DESIGN OF A 32-BIT ACCURACY-CONTROLLABLE APPROXIMATE MULTIPLIER FOR FPGAS

[1]Asifa, [2]Dr. P. Nithiyanantham
[1]PG Scholar, M.Tech, Dept of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, TS.
sasifa640@gmail.com
[2]Professor, Dept of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, TS.
palanisamy.nithiyanantham@gmail.com

ABSTRACT
In this research, we provide an estimated 32 x 32-bit multiplier for FPGAs. The suggested approximation multiplier contains two operands as well as a specific input to dynamically regulate the accuracy. When great precision is desired, the multiplier can conduct slow multiplication instead of quick multiplication using the special input. This study also examines the suggested multiplier's latency and maximum error, demonstrating that there is a trade-off between the two. Many error-tolerant applications depend on the fundamental function of multiplication. An effective method for balancing energy usage with efficiency and precision is approximate multiplication. This article suggests an adder with carry masking that produces the end result of an accuracy-controllable multiplier. The suggested technique may flexibly meet the accuracy requirements by dynamically choosing the carry propagation's length. The suggested tree compressor makes an approximation of the multiplier's partial product tree. The carry maskable adder and the compressor are used to construct an 8x8 multiplier architecture.

1. INTRODUCTION

Majority of the signal processing applications have convolutional units as its computationally intensive and performance determining operational units. Adders and multipliers are mainly used in convolutional units among which multipliers significantly contributes to the area, delay and power. High speed multipliers that are optimized for area and power are in great demand in real life applications involving computational units which are used in convolutional neural networks, multimedia, etc. Multiplication operation can be broadly divided into three stages 1) generation of partial products using an array of AND gates 2) partial product accumulation and 3) final partial product addition. 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. Algorithms introduced by Dadda [1] 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 V 3, 7 V 2, etc [3]. In the recent times, multipliers are explored in the light of approximation as they find huge demand in realising area and power optimised design for error tolerant applications like multimedia processing, neural network, signal processing, etc. Recently, such optimisations in CMOS [4]- [9], FPGA [10], pass transistor [11] and FinFET [12] based multiplier realization's are being researched.

1.2 EXISTING SYSTEM

Akbari *et al.* [4] 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. An XOR-less architecture was proposed by Esposito *et al.* [5] with 56.25% ER. However, high ER makes it inefficient in image processing applications. Reddy *et al.* [6] and Edavoor *et al.* [7] proposed compressors based on multiplexers. These designs are more appropriate for conventional gate level implementation. Gorantla and Deepa [8] have proposed 4:2 compressors with optimised delay, but the area/gate count is high.

Strollo *et al.* [9] have proposed a 4:2 compressor design by changing the method of stacking circuits. Above mentioned designs are optimised for CMOS based applications. 4:2 compressors optimised for FPGA based architecture is presented in [10]. A compact 4:2 compressor for FPGAs with low accuracy losses and higher electrical performance is proposed by Toan and Lee [10]. Chang *et al.* [11] have proposed approximate 4:2 compressor optimised for pass transistor-based implementation. FinFET based implementation for approximate 4:2 compressor was proposed by Zakian and Asli [12].

All the above-mentioned architectures are restricted by the physical limitations set by Moore's law [13]. In irreversible computations, for every bit lost, there is $KTln2$ joules of energy loss. This dissipation was insignificant in higher technologies [14]. 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. In 1973, 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 [15].

Realisation of circuits and systems using reversible logic gates is an emerging area of research. The efficacy of circuit realisations using reversible logic is measured in scales of one or a combination of reversible logic realization parameters - QC, GC, GO and AI [16]-[27]. Chattopadhyay and Baksi [16] proposed two low quantum cost circuit construction for symmetric Boolean functions. The authors have demonstrated the implementation of adder circuits and has measured and compared the efficacy of the proposed adder by projecting GO, GC, and QC. The first construction has reduced GO and the second construction has reduced GC. Norwin *et al.* [17] have proposed a reversible logic based bidirectional barrel shifter for input widths that are integer powers of 2. The proposed n-bit barrel shifter has *logn* select lines with a maximum shift of (*n* - 1) bits. The comparative analysis shows that the proposed 8-bit barrel shifter is able to reduce GC by 19.44%,

GO by 36.84% and QC by 12.93% as compared to [18] that has the least reversible logic realisation parameters in the literature. Khan and Rice [19] introduced a Min-Max algebra-based synthesis technique to realise 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 [20] proposed a method to realise 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 of 23.78% and 66.63% reduction in QC and GO compared to the existing replacement technique [21].

Molahosseini *et al.* [22] have proposed a technique to leverage the parallelism in residue number systems to improve the efficiency of reversible circuit realisations. A parallel-prefix modulo-(2*n*□1) adder proposed is able to reduce the overhead of QC by 19.81% as compared to regular Brent-Kung adder.

