International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 14, No. 2, July 2025, pp. 388~397
ISSN: 2089-4864, DOI: 10.11591/ijres.v14.i2.pp388-397

Exploring the landscape of approximate subtraction methods in ASIC platform

M. Priyadharshni[1], Rajermani Thinakaran[2], Grande Naga Jyothi[3], Vijayakumar Varadarajan[4,5,6], C. Srinivasa Murthy[7]

[1]Department of Electrical and Computer Engineering, Panimalar Engineering College, Chennai, India
[2]Faculty of Data Science and Information Technology, INTI International University, Nilai, Malaysia
[3]Department of Electrical and Computer Engineering, Madanapalle Institute of Technology and Science, Madanapalle, India
[4]Global Students Nest, Pune, India
[5]Ajeenkya DY Patil University (ADYPU), School of Engineering, Pune, India
[6]Swiss School of Business and Management (SSBM), Geneva, Switzerland
[7]Department of Electrical and Computer Engineering, Chinthalapudi Engineering College, Ponnur, India

Article Info

Article history:

Received Oct 9, 2023
Revised Apr 10, 2025
Accepted Jun 10, 2025

Keywords:

Approximate subtractor
Approximation imprecision
ASIC platform
Design metrics
Error-tolerant applications
Inexact computing

ABSTRACT

Approximate computing has emerged as a crucial technique in modern computing, offering significant benefits for error-resilient applications. Error resilient applications include signal, image, audio processing, and multimedia. These applications will accept the errored results with some degree of tolerance. This approach allows these applications to process and embrace data that may deviate slightly from perfect accuracy. The utility of approximate computing extends to both hardware and software domains. In hardware, arithmetic units are particularly important, among that approximate subtractors have gained attention for their role in these units. A comparative study was conducted on various approximate subtractors from existing literature, considering structural analysis in all scenarios. These approximate subtractors are coded in Verilog hardware description language (HDL) and synthesized in Synopsys electronic design automation (EDA) Tool using Taiwan Semiconductor Manufacturing Company (TSMC) 65 nm technology. Out of the available choices, approximate subtractor 3 is particularly well-suited for processing higher bit data due to its reduced hardware complexity and minimal error. Notably, it outperforms exact subtractors by achieving a notable reduction of 20% in the area delay product (ADP) and 15% in the power delay product (PDP) as process innovation. These improvements highlight the efficiency and effectiveness of approximate subtractor 3, making it a compelling option for various computing applications which accept the inaccurate results.

Corresponding Author:

Grande Naga Jyothi
Department of Electronics and communications, Madanapalle Institute of Technology and Science
Angallu, Andhra pradesh, India
Email: nagajyothisai221@gmail.com

1. INTRODUCTION

Most of the computer applications that follow arithmetic sequence can be done with more precision by using electronic circuits that do use data and sorts of information that are encoded in a binary form [1]. Although the precision can be made by using digital logic circuits, there are many more applications [2] that doesn't require such a high level of accuracy, that includes image processing, and also multimedia platforms such as video, audio, and computer games. The above-mentioned application has a high tolerance of error by

producing more reliable results provided that the severities are within the specified thresholds. When the production is done in a large scale to conserve the budget from being exploited, the flexibility of the circuits in maintaining the level of accuracy and Specificity can be compromised [3].

In the realm of computer science, there exists a paradigm known as inexact computing. Unlike the traditional approach of exact computing, which prioritizes utmost accuracy in its results, inexact computing [4] introduces a fascinating concept: it allows for a certain level of errors and imprecision in calculations while employing approximation techniques. This paradigm finds particular relevance and benefits in the domain of basic arithmetic circuits, which encompass fundamental operations like addition, subtraction, and multiplication.

The adoption of inexact computing brings forth numerous advantages. By accepting a tolerable margin of errors, it enables more efficient utilization of computational resources and can potentially enhance the overall performance of arithmetic circuits [5]–[11]. These circuits, being integral to various computational tasks, can leverage the inherent imprecision to optimize their operations and improve efficiency. In other words, inexact computing challenges the traditional belief that absolute accuracy is paramount, highlighting instead the potential gains in efficiency and resource utilization that can be achieved by embracing approximation and imprecision.

Several studies focus on arithmetic circuits, exploring the advantages and hurdles of integrating full adder circuits using fin-shaped field-effect transistor (FinFET) technology, thereby providing valuable insights to enhance efficiency and performance in arithmetic circuits [12]. Additionally, varying designs of full adders are discussed in [13]–[17].

Through the incorporation of approximation techniques [6], inexact computing offers a fresh perspective on the relationship between precision and efficiency in computations. By acknowledging the inherent trade-off between accuracy [18], [19] and resource utilization, this paradigm paves the way for innovative approaches to computing, revolutionizing the field and opening up new avenues for exploration and advancement [20]–[22]. The design methodologies, evaluation metrics and trade off parameters related to approximate arithmetic circuits are discussed in [23]. By using these arithmetic circuits Multiply and accumulate unit can be designed and remain suitable for many applications [24].

