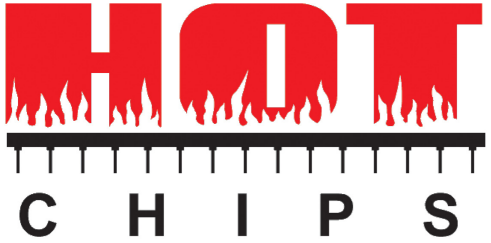


KLIMA: Low-latency mixed-signal In-Memory Computing accelerator for solving arbitrary-order Boolean Satisfiability

Tinish Bhattacharya[1], Dongseok Kwon[1], George Hutchinson[1], Xiangyi Zhang[2], Giacomo Pedretti[3], Fabian Bohm[3], John Paul Strachan[4], Thomas Van Vaerenbergh[3], Ray Beausoleil[3], Ignacio Rozada[2], Dmitri Strukov[1]

[1]University of California Santa Barbara

[2]1QB Information Technologies (1QBit)

[3]Hewlett Packard Labs

[4]RWTH Aachen University & Peter Grunberg Institute (PGI-14), Forschungszentrum Juelich GmbH

Acknowledgement

Contributors


Tinish Bhattacharya


Dongseok kwon


George Hutchinson


Xiangyi Zhang


Giacomo Pedretti


Fabian Bohm


John Paul Strachan


Thomas Van Vaerenbergh


Ray Beausoleil


Ignacio Rozada


Dmitri Strukov

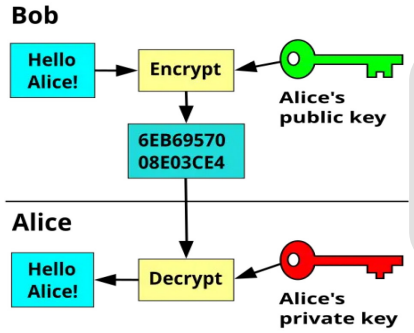
Funding Agency

This project is supported by DARPA Quantum-Inspired Classical Computing (QUICC) Program

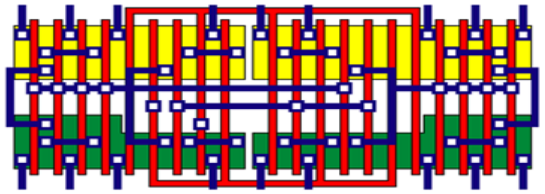
Introduction and Motivation

Hardest and most interesting problems solve some kind of Optimization

- **Learning in biology** and artificial neural networks is an optimization process; as are other problems in several industrial and scientific scenarios as highlighted below (applications in bold **have high-order cost function**).



- ➢ **ML model robustness verification**
- ➢ **Test Pattern Generation**
- ➢ **Cryptography**
- ➢ Logistics
- ➢ Finance
- ➢ **Network/ Circuit Routing**
- ➢ **AI/ Mission Planning**
- ➢ **Gene prediction**
- ➢ **Material design**



- This highlights the need for **high-order optimizers** !!

Solving High-Order Optimization Problems

Second-Order Cost function

$$H = \sum_{i,j=1}^{N} J_{ij} x_i x_j + \sum_{i=1}^{N} b_i x_i$$

Traditional Approach ☹

Arbitrary-Order Cost function

$$H = \sum_{i_1} J_{i_1}^{(1)} x_{i_1} + \cdots + \sum_{i_1 < \cdots < i_k} J_{i_1 \ldots i_k}^{(k)} x_{i_1} \ldots x_{i_k}$$

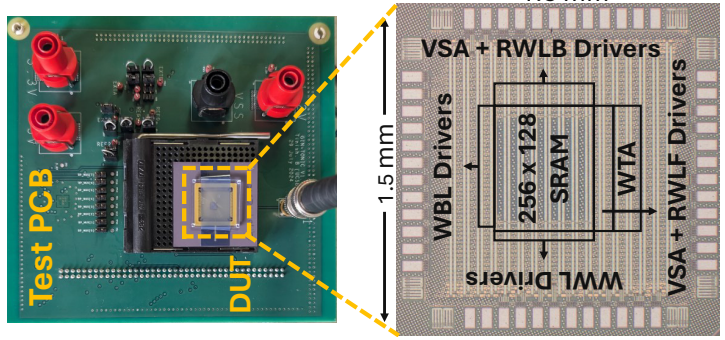
Other Approach ☹

- **Dedicated SAT solvers** that are either restricted to third-order only or are sequential in nature.