Gaur *et al.* [23] proposed an approach to realise 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. The proposed approach reduces GO as compared to previous work in the testable reversible circuit design. Gaur *et al.* [24] proposed a reversible logic-based realisation for Arithmetic Logic Unit (ALU) that can be scaled for N-bits using novel parity-preserving structure. The proposed circuit attains single-bit fault coverage by using two-fold gate placement method and Fredkin gates. The efficiency is projected in terms of GC, QC, GO and AI. This approach achieves an improvement of 61%, 74%, 27% and 14% in GC, QC, GO and AI for a 4-bit ALU design. Datta *et al.* [25] proposed an optimisation technique using multiple-control Toffoli gate netlist. This approach has repeated application of replacement and pair-wise gate merging rules. The proposed technique is tested on reversible benchmark circuits [28] and was able to obtain improvement of 64.9% for QC and 73.8% for GC. Raveendran *et al.* [26] have proposed reversible logic circuit realisation 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 *et al.* [27] 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 optimised for reversible logic with 25% ER and _2 ED. 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).

1.3 PROPOSED SYSTEM

To address the need to realise low power computational units for error resilient applications, this paper proposes an inexact Baugh-Wooley Wallace tree multiplier. This paper achieve approximation in the multiplier architecture by introducing a novel architecture for inexact 4:2 compressor which is compatible for implementation using reversible logic by optimising reversible logic realisation parameters GC, QC, GO and AI. 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 rationalised 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.

2. LITERATURE SURVEY

"A new design technique for column compression multipliers,", Z. Wang, G. A. Jullien, and W. C. Miller.

In this paper, a new design technique for column-compression (CC) multipliers is presented. Constraints for column compression with full and half adders are analyzed and, under these constraints, considerable flexibility for implementation of the CC multiplier, including the allocation of adders, and choosing the length of the final fast adder, is exploited. Using the example of an 8/spl times/8 bit CC multiplier, we show that architectures obtained from this new design technique are more area efficient, and have shorter interconnections than the classical Dadda CC multiplier. We finally show that our new technique is also suitable for the design of twos complement multipliers.

"**Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers**", O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram.

In this paper, we propose four 4:2 compressors, which have the flexibility of switching between the exact and approximate operating modes. In the approximate mode, these dual-quality compressors provide higher speeds and lower power consumptions at the cost of lower accuracy. Each of these compressors has its own level of accuracy in the approximate mode as well as different delays and power dissipations in the approximate and exact modes. Using these compressors in the structures of parallel multipliers provides configurable multipliers whose accuracies (as well as their powers and speeds) may change dynamically during the runtime. The efficiencies of these compressors in a 32-bit Dadda multiplier are evaluated in a 45-nm standard CMOS technology by comparing their parameters with those of the state-of-the-art approximate multipliers. The results of comparison indicate, on average, 46% and 68% lower delay and power consumption in the approximate mode. Also, the effectiveness of these compressors is assessed in some image processing applications.

"**Approximate multipliers based on new approximate compressors**", D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra.

Approximate computing is an emerging trend in digital design that trades off the requirement of exact computation for improved speed and power performance. This paper proposes novel approximate compressors and an algorithm to exploit them for the design of efficient approximate multipliers. By using the proposed approach, we have synthesized approximate multipliers for several operand lengths using a 40-nm library. Comparison with previously presented approximated multipliers shows that the proposed circuits provide better power or speed for a target precision. Applications to image filtering and to adaptive least mean squares filtering are also presented in the paper.

"Design and analysis of multiplier using approximate 4-2 compressor", K. Manikantta Reddy, M. H. Vasantha, Y. B. Nithin Kumar, and D. Dwivedi.

Approximate computing has received significant attention as an attractive paradigm for error-tolerant applications to reduce the power consumption, delay and area with some trade-off in accuracy. This paper proposes the design of a novel approximate 4–2 compressor. A modified architecture of Dadda Multiplier is presented for the effective utilization of the proposed compressor and to reduce the error at the output. Through extensive experimental evaluation, the efficiency of the proposed compressor and multiplier are evaluated in a 45 nm standard CMOS technology and their parameters are compared with those of the state-of-the-art approximate multipliers. The results show that the proposed compressor accomplish a significant reduction in error rate compared to other approximate compressors available in the literature. In addition, the proposed multiplier shows 35%, 36% and 17% reduction in power consumption, delay and area respectively compared to those of exact multiplier. The effectiveness of multiplier is assessed by some of the image processing applications. On an average, the proposed multiplier processes images with 85% structural similarity compared to the exact output image.

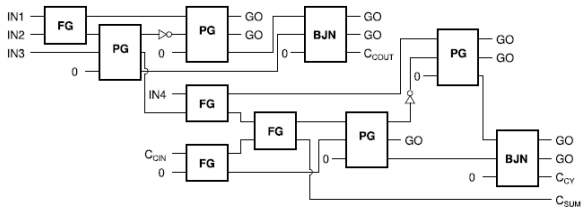
"Approximate multiplier design using novel dual-stage 4:2 compressors," P. J. Edavoor, S. Raveendran, and A. D. Rahulkar,

High speed multimedia applications have paved way for a whole new area in high speed error-tolerant circuits with approximate computing. These applications deliver high performance at the cost of reduction in accuracy. Furthermore, such implementations reduce the complexity of the system architecture, delay and power consumption. This paper explores and proposes the design and analysis of two approximate compressors with reduced area, delay and power with comparable accuracy when compared with the existing architectures. The proposed designs are implemented using 45 nm CMOS technology and efficiency of the proposed designs have been extensively verified and projected on scales of area, delay, power, Power Delay Product (PDP), Error Rate (ER), Error Distance (ED), and Accurate Output Count (AOC). The proposed approximate 4 : 2 compressor shows 56.80% reduction in area, 57.20% reduction in power, and 73.30% reduction in delay compared to an accurate 4 : 2 compressor. The proposed compressors are utilised to implement 8 × 8 and 16 × 16 Dadda multipliers. These multipliers have comparable accuracy when compared with state-of-the-art approximate multipliers. The analysis is further extended to project the application of the proposed design in error resilient applications like image smoothing and multiplication.

