iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** IV **Month of publication:** April 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Efficient Array Based Approximate Arithmetic Computing Multiplier and Squarer

E. Pavithra[1], Dr. B. Sargunam[2]
Department of Electronics and Communication Engineering,
Avinashilingam University for Women, Coimbatore.

Abstract— Power consumption is one of the most important challenges in arithmetic circuit designs. Exact multiplier produces exact result but it consumes more power which is the main drawback of exact multipliers. The results of approximate multiplier are not exact but the area and power consumption are much less when compared with the exact multipliers. Little accuracy deviation is not an important issue in many applications, hence approximate multiplier is preferred rather than the exact multiplier. The Array based Approximate Arithmetic Computing(AAAC) circuit consists of a Booth multiplier along with the error compensation unit. The Booth multiplier has an adder which combines the outputs of accurate part and the truncated part. An error compensation unit is used to compensate the errors during approximation. The existing approximate and exact multipliers are designed with carry look-ahead adder (CLA) which occupies more area and power requirement is large. This problem is eliminated in efficient AAAC multiplier by carry select adder(CSLA), which helps to improve the performance of the multiplier in terms of area, delay and power. The similar technique is applied to the squarer circuit and significant reduction in area, delay and power are obtained. The Multiplier and Squarer are simulated using Modelsim SE 10.0b software and implemented in Spartan3 FPGA device using Xilinx ISE 9.1i software. Key words—Multiplier, Squarer, Approximate arithmetic computing, Adder

I. INTRODUCTION

Area, Power consumption and delay are the most important issues in VLSI design . Array-based arithmetic computing circuits such as adders [1], multipliers[2] and squarers[3] have been used widely for many applications. These circuits consume a significant amount of power in a chip design. Hence, reduction of power consumption of AAAC circuits is an important design consideration.

There are many existing methods [4]-[7] available for approximate multipliers and squarers. The performance of those circuits is less in terms of area, delay and power. This paper presents an efficient array based approximate arithmetic multiplier and squarer which consumes less area, power and delay.

The AAAC model uses a Booth multiplier [8][10] and an error compensation unit. Encoding, Selection, Compression and Addition are the four main blocks inside the multiplier [11]. Common exact and approximate multipliers use CLA to generate the final product, which occupies more area and power. The efficiency of the multiplier is improved by replacing the adder with CSLA[12].

II. ERROR FREE COMPUTING METHOD

The error free multiplication[13] needs complex circuits hence the area and power requirement for this method are more. The results of this multiplier is accurate but processing speed is less. It uses AND gates for partial product generation. For example in order to perform 16x16 bits multiplication it needs 16 partial products. Number of partial products needed is high which makes the circuit more complex.