- Solvers like **Fujitsu DA, D-WAVE** etc, are limited to second-order interactions and incur orders of magnitude penalty due to order reduction.

Our approach ☺

- Prototype of our **arbitrary-Order solver**, that preserves high-order interactions and accelerates high-order problems like Boolean Satisfiability.



KLIMA Overview

K-Boolean Satisfiability (*K*-SAT) Problem

- Repeated memory access on sparse data makes solving large problems inefficient on Von Neumann machines.

Goal: Find variable set satisfying all clauses in conjunctive normal form

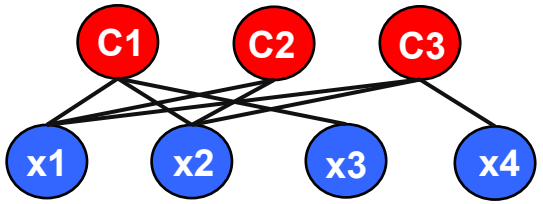
ex: 3-SAT, $f = (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_4} \vee x_2 \vee x_1)$

$(x_1, x_2, x_3, x_4) = (0,1,0,0)$ satisfies the formula, f

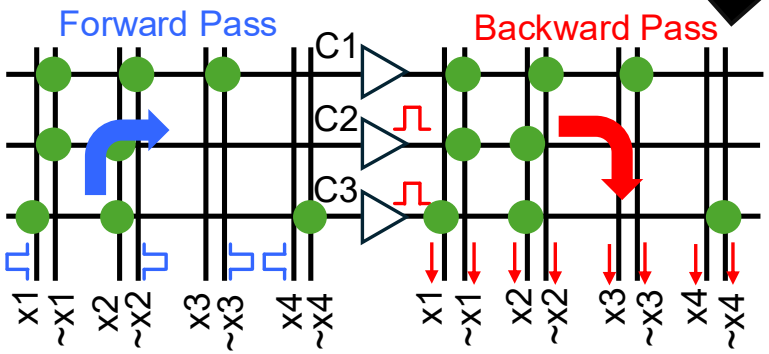
- Current K-SAT solver ASICs/FPGAs have limited performance due to sequential gradient computation.

Key Features of our Hardware

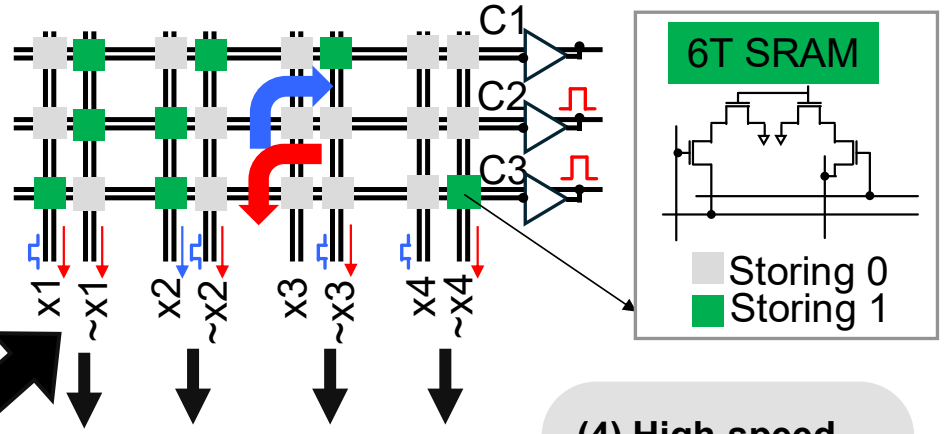
(1) Bipartite Graph-Based Problem Embedding : supports arbitrary k-SAT mapping.



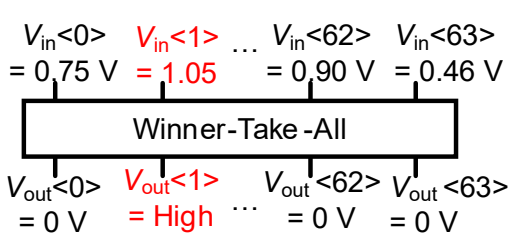
(2) In-Memory Computing: efficient and massively parallel gradient computation [1].



