2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID) | 978-1-6654-8505-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/VLSID2022.2022.00059

Design of 8-bit Dadda Multiplier using Gate Level Approximate 4:2 Compressor

KATTEKOLA NARESH
ECE Department
VNR VJIET
Hyderabad, Telangana, India
nareshkattekola@gmail.com

Y. PADMA SAI
ECE Department
VNR VJIET
Hyderabad, Telangana, India
padmasai_y@vnrvjiet.in

SHUBHANKAR MAJUMDAR
ECE Department
NIT Meghalaya
Shillong, Meghalaya, India
shubuit@gmail.com

***Abstract*—Power consumption and reliability have been the main design issues in the digital world in recent years. New low-power architecture models have been investigated using power density and energy as boundaries. In the hardware implementation domain, one such ideal model is approximate computation, where our approach proposes to design an approximate Dadda multiplier by reducing large and costly Compressors to some steering logic and a much smaller exact logic. Here four cases of 8–bit dada multiplier are designed in such a way that; each case has less error rate to get less affected function with less area and power dissipation. Furthermore, the proposed multipliers are implemented in the image blending application. The design of Dadda Multiplier shows the power and delay gain of 60% and 35% respectively which is designed with gate-level approximation technique.**

Keywords— Approximate Computing, Compressor, Dadda Multiplier, Image processing.

I. INTRODUCTION

Approximate computing is one strategy that takes advantage of an application's error resilience to improve the system's overall resource performance. Traditional strategies for achieving substantial efficiency gains, such as Dynamic Voltage and Frequency Scaling (DVFS), power gating, and power/energy-aware application mapping, are insufficient to meet the increasing computing-efficiency demands, providing improvements to a limited degree. Approximate computing allows the designer to take advantage of this error tolerance to introduce a new dimension to design optimization, where numerical precision is exchanged for power consumption and design complexity reductions.

Approximate calculation results are those that are estimated and subject to a given error bound. Approximate arithmetic circuits are a critical subject that has gotten a lot of attention in the VLSI design community. Hardware elements are optimized in this approach to ramp down power consumption and design area, therefore adding inaccuracies. Due to their widespread use in various applications, these approximations are often based on simple arithmetic building blocks (e.g., adders and multipliers).

Every application doesn't need an accurate computation unless it is very sensitive to errors. Using the full custom design, there is a maximum possibility for optimization of the design. Even with the precedence, as the complexity increases, an optimized design with error-tolerant circuits doesn't affect the functionality in some applications. So, a multiplier is being proposed here with approximation techniques that can be used in Image processing applications. Approximate computing is a major advantage for image processing applications where accuracy can be traded off. For image compression and image format conversion applications are needed combinational blocks like adders [1], compressors [2], multipliers [2], [3].

The approximate design work done on multipliers by [4] is being proposed by analyzing four different compression techniques on the dadda multiplier. [5] proposed a low-power, high-accuracy 8-bit multiplier with two major characteristics, different weights use different compressors to gather their product terms, depending on the importance to simplify the logic of carrying chains, they use high-order approximate compressors (e.g., 8:2 compressor) for the middle significance weights. In [6], suggested high-speed, low-power 3- 2, 4-2, and 5-2 compressors that can operate at ultra-low voltages and have outputs that are effectively tailored to boost the compressor's performance. Most of the approximate circuit work is done by using combinational modules [7-15]

Compressors are basic circuits that are made of full adders and half adders to count the number of "ones" in the input. Several compressors are required in the partial product reduction stage which will affect multipliers. Various compressors with different approaches and low power techniques, at gate level and transistor levels such as 3-2, 4-2, 5-2, and 5-3 were proposed by researchers in the last 20 years. These are useful only when the preferred multiplier is small in size. 16 & 32-bit multipliers [15] require a large number and size of compressors which is a major score in controlling the design constraints. As the approximation technique is most suitable for image processing applications, image blending, denoising, edge detection, type of work is done using arithmetic circuits like compressors and multipliers [16-17].

The paper's primary contributions are as follows:

- Design of approximate full adder for the compressor.
- Development of approximate compressor using a gate-level replacement for a better area.

978-1-6654-8505-0/22/$31.00 ©2022 IEEE
DOI 10.1109/VLSID2022.2022.00059

- Analysis of error metrics in approximate compressors.
- Design of image processing application using an approximate compressor-based multiplier.
- Design constraints comparison of approximate multipliers and compressor designs.

