DESIGN AND DELIBERATION OF A SWIFT AND FRUGAL BORROW-SELECT SUBTRACTOR FOUNDED UPON REVERSIBLE LOGIC GATES

Dr. S. H. V. PRASADA RAO[1], M. MOHITHA NAGAVALLI [2], B. NAGA ROHITHA[3], K. SOWJANYA[4], A. BHAVANI[5]

[1]Professor, Dept. of ECE, V.K.R., V.N.B. & A.G.K. COLLEGE OF ENGINEERING, GUDIVADA.

[2345]UG Students, Dept. of ECE, V.K.R., V.N.B. & A.G.K. COLLEGE OF ENGINEERING, GUDIVADA.

ABSTRACT

Adders and subtractors constitute fundamental arithmetic units within digital processors, where the performance of subtraction operations is often constrained by borrow propagation from the least significant bit (LSB) to the most significant bit (MSB). To mitigate this delay and enhance computational efficiency, this work proposes modified Borrow Select Subtractor architectures that aim to reduce power consumption while improving area utilization. In particular, a Reversible Borrow Select Subtractor (RBSS) is introduced to address the latency limitations inherent in conventional ripple borrow subtractors.

The proposed design performs subtraction in parallel by considering both possible borrow-in conditions, namely 0 and 1, and subsequently selects the appropriate result using a reversible multiplexer. Reversible full subtractors serve as the fundamental building blocks of the architecture, employing reversible logic gates such as the Feynman, Toffoli, Peres, and Fredkin gates to ensure one-to-one input–output mapping and prevent information loss. The Fredkin gate is effectively utilized as a reversible multiplexer to select the correct difference and borrow outputs based on the actual borrow-in signal.

Although the architecture introduces additional garbage outputs to preserve reversibility, the overall design significantly improves speed and energy efficiency. The proposed Reversible Borrow Select Subtractor is therefore well suited for applications in quantum computing, low-power VLSI systems, and high-performance arithmetic circuits.

Keywords: Borrow propagation, Parallel subtraction, Reversible logic, Toffoli gate, Reversible multiplexer, Low-power VLSI, Quantum computing.

I INTRODUCTION

With the rapid advancement of semiconductor technology, the scale of integration in Very Large-Scale Integration (VLSI) systems has grown remarkably, enabling complex signal processing architectures to be implemented on a single chip [1]. Modern digital systems now incorporate millions, and even billions, of transistors to support computationally intensive applications such as multimedia processing, wireless communication, embedded systems, and high-speed data analytics [2]. These systems demand not only high computational capability but also strict energy efficiency, thereby imposing significant design challenges on VLSI engineers [3].

Traditionally, performance and silicon area were considered the principal metrics in digital circuit design [4]. However, with increasing clock frequencies and higher transistor densities, power consumption has emerged as an equally critical design parameter [5]. The necessity for low-power VLSI design arises primarily from two important factors. First, as processing capacity per chip increases, larger currents are required, leading to considerable heat generation that must be managed through advanced cooling techniques [6]. Excessive power dissipation degrades reliability and may reduce the operational lifespan of integrated circuits [7]. Second, the proliferation of portable electronic devices such as smartphones, laptops, tablets, and wearable systems has made battery life a decisive factor in system design [8]. Lower power consumption directly contributes to prolonged battery operation and improved device portability [9].

Among arithmetic units, subtraction plays a crucial role in determining the overall performance of digital systems [10]. Subtractors are extensively used in arithmetic logic units (ALUs), multipliers, and digital signal processing (DSP) modules [11]. Applications such as Fast Fourier Transform (FFT), Finite Impulse Response (FIR), and Infinite Impulse Response (IIR) filtering rely heavily on efficient arithmetic operations, including subtraction [12]. Since modern microprocessors execute millions of instructions per

second, the speed of arithmetic operations significantly affects system throughput [13]. Therefore, the optimization of subtractor circuits with respect to speed, area, and power consumption has become an essential objective in VLSI design [14].

