VLSI implementation of Wallace Tree Multiplier using Ladner-Fischer Adder

Kommalapati Salomi Monica[1]*　　**Dereddy Anuradha[2]**　　**Syed Haroon Rasheed[2]**
Barnala Shereesha[2]

[1]Department of ECE, Marri Laxman Reddy Institute of Technology and Management, Hyderabad, India
[2]Department of ECE, SVR College of Engineering and Technology, Nandyal, India
* Corresponding author's Email: monicasalomi2014@gmail.com

Abstract: Nowadays, most of the application depends on arithmetic designs such as an adder, multiplier, divider, etc. Among that, multipliers are very essential for designing industrial applications such as Finite Impulse Response, Fast Fourier Transform, Discrete cosine transform, etc. In the conventional methods, different kind of multipliers such as array multiplier, booth multiplier, bough Wooley multiplier, etc. are used. These existing multipliers are occupied more area to operate. In this study, Wallace Tree Multiplier (WTM) is implemented to overcome this problem. Two kinds of multipliers have designed in this research work for comparison. At first, existing WTM is designed with normal full adders and half adders. Next, proposed WTM is designed using Ladner Fischer Adder (LFA) to improve the hardware utilization and reduce the power consumption. Field Programmable Gate Array (FPGA) performances such as slice Look Up Table (LUT), Slice Register, Bonded Input-Output Bios (IOB) and power consumption are evaluated. The proposed WTM-LFA architecture occupied 374 slice LUT, 193 slice register, 59 bonded IOB, and 26.31W power. These FPGA performances are improved compared to conventional multipliers such asModified Retiming Serial Multiplier (MRSM), Digit Based Montgomery Multiplier (DBMM), and Fast Parallel Decimal Multiplier (FPDM).

Keywords: Array multiplier, Bough wooley multiplier, Field programmable gate array, Ladner fischer adder, Wallace tree multiplier.

1 Introduction

In recent years, multiplication is one of the major operations for any kind of circuit design. Normally, multiplication contains partial product and their sum and serial or parallel process operation [1, 2]. The multiplication is used in the arithmetic process which differentiates the signal processing and scientific application [3]. Normally, the perfect multiplier contains good physical character and high-speed unit. The normal multiplication method is called add and shift method [4]. In the process of a parallel multiplier, the partial products need to be added which determines the performance of multiplier. Additionally, the Booth algorithm helps to reduce the partial products [5].

The existing multipliers are high-speed Baugh Wooley [6], Baugh Wooley with unsigned [7], approximate multiplier with carry predictor [8], energy-efficient approximate multiplier [9], modified retiming serial multiplier [10], fast multiplier [11], Montgomery multiplier [12], array multiplier [13], etc. These multipliers require more power and area for performing the multiplication. With the help of the Wallace tree method, the Wallace multiplier has been designed that operates at high speed and consumes less power [14, 15]. The major point of the proposed method is explained below,

- In this paper, the WTM- LFA method is designed for improving FPGA performances. LFA is one of the essential circuits to reduce hardware utilization.
- Efficient WTM is designed with the help of LFA which helps to perform the multiplication operation with less delay.
- The LFA has less number of logical elements to perform the arithmetic operation. Due to the

usage of gate-level logic, the adder works very speed and occupied less memory usage.

This research paper is composed as follows. Section 2 presents a broad survey of recent papers in various multipliers. In section 3, an effective proposed system is developed for multiplication. In section 4, quantitative and comparative analysis of proposed and existing systems are presented. The conclusion is made in section 5.

2   Literature review

H. Saadat, H. Bokhari and S. Parameswaran [16] proposed biased multipliers for approximate integer multiplication. This method was developed by coupling a unique error reduction mechanism with an optimization integer. The floating point multiplier lies on the Pareto front in the design space area. This architecture has been designed in TSMC 45nm standard library. This proposed architecture required 686um^2 area to operate the multiplier design. The peak error increased while detecting the multiplication output.

D. Bhattacharjee, A. Siemon, E. Linn and A. Chattopadhyay [17] presented switch-based crossbar array booth multiplier. In this paper, the radix based algorithm was used to perform the multiplication process which helps to improve the speed of the operation. Complimentary Resistive Switch (CRS) logic operation helps to move the value to one register to another register. This architecture works based on ReRAM adder which causes more temperature (300K). Booth multiplier consumed more voltage (4.2V) and more cycle duration (50ns) to write the data in output terminal.

E. Pouraliakbar and M. Mosleh [18] proposed an efficient design for reversible Wallace unsigned multiplier. In this paper, two 4x4 reversible unsigned multipliers were used to design WTM. Toffoli gate (TG), Feynman Gate (FG) and reversible gates were used to design the full adder circuits which help to design the WTM. In this method, two kinds of stages were utilized to reduce the delay value. It is too difficult to design reversible gates, because it should satisfy certain conditions. After design the reversible gates, it is too difficult to give random number as an input.

R. De Rose, P. Romero and M. Lanuzza [19] presented double-precision Dual Mode Logic (DML) carry-save multiplier. This DML method worked in a mixed operation mode by using high precision operation. The DML operation achieved better energy consumption, low area and low power. Carry save adder plays a vital role in the multiplier. But,

more complex logical block was used to design the DML multiplier. Due to this complex module design, the error identification and rectification was too difficult.