Recent works on inexact computing, a special type of subtractor steals the limelight, it is termed as the approximate subtractor [9]. The ideology of approximate subtractors (APSC) is the exact subtractors with tolerable errors in results and the final output will not have noticeable change. The above-mentioned designs are developed as a result of introducing the concept of approximation in logic level and k-map is used to reduce its complexity. To establish a connection that maintains balance between energy efficiency and accuracy, the three APSC [10] are designed for its hardware efficiency and power delay product (PDP).

The primary drive behind this endeavor is to pinpoint the most suitable subtractor, crucial in crafting high-speed division logic. Given the pivotal role of arithmetic units in digital architecture design, the focus lies on selecting an optimal subtractor to significantly enhance the speed and efficiency of division operations. This pursuit aligns with the imperative need for streamlined arithmetic processes within digital systems, ensuring their overall performance and functionality.

The organization of this paper has been thoughtfully structured to provide a cohesive and comprehensive exploration of the topic. It begins with section 2 with a clear understanding of the subtractor's functionality, underlying principles, and relevant contextual information. Section 3 shifts the focus to result analysis, the analysis encompasses a thorough evaluation of the subtractor's performance, identification of any potential limitations or challenges. Concluding the paper, section 4 encapsulates the key takeaways from the study and offers a comprehensive summary.

2. EXISTING APPROXIMATE SUBTRACTORS

In this specific section, we explore the currently available approximate subtractors, which are approximated at both the transistor and gate levels. Within the existing literature, there are two primary approaches. The first approach involves approximations in the "difference" (D) part while maintaining exactness in the "borrow out" (Bout) part. The second approach is the opposite, where the approximations are made in Bout while ensuring exactness in D.

2.1. Approximate subtractor 1 (ASUB_1)

The ASUB_1 [9] is executed by preserving the precise Bout and approximated D, with Bout being of higher order compared to D. The accompanying logical diagram is displayed below. The gate level implementation of approximate subtractor1 is shown in Figure 1. ASUB_1 is designed using 1 XOR, 2 AND, 2 OR, 2 NOT gates. (1 XOR=5 basic gates). Totally, ASUB_1 utilizes 11 basic gates.

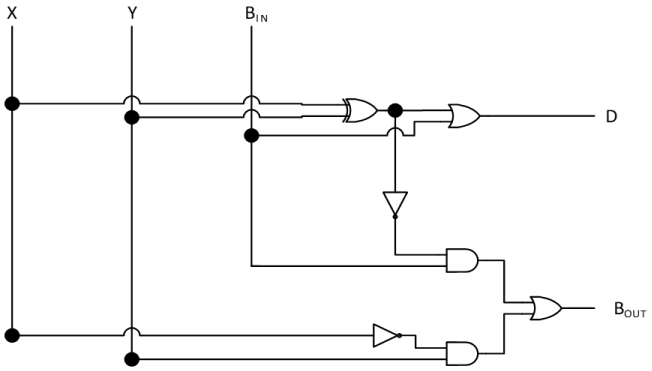

Figure 1. Approximate subtractor 1 (ASUB_1)

2.2. Approximate subtractor 2 (ASUB_2)

The implementation of ASUB_2 [9] involves preserving the exact value of D while setting Bout to be equal to D. However, this approximation leads to two errors in Bout when compared to the exact value. The logical diagram of approximate subtractor 2 is shown in Figure 2. The total number of logic gates used in approximate subtractor 2 is 2 XOR gates. 10 basic gates are utilized to develop ASUB_2.

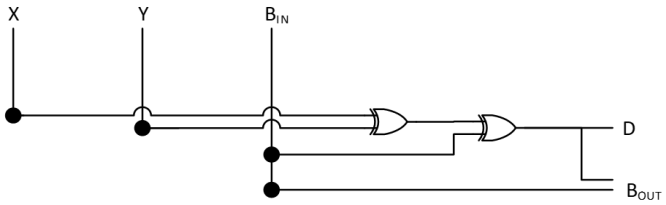

Figure 2. Approximate subtractor 2 (ASUB_2)

2.3. Approximate subtractor 3 (ASUB_3)

In ASUB_3 [9], the implementation maintains Bout as an exact value, equal to D. However, this approximation results in the introduction of two errors in the D output. The logical diagram corresponding to this configuration is provided below. The Figure 3 shows the implementation of ASUB_3. The total number of logic gates used in approximate subtractor 3 is 1 XOR gates, 2 AND, 1 OR, 2 NOT gates. ASUB_3 is implemented using 10 basic gates.