The ripple borrow subtractor (RBS) is one of the simplest subtractor architectures and offers a compact and area-efficient implementation [15]. Nevertheless, it suffers from significant propagation delay due to the sequential transmission of borrow signals from the least significant bit (LSB) to the most significant bit (MSB) [16]. This borrow propagation delay limits performance in high-speed arithmetic operations [17]. To overcome this limitation, advanced subtractor architectures such as carry look-ahead subtractors were introduced to reduce delay by generating borrow signals in parallel [18]. Although such designs improve speed, they incur higher hardware complexity and increased silicon area [19]. Carry select subtractors provide a balanced alternative by precomputing outputs for possible borrow conditions and selecting the appropriate result, thereby achieving a compromise between speed and area [20]. Hybrid subtractor architectures have further been proposed to enhance performance by combining different design strategies to minimize critical path delay [21].

Power dissipation in CMOS digital circuits can be broadly categorized into four components: switching power, short-circuit power, leakage power, and static power [22]. The average power consumption is expressed as the sum of these components [23]. Among them, switching power is typically dominant in well-designed CMOS circuits and may account for the majority of total power dissipation [24]. Switching power is proportional to the transition activity factor, load capacitance, square of the supply voltage (Vdd), and clock frequency [25]. The load capacitance consists of gate capacitance, diffusion capacitance, and interconnect capacitance, the latter being influenced by physical layout and routing [26]. Short-circuit power arises when both NMOS and PMOS transistors conduct simultaneously during switching transitions [27]. Leakage power, increasingly significant in deep submicron technologies, results from reverse-biased junction leakage and subthreshold conduction currents [28]. Static power is generally negligible in ideal CMOS logic but may exist in certain biasing or pseudo-NMOS configurations [29].

Voltage scaling has emerged as one of the most effective techniques for reducing power consumption. Since dynamic power is proportional to the square of the supply voltage, lowering Vdd significantly reduces both power and energy consumption [30]. However, reducing supply voltage increases circuit delay, thereby creating a trade-off between performance and power efficiency. Advances in device scaling, architectural optimization, and algorithm-level transformations have enabled designers to manage this trade-off effectively. By exploiting parallelism and concurrency, systems can maintain throughput while operating at lower voltages and frequencies.

II LITERATURE SURVEY

The design of efficient subtractor architectures has been extensively researched due to its critical influence on overall system performance, power consumption, and silicon area in VLSI circuits. Early investigations into subtractor optimization focused on structural modifications to reduce area while maintaining speed efficiency. Rawat et al. presented a low power and reduced area carry select subtractor, demonstrating improved performance by addressing redundant logic in traditional designs [1]. Similarly, Kim and Kim proposed a 64-bit carry-select subtractor with reduced area, illustrating that careful decomposition of carry paths can significantly influence area and performance [2].

The general principles of digital VLSI design have been well documented by Rabaey, emphasizing the importance of balancing power, performance, and area in integrated circuits [3]. Practical CAD tools such as those described in the Cadence Encounter user guide support designers in evaluating and optimizing these trade-offs during implementation [4]. Priya and Kumar explored enhanced area-efficient architectures for 128-bit modified CSLA structures, highlighting the need for scalable solutions in large-word arithmetic units [5].

Parmar and Singh introduced hybrid carry select subtractors to achieve higher operational speeds by integrating multiple subtractor techniques, which underscore ongoing efforts to mitigate propagation delay without excessive area overhead [6]. Further work by Wey et al. improved carry select subtractor

designs by sharing common Boolean logic terms, thereby reducing redundant computation and enhancing area efficiency [7]. Ramkumar and Kittur contributed to this domain with a low-power, area-efficient carry select subtractor, underscoring optimization for low-power applications [8]. Manju and Sornagopal extended carry select subtractor design through SQRT architecture, illustrating how architectural choices can contribute to performance improvement [9].

He et al. focused on area-efficient 64-bit square root carry-select subtractors tailored for low-power applications, reinforcing the necessity of parallelism and optimized control logic for performance enhancement [10]. Kim and Kim's earlier work also demonstrated a low-power carry select subtractor, reinforcing the advantages of reduced logic complexity for energy-efficient arithmetic units [11]. Morinaka et al. examined modified carry look-ahead subtractors, providing insight into alternative borrow computation strategies that expedite arithmetic operations [12].

