

Received July 8, 2021, accepted July 15, 2021, date of publication July 28, 2021, date of current version August 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100892

# Inexact Signed Wallace Tree Multiplier Design Using Reversible Logic

SITHARA RAVEENDRAN<sup>1</sup>, PRANOSE J. EDAVOOR<sup>2</sup>,  
Y. B. NITHIN KUMAR<sup>1</sup>, (Senior Member, IEEE),  
AND M. H. VASANTHA<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Department of Electronics and Communication Engineering, National Institute of Technology Goa, Farmagudi, Goa 403401, India

<sup>2</sup>Department of Electrical and Electronics Engineering, National Institute of Technology Goa, Farmagudi, Goa 403401, India

Corresponding author: Sithara Raveendran (sithararaveendrann@gmail.com)

This Publication is an outcome of the R&D work supported by Visvesvaraya Ph.D. Scheme, Ministry of Electronics and Information Technology (MeitY), Govt. of India: MEITY-PHD-2152, and the Ministry of Education, Govt. of India for scholarship support.

**ABSTRACT** This paper proposes an inexact Baugh-Wooley Wallace tree multiplier with novel architecture for inexact 4:2 compressor optimised for realisation using reversible logic. The proposed inexact 4:2 compressor has  $\pm 1$  Error Distance (ED) and 12.5% Error Rate (ER). The efficacy of the proposed reversible logic based realisation of the proposed inexact 4:2 compressor and Baugh-Wooley Wallace tree multiplier is measured in scales of Gate Count (GC), Quantum Cost (QC), Garbage Output (GO) and Ancilla Input (AI). The proposed inexact 4:2 compressor is able to reduce reversible logic realisation metrics GC, QC, GO and AI by 50%, 15%, 25% and 11.11% as compared to reversible logic realisation of exact 4:2 compressor. An  $8 \times 8$  Baugh-Wooley Wallace tree multiplier is implemented in this paper. 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. MED and MRED of the proposed multiplier is 59.16 and 0.0109 respectively. The proposed multiplier is utilised in two applications 1) image processing - one level decomposition using rationalised db6 wavelet filter bank and image smoothing and 2) Convolutional Neural Networks (CNN). The efficacy of the proposed multiplier in image processing applications is estimated by measuring Structural Similarity Index Measure (SSIM) which is found to be 0.96 and 0.84 for image decomposition and smoothing respectively. In CNN based application, the efficiency is measured in scales of accuracy and is found to be 97.1%.

**INDEX TERMS** Inexact Wallace tree multipliers, Baugh-Wooley algorithm, inexact 4:2 compressor, reversible logic, image processing, wavelet transform, convolutional neural networks.

## I. 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 optimised 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

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang<sup>1</sup>.

and 3) final partial product addition. Partial product accumulation stage contributes to the overall delay and hence research has been carried out to optimise 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-optimised 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 realise regularly structured architectures compared to other topology like  $5 : 3$ ,  $7 : 2$ , 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 realisations are being researched.

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  $KT \ln 2$  joules of energy loss. This dissipation was insignificant in higher technologies [14]. As scaling of devices are happening at a rapid rate, the  $KT \ln 2$  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 realisation 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  $\log_n$  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 efficacy 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 efficacy 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  $\pm 2$  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).

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.

The paper is organised as follows. Section II briefs about Baugh-Wooley Wallace tree multipliers along with the basic computational units used for carrying out multiplication operation. Section III introduces the need for compressor and the proposed compressor design. Section IV discusses reversible logic implementation in detail and the design of basic computational units including the proposed reversible compressor using reversible logic gates. Experimental results are presented in V followed by concluding remarks in VI.

**II. 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.

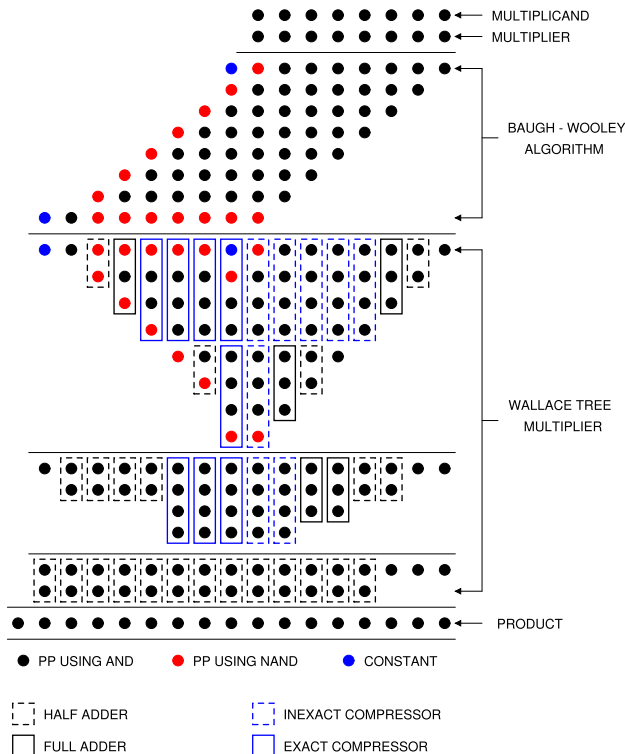
$$P = -p_{N-1}2^{N-1} + \sum_{x=0}^{N-1} p_x2^x \tag{1}$$

$$Q = -q_{N-1}2^{N-1} + \sum_{y=0}^{N-1} q_y2^y \tag{2}$$

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

$$\begin{aligned}
 P \times Q = & p_{N-1}q_{N-1}2^{2N-2} + \sum_{x=0}^{N-2} \sum_{y=0}^{N-2} p_xq_y2^{x+y} \\
 & + 2^{N-1} \left( -2^{N-1} + \sum_{y=0}^{N-2} \overline{p_{N-1}q_y}2^y + 1 \right) \\
 & + 2^{N-1} \left( -2^{N-1} + \sum_{x=0}^{N-2} \overline{q_{N-1}p_x}2^x + 1 \right) \tag{3}
 \end{aligned}$$

Constant ‘1’ is added at the 2N-1<sup>th</sup> and N<sup>th</sup> columns. MSB of the first N-1 partial product rows are inverted. All the elements in the N<sup>th</sup> row except the MSB is inverted. These defines the final partial product array in Baugh Wooley algorithm which is then reduced using any partial product reduction techniques.



