Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 8(80)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874124

Design of high speed approximate multipliers with inexact compressor adder

B. Sudharani[1*] and G. Sreenivasulu[2]
Research Scholar, Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, Sri Venkateswara University, Tirupati (Andhra Pradesh), India[1]
Professor, Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, Sri Venkateswara University, Tirupati (Andhra Pradesh), India[2]

Received: 18-May-2021; Revised: 14-July-2021; Accepted: 17-July-2021

Abstract
In most practical applications, approximate computation is being used. By using approximate computing, the system performance metrics like area, power and speed can be improved. In this paper an approximate circuit was proposed and developed by modifying the circuit architecture but not the circuit operation. An approximate multiplier using AND-OR logic approximation with Wallace tree reduction, and 3:2 inexact additive designs were proposed for partial product generation and addition. Four different kinds of Approximate Wallace Multiplier (AWM) were implemented using 3:2 compressor adder designs. The concept was discussed, considering an 8×8-bit multiplication as an example. The proposed multipliers achieve substantial improvements in terms of both area and delay. Compared to the conventional multipliers, the AWM1 achieves up to 35.577% reduction in area and 35.224% in delay. AWM2 has an area and delay reductions of up to to 48.077% and 36.532% respectively. AWM3 has area savings of up to 48.077% and delay reductions of up to 46.633%. Finally, the AWM4 has area savings of up to 53.846% and delay reductions of up to 56.482%.

Keywords
Approximate circuits, 3:2 compressor adder design, AND-OR logic, Approximation, Wallace tree reduction.

1.Introduction
Higher computing performance at a lower energy cost is a persistent requirement for emerging applications. The primary idea underlying approximate computing is to replace traditional complex and power-consuming data processing blocks with simpler, few gate ones. The chip area is constrained and they reduce power consumption because of the inaccuracies introduced into the processed data. This might improve energy-efficient systems for present and future system trends in specific applications [1, 2].

The architecture of the multiplier comprises 3 main stages, as depicted in *Figure 1:*
a) Partial Product Generation (PPG)
b) Partial Product Reduction (PPR), and
c) Final Addition.

By using the Partial Product Perforation (PPP) method, various approximate multipliers are designed. The PPP technique comprises perforating any two rows from the original partial products generated by the conventional multipliers [3, 4].

Approximate computing
Electronic device utilization has increased in day-to-day life. The idea of digital data computation has made a dramatic effect on our society. On recent computing platforms, they perform the computations precisely depending on application requirements. The design performance and efficiency of computing platforms have grown exponentially in recent years. From an application perspective, all computations are not equally important.

Humans have lower observational abilities for identifying inaccuracies in processing images or videos. Algorithms and exact models are inefficient in these applications, allowing inaccurate computation to the current digital logic circuits by minimizing logic complexity and increasing performance with a trade-off inaccuracy.

*Author for correspondence

This field of study is referred to as approximate or inaccurate computing and is often called error-tolerant computing [5]. Approximate computing is an excellent technique for improving overall performance without compromising accuracy. For the sake of accuracy, approximate computing permits optimizing logic complexity and speed. Someone can tolerate computational error as long as it is small enough to keep the system operating.

Many popular applications, such as image processing and recognition, are tolerant of minor errors [6]. These are computationally intensive applications, and multiplication is their core arithmetic function, allowing them to trade-off computational accuracy for ease.

In many image processing, video processing, and multi-media systems, the real-world inputs are noisy, so there is not much attention to these systems. The clustering and recognition methods employed in data processing are statistical / probabilistic. Because of the statistical/probabilistic nature of the computation, minor errors will not have a noticeable impact on performance. As a result, we may use approximate computing in several applications that can bear some accuracy loss.

The growth in demand on computing platforms expected to continue unabated. The workloads on the computing platform go through a lot of data processing to achieve best results. For enhancing energy efficiency, approximate computing is a prospective method.

Approximate multipliers in approximate computing

In almost all electronic systems, especially portable systems like smartphones, tablets, and other accessories, minimizing power consumption has become one of the fundamental design requirements. It is extremely desirable to achieve this minimization with an inconsiderable amount of performance (speed) penalty possible. Digital Signal Processing (DSP) blocks are major elements of these portable devices, so we use them to implement a variety of multimedia applications.

The Arithmetic Logic Unit (ALU) is the computational core of these blocks, where multiplication and addition are the key parts. The multiplications play a high percentage of operation in the processing elements, which can lead to high energy and power consumption. Power and energy efficiency play an important role in processor efficiency. Despite computing errors are typically undesirable, applications like multimedia, recognition, and data mining may tolerate some errors. Because of the constraints of human perception, these errors are not substantially different in domains like image, audio, and video processing. Additionally, input from the outside world is quite noisy in many DSP systems, limiting the accuracy of the computational results.

An efficient approach for improving computational efficiency while using fewer resources in imprecision-sensitive systems. The basic aim behind approximate computing is to alter traditional complicated & more energy required for information processing modules with low-complexity, low-logic-count one's [5]. As a result, it minimized efficient chip size and energy usage at the risk of an additional inaccuracy in the stored results.

In many of the applications, approximate arithmetic, which involves approximate adders and multipliers, may reduce energy demands, increase speed, reduce expenditure, and improve consistency. Multipliers are vital in calculating units in several aforementioned applications for two important reasons [7]. At first glance, it distinguishes them by their complex logic architecture, which makes them one of the most energy-intensive information processing devices found in today’s microprocessors. Second, compute-intensive applications often perform many multiplication operations.

The functional modifications entail logic reduction approaches, which may be achieved by easing the requirements for accurate Boolean equivalents to minimize power consumption and circuit size. For example, truncating the multiplier product terms may eliminate some of the less effective partial product terms [8]. Energy consumption decreases when more columns we delete, but errors increase. Some other powerful methods are efficient large multipliers that use small inaccurate multiplier blocks [9]. However, despite increasing the size of the multiplier, the approximate small block hierarchy may not considerably decrease the critical path and may propagate additional errors later.

In this study, we designed an approximate multiplier in which partial products are generated and reduced to half using AND-OR logic approximation. Then it is reduced again using the Wallace tree strategy, and finally, the reduced set of partial products is added using 3:2 inexact additive designs [4].

