ACTA IMEKO
ISSN: 2221-870X
June 2022, Volume 11, Number 2, 1 - 6

A modified truncation and rounding-based scalable approximate multiplier with minimum error measurement

Yamini Nagaratnam[1], Sudanthiraveeran Rooban[1]

[1] *Department of ECE, Koneru Lakshmaiah Education Foundation, Green fields, Vaddeswaram,522502, Guntur, AP, India*

ABSTRACT
Multiplication necessitates more hardware resources and processing time. In a scalable method of approximate multiplier, the truncated rounding technique is added to reduce the number of logic gates in partial products with the help of leading one-bit architecture. Truncation and Rounding based Scalable Approximate Multiplier (TOSAM) has few modes of error measurement based upon height (h) and truncated (t) named as (h,t). These multipliers are named as TOSAM(0,2), TOSAM(0,3), TOSAM(1,5), TOSAM(2,6), TOSAM(3,7), TOSAM(4,8), and TOSAM(5,9). Multiplication provides a substantial impact on metrics like power dissipation, speed, size and power consumption. A modified approximate absolute unit is proposed to enhance the performance of the existing approximate multiplier. The existing 16-bit (3,7) error measurement multiplier shows an error measurement value of 0.4 %. The proposed 16-bit multiplier for the same error measurement possesses the error measured value is of 0.01%, mean relative error measured value of 0.3 %, mean absolute relative error measured value of 1.05, normalized error distance measured value of 0.0027, variance of absolute error measured value of 0.52, delay of 1.87 ns, power of 0.23 mW, energy of 0.4 pJ. The proposed multiplier can be applied in image processing. The work is designed in Verilog HDL and simulated in Modelsim, Synthesized in Vivado.

Section: RESEARCH PAPER

Keywords: Approximate multiplier; hardware computation; mean absolute relative error; truncation-based multiplier; rounding operation; absolute error

Citation: Yamini Nagaratnam, Sudanthiraveeran Rooban, A modified truncation and rounding-based scalable approximate multiplier with minimum error measurement, Acta IMEKO, vol. 11, no. 2, article 37, June 2022, identifier: IMEKO-ACTA-11 (2022)-02-37

Section Editor: Md Zia Ur Rahman, Koneru Lakshmaiah Education Foundation, Guntur, India

Received February 7, 2022; **In final form** May 11, 2022; **Published** June 2022

Corresponding author: Sudanthiraveeran Rooban, e-mail: sroban123@gmail.com

1. INTRODUCTION

In a recent technology of digital signal processing application, a multiplier is a priority one with low area and low power utilizations. Nowadays, approximate multiplier is functioning in good manner to reduce area, delay, power, error measurement and energy utilization. As a result of these specifications, approximate computing becomes a trendy trend in the world of digital design 0. Because of the high speed, fault tolerance, and power efficiency, the demand for efficient approximate multipliers is growing. The method of approximation computing encompasses a number of models, including data mining and multimedia processing [2]. Multipliers are critical components in applications like digital signal processing, microprocessors, and embedded systems to accomplish operations like filtering and neural network convolution. These multipliers are made with complicated logic blocks, which increases the amount of energy consumed when the size of the circuit is increased. Because the multiplier is a fundamental component of mathematical units, the configuration of approximation multiplier design has been a research topic for many years [3]. The approximation multiplier is made up of a few basic blocks in which the approximate technique is performed by using any one of the various phases [3]. When it comes to approximation techniques, truncation of partial products is one of the most effective methods for reducing the error by using correction functions [4]. There are various types of error measurement approximate multipliers depending on the operand size. This error measurement is used to overcome the problem of high latency and energy utilization, this work introduced a measurement of scalable approximate multiplier using truncated rounding-based technique which is used to minimize the measure of partial products based on leading one bit position [5]. The proposed error measurement approximate multiplier is of different bitlengths.

2. GENERALISED APPROXIMATE COMPUTING

Approximation computing process is executed at multiple architecture layers, software, and other circuits [6] and the study

of approximate computing is also applied in deep learning. Arithmetic computation is performed by using the design of addition (in some cases can be termed as accumulation) and multiplication and for some applications, for instance as DSP and machine learning. To achieve the power and latency savings, many approximate adders have been developed. Speculative adders and non-speculative transistor-level complete adders are collaborated to create the current approximate adder designs.