Section - II discusses approximate multiplier Design using four approximation possibilities. Section-III describes the Simulation Results. Section -IV Compares the Design Constraints, Section-V concludes the work done.

II. Approximate Dadda Multiplier

A. 8-Bit Dadda Multiplier

It's a certain kind of parallel multiplier. It is shorter (for all operand sizes) and requires fewer gates than the array multiplier (for all but the lowest operand sizes). Dadda multiplier's lowering phase is less expensive. The Dadda multiplier's reduction phase requires only six complete adders and half adders, compared to ten for a Wallace tree. In comparison to the Dadda Multiplier, the Wallace requires more hardware. The compressor is the major component of the multiplier. There is a large delay during the partial product addition stage, which increases the power consumption. Compressor adders minimize the amount of full and half adders, and hence the power consumed, by adding four, five, six, or seven bits at a time. Compressors are components that are utilized in the multiplication process to sum the partial products. The main principle behind an n: 2 compressors is performing addition while keeping the carry and sums separated, n operands can be reduced to two. In this paper, gate level approximate 4:2 compressors are used to implement an 8-bit Dadda multiplier.

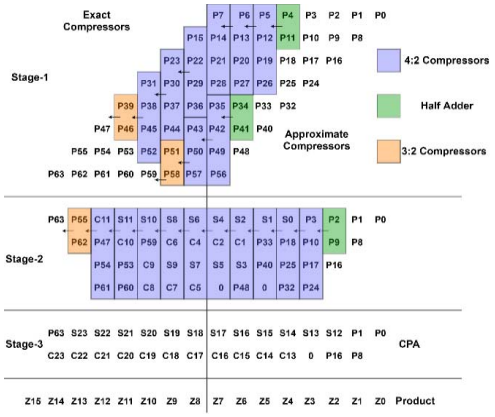


Fig. 1. 8-bit Dadda Multiplier

The proposed compressors are implemented in unsigned 8-bit Dadda tree multipliers. The first step of an accurate multiplier reduction circuit is depicted in the figure. To limit the PP to a maximum of four rows, two 3:2 compressors, two half-adders, and eight 4:2 compressors are used. In the next step, one half-adder, one 3:2 compressor, and ten 4:2 compressors are used to calculate the final two stages of partial products. The final product is obtained by using a carry propagate adder in the previous stage. Fig.1 shows the design of an 8-bit Dadda Multiplier using 3 stages.

B. Conventional Full Adder

The 3:2 accurate compressor is a conventional full adder where the outputs are the sum and carry. A complete adder circuit takes three numbers as input (A, B, and C) and results in the sum and carry of bit values. Digital logic gates are utilized to build the adder, which performs the addition of the provided input function and outputs the sum and carry.

Two serially coupled full adders make up a conventional 4:2 compressor. Instead of using another adder, employ a compressor adder with carrying propagation.

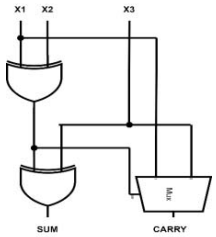


Fig. 2. Conventional full adder

The Fig.2. shows the logic diagram of conventional full adder. The equations of the sum and carry are followed as

$$Sum = x1 \oplus x2 \oplus x3 \quad (1)$$

$$Carry = x1.x2 + x3(x1 \oplus x2) = x3(x1 \oplus x2) + (x1 \oplus x2)'.x1 \quad (2)$$

C. Approximate Full Adder

Each case of approximate full adder from table I is described as a respected approximate full adder case (AFAC).

TABLE I Approximate Full Adder Truth Table

Inputs			Accurate		Case-1		Case-2		Case-3		Case-4	
x 1	x 2	x 3	S	C	S	C	S	C	S	C	S	C
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	1	0	1	0	1	**1**
0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	1	0	1	0	1	**1**	1	0	1	0	1
1	0	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	1	1	**1**	1	**1**	**0**	0	1
1	1	0	0	1	**1**	**0**	0	1	0	1	0	**0**
1	1	1	1	1	**0**	1	1	1	1	1	1	1

The approximate computing depends on the number of errors introduced concerning the area affected in the design. as the errors increase, the functionality is more affected. Hence there should be valid error insertion to reduce the redundant logic which may not affect the overall functionality. With proper error position, the design constraints may improve.