M. Vestias and H. Neto [20] proposed fast parallel Decimal multipliers which improve the area of previous multipliers. Based on BCD/excess-6 process, the multiplication operation was performed which has 5221 recoding multiplier digits. The overall area was reduced by 20% compared to existing works. A normal digital adder has been used to perform the multiplication, which causes more delay (6.4ns). Due to the usage of BCD to excess-6 converter, more hardware (1268 LUT) was occupied to design the parallel decimal multiplier.

3   WTM-LFA methodology

In this research, two inputs are performed both multiplication operation such as conventional WTM and WTM-LFA. These two inputs are represented as 8-bit which is simulated in Verilog using Xilinx Vivado 2014.4 version. This paper proved that the proposed method consumes less power and used a less number of LUTs and Registers than the existing method. This section discusses the existing method, i.e., conventional Wallace Tree Multiplier and Wallace tree multiplier with Ladner-Fischer parallel prefix adder with an example. The block diagram of the proposed method is shown in Fig. 1.

3.1 Structure of conventional WTM

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers, invented by Australian Computer Scientist Chris Wallace in 1964.
The Wallace tree has three steps:
- Multiply (that is – AND) each piece of one of the contentions, by each piece of the other, yielding {\display style n^ {2}} results. Contingent upon the position of the increased bits, the wires convey diverse loads, for instance, wire of bit conveying consequence of {\display style a_ {4} b_ {3}} is 128.
- Reduce the number of incomplete items to two by layers of the full and half adders.
- Group the wires in two numbers, and include them with a regular adder.

In an ordinary WTM, partial products are created first. At that point, these are aggregated in various stages. The methodology repeats until the last stage contains just two columns.

A 4x4 bit customary WTM is created and its stream outline for structuring an ordinary WTM is
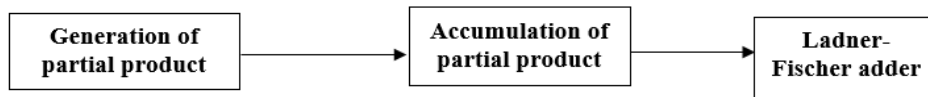
24
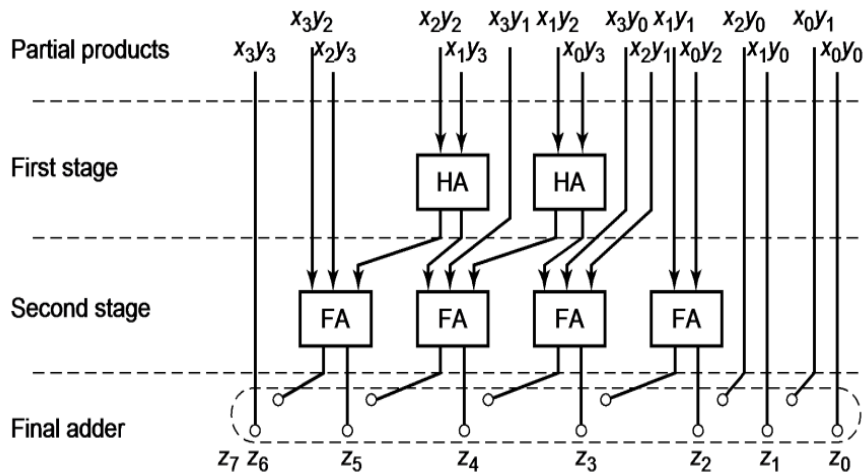

Figure. 1 Block diagram of proposed method

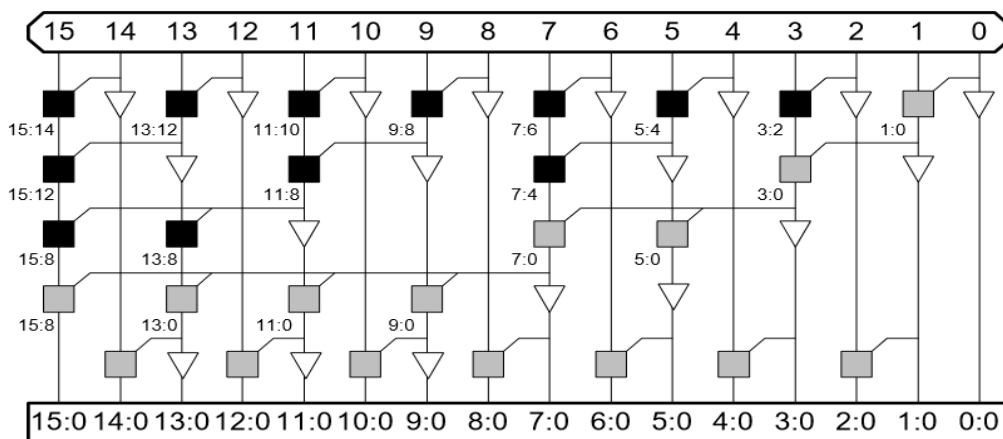Figure. 2 Conventional wallace tree multiplier design

Figure. 3 Design of 16-bit ladner fischer adder

shown in Fig. 2. The design of conventional WTM is done in three steps:

- Partial product generation
- Accumulation of partial product
- Final addition

3.2 Ladner fischer adder

In 1980, Fischer and Richard Ladner exhibited a parallel calculation for registering prefix sums efficiently in LFA. Fischer and Richard Ladner presented a parallel algorithm for computing prefix sums efficiently. They showed how to construct a circuit that computes the prefix sums in the circuit, each node performs an addition of two numbers. Their developed model helps to pick a trade-off between the circuit depth and the number of hubs. A 16-bit Ladner-Fischer adder has been shown as an example in Fig. 3. This architecture realizes how to process the aggregate bits if the addends input is one bit. This architecture is presently prepared the combinational circuit for the prefix issue. This adder utilizes just a single building square, in particular, it is a gate. The expenses should be accepted and the postponement of a gate is steady. Two circuits are essential in LFA; one with the ideal cost and the second with ideal delay.