The basic four parts of approximate multiplier are:

- Approximation of operands
- Approximation of partial product generation
- Approximation of partial product tree
- Approximation of compressors

2.1. Approximation of operands

Mitchell proposed the concept of a logarithmic multiplier (LM), which uses estimated operands to perform multiplication. The LM performs the operation by changing the operands to approximate logarithmic numbers using shifting and addition operations. Using precise piecewise linear approximation [7] and iterative methodology, the accuracy of contemporary designs of logarithmic multipliers is increased. The usage of estimated operands by the Error-Tolerant Multiplier (ETM) and a Dynamic Range Unbiased Multiplier (DRUM) [8] are the further enhancement in multipliers. The ETM [9] is found with a technique known as multiplier partitioning, which divides a multiplier into accurate multiplication and non-multiplication parts. The Least Significant Bits (LSBs) decide non-multiplication, while the Most Significant Bits (MSBs) determine proper multiplication.

2.2. Approximation of Partial Product generation

To obtain the measured final product, we execute some specific processes, but before that the partial products had to be generated and these undergo some compression operations. The Under Designed Multiplier (UDM), is being forwarded by substituting one entry of the Karnaugh-map based on 2×2 approximation multipliers. The approximation 2×2 multipliers are utilised as fundamental units for bigger size multipliers to yield approximate partial products which are collected by using the correct adder tree. During the partial product accumulation stage, the generalised design of UDM is examined for further utilising carry-in prediction [10]. On approximation booth encoders, a study [11] is carried out. It uses two efficient radix-4 approximation booth encoders.

2.3. Approximation of Partial Product tree

In general, the truncation approach is used for incomplete product trees. The fixed-width multiplier estimates the least significant partial products as unchanged. Some of the least significant columns are omitted in the inexact array multiplier, resulting in constant partial product columns. Among the reduction and rounding strategies, the truncated multiplier that employs the correction constant is chosen. Variable correction is required for truncated multipliers to avoid excessive mistakes.

2.4. Approximation of compressors

Compressors are commonly employed in the construction of high-speed multipliers [12] to accelerate the accumulation of Partial Products (PP). Some error compensation algorithms for fixed-width booth multipliers [13] have recently been proposed, which increases the multipliers accuracy. The error compensation circuit is developed using a simpler sorting network. Several researches have been undertaken on how to determine or identify a number's logarithm and antilogarithm, with the replica being found. Mitchell suggested a simple method for calculating a number's logarithm and antilogarithm, which is then utilised to generate approximate multiplication results (Mitchell multiplier). Although the multiplier is proposed, it falls short of the mark, hence more research has been done to improve the approximation in the measurement of Mitchell-based logarithmic multipliers.

3. PROPOSED APPROXIMATE MULTIPLIER

The proposed approximate multiplier having an error measurement of 16-bit for the rounding and truncation parameters of measurement (3,7) consists of blocks namely Approximate Absolute Unit (AAU), Leading One Detector unit also referred as foremost one detector unit (LOD), Truncation unit (TU), Arithmetic Unit (AU), Shift Unit (SU), Sign and Zero Detector unit (SZD), and is represented in Figure 2.

3.1. Approximate Absolute Unit

The AAU is given by inputs of A and B. If the input operand is negative, the results are inverted; if the input operand is positive, the results are unchanged. This AAU can be removed for unsigned multipliers. It appears as $|A|_{app}$ and $|B|_{app}$, for the measured values of A and B as described in [14].

3.2. Leading One Detector Unit

The LOD unit or foremost one detector unit takes input as $|A|_{app}$ and $|B|_{app}$, values. By using these measurement values the k_A and k_B are detected, which detects the '1' in the MSB. These k_A and k_B are responsible for shifting operation.

3.3. Truncation Unit

The inputs for this TU are measured as k_A and k_B, and also this is having another two inputs which are measured as $|A|_{app}$ and $|B|_{app}$, values. The approximation inputs [15] are trimmed and converted to fixed width operands rely on the leading one position of the input operands. The output is obtained from truncation unit are measured as $(Y_A)_t$ and $(Y_B)_t$ these are given as inputs to the arithmetic unit. The terms $(Y_A)_t$ and $(Y_B)_t$ acquired from the truncated unit is the measured value which is represented in the following equation:

$$TU = 1+(Y_A)_t + (Y_B)_t + (Y_A)_{APX} + (Y_B)_{APX}. \quad (1)$$

3.4. Arithmetic Unit

This AU will perform addition on the truncated fixed width operand as well as the product of approximation input, which is denoted by the value '1' and can be written as TU. It's worth noting that the MSBs of $(Y_A)_{APX}$ and $(Y_B)_{APX}$ are identical to those of measured values of $(Y_A)_t$ and $(Y_B)_t$. Some adders and logical AND gates in the Arithmetic Unit require power gating, this is determined by the operating mode. This is done to improve the design's energy efficiency. The arithmetic block is the same for all the bit lengths.

