International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1014-1023

Design and Implementation of approximate High-Order Compressors with Low Power and High Accuracy Approximate Multiplier on FPGA

Dr.P.Prasanna Murali Krishna [(1)] , Mrs.N.Swarupa Rani [(2)] , Gantepogu Rahul Teja [(3),] Thanugundla Hanumareddy [(4),] Pasupuleti Rama Krishna [(5)]
[1,2] *Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.*
[3,4,5,] *Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.*

Abstract In this paper, for many error-resilient applications, the approximation multiplier design seems to be a viable way to cut down on power usage. We present a low-power, high-accuracy approximation of an 8 x 8 multiplier design in this study. There are two key elements to the suggested design. First, in accordance with the significance, various weights accumulate their product terms using various compressors (to varying degrees of accuracy). Consequently, a tiny mistake can save power consumption. Second, we employ high-order approximate compressors (such as the 8:2 compressor) to lessen the logic of carry chains for the medium significance weights. It is the first design that we are aware of that effectively incorporates higher order approximation compressors into the approximate multiplier architecture. The experimental results demonstrate that the suggested approximation multiplier may achieve both low power and good accuracy when compared to an exact multiplier (Dadda tree multiplier) by using Xilinx Vivado on FPGA.

Keywords: Low Power VLSI Design,6G Communications, FPGA, Verilog HDL, Xilinx Tool.

I.INTRODUCTION

All these techniques perform the exact computation and modules produce the correct result. Accuracy of the module/device is always 100% in exact computing. But exact computing has one major drawback. It is not possible to optimize all the parameters of the circuit in exact computing. However, exact computing is not essential for every application. There are some applications like image processing and multimedia that can tolerate errors and provide meaningful results. Inexact (approximate) computing techniques have become popular because of their low complexity and less power consumption. Inexact computing produces reasonable results, even if it has low accuracy. In approximate computing, the value of error rate (ER), error distance (ED) and normalized error distance (NED) play an important role in calculating the final output. Error rate is given by several erroneous outputs over the total number of outputs. Error Distance is the arithmetic distance between an erroneous output and the correct one. Normalized Error Distance is the ratio of mean error distance over all inputs by maximum input of the circuit.

Microprocessors and Digital Signal Processors (DSP) are playing a significant role to handle the complexity of digital signal. About 95% of the processors in the market are based on digital signal [1]. Digital signal processors take care of

convolution, correlation and filtering of digital signal [2]. Multipliers, shifters and adders are mainly used to accomplish these tasks. Among the three modules, multiplier is the most complex one. Multipliers take more time and consume

higher power than other two modules [3]. Multipliers have three phases - generation of partial products, reduction of partial products and final stage addition. Reduction of partial products take much time and power in the multiplier. Many techniques were proposed to reduce the critical path in the multiplier [4], [5]. Among them, the use of compressors in partial product reduction stage is the most popular. Compressors are basic circuits which are made of full adders or half adders to count the number of ‘‘ones’’ in the input. Several compressors are required in the partial product reduction stage. Various compressors such as 3-2, 4-2, 5-2 and 5-3 were proposed by researchers in the last 20 years [6]–[9]. These are useful only when the size of multiplier is small. 16 × 16,32×32-bit multipliers require large size of compressors. High order compressors provide better results in terms of power and speed [10]–[12]. But it consumes more area than low order compressors. Several approximation techniques were proposed for adders and multipliers [14]–[32]. From central point to the most significant bit (MSB) is called accurate and to the least significant bit (LSB) is called inaccurate part of adders which was discussed by Zhu et al. [14]. Inaccurate computing in MSB side causes large error. The normal addition rule is applied in accurate part whereas a special method of addition takes place in inaccurate part. Output ‘‘sum’’ value is calculated normally when any one of the operand value of adder is ‘‘0’’. When both operands are ‘‘1’’, ‘‘sum’’ value can be fixed as ‘‘1’’ from that bit position to least significant bit. This technique is used to minimize the error distance of the adder Approximate XOR/XNOR adder for inexact computing is proposed by Yang et al. [15]. Both XOR and XNOR gates are required to calculate the output of the adder. Three different approximate methods were proposed. The output expression for ‘‘sum’’ and ‘‘carry’’ is approximated. Instead of using two XOR gates, only one XNOR gate has been utilized to calculate ‘‘sum’’. Similarly, one XOR, one OR and two AND gates are used for ‘‘carry’’. Low power imprecise adder were proposed by Jiang et al. [16] where they optimized transistor count, power consumption and power delay product (PDP) of the adder. Moreover, the number of incorrect outputs of the adder is also small. In [17] and [18], accuracy, error distance and various design parameters of approximate adders are analyzed and compared. Approximation methodologies were applied in generating the partial product phase [19]. A 2 × 2 bit approximate multiplier is designed by altering the one output combination. In this technique, multiplier produces ‘‘7d’’ when it multiplies ‘‘11’’ by ‘‘11’’. But the actual output of the multiplier is ‘‘1001’’. The probability of getting error in this multiplier is 0.0625. An error has been introduced in the partial product generation phase. Adder tree (reduction tree) of this multiplier is same as accurate multiplier.