**FIGURE 1.** Multiplication operation using Baugh Wooley Wallace tree multiplier.

**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.

**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.

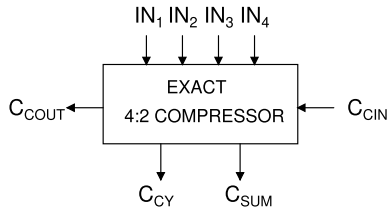


FIGURE 2. Exact 4:2 compressor.

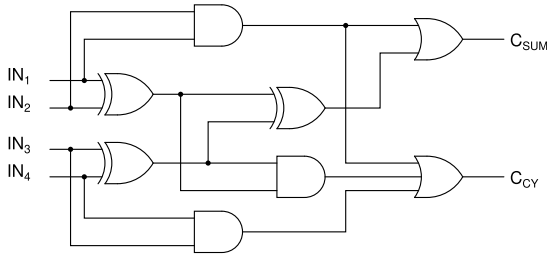


FIGURE 3. Proposed inexact 4:2 compressor using conventional logic gates.

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.

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.

### III. 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 find the number of logic ‘1’ in the input. The outputs can be represented as

$$C_{SUM} = C_{CIN} \oplus IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1 \quad (4)$$

$$C_{COUT} = IN_1(\overline{IN_2} \oplus \overline{IN_1}) + IN_3(IN_2 \oplus IN_1) \quad (5)$$

$$C_{CY} = IN_4(\overline{IN_4} \oplus \overline{IN_3} \oplus \overline{IN_2} \oplus \overline{IN_1}) + C_{CIN}(IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1) \quad (6)$$

*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

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

| IN <sub>1</sub> | IN <sub>2</sub> | IN <sub>3</sub> | IN <sub>4</sub> | Exact              |                 |                  | Proposed Inexact |                  |
|-----------------|-----------------|-----------------|-----------------|--------------------|-----------------|------------------|------------------|------------------|
|                 |                 |                 |                 | C <sub>COU</sub> T | C <sub>CY</sub> | C <sub>SUM</sub> | C <sub>CY</sub>  | C <sub>SUM</sub> |
| 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                  | 1               | 0                | 0                | 1                |

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 *C<sub>COU</sub>T* as it is ‘1’ only for one input combination ‘1111’. Introduction of this approximation technique allows pruning of *C<sub>CIN</sub>*. The output for second compressor approximation technique can be expressed as

$$C_{SUM} = IN_4IN_3IN_2IN_1 + IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1 \quad (7)$$

$$C_{CY} = IN_4IN_3 + IN_2IN_1 + (IN_4 + IN_3)(IN_2 + IN_1) \quad (8)$$

$$IN_4 + IN_3 + IN_2 + IN_1 = C_{SUM} + 2 \times C_{CY} \quad (9)$$

#### 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

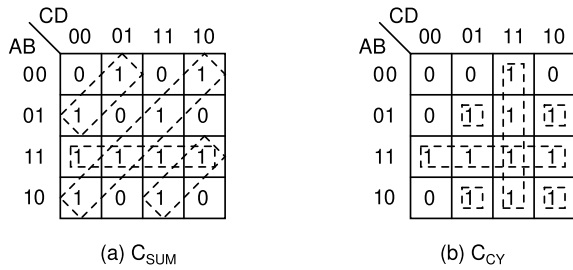


FIGURE 4. K-Map for CARRY and SUM of proposed 4:2 inexact compressor.

conventional logic gates is presented in Figure 3. In this design, maximum operations in the logical expression for Carry ( $C_{CY}$ ) and Sum ( $C_{SUM}$ ) is limited to XOR operations as they can be easily implemented using reversible logic gates with minimum QC and GC. The K-map for  $C_{CY}$  and  $C_{SUM}$  is shown in Figure 4. The  $C_{SUM}$  and  $C_{CY}$  of the proposed inexact 4:2 compressor can be represented by Eq. 10 and Eq. 11 respectively.

$$C_{SUM} = IN_2IN_1 + IN_4 \oplus IN_3 \oplus IN_2 \oplus IN_1 \quad (10)$$

$$C_{CY} = IN_4IN_3 + IN_2IN_1 + (IN_4 \oplus IN_3)(IN_2 \oplus IN_1) \quad (11)$$

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 +1 and for ‘1111’, ED is -1. Therefore the ER is  $\frac{2}{16} = 12.5\%$ . The errors introduced ensures equal deviation in +1 and -1 directions which aids in reducing the MED and MRED in multiplier implementations [7].

#### IV. REVERSIBLE LOGIC AND IMPLEMENTATION OF COMPUTATIONAL UNITS USING REVERSIBLE LOGIC

An  $n \times n$  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 BJJ 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,  $A \oplus B$ ) at the output of Feynman gate. FYG has a QC of 1.

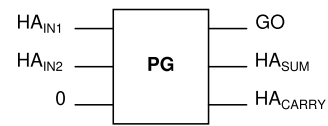


FIGURE 5. Half adder using reversible logic.

##### C. 3-BIT GATES

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

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

**BJJ 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 BJJ 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.

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 ( $HA_{IN1}$  and  $HA_{IN2}$ ) and two outputs ( $HA_{SUM}$  and  $HA_{CARRY}$ ). The expression for  $HA_{SUM}$  and  $HA_{CARRY}$  are given in Eq. 12 and Eq. 13 respectively.

$$HA_{SUM} = HA_{IN1} \oplus HA_{IN2} \quad (12)$$

$$HA_{CARRY} = HA_{IN1} \cdot HA_{IN2} \quad (13)$$

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 ( $FA_{IN1}$ ,  $FA_{IN2}$  and  $FA_{CIN}$ ) and two outputs ( $FA_{SUM}$  and  $FA_{CARRY}$ ).  $FA_{SUM}$  and  $FA_{CARRY}$  can be expressed

TABLE 2. Reversible logic realisation parameters for half adder.

