MANTECH Publications

Journal of Research in VLSI Design Tools and Technology
Volume 11, Issue 1, January-April 2026

Open-Source Autonomous VLSI Design Tools: A Review

Rohan Saini[1], Kapil Singh[2], Priyanka Gusain[3]

Professor[1], Students[2, 3]

Department of ECE

Sveri's College of Engineering

Email ID: *rohansainibn@yahoo.com[1], kapils1singh@rediffmail.com[2], Priyanka.gusain78@gmail.com[3]*

DOI: *https://doi.org/10.5281/zenodo.19762292*

ABSTRACT

The evolution of Very Large Scale Integration (VLSI) design has witnessed a paradigm shift with the advent of autonomous and open-source tools. Traditionally, VLSI design relied on proprietary Electronic Design Automation (EDA) tools, which were expensive and limited in flexibility. The emergence of open-source autonomous design tools has democratized chip design, providing accessible, customizable, and intelligent solutions. This paper reviews the current state of open-source autonomous VLSI design tools, discussing their architectures, capabilities, and the integration of Artificial Intelligence (AI) and Machine Learning (ML) for tasks such as placement, routing, power optimization, and verification. Key challenges, limitations, and potential research directions are also explored. The review aims to provide a comprehensive guide for researchers, educators, and industry practitioners interested in adopting open-source autonomous approaches for VLSI design.

KEYWORDS: *VLSI, EDA, Open-source, Autonomous Design, AI, Machine Learning, Placement, Routing, Verification*

INTRODUCTION

Very Large Scale Integration (VLSI) refers to the process of integrating millions of transistors onto a single chip. As the complexity of integrated circuits (ICs) grows, traditional manual design methods become impractical. Historically, VLSI designers relied on proprietary Electronic Design Automation (EDA) tools for tasks like synthesis, placement, routing, timing

analysis, and verification. While these tools are mature and powerful, they are often expensive and limit accessibility for smaller institutions or independent researchers.

In recent years, there has been growing interest in **open-source autonomous VLSI design tools**. These tools leverage **AI and ML algorithms** to automate various stages of chip design while being freely available for academic and industrial use. Autonomous tools aim to reduce design time, improve efficiency, and minimize human error in complex design flows.

The paper focuses on reviewing prominent open-source autonomous VLSI tools, their design capabilities, architectural frameworks, and applications. We also discuss the integration of AI-driven methods into VLSI workflows and provide insights into future research directions.

BACKGROUND AND MOTIVATION

Very Large Scale Integration (VLSI) has enabled the integration of millions to billions of transistors on a single silicon die, allowing the development of modern microprocessors, memory chips, and SoCs (System-on-Chip). However, as designs have become more complex, traditional manual design methods have become increasingly impractical. Understanding the traditional VLSI design flow provides the necessary context to appreciate the emergence of **autonomous and open-source design tools**.

1. Traditional VLSI Design Flow

The traditional VLSI design process is a sequential multi-stage workflow, where each stage transforms a high-level specification into a manufacturable chip. Each step requires human expertise and careful consideration of trade-offs between performance, area, power, and reliability.

a) Specification

- The first stage defines the functionality, performance, and constraints of the desired chip.
- Inputs include:
  - Performance targets (e.g., clock frequency, latency)
  - Area constraints
  - Power budget
  - Interface requirements (I/O, memory)

- The specification serves as the blueprint for the entire design process.

b) RTL Design

- RTL (Register-Transfer Level) design is the translation of specifications into **hardware description languages (HDLs)** such as Verilog or VHDL.
- Designers describe the functionality of the circuit in terms of registers, combinational logic, and data flow between them.
- This stage is critical because errors at RTL propagate to later stages.

c) Logic Synthesis

- RTL code is transformed into a **gate-level netlist**, consisting of standard cells (AND, OR, Flip-Flops, etc.).
- Synthesis tools optimize the design for **area, speed, and power** while preserving logical correctness.
- Example tools: Synopsys Design Compiler, Cadence Genus (proprietary).
- Limitations: These tools are often expensive and require expert configuration.

d) Floorplanning

- Floorplanning determines the rough physical placement of **macros and large blocks** on the chip.
- Goals: minimize routing congestion, optimize area utilization, and maintain timing requirements.
- Manual adjustments are often needed for large, complex designs.

e) Placement

- Placement assigns exact coordinates to standard cells within the floorplan.
- Proper placement is crucial to reduce **interconnect length**, which directly affects **timing, power, and congestion**.
- Traditional tools rely on heuristic algorithms; human designers may intervene for optimization.

