Design and Implementation of High Speed Dadda Multiplier with Parallel Gates

Narsing Ashok[1], K. Snehalatha[2]

M. Tech Student[1], Department of VLSI Design[2]

[1]J.B.Institute of engineering & Technology. Bhaskar nagar, yenkapally, moinabad(m), Rangareddy (dist.), Hyderabad, Telangana, India

[2]Associate Professor M. E. (E&C) M. Tech. (WMC), J.B.Institute of engineering & Technology. Bhaskar nagar, yenkapally, moinabad(m), Rangareddy (dist.), Hyderabad, Telangana, India

ABSTRACT

The inexact 4:2 compressor proposed in this study has a unique architecture that is optimized for realization using reversible logic. It also includes an inexact Baugh-Wooley Wallace tree multiplier. Measured in scales of Gate Count (GC), Quantum Cost (QC), Garbage Output (GO), and Ancilla Input, the effectiveness of the suggested reversible logic-based realization of the inexact 4:2 compressor and Baugh-Wooley Wallace tree multiplier is examined (AI). This paper proposes an implementation of an 8 8 Baugh-Wooley Wallace tree multiplier. The accuracy metrics MED and MRED are measured for the proposed multiplier and is found to be the least among existing inexact compressor-based multiplier designs.

Keywords — Inexact Wallace tree multipliers, Baugh-Wooley algorithm, inexact 4:2 compressor, reversible logic, image processing, wavelet transform, convolutional neural networks

I. INTRODUCTION

Partial product accumulation stage contributes to the overall delay and hence research has been carried out to optimize this stage to generate the final two terms for stage three using parallel and high-speed accumulation algorithms [1]. Algorithms introduced by Dadda and Wallace [2] have significantly contributed in achieving delay-optimized architectures in accumulation stage.

Delay in accumulation phase is further reduced by using compressors instead of full adders and half adders. The most commonly used compressor topology is the 4:2 compressor as it can realize regularly structured architectures compared to another topology like 5: 3, 7: 2, etc. In the recent times, multipliers are explored [3] in the light of approximation as they find huge demand in realizing area and power optimized design for error tolerant applications like multimedia processing, neural network, signal processing, etc. Recently, such

optimizations in CMOS, FPGA, pass transistor and FinFET based multiplier realizations are being researched.

Akbari have proposed four 4:2 compressors that are configurable between exact and approximate modes. The switching logic between exact and approximate modes incur extra hardware, which creates an overhead in terms of area, even though approximation is introduced [4]. An XOR-less architecture was proposed by Esposito with high ER. However, high ER makes it inefficient in image processing applications. Reddy and Edavoor proposed compressors based on multiplexers. These designs are more appropriate for conventional gate level implementation. In the beginning step, a 64-bit simple text fragment is supplied to an initial permutation (IP) algorithm.

All the above-mentioned architectures are restricted by the physical limitations set by Moore's law. In irreversible computations, for every bit lost, there is KTln2 joules of energy loss. This dissipation was insignificant in higher technologies [5]. As scaling of devices are happening at a rapid rate, the KTln2 joules of energy per bit is becoming crucial and methods are being researched to address this power loss. Hence, newer technologies need to be explored to reduce the power dissipation. Reversible approach in designing circuits and systems is an emerging area of research to address this issue. Bennett's comparative study on conventional irreversible and reversible systems showed that if a reversible model is designed for a circuit/system, the power dissipation is reduced to zero or to negligible amounts.
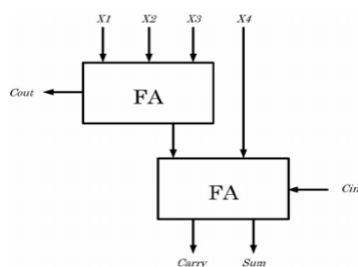

Fig 1: Exact Compressor

The comparative analysis shows that the proposed 8-bit barrel shifter is able to reduce GC, GO and QC, as compared that has the least reversible logic realization parameters in the literature. Khan and Rice introduced a Min-Max algebra-based synthesis technique to realize reversible circuits for ternary logic functions. This circuit mapping is achieved using ternary multiple-controlled unary gates. The proposed technique outperforms existing Ternary Galois Field Sum of Products (TGFSOP) based technique in scales of AI and QC.