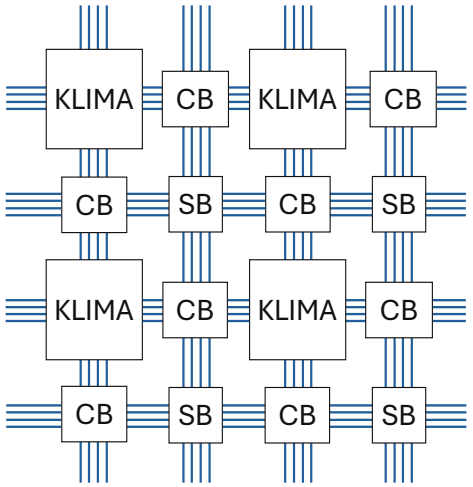
(3) Bi-directional 10T SRAM cell: allows simultaneous forward and backward pass in same array.



(4) High-speed Analog Winner-Take-All: allowing gradient descent with ADC-free sensing.



(5) One-shot parallel gradient computation: enables wide range of heuristics including but not limited to High-Order Hopfield Neural Networks, G2WSAT, High-order Simulated Bifurcation Machine.



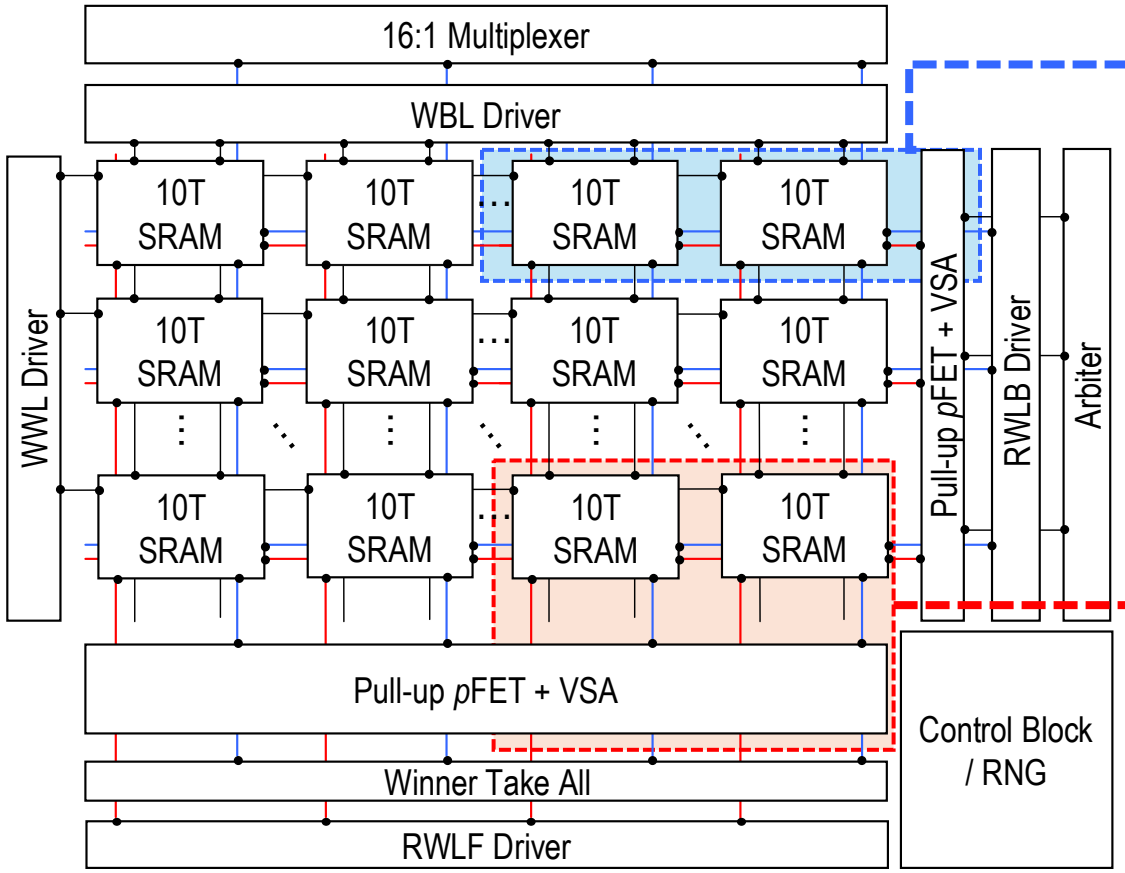
SB: Switch Block
CB: Connection Block

(6) Highly Scalable: enable parallel multi-tile communication with Field-Programmable Ising Array architecture.

