



REVIEW

Machine Learning-Assisted Performance Prediction and Hardware Security Analysis of Digital Adder Architectures: A Comprehensive Survey and Future Directions

F. M. Karlyn Cynthia¹, Akshat Gaurav², Varsha Arya^{3,4}, Razaz Waheeb Attar⁵, Kwok Tai Chui³ and Brij B. Gupta^{2,6,7,8,*}

¹Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Tiruchirappalli, India

²Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

³Electronic Engineering and Computer Science, Hong Kong Metropolitan University, Hong Kong

⁴Center for Interdisciplinary Research, University of Petroleum and Energy Studies (UPES), Dehradun, India

⁵Management Department, College of Business Administration, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

⁶Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan

⁷Symbiosis Centre for Information Technology (SCIT), Symbiosis International University, Pune, India

⁸School of Cybersecurity, Korea University, Seoul, Republic of Korea

*Corresponding Author: Brij B. Gupta. Email: bbgupta@asia.edu.tw

Received: 08 June 2026; Accepted: 17 September 2026; Published: 21 October 2026

ABSTRACT: The increasing complexity of modern Very Large-Scale Integration (VLSI) systems has intensified the demand for digital adder architectures that simultaneously achieve high performance, low power consumption, efficient area utilization, and strong security guarantees. Conventional adder design methodologies rely heavily on exhaustive simulation and iterative optimization processes, which become increasingly time-consuming and computationally expensive as circuit complexity grows. Although machine learning (ML) has recently emerged as a promising solution for rapid performance estimation and design-space exploration, existing review studies primarily focus on performance optimization while providing limited coverage of hardware security concerns and their integration with ML-based methodologies. To address this gap, this survey presents a comprehensive review of ML-driven approaches for digital adder design, covering performance prediction of delay, power, and area, as well as security-oriented applications related to hardware Trojan detection, side-channel attack mitigation, and trustworthy hardware design. The surveyed literature is systematically categorized according to learning paradigms, adder architectures, prediction objectives, attack models, and defense mechanisms. Furthermore, the paper analyzes current challenges including dataset limitations, model generalizability, explainability, security-performance trade-offs, and deployment complexity. Key open challenges include limited adder-specific datasets, weak cross-technology-node generalization, explainability of black-box predictors, security-performance trade-offs, and the cost of collecting labeled side-channel or Trojan data. Finally, emerging research directions—including explainable AI, graph-based netlist learning, LLM-assisted RTL exploration, automated electronic design automation (EDA), transfer learning, and secure ML-assisted hardware design—are discussed to guide future developments in intelligent and secure digital arithmetic systems.

KEYWORDS: Adder architectures; hardware-level security; machine learning; power analysis modeling; side-channel threats

1 Introduction

Continuous evolution of semiconductor fabrication allows incorporation of billions of transistors within a single chip, giving rise to the capability and complexity of efficient digital systems [1–5]. Digital systems form the core of these arithmetic units, where adders are required for handling operations such as arithmetic computations and signal processing tasks [6–9]. The performance efficiency of digital adders affects the speed of the processor, power usage, and chip utilization. This reduces the propagation delay, power, and area parameters, which are primary concerns in modern VLSI implementations [2,10–13]. In order to satisfy these requirements, different adder architectures have been introduced. Though the Ripple Carry Adder (RCA) stands out for its simple structure and reduced hardware utilization, it suffers from high propagation delay as each carry must wait to propagate through all the progressing stages [14–18]. To overcome this limitation, high-performance adder designs like Carry Lookahead Adders (CLA), Carry Select Adders (CSLA), and Parallel Prefix Adders, including Kogge-Stone adders and Brent-Kung adders, are utilized through parallel carry generation to improve their performance, but they need more hardware resources [19–23]. This illustrates the balance between achieving high speed and managing the structural complexity in adder architectures [24–28].

Existing design strategies for digital adders rely on analytical techniques, and they are often evaluated through simulation for performance [4,5,29–31]. Although these techniques provide accurate results, they become less efficient as the complexity of the circuit increases. This results in increased design time and computational requirements [32–34]. As a solution to this challenge, machine learning methods have been developed as efficient alternatives by enabling quick prediction of performance parameters like power, delay, and area [32,35–38]. Tools like regression models, support vector machines, ensemble learning approaches, and neural networks show high proficiency in capturing complex circuit operations and improving design space analysis [32,35–37]. At the same time, hardware security has led to an increase in global supply chains and third-party components, which is developing as a major issue [39–41]. Modern digital systems can be vulnerable to security threats, including side-channel attacks and hardware Trojans, which have the capability of degrading the system functionality and exposing sensitive information. To reduce and safeguard against these threats, strategies like masking, hiding, and logic locking are implemented. Such strategies can result in extra design complexity and operational cost [42–46]. In recent years, studies show that machine learning algorithms are enhancing security by analysing the behaviour of the circuit and identifying the deviations and hidden vulnerabilities in circuit functionality [47–51].

The collaboration of machine learning techniques, performance evaluation and hardware security delivers a comprehensive approach for next-generation VLSI development [52–56]. By combining prediction techniques with statistical power analysis and security mechanisms, optimized performance can be achieved [35,37,48,57,58]. This maintains system reliability and stability [59].

This survey presents a detailed analysis of machine learning approaches for predicting the performance and hardware security considerations for digital adder architectural systems. Although supporting background is drawn from broader VLSI, electronic design automation (EDA), and hardware-security literature, the organizing focus remains digital adder architectures (e.g., ripple-carry, carry-lookahead, carry-select, and parallel-prefix topologies). Broader VLSI material is retained only when it supplies transferable methods, datasets, or threat models that inform adder-centric performance prediction and security analysis. Reported operating frequencies, operand widths, and technology nodes vary widely across the surveyed sources; therefore, cross-study numerical comparisons in this review are interpretive rather than a single unified benchmark. The primary contributions include:

- An adder-centered synthesis of ML techniques for predicting propagation delay, power consumption, and area, together with the structural features that differentiate adder topologies
- A structured taxonomy of hardware attacks and defense mechanisms with explicit links to arithmetic/carry-path vulnerabilities
- A classification of existing design methodologies into conventional, ML-based, and security-aware methods, including critical discussion of limitations and research gaps
- Comparative analysis of approaches with clearly labeled qualitative indicators and conceptual trade-off illustrations
- An overview of open challenges mapped to corresponding future research directions

Fig. 1 shows the overall architectural workflow of a machine learning-based adder design, which highlights the process that begins with selecting adder designs and feature extraction, followed by predicting performance and security evaluation in order to achieve optimized results. Although purely “ML-for-adder-only” papers remain fewer than general ML-EDA studies, adders are ubiquitous, security-sensitive arithmetic kernels; a focused survey is therefore useful to consolidate transferable methods and to expose adder-specific gaps [12,52].

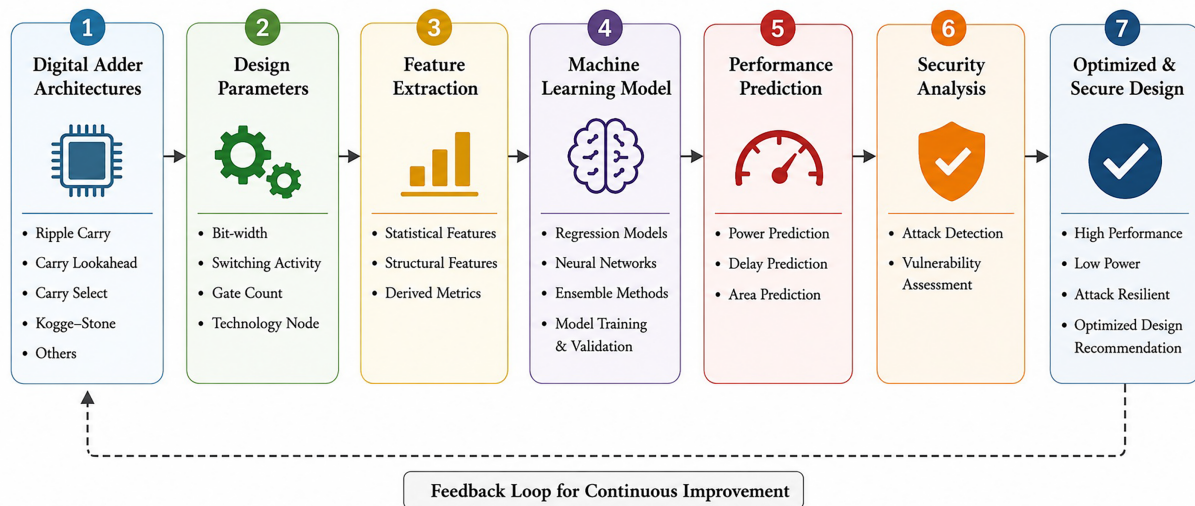


Figure 1: Conceptual overview of ML-assisted digital adder design and security analysis.

Fig. 2 provides an illustrative summary of the rising activity in machine learning-based VLSI design and hardware security. The underlying literature screening emphasized peer-reviewed sources indexed in IEEE Xplore, ScienceDirect, and SpringerLink for the approximate period 2015–2024; Google Scholar was used only as a supplementary discovery tool (not as an independent bibliographic database) to locate additional candidate works that were then traced to primary publishers.

2 Core Concepts and Preliminaries

The development and evaluation of advanced digital systems requirements are integrated with the understanding of arithmetic circuit structures, machine learning techniques, power analysis, and security-aware hardware design strategies [60–64]. Traditional design methods often struggle to keep up with the increasing complexity of integrated circuits, and in order to address the modern necessities for better performance, lower utilization of power and enhanced secure system operation [65–68]. This section

summarizes only the essential background on digital adder families, ML prediction basics, power modelling, and hardware-security foundations [69–72]. Detailed methodological comparisons are deferred to the taxonomy and comparative-analysis sections to reduce overlap.

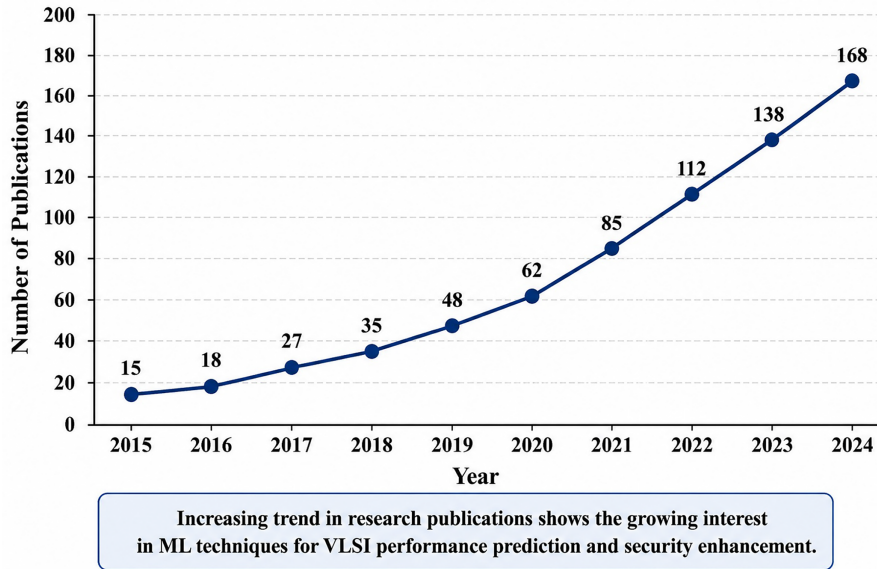


Figure 2: Illustrative growth trend of research in ML-based VLSI design and hardware security (conceptual summary of publication activity).

2.1 Digital Adder Architectures

In modern electronic architectures, digital adders act as the key computational units that are widely used in arithmetic logic units (ALUs), microprocessors, and digital signal processing applications [73–76]. As addition happens frequently in computations, this circuit’s performance directly influences the computational speed, efficiency of power, and hardware resources. Thus, optimizing delay, power consumption, and minimal area is essential in VLSI system design [77–79]. The Ripple Carry Adder (RCA) is valued for its simple architecture, easy to implement, and has low hardware cost. However, its delay increases proportionally with the number of bits because of the sequential carry propagation, making it unsuitable for high-performance systems [80–83]. In order to address this limitation, the Carry Lookahead Adder (CLA) improves its performance by precomputing carry outputs through generate and propagate mechanisms. This leads to the improvement of the overall circuit complexity [84–87].

