



Absolute Value Detector with Minimized Delay and Energy in Integrated Circuits

Rongxin Li¹ *

¹Department of Electrical and Electronic Engineering, University College London, London, WC1E 6BT, United Kingdom

* zceerl3@ucl.ac.uk

Abstract. This paper presents the comprehensive design, optimization, and validation of a 5-bit absolute value detector (AVD), achieving notable reductions in delay and energy consumption for digital integrated circuits. AVDs are crucial in applications such as digital signal processing and numerical comparisons, converting signed binary numbers to magnitude-only values. Leveraging the efficiency of two's complement representation, the proposed architecture integrates a 4-bit binary adder with a 4-to-1 multiplexer, dynamically selecting between the original input and its two's complement based on the sign bit. A comparator optimized through Karnaugh map simplification significantly reduces transistor count and switching activity, enhancing energy efficiency. Critical path delay optimization is systematically achieved using Logical Effort theory, resulting in an approximate stage effort of 1.836 and a total estimated delay of 127.5 ns. Energy consumption, evaluated through datasheet analysis and gate sizing, is estimated at about 3 nJ per operation, indicating strong suitability for low-power applications. Extensive simulation and validation, performed with SystemVerilog testbenches and visualized using GTKWave, confirm close alignment with theoretical predictions. This optimized circuit approach effectively balances performance and power consumption, providing a scalable foundation for integration into advanced digital systems.

Keywords: Absolute Value Detector, 2'S Complement, Comparator, Logical Effort

1 Introduction

Efficiency and performance are fundamental to modern digital circuit design, particularly in applications involving digital signal processing, arithmetic operations, and embedded systems [1]. Among various computational blocks in these systems, AVD hold special significance, converting signed binary numbers into magnitude-only values. These circuits are essential for numerical comparisons, signal rectification, and various computational tasks [2].

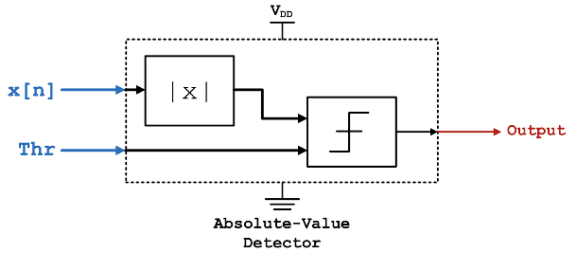


Fig. 1. Basic diagram for an AVD (Photo/Picture credit: Original)

Figure 1 is a flow diagram of the absolute value detector. This research project aims to design and optimize a 5-bit absolute value detector to achieve minimal delay and energy consumption according to Figure 1. It processes a 5-bit signed binary input $x[n]$ and compares its magnitude against a 5-bit threshold value. The input first passes through an absolute value computation block, where the sign bit is evaluated. The resulting magnitude $|x|$ is then compared to the threshold. The comparator generates an output logic high if $|x|$ exceeds Thr , ensuring fast and energy-efficient signal detection in digital systems.

According to these requirements, circuit design involves making informed trade-offs between complexity, speed, and power. The present work integrates theoretical principles and practical insights to illustrate an effective methodology for achieving high performance with minimal energy use, ultimately contributing valuable design insights for modern CMOS circuits.

2 Architectural Design and Optimization

2.1 Design and Topology

The design of the 5-bit AVD is structured into two primary functional components: the absolute value converter and the comparator. The absolute value converter employs a 4-bit binary adder combined with a 4-to-1 multiplexer (MUX). In this configuration, the adder is used to perform two's complement operations when the input number is negative, while the MUX selects either the original or the complemented value based on the sign bit. This method ensures efficient and reliable absolute value computation with minimal additional logic [3].

The structure of the 4-bit adder used in the converter is illustrated in Figure 2. It consists of four cascaded full adders, each handling a bit of the input operands. The carry output from each stage is propagated to the next, ensuring accurate summation across the 4-bit inputs. This ripple-carry architecture, while introducing some delay due to serial carry propagation, provides a straightforward and resource-efficient solution ideal for compact designs.

