Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, June : 2024

ENERGY AND DENSITY EFFICIENT ROUNDING-BASED APPROXIMATE MULTIPLIER FOR IMAGE ENHANCEMENT

JANGA BHARGAVI, M. Tech, Dept. of ECE, Bonam Venkata Chalamayya Institute of Technology & Science, Bhatlapalem, A.P, India.
Dr KORIMILLI SIRISHA Guide, Dept. of ECE, Bonam Venkata Chalamayya Institute of Technology & Science, Bhatlapalem, A.P, India.

Abstract: This project proposes an efficient approximate multiplier using hybrid algorithm. Many multipliers based on approximate compressors have been developed for error-tolerant applications such as image processing to reduce power, but the combination of them has not been fully satisfied. Proposes a novel 4 gate 4-2 approximate compressor which is complementary with other compressors from earlier work and constructs a hybrid multiplier based on the compressors, a constant approximation, and error correction AND gate. Further this project is enhanced by using Rounding-Base Approximate Multiplier for High-Speed. The proposed 8-bit ROBA multiplier multiplication offers better efficiency in energy consumption when compared with other existing accurate and approximate multipliers. The proposed approach is applicable to both signed and unsigned multiplications. proposed three hardware implementations of the approximate multiplier that includes on for the unsigned and two for the signed operations.

Keywords: Rounding-Base Approximate Multiplier, Partial products, Approximate computing, 4-2 compressor, digital multiplier, compressor combination, error compensation

I**NTRODUCTION:** 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. 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. 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. 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. At various level of abstraction including circuit, logic and architecture levels the approximation is processed. 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. Broken array multiplier was designed for efficient VLSI implementation. 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. 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 .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 trade-offs, accuracy, speed, energy and power consumption.

LITERATURE SURVEY:

A traditional method to reduce the aging effects is overdesign which includes techniques like guard-banding ad gate oversizing. This approach can be area and power inefficient. To avoid this problem, an NBTI- aware technology mapping technique was proposed in which guarantee the performance of the circuit during its lifetime. Another technique was an NBTI- aware sleep transistor in which improve the lifetime stability of the power gated circuits under considerations. A joint logic restructuring and pin reordering method 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 bogy-biasing techniques were proposed 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 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. Fornon critical 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 to improve the accuracy of the hold logic and to optimize the performance of the variable-latency circuit. An instruction scheduling algorithm was proposed to schedule the operations on non-uniform latency functional units and improve the performance of Very Long Instruction Word processors. A variable-latency pipelined multiplier architecture with a Booth algorithm was proposed. 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. Chen et al (2003) presented low-power 2's complement multipliers by minimizing the switching activities of partial products using the radix-4 Booth algorithm. Before computation for two input data, the one with a smaller effective dynamic range is processed to generate Booth codes, thereby increasing the probability that the partial products become zero.

EXISTING METHOD:

UNSIGNED APPROXIMATE MULTIPLIER

If we compel the erroneous output to occur at specific input characteristics, as shown in [8], we can remedy compressor errors with a simple logic gate. The method can be extended to scenarios with more than four faults of the total sixteen input patterns. A new area efficient 4-2 compressor is designed in an effort to make the sum of *S* (sum) and *C* (carry) as close to the exact result as possible and to improve the regularity of the Karnaugh map. Table I shows the proposed compressor's outputs *S* and *C*. The *C* derived from Table I is OR as shown in equation (1). According to De Morgan's laws, the proposed expression of *S* is shown in equation (2) under the assumption of using the fewest number of logic gates possible. It is worth noting that

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, June : 2024

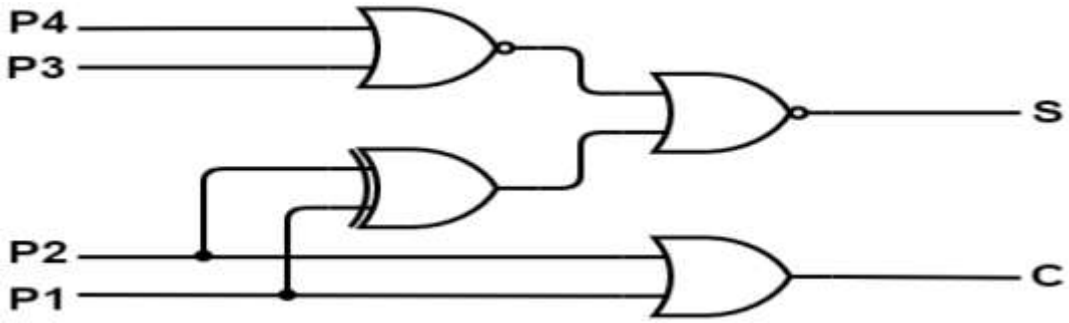

Fig. 1. Architecture of the proposed compressor.