The Carry Select Adder (CSLA) generally improves computational speed by precomputing candidate sums in parallel and selecting the final result with multiplexers; this speed gain typically comes at the cost of additional area (and often power), so “area efficiency” is implementation-dependent rather than universal [88–90]. High-performance Parallel Prefix Adders such as Kogge-Stone and Brent-Kung adders enhance their computational speed by utilizing a tree-like structure for an efficient carry computation mechanism [91]. While the Kogge-Stone adder is optimized for minimum delay but comes with higher area and design complexity, the Brent-Kung adder achieves reduced usage of area with a small delay increase [21,92]. The presence of these trade-offs highlights the intelligent design methodologies, where machine learning helps to evaluate system performance and select the most suitable architectures [32,35].

From a machine-learning perspective, structural differences among adder families can be encoded as features such as bit width, logic depth of the carry network, maximum fan-out, number of generate/propagate

cells, estimated wire complexity, and switching-activity proxies. Linear carry chains (e.g., RCA) emphasize long sequential dependence, whereas prefix trees (e.g., Kogge–Stone and Brent–Kung) emphasize parallel prefix depth and wiring congestion; these distinctions help models relate topology to delay, power, and area predictions without requiring a full gate-level re-simulation for every candidate [12,22,35].

2.2 Machine Learning in VLSI

In modern VLSI design, as system complexity grows, machine learning offers an efficient alternative to traditional simulation methods, particularly for handling complex systems where the simulation costs are very high [93]. ML approaches analyse the use of past design data in order to predict important performance measures including power, delay, and area, which improves computational efficiency [54,60]. In VLSI applications, supervised learning methods are enhanced by using structured datasets that provide relationships between the inputs and outputs [94]. Simple prediction problems can be handled by using linear regression, whereas complex analytical approaches require advanced algorithms such as support vector machines (SVM) and ensemble models such as random forests are employed for complex regression and classification tasks [35–37].

Circuit performance prediction can be effectively predicted by these models using factors such as bit-width, switching activity, and gate count [69]. Advancements in deep learning approaches have significantly improved the modelling of nonlinear circuit behaviour, resulting in better accuracy in timing and power analysis [59,95]. The adoption of machine learning is increasingly applied at various stages of electronic design automation (EDA) phases, including synthesis, routing, and placement, to speed up optimization and enhance the evaluation of diverse design configurations [96–98]. For digital adders specifically, open issues include circuit-specific feature engineering (RTL vs. gate-level views), graph or netlist representations of prefix networks [99,100], technology-node transferability of trained predictors, construction of balanced datasets across bit widths, and quantification of prediction uncertainty under process–voltage–temperature variation [33,34]. Backend physical effects such as routing congestion and IR drop can further degrade the fidelity of early ML estimates if training features omit post-route information [33,53].

Table 1 is a qualitative synthesis intended to compare model families for VLSI-oriented prediction tasks; the “Accuracy” column uses ordinal labels rather than re-computed metrics (e.g., RMSE/ R^2 for regression or F1/precision/recall for classification). Deep learning and neural models are typically reported as stronger on nonlinear circuit behaviour, whereas linear regression remains a simpler baseline with more limited capacity.

Table 1: Comparison of ML models for VLSI prediction (qualitative survey synthesis). Ordinal labels: Low < Medium < High < Very High; Partial denotes limited or task-dependent coverage rather than a measured percentage.

Reference	Model	Delay Prediction	Power Prediction	Area Prediction	Accuracy	Training Complexity	Scalability	Suitability for VLSI
[101]	Linear Regression	Partial	Partial	×	Medium	Low	High	Basic Designs
[51]	SVM	✓	Partial	Partial	High	Medium	Medium	Classification Tasks
[102]	Random Forest	✓	✓	Partial	High	Medium	High	Power Estimation
[48]	Neural Networks	✓	✓	✓	Very High	High	High	Complex VLSI Designs
[49]	Deep Learning	✓	✓	✓	Very High	Very High	High	Advanced Prediction

Fig. 3 outlines the architecture of a machine learning approach for estimating performance and incorporating processes like feature extraction, preprocessing, choice of the model, and evaluation in order to assess power, delay, and area.

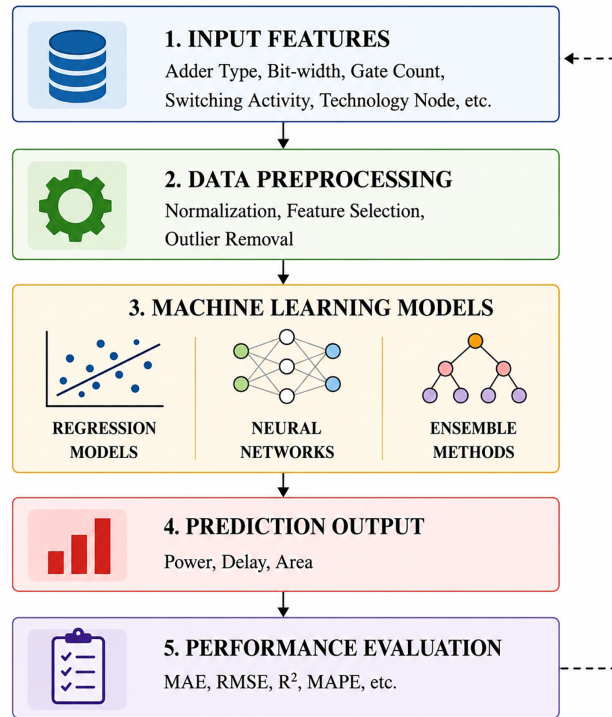


Figure 3: ML-based prediction of performance (feature extraction to evaluation). Emerging LLM-assisted RTL/design exploration can feed the same pipeline as an upstream generator of candidates or features [103].

2.3 Power Modelling Techniques

In modern VLSI systems, minimizing the usage of power is a major design constraint, particularly in battery-driven and low-energy applications [104,105]. As transistor size continues to become smaller, it leads to higher dynamic power due to frequent switching activity and higher static power, which requires reliable power modelling techniques [106,107]. Earlier power estimation techniques concentrated on switching activity patterns, which utilized approaches such as transition density calculations and probabilistic modelling techniques. Even though these approaches are simple and resource-efficient, they may often fail to accurately capture the model's complex behaviour with advanced circuit designs [108–111].

Statistical power modelling enhances the overall prediction accuracy by observing variability and probabilistic behaviour within the system [57,58,112]. Recently, machine learning approaches have been used to analyse, understand, and to predict the power from available circuit design features. This model learns the relationship between circuit parameters and power consumption based on training datasets [60,61]. The use of regression techniques, along with neural networks and ensemble learning approaches, improves the accuracy of prediction and ensures efficient power modelling for large-scale VLSI circuits [35,37].

2.4 Hardware Security Fundamentals

As the increasing use of integrated circuits in sensitive applications has become more critical, the importance of hardware security in VLSI systems has significantly increased [113,114]. Hardware-level

attacks frequently target the physical properties or design-stage weaknesses that are introduced during the time of digital circuit design process, resulting in exposure of data or performance efficiency loss. Wide research on side-channel attacks highlights the advantage of how physical leakages such as consumption of power, timing behaviour, and electromagnetic signals can be used to analyse and to extract confidential data [101,115–117]. Approaches like Simple Power Analysis (SPA) and Differential Power Analysis (DPA) analyse power signature usage to extract secret or sensitive information, especially in secure hardware environments [102,118,119].

In addition to that, Hardware Trojans are another major threat to hardware security, as they involve hiding malicious alterations at the time of design or fabrication stages, staying undetected under standard operating conditions but capable of triggering harmful effects upon activation [120,121]. In fault injection-based attacks, it allows attackers to manipulate the circuit operations by using disturbances like voltage variations or electromagnetic pulses, which results in faulty outputs that may expose the hidden information [122–124]. In order to enhance the security of the system, techniques such as masking, logic locking, and hiding are implemented to safeguard against these threats [125–127]. The integration of machine learning approaches facilitates the system to recognize unusual behaviour and strengthen security in a flexible and reliable manner [49].

3 Literature Review

Literature Selection Approach

This survey follows a narrative (scoping) review approach rather than a full PRISMA systematic review with prospectively registered protocols. Candidate works were identified primarily through IEEE Xplore, ScienceDirect, and SpringerLink, with Google Scholar used only for supplementary discovery of titles that were subsequently verified against publisher records. Inclusion emphasized (i) digital adder or closely related arithmetic-unit architectures, (ii) machine-learning methods for delay/power/area prediction or design-space exploration, and/or (iii) hardware-security topics (side-channel leakage, hardware Trojans, fault injection, and defenses) with relevance to arithmetic datapaths. Exclusion emphasized non-hardware ML applications without a clear VLSI/security transfer argument and duplicate secondary summaries lacking primary technical content. Exact query-hit counts and a PRISMA flow diagram are not claimed here because a formal multi-stage screening log was not retained for this revision; instead, the taxonomy and tables organize the retained literature thematically around adder-relevant ML and security questions.

In VLSI design, digital adder systems remain an important area of research [1,2,6,24]. Due to its key role in arithmetic processing tasks, initial studies are focused on conventional structures such as Ripple Carry Adder (RCA), Carry Lookahead Adder (CLA) and Parallel Prefix Adders (PPA) [14,15,22,128,129]. The design architectures which are based on parallel prefix techniques including Kogge-Stone and Brent-Kung adders improves their speed by minimizing delay, but they need higher implementation cost due to their design space [12,77,92]. Hybrid adder architectures were introduced to provide an effective balance between performance, efficiency of power, and hardware resource utilization [130–133]. As the complexity of the circuit grows, it minimizes the flexibility of the traditional simulation and analytical methods, which results in higher design time and computational expenses [5,29,30,34]. In order to improve the prediction accuracy, machine learning techniques that are based on predictive models have been used for efficient circuit performance analysis. Algorithms which include linear regression, support vector machines and neural networks are usually applied to model the parameters of the circuit from the given input features, minimizes the requirements for simulations, thereby speeding up the efficiency of the design [32,35–37,94]. Accordingly, they support faster and efficient design space analysis, particularly during the early development stages [52,53,134,135].

The importance of power analysis has grown significantly and plays a critical role in VLSI design, particularly for applications that need less amount of power consumption [2,10,11,58]. Existing approaches that depend based on the switching behaviour and probabilistic modelling, including transition density-based simulation, suffer from increased complexity of the circuits [13,57,112,136]. In order to achieve better accuracy in predictions, statistical power consumption models were used and enhanced for fabrication variability and in real-time conditions [13,57,58,112,136]. In recent years, the integration of neural networks and ensemble learning algorithms which has optimized the accuracy of predicting power and also makes them as an ideal application for complex VLSI circuit designs [35–38]. The use of machine learning approaches is applied to predict delay and area which offers a complete view of power, performance and area (PPA) prediction system [29,32,33]. By depending on RTL-level representations instead of complete simulations and synthesis, these techniques optimize the speed of the design and improve the design efficiency [34,55,96,103].

Machine learning techniques are increasingly applied in multiple stages of the electronic design automation (EDA) process, which include placement, routing, and synthesis in order to promote better scalability and efficient direction of design space alternatives [34,52,55,99,100]. In today's VLSI design architectures, improving performance with hardware security has become an important part of the performance optimization of the system [39–41]. Electronic circuits face security risks and are vulnerable to side-channel threats that use physical information leakage as well as hardware Trojans that may be added at the time of development or fabrication [39,44,45,137,138]. Multiple design methods such as masking, hiding, and logic locking are implemented in order to overcome these threats; these solutions frequently result in higher consumption of power, improved chip area, and minimized operating speed [41,42,90,139,140]. Over the years, data-driven security methods have gained significant attention as they effectively analyse circuit operations and enhance the overall prediction of the system [47–49,141]. Recent trends show that the direction of research indicates a transition from conventional methods to data-driven approaches with power modelling in order to improve efficiency and reliability of the performance and secure digital design of circuits [35,142,143].

Table 2 summarizes a structured comparison of existing survey works, focusing on their main themes, utilization of ML models, security focus, and limitations. Checkmarks indicate topical coverage rather than a claim of exhaustive superiority; the novelty of the present survey is the joint, adder-centered treatment of ML-based PPA prediction and hardware-security analysis, together with an explicit mapping from open challenges to future directions.

Table 2: Overview of existing surveys.