[1] Bhattacharya, Tinish. et al., *Nature Communications*, 2024.

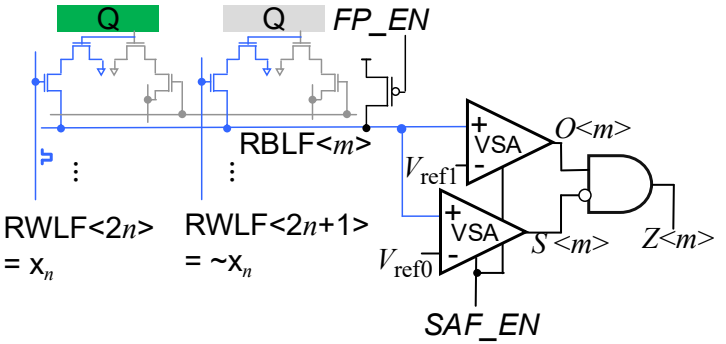
Chip Architecture

System-Level Block Diagram

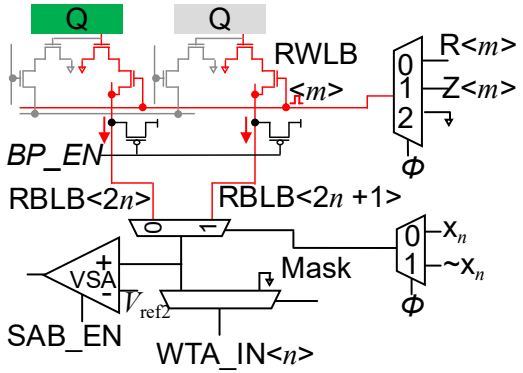


- Forward Pass (FP) VMM evaluates all clauses in parallel.
- Backward Pass (BP) VMM computes gradients of all variables in parallel.
- Arbiter is used to select one clause/variable at random out of a set of valid ones.
- Winner-Take-All selects variable with most promising gradient.