Khan proposed a method to realize reversible logic based synchronous sequential circuit with the output functions and state transitions represented using pseudo-Reed Muller expressions. Synchronous sequential circuits such as registers and counters are implemented and is able to achieve an average and reduction in QC and GO compared to the existing replacement technique. Molahosseini have proposed a technique to leverage the parallelism in residue number systems to improve the efficacy of reversible circuit realizations [6]. A parallel-prefix modulo-(2n−1) adder proposed is able to reduce the overhead of QC as compared to regular Brent-Kung adder. Gaur proposed an approach to realize a testable design for an arbitrary circuit by generating modified testable cells from parity preserving logic gates. The efficiency of the circuit is projected in terms of QC, GO and AI.

Dadda proposed an optimization technique using multiple-control Toffoli gate net list. This approach has repeated application of replacement and pair-wise gate merging rules [7]. The proposed technique is tested on reversible benchmark circuits and was able to obtain improvement for QC and GC. Raveendran have proposed reversible logic circuit realization for image kernels for processing/enhancing images. The implementation efficiency of the circuit is measured in terms of QC, GO, AI and GC. Further, the quality of the processed images are measured in terms of SSIM. Raveendran have proposed reversible logic based design for Haar Wavelet Transform (HWT) and lifting for HWT. The authors have proposed

approximate full adder architectures that are optimized for reversible logic with ER and ED [8]. The efficiency of the reversible circuits presented are projected in scales of QC, GC, GO and AI and these parameters of the proposed designs are found to be lesser than the existing approximate adder designs. The efficiency of image processing is measured in terms of SSIM and Peak Signal to Noise Ratio (PSNR). In this research, an imperfect Baugh-Wooley Wallace tree multiplier is suggested as a solution to the problem of realising low power computational units for error tolerant applications. This study introduces a unique architecture for an approximate 4:2 compressor that is consistent with implementation using reversible logic by optimizing the realization parameters for reversible logic (GC, QC, GO, and AI) [9]. This architecture achieves approximation in the multiplier architecture. The proposed inexact compressor-based Baugh-Wooley Wallace tree multiplier is verified for efficiency in image processing and CNN based applications. One level decomposition using rationalized db6 wavelet filter bank and image smoothing applications are performed for the experimental analysis and the efficiency is measured in terms of SSIM. In CNN based application, accuracy of the model is measured to evaluate the efficacy.

II. EXISTING METHOD

In VLSI circuits, heat is a major problem. However, the logic of reversibility results in no heat dissipation at all. It therefore plays a crucial part in nanotechnology, low-energy complementary metal oxide semiconductor architectures, etc. One of the most recent technologies utilizing reversible logic gates is said to be quantum computing. It is being thought about that the limits of conventional technologies will be imposed by the accumulation extending of semiconductor concentration as well as energy deprivation. The erasure of orientational bits under the effect of logic in a regular space or range produces a tremendous amount of power satisfaction.

Results are preserved according to reversible logic. Although there is little hardware involved, this reduces delays. In order to reduce energy dispersion, dissipate heat waves, increase speed, etc., we can employ reversible logic technology. This is done in order to optimize speed while minimizing energy use. The reversible logic gates Fredkin, Peres, Feynmen, and Toffoli gate are only a few examples that we may discuss in this.

1-BIT GATES

NOT gate - a 1-bit gate represented by NOT. It negates the input at the output. It has a QC of 1.

A. 2-BIT GATES

Feynman gate - a 2-bit gate represented by FYG. Input (A, B) produces (A, A $\oplus$ B) at the output of Feynman gate. FYG has a QC of 1.

B. 3-BIT GATES

Toffoli gate - a 3-bit gate represented by TG. Input combination (A, B, C) produces output (A, B, AB $\oplus$ C). It has a QC of 5.

Fredkin gate - a 3-bit gate represented by FRG. Input combination (A, B, C) produces output (A, AB $\oplus$ AC, AB $\oplus$ AC). It has a QC of 5.

BJN gate - a 3-bit gate in which an input combination (A, B, C) produces (A, B, (A+B) $\oplus$ C) at the output. It is represented as BJN and has a QC of 5.