3.2.1. Example of addition

The normal process is shown in Fig. 4 that started with the state diagram of the serial adder, attached a
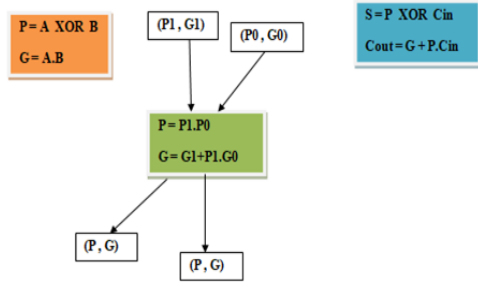
P = A XOR B
G = A.B

(P1, G1)
(P0, G0)

S = P XOR Cin
Cout = G + P.Cin

P = P1.P0
G = G1+P1.G0

(P, G)
(P, G)

function δi to each input symbol σi, and computed the composition of the function. This Fig. 4 is the basic information to design the N bit of LFA.

Besides, this leads to a fast and cheap adder design. The example of Ladner-Fischer adder is shown in Fig. 5 which takes two inputs and generate the sum and carry in the most efficient and fast way.

Figure. 4 Process of addition

LADNER-FISCHER ADDER

EXAMPLE : A = 0010(2) B = 0011(3) SUM(S) = 0101(5)

(A3=0, B3=0)
(A2=0, B2=0)
(A1=1, B1=1)
(A0=0, B0=1)

P3 = 0
G3 = 0

P2 = 0
G2 = 0

P1 = 0
G1 = 1

P0 = 1
G0 = 0

P30 = 0
G30 = 0

P10 = 0
G10 = 1

P31 = 0
G31 = 0

P20 = 0
G20 = 0

S = 0
C = 0

S = 1
C = 0

S = 0
C = 1

S = 1
C = 0
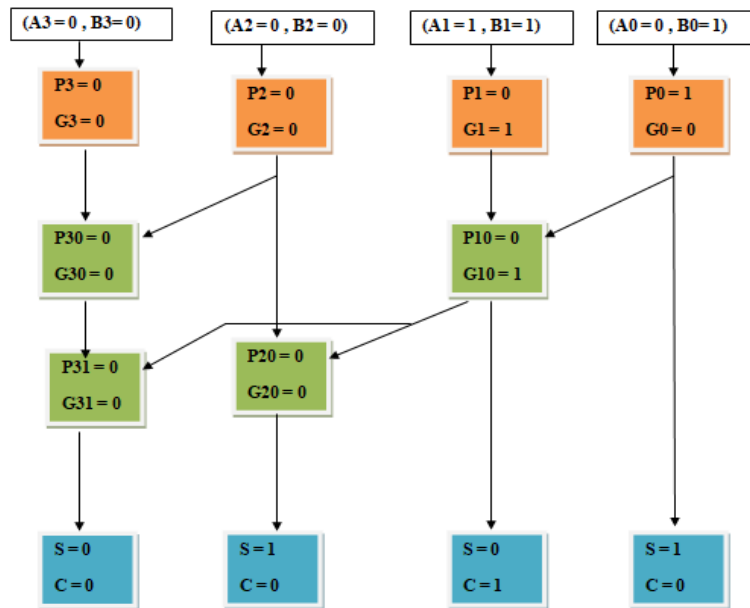
Figure. 5 Mathematical example of ladner-fischer adder

Partial products

x_3y_2
x_3y_3 x_2y_3
x_2y_2 x_3y_1 x_1y_2 x_3y_0 x_1y_1 x_2y_0 x_0y_1
x_1y_3 x_0y_3 x_2y_1 x_0y_2 x_1y_0 x_0y_0

First stage

LFA LFA

Second stage

LFA LFA LFA LFA

Final adder
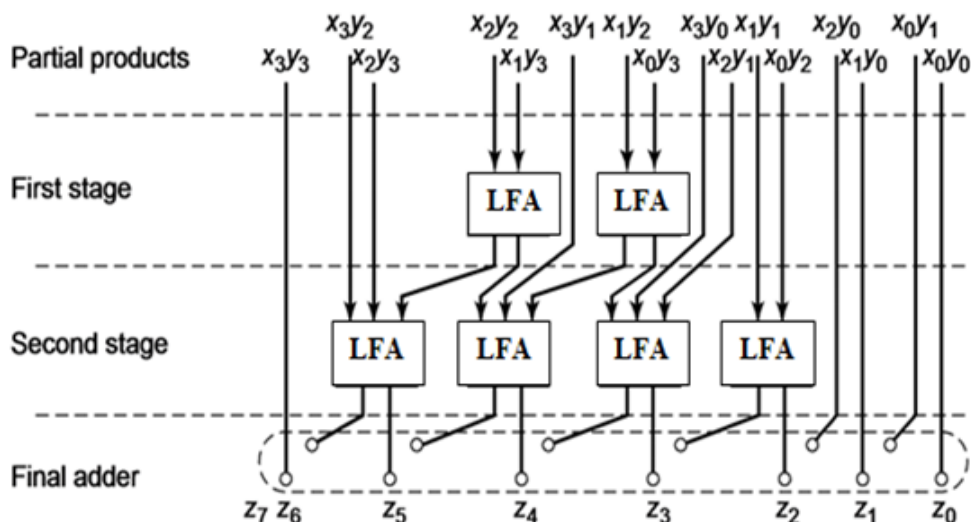
z_7 z_6 z_5 z_4 z_3 z_2 z_1 z_0

Figure. 6 Structure of the wallace tree multiplier with ladner fischer adder

3.3 Structure of the proposed WTM

In this method, WTM partial products generate the same as in the conventional method. In the proposed method, Ladner-Fischer adder is used to replace the half adder and the Full adder of the conventional WTM. The structure of the proposed technique is shown in Fig. 6. Convolution, DWT, and FFT have been used for all DSP applications.