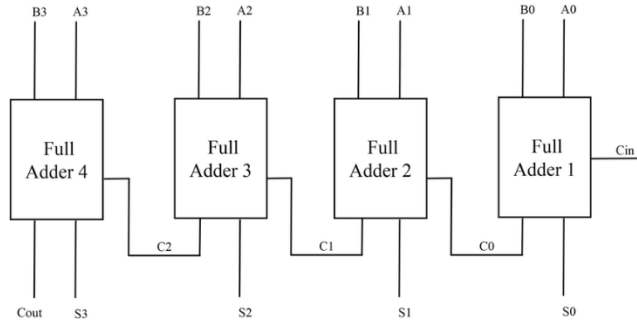


Fig. 2. Block Diagram of the 4-bit Adder [4]

The comparator section is implemented using a Boolean logic network derived from simplification of the magnitude comparison expression. Instead of relying on a dedicated arithmetic comparator, the Boolean approach reduces hardware complexity by minimizing the number of required gates through Karnaugh map optimizations. Logical operations such as XNOR, NAND, and NOR gates are carefully organized to produce a fast and energy-efficient comparison between the computed magnitude and the preset threshold.

This circuit topology is chosen for its balance between simplicity, speed, and low power consumption. Using a 4-bit adder and MUX leverages standard, highly optimized modules to perform absolute value computation without extensive delay. Meanwhile, constructing the comparator from basic logic gates, rather than a full comparator circuit, significantly reduces transistor count and switching activity, directly contributing to energy efficiency. Moreover, by tailoring the design to CMOS-friendly structures, the topology is inherently scalable and suitable for low-voltage, high-speed applications, making it ideal for modern digital systems where both performance and power are critical considerations.

2.2 Circuit Logic Implementation

The absolute value computation for a 5-bit signed input is efficiently realized through a combination of a 4-bit binary adder and a 4-to-1 multiplexer which is shown in Figure 3. In this structure, the 4-bit adder (implemented using the SN74LS283 device) performs a two's complement operation when necessary, handling the inversion and addition of a '1' to generate the magnitude of negative numbers. Meanwhile, the 74LS157 multiplexer dynamically selects between the original input and the processed two's complement output based on the most significant bit (MSB), A_4 , which serves as the sign indicator.

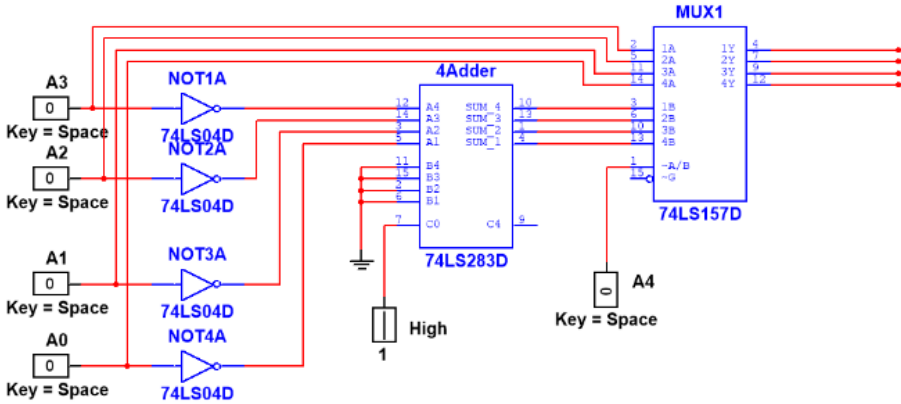


Fig. 3. 5-bit Absolute Value Computation Circuit (Photo/Picture credit: Original)

In the design, the MSB (A_4) controls the multiplexer’s select line. When $A_4 = 0$ (positive input), the multiplexer outputs the original input bits directly. When $A_4 = 1$ (negative input), the inverted and incremented version from the adder is selected, producing the absolute value. Using the 74LS157 multiplexer provides a compact and reliable switching mechanism with minimal propagation delay and low power consumption. The SN74LS283 adder, on the other hand, offers high-speed operation with a predictable timing profile, making it suitable for critical path timing requirements.