Peres gate - a 3-bit gate represented as PG. For input combination (A, B, C), the output is (A, A $\oplus$ B, AB $\oplus$ C). Peres gate has a QC of 4.

Quantum Cost (QC):

It refers to the number of primitive quantum gates in the circuit.

Gate Count (GC):

It refers to the number of reversible gates in the circuit.

Ancilla Inputs (AI):

It refers to the number of additional inputs included to attain physical reversibility.

Garbage Output (GO)

It refers to the number of additional outputs included to make the circuit reversible.

An optimized reversible circuit synthesis should use minimum ancilla inputs, minimum garbage outputs, minimum gate count and minimum quantum cost. The reversible logic realization of the basic computational units is presented below.

C. Half Adder

Half adders are used to find the sum of two bits and have two inputs (HAIN1 and HAIN2) and two outputs (HASUM and HACARRY). The expression for HASUM and HACARRY are given below.

$HASUM = HAIN1 \oplus HAIN2$

$HACARRY = HAIN1 \cdot HAIN2$

Figure 2 shows the reversible logic gate based implementation of a half adder. A Peres gate is used in which one ancilla input and one garbage output is used.
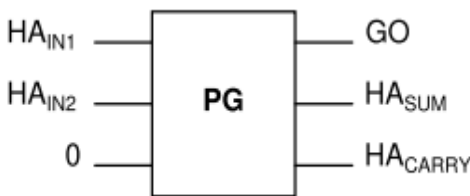
Figure 2: Half adder using Reversible Logic

D. FULL ADDER:

To find the sum of three bits, a full adder is used. Full adder has three inputs (FAIN1, FAIN2 and FACIN) and two outputs (FASUM and FACARRY). FASUM and FACARRY can be expressed below

$FASUM = FAIN1 \oplus FAIN2 \oplus FACIN$

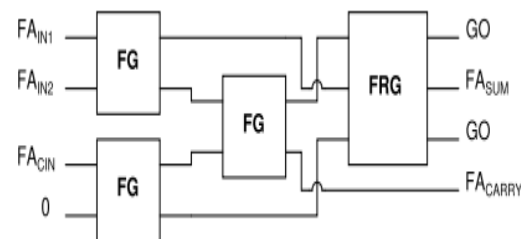$FACARRY = (FAIN1 \oplus FAIN2) \cdot FACIN + FAIN1 \cdot FAIN2$

Figure 3: Full adder using reversible logic

E. Exact 4:2 Compressor:

The reversible logic realization of an exact compressor is presented in Figure 4 and has four

Feynman gates, one NOT gate, two BJN gates and four Peres gates with eight AI and nine GO. Table 4 presents the summary of reversible logic realization parameters for an exact compressor.
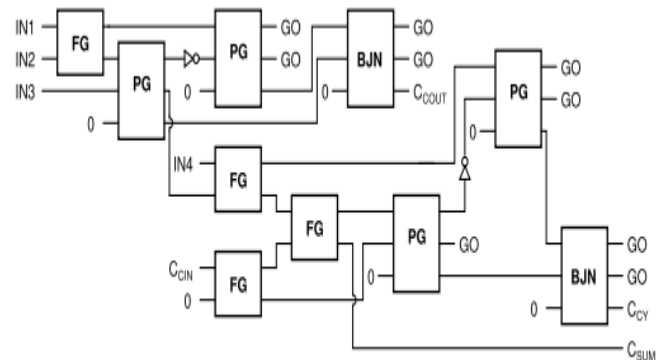

Figure 4: Exact compressor design using Reversible logic

F. Proposed Inexact 4: 2 Compressor

Using reversible logic gates, Figure 5 illustrates the circuit realisation of the proposed inexact 4:2 compressor. With three BJN gates, three Peres gates, six AI gates, and eight GO gates, the proposed 4:2 inexact compressor has twelve gates in total.
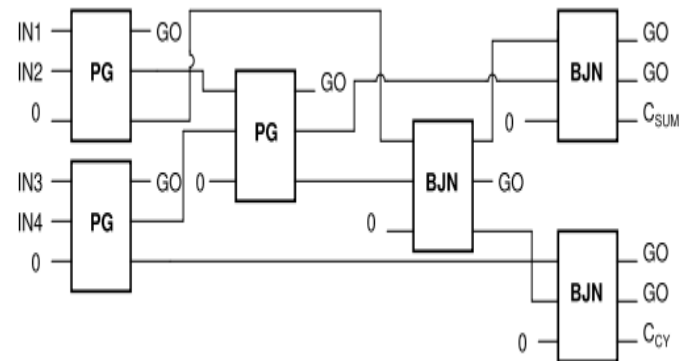