Reference	Year	ML-Based Prediction	Power Analysis	Hardware Security	Side-Channel Attacks	Hardware Trojans	Defence Mechanisms	Hybrid ML + Security	Research Gaps Discussed
[53]	2022	✓	✓	×	×	×	×	×	Partial
[52]	2024	✓	✓	Partial	×	×	×	×	×
[35]	2021	✓	×	Partial	×	×	×	×	×
[40]	2023	✓	×	✓	✓	Partial	✓	×	Partial
[59]	2020	Partial	✓	✓	✓	×	Partial	×	×
This Survey	2026	✓	✓	✓	✓	✓	✓	✓	✓

4 Taxonomy of Attacks

This paper pursues a dual focus: (i) ML-assisted performance prediction for digital adders and (ii) hardware-security analysis of arithmetic datapaths. The attack taxonomy below supports the second focus while remaining connected to adder carry/sum logic that also drives PPA behaviour. A well-structured classification of hardware security threats is required for analysing the validation of system weakness which may leak sensitive information in integrated circuits. With the development in digital systems and their integration in security and high-dependency domains, the increasing risk of threats from the combined use of physical dynamics with structural vulnerabilities has considerably strengthened the results of the system performance [39,40]. Such vulnerabilities can cause data leakage, manipulation of functionality, and overall system-level failures. Hence, proper classification of the type of threats supports the development of carrying out efficient defense mechanisms and safeguarding digital adder architecture and VLSI systems [42]. Fig. 4 illustrates the classification of hardware attacks, dividing them into categories such as side-channel attacks, hardware Trojans, fault injection techniques, and machine learning-based threats, along with their associated strategies.

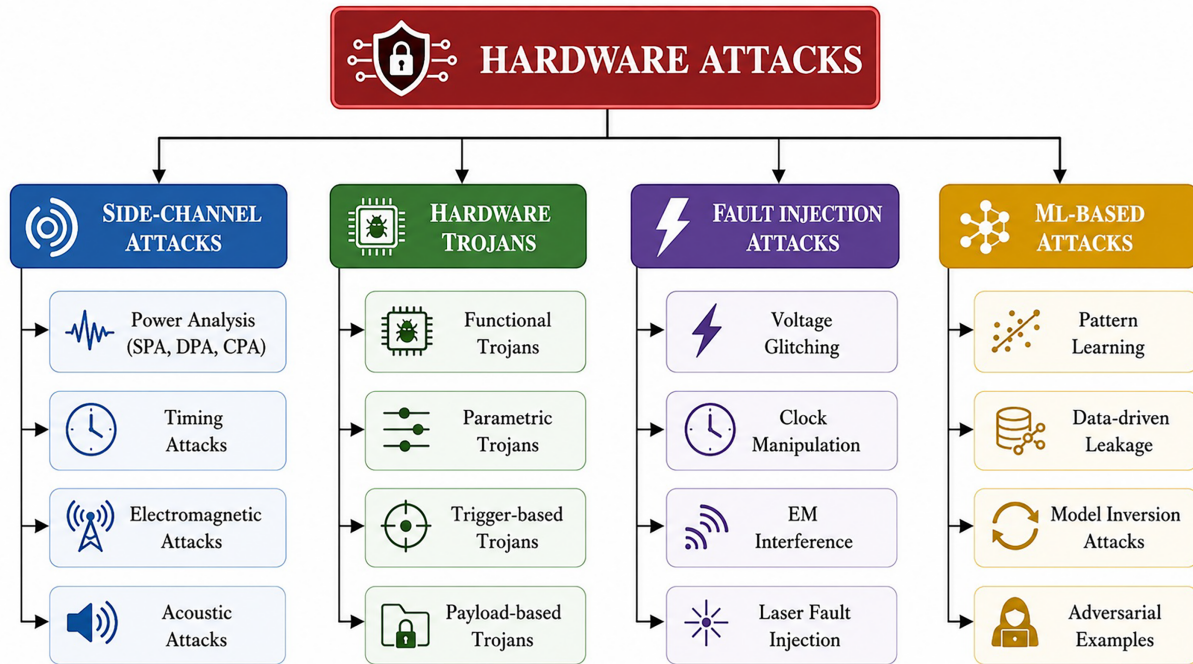


Figure 4: Taxonomy of hardware attacks.

4.1 Side-Channel Attacks

In hardware security systems, side-channel attacks are considered as a serious threat where attackers take advantage of leaking sensitive information from side-channel attacks in the physical characteristics of the behaviour of the circuit instead of attacking or directly targeting the logic of the algorithmic design [115,137]. This kind of leakage might result from variations in the utilization of power, timing responses, electromagnetic signals, or outputs and similar physical effects at the time of circuit operation. As these attacks do not completely rely on direct interaction with internal data, such threats are highly suitable for carrying out and continue to remain difficult in order to recognize in real-world scenarios [101,117]. Power analysis threats remain as a primary attack mechanism with simple analysis of power, which tries

to understand direct observation of capturing power patterns, while Differential Power Analysis (DPA) tends to apply statistical processing approaches in order to extract sensitive information, including private keys [102,144].

The switching behaviour of the circuit in digital adder architectures differs with input data, which makes them vulnerable to side-channel leakage. The variation that happens in the input data can lead to multiple logic transitions, resulting in variations of power that might direct the attackers to analyse and expose the sensitive information and also the hidden internal values [57,72]. Even minute power fluctuations can be observed using the statistical tools in order to extract secure data, thus making adders become the primary target for security threats [112,145]. The use of machine learning concepts has improved these threats by enabling efficient pattern identification from large datasets, which improves the system accuracy [48,141]. As a result, safeguarding VLSI design circuits against side-channel attacks is still a major concern [138,143].

Table 3 highlights the comparison of various side-channel attacks by analysing their leakage patterns, affected parameters, and impact on hardware protection. Approaches based on electromagnetic leakage and power analysis are used to identify hidden information from VLSI circuits and encryption devices.

Table 3: Comparison of side-channel attacks.

Reference	Attack Type	Leakage Source	Target Metric	Impact
[102,137]	SPA	Power traces	Secret key	Information leakage
[122,145]	DPA	Statistical power variation	Cryptographic data	Key extraction
[124,146]	Timing Attack	Execution timing	Computation delay	Data exposure
[101,147]	EM Attack	Electromagnetic radiation	Internal switching	Signal leakage
[48,49]	ML-assisted Side-Channel Attack	Power and timing patterns	Hidden circuit behavior	Improved attack accuracy

4.2 Hardware Trojan Attacks

The presence of hardware Trojans is a serious threat to maintaining the accuracy and dependability of integrated circuit performance. This type of threat arises from the integration of malicious or harmful components during the design or fabrication phases, which can result in the operation of circuits, minimizing the performance of the system or loss of efficiency and leakage of sensitive information [39,40]. Risk levels have been increased due to the fabrication becoming distributed globally, and it is mostly reliant on external components [41]. Detection of hardware Trojans are very difficult to expose as they are being designed in order to remain inactive at the time of functioning of the system and being active only under rare inputs or environmental factors [39,50]. Once it is activated, they have the capacity to degrade the system functionality or system behaviour and result in sensitive data being exposed [46].

As essential arithmetic building blocks, digital adders are attractive Trojan targets at RTL, gate-level, and layout stages [39,41]. Because topologies differ in carry organization, the same Trojan payload can have topology-dependent effects: in an RCA, a stealthy modification along the serial carry chain can corrupt many sum bits with limited local area cost; in prefix adders, tampering with generate/propagate or sparse wiring nodes can create rare-trigger timing/power anomalies that are harder to localize; in CSLA structures, duplicated sum paths and select multiplexers enlarge the attack surface for selective corruption under

particular carry assumptions [22,98,148]. ML detectors commonly rely on power, timing, or structural features to flag such anomalies [47,48]; class imbalance (few Trojanized samples) remains a practical obstacle and motivates careful sampling, synthetic augmentation, or anomaly-detection formulations [142,143].

4.3 Fault Injection Attacks

Fault injection attacks refer to a category of hardware-based threats that involve the manipulation of the behaviour of circuits by injecting faults that are implemented into a system's hardware to expose weaknesses and affect its operational security [149,150]. Analysing the system's behaviour in the presence of faults that are being triggered can be utilized to leak the secret or hidden information or degrade the security mechanisms [149,150]. Common fault injection approaches include voltage glitching, manipulation of clock signals, electromagnetic interference, and laser injection faults are often used to introduce faults into circuits. Such breakdowns can lead to incorrect or abnormal outputs, which helps to expose hidden data or system vulnerabilities [149,151].

The presence of fault injection can affect the performance of digital systems, where even small inaccuracies in sum or carry generation tend to influence the overall system output [149,150]. In a cryptographic environment, such faults can be utilized for key leakage or to degrade encryption security, thus increasing the potential impact associated with these attacks [149,150]. Common techniques and broadly used solutions include redundant configurations, error verification, and fault-tolerant architectures [152,153]. Recent studies analysed the use of machine learning approaches that are applied in detecting anomalies, but still a certain number of challenges need to be resolved [154,155].

5 Taxonomy of Defense Mechanisms

Defence security mechanisms in hardware are designed in order to safeguard the digital circuits by reducing data leakage and blocking improper interference. As attack methodologies increase due to their complexity, conventional approaches struggle to provide sufficient protection measures [42,140]. A systematic classification of defense mechanisms supports the evaluation of their capability and performance in order to overcome threats like side-channel attacks, hardware Trojans, and fault injection [43,46]. Fig. 5 provides different hardware security mechanisms such as masking, hiding, logic locking, and AI-based strategies that contribute to enhanced system security and protection against potential threats.

5.1 Masking Techniques

Masking is a well-known and operates as an effective hardware security mechanism to secure confidential data and ensures protection by introducing randomness into the computation process. It minimizes the reliability with the relationship between actual data and observable signals like power consumption, which limits the capability to obtain useful information and makes it more challenging [137,144]. This approach is carried out by splitting the data into multiple randomized shares, thereby preventing a single share from revealing the original information [44,139]. Masking techniques can be implemented at different levels of digital adders in order to reduce the leakage patterns, which enhances the defence against threats such as differential power analysis [72,102].

Masking is one of the efficient mechanisms, but adds hardware complexity and leads to higher power utilization and chip area because of the need for random value generation [10,53]. If the system is not designed or implemented carefully, it may result in failure of side-channel leakage and minimize their reliability, thereby limiting the effectiveness in secure systems [72,102]. Current research aims to improve the masking efficiency and incorporate it with machine learning algorithms in order to achieve the balance between protection and system performance [47,142].

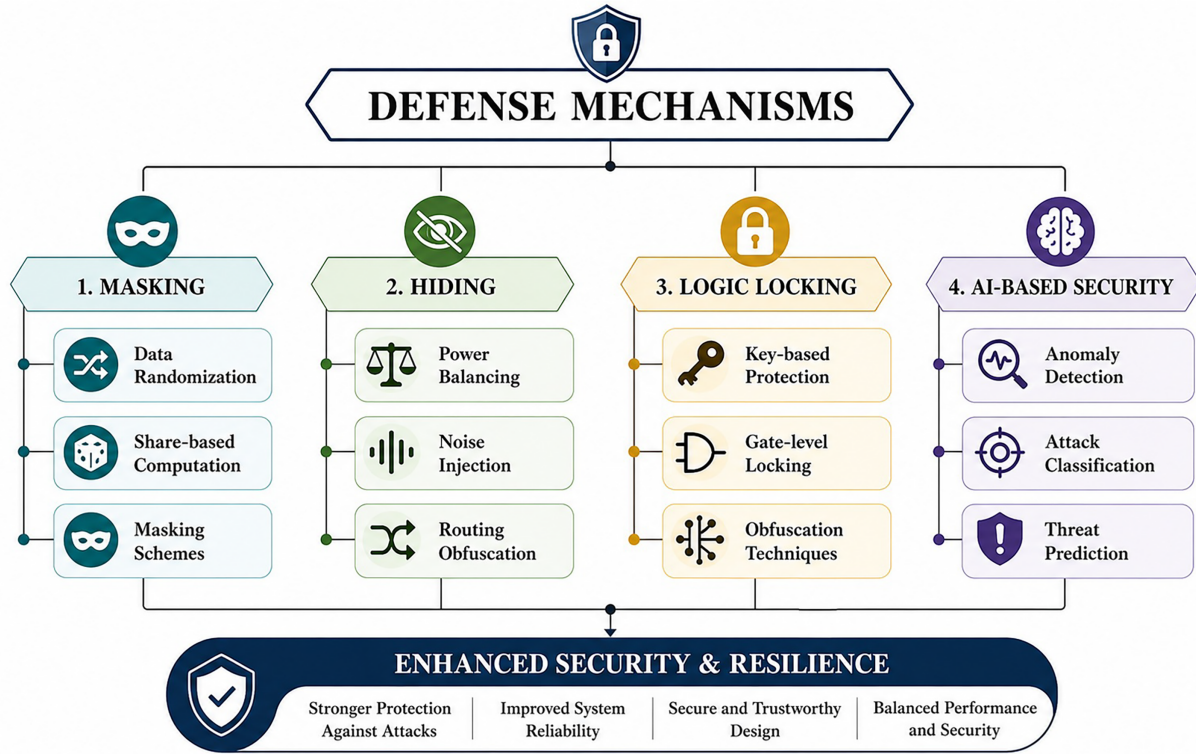


Figure 5: Classification of defense mechanisms against hardware attacks.

