International Journal of Electrical Electronics & Computer Science Engineering
Volume 6, Issue 1 (February, 2019) | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

ASIC Design of Approximate Booth Multiplier

N. Arun[1], V. Govarthini[2], K. Semmalar[3], K. Saranya[4]
[1-3]UG Scholar, [4]Assistant Professor, Electrical and Electronics Engineering Department
Dr. Mahalingam College of Engineering and Technology, Coimbatore, Tamil Nadu, India
[1]heroarun97@gmail.com, [2]govasweet98@gmail.com, [3]ksemmalar1998@gmail.com, [4]saranya@drmcet.ac.in

Abstract: ***Approximate computing or Inexact computing is a methodology designed to achieve low power, high performance multipliers and also reduces the circuit complexity by relaxing the requirement of strict accuracy. Multipliers contribute to a lot of delay and consumes high power due to increased number of switching operations. With the advancement of technology, the demand for high speed processors and thus the demand for energy- efficient arithmetic units has increased. In this brief, we present the design of approximate booth multipliers based on approximate radix 4 booth encoding algorithms and a regular partial product array.***

I. INTRODUCTION

Multipliers are widely used in the arithmetic units of microprocessors, multimedia and digital signal processors. But high performing low power consuming multipliers are in high demand for embedded systems and other application which involves approximate results. It is extremely difficult to further improve the performance of the multipliers and to reduce the power consumption of the multipliers under the requirement of full accuracy. In accordance to the human perception, the requirement high precision and exactness is not so strict for many applications. This requirement of exactness and high precision in the operations of digital logic circuits are generally related to the acceptance of correctness of information processing.

By relaxing the requirement of strict accuracy, performance and power consumption can be substantially improved. This design principle is known as Approximate or Inexact computing. This principle is generally used in numerous error- tolerant applications.

The basic operations of an arithmetic processor are addition and multiplication. Multiplication is quite complex when compared with addition, because it requires the accumulation of partial product rows. The most widely used high performance multiplier consists of a modified booth encoding. This reduces the number of partial product rows by its half in its first step. Here in the paper, the performance of approximate radix 4 booth multipliers are discussed. The new booth encoding method is introduced to reduce the partial products and also to reduce the error rate.

Section 2 describes the basic Booth multiplication. Section 3 discusses about the various methods of booth approximation in Ref [9] and the proposed method. Section 4 presents the comparison of performance of proposed method.

II. BOOTH MULTIPLIER

A Booth multiplier consists of Partial product generation using a booth encoder, Partial product accumulation using a compressor and Final product generation using a fast adder as shown in figure 2.1.The partial products are generated using the radix 4 booth encoding shown in Table 2.1.

Table 2.1. Booth Encoding Table

B	Zn	Partial Product
000	0	0
001	1	1 x Multiplicand
010	1	1 x Multiplicand
011	2	2 x Multiplicand
100	-2	-2 x Multiplicand
101	-1	-1 x Multiplicand
110	-1	-1 x Multiplicand
111	0	0

The final product generation consists of a fast adder. The fast adder used here is "Carry look- ahead adder".This adder is superior to n^{th} full adder. The speed is high compared to other adders. Generally, full adders are dependent on the previous output. This owes to the delay in the circuit. But carry look- ahead adder generates its Generate, Propagate and Carry in advance so that the adders need not depend on the previous output and need not wait for the previous adder to generate it. The generate, propagate and carry are generated in parallel to each adder. Thus, the delay is reduced. Hence, the carry look- ahead adder is referred as a fast adder. The Carry save adder will also fall under the category of fast adder. But the circuit size of carry save adder is high in complexity compared with carry look- ahead adder. Hence, the usage of the carry look- ahead adder is an advantage both in the case of delay maintenance and size of the circuit as discussed in Ref[9].

Consider the multiplication of two N bit integers. Here, multiplicand A and Multiplier B in two's complement is given as follows:

$$A= -a_{N-1}2^{N-1}+\sum_{i=0}^{N-2}a_i2^i \quad (1)$$

$$B= -b_{N-1}2^{N-1}+\sum_{i=0}^{N-2}b_i2^i \quad (2)$$

International Journal of Electrical Electronics & Computer Science Engineering
Volume 6, Issue 1 (February, 2019) | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

The output equation of the exact booth encoder is given in equation (3).