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

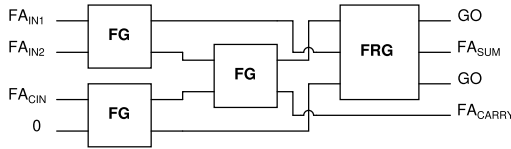


FIGURE 6. Full adder using reversible logic [40].

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

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

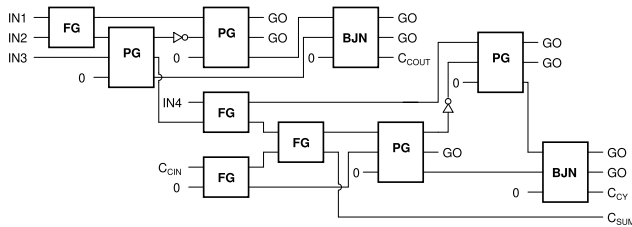


FIGURE 7. Exact 4:2 compressor using reversible logic.

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

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

by Eq. 14 and Eq. 15 respectively.

$$FA_{SUM} = FA_{IN1} \oplus FA_{IN2} \oplus FA_{CIN} \tag{14}$$

$$FA_{CARRY} = (FA_{IN1} \oplus FA_{IN2}) \cdot FA_{CIN} + FA_{IN1} \cdot FA_{IN2} \tag{15}$$

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 realisation 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 BJK 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 BJK gates and

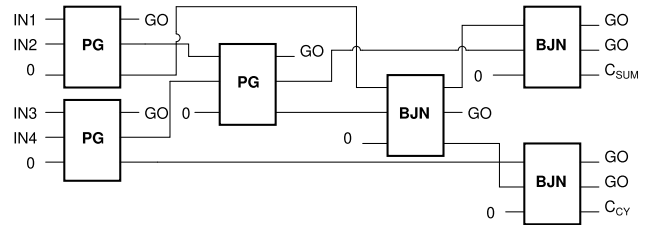


FIGURE 8. Proposed Inexact 4:2 compressor using reversible logic.

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

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

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.

## V. RESULTS

In this section, the proposed inexact 4:2 compressor is compared with the existing state-of-the-art 4:2 inexact compressor designs. Further, to analyse the efficacy of the proposed compressor, they are employed to realise an 8 × 8 Baugh-Wooley Wallace Tree multiplier. The multiplier designs are used in two applications - image processing and CNN based hand written digit recognition system.

### A. EXPERIMENTAL ANALYSIS OF PROPOSED COMPRESSOR DESIGN

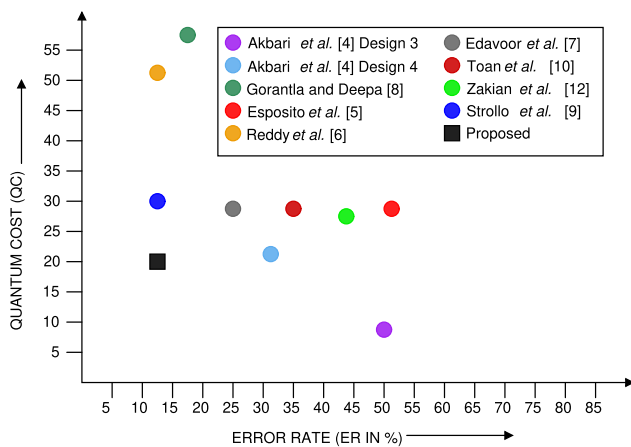
A comparison of ED and ER of proposed and existing inexact 4:2 compressor architectures is presented in Table 6. The inexact compressor design is said to be best optimised when the design is able to achieve minimum gate count with minimum ED and ER. Maximum ED of ‘2’ is for compressor designs proposed by Akbari *et al.* [4] and Gorantla and Deepa [8]. All the other designs have maximum ED ‘1’. Introducing error in equal numbers in +1 and -1 directions will reduce the MED and MRED in multiplier architectures where the inexact compressors are cascaded in partial product accumulation phase [7]. The proposed design for inexact compressor has an ED of ±1, with one error in +1 direction and one error in -1 direction. This approach aids in the reduction in MED and MRED in multiplier designs. A high ER of 62.5% is seen in the case of Akbari *et al.* [4] Design 1 and Design 2. High ER is not desirable for image processing applications.

Lowest ER is for proposed compressor design, compressors design proposed by Reddy *et al.* [6] and Strollo *et al.* [9]. The proposed inexact compressor design is able to achieve minimum ED and minimum ER among the other designs studied in this experiment which makes it ideal for image processing applications.

To measure the efficiency of the proposed design in reversible logic based realisation that is based on the

**TABLE 6.** Comparison of ED and ER for proposed and existing inexact 4:2 compressor architectures.

| Compressor Used                     | Max. ED | ER     |
|-------------------------------------|---------|--------|
| Akbari <i>et al.</i> [4] (Design 1) | $\pm 2$ | 62.5%  |
| Akbari <i>et al.</i> [4] (Design 2) | $\pm 2$ | 62.5%  |
| Akbari <i>et al.</i> [4] (Design 3) | -2      | 50%    |
| Akbari <i>et al.</i> [4] (Design 4) | +2      | 31.25% |
| Gorantla and Deepa [8]              | -2      | 18.75% |
| Esposito <i>et al.</i> [5]          | $\pm 1$ | 56.25% |
| Reddy <i>et al.</i> [6]             | $\pm 1$ | 12.5%  |
| Edavoor <i>et al.</i> [7]           | $\pm 1$ | 25%    |
| Toan <i>et al.</i> [10]             | +1      | 37.5%  |
| Zakian <i>et al.</i> [12]           | -1      | 43.75% |
| Strollo <i>et al.</i> [9]           | -1      | 12.5%  |
| <b>Proposed Compressor</b>          | $\pm 1$ | 12.5%  |

