FPGA Implementation of Logarithmic Multiplier

P. Anusha, G. Kalpana, T. Vigneswaran

Abstract: ***logarithmic multiplier is the vital procedure mainly for DSP, image processing and 3-D graphic applications. Log multiplier converts the multiplication into addition; hence it will reduce the number of computation steps to speed up the multiplication. In multiplication process, the reduction of partial products contributes most to the overall delay, power and area. Adder Compressors are employed to reduce the latency of this step. Analysis is done by coding the designs in HDL and synthesized with Xilinx ISE 14.7 using Virtex6 or spartan3 series of FPGA. Optimized architectures are synthesized using Encounter RTL Compiler Tool in Cadence and obtained the reports on power and area. The results indicate the better speed high performance and overall efficiency of logarithmic multiplication***

Key Words: LNS (logarithmic number systems), Arithmetic circuit, multiplication, LUT, Mitchell.

I. INTRODUCTION

From past few decades the multiplier is the vital hardware unit in most the digital processing systems such as in microprocessors and processors [8]. With the advance in technology, the literature has found with aim of lower power consumption and occupying the less silicon area in most of the portable devices. It is difficult to compromise the all constraints like area, power and speed. In general area and speed are the two conflict factors because area will increase if the device has speed performance. In this paper, the trade of between the two factors is realized. The architecture of logarithmic multiplier is employed with the compressor adders to achieve the best synthesizable results also enhances the speed of the device. The traditional logarithmic multiplier uses the Mitchell algorithm that has high performance and error in multiplication. Later Mitchell modified the algorithm with simple addition and shifting operations achieves error free logarithm multiplication. For computing the partial products in the multiplier ripple carry adder was used in order to obtain the propagation of the carry. But by employing the compressors reduces the latency and increases the speed performance. The remaining paper is arranged as follows: section 2, provides the outline of the logarithmic multiplication. Section 3, Deals with detailed view of compressors and its architectures. Section 4, Give the experimental results and discussion and section 5. Concludes the research work.

Revised Manuscript Received on 8 February 2019.

Anusha. P, Department of Electronics Engineering, VIT Chennai, Tamil Nadu India

Kalpana. G, Department of Computer Science, FSH, SRM Institute of Science and Technology, Kattankulathur (Tamil Nadu), India.

Vigneswaran. T, School of Electronics engineering, VIT Chennai, *Tamil Nadu India*

II. BACKGROUND REVIEW

Real time 3-D graphics are one of the most interesting applications in gaming mobile networks which demands low power [5,6] and relatively lesser complexity. Where logarithmic multiplication plays vital role which reduces the computation time enhances the speed. Mitchell algorithm is one which replaces the multiplication with addition because of it is having some loss of accuracy *V.mahalingam et.al* [4] provided a solution to improve accuracy operand decomposition is implemented. In logarithm conversion to achieve higher precision more number of approximation regions has been used. *Tso-bin-juang et al* proposed solution by portioning curve into two symmetric regions which is most useful in 3-D applications and DSP. For the designing of digital filters there is impact of partitioning of LUT *J.kouretas et al* implemented LNS based multiplier accumulator to achieve power and delay. *Bansal Y, Madhu C, Kaur P et al* and *M. Fonseca et al [1, 2]* designed the different multipliers using the compressors which reduce the number of partial product of the multiplication and to obtain the optimized synthesis results.

III. LOGARITHMIC MULTIPLIER

Logarithm number system is widely used to represent the large numbers it is the one of the modified version of the floating number system and it is IEEE 754 standard. The research has found use of various methods for the logarithm multiplication like lookup table method, Mitchell algorithm and interpolation techniques. Scientists explored their interest to achieve best conversion for logarithm and antilogarithm convertors [7] by adapting the various methods to achieve the error free operations. Logarithm multiplication [9] shows best approach to fast the multiplication process while calculating the partial products of the bits. It reduces the area overhead just by simple addition and shifting techniques. The logarithmic computation steps are as shown in the figure1. The first major step is to converting the decimal number into the binary number format [3]. Here the input A and input B has 8bits. The function of leading one detector (LOD) is to detect the leading one bit of the input a and input b. the value of the leading bit indicates the position the leading bit in the given input which helps in the further optimization and shifting process. The selection of remaining bits just after detecting the leading bits has been taken into account as a process. The output bits of LOD represented as K_a and K_b as well as the output bits of after LOD selection of remaining bits is represented as M_a and M_b . Concatenate the bits K_a M_a and K_b M_b. perform the addition of input a and input b with the conventional ripple carry adder in this the output of each carry is the succeeding input of next bit of full adder.