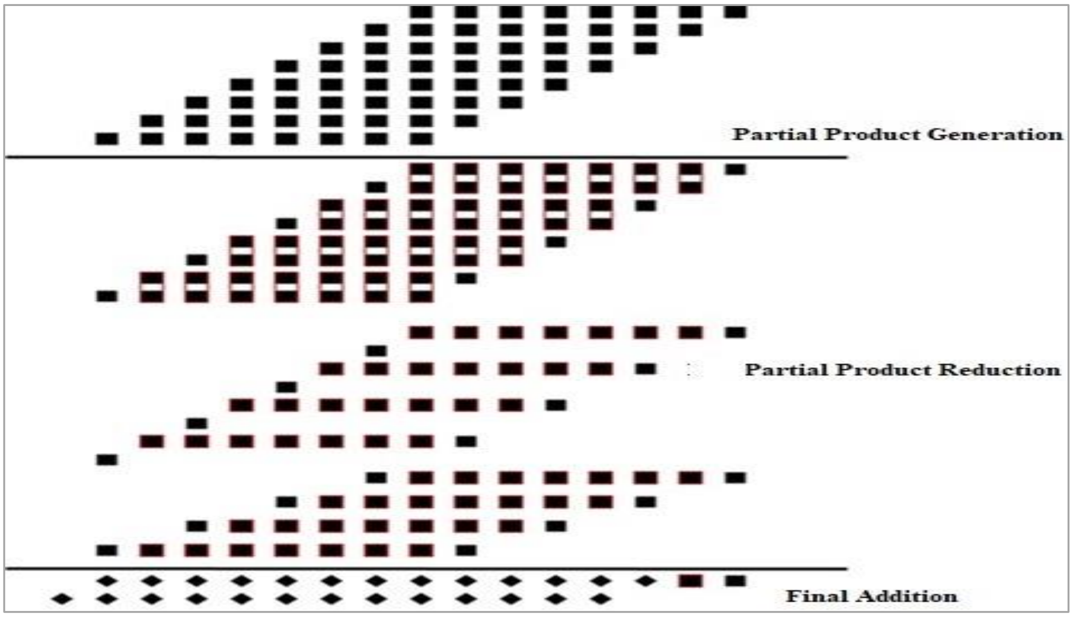

Figure 1 Architecture of a multiplier

2.Literature survey

Gupta et al. [10] developed a novel tree multiplication structure-based design. A divide-and-conquer method was used to produce partially generated products. They have suggested a novel tree structure for adding partially generated products. To improve the energy and power efficiency of ALU, a new reversible logic gate, like Fredkin Gate, was proposed. By combining these modules, they have developed a high-speed, power-efficient ALU.

Momeni, et al. [1] introduced a new design method for compressor-based approximate multipliers, in which they proposed it to manipulate partial product partitions using multiplication recursively. Using 4:2 approximate compressors, four multiplier designs were introduced. Despite approximation reduces accuracy, it yields substantial and faster results while consuming less power; this is desirable for arithmetic circuits. Further simulation results show that the suggested design improves accuracy while decreasing power and delay when compared to the prior approximate design.

Lin and Lin [11] implemented a new inaccurate 4:2 counter, which effectively reduces the partial production stage of the Wallace tree multiplier. This author also designed the error correction module to reduce the error rate or to get accurate results. By using both approximate and exact compressors, they have implemented the proposed design. When tried to compare to the existing Wallace tree multiplier the multiplier that was implemented in this paper will have less area overhead and less delay.

Liu et al. [12] suggested an approximation multiplier for high performance that uses less power and has a short critical path than traditional multipliers. This multiplier uses a newly developed approximate order that confines carry propagation to the nearest neighbors to speed up a partial product accumulation. Accuracy levels may be customized by decreasing errors by varying the number of Most Significant Bits (MSBs) with configurable error recovery. Most errors are negligible since the approximate multiplier has a low mean error distance. This new multiplier has a smaller delay compared to other existing multipliers.

Jiang et al. [13] proposed both unsigned and signed 16×16-bit radix-8 Booth multiplier. The authors also presented a 2-bit adder with a 3-input XOR gate, which reduces gate count and increases overall performance. The authors also proposed circuits for error detection, compensation, and recovery. The parallel processing method of the Wallace tree was used to speed partial product addition. As a result, the presented multiplier decreased the area and delay parameters when compared with the conventional Wallace tree multiplier.

Bhardwaj et al. [14] demonstrated a novel Error-Tolerant Approximate Wallace Tree Multiplier (AWTM) with a small area and low power. For optimal multiplier design, have used the proposed bitwidth aware approximation multiplication method. To reduce the crucial path delay, they employed a carry prediction algorithm. It was further improved by using hardware-efficient carry-in pre-computation. Wallace trees were also used to reduce the delay, power, and area. They had implemented on 4, 8, and 16-bits with high accuracy. The designed multiplier had less area and low power consumption with slightly increasing delay.

Maheshwari et al. [15] proposed a new design method, which uses recursive multiplication to exploit partitions of the partial product of the compressor-based approximate multiplier. A design of four multipliers using approximately 4:2 compressors were proposed. Thorough simulation results show that, compared with previous approximate designs, the proposed design has significantly improved accuracy and reduced power consumption and delay. The image processing application also has a function and shows how efficient the proposed design.

Akbari et al. [16] developed four 4: 2 compressors that can shift between accurate and approximate operating modes. These dual-quality compressors can accomplish high speed and low power usage in approximate mode, but at the expense of accuracy. Each of these compressors has a different accuracy, delays, and power consumption in approximate mode. These compressors provide configurable multipliers whose accuracies can alter dynamically during runtime when they are applied in parallel multiplier systems.

Kulkarni et al. [17] presented a new multiplier architecture with tunable error properties based on a modified inaccurate 2×2 building block. The inaccurate multiplier works better than the correct multiplier in terms of performance and error rate. They showed that their design improves Signal-To-Noise Ratio (SNR) and outperforms the methodologies they presented across several image processing applications.

Anitha et al. [18] designed and implemented a Vedic Multiplier and Reversible Logic Gates in the multiply and accumulate unit. The Urdhava Triyagbhayam sutra was used to develop a Vedic multiplier, and a reversible logic gate was used to develop an adder. Reversible logic was also essential for quantum computation, which was just getting started. For the multiplication unit, the Vedic multiplier was used to minimize partial products and achieve high performance while taking up less space. To get less power, reversible logic was used.

Narayanamoorthy et al. [19] suggested approximate multiplier takes in consecutive bits of a n-bit operand, either starting at the MSB or ending at the Least Significant Bit (LSB) and applies two segments to a m×m multiplier, consuming significantly less power and area.

Pei et al. [20] developed three new, approximate 4-2 multiplier compressors for 8-Bit utilization. In addition, an Error Correction Module (ECM) is suggested for improved error performance of the approximate multiplier during implementing the 4-2 compressors. In this paper, the number of outputs of the approximate 4-2 compressor is decreased to one and the energy efficiency is also improved.

Reddy et al. [3] introduced a novel approximate 4–2 compressor design. They describe the modified Dadda multiplier design to improve the efficiency and reduce the output error of the proposed compressor. A rigorous test study examines the proposed compressor and multiplier efficiency in 45 nm Complementary Metal Oxide Semiconductor (CMOS) standard technologies and compares their parameters to those of existing approximation compressors, yielding a significant reduction in the proposed compressor's error rate.

Esposito et al. [4] developed new approximate compressors and a methodology for exploiting them in the design of effective approximate multipliers. The authors synthesized these approximate multipliers using a 40 nm library for many operand lengths using the proposed method. When compared to previous reporting approximate multipliers, the proposed circuits showed better performance in terms of power or speed for a target precision. The study also contains applications to filter images and adaptive least mean squares filtering.

In approximate computing, a compressor-based multiplier will produce decent results in terms of design and circuit error rate. From the literature review, we can find that there is a scope for improvement in the compressor-based design to improve the speed of the multiplier. Finally, modifying this accurate and approximate based compressors and using it in different multipliers will improve the speed of the multiplier. *Table 1* shows some more studies on approximate multipliers along with the pros and cons.