**FIGURE 9.** QC vs ER graph for proposed and existing 4:2 approximate compressor designs.

principles of quantum computing, reversible logic implementation parameters (GC, QC, AI and GO) are estimated for each compressor design and presented in Table 7. The best optimised design using reversible logic has minimum GC, minimum QC, minimum GO, and minimum AI. In exact 4:2 compressor,  $C_{COUT}$ ,  $C_{CY}$  and  $C_{SUM}$  are generated in five, eight and four stages respectively. Exact compressor design consists of two BJN gates, four Peres gates, four Feynman gates and two NOT gates which adds upto a GC of 12, QC of 32 with eight AI and nine GO. Among the inexact compressor designs, Reddy *et al.* [6] has the highest number of stages to generate  $C_{CY}$  and  $C_{SUM}$  as it has a MUX based architecture. This design has four BJN gates, two Toffoli gates, four Peres gates, four Feynman gates and one NOT gate. GC and QC count is high for compressor design by Gorantla and Deepa [8] as the design is not optimised of gate count. Esposito *et al.* [5], Akbari Design-3 [4] and Akbari Design-4 [4] generates  $C_{CY}$  and  $C_{SUM}$  in less number of stages when compared to other designs. However, these designs have high ER which makes them undesirable in real life applications. Among all the designs, proposed compressor design is able to generate  $C_{CY}$  and  $C_{SUM}$  in four stages

each with an ER as low as 12.5%. It can be observed that the reversible logic realisation parameters are also less for the proposed design, when compared to the other compressor designs. Therefore, the proposed design is the best optimised design for reversible logic realisation with a low ER. This makes the proposed design ideal for image processing and CNN based applications using reversible logic. To further analyse the efficiency of the proposed design in comparison with the existing compressor architectures, QC vs ER is plotted for each design. It can be observed that Strollo *et al.* [9], Reddy *et al.* [6] and Gorantla and Deepa *et al.* [8] have low ER, but QC is more than 30 in all these designs, which makes it less optimised for reversible logic realisations. Akbari *et al.* Design-3 [4] and Akbari *et al.* Design-4 [4] have the lowest QC, but have high ER of 50% and 31.25% respectively. Among all the designs, the best design with optimum ER and QC is the proposed design with a QC of 27 and ER of 12.5%.

## B. EXPERIMENTAL ANALYSIS OF PROPOSED INEXACT BAUGH-WOOLEY WALLACE TREE MULTIPLIER

In this work, an  $8 \times 8$  Baugh-Wooley Wallace tree multiplier is implemented employing full adders, half adders, exact and inexact compressors as shown in Figure 1. In exact compressor based multiplier design, twenty three half adders, four full adders and sixteen exact 4:2 compressors are used. In approximate compressor based design, twenty three half adders, four full adders, eight exact 4:2 compressors and eight inexact 4:2 compressors are used. Including the exact compressor based multiplier design, eleven Baugh-Wooley Wallace tree multipliers are implemented (MULT-1, MULT-2, ..., MULT-11) using reversible logic in this analysis. Table 8 shows the comparison for reversible logic realisation parameters for all the multiplier designs. MULT-1 (exact) has a GC of 295, QC of 892, AI count of 221 and GO count of 299. The reversible logic realisation parameters are the highest in the case of MULT-6 [6] and MULT-4 [8]. Both these designs have GC greater than three hundred and QC greater than thousand, which makes them undesirable in reversible logic based realisations. Multipliers MULT-2 [4], MULT-3 [4], MULT-5 [5] and MULT-9 [12] have lower reversible logic realisation parameters, when compared to other inexact compressor based multiplier designs, However, the ER in these compressors employed are greater than 31%, which is not desirable. The proposed design is able to achieve low GC as the above mentioned designs with a QC of 852. AI and GO counts are also comparable. The proposed design is able to achieve good optimisation in reversible logic realisations parameters.

Analogous to measuring ED and ER for inexact 4:2 compressors, MED, MRED, and error rate (in %) [6] are estimated for the proposed inexact Baugh Wooley Wallace tree multipliers. The difference of the exact and approximate output is referred to as the error distance (ED). For a multiplier design, ED is estimated for a set of input combinations, and average error distance is calculated and this average is

**TABLE 7.** Reversible logic realisation parameters for proposed reversible compatible approximate compressor with existing reversible logic based exact and approximate compressor designs.