NAND and NOR gates have better performance than AND

and OR gates

$$C = P1 + P2 \quad (1)$$

$$S = (\overline{P1} \cdot \overline{P2} + P1 \cdot P2)(P3 + P4)$$
$$= \overline{(P1 \oplus P2)} + \overline{(P3 + P4)} \quad (2)$$

The architecture of the proposed 4-2 compressor is presented in Fig. 1. This compressor only has two NOR gates and one XOR gate and one OR gate, which makes the compressor consume very few transistors. So even though it has six errors out of sixteen input patterns, the design is still attractive and competitive.

Error Adjustment With Probability Table I shows the probability of each input and the total probability of error is 70/256 ≈ 0.273, which is rather large. The error rates of the plus-one error is 54/256 ≈ 0.211 and the minus-one error is 16/256 ≈ 0.063. Since the probability of being 1 is 1/4 and the probability of being 0 is 3/4 for the initial partial products, the corresponding plus-one error for 0100 and 1000 is very likely to happen. To solve this unfavorable situation, it is essential to avoid using the proposed compressors at the first level, where the inputs of the compressors are directly connected to the initial partial products. The error probability in the first level is modified by using Esposito's approximate compressors in [7] at the first level and the error produced by the proposed compressors in the second level is compensated. The behavior of Esposito's 4-2 compressor is shown in Table I. The output of Esposito's 3-2 or 4-2 compressor will be 11 if two or more inputs are 1. We find that the error probability can be somewhat compensated by combining Esposito's compressors for the initial partial products and the proposed compressor at the second layer. The structure of the proposed 8-bit hybrid approximate multiplier is illustrated in Fig. 2, which combines two distinct approximate compressors and makes use of the error correction module (i.e., AND gate) and constant approximation from [8]. For the final addition, a ripple carry adder (RCA) is employed. As Fig. 2 shows, two types of Esposito's compressors, 3-2 and 4-2, are used in the first level. For Esposito's 4-2 compressor, the probability of outputs $w1$ and $w2$ being 0 are both 135/256 ≈ 0.527 as

shown in Table I. For Esposito's 3-2 compressor, the probability of an output $w1$ being 0 is 36/64 ≈ 0.563, and the other output $w2$ being 0 is 45/64 ≈ 0.703. The four inexact columns with unique input patterns are named $C4$ to $C1$ respectively.

TABLE I TRUTH TABLE OF APPROXIMATE 4-2 COMPRESSOR

Input *p1-p4*	Prob.	Exact	Esposito [7]			Proposed	
		C S	w_2	w_1	*E*	*CS*	*E*
0000	81/256	00	0	0		00	
0001	27/256	01	0	1		01	
0010	27/256	01	0	1		01	
0011	9/256	10	1	1		01	-1
0100	27/256	01	1	0		10	+1
0101	9/256	10	1	1		10	
0110	9/256	10	1	1		10	
0111	3/256	11	1	1	-1	10	-1
1000	27/256	01	1	0		10	+1
1001	9/256	10	1	1		10	
1010	9/256	10	1	1		10	
1011	3/256	11	1	1	-1	10	-1
1100	9/256	10	1	1		10	
1101	3/256	11	1	1	-1	11	
1110	3/256	11	1	1	-1	11	
1111	1/256	100	1	1	-2	11	-1

