Journal of Electronic Testing (2022) 38:353–370
https://doi.org/10.1007/s10836-022-06010-1

CMOS Implementation and Performance Analysis of Known Approximate 4:2 Compressors

Parthibaraj Anguraj[1] · Thiruvenkadam Krishnan[1] · Saravanan Subramanian[2]

Received: 22 December 2021 / Accepted: 19 June 2022 / Published online: 26 July 2022

Abstract
Approximate computing is one of the emerging concepts in multimedia applications like image processing applications. In the research world, it is getting more attention from researchers. Because of sacrificing a smaller scale in the accuracy of the design, it reduces the circuit parameters like area complexity, delay, and power. The purpose of this work is to survey the Field-Programmable Gate Array (FPGA) and Application-Specific Integrated Circuit (ASIC) implementation of modified Dadda multiplier architecture using various approximate 4:2 compressor designs presented for the last few decades. Based on implementation outcomes, this survey examines the approximate modified Dadda multiplier design performance for its closeness to the exact computation. In addition, the comparison is carried out based on approximate 4:2 compressors performance, an error rate of the particular design, the accuracy analysis metrics of approximate multiplier and its area utilization, power consumption, and delay.

Keywords Arithmetic circuits · Approximate multiplier · Approximate compressor · Circuit parameters · ASIC and FPGA.

1 Introduction

An approximate computing scheme is the most attractive and preferable for multimedia applications [25]. Especially in image processing applications [13], the inaccurate outcome gives an acceptable performance with lesser hardware costs. Consequently, numerous scientists from diverse nations have been getting significant attention to creating more circuits using an approximating methodology than exact computation [17]. During the design of approximate circuits, the researchers reduce the circuit parameters by tolerating the accuracy of the overall performance. The researchers try to maintain a good trade-off between circuit performance and precision [15]. Even though it is one of the biggest challenges, researchers present various approximate computing-based circuits to the scientific world sequentially [20].

Generally, the Dadda multiplier is the basic arithmetic unit with two variables: multiplier and multiplicand. It follows three stages to produce the final output: partial product generation, partial product reduction, and final product computation [3]. The Full Adder (FA) and Half Adder (HA) blocks are typically used to get the final output in the conventional multiplier design. However, in recent days, the authors are introduced various mapping styles to minimize the partial product values. It means that to find out the result, the grouping of the partial product value varies depending on the author's novelty. As a result, the exact 4:2 compressor block is presented by researchers to get the final outcome collaboration with FA and HA unit. As mentioned earlier, the approximation concept is the well-known scheme used in error-tolerate applications [4]. Hence, the authors have developed an approximate 4:2 compressor instead of an exact 4:2 compressor. Not only approximate 4:2 compressor, but also various kinds of the compressor such as approximate 5:2 compressor, approximate 7:2 compressor, and 15:2 compressor [21], etc. are designed in the last two decades.

Responsible Editor: S. T. Chakradhar

✉ Parthibaraj Anguraj
parthi63raja@gmail.com

✉ Thiruvenkadam Krishnan
thiruvlsi@gmail.com

Saravanan Subramanian
saran.vpm@gmail.com

[1] Department of Electronics and Communication Engineering, K.Ramakrishnan College of Technology, Samayapuram, Trichy, Tamil Nadu 621112, India

[2] Department of Electronics and Communication Engineering, K. S. Rangasamy College of Technology, Tamil Nadu 637215 Namakkal, India

Though the varieties of the approximate compressor are continuously presented in the research environment, there is still presenting research gap between circuit performance and accuracy. So, the authors are still working to build further modifications in the exact 4:2 compressor based on the approximate methodology for emerging real-time applications. One of the modified architectures of the approximate Dadda multiplier is presented in Fig. 1. In the first stage of Fig. 1, the dots represent the partial product values produced by the AND gate. After that, to get the final outcome (P0-P15) in the remaining stages, the FA, HA, and approximate 4:2 compressor are used. The HA has two inputs (*A* and *B*) and produces two outputs (*Sum* and *Carry*). The FA has three inputs (*A*, *B*, and *C*) and produces two outputs (*Sum* and *Carry*). The output expression of HA is given in Eqs. (1) and (2) and FA output expression in Eqs. (3) and (4).

$$Sum = A \oplus B \tag{1}$$

$$Carry = AB \tag{2}$$

$$Sum = A \oplus B \oplus C \tag{3}$$

$$Carry = AB + BC + CA \tag{4}$$

1.1 The Motivation of the Work

Many scientists from various countries are researching approximation-based arithmetic circuits for error-resilient applications like image processing applications. Among them, as the approximate Dadda multiplier role is essential, this is chosen by us for detailed survey. In this work, the modified Dadda multiplier architecture performance parameters are analysed by using various approximate 4:2 compressors. As a result, this survey paper will be helpful for those who are all doing research related to the performance of approximate 4:2 compressor concepts. The end of this work highlights the area utilization, delay, power, error rate of the particular approximate compressor design, and the accuracy metric (Mean Error Distance (MED), Mean Relative Error Distance (MRED), Normalize MED (NED), and the Number of Correct Outputs (#CO)) of the approximate multiplier design.

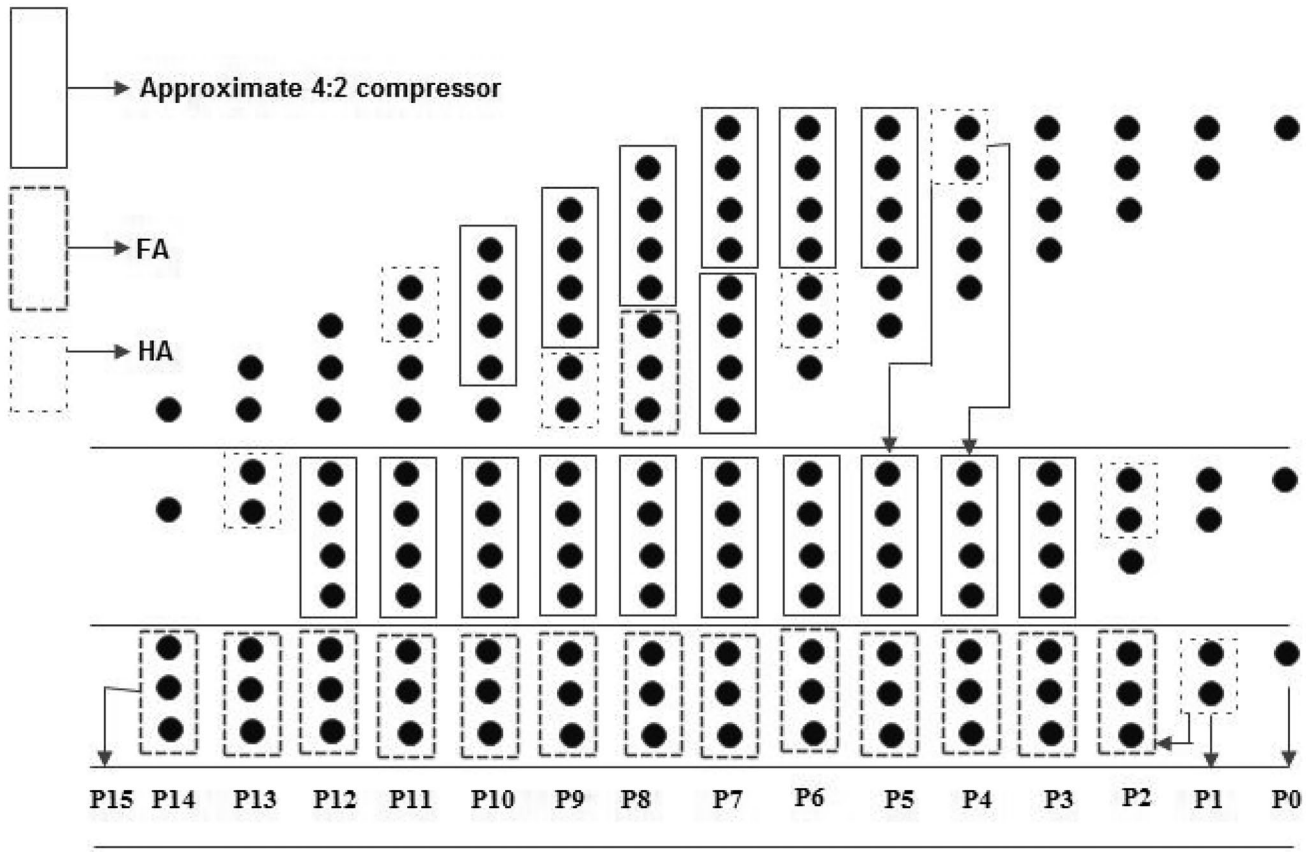

Fig. 1 The approximate 8 × 8 Dadda Multiplier reduction circuitry [19]