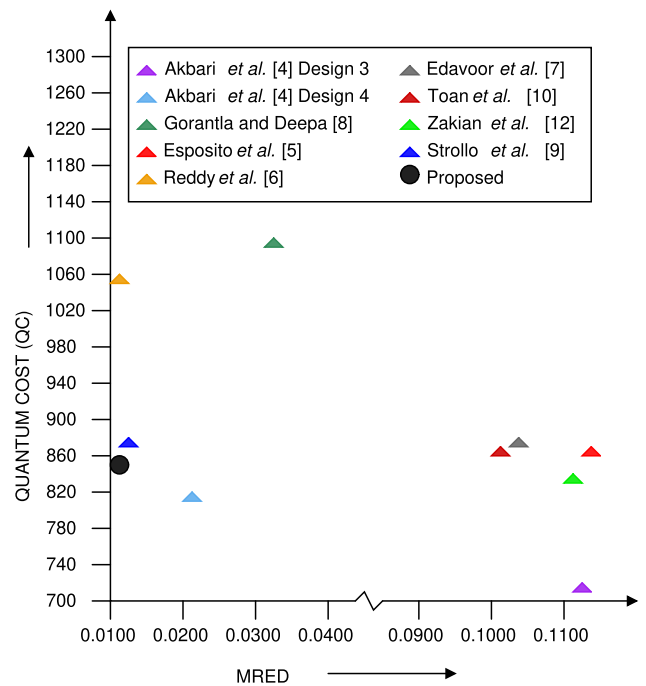
| Design                       | No. of Stages |           | Gates Used                           | Realisation Parameters |    |    |    |
|------------------------------|---------------|-----------|--------------------------------------|------------------------|----|----|----|
|                              | $C_{CY}$      | $C_{SUM}$ |                                      | GC                     | QC | AI | GO |
| Accurate                     | 8             | 4         | BJN(2), PG(4), FYG(4), NOT(2)        | 12                     | 32 | 8  | 9  |
| Design 3 - Akbari et al. [4] | 1             | 4         | PG(1), FYG(2), NOT(3)                | 6                      | 9  | 1  | 3  |
| Design 4 - Akbari et al. [4] | 4             | 4         | PG(4), NOT(6)                        | 10                     | 22 | 4  | 6  |
| Gorantla and Deepa [8]       | 5             | 7         | BJN(7), PG(3), TG(1), FYG(1), NOT(4) | 16                     | 57 | 12 | 14 |
| Esposito et al. [5]          | 3             | 3         | BJN(4), PG(2)                        | 6                      | 28 | 6  | 8  |
| Reddy et al. [6]             | 11            | 9         | BJN(4), TG(2), PG(4), FYG(4), NOT(1) | 15                     | 51 | 11 | 13 |
| Edavoor et al. [7]           | 1             | 6         | BJN(3), PG(3), FYG(1), NOT(1)        | 8                      | 29 | 6  | 7  |
| Toan et al. [10]             | 5             | 3         | BJN(4), PG(2)                        | 6                      | 28 | 6  | 8  |
| Zakian et al. [12]           | 3             | 5         | BJN(3), PG(2), NOT(1)                | 6                      | 24 | 5  | 6  |
| Strollo et al. [9]           | 5             | 3         | BJN(3), PG(3), FYG(3)                | 9                      | 30 | 7  | 9  |
| <b>Proposed Compressor</b>   | 4             | 4         | BJN(3), PG(3)                        | 6                      | 27 | 6  | 8  |

**TABLE 8.** Reversible logic realisation parameters for proposed reversible compatible approximate Baugh-Wooley Wallace tree multiplier with existing reversible logic based exact and approximate compressor designs.

| Compressor Design                     | Implementation Parameters |      |     |     |
|---------------------------------------|---------------------------|------|-----|-----|
|                                       | GC                        | QC   | AI  | GO  |
| MULT-1 Exact                          | 295                       | 892  | 221 | 299 |
| MULT-2 (Design 3 - Akbari et al. [4]) | 247                       | 708  | 165 | 251 |
| MULT-3 (Design 4 - Akbari et al. [4]) | 279                       | 812  | 189 | 275 |
| MULT-4 (Gorantla and Deepa [8])       | 327                       | 1092 | 253 | 339 |
| MULT-5 (Esposito et al. [5])          | 247                       | 860  | 205 | 291 |
| MULT-6 (Reddy et al. [6])             | 319                       | 1044 | 245 | 331 |
| MULT-7 (Edavoor et al. [7])           | 263                       | 868  | 205 | 283 |
| MULT-8 (Toan et al. [10])             | 247                       | 860  | 205 | 291 |
| MULT-9 (Zakian et al. [12])           | 247                       | 828  | 197 | 275 |
| MULT-10 (Strollo et al. [9])          | 271                       | 876  | 213 | 299 |
| MULT-11 (Proposed)                    | 247                       | 852  | 205 | 291 |

referred to as Mean Error Distance (MED). For an n-bit multiplier, MED refers to the average of error distances for all  $2^{2n}$  input combinations. MRED is the mean of ratios of ED and corresponding input for all  $2^{2n}$  input combinations. Finally error rate (in %) is also estimated which gives the number of erroneous outputs for all  $2^{2n}$  input combinations. The analysis is carried out for the existing and proposed inexact 4:2 compressor based multiplier architectures for seeded random 32000 input combinations and a comparison of the same is presented in Table 9.

MED is the highest for MULT-5 [5], MULT-2 [4], MULT-9 [12] and MULT-4 [8] due to high ER in the compressors used. Low MED is for MULT-11, MULT-6 [6], MULT-10 [9] and MULT-7 [7] as the ER in these design are low. The least MED among all the designs are for the proposed compressor based MULT-11. The proposed MULT-11 has an MED of 59.16. MRED is the highest for MULT-5 [5] and is 0.1144. MULT-2 [4], MULT-9 [12] and MULT-8 [10] also have high MRED which is not desirable for multiplier implementation. Low MRED is seen in MULT-11, MULT-6 [6], MULT-10 [9] and MULT-7 [7]. Among all the designs, lowest MRED is for the proposed compressor based MULT-11. High error rate of 95.8% and 94.6% is seen in the case of MULT-2 [4] and MULT-5 [5]. This is due to high ER of the compressors



**FIGURE 10.** QC vs MRED graph for proposed and existing 4:2 approximate compressor based Baugh-Wooley Wallace tree multiplier architectures.

employed. Low error rate are for MULT-10 [9], MULT-6 [6] and MULT-11. Among all these multipliers implemented, lowest error rate of 85.4% is for the proposed compressor based multiplier MULT-11. The accuracy metrics of the proposed compressor based MULT-11 makes it ideal for implementation of multipliers when compared to the existing state-of-the-art 4:2 approximate compressor based multiplier designs. To analyse the efficacy of the proposed design when approximation is introduced, QC vs MRED is plotted for each design. It can be observed that low MRED is for MULT-6 [6], MULT-10 [9] and the proposed compressor based MULT-11. Among these designs, the proposed MULT-11 has the least QC which is desirable for reversible logic based realisation.