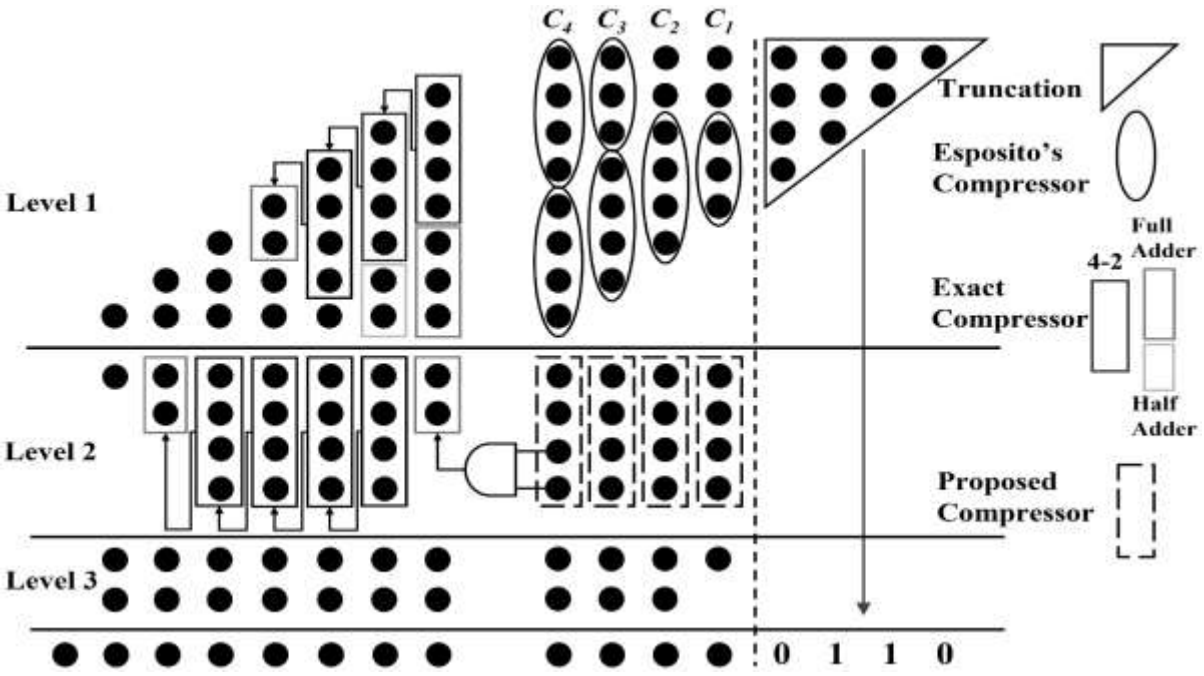

Fig. 2. Existing hybrid approximate multiplier structure

TABLE II PROBABILITY TABLE OF THE PROPOSED COMPRESSOR

Input	Probability					Proposed	
p1-p4	***Origin***	C_4	C_3	C_2	C_1	***CS***	***E***
0000	0.316	0.077	0.110	0.156	0.222	00	
0001	0.105	0.069	0.099	0.140	0.094	01	
0010	0.105	0.069	0.099	0.140	0.173	01	
0011	0.035	0.062	0.088	0.126	0.073	01	-1
0100	0.105	0.069	0.046	0.052	0.074	10	+1
0101	0.035	0.062	0.042	0.047	0.031	10	
0110	0.035	0.062	0.042	0.047	0.058	10	
0111	0.012	0.056	0.037	0.042	0.024	10	-1
1000	0.105	0.069	0.086	0.052	0.074	10	+1
1001	0.035	0.062	0.077	0.047	0.031	10	
1010	0.035	0.062	0.077	0.047	0.058	10	
1011	0.012	0.056	0.069	0.042	0.024	10	-1
1100	0.035	0.062	0.036	0.017	0.025	10	
1101	0.012	0.056	0.032	0.016	0.010	11	
1110	0.012	0.056	0.032	0.016	0.019	11	
1111	0.004	0.050	0.029	0.014	0.008	11	-1
P_{Total}	0.273	0.362	0.355	0.328	0.278		
P_{-1}	0.063	0.223	0.223	0.223	0.130		
P_{+1}	0.211	0.139	0.132	0.104	0.148		

plus-one error and minus-one error at each column are reported in Table II. In Table II, origin means the probability of the corresponding input patterns of compressors at the first level in which initial partial products are directly connected (the same as “Prob.” in Table I). With the usage of Esposito’s compressorsat the first level, although the total error possibility (i.e. PTotal*)* of every single approximate compressor at the second level has been increased, the higher minus-one error possibility and lower plus-one error possibility make the proposed hybrid approximate multiplier more reasonable to use the error correcting AND gate. The proposed hybrid concept is applied to a 16-bit approximate multiplier, where Esposito’s compressors are used at the first and third levels. Two error correcting AND gates are carried out in the second level to reduce the error distance without sacrificing too many resources. Besides, the truncation is adjusted to 6 bits, and the compensation constant is set to the average value of 011110.