:

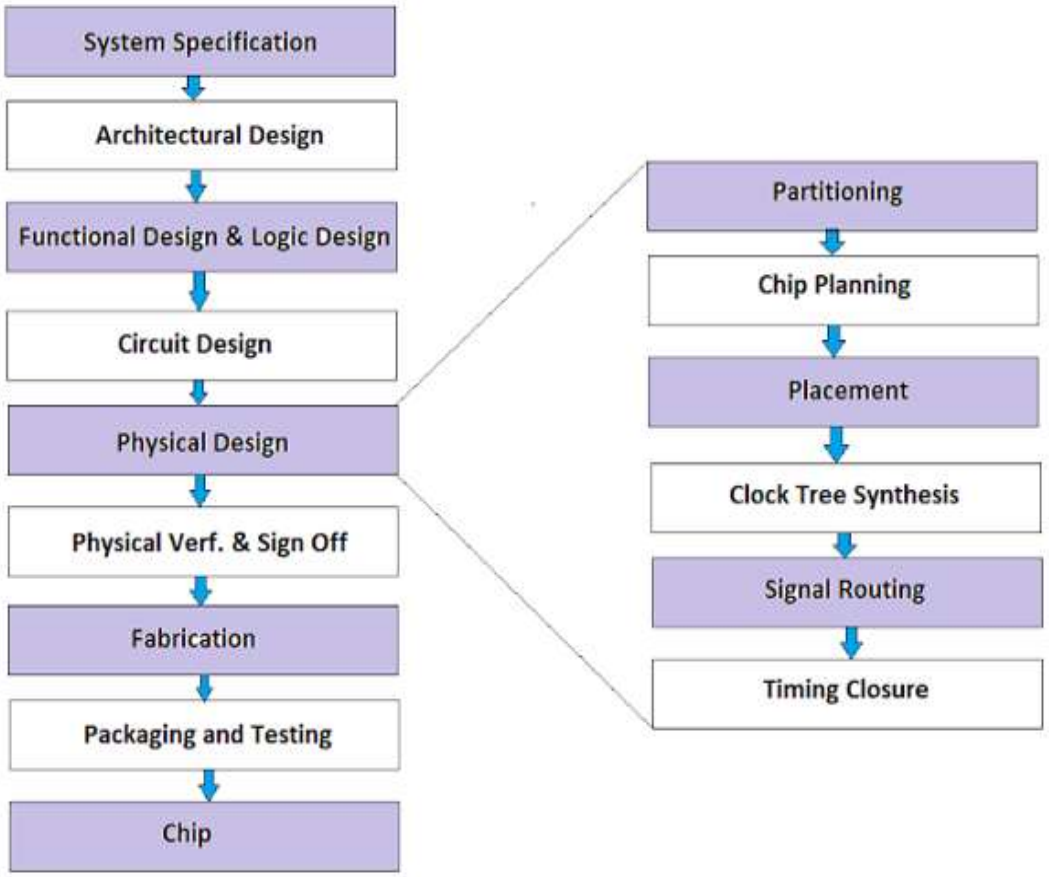

Figure 1: Traditional VLSI Design Flow

2. Motivation for Open-Source Autonomous Tools

The increasing complexity of modern integrated circuits (ICs) has exposed several limitations in traditional, proprietary VLSI design flows, including high cost, long design cycles, and limited flexibility. Open-source autonomous VLSI tools aim to address these challenges by offering accessible, adaptable, and intelligent solutions. The motivations for adopting open-source autonomous tools can be understood in terms of accessibility, flexibility, collaboration, and AI integration.

a) Accessibility

Open-source VLSI tools remove financial and logistical barriers that have historically restricted access to advanced design tools.

- **Cost-Effectiveness:** Proprietary EDA tools, such as Synopsys or Cadence suites, often require expensive licenses that can cost tens or hundreds of thousands of dollars per year. In contrast, open-source tools are freely available, enabling universities, research labs,

startups, and independent designers to perform full-chip design without financial constraints.

- **Educational Benefits:** Students can experiment with complete VLSI flows—from RTL design to GDSII generation—without depending on institutional licenses. Tools like **OpenROAD** or **Qflow** allow students to learn industry-relevant workflows on affordable hardware.
- **Democratization of Design:** Accessibility empowers a wider audience to contribute to VLSI research and design innovation, promoting a more inclusive ecosystem.

b) Flexibility

Open-source tools provide the flexibility to adapt, customize, and extend design capabilities according to specific needs.