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

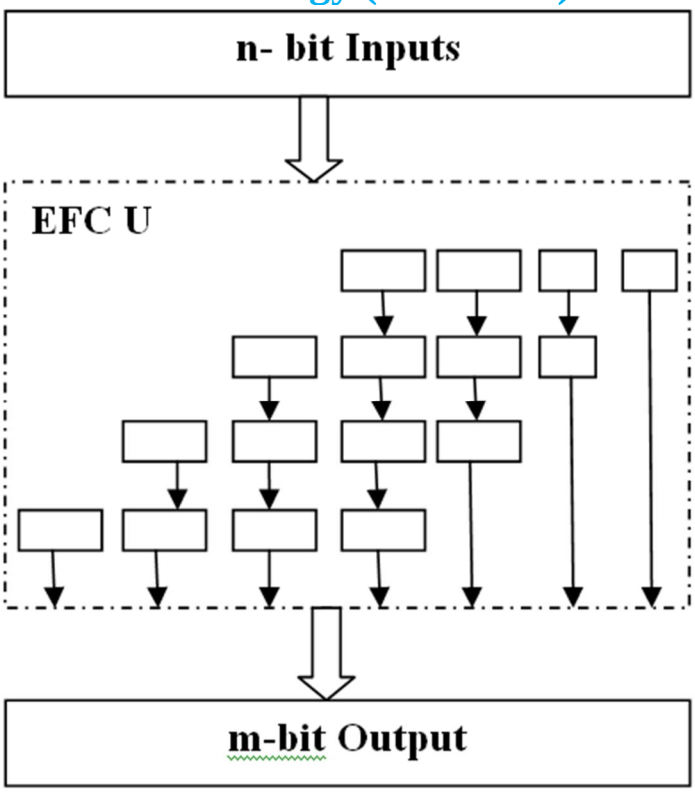


Fig. 1 Error Free Computing Method

The error free computing multiplier has 'n' number of bits which are applied to the Error Free Computing Unit (EFCU) as shown in Fig1. The EFCU unit generates the partial products and then using an adder the final m-bit output is obtained which has high delay, area and power consumption.

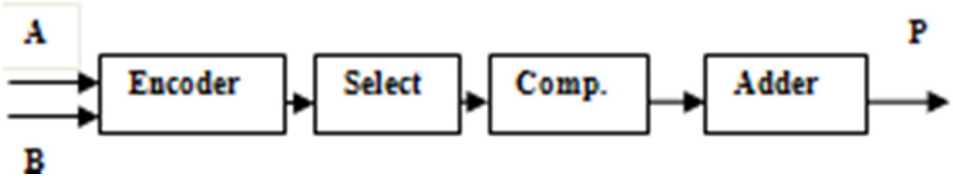


Fig. 2 Block Diagram of Error Free Multiplier

The general block diagram of error free multiplier is shown in Fig 2. A and B are the two 'n' bit inputs which are applied to the inputs of the error free multiplier. These bits are processed by selection block and then compression block. Finally CLA is used to generate the final product.

III. AAAC MODEL

The AAAC circuits are based on approximation [13] and rounding off between the values. Hence the results of this multiplier and squarer are less accurate than the exact multiplier. These circuits are preferred rather than the exact computation circuits because of the less area and power consumption. Though they contain some error in the output, they can be reduced by adding an error compensation circuit. The general structure for the AAAC model shown in fig 3.

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

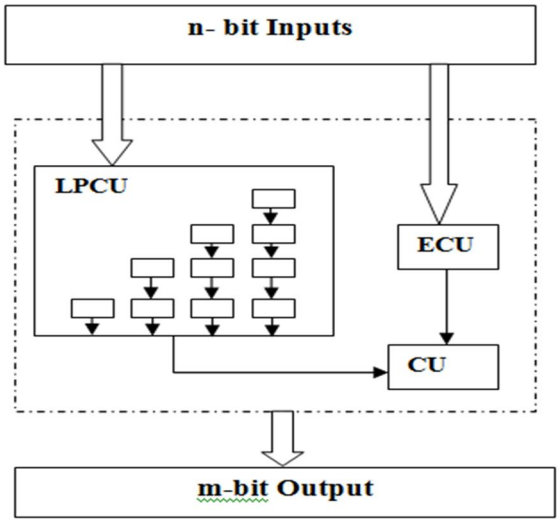


Fig. 3 Structure of AAAC Model

Inside the AAAC model there are three major units. They are(i) Low Precision Computing Unit(LPCU), (ii)Error Compensation unit(ECU) and (iii)Combine unit(CU).

The inputs of the multiplier is divided into two parts which are most significant bits(MSB) and least significant bits(LSB). The MSB bits are associated with the LPCU unit which produces the accurate result and approximation is done at the LSB bits which are associated with the ECU unit. The ECU unit is used to compensate the errors during approximation. An ECU unit is designed with the help of a signature generator and a multiplexer. Then the outputs of LPCU unit and the ECU units are combined using the Combine unit(CU). The final product obtained from the combine unit.

IV. EFFICIENT AAAC MULTIPLIER AND SQUARER

An AAAC multiplier and squarer are proposed with the help of AAAC model to reduce the area, power consumption and delay. An approximate 16x16 fixed width Booth multiplier and 16x16 fixed widths squarer with error compensation unit are proposed. The block diagram for the proposed multiplier is shown in fig.4.