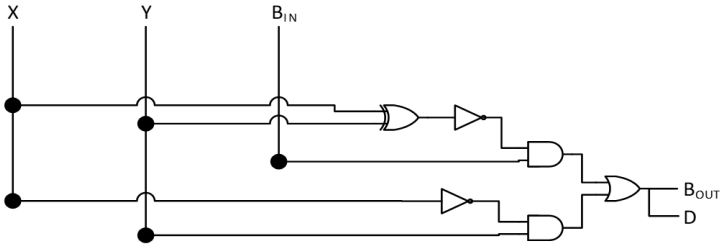

Figure 3. Approximate subtractor 3 (ASUB_3)

Additionally, the discussion includes four other APSC designs, which are known to exhibit lower errors compared to the previously mentioned models. These designs, namely APSC4, APSC5, APSC6, and APSC7, are developed using k maps for simplification. It is important to note that during the development of these proposed APSCs, only the approximation of d is performed. In Table 1, the inputs and the outcomes of ASUB_1, ASUB_2, and ASUB_3 are compared with exact subtractor (EXSC) and observed.

Table 1. Observation of outcomes of ASUB_1, ASUB_2, and ASUB_3

Inputs			EXSC		ASUB_1 [9]		ASUB_2 [9]		ASUB_3 [9]	
X	Y	Bin	Bout	D	Bout	D	Bout	D	Bout	D
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1
0	1	1	1	0	1	1×	0×	0	1	1×
1	0	0	0	1	0	1	1×	1	0	0×
1	0	1	0	0	0	1×	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1

2.4. Approximate subtractor 4 (ASUB_4)

In the case of APSC4 [10], the difference is approximated, while maintaining Bout as an exact value. This approximation leads to incorrect results in one out of the eight cases, while correct by computing the remaining cases. Specifically, the exceptional case arises when X=0, Y=1, and Bin=1. Figure 4 shown is logical diagram ASUB_4. 3 NOT gate, 5 AND gate, 4 OR gate, 1 XNOR gate are utilized for the ASUB_4 designing. Totally, 18 basic gats are utilized in designing ASUB_4 design.

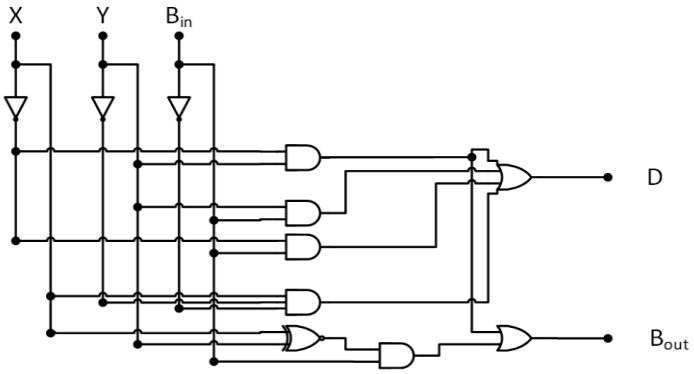

Figure 4. Approximate subtractor 4 (ASUB_4)

2.5. Approximate subtractor 5 (ASUB_5)

In APSC5 [10], the difference is approximated while maintaining Bout as an exact value. The results are recorded for eight cases, out of which seven cases yield correct outputs. The only exception occurs when X=1, Y=0, and Bin=1, where the result is incorrect. However, for all the remaining cases, APSC5 produces accurate results. The Figure 5 shows the implementation of ASUB_5. 3 NOT gate, 6 AND gate, 4 OR gate, 1 XNOR gate are utilized for the ASUB_5 designing. Totally, 19 basic gats are utilized in designing ASUB_4 design.

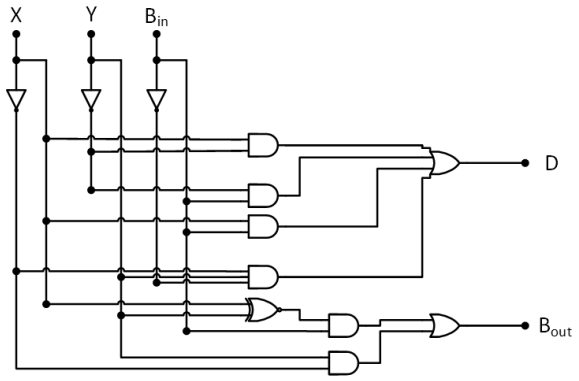

Figure 5. Approximate subtractor 5 (ASUB_5)

2.6. Approximate subtractor 6 (ASUB_6)

For the APSC6 [10] model discussed, the difference is correct for seven out of eight cases. The exceptional case occurs when X=1, Y=0, and Bin=0. However, it is worth noting that the value of Bout remains correct for all eight cases. and the Figure 6 shows the gate level implementation of the ASUB_6 using basic gates. ASUB_6 is designed using 3 NOT gate, 5 AND gate, 3 OR gate, 1 XNOR gate. 17 basic gates are required to implement ASUB_6.