5.2 Hiding Techniques

The main objective of hiding techniques is to limit the information leakage by minimizing the power fluctuations and execution timing during the operation of circuits [126,137]. Instead of changing the data, their goal is to make the leakage independent of the input values. Such outcomes are accomplished by using common techniques like power balancing, noise injection, and constant-time execution [139,140]. These solutions help to reduce the risk of attackers that depend on timing or power variations at the time of execution [72,112].

The technique of hiding in digital adders highly depends on maintaining consistent switching behaviour, achieved by using dual-rail logic that balance the operation of circuit transitions [139,140]. However, ensuring perfect consistency is difficult to attain due to process-induced variations and real-world influences [112,145]. These approaches may lead to higher complexity in circuits and can reduce the performance efficiency of the system [10,53]. As a result, it is important for the designers to balance between the improved security effectiveness and the performance efficiency when applying the technique of hiding [126,137].

5.3 Logic Locking Techniques

Logic locking refers to a hardware-level protection method that is used to safeguard integrated circuits that limit illegal interaction and data manipulation. Additional logic is integrated into the design so that the circuit can perform accurately only with the correct secret key [39,41]. An invalid key may result in the circuit generating the incorrect outputs, preventing unauthorized access [125,140]. This approach can be integrated at various design stages of the design process, including gate-level to layout-level, by using XOR/XNOR-based locking as well as improved security techniques [50,143].

To safeguard this functionality of the circuits, logic locking in digital adders is integrated into critical key modules such as carry propagation and sum calculation units in order to enhance secure circuit operation [39,41]. This approach plays a key role in preventing the intellectual property and avoiding unauthorized manipulation [46,125]. This also contributes to higher hardware cost, power consumption, and greater latency [98,148]. In addition to that, advanced attack methodologies, including SAT-based attack strategies, have analysed the limitations in certain locking schemes, which indicates the need for stronger and more efficient future solutions [50,143].

5.4 AI-Based Security Approaches

The use of ML-driven security mechanisms has seen significant growth, and they are effective tools for detecting and identifying hardware attacks [47,48]. By analysing the operational patterns of the circuits, these models recognize abnormal deviations that can point to threats such as side-channel attacks and hardware Trojans [142,143]. Common machine learning algorithms like neural networks, detection of anomalies, and support vector machines depend on inputs such as power patterns, timing data, and structural information in order to identify whether a system can operate under secure conditions or indicate abnormal behaviour [138,141].

In digital adders, machine learning techniques are applied to allow continuous observation of fluctuations in power and timing deviations and also to analyse abnormal behaviour, facilitating the rapid detection of possible security threats [47,48]. However, these techniques depend on large-scale training data and suffer from poor transparency, and they can be vulnerable to advanced targeted threats. In spite of these limitations, machine learning-based security solutions hold strong potential and are expected to be the key elements to the evolution of advanced VLSI security architectures [142]. Importantly, ML-driven PPA prediction and side-channel mitigation are complementary within an integrated adder design flow: early predictors can estimate the delay/power/area overhead of candidate countermeasures (masking, hiding, locking), while security-oriented models monitor whether the resulting switching behaviour remains acceptably hardened [35,42,48]. Public resources commonly used to prototype such pipelines include open EDA flows and circuit benchmarks (e.g., OpenROAD-oriented academic flows and standard Trojan/side-channel evaluation suites reported in the hardware-security literature), although adder-specific labeled corpora remain comparatively scarce [34,40].

Table 4 provides a comparative overview of security strategies by describing their functions, advantages and challenges and showcasing the balance between security strength and hardware consumption.

Table 4: Comparison of security-based approaches.

Reference	Security Technique	Side-Channel Protection	Trojan Detection	Overhead	Security Strength	ML Integration Possible
[102,137]	Masking	✓	×	High	High	Partial
[101,147]	Hiding	✓	×	Medium	Medium	Partial
[51,124]	Logic Locking	Partial	✓	Medium	High	✓
[47,48]	ML Detection	✓	✓	Medium	Very High	✓

6 Classification of Existing Approaches

Analysing multiple digital adder architectures indicates a transition from standard architectures towards advanced and optimized architectures for complex environments [35,52]. The conventional simulation methods provide strong accuracy, but they are often not scalable for large-scale complex systems [32,134]. As a result, the use of machine learning and security-focused mechanisms allows efficient performance validation and protects the systems from threats such as hardware Trojans and side-channel attacks [36,47]. Combining these strategies ensures an optimal and balanced trade-off among system performance, operational efficiency, and security aspects [48]. Fig. 6 outlines a systematic classification of various adder design methodologies, organizing them into traditional, ML-based, security-aware, and hybrid techniques along with their respective key features.

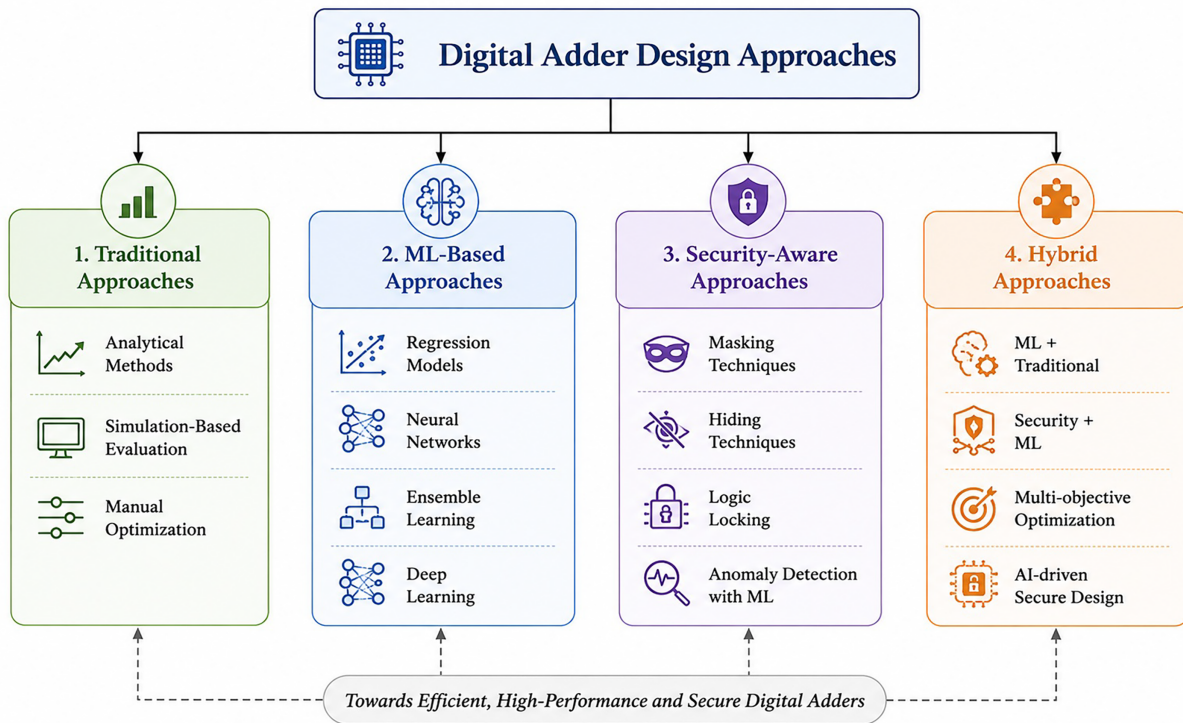


Figure 6: Classification of digital adder design techniques.

Table 5 categorizes different conventional approaches by highlighting their techniques, goals, advantages and limitations in order to offer a structured comparison of multiple design methodologies.

6.1 Traditional Design Approaches

Classical design techniques depend on analytical models and simulation tools in order to study the efficiency of the circuit performance [1,6]. They provide accurate calculations of power consumption, delay, and hardware area, which contribute to reliable circuit performance in high-priority applications [32,58]. Common techniques such as static timing analysis and probabilistic power modelling benefit from their effective integration with next-generation and widely accepted EDA tools [10,57].

Table 5: Feature comparison of digital adder design techniques.

Reference	Features	Traditional	ML Based	Security-Aware	Hybrid
[102]	Delay Optimization	✓	✓	Partial	✓
[48,102]	Power Optimization	Partial	✓	Partial	✓
[49]	Area Efficiency	✓	✓	×	✓
[47,48]	Fast Prediction	×	✓	×	✓
[137,147]	Hardware Security	×	Partial	✓	✓
[101]	Attack Detection	×	Partial	✓	✓
[48]	Scalability	×	✓	Partial	✓
[124,146]	Design Complexity	Low	Medium	High	High
[47,51]	Real-Time Adaptation	×	✓	Partial	✓

However, existing methods struggle to address the need for today's large-scale systems. As the complexity of the circuit size increases, it leads to slower simulations and higher resource requirements, which require multiple executions in order to observe changes in the design, contributing to higher time and cost [32,58]. Although these methods provide high accuracy, they need the support of faster techniques like machine learning for efficient handling of complex circuit designs [35,57].

6.2 Machine Learning-Based Approaches

With the growing need for fast performance, machine learning approaches are becoming an essential alternative to traditional design methods because of their efficiency and scalability. They utilize data-driven models, as they can predict key performance indicators such as power, delay, and area, which minimizes the reliance on detailed simulation efforts [32,35]. Regression models, random forest, support vector machines, and neural networks are trained and identify the complex relationships between input design parameters and system performance [36,37]. Though neural networks are efficient for non-linear relationships, ensemble learning methods strengthen the overall performance of the system and improve reliability [34,37].

By applying machine learning approaches, designers can observe various design alternatives that can enhance efficiency and optimize the design space, thereby speeding up the overall design process [53,156]. However, these techniques rely heavily on high-quality datasets and may suffer from overfitting as well as reduced transparency. The quality of the data which is used might directly affect their accuracy. Though quite a number of challenges remain, machine learning continues to be an important component in the field of modern VLSI design methods [52].

6.3 Security-Aware Approaches

Hardware systems that are designed with security safeguards protect the techniques within the hardware against risks such as hardware Trojans, fault injection, and side-channel attacks [43]. Their primary objective is to concentrate on maintaining the reliability of the system while safeguarding the data from illegal data access to sensitive information [47,48]. Most widely used approaches include masking for data complication through randomization, hiding is performed in order to reduce variations in power and timing behaviour, and logic locking to confirm that the circuit operates in the correct functionality only when the secret key that is applied is valid [41,42].

The application of machine learning in security mechanisms has recently gained significant attention and appeared as an effective approach in order to detect attacks by monitoring the behaviour of the circuit and spotting anomalies for normal operation [47,48]. Their flexible nature makes them well-suited and enhances their effectiveness against newly evolving threats. However, they frequently result in increased power consumption and chip area and may complicate the design process. Thus, recent studies aim to improve these techniques in order to achieve a balanced trade-off between security strength and improved operational efficiency [142].

7 Comparative Analysis

A comparative evaluation is important, and it provides an understanding of analysing various digital adder architectural designs in order to determine their efficiency and limitations [6,19]. Multiple methods are commonly compared using parameters in terms of delay, power utilization, efficiency of the hardware resources, and security, involving traditional, ML-driven, and security-aware solutions [37,52]. Methods which are based on analytical modelling and simulations provide a high level of accuracy and reliable results [10,57]. However, they require high computational resources, which makes them less efficient for large-scale systems due to their extended design period [5,32].