3.5. Shift unit

The Arithmetic Unit's output must be left shifted by the measured value of $k_A + k_B$ times (k_A and k_B are the leading one-bit values of A and B) The term $2^{k_A+k_B}(1+(Y_A)_t + (Y_B)_t + (Y_A)_{APX} (Y_B)_{APX})$ can be obtained by conducting the shifting operation, as shown in [16]. For the greatest truncation 't' and rounding 'h' values ($h = 5$ and $t = 9$ in this case), the TOSAM multiplier should be developed.

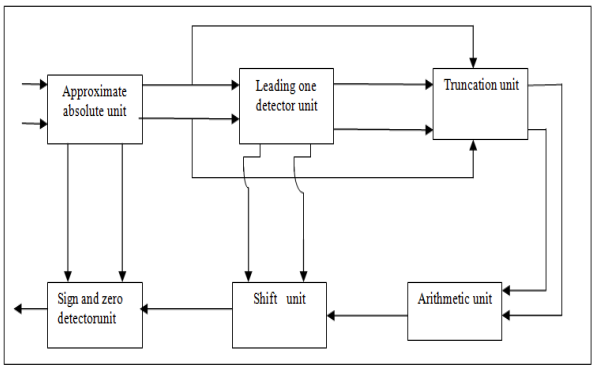

Figure 2. Block diagram of truncated multiplier.

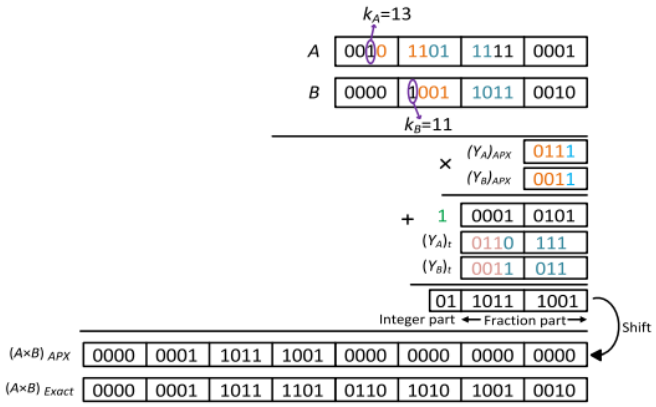

Figure 1. 16-bit TOSAM numerical example of measurement for truncation t = 7 and rounding h = 3.

3.6. Sign and Zero Detector Unit

The output operands sign is determined by the sign of the input operands, and if at least one of the inputs is zero, the output is set to zero. The AAU should be eliminated, due to the unsigned input operands, and the sign and zero detector unit should be restored with a Zero Detector Unit (ZD), as the measurement of the sign unit is unnecessary when the input operands are unsigned.

The proposed approximate absolute unit is implemented in the truncated approximate multiplier. The error measurement values $(Y_A)_{APX}$ ($(Y_B)_{APX}$) are denoted as h + 1 bits in this example. When compared to the exact 16-bit multiplier of measurement of (3,7) in Figure 1, the below dot diagram gives an overview of the procedure for a specific measurement where truncation 't' and rounding 'r' and their values are treated as t = 7 and h = 3 respectively. In the dot diagram below, the green square represents the "1" bit in the measured term $1+(Y_A)_t + (Y_B)_t + (Y_A)_{APX}\ (Y_B)_{APX}$. On the MSB side, the orange circles represent partial products of $(Y_A)_{APX}$ and $(Y_B)_{APX}$, while the purple triangles represent the MSB bits of $(Y_A)_t$ and $(Y_B)_t$. The remaining grey circles and triangles in the dot diagram are not included in the current operations, but they will be considered for future multiplier computations. The measurement of partial multipliers in the approximation multiplier for 16-bit is illustrated in Figure 3.