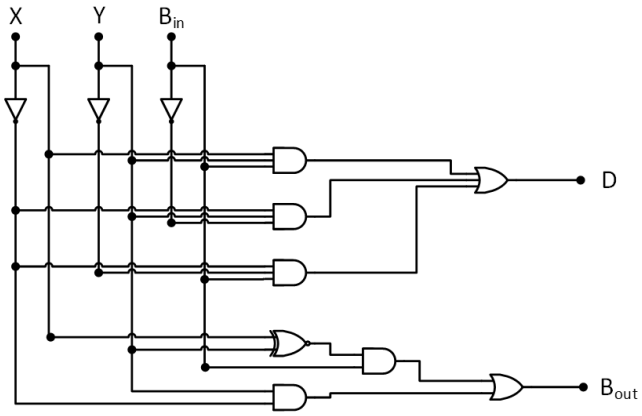

Figure 6. Approximate subtractor 6 (ASUB_6)

2.7. Approximate subtractor 7 (ASUB_7)

It has been observed that the proposed design, APSC7 [10], exhibits correct difference values in seven out of eight cases. The exceptional case arises when X=0, Y=1, and Bin=0. However, it is important to note that the value of Bout remains correct for all eight cases. These results have been accurately recorded in the truth table. ASUB_7 is designed using 3 NOT gate, 8 AND gate, 3 OR gate, 1 XNOR gate. 20 basic gates are required to implement the ASUB_7.

When comparing ASUB_1 to ASUB_3 with ASUB_4 to ASUB_7, it becomes evident that the former group (ASUB_1 to ASUB_3) has a higher probability of errors. Unlike ASUB_2, which introduces the concept of approximation only in D, ASUB_1 and ASUB_3 follow a similar approach. Figure 7 shows the diagrammatic representation of ASUB_7.

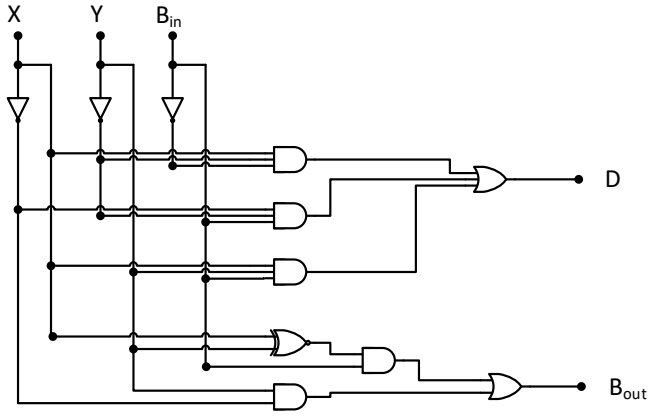

Figure 7. Approximate subtractor 7 (ASUB_7)

Table 2 shows the comparison of inputs and the outcomes of ASUB_4, ASUB_5, ASUB_6, and ASUB_7 with EXSC. Each subtractor produces error in the difference and maintains the Bout as the same. Among 7 input combinations, one pair of input got the errored result, for the remining combinations the result maintained to be exact.

Table 2. Observation of outcomes of ASUB_4, ASUB_5, ASUB_6, and ASUB_7

Inputs			EXSC		ASUB_4 [10]		ASUB_5 [10]		ASUB_6 [10]		ASUB_7 [10]	
X	Y	Bin	Bout	D	Bout4	D4	Bout5	D5	Bout6	D6	Bout7	D7
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	1	0×
0	1	1	1	0	1	1×	1	0	1	0	1	0
1	0	0	0	1	0	1	0	1	0	0×	0	1
1	0	1	0	0	0	0	0	1×	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1

3. RESULTS EXAMINATION

The gate-level implementation of the approximate subtractors involved individual Verilog code composition for each design. Thorough functional verification was executed using the Synopsis Verilog compiler and simulator. Synthesis reports were produced via design compiler, employing Taiwan Semiconductor Manufacturing Company (TSMC) 65 nm technology [11]. The comprehensive analysis covered all process corners specified within the library files. The tabulated outcomes are detailed below.

Table 3 displays the typical corner performance of the approximated subtractors. It showcases observed values for various features such as power, area, and delay, providing insights into the characteristics and efficiency of each subtractor model. The table allows for comparison between the best and worst-case scenarios for subtractor designs, considering factors such as power consumption, delay, and area.