Approaches that are based on machine learning support fast prediction of performance parameters such as power consumption, propagation delay, and chip area, facilitating simple analysis and enhancing design configurations in complex environments [35,36]. Their performance is reliable on high-quality training data, and they can often struggle in terms of model transparency [32]. Security-focused techniques such as masking, logic locking, and hiding improve the protection against threats, which often results in higher hardware cost, enhanced power utilization, and increased latency [41,42].

Overall, each approach complements the others, which brings unique advantages, and conventional methods concentrate on accuracy while machine learning supports efficient circuits, enhances performance and scalable processing, and security-aware measures in order to safeguard the system [35,52]. As a result, the integration of these mechanisms is necessary for enhancing and optimizing the overall performance of circuits while maintaining the system efficiency and security aspects in digital adder architecture designs [47,48]. Critically, published ML results are often difficult to compare directly because datasets, bit widths, technology nodes, and threat models differ; accuracy-like scores for regression-based PPA prediction are not interchangeable with classification scores for Trojan or side-channel detection. Designers should therefore treat cross-paper rankings cautiously and prefer studies that report task-appropriate metrics, hardware platforms, and overhead. Fig. 7 is a conceptual illustration summarizing commonly reported qualitative trends among conventional, ML-assisted, security-aware, and hybrid strategies. Bars are normalized for readability within the figure and are not derived from a single unified experimental campaign in this survey; where a metric is “lower is better” (e.g., delay, power, PDP), smaller conceptual scores indicate preferable behaviour. Fig. 8 conceptually highlights that hybrid strategies are often discussed as improving PDP relative to purely conventional, purely ML-driven, or purely security-hardened flows; the ordering is illustrative and should not be read as statistically aggregated measurements. Fig. 9 conceptually illustrates the performance–security tension and the motivation for hybrid adder design flows; values are schematic.

Table 6 summarizes representative recent studies touching ML-based prediction and/or hardware security. Several entries address general VLSI or security settings rather than adder-only case studies; they are retained when their methods transfer to arithmetic datapaths. Approximate reported scores should not be compared across regression and classification tasks without consulting the original metric definitions.

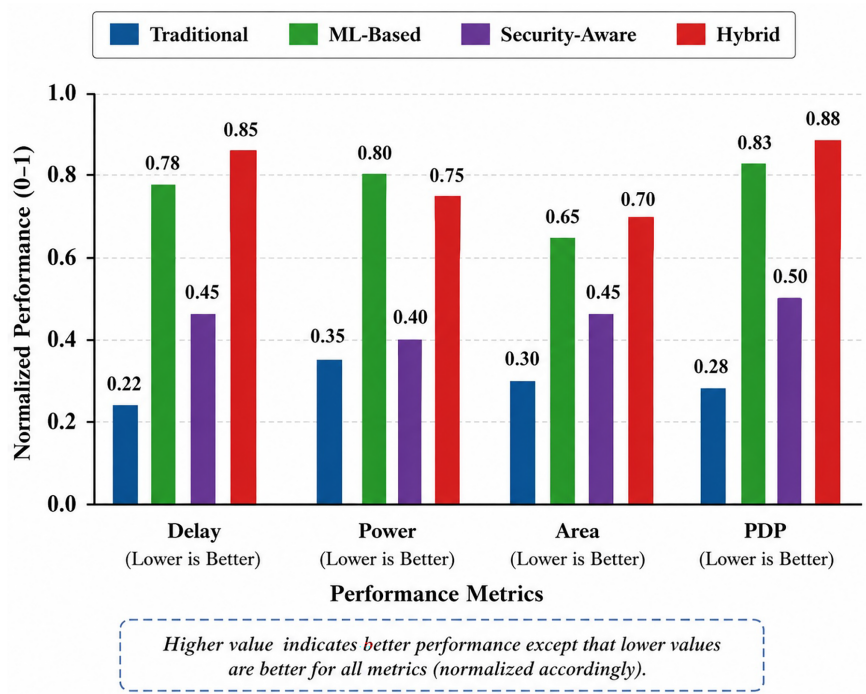


Figure 7: Conceptual comparison of design strategies in terms of delay, power, area, and PDP (illustrative; not a re-simulated benchmark).

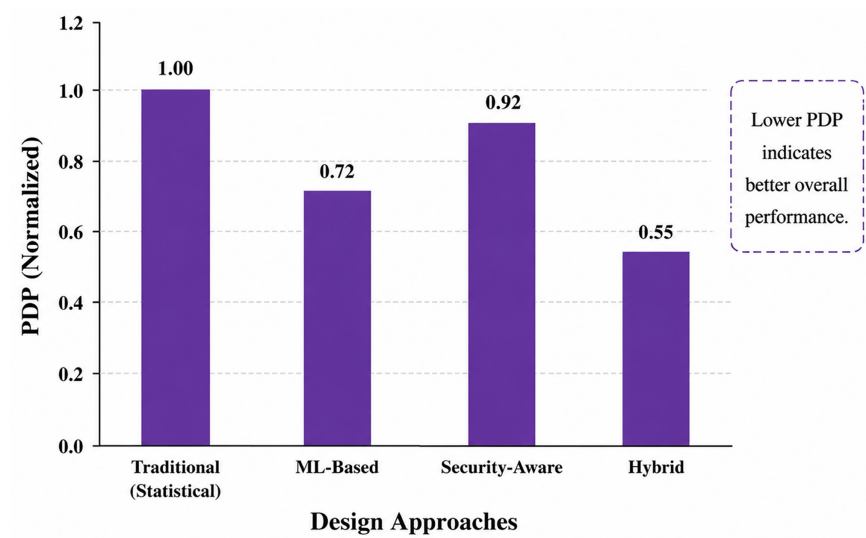


Figure 8: Conceptual comparison of power–delay product trends across design approaches (illustrative).

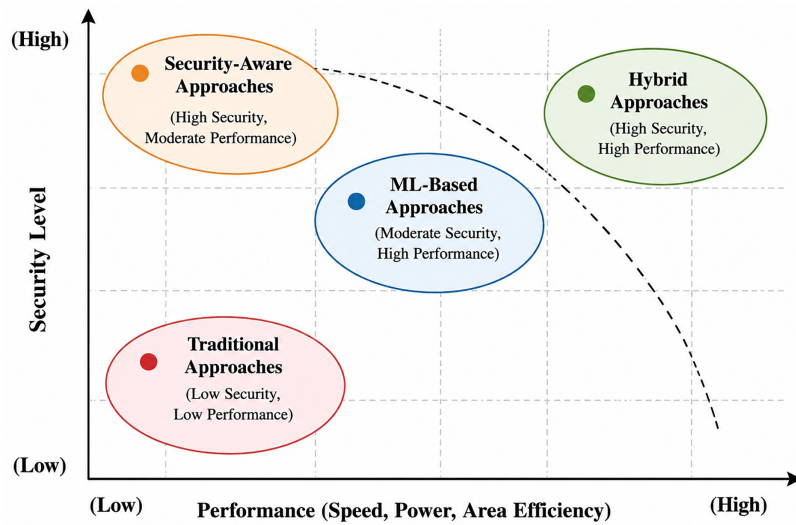


Figure 9: Conceptual trade-off between performance and security emphasis (illustrative).

Table 6: Comparative analysis of recent design approaches related to ML-assisted prediction and/or hardware security (2020–2025), ordered by year.

Ref	Year	Approach	Dataset/Design	Metrics Predicted	ML Model	Security Feature	Approx. Reported Score	Hardware Support
[35]	2021	ML Timing Prediction	Benchmark Circuits	Delay, Power	Regression	No	~94%	FPGA
[59]	2021	Fault Injection Detection	Cryptographic Circuits	Security Metrics	SVM	Yes	~94%	FPGA
[36]	2022	ML-based Power Prediction	Digital Circuit Dataset	Power	Neural Network	No	~95%	ASIC
[53]	2022	EDA Optimization using ML	Simulation Dataset	Power, Area	Random Forest	No	~95%	ASIC
[40]	2023	ML-based Trojan Detection (survey context)	Security Dataset	N/A	SVM	Yes	—	No
[60]	2023	ML-assisted VLSI Prediction	Switching Activity Dataset	Power, Delay	Random Forest	No	~96%	ASIC
[52]	2024	Hybrid ML Security Model	Secure IC Design	Power, Delay	Ensemble Learning	Yes	~97%	FPGA
[135]	2024	High-speed Circuit Optimization	Custom Hardware Dataset	Delay	Neural Network	No	~96%	ASIC
[54]	2025	AI-assisted VLSI Prediction	Large-scale Dataset	Power, Delay, Area	Deep Learning	Partial	~98%	ASIC

8 Open Challenges and Future Directions

8.1 Open Challenges

While advancements have been achieved, multiple challenges continue to slow down the effective use of machine learning techniques in VLSI and hardware security [52,53,134,157]. A major challenge is the limited availability of high-quality datasets, since accurate power modelling, propagation delay, and hardware resources depend on large-scale and wide-ranging data that are cost-effective and time-consuming to generate [32,33,36,37]. The relationship between accuracy and computational efficiency continues to remain as a concern, where advanced techniques like deep neural networks can provide improved accuracy and higher performance, though it results in higher complexity and make the model less transparent [32,35,94,135].

Maintaining an optimal balance between computational performance and effective system security is a significant challenge in VLSI design [2,10,11,52]. Security mechanisms, including masking, data hiding and logic locking, boost and strengthen the protection, often resulting in higher power consumption, longer propagation delay, and complexity of circuits [41,42,48,90]. Moreover, variations in technology and implementations in designs may restrict the capability of machine learning models to adapt efficiently across a wide range of circuit architectures [33,34,135]. Addressing these challenges is required to advance machine learning for secure, high-performance VLSI and adder-centric arithmetic units [40,142,143]. In summary, the dominant open issues are: (1) scarce adder-labeled datasets, (2) limited cross-node/bit-width generalization, (3) explainability and uncertainty quantification, (4) security–PPA overhead trade-offs, and (5) reproducible benchmarking protocols.

8.2 Future Directions

Advancements in machine learning techniques aim to focus on enhancing the speed and more accurate predictions in VLSI system design [52,53,55,134]. Reinforcement learning is expected to improve the possibilities of automated design space exploration and identifying optimal solutions, which leads to shorter design cycles and also reduces the overall design time [34,99,100,156]. Explainable AI is another growing domain that helps designers understand model outputs when reliability and transparency are required [94,103]. Graph-learning methods for netlists and placement/routing context [99,100], together with LLM-assisted RTL generation/exploration [103], are particularly relevant future tools for adder topology search and for auditing security-sensitive arithmetic logic—directly addressing the dataset, generalization, and explainability challenges noted above. Future studies will also concentrate and prioritize on enhancing hardware security with the help of AI-powered strategies. By observing the activity of the circuit, machine learning techniques can detect the possible potential threats like hardware Trojans in real time [48,49,142,143]. Combining classical design techniques with machine learning and security mechanism solutions helps to achieve and enhance better system efficiency and improved protection [35,40].

The utilization of generative AI and data augmentation methods has the capacity to overcome the problem of limited datasets, which supports the development of improved stability and protection mechanisms in VLSI circuit design [52,55,103,135]. It can also focus on exploring the use of hybrid machine learning techniques that combine multiple algorithms in order to achieve enhanced accuracy and reliability in diverse circuit configurations. Such approaches help to gain stronger generalization performance while facilitating efficient design space across a wide range of operating conditions [32,33,35,37]. The integration of multiple learning strategies serves these models effectively to handle complex nonlinear system behaviour and enhance the consistency of predictions in VLSI applications [35,36,38,94]. Also, future work can concentrate on the development of compact and fast ML architectures for real-time VLSI implementations [34,56,95,158].

Due to the increasing trend of VLSI systems in edge computing applications and embedded technologies, there is a strong requirement for lightweight models in terms of computational load and limited hardware resources [53,56,64,158]. By applying optimization techniques including pruning, quantization, and model compression minimize the complexity, while maintaining better security, which facilitates faster and more efficient performance in practical scenarios [34,55,103,135]. Further directions must focus on improving the combination of the process and environmental fluctuations in order to enhance the prediction methodologies [33,58,112,136]. Changes in operating conditions such as temperature variations, voltage fluctuations, and fabrication inconsistencies can reduce the model accuracy, which requires reliability with uncertainty considerations [33,58,136]. Such strategies strengthen the reliability of prediction and support the designers towards efficient decision-making at the beginning of the design process [32,36,37].