1.2 The Organisation of the Work

This survey work is arranged as follows. In Sect. 2, the varieties of approximate 4:2 compressors and their corresponding truth table are discussed. Then, simulation and implementation results of the Dadda multiplier using an approximate 4:2 compressor are tabulated in Sect. 3. After that, the work summary is reported in the Sect. 4.

2 Related Survey on Various Kinds of 4:2 Compressor

This section explains the exact and approximate 4:2 compressor designs presented at the national and international levels. As the approximate 4:2 compressor provides reasonable performance with less computational complexity, in recent trends, instead of the exact 4:2 compressor, these kinds of approximate 4:2 compressors are majority used in the Dadda multiplier.

2.1 The Exact 4:2 Compressor

The exact 4:2 compressor is built by using the two full adder designs [6], which have five inputs $(X_1, X_2, X_3, X_4, C_{in})$ and three outputs $(C_{out}, Carry, Sum)$. Figure 2 exhibits the exact 4:2 compressor design, and the output expressions of the same are given in Eqs. (5–7). In addition, Table 1 exhibits the truth table of the exact 4:2 compressor.

$$Sum = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in} \tag{5}$$

$$C_{out} = (X_1 \oplus X_2)X_3 + \left(\overline{X_1 \oplus X_2}\right)X_1 \tag{6}$$

$$Carry = (X_1 \oplus X_2 \oplus X_3 \oplus X_4)C_{in} + \overline{(X_1 \oplus X_2 \oplus X_3 \oplus X_4)}X_4 \tag{7}$$

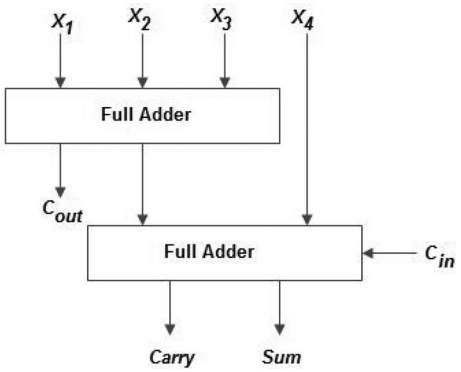

Fig. 2 The exact 4:2 compressor [6]

2.2 The Approximate 4:2 Compressors

As mentioned earlier, in error-resilient applications, to reduce the circuit parameters, the authors have designed approximate 4:2 compressors with an acceptable error rate. It is constructed by modifying the truth table of the exact outcome. Here, the more than thirty various approximate 4:2 compressors, which are proposed by various scientists from various countries are discussed. The Boolean output expressions of approximate 4:2 compressors are mentioned in Tables 2, 3 and 4. For multimedia applications, the researchers from Taiwan designed the inaccurate 4-2 counter to minimize the partial product steps in the Wallace multiplier [16]. The proposed inaccurate 4-2 compressor is implemented using TSMC 180nm standard cell library and compared with work [18]. Based on the implementation results, the presented inaccurate 4-2 counter has reduced the area and delay by 9.30% and 7.01%, respectively. In [23], two approximate compressors are designed with different input and output counts. The first design has five inputs and three outputs and the second one has four inputs and two outputs. It is specially designed for the Dadda multiplier, and the proposed designs are simulated using 32 nm, 22 nm, and 16 nm technologies. In [35], the authors from china are designed three different inaccurate 4:2 compressors in the year 2015. Compared with previous works, these proposed designs have notably reduced the area and power while attaining a high signal-to-noise ratio (SNR). High speed and low power consumption approximate 4:2 compressors were built by researchers from Iran in 2016, detailed in work [2].

For better improvement, Anusha gorantla and his team have designed three different types of 4:2 approximate compressors with less error rate [11]. Not only concentrate on the 4:2 approximate compressors, but this team also designed two various 5:2 approximate compressors for the Dadda multiplier. These proposed designs were simulated and implemented using 180 nm, 45 nm, and 90 nm technologies. Compared with Momeni et al. research work [23], the proposed designs have effectively reduced the area, power, and delay with reasonable performance. Further, in 2017, the authors from Canada have designed the power and area efficient approximate multiplier using approximation concept-based half adder, full adder, and 4:2-compressor. It is explained in the paper [34]. In a letter [10], for the modified Dadda multiplier structure, the power-efficient approximate 4:2 compressor is proposed and compared with the research article [35]. There are three regions of operations (truncation region, accurate region, and approximate region) are introduced in this letter by researchers to find out the final product outcome.

For image filtering applications, in 2018, the authors from Italy designed three different sizes of the compressor (4-2 compressor,5-3 compressor, and 6-3 compressor) based on

Table 1 Truth table of exact 4:2 compressor

C_{in}	X_4	X_3	X_2	X_1	C_{out}	*Carry*	*Sum*
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0
0	0	1	0	0	0	0	1
0	0	1	0	1	1	0	0
0	0	1	1	0	1	0	0
0	0	1	1	1	1	0	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	0
0	1	1	0	1	1	0	1
0	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	0
1	0	0	1	1	1	0	1
1	0	1	0	0	0	1	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	0	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	0	0	1	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	1	1
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

Table 2 Boolean expression of approximate 4:2 compressors (Set-1)

Ref	*Sum′*	*Carry′*
[16]	$X_1 \oplus X_2 \oplus X_3 \oplus X_4$	$(X_1 + X_2)(X_3 + X_4) + X_4X_3 + X_2X_1$
[23] D_2	$\left(\overline{X_1 \oplus X_2} + \overline{X_3 \oplus X_4}\right)$	$\overline{\left(\overline{X_1X_2} + \overline{X_3X_4}\right)}$
[35] D_1	$\overline{X_4X_3}(X_2 \oplus X_1) + \overline{X_2X_1}(X_4 \oplus X_3) + X_3(X_2X_1 + X_4X_1 + X_4X_2) + X_4X_2X_1$	$X_2X_1 + X_3(X_1 + X_2) + X_4(X_3 + X_1 + X_2)$
[35] D_2	$\overline{X_4X_3}(X_2 \oplus X_1) + X_4X_3 + (X_4 + X_3)(X_2 \odot X_1)$	$X_2X_1 + X_3(X_1 + X_2) + X_4(X_3 + X_1 + X_2)$
[35] D_3	$\overline{X_4X_3}(X_2 \oplus X_1) + (X_4 + X_3)(X_2 \odot X_1)$	$X_2X_1 + X_3(X_1 + X_2) + X_4(X_3 + X_1 + X_2)$
[2] D_1	X_1	X_4
[2] D_3	$\overline{(X_1 \odot X_2)(X_3 \odot X_4)}$	X_4
[2] D_4	$\overline{(X_1 \odot X_2)(X_3 \odot X_4)}$	$\overline{\left(\overline{X_1X_2}\right)\left(\overline{X_3X_4}\right)}$
[11] D_1	$X_1X_2 + X_3X_4 + \overline{X_1X_2}(X_3 + X_4) + \overline{X_3X_4}(X_1 + X_2)$	$(X_1 + X_2)(X_3 + X_4)$
[11] D_2	$X_4X_3 + (X_3 + X_4)\overline{(X_2 \oplus X_1)} + \overline{X_4X_3}(X_2 \oplus X_1)$	$Carry' = (X_1 + X_2)(X_3 + X_4)$
[11] D_3	$X_2X_1(X_3 + X_4) + X_4X_3(X_1 + X_2) + \overline{X_4X_3}(X_1 \oplus X_2) + \overline{X_2X_1}(X_4 \oplus X_3)$	$(X_1 + X_2)(X_3 + X_4)$