Several other approximation techniques were proposed in the partial product reduction stage [20]–[27]. One need not consider the row of partial products when the value of multiplier bit is ‘‘0’’ also particular column value can be skipped when multiplicand bit is ‘‘0’’. This technique is called row and column

bypassing [20], [21]. In [22], some of carry-save adders are skipped in both horizontal and vertical directions based on the number of zeros in the multiplier input. In [23], partial product tree is splitted into two parts. Accurate multiplier was used in MSB side of the multiplier.No multipliers was used in LSB side where approximation

rule was applied.

In [24], ‘‘n’’ bit multiplier was implemented by two ‘‘n/2’’ bit sub-multipliers. Then, most significant ‘‘n/2’’ multiplier was implemented by two further ‘‘n/4’’ sub multipliers and least significant ‘‘n/2’’ multiplier was implemented by an approximate ‘‘n/4’’ multiplier. Then, all partial products are accumulated by a Wallace tree. Every proposed technique has its own merits and demerits. But in accuracy point of view, all techniques have high normalized error distance (NED) and a low pass rate. Pass rate of the multipliers discussed so far is almost 0%. The performance of the approximate multiplier is not only decided by the circuit metrics. It is also based on the error tolerance [28]. Getting low error and better circuit performance are always challenging.

Various inaccurate compressors were introduced in the reduction tree [29]–[33]. Compressors can handle large number of inputs than half and full adders. 4-2 compressor is widely used by various researchers which reduces the four partial products into two partial products. The probability of getting an error in approximate 4-2 compressor is 0.003 which is very minimum than any inaccurate adder circuit [29]. In [30], different approximate compressors are proposed and utilized in 16 × 16-bit Vedic multiplier.4-2 compressor is further optimized to get the best power and lower delay with compromising accuracy [31]. Two designs of approximate 4-2 compressor are proposed in this paper. Design 2 of approximate 4-2 compressor is the best compressor in recent days. In [28], the performance of various multipliers was compared. It was found that, introducing compressors in the partial production tree gives the lowest error rate, minimum normalized error distance and decent circuit metrics. In this paper, we have proposed higher order compressor for 16 × 16-bit multiplier. Use of several approximate 4-2 compressors in large size multiplier causes large error. It is necessary to get a minimum error as well as decent circuit performance.

II.EXISTING SYSTEM AND PROPO

1.Existing System:

All these techniques perform the exact computation and modules produce the correct result. Accuracy of the module/device is always 100% in exact computing. But exact computing has one major drawback. It is not possible to optimize all the parameters of the circuit in exact computing. However, exact computing is not essential for every application. There are some applications like image processing and multimedia that can tolerate errors and provide meaningful results. Inexact (approximate) computing techniques have become popular because of their low complexity and less power consumption. Inexact computing produces reasonable results, even if it has low accuracy. In approximate computing, the value of error

rate (ER), error distance (ED) and normalized error distance (NED) play an important role in calculating the final output. Error rate is given by several erroneous outputs over the total number of outputs. Error Distance is the arithmetic distance between an erroneous output and the correct one. Normalized Error Distance is the ratio of mean error distance over all inputs by maximum input of the circuit.