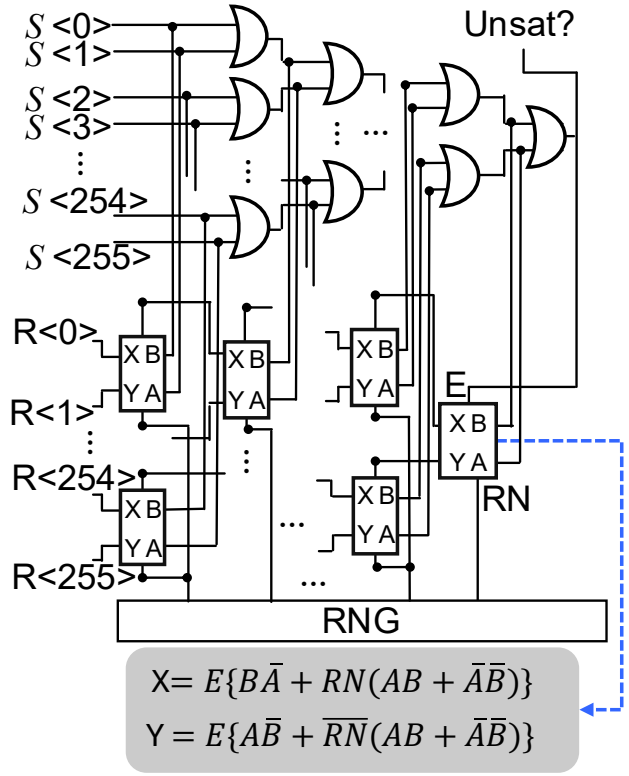
Forward Pass Operation



Backward Pass Operation

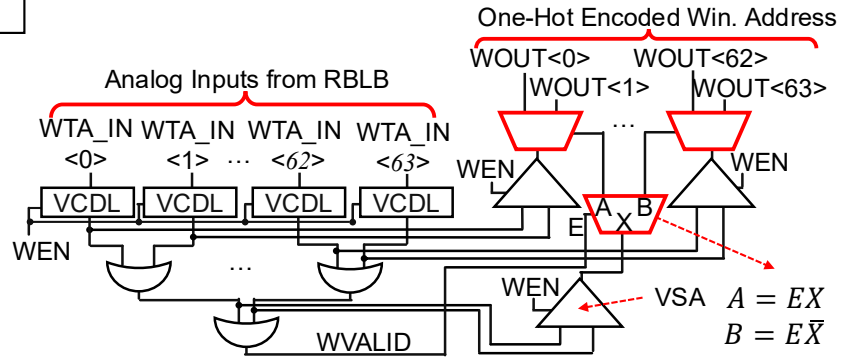


Arbiter

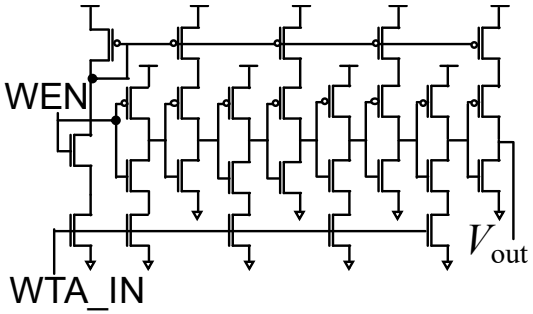


$$X = E\{B\bar{A} + RN(AB + \bar{A}\bar{B})\}$$
$$Y = E\{A\bar{B} + \overline{RN}(AB + \bar{A}\bar{B})\}$$

Winner-Take-All



Voltage Controlled Delay Line

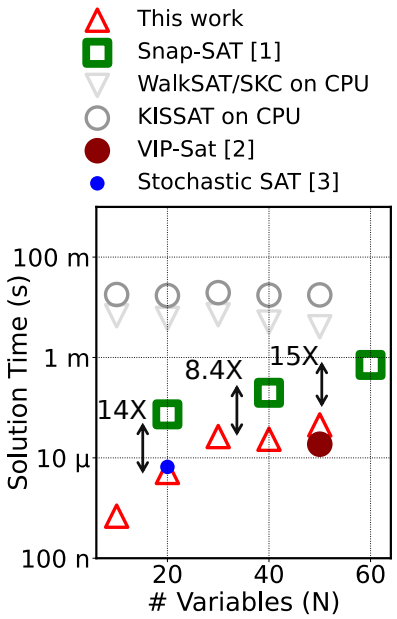


[1] Bhattacharya, Tinish. et al., *VLSI*, 2025.