Table 1 Study of some more works of literature

S.No	Authors	Year	Methods	Pros	Cons
1	Masadeh et al. [21]	2018	Approximate full adders in Partial Product Summation	Tree multiplier designs have lower power consumption, lower delay, and smaller size than array multiplier designs in terms of architecture	The quality loss, increased for higher-order multiplication
2	Arya and Nair [22]	2018	Rounding the operands to the nearest exponent	Both signed and unsigned multiplications are supported.	Implementation of hardware is quite complex
3	Guo et al. [23]	2018	Approximate multiplier design with inexact compressors.	Normalized Mean Error Distance and Mean Relative Error Distance are low and more Hardware Reduction	Higher Error Rate
4	Maddisetti and Ravindra [24]	2018	For partial product reduction, inexact 5:2 compressors are used.	Average Power Decreased (Low Power Consumption)	Propagation delay increased
5	Kiruthika and Suguna [25]	2019	Symmetric bit stacking approach	Multiplication is made easier by reducing the complexity of the Wallace tree structure.	Large Errors in the output
6	Yi et al. [26]	2019	4-2 compressor with a low supply voltage	Under low supply voltage, it achieved the smallest delay and power usage, showing high energy efficiency	Worse Normalized Mean Error
7	Strollo et al. [27]	2020	Partial Product Reduction with error recovery module	Reduced Mean and Maximum Error Distances	Marginally slower
8	Balasubramanian et al. [28]	2021	Vertical or horizontal cuts in an array	Less power-delay product and reduction in critical path delay	Moderately lesser PSNR and SSIM
9	Guo et al. [29]	2018	Two different techniques were used for different bit positions. For lower bit positions OR-based compressor is used, whereas for upper bit positions to carry propagation compressor is utilized	Reduces circuit complexity in Partial Product Accumulation and reduces the critical path delay	Building a large multiplier from 2x2 blocks costs more delay Consumption
10	Sabetzadeh and Moaiyeri [30]	2019	Finfets are used to implement an ultra-efficient imprecise 4:2 compressor at the transistor level	Lower power and propagation delay due to fewer transistors and a shorter critical path.	Ignored Capacitance effects for enhancing design efficiency
11	Venkatachalam et al. [31]	2019	Approximate Booth Multiplier Models with Radix-4 modified Booth encoding technique	Over the existing approximate Booth multipliers, there is a significant increase in accuracy and hardware performance	Error-values increased as the approximation factor increased
12	Senthilkumar et al. [32]	2020	Constrained Cartesian Genetic Programming (CCGP)	Improvement in cell area and maximum power reduction	Only multipliers with fewer gates are considered

3.Challenges and motivation

3.1Challenges

(a) Today and also in the future, the real challenge for the system design is to develop high-speed systems that can perform high-speed and process and transmit the signals with very high signal integrity.

(b) The design of every Very Large-Scale Integration (VLSI) system relies heavily on metrics like area, power, and delay. Multipliers are the central element for most applications and determine overall circuit performance in speed, power consumption, and size.

(c) To improve the speed of the application, the multiplier must comprise less delay. Hence, the goal of this work is to design a multiplier with less delay and compact area.

(d) The multiplier's performance governs the performance of a system because the multiplier is the slowest component in the system and the area consuming most. As a result, optimizing the multiplier speed and area is a critical technology issue.

3.2Motivation

a) For high-performance processors and systems, high-speed multiplication has always been a fundamental requirement.
b) The multiplier is the important structure for energy-efficient processor design, and processor efficiency is determined by multiplier design.
c) Since multipliers are complex circuits and usually have to operate at a high system clock rate, it is vital to fulfilling the overall design to decrease the delay of multipliers.
d) In general, two methods can improve the performance of multiplier in terms of area, power consumption, and. The first one is based on efficient implementation of multiplier function, whereas another relies on the proper selection of logic style for its implementation.
e) In recent decades, several strategies have been suggested in this partial product reduction phase for the realisation of high-speed multipliers. The paper mainly focused on the partial product reduction stage by using approximate adders.

4.Objectives

1. To propose and analyze the novel 8-Bit multiplier architecture with optimized delay and area using the concepts of AND-OR logic compression.
2. To design efficient inexact compressor adder circuits.
3. To design the approximate multipliers using the proposed Inexact Compressor adder circuits.
4. To compare the parameters of conventional multipliers with approximate multiplier designs.

5.Earlier work

A multiplier is an important mathematical unit in a huge number of applications, for two main reasons. Primarily, multipliers are described by a complex logical configuration, being a standout unit for signal processing that requires the most power in current microprocessors. Second, to calculate a concentrated application regularly takes an extensive number of multiplication activities to compute results.

The basic multiplication technique is adding and shift technique. The performance of the parallel multipliers depends on the number of partial products. The multiplication of the multiplicand produces every partial product with one multiplier bit [4]. The partial products are then shifted by their bit orders and afterward added. They often perform the addition with an ordinary carry propagate adder. An array multiplier is one of the foremost basic parallel multiplier circuits. It has a regular design and its pipelined architecture is simple to design. But the major limitation of an array multiplier is its size. As the operand size increases array grows in size at a rate equal to the square of the operand size. This increase in the area results in more power consumption. The propagation delay is high for the internal partial product addition.

To overcome the drawbacks of an array multiplier we may have some approximate multiplication techniques where we have the better performance parameters [13]. That approximate multiplication is proposed in the next section.

6.Methodology for the proposed approximate multiplier designs

The approach for the proposed approximate multiplier designs is shown in the form of a block diagram as shown in *Figure 2*.

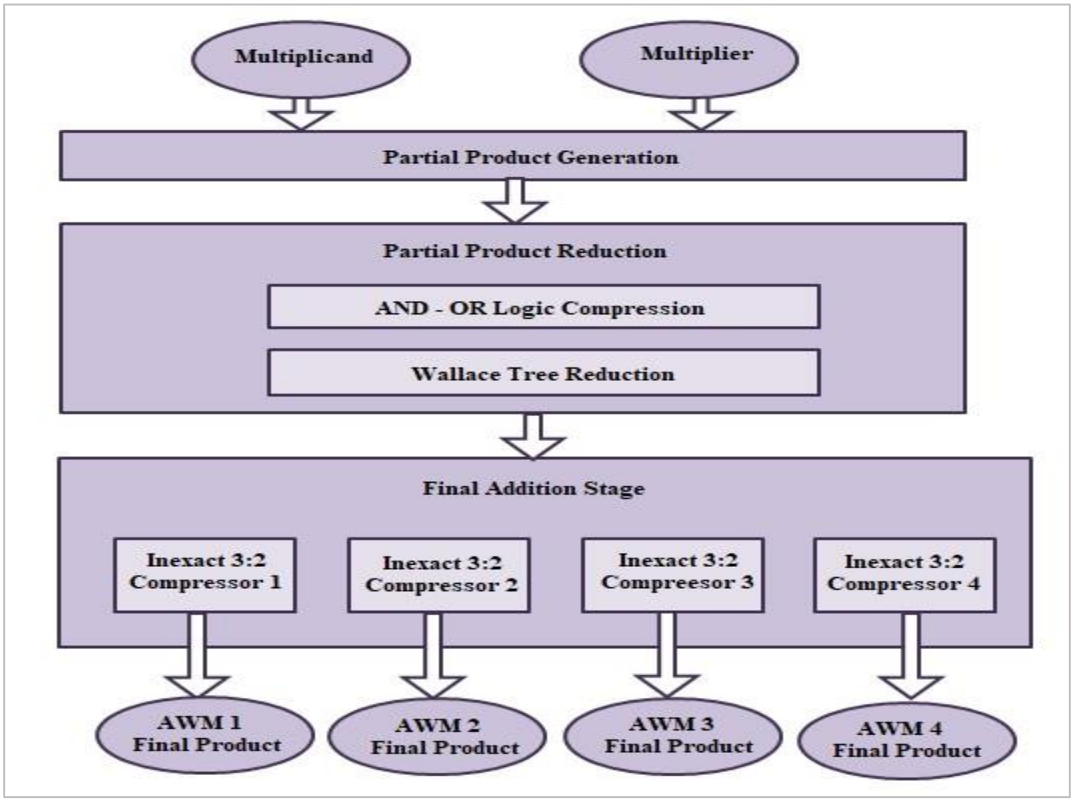

Figure 2 Block diagram of proposed method

6.1Partial product generation

By performing an AND operation between each multiplicand and the multiplier bit, they can easily generate partial products. A row of partial products is formed by ANDing the entire multiplier number with each multiplicand bit. The number of bits in the multiplicand determines the size of the partial product matrix.