The expressions for all cases are given below.

1) Approximate Full Adder Case-1(AFAC-1):

$$Sum = x2.x3' + x1.x3' + x1'.x3 \quad (3)$$

$$Carry = x2.x3 + x1.x3 = x3(x1+x2) \quad (4)$$

2) Approximate Full Adder Case-2(AFAC-2):

$$Sum = x3 + x1 \oplus x2 \quad (5)$$

$$Carry = x3(x1 \oplus x2) + (x1 \oplus x2)'x1 \quad (6)$$

3) Approximate Full Adder case-3(AFAC-3):

$$Sum = x1(x2' + x3) + x2'x3 + x1'x2x3' \quad (7)$$

$$Carry = x2\,(x3+x1) \quad (8)$$

4) Approximate Full Adder case-4(AFAC-4):

$$Sum = x1 \oplus x2 \oplus x3 \quad (9)$$

$$Carry = x3 \quad (10)$$

5) Approximate Full Adder case-5(AFAC-5):

$$Sum = x1 + x2 + x3 \quad (11)$$

$$Carry = x2.x3 + x1.x3 = x3(x1+x2) \quad (12)$$

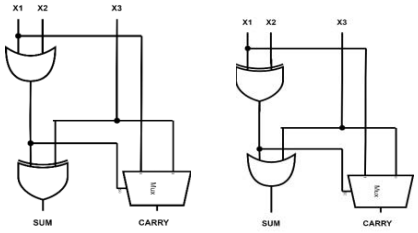


Fig. 3. Design of AFAC-1 and AFAC-2

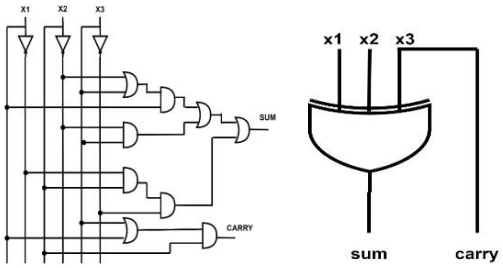


Fig. 4. Design of AFAC-3 and AFAC-4

The XOR logic consumes more power and area because of its logic. All the arithmetic circuits are based on XOR/XNOR functions. Here in gate level approximation, XOR is replaced with OR for better efficiency. The case is described as AFAC-5.

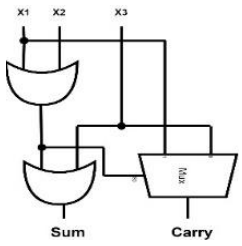


Fig. 5. Design of AFAC using OR logic

D. Conventional 4:2 Compressor

A compressor is a device that is commonly seen in multipliers and is used to decrease the operands while adding partial product terms. M equally weighted input bits are fed into a conventional M-N compressor, which outputs an N-bit binary integer. As seen in Fig. 6., a 4-2 compressor is traditionally implemented with two serially connected full adders.

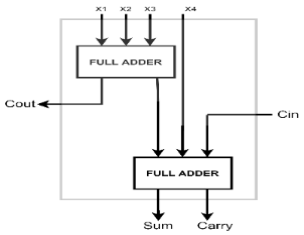


Fig. 6. Conventional of 4:2 Compressor

The sum, Carry and Cout equations are as shown below.

$$Sum = X1 \oplus X2 \oplus X3 \oplus X4 \oplus Cin. \quad (13)$$

$$Cout = (X1 \oplus X2)\,X3 + (X1 \oplus X2)\,X1. \quad (14)$$

$$Carry = (X1 \oplus X2 \oplus X3 \oplus X4)\,Cin + (X1 \oplus X2 \oplus X3 \oplus X4)\,X4 \quad (15)$$

By Approximating the frequently used compressor module, there is large scope for constraints minimization.

E. Approximate 4:2 Compressor

The Approximate full adder is replaced at the MSB and LSB positions with an accurate adder in AFAC-1M, AFAC-1L modules in Fig. 7 a) and b), These designs are named as Approximate Compressor Modules (ACM)-1M and 1L respectively. Similarly, ACM-2M and 2L are designed.