With the recent innovative technology, the interest for the improvement of the fast multiplier is exponentially rising. Since the last few decades, a number of literatures have been presented which describes the multipliers and it is operational. Even though existing multipliers consume more area and power, it operates at high speed. A comparative attempt has been made in this paper to develop a WTM that utilizes compressors and a parallel prefix adder to increase the speed of calculation. With the help of less delay LFA helps to design the WTM which performs the efficient multiplication with less memory usage.

4. Simulation results and discussion

In this section, the simulation result of the proposed methodology is discussed and described the simulation set-up and performance measure. The performance of the proposed methodology was evaluated by FPGA performances.

4.1 Simulation setup

The proposed approach was simulated in a Personal Computer which runs of 4GB RAM with 3.30 GHz, i3 processor, and 500GB hard disk. The WTM-LFA architecture is implemented using Verilog language. Xilinx Vivado14.4 is used for getting simulation output waveform and evaluating FPGA performances like LUT, flip flop, slices, and frequency.

4.2 FPGA synthesis

This FPGA synthesis is implemented in Xilinx Vivado for Virtex 4, Virtex 6, Virtex 7, and Zync-7000. The performance such as LUT, Slice registers, Bonded IOB, and Frequency is calculated using Xilinx Vivado.

The comparison of the different FPGA performances is tabulated in Table 1. In this table, value of LUT, slice register, Bonded IOB, and power consumption are presented for different Virtex devices. The existing multiplier such as Modified Retiming Serial Multiplier (MRSM) [10], Digit Based Montgomery Multiplier (DBMM) [12], Fast parallel Decimal Multiplier (FPDM) [20] and WTM-NA are implemented in different FPGA devices such as Virtex 4, Virtex 6, Virtex 7, and Zync-7000.
In MRSM [10], Finite Impulse Response (FIR) filter based ring topology was designed which contained adders and multipliers. These arithmetic circuits are designed in gate level. To increase the MRSM multiplier performances, carry look ahead and carry save adder was used.

DBMM was designed to utilizing the DSP resources [12]. The operand and digit size are denoted as 528 bits and 48 bits. Each and every iteration, the multiplied results are accumulated and the performance was improved in Virtex 7 FPGA.
In [20], FADM was designed to calculate the FPGA performances. Decimal multiplication helps to generate the output without any degradation. So, this decimal multiplication was frequently used in all the applications. Different kind of digits multiplier was designed such as 2, 4, 8, 16, 32, and 34. This proposed FADM achieved 20% of better area performances.

Table 1. FPGA performances for different method

Design	FPGA device	Slice LUT	Slice Register	Bonded IOB	On chip power consumption (W)	Frequency (MHz)	Delay (ns)	Clock cycle
MRSM[10]	Virtex 4	144	118	60	35.14	566.572	2.602	24
DBMM[12]	Virtex 7	403	226	75	30.17	454.20	4.80	91
FPDM[20]	Virtex 6	395	254	65	28.41	440.36	4.2	65
WTM-LFA	Virtex 4	124	94	54	30.24	464.20	3.2	55
WTM-LFA	Virtex 7	307	193	59	26.31	498.25	3.6	51
WTM-LFA	Virtex 6	362	221	51	22.63	488.21	3.4	44
Existing WTM	Zynq-7000	85	16	33	13.313	578.25	3.6	32
WTM-LFA	Zynq-7000	27	15	49	10.502	610.21	3.1	28

Above mentioned conventional multipliers occupied more area utilization and consumed more power. All the conventional multipliers are implemented in Virtex 4, Virtex 7, and Virtex 6 respectively. The proposed (WTM-LFA) multiplier also implemented in different Virtex devices. Moreover, the proposed multiplier is also implemented in Zynq-7000 FPGA. This table shows that WTM-LFA has achieved less number of LUT, slice registers, and power consumption. Moreover, frequency, delay and click cycle have also improved in WTM-LFA compared to conventional methods.