Proposed System Block diagram

Typically, a multiplier consists of three parts. In the first part, AND gates are utilized to generate partial products. In the second part, the maximum height of PPM (partial product matrix) is reduced by using a carry save adder tree. In the third part, a carry propagation adder is used to produce the result. The design complexity of a multiplier is primarily related to the PPM reduction circuitry (i.e., the multiplier is primarily related to the PPM reduction circuitry (i.e., the second part). Thus, the study of multiplier design is focuses on the optimization of the PPM reduction circuitry.

we propose an approximate 16 x 16 multiplier design. Fig. 5 gives the overall structure of our PPM reduction circuitry. According to the significance, the weights are classified into three categories: the higher significance weights, the middle significance weights, and the lower significance weights. Note that the designers are allowed to configure the number of higher significance weights, the number of middle significance weights and the number of lower significance weights for the trade-off between the power consumption and the computational accuracy. To reduce the power consumption with a small error, our PPM reduction circuitry applies the significance driven logic compression technique as below: the higher significance weights use accurate (i.e., exact) 4:2 compressors; the middle significance weights use our approximate high-order compressors (i.e., the approximate n:2 compressors proposed in Section II); the lower significance weights use inaccurate compressors (OR-tree based approximation). Our PPM reduction circuitry has two stages. The first stage is for all the weights. The second stage is only for the higher significance weights. After the second stage is completed, each weight has at most two product terms. Thus, a carry propagation adder can be used to produce the final result. In the following, we elaborate the details of these two stages.

A. The First Stage

For each lower significance weight, we use a simple ORtree based approximation for power saving. Suppose that the number of inputs is n. If $n \leq 2$, no action is performed. On the other hand, if $n > 2$, we use an OR tree for n-1 inputs to approximate the accumulation result of these n-1 inputs. Thus, after the first stage is done, each lower significance weight has at most two product terms. For each middle significance weight, we use our approximate n:2 compressor (as described in Section II) for power saving, where n is the number of product terms in this weight. As described in Section II, the designers can choose one of the following two implementations: one implementation is with accurate Sum and approximate Carry and the other implementation is with approximate Sum and approximate Carry. After the first stage is done, each middle significance weight has at most two product terms. To achieve high

accuracy, for each higher significance weight, we use accurate (i.e., exact) 4:2 compressors. For each accurate 4:2 compressor, if the number of product terms is less than 4, the values of other inputs to this compressor are set to be 0. In the rightmost higher significance weight, the carry bit Cin of one accurate 4:2 compressor is from the Carry output of the leftmost middle significance weight, and the carry bit Cin of the other one accurate 4:2 compressor is set to be 0.

B. The Second Stage

Note that the second stage is only for the higher significance weights. To achieve high accuracy, we use accurate (i.e., exact) 4:2 compressors to reduce the maximum height of the PPM. The carry bit Cin of the rightmost accurate 4:2 compressor is set to be 0. As shown in Fig. 5, after the second stage is completed, each higher significance weight has two product terms.