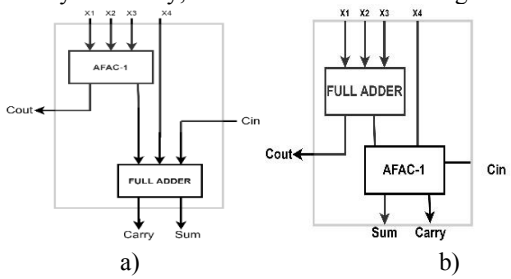


Fig. 7. Design of a) ACM1M b) ACM-1L

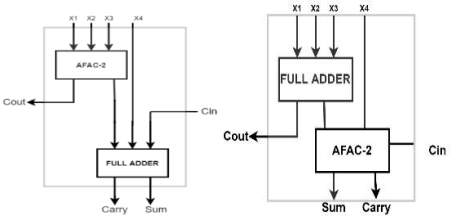


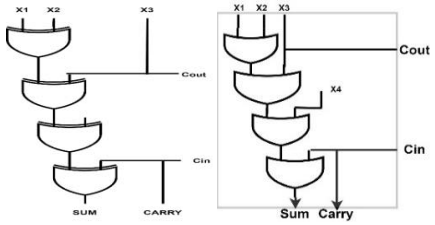
Fig. 8. Design of a) ACM-2M b) ACM-2L

From the truth table of accurate 4:2 compressor around 75% of carry reflects the exact value of carry input (Cin) and Cout matches the exact value of X3. Therefore

$$X3 = Cout \quad (16)$$

$$Cin = Carry. \quad (17)$$

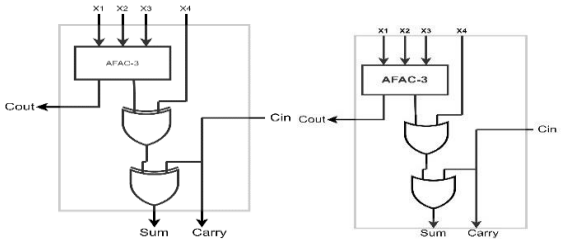
These expressions are resultant of the AFAC-4 approximate 3:2 compressor model. This case is designed with the name of Approximate Compressor Module-3(ACM-3). ACM-3OR is designed by replacing XOR with OR logic.



a) b)

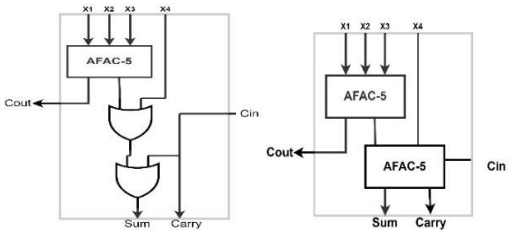
Fig. 9. Design of a) ACM-3 b) ACM-3OR

ACM-4 is based on the AFAC-3 compressor. ACM-4OR is designed by replacing XOR with OR logic. ACM-45 and 5 are designed based on AFAC-4,5 combinations and 5 respectively.



a) b)

Fig. 10. Design of a) ACM-4 b) ACM-4OR



a) b)

Fig. 11. Design of a) ACM-45 b) ACM-5

Since the AFAC-1 and 2 have better area efficiency, 4:2 compressor is implemented with all these possibilities which are named as ACM-11,12,21, and 22 shown in Fig. 12.

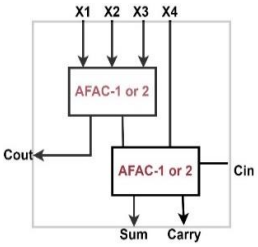
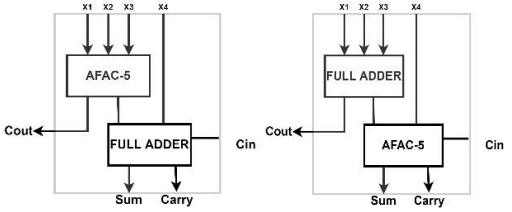


Fig. 12. Design for combinations of ACM-11,12,21,22

A specific OR logic has more error possibility in multiplication process, from ACM-5 compressor, OR logic is chosen at MSL, LSB respectively with ACM-5M and 5L.



a) b)