The process involved in multiplying input operands A by B for a specific measurement of truncation t = 7 and rounding h = 3 is shown in Figure 1. The TOSAM (X, Y) structures, where X and Y correlate to the rounding 'h' and truncation 't' parameters, and the correctness of this multiplier technique is mostly determined by these parameters 't' and 'h'.

As a result, the relationship between these two parameters, 't' and 'h' must be satisfied in order to ensure maximum precision, as well as a speed and energy consumption that is reasonable. Finally, by examining several approaches, this multiplication strategy can be employed for both signed and unsigned operands. To apply this approach to signed multipliers, we must first determine the absolute value of the input operands A and B, as well as the multiplier's sign. Calculation time can be reduced by finding the input operands with exact absolute values.

In the example, the input operand A is 16-bit and has a decimal value of 11761, whereas the input operand B is 16-bit and has a decimal value of 2482. A and B's exact measurement value is written as $(A \times B)_{exact}$ The exact result in binary format is 0000 0001 1011 1101 0110 1010 1001 0010 which is represented in decimal format as 29 190 802, but using the existing method, we get the value as $(A \times B)_{Existed}$ in the binary format is 0000 0001 1011 1001 0000 0000 0000 0000, the decimal format is 28 901 376. The difference between Existed and exact values in this situation is 289 426.

The value of $(A \times B)_{Proposed}$ is calculated using the approximation technique which is explained in Figure 4, the binary format 0000 0001 1011 1001 1111 1111 1111 1111, the decimal format is 28 966911. The difference in between the exact and proposed is 223 891. The K_A and K_B values reflect the leading one-bit location in the input operands A and B. The measured values of K_A and K_B numbers in this case are 13 and 11, respectively. Various (h, t) combinations have a slight modification in the numerical example. Various study is being done to build a new measurement of approximate multiplier. In the Dynamic Segment Method (DSM) [17] design, the input operands are trimmed to 'm' bits depends on the location of the leading one bit (value of that particular position, i.e., 1,2, …), and fixed-width multiplication is implemented on the values, that are obtained from truncation operation. While applying this method of truncation, the produced output value in most of the cases is less than the exact one, resulting in a negative Mean Relative Error (MRE).

When considering digital signal processing applications, strive to keep the mean error as low as feasible to achieve a good Signal-to-Noise Ratio (SNR). The DRUM structure is truncated to yield the solution [18]. We seek to bring the MRE value near zero, the LSB of the shorter input, which is assigned to the value "1," to limit the erroneous outcome. The truncation of the input operands is performed in the multiplication stage in the Low

Figure 3. Representation of measured term $1+(Y_A)_t + (Y_B)_t + (Y_A)_{APX} + (Y_B)_{APX}$ in dot diagram with truncation 't' and rounding parameters 'h' are 7 and 3 respectively.

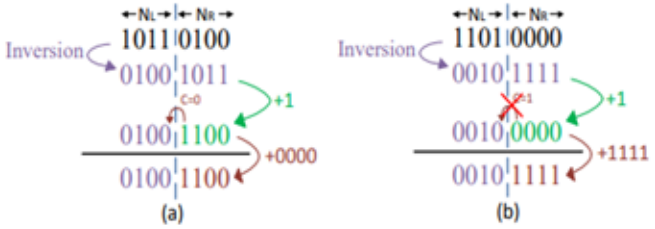

Figure 4. Example for generation of approximate absolute value with two negative numbers.

Energy Truncation-based Approximate Multiplier (LETAM) [19] structure, and there is a chance of omitting half of the partial products.

maxARE is specified as maximum absolute relative error (considered from relative error RE)

MRE is specified as mean relative error

MARE is specified as mean absolute relative error

VARE is specified as variance of absolute relative error

NED is specified as normalized error distance

max_NED is specified as maximum normalized error distance

Comparison of the accuracy of TOSAM against other approximation multipliers like DSM, DRUM, LETAM, and U-RoBA in terms of MRE, maxARE, MARE, VARE, max NED, and NED using random vectors [5] is performed on the parameters of maxARE, MRE, MARE, VARE, max NED, and NED. All these findings are summarised in Table 1.

