Formal Approaches to Characterize Emerging Arithmetic Realizations

Paul Jiang[1*], Iana Lin[2*], Ganesh Gopalakrishnan[3] (Advisor)

[1]Department of Computer Science, Purdue University
[2]Department of Electrical Engineering & Computer Science, UC Berkeley
[3]Kahlert School of Computing, University of Utah
[*]Both authors contributed equally to this research

SC24 Atlanta, GA | hpc creates.

PURDUE UNIVERSITY · Berkeley UNIVERSITY OF CALIFORNIA · THE UNIVERSITY OF UTAH

Research under REU Site NSF 2244492 Trust and Reproducibility Education for Undergraduates

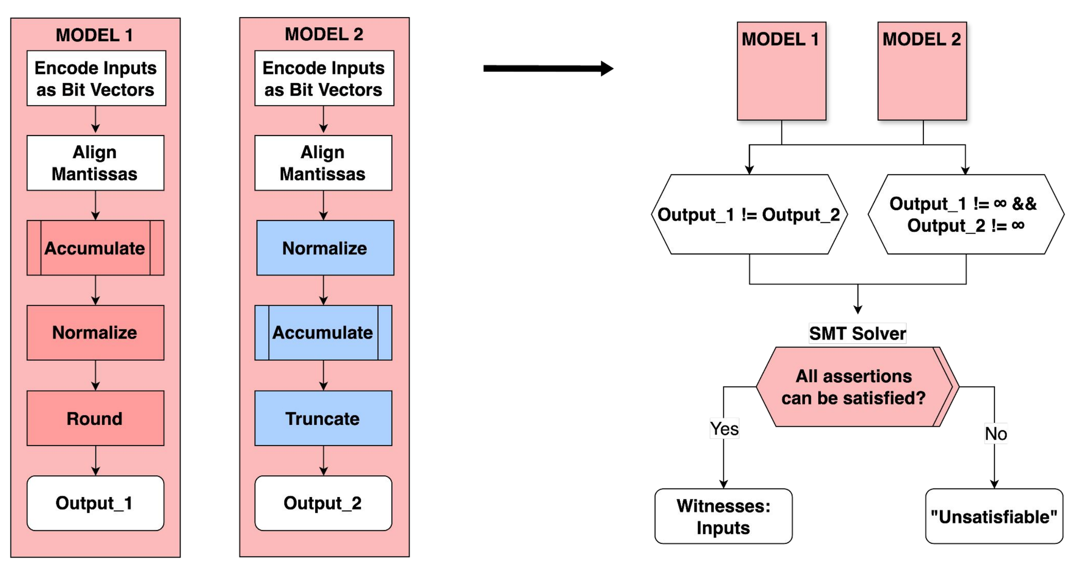
INTRODUCTION

- Advances in AI/Scientific Computing are critically dependent on **efficient computer arithmetic**
  - Alexnet : 0.01 petaflop-days → GPT4 : 10^6 petaflop-days
  - 4 tera-ops in single precision → 4 peta-ops in FP8 (1000x) in 10 years
- Emerging arithmetic schemes significantly deviate (often **undocumented)** from the IEEE standard - e.g. Matrix Accelerators (MA)
  - A single MatMul shown to yield 0s, 255s, or 192s across five MA (Li [3])
  - Automated formal methods based on SMT help create and maintain **unambiguous specifications**, accurately capturing feature **differences**
    - Helps address reproducibility – a concern both in AI ("Grand Illusion" (Mince [4]) and Scientific Computing (e.g. Climate Research)
- **RESEARCH QUESTIONS**
  - Are SMT solvers scalable/expressive in characterizing modern arithmetic realizations?
  - Can downstream tasks (e.g., test generation) benefit from these formal methods?
  - What Floating-Point (FP) properties are implicitly assumed - do they hold upon extrapolation?
  - How best to discover undocumented features (e.g. AI integration)?

PRELIMINARIES

Satisfiability Modulo Theory (SMT) Solvers– Z3 (de Moura, Bjørner [2]) **:**

- SMT is used in formal methods to **automatically analyze the satisfiability of logical formulas** in a variety of theories
  - e.g. generate witnesses ***x, y*** such that **x + y = 10** and **x - 3 = 1** → **4, 6**
- It is rigorous, i.e. mathematical property-guided enumeration of tests can provide enhanced confidence of coverage
  - Solver speed is active research - SMT has sped-up by 100x in a decade
- Key to our work: Leverage Bit-Vector Theories to model non-standard HW
  - FP represented as Sign(S), Exponent(E), Mantissa(M), GRS



METHODS

- **Current Methods** (Fasi, Higham, Mikaitis [1]): **Manual Feature-Test Generation for Physical Hardware**
  - Limited nuanced formalizations of FP numerical behavior
- **Hypothesis:** Use of SMT to **automate feature-discriminating test generation** has several benefits
  - Focus on high-impact arithmetic features for matrix multiplication (see table)
- **Approach**: Use SMT-LIB model arithmetic from first principles
  - Corroborated by findings in (Valpey, Pai [7])
  - Encode desired behavior as SMT constraints

Feature
Subnormal support
Extra precision bits
Rounding modes
Block FMA
FMA unit width
Accumulation order
Intermediate precision
Monotonicity
Product rounding