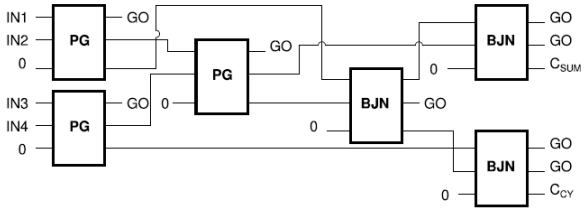
"Design of approximate compressors for multiplication," A. Gorantla and P. Deepa,
Approximate computing is a promising technique for energy-efficient Very Large Scale Integration (VLSI) system design. It is best suited for error-resilient applications such as signal processing and multimedia. Approximate computing reduces accuracy but still provides significant and faster results with lower power consumption. This is attractive to arithmetic circuits. In this article, various novel design approaches of approximate 4-2 and 5-2 compressors have been proposed for reduction of the partial product stages in multiplication. Three approximate 8 × 8 Dadda multiplier designs using three novel approximate 4-2 compressors and two approximate 8 × 8 Dadda multiplier designs using two novel approximate 5-2 compressors have proposed. The synthesis results show that the proposed designs achieved significant accuracy improvement together with power and delay reductions compared to the existing approximate designs.

"Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers," A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo
Approximate multipliers attract a large interest in the scientific literature that proposes several circuits built with approximate 4-2 compressors. Due to the large number of proposed solutions, the designer who wishes to use an approximate 4-2 compressor is faced with the problem of selecting the right topology. In this paper, we present a comprehensive survey and comparison of approximate 4-2 compressors previously proposed in literature. We present also a novel approximate compressor, so that a total of twelve different approximate 4-2 compressors are analyzed. The investigated circuits are employed to design 8 × 8 and 16 × 16 multipliers, implemented in 28nm CMOS technology. For each operand size we analyze two multiplier configurations, with different levels of approximations, both signed and unsigned. Our study highlights that there is no unique winning approximate compressor topology since the best solution depends on the required precision, on the signedness of the multiplier and on the considered error metric.

3. PROPOSED METHODOLOGY
BLOCK DIAGRAM



Exact 4:2 compressor using reversible logic.



Proposed Inexact 4:2 compressor using reversible logic.

3.2 BAUGH-WOOLEY WALLACE TREE MULTIPLIER

A. BAUGH-WOOLEY ALGORITHM FOR MULTIPLICATION

The Baugh-Wooley algorithm is a partial product generation algorithm that operates on 2's complement operands to perform multiplication of signed and unsigned numbers. Consider two N-bit numbers P and Q in 2's complement integer representation as given in Eq. 1 and Eq. 2.

$$P = -p_{N-1}2^{N-1} + \sum_{x=0}^{N-1} p_x 2^x$$

$$Q = -q_{N-1}2^{N-1} + \sum_{y=0}^{N-1} q_y 2^y$$

The product defined by Baugh-Wooley algorithm [29], [30] [31] can be represented as

$$P \times Q = p_{N-1}q_{N-1}2^{2N-2} + \sum_{x=0}^{N-2}\sum_{y=0}^{N-2} p_x q_y 2^{x+y} + 2^{N-1}\left(-2^{N-1} + \sum_{y=0}^{N-2} \overline{p_{N-1}q_y}2^y + 1\right) + 2^{N-1}\left(-2^{N-1} + \sum_{x=0}^{N-2} \overline{q_{N-1}p_x}2^x + 1\right)$$

Constant `1' is added at the 2N-1*th* and N*th* columns. MSB of the first N-1 partial product rows are inverted. All the elements in the N*th* row except the MSB is

inverted. These defines the final partial product array in BaughWooley algorithm which is then reduced using any partial product reduction techniques.

B. WALLACE ADDER TREE REDUCTION TECHNIQUE

Various adder-tree reduction techniques are presented in literature (Dadda [1], Wallace [2], Braun [32]) to increase the speed of operation during partial product accumulation. Wallace tree adder reduction technique was introduced by Chris Wallace in 1964 for fast multiplication [2]. The partial product array is rearranged in a tree like structure. In the implementation of conventional Wallace tree multiplier, a series of full adders are used to generate the final carry and sum in a column with multiple partial product bits. However, this method has varying number of bits to be added in each column, which demands a complex circuit for accumulation. Use of compressors can reduce the complexity in this phase. Various topology (3 : 2, 4:2, 5 : 2, 7 : 3) have been proposed in the past and 4:2 compressors have gained popularity in multiplier design due to the regularity that it can achieve in the accumulation phase [3]. In this work, Wallace tree reduction technique is used for accumulation. The partial product bits in each columns are grouped in groups of four, three or two. Compressors are used for groups of four bits. Full adders are for groups of three bits and half adders for groups of two bits.