Figure 5: Proposed Inexact 4: 2 compressors

By using these adders and compressors with reversible logic, an 8x8 multiplier is implemented below.

III. PROPOSED METHOD

Multiplication is unquestionably a performance determining operation in AI and DSP applications. These applications demand high speed multiplier architectures to necessitate high speed parallel operations with acceptable levels of accuracy.

Introduction of approximation in multipliers leads to realization of faster computations with reduced hardware complexity, delay and power, with accuracy in desirable levels. Partial product summation is the speed limiting operation in multiplication due to the propagation delay in adder networks. In order to reduce the propagation delay, compressors are introduced. Compressors compute the sum and carry at each level simultaneously. The resultant carry is added with a higher significant sum bit in the next stage.

As a measure to optimize the hardware utilization of the proposed design, this paper proposes an alternate architecture for multipliers with more than three stages of cascaded compressors. In the high-speed area-efficient compressor architecture (as shown in Figure 4), apart from the MUX, one XOR, one AND and two OR gates are required. OR and AND gates each need 8 transistors in CMOS logic implementation. In order to reduce the transistor count, this paper proposes an architecture with NAND and NOR gates. Even though the SUM and CARRY generated by the modified architecture is not as same as that of the proposed 4: 2 compressor architecture, with cascading of the compressor in multiples of 2, the error is nullified.
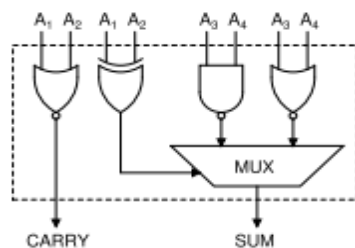

Fig.6 Basic building block for proposed modified Dual-stage 4: 2 compressors

On applying approximation to 4: 2 compressor, output count can be reduced to 2. Approximation is done by eliminating COUT. This incurs an error only when the input combination is '1111'. When the input bits are '1111' the CARRY and SUM are set to '11' and an error of –1 is introduced.

In the majority of MAC modules, multiplier components take up half of the space. Consequently, the design of a low-power VLSI system can benefit from the use of an energy-efficient multiplier.

The suggested technique only approximates the least significant portion of the outcome while precisely performing the higher - level multiplication functions to minimize error severity. Accordingly, considering the effectiveness of numerous approximate multipliers with different sequences of estimated digits, we now recommend using approximate array multipliers with 9-bit of the outcome approximated out from 16-bit in the predicted MAC units as depicted in Figs. Approximating that over 9-bits results in greater improvements in area, power, as well as latency decrement.

However, the quality degradation is also quite significant. In this method, a mac unit is designed using 4 approximate Adder designs for implementing multiplier designs to achieve better area, power and speed. The multiplier, adder, accumulator, and controller are the four parts of the proposed MAC unit. As a result, any of these components can be roughly calculated. The multiplier blocks are the focus of this paper's efforts to increase energy efficiency. Multiplier is a significant arithmetic module in the digital signal processing (DSP) system. It contributes mainly in the power consumption and speed, and efficient multipliers are the need of the hour. Approximate computing has added a unique dimension in the area of digital design by reducing area, power and delay.