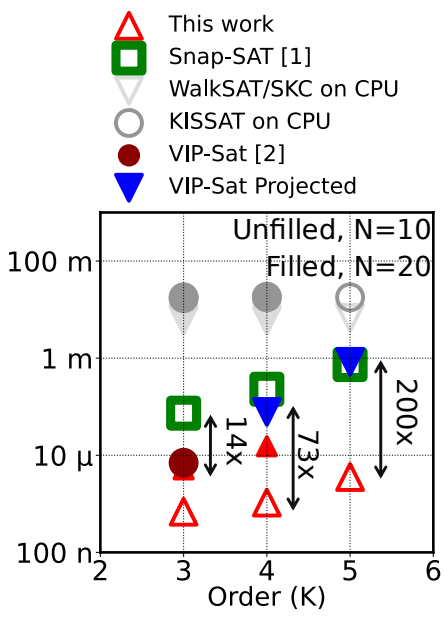
Measurement Results

Time to Solution

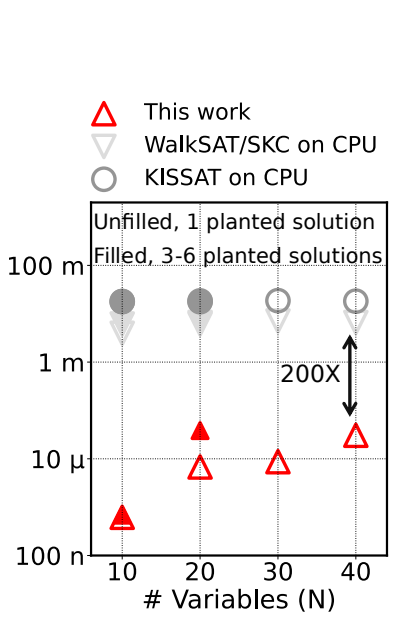
Uniform Random 3-SAT



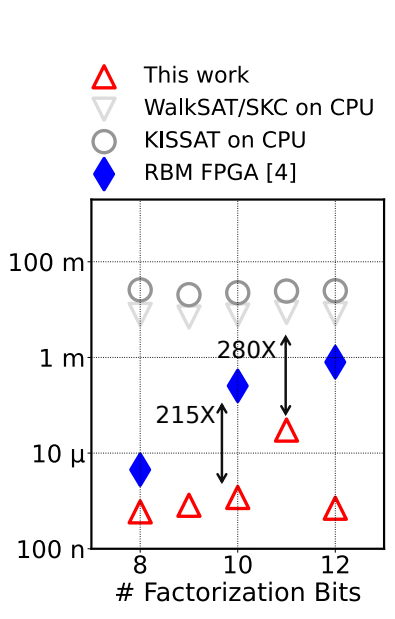
Uniform Random K-SAT



Planted Hard 4-SAT

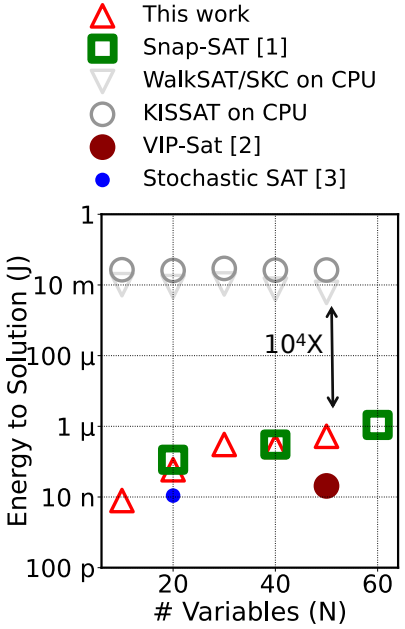


Semiprime Factoring

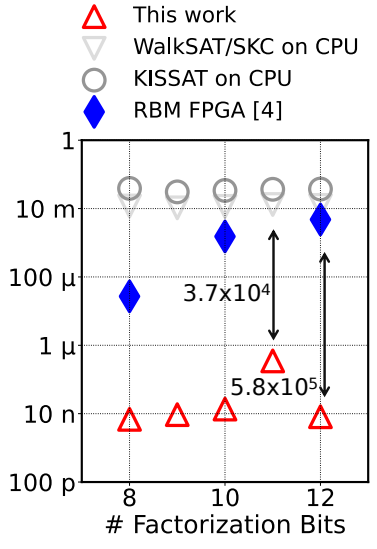


Energy to Solution

Uniform Random 3-SAT



Semiprime Factoring



[1] Xie, Shanshan, et al. *ISSCC*, 2023. [2] Shim, Chaeyun, et al., *ISSCC,* 2024. [3] Zhang, Qiaochu, et al, *VLSI,* 2024. [4] S. Patel, *Nat Elec*, 2022. [5] Hizzani, M., et al., *ISCAS*, 2024. [6] Bhattacharya, Tinish. et al., *VLSI*, 2025.

Summary

- 10x faster than single-variable update ASIC (Snap-SAT [1]) on uniform random 3-SAT.
- Solution times at par with Stochastic SAT [3] and VIP-Sat [2], despite the latter two solvers implementing multiple-variable update per iteration, while our solver updates a single variable per iteration. Multi-variable update heuristics exhibit faster convergence and can be readily adopted using KLIMA architecture in future prototypes [5].
- 10x to 200x faster than ASIC/FPGAs on K-SAT problems with K ≥ 4.
- 73x and 200x faster than current 3-SAT-specific solvers (VIP-Sat) on 4- and 5-SAT problems.