Table 3. Tabulation of typical corner

Designs	Area	Delay	Power
EXACT	47.2	0.12	0.1198
ASUB_1 [9]	32.8	0.12	0.0852
ASUB_2 [9]	32.8	0.12	0.0852
ASUB_3 [9]	10.8	0.11	0.0201
ASUB_4 [10]	25.2	0.12	0.0596
ASUB_5 [10]	32.4	0.12	0.0711
ASUB_6 [10]	34	0.12	0.0768
ASUB_7 [10]	23.2	0.11	0.0336

The power consumption comparison between the approximate subtractor models and the EXACT subtractor reveals interesting findings. Firstly, both ASUB_1 and ASUB_2 models consume approximately 28.88% less power compared to the EXACT subtractor. This power reduction in ASUB_2 is equivalent to the power consumption observed in ASUB_1. Moving on, ASUB_3 demonstrates a substantial decrease in power consumption, with approximately 83.10% less power consumed than the EXACT subtractor.

ASUB_4 also exhibits notable power savings, consuming around 50.144% less power than the exact subtractor models. Additionally, ASUB_5 shows a significant reduction in power consumption, being approximately 40.66% less than the EXACT subtractor. Furthermore, ASUB_6 and ASUB_7 models showcase commendable power efficiency, consuming approximately 36.0024% and 71.95% less power, respectively, when compared to the exact subtractor model. These findings highlight the power-saving potential of the approximate subtractor designs. The graph below illustrates the comparison of the area delay product (ADP), PDP, and power area product (PAP) of approximate subtractors at best corner. Figure 8 is the plotting of the typical corner.

In addition to power consumption, the delay in the operation of the subtractors is also a crucial factor. When considering the operation delay, the values produced by the existing and proposed models are compared to the existing models. It is observed that models ASUB_1, ASUB_2, ASUB_4, ASUB_5, and ASUB_6 exhibit equal delay values. However, ASUB_3 and ASUB_7 demonstrate approximately 0.8333% less delay compared to the existing models.

Furthermore, when evaluating the area as a design metric, the values of the existing and proposed subtractor models are compared to the exact subtractor model. The findings indicate that ASUB_1 and ASUB_2 have an area approximately 30.508% less than the EXSC model. On the other hand, APSC3 shows a significant reduction in area by 77.12%. Similarly, ASUB_4, ASUB_5, ASUB_6, and ASUB_7 have area values lower than the exact subtractor model by 46.61%, 31.3%, 27.9661%, and 50.847%, respectively. Table 4 presents the analysis of the best corners for the approximate subtractors.

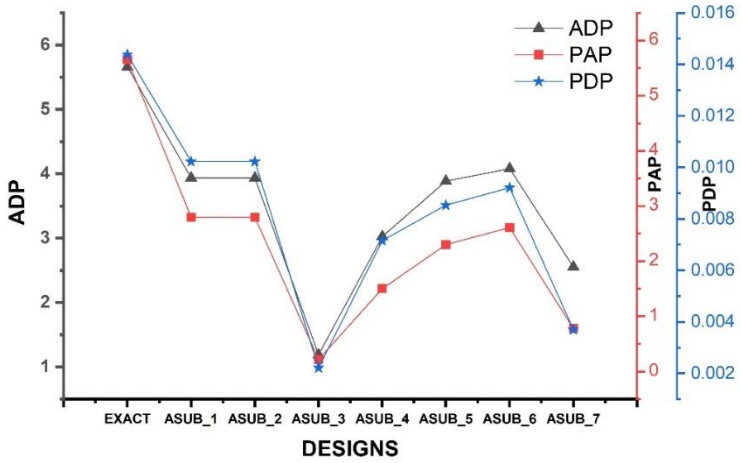

Figure 8. Graph of typical corner

Table 4. Tabulation of best-case scenarios

Designs	Area	Delay	Power
EXACT	44	0.09	0.0929
ASUB_1 [9]	44	0.09	0.0929
ASUB_2 [9]	42.2	0.09	0.1072
ASUB_3 [9]	10.8	0.07	0.0661
ASUB_4 [10]	24.8	0.09	0.1803
ASUB_5 [10]	32.8	0.09	0.0554
ASUB_6 [10]	34	0.09	0.0808
ASUB_7 [10]	22.8	0.09	0.1439

From the analysis, the power consumed by ASUB_2 is approximately 15.336% greater than the exact subtractor, while ASUB_1 consumes the same power as the exact model. ASUB_3 demonstrates a power reduction of approximately 28.96%, while ASUB_4 consumes significantly more power (93%) compared to the exact subtractor. ASUB_5 and ASUB_6 exhibit power reductions of approximately 40.4% and 13.1062 respectively. On the other hand, ASUB_7 consumes approximately 54.75% more power.

Considering the area factor, ASUB_2 has an area approximately 4.09% less than the exact subtractor, while ASUB_1 has the same area. ASUB_3 shows a substantial area reduction of approximately 75.45%. ASUB_4, ASUB_5, ASUB_6, and ASUB_7 demonstrate area reductions of approximately 43.6%, 25.45%, 22.73%, and 48.18%, respectively. Figure 9 shows the plotting of the best corners.