6.2Partial product reduction

This method uses a logic approximation to perform lossy compression. The compressed terms are then reduced, remapped, and added to produce the product of the input terms. In an (N×N) multiplier, N2 AND gates can be used in parallel to get the partial product bit-matrix. To get the final product, we accumulate the matrix column-wise using a carry-propagate adder to get the final product. The proposed method begins with the generation of all the partial products making use of the equal number of AND gates just as the existing multiplications. In the partial product matrix, they reduce the number of bits before continuing to the accumulation stage by using lossy logic approximation with AND-OR logic approximation.

6.3AND- R logic compression

An AND gate is used to make a partial product, which is then approximated by half using an array of OR gates. As a result, a reduced set of pre-processed partial product matrices are available for accumulation using any convenient multiplication scheme, such as Carry Select Adder (CSA), Wallace, or Dadda tree. Theoretically, for adding two binary numbers, an OR gate is required. For adding two bits, if their value is high, then the result should be ‘10’ but OR gate gives the result as high. The logic diagram of the AND-OR Logic approximate circuit is given in below *Figure 3* and the truth table is shown in *Table 2*.

The brief impact is to create an energy-efficient approximate multiplier with Reduced Approximate error. Because they handle the most important bits with gradually greater accuracy as one moves down the partial product matrix, the size of the AND-OR logic approximate circuit is reduced. By approximating the most significant bit of partial products it will cause a more approximate error.

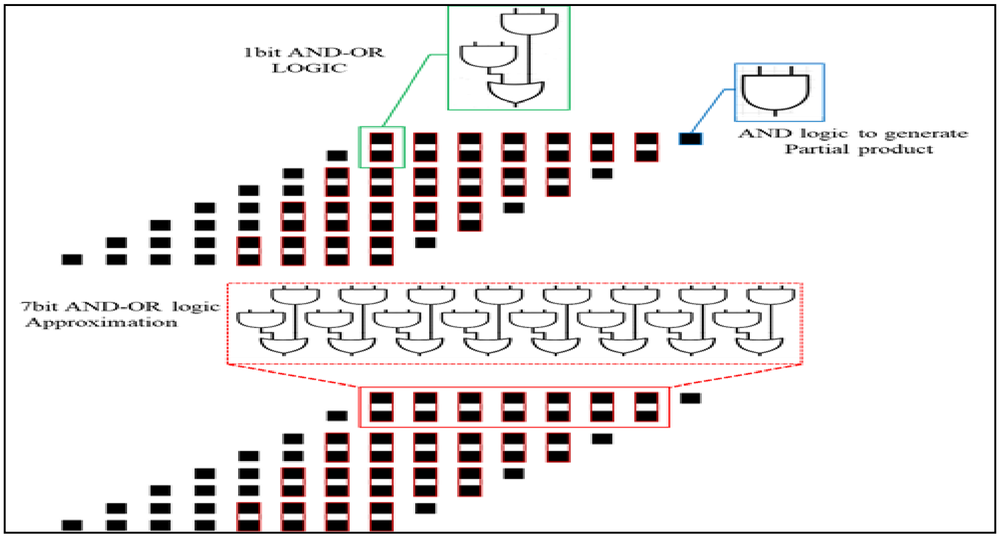

Figure 3 AND-OR logic approximation

Table 2 Truth table of AND-OR logic compression

Inputs		Or Output	Half adder output	
A	B	Y	S	C
0	0	0	0	0
0	1	1	1	0
1	0	1	1	0
1	1	1	0	1

6.4 Wallace tree reduction

The partial products have been reduced after the AND-OR logic approximation as shown in *Figure 4*. It can be again reduced using the Wallace tree reduction technique.

The partial product conditions were remapped based on the bit's commutative property, so we group bits of the same weight in the same column. Because there are fewer rows, the critical path delay is significantly reduced.

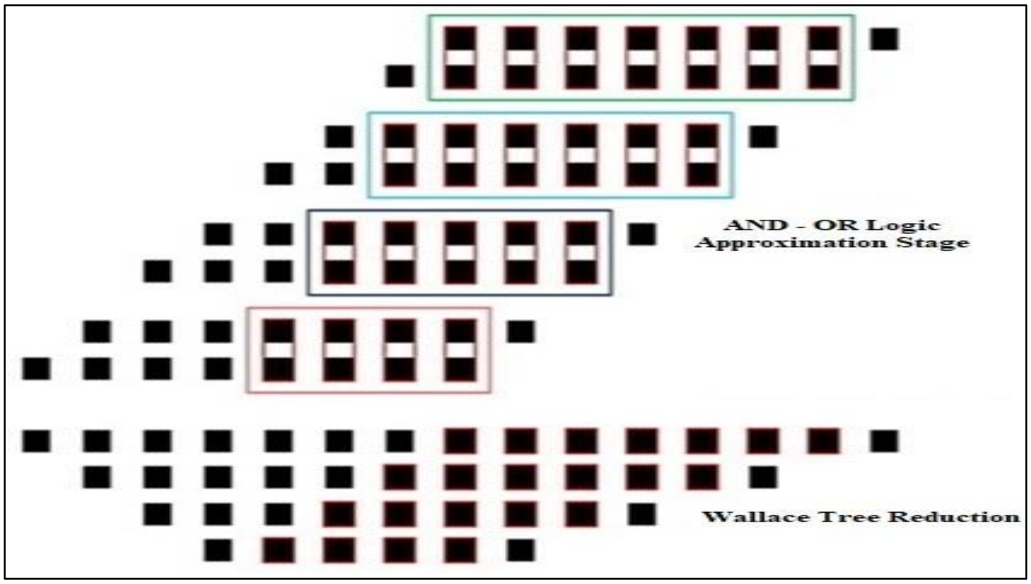

Figure 4 Wallace tree reduction

6.5 Final partial product addition

To get the product of the inputs, accumulate the reduced set of partial products after Wallace tree reduction [14]. We used a new inexact half adder and full adders to add the partial products by changing the design. So that it would achieve better outcomes concerning delay [2, 20]. In this paper, few designs have been suggested for inexact adders with reduced gate counts.

7.Proposed inexact compressor adders

A half adder comprises an AND gate and an XOR gate. We replace the XOR gate with an OR gate such that the error output is obtained only when both the inputs are high. It is shown in Equation 1 and 2.

$$\{\text{carry}(C_0),\text{Sum}(S)\}_{\text{Exact}} = \{\text{AandB}\,,\text{AxorB}\} \quad (1)$$

In *Figure 5*, a new half-adder design was proposed that gives a faulty output when the inputs {A, B} = 11, where the sum output (s) is 1 instead of 0. The output expression is given in Equation 2. Four new 3:2 compressor adder designs were presented in this paper.

$$\{\text{carry}(C_0),\text{Sum}(S)\}_{\text{InExact}} = \{\text{AandB}\,,\text{AorB}\} \quad (2)$$

A fully functional 3:2 compressor adder was given in *Figure 6* and the output expression is given in Equation 3.