Summary and Outlook

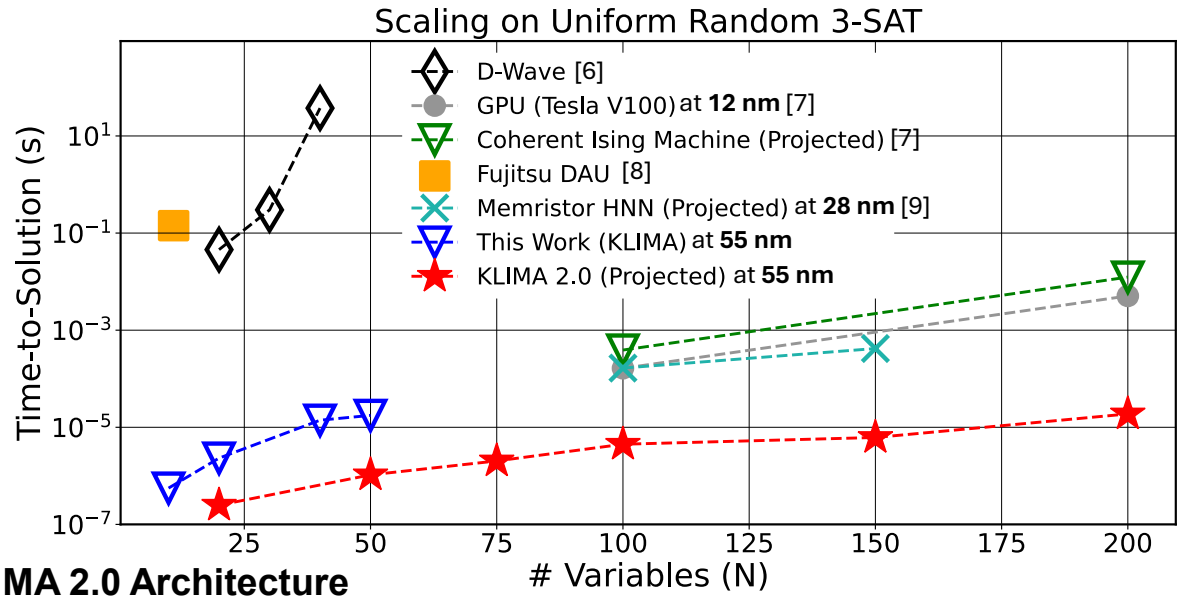
Comparison Table on Uniform Random 3-SAT

Specifications	Stochastic SAT [1]	VIP-Sat [2]	Sci Rep 2024 [3]	Snap-SAT [4]	KLIMA (This work)
Technology	65nm	65nm	65 nm	65nm	55nm
Type	Analog	Digital	Digital	Digital	Mixed
Architecture	Stochastic CT	Near-Memory	Ring Oscillator	SRAM In/Near-Memory	SRAM In-Memory
Flips / iteration	Multiple	Multiple	Multiple	Single	Single
Chip Area (mm^2)	0.37	1.115	1.8	0.93	0.544
Maximum order	3	3	2	2 - 128	64
# Variables (N)	20	50	20	128	64
# Clauses (M)	91	218	91	1024	256
Solvability	100%‡	100%‡, 98% ‼	100%‡	72%◊	100%‡, 98%‼
Solution Time	6.6 us‡	7 us‡, 18.7 us ‼	15.7 ms‡	70 us‡, 710 us◊	5.12 us‡, 45 us‼
Solution Energy	11 nJ‡	20.8 nJ‼	0.15 mJ‡	1098 nJ◊	59 nJ‡, 518 nJ‼

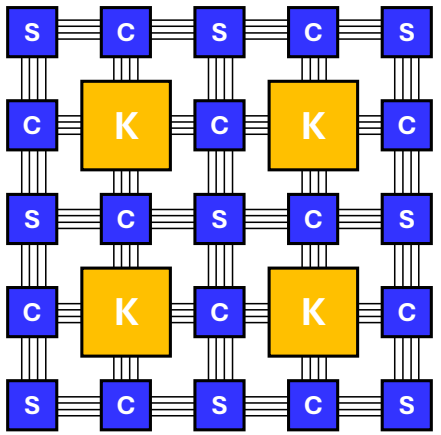