EDP is defined as energy-delay product

PDA is defined as power-area-delay product

Delay, Power, Area, Energy, EDP, PDA, and MARE of the approximation multiplier are calculated and compared with the existing multiplier design and is tabulated in Table 2. From the data, the proposed modified multipliers show better results than the other existing approximation multiplier configurations with respect to speed and energy usage while maintaining almost identical MARE values. Table 2 shows a comparison between DSM, DRUM, LETAM, U-RoBA with the proposed measurement approach of TOSAM (3,7) approximate multiplier.

4. Results and discussions

The proposed approximation multiplier with output measurement of 32-bit truncated based multiplier that produces a result that is more approximate than the existing multiplier. 11761 is the A value, and 2482 is the B value. The exact values of A and B are as follows: $(A \times B)_{\text{exact}}$ value 29 190 802, and the Existed of A and B is $(A \times B)_{\text{Existed}}$ equals 28 901 376. The difference between Existed and exact values in this situation is 289 426. The value of $(A \times B)_{\text{Proposed}}$ is 28 966911when utilising the proposed approximate technique; the difference between Proposed and exact is 223 891, indicating that the value is more approximate. The output is generated in the next cycle and the error value is also shown in Figure 5.

The Internal structure shows the blocks of the proposed approximate multiplier namely Approximate absolute unit AAU, Lead One Detector LOD, Truncation Unit TU, Arithmetic Unit AU, shifter, sign-set. Also, it represents the flow of data from one block to the other and is shown in Figure 6.

5. CONCLUSION AND FUTURE SCOPE

Low-energy and area-efficient 16-bit approximation multiplier is proposed. Truncation on input operands is performed with two different parameters namely truncation 't' and rounding 'h'. The existing 16-bit multiplier with rounding and truncation measurement (3,7) shows a measurement error of 0.4 %. The proposed 16-bit multiplier for the same truncation and rounding measurement (3,7) with the measured error of 0.01 % (the error is less than 1 %). The error is reduced by rounding the input operands to the next odd value. The

Table 1. Representation of various Approximate Multiplier with maxARE, MRE, MARE, VARE, max NED, and NED.

Architecture	MaxARE (%)	MRE (%)	MARE (%)	VARE (%)	max NED	NED
DSM(3) [20]	36.00	-16.1	16.10	40.43	0.2344	0.0399
TOSAM(0,2) [20]	31.25	-9.1	10.90	46.63	0.3125	0.0309
TOSAM(0,3) [20]	25.00	-3.3	7.61	28.81	0.2500	0.0213
DRUM(3) [8]	56.25	2.1	11.90	79.96	0.2344	0.0281
TOSAM(1,5) [20]	13.89	-0.7	3.95	7.60	0.1250	0.0104
TOSAM(2,6) [20]	6.87	-0.6	2.06	2.00	0.0664	0.0053
Proposed TOSAM(3,7)	3.65	-0.3	1.05	0.52	0.0342	0.0027
LETAM(3) [14]	9.72	-4.0	4.00	2.54	0.0859	0.0104
U-RoBA [15]	11.10	0	2.89	6.37	0.0625	0.0069
TOSAM(4,8) [20]	1.88	-0.2	0.53	0.13	0.0173	0.0013

Table 2. Comparisons of Delay, Power, Area, Energy, EDP, PDA, and MARE of the approximate multipliers.

Architecture	Delay (ns)	Power (mW)	Area (μm²)	Energy (pJ)	EDP (pJ · ns)	PDA (pJ · μm²)	MARE (%)
TOSAM(0,2) [20]	0.74	0.16	342	0.12	0.09	40	10.9
TOSAM(0,3) [20]	0.84	0.21	423	0.18	0.15	76	7.6
DSM(3) [20]	0.97	0.20	344	0.19	0.19	67	16.1
TOSAM(1,5) [20]	1.00	0.35	532	0.35	0.35	185	4.0
DRUM(3) [8]	0.88	0.13	257	0.11	0.10	29	11.9
TOSAM(2,6) [20]	1.00	0.35	532	0.35	0.35	185	2.06
LETAM(3) [14]	1.16	0.39	608	0.46	0.53	278	4.0
U-RoBA [15]	1.05	0.55	1438	0.57	0.60	826	2.9
Proposed TOSAM(3,7)	1.87	0.23	593	0.4	0.748	255	1.05

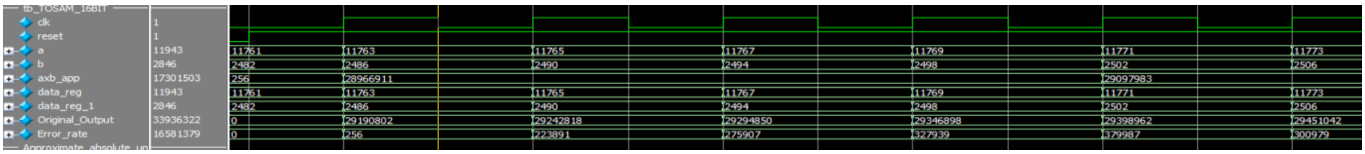

Figure 5. Result of the proposed approximate multiplier with measurement of (3,7).