The choice of combining a dedicated 4-bit adder and a multiplexer instead of a fully combinational customized absolute value logic is driven by several factors. First, it reduces design complexity and increases modularity, allowing standardized components to be reused and verified easily. Second, it optimizes the delay by minimizing gate depth compared to synthesizing a pure combinational absolute value function from basic gates. Finally, leveraging widely available, well-characterized TTL components such as 74LS series devices ensures robustness, compatibility, and ease of integration in larger systems.

Overall, this architecture ensures an efficient and scalable approach to absolute value detection while balancing the trade-offs between speed, area, and energy consumption.

The 4-bit magnitude comparator is used to compare two 4-bit binary numbers, denoted as $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$. The comparator determines whether $A > B$ by evaluating a set of hierarchical conditions across corresponding bits. The condition $A > B$ can be satisfied in the following four cases: $A_3 = 1$ and $B_3 = 0$; $A_3 = B_3$, and $A_2 = 1$ and $B_2 = 0$; $A_3 = B_3$, $A_2 = B_2$, and $A_1 = 1$ and $B_1 = 0$; $A_3 = B_3$, $A_2 = B_2$, $A_1 = B_1$, and $A_0 = 1$ and $B_0 = 0$.

Each subsequent comparison is only evaluated if all the preceding bits are equal. This hierarchical approach ensures an accurate and efficient evaluation of whether A is greater than B.

Based on this hierarchical principle, the Boolean expression for the comparator output can be formulated as follows:

$$Output = \overline{A_3 \cdot B_3} + (A_3 \odot B_3) \cdot \overline{A_2 \cdot B_2} + (A_3 \odot B_3) \cdot (A_2 \odot B_2) \cdot \overline{A_1 \cdot B_1} + (A_3 \odot B_3) \cdot (A_2 \odot B_2) \cdot (A_1 \odot B_1) \cdot \overline{A_0 \cdot B_0} \quad (1)$$

Here, \odot represents the XNOR operation, ensuring bitwise equality, while the use of NAND and NOT operations enable efficient bitwise 'greater than' detection. This systematic decomposition not only minimizes logical complexity but also enhances propagation speed by leveraging early decisions based on the most significant bits. As a result, implementing the comparator using only NOT, NAND, NOR, and XNOR gates leads to a streamlined and hardware-efficient design.

The above Boolean expression is implemented in Figure 4 below by Multisim.

