



VIJAYA KRISHNA KASULA

Principal Engineer, Processor Verification @ Synopsys | Jul 2010 - Present (15+ years)

Worked as a contractor for Synopsys from Jul 2010 - Feb 2014 at SoCronics ([full](#))

KEY PROJECTS (all)

Random Program Generators (RPGs) (details)

Dec 2011 - Present

Lead Developer & Team Lead, Synopsys

- Designed and developed 3 generations of RPGs from scratch for ARC processor verification, each addressing different design philosophies
- G1 (Perl-based, 2011-2016): Fast offline instruction generator supporting 20-30 engineers, primary RPG for 6+ years, phased out for G2
- G2 (SystemVerilog-based, 2016-Present): State-aware generator with coverage-driven features, longest-running tool, 50+ engineers, generates millions of instructions daily, with ~1000 test templates
- G3 (C-based, Sep 2025-Present): High-level scenario generator using CSmith, focuses on test scenarios rather than instruction-level details
- Managed the tools end-to-end, from planning to maintenance, supporting 50+ verification engineers across multiple parallel product lines
- Handled complex challenges: multi-issue scheduling, multicore systems, cache coherency, MMU/MPU, VLIW, SIMD, virtualization, APEX extensions
- Built advanced Memory Manager and Data Manager for architecture-aware instruction streams and complex scenario generation (e.g., cache coherency, DMP)
- Supports automatic extraction of ISA and CSR definitions from processor specs, enabling rapid support for new architectures and near-zero setup for new processors
- Provided SV-Python wrappers to load the test information generated in Text/YAML/JSON formats for seamless testbench integration, includes VSCode IDE with Ctags and extensive Doxygen/Sphinx documentation

AI-ML Log Anomaly Detector (details)

Feb 2022 - Present (Staggered)

Master's Thesis Project, BITS Pilani | Deployed in Production at Synopsys

- Developed intelligent log anomaly detector using scikit-learn, NLTK, and unsupervised learning algorithms
- Multi-stage ML pipeline: preprocessing (noise filtering, lemmatization), hashing vectorizer with trigram N-grams, KD-tree nearest neighbor, agglomerative clustering
- Handles noise (timestamps, file paths), unordered lines, parallel jobs, and template messages with variable parts
- Deployed for a while in production at Synopsys, automating verification failure log triaging
- Currently integrating LLMs into the flow for more intelligent clustering

C-Forge (details)

Mar 2025 - Present

Sole Developer, Synopsys

- Designed ML-enhanced configuration space optimization tool for ARC Processor IP verification
- Used Synopsys VCS Intelligent Coverage Optimization (ICO) with custom cov2gen flow and heuristics
- Auto analyzes uncoverable combinations, triages build failures using Random Forest and Decision Tree classifiers
- Presented at Purple Poster event (Synopsys internal innovation showcase) 2025
- Impact: Reduced verification effort by more than 50% by reducing configurations to verify by at least 50%

Co-Simulation Environment (details)

Dec 2012 - Mar 2013

Lead Developer, Synopsys

- Prototyped experimental co-simulation flow to automate processor-level verification
- Enabled seamless integration between SystemVerilog testbenches and C-based golden reference model
- The tool changed the verification methodology, became critical component of verification infrastructure, enabled robust test development framework

SKILLSET

Programming Languages:

Python, C, C++, SystemVerilog, MySQL, E, Perl, Shell, Bash, Tcsh, Zsh

AI & Machine Learning:

Scikit Learn, NLTK, Pandas, Anomaly Detection, Clustering, Hashing Vectorizer, Kd Tree, Decision Tree, Random Forest, Vibe Coding, Windsurf, GithubCopilot, AI, AI IDE

Data Visualization:

Streamlit, Plotly Dash, Grafana

Tools & Libraries:

Bison, Flex, Yacc, Graphviz, Dot, Compilers, Argparse, VSCode, Cursor, Gvim, SLY, BNF, BeautifulSoup, Lex, Makefile, Verilator, WSL, Ctags, JSON, Parquet, YAML

Version Control & CI/CD:

Git, GitHub, Gitlab, SVN, Perforce, LSF, Farm, CI/CD, OGE

Documentation:

Docusaurus, Sphinx, Doxygen, Markdown, RST, DITA, Documentation, Templates

EDA & Verification:

Verdi, VCS, Specman, ICO, VCS DVE, CSmith, Testbench, FPGA, FSDB, HVP, Synopsys, Debugging, IPXACT, Verification Plans

Project Execution:

Jama, Jira, 8D, Project Dashboard

Processor Architecture:

MMU, MPU, ISA, ISA Coverage, Action Points, ARC ISA, Debug Unit, RISC V, Processor State, Assembly Language, Cache Coherency, Caches, CSRs

Processor Verification:

Verification, Test Generation, Co Simulation, PLV, MLV, Coverage, Cov2Gen, Constraints, Configuration Space

Process & Methodology:

Automation, Best Practices, Checklists, Guidelines, Leadership, Migration, Prototyping, Regression Automation, Sessions, Training, Workaround Management

LEADERSHIP & MANAGEMENT

Verification Best Practices Team (VBPT) [\(details\)](#)

2020 - Present

Team Lead, Synopsys

Team Contributions:

- Led cross-functional team responsible for verification infrastructure and process improvement
- Established best practices, guidelines, and standards for verification across product lines
- Managed innovation initiatives and explored AI/ML capabilities for verification automation
- Led verification plans evolution through 3 generations (Excel → HVP/Verdi → Jama), built automation tools for spec linking and coverage back annotation, established templates and guidelines, conducted training sessions
- Introduced SystemVerilog coding guidelines across verification team
- Mentored engineers on verification methodologies, tools, and industry best practices

Individual Contributions:

- **Workaround Management System** [\(details\)](#) - Standardized automated workaround process, made automation generic and reusable across environment, auto-generated Markdown/RST documentation for stale workarounds
- **AI Integration Initiative** [\(details\)](#) - Worked with central engineering to enable internal LLM search engine, internal AI enabled tools, introduced AI editors (GitHub Copilot, Cursor, Windsurf) in the team

KEY PROJECTS (CONTINUED)

Verification Dashboard & Infrastructure [\(details\)](#)

2017 - Present

Sole Developer, Synopsys

- Gen-1 (2017): Built Grafana-based dashboard with MySQL backend, established templates and data structures, actively used and maintained by team
- Gen-2 (2024): Developed Streamlit-based dashboard (Darpan) with Python frontend, added disk caching and extensive documentation using Docusaurus
- Both dashboards in active use by engineers and management up to SVP level to track project health metrics

Other Projects

- **Git Migration Project** [\(details\)](#) - Led Perforce to Git migration for entire ARC IP codebase, zero data loss, minimal downtime, established CI/CD infrastructure (Dec 2023 - Feb 2024)
- **Regression Automation System** [\(details\)](#) - End-to-end automation in use for 10+ years with auto error bucketing, Jira integration, and coverage management. Guided the OGE to LSF migration in team (2012-2016)
- **FSDB Tracer for Debugging** [\(details\)](#) - A tool that generates human-readable instruction traces from FPGA waveforms, bridging the gap between FPGA and RTL simulation

PUBLICATIONS & PRESENTATIONS [\(all\)](#)

Kasula, V. K. (2025). *C-Forge, Efficient Configuration Generator for Configuration Space Verification of ARC Processor IP*. Purple Poster, Synopsys.

Kasula, V. K., & Chedella, S. S. (2016). *Barrier Insertion in Test Programs for Controlled Verification of Cache Coherency*. Synopsys India Technical Conference (SITC), Bangalore.

Kasula, V. K. (2016). *Reverse Generation Technique for Functional Verification of Processor Cores*. Synopsys India Technical Conference (SITC), Bangalore.

Kasula, V. K., & Tadiboina, G. K. (2016). *Test Generation for Processors with Extended Address Capability*. Synopsys India Technical Conference (SITC), Bangalore.

OPEN SOURCE CONTRIBUTIONS [\(all\)](#)

argparse-enh [\(details\)](#) • Python Library, PyPI

Active

- Enhanced argparse with API, shell, and automatic GUI generation modes