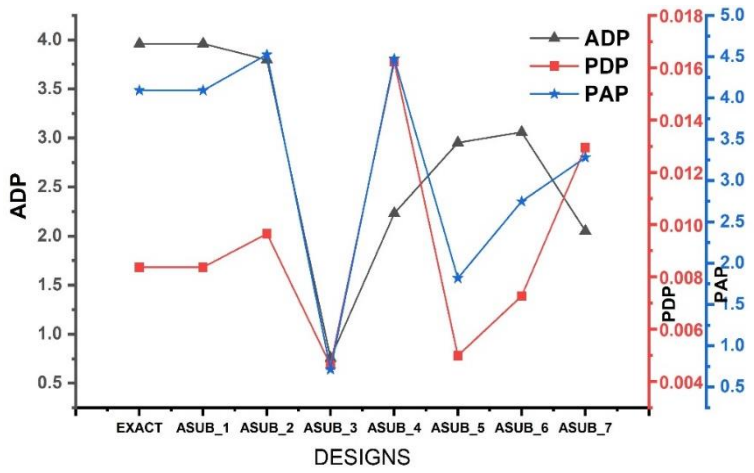

Figure 9. Graph of best-case scenarios

In terms of operation delay, all the models, except ASUB_3, exhibit the same delay as the exact subtractor. ASUB_3 demonstrates a delay reduction of approximately 22.22%. Figure 10 is the graph illustrating the comparison of the ADP, PDP, and PAP for the approximate subtractors at the worst corner. Table 5 provided showcases the analysis of the worst-case scenarios for the approximate subtractors. These results are obtained from the Synopsis design compiler and these results comparts the area, power, and delay values of the approximate subtractors.

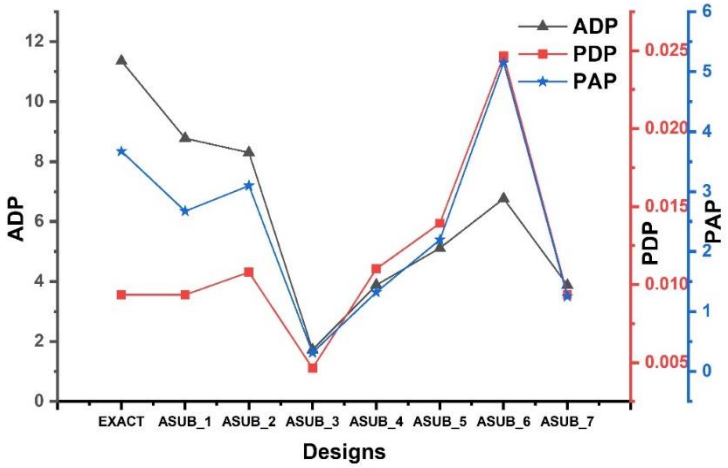

Figure 10. graph of worst-case scenarios

Table 5. Tabulation of worst-case scenarios

Designs	Area	Delay	Power
EXACT	66.8	0.17	0.0549
ASUB_1 [9]	51.6	0.17	0.0549
ASUB_2 [9]	48.8	0.17	0.0634
ASUB_3 [9]	10.8	0.16	0.029
ASUB_4 [10]	21.6	0.18	0.0612
ASUB_5 [10]	28.4	0.18	0.0774
ASUB_6 [10]	37.6	0.18	0.1369
ASUB_7 [10]	22.8	0.17	0.0549

When comparing the area of the approximate subtractors to the exact subtractor, the former demonstrates reductions of 22.79%, 26.95%, 83.83%, 67.7%, 57.5%, 43.6%, and 65.87%. Among that ASUB_3 achieves less reduction in area compared to all other designs. In terms of the basic gates ASUB_3 is implemented using only 10 basic gates which is very less compared to other designs.

In this section, we analyse the design metrics with a focus on operational delay as the primary factor. The operational delay values of models ASUB_1, ASUB_2, ASUB_3, ASUB_4, ASUB_5, ASUB_6, and ASUB_7 are compared to the EXACT subtractor models. It is observed that ASUB_1, ASUB_2, and ASUB_7 have the same delay values as the EXACT model. However, APSC3 shows a delay reduction of approximately 5.88%, while ASUB_4, ASUB_5, and ASUB_6 exhibit delays that are 5.88% greater than the exact subtractor.

Furthermore, the analysis considers the main factor of power consumption. The power consumed by models ASUB_1 and ASUB_7 is the same as the exact subtractors. However, the other models differ in power consumption. ASUB_2 consumes approximately 15.47% more power, while ASUB_3 demonstrates a power reduction of approximately 47.20%. Additionally, ASUB_4 and ASUB_5 consume approximately 11.28% and 40.70% more power, respectively. Figure 10 analyses the worst case scenarios of designs.