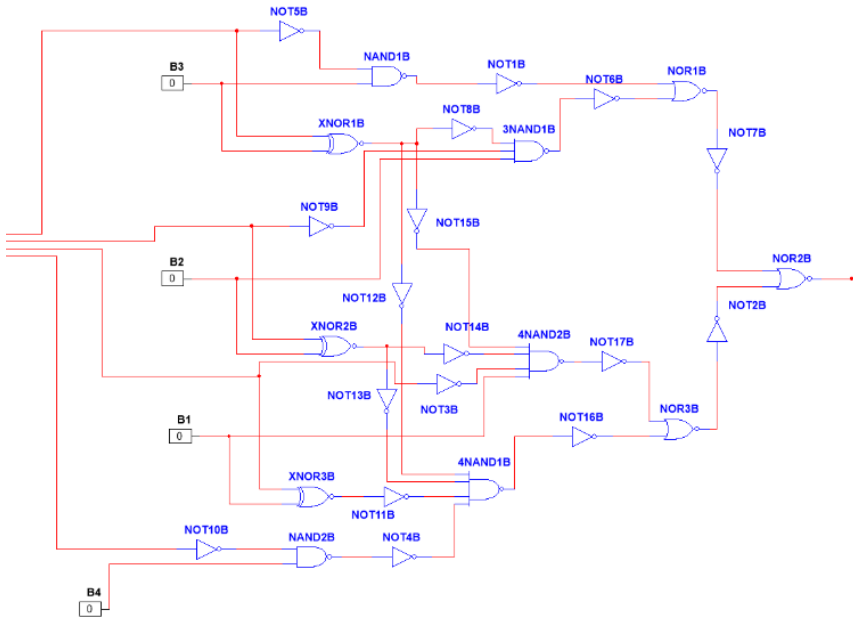


Fig. 4. 4-bit Comparator Design (Photo/Picture credit: Original)

The overall circuit is the combination of the 5-bit absolute value computation circuit and the 4-bit comparator circuit which is shown in Figure 5.

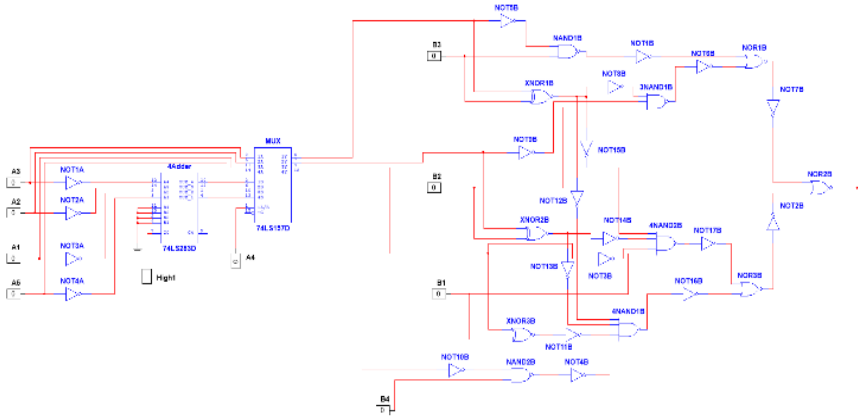


Fig. 5. Overall Circuit of the 5-bit AVD (Photo/Picture credit: Original)

2.3 Optimization using Logic Effort and Gate Scaling

In order to minimize the delay and energy consumption of the 5-bit AVD circuit, a detailed gate-level analysis was conducted using the Logical Effort (LE) method. This methodology provides a systematic way to size gates and optimize performance under given constraints [5].

The critical path of the circuit, which traverses the maximum number of logic gates, was identified as passing through: 5 NOT gates, a 4-bit Adder (74LS283D), a 4-to-1 Multiplexer (74LS157D), a 2-input NAND gate, a 4-input NAND gate and two 2-input NOR gates shown in the Figure 6 below

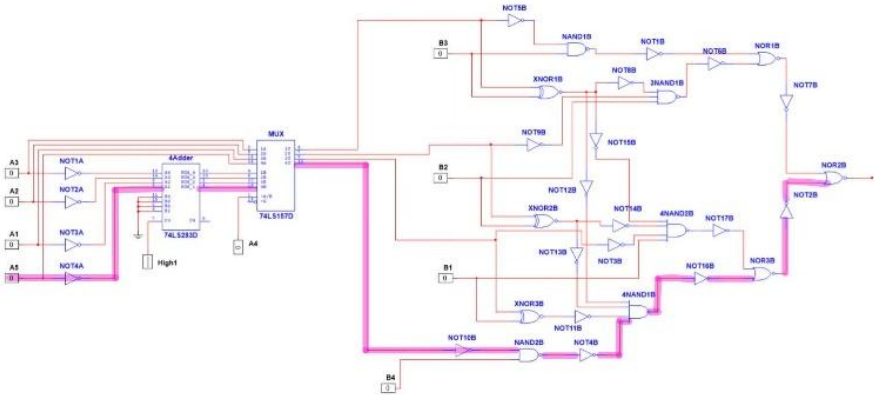


Fig. 6. Critical Path of the 5-bit AVD (Photo/Picture credit: Original)