The demand of efficient approximate multipliers is enhancing due to the high speed and fault tolerance as well as its power efficiency. In this paper 8bit multiplier designed using proposed approximate compressors.
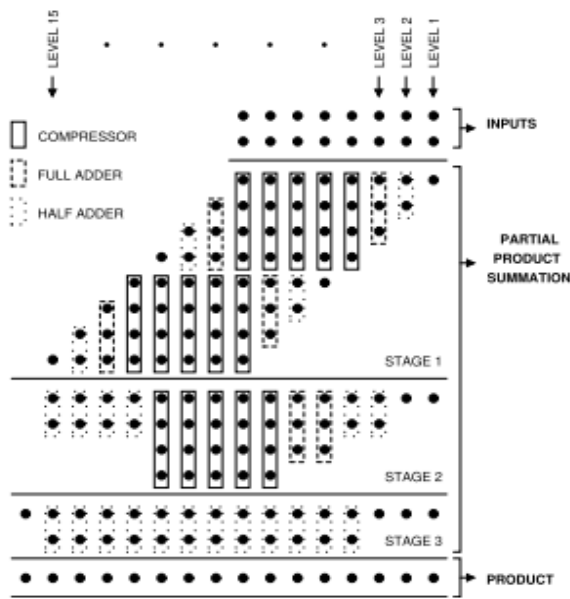
Fig.7 8 × 8 approximate multipliers.

Parallel Prefix adder:

The Ladner-Fischer adder is flexible to speed up the binary addition and the structure looks like tree structure for the high performance of arithmetic operations. Research on binary operation elements and motivation gives development of devices. Field programmable gate arrays [FPGA's] are most popular in recent years because they improve the speed of microprocessor-based applications like mobile DSP and telecommunication. The construction of Ladner-Fischer adder consists of three stages. They are pre-processing stage, carry generation stage, post-processing stage.

Pre-Processing Stage:

In the pre-processing stage, generate and propagate are from each pair of inputs. The propagate gives "XOR" operation of input bits and generates gives "AND" operation of input bits. The propagate (Pi) and generate (Gi) are shown in below equations 1 & 2.

$$Pi = Ai \text{ XOR } Bi \text{ ------ } (1)$$
$$Gi = Ai \text{ AND } Bi \text{ ------ } (2)$$

Carry Generation Stage:

In this stage, carry is generated for each bit and this is called as carry generate (Cg). The carry propagates and carry generate is generated for the further operation but final cell present in each bit operation gives carry. The last bit carry will help to produce

sum of the next bit simultaneously till the last bit. The carry generates and carry propagate are given in below equations 3 & 4.

$$Cp = P1 \text{ AND } P0 \text{ ---------------- } (3)$$
$$Cg = G1 \text{ OR } (P1 \text{ AND } G0) \text{ ------ } (4)$$

The above carry propagates Cp and carry generation Cg in equations 3&4 is black cell and the below shown carry generation in equation 5 is gray cell. The carry propagate is generated for the further operation but final cell present in each bit operation gives carry. The last bit carry will help to produce sum of the next bit simultaneously till the last bit. This carry is used for the next bit sum operation, the carry generate is given in below equations 5.

$$Cg = G1 \text{ OR } (P1 \text{ AND } G0) \text{ ------ } (5)$$

Post-Processing Stage:

It is the final stage of an Ladner-Fischer adder, the carry of a first bit is XORed with the next bit of propagates then the output is given as sum and it is shown in equation 6.

$$Si = Pi \text{ XOR } Ci-1 \text{ ------- } (6)$$
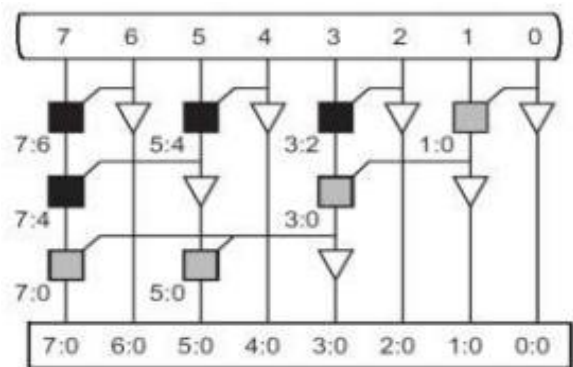

Fig.8 Ladner-Fischer Adder.