The carry of each stage is rippled to next stages to obtain the efficient addition. The output of the addition is the logarithmic addition of the two input values. To get the product convert the value to antilogarithmic value [10] just by appending 1 or 0 depends on the mantissa value generated by the logarithmic conversion.

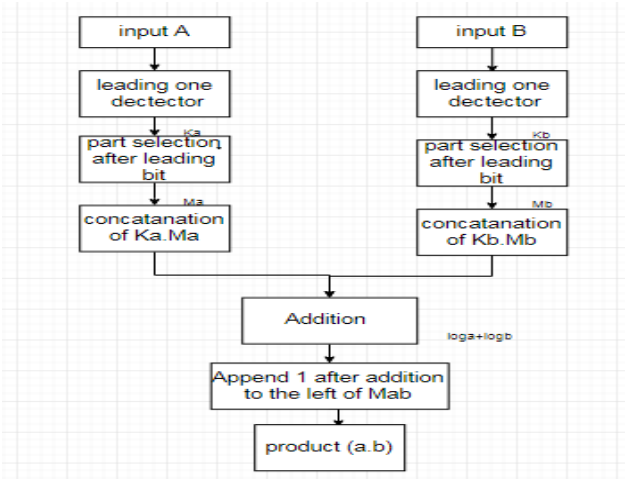


Figure 1: computation flow of logarithmic multiplication

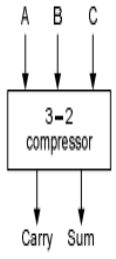
The architecture of the logarithmic multiplier is as shown in the figure 1 which converts the inputs to the binary number. Using LOD and part selection it converts to logarithm number. To obtain the overall product of number again antilogarithm is used

IV. COMPRESSOR ADDER

Compressor is the modern circuit which reduces the area and enhances the speed of the circuit by generating the minimal carry propagation than other.

Adders this makes the use of compressors adders in the fast multiplication for fast computations in the processors and many of the graphical processing units. Compressor has the principle of the parallel connected counters which has fast carry propagation and it will limit the carry propagation to a single stage thereby, reduce the overall propagation delay.

The basic full adder compressor is the 3:2 compressor adders as shown in the figure 2. It indicates that compression of three operands into two. The outputs are sum and carry. The set of equation of 3:2 compressors is given as



A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 2: 3:2 compressor adder architecture

The expressions for this adder are governed with the following equations:

Sum = A ⊕ B ⊕ C (1)

Carry = A ⊕ B * C $\overline{A \oplus B}$ * A (2)

The logic style of the 3:2 compressor is shown in the figure 3 which has two methods one with the XOR gate. Second method uses the XOR-XNOR with the combination of multiplexer which will reduces the critical path

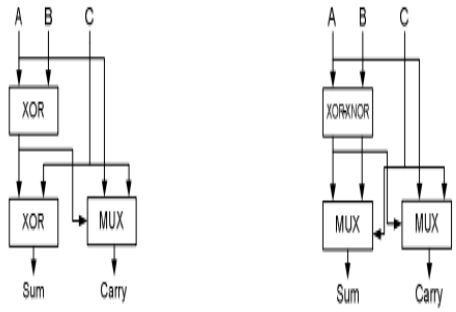


Figure 3: 3:2 compressor adder using XOR and XOR-XNOR

IV.A. 4:2 compressors

The compressor n: 2 have n operands can be reduced by two which will accumulate the partial products at each stage. It will compute the sum and carry separately that means all the bits are added parallel without relying on the previous outputs. 4:2 compressor can be computed using the basic 3:2 compressor full adder compressor with the 3 operands and other compressor with one more operand produces sum and carry as the output Figure 4 shows the 4:2 compressors which has 4 inputs carry input and sum, carry and Cout as outputs. For the logical implementation of 4:2 the two 3:2 compressor full adder has been used. Here sum output of the first adder is given to the second adder and computed parallel without waiting for carry and sum and carry computed separately. 4:2 compressor XOR-XNOR is shown in the figure 5 consists of only two XOR-XNOR operations and uses four multiplexers for producing the outputs where cin is the carry input cout is the carry output which are computed by parallel operation of full adders. The expression for the 4:2 compressor based on XOR gate is governed by the following equations:

Sum = X1⊕X2⊕X3⊕X4⊕C_{in} (3)

