-2 Multiplier can be used in applications with less latency.

REFERENCES

Bharathi, M., & Shirur, Y.J.M. (2021). Vlsi Implementation Of Multiply And Accumulate Unit Using Offset Binary Coding Distributed Arithmetic. Turkish Journal of Computer and Mathematics Education, 12(11), 4739-4749.

P.L. Lahari, M. Bharathi and Y. Jyothi Shirur, "An Efficient Truncated MAC using Approximate Adders for Image and Video Processing Applications," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), 2020, pp. 1039-1043, doi: 10.1109/ICOEI48184.2020.9142930.

Bharathi, M., Shirur, Y.J.M., & Lahari, P.L. (2020, July). Performance evaluation of Distributed Arithmetic based MAC Structures for DSP Applications. In 2020 7th International Conference on Smart Structures and Systems (ICSSS) (pp. 1-5). IEEE.

Vardhan, A.S., Bharathi, M., & Padmaja, N. (2019). Convolution Techniques using Modified Booth Multiplication. I-Manager's Journal on Digital Signal Processing, 7(2), 33.

Rani, D.L., Bharathi, M., & Padmaja, N. (2019, November). Performance Comparison of FFT, DCT, DWT and DDDWT-OFDM in Rayleigh Channel. In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 392-394). IEEE.

Ashreetha. B, Anil Kumar N, D Venkataramireddy, D. Badrinarayanan, V. Vijayaraghavan, M. Sumithra. (2022). Role of Embedded Systems in Industrial Section-By considering the Automotive Industry as an Example. Journal of Optoelectronics Laser, 41(3), 100–103.

Ashreetha, B., Devi, M. R., Kumar, U. P., Mani, M. K., Sahu, D. N., & Reddy, P. C. S. (2022). Soft optimization techniques for automatic liver cancer detection in abdominal liver images. International Journal of Health Sciences, 6(S1), 10820–10831.

K.A. Swamy, Z.C. Alex, P. Ramachandran, T.L. Mathew, C. Sushma and N. Padmaja, & quot; Real-time Implementation of Delay Efficient DCT Based Hearing Aid Algorithm Using TMS320C5505 DSP Processor, & quot; 2021 Innovations in Power and Advanced DSP Processor, & quot; 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), 2021, pp. 1-8, doi: 10.1109/i-PACT52855.2021.9696632

Lahari, P. L., Bharathi, M., &Shirur, Y. J. M. (2020, July). High Speed Floating Point Multiply Accumulate Unit using Offset Binary Coding. In 2020 7th International Conference on Smart Structures and Systems (ICSSS) (pp. 1-5). IEEE.

Preetha, M., Anil Kumar, N., Elavarasi, K. et al. A Hybrid Clustering Approach Based Q-Leach in TDMA to Optimize QOS- Parameters. Wireless Pers Commun., 123, 1169–1200 (2022). https://doi.org/10.1007/s11277-021-09175-8

G. Challa Ram, D. Sudha rani, Y. Rama Lakshmanna, K. Bala Sindhuri (2016). Area Efficient Modified Vedic Multiplier. International Conference on Circuit, Power and Computing Technologies {ICCPCT].

M. Bala Murugesh, S. Nagaraj, G. Vijay Kumar Reddy, J. Jayasree. Modified High Speed 32-bit Vedic Multiplier Design and Implementation. Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020) IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4.

A.S. Prabhu, V. Elakya. Design of Modified Low Power Booth Multiplier. IEEE Conference.

Ravindra P Rajput, M. N Shanmukha Swamy. High speed Modified Booth Encoder multiplier for signed and unsigned numbers. 2012 14th International Conference on Modelling and Simulation.

G. Challa Ram, D. Sudha Rani, R. Balasaikesava, K. BalaSindhuri. Design of Delay Efficient Modified 16 bit Wallace Multiplier. IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016, India.