- **Algorithm Customization:** Designers can modify placement, routing, and optimization algorithms to suit particular design goals, such as minimizing power consumption for IoT devices or optimizing speed for high-performance computing.
- **Integration with AI Models:** Open-source platforms often allow users to integrate their AI or ML models directly into the design flow. For example, a student could plug a custom reinforcement learning model into OpenROAD's placement engine.
- **Experimentation and Research:** Researchers can experiment with new algorithms, heuristics, and AI-driven methodologies without restrictions imposed by closed-source environments. This flexibility encourages innovation in emerging areas like **autonomous chip design** and **self-driving CAD flows**.

c) Collaboration and Community-Driven Development

Open-source tools foster a collaborative ecosystem where developers, researchers, and students contribute to continuous improvement of the software.

- **Shared Knowledge:** Community-driven projects allow developers to share best practices, optimization techniques, and bug fixes, reducing redundant effort and accelerating tool evolution.
- **Rapid Iteration:** Multiple contributors can implement and test new features in parallel, leading to faster development cycles.
- **Global Accessibility:** Open-source projects like **VTR (Verilog-to-Routing)** or **DREAMPlace** have contributors worldwide, which helps improve robustness, scalability,

and applicability across different technologies.

- **Standardization:** Community involvement often leads to the establishment of open standards, making it easier to integrate different tools in a coherent workflow.

d) AI Integration

One of the most transformative aspects of modern open-source autonomous VLSI tools is the integration of **Artificial Intelligence (AI) and Machine Learning (ML)** techniques.

- **Intelligent Placement and Routing:** AI models can predict optimal placement positions for cells and route connections efficiently, minimizing wirelength, congestion, and timing violations. For instance, **Graph Neural Networks (GNNs)** are used to capture circuit connectivity patterns, and reinforcement learning models iteratively improve placement.
- **Performance Optimization:** AI-driven tools can optimize for multiple objectives simultaneously, such as timing, power, area, and reliability—something that traditional heuristic tools often struggle with.
- **Automation of Repetitive Tasks:** Routine tasks such as netlist extraction, congestion checking, and timing analysis can be automated using AI, freeing designers to focus on high-level architectural decisions.
- **Adaptive Learning:** Models trained on previous designs can generalize to new designs, continuously improving efficiency and effectiveness across multiple projects.

e) Accelerated Design Cycles

By combining accessibility, flexibility, collaboration, and AI-driven automation, open-source autonomous VLSI tools significantly reduce design cycle time.

- Designers spend less time manually performing repetitive tasks and troubleshooting complex layouts.
- Faster iteration allows rapid prototyping and experimentation, which is particularly valuable in academic and startup environments.
- Ultimately, open-source autonomous tools make it feasible to handle **modern SoCs with millions of cells**, which would be extremely challenging using manual or semi-automated traditional flows.

3. Open-Source Autonomous VLSI Tools

The increasing complexity of VLSI designs has led to the emergence of **open-source**

autonomous tools that automate key stages of chip design while being accessible and flexible. Unlike traditional proprietary EDA tools, these platforms are freely available, allow algorithmic customization, and are often designed to integrate AI-driven optimization. Open-source tools span multiple stages of the VLSI flow, including **logic synthesis, placement, routing, verification, and full-chip automation**.

Several notable tools have been developed and widely used in academia and industry research.

a) Summary of Key Open-Source Tools

Table 1: Open-Source Autonomous VLSI Design Tools

Tool Name	Focus Area	Key Features	Language / Platform
OpenROAD	Full-flow Automation	End-to-end autonomous design flow; placement, routing, timing optimization, congestion reduction	C++ / TCL
Qflow	Digital ASIC Flow	Logic synthesis, placement, routing, GDSII export; lightweight, suitable for small ASICs	Makefile / Perl
VTR (Verilog-to-Routing)	FPGA CAD	Logic synthesis, packing, placement, routing; FPGA-specific benchmarks; research-oriented	C++
Odin-II	Logic Synthesis	Technology mapping, RTL-to-gate synthesis, simulation support	C++
GrayWolf	Optimization	AI-based placement and routing optimization, congestion prediction	Python
DREAMPlace	Placement	Deep learning-based placement engine; GPU-accelerated for large designs	C++ / CUDA
RePlAce	Placement	Gradient-based analytical placement; minimizes wirelength and congestion	C++
Kam1n0	Verification / AI	Pattern-matching for reusable IP blocks; assists in automated verification	Java

b) Tool Descriptions and Applications

OpenROAD

OpenROAD (Open-source Rapid Object-Oriented Design Automation) is a full-flow autonomous VLSI design platform.