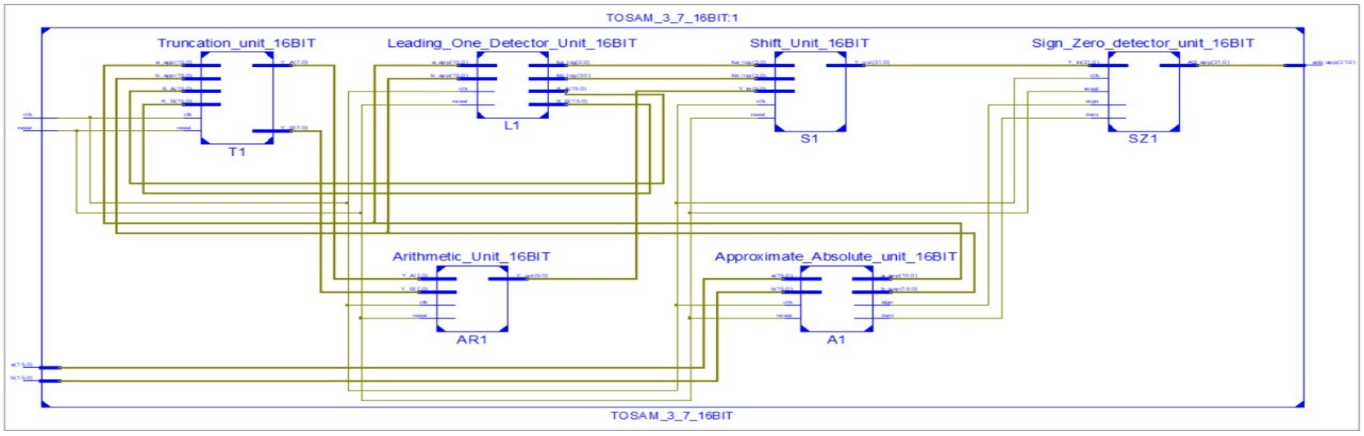

Figure 6. RTL schematic of the proposed approximate multiplier with error measurement of (3,7).

recommended approximation multiplier is scalable and outperforms the correct multiplier in regard of speed, area, and energy. The proposed approximation multiplier consumes 0.23 mW which is lesser than the existing approximate multipliers. Various types of approximate multipliers are used in sharpening the images. In future there is also the possibility of using the multiplier and accumulator unit to create an image sharpening module, and this may be used to measure the energy consumption for various approximate multipliers. Also, in other applications, we can use the JPEG technique to compress many images, and this can be used for approximation multipliers in the discrete cosine transform unit.

REFERENCES

[1] J. Han, M. Orshansky, Approximate computing: An emerging paradigm for energy-efficient design, Proc. 18th IEEE Eur. Test Symp., Avignon, France, 27-30 May 2013, pp. 1–6.
DOI: 10.1109/ETS.2013.6569370

[2] V. K. Chippa, S. T. Chakradhar, K. Roy, A. Raghunathan, Analysis and characterization of inherent application resilience for approximate computing, Proc. 50th ACM/EDAC/IEEE Des. Automat. Conf. (DAC), Austin, TX, USA, 29 May - 7 June 2013, pp. 1–9.
DOI: 10.1145/2463209.2488873

[3] H. Jiang, C. Liu, N. Maheshwari, F. Lombardi, J. Han, A comparative evaluation of approximate multipliers, in Proc. IEEE/ACM Int. Symp. NANOARCH, Beijing, China, 18-20 July 2016, pp. 191–196.
DOI: 10.1145/2950067.2950068

[4] S. Balamurugan, P. S. Mallick, Error Compensation Techniques for Fixed-Width Array Multiplier Design - A Technical Survey, J. Circuits, Syst. Comput., vol. 26, no. 3 (2017), p. 1730003.
DOI: 10.1142/S0218126617300033