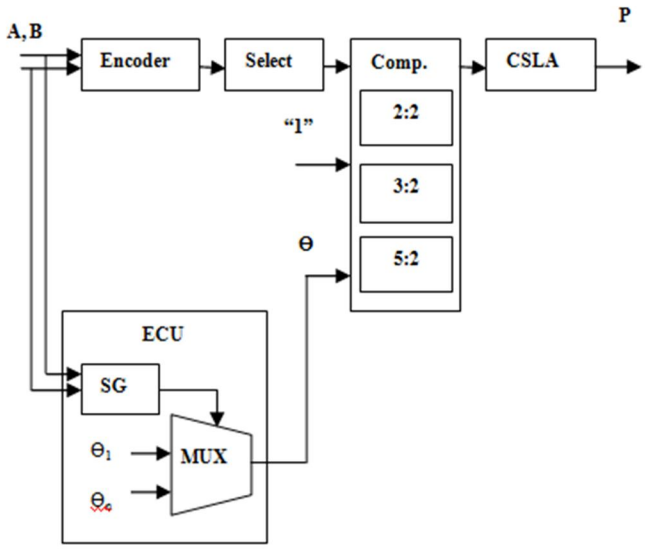


Fig.4 Block Diagram of Efficient AAAC Method

The Booth multiplier has two outputs which are accurate part(AP) output and truncated part(TP) output. Both the outputs are combined using an adder. The selection of adder plays the main role in area and power consumption reduction.

The inputs applied to the multiplier are kept as A and B, where A is the multiplicand and B is the multiplier. The encoder block is

designed by applying the Booth algorithm. Then at the selection block it will allow only the selected partial products for further processing. The values of the partial products will be any one of the following: 0, A, -A, 2A, -2A, where '-' sign indicates the complement function, 'A' means 1's complement and '2A' represents 2's complement. Then all the partial products are compressed using a compressor. ECU block is used to compensate the errors during the approximation. It is designed with the help of a signature generator and a multiplexer. The signature generator will generate many signatures depending upon the input applied and the multiplexer selects any one of the predetermined values which is given to the compressor block. At the compressor block the output of selection block and ECU block are combined. Then the final output of the multiplier is obtained at the adder unit.

The CSLA is used to perform the final addition part. The speed of addition in digital circuits is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position. The CSLA is used to moderate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum.

The carry select adder is a particular way to implement an adder which is a logic element that computes the (n+1) bit sum of two (n) bit numbers. In the existing methods CLA is used which produces more delay at the output. So, it is replaced by CSLA which will make the multiplier more efficient than the error free computing multiplier and the AAAC multiplier in terms of area, delay and power.

The main advantage of CSLA is it performs the arithmetic calculations faster and it reduces the problem of carry propagation delay. Using the similar diagram a squarer is also proposed and reduction in area, delay and power consumption are achieved. Both the the multiplier and squarer are coded in hardware description language and simulated. The FPGA implementation results show that the proposed method is more efficient in terms of area, delay and power consumption.

V. RESULTS AND DISCUSSION

The efficient array based approximate arithmetic computing multiplier and squarer are designed in(Very High Speed Integrated Circuit Hardware Description Language) VHDL and simulated using Modelsim SE 10.0b software. Area, delay and power consumption are measured by implementing the multipliers in Spartan 3 FPGA device. The obtained FPGA implementation results are compared with the exact multiplier and approximate multipliers.

The EFCU method is one of the accurate arithmetic circuit which produces the exact result but area, delay and power consumption for this method is very high. This is the main drawback of error free computing method.

The AAAC is the approximate computation method and the results of this method will not be the accurate. The multiplier and squarer designed using this method have approximated results. The area, delay and power consumption for this method are less when compared with the error free computing method.

The efficient AAAC multiplier and squarer are the approximate computation circuits. The results of these circuits are approximated values but accuracy is improved when compared with the AAAC method. The area, delay and power consumption requirement is much less when compared with the error free computing method and the AAAC method.