Table 3 Boolean expression of approximate 4:2 compressors (Set-2)

Ref	*Sum′*	*Carry′*
[34]	$(X_1 \oplus X_2) + (X_3 \oplus X_4) + X_4X_3X_2X_1$	$(X_4X_3 + X_2X_1)$
[10]	$(X_1 \oplus X_2) \oplus (X_3 + X_4)$	$X_2 + X_1X_3 + X_1X_4 + X_2X_3 + X_2X_4$
[4]	$(X_4 + X_3) + (X_2 + X_1)$	$X_4X_3 + X_2X_1$
[9]	$\overline{X_4X_3}(X_2 \oplus X_1) + \overline{X_2X_1}(X_4 \oplus X_3)$	$X_1(X_2 + X_3 + X_4) + X_2(X_3 + X_4) + X_4X_3$
[19]	$(X_1 \oplus X_2 \oplus X_3)\overline{X_4} + (X_3X_2 + X_3X_1 + X_2X_1)X_4$	$(X_3X_2 + X_3X_1 + X_2X_1)\overline{X_4} + X_4$
[1]	$X_1 + X_2 + X_3 + X_4$	$(X_1 + X_2)(X_3 + X_4)$
[8]	$(X_4 \oplus X_3)X_2X_1 + \overline{(X_4 \oplus X_3)}(X_2 + X_1)$	$X_4 + X_3$
[28]	$1'b1$	$X_4(X_1 + X_3) + (X_1X_3)$
[29]	$1'b1$	$X_1(X_3 + X_4)$
[32] D_1	$(X_4X_3X_2X_1) + (X_4 \oplus X_3 \oplus X_2 \oplus X_1)$	$(X_4 + X_3)(X_2 + X_1) + X_4X_3 + X_2X_1$
[32] D_2	$(X_2 + X_1) + (X_4 \oplus X_3)$	$(X_4 + X_3)(X_2 + X_1) + X_4X_3$
[27]	$X_2X_1 + X_4X_3 + \overline{X_4X_3}(X_2 + X_1) + \overline{X_2X_1}(X_3 + X_4)$	$(X_1 + X_2)(X_3 + X_4)$

Table 4 Boolean expression of approximate 4:2 compressors (Set-3)

Ref	*Sum′*	*Carry′*	C_{out}
[27]	$X_4 \oplus (X_3 + X_1X_2) \oplus (X_1 + X_2)$	$X_4(X_3 + X_1X_2) + (X_3 + X_1X_2)(X_1 + X_2) + (X_1 + X_2)X_4$	
[27] D_1	$X_4X_3 + (X_4 + X_3)(X_2 + X_1) + X_2X_1$	-	-
[27] D_2	$(X_4 + X_3)(X_2 + X_3)$	-	-
[27] D_3	$(X_1 + X_3)$	-	-
[23] D_1	$\overline{C_{in}}\left(\overline{X_1 \oplus X_2} + \overline{X_3 \oplus X_4}\right)$	C_{in}	$\overline{\left(\overline{X_1X_2} + \overline{X_3X_4}\right)}$
[2] D_2	X_1	X_4	X_3
[7]	$\overline{((X_3 \oplus X_2) \odot X_1)(X_5 \odot X_4)}$	$\overline{\left(\overline{X_3X_2}\right)\overline{(X_3 \oplus X_2)X_1}}$	X_5X_4
[31]	C_{in}	X_4	$X_2X_1 + X_3X_1 + X_3X_2$

Table 5 Truth table of Approximate 4:2 compressors (Set-1)

Ref	[6]	[19]		[23] D_2		[2] D_1		[2] D_3		[2] D_4		[10]		[35] D_1		[35] D_2		[35] D_3	
$X_4X_3X_2X_1$	*CS*	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E
0000	00	00	0	**01**	+1	00	0	00	0	00	0	00	0	00	0	00	0	00	0
0001	01	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0
0010	01	01	0	01	0	**0**0	−1	01	0	01		01	0	01	0	01	0	01	0
0011	10	10	0	**01**	−1	**01**	−1	**00**	−2	10	0	10	0	10	0	10	0	10	0
0100	01	01	0	01	0	**00**	−1	**1**1	+2	01	0	01	0	01	0	01	0	01	0
0101	10	10	0	10	0	**01**	−1	**1**1	+1	**01**	−1	10	0	10	0	10	0	10	0
0110	10	10	0	10	0	**0**0	−2	**1**1	+1	**01**	−1	10	0	10	0	10	0	10	0
0111	11	11	0	11	0	**0**1	−2	11	0	11	0	11	0	11	0	11	0	11	0
1000	01	**10**	+1	01	0	**10**	1	01	0	01	0	01	0	01	0	01	0	01	0
1001	10	10	0	10	0	**1**1	1	**01**	−1	**01**	−1	10	0	10	0	10	0	10	0
1010	10	10	0	10	0	10	0	**01**	**−1**	**01**	−1	10	0	10	0	10	0	10	0
1011	11	11	0	11	0	11	0	**0**1	−2	11	0	11	0	11	0	11	0	11	0
1100	10	10	0	**01**	−1	10	0	10	0	10	0	**01**	−1	10	0	**1**1	+1	**1**1	+1
1101	11	11	0	11	0	11	0	11	0	11	0	**1**0	−1	11		11	0	**1**0	−1
1110	11	11	0	11	0	**1**0	−1	11	0	11	0	**1**0	−1	11	0	11	0	**1**0	−1
1111	00	**11**	−1	**11**	−1	11	−1	**1**0	−2	**1**0	−2	**11**	−1	**11**	−1	**11**	−1	**11**	−1

Table 6 Truth table of Approximate 4:2 compressors (Set-2)

Ref	[32]D_1		[32]D_2		[34]		[26]D_1		[26]D_2		[26]D_3		[9]		[4]		[8]	
$X_4X_3X_2X_1$	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E
0000	00	0	00	0	00	0	0	0	0	0	0	0	00	0	00	0	00	0
0001	01	0	01	0	01	0	**0**	−1	**0**	−1	**1**	0	01	0	01	0	01	0
0010	01	0	01	0	01	0	**0**	−1	**0**	−1	**0**	−1	01	0	01	0	01	0
0011	10	0	**01**	−1	10	0	**1**	−1	**0**	−2	**1**	−1	10	0	1**1**	1	**01**	−1
0100	01	0	01	0	01	0	**0**	−1	**0**	−1	**1**	0	01	0	01	0	**10**	+1
0101	10	0	1**1**	1	**01**	−1	**1**	−1	**1**	−1	**1**	−1	10	0	**01**	−1	10	0
0110	10	0	1**1**	0	**01**	−1	**1**	−1	**1**	−1	**1**	−1	10	0	**01**	−1	10	0
0111	11	0	11	0	11	0	**1**	−2	**1**	−2	**1**	−2	1**0**	−1	11	0	11	0
1000	01	0	01	0	01	0	**0**	−1	**0**	−1	**0**	−1	01	0	01	0	**10**	+1
1001	10	0	1**1**	1	**01**	−1	**1**	−1	**1**	−1	**1**	−1	10	0	**01**	−1	10	0
1010	10	0	1**1**	1	**01**	−1	**1**	−1	**1**	−1	**0**	−2	10	0	**01**	−1	10	0
1011	11	0	11	0	11	0	**1**	−2	**1**	−2	**1**	−2	1**0**	−1	11	0	11	0
1100	10	0	10	0	10	0	**1**	−1	**0**	−2	**1**	−1	10	0	1**1**	1	10	0
1101	11	0	11	0	11	0	**1**	−2	**1**	−2	**1**	−2	1**0**	−1	11	0	11	0
1110	11	0	11	0	11	0	**1**	−2	**1**	−2	**1**	−2	1**0**	−1	11	0	11	0
1111	**11**	**−1**	11	**−1**	**11**	**−1**	**1**	**−3**	**1**	−3	**1**	−3	**10**	−2	**11**	−1	**11**	−1