- Provides **end-to-end automation** from RTL to GDSII.
- Incorporates AI-assisted placement and routing for **timing and congestion optimization**.
- Well-suited for both **academic research** and **industrial prototyping**.
- Example: Students can take a Verilog design, run OpenROAD, and obtain a fully routed GDSII layout without manual intervention.

Qflow

Qflow is an **ASIC design flow** focused on simplicity and accessibility.

- Designed to help students and researchers learn the digital ASIC design process.
- Supports **RTL synthesis, placement, routing, and GDSII export**.
- Lightweight and modular, making it easy to integrate with other tools or customize scripts.
- Limitations: Less scalable for very large or complex designs.

VTR (Verilog-to-Routing)

VTR is a research-oriented tool for **FPGA design automation**.

- Converts Verilog netlists to FPGA configurations through **synthesis, packing, placement, and routing**.
- Includes benchmarking capabilities to compare FPGA architectures and routing strategies.
- Enables experimentation with new optimization algorithms, particularly in **AI-assisted placement and routing**.

Odin-II

Odin-II is a **logic synthesis tool** that converts RTL designs into gate-level netlists suitable for FPGA or ASIC flows.

- Performs **technology mapping**, simulation, and supports multiple backends.
- Often used in combination with VTR for FPGA research.
- Enables researchers to test **AI-driven synthesis optimization** techniques.

GrayWolf

GrayWolf is an emerging tool for **AI-based placement and routing optimization**.

- Uses **machine learning models** to predict congestion and optimize placement automatically.
- Can adaptively refine designs through iterative evaluation.
- Particularly useful in **large-scale ASICs** where conventional heuristics are insufficient.

AI-DRIVEN AUTONOMOUS VLSI DESIGN

The integration of AI and ML into VLSI design tools enables intelligent decision-making in complex tasks:

a) Placement Optimization

Placement determines the physical locations of circuit components on a chip. AI-based placement methods use **reinforcement learning (RL)** and **graph neural networks (GNNs)** to optimize:

- Wirelength
- Congestion
- Timing

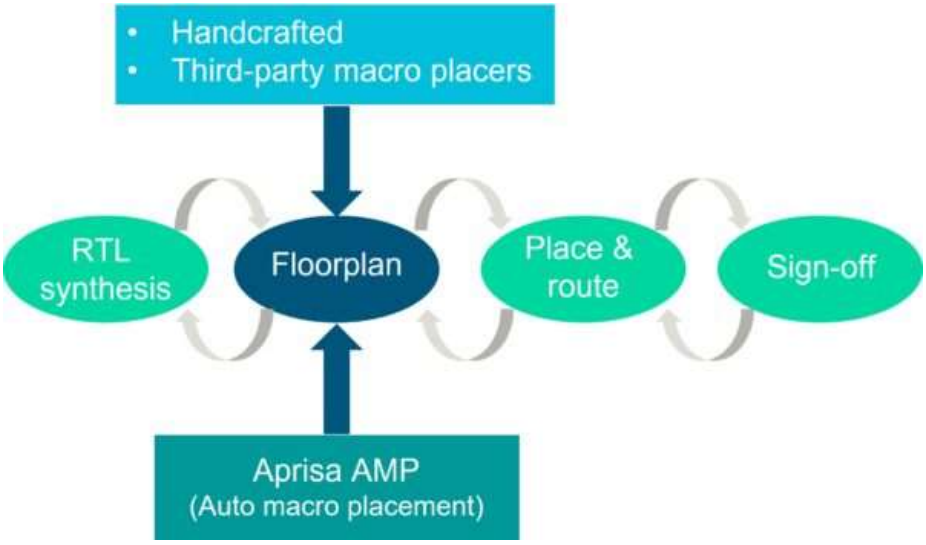

Figure 2: AI-Driven Placement Flow

b) Routing

Routing connects the placed components while minimizing:

- Delay

- Crosstalk
- Power consumption

Autonomous routing engines leverage **reinforcement learning** and **constraint-based algorithms** to reduce congestion and improve signal integrity. OpenROAD and GrayWolf incorporate such AI-assisted routing techniques.

c) Power and Timing Analysis

Predictive ML models can estimate:

- Timing violations
- Power consumption
- Thermal hotspots

Before full simulation, these predictions allow designers to focus only on critical areas, reducing computational time.

d) Verification and Fault Detection

Autonomous verification tools use:

- ML-based bug prediction
- Pattern matching
- Regression testing automation

Such methods reduce human intervention and improve error detection in large designs.