Another essential focus of research includes the creation of consistent datasets and performance validation frameworks for machine learning approaches in VLSI design systems [52,53,134,157]. Limited access is given to the availability of high-quality datasets which negatively affects the efficiency and scalability of the models and also minimizes their potential effectiveness [32,33,35,36]. Applying standardized evaluation of performance criteria facilitates consistency in performance analysis and promotes advanced innovations [35,57,58,112]. Upcoming research directions will prioritize the integrated hardware-software design methodologies in order to strengthen the effective interaction between machine learning systems and the development of hardware circuit design implementations [47,48,55,95]. These strategies result in improved performance, maximize the use of hardware resources, and stronger security mechanisms [56,142]. The combination of advanced prediction approaches and security-focused design methodologies contributes to the growth of modern VLSI systems that achieve operational efficiency and reliability while keeping the system flexible [35,40,52,143].

9 Conclusion

This survey presents a detailed overview of machine learning-based performance prediction and hardware security mechanisms in digital adder systems. Although classical methods achieve reliable accuracy, their effectiveness is not suitable and struggle to adapt to system performance, and are less capable of handling the complexities of modern system environments. Machine learning algorithms support rapid and more accurate computation in order to predict important parameters such as power consumption, propagation delay, and hardware area, which leads to optimized design efficiency. The survey also analysed major security threats with defensive mechanisms like data hiding, masking, logic locking, and AI-powered detection mechanisms. To conclude, the survey makes clear that no single approach is sufficient; instead, a hybrid approach is required in combining conventional methods, machine learning techniques, and security-aware designs in order to achieve optimized efficiency and reliable system behaviour.

Acknowledgement: Not applicable

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R343), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: The authors confirm contribution to the paper as follows: Supervision, data collection, study conception and design: Brij B. Gupta, Kwok Tai Chui; analysis and interpretation of results, methodology development: F. M. Karlyn Cynthia, Akshat Gaurav, Varsha Arya, ; draft manuscript preparation, figure and tables: Razaz Waheeb Attar. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. De Micheli G. Synthesis and optimization of digital circuits. New York, NY, USA: McGraw-Hill Higher Education; 1994.
2. Gu RX, Sharaf KM, Elmasry MI. High-performance digital VLSI circuit design. Vol. 338. Boston, MA, USA: Springer; 2012.
3. Lee MTC. High-level test synthesis of digital VLSI circuits. Norwood, MA, USA: Artech House, Inc.; 1997.
4. Marple DP. Performance optimization of digital VLSI circuits. Stanford, CA, USA: Stanford University; 1986.
5. Litovski V, Zwolinski M. VLSI circuit simulation and optimization. Boston, MA, USA: Springer; 1996.
6. Kumar AA. Fundamentals of digital circuits. 4th ed. New Delhi, India: PHI Learning Pvt. Ltd.; 2016.
7. Gurjar P, Solanki R, Kansliwal P, Vucha M. VLSI implementation of adders for high speed ALU. In: Proceedings of the 2011 Annual IEEE India Conference; 2011 Dec 16–18; Hyderabad, India. p. 1–6.
8. Sengupta S, Sarkar P, Dastidar A. Design of a 4 bit arithmetic & logic unit, evaluation of its performance metrics & its implementation in a processor. In: Proceedings of the 2020 International Conference for Emerging Technology (INCET); 2020 Jun 5–7; Belgaum, India. p. 1–8.
9. Santhanam S, Prasad SR. Performance comparison of 16-bit ALU and 32-bit ALU using Verilog. In: Proceedings of the 2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS); 2025 Feb 21–22; Raichur, India. p. 1–6.
10. Bellaouar A, Elmasry M. Low-power digital VLSI design: circuits and systems. Boston, MA, USA: Springer; 2012.
11. Yeap GK. Practical low power digital VLSI design. Vol. 170. Boston, MA, USA: Springer; 1998.
12. Oklobdzija VG, Zeydel BR, Dao HQ, Mathew S, Krishnamurthy R. Comparison of high-performance VLSI adders in the energy-delay space. IEEE Trans Very Large Scale Integr VLSI Syst. 2005;13(6):754–8. doi:10.1109/tvlsi.2005.848819.
13. Vaidya S, Dandekar D. Delay-power performance comparison of multipliers in VLSI circuit design. Int J Comput Netw Commun. 2010;2(4):47–56. doi:10.5121/ijcnc.2010.2405.
14. Archana S, Durga G. Design of low power and high speed ripple carry adder. In: Proceedings of the 2014 International Conference on Communication and Signal Processing; 2014 Apr 3–5; Melmaruvathur, India. p. 939–43.
15. Burgess N. Fast ripple-carry adders in standard-cell CMOS VLSI. In: Proceedings of the 2011 IEEE 20th Symposium on Computer Arithmetic; 2011 Jul 25–27; Tuebingen, Germany. p. 103–11.
16. Pudi V, Sridharan K. Low complexity design of Ripple carry and Brent–Kung Adders in QCA. IEEE Trans Nanotechnol. 2012;11(1):105–19. doi:10.1109/tnano.2011.2158006.
17. Thangam C, Kumari KN, Nagakumararaj S. Implementation of ripple carry adder design for optimal VLSI performance. Int J Comput Appl. 2023;2(11):29–39. doi:10.59544/pbmcl528/ijatemv02i11p3.
18. Kumaar AD, Ajithkumar PR, Giriprasath TK, Selvi RS. An efficient ripple carry adder using pipelining. In: Proceedings of the 2024 International Conference on Innovation and Novelty in Engineering and Technology (INNOVA); 2024 Dec 20–21; Vijayapura, India. p. 1–5.
19. Brent, Kung. A regular layout for parallel adders. IEEE Trans Comput. 1982;C-31(3):260–4. doi:10.1109/tc.1982.1675982.
20. Ladner RE, Fischer MJ. Parallel prefix computation. J ACM. 1980;27(4):831–8. doi:10.1145/322217.322232.
21. Han T, Carlson DA. Fast area-efficient VLSI adders. In: Proceedings of the 1987 IEEE 8th Symposium on Computer Arithmetic (ARITH); 1987 May 18–21; Como, Italy. p. 49–56.
22. Harris D. A taxonomy of parallel prefix networks. In: Proceedings of the The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003; Pacific Grove, CA, USA. p. 2213–7.
23. Ling H. High-speed binary Adder. IBM J Res Dev. 1981;25(3):156–66. doi:10.1147/rd.252.0156.

24. Zimmermann R. Binary adder architectures for cell-based VLSI and their synthesis. Konstanz, Germany: Hartung-Gorre Verlag; 1998.
25. Jayanthi A, Ravichandran C. Comparison of performance of high speed VLSI adders. In: Proceedings of the 2013 International Conference on Current Trends in Engineering and Technology (ICCTET); 2013 Jul 3; Coimbatore, India. p. 99–104.
26. Saini J, Agarwal S, Kansal A. Performance, analysis and comparison of digital adders. In: Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications; 2015 Mar 19–20; Ghaziabad, India. p. 80–3.
27. Sarkar S, Sarkar S, Mehedi J. Comparison of various adders and their VLSI implementation. In: Proceedings of the 2018 International Conference on Computer Communication and Informatics (ICCCI); 2018 Jan 4–6; Coimbatore, India. p. 1–9.
28. Vijay V, Sreevani M, Rekha EM, Moses K, Pittala CS, Shaik KS, et al. A review on n-bit ripple-carry adder, carry-select adder and carry-skip adder. *J VLSI Circuits Syst*. 2022;4(01):27–32. doi:10.31838/jvcs/04.01.05.
29. Chen C, Yang X, Sarraftadeh M. Predicting potential performance for digital circuits. *IEEE Trans Comput Des Integr Circuits Syst*. 1988;21(3):253–62. doi:10.1109/43.986420.
30. Visweswariah. Optimization techniques for high-performance digital circuits. In: Proceedings of the 1997 Proceedings of IEEE International Conference on Computer Aided Design (ICCAD); 1997 Nov 9–13; San Jose, CA, USA. p. 198–207.
31. Lewis DM. Hardware accelerators for timing simulation of VLSI digital circuits. *IEEE Trans Comput-Aided Des Integr Circuits Syst*. 2002;7(11):1134–49. doi:10.1109/43.9184.
32. Kumar AKA, Al-Salamin S, Amrouch H, Gerstlauer A. Machine learning-based Microarchitecture- level power modeling of CPUs. *IEEE Trans Comput*. 2023;72(4):941–56. doi:10.1109/tc.2022.3185572.
33. Zhang X, Zhu B, Liu F, Jiang J, Wang Z, Xu P, et al. Prerouting timing prediction across different technology nodes. *IEEE Trans Comput-Aided Des Integr Circuits Syst*. 2025;44(7):2697–710. doi:10.1109/tcad.2024.3523426.
34. Khailany B. Accelerating chip design with machine learning. In: Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD; 2020 Nov 16–20; Virtual. 33 p.
35. Govindaraj V, Arunadevi B. Machine learning based power estimation for CMOS VLSI circuits. *Appl Artif Intell*. 2021;35(13):1043–55. doi:10.1080/08839514.2021.1966885.
36. Bhavesh MD, Anilkumar NA, Patel MI, Gajjar R, Panchal D. Power consumption prediction of digital circuits using machine learning. In: Proceedings of the 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP); 2022 Feb 12–14; Vijayawada, India. p. 1–6.
37. Gowrishankar R, Govindaraj V, Devi S, Ramesh C. Intelligent power estimation of digital circuits using random forest and neural network models. *Int J Comput Intell Syst*. 2025;18(1):263. doi:10.1007/s44196-025-01000-5.
38. Krishnamoorthy HS, Aayer NT. Machine learning based modeling of power electronic converters. In: Proceedings of the 2019 IEEE Energy Conversion Congress and Exposition (ECCE); 2019 Sep 29–Oct 3; Baltimore, MD, USA. p. 666–72.
39. Karri R, Rajendran J, Rosenfeld K, Tehranipoor M. Trustworthy hardware: identifying and classifying hardware Trojans. *Computer*. 2010;43(10):39–46. doi:10.1109/mc.2010.299.
40. Koblah D, Acharya R, Capecci D, Dizon-Paradis O, Tajik S, Ganji F, et al. A survey and perspective on artificial intelligence for security-aware electronic design automation. *ACM Trans Des Autom Electron Syst*. 2023;28(2):1–57. doi:10.1145/3563391.
41. Rinsy JJ, Sivamangai N, Naveenkumar R, Napoleon A, Puviarasu A, Janani V. Review on logic locking attacks in hardware security. In: Proceedings of the 2022 6th International Conference on Devices, Circuits and Systems (ICDCS); 2022 Apr 21–22; Coimbatore, India. p. 342–7.
42. Tiri K, Verbaudhede I. A VLSI design flow for secure side-channel attack resistant ICs. In: Proceedings of the design, automation and test in Europe; 2005 Mar 7–11; Munich, Germany. p. 58–63.
43. Khan AK, Mahanta HJ. Side channel attacks and their mitigation techniques. In: Proceedings of the 2014 First International Conference on Automation, Control, Energy and Systems (ACES); 2014 Feb 1–2; Adisaptagram, India. p. 1–4.