inaccurate computing methods [9]. These proposed designs are simulated using 40 nm technology, and the proposed approximate 4-2 compressor is compared with research work [2, 23, 34]. Based on the implementation results, the presented designs were maintained a good trade-off between error performance and circuit complexity. Researchers from Taiwan designed the novel approximate 4:2 compressor for image sharpening applications and implemented it using 90 nm technology [7]. When compared with previous approximate multipliers presented in [2, 6, 9, 10, 23], this work has effectively reduced power and delay with acceptable circuit complexity.

A few years back, area and energy-efficient approximate compressors (4-2 & 5-2) were designed using FinFET, and simulations were done by HSPICE at 7 nm technology [1]. With the acceptable error rate, the proposed imprecise 4-2 compressor has effectively minimized the area, delay, and power when compared with previous methods [2, 4, 5, 10, 11, 23, 34]. Similarly, the proposed approximate 5-2 compressors have minimized the circuit parameter significantly compared to research work [11] and [33]. These comparisons are tabulated in [1]. Ferdos Salmanpour and his team have built a hybrid approximate 4-2 compressor for image multiplication applications. This proposed design is

Table 7 Truth table of Approximate 4:2 compressors (Set-3)

Ref	[11] D_1		[11] D_2		[11] D_1		[27]		[28]		[29]		[30]		[16]		[1]	
$X_4X_3X_2X_1$	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E	*CS*	E
0000	00	0	00	0	00	0	00	0	0**1**	+1	0**1**	+1	00	0	00	0	00	0
0001	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0
0010	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0
0011	**01**	−1	**0**0	−2	**0**0	−2	**01**	−1	**01**	−1	**01**	−1	10	0	10	0	**01**	−1
0100	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0
0101	10	0	10	0	10	0	10	0	1**1**	+1	1**1**	+1	10	0	10	0	1**1**	+1
0110	10	0	10	0	10	0	10	0	**01**	−1	**01**	−1	10	0	10	0	1**1**	+1
0111	11	0	11	0	11	0	11	0	11	0	11	0	1**0**	−1	11	0	11	0
1000	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0	01	0
1001	10	0	10	0	10	0	10	0	1**1**	+1	1**1**	+1	10	0	10	0	1**1**	+1
1010	10	0	10	0	10	0	10	0	**01**	−1	**01**	−1	10	0	10	0	1**1**	+1
1011	11	0	11	0	11	0	11	0	11	0	11	0	11	0	11	0	11	0
1100	**01**	−1	**01**	−1	**0**0	−2	**01**	−1	1**1**	+1	**01**	−1	10	0	10	0	**01**	−1
1101	11	0	11	0	11	0	11	0	11	0	11	0	11	0	11	0	11	0
1110	11	0	11	0	11	0	11	0	11	0	0**1**	−2	11	0	11	0	11	0
1111	**11**	−1	**11**	−1	**11**	−1	**11**	−1	**11**	−1	**11**	−1	**11**	−1	1**0**	−2	**11**	−1

implemented in 7-nm FinFET technology and it has better results than previous works [1, 2, 4, 5, 8, 10, 11, 19, 23, 28, 34]. This comparison is detailed in research work [29]. The FPGA-based multi-level approximate multiplier is constructed in [32] for error-resilient applications. Approximately 3-2 and 4-2 compressor is designed to reduce partial product outcomes. In the paper [31], the majority based approximate 4-2 compressor has been carried out using FinFET and Quantum-dot cellular automata technologies by authors from Iran for digital signal processing applications. The high-performance 4-2 approximate compressor is designed using the synapsis 90 nm technology library [27]. Furthermore, the low power approximate 4-2 compressors are designed in [30] and [26] for multi-media applications. Compared with the exact 4-2 compressor, the proposed designs have reasonable performance with less complexity.

In this paper, the Sum',$Carry'$, C_{out} and $(C_{in}, X_1, X_2, X_3, X_4)$ denotes the output and input of the approximate compressor. Also, $\oplus, \odot, +$ indicates XOR, XNOR and OR logical operators. The truth table of the approximate 4:2 compressor designs and their Error distances (E) are mentioned in Tables 5, 6, 7, and 8. In these tables, the error values are highlighted in bold. Further, The error rate and pass rate of the corresponding approximate 4:2 compressor designs are mentioned in Table 9.

The error rate and pass rate is calculated as follows [11]:

$$\text{Error rate in } (\%) = \frac{\text{Number of error conditions}}{\text{Total number of conditions}} \times 100 \tag{8}$$

$$\text{Pass rate in } (\%) = (1 - \text{Error rate}) \times 100 \tag{9}$$

$$\text{Error Distance (E)} = \text{Approximate Outcome} - \text{Exact Outcome} \tag{10}$$

During the design, the authors are designed the approximate compressors with different input and output counts. In Tables 5–8, the input (X_4, X_3, X_2, X_1) and output $(Carry(C), Sum(S))$ count of the approximate 4:2 compressors are four and two. In the research articles [, 6, 23], and [31], the researchers have designed the approximate compressors to the length of five inputs $(C_{in}, X_4, X_3, X_2, X_1)$ and three outputs $(C_{out}(C_0), Carry(C), Sum(S))$.The Error distance (E) is calculated by the difference between the inaccurate outcome of the approximate compressor and the exact outcome [23].

3 Simulation and Implementation Results

In this session, FPGA and ASIC implementation of approximate Dadda multipliers is presented. Before that, the approximate 4:2 compressor results are tabulated in Tables 10 and 11. More than thirty various approximate 4:2 compressors are taken for this comparison. These are synthesized in three different types of FPGA families, such as

Table 8 Truth table of Approximate 4:2 compressors (Set-4)

$X_5X_4X_3X_2X_1$	C_oCS	C_oCS	E	C_oCS	E
00000	000	**001**	1	000	0
00001	001	001	0	**000**	−1
00010	001	001	0	**000**	−1
00011	100	**001**	−1	100	0
00100	001	001	0	**000**	−1
00101	100	100	0	100	0
00110	100	100	0	100	0
00111	101	101	0	**100**	−1
01000	001	001	0	**010**	1
01001	010	100	0	010	0
01010	010	100	0	010	0
01011	101	101	0	**110**	1
01100	010	**001**	−1	010	0
01101	101	101	0	**110**	1
01110	101	101	0	**110**	1
01111	110	**101**	−1	110	0
10000	001	**010**	1	001	0
10001	010	010	0	**001**	−1
10010	010	010	0	**001**	−1
10011	101	**010**	−1	101	0
10100	010	010	0	**001**	−1
10101	101	**110**	1	101	0
10110	101	**110**	1	101	0
10111	110	110	0	**101**	−1
11000	010	010	0	**011**	1
11001	011	**110**	1	011	0
11010	011	**110**	1	011	0
11011	110	110	0	**111**	1
11100	011	**010**	−1	011	0
11101	110	110	0	**111**	1
11110	110	110	0	**111**	1
11111	111	**110**	−1	111	0