ARCHITECTURAL OVERVIEW OF AUTONOMOUS TOOLS

Autonomous VLSI design tools generally follow a modular architecture:

1. **Input Module:** Accepts HDL (Verilog/VHDL) and design constraints.
2. **Synthesis Engine:** Converts RTL to gate-level netlists.
3. **Optimization Engine:** AI/ML models for placement, routing, and power optimization.
4. **Verification Module:** Automated functional and timing verification.
5. **Output Module:** Generates GDSII/DEF/LEF files for fabrication.

ADVANTAGES OF OPEN-SOURCE AUTONOMOUS TOOLS

- **Cost-Effective:** No licensing fees for academic use.
- **Customizable:** Users can modify algorithms or integrate new ML models.
- **Community Support:** Contributions from global researchers.
- **Faster Prototyping:** Automated optimization reduces design cycle time.
- **Educational Value:** Students can experiment with full-chip design flows.

CHALLENGES AND LIMITATIONS

Despite significant progress, several challenges remain:

1. **Scalability:** Some AI models struggle with very large designs.
2. **Tool Integration:** Combining multiple open-source tools into a coherent flow is non-trivial.
3. **AI Model Training:** Requires large datasets and high computational resources.
4. **Verification Completeness:** Autonomous tools may miss corner-case errors.
5. **Community Adoption:** Some tools lack extensive documentation or user-friendly interfaces.

FUTURE DIRECTIONS

1. **Enhanced AI Models:** Incorporation of more advanced RL, GNNs, and transformer-based models for VLSI design.
2. **Hybrid Flows:** Integration of open-source autonomous tools with proprietary EDA for better scalability.
3. **Cross-Domain Optimization:** Combining power, timing, thermal, and reliability optimization in a unified framework.
4. **Educational Platforms:** Development of low-cost, open-source VLSI labs for students.
5. **Cloud-Based Design Automation:** SaaS platforms for autonomous VLSI design using open-source engines.

CASE STUDIES

1. OpenROAD in Academic Research

OpenROAD has been used in academic projects to implement small to medium digital ASICs. Students reported significant reduction in placement and routing time compared to manual methods.

2. DREAMPlace for AI-Assisted Placement

DREAMPlace uses GPU-accelerated deep learning to optimize placement. In benchmarks, it reduced total wirelength by 12–18% compared to traditional analytical placers.

CONCLUSION

Open-source autonomous VLSI design tools have emerged as a promising alternative to traditional EDA solutions. By integrating AI and ML into design automation, these tools accelerate design cycles, optimize performance, and make VLSI design more accessible. While challenges like scalability, verification, and integration remain, ongoing research and community efforts are steadily improving these tools. Future developments are expected to include more intelligent, hybrid, and cloud-based autonomous VLSI design flows, enabling both academia and industry to leverage these cost-effective and adaptable solutions.

REFERENCES

1. Kahng, A., et al. "OpenROAD: Toward a Self-Driving CAD Flow for Open-Source VLSI." *IEEE Transactions on CAD*, 2020.
2. Chen, W., et al. "DREAMPlace: Deep Learning Toolkit for VLSI Placement." *ACM TODAES*, 2019.
3. Marquardt, A., et al. "Qflow: Open-Source Digital ASIC Design Flow." *FPL*, 2018.
4. Betz, V., Rose, J. "VTR: Verilog-to-Routing for FPGA CAD Research." *FPGA Conference*, 2012.
5. Jiang, L., et al. "AI-Assisted Placement and Routing for VLSI Chips." *IEEE Design & Test*, 2021.
6. Zhou, Y., et al. "GrayWolf: Autonomous AI Optimization for VLSI Layout." *Integration, the VLSI Journal*, 2022.
7. Zhang, Q., et al. "RePlAce: Fast Analytical Placement via Gradient Descent." *IEEE Transactions on CAD*, 2017.
8. Sharma, A., Joshi, M. "Open-Source VLSI Tools for Education and Research." *Journal of VLSI Design Education*, 2021.
9. Lin, Y., et al. "AI in Verification: Machine Learning Approaches for Chip Validation." *ACM SIGDA*, 2020.
10. Desai, R., et al. "Open-Source Autonomous VLSI Design: Trends and Challenges." *International Journal of Electronics*, 2022.

Cite as:

Rohan Saini, Kapil Singh, Priyanka Gusain (2026). Open-Source Autonomous VLSI Design Tools: A Review. Journal of Research in VLSI Design Tools and Technology, 11(1), 52-64.

https://doi.org/10.5281/zenodo.19762292