TABLE 1
COMPARISON OF DIFFERENT MULTIPLIERS

Parameter	EFCU Method	AAAC Method	Efficient AAAC Method
Area(No. of Gates)	21,672	13,268	10,610
Power (mW)	117mW	82mW	71mW
Delay (nS)	6.151nS	6.151nS	3.886 nS

Table 1 shows the comparison results of three different multipliers in terms of area, delay and power consumption. On compared with the Existing EFCU method and AAAC method, the Efficient AAAC method utilizes less area, delay and power requirements.

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

TABLE 2
COMPARISON OF DIFFERENT SQUARERS

Parameter	EFCU Method	AAAC Method	Efficient AAAC Method
Area(No. of Gates)	17,043	9,879	5,727
Power (mW)	13	75	70
Delay (nS)	7.008	7.008	5.254

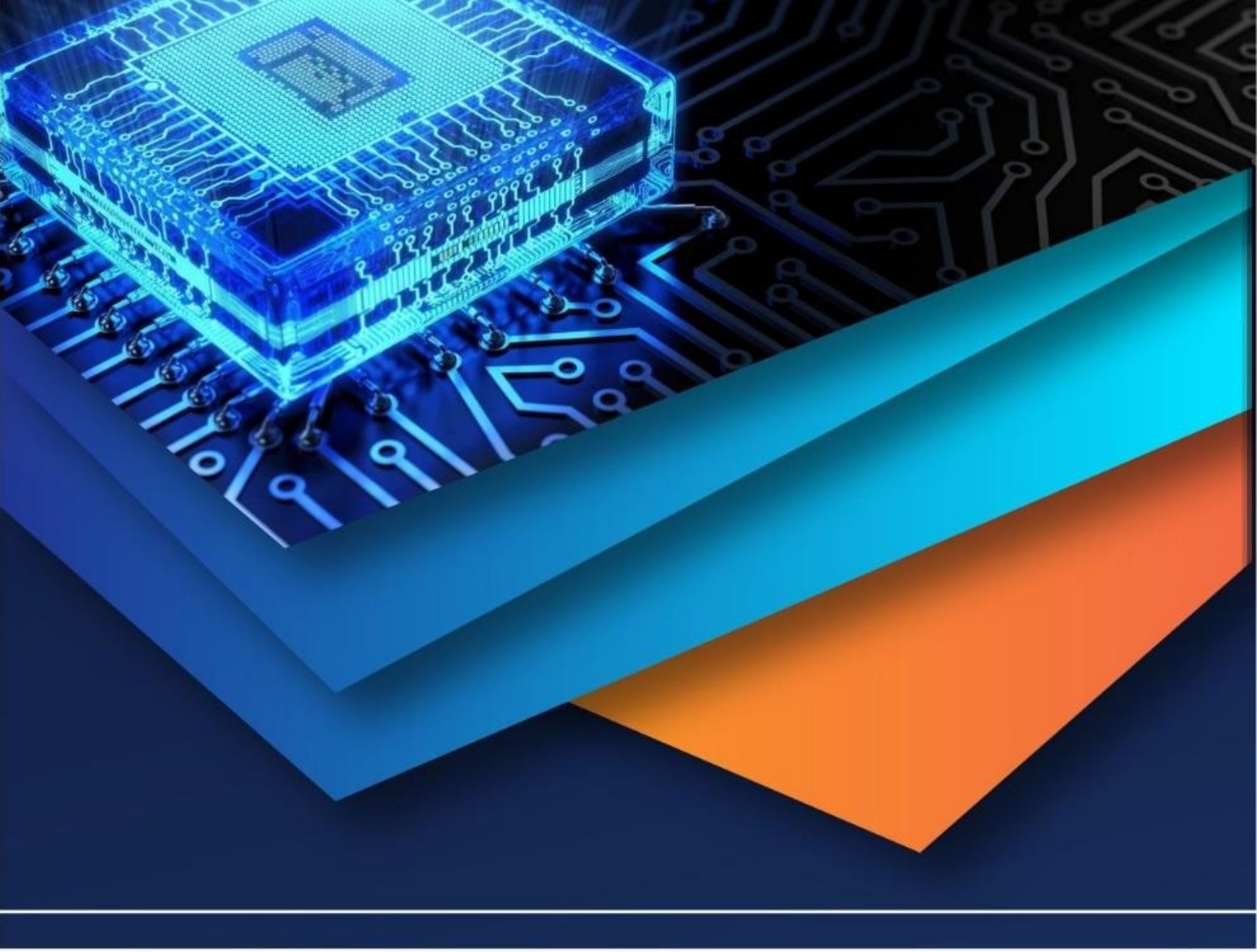
Table 2 shows the comparison results of three different squarers in terms of area, delay and power. It is noted that the efficient AAAC based squarer occupies less area, delay and power consumption compared with the existing EFCU method and AAAC method.