[5] A. Momeni, J. Han, P. Montuschi, F. Lombardi, Design and Analysis of Approximate Compressors for Multiplication, IEEE Trans. Comput., vol. 64, no. 4 (2015), pp. 984–994.
DOI: 10.1109/TC.2014.2308214

[6] S. Venkataramani, S. Chakradhar, K. Roy, A. Raghunathan, Approximate computing and the quest for computing efficiency, Proc. 52nd Annual Design Automation Conference (DAC), San Francisco, CA, USA, 8-12 June 2015, Article 120, pp.1-6.
DOI: 10.1145/2744769.2744904

[7] J. Low, C. Jong, Unified Mitchell-Based Approximation for Efficient Logarithmic Conversion Circuit, IEEE Trans. Computers, vol. 64, no. 6 (2015), pp. 1783-1797.
DOI: 10.1109/TC.2014.2329683

[8] S. Hashemi, R. Bahar, S. Reda, DRUM: A Dynamic Range Unbiased Multiplier for Approximate Applications, Proc. IEEE/ACM International Conference on Computer Design, Austin, TX, USA, 2-6 November 2015, pp. 418-425.
DOI: 10.1109/ICCAD.2015.7372600

[9] S. Rooban, S. Saifuddin, S. Leelamadhuri, S. Waajeed , Design of fir filter using wallace tree multiplier with kogge-stone adder, International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6 (2019), pp.92-96.

[10] V. Leon, G. Zervakis, D. Soudris, K. Pekmestzi, Approximate Hybrid High Radix Encoding for Energy-Efficient Inexact Multipliers, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 3 (2018), pp. 421-430.
DOI: 10.1109/TVLSI.2017.2767858

[11] S. Rooban , D. L. Prasanna, K. B. D. Teja, P. V. M. Kumar, Carry Select Adder Design with Testability using Reversible Gates, International Journal of Performability Engineering, vol. 17, no. 6 (2021), pp. 536–542.
DOI: 10.23940/ijpe.21.06.p6.536542

[12] S. Venkatachalam, E. Adams, H. J. Lee, S. B. Ko, Design and Analysis of Area and Power Efficient Approximate Booth Multipliers, IEEE Transactions on Computers, vol. 68, no. 11 (2019), pp.1697-1703
DOI: 10.1109/TC.2019.2926275

[13] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, N. S. Kim, Energy-Efficient Approximate multiplication for Digital Signal Processing and classification applications, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6 (2015), pp. 1180–1184.
DOI: 10.1109/TVLSI.2014.2333366

[14] S. Vahdat, M. Kamal, A. Afzali-Kusha, M. Pedram, LETAM: A low energy truncation-based approximate multiplier, Comput. Elect. Eng., vol. 63 (2017), pp. 1–17.

DOI: 10.1016/j.compeleceng.2017.08.019
[15] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, M. Pedram, RoBa multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 2 (2017), pp. 393–401.
DOI: 10.1109/TVLSI.2016.2587696
[16] M. Ha, S. Lee, Multipliers With Approximate 4–2 Compressors and Error Recovery Modules, IEEE Embedded Syst. Lett., vol. 10, no. 1 (2018), pp. 6–9.
DOI: 10.1109/LES.2017.2746084
[17] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, Approximate Multipliers Based on New Approximate Compressors, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 12 (2018), pp. 4169–4182.
DOI: 10.1109/TCSI.2018.2839266
[18] I. Alouani, H. Ahangari, O. Ozturk, S. Niar, A Novel Heterogeneous Approximate Multiplier for Low Power and High Performance, IEEE Embedded Syst. Lett., vol. 10, no. 2 (2018), pp. 45–48.
DOI: 10.1109/LES.2017.2778341
[19] M. Masadeh, O. Hasan, S. Tahar, Comparative Study of Approximate Multipliers, GLSVLSI'18: proceedings of the 2018 on great lakes symposium on VLSI, 2018, pp. 415-418.
DOI: 10.1145/3194554.3194626
[20] S. Vahdat, M. Kamal, A. Afzali-Kusha, M. Pedram, TOSAM: An Energy-Efficient Truncation- and Rounding Based Scalable Approximate Multiplier, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.27, no.5 (2019), pp. 1161-1173.
DOI: 10.1109/TVLSI.2018.2890712