Further the efficiency of approximate subtractors are evaluated based on their error metrics such error distance (ED), mean relative error distance (MRED) [12], [13] and image quality metrics such as structural similarity index and peak signal to noise ratio (PSNR) [14], [25], [26].

4. CONCLUSION

In summary, the analysis of the approximate subtractors has provided valuable insights into their performance and characteristics when compared to the exact subtractor model. These subtractors have

undergone gate-level implementation, and their functional verification, synthesis, and analysis have been carried out using Synopsis. The results emphasize the potential benefits of using approximate subtractors in terms of power consumption, area efficiency, and operational delay. The model ASUB_3 will occupy less area compared to other subtractor which makes it as a stand out model, that trade-off between the values of area, power consumed and operational delay. Ultimately, selecting the best subtractor depends on specific design requirements and priorities, taking into account the trade-offs between power, area, and delay. The study identified the optimal approximate subtractor, demonstrating its potential to significantly enhance the speed of division operations. Further utilization of this identified subtractor promises to expedite the division process, paving the way for high-speed computations in various applications.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proceedings of the 2013 IEEE European Test Symposium (ETS'13)*, May. 2013, pp. 1–6, doi: 10.1109/ETS.2013.6569370.

[2] R. E. Blahut, *Fast algorithms for digital signal processing*. USA: Addison-Wesley, 1985.

[3] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 6, pp. 813–823, Dec. 2001, doi: 10.1109/92.974895.

[4] W. Wolf, *High-performance embedded computing: Architectures, applications, and methodologies*, 2nd ed. USA: Morgan Kaufmann, 2014.

[5] M. Priyadharshni and S. Kumaravel, "A comparative exploration about approximate full adders for error tolerant applications," in *ternational Symposium on VLSI Design and Test*, 2018, pp. 61–74, doi: 10.1007/978-981-13-5950-7.

[6] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*, Mar. 2010, pp. 957–960, doi: 10.1109/DATE.2010.5456913.

[7] G. N. Jyothi and S. SriDevi, "Distributed arithmetic architectures for FIR filters-A comparative review," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2017, pp. 2684–2690, doi: 10.1109/WiSPNET.2017.8300250.

[8] K. V. Palem, "Energy aware computing through probabilistic switching: a study of limits," *IEEE Transactions on Computers*, vol. 54, no. 9, pp. 1123–1137, Sep. 2005, doi: 10.1109/TC.2005.145.

[9] G. N. Jyothi and S. Sridevi, "High speed and low area decision feed-back equalizer with novel memory less distributed arithmetic filter," *Multimedia Tools and Applications*, vol. 78, no. 23, pp. 32679–32693, Dec. 2019, doi: 10.1007/s11042-018-7038-6.

[10] A. Gorantla and P. Deepa, "Design of approximate subtractors and dividers for error tolerant image processing applications," *Journal of Electronic Testing*, vol. 35, no. 6, pp. 901–907, Dec. 2019, doi: 10.1007/s10836-019-05837-5.

[11] M. Priyadharshni, A. R. Gupta, V. N. Kumar, and S. Kumaravel, "An error efficient and low complexity approximate multi-bit adder for image processing applications," *International Journal of Circuit Theory and Applications*, vol. 49, no. 8, pp. 2373–2381, Aug. 2021, doi: 10.1002/cta.3074.

[12] A. S. Roy and A. S. Dhar, "A novel approach for fast and accurate mean error distance computation in approximate adders," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351171.

[13] Y. Wu, Y. Li, X. Ge, Y. Gao, and W. Qian, "An efficient method for calculating the error statistics of block-based approximate adders," *IEEE Transactions on Computers*, vol. 68, no. 1, pp. 21–38, Jan. 2019, doi: 10.1109/TC.2018.2859960.

[14] G. Palubinskas, "Mystery behind similarity measures mse and SSIM," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 575–579, doi: 10.1109/ICIP.2014.7025115.

[15] R. Hajare and C. Lakshminarayana, "Design and software characterization of finFET based full adders," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 8, no. 1, pp. 51–60, Feb. 2019, doi: 10.11591/ijres.v8.i1.pp51-60.

[16] M. Z. Hussain and K. N. Parvin, "Low power and high performance FFT with different radices," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 8, no. 2, pp. 99–106, Jul. 2019, doi: 10.11591/ijres.v8.i2.pp99-106.

[17] C.-J. Soong, R. A. Rahman, R. Ramli, M. S. A. Manaf, and C.-C. Ting, "An evolutionary algorithm: an enhancement of binary tournament selection for fish feed formulation," *Complexity*, vol. 2022, no. 1, Jan. 2022, doi: 10.1155/2022/7796633.

[18] P. Pritty, M. Kumar, and M. Zunairah, "A body bias technique for low power full adder using XOR gate and pseudo NMOS transistor," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 8, no. 3, pp. 162–168, Nov. 2019, doi: 10.11591/ijres.v8.i3.pp162-168.