**TABLE 9.** Accuracy metrics for proposed reversible compatible inexact Baugh-Wooley Wallace tree multiplier with existing reversible logic based exact and inexact 4:2 compressor designs.

| Multiplier Design                                   | Accuracy metrics |        |                |
|---|------------------|--------|----------------|
|   | MED              | MRED   | Error Rate (%) |
| <b>MULT-1</b> (Accurate)                            | -                | -      | -              |
| <b>MULT-2</b> (Design 3 - Akbari <i>et al.</i> [4]) | 301.27           | 0.1118 | 95.8           |
| <b>MULT-3</b> (Design 4 - Akbari <i>et al.</i> [4]) | 144.91           | 0.0215 | 92.3           |
| <b>MULT-4</b> (Gorantla and Deepa [8])              | 280.91           | 0.0328 | 88.7           |
| <b>MULT-5</b> (Esposito <i>et al.</i> [5])          | 329.35           | 0.1144 | 94.6           |
| <b>MULT-6</b> (Reddy <i>et al.</i> [6])             | 59.78            | 0.0110 | 85.5           |
| <b>MULT-7</b> (Edavoor <i>et al.</i> [7])           | 70.54            | 0.0141 | 88.2           |
| <b>MULT-8</b> (Toan <i>et al.</i> [10])             | 160.48           | 0.1098 | 90.4           |
| <b>MULT-9</b> (Zakian <i>et al.</i> [12])           | 293.26           | 0.1106 | 91.6           |
| <b>MULT-10</b> (Strollo <i>et al.</i> [9])          | 65.83            | 0.0123 | 86.1           |
| <b>MULT-11</b> (Proposed )                          | 59.16            | 0.0109 | 85.4           |

### C. IMAGE PROCESSING APPLICATION

In this analysis, two image processing applications are explored to gauge the effectiveness of the proposed inexact Baugh-Wooley Wallace tree multiplier in real time applications. The two image processing applications are

- One level decomposition using rationalised db6 wavelet filter bank.
- Image smoothing.

Since these applications are implemented and analysed to measure the efficacy of the  $8 \times 8$  Baugh Wooley Wallace tree multiplier implemented using proposed inexact 4:2 compressors, the filter coefficients and pixels are scaled to 8-bit 2's complement representation.

#### 1) ONE LEVEL DECOMPOSITION USING RATIONALISED db6 WAVELET FILTER BANK

Multirate signal processing has garnered huge popularity in the recent times due to its multi resolution capabilities when compared to signal processing using conventional transform techniques like Fourier transform, Discrete Cosine transform, etc [41]. Wavelet transform is a transform technique that is used for multi-rate signal processing. Wavelet Filter Banks (FBs) are broadly classified into orthogonal and bi-orthogonal wavelet transforms. In orthogonal wavelet FBs, the synthesis and analysis are same which aids in energy preservation. However, orthogonal wavelet FBs are not symmetric. Bi-orthogonal wavelet FBs are symmetric, which improves regularisation in FBs.

Various orthogonal wavelet FBs like db<sub>1</sub>, db<sub>2</sub>, db<sub>3</sub>, ..., db<sub>*n*</sub> can be realised by varying the filter length. While db<sub>1</sub> has rational filter coefficients, all other higher order filter banks have irrational filter coefficients that creates an over head in terms of computational intensity, gate count and delay. In order to address this, rationalisation techniques were introduced to reduce floating point (irrational) operations to fixed point (rational) operations. Murugesan and Tay [41] have introduced a new rationalisation technique for orthogonal wavelet FB using lattice structure realisation. This paper realises the db6 wavelet FB realisation using the

rationalisation technique proposed by [41] by adding a slight modification to realise dyadic filter coefficients. In [41], the lattice parameters  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  for db6 wavelet FB are found to be  $[\frac{-9}{4}]$ ,  $[\frac{1}{2}]$  and  $[\frac{-3}{31}]$ , for which divider circuits are required to realise  $[\frac{-3}{31}]$ . To achieve a simple realisation using fixed point multipliers and adders that are computationally less intensive compared to dividers, this paper slightly modified  $\alpha_3$  to  $[\frac{-3}{32}]$ . This ensures shifter based implementation for division operation in reversible logic based realisations. This change slightly varies the perfect reconstruction conditions of wavelet FB but can be effectively used in error resilient multimedia applications.

#### 2) IMAGE SMOOTHING

A  $5 \times 5$  smoothing kernel  $IK_{sm}$  is applied on the input image  $Img_{input}$ .  $IK_{sm}$  can be represented as

$$IK_{sm} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 4 & 4 & 4 & 1 \\ 1 & 4 & 12 & 4 & 1 \\ 1 & 4 & 4 & 4 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (16)$$

The final image  $Img_{final}$  in terms of  $Img_{input}$  and  $IK_{sm}$  can be represented as

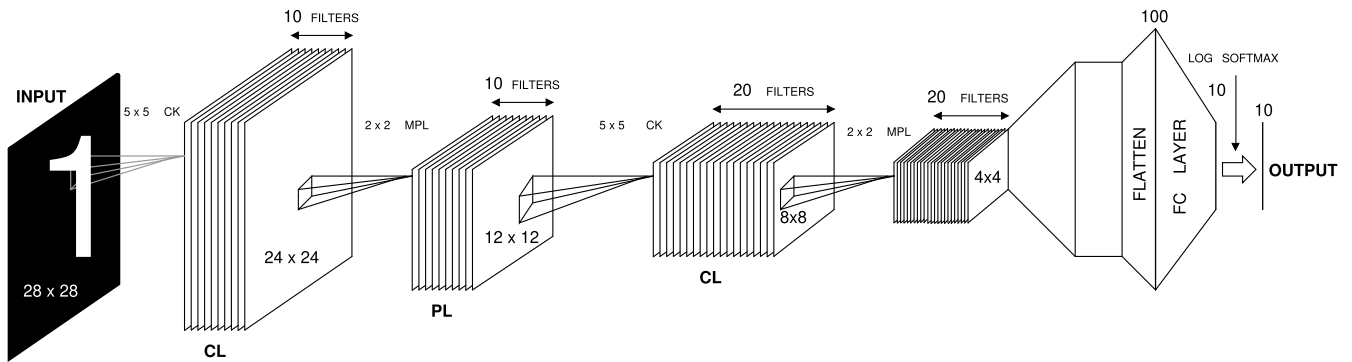
$$Img_{final}(a, b) = \frac{1}{60} \sum_{s=-2}^2 \sum_{t=-2}^2 \left( Img_{input}(a+s, b+t) \times IK_{sm}(s+3, t+3) \right) \quad (17)$$