Fig. 13. Design of a) ACM-5M b) ACM-5L

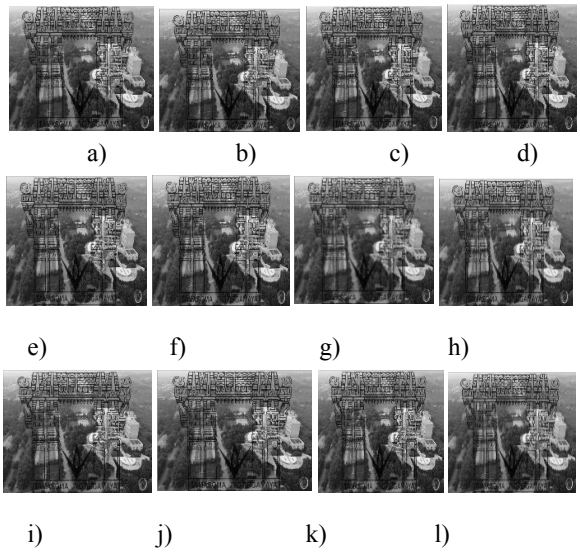
III. SIMULATION RESULTS

The approximated compressor-based multipliers are used in image processing to perform pixel-by-pixel multiplication, resulting in merging two images into one. The two images with similar properties are multiplied to produce an effectively blended output image.

The below simulation results show the MATLAB results on image blending using an 8-bit Dadda multiplier using all the possible 16 approximation cases. Fig.15 shows the image blending applications of Dadda multipliers with proposed cases.

a) b) c)

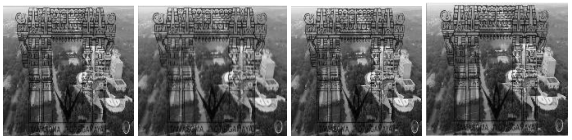
Fig. 14. Input images for Multiplication a) Campus b) VNRVJIET logo c) accurate multiplier output



a) b) c) d)

e) f) g) h)

i) j) k) l)



m) n) o) p)

Fig. 15. Blending image of a) ACM-1M b) ACM-1L c) ACM-2M d) ACM-2L e) ACM-3 f) ACM-3OR g) ACM-4 h) ACM-4OR i) ACM-45 j) ACM-5 k) ACM-11 l) ACM-12 m) ACM-21 n) ACM-22 o) ACM-5M p) ACM-5L

IV. Design Constraints Comparison

Most of the compressors which were designed earlier were introduced with a greater number of error rates, whereas the proposed models usually represent fewer error rates with better efficiency. The simulation results were done by using Synopsys DC compiler under 90nM Technology.

TABLE II Multiplier Design Constraints

Dadda Multiplier	POWER (mW)	AREA (μm^2)	DELAY (uS)	PSNR	PDP (fJ)
Accurate	857.58	2588.18	2.23	-	1912.4
ACM-1M	802.49	2438.87	2.14	43.80	1717.3
ACM-1L	777.74	2438.87	2.04	43.69	1586.5
ACM-2M	787.62	2438.87	2.11	43.03	1661.8
ACM-2L	816.31	2438.87	2.05	45.1	1673.4
ACM-3	763.22	1957.78	2.10	41.75	160.2.7
ACM-3OR	386.18	1476.99	1.91	42.18	737.6
ACM-4	652.03	2007.57	1.95	41.91	1271.4
ACM-4OR	468.53	1775.59	1.91	41.56	894.8
ACM-45	527.4	1792.17	1.91	42.44	1007.3
ACM-5	631.08	2090.52	1.91	44.05	1205.3
ACM-11	716.84	2322.75	1.96	42.37	1405
ACM-12	745.07	2322.75	1.98	43.47	1475.2
ACM-21	721.74	2322.75	2.03	44.39	1465.1
ACM-22	754.79	2322.75	2.05	42.59	1547.3
ACM-5M	716.39	2322.75	2.04	43.71	1461.4
ACM-5L	759.51	2322.75	2.00	43.45	1519

Table II discusses the design constraints of approximate multiplier models. The results are shown with the power delay product. Out of all gate level replacement approximate multiplier models, the ACM-3,3OR 4,4OR, and 45 models have shown better efficiency due to their OR logic and less area. XOR is a power-hungry logic that can be replaced for better efficiency but the error rate should be acceptable. As the primitives are predefined and easy to use for any module, the gate level approximation has a better scope from a design perspective. Because of the gate level, the dadda multiplier is chosen, which occupies fewer gates than the Wallace tree multiplier.