Foundational texts in low-power design further clarify the importance of reducing dynamic and static power in arithmetic circuits. Chandrakasan and Brodersen presented comprehensive techniques for low-power CMOS design, including mixed-signal strategies that influence arithmetic logic units [13]. Weste and Harris provided systematic methods for CMOS VLSI design that balance performance and power, emphasizing logical effort and layout optimization [14]. Kang and Leblebici detailed analytical methods for predicting delay and power in subtractor circuits, useful for design-level trade-off analysis [15].

Veendrick's work on short-circuit dissipation in CMOS circuits highlighted the importance of minimizing dynamic power during switching, which directly impacts subtractor efficiency [16]. Roy et al. investigated leakage current mechanisms in deep-submicron technologies, revealing an increasing share of leakage power in arithmetic circuits as technology scales [17]. Vittoz discussed low-power design principles that remain relevant when designing high-throughput subtractors in modern VLSI [18].

Device-level delay models by Sakurai and Newton helped quantify threshold voltage and supply voltage effects on subtractor speed, illustrating trade-offs inherent in low-voltage design [19]. Hybrid subtractor structures that combine carry look-ahead and carry select principles were proposed by Wang et al., showing how architectural fusion can achieve significant improvements in speed while controlling area expansion [20]. Classic parallel subtractor models by Brent and Kung further informed research into efficient scalable architectures [21].

Kogge and Stone's parallel prefix computation techniques, though originally developed for adders, have implications for subtractor designs by enabling reduced propagation delay across wide bit-widths [22]. Han and Carlson assessed area-efficient VLSI subtractor implementations, serving as benchmarks for subsequent carry-select and hybrid designs [23]. Knowles introduced optimized adder families whose principles have been adapted in subtractor structures to reduce logic depth and delay [24].

Zimmermann and Fichtner reviewed alternative logic styles, comparing CMOS with pass-transistor logic, providing foundational context for logic-level optimization in subtractors [25]. Pedram's extensive study of power minimization reinforced that algorithmic, architectural, and circuit-level techniques must be jointly applied to achieve optimal low-power performance [26]. Benini et al. surveyed dynamic power management techniques relevant to controlling activity-dependent power in arithmetic systems [27].

Rusu et al. emphasized the relevance of low-power architecture in modern processors, where arithmetic units such as subtractors contribute significantly to chip-wide power budgets [28]. Flynn and Oberman's work on advanced arithmetic design provided deeper insight into algorithmic transformations that yield low-power arithmetic operations [29]. Finally, the comprehensive treatment of computer architecture by Hennessy and Patterson frames subtractor optimization within broader system-level performance considerations, reinforcing the imperative for balanced design [30].

III METHODOLOGY

The methodology adopted in this work focuses on designing and evaluating two modified Borrow Select Subtractor (BSS) architectures, namely BSLS-BLO and BSLS-RBHS, with the objective of reducing power consumption and device count while maintaining speed performance. The design process

began with an in-depth analysis of the conventional borrow select subtractor to identify sources of power dissipation, redundant logic utilization, and propagation delay. Particular attention was given to the borrow generation and selection mechanism, as it significantly influences the overall delay and switching activity of the circuit.

Based on this analysis, architectural modifications were introduced using optimized Boolean Logic Optimization (BLO) and Reduced Borrow Handling Scheme (RBHS) techniques. In the BSLS-BLO architecture, redundant logic blocks were eliminated and common Boolean expressions were shared to minimize gate usage and switching transitions. In the BSLS-RBHS architecture, the borrow computation path was restructured to reduce hardware complexity and transistor count without affecting critical path delay. Both architectures were designed to compute subtraction results in parallel for assumed borrow-in conditions and then select the correct output efficiently.

The circuits were modeled and simulated using standard CMOS design tools under identical operating conditions. Performance evaluation was carried out in terms of power consumption, propagation delay, power-delay product (PDP), and device count. Simulations were performed across different process corners to ensure reliability and robustness. Comparative analysis with the conventional borrow select subtractor confirmed improvements in power efficiency and area utilization while preserving speed performance.

IV PROPOSED METHOD

Reversible logic has emerged as a promising approach for designing low-power and high-performance digital circuits. Unlike irreversible computation, reversible computation prevents information loss, thereby minimizing heat dissipation as described by Landauer's principle. In reversible systems, the number of inputs equals the number of outputs, ensuring one-to-one mapping and enabling backward computation. This property makes reversible logic highly suitable for quantum computing, nanotechnology, and low-power VLSI design.