$\{C_O, S\}_{\text{Exact}}$ = {(A and B) or ((A xor B) and C_i), (A xor B xor C_i)) (3)

In *Figure 7* a new 3:2 compressor adder design 1 was proposed which provides the faulty output when the inputs {a, b, c_i} = 001 where the carry output (Co) is 1 instead of Zero. The output expression is given below in Equation 4 and the truth table for this new 3:2 compressor adder is shown in *Table 3.*

$\{CO, S\}_{\text{Inexact 1}}$ = {(A and B) or Ci, (A xor B xor Ci)} (4)

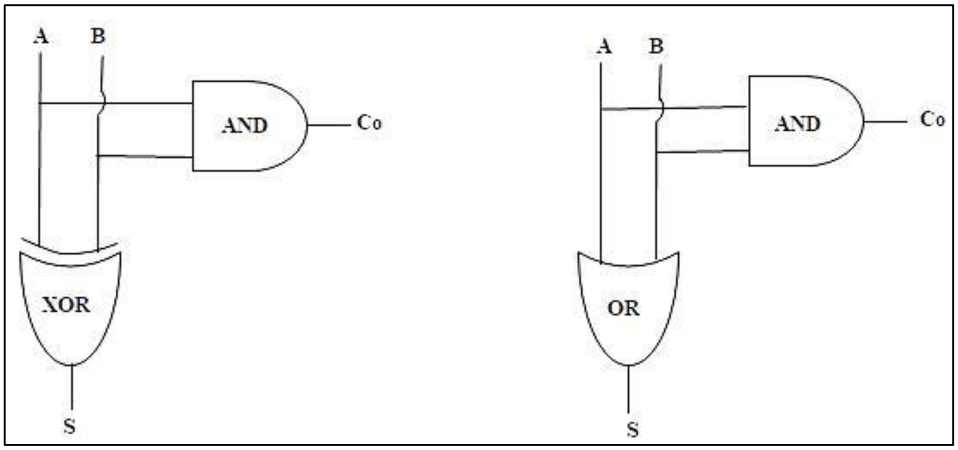

Figure 5 (a) Exact half adder and inexact half adder

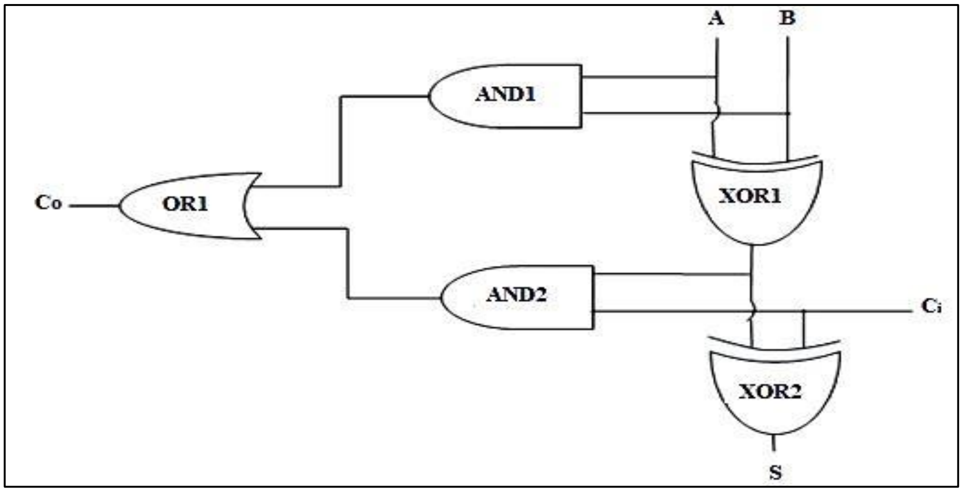

Figure 6 Fully functional 3:2 compressor adder

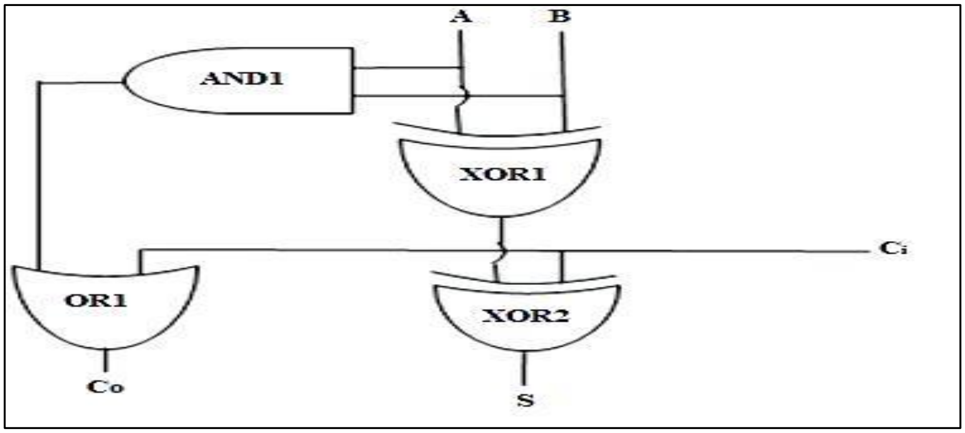

Figure 7 New 3:2 compressor adder design 1

Table 3 Truth table of inexact 3:2 compressor adder design 1

Inputs			**Inexact output**	**Exact compressor output**	
A	B	C_i	C_O	S	C_O
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	1

In *Figure 8* a new 3:2 compressor adder design 2 was proposed which provides the faulty output when the inputs {a, b, c_i} = 110,111 where the carry output (Co) is 0 instead of 1. The output expression is given below in Equation 5 and the truth table for this new 3:2 compressor adder is shown in *Table 4.*

$$\{C_O, S\}_{\text{Inexact 2}} = \{(A \text{ xor } B) \text{ and } C_i, ((A \text{ xor } B) \text{ xor } C_i) \quad (5)$$

In *Figure 9* a new 3:2 compressor adder design 3 was proposed which provides the faulty output when the inputs {a, b, c_i} = 110,111. If it is 110 then the carry output (Co) is 0 instead of 1, the sum output (s) is 1 instead of 0. If the input is 111 than the sum generated is 0 while the actual output is 1. The output expression is given below in Equation 6 and the truth table for this new 3:2 compressor adder is shown in *Table 5.*

$$\{CO, S\}_{\text{Inexact 3}} = \{(A \text{ or } B) \text{ and } Ci, ((A \text{ or } B) \text{ xor } Ci)\} \quad (6)$$

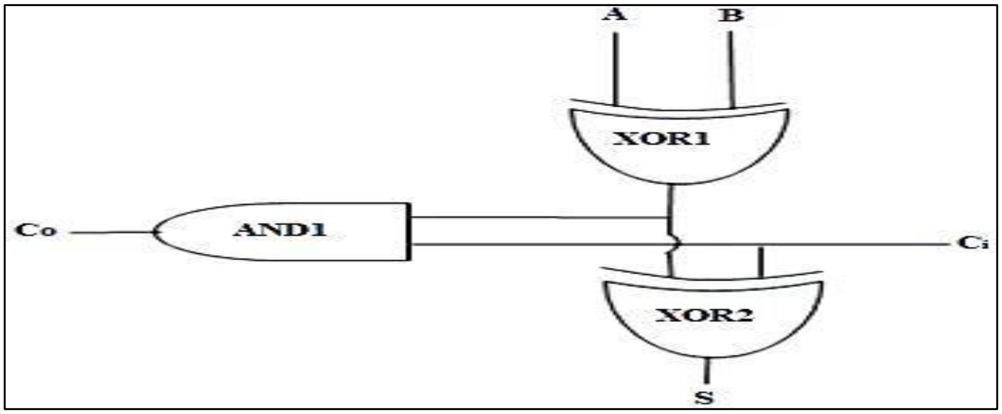

Figure 8 New 3:2 compressor adder design 2

Table 4 Truth table of Inexact 3:2 compressor adder design 2

Inputs			Inexact output	Exact compressor output	
A	B	C_i	C_O	S	C_O
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	0	1	1