[19] A. Sandhu and S. Gupta, "Performance evaluation of an efficient five-input majority gate design in QCA nanotechnology," *Iranian Journal of Science and Technology - Transactions of Electrical Engineering*, 2019, doi: 10.1007/s40998-019-00296-2.

[20] N. Saravanakumar, K. S. Sudhan, K. N. Vijeyakumar, and S. Saranya, "Design and implementation of reduced power energy efficient binary coded decimal adder," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 8, no. 3, pp. 185–193, Nov. 2019, doi: 10.11591/ijres.v8.i3.pp185-193.

[21] C. M. R. Prabhu, T. W. X. Wilson, and T. Bhuvaneswari, "Low power 11T adder comparator design," *International Journal of Reconfigurable and Embedded Systems(IJRES)*, vol. 9, no. 1, pp. 28–33, 2020, doi: 10.11591/ijres.v9.i1.pp28-33.

[22] N. D. Kumar, S. R. Prasad, C. R. Kumari, and C. D. Naidu, "Design and analysis of different full adder cells using new technologies," *Lecture Notes in Electrical Engineering*, vol. 661, pp. 585–597, 2021, doi: 10.1007/978-981-15-4692-1_45.

[23] H. Jiang, L. Liu, F. Lombardi, and J. Han, "Approximate arithmetic circuits: design and evaluation," *Approximate Circuits*, pp. 67–98, 2019, doi: 10.1007/978-3-319-99322-5_4.

[24] C. G. N. and S. Kulkarni, "Enhanced MAC controller architecture for 2D processing based on FPGA with configurable resource allocation," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 10, no. 3, pp. 212–220, Nov. 2021, doi: 10.11591/ijres.v10.i3.pp212-220.

[25] D. R. I. M. Setiadi, "PSNR vs SSIM: imperceptibility quality assessment for image steganography," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8423–8444, Mar. 2021, doi: 10.1007/s11042-020-10035-z.

[26] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 2366–2369, doi: 10.1109/ICPR.2010.579.

BIOGRAPHIES OF AUTHORS

Dr M. Priyadharshni received her B.Tech. degree in Electronics and Communication Engineering from Anna University, Chennai, India in 2010, M.Tech. degree in VLSI Design from Veltech University, Chennai, India, in 2013 and Ph.D. degree in VLSI from Vellore institute of Technology, Vellore, India, in 2021. She is currently working at Panimalar Engineering College, Chennai. She can be contacted at email: priyadharshni.panimalar@gmail.com.

Rajermani Thinakaran received the Bachelor of Science degree in Computer Science from Universiti Teknologi Malaysia (UTM), Malaysia, in 1995, the Master's degree in IT from Univer- siti Kebangsaan Malaysia (UKM), in 2012, and the Ph.D. degree from UTM, in 2019. She is currently an Associate Professor with the Faculty of Data Science and Information Technology, INTI International University, Negeri Sembilan, Malaysia. She supervises both Bachelor's and Master's students (Master's and Ph.D. levels). She has more than 30 papers in international and local journals, international and national conference proceedings, as well as book chapters and lecture notes. Her research interests include artificial intelligence, assistive technology in empowering disabled students, elearning, and gaming ranging from theory to design to implementation. She also serves as a member of the Editorial Board and a Technical Reviewer for local and international journals and conferences. She can be contacted at email: rajermani.thina@newinti.edu.my.

Dr. Grande Naga Jyothi working as Asst. Professor in Madanapalle Institute of Technology and Science. She has completed her B.Tech. degree in the year of 2006 in JNTUA. She completed her M.Tech. from VIT University in the year of 2008. She completed her full time Ph.D. in VIT University in the year of 2020. She has around 20 scopus indexed journals and conferences. She has around 15 years of teaching experience. She can be contacted at email: nagajyothisai221@gmail.com.

Dr. Vijayakumar Varadarajan currently working as Professor and program leader in the Ajeenkya D Y Patil University, India. He completed his Ph.D. in Anna University in the year of 2012. He had around 205 journal papers and 20 conference papers. He is editorial member for many journals, book chapters and conferences. He had several funding projects and patents. He devoted computer scientist and engineer with a doctorate, 20 years of experience in industry and education, and creative managerial background at both domestic and foreign organizations. He can be contacted at email: vijayakumar.varadarajan@gmail.com.

C. Srinivasa Murthy obtained his B.E. from B.A.M.University Aurangabad in 2000. M.Tech. VLSI Design from Bharath University Chennai in 2005. He completed M.Tech. Computer Science and Engineering from Acharya Nagarjuna University Guntur 2010. He has more than 20 years of experience. He has a patent in the Bio Medical Engineering domain. His field of interest is VLSI, AI based bio medical engineering, and embedded systems. He can be contacted at email: srinivasamch@gmail.com.