Table 9 Error rate and Pass rate of the Approximate Compressor

Ref	Total conditions	Number of Error conditions	Error occurred conditions $(C_{in}X_4X_3X_2X_1)$	Error rate	Error rate in (%)	Pass rate	Pass rate in (%)
[16]	16	1	1111	0.0625	6.25	0.9375	93.75
[23] D_1	32	12	00000,00011,01100,01111,10000,10011,10101,10110,11001,11010,11100,11111	0.375	37.5	0.625	62.5
[23] D_2	16	4	0000,0011,1100,1111	0.25	25	0.75	75
[35] D_1	16	1	1111	0.0625	6.25	0.9375	93.75
[35] D_2	16	2	1100,1111	0.125	12.5	0.875	87.5
[35] D_3	16	4	1100,1101,1110,1111	0.25	25	0.75	75
[2] D_1	16	10	0010,0011,0100,0101,0110,0111,1000,1001,1110,1111	0.625	62.5	0.9375	93.75
[2] D_2	16	8	0011,0100,0101,0110,1001,1010,1011,1111	0.5	50	0.5	50
[2] D_3	16	5	0101,0110,1001,1010,1111	0.3125	31.25	0.6875	68.75
[11] D_1	16	3	0011,1100,1111	0.1875	18.75	0.8125	81.25
[11] D_2	16	3	0011,1100,1111	0.1875	18.75	0.8125	81.25
[11] D_3	16	3	0011,1100,1111	0.1875	18.75	0.8125	81.25
[34]	16	5	0101,0110,1001,1010,1111	0.3125	31.25	0.6875	68.75
[10]	16	4	1100,1101,1110,1111	0.25	25	0.75	75
[4]	16	7	0011,0101,0110,1001,1010,1100,1111	0.4375	43.75	0.5625	56.25
[9]	16	5	0111,1011,1101,1110,1111	0.3125	31.25	0.6875	68.75
[19]	16	2	1000,1111	0.125	12.5	0.875	87.5
[7]	32	8	01001,01010,01100,01111,10001,10010,10100,10111	0.25	25	0.75	75
[1]	16	7	0011,0101,0110,1001,1010,1100,1111	0.4375	43.75	0.5625	56.25
[28]	16	8	0000,0011,0101,0110,1001,1010,1100,1111	0.5	50	0.5	50
[32]	16	1	1111	0.0625	6.25	0.9375	93.75
[32]	16	6	0011,0101,0110,1001,1010,1111	0.375	37.5	0.625	62.5
[8]	16	4	0011,0100,1000,1111	0.25	25	0.75	75
[31]	32	16	00001,00010,00100,00111,01000,01011,01101,01110,10001,10010,10100,10111,11000,11011,11101,11110	0.5	50	0.5	50
[27]	16	3	0011,1100,1111	0.1875	18.75	0.8125	81.25
[26]	16	15	Except 0000	0.9375	93.75	0.0625	6.25
[26] D_1	16	15	Except 0000	0.9375	93.75	0.0625	6.25
[26] D_2	16	13	Except 0000, 0001,0100	0.8125	81.25	0.8125	81.25
[29]	16	9	0000,0011,0101,0110,1001,1010,1100,1110,1111	0.5625	56.25	0.4375	43.75
[30]	16	2	0111,1111	0.125	12.5	0.875	87.5

Spartan-6 (xc6slx4-2tqg144), Virtex-4 (xc4vfx12-12sf363), and Virtex-5 (xc5vlx20t-2ff323). Also, the circuit parameters such as the number of LUTs, delay, and frequency are calculated. In this work, The $D_1, D_2, D_3, andD_4$ denotes the Design-1, Design-2, Design-3, and Design-4 of the circuits presented in corresponding research articles. Similarly, in Table 12, the approximate 4:2 compressors are implemented using 32nm, 45nm, 22nm, 16nm, 180nm, 90nm, 7nm FinFET technologies with different supply voltage at various operating frequencies. In addition, the following design parameters are figured: area requirement, critical path delay, and power consumption.

Table 10 FPGA Implementation of Approximate 4:2 Compressors Set-1

Family	Spartan-6			Virtex-4				Virtex-5		
Device	xc6slx4-2tqg144			xc4vfx12-12sf363				xc5vlx20t-2ff323		
Ref	Numberof LUT	Delay (ns)	Frequency (MHz)	Number of Slices	Number of 4 input LUTs	Delay (ns)	Frequency (MHZ)	Number of LUT	Delay (ns)	Frequency (MHz)
[6]	3	5.536	180.635	2	4	5.663	176.584	3	4.032	248.015
[16]	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[23] D_1	2	3.875	258.064	2	0	3.875	258.064	2	3.875	258.064
[23] D_2	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[35] D_1	-	4.372	228.728	-	-	4.275	233.918	-	3.124	320.102
[35] D_2	-	4.372	228.728	-	-	4.275	233.918	-	3.124	320.102
[35] D_3	1	5.439	183.857	1	1	4.989	200.441	1	3.875	258.064
[35] D_4	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[2]	2	6.177	161.890	1	2	4.989	200.441	2	3.875	258.064
[11] D_1	2	6.117	161.890	1	2	4.989	200.441	2	3.875	258.064
[11] D_2	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[11] D_3	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[34] D_1	2	6.177	161.890	1	2	4.989	200.441	2	3.875	258.064
[34] D_2	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[10]	2	6.177	161.890	1	2	4.989	200.441	2	3.875	258.064

Table 11 FPGA Implementation of Approximate 4:2 Compressors Set-2

Family	Spartan-6			Virtex-4				Virtex-5		
Device	xc6slx4-2tqg144			xc4vfx12-12sf363				xc5vlx20t-2ff323		
Ref	Number of LUT	Delay (ns)	Frequency (MHz)	Number of Slices	Number of 4 input LUTs	Delay (ns)	Frequency (MHZ)	Number of LUT	Delay (ns)	Frequency (MHz)
[4]	1	5.385	185.701	1	1	4.936	202.593	1	3.809	262.536
[9] D_1	1	5.402	185.116	1	1	4.976	200.964	1	3.871	258.331
[9] D_2	1	5.385	185.701	1	1	4.936	202.593	1	3.809	262.536
[9] D_3	1	5.259	190.150	1	1	4.858	205.846	1	3.696	270.562
[19]	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[7]	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[1]	2	5.402	185.116	1	2	4.976	200.964	2	3.871	258.331
[28] D_1	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[28] D_2	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[28] D_3	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[8]	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[31]	1	5.385	185.701	1	1	4.936	202.593	1	3.809	262.536
[27]	1	5.385	185.701	1	1	4.936	202.593	1	3.809	262.536
[26]	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[29]	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064
[30]	2	5.439	183.857	1	2	4.989	200.441	2	3.875	258.064

Table 12 ASIC Implementation of Approximate 4:2 Compressors

Ref	Area (μm^2)	Delay (ps)	Power (nW)
[19]	33.6	48.8	3540
[6]	42.6	95.7	5800
[23] D_1	34.0	47.6	3450
[23] D_2	31.0	40.6	3390
[2] D_1	4.02	12.2	860
[2] D_2	7.55	12.2	1080
[2] D_3	24.2	41.5	2630
[2] D_4	30.5	40.3	3410
[10]	35.4	47.1	3620
[12]	36.8	52.4	3780
[11]	37.6	61.84	3970
[9]	13.66	17.31	1830
[7]	29.42	44.3	3380
[8]	18.37	25.48	2480
	32nm Technology @ 1GHz		
[6]	-	60.36	2980
[23] D_1	-	58.32	1270
[23] D_2	-	44.35	1140
	22nm Technology @ 1GHz		
[6]	-	55.82	1500
[23] D_1	-	56.79	6200
[23] D_2	-	41.69	5800
	16nm Technology @ 1GHz		
[6]	-	47.59	9500
[23] D_1	-	37.16	3900
[23] D_2	-	24.44	3600
	180nm Technology		
[34]	186.28	810	5.15
[35]	199.58	860	6.57
[10]	259.46	870	5.99
[9]	96.47	570	2.72
[26] D_1	133.06	720	3.52
[26] D_2	59.88	510	1.88
[26] D_3	19.96	350	0.74
	180nm 0.9V@ 1GHz		
[23] D_1	77	426	2838
[23] D_2	83	362	3124
[11] D_1	73	442	2695
[11] D_2	110	899	4444
[11] D_3	173	737	6581
	90nm 1V@ 1GHz		
[23] D_1	20	104	902
[23] D_2	22	88	1163
[11] D_1	18	116	745
[11] D_2	29	242	1674
[11] D_3	39	157	1783