PROPOSED METHOD:

ROUNDING BASED MULTIPLIER AND ITS INACCURACY (ROBA):

The main concept of conventional rounding based approximate multiplier is selecting the rounded values for both the inputs which are in form of 2n and both the inputs should be in the form of 3x2p-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 Ar (rounded input value of A) and Br (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 (2n) .To elaborate on the process of the approximate multiplier, first, let us denote of the input of A and B rounded value by Ar and Br, respectively. The multiplication of A by B can be write as A x B = (Ar - A) x (Br – B) + Ar x B + Br x A – Ar x Br ---- 1 Key observation is to facilitate the multiplications of Ar * Br, Ar * B, and Br X A may be implemented just by the operation of shifting which is publicized in the eqn. The hardware implementation (Ar – A) X (Br – 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 (Ar -A) x (Br – B), helping simplify the multiplication operation shown in the eqn. Hence, to perform the multiplication process, the following expression is used A x B = Ar x B + Br x A – Ar x Br ----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 1 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 Ar and Br 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 (Ar − A) × (Br − B) will be neglected. Since the differentiation 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 Ar and Br, then the approximate result is lesser than the exact result. Hence, before the multiplication operation starts, the values of both input are absolutes 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.

2.2 STRUCTURE LEVEL DESIGN OF ROBA MULTIPLIER:

From the equation 1 and 2 the structure level implementation of the multiplier were designed. 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 2n. 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. Ar is high and Br is low. 2. Ar is low and Br is high. 3. Ar is high and Br is high. 4. Ar is low and Br 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 2 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 will be seen later, the impact on the error decreases when an input width increases. If the negation is performed exactly (approximately), the implementation is called signed ROBA (S-ROBA) multiplier [approximate S-ROBA (AS-ROBA) multiplier]. If the inputs are always positive, to speed up and decrease the power consumption, the sign detector and sign set blocks are omitted from the architecture, providing us with the architecture called unsigned ROBA (UROBA) multiplier

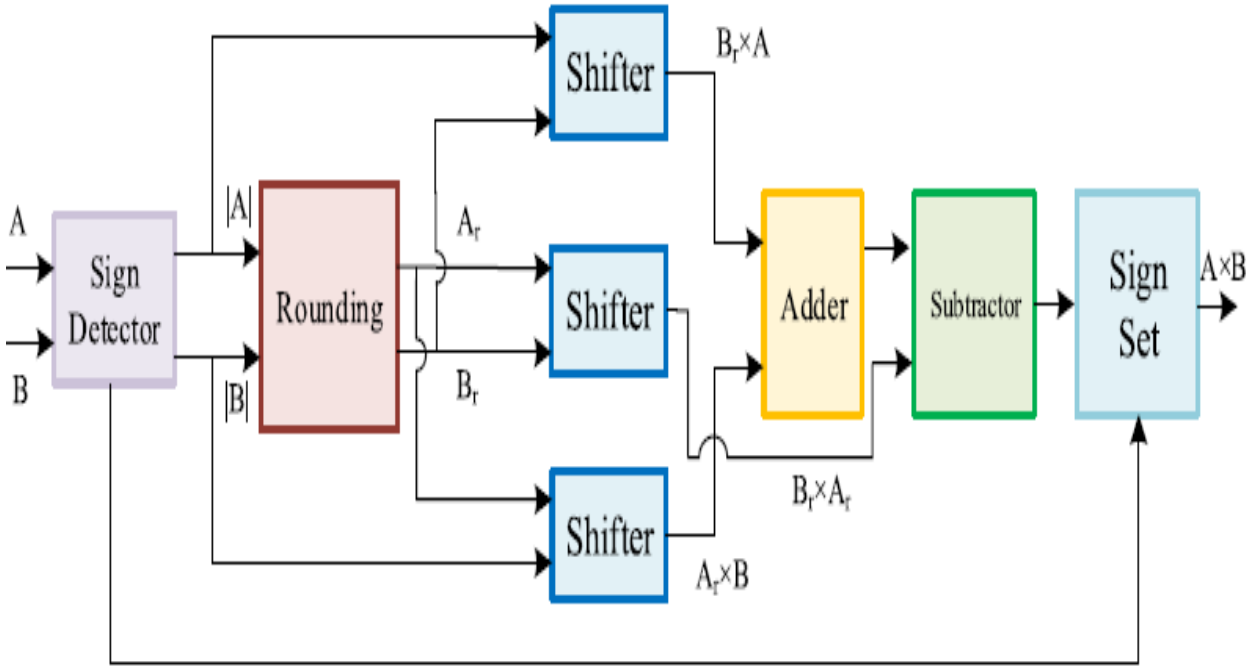

Fig3. Block diagram for the hardware implementation of the proposed multiplier.