$Pp_{ij}=(b_{2i}\hat{}b_{2i-1})(b_{2i+1}\hat{}a_j)+(\sim(b_{2i}\hat{}b_{2i-1}))(b_{2i+1}\hat{}b_{2i})(b_{2i+1}\hat{}a_{j-1})$ (3)

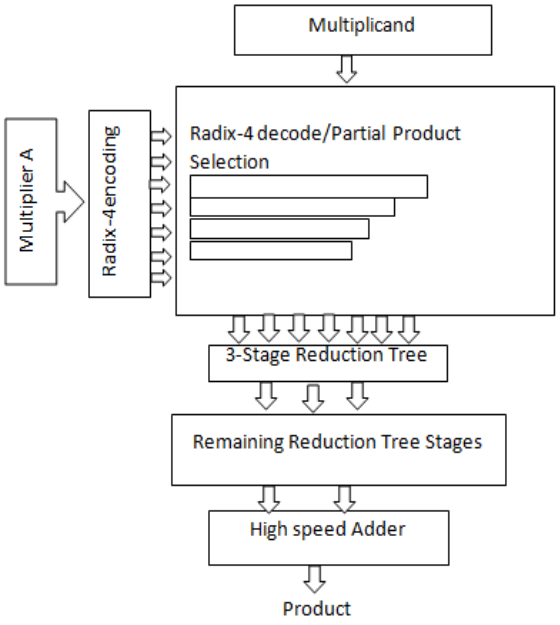

Fig. 2.1. Block Diagram of Booth Multiplication

III. Approximate Radix 4 Booth Encoding

3.1. Booth Approximation 1 from [9]:

Here, the K map of R4ABE1 is shown where a few 1 is replaced by 0. This approximation is aimed at making the table as symmetrical as possible. The advantage is a very small error occurs, as only 4 entries are modified. The output equation of the approximate booth encoder 1 is given as follows:

$Pp_{ij} = a_j\sim(b_{2i+1}b_{2i})b_{2i-1} + a_j\sim(b_{2i+1})b_{2i}\sim(b_{2i-1}) + \sim a_j(b_{2i+1}b_{2i})b_{2i-1} + \sim a_j b_{2i+1}\sim(b_{2i})b_{2i-1} = (b_{2i}\hat{}b_{2i-1})(b_{2i+1}\hat{}a_j)$ (4)

(0) = entry where 1 is modified to 0.The error rate is given by: 4/32= 12.5%.4 bits out of 32 bits has been changed which adds to 12.5% of error.

3.2. Booth Approximation 2 from [9]:

In this method, a few entries are changed from 0 to 1. The strategy for R4ABE2 is that in addition to having a symmetric truth table at a small error, the number of prime implicants (identified by the rectangular) should be as small as possible. The modification of R4ABE2 is not only achieved by changing 1 to 0, but also changing 0 t0 1.Thus, the approximate product can be either larger or smaller than the exact product and errors can complement each other in partial product reduction process. The output equation of the approximate booth encoder 2 is given as follows:

$Pp_{ij} = a_j\sim(b_{2i+1})+\sim(b_{2i+1}) \quad = b_{2i+1}\hat{}a_j\%$ (5)

(0) = entry where 1 is modified to 0. (1) = entry where 0 is modified to 1.The error rate is given by: 8/32= 25. 8 bits out of 32 bits is changed which adds to 25% of error.

3.3. Proposed Booth Approximation 1:

Here, the K map of the R4ABE3 is shown in Table 3.1. The modification is achieved by changing a few 1 to 0 and 0 to 1. The number of modifications made in R4ABE1 is 4 and the number of modifications made in R4ABE2 is 8 whereas the number of modifications made in R4ABE3 is 6. Thus, the R4ABE3 is capable of producing an error rate less than R4ABE2.

Table 3.1. K-map of R4ABE3

$b_{2i+1}b_{2i}b_{2i-1}$/ a_ja_{j-1}	000	001	011	010	110	111	101	100
00	1	0	1	0	1	0	1	0
01	1	0	1	0	1	0	1	0
11	0	1	0	1	0	1	0	1
10	0	1	0	1	0	1	0	1

The output equation of the approximate booth encoder 3 is given as follows:

Ppij=b2i^(b2i-1)^aj^(b2i+1) (6)

The error rate is given by: 6/32= 18.75%.6 bits out of 32 bits has been changed which produces 18.75% of error rate.