III. RESULTS AND ANALYSIS DISCUSSION

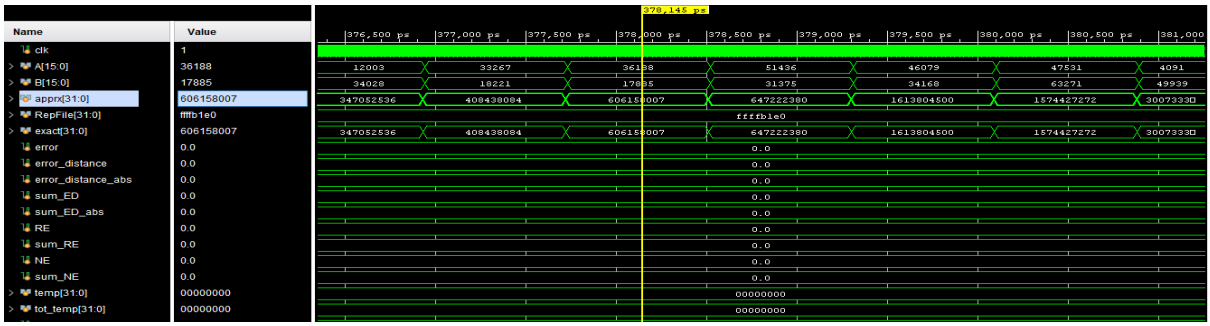

FIG1.16X16 BIT MULTIPLIER SIMULATION OUTPUT

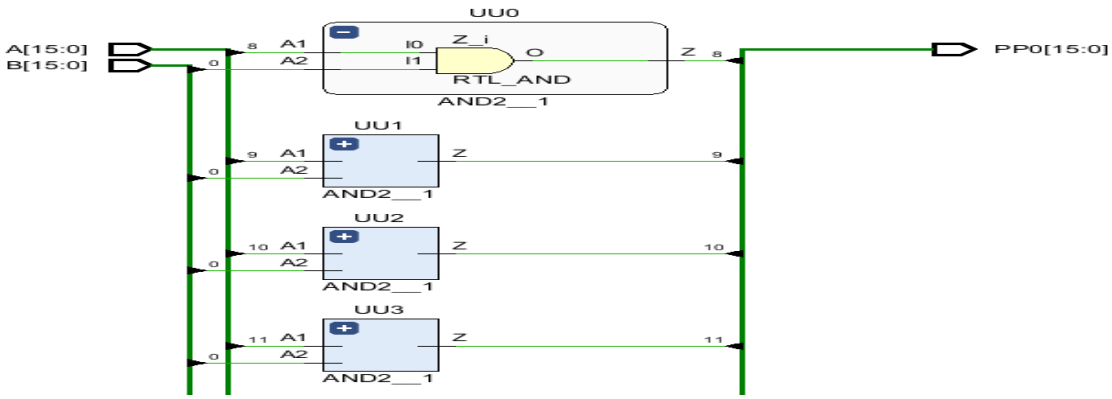

Fig2.RTL schematic

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1014-1023

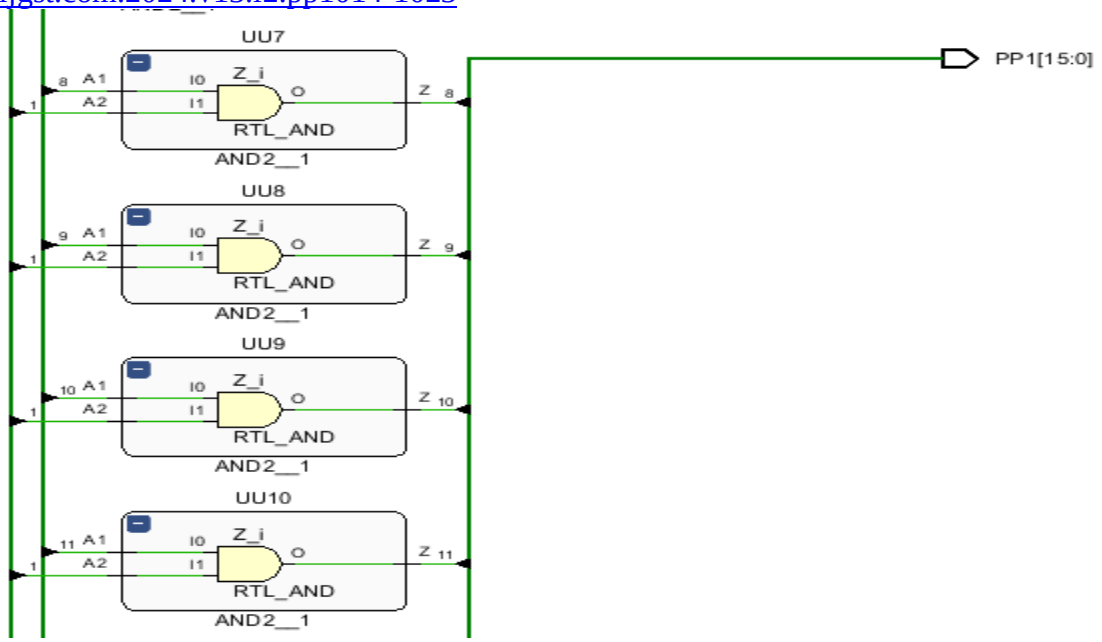

Fig3. 16X16 BIT MULTIPLIER SYNTHESIS DESIGN1

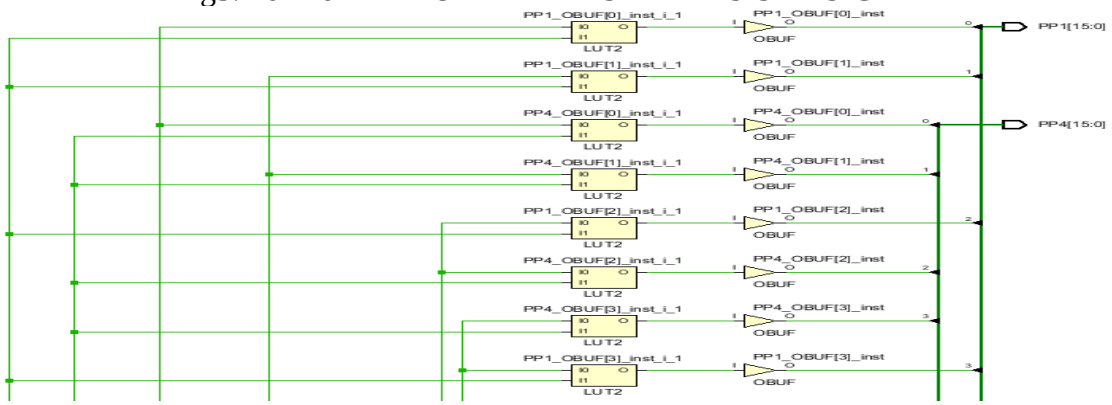