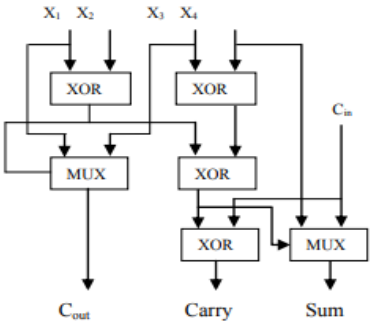


Figure 4: 4:2 Compressor based on XOR gate

C_{out} = (X1 ⊕X2).X3 + ($\overline{X1 + X2}$). X3 (4)

Carry = (X1⊕X2⊕X3⊕X4).C_{in}+ ($\overline{X1 \oplus X2 \oplus X3 \oplus X4}$). X4 (5)

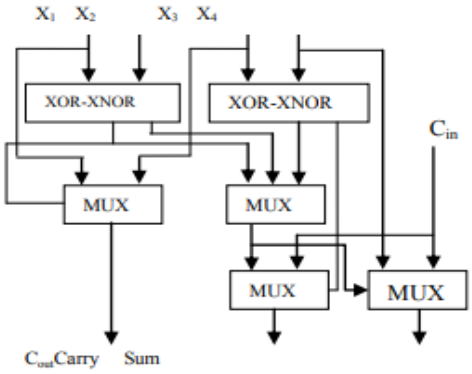


Figure 5: 4:2 compressors based on XOR-XNOR gate

The expressions for the 4:2 compressor is governed by the following equations:

$Sum = (X1 \oplus X2).(X3 \oplus X1).(X1 \oplus X2).(X3 \oplus X1).C_{in}$ (6)

$C_{out} = (X1 \oplus X2).X3 + (\overline{X1 \oplus X2}).X1$ (7)

$Carry = (X1 \oplus X2 \oplus X3 \oplus X4).C_{in} + (\overline{X1 \oplus X2 \oplus X3 \oplus X4}).X4$ (8)

IV.B. Proposed 5:2 compressors:

The proposed 5:2 compressors has x1,x2,x3,x4,x5,x6 and x7 as inputs and sum ,carry ,cout1and cout2 as the outputs. The architecture normally shows the two possibilities to implement the adder architecture. It is very important that multiplication has more space exploration for computing the partial products it better to consider the different logic styles to verify the area, power and delay of the multiplication architectures.

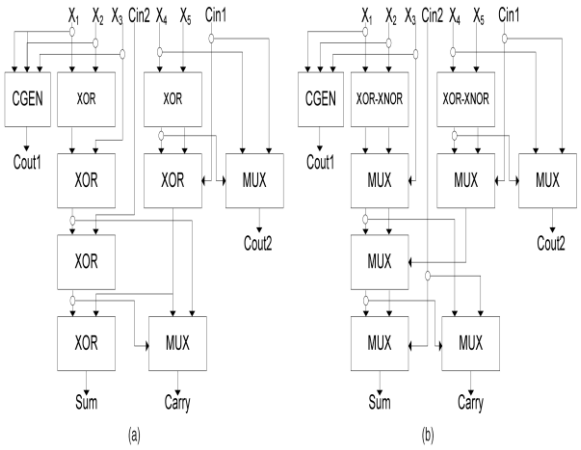


Figure: 6(a) proposed compressor architecture based on XOR 8(b) based on XOR –XNOR

Figure 6(a) and 6(b) show the proposed architectures of the compressor adder designed using the basic 3:2 compressor adder logic styles which is basic full adder architecture it having the carry generation block where it will compute the carry individually and its evaluated later. The overall operation is performed parallel so that the computed time is also reduced. In XOR based adder it used 6 XOR logics even it gives the better optimized results occupies more area. In figure 6(b) the number of XOR logic used is reduce from 6 to 2 and replace with XOR and XNOR blocks.

V. EXPERIMENT AND RESULTS

The experimental results has been obtain for the 5:2 architecture of compressor adder and compared with the traditional ripple carry adder. Xilinx 14.7 ISE is used to obtain the synthesis results. The report of area and power has been taken form cadence encounter RTL tool it is analyzed that the power consumed by the proposed architecture achieved 15% less. All the results are optimized with no violations. No negative slacks are observed in the design for the proposed architectures.