3.4. Proposed Booth Approximation 2:

Table 3.2. K-map of R4ABE4

$b_2b_1b_0$/ a_1a_0	000	001	011	010	110	111	101	100
00	0	0	0	0	1	0	1	1
01	0	0	1	0	1	0	1	1
11	0	1	1	1	0	0	1	1
10	0	1	0	1	0	0	1	1

The Error Rate for 4/32=12.5%

Ppij=b[2]|((~a[1]) ((b[2]& ~b[1]& b[0])| (b[2]& b[1]&~ b[0])) | ((a[1]) (~(b[2]& ~b[1]& b[0])| (~b[2]& b[1]&~ b[0])) | (a[0]& (~(b[2]& b[1]& b[0])) (7)

3.5. Proposed Booth Approximation 3:

The error rate is given by: 4/32.4 bits out of 32 bits is changed and it produces 12.5% of error rate.

International Journal of Electrical Electronics & Computer Science Engineering
Volume 6, Issue 1 (February, 2019) | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

ppij=((a[1]&~b[2])&(b[0]|b[1]))|((~a[1]&b[2])&((b[1]&~b[0])|~b[1]))) (8)

The truth table illustrating the change of bits is shown in Table3.2.

Table 3.2. K-map of R4ABE5

$b_2b_1b_0$/ a_1a_0	000	001	011	010	110	111	101	100
00	0	0	0	0	1	0	1	1
01	0	0	0	0	1	0	1	1
11	0	1	1	1	0	0		0
10	0	1	1	1	0	0		0

Table 3.1. Comparison of Performance Parameters

Design	No of Approximated Bits	Error Rate=N/32	Cell	Cell Area	Delay (ps)	Leakage Power (nW)	Switching Power (nW)	Total Power in nW
Exact	0	0	312	5691	16381	218.12	337234	337452
R4ABE1[9]	4	12.5	211	4228	16179	178.671	248582.79	248767.461
R4ABE4(Proposed)	4	12.5	237	4045	16128	147.66	184880	185027.688
R4ABE5(Proposed)	4	12.5	211	3769	16116	123	226957.8	227081.362
R4ABE3(Proposed)	6	18.75	153	2874	16002	109	115634.98	115743.09
R4ABE2[9]	8	25	102	1956	15590	72	106383.5	106461

IV. PERFORMANCE COMPARISON

The partial products are generated using the Booth encoding algorithm and are accumulated using an exact 4-2 compressor. Radix 4 booth encoders of different types were designed; one using exact irregular partial product array and the others using inexact regular partial product array. Finally, the accumulated partial products were added using the fast adder, ie., the carry-look ahead adder. The approximate booth multipliers implemented in 180 nm technology file Cadence EDA Tool and performance of various parameters is compared in Table 4.1.

V. CONCLUSION

This paper presents a highly efficient method of multiplication- "ASIC Design of Error Tolerant Radix 4 Booth Multiplier" based on Booth encoding methods. This Booth Algorithm can be used for Low Power VLSI Techniques for Digital Filter for Hearing aid applications. With the inherent concept of Inexact computing, it can be used as a low power multiplier which has a minimal delay in operation. It is observed that the static and dynamic power consumed by the circuits is much lesser than the conventional circuits.

VI. REFERENCES

[1] S. Kuang, J. Wang, and C. Guo, "Modified Booth multiplier with a regular partial product array," IEEE Trans. Circuits Syst. II: Express Briefs, Vol. 56, pp. 404–408, May 2009.

[2] J.-P. Wang, S.-R. Kuang, and S.-C. Liang, "High-accuracy fixed-width modified Booth multipliers for lossy applications," IEEE Trans. VLSI Systems, vol. 19, no. 1, pp. 52–60, 2011.

[3] Y.-H. Chen, C.-Y. Li, and T.-Y. Chang, "Area-effective and power efficient fixed-width Booth multipliers using generalized probabilistic estimation bias," IEEE J. Emerging and Selected Topics in Circuits and Systems, Vol. 1, no. 3, pp. 277–288, 2011.

[4] J. Han and M. Orshansky, "Approximate computing: an emerging paradigm for energy-efficient design," Proc. 18th IEEE European Test Symposium,2013, pp.1-6.

[5] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Computers, vol. 63, pp.1760-1771, Sep. 2013.

[6] C.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," Proc. IEEE Int. Conf. Computer Design, 2013, pp. 33–38.

[7] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," Proc. Design Automation and Test in Europe, 2014, pp. 95-98.

[8] H. Jiang, J. Han, F. Qiao, F. Lombardi, "Approximate radix-8 Booth multipliers for low-power and high performance operation", IEEE Trans. Computers, vol. 65, pp. 2638 – 2644, Aug. 2016.

[9] W. Yeh and C. Jen, "High-speed Booth encoded parallel multiplier design," IEEE Trans. Comput., vol. 49, pp. 692–701, Jul. 2000.