TABLE III 3:2 Compressor Design Constraints

3:2 Compressor	Gate Count	Error Rate	POWER (μW)	AREA (μm^2)	DELAY (nS)	PDP (fJ)
Accurate	14	0	10.36	38.70	0.24	2.48
AFAC-1	10	0.18	8.12	32.25	0.22	1.78
AFAC-2	10	0.12	8.27	32.25	0.22	1.81
AFAC-3	12	0.12	3.54	24.88	0.10	0.35
AFAC-4	10	0.12	6.31	22.11	0.21	1.32
AFAC-5	6	0.25	6.27	25.80	0.16	1.00

Table III represents the gate count with an error rate and design constraints of 3:2 approximate Compressors. From the table, AFAC-3 results in giving better efficiency by eliminating the XOR logic. Even the model AFAC-5 gives better efficiency, but it has more error rate because of OR logic.

.TABLE IV 4:2 Compressor Design Constraints

4:2 Compressor	Gate Count	Error Rate	POWER (μW)	AREA (μm^2)	DELAY (nS)	PDP (fJ)
Accurate	28	0	21.03	77.41	0.48	10.09
ACM-1M	24	0.16	19.19	70.96	0.43	8.25
ACM-1L	24	0.12	18.81	70.96	0.41	7.71
ACM-2M	24	0.12	20.23	70.96	0.48	9.71
ACM-2L	24	0.08	19.94	70.96	0.45	8.97
ACM-3	20	0.16	14.72	44.23	0.30	4.41
ACM-3OR	4	0.32	3.90	17.52	0.19	0.74
ACM-4	22	0.16	6.54	22.11	0.32	2.09
ACM-4OR	14	0.26	5.69	34.11	0.18	1.02
ACM-45	8	0.28	8.34	35.03	0.25	2.08
ACM-5	12	0.27	13.06	51.61	0.33	4.30
ACM-11	20	0.22	16.86	64.51	0.35	5.90
ACM-12	20	0.20	16.98	64.51	0.40	6.79
ACM-21	20	0.18	16.77	64.51	0.40	6.70
ACM-22	20	0.16	17.98	64.51	0.45	8.09
ACM-5M	20	0.22	17.36	64.51	0.42	7.29
ACM-5L	20	0.16	16.94	64.51	0.38	6.43

Table IV represents the gate count with an error rate and design constraints of 4:2 approximate compressors. From the table, ACM-3OR and ACM4-OR result in giving better efficiency by eliminating the XOR logic. But these models have more error rates because of OR logic. The model ACM-4 has better efficiency and a considerable error rate.

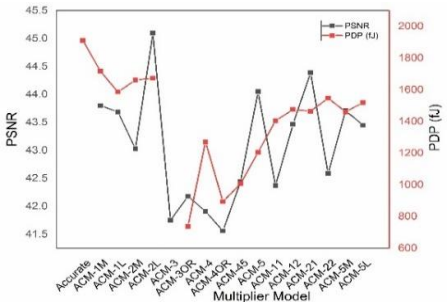


Fig. 16. Power and Delay comparisons for dadda multiplier models

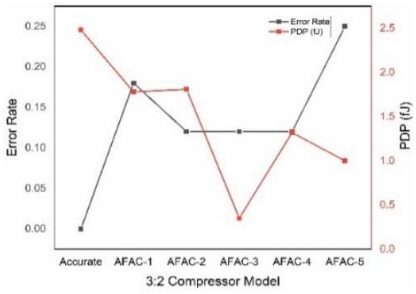


Fig. 17. Error rate and PDP comparisons for 3:2 compressor models

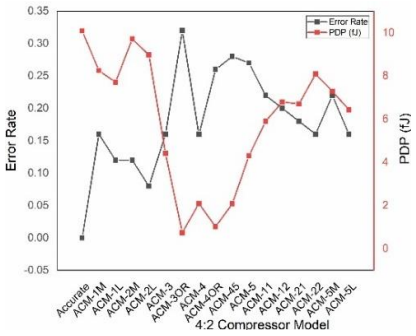


Fig. 18. Error rate and PDP comparisons for 4:2 compressor models

Fig. 16,17 & 18 shows the error rate, PDP from the designed modules of compressors and multiplier designs.

V. CONCLUSION

Dadda Multiplier is implemented using approximate compressors. For that, 4:2 Compressor is implemented in various cases using different approaches of gate-level approximation. The exactness or accuracy rates for better cases are 92, 87.5, 85.5, 78. The proposed approximate compressors with a Dadda multiplier implementation have been discussed along with image processing applications. The respected Peak Signal to Noise Ratio (PSNR) is shown with a particular model for each multiplier.