The comparison graph of LUT, slice register, Bonded IOB, and power consumption are shown in Fig. 7, Fig. 8, Fig. 9 and Fig. 10. These all the images are plotted for Virtex 4 FPGA devices. In this graph, X-axis represented as different multipliers and Y-axis represented as FPGA performances. X axis results are compared with LFA based WTM. From this graph, it's clear that all the FPGA performances improved in WTM-LFA. Due to the usage of less logical element present in the LFA, the performances have improved in LFA based WTM compared to normal FA based WTM
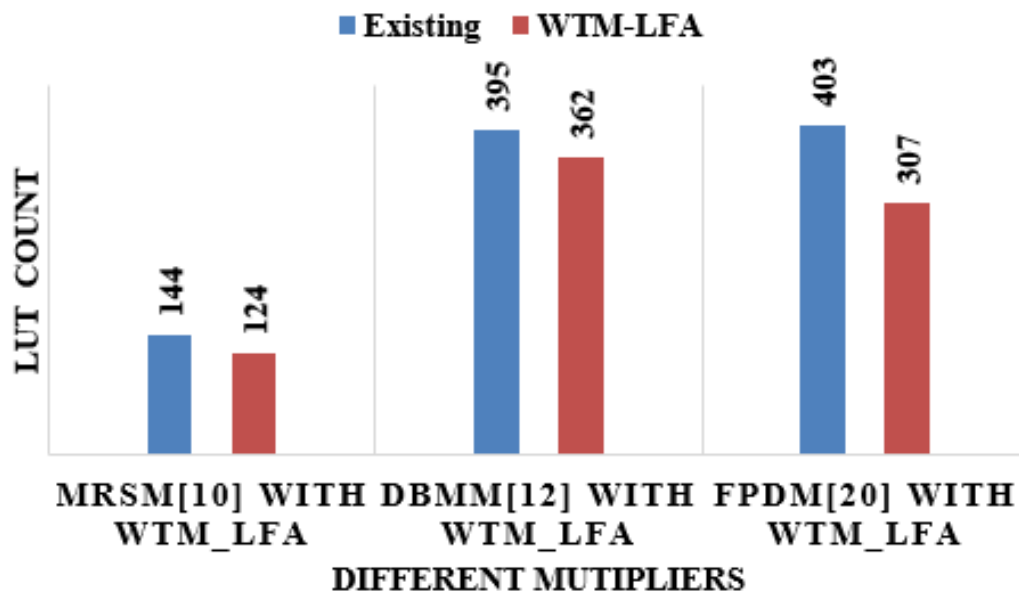

Figure. 7 Comparison of different virtex devices LUTs performances for different multipliers
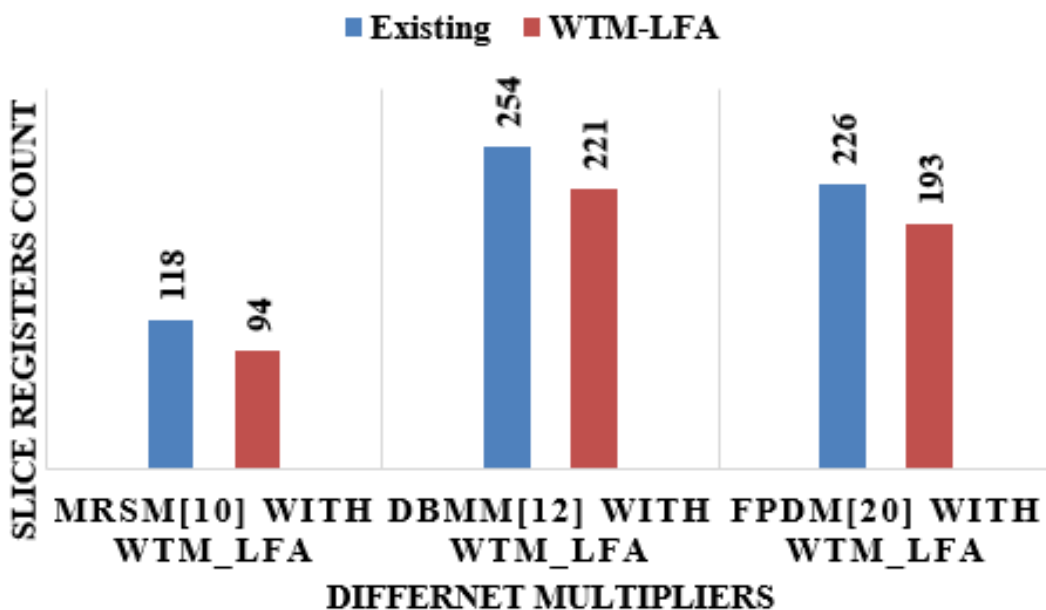

Figure. 8 Comparison of different virtex devices slice registers performances for different multipliers
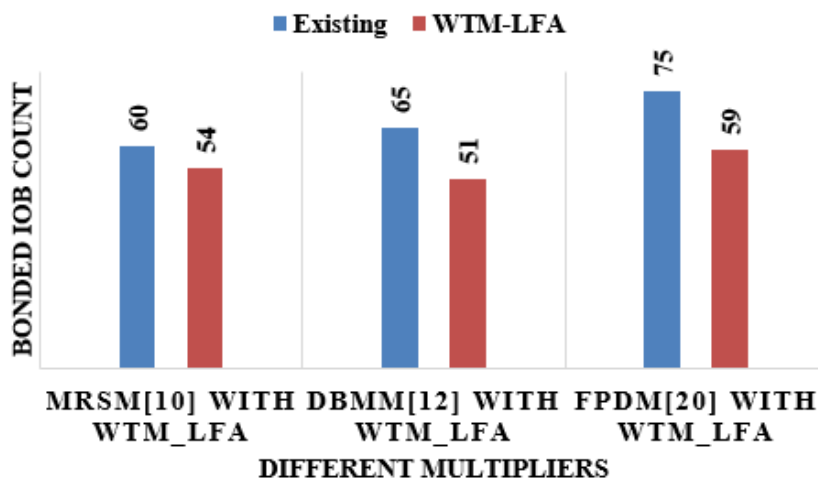
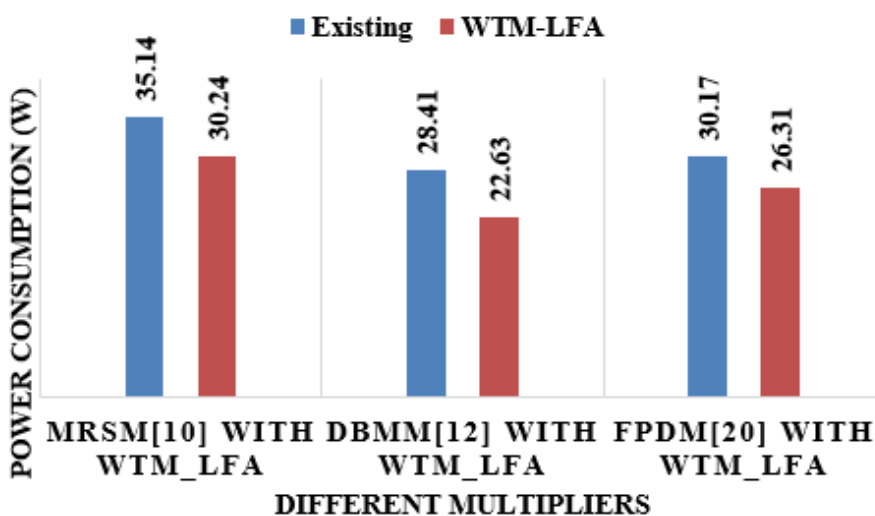Figure. 9 Comparison of different virtex devices bonded IOB performances for different multipliers