Fig4. 16X16 BIT MULTIPLIER SYNTHESIS DESIGN2

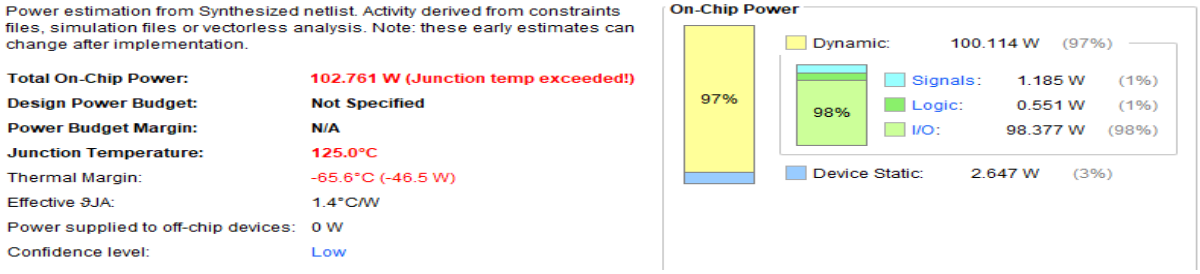

Fig5.power estimated.

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1014-1023

Resource	Utilization	Available	Utilization %
LUT	128	117120	0.11
IO	288	204	141.18

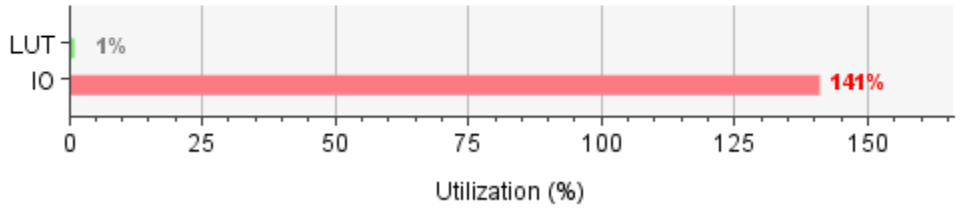

Fig6.Area Utilization (LUTs and IOBs).

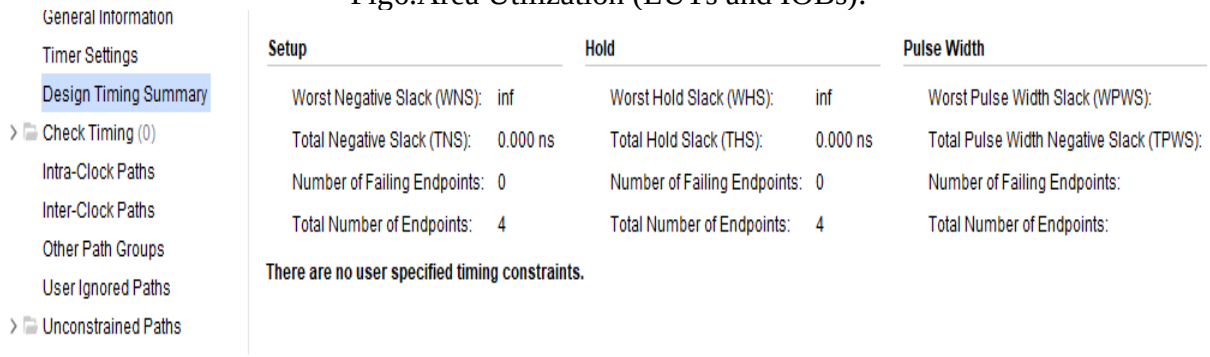

Fig7.Timing report

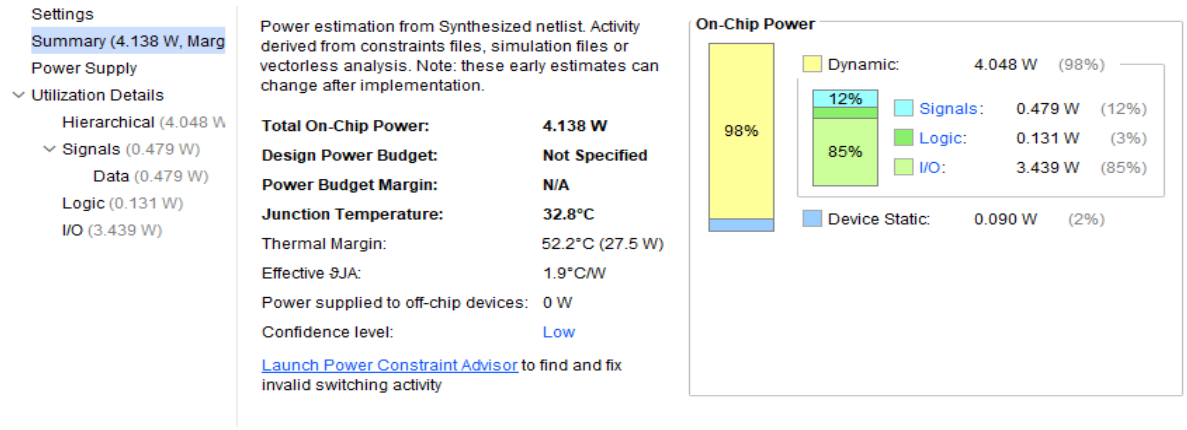

Fig8. Optimised power.

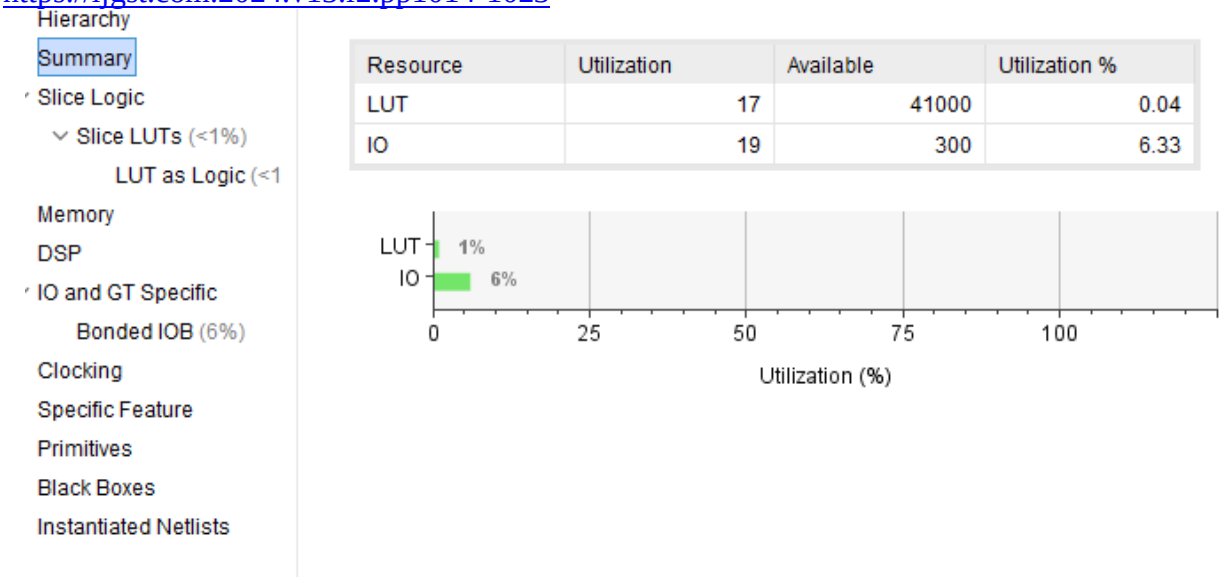

Fig9. Optimised Utilization (LUTs and IOBs)

Conclusion

Approximate 16 × 16 bit multipliers are designed using those proposed 4-2 compressors. Approximate multipliers provide better performance than accurate multipliers with compromising of error rate. Moreover, we have achieved high pass rate and the normalized error distance value of multipliers designed using proposed 4X2 compressor is very small. Latency of the proposed multiplier is almost equal as compared to the accurate multipliers.

REFERENCES

[1] D. Liu, Embedded DSP Processor Design, 1st ed. San Mateo, CA, USA:Morgan Kaufmann, 2008.

[2] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, 1st ed. Hoboken, NJ, USA: Wiley, 1999.

[3] Y. Kim, Y. Zhang, and P. Li, ‘‘An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems,’’ in Proc. Int.Conf. Comput.-Aided Design (ICCAD), Nov. 2013, pp. 130–137.

[4] A. Pishvaie, G. Jaberipur, and A. Jahanian, ‘‘Improved CMOS (4; 2) compressor designs for parallel multipliers,’’ Comput. Elect. Eng., vol. 38,no. 6, pp. 1703–1716, Nov. 2012.

[5] D. Baran, M. Aktan, and V. G. Oklobdzija, ‘‘Energy efficient implementation of parallel CMOS multipliers with improved compressors,’’ in Proc.ACM/IEEE Int. Symp. Low-Power Electron. Design (ISLPED), Aug. 2010,pp. 147–152.

[6] S. Veeramachaneni, K. M. Krishna, L. Avinash, S. R. Puppala, and M. B. Srinivas, ‘‘Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors,’’ in Proc. Int. Conf. VLSI Design (VLSID),Jan. 2007, pp. 324–329.

[7] R. Menon and D. Radhakrishnan, ‘‘High performance 5 : 2 compressor architectures,’’ Proc. IEE-Circuits, Devices Syst., vol. 153, no. 5,pp. 447–452, Oct. 2006.

