2025/08/22

Vu Phan, PhD

Resume

Astrodome, Houston, Texas
vuphan314.github.io
vuphan314
vuphan314

Vu Phan is skilled in **computation research** (data structures, algorithms), **software development** (C++, Java, Scala), **hardware validation** (SystemVerilog, TCL), **data analysis** (Python), and **technical writing** (LaTeX).

- Vu validated semiconductor *Intellectual Property (IP)* as a Formal Verification engineer at Intel Corporation. His PhD in Computer Science was supervised by Professor Moshe Vardi at Rice University.
- Vu is a permanent resident of the United States. He is fluent in English and Vietnamese.

Professional Experience

End 2025/07 **IP Formal Verification Engineer**, *Intel Corporation*, Oregon (hybrid).
Start 2022/10
- Applied pre-Silicon *Formal Property Verification (FPV)* to 2 generations of System-on-Chip designs.
- Checked *Register-Transfer Level (RTL)* implementations against high-level architectural specifications.
- Partnered with RTL designers and system architects to solve problems.
- Found 12 bugs, wrote reports, and validated fixes for functional correctness.
- Co-owned the cache coherence fabric, which orchestrates communication among CPU cores:
  - Error Correction Code in Static Random-Access Memory: reliability/availability/serviceability;
  - Machine Check Architecture: error type, overwriting rule;
  - Power Management: request/acknowledgment handshake, finite state machine, firmware micro-code.
- Used Cadence JasperGold to prove assertions and Synopsys VCS to debug assumptions.
- Used SystemVerilog to develop proofs, TCL to configure tests, and Python to analyze results.

End 2021/08 **IP Formal Verification Engineering Intern**, *Intel Corporation*, Texas (remote).
Start 2021/05
- Used Python to benchmark FPV platforms (Cadence, Synopsys) and analyze their performance.

End 2022/08 **Graduate Research Assistant**, *Rice University*, Texas (onsite).
Start 2017/08
- Read prior works, wrote technical papers, and responded to expert reviewers with informational rebuttals.
- Engineered software to exactly solve stochastic satisfiability:
  - Binary-Decision-Diagram data structures: lossless compressions (C/C++ APIs);
  - Divide-and-Conquer algorithms: scalable solutions (C++ back-end, Python front-end).
- Used C++ with STL to develop efficient solvers and Python with Pandas to analyze large datasets.

Industrial Certification

2025/04 **JasperGold Scoreboard v1.0**, *Cadence Design Systems*.

2025/01 **JasperGold Formal Expert v22.09**, *Cadence Design Systems*.

Higher Education

End 2022/08 **PhD in Computer Science**, Rice University, `https://github.com/vuphan314/phd-thesis`.
Start 2020/01
- Published peer-reviewed conference papers [DPV20a; DPV21] and thesis [Pha22].

End 2019/12 **MS in Computer Science**, Rice University, `https://github.com/vuphan314/ms-thesis`.
Start 2017/08
- Published peer-reviewed conference paper [DPV20b] and thesis [Pha19].

End 2017/07 **Dual BS in Computer Science and Mathematics**, Texas Tech University.
Start 2014/08
- Published peer-reviewed workshop paper [Pha18].

Software Engineering

End 2025/07 **DPO**, *Dynamic-Programming Optimizer*, `https://github.com/vuphan314/DPO`.
Start 2022/02
- Developed, maintained, and documented a SAT framework with weight optimization (C++).

End 2024/08 **DPMC**, *Dynamic-Programming Model Counter*, `https://github.com/vardigroup/DPMC`.
Start 2020/07
- Developed, maintained, and documented a #SAT framework with parallel computing (C++).

End 2024/06 **ADDMC**, *Algebraic-Decision-Diagram Model Counter*, `https://github.com/vardigroup/ADDMC`.
Start 2018/01
- Developed, maintained, and documented a #SAT solver (C++).
- Won the weighted track of the Model Counting Competition 2020 (tied with another solver).

End 2017/08 **LED**, *Language of Effective Definitions*, `https://vuphan314.github.io/LED`.
Start 2016/04
- Developed and documented a translator from LED to SequenceL and LaTeX (Python).

End 2017/01 **L**, *Logic*, `https://github.com/iensen/LtoASPtranslator`.
Start 2015/06
- Developed and documented a translator from L to Answer Set Prolog (Python).
- Invented CertWare Safety Case Workbench (NASA new technology report LAR-18868-1).

Select Coursework

End 2022 **Graduate Level**.
Start 2017
1. Statistical Machine Learning
2. Artificial Intelligence
3. Automated Program Verification
4. Compiler Construction
5. Multi-Core Computing
6. Bioinformatics: Sequence Analysis

End 2017 **Undergraduate Level**.
Start 2015
1. Data Structures
2. Algorithms
3. Object-Oriented Programming
4. Database Systems
5. Operating Systems
6. Computer Organization and Assembly Language

Bibliography

Theses

[Pha22] Vu Phan. "Quantitative Reasoning on Hybrid Formulas with Dynamic Programming". PhD thesis. Rice University, 2022. URL: `https://repository.rice.edu/items/2e464125-244d-431b-b998-612f0dc2b41a`.

[Pha19] Vu Phan. "Weighted Model Counting with Algebraic Decision Diagrams". MS thesis. Rice University, 2019. URL: `https://repository.rice.edu/items/a1a5e73d-a001-44ca-9730-25a7277c8af1`.

Conference Papers

[DPV21] Jeffrey Dudek, Vu Phan, and Moshe Vardi. "ProCount: Weighted Projected Model Counting with Graded Project-Join Trees". In: *Conference on Theory and Applications of Satisfiability Testing*. 2021. URL: `https://kasekopf.github.io/papers/sat21_procount.pdf`.

[DPV20a] Jeffrey Dudek, Vu Phan, and Moshe Vardi. "DPMC: Weighted Model Counting by Dynamic Programming on Project-Join Trees". In: *Conference on Principles and Practice of Constraint Programming*. 2020. URL: `https://arxiv.org/abs/2008.08748`.

[DPV20b] Jeffrey Dudek, Vu Phan, and Moshe Vardi. "ADDMC: Weighted Model Counting with Algebraic Decision Diagrams". In: *AAAI Conference on Artificial Intelligence*. 2020. URL: `https://arxiv.org/abs/1907.05000`.

Workshop Paper

[Pha18] Vu Phan. "Syntactic Conditions for Antichain Property in Consistency Restoring Prolog". In: *Workshop on Answer Set Programming and Other Computing Paradigms*. 2018. URL: `https://arxiv.org/abs/1809.09319`.