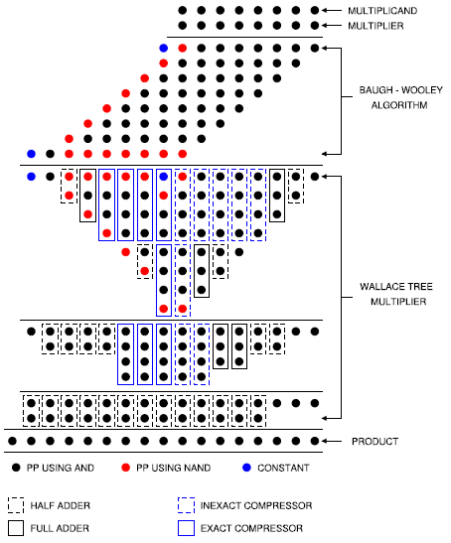


Fig 1. Multiplication operation using Baugh Wooley Wallace tree multiplier.

C. BAUGH-WOOLEY WALLACE TREE MULTIPLIER

In this work, Baugh-Wooley algorithm is used to generate partial products and Wallace tree adder tree reduction technique is used in accumulation stage. To improve the speed of operation, 4:2 compressors are used instead of full adders in some columns. To reduce gate count, inexact compressors are utilised in lower significant columns in partial product array.

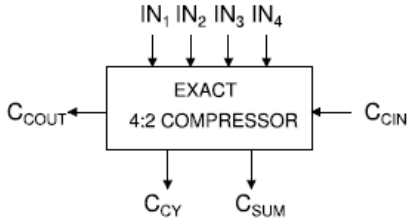


FIGURE 2. Exact 4:2 compressor.

The complete architecture for Baugh Wooley Wallace tree multiplier implemented in this paper is presented in Figure 1. In this architecture, eight exact 4:2 compressors, eight inexact 4:2 compressors, five full adders and twenty-three half adders are used.

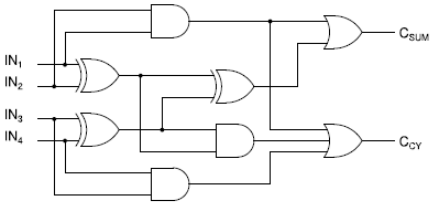


FIGURE 3. Proposed inexact 4:2 compressor using conventional logic gates.

To avoid the delay introduced due to carry propagation during addition operation, the *Sum* and *Carry* bits of all the units are passed onto the next stage for further addition. The *Sum* is passed onto the column with same bit weight and carry is passed onto next higher significant column. This approach is followed until the final stage where the partial product bits are reduced to two rows. In the final stage, a carry propagate addition operation is carried to obtain the final product.

EXACT AND PROPOSED INEXACT 4:2 COMPRESSORS

A. EXACT 4:2 COMPRESSOR

4:2 compressors are used in high performance parallel multipliers to reduce the delay in partial product accumulation stage. Figure 2 shows an exact 4:2 compressor. An exact 4:2 compressor has five inputs and three outputs. The underlying operation in a 4:2 exact compressor is to _nd the number of logic `1' in the input. The outputs can be represented as

$$C_{SUM} = C_{CIN} \oplus IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1$$
$$C_{COUT} = IN_1(\overline{IN_2 \oplus IN_1}) + IN_3(IN_2 \oplus IN_1)$$
$$C_{CY} = IN_4(\overline{IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1}) + C_{CIN}(IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1)$$

Carry and *Cout* has equal and higher weight than *Sum* and are propagated to the next compressor.

B. INEXACT 4:2 COMPRESSOR

Compressors are explored in the light of approximation to increase performance and reduce gate count. Introducing error in full adders can adversely affect accuracy with an error rate of approximately 53% [33], [34]. In order to reduce power and logical complexity, approximation can be introduced in compressors using two approximation techniques. In the first approximation technique, the input and output count of the compressor are maintained. By changing output values, simple logical realisations are obtained. In second approximation technique, the output count is reduced from three to two by eliminating *CCOUT* as it is `1' only for one input combination `1111'. Introduction of this approximation technique allows pruning of *CCIN*. The output for second compressor approximation technique can be expressed as

$$C_{SUM} = IN_4IN_3IN_2IN_1 + IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1$$
$$C_{CY} = IN_4IN_3 + IN_2IN_1 + (IN_4 + IN_3)(IN_2 + IN_1)$$
$$IN_4 + IN_3 + IN_2 + IN_1 = C_{SUM} + 2 \times C_{CY}$$

IN_1	IN_2	IN_3	IN_4	Exact			Proposed Inexact	
				C_{COUT}	C_{CY}	C_{SUM}	C_{CY}	C_{SUM}
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	1
0	0	1	0	0	0	1	0	1
0	0	1	1	0	1	0	1	0
0	1	0	0	0	0	1	0	1
0	1	0	1	0	1	0	1	0
0	1	1	0	0	1	0	1	0
0	1	1	1	0	1	1	1	1
1	0	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1	0
1	0	1	0	0	1	0	1	0
1	0	1	1	0	1	1	1	1
1	1	0	0	0	1	0	**1**	**1**
1	1	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	**1**	**1**

TABLE 1. Truth table for proposed inexact 4:2 compressor.