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 2^n. It should be noted that the bit width of the output of this block is n (the most significant bit of the absolute value of an n-bit number in the two's complement format is zero). To find the nearest value of input A, we use the following equation to determine each output bit of the rounding block:

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, June : 2024

$$
\begin{aligned}
A_r[n-1] &= \overline{A[n-1]} \cdot A[n-2] \cdot A[n-3] \\
&\quad + A[n-1] \cdot \overline{A[n-2]} \\
A_r[n-2] &= (\overline{A[n-2]} \cdot A[n-3] \cdot A[n-4] \\
&\quad + A[n-2] \cdot \overline{A[n-3]}) \cdot \overline{A[n-1]} \\
&\vdots \\
A_r[i] &= (\overline{A[i]} \cdot A[i-1] \cdot A[i-2] + A[i] \cdot \overline{A[i-1]}) \cdot \prod_{i=i+1}^{n-1} \overline{A[i]} \\
&\vdots \\
A_r[3] &= (\overline{A[3]} \cdot A[2] \cdot A[1] + A[3] \cdot \overline{A[2]}) \cdot \prod_{i=4}^{n-1} \overline{A[i]} \\
A_r[2] &= A[2] \cdot \overline{A[1]} \cdot \prod_{i=3}^{n-1} \overline{A[i]} \\
A_r[1] &= A[1] \cdot \prod_{i=2}^{n-1} \overline{A[i]} \\
A_r[0] &= A[0] \cdot \prod_{i=1}^{n-1} \overline{A[i]}.
\end{aligned}
$$

In the proposed equation, Ar [i] is one in two cases. In the first case, A[i] is one and all the bits on its left side are zero while A[i − 1] is zero. In the second case, when A[i] and all its left-side bits are zero, A[i − 1] and A[i − 2] are both one. Having determined the rounding values, using three barrel shifter blocks, the products $Ar \times Br$, $Ar \times B$, and $Br \times A$ are calculated. Hence, the amount of shifting is determined based on logAr 2− 1 (or logBr 2 − 1*)* in the case of A (or B) operand. Here, the input bit width of the shifter blocks is n, while their outputs are 2n. A single 2n-bit Kogge-Stone adder is used to calculate the summation of $Ar \times B$ and $Br \times A$. The output of this adder and the result of $Ar \times Br$ are the inputs of the *subtractor* block whose output is the absolute value of the output of the proposed multiplier. Because Ar and Br are in the form of 2n, the inputs of the *subtractor* may take one of the three input patterns. where P is $Ar \times B + Br \times A$ and Z is $Ar \times Br$. The corresponding circuit for implementing this expression is smaller and faster than the conventional subtraction circuit. Finally, if the sign of the final multiplication result should be negative, the output of the subtractor will be negated in the *sign set* block. To negate values, which have the two's complement representation, the corresponding circuit based on $\bar{X}$ +1 should be used. To increase the speed of negation operation, one may skip the incrementation process in the negating phase by accepting its associated error. As will be seen later, the significance of the error decreases as the input widths increases. In this paper, if the negation is performed exactly (approximately), the implementation is called signed ROBA (S-ROBA) multiplier [approximate S-ROBA (AS-ROBA) multiplier]. In the case where the inputs are always positive, to increase the speed and reduce the power consumption, the *sign detector* and *sign set* blocks are omitted from the architecture, providing us with the architecture called unsigned ROBA (U-ROBA) multiplier. In this case, the output width of the

rounding block is n +1 where this bit is determined based on $Ar[n] = A[n-1] \cdot A[n-2]$. This is because in the case of unsigned $11x \ldots x$ (where x denotes do not care) with the bit width of n, its rounding value is 10…0 with the bit width of $n + 1$. Therefore, the input bit width of the shifters is $n + 1$. However, because the maximum amount of shifting is $n - 1$, $2n$ is considered for the output bit width of the shifters.

