CONTACT

vijayakrishnakasula@gmail.com

+91 92931 94921

LinkedIn • GitHub • GitLab

Detailed Resume / CV

PROFESSIONAL SUMMARY

Principal Engineer with 15+ years in processor verification. Led development of 3 generations of Random Program Generators serving 50+ engineers across multiple product lines. Specialized in verification infrastructure, ML/AI applications, and process improvement.

EDUCATION (full)

M.Tech in Data Science & Engineering
BITS Pilani (WILP)
Nov 2019 - Feb 2022 | CGPA: 8.75

B.Tech in Electronics & Communication
SVIT (JNTU Hyderabad)
Jun 2006 - Jun 2010 | 71.40%

AWARDS & RECOGNITION

Harvard University Certificate
Certificate of recommendation from Harvard University, Cambridge, MA, USA, for performance in online coding competition held in collaboration with NASA on TopCoder (2012)

Industry Recognition
Multiple Synopsys awards for technical contributions and innovation

More awards and certifications on LinkedIn

VIJAYA **KRISHNA** KASULA

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

Worked as a contractor from SoCtronics from Jul 2010 - Feb 2014 (full)

KEY PROJECTS (all)

Random Program Generators (RPGs) (details) — Dec 2011 - Present

Lead Developer & Product Manager, Synopsys

- Designed and developed 3 generations of RPGs from scratch for ARC processor verification, each addressing different design philosophies
- G1 (Perl-based, 2011-2020): Fast offline instruction generator for random instruction sequences
- G2 (SystemVerilog-based): State-aware generator with coverage-driven features (Cov2gen) for targeted test generation
- G3 (C-based): Advanced scenario generator with flexible API framework
- Managed product lifecycle end-to-end, supporting 50+ verification engineers across multiple parallel product lines
- Handled complex challenges: multi-issue scheduling, multicore systems, cache coherency, MMU/MPU, VLIW, SIMD
- Impact: Significantly reduced test development time, boosted productivity across product lines

AI-ML Log Anomaly Detector (details) — Nov 2021 - Feb 2022

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

- Developed intelligent log analyzer using scikit-learn, NLTK, NLP, and unsupervised learning algorithms
- Evaluated on 10 diverse datasets with 400M+ reference log lines, tested 36 model combinations
- Implemented advanced techniques: hashing vectorizer, KD-tree, decision tree, random forest, clustering
- Successfully deployed in production at Synopsys, automating verification failure log triaging
- Impact: Reduced manual triaging effort, accelerated failure analysis workflow

C-Forge (details) — 2024 - 2025

Sole Developer, Synopsys

- Designed efficient configuration generator for configuration space verification of ARC Processor IP
- Automated generation of diverse processor configurations covering complex configuration spaces
- Presented at Purple Poster event (Synopsys internal innovation showcase) 2025
- Impact: Improved configuration space coverage, reduced manual configuration efforts

Co-Simulation Environment (details) — Dec 2012 - Mar 2014

Sole Developer, Synopsys

- Designed and implemented robust co-simulation environment for processor-level verification
- Enabled seamless integration between SystemVerilog testbenches and C-based golden reference models
- Developed efficient communication protocols for real-time state synchronization
- Impact: Enhanced verification accuracy through comprehensive state comparison

Verification Dashboard & Infrastructure — 2018 - Present

Technical Lead, Synopsys

- Built comprehensive verification dashboard using Grafana, Plotly-Dash, and Streamlit
- Developed regression automation framework and monitoring infrastructure
- Integrated with GitLab CI/CD for continuous verification workflows
- Created documentation systems using Docusaurus, Sphinx, and Doxygen
- Impact: Improved team visibility, reduced debugging time, enhanced collaboration

SKILLSET

Programming & Scripting 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

Data Visualization:

Streamlit, Plotly Dash, Grafana

Tools & Libraries:

Bison, Flex, Yacc, Graphviz, Dot, Compilers, Argparse

Version Control & CI/CD:

Git, GitLab, GitHub, SVN, Perforce, LSF, Farm

Documentation Tools:

Docusaurus, Sphinx, Doxygen

EDA & Verification Tools:

Verdi, VCS, Specman, ICO, VDS DVE, CSmith, Testbench

Processor Architecture:

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

Processor Verification:

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

Data Formats:

JSON, Parquet, YAML

Others:

Project Dashboard

LEADERSHIP & MANAGEMENT

Verification Best Practices Team (VBPT) **2020 - Present**

Team Lead, Synopsys

- 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
- Drove Git migration project, transitioning team from legacy version control systems
- Mentored engineers on verification methodologies, tools, and industry best practices

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) **Active**

Python Library, PyPI

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

txtoflow (details) **Active**

Python Library, PyPI

- Automatically generates flowcharts from pseudo-code written in C-like languages