III. RESULTS AND DISCUSSION

We have to give force constant values to our design once the waveform window gets open, otherwise the second option we have to provide some different input pairs as a test module and we can run the module directly to cross verify the results.
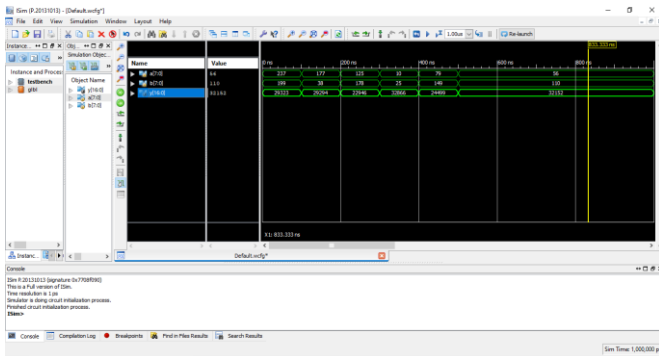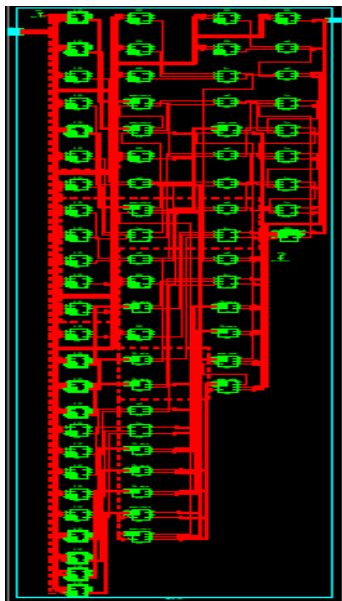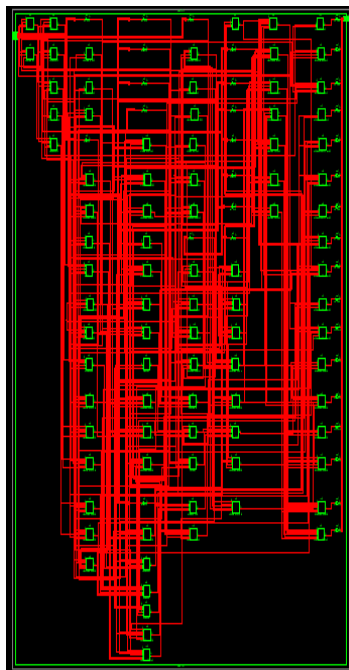
Figure 9. Simulation outcomes for 8x8 multiplier

RTL Schematic

Technology Schematic

Evaluation of Area, Delay report:

	Area	Delay
Existing	97	6.119
Proposed	88	5.944

IV. CONCLUSION

In this paper, a 4:2 inexact compressor again modified with complementary gate that has the least reversible logic realization metrics when compared to reversible logic-based realization of existing state-of-the-art architectures is proposed. By using parallel prefix adder we can reduce the delay and by using complementary gate based compressor we can reduce hardware complexity. From the experimental analysis, it can be concluded that the proposed design is able to achieve the best optimization in scales of reversible logic realization parameters and is able to achieve comparable accuracy metrics with the exact Baugh-Wooley Wallace tree multiplier.

V. REFERENCES

[1]. Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," IEEE Trans. Comput., vol. 44, no. 8, pp. 962–970, Aug. 1995.

[2]. O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 4, pp. 1352–1361, Apr. 2017.

[3]. 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.

[4]. K. Manikantta Reddy, M. H. Vasantha, Y. B. Nithin Kumar, and D. Dwivedi, "Design and

analysis of multiplier using approximate 4- 2 compressor," AEU Int. J. Electron. Commun., vol. 107, pp. 89–97, Jul. 2019.

[5]. P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, "Approximate multiplier design using novel dual-stage 4:2 compressors," IEEE Access, vol. 8, pp. 48337–48351, 2020. [6] A. Gorantla and P. Deepa, "Design of approximate compressors for multiplication," ACM J. Emerg. Technol. Comput. Syst., vol. 13, no. 3, p. 44, May 2017.

[6]. A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for lowpower approximate multipliers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9, pp. 3021–3034, Sep. 2020.

[7]. N. Van Toan and J.-G. Lee, "FPGA-based multi-level approximate multipliers for high-performance error-resilient applications," IEEE Access, vol. 8, pp. 25481–25497, 2020.

[8]. C.-H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 10, pp. 1985–1997, Oct. 2004

Cite this article as :