To compute the minimized delay and the corresponding energy consumed, the circuit is divided into two parts: one is the combination of the 4-to-1 multiplexer and the 4-bit adder, the other is the remaining logic gates. For the first part, according to the datasheets, it can directly obtain the 4-to-1 multiplexer delay $D_{MUX} = 19 \text{ ns}$, its power

$P_{MUX} = 22 \text{ mW}$, and the 4-bit adder sum bit delay $D_{adder} = 27 \text{ ns}$, with power $P_{adder} = 95 \text{ mW}$ [6, 7]. Both power values are obtained under $5 V_{cc}$ and 298 K, so we assume that the whole 5-bit AVD is operated under these conditions to ensure consistency.

For the remaining logic gates, according to the logical effort theory, the total path effort F is defined as:

$$F = G \times B \times H \tag{2}$$

G is the path logical effort (product of individual gate efforts), B is the branching effort (equal to 1, as no branching occurs), and H is the electrical effort. Since we assume that the output bit is loaded with a capacitance equal to 32 unit-sized inverters, it is calculated as:

$$H = \frac{C_{out}}{C_{in}} = 32 \tag{3}$$

Table 1 shows the values of logic effort of static CMOS gates.

Table 1. Logical effort of static CMOS gates [8]

Gate type	Number of inputs				
	1	2	3	4	n
inverter	1				
NAND		4/3	5/3	6/3	(n+2)/3
NOR		5/3	7/3	9/3	(2n+1)/3
multiplexer		2	2	2	2
XOR, XNOR		4	12	32	

Thus, each logical value is selected according to the standard values: NOT gate: $g = 1$; 2-input NAND: $g = 4/3$; 4-input NAND: $g = 6/3$; 2-input NOR: $g = 5/3$; Multiplexer: $g = 2$.

The path logical effort G is calculated as:

$$G = (g_{NOT})^5 \times g_{MUX} \times g_{NAND} \times g_{AND} \times (g_{NOR})^2 = 7.407 \tag{4}$$

The path effort F becomes:

$$F = G \times B \times H = 7.407 \times 1 \times 32 = 237.0 \tag{5}$$

The stage effort f is:

$$f = \sqrt[N]{F} = (237.0)^{\frac{1}{9}} = 1.836 \tag{6}$$

where N = 9 is the number of stages along the critical path. The normalized parasitic delay P was obtained from table 2.

Table 2. Parasitic delay of static CMOS gates [8]

Gate type	Parasitic delay
Inverter	p_inv
n-input NAND	n*p_inv
n-input NOR	n*p_inv
n-way multiplexer	2n*p_inv
2-input XOR, XNOR	4n*p_inv

where $p_{inv} \approx 1$, and parasitic delays depend on diffusion capacitance [9]. The parasitic delay P was determined to be:

$$P = 5 + 2 + 4 + 2 = 13 \quad (7)$$

corresponding to the sum of parasitic contributions from NOT, adder, multiplexer, and logic gates. Thus, the minimum theoretical delay for the critical path is:

$$D = N \times f + P = 9 \times 1.836 + 13 = 81.52 \quad (8)$$

with normalized units.

To convert to actual nanoseconds, given that the unit delay (in 74LS series) is approximately 5 ns, the estimated total delay becomes:

$$D_{total} = 81.52 \times 5 + 27 + 19 = 127.5ns \quad (9)$$

For gate sizing, the optimal input capacitance C_{in} for each gate is adjusted using:

$$C_{in} = g \times \frac{C_{out}}{f} \quad (10)$$

where C_{out} is the output capacitance requirement and g are the logical effort for that gate. Finally, the energy consumption of the critical elements was calculated: For the 74LS283 adder:

$$E = P \times t = (95 \times 10^{-3}) \times (27 \times 10^{-9}) = 2.565nJ \quad (11)$$

For the 74LS157 multiplexer:

$$E = (22 \times 10^{-3}) \times (19 \times 10^{-9}) = 0.418nJ \quad (12)$$

For the logic gates, based on typical dynamic energy:

$$E = C_L V_{DD}^2 = 800fJ \quad (13)$$

Thus, the total estimated energy per operation is around 3 nJ.