As per the simulation results, the approximate multiplier of ACM-3,4,5 models gives better results with an acceptable error rate. These models have the reduction of design constraints with a better PDP of around 55% to 65%. Hence, the top module of Dadda ACM-3,4,5 has a significant reduction in power and delay gain of 55% and 15% respectively. The output images have a high Peak-to-Signal Noise Ratio (PSNR) value when these designed multipliers are utilized in image processing for pixel-by-pixel multiplication, resulting in a high-quality output image. These techniques can be extended to a further size of multiplication and can be implemented in video mattifying and image-based rendering.

REFERENCES

[1] Manickam Ramasamy, G. Narmadha, S. Deivasigamani, “Carry based approximate full adder for low power approximate computing,” in 2019 7th International Conference on Smart Computing & Communications (ICSCC), 2019.

[2] Minho Ha, Sunggu Lee., “Multipliers with Approximate 4–2 Compressors and Error Recovery Modules,” in IEEE Embedded Systems Letters, 2018.

[3] Antonio Giuseppe Maria Strollo, Ettore Napoli, Davide De Caro, “Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers,” in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–I: REGULAR PAPERS, 2020.

[4] Ms.N. Savitha, Ms.A. Poornima, “A High-speed Area Efficient Compression technique of Dadda multiplier for Image BlendingApplication”, Third International Conference on I-SMAC, 2019.

[5] Che-Wei Tung, Shih-Hsu Huang, “Low-Power High-Accuracy Approximate Multiplier Using Approximate High Order Compressors”, 2nd International Conference on Communication Engineering and Technology, 2019.

[6] Sreehari Veeramachaneni, Kirthi Krishna M, Lingamneni Avinash, Sreekanth Reddy Puppala, M.B. Srinivas, “Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors”, 20th International Conference on VLSI Design (VLSID’07), 2007.

[7] Manickam Ramasamy, G.Narmadha, S. Deivasigamani “Carry based approximate full adder for low power approximate computing,” in 2019 7th International Conference on Smart Computing & Communications (ICSCC), 2019.

[8] Minho Ha, Sunggu Lee., “Multipliers with Approximate 4–2 Compressors and Error Recovery Modules,” in IEEE Embedded Systems Letters, 2018.

[9] Antonio Giuseppe Maria Strollo, Ettore Napoli, Davide De Caro, “Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers,” in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–I: REGULAR PAPERS, 2020.

[10] K. Manikantta Reddy, M.H. Vasantha, Y.B Nithin Kumar, Devesh Dwivedi, “Design of Approximate Dividers for Error Tolerant Applications,” in 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), 2018.

[11] Mohammad Heidary Takaby, Sayed Masoud Sayedi., “Low Power Approximate Restoring Divider Design Using Gate Diffusion Input Logic,” in 27th Iranian Conference on Electrical Engineering (ICEE2019), 2019.

[12] Weiqiang Liu, Jing Li, Tao Xu, Chenghua Wang, Paolo Montuschi, Fabrizio Lombardi, “Combining Restoring Array and Logarithmic Dividers into an Approximate Hybrid Design,” in 2018 IEEE 25th Symposium on Computer Arithmetic (ARITH), 2018.

[13] Linbin Chen, Jie Han, Weiqiang Liu, Paolo Montuschi, Fabrizio Lombardi., “Design, Evaluation, and Application of Approximate High- Radix Dividers,” in IEEE Transactions on Multi-Scale Computing Systems, 2018.

[14] Suganthi Venkatachalam, Elizabeth Adams, Seok-Bum Ko., “Design of Approximate Restoring Dividers,” in 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019.

[15] R. Marimuthu, Y. E. Rezinold, and P. S. Mallick, "Design and Analysis of Multiplier Using Approximate 15-4 Compressor," in IEEE Access, vol. 5, pp. 1027-1036, 2017.

[16] J. A. Arjun and S. Majumdar, "Development of Approximate Compressor Based Hybrid Dadda Multiplier for Image De-noising Applications," 2019 IEEE 16th India Council International Conference (INDICON), Rajkot, India, pp. 1-4, 2019.

[17] Kattekola, N., Jawale, A., Nath, P.K. and Majumdar, S., "Efficient partial product reduction for image processing application using approximate 4:2 compressor", Circuit World, DOI:10.1108/CW-09-2020-0220, 2021.