#### 3) ANALYSIS OF EFFICACY OF PROPOSED BAUGH-WOOLEY WALLACE TREE MULTIPLIER IN IMAGE PROCESSING APPLICATIONS

To measure the accuracy of image processing units designed using approximate multipliers, SSIM is estimated. SSIM analyses local patterns of pixel intensities from the frame of reference of formation of images. Illumination and reflectance mainly determines the luminance of object surfaces being observed. However, the information about the structure of objects do not depend on illumination. Influence

**TABLE 10.** SSIM for one level decomposition using rationalised db6 wavelet filter bank and image smoothing application using proposed and existing inexact compressor based Baugh-Wooley Wallace tree multiplier architectures.

| Multiplier Used                       | SSIM  |                 |
|---------------------------------------|---|-----------------|
|                                       | One level decomposition using rationalised db6 wavelet FB | Image Smoothing |
| MULT-2 (Design 3 - Akbari et al. [4]) | 0.71  | 0.57            |
| MULT-3 (Design 4 - Akbari et al. [4]) | 0.88  | 0.64            |
| MULT-4 (Gorantla and Deepa [8])       | 0.89  | 0.69            |
| MULT-5 (Esposito et al. [5])          | 0.79  | 0.29            |
| MULT-6 (Reddy et al. [6])             | 0.95  | 0.74            |
| MULT-7 (Edavoor et al. [7])           | 0.91  | 0.74            |
| MULT-8 (Toan et al. [10])             | 0.89  | 0.43            |
| MULT-9 (Zakian et al. [12])           | 0.86  | 0.38            |
| MULT-10 (Strollo et al. [9])          | 0.95  | 0.82            |
| MULT-11 (Proposed )                   | 0.96  | 0.84            |



**FIGURE 11.** HDR system using LeNet 1 model.

of illumination has to be removed to obtain structural information. Therefore, structural information that contributes to the structure of objects in image is extracted by normalising it for contrast and luminance. SSIM can be represented by Eq. 18 [42].

$$SSIM(x, y) = \frac{(a\sigma_{xy} + K_2)(2\mu_y\mu_x + K_1)}{(\sigma_y^2 + \sigma_x^2 + K_2)(\mu_y^2 + \mu_x^2 + K_1)} \quad (18)$$

$\mu_x$  and  $\mu_y$  are mean intensities.  $\sigma_x$  and  $\sigma_y$  represent standard deviation. To avoid instabilities as  $\mu_x^2 + \mu_y^2$  approaches zero, two constants  $K_1$  and  $K_2$  are introduced. Table 10 presents the experimental results for image processing applications. For decomposition using wavelets, least SSIM of 0.71 and 0.79 are for MULT-2 [4] and MULT-5 [5] respectively, due to high ER. Highest SSIM of 0.96 is for MULT-11 which utilises the proposed 4:2 compressor. This is achieved due to the low ER and ED of the proposed design. MULT-6 [6] and MULT-10 [9] are also able to achieve high SSIM of 0.95. However, the proposed MULT-11 has higher SSIM than these designs. In smoothing application, MULT-5 [5] and MULT-9 [12] have the lowest SSIM due to high ER and ED. MULT-10 [9] also has one of the best SSIM of 0.82. The best SSIM is for MULT-11 with the proposed inexact 4:2 compressor with an SSIM of 0.84. Proposed MULT-11 is able to achieve high SSIM as the ER is kept at 12.5% and ED is  $\pm 1$  with equal number of deviations in +1 and -1 directions.

#### D. CNN APPLICATION

The Baugh-Wooley Wallace tree multipliers designed are utilised in convolutional kernels for Handwritten Digit Recognition (HDR) system. A CNN based implementation is carried out to realise Lenet-1 [43] handwritten digit recognition system using MNIST dataset [44]. Lenet-1 handwritten digit recognition system is shown in Figure. 11. In Figure. 11, CK represents Convolutional Kernels, CL represents Convolutional layers, PL represents Pooling layers, MPL represents Max Pooling layer and FCL represents Fully Connected Layer. The input images to the Lenet-1 model are handwritten digit images of size  $28 \times 28$ . These images are applied to first CNN layer of size  $24 \times 24 \times 10$  where they are processed using  $5 \times 5$  CK. The filters perform the multiply-accumulate operation on the three color components of each pixel in the  $28 \times 28$  images in parallel.

In these filters, the proposed inexact multiplier is utilised to perform multiplication operation with reduced delay and gate count.  $2 \times 2$  MPLs after the CNN layer results in pooling layers of size  $(12 \times 12 \times 10)$ . The output of MPL layer is fed to the second CNN layer  $(8 \times 8 \times 20)$ . To optimise the gate count, this layer utilises proposed inexact multipliers. The  $4 \times 4 \times 20$  output from the max pooling layer is flattened which is processed by two FCL to generate the final ten outputs using LogSoftmax.