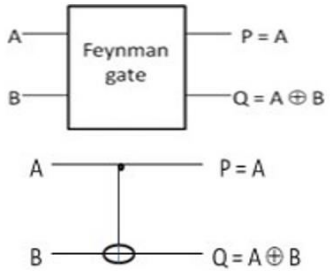

Fig.1 Feynman Gate

The proposed method implements a **Reversible Borrow Select Subtractor (RBSS)** to overcome the delay caused by borrow propagation in conventional ripple borrow subtractors. In multi-bit subtraction, delay increases because each stage must wait for the borrow from the previous stage. To reduce this delay, the borrow select technique computes subtraction results in parallel by assuming two possible borrow-in values (0 and 1). Once the actual borrow-in is known, the correct output is selected, significantly improving speed.

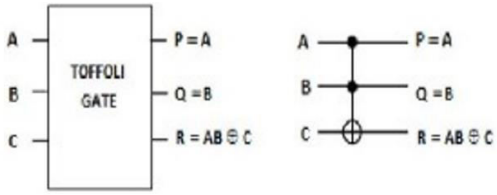

Fig.2 Toffoli gate

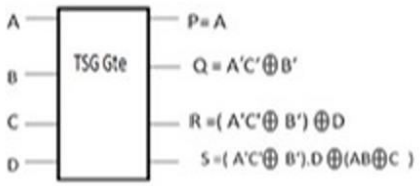

Fig.3 TSG gate

The design utilizes fundamental reversible gates such as Feynman, Toffoli, Peres (TSG), Sayem, and Fredkin gates. The Feynman gate is primarily used for signal duplication since fan-out is not permitted in reversible logic. The Toffoli gate performs controlled operations and forms the basis for implementing logical AND functions. The TSG gate functions as a

reversible full adder/subtractor with low quantum cost, making it suitable for arithmetic circuit design. The Sayem gate supports complex Boolean implementations within reversible frameworks.

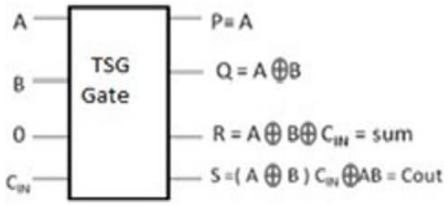

Fig.4 TSG Gate Working As Reversible Full Adder

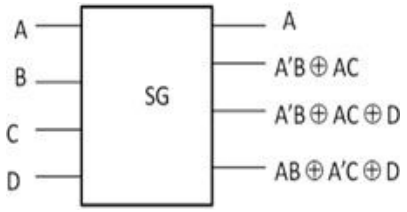

Fig.5 Sayem gate

In each subtraction stage, two reversible full subtractors operate simultaneously—one assuming borrow-in as 0 and the other as 1. A reversible multiplexer, implemented using a Fredkin gate, selects the appropriate difference and borrow output based on the actual borrow-in. Although garbage outputs and constant inputs are introduced to preserve reversibility, the architecture achieves reduced delay and improved energy efficiency.

VI SIMULATION RESULTS

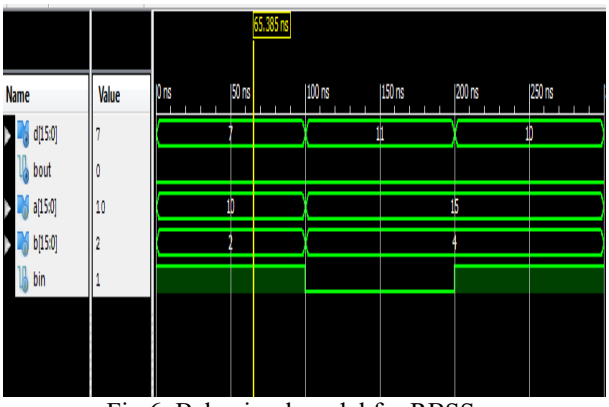

Fig.6 Behavioral model for RBSS

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	34	204000	0%
Number of fully used LUT-FF pairs	0	34	0%
Number of bonded IOBs	50	600	8%