VI. CONCLUSION

An efficient array based approximate arithmetic computing multiplier and squarer are designed using carry select adder with less area, power consumption and delay. Simulation is performed for different multipliers and squarers using Modelsim SE 10.0b software. In order to analyze the performance of the proposed method area, delay and power consumption are measured by implementing in Spartan3 FPGA device using Xilinx ISE 9.1i software. The proposed multiplier and squarer are compared with the existing exact and approximate arithmetic computation circuits. The FPGA implementation results show that the performance of the efficient AAAC based multiplier and squarer is increased in terms area, delay and power consumption.

REFERENCES

[1] K. Yongtae, Y. Zhang, and P. Li, "An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems," Proc. 2013 IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), pp. 130–137.\
[2] E. G. Walters, III and M. J. Schulte, "Efficient function approximation using truncated multipliers and squarers," in Proc. 17th IEEE Symp. Comput. Arith. (ARITH-17 2005).
[3] E. G. Walters, III, M. J. Schulte, and M. G. Arnold, "Truncated squarers with constant and variable correction," Proc. SPIE, vol. 5559, pp. 40–50, 2004.
[4] E. J. King and E. E. Swartzlander, "Data-dependent truncation scheme for parallel multipliers," Conf. Rec. 31st Asilomar Conf. Signals, Syst. Comput., 1997.
[5] V.-P. Hoang and C.-K. Pham, "Low-error and efficient fixed-width squarer for digital signal processing applications," in Proc. 2012 4th Int. Conf. Commun. Electron. (ICCE), pp. 477–482.
[6] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," Proc. Design, Autom., Test Eur. Conf. Exhib. (DATE) 2012, pp. 1257–1262.
[7] M. J. Schulte and E. E. Swartzlander, "Truncated multiplication with correction constant," Proc. Workshop VLSI Signal Processing VI, pp. 388–396, 1993.
[8] S.-J. Jou, M.-H. Tsai, and Y.-L. Tsao, "Low-error reduced-width Booth multipliers for DSP applications," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 11, pp. 1470–1474, 2003.
[9] J.-P. Wang, S.-R. Kuang, and S.-C. Liang, "High-accuracy fixed-width modified booth multipliers for lossy applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 1, pp. 52–60, 2011.
[10] Y.-H. Chen, "An accuracy-adjustment fixed-width booth multiplier based on multilevel conditional probability," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 1, pp. 203–207, 2014.
[11] S. O. N. G. Min-An, V. A. N. Lan-Da, and K. U. O. Sy-Yen, "Adaptive low-error fixed-width booth multipliers," IEICE Trans. Fundam. Electron., Commun., Comput. Sci., vol. 90, no. 6, pp. 1180–1187, 2007.
[12] S. Lu, "Speeding up processing with approximation circuits," in Computer, vol. 37, pp. 67–73, 2004.
[13] K.-J. Cho et al., "Design of low-error fixed-width modified booth multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522–531, 2004.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY