

RELIABLE AND ACCURATE LOW DENSE EFFICIENT MULTIPLIER

1. CHEEPURI SAI SURYA SRI, 2. GANGA RAMESH J

1. M.Tech, Dept. of ECE, V S Lakshmi Engineering College for Women, Kakinada, A.P
2. ASSOCIATE PROFESSOR, Dept. of ECE, V S Lakshmi Engineering College for Women, Kakinada, A.P

ABSTRACT:

Consumption of Energy is the major factor, in the various processing application like DSP, ASIC, and FPGA. The motive of this work is to approximate the multiplication process. The multiplier operands are rounding off to the two power N format which is nearest to the input values. With a small penalty of error, the speed and energy considerably increased. Literature survey reveals that earlier works are based on modifying the structure or complexity reduction of a specific accurate multiplier. This multiplier leads to better error rate when compared with other multipliers. So the rounding based inexact multiplication provides high speed and energy efficient for various processors. The area, speed, and timing analysis are performed for this approach and for some existing accurate and approximate multipliers. The proposed 8-bit RoBA multiplier multiplication offers better efficiency in energy consumption when compared with other existing accurate and approximate multipliers. Further this project is enhanced by using Radix-8 modified booth encoding algorithm.

KEYWORDS: Digital signal processing (DSP), Round based approximation (ROBA), Radix, Modified booth encoding,

INTRODUCTION:

Energy minimization is major requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. It is extremely desired to attain this minimization with minimal performance (speed) penalty [1]. Digital signal processing (DSP) blocks are most wanted in transportable components for realizing various multimedia applications. The computational core of these blocks is the ALU where the multiplications and additions are the major part [6]. The multiplications plays foremost operation in the processing elements which can leads to high consumption of energy and power. Many of the DSP cores implement image and video processing algorithms where final outputs are either images or videos prepared for human consumptions. It facilitates to go for approximations for improving the speed and energy in the arithmetic circuits. This originates from the limited perceptual abilities in observing an image or a video for human beings. In addition to the image and video processing applications, there are other areas where the exactness of the arithmetic operations is not critical to the functionality of the system (see [2],[3]). Approximate computing provides an accuracy, speed and power/energy consumption. The advantage of approximate multiplier reduces the error rate and gain high speed. For correcting the division error compare operation and a memory look up is required for the each operand is required

which increases the time delay for entire multiplication process [4]. At various level of abstraction including circuit, logic and architecture levels the approximation is processed [5]. In the category for approximation methods in function, a number of approximating arithmetic building blocks, such as adders and multipliers, at different design levels have been suggested in various structures [6],[7]. Broken array multiplier was designed for efficient VLSI implementation[8]. The error of mean and variance of the imprecise model increase by only 0.63% and 0.86% with reverence to the precise WPA and the maximum error increases by 4%. Low-Power DSP uses approximate adders which are employed in different algorithms and design for signal processing. In contrast with standard multiplier, the dissipated power for the ETM dropped from 75% to 90%. While maintaining the lower average error from the conventional method, the proposed ETM achieves an impressive savings of more than 50% for a 12 x 12 fixed-width multiplication. The crucial part of the arithmetic units are basically built by the multiplier hardware, so multipliers play a prominent role in any design. [1] If we consider a Digital signal processing (DSP) the internal blocks of arithmetic logic designs, where multiplier plays a major role among other operations in the DSP systems [1]. So, in the design of multiplier and accumulate unit (MAC) multipliers play an important role. Next,

important design in the MAC unit is the Adder. Adders also share the equal important in this design. By the appropriate function methods different kinds of adders and multipliers designs are been suggested. By the approximate computing the designer can make tradeoffs, accuracy, speed, energy and power consumption. In this paper we proposed the modified form of rounding based approximate multiplier which is low power design, high speed and energy efficient. The multiplier designed was built using the conventional multiplier approach at the algorithm level by considering the rounded input values which are not in the form of 2^n so, we call this multiplier the modified rounding based approximate multiplier. This multiplier can be applied for Signed and Unsigned operations by which three different architectures are implemented. **ENERGY** minimization is one of the main design requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets [1]. It is highly desired to achieve this minimization with minimal performance (speed) penalty [1]. Digital signal processing (DSP) blocks are key components of these portable devices for realizing various multimedia applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems [2]. Therefore, improving the speed and power/energy-efficiency characteristics of multipliers plays a key role in improving the efficiency of processors. Many of the DSP cores implement image and video processing algorithms where final outputs are either images or videos prepared for human consumptions. use approximations for improving the speed/energy efficiency. This originates from the limited perceptual abilities of human beings in observing an image or a video. In addition to the image and video processing applications, there are other areas where the exactness of the arithmetic operations is not critical to the functionality of the system.

LITERATURE SURVEY:

A traditional method to reduce the aging effects is overdesign which includes techniques like guard-banding and gate oversizing. This approach can be area and power inefficient [8]. To avoid this problem, an NBTI-aware technology mapping technique was proposed in [7] which guarantee the performance of the circuit during its lifetime. Another technique was an NBTI-aware sleep transistor in [3] which improve the lifetime stability of the power gated circuits under considerations. A joint logic restructuring and pin reordering method in [6] is based on detecting functional symmetries and transistor stacking effects. This approach is an NBTI optimization method that considered path sensitization. Dynamic voltage scaling and body-biasing techniques were proposed in [4] and [5] to reduce power or extend circuit life. These

techniques require circuit modification or do not provide optimization of specific circuits. Every gate in any VLSI circuit has its own delay which reduces the performance of the chip. Traditional circuits use critical path delays as the overall circuit clock cycle in order to perform correctly. However, in many worst-case designs, the probability that the critical path delay is activated is low. In such cases, the strategy of minimizing the worst-case conditions may lead to inefficient designs. For noncritical path, using the critical path delay as the overall cycle period will result in significant timing waste. Hence, the variable latency design was proposed to reduce the timing waste of traditional circuits. A short path activation function algorithm was proposed in [16] to improve the accuracy of the hold logic and to optimize the performance of the variable-latency circuit. An instruction scheduling algorithm was proposed in [17] to schedule the operations on nonuniform latency functional units and improve the performance of Very Long Instruction Word processors. In [8], a variable-latency pipelined multiplier architecture with a Booth algorithm was proposed. In [9], process-variation tolerant architecture for arithmetic units was proposed, where the effect of process-variation is considered to increase the circuit yield. In addition, the critical paths are divided into two shorter paths that could be unequal and the clock cycle is set to the delay of the longer one. These research designs were able to reduce the timing waste of traditional circuits to improve performance, but they did not consider the aging effect and could not adjust themselves during the runtime. A variable-latency adder design that considers the aging effect was proposed in [2] and [1].

ROUNDING BASED MULTIPLIER AND ITS INACCURACY (ROBA):

The main concept of conventional rounding based approximate multiplier [1] is selecting the rounded values for both the inputs which are in form of 2^n and both the inputs should be in the form of $3 \times 2^{p-1}$ (p is considered as arbitrary positive integer value which is greater than 1) in this case of the conventional approach the final value obtained by the multiplier would be less or more than the exact result obtained. Depending on the A_r (rounded input value of A) and B_r (rounded input value of B) respectively and the result obtained is inaccurate. The motive behind this approximate multiplier is to make use of the ease of operation of power n (2^n). To elaborate on the process of the approximate multiplier, first, let us denote of the input of A and B rounded value by A_r and B_r , respectively. The multiplication of A by B can be write as $A \times B = (A_r - A) \times (B_r - B) + A_r \times B + B_r \times A - A_r \times B_r$ --- (1) Key observation is to facilitate the multiplications of $A_r \times B_r$, $A_r \times B$, and $B_r \times A$ may be implemented just by the operation of shifting which is publicized in the eqn (1). The hardware implementation $(A_r - A) \times (B_r - B)$, however, is rather complex. The weight of

this term in the concluding result, depends on differences of the exact numbers from their rounded ones, is typically small. Hence, it is proposed to omit this part from $(A_r - A) \times (B_r - B)$, helping simplify the multiplication operation shown in the eqn (2). Hence, to perform the multiplication process, the following expression is used $A \times B = A_r \times B + B_r \times A - A_r \times B_r \div 2$. While both values lead to same effect on the accuracy of the multiplier, selecting the larger one (expect for the value $p=2$) leads to a smaller hardware implementation for determining the nearest rounded value. It originates from the detail that the number in the composition of $3 \times 2^{p-2}$ considered as do not care in the both rounding process up and down manner, and smaller logic expressions may be achieved. With the help of accurate and approximate equation the proposed architecture can be designed. Fig provides the detail block diagram for the RoBA multiplier which is applicable for the two processing such as unsigned multiplication, signed multiplication. If the operation is for unsigned multiplication the sign detector and sign set is disabled which can speed up the multiplication process. The two inputs are provided to the detector block which detects MSB of the input and it is provided to the sign set block to denoted signed or unsigned multiplication. Rounding and shifter are worn to reduce the operands value to the nearest power of 2 and it can be shifted with the help of barrel shifter. There are 3 levels of shifter for the following terms obtained in the approximate equation. The kongee stone adder is used to add the two functions from the shifter. The sign can be set with the help detector block. If the output is negative the error value is calculated by inverting the output equation and it is added with binary value of 1. It supposed to be noted that contrary to the previous work where the approximate result is lesser than the exact result, the final result calculated by the RoBA multiplier may be either larger or lesser than the exact result depending on the magnitudes of A_r and B_r compared with those of A and B , respectively. Note that if one of the operands (say A) is lesser than its equivalent rounded value while the other operand (say B) is larger than its equivalent rounded value, then the approximate result will be larger than the exact result. Because the term $(A_r - A) \times (B_r - B)$ will be neglected. Since the differentiation between (1) and (2) is precisely this product, the approximate result becomes higher than the exact one. Similarly, if both A and B are larger or both are lesser than A_r and B_r , then the approximate result is lesser than the exact result. Hence, before the multiplication operation starts, the values of both input are absolute and the output sign of the result are based on the inputs signs be determined and then the operation be performed for unsigned numbers and, at the last stage, the proper sign be applied to the unsigned result.

The inputs are represented in the format of two's complement. First, the signs of the inputs are

determined, and for each negative value, the unconditional value is generated. Next, the rounding block extracts the nearest value for each unconditional value in the form of 2^n . The bit width of the output of this block is n (the most significant bit of the absolute value of an n -bit number is zero for two's complement format). To determine the nearest value of input A , the operands are rounding off to the power of 2 with the help of rounding criteria. There are four cases for selecting final rounded of value from the original input values there are discussed below 1. A_r is high and B_r is low. 2. A_r is low and B_r is high. 3. A_r is high and B_r is high. 4. A_r is low and B_r is low. By selecting the case one, the approximate result is larger when observed with exact. The error rate is the important factors that should be considered while designing the approximate multiplier. The distance between exact and inexact results for the approximate multiplier is calculated before calculating the error rate of the rounding based approximate multiplier. The hardware architectures of the sign detector, rounding, barrel shifter, kongee stone, subtractor and the sign set modules. The RTL architecture for RoBA multiplier is shown in Fig taken by cadence encounter tool 180-nm technology. The sign set block is used to negate the output if the final output is negative valued. To negate values, which have the representation of two's complement, the corresponding circuit based on $X+1$ should be used. To speed up negation operation, one may skip the incrementation process in the negating phase by accepting its associated error. As result. From the case two and three, the approximate result is somewhat larger than the accurate result in contrast with case one. For case four, the approximate result is lower than the exact result. The program should be slightly modified for each one of the cases. The rate or error is extremely low down for case one and four in contrast with other two cases.

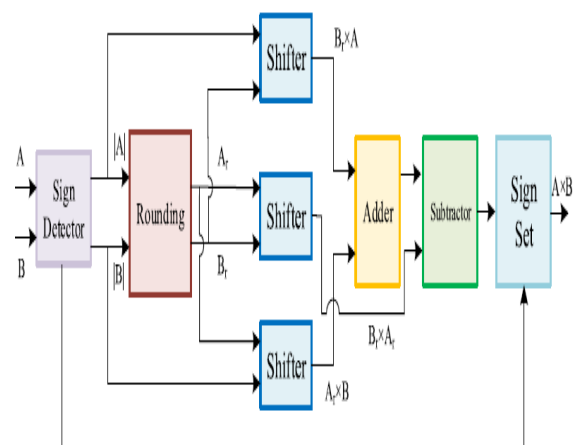


Fig. Block diagram for the hardware implementation of the proposed multiplier.

provide the block diagram for the hardware implementation of the proposed multiplier in Fig where the inputs are represented in two's

complement format. First, the signs of the inputs are determined, and for each negative value, the absolute value is generated. Next, the rounding block extracts the nearest value for each absolute value in the form of $2n$. It should be noted that the bit width of the output of this block is n .

RADIX-8 MODIFIED BOOTH ALGORITHM:

The Booth algorithm consists of repeatedly adding one of two predetermined values to a product P and then performing an arithmetic shift to the right on P .

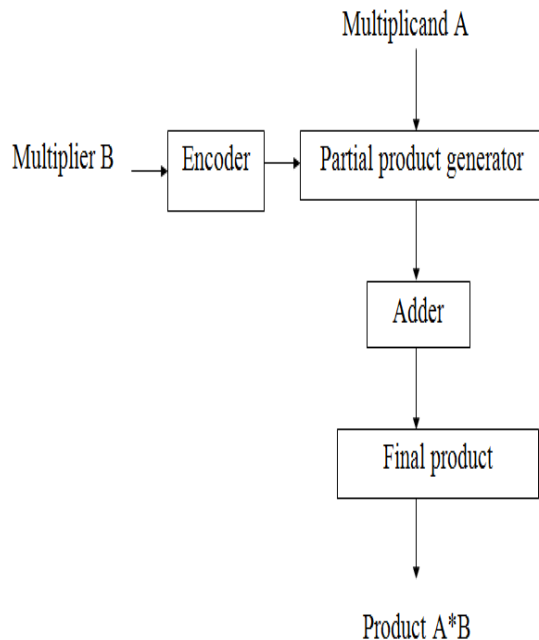


Fig. Booth algorithm

The multiplier architecture consists of two architectures, i.e., Modified Booth. By the study of different multiplier architectures, we find that Modified Booth increases the speed because it reduces the partial products by half. Also, the delay in the multiplier can be reduced by using Wallace tree. The energy consumption of the Wallace Tree multiplier is also lower than the Booth and the array. The characteristics of the two multipliers can be combined to produce a high-speed and low-power multiplier.

The modified stand-alone multiplier consists of a modified recorder (MBR). MBR has two parts, i.e., Booth Encoder (BE) and Booth Selector (BS). The operation of BE is to decode the multiplier signal, and the output is used by BS to produce the partial product.

Then, the partial products are added to the Wallace tree adders, similar to the carry-save-adder approach. The last transfer and sum output line are added by a carry look-ahead adder, the carry being stretched to the left by positioning.

Table . Quartet coded signed-digit table

Quartet value	Signed-digit value
0000	0
0001	+1
0010	+1
0011	+2
0100	+2
0101	+3
0110	+3
0111	+4
1000	-4
1001	-3
1010	-3
1011	-2
1100	-2
1101	-1
1110	-1
1111	0

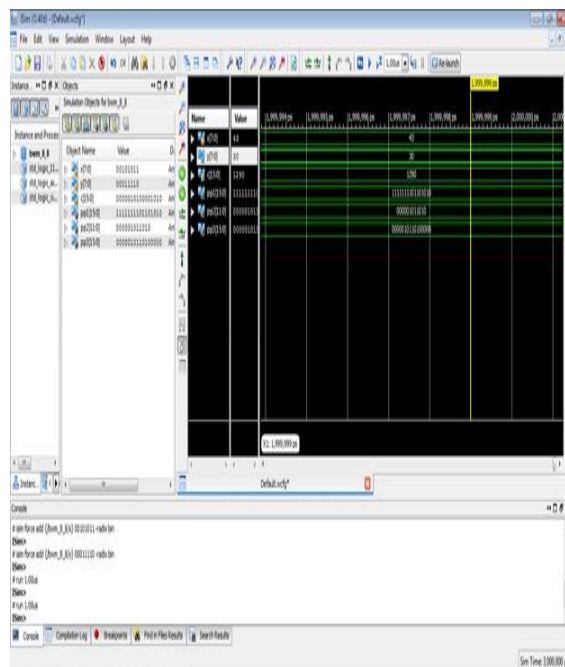
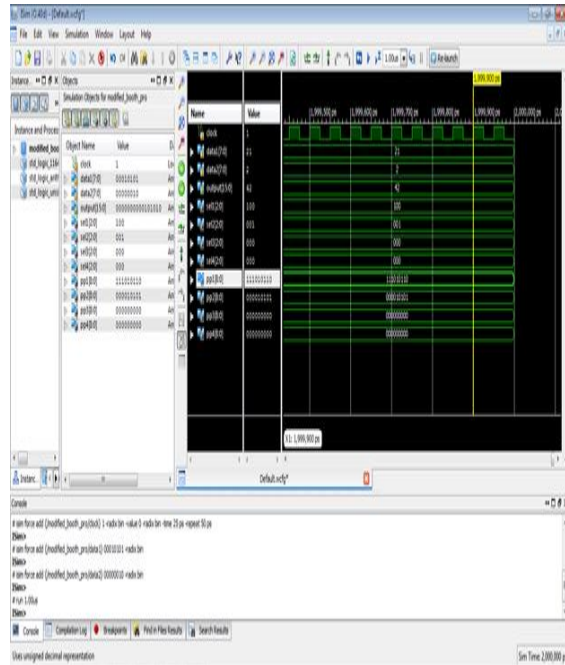
Here we have a multiplication multiplier, $3Y$, which is not immediately available. To Generate it, we must run the previous addition operation: $2Y + Y = 3Y$. But we are designing a multiplier for specific purposes and then the multiplier belongs to a set of previously known numbers stored in a memory chip. We have tried to take advantage of this fact, to relieve the radix-8 bottleneck, that is, $3Y$ generation. In this way, we try to obtain a better overall multiplication time or at least comparable to the time, we can obtain using a radix-4 architecture (with the added benefit of using fewer transistors). To generate $3Y$ with 21-bit words you just have to add $2Y + Y$, ie add the number with the same number moved to a left position.

A product formed by multiplying it with a multiplier digit when the multiplier has many digits. Partial products are calculated as intermediate steps in the calculation of larger products.

The partial product generator is designed to produce the product multiplying by multiplying A by 0, 1, -1, 2, -2, -3, -4, 3, 4. Multiply by zero implies that the product is "0 ". Multiply by "1 "means that the product remains the same as the multiplier. Multiply by "-1" means that the product is the complementary form of the number of two.

Multiplying with "-2" is to move left one as this rest as per table.

RESULT:



CONCLUSION:

High-speed and energy efficient approximate multiplier were proposed. The RoBA multiplier had a high accuracy depend upon the 2n input form. The high exhaustive computation part is neglected to provide high performance. So hardware structural design is designed for S-RoBA, RoBA and AS-RoBA multiplier. The efficiencies of the RoBA multiplier were compared with some existing accurate and approximate multipliers with different parameters. With the help of comparison table, RoBA multiplier provides the better area, power, and energy efficient when compared with some

already proposed accurate and approximate multiplier.

REFERENCE:

- [1] Wen-Chang Yeh and Chein-Wei Jen, "High-speed Booth encoded parallel multiplier design," IEEE Trans. on Computers, vol. 49, issue 7, pp. 692-701, July 2000.
- [2] Jung-Yup Kang and Jean-Luc Gaudiot, "A simple high-speed multiplier design," IEEE Trans. on Computers, vol. 55, issue 10, Oct. pp. 1253-1258, 2006.
- [3] Shiann-Rong Kuang, Jiun-Ping Wang and Cang-Yuan Guo, "Modified Booth multipliers with a regular partial product array," IEEE Trans. on Circuit and Systems, vol.56, Issue 5, pp. 404-408, May 2009.
- [4] Li-rong Wang, Shyh-Jye Jou and Chung-Len Lee, "A well-structured modified Booth multiplier design," Proc. of IEEE VLSI-DAT, pp. 85-88, April 2008.
- [5] A. A. Khatibzadeh, K. Raahemifar and M. Ahmadi, "A 1.8V 1.1GHz Novel Digital Multiplier," Proc. of IEEE CCECE, pp. 686-689, May 2005.
- [6] S. Hus, V. Venkatraman, S. Mathew, H. Kaul, M. Anders, S. Dighe, W. Burleson and R. Krishnamurthy, "A 2GHZ 13.6mW 12x9b multiplier for energy efficient FFT accelerators," Proc. of IEEE ESSCIRC, pp. 199-202, Sept. 2005.
- [7] Hwang-Cherng Chow and I-Chyn Wey, "A 3.3V 1GHz high speed pipelined Booth multiplier," Proc. of IEEE ISCAS, vol. 1, pp. 457-460, May 2002.
- [8] M. Aguirre-Hernandez and M. Linarse-Aranda, "Energy-efficient high-speed CMOS pipelined multiplier," Proc. of IEEE CCE, pp. 460-464, Nov. 2008.
- [9] Yung-chin Liang, Ching-ji Huang and Wei-bin Yang, "A 320-MHz 8bit x 8bit pipelined multiplier in ultra-low supply voltage," Proc. of IEEE A-SSCC, pp. 73-76, Nov. 2008.
- [10] S. B. Tatapudi and J. G. Delgado-Frias, "Designing pipelined systems with a clock period approaching pipeline register delay," Proc. of IEEE MWSCAS, vol. 1, pp. 871-874, Aug. 2005.
- [11] A. D. Booth, "A signed binary multiplication technique," Quarterly J. Mechanical and Applied Math, vol. 4, pp.236-240, 1951.
- [12] M. D. Ercegovac and T. Lang, *Digital Arithmetic*, Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 2003.
- [13] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. On Computers, vol. BC13, pp. 14-17, Feb. 1964.
- [14] M.D. Ercegovac et al., "Fast Multiplication without Carry- Propagate Addition," IEEE Trans. Computers, vol. 39, no. 11, Nov. 1990.
- [15] R.K. Kolagotla et al., "VLSI Implementation of a 200-Mhz 16 _ 16 Left-to-Right Carry-Free Multiplier in 0.35_m CMOS Technology for Next-Generation DSPs," Proc. IEEE 1997 Custom Integrated Circuits Conf., pp. 469-472, 1997.

Copyright @ 2019 ijeearst. All rights reserved.

INTERNATIONAL JOURNAL OF ENGINEERING IN ADVANCED RESEARCH
SCIENCE AND TECHNOLOGY

Volume.02, IssueNo.03, August -2019, Pages: 330-335

- [16] P.F. Stelling and V.G. Oklobdzija, ^aOptimal Designs for Multipliers and Multiply-Accumulators,^o Proc. 15th IMACS WorldCongress Scientific Computation, Modeling, and Applied Math., A. Sydow, ed., pp. 739-744, Aug. 1997.
- [17] Passport 0.35 micron, 3.3 volt, Optimum Silicon SC Library, CB35OS142, Avant! Corporation, Mar. 1998.
- [18] G. Goto et al., ^aA 4.1ns compact 54 _ 54-b Multiplier Utilizing Sign-Select Booth Encoders,^o IEEE J. Solid-State Circuits, vol. 32, no. 11, pp. 1,676-1,682, Nov. 1997.
- [19] G. Goto et al., ^aA 54 _ 54-b Regularly Structured Tree Multiplier,^o IEEE J. Solid-State Circuits, vol. 27, no. 9, Sept. 1992.
- [20] R. Fried, ^aMinimizing Energy Dissipation in High-Speed Multipliers, ^o Proc. 1997 Int'l Symp. Low Power Electronics and Design, pp. 214-219, 1997.
- [21] N.H.E. Weste and K. Eshraghian, Principles of CMOS VLSI Design: A Systems Perspective, second ed., chapter 8, p. 520. Addison Wesley, 1993.