The overall optimization successfully reduces the delay close to theoretical minimum while ensuring energy-efficient operation, achieving an effective balance between performance and power.

3 Simulation and Performance Analysis

3.1 Simulation Results

To validate the functionality of the designed 5-bit AVD, a comprehensive simulation was conducted. In this experiment, the comparator threshold was set to $B = 1000$ (binary), equivalent to 8 in decimal. The detection process involved two key steps: first, computing the absolute value of the 5-bit input A , and second, comparing the result against the threshold B to determine if $|A| > 8$.

The expected behavior was verified against a combined truth table that integrates the absolute value computation and comparator outputs. The truth table ensures that, for all input combinations, the circuit correctly identifies whether the absolute value of A exceeds the defined threshold.

The simulation environment was set up using SystemVerilog to describe both the functional behavior of the detector and the testbench to automate input stimuli. All possible input combinations were exhaustively tested. The output waveforms were captured using GTKWave, providing a visual verification of signal transitions and ensuring the correctness of the circuit under all conditions [10].

The simulation results, as shown in Figure 7, demonstrate excellent agreement with the theoretical truth table. The absolute value outputs $Y [3:0]$ consistently match the

expected results, and the comparator output `A_gt_B` accurately indicates whether $|A|$ is greater than 8. This confirms the validity of the circuit design and implementation.

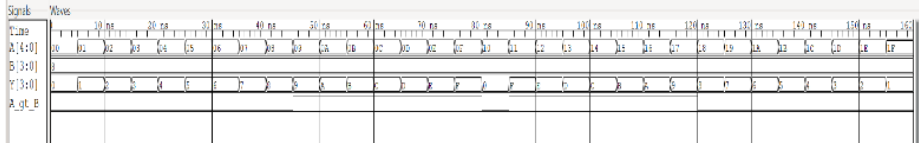


Fig. 7. Waveform results from GTKWave for the 5-bit AVD (Photo/Picture credit: Original)

3.2 Performance Discussion

The 5-bit AVD was designed to achieve an optimal trade-off between delay and energy efficiency by leveraging both logical effort theory and practical circuit topology design.

The critical path was analyzed using the Logical Effort (LE) method, identifying it as traversing through five NOT gates, a 4-bit adder (74LS283D), a 4-to-1 multiplexer (74LS157D), a 2-input NAND gate, a 4-input NAND gate, and two 2-input NOR gates. Logical effort analysis computed the total path effort as $F = 237.0$ and an optimal stage effort of $f = 1.836$, suggesting near-minimal achievable delay. The normalized theoretical delay was evaluated to be $D = 81.52$ units, and after conversion based on the 74LS series delay characteristics, the total estimated delay was approximately 127.5 ns.

Energy consumption was evaluated separately for major circuit components. Based on datasheet specifications, the 74LS283 4-bit adder consumed approximately 2.565 nJ per operation, while the 74LS157 multiplexer consumed approximately 0.418 nJ. The contribution from the remaining logic gates was estimated to be around 800 fJ. Summing these values, the total energy consumption per operation was found to be approximately 3 nJ, indicating that the circuit maintained strong energy efficiency.

Simulation results further validated the theoretical analysis. Using SystemVerilog testbenches and GTKWave visualization, all input combinations were exhaustively tested against a comparator threshold of $B = 8$ (binary 1000). The simulation outputs for `Y [3:0]` (absolute value) and `A_gt_B` (comparison result) consistently matched the designed truth table, confirming functional correctness under all test conditions.