C. PROPOSED INEXACT 4:2 COMPRESSOR

In this work, the second approximate technique for compressors is used. It is known that, the primitive/basic reversible logic gates with least QC performs XOR or NOT operations. Even though there are advanced reversible logic gates that can realise other logical operations such as AND, OR, NAND, NOR, etc, the QC of such gates are very high when compared to primitive/basic reversible logic gates. Therefore, a design can be best optimised for reversible logic gates-based implementation if the logical operations in a function are predominantly XOR operations. In the case of introducing approximation in compressors, the best approach would be to reduce the gate count (reduce the logical expression) by introducing minimum number of errors with minimum ED of +1 or -1.

In addition, reduction of error in cascaded compressor architectures (seen in the case of multipliers) can be achieved if equal number of errors are introduced [7]. Therefore, this paper takes into account the introduction of equal number of errors in positive and negative directions with minimum ED and to realise the circuit best suited for reversible logic-based implementation by modifying K-map based reduction technique. The proposed inexact compressor architecture using conventional logic gates is presented in Figure 3. In this design, maximum operations in the logical expression for *Carry* (*CCY*) and *Sum* (*CSUM*) is limited to XOR operations as they can be easily implemented using reversible logic gates with minimum QC and GC. The K-map for *CCY* and *CSUM* is shown in Figure 4. The *CSUM* and *CCY* of the proposed inexact 4:2 compressor can be represented by Eq. 10 and Eq. 11 respectively.

$$C_{SUM} = IN_2IN_1 + IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1$$
$$C_{CY} = IN_4IN_3 + IN_2IN_1 + (IN_4 \oplus IN_3)(IN_2 \oplus IN_1)$$

The truth table for the proposed 4:2 inexact compressor is presented in Table 1. The errors are introduced in input combinations `1100' and `1111'. For input combination `1100', ED is C1 and for `1111', ED is -1. Therefore, the ER is 2/16 = 12.5%. The errors introduced ensures equal deviation in +1 and -1 directions which aids in reducing the MED and MRED in multiplier implementations [7].

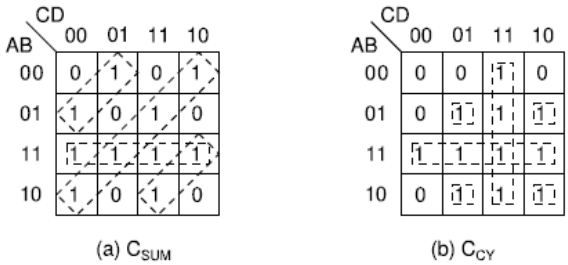


FIGURE 4. K-Map for CARRY and SUM of proposed 4:2 inexact compressor.

REVERSIBLE LOGIC AND IMPLEMENTATION OF COMPUTATIONAL UNITS USING REVERSIBLE LOGIC

An *n_n* gate is called reversible if there is a bijective mapping between the 2*n* input combinations and 2*n* output combinations. This is possible only when the number of outputs equal the number of inputs, contrary

to the surjective or injective mapping in conventional irreversible gates. A reversible circuit/system can be realised using reversible gates. However, reversible circuit realisation is different from conventional circuit realisation, as it does not allow fan-out and feedback from output to input. In this work, the reversible logic gates used are NOT gate, Toffoli gate [35], Feynman gate [36], Fredkin gate [37], Peres gate [38] and BJN gate [39]. A brief description of the gates is given below.

A. 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.

B. 2-BIT GATES

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

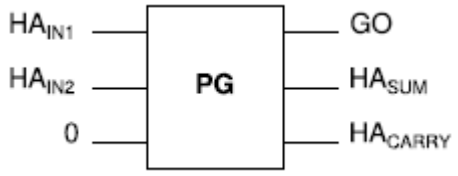


FIGURE 5. Half adder using reversible logic.

C. 3-BIT GATES

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

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

BJN gate - a 3-bit gate in which an input combination (A, B, C) produces (A, B, (ACB) _ 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 + B, AB + C). Peres gate has a QC of 4.

Four commonly used reversible logic realisation parameters are QC, GC. AI and GO [16] - [27].

Quantum Cost (QC) - refers to the number of primitive quantum gates in the circuit.

Gate Count (GC) - refers to the number of reversible gates in the circuit.

Ancilla Inputs (AI) - refers to the number of additional inputs included to attain physical reversibility.

Garbage Output (GO) - refers to the number of additional outputs included to make the circuit reversible.

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

D. HALF ADDER

Half adders are used to find the sum of two bits and have two inputs (*HAIN*1 and *HAIN*2) and two outputs (*HASUM* and *HACARRY*). The expression for *HASUM* and *HACARRY* are given in Eq. 12 and Eq. 13 respectively.

$$HA_{SUM} = HA_{IN1} \oplus HA_{IN2}$$

$$HA_{CARRY} = HA_{IN1} \cdot HA_{IN2}$$

Figure 5 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. Summary of the reversible logic realisation parameters is presented in Table 2.