Table1: Device Utilization summary

Design	No. of slices	No of LUTs	Total Power (nW)	MCDP (ns)
Logarithmic multiplier(ripple adder)	8	14	45.8884	17.432
Logarithmic multiplier(XOR Gate)	9	14	42.356	16.231
Logarithmic multiplier(XOR-XNOR Gate)	8	13	41.3452	15.661

Table 1 shows the summary of the logarithmic multiplication for various designs. Logarithmic multiplier with XOR-XNOR gate has less number of slices when compared to conventional logarithmic multiplier with ripple adder. Also for the number of LUT occupied by the XOR-XNOR design has less area.

VI. CONCLUSION

This paper presented the various compressors architecture by using XOR based and XOR-XNOR based logarithmic multiplication. The analysis shows that using XOR-XNOR based approach utilizes the maximum combinational delay path is 2.2 % less when compared to the conventional approach. From the experimental results it is analyzed that if area is the constraint then logarithmic multiplier with XOR-XNOR based approach produce efficient results. Similarly for low power approach logarithmic multiplier XOR-XNOR provides best result of 4.54% when compared to the traditional logarithmic multiplier.

REFERENCES

1. Bansal Y, Madhu C and Kaur P. (2014) High speed Vedic multiplier designs-A review on IEEE Recent Advances in Engineering and Computational Sciences (RAECS), (pp. 1-6).
2. M. Fonseca (2011) “Design of Pipelined Butterflies from Radix-2 FFT with Decimation in Time Algorithm using Efficient Adder Compressors,” in Circuits and Systems (LASCAS), IEEE Second Latin American Symposium on, feb. 2011, pp. 1-4
3. John N Mitchell.(1962) Computer multiplication and division using binary logarithms. IRE Transactions on Electronic Computers, (4):pp. 512-517.
4. V. Mahalingam, N. Rangantathan, (2006) Improving Accuracy in Mitchell’s Logarithmic Multiplication Using Operand Decomposition, IEEE Transactions on Computers, Vol. 55, No. 2, pp. 1523-1535
5. Ellaithy DM, El. Moursy MA, Ibrahim GH, Zaki A and Zekry (2017) A. Double Logarithmic Arithmetic Technique for Low-Power 3-D Graphics Applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems: pp. 2144-52.
6. Ioannis Kouretas, Charalambos Basetas and Vassilis Paliouras. (2014) Low-power logarithmic number system addi- tion/subtraction and their impact on digital filters. IEEE transactions on computers, 62(11), pp. 2196-2209.
7. R. R. Selina, (2013) “VLSI implementation of piecewise approximated antilogarithmicconverter,” in Proc. Int. Conf. Commun. Signal Process. . (ICCSP), pp. 763–766.

8. Rabaey, J.M., Chandrakasan and Nikolic, B. (2002): 'Digital integrated circuits' (Prentice Hall).
9. K. Johansson, O. Gustafsson and L. Wanhammar, (2008) "Implementation of elementary functions for logarithmic number systems," IET Comput. Digit.Tech., vol. 2, no. 4, pp. 295–304.
10. C.T. Kuo and T.B. Juang, (2012) "A lower error antilogarithmic converter using novel four-region piecewise-linear approximation," in Proc. IEEE Circuits Syst. Conf., vol. 2. Dec., pp. 507 510.

AUTHORS PROFILE

P.Anusha , student, is currently pursuing her M.tech VLSI, School of Electronics Engineering, VIT Chennai, she has completed B.Tech in Electronics and Communication Engineering in JNTU Hyderabad. Her research areas are VLSI, Digital IC Design, and VLSI and signal processing.

G.kalpana received Ph.D degree in Job Scheduling in Grid Computing from SRM University, India . In 2006, she joined as lecturer in the Department of Computer Science, Faculty of Science & Humanities, Kattankulathur Campus, SRM Institute of Science & Technology.Her research interests include different aspects of Cloud Computing, Big Data Analytics.Presently he is working as an associate professor, Department of Computer Science, Faculty of Science & Humanities, Kattankulathur Campus, SRM Institute of Science & Technology.

Dr. T.Vigneswaran received the graduate degree in Electrical and Electronics Engineering from Bharathidhasan University in 2000, M.E (VLSI Design) from College Of Engineering, Guindy, Anna University in 2003 and the Ph.D degree in Low power VLSI design from SRM University, India in 2009. In 2003, he joined as lecturer in the department of ECE, SRM University.His research interests include different aspects of Low power VLSI circuit design and high speed algorithm and architecture development. He is a member of Indian science congress.Presently he is working as a professor, School of Electronics engineering, VIT University, Chennai.