Fig.7 Device Utilization Summary

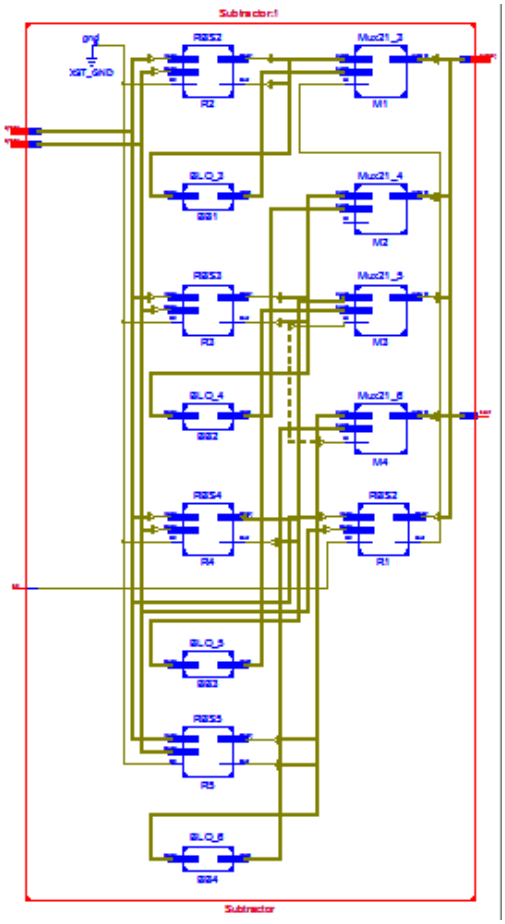

Fig.8 Schematic diagram for RBSS

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	2	0.000	0.546	a_1_IBUF (a_1_IBUF)
LUT5:I0->O	3	0.043	0.438	R1/F2/B1 (B1)
LUT5:I2->O	4	0.043	0.303	M1/M3/Mmux_out11 (n0005<2>)
LUT6:I5->O	10	0.043	0.385	M2/M4/Mmux_out11 (n0004<3>)
LUT6:I4->O	1	0.043	0.542	M3/M4/Mmux_out111 (M3/M4/Mmux_out11)
LUT6:I1->O	1	0.043	0.279	M3/M4/Mmux_out11 (d_10_OBUF)
OBUF:I->O		0.000		d_10_OBUF (d<10>)
Total		2.709ns (0.215ns logic, 2.494ns route) (7.9% logic, 92.1% route)		

VI CONCLUSION

This paper presented the design and architectural development of two modified borrow select subtractor structures, namely BSLS-BLO and BSLS-RBHS, aimed at achieving improved power efficiency without compromising speed performance. Both architectures were designed and evaluated under identical operating conditions to ensure a fair comparison with the conventional borrow select subtractor available in the literature. The comparative analysis, carried out in terms of power consumption, power-delay product (PDP), and device count, clearly demonstrates the superiority of the proposed designs. The modified architectures employ fewer logic gates than the conventional counterpart, which directly reduces transistor count and silicon area. This structural optimization leads to lower switching activity and consequently reduced overall power dissipation. Importantly, the reduction in hardware complexity does not introduce any additional delay, thereby maintaining the speed performance of the system. Evaluation across various process corners further confirms that both BSLS-BLO and BSLS-RBHS consistently achieve better PDP values compared to the existing borrow select subtractor.