Figure. 10 Comparison of different virtex devices power consumption performances for different multipliers
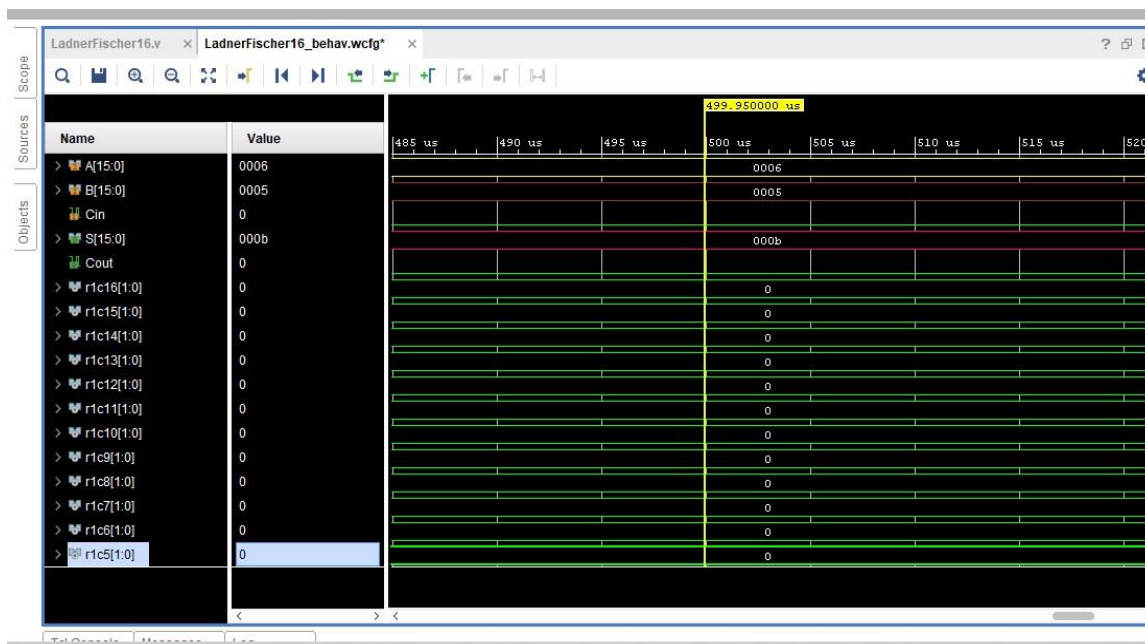

Figure. 11 LFA waveform

29

The LFA waveform result is presented in Fig. 11. Here, both inputs are represented as 16-bit such as A [15:0], and B [15:0], i.e. A = 0110(6) B = 0101(5) C_{in} =0(0) S=1011(b). The inputs performed LFA operation which produced the sum output. The simulation run time is 600us.

The multiplier output waveform is shown in Fig. 12. The inputs and outputs are represented as 8-bit to perform the multiplication. Sometimes 16-bits output also represents based on the input values. Here, X = 0011(3), Y = 0010(2), and PRODUCT = 0110(6).
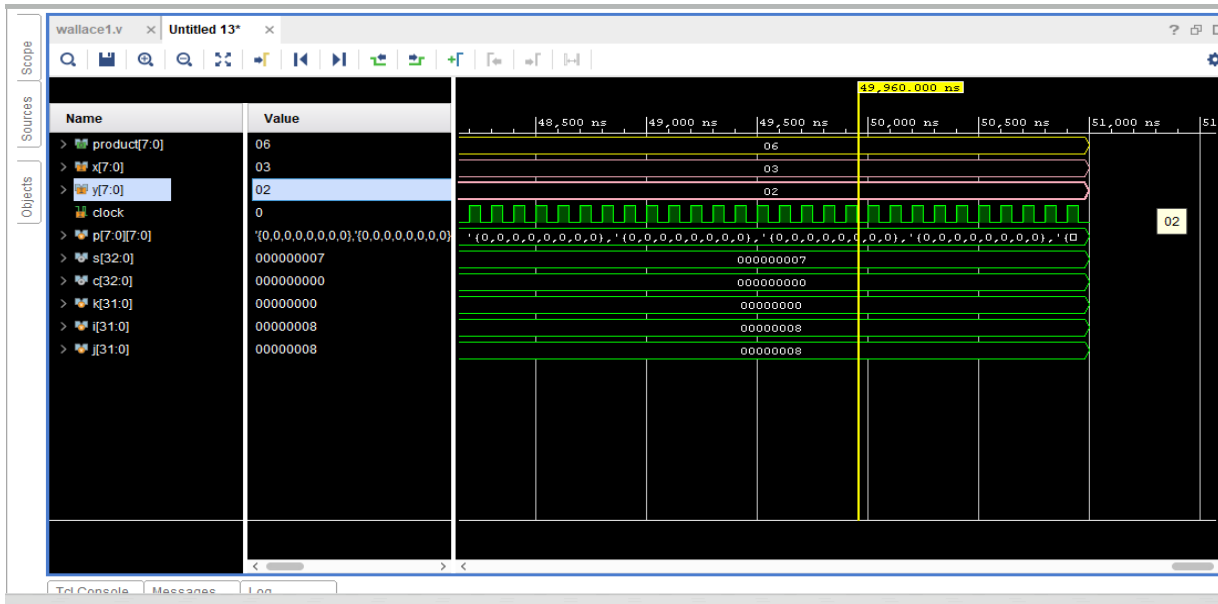

