Design and Analysis of a Power and Delay Optimized Radix -4 Approximate Subtractor.

Nithyashree R V, Shashidhar Tantry

Department of Electronics and Communication Engineering PES University Bengaluru, Karnataka

DOI: https://doi.org/10.51244/IJRSI.2026.1306000438

Received: 21 June 2026; Accepted: 26 June 2026; Published: 16 July 2026

ABSTRACT

Subtraction is a fundamental arithmetic operation used in arithmetic logic units (ALUs), digital signal processing, image processing, and machine learning applications. Conventional subtractor architectures provide exact computation but often incur high hardware complexity, power consumption, and propagation delay due to extensive borrow propagation. Although approximate computing has emerged as an effective approach for improving hardware efficiency in error-tolerant applications, most existing approximate subtractor designs are based on conventional Radix-2 architectures, offering limited reduction in the borrow critical path. To address this limitation, this work proposes accurate and approximate Radix-4 subtractor architectures that simplify the borrow-generation logic while preserving acceptable computational accuracy. The proposed approximation reduces logic depth, gate count, power consumption, and propagation delay by selectively modifying the borrow path. The architectures are implemented using Verilog HDL and implemented on tool FPGA synthesis analysis.Experimental results demonstrate that the proposed approximate Radix-4 subtractors achieve lower power consumption and delay than their accurate counterparts while maintaining a low error rate, making them suitable for error-resilient applications such as multimedia processing, image processing, and machine learning.

Keywords: Approximate Computing, Approximate Subtractor, Low Power VLSI, Error-Tolerant Systems, Digital Systems.

INTRODUCTION

Subtraction is a fundamental arithmetic operation extensively employed in arithmetic logic units (ALUs), digital filters, signal processing systems, control systems, and various numerical computing applications. It plays a vital role in image enhancement, digital communication, and many other computational tasks where efficient arithmetic processing is essential.

The growing demand for energy-efficient and high-performance electronic systems has increased the need for optimized arithmetic circuits. Conventional accurate subtractors provide precise computation but often suffer from high hardware complexity, increased power consumption, and longer propagation delay due to complex borrow-generation and propagation logic. However, many emerging applications, including multimedia processing, image processing, machine learning, and artificial intelligence, can tolerate small computational inaccuracies without significantly affecting the quality of the final output. Approximate computing has therefore emerged as an effective design paradigm that intentionally relaxes computational accuracy to achieve substantial improvements in power efficiency, circuit area, and operating speed.

Approximate arithmetic circuits have attracted significant research interest because they reduce hardware complexity by simplifying arithmetic logic. Several researchers have proposed approximate adders and subtractors to improve the energy efficiency of digital systems while maintaining acceptable computational accuracy [1]–[3]. Approximate arithmetic techniques have been successfully applied to low-power arithmetic units, digital signal processing systems, divider architectures, and error-resilient computing applications [4]–[12]. Furthermore, architectural optimization and logic approximation methods have demonstrated significant reductions in power consumption, propagation delay, and silicon area in modern VLSI systems [13]–[17].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Although considerable research has been carried out on approximate arithmetic circuits, most reported approximate subtractor architectures are based on conventional Radix-2 designs. These architectures primarily achieve hardware savings through logic simplification but still rely on borrow propagation across every bit position, resulting in increased critical path delay for higher-order subtractors. Moreover, existing approximation techniques generally focus on reducing logic complexity without exploiting higher-radix arithmetic to shorten the borrow propagation path. Consequently, there remains a need for an efficient approximate Radix-4 subtractor architecture that simultaneously reduces gate count, borrow-path complexity, propagation delay, and power consumption while maintaining acceptable computational accuracy.

To address these limitations, this work proposes accurate and approximate Radix-4 subtractor architectures in which the borrow-generation network is selectively simplified while preserving accurate difference computation. Unlike conventional approximation approaches that modify the complete arithmetic logic, the proposed architecture focuses only on the borrow path, thereby reducing logic depth, gate count, and propagation delay while minimizing error propagation. In addition, the proposed Radix-4 architecture performs two-bit subtraction in a single stage, reducing the borrow propagation path compared with conventional Radix-2 implementations. As a result, the proposed design offers improved hardware efficiency while maintaining low computational error, making it well suited for error-tolerant applications.