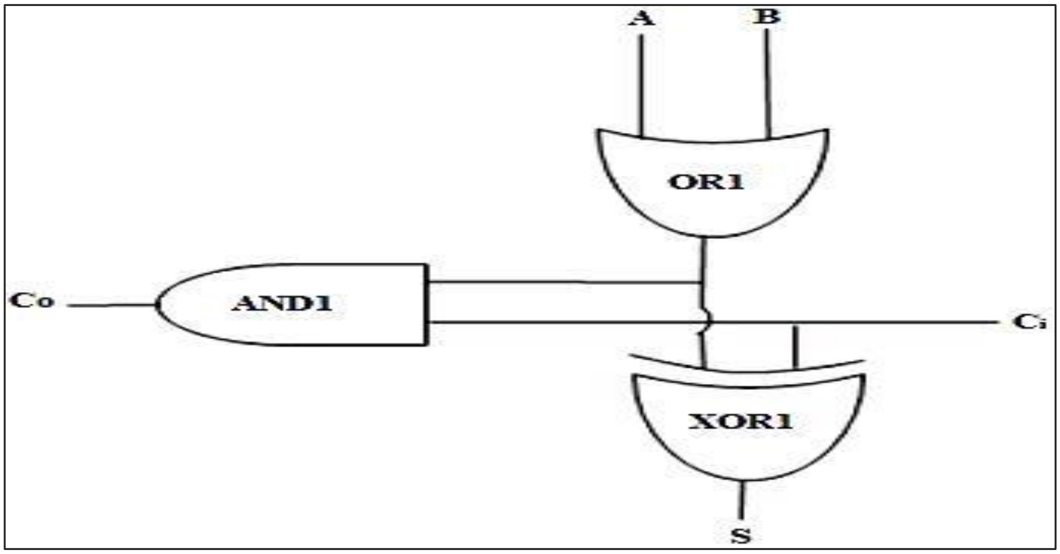

Figure 9 New 3:2 compressor adder design 3

Table 5 Truth table of inexact 3:2 compressor adder design 3

Inputs			Inexact output		Exact compressor output	
A	B	C_i	S	C_O	S	C_O
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	0	1	1	1

In *Figure 10* a new 3:2 compressor adder design 4 was proposed which provides the faulty output when the inputs {a, b, c_i} = 011,100 where the carry output (Co) is reflected with accurate. The output expression is given below in equation 7 and the truth table for this new 3:2 compressor adder is shown in *Table 6.*

$$\{C_O, S\}_{\text{Inexact 4}} = \{A, ((A \text{ xor } B) \text{ xor } C_i)\} \qquad (7)$$

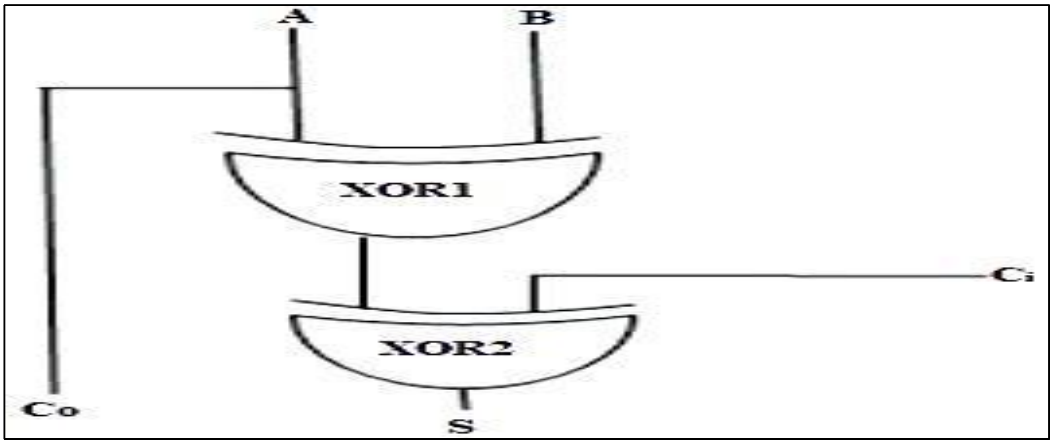

Figure 10 New 3:2 compressor adder design 4

Table 6 Truth table of Inexact 3:2 compressor adder design 4

Inputs			**Inexact output**	**Exact compressor output**	
A	B	C_i	C_O	S	C_O
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	1

8.Results

The simulation results for different operand values of conventional Wallace multiplier and Approximate Wallace Multiplier (AWM) using AND-OR logic compression is shown in *Figure 11* and *Figure 12.* Four different kinds of AWM's were implemented by using above mentioned 3:2 compressor adder designs. Here Approximate Wallace Multiplier 1 (AWM1) was implemented with 3:2 compressor adder design 1. Approximate Wallace Multiplier 2 (AWM2) was implemented with compressor adder design 2. Approximate Wallace Multiplier 3 (AWM3) was implemented with compressor adder design 3. Finally Approximate Wallace Multiplier 4 (AWM4) was implemented by making use of compressor adder design 4. The simulation results of proposed multiplier i.e. AWM1, AWM2, AWM3 and AWM4 for different operand values are shown in *Figure 13*, *Figure 14*, *Figure 15* and *Figure 16* respectively. The proposed multipliers were designed and tested in the Xilinx environment and are simulated in the ISE Simulator (ISim). Complete list of abbreviations is shown in *Appendix* I.