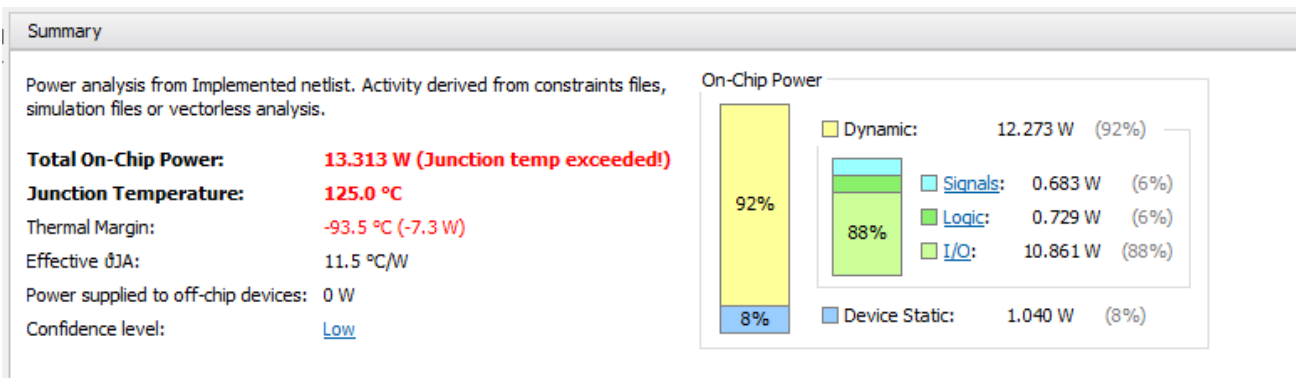Figure. 12 Wallace tree multiplier waveform
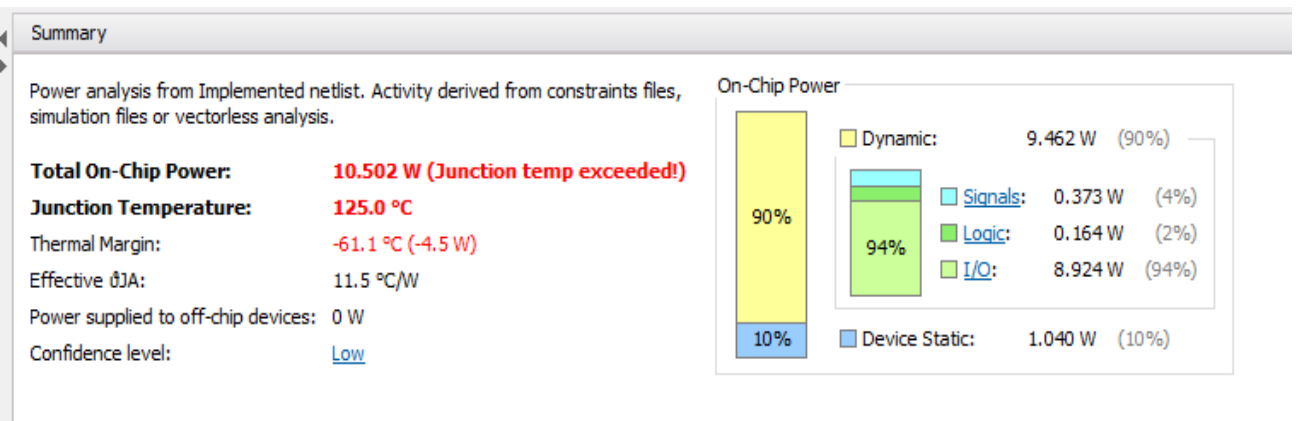

Figure. 13 Power for WTM with existing FA

Figure. 14 Power for the WTM with proposed LFA

In the above Figs. 13 and 14 show the total on-chip power of the existing and Proposed WTM and the Junction Temperature of the Proposed WTM with Ladner-Fischer adder. The total on-chip power of the existing WTM and the proposed multiplier is 13.313w and 10.502w respectively.