Table 12 (continued)

	HSPICE with 7nm FinFET 0.7V@ 2GHz		
[23]	0.307	19.55	188
[2]	0.307	20.01	190
[34]	0.443	25.99	270
[10]	0.329	29.91	214
[4]	0.176	28.61	137
[28]	0.105	17.21	70
[29]	0.070	14.9	60
[1]	0.125	18.9	116
[8]	0.198	20.1	331
[11]	0.355	24.7	262
[19]	0.484	32.5	294
[1]	0.125	18.9	116
[10]	0.329	29.9	214
[5]	0.484	34.4	464

3.1 Performance Analysis of the Approximate Dadda Multiplier

Using various CMOS technologies, in this sub-section, the approximate Dadda multipliers are synthesized and implemented. Notably, the researchers are introduced various mapping styles to reduce the partial product values. That means that not all researchers follow the same style to group the partial product values. It depends based on the novelty of the researchers. It is highlighted in the following tables. In Table 13, the comparison is carried out using various technologies and found the area requirement, delay, and power consumption of the particular design. Some of the approximate Dadda multiplier structures are synthesized and implemented using HSPICE and 7nm FinFET technology and calculated the circuit parameters such as transistor count, delay, and power consumption of the circuits tabulated in Table 14. Based on the comparison, it is clearly shown that the approximate multiplier has consumed less area and low power consumption than accurate computation.

3.2 Accuracy Metrics Analysis of the Approximate Dadda Multiplier

Even though the approximate multipliers have less circuit complexity, the accuracy of the outcome is slightly affected. However, this concept is mainly suited for error-tolerate applications like image processing applications due to providing acceptable meaningful results. The accuracy metrics analysis is clearly noticed that the approximate multiplier performances are how much near to the accurate computations. It is tabulated in Table 15. The metric accuracy parameters are found out by using the following equations:

$$MED = \frac{1}{2^{2n}} \sum_{i=1}^{2^{2n}} |ED_i| \tag{11}$$

$$\text{MRED} = \frac{1}{2^{2n}} \sum_{i=1}^{2^{2n}} \frac{|ED_i|}{S_i} \tag{12}$$

$$\text{NED} = \frac{\text{MED}}{\text{D}} \tag{13}$$

Table 13 Performance analysis of the approximate $N \times N$ Dadda multipliers

Utilized Approximate compressors are presented in work	The proposed multiplier structure is presented in work	$N \times N$	Area (μm^2)	Delay (ns)	Power (μW)
65 nm CMOS technology					
Accurate	[10]	8×8	674.56	3.04	215.3
[23]			620.80	2.79	176.0
[35]			521.28	2.70	172.4
[10]			510.08	2.70	164.6
Accurate	[10]	16×16	2667.52	5.90	977.2
[23]			2401.60	5.55	824.9
[35]			2101.12	5.07	798.5
[10]			2048.32	5.02	758.4
Accurate	[10]	32×32	10750.40	11.66	4520.6
[23]			9453.44	11.35	3620.5
[35]			8391.68	9.79	3587.4
[10]			8163.52	9.73	3414.9
TSMC 65 nm standard cell library					
Accurate	[34]	16×16	4859.28	0.68	1776.49
[34]			2158.56	0.47	503.15
[34]			3319.20	0.66	1102.03
[23]			2871.72	0.40	435.31
[23]			3782.16	0.63	1250.70
[24]			3953.88	0.69	1225.29
[36]			4547.52	0.64	1570.79
[14]			3938.00	0.67	1318.51
180 nm CMOS technology					
Accurate	[26]	8×8	8249.07	5.07	171.7
[23]			7344.69	4.57	115.9
[23]			7384.12	4.54	107.5
[34]			7610.80	4.54	107.7
[35]			7717.25	4.54	123.5
[10]			8089.80	4.54	126.6
[9]			6812.47	4.54	103.2
[26]			6586.27	4.33	087.9
[26]			6636.17	4.33	091.8
[26]			6380.04	4.33	086.4
[26]			6054.40	4.33	083.0
45-nm CMOS technology					
Accurate	[8]	8×8	3979	0.63	101
[19]			3312	0.40	65.74
[23]			3261	0.41	64.85
[23]			3092	0.37	59.69
[2]			682.2	0.29	33.53
[2]			713.8	0.28	34.51
[2]			2618	0.34	53.92
[2]			3070	0.42	61.47
[10]			3624	0.50	79.24
[11]			3715	0.49	80.19
[9]			834	0.31	39.22
[7]			3683	0.47	80.60

Table 13 (continued)

45-nm CMOS technology					
[8]			2468	0.33	51.01
180-nm CMOS technology					
[11]	[11]	8 × 8	1506	2.5	43.212
[11]			1656	4.1	45.064
[11]			1890	3.75	47.165
90-nm CMOS technology					
[11]	[11]	8 × 8	726	1.432	15.988
[11]			749	1.516	16.674
[11]			767	1.464	18.309
45-nm CMOS technology					
[11]	[11]	8 × 8	645	0.682	25.58
[11]			741	0.748	26.68
[11]			757	0.753	29.29
90-nm CMOS technology					
[6]	[7]	8 × 8	-	0.892	15.09
[23] D_1			-	0.576	09.83
[7]			-	0.625	10.09
[23] D_2			-	0.650	09.96
[2] D_4			-	0.735	11.33
[10]			-	0.880	12.99
[9]			-	0.578	09.42
HSPICE and 7nm FinFET technology					
[29]	[29]	8 × 8	0.070	0.0149	0.060
[1]			0.125	0.0189	0.116
[2]			0.307	0.0199	0.190
[4]			0.176	0.0286	0.137
[8]			0.198	0.0201	0.331
[11]			0.355	0.0247	0.262
[10]			0.329	0.0299	0.214
[23]			0.307	0.0195	0.188
[19]			0.484	0.0325	0.294
[28]			0.105	0.0175	0.071
[34]			0.443	0.0261	0.270
[5]			0.484	0.0344	0.464

Table 14 Performance analysis of the approximate Dadda multiplier (Set-2)

Utilized Approximate compressors are presented work in	The proposed multiplier structure is presented in work	$N \times N$	Transistors Count	Delay (ns)	Power (μW)
HSPICE and 7nm FinFET technology					
[28]	[28]	[8 × 8]	852	0.1434	7.27
[23]			1180	0.2224	10.94
[2]			1180	0.2226	11.14
[34]			1300	0.2365	11.73
[10]			1228	0.2445	12.73
[11]			1228	0.2331	12.85
[4]			1108	0.2398	10.06
[5]			1530	0.2528	18.79
[1]	[1]	[8 × 8]	944	0.2218	8.47
[1]			920	0.2218	7.40
[23]			1104	0.2224	9.32
[11]			1168	0.2329	11.82
[11]			1072	0.2272	9.73
[10]			1168	0.2441	11.78
[2]			1104	0.2226	10.11
[34]			1264	0.2365	10.81
[9]			1008	0.2398	8.80
[5]			1530	0.2528	18.79