SIMULATION RESULT:

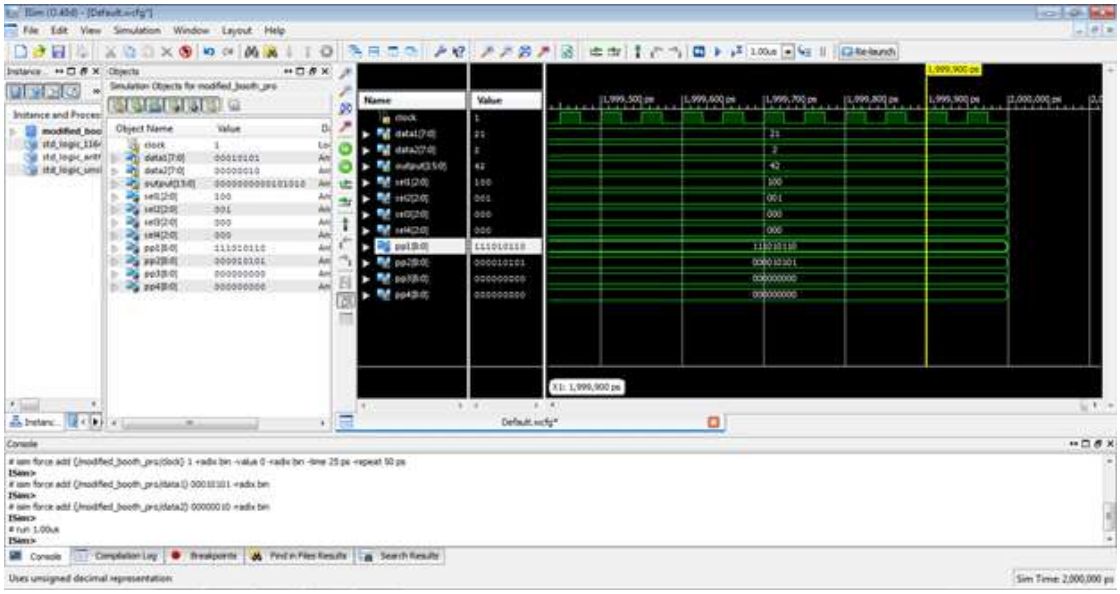

Fig a: Existing method simulation

Above simulation diagram shows output screen shot of existing technique. data1, data2 are as shown as decimal given as inputs , final output is produced at “output” signal. Multiple number of intermediate signals also shown in above simulated waveform.