5 Conclusion

This paper explained the importance of the optimal multiplier and its operations. The Wallace tree multiplier with conventional adder is considered as an existing method which occupied more area. So, WTM with LFA has been designed to improve FPGA performance. The operation of LFA has been explained clearly which helps for efficient multiplication with more operating speed. In this work, FPGA performances such as slice LUT, Slice Register, Bonded IOB, and power consumption have been evaluated for different FPGA devices such as Virtex 4, Virtex 6, Virtex 7, and Zynq-7000. The proposed WTM-LFA method occupied 27 slice LUT, 15 slice register, 49 bonded IOB, and 10.502W power in Zynq-7000 FPGA. In Virtex 7, 374 Slice LUT, 193 slice register, 59 bonded IOB have been occupied in WTM-LFA. Compared to conventional methods, the proposed architecture occupied less FPGA performances in all the FPGA devices. In the future, different kind of applications will be designed by using WTM-LFA to evaluate the FPGA performances.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, have been done by 1st author. The resources, data curation have been done by 3rd author. The writing-original draft preparation, writing-review and editing, visualization have been done by 4th author. The supervision, and project administration, have been done by 2nd author.

References

[1] M. R. Dhanya, "Design and implementation of Wallace tree multiplier using higher order compressors", *International Journal of VLSI System Design and Communication Systems*, Vol. 4, No. 6, pp. 0442-0448, 2016.

[2] V. B. Biradar, P. G. Vishwas, C. S. Chetan, and B. S. Premananda, "Design and performance analysis of modified unsigned braun and signed Baugh-Wooley multiplier", In*: Proc. of IEEE International Conf. on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT),* pp. 1-6, 2017.

[3] D. K. Shruti and G. N. Zade, "Design of baugh wooley multiplier using Verilog HDL", *IOSR Journal of Engineering,* Vol. 5, No. 10, pp. 25-29, 2015.

[4] P. S. Bhupender and R. Kumar, "Design and implementation 8 bit Wallace tree multiplier", *International Journal of Advanced Research in Electrical, Electronics, and Instrumentation Engineering,* Vol. 5, No. 4, pp. 2307-2312, 2016,

[5] P. S. Aswale, M. P. Mahajan, M. V. Nikumbh, and O. S. Vaidya, "Implementation of Baugh-Wooely Multiplier and Modified Baugh Wooely Multiplier Using Cadence (Encounter) RTL", *International Journal of Science, Engineering and Technology Research,* Vol. 4, No. 2, pp. 293-298, 2015.

[6] J. Antony and J. Pathak, "Design and implementation of high speed Baugh Wooley and modified booth multiplier using cadence RTL", *International Journal of Research in Engineering and Technology,* pp. 2319-1163, 2014.

[7] P. Mohanty and R. Ranjan, "An Efficient Baugh-Wooley Architecture for both Signed & Unsigned Multiplication", *International Journal of Computer Science and Engineering Technology,* Vol. 3, No. 4, pp. 94-99, 2012.

[8] A. Sunny, B. K. Mathew, and P. B. Dhanusha, "*Area Efficient High Speed Approximate Multiplier with Carry Predictor*", Procedia Technology, Vol. 24, pp. 1170-1177, 2016.

[9] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications", *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 23, No. 6, pp. 1180-1184, 2015.

[10] B. Rashidi, "High performance and low-power finite impulse response filter based on ring topology with modified retiming serial multiplier on FPGA", *IET Signal Processing*, Vol. 7, No. 8, pp. 743-753, 2013.

[11] A. Kakacak, A. E. Guzel, O. Cihangir, S. Gören, and H. F. Ugurdag, "Fast multiplier generator for FPGAs with LUT based partial product generation and column/row compression", *Integration, the VLSI Journal*, Vol. 57, pp. 147-157, 2017.

[12] E. Özcan and S. S. Erdem, "A fast digit based Montgomery multiplier designed for FPGAs

with DSP resources", *Microprocessors and Microsystems*, Vol. 62, pp. 12-19, 2018.

[13] N. Ravi, A. Satish, T. J. Prasad, and T. S. Rao, "A new design for array multiplier with trade off in power and area", *arXiv Preprint arXiv:1111.7258*, 2011.

[14] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power-and area-efficient approximate wallace tree multiplier for error-resilient systems", In*: Proc. of 15ᵗʰInternational Symposium on Quality Electronic Design (ISQED),* pp. 263-269, 2014.

[15] A. C. Swathi, T. Yuvraj, J. Praveen, and A. Raghavendra Rao, "A Proposed Wallace Tree Multiplier Using Full Adder and Half Adder", *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering,* Vol. 4, No. 5, pp. 472-474, 2016.

[16] H. Saadat, H. Bokhari, and S. Parameswaran, "Minimally Biased Multipliers for Approximate Integer and Floating-Point Multiplication", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 37, No. 11, pp. 2623-2635, 2018.

[17] D. Bhattacharjee, A. Siemon, E. Linn, and A. Chattopadhyay, "Efficient complementary resistive switch-based crossbar array Booth multiplier", *Microelectronics Journal*, Vol. 64, pp. 78-85, 2017.

[18] E. Pour AliAkbar and M. Mosleh, "An Efficient Design for Reversible Wallace Unsigned Multiplier", *Theoretical Computer Science*, Vol. 773, pp. 43-52, 2018.

[19] R. De Rose, P. Romero, and M. Lanuzza, "Double-precision Dual Mode Logic carry-save multiplier", *Integration*, Vol. 64, pp. 71-77, 2019.

[20] M. Véstias and H. Neto, "Improving the area of fast parallel decimal multipliers", *Microprocessors and Microsystems*, Vol. 61, pp. 96-107, 2018.