44. Lomne V, Dehaboui A, Maurine P, Torres L, Robert M. Side channel attacks. In: Security trends for FPGAs: from secured to secure reconfigurable systems. New York, NY, USA: Springer; 2011. p. 47–72.
45. Lyu Y, Mishra P. A survey of side-channel attacks on caches and Countermeasures. *J Hardw Syst Secur.* 2018;2(1):33–50. doi:10.1007/s41635-017-0025-y.
46. Gupta H, Mondal S, Majumdar R, Ghosh NS, Khan SS, Kwanyu NE, et al. Impact of side channel attack in information security. In: Proceedings of the 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE); 2019 Dec 11–12; Dubai, United Arab Emirates. p. 291–5.
47. Mushtaq M, Akram A, Bhatti MK, Chaudhry M, Yousaf M, Farooq U, et al. Machine learning for security: the case of side-channel attack detection at run-time. In: Proceedings of the 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS); 2018 Dec 9–12; Bordeaux, France. p. 485–8.
48. Hettwer B, Gehrler S, Güneysu T. Applications of machine learning techniques in side-channel attacks: a survey. *J Cryptogr Eng.* 2020;10(2):135–62. doi:10.1007/s13389-019-00212-8.
49. Levina A, Sleptsova D, Zaitsev O. Side-channel attacks and machine learning approach. In: Proceedings of the 2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT); 2016 Apr 18–22; St-Petersburg, Russia. p. 181–6.
50. Phan QS, Bang L, Pasareanu CS, Malacaria P, Bultan T. Synthesis of adaptive side-channel attacks. In: Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF); 2017 Aug 21–25; Santa Barbara, CA, USA. p. 328–42.
51. Naserelden S, Alias N, Altigani A, Mohamed A, Badreddine S. Advance attacks on AES: a comprehensive review of side channel, fault injection, machine learning and quantum techniques. *Edelweiss Appl Sci Technol.* 2025;9(4):2471–86. doi:10.55214/25768484.v9i4.6586.
52. Attaoui Y, Chentouf M, Ismaili ZEAA, Elmourabit A. Machine learning in VLSI design: a comprehensive review. *J Integr Circuits Syst.* 2024;19(2):1–14.
53. Ren H, Hu J. Machine learning applications in electronic design automation. Cham, Switzerland: Springer; 2022.
54. Khan MJ, Gaur P, Singh I, Ahmad S. Role of machine learning applications in VLSI design. In: Advancing VLSI through machine learning. Boca Raton, FL, USA: CRC Press; 2025. p. 149–83.
55. Maheswari S, Gokilam G, Navaneethan S, Amer A, Sunheriya N, Parthi PS. AI-powered VLSI design automation for faster chip production in emerging technologies. In: Proceedings of the 2025 3rd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT); 2025 Mar 21–22; Dehradun, India. p. 445–50.
56. Nagar P, Boruah S, Bhoi AK, Patel A, Sarda J, Darjij P. Emerging VLSI technologies for high performance AI and ML applications. In: Proceedings of the 2024 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC); 2024 Jan 27–29; Bhubaneswar, India. p. 1–5.
57. Najm FN. Transition density: a new measure of activity in digital circuits. *IEEE Trans Comput-Aided Des Integr Circuits Syst.* 1993;12(2):310–23. doi:10.1109/43.205010.
58. Roy K, Mukhopadhyay S, Mahmoodi-Meimand H. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proc IEEE.* 2003;91(2):305–27. doi:10.1109/jproc.2002.808156.
59. Gaber L, Hussein AI, Moness M. Fault detection based on deep learning for digital VLSI circuits. *Procedia Comput Sci.* 2021;194:122–31. doi:10.1016/j.procs.2021.10.065.
60. Kumar A, Tripathi SL, Rao KS. Machine learning for VLSI chip design. Hoboken, NJ, USA: John Wiley & Sons; 2023.
61. Khan IU, Mittal N, Ansari MA. Applications of VLSI design in artificial intelligence and machine learning. In: Machine learning for VLSI chip design. Hoboken, NJ, USA: Wiley; 2023. p. 1–17.
62. Lata K, Saini S, Sinha G. VLSI and hardware implementation using machine learning methods: a systematic literature review. In: VLSI and hardware implementations using modern machine learning methods. Boca Raton, FL, USA: CRC Press; 2021. p. 1–21.
63. Tigadi A. Survey on VLSI design for artificial intelligence and machine learning applications. *Int J Res Anal Rev.* 2023;10(2):736–40. doi:10.1002/9781119910497.ch1.

64. Ande JRPKR, Khair MA. High-performance VLSI architectures for artificial intelligence and machine learning applications. *Int J Recipr Symmetry Theor Phys*. 2019;6(1):20–30. doi:10.1002/9781119910497.ch1.
65. Lorandel J, Prevotet JC, Helard M. Fast power and performance evaluation of FPGA-based wireless communication systems. *IEEE Access*. 2016;4:2005–18. doi:10.1109/access.2016.2559781.
66. Mametjanov A, Balaprakash P, Choudary C, Hovland PD, Wild SM, Sabin G. Autotuning FPGA design parameters for performance and power. In: *Proceedings of the 2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*; 2015 May 2–6; Vancouver, BC, Canada. p. 84–91.
67. Kulkarni A, Heyse K, Davidson T, Stroobandt D. Performance evaluation of dynamic circuit Specialization on Xilinx FPGAs. In: *Proceedings of the FPGA World Conference 2014*; 2014 Sep 9–11; Stockholm and Copenhagen, Sweden. p. 1–6.
68. Coffman K. *Real world FPGA design with verilog*. Upper Saddle River, NJ, USA: Pearson Education; 1999.
69. Singh KS, Mittra S, Singh N, Birla S. Machine learning-driven prediction of optimal design parameters for Ripple carry Adder. *IEEE Access*. 2026;14(24):16296–305. doi:10.1109/access.2026.3658174.
70. Ain NU, Ahmadpour SS, Navimipour NJ, Diakina E, Kassa SR. Secure quantum-based Adder design for protecting machine learning systems against side-channel attacks. *Appl Soft Comput*. 2025;169(3):112554. doi:10.1016/j.asoc.2024.112554.
71. Bachir M, Salah AM, Said S, Redouane K, Silva B. Optimized modular Adder architecture for Cryptographic applications on FPGAs. *IEEE Trans Comput-Aided Des Integr Circuits Syst*. 2025;44(6):2168–80. doi:10.1109/tcad.2024.3518412.
72. Gattu N, Khan MNI, De A, Ghosh S. Power side channel attack analysis and detection. In: *Proceedings of the 39th International Conference on Computer-Aided Design*; 2020 Nov 2–5; Virtual. p. 1–7.
73. Saikumar M, Samundiswary DP. Design and performance analysis of various adders using verilog. *Int J Comput Sci Mob Comput*. 2013;2:128–38. doi: 10.5281/zenodo.32564.
74. Dubey N, Akashe S. Implementation of an Arithmetic logic using area efficient carry Lookahead Adder. *Int J VLSI Des Commun Syst*. 2014;5(6):29–34. doi:10.5121/vlsic.2014.5604.
75. Ganesh KS, Agrawal S, Paramasivam C. Performance comparison of advanced 32-bit ALU architectures using optimized adders. In: *Proceedings of the 2025 IEEE 2nd International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*; 2025 Aug 29–30; Bangalore, India. p. 1–6.
76. Chopde A, Kale J, Kamble R, Paigude K. Design of a 4-bit 4-operand adder using Verilog: an abstraction analysis. In: *Proceedings of the 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*; 2024 Feb 24–25; Bhopal, India. p. 1–9.
77. Zeydel BR, Baran D, Oklobdzija VG. Energy-efficient design methodologies: high-performance VLSI Adders. *IEEE J Solid-State Circuits*. 2010;45(6):1220–33. doi:10.1109/jssc.2010.2048730.
78. Samanta J, Halder M, De BP. Performance analysis of high speed low power carry look-ahead adder using different logic styles. *Int J Soft Comput Eng*. 2013;2(6):330–6. doi:10.6088/ijaser.0020101045.
79. Shubhangi, Malhotra D, Rani R, Usmani H, Kumar A, Goel N. Performance analysis of carry look ahead adder and ripple carry adder. In: *Proceedings of the 2025 5th International Conference on Advancement in Electronics & Communication Engineering (AECE)*; 2025 Nov 21–22; Ghaziabad, India. p. 594–8
80. Nithyashree R, Afreen S, Tantry S. Analysis of various approximate adders in Ripple carry adder design. In: *Proceedings of the 2023 4th IEEE Global Conference for Advancement in Technology (GCAT)*; 2023 Oct 6–8; Bangalore, India. p. 1–5.
81. Mei OS, Dhahi TS, Rahman SFA, Fathil M, Zakaria MR, Zawawi MAM. Improved Ripple carry Adder with reduced delay and low power consumption at circuit level using 180 nm technology. *J Phys Conf Ser*. 2025;3020(1):012001. doi:10.1088/1742-6596/3020/1/012001.
82. Koyada B, Meghana N, Jaleel MO, Jeripotula PR. A comparative study on adders. In: *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*; 2017 Mar 22–24; Chennai, India. p. 2226–30.
83. Balasubramanian P, Yamashita S. Area/latency optimized early output asynchronous full adders and relative-timed ripple carry adders. *SpringerPlus*. 2016;5(1):440. doi:10.1186/s40064-016-2074-z.

84. Wang Z, Jullien GA, Miller WC, Wang J. New concepts for the design of carry lookahead adders. In: Proceedings of the 1993 IEEE International Symposium on Circuits and Systems; 1993 May 3–6; Chicago, IL, USA. p. 1837–40.
85. Doran RW. Variants of an improved carry look-ahead adder. *IEEE Trans Comput.* 1988;37(9):1110–3. doi:10.1109/12.2261.
86. Boateng EA. Design and implementation of a 16 bit carry-lookahead adder. Norfolk, VA, USA: Norfolk State University; 2019.
87. Ahmadpour SS, Jafari Navimipour N. A new nano-design of 16-bit carry look-ahead adder based on quantum technology. *Phys Scr.* 2023;98(12):125108.
88. Kim Y, Kim LS. A low power carry select adder with reduced area. In: ISCAS 2001. The 2001 IEEE international symposium on circuits and systems (Cat. No. 01CH37196). Vol. 4. New York, NY, USA: IEEE; 2001. p. 218–21.
89. Dhandapani V. An efficient architecture for carry select adder. *World J Eng.* 2017;14(3):249–54. doi:10.1108/wje-08-2016-0043.
90. Vasudevan DP, Lala PK, Parkerson JP. Self-checking carry-select Adder design based on two-rail encoding. *IEEE Trans Circuits Syst I Regul Pap.* 2007;54(12):2696–705. doi:10.1109/tcsi.2007.910537.
91. Macedo M, Soares L, Silveira B, Diniz CM, da Costa EA. Exploring the use of parallel prefix adder topologies into approximate adder circuits. In: Proceedings of the 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS); 2017 Dec 5–8; Batumi, Georgia. p. 298–301.
92. Srinivas HR, Parhi KK. A fast VLSI adder architecture. *IEEE J Solid-State Circuits.* 1992;27(5):761–7. doi:10.1109/4.133165.
93. Singh AP, Mishra VK, Akhter S. A perspective view of silicon based classical to non-classical MOS Transistors and their extension in machine learning. *Silicon.* 2023;15(16):6763–84.
94. Pandey D, Niwaria K, Chourasia B. Machine learning algorithms: a review. *Mach Learn.* 2019;6(2):916–22. doi:10.21275/art20203995.
95. Sai ADN, Kumar AUS, Mathan N, GD AJ, Nandhitha NM. AI driven fault detection for advanced digital VLSI systems. In: Proceedings of the 2025 11th International Conference on Communication and Signal Processing (ICCSP); 2025 Jun 5–7; Melmaruvathur, India. p. 125–30.
96. Jamieson P, Rose J. A verilog RTL synthesis tool for heterogeneous FPGAs. In: Proceedings of the International Conference on Field Programmable Logic and Applications; 2005 Aug 24–26; Tampere, Finland. p. 305–10.
97. Mendez T, Ghanashyam N, Balakrishnan A, Puthalath A. Implementation and performance evaluation of fundamental sorting algorithms using Verilog. In: Proceedings of the 2025 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER); 2025 Oct 17–18; Mangalore, India. p. 679–85.
98. Atwell ID, Perera DG. HDL code variation: impact on FPGA performance metrics and CAD tools. In: Proceedings of the 2024 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM); 2024 Aug 21–24; Victoria, BC, Canada. p. 1–6.
99. Dinh TX, Do TD. Graph-based ranking techniques for improving VLSI placement. In: Proceedings of the 2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON); 2024 Oct 17–19; Yorktown Heights, NY, USA. p. 715–9.
100. Cheng R, Yan J. On joint learning for solving placement and routing in chip design. *Adv Neural Inf Process Syst.* 2021;34:16508–19. doi: 10.48550/arxiv.2111.00234.
101. Sayakkara A, Le-Khac NA, Scanlon M. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. *Digit Investig.* 2019;29(11):43–54. doi:10.1016/j.diin.2019.03.002.
102. Gamaarachchi H, Ganegoda H. Power analysis based side channel attack. *arXiv:1801.00932.* 2018. doi: 10.48550/arxiv.1801.00932.
103. Thorat K, Zhao J, Liu Y, Hasan A, Peng H, Xie X, et al. LLM-VeriPPA: power, performance, and area optimization aware Verilog code generation with large language models. In: Proceedings of the 2025 ACM/IEEE 7th Symposium on Machine Learning for CAD (MLCAD); 2025 Sep 8–10; Santa Cruz, CA, USA. p. 1–7.
104. Bahadori M, Kamal M, Afzali-Kusha A, Pedram M. High-speed and energy-efficient carry skip Adder operating under a wide range of supply voltage levels. *IEEE Trans Very Large Scale Integr VLSI Syst.* 2016;24(2):421–33. doi:10.1109/tvlsi.2015.2405133.