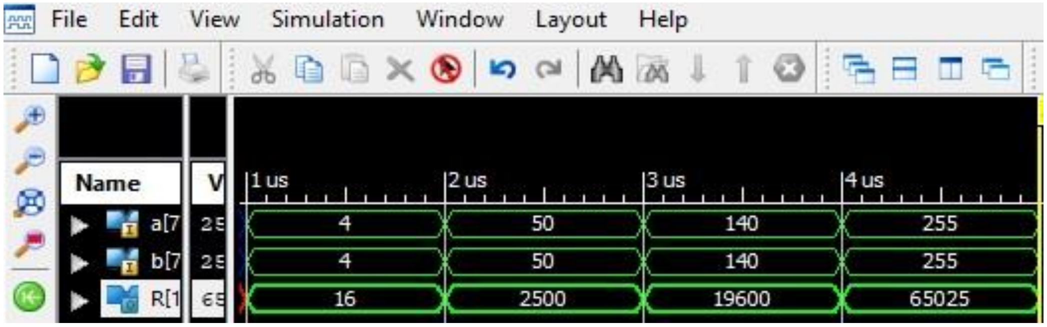

Figure 11 Simulation output of conventional Wallace multiplier

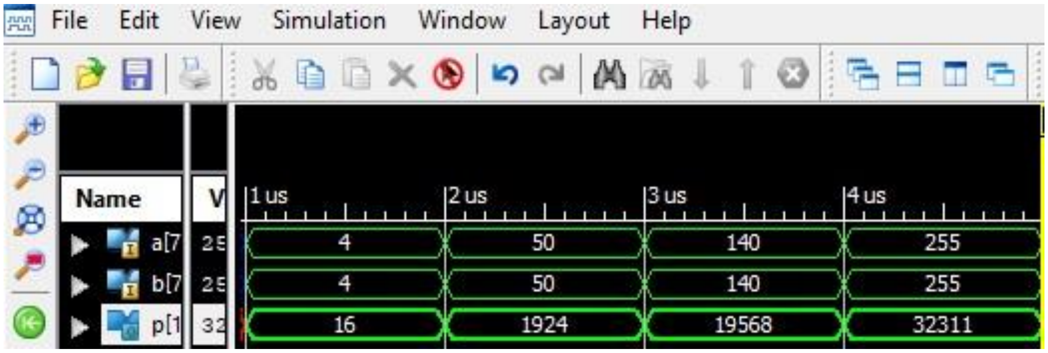

Figure 12 Simulation output of AWM using AND-OR logic compression

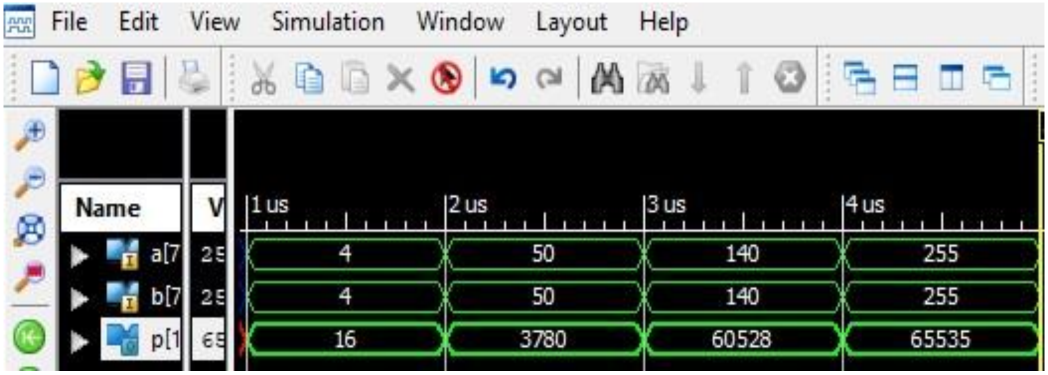

Figure 13 Simulation output of AWM1

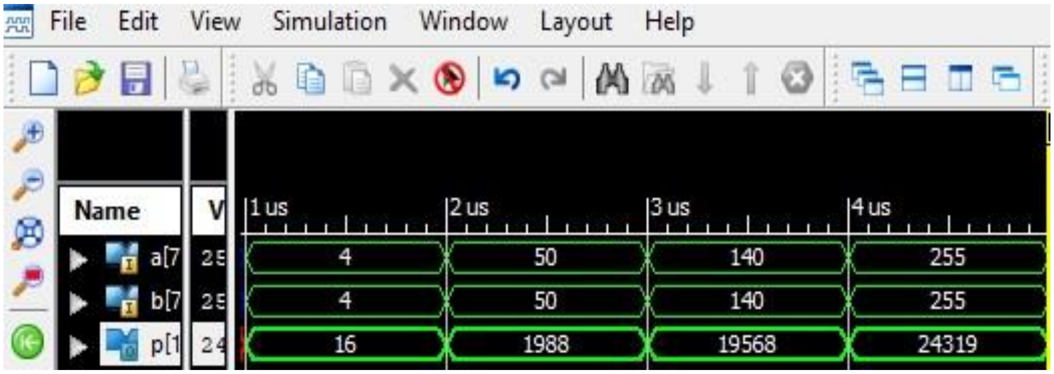

Figure 14 Simulation output of AWM2

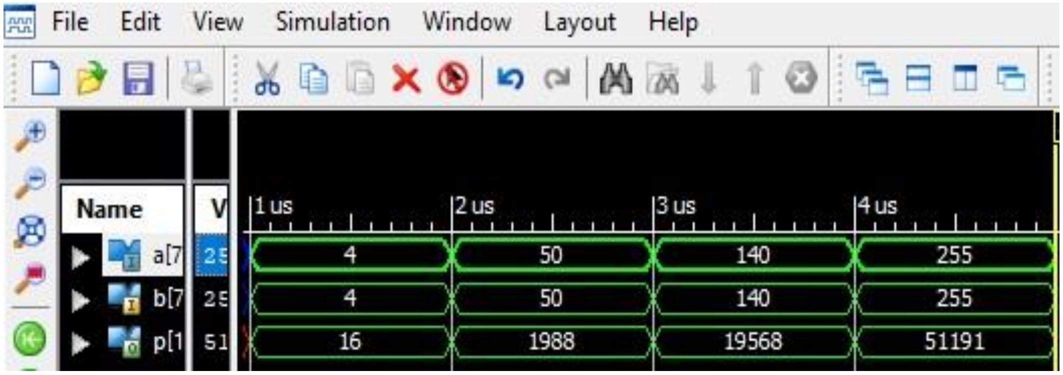

Figure 15 Simulation output of AWM3

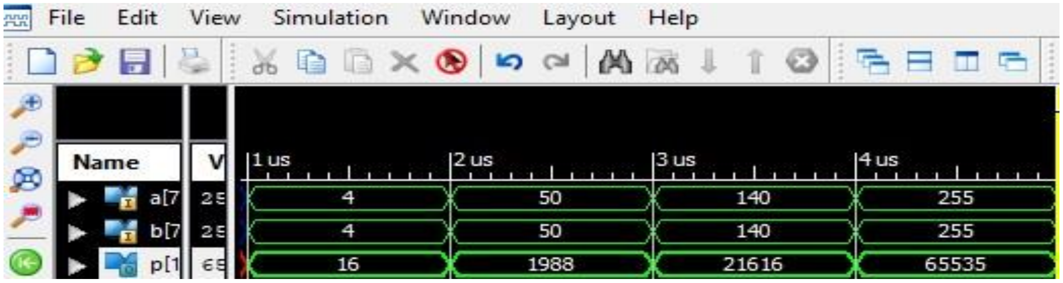

Figure 16 Simulation output of AWM4

9.Discussion

It is observed from the simulation results that error is significantly higher at large values of operands compression (*Figure 12*). So, to minimize the error arising in AWM using AND-OR logic compression for different inexact 3:2 compressor adder designs were used in the design.

The area, delay metrics, and the percentage reduction in area and delay parameters as compared to the conventional multiplier is shown in *Table 7*. The proposed approximate multipliers were implemented in Verilog HDL and synthesized with the Xilinx ISE 14.1 version tool. Compared to the conventional multiplier, all four proposed designs yield significant area and delay reductions. The proposed AWM1 designs reduce 35.577% area and 35.224% delay. Similarly, AWM2 design reduces 48.077% area and 36.532% delay, whereas AWM3 design reduces 48.077% area and 46.633% delay and finally AWM4 design reduces 53.846% area and 56.482% delay compared to the conventional multiplier.

Table 7 Comparison of proposed approximate Wallace multipliers with conventional multiplier

Type	Area (lut's)	Actual (%)	Reduction (%)	Delay (ns)	Actual (%)	Reduction (%)
Conventional Multiplier	104	100	0	16.662	100	0
Conventional Wallace Multiplier	114	109.615	-9.615	12.737	76.443	23.557
AWM using AND-OR logic	73	70.192	29.808	9.735	58.426	41.574
AWM1	67	64.243	35.577	10.793	64.776	35.224
AWM2	54	51.923	48.077	10.575	63.468	36.532
AWM3	54	51.923	48.077	8.892	53.367	46.633
AWM4	48	46.154	53.846	7.251	43.518	56.482

10.Conclusion and future scope

An approximate multiplier was developed by using an inexact half adder and four 3:2 compressor adders. It is with four different design configurations and compared with the accurate conventional multiplier. The proposed multiplier is found to be better in terms of area and delay performances. The present work mainly concentrated in the area and delay reduction techniques further it can be carried out for power reduction. In the future, the same can be implemented with different design configurations for the full adders.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Momeni A, Han J, Montuschi P, Lombardi F. Design and analysis of approximate compressors for multiplication. IEEE Transactions on Computers. 2014; 64(4):984-94.

[2] Gupta V, Mohapatra D, Raghunathan A, Roy K. Low-power digital signal processing using approximate adders. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2012; 32(1):124-37.

[3] Reddy KM, Vasantha MH, Kumar YN, Dwivedi D. Design and analysis of multiplier using approximate 4-2 compressor. AEU-International Journal of Electronics and Communications. 2019; 107:89-97.