Table 15 Accuracy metrics analysis of the approximate Dadda multipliers)

Utilized Approximate compressors are presented in work	The proposed multiplier structure is presented in work	MED	MRED	NED	#CO
[19]	[19]	502.9	0.0728	0.0077	19527
[23] D_1		3873.8	4.5483	0.0596	103
[23] D_2		3508.8	4.2843	0.0540	503
[2] D_1		4376	0.4424	0.0673	690
[2] D_2		2787.8	0.3135	0.0429	624
[2] D_3		3095.7	0.3907	0.0476	1885
[2] D_4		1376.6	0.0854	0.0212	11052
[10]		25.212	0.0326	0.0023	15863
[12]		1584.2	0.8264	0.0387	1216
[23] D_1	[2]	3999	4.5410	0.0613	111
[23] D_2		3520	4.2463	0.0541	458
[2] D_1		3540	0.3692	0.0545	968
[2] D_2		2400	0.2932	0.0370	974
[2] D_3		3220	0.3281	0.0496	3349
[2] D_4		1399	0.0809	0.0213	11012
[23] D_1	[10]	48.68	-	0.0007	4928
[35]		31.76	-	0.00048	11200
[10]		28.05	-	0.00043	11200
[22]	[31]	2520.87	-	0.038	-
[31]		3290.92	-	0.0506	-
[23] D_1	[7]	212.59	0.1302	0.0033	690
[7]		115.41	0.0159	0.0018	19766
[23] D_2		140.80	0.1095	0.0022	1263

Table 13 (continued)

Utilized Approximate compressors are presented in work	The proposed multiplier structure is presented in work	MED	MRED	NED	#CO
[2]		143.91	0.0205	0.0022	17779
[10]		49.05	0.0092	0.0008	37726
[9]		329.66	0.1142	0.0051	4096
[29]	[29]	1148	-	0.017	655
[1]		547	-	0.008	4587
[2]		644	-	0.009	5898
[4]		648	-	0.010	5242
[8]		759	-	0.012	1310
[11]		417	-	0.006	7208
[10]		326	-	0.005	8519
[23]		911	-	0.014	655
[19]		293	-	0.005	4587
[28]		924	-	0.014	655
[34]		623	-	0.009	5898
[1]	[1]	1260	-	0.019	-
[1]		1190	-	0.018	-
[23]		3477	-	0.054	-
[11]		892	-	0.014	-
[11]		1354	-	0.021	-
[10]		709	-	0.011	-
[2]		1319	-	0.020	-
[34]		1231	-	0.019	-
[4]		1363	-	0.021	-

4 Conclusion

This survey concludes with the FPGA and ASIC implementation of the modified Dadda multiplier design using various approximate 4:2 compressors. The modifications and development of arithmetic circuits using imprecise concepts are desired for emerging real-time applications in current trends. Moreover, especially in image processing applications, it is mostly preferred by many researchers. Hence, in this work, using various CMOS technologies, one of the arithmetic circuits (Dadda Multiplier) is taken for reviewing the performance based on multiple approximate 4:2 compressors.

Based on investigations, the following things are noticed by us.

1. In human realization-based applications like image processing applications, the slight deviation in the accuracy of the outcome is not considered the major issue. As a result, many researchers have taken the approximate computing concept to minimize the design complexity. In addition, it is one of the promising approaches in emerging real-time applications.
2. The performance analysis of the Dadda multiplier is done based on various aspects. For example, in partial product reduction, the kinds of 4:2 compressors are utilized, which act as a brilliant role to find out the final outcomes. So, firstly, this survey is focused on detailing the exact and approximate 4:2 compressor error rate and design parameters.
3. Based on approximate 4:2 compressors performance, the complexity of the Dadda multiplier structure is analysed and tabulated the design parameters. The results show that the approximate designs have attained reasonable performance with low circuit complexity than the exact outcome.
4. The accuracy matrix table shows that even though the authors introduced the error conditions to reduce the complexity of the design, the researchers maintain the trade-off between overall accuracy and circuit performance.

On the other hand, many researchers are still doing research related to modified multiplier design based on approximation concepts. This survey also focuses the CMOS implementation of known approximate multipliers. Through

accuracy metrics analysis, this survey is highlighted, how the approximate multipliers are close to exact computation in terms of acceptable area utilization, delay requirement and power consumption. So, we hope that this survey will be most helpful for researchers who are doing research related to error-resilient applications based on multipliers using approximate 4:2 compressor designs.

Data Availability Statement Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflicts of Interest I certify that there is no actual or potential conflict of interest in relation to this article.