The main contributions of this work are summarized as follows:Design and implementation of accurate Radix-4 full and half subtractor architectures capable of performing two-bit subtraction in a single stage.Development of approximate Radix-4 full and half subtractors through selective borrow-path approximation while preserving accurate difference computation.Reduction of the borrow-path logic depth from four logic levels to three, resulting in lower propagation delay, reduced gate count, and improved hardware efficiency.Hardware implementation and functional verification of the proposed architectures using Verilog HDL and FPGA synthesis tools.Comparative evaluation of the proposed subtractors with accurate architectures in terms of gate count, power consumption, propagation delay, and computational error to demonstrate the effectiveness of the proposed approach.

BACKGROUND AND RELATED WORK

Accurate Radix-2 Full Subtractor

A conventional full subtractor generates difference and borrow outputs using XOR and combinational logic.

A conventional full subtractor performs subtraction of three input bits: minuend (A), subtrahend (B), and borrow input (Bin), producing difference (D) and borrow output (Bout).

This accurate implementation requires multiple logic gates and contributes to increased power consumption and propagation delay. Existing Radix-2 full subtractor is represented in Figure 1.The difference and borrow expressions are represented in equation (1) and (2) respectively.

The exact full subtractor logic is defined as:

$$\text{Difference} = A \oplus B \oplus Bin \quad (1)$$

$$\text{Borrow} = A'B + A' Bin + B Bin \quad (2)$$

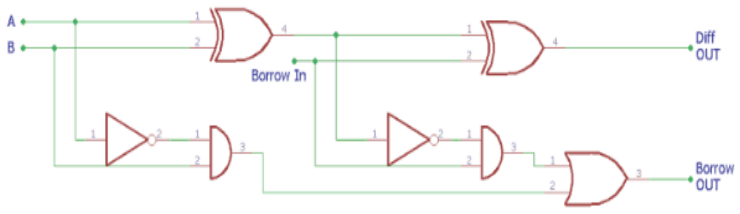

Figure 1: Radix-2 Accurate full subtractor.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

A	B	Bin	Diff Out	Borrow Out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Table 1: Truth table of Radix-2 Accurate full subtractor.

Approximate subtractors simplify the logic expressions by modifying borrow and difference generation, thereby reducing hardware complexity. Previous research has demonstrated that approximate arithmetic circuits can achieve significant improvements in power efficiency, area reduction, and speed while maintaining acceptable error levels.

Approximate arithmetic circuits have been widely applied in adders and multipliers. However, approximate subtractors remain relatively less explored, despite their importance in arithmetic processing systems.

This work proposes approximate subtractor designs with simplified logic expressions to reduce gate count, power consumption, and delay. In the further section will be discussing on radix-4 accurate and approximate subtractor.

Accurate Subtractor Architecture

Since Radix-2 subtractors are already available we have concentrated to design and test Radix- 4 subtractors where two bits can undergo subtraction simultaneously where critical path could be reduced. In this paper we have designed, tested accurate Radix- 4 subtractors and further approximated these Radix -4 subtractors and compared them with accurate Radix- 4 subtractors in next section . We have 2 cases in this design of accurate Radix-4 subtractors, in first case the design involves Radix- 4 accurate full subtractors with Bin which could be used in all bit positions other than LSB and the next case is Radix- 4 accurate half subtractors without Bin which could be used in LSB where there is no previous borrow in(Bin) .

Accurate Radix - 4 Full Subtractor.

Existing approximate subtractors are largely derived from Radix-2 arithmetic, where borrow propagation occurs across every bit position. As operand width increases, this borrow chain becomes a dominant contributor to propagation delay and energy consumption. Although several approximation methods simplify borrow logic, they do not exploit higher-radix subtraction to shorten the borrow propagation path. Therefore, an efficient approximate Radix-4 subtractor remains an open design opportunity.