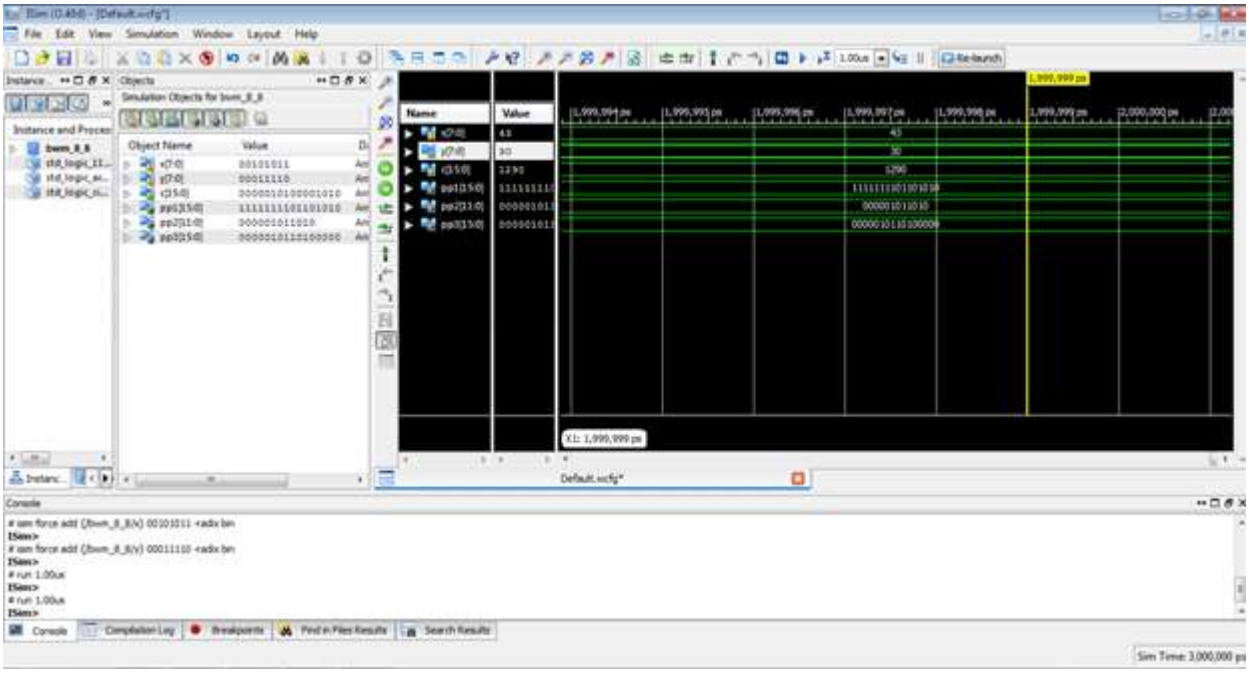

Fig b: Proposed method simulation

Above simulation diagram shows output screen shot of proposed technique. X, Y are as shown as decimal given as inputs , final output is produced at “OP” signal. Multiple number of intermediate signals also shown in above simulated waveform.

CONCLUSION AND FUTURE SCOPE: 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. The proposed approach was applicable to both signed and unsigned multiplications. 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. Further, this project is enhanced by modifying partial product addition structures with advanced adders like csa,csla,cla, to decrease parameters.

REFERENCES

[1] Wen-Chang Yeh and Chein-Wei Jen, “High-speed Booth encoded parallel multiplier design,” IEEE Trans. on Computers, vol. 49, isseu 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. onCircuit 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 mutiplier 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., ªFast Multiplication without Carry- Propagate Addition,º IEEE Trans. Computers, vol. 39, no. 11, Nov. 1990.

[15] R.K. Kolagotla et al., ªVLSI Implementation of a 200-Mhz 16 _ 16 Left-to-Right Carry-Free Multiplier in 0.35_m CMOS Technology for Next-Generation DSPs,º Proc. IEEE 1997 Custom Integrated Circuits Conf., pp. 469-472, 1997.

[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 UtilizingSign-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.

[22] J. Fadavi-Ardekani, ªM_N Booth Encoded Multiplier Generator Using Optimized Wallace Trees,º IEEE Trans. VLSI Systems, vol. 1, no. 2, June 1993.

[23] A.A. Farooqui et al., ªGeneral Data-Path Organization of a MAC Unit for VLSI Implementation of DSP Processors,º Proc. 1998 IEEE Int'l Symp. Circuits and Systems, vol. 2, pp. 260-263, 1998.

[24] K. Hwang, Computer Arithmetic: Principles, Architecture, and Design, chapter 3, p. 81. John Wiley & Sons, 1976.

[25] K.H. Cheng et al., ªThe Improvement of Conditional Sum Adder for Low Power Applications,º Proc. 11th Ann. IEEE Int'l ASICConf., pp. 131-134, 1998.

[26] H. Pei, X. Yi, H. Zhou, and Y. He, “Design of ultra-low power consumption approximate 4-2 compressors based on the compensation characteristic,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 1 pp. 461–465, Jan. 2021.

[27] D. Esposito, A. G. M. Strollo, E. Napoli, D. de Caro, and N. Petra, “Approximate multipliers based on new approximate compressors,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 12, pp. 4169–4182, Dec. 2018.

[28] U. Anil Kumar, S. K. Chatterjee, and S. E. Ahmed, “Lowpower compressor-based approximate multipliers with error correcting module,” *IEEE Embdded Syst. Lett.*, vol. 14, no. 2, pp. 59–62, Jun. 2022.

[29] X. Yi, H. Pei, Z. Zhang, H. Zhou, and Y. He, "Design of an energyefficient approximate compressor for error-resilient multiplications," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2019, pp. 1–5.

[30] M. Ha and S. Lee, "Multipliers with approximate 4-2 compressors an error recovery modules," *IEEE Embdded Syst. Lett.*, vol. 10, no. 1, pp. 6–9, Mar. 2018.

[31] M. Ahmadinejad, M. H. Moaiyeri, and F. Sabetzadeh, "Energy and area efficient imprecise compressors for approximate multiplication at nanoscale," *AEU Int. J. Electron. Commun.*, vol. 110, Oct. 2019 Art. no. 152859.