[4] Esposito D, Strollo AG, Napoli E, De CD, Petra N. Approximate multipliers based on new approximate compressors. IEEE Transactions on Circuits and Systems I: Regular Papers. 2018; 65(12):4169-82.

[5] Yang Z, Jain A, Liang J, Han J, Lombardi F. Approximate XOR/XNOR-based adders for inexact computing. In international conference on nanotechnology 2013 (pp. 690-3). IEEE.

[6] Han J, Orshansky M. Approximate computing: an emerging paradigm for energy-efficient design. In European test symposium 2013 (pp. 1-6). IEEE.

[7] Kyaw KY, Goh WL, Yeo KS. Low-power high-speed multiplier for error-tolerant application. In international conference of electron devices and solid-state circuits 2010 (pp. 1-4). IEEE.

[8] Naaz SA, Pradeep MN, Bhairannawar S, Halvi S. FPGA implementation of high speed vedic multiplier using CSLA for parallel FIR architecture. In international conference on devices, circuits and systems 2014 (pp. 1-5). IEEE.

[9] Baran D, Aktan M, Oklobdzija VG. Energy efficient implementation of parallel CMOS multipliers with improved compressors. In proceedings of the international symposium on low power electronics and design 2010 (pp. 147-52).ACM

[10] Gupta A, Malviya U, Kapse V. Design of speed, energy and power efficient reversible logic based vedic ALU for digital processors. In Nirma university international conference on engineering 2012 (pp. 1-6). IEEE.

[11] Lin CH, Lin C. High accuracy approximate multiplier with error correction. In international conference on computer design 2013 (pp. 33-8). IEEE.

[12] Liu C, Han J, Lombardi F. A low-power, high-performance approximate multiplier with configurable partial error recovery. In design, automation & test in Europe conference & exhibition 2014 (pp. 1-4). IEEE.

[13] Jiang H, Liu C, Maheshwari N, Lombardi F, Han J. A comparative evaluation of approximate multipliers. In international symposium on nano scale architectures 2016 (pp. 191-6). IEEE.

[14] Bhardwaj K, Mane PS, Henkel J. Power-and area-efficient approximate Wallace tree multiplier for error-resilient systems. In fifteenth international symposium on quality electronic design 2014 (pp. 263-9). IEEE.

[15] Maheshwari N, Yang Z, Han J, Lombardi F. A design approach for compressor based approximate multipliers. In international conference on VLSI design 2015 (pp. 209-14). IEEE.

[16] Akbari O, Kamal M, Afzali-kusha A, Pedram M. Dual-quality 4: 2 compressors for utilizing in dynamic accuracy configurable multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2017; 25(4):1352-61.

[17] Kulkarni P, Gupta P, Ercegovac MD. Trading accuracy for power in a multiplier architecture. Journal of Low Power Electronics. 2011; 7(4):490-501.

[18] Anitha R, Deshmukh N, Agarwal P, Sahoo SK, Karthikeyan SP, Reglend IJ. A 32-bit mac unit design using vedic multiplier and reversible logic gate. In international conference on circuits, power and computing technologies 2015 (pp. 1-6). IEEE.

[19] Narayanamoorthy S, Moghaddam HA, Liu Z, Park T, Kim NS. Energy-efficient approximate multiplication for digital signal processing and classification applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2014; 23(6):1180-4.

[20] Pei H, Yi X, Zhou H, He Y. Design of ultra-low power consumption approximate 4–2 compressors based on the compensation characteristic. IEEE Transactions on Circuits and Systems II: Express Briefs. 2020; 68(1):461-5.

[21] Masadeh M, Hasan O, Tahar S. Comparative study of approximate multipliers. In proceedings of the on great lakes symposium on VLSI 2018 (pp. 415-8). ACM

[22] Arya NS, Nair RM. Approximate computing: a new trend in VLSI based multipliers for error resilient DIP applications. International Research Journal of Engineering and Technology (IRJET) 2018; 5(4): 3866-9.

[23] Guo Y, Sun H, Kimura S. Design of power and area efficient lower-part-OR approximate multiplier. In TENCON 2018 (pp. 2110-5). IEEE.

[24] Maddisetti L, Ravindra JV. Performance metrics of inexact multipliers based on approximate 5: 2 compressors. In international SoC design conference 2018 (pp. 84-5). IEEE.

[25] Kiruthika R, Suguna S. Low latency and power efficient approximate multipliers using compressors. International Journal of Engineering Research & Technology. 2019; 8(4):593–6.

[26] Yi X, Pei H, Zhang Z, Zhou H, He Y. Design of an energy-efficient approximate compressor for error-resilient multiplications. In international symposium on circuits and systems 2019 (pp. 1-5). IEEE.

[27] Strollo AG, De CD, Napoli E, Petra N, Di MG. Low-power approximate multiplier with error recovery using a new approximate 4-2 compressor. In international symposium on circuits and systems 2020 (pp. 1-4). IEEE.

[28] Balasubramanian P, Nayar R, Maskell DL. Approximate array multipliers. Electronics. 2021; 10(5):1-20.

[29] Guo Y, Sun H, Guo L, Kimura S. Low-cost approximate multiplier design using probability-driven inexact compressors. In Asia pacific conference on circuits and systems 2018 (pp. 291-4). IEEE.

[30] Sabetzadeh F, Moaiyeri MH, Ahmadinejad M. A majority-based imprecise multiplier for ultra-efficient approximate image multiplication. IEEE Transactions on Circuits and Systems I: Regular Papers. 2019; 66(11):4200-8.

[31] Venkatachalam S, Adams E, Lee HJ, Ko SB. Design and analysis of area and power efficient approximate booth multipliers. IEEE Transactions on Computers. 2019; 68(11):1697-703.

[32] Senthilkumar KK, Kumarasamy K, Dhandapani V. Approximate multipliers using bio-inspired algorithm. Journal of Electrical Engineering & Technology. 2021; 16:559-68.

B. Sudharani, Research Scholar in the Department of Electronics and Communication Engineering at Sri Venkateswara University College of Engineering, Tirupati, India. She obtained her B.Tech. Degree in Electronics and Communication Engineering from S.V. University and M.Tech degree in VLSI System Design from Jawaharlal Nehru Technological University Hyderabad. She has 15 years of teaching experience. Her research area of interest includes Analog and Digital Circuits.
Email: bsudhasvu@gmail.com

Dr. G. Sreenivasulu is working as a Professor in the Department of Electronics and Communication Engineering at Sri Venkateswara University College of Engineering, Tirupati, India. He obtained his B.Tech Degree in Electronics and Communication Engineering, M.Tech Degree in Instrumentation and Control Systems, and Ph.D. in the area of Process control from Sri Venkateswara University in 1990, 1992, and 2007 respectively. He has 30 years of teaching and research experience. He is a fellow of IETE and a life member of ISTE. He has published 27 papers in referred Journals and also presented 17 papers at the national and international levels. His research area of interest includes Analog Circuits, Digital Circuits, Electronic Instrumentation, and Process Control.
Email: gunapatieee@gmail.com

Appendix I

S.No.	Abbreviation	Description
1	ALU	Arithmetic Logic Unit
2	AWM	Approximate Wallace Multiplier
3	AWTM	Approximate Wallace Tree Multiplier
4	CMOS	Complementary Metal Oxide Semiconductor
5	CSA	Carry Select Adder
6	DSP	Digital Signal Processing
7	ECM	Error Correction Module
8	ISim	ISE Simulator
9	LSB	Least Significant Bit
10	MSB	Most Significant Bits
11	PPG	Partial Product Generation
12	PPP	Partial Product Perforation
13	PPR	Partial Product Reduction