References

1. Ahmadinejad M, Moaiyeri MH, Sabetzadeh F (2019) Energy and area efficient imprecise compressors for approximate multiplication at nanoscale. AEU-Int J Electron C 110:152859. https://doi.org/10.1016/j.aeue.2019.152859
2. Akbari O, Kamal M, Afzali-Kusha A, Pedram M (2017) Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers. IEEE Trans Very Large Scale Integr VLSI Syst 25(4):1352–1361. https://doi.org/10.1109/TVLSI.2016.2643003
3. Anguraj P, Krishnan T (2021) Design and implementation of modified BCD digit multiplier for digit-by-digit decimal multiplier. Analog Integr Circ Sig Process 107:1–12
4. Ansari MS, Jiang H, Cockburn BF, Han J (2018) Low-power approximate multipliers using encoded partial products and approximate compressors. IEEE J Emerging Sel Top Circuits Syst 8(3):404–416. https://doi.org/10.1109/JETCAS.2018.2832204
5. Arasteh A, Moaiyeri M, Taheri M, Navi K, Bagherzadeh N (2017) An energy and area efficient 4:2 compressor based on FinFETs. Integration 60. https://doi.org/10.1016/j.vlsi.2017.09.010
6. Chang C-H, Gu J, Zhang M (2004) Ultra low-voltage low-power CMOS 4–2 and 5–2 compressors for fast arithmetic circuits. IEEE Trans Circuits Syst Regul Pap 51(10):1985–1997. https://doi.org/10.1109/TCSI.2004.835683
7. Chang Y-J, Cheng Y-C, Lin Y-F, Liao S-C, Lai C-H, Wu T-C (2019) Imprecise 4–2 compressor design used in image processing applications. IET Circuits Devices Syst 13(6):848–856. https://doi.org/10.1049/iet-cds.2018.5403
8. Edavoor PJ, Raveendran S, Rahulkar AD (2020) Approximate multiplier design using novel dual-stage 4:2 compressors. IEEE Access 8:48337–48351. https://doi.org/10.1109/ACCESS.2020.2978773
9. Esposito D, Strollo AGM, Napoli E, De Caro D, Petra N (2018) Approximate multipliers based on new approximate compressors. IEEE Trans Circuits Syst Regul Pap 65(12):4169–4182. https://doi.org/10.1109/TCSI.2018.2839266
10. Ha M, Lee S (2018) Multipliers with approximate 4–2 compressors and error recovery modules. IEEE Embed Syst Lett 10(1):6–9. https://doi.org/10.1109/LES.2017.2746084
11. Gorantla A, Deepa P (2017) Design of approximate compressors for multiplication. ACM J Emerg Technol Comput Syst 13(3):44:1–17. https://doi.org/10.1145/3007649
12. Guo Y, Sun H, Guo L, Kimura S (2018) Low-cost approximate multiplier design using probability-driven inexact compressors. In: Proc. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2018. pp 291–294. https://doi.org/10.1109/APCCAS.2018.8605570
13. Kong T, Li S (2021) Design and analysis of approximate 4–2 compressors for high-accuracy multipliers. IEEE Trans Very Large Scale Integr VLSI Syst 29(10):1771–1781. https://doi.org/10.1109/TVLSI.2021.3104145
14. Kulkarni P, Gupta P, Ercegovac M (2011) Trading accuracy for power with an underdesigned multiplier architecture. In: Proc. 24th Internatioal Conference on VLSI Design. pp 346–351. https://doi.org/10.1109/VLSID.2011.51
15. Kumar UA, Chatterjee SK, Ahmed SE (2021) Low-power compressor-based approximate multipliers with error correcting module. IEEE Embed Syst Lett (Early Access) 1–1. https://doi.org/10.1109/LES.2021.3113005
16. Lin C-H, Lin I-C (2013) High accuracy approximate multiplier with error correction. In: Proc. IEEE 31st International Conference on Computer Design (ICCD). pp 33–38. https://doi.org/10.1109/ICCD.2013.6657022
17. Liu W, Zhang T, McLarnon E, O'Neill M, Montuschi P, Lombardi F (2021) Design and analysis of majority logic-based approximate adders and multipliers. IEEE Trans Emerg Top Comput 9(3):1609–1624. https://doi.org/10.1109/TETC.2019.2929100
18. Ma J, Man KL, Krilavičius T, Guan S, Jeong T (2011) Implementation of high performance multipliers based on approximate compressor design. In: Proc. 6th International Conference on Electrical and Control Technologies, ECT 2011. pp 96–100
19. Manikantta Reddy K, Vasantha MH, Nithin Kumar YB, Dwivedi D (2019) Design and analysis of multiplier using approximate 4-2 compressor. AEU-Int J Electron C 107:89–97. https://doi.org/10.1016/j.aeue.2019.05.021
20. Manikantta Reddy K, Vasantha MH, Nithin Kumar YB, Dwivedi D (2020) Design of approximate booth squarer for error-tolerant computing. IEEE Trans Very Large Scale Integr VLSI Syst 28(5):1230–1241. https://doi.org/10.1109/TVLSI.2020.2976131
21. Marimuthu R, Rezinold YE, Mallick PS (2017) Design and analysis of multiplier using approximate 15–4 compressor. IEEE Access 5:1027–1036. https://doi.org/10.1109/ACCESS.2016.2636128
22. Moaiyeri MH, Sabetzadeh F, Angizi S (2018) An efficient majority-based compressor for approximate computing in the nano era. Micro system Technologies 24(3):1589–1601. https://doi.org/10.1007/s00542-017-3587-2
23. Momeni A, Han J, Montuschi P, Lombardi F (2015) Design and analysis of approximate compressors for multiplication. IEEE Trans Comput 64(4):984–994. https://doi.org/10.1109/TC.2014.2308214
24. Narayanamoorthy S, Moghaddam HA, Liu Z, Park T, Kim NS (2015) Energy-efficient approximate multiplication for digital signal processing and classification applications. IEEE Trans Very Large Scale Integr VLSI Syst 23(6):1180–1184. https://doi.org/10.1109/TVLSI.2014.2333366
25. Park G, Kung J, Lee Y (2021) Design and analysis of approximate compressors for balanced error accumulation in MAC operator. IEEE Trans Circuits Syst Regul Pap 68(7):2950–2961. https://doi.org/10.1109/TCSI.2021.3073177
26. Pei H, Yi X, Zhou H, He Y (2021) Design of ultra-low power consumption approximate 4–2 compressors based on the compensation characteristic. IEEE Trans Circuits Syst Express Briefs 68(1):461–465. https://doi.org/10.1109/TCSII.2020.3004929

27. Rammohan SR, Jayashri N, Bivi MA, Nayak CK, Niveditha VR (2020) High performance hardware design of compressor adder in DA based FIR filters for hearing aids. Int J Speech Technol 23(4):807–814. https://doi.org/10.1007/s10772-020-09759-y
28. Sabetzadeh F, Moaiyeri MH, Ahmadinejad M (2019) A majority-based imprecise multiplier for ultra-efficient approximate image multiplication. IEEE Trans Circuits Syst Regul Pap 66(11):4200–4208. https://doi.org/10.1109/TCSI.2019.2918241
29. Salmanpour F, Moaiyeri MH, Sabetzadeh F (2021) Ultra-compact imprecise 4:2 compressor and multiplier circuits for approximate computing in deep nanoscale. Circuits, Systems, and Signal Processing 40(9). https://doi.org/10.1007/s00034-021-01688-8
30. Strollo AGM, Napoli E, De Caro D, Petra N, Meo GD (2020) Comparison and extension of approximate 4–2 compressors for low-power approximate multipliers. IEEE Trans Circuits Syst Regul Pap 67(9):3021–3034. https://doi.org/10.1109/TCSI.2020.2988353
31. Taheri M, Arasteh A, Mohammadyan S, Panahi A, Navi K (2020) A novel majority based imprecise 4:2 compressor with respect to the current and future VLSI industry. Microprocessors and Microsystems 73:102962
32. Van Toan N, Lee J-G (2020) FPGA-based multi-level approximate multipliers for high-performance error-resilient applications. IEEE Access 8:25481–25497. https://doi.org/10.1109/ACCESS.2020.2970968
33. Veeramachaneni S, Krishna KM, Avinash L, Puppala SR, Srinivas MB (2007) Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors. In: Proc. 20th International Conference on VLSI Design and 6th International Conference on Embedded Systems (VLSID'07). pp 324–329. https://doi.org/10.1109/VLSID.2007.116
34. Venkatachalam S, Ko S-B (2017) Design of power and area efficient approximate multipliers. IEEE Trans Very Large Scale Integr VLSI Syst 25(5):1782–1786. https://doi.org/10.1109/TVLSI.2016.2643639
35. Yang Z, Han J, Lombardi F (2015) Approximate compressors for error-resilient multiplier design. In: Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2015 183–186. https://doi.org/10.1109/DFT.2015.7315159
36. Zervakis G, Tsoumanis K, Xydis S, Soudris D, Pekmestzi K (2016) Design-efficient approximate multiplication circuits through partial product perforation. IEEE Trans Very Large Scale Integr VLSI Syst 24(10):3105–3117. https://doi.org/10.1109/TVLSI.2016.2535398

Parthibaraj Anguraj received the B. E. (ECE) degree from Narasu's Sarathy Institute of Technology, Salem, Tamil Nadu, India in the year 2015 and his M.E. (VLSI Design) degree in Anna University Regional Campus, Coimbatore, Tamil Nadu, India in the year 2018. After that, he was working as a Junior Research Fellow (JRF) under the SERB funded project in the ECE department of the Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India. Currently, he is working as an Assistant Professor in K.Ramakrishnan College of Technology, Trichy, Tamil Nadu, India. In addition, he is working towards the Ph.D. degree (Part-time) in information and communication engineering under the guidance of Dr. Thiruvenkadam Krishnan in K.Ramakrishnan College of Technology, Trichy, Tamil Nadu, INDIA. His interests include Wireless Communications and Low Power VLSI design and Approximate computing.

Thiruvenkadam Krishnan received B.E. (ECE) degree from Adhiyamaan College of Engineering, Hosur, Tamil Nadu, India in 2007 and his M.E. degree in (VLSI Design) from PSG College of Technology Coimbatore, Tamil Nadu, India in 2009. He completed his Ph.D. in the year 2018 from Anna University, Chennai. Currently he is working as an Associate Professor in K.Ramakrishnan College of Technology, Trichy, Tamil Nadu, India. His general research areas include Low Power VLSI design & testing, Approximate computing and Image processing applications.

Saravanan Subramanian received B. E (ECE) degree from Adhiyamaan College of Engineering, Hosur, Tamil Nadu, India in 2006 and his M.E. degree in VLSI Design from Kongu Engineering College Erode, Tamil Nadu, India in 2010. Currently, he is working as an Assistant Professor in K. S. Rangasamy College of Technology, Tiruchengode, Tamil Nadu, India. His area of interests includes Low Power VLSI design and Approximate computing.