105. Patel S, Garg B, Mahajan A, Rai S. Area-delay efficient and low-power carry skip adder for high performance computing systems. In: Proceedings of the 2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS); 2019 Dec 16–18; Rourkela, India. p. 300–3.
106. Mahapatra NR, Tareen A, Garimella SV. Comparison and analysis of delay elements. In: Proceedings of the 2002 45th Midwest Symposium on Circuits and Systems MWSCAS-2002; 2002 Aug 4–7; Tulsa, OK, USA. p. II-473–6.
107. Del Barrio AA, Hermida R, Memik SO. A partial carry-save on-the-fly correction multispeculative multiplier. *IEEE Trans Comput.* 2016;65(11):3251–64.
108. Alioto M, Palumbo G. A simple strategy for optimized design of one-level carry-skip adders. *IEEE Trans Circuits Syst I Fundam Theory Appl.* 2003;50(1):141–8. doi:10.1109/tcsi.2002.807517.
109. Guyot, Hochet, Muller. A way to build efficient carry-skip adders. *IEEE Trans Comput.* 1987;100(10):1144–52. doi:10.1109/tc.1987.1676855.
110. Chan PK, Schlag MD, Thomborson CD, Oklobdzija VG. Delay optimization of carry-skip adders and block carry-lookahead adders. In: Proceedings of the IEEE Symposium on Computer Arithmetic; 1991 Jun 29–Jul 2; Windsor, ON, Canada. p. 154–64.
111. Lin YS, Radhakrishnan D. Delay efficient 32-bit carry-skip adder. *VLSI Des.* 2008;2008(1):218565.
112. Moos T, Moradi A, Richter B. Static power side-channel analysis—an investigation of measurement factors. *IEEE Trans Very Large Scale Integr VLSI Syst.* 2020;28(2):376–89. doi:10.1109/tvlsi.2019.2948141.
113. Alahmadi AN, Rehman SU, Alhazmi HS, Glynn DG, Shoaib H, Solé P. Cyber-security threats and side-channel attacks for digital agriculture. *Sensors.* 2022;22(9):3520. doi:10.3390/s22093520.
114. Ometov A, Levina A, Borisenko P, Mostovoy R, Orsino A, Andreev S. Mobile social networking under side-channel attacks: practical security challenges. *IEEE Access.* 2017;5:2591–601. doi:10.1109/access.2017.2665640.
115. Le TH, Canovas C, Clédiere J. An overview of side channel analysis attacks. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security; 2008 Mar 18–20; Tokyo Japan. p. 33–43.
116. Lomné V, Prouff E, Roche T. Behind the scene of side channel attacks. In: International Conference on the Theory and Application of Cryptology and Information security. Berlin/Heidelberg, Germany: Springer; 2013. p. 506–25.
117. Spreitzer R, Moonsamy V, Korak T, Mangard S. Systematic classification of side-channel attacks: a case study for mobile devices. *IEEE Commun Surv Tutor.* 2018;20(1):465–88.
118. Schellenberg F, Gnad DRE, Moradi A, Tahoori MB. Remote inter-chip power analysis side-channel attacks at board-level. In: Proceedings of the International Conference on Computer-Aided Design; 2018 Nov 5–8; San Diego, CA, USA. p. 1–7.
119. Yucebas D, Yuksel H. Power analysis based side-channel attack on visible light communication. *Phys Commun.* 2018;31(9):196–202. doi:10.1016/j.phycom.2018.04.013.
120. Ramesh C, Patil SB, Dhanuskodi SN, Provelengios G, Pillement S, Holcomb DE, et al. FPGA side channel attacks without physical access. In: Proceedings of the 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM); 2018 Apr 29–May 1; Boulder, CO, USA. p. 45–52.
121. Zhao M, Suh GE. FPGA-based remote power side-channel attacks. In: Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP); 2018 May 20–24; San Francisco, CA, USA. p. 229–44.
122. Peeters E, Standaert FX, Donckers N, Quisquater JJ. Improved higher-order side-channel attacks with FPGA experiments. In: International workshop on cryptographic hardware and embedded systems. Berlin/Heidelberg, Germany: Springer; 2005. p. 309–23.
123. Elaabid MA, Meynard O, Guilley S, Danger JL. Combined side-channel attacks. In: International workshop on information security applications. Berlin/Heidelberg, Germany: Springer; 2010. p. 175–90.
124. Karimi E, Jiang ZH, Fei Y, Kaeli D. A timing side-channel attack on a mobile GPU. In: Proceedings of the 2018 IEEE 36th International Conference on Computer Design (ICCD); 2018 Oct 7–10. Orlando, FL, USA. p. 67–74.
125. Wittteman M, Oostdijk M. Secure application programming in the presence of side channel attacks. In: Proceedings of the RSA Conference; 2008 Apr 7–11; San Francisco, CA, USA.
126. Page D. Defending against cache-based side-channel attacks. *Inf Secur Tech Rep.* 2003;8(1):30–44. doi:10.1016/s1363-4127(03)00104-3.
127. Rao JR, Rohatgi P. Empowering side-channel attacks. *Cryptology EPrint Archive.* 2001.

128. Balasubramanian P, Maskell DL. A new carry look-ahead Adder architecture optimized for speed and energy. *Electronics*. 2024;13(18):3668. doi:10.3390/electronics13183668.
129. Balasubramanian P, Mastorakis NE. High-speed and energy-efficient carry look-ahead Adder. *J Low Power Electron Appl*. 2022;12(3):46.
130. Wang Y, Pai C, Song X. The design of hybrid carry-lookahead/carry-select adders. *IEEE Trans Circuits Syst II Analog Digit Signal Process*. 2002;49(1):16–24.
131. Mohanty BK, Patel SK. Area–delay–power efficient carry-select Adder. *IEEE Trans Circuits Syst II Express Briefs*. 2014;61(6):418–22. doi:10.1109/tcsii.2014.2319695.
132. Jothin R, Sreelatha P, Ahilan A, Mohamed MP. High-performance carry select Adders. *Circuits Syst Signal Process*. 2021;40(8):4169–85. doi:10.1007/s00034-021-01658-0.
133. Rajagopal T, Chakrapani A. A novel high-performance hybrid full Adder for VLSI circuits. *Circuits Syst Signal Process*. 2021;40(11):5718–32. doi:10.1007/s00034-021-01725-6.
134. Gubbi KI, Beheshti-Shirazi SA, Sheaves T, Salehi S, PD SM, Rafatirad S, et al. Survey of machine learning for electronic design automation. In: *Proceedings of the Great Lakes Symposium on VLSI 2022*; 2022 Jun 6–8; Irvine, CA, USA. p. 513–8.
135. Melikyan V. *Machine learning-based design and optimization of high-speed circuits*. Cham, Switzerland: Springer; 2024.
136. Banerjee K, Pedram M, Ajami AH. Analysis and optimization of thermal issues in high-performance VLSI. In: *Proceedings of the 2001 International Symposium on Physical Design*; 2001 Apr 1–4; San Diego, CA, USA. p. 230–7.
137. Mangard S, Oswald E, Popp T. *Power analysis attacks: revealing the secrets of smart cards*. New York, NY, USA: Springer; 2007.
138. Wang Z, Meng FH, Park Y, Eshraghian JK, Lu WD. Side-channel attack analysis on in-memory computing architectures. *IEEE Trans Emerg Top Comput*. 2024;12(1):109–21. doi:10.1109/tetc.2023.3257684.
139. Das D, Maity S, Nasir SB, Ghosh S, Raychowdhury A, Sen S. High efficiency power side-channel attack immunity using noise injection in attenuated signature domain. In: *Proceedings of the 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*; 2017 May 1–5; Mclean, VA, USA. p. 62–7.
140. Bayrak AG, Velickovic N, Ienne P, Burleson W. An architecture-independent instruction shuffler to protect against side-channel attacks. *ACM Trans Archit Code Optim*. 2012;8(4):1–19. doi:10.1145/2086696.2086699.
141. Lerman L, Bontempi G, Markowitch O. Side channel attack: an approach based on machine learning. *Cent Adv Secur Res Darmstadt*. 2011;29(2):29–41. doi:10.1504/ijact.2014.062722.
142. Sharma Y, Kaushik P, Malhotra I. Enhancing FPGA security: an AI-driven approach to root of trust architecture. In: *Proceedings of the 2025 7th International Conference on Signal Processing, Computing and Control (ISPPCC)*; 2025 Mar 6–8; Solan, India. p. 1080–5.
143. Bepary MK, Zhang T, Farahmandi F, Tehranipoor M. PreSCAN: a comprehensive review of pre-silicon physical side-channel vulnerability assessment methodologies. *Chips*. 2024;3(4):311–33. doi:10.3390/chips3040016.
144. Standaert FX, Malkin TG, Yung M. A unified framework for the analysis of side-channel key recovery attacks. In: *Annual International Conference on the Theory And Applications of Cryptographic Techniques*. Berlin/Heidelberg, Germany: Springer; 2009. p. 443–61.
145. Fei Y, Ding AA, Lao J, Zhang L. A statistics-based fundamental model for side-channel attack analysis. *Cryptol EPrint Arch*. 2014.
146. Molnar D, Piotrowski M, Schultz D, Wagner D. The program counter security model: automatic detection and removal of control-flow side channel attacks. In: *International Conference on Information Security and Cryptology*. Berlin/Heidelberg, Germany: Springer; 2005. p. 156–68.
147. Tang M, Luo M, Zhou J, Yang Z, Guo Z, Yan F, et al. Side-channel attacks in a real scenario. *Tsinghua Sci Technol*. 2018;23(5):586–98. doi:10.26599/tst.2018.9010047.
148. Gaskin T. *Exploring the performance impacts of harmful FPGA configurations*. Provo, UT, USA: Brigham Young University; 2021.
149. Breier J, Hou X. How practical are fault injection attacks, really? *IEEE Access*. 2022;10:113122–30. doi:10.1109/access.2022.3217212.

150. Barenghi A, Breveglieri L, Koren I, Naccache D. Fault injection attacks on Cryptographic devices: theory, practice, and Countermeasures. *Proc IEEE*. 2012;100(11):3056–76. doi:10.1109/jproc.2012.2188769.
151. Moro N, Dehbaoui A, Heydemann K, Robisson B, Encrenaz E. Electromagnetic fault injection: towards a fault model on a 32-bit Microcontroller. In: *Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*; 2013 Aug 20; Los Alamitos, CA, USA. p. 77–88.
152. Richter-Brockmann J, Rezaei Shahmirzadi A, Sasdrich P, Moradi A. FIVER—robust verification of countermeasures against fault injections. *IACR Trans Cryptogr Hardw Embed Syst*. 2021;2021(4):447–73. doi:10.46586/tches.v2021.i4.447-473.
153. Arribas V, Wegener F, Moradi A, Nikova S. Cryptographic fault diagnosis using VerFI. In: *Proceedings of the 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*; 2020 Dec 7–11; San Jose, CA, USA. p. 229–40.
154. Gangolli A, Mahmoud QH, Azim A. A systematic review of fault injection attacks on IoT systems. *Electronics*. 2022;11(13), 2023.
155. Jiang W. Machine learning methods to detect voltage Glitch attacks on IoT/IIoT infrastructures. *Comput Intell Neurosci*. 2022;2022:6044071–7. doi:10.1155/2022/6044071.
156. Li L, Cai Y, Zhou Q. A survey on machine learning-based routing for VLSI physical design. *Integration*. 2022;86:51–6. doi:10.1016/j.vlsi.2022.05.003.
157. Khetarpal V, Gupta L, Dhand R, Sharma P. Machine learning techniques for VLSI circuit design: a review. In: *International Conference on Intelligent Systems Design and Applications*. Berlin/Heidelberg, Germany: Springer; 2023. p. 191–9.
158. Jiang W. VLSI design for AI applications. In: *Machine learning in nanoelectronics: devices, circuits and systems*. Hoboken, NJ, USA: Wiley; 2026. p. 197–230.