E. FULL ADDER

To find the sum of three bits, a full adder is used. Full adder has three inputs (*FAIN*1, *FAIN*2 and *FACIN*) and two outputs (*FASUM* and *FACARRY*). *FASUM* and *FACARRY* can be expressed by Eq. 14 and Eq. 15 respectively.

$$FA_{SUM} = FA_{IN1} \oplus FA_{IN2} \oplus FA_{CIN}$$

$$FA_{CARRY} = (FA_{IN1} \oplus FA_{IN2}) \cdot FA_{CIN} + FA_{IN1} \cdot FA_{IN2}$$

Module Name	AI	GO	GC	QC
Half Adder	1	1	1	4

TABLE 2. Reversible logic realisation parameters for half adder.

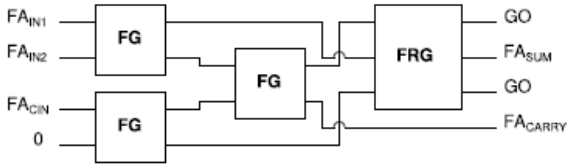


FIGURE 6. Full adder using reversible logic [40].

Module Name	AI	GO	GC	QC
Full Adder	1	1	4	8

TABLE 3. Reversible logic realisation parameters for a full adder.

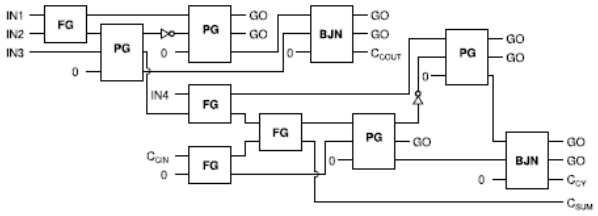


FIGURE 7. Exact 4:2 compressor using reversible logic.

Module Name	AI	GO	GC	QC
Exact Compressor	8	9	12	32

TABLE 4. Reversible logic realisation parameters for an exact compressor.

The reversible logic realisation of a full adder proposed by Pujar et al. [40] is used in this paper and is shown in Figure 6. The full adder realisation has three Feynman gates and one Fredkin gate with one ancilla input and two garbage outputs. Table 3 presents the summary of reversible logic realization parameters for a full adder.

F. EXACT 4:2 COMPRESSOR

The reversible logic realisation of an exact compressor is presented in Figure 7 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 realisation parameters for an exact compressor.

G. PROPOSED INEXACT 4:2 COMPRESSOR

The circuit realisation of proposed inexact 4:2 compressor using reversible logic gates is presented in Figure 8. The proposed 4:2 inexact compressor has three BJN gates and three Peres gates with six AI and eight GO. Table 5 presents the summary of reversible logic realisation parameters for proposed inexact 4:2 compressor.

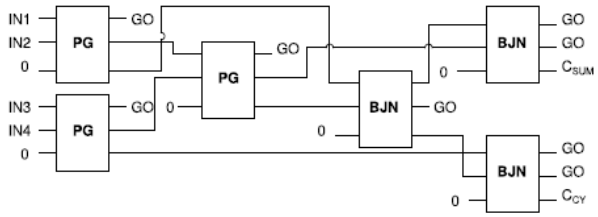
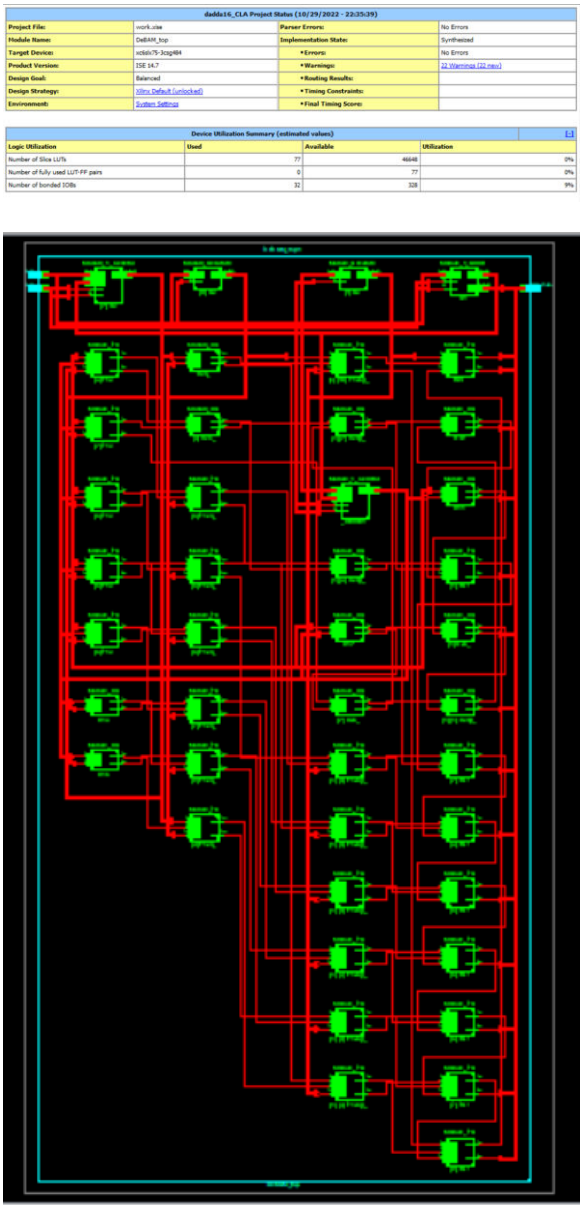


FIGURE 8. Proposed Inexact 4:2 compressor using reversible logic.