**TABLE 11.** Accuracy measure for CNN application using proposed and existing inexact 4:2 compressor based Baugh-Wooley Wallace tree multiplier architectures.

| Multiplier Used                              | Accuracy % |
|--|------------|
| MULT-1 (Exact)                               | 97.6       |
| MULT-2 (Design 3 - Akbari <i>et al.</i> [4]) | 84.7       |
| MULT-3 (Design 4 - Akbari <i>et al.</i> [4]) | 86.6       |
| MULT-4 (Gorantla and Deepa [8])              | 81.7       |
| MULT-5 (Esposito <i>et al.</i> [5])          | 86.4       |
| MULT-6 (Reddy <i>et al.</i> [6])             | 97.1       |
| MULT-7 (Edavoor <i>et al.</i> [7])           | 96.2       |
| MULT-8 (Toan <i>et al.</i> [10])             | 88.3       |
| MULT-9 (Zakian <i>et al.</i> [12])           | 85.6       |
| MULT-10 (Strollo <i>et al.</i> [9])          | 96.1       |
| MULT-11 (Proposed)                           | 97.1       |

Sixty thousand images ( $28 \times 28 \times 3$ ) from MNIST database is used for training and validation. Batch size is 128 with 25 epochs. The inputs and trained parameters are scaled to corresponding 8-bit 2's complement representation. The output prediction using LogSoftmax results in a probability distribution from 0 to 1. The Softmax layer output with the maximum probability is considered as the final output. Inexact computation do not adversely affect the accuracy in such applications as long as the error rate and error distance is kept minimum. In addition, the probability driven operation of such systems also allows the introduction of inexact computations. Hence, CNN based applications can be used to verify the operations of inexact multipliers. The experimental analysis in this work is carried out using simulations in Matlab for two thousand random images. The efficacy is measured in scales of accuracy presented in %. The accuracy for various multiplier architectures is presented in Table 11. With exact compressor based MULT-1, accuracy obtained is 97.6%. The next best designs using approximate compressors are MULT-6 [6] and MULT-11 (based on proposed inexact 4:2 compressor). Both these designs have an accuracy of 97.1%. MULT-10 [9] and MULT-7 [7] also have comparable accuracy. Lowest accuracy is for MULT-4 [8], MULT-5 [5] and MULT-9 [12]. The proposed compressor based MULT-11 has comparable accuracy as MULT-1 (exact) and therefore can be used in CNN based applications.

## VI. CONCLUSION

A novel architecture for inexact Baugh-Wooley Wallace tree multiplier optimised for reversible logic realisation is proposed in this paper. In this, a 4:2 inexact compressor that has the least reversible logic realisation metrics when compared to reversible logic based realisation of existing state-of-the-art architectures is proposed. The proposed inexact compressor is utilised to realise Baugh-Wooley Wallace tree multiplier to perform multiplication operation in two image processing applications - one level decomposition using rationalised db6 wavelet filter bank and image smoothing and CNN based handwritten digit recognition system. In both the image processing applications, the SSIM was

found to be the best in the case of proposed design. The proposed design based architectures were able to achieve SSIM of 0.96 and 0.84 for one level decomposition using rationalised db6 wavelet filter bank and image smoothing respectively. Similarly, accuracy was estimated to analyse the efficiency of the proposed multiplier architecture in CNN based application. The proposed design is able to achieve an accuracy of 97.1% which is comparable with accurate multiplier based architecture. Therefore, from the experimental analysis, it can be concluded that the proposed design is able to achieve the best optimisation in scales of reversible logic realisation parameters and is able to achieve comparable accuracy metrics with the exact Baugh-Wooley Wallace tree multiplier. The proposed reversible logic realisation along with the introduction of approximation can find application in realising power efficient image processing and CNN based applications 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 configurable 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 efficient 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. 56–64, 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. 54–61, 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. 1208–1214, 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. 4110–4121, Dec. 2020.
- [27] S. Raveendran, P. J. Edavoor, N. Y. B. Kumar, and M. H. Vasanthan, "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. 984–994, 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. 409–414.
- [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. 219–253, 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. Vasanthan, 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.
- [41] S. Murugesan and D. B. H. Tay, "New techniques for rationalizing orthogonal and biorthogonal wavelet filter coefficients," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 3, pp. 628–637, Mar. 2012.
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [43] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [44] Y. LeCun, C. Cortes, and C. J. C. Burges. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>



**SITHARA RAVEENDRAN** received the bachelor's degree in electronics and communication engineering from Cochin University of Science and Technology, Kerala, India, in 2008, and the master's degree in very large-scale integration (VLSI) from the National Institute of Technology Goa, India, in 2016, where she is currently pursuing the Ph.D. degree with the Electronics and Communication Department. She has more than five years of industrial experience in design and verification of digital circuits (SOC level). Her research interests include digital design, reversible logic, FPGA, and ASIC design.



**PRANOSE J. EDAVOOR** received the bachelor's degree in electronics and communication engineering from Cochin University of Science and Technology, Kerala, India, in 2012, and the master's degree in very large-scale integration (VLSI) from the National Institute of Technology Goa, India, in 2016, where he is currently pursuing the Ph.D. degree with the Electrical and Electronics Department. His research interests include digital design, FPGA accelerators, ASIC design, wavelets, and deep neural networks.



**Y. B. NITHIN KUMAR** (Senior Member, IEEE) received the B.E. degree from Manipal Institute of Technology, Manipal, India, the M.Tech. degree from the National Institute of Technology Karnataka, Surathkal, India, and the Ph.D. degree from IIT Kharagpur, Kharagpur, India. He was a Visiting Researcher with the University of Pavia, Italy. He is currently an Assistant Professor with the National Institute of Technology Goa, India. He has published over 75 research papers in journals/conferences and filed several patents. He was a recipient of the Young Indian Researcher Award from the Government of Italy, in 2007 and 2010, the Young Scientists in Engineering Sciences by DST, Government of India, in 2015, and the Young Faculty Research Fellowship (YFRF) of Visvesvaraya Ph.D. Program of Ministry of Electronics and Information Technology, MeitY, Government of India, in 2018.



**M. H. VASANTHAN** (Member, IEEE) received the B.Tech. degree in electrical and electronics engineering from the NMAM Institute of Technology, Nitte, India, in 1998, the M.Tech. degree in microelectronics and very large scale integration design from IIT Madras, Chennai, India, in 2003, and the Ph.D. degree from the Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, India, in 2014. He is currently an Associate Professor with the National Institute of Technology Goa, India. He has published over 75 research papers and filed several patents.

...