- **Basis for our work** – Li [3] characterized the arithmetic model based on evidence gathered from limited tests
  - Arithmetic samples administered in normal and subnormal ranges
- We extend test-based understanding to a logic-based general model

RESULTS

SMT-generated test for Subnormal Handling:

Operation	⇒	Subnormal Support	Flush to Zero
$0.0000100000 \times 2^{-15} + 1.0000000001 \times 2^{-14}$	⇒	$1.0000100001 \times 2^{-14}$	$1.0000000001 \times 2^{-14}$

- Verified these inputs and outputs using NVIDIA V100 & AMD MI250X

Assessing bounds of FP properties without direct testing on hardware, e.g. Sterbenz Lemma (Torstensson, Weber [6]), Innocuous Double Rounding (Roux [5])

- Just 1 ulp error was found to violate both these properties
- Our models showed the ability to generate **14,400+ candidate witnesses** for these tests (better coverage) in **<5 minutes**

Revisiting Fasi, Higham, Mikaitis [1]: Lack of intermediate normalization results in non-monotonicity from dot products in Tensor Cores. Proposed this behavior underlies all architecture without partial sum normalization.

Formulated as constraints in our SMT model:

a ⊕ b ⊕ b ⊕ b = c # a, b, c := floating point symbolic variables
|exp(b) - exp(a)| ≥ m # m := mantissa bits + extra precision bits
|exp(b) - exp(a ⊖ b)| ≤ m
Then: a ⊕ b ⊕ b ⊕ b = c < (a ⊖ b) ⊕ b ⊕ b ⊕ b = c ⊕ b ⊕ b

Key insight: Without direct access to hardware, we **formally verified** this non-monotonicity extends beyond FP16 precision to BFloat precision as well.

Additionally, we verified this non-monotonicity to hold on all standard rounding modes and any number of extra precision bits.

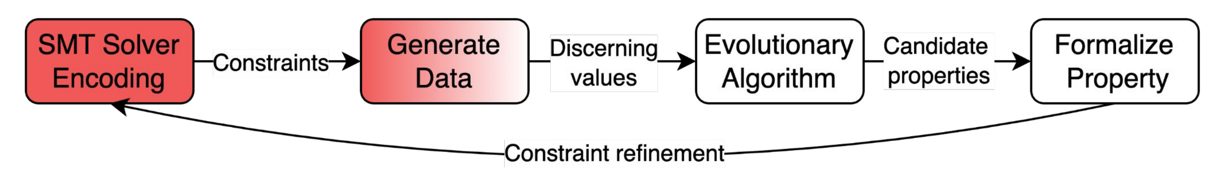
FUTURE DIRECTIONS

Our results demonstrate the ability to **generate a large number of witnesses** within a reasonable time frame for arithmetic features and FP properties

- This supports feasibility in generating **extensive tailored datasets per arithmetic realization for AI-integrated analysis**

Using our SMT-models, we propose a future pipeline for automated property discovery. This pipeline will:

- Analyze complex non-standard FP arithmetic realizations
- Identify key properties and invariants [over varying precisions] to be verified via SMT



By automating property discovery, we can **accelerate the formal verification process** and make it more scalable for the growing diversity of non-standard hardware. This verification is necessary for safely and efficient harnessing the power of novel architectures.

CONCLUSION

Our contributions:

- Validated **software models** of arithmetic features to
  - Automate feature-targeted test generation
  - Verify FP behavior without direct hardware testing

Working Towards:

- Discover FP properties by analysing SMT-generated test data via **genetic programming**

REFERENCES

[1] Fasi M, Higham NJ, Mikaitis M, Pranesh S. Numerical behavior of NVIDIA tensor cores. PeerJ Comput Sci. 2021 Feb 10;7:e330. doi: 10.7717/peerj-cs.330. PMID: 33816984; PMCID: PMC7959640.
[2] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: an efficient SMT solver. In Proceedings of the Theory and practice of software, 14th international conference on Tools and algorithms for the construction and analysis of systems (TACAS'08/ETAPS'08). Springer-Verlag, Berlin, Heidelberg, 337–340.
[3] Li, X.; Li, A.; Fang, B.; Swirydowicz, K.; Laguna, I. & Gopalakrishnan, G. (2024), 'FTTN: Feature-Targeted Testing for Numerical Properties of NVIDIA & AMD Matrix Accelerators.', CoRR abs/2403.00232.
[4] Mince, F., Dinh, D., Kgomo, J., Thompson, N., & Hooker, S. (2024). The Grand Illusion: The Myth of Software Portability and Implications for ML Progress. Advances in Neural Information Processing Systems, 36.
[5] Pierre Roux. Innocuous Double Rounding of Basic Arithmetic Operations. Journal of Formalized Reasoning, 2014, 7 (1), pp.131-142. ff10.6092/issn.1972-5787/4359ff. Ffhal-01091186f.
[6] Torstensson, O., Weber, T. (2023). Hammering Floating-Point Arithmetic. In: Sattler, U., Suda, M. (eds) Frontiers of Combining Systems. FroCoS 2023. Lecture Notes in Computer Science(), vol 14279. Springer, Cham. https://doi.org/10.1007/978-3-031-43369-6_12
[7] B. Valpey. 2023. A Formal Specification of Tensor Cores via Satisfiability Modulo Theories. In SC '23: Student Research Competition Graduate Poster Session, November 14–16, 2023, Denver, CO. ACM, New York, NY, USA, 2 pages.