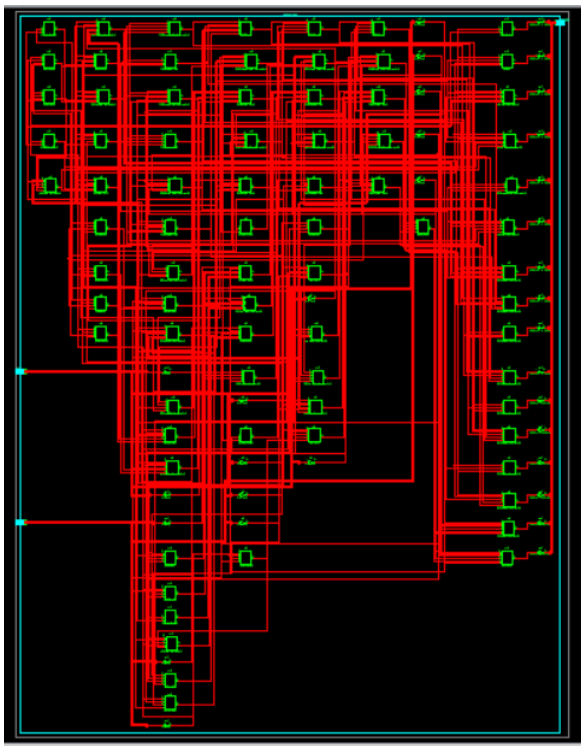
Module Name	AI	GO	GC	QC
Proposed Approximate 4 : 2 Compressor	6	8	6	27

TABLE 5. Reversible logic realisation parameters for proposed inexact 4:2 compressor.

SIMULATION RESULTS:



Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	0	93,296	0%	
Number of Slice LUTs	59	46,648	1%	
Number used as logic	59	46,648	1%	
Number using O6 output only	41			
Number using O5 output only	0			
Number using O5 and O6	18			
Number used as ROM	0			
Number used as Memory	0	11,072	0%	
Number of occupied Slices	23	11,662	1%	
Number of MUXCYs used	0	23,324	0%	
Number of LUT Flip Flop pairs used	59			
Number with an unused Flip Flop	59	59	100%	
Number with an unused LUT	0	59	0%	
Number of fully used LUT-FF pairs	0	59	0%	
Number of slice register sites lost to control set restrictions	0	93,296	0%	
Number of bonded IOBs	32	328	9%	
Number of RAMB16BWERs	0	172	0%	
Number of RAMB8BWERs	0	344	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	0	16	0%	
Number of DCM/DCM_CLKGENs	0	12	0%	
Number of ILOGIC2/ISERDES2s	0	442	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	442	0%	



CONCLUSION

This study proposes a unique design for the inexact Baugh-Wooley Wallace tree multiplier that is optimised for the realisation of reversible logic. In this, a 4:2 inexact compressor is presented, which, when compared to reversible logic-based realisation of existing state-of-the-art designs, has the least reversible logic realisation metrics. In two image processing applications—one level decomposition utilising a rationalised db6 wavelet filter bank and image smoothing and a CNN-based handwritten digit identification system—the suggested inexact compressor is used to realise the Baugh-Wooley Wallace tree multiplier. The SSIM was discovered to be the best in the case of the suggested design in both image processing applications. By applying a rationalised db6 wavelet filter bank and picture smoothing, the suggested design-based architectures were able to obtain SSIM values of 0.96 and 0.84 for one level decomposition, respectively. In a CNN-based application, accuracy was assessed to analyse the effectiveness of the multiplier design. The precision of the suggested design, which is 97.1%, is comparable to that of accurate multiplier-based architecture. The suggested design is therefore capable of achieving the optimal optimization in scales of reversible logic realisation parameters and is able to reach equivalent accuracy metrics with the precise Baugh-Wooley Wallace tree multiplier, according to the experimental study. With the addition of approximation, the suggested reversible logic realisation may be used to implement CNN-based applications and power-efficient image processing in quantum computing.