[8] A. Pishvaie, G. Jaberipur, and A. Jahanian, ‘‘High-performance CMOS (4:2) compressors,’’ Int. J. Electron., vol. 101, no. 11, pp. 1511–1525,Jan. 2014.
[9] O. Kwon, K. Nowka, and E. Swartzlander, Jr., ‘‘A 16-bit by 16-bit MAC design using fast 5:3 compressor cells,’’ J. VLSI Signal Process. Syst.Signal, Image Video Technol., vol. 31, no. 2, pp. 77–89, Jul. 2002.
[10] S. Mehrabi, R. F. Mirzaee, S. Zamanzadeh, K. Navi, and O. Hashemipour,‘‘Design, analysis, and implementation of partial product reduction phase by using wide m:3 ($4 \leq m \leq 10$) compressors,’’ Int. J. High Perform. Syst. Archit., vol. 4, no. 4, pp. 231–241, Jan. 2013.
[11] A. Dandapat, S. Ghosal, P. Sarkar, and D. Mukhopadhyay, ‘‘A 1.2-ns16 × 16 bit binary multiplier using high speed compressor,’’ Int. J. Elect.Electron. Eng., vol. 4, no. 3, pp. 234–239, 2010.
[12] R. Marimuthu, M. Pradeepkumar, D. Bansal, S. Balamurugan, and P. S. Mallick, ‘‘Design of high speed and low power 15-4 compressor,’’ in Proc. Int. Conf. Commun. Signal Process. (ICCSP), Apr. 2013,Pp. 533–536.
[13] J. Liang, J. Han, and F. Lombardi, ‘‘New metrics for the reliability of approximate and probabilistic adders,’’ IEEE Trans. Comput., vol. 62, no. 9, pp. 1760–1771, Sep. 2013.
[14] N. Zhu, W. L. Goh, and K. S. Yeo, ‘‘An enhanced low-power highspeed adder for error-tolerant application,’’ in Proc. 12th Int. Symp. Integr.Circuits (ISIC), Nov. 2009, pp. 69–72.
[15] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, ‘‘Approximate XOR/XNOR-based adders for inexact computing,’’ in Proc. IEEE Int.Conf. Nanotechnol. (IEEE-NANO), Aug. 2013, pp. 690–693.
[16] H. Jiang, J. Han, and F. Lombardi, ‘‘A comparative review and evaluation of approximate adders,’’ in Proc. ACM Great Lakes Symp. VLSI (GLSVLSI),May 2015, pp. 343–348.
[17] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy,‘‘IMPACT: IMPrecise adders for low-power approximate computing,’’in Proc. Int. Symp. Low Power Electron. Design (ISLPED), Aug. 2011,pp. 409–414.
[18] C. Liu, J. Han, and F. Lombardi, ‘‘A low-power, high-performance approximate multiplier with configurable partial error recovery,’’ in Proc. Int.Conf. Design Autom. Test Eur. (TEST), Mar. 2014, Art. no. 95.
[19] P. Kulkarni, P. Gupta, and M. D. Ercegovac, ‘‘Trading accuracy for power in a multiplier architecture,’’ J. Low Power Electron., vol. 7, no. 4,pp. 490–501, Dec. 2011.
[20] S. Balamurugan and P. S. Mallick, ‘‘Fixed-width multiplier circuits using column bypassing and decompositon logic techniques,’’ Int. J. Elect. Eng.Inform., vol. 7, no. 4, pp. 655–664, Dec. 2015.
[21] S. Balamurugan, S. Ghosh, Atul, S. Balakumaran, R. Marimuthu, and P. S. Mallick, ‘‘Design of low power fixed-width multiplier with row bypassing,’’ IEICE Electron. Exp., vol. 9, no. 20, p. 1568–1575, Oct. 2012.
[22] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, ‘‘Bio-inspired imprecise computational blocks for efficient VLSI implementation of softcomputing applications,’’ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57,no. 4, pp. 850–862, Apr. 2010.
[23] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, ‘‘Low-power high-speed multiplier for error-tolerant application,’’ in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.

https://ijgst.com.2024.v13.i2.pp1014-1023

[24] K. Bhardwaj, P. S. Mane, and J. Henkel, ‘‘Power- and area-efficient approximate Wallace tree multiplier for error-resilient systems,’’ in Proc. 15th Int. Symp. Quality Electron. Design (ISQED), Mar. 2014, pp. 263–269.

[25] M. S. K. Lau, K.-V. Ling, and Y.-C. Chu, ‘‘Energy-aware probabilistic multiplier: Design and analysis,’’ in Proc. Int. Conf. Compil., Archit.,Synth. Embedded Syst., Oct. 2009, pp. 281–290.

[26] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, ‘‘MACACO:Modeling and analysis of circuits for approximate computing,’’ in Proc.Int. Conf. Compil., Archit., Synth. Embedded Syst. (ICCAD), Nov. 2011,pp. 667–673.

[27] F. Farshchi, M. S. Abrishami, and S. M. Fakhraie, ‘‘New approximate multiplier for low power digital signal processing,’’ in Proc. 17th Int. Symp.Comput. Archit. Digit. Syst. (CADS), Oct. 2013, pp. 25–30.

[28] H. Jiang, C. Liu, N. Maheshwari, F. Lombardi, and J. Han, ‘‘A comparative evaluation of approximate multipliers,’’ in Proc. Int. Symp. Nanosc.Archit. (NANOARCH), Jul. 2016, pp. 191–196.

[29] C.-H. Lin and I.-C. Lin, ‘‘High accuracy approximate multiplier with error correction,’’ in Proc. IEEE 31st Int. Conf. Comput. Design (ICCD),Oct. 2013, pp. 33–38.

[30] Y. Bansal and C. Madhu, ‘‘A novel high-speed approach for 16 × 16 vedic multiplication with compressor adders,’’ Comput. Elect. Eng., vol. 49,pp. 39–49, Jan. 2016.