Instead of using radix -2 accurate subtractors the Accurate Radix - 4 Subtractor can simultaneously subtract 2-bits which reduces critical path, delay and power. We have designed and tested accurate Radix-4 full and half subtractor and represented the results.

The Accurate Radix - 4 full subtractor is represented in Figure 2. This is with Bin circuit that can be used in all MSB positions other than LSB since all MSB bits has borrow in from previous bits. It can also be used on LSB bit position by making Bin as 0. The truth table Accurate Radix – 4 Full Subtractor is shown in Table I which involves two cases first case is when Bin is 0 and second case is when Bin is 1 since borrow can either be 0 or 1.Equation (3),(4),(5) represents the difference and borrow expressions for Accurate Radix – 4 Full Subtractor.

$$D0=A0 \oplus B0 \oplus \underline{Bin} \quad (3)$$

$$D1=A1 \oplus B1 \oplus (A0B0+(\overline{\underline{A0} \oplus B0})Bin) \quad (4)$$

$$Bout=\overline{A1}B1+(\overline{A1 \oplus B1})(A0B0+(\overline{A0 \oplus B0})Bin) \quad (5)$$

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

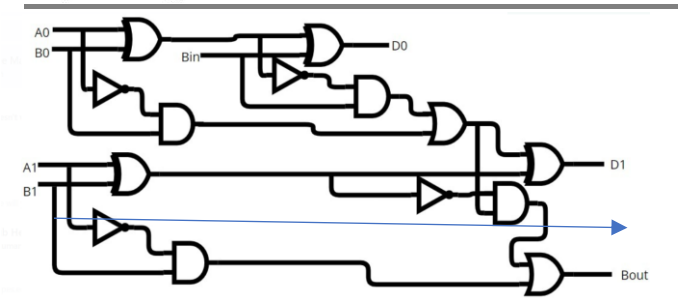

Figure 2: Accurate Radix – 4 Full Subtractor.

Inputs				When Bin = 0			When Bin = 1		
a1	a0	b1	b0	bout	Out1	Out0	bout	Out1	Out0
0	0	0	0	0	0	0	1	1	1
0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	0	1	0	1
0	0	1	1	1	0	1	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	1	1	1
0	1	1	0	1	1	1	1	1	0
0	1	1	1	1	1	0	1	0	1
1	0	0	0	0	1	0	0	0	1
1	0	0	1	0	0	1	0	0	0
1	0	1	0	0	0	0	1	1	1
1	0	1	1	1	1	1	1	1	0
1	1	0	0	0	1	1	0	1	0
1	1	0	1	0	1	0	0	0	1
1	1	1	0	0	0	1	0	0	0
1	1	1	1	0	0	0	0	1	1

Table 2: Truth table of Accurate Radix - 4 Full Subtractor.

Accurate Radix - 4 Half Subtractor.

The Accurate Radix – 4 half subtractor is designed without Bin that can be used in LSB position where there is no requirement of Bin pin that is where there is no previous borrow. Using this without Bin circuit in turn reduces gate count from 14 (Radix -4 full subtractor) to 10 (Radix-4 half subtractor). Figure 3 represents Accurate Radix - 4 half subtractor and the truth table for Accurate Radix - 4 half subtractor is shown in Table 3. In accurate Radix - 4 half subtractor critical path to Bout path involves 4 gates represented in blue color. Equation (6),(7),(8) represents the difference and borrow expressions for Accurate Radix – 4 Half Subtractor.

$$D0=A0 \oplus B0 \quad (6)$$

$$D1=(A1 \oplus B1) + \overline{A0}B0 \quad (7)$$

$$Bout=\overline{A1}B1+\overline{A0}B0(A1 \oplus B1) \quad (8)$$

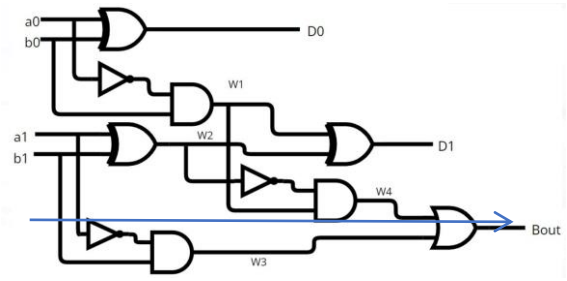

Figure 3: Accurate Radix - 4 half subtractor.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Inputs				Outputs		
a1	a0	b1	b0	out1	out0	bout
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	1	0	1
0	0	1	1	0	1	1
0	1	0	0	0	1	0
0	1	0	1	0	0	0
0	1	1	0	1	1	1
0	1	1	1	1	0	1
1	0	0	0	1	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	1	1	0
1	1	0	1	1	0	0
1	1	1	0	0	1	0
1	1	1	1	0	0	0

Table 3: Truth table of Accurate Radix - 4 half subtractor.

These accurate Radix - 4 full and half subtractors can be approximated to further reduce delay and power but it does give some error bits but these can be used in error tolerant applications. Also, further in higher order bit subtractors, the LSB bits could be replaced with approximate subtractors to reduce hardware cost, MSB bits may retain accurate subtractors to limit error propagation.

Approximate Subtractor Architecture

Approximate Radix - 4 Full Subtractor.

Approximate subtractors are designed by simplifying the difference and borrow-generation logic to reduce hardware complexity, gate count, propagation delay, and power consumption compared with accurate subtractors. In the proposed Radix-4 full subtractor, the approximation is performed primarily in the borrow-generation network, as timing analysis of the accurate Radix-4 subtractor shows that the borrow path forms the critical path due to signal propagation through multiple logic levels before reaching the output.

Unlike conventional approximate subtractors that simplify the complete arithmetic logic, the proposed architecture selectively approximates only the borrow-generation network while preserving accurate difference computation. This targeted approximation minimizes error propagation while reducing the logic depth of the borrow path from four logic levels to three. Consequently, the proposed design achieves lower propagation delay, reduced gate count, and improved power efficiency without significantly affecting computational accuracy.

The Approximate Radix-4 Full Subtractor is derived from the Accurate Radix-4 Full Subtractor shown in Figure 2. The simplified borrow-generation logic results in a shorter critical path for the borrow output (Bout), as illustrated by the blue-highlighted path in Figure 4. The difference and borrow expressions of the proposed Approximate Radix-4 Full Subtractor are given in Equations (9)-(11), while its corresponding truth table is presented in Table 4. The erroneous output combinations introduced by the approximation are highlighted in red to illustrate the limited impact of the proposed simplification.

$$D0 = A0 \oplus B0 \oplus Bin \quad (9)$$

$$D1=A1\oplus B1\oplus(\overline{A0}B0+(\overline{A0\oplus B0})Bin) \quad (10)$$

$$Bout = \overline{A1}(B1+\overline{A0}B0+\overline{A0}Bin+B0Bin) \quad (11)$$

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

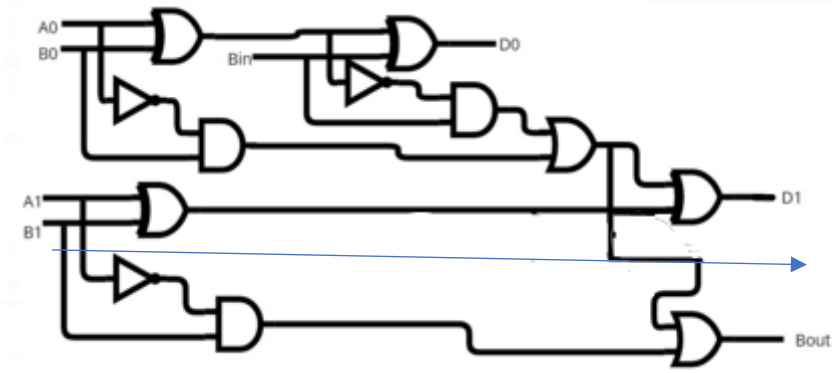

Figure 4: Approximate Radix - 4 full subtractor.

Inputs				When Bin = 0			When Bin = 1		
a1	a0	b1	b0	bout	Out1	Out0	bout	Out1	Out0
0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	1	1	0	1	0
0	0	1	0	1	1	0	1	0	1
0	0	1	1	1	0	1	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	1	1
0	1	1	0	1	1	1	1	1	0
0	1	1	1	1	1	0	1	0	1
1	0	0	0	0	1	0	0	0	1
1	0	0	1	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	1
1	0	1	1	0	1	1	0	1	0
1	1	0	0	0	1	1	0	1	0
1	1	0	1	0	1	0	0	0	1
1	1	1	0	0	0	1	0	0	0
1	1	1	1	0	0	0	0	1	1

Table 4 : Truth table of Approximate Radix - 4 full subtractor.

Approximate Radix - 4 Half Subtractor.

The Approximate Radix-4 Half Subtractor is designed for the least significant bit (LSB) position, where a borrow input (Bin) is not required. Since the LSB stage does not receive a borrow from a preceding stage, approximating the half subtractor introduces fewer computational errors than approximating subtractors in higher-order bit positions, making it well suited for error-tolerant arithmetic architectures.

The proposed Approximate Radix-4 Half Subtractor is derived from the Accurate Radix-4 Half Subtractor by selectively simplifying the borrow-generation network while preserving accurate difference computation. Similar to the approximate full subtractor, the approximation is confined to the borrow output (Bout) path, which constitutes the critical path of the circuit. This targeted simplification reduces the logic depth of the borrow path, leading to lower gate count, reduced propagation delay, improved area efficiency, and lower power consumption.

The proposed architecture is implemented using Verilog HDL and verified through functional simulation. The Approximate Radix-4 Half Subtractor is illustrated in Figure 5, and its corresponding truth table is presented in Table 5, where the erroneous output combinations are highlighted in red. The simulation results show that the approximation affects only the borrow output (Bout), while both difference outputs (D1 and D0) remain accurate for all input combinations. Furthermore, the critical path of the borrow output is reduced to three logic levels, as indicated by the blue-highlighted path in Figure 5, and the overall design requires only eight logic gates. The difference and borrow expressions of the proposed Approximate Radix-4 Half Subtractor are given in Equations (12)–(14).

$D0=A0 \oplus B0 \quad \_\_ \quad (12)$

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

$D1=(A1 \oplus B1) + A0B0$ (13)

$Bout=\overline{A0}B0+\overline{A1}B1$ (14)

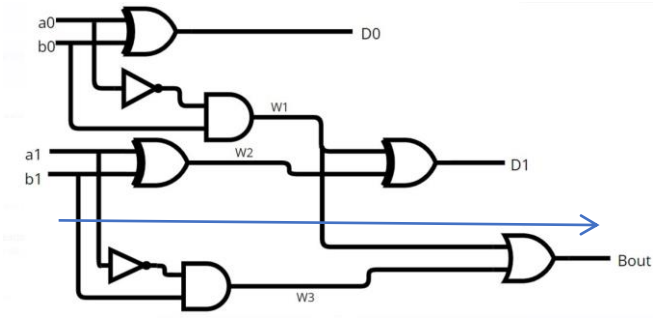

Figure 5: Approximate Radix-4 half subtractor .

Inputs				Outputs		
a1	a0	b1	b0	out1	out0	bout
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	1	0	1
0	0	1	1	0	1	1
0	1	0	0	0	1	0
0	1	0	1	0	0	0
0	1	1	0	1	1	1
0	1	1	1	1	0	1
1	0	0	0	1	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	1	1	0
1	1	0	1	1	0	0
1	1	1	0	0	1	0
1	1	1	1	0	0	0

Table 5:Truth table of approximate Radix-4 half subtractor .