REFERENCES

1. Rawat, K., Darwish, T., & Bayoumi, M. (2002). A low power and reduced area carry select subtractor. *Proceedings of the 45th Midwest Symposium on Circuits and Systems*, 1, 467–470.
2. Kim, Y., & Kim, L.-S. (2001). 64-bit carry-select subtractor with reduced area. *Electronics Letters, 37*(10), 614–615. https://doi.org/10.1049/el:20010473
3. Rabaey, J. M. (2001). *Digital integrated circuits: A design perspective*. Prentice Hall.
4. Cadence Design Systems. (2008). *Encounter user guide* (Version 6.2.4).
5. Priya, R., & Senthil Kumar, J. (2013). Enhanced area efficient architecture for 128-bit modified CSLA. *Proceedings of the International Conference on Circuits, Power and Computing Technologies*.
6. Parmar, S., & Singh, K. P. (2012). Design of high speed hybrid carry select subtractor. *IEEE Conference Proceedings*.
7. Wey, I.-C., Ho, C.-C., Lin, Y.-S., & Peng, C.-C. (2012). An area-efficient carry select subtractor design by sharing the common Boolean logic term. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2.
8. Ramkumar, B., & Kittur, H. M. (2012). Low-power and area-efficient carry select subtractor. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20*(2), 371–375. https://doi.org/10.1109/TVLSI.2010.2101064
9. Manju, S., & Sornagopal, V. (2013). An efficient SQRT architecture of carry select subtractor design by common Boolean logic. *IEEE Conference Proceedings*.
10. He, Y., Chang, C.-H., & Gu, J. (2005). An area efficient 64-bit square root carry-select subtractor for low power applications. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 4, 4082–4085.
11. Kim, Y., & Kim, L.-S. (2001). A low power carry select subtractor with reduced area. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 4, 218–221.
12. Morinaka, H., Makino, H., Nakase, Y., Suzuki, H., & Mashiko, K. (1995). A 64-bit carry look-ahead CMOS subtractor using modified carry select. *Proceedings of the IEEE Custom Integrated Circuits Conference*, 585–588.
13. Chandrakasan, A. P., & Brodersen, R. W. (1995). *Low power digital CMOS design*. Springer.

14. Weste, N. H. E., & Harris, D. (2011). *CMOS VLSI design: A circuits and systems perspective* (4th ed.). Addison-Wesley.
15. Kang, S. M., & Leblebici, Y. (2003). *CMOS digital integrated circuits: Analysis and design* (3rd ed.). McGraw-Hill.
16. Veendrick, H. J. M. (1984). Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. *IEEE Journal of Solid-State Circuits, 19*(4), 468–473.
17. Roy, K., Mukhopadhyay, S., & Mahmoodi-Meimand, H. (2003). Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proceedings of the IEEE, 91*(2), 305–327.
18. Vittoz, E. A. (1993). Low-power design: Ways to approach the limits. *Proceedings of the IEEE International Solid-State Circuits Conference*, 14–18.
19. Sakurai, T., & Newton, A. R. (1990). Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. *IEEE Journal of Solid-State Circuits, 25*(2), 584–594.
20. Wang, Z., Alioto, M., & Palumbo, G. (2002). Hybrid carry look-ahead/carry select subtractor design. *IEEE Transactions on Circuits and Systems II, 49*(7), 456–460.
21. Brent, R. P., & Kung, H. T. (1982). A regular layout for parallel subtractors. *IEEE Transactions on Computers, 31*(3), 260–264.
22. Kogge, P. M., & Stone, H. S. (1973). A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Transactions on Computers, 22*(8), 786–793.
23. Han, T., & Carlson, D. A. (1987). Fast area-efficient VLSI subtractors. *Proceedings of the IEEE 8th Symposium on Computer Arithmetic*, 49–56.
24. Knowles, S. (2001). A family of adders. *Proceedings of the IEEE Symposium on Computer Arithmetic*, 277–281.
25. Zimmermann, R., & Fichtner, W. (1997). Low-power logic styles: CMOS versus pass-transistor logic. *IEEE Journal of Solid-State Circuits, 32*(7), 1079–1090.
26. Pedram, M. (1996). Power minimization in IC design: Principles and applications. *ACM Transactions on Design Automation of Electronic Systems, 1*(1), 3–56.
27. Benini, L., Bogliolo, A., & De Micheli, G. (2000). A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 8*(3), 299–316.
28. Rusu, S., Tam, S., Muljono, H., Ayers, D., & Chang, J. (2003). A 65-nm dual-core multithreaded Xeon processor with 16 MB L3 cache. *IEEE Journal of Solid-State Circuits, 42*(1), 17–25.
29. Flynn, M. J., & Oberman, S. F. (2001). *Advanced computer arithmetic design*. Wiley.
30. Hennessy, J. L., & Patterson, D. A. (2017). *Computer architecture: A quantitative approach* (6th ed.). Morgan Kaufmann.