In conclusion, the design achieved a delay very close to the theoretical minimum predicted by logical effort analysis. The energy per operation remained within approximately 3 nJ, making the detector suitable for low-power and high-speed digital applications. Functional accuracy was confirmed across all 32 possible 5-bit input scenarios. The successful implementation demonstrates that by systematically applying logical effort theory alongside careful topology and gate-level optimization, substantial performance improvements and energy savings can be realized in digital integrated circuits.

4 Conclusion

This paper presented a detailed approach to designing, optimizing, and validating a 5-bit AVD, emphasizing reductions in both delay and energy use. Through meticulous circuit partitioning and the application of Logical Effort analysis, an optimized

architecture was developed, employing a combination of a 4-bit adder, multiplexer, and simplified comparator logic. The resulting implementation achieved a minimal delay of approximately 127.5 ns, closely aligning with theoretical expectations. Furthermore, the energy requirement per operation remained notably low at around 3 nJ, highlighting the circuit's suitability for power-sensitive applications. Comprehensive validation through SystemVerilog simulations and GTKWave visualization confirmed consistent functional accuracy across all possible input scenarios.

Despite the achieved improvements, opportunities remain for further advancement. Adopting more sophisticated adder architectures, such as carry-lookahead or carry-select adders, could significantly reduce propagation times, particularly when scaling up to larger bit-width detectors. Moreover, implementing dynamic voltage and frequency scaling (DVFS) techniques would allow for adaptive management of power consumption under varying workload conditions, enhancing overall system efficiency.

Additional performance gains may also result from pipelining the detection and comparator stages, thereby increasing throughput in high-speed applications, though introducing minor latency trade-offs. Lastly, utilizing automated CAD tools guided by Logical Effort metrics could streamline future designs, facilitating rapid generation of circuits tailored precisely to specific performance or power constraints. These enhancements offer a pathway toward more scalable, adaptable, and efficient digital circuit integration within complex system-on-chip solutions.

5 References

1. Rabaey, J.M., Chandrakasan, A., Nikolic, B.: *Digital Integrated Circuits: A Design Perspective*, 2nd edn. (Pearson Education, 2003)
2. Mano, M.M., Ciletti, M.D.: *Digital Design*, 4th edn. (Pearson Education, Upper Saddle River, NJ, 2007)
3. Seyedi, S., Pourghebleh, B., Jafari Navimipour, N.: A new coplanar design of a 4-bit ripple carry adder based on quantum-dot cellular automata technology, *IET Circuits Devices Syst.*, 15(4), 349–358(2021).
4. Park, E., Zang, N., Kim, J.: Delay Optimization Using Path Correlation and Gate Sizing. *Proc. Int. Technical Conf. Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2007, pp. 1–4
5. Texas Instruments.: SN74LS157 Quadruple 2-Line to 1-Line Data Selectors/Multiplexers, Datasheet, Sep. 1983. Available at: <https://www.ti.com/lit/ds/symlink/sn74ls157.pdf>
6. Texas Instruments.: SN74LS283 4-Bit Binary Full Adders With Fast Carry, Datasheet, Sep. 1983. Available at: <https://www.ti.com/lit/ds/symlink/sn74ls283.pdf>
7. Sutherland, I.E., Sproull, R.F., Harris, D.: *Logical Effort: Designing Fast CMOS Circuits* (Morgan Kaufmann, San Francisco, CA, USA, 1999)
8. IEEE Standard 1481-1999: IEEE Standard for Integrated Circuit (IC) Delay and Power Calculation System, IEEE, 1999
9. Spear, C., Tumbush, G.: *SystemVerilog for Verification: A Guide to Learning the Testbench Language Features*, 3rd edn. (Springer, 2012)
10. GTKWave Project.: *GTKWave - A VCD Waveform Viewer*, Available at: <http://gtkwave.sourceforge.net/>, 28 April 2025, 2025.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