A gate-level comparison between the accurate and approximate Radix-4 subtractor architectures demonstrates the hardware efficiency achieved through selective borrow-path approximation. The Accurate Radix-4 Full Subtractor shown in Figure 2 requires 14 logic gates, whereas the proposed Approximate Radix-4 Full Subtractor shown in Figure 4 requires only 12 logic gates. Similarly, the Accurate Radix-4 Half Subtractor shown in Figure 3 uses 10 logic gates, while the proposed Approximate Radix-4 Half Subtractor shown in Figure 5 requires only 8 logic gates. The reduction in gate count is achieved by simplifying the borrow-generation network, which forms the critical path in higher-order subtractor circuits. Consequently, the proposed architectures reduce hardware complexity, shorten the critical path, and improve power and delay performance.

The proposed subtractor architectures are implemented using Verilog HDL and synthesized using FPGA design tools to evaluate key performance metrics, including power consumption, propagation delay, and error rate. Functional simulation confirms the correctness of the proposed designs, while synthesis results demonstrate improvements in hardware efficiency compared with the corresponding accurate architectures. Furthermore, the proposed Radix-4 subtractors can be employed as efficient building blocks for higher-order arithmetic circuits, such as multi-bit ripple-borrow subtractors.

IMPLEMENTATION AND RESULTS

The proposed Radix-4 subtractor architectures were implemented using Verilog HDL in the Quartus design environment and synthesized on an FPGA on tool platform to evaluate their performance. The evaluation focuses on key performance metrics, including error rate, power consumption, and propagation delay.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Table 6 presents a comparison of the error characteristics of the accurate and approximate Radix-4 full and half subtractors. As expected, the accurate subtractors produce error-free outputs, whereas the proposed approximate subtractors introduce only a small error rate, demonstrating that the approximation achieves improved hardware efficiency while maintaining acceptable computational accuracy.

Table 7 compares the power consumption of the proposed architectures. The results indicate that the Radix-4 subtractor consumes less power than the equivalent implementation using two Radix-2 subtractors. Furthermore, the proposed approximate Radix-4 subtractors achieve an additional reduction in power consumption due to the simplified borrow-generation logic.

Table 8 compares the propagation delay of the evaluated architectures. The proposed approximate Radix-4 subtractors exhibit lower propagation delay than the corresponding accurate Radix-4 subtractors and significantly outperform the equivalent two-stage Radix-2 implementation. The reduction in delay is primarily attributed to the shortened borrow critical path achieved through selective borrow-path approximation.

Sl. No	**Subtractor types**	**Error (%)**
1.	Radix-2 accurate full subtractor	0
2.	Radix-4 accurate full subtractor	0
3.	Radix-4 approximate full subtractor	7.29
4.	Radix-4 accurate half subtractor	0
5.	Radix-4 approximate half subtractor	4.16

Table 6: Comparison of error results of accurate and approximate subtractor.

Sl. No	**Subtractor types**	**Power (µW)**
1.	Radix-2 accurate full subtractor	0.973 (each radix-2) 0.973x2=**1.976** (2 radix-2 subtractors)
2.	Radix-4 accurate full subtractor	1.672
3.	Radix-4 approximate full subtractor	1.564
4.	Radix-4 accurate half subtractor	1.404
5.	Radix-4 approximate half subtractor	1.294

Table 7: Comparison of power results of accurate and approximate subtractor.

Sl. No	**Subtractor types**	**Delay (ns)**
1.	Radix-2 accurate full subtractor	6.798 (each radix-2) 6.798x2=**13.596** (2 radix-2 subtractors)
2.	Radix-4 accurate full subtractor	7.377

3.	Radix-4 approximate full subtractor	7.176
4.	Radix-4 accurate half subtractor	6.976
5.	Radix-4 approximate half subtractor	6.776

Table 8: Comparison of delay results of accurate and approximate subtractor.

CONCLUSION

This work presented the design and analysis of accurate and approximate Radix-4 subtractor architectures for low-power and high-speed arithmetic applications. The proposed approximate architectures were developed by selectively simplifying the borrow-generation logic while preserving accurate difference computation. This targeted approximation reduced the critical path in the borrow network, resulting in lower hardware complexity, power consumption, and propagation delay compared with the corresponding accurate Radix-4 subtractors. Functional verification and FPGA on tool synthesis confirmed the effectiveness of the proposed designs.