REFERENCES

[1] L. Dadda, ``Some schemes for parallel multipliers,'' Alta Frequenza, vol. 34, no. 5, pp. 349-356, Mar. 1965.
[2] C. S. Wallace, ``A suggestion for a fast multiplier,'' IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14-17, Feb. 1964.
[3] 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.
[4] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, ``Dual-quality 4:2 compressors for utilizing in dynamic accuracy congurable multipliers,'' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 4, pp. 1352-1361, Apr. 2017.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, ``Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers,'' IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9, pp. 3021-3034, Sep. 2020.
[10] 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.
[11] 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.
[12] P. Zakian and R. N. Asli, ``An efficient design of low-power and high-speed approximate compressor in FinFET technology," Comput. Electr. Eng., vol. 86, Sep. 2020, Art. no. 106651.
[13] G. E. Moore, ``Cramming more components onto integrated circuits," Electronics, vol. 38, no. 8, pp. 114-117, Apr. 1965.
[14] R. Landauer, ``Irreversibility and heat generation in the computing process," IBM J. Res. Develop., vol. 5, no. 3, pp. 183-191, Jul. 1961.
[15] C. H. Bennett, ``Logical reversibility of computation," IBM J. Res. Develop., vol. 17, no. 6, pp. 525-532, Nov. 1973.
[16] A. Chattopadhyay and A. Baksi, ``Low-quantum cost circuit constructions for adder and symmetric Boolean functions," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2016, pp. 2294-2297.
[17] S. Nowrin, L. Jamal, and H. M. H. Babu, ``Design of an optimized reversible bidirectional barrel shifter," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2016, pp. 730-733.
[18] I. Hashmi and H. M. H. Babu, ``An efficient design of a reversible barrel shifter," in Proc. 23rd Int. Conf. VLSI Design, Jan. 2010, pp. 93-98.
[19] M. Khan and J. E. Rice, ``Synthesis of reversible logic functions using ternary max-min algebra," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2016, pp. 1674-1677.
[20] M. H. A. Khan, ``Design of reversible synchronous sequential circuits using pseudo reed-muller expressions," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 11, pp. 2278-2286, Nov. 2014.
[21] H. Thapliyal and N. Ranganathan, ``Design of reversible sequential circuits optimizing quantum cost, delay and garbage outputs," ACM J. Emerg. Technol. Comput. Syst., vol. 6, no. 4, pp. 14:1-14:35, 2008.
[22] A. S. Molahosseini, A. Asadpoor, A. A. E. Zarandi, and L. Sousa, ``Towards efcient modular adders based on reversible circuits," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2018, pp. 1-5.
[23] H. M. Gaur, A. K. Singh, and U. Ghanekar, ``Testable design of reversible circuits using parity preserving gates," IEEE Des. Test. Comput., vol. 35, no. 4, pp. 5664, Aug. 2018.
[24] H. M. Gaur, A. K. Singh, and U. Ghanekar, ``Design of reversible arithmetic logic unit with built-in testability," IEEE Des. Test. Comput., vol. 36, no. 5, pp. 5461, Oct. 2019.
[25] K. Datta, I. Sengupta, and H. Rahaman, ``A post-synthesis optimization technique for reversible circuits exploiting negative control lines," IEEE Trans. Comput., vol. 64, no. 4, pp. 12081214, Apr. 2015.
[26] S. Raveendran, P. J. Edavoor, N. K. Y. Balachandra, and V. M. Harishchandra, ``Design and implementation of image kernels using reversible logic gates," IET Image Process., vol. 14, no. 16, pp. 41104121, Dec. 2020.
[27] S. Raveendran, P. J. Edavoor, N. Y. B. Kumar, and M. H. Vasantha, ``An approximate low-power lifting scheme using reversible logic," IEEE Access, vol. 8, pp. 183367-183377, 2020.
[28] R. Wille, D. Grosse, L. Teuber, G. W. Dueck, and R. Drechsler, ``RevLib: An online resource for reversible functions and reversible circuits," in Proc. Int. Symp. Multi-Valued Log., May 2008, pp. 220-225.
[29] C. R. Baugh and B. A. Wooley, ``A two's complement parallel array multiplication algorithm," IEEE Trans. Comput., vol. C-22, no. 12, pp. 1045-1047, Dec. 1973.
[30] K. Hwang, Computer Arithmetic: Principles, Architecture, and Design. Hoboken, NJ, USA: Wiley, 1979.
[31] F. Cavanagh, Digital Computer Arithmetic: Design and Implementation. New York, NY, USA: McGraw-Hill, 1984.
[32] E. L. Braun, Digital Computer Design: Logic, Circuitry, and Synthesis. New York, NY, USA: Academic, 2014.
[33] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, ``Design and analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984994, Feb. 2015.
[34] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, ``IMPACT: IMPrecise adders for low-power approximate computing," in Proc. IEEE/ACM Int. Symp. Low Power Elect. Design, Aug. 2011, pp. 409414.
[35] T. Toffoli, ``Reversible computing," in Automata, Languages and Programming, J. de Bakker and J. van Leeuwen, Eds. Berlin, Germany: Springer, 1980, pp. 632-644.
[36] R. P. Feynman, ``Quantum mechanical computers," Opt. News, vol. 11, pp. 11-20, Feb. 1985.
[37] E. Fredkin and T. Toffoli, ``Conservative logic," Int. J. Theor. Phys., vol. 21, nos. 3-4, pp. 219253, 1982.
[38] A. Peres, ``Reversible logic and quantum computers," Phys. Rev. A, Gen. Phys., vol. 32, no. 6, pp. 3266-3276, Dec. 1985.
[39] L. Gopal, N. S. M. Mahayadin, A. K. Chowdhury, A. A. Gopalai, and A. K. Singh, ``Design and synthesis of reversible arithmetic and Logic Unit (ALU)," in Proc. IEEE Int. Conf. Comput., Commun., Control Technol. (I4CT), Sep. 2014, pp. 289-293.

[40] J. Pujar, S. Raveendran, T. Panigrahi, M. H. Vasantha, and Y. B. N. Kumar, ``Design and analysis of energy efficient reversible logic based full adder," in Proc. IEEE 62nd Int. Midwest Symp. Circuits Syst. (MWSCAS), Aug. 2019, pp. 339-342.