The experimental results show that the proposed approximation introduces errors only in the borrow output, while both difference outputs remain exact. Since borrow errors occur only for a limited number of input combinations, the overall error rate remains low, making the proposed architecture suitable for error-tolerant applications such as image processing, multimedia systems, and machine learning accelerators. Furthermore, the proposed Radix-4 architecture provides an efficient building block for higher-order subtractors by reducing the borrow critical path and improving hardware efficiency. Future work will focus on extending the proposed design to multi-bit subtractor architectures and investigating advanced approximation techniques to further optimize the trade-off between accuracy, power consumption, and delay.

REFERENCES

1. Ferreira, R., M. Leme, M. Corrêa, L. Agostini, C. Diniz and B. Zatt, "Approximate Subtractor Operator for Low-Power Video Coding Hardware Accelerators," 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 2019, pp. 426-429, doi: 10.1109/ICECS46596.2019.8964783.
2. Gupta, V., D. Mohapatra, A. Raghunathan, and K. Roy, “Low-power digital signal processing using approximate adders,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124–137, Jan. 2013.
3. Han, J. and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in Proceedings of the IEEE European Test Symposium, 2013, pp. 1–6.
4. Hashemi, S., R. Bahar, and S. Reda, “DRUM: A dynamic range unbiased multiplier for approximate applications,” in Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2015, pp. 418–425.
5. Jiang, H., C. Liu, J. Han, and F. Lombardi, “Approximate arithmetic circuits: A survey, characterization, and recent applications,” IEEE Access, vol. 5, pp. 20297–20317, 2017.
6. Kaushik, N. and B. Srinivasu, "High-Speed and Area-Efficient Serial IMPLY-Based Approximate Subtractor and Comparator for Image Processing and Neural Networks," IEEE Transactions on Nanotechnology, vol. 23, pp. 748-757, 2024, doi: 10.1109/TNANO.2024.3487223.
7. Liu, C., J. Han, and F. Lombardi, “A low-power approximate adder with configurable error recovery,” in Proceedings of Design, Automation & Test in Europe (DATE), 2014, pp. 1–4.
8. Nithyashree, R. V., et al., “Analysis of various approximate adders,” in Proceedings of IEEE International Conference on Electronics and Communication Systems, 2023, pp. 1–5.
9. Palem, K. V., “Energy aware computing through probabilistic switching: A study of limits,” IEEE Transactions on Computers, vol. 54, no. 9, pp. 1123–1137, Sept. 2005.
10. Raha, A., H. Jayakumar, and V. Raghunathan, “Input-based dynamic reconfiguration of approximate arithmetic units for video encoding,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 3, pp. 846–857, Mar. 2016.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

11. Rehman, S., M. Shafique, A. Kumar, and J. Henkel, “Design and analysis of approximate ripple carry adders,” in Proceedings of IEEE International Conference on Computer Design (ICCD), 2014, pp. 1–6.
12. Rehman, S., W. El-Harouni, M. Shafique, and J. Henkel, “Architectural-space exploration of approximate multipliers,” in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2016, pp. 1–8.
13. Shafique, M., S. Rehman, R. Hafiz, and J. Henkel, “Approximate computing: From circuits to systems,” in Proceedings of the Design Automation Conference (DAC), 2015, pp. 1–6.
14. Shafique, M., W. Ahmad, R. Hafiz, and J. Henkel, “Approximate computing: Techniques and applications,” IEEE Design & Test, vol. 33, no. 1, pp. 8–22, Feb. 2016.
15. Venkataramani, S., A. Sabne, V. Kozhikkottu, K. Roy, and A. Raghunathan, “SALSA: Systematic logic synthesis of approximate circuits,” in Proceedings of the 49th Design Automation Conference (DAC), 2012, pp. 796–801.
16. Verma, A. K., P. Brisk, and P. Ienne, “Variable latency speculative adders for approximate computing,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 11, pp. 1874–1885, Nov. 2008.
17. Xu, Q., T. Mytkowicz, and N. S. Kim, “Approximate computing: A survey,” IEEE Design & Test, vol. 33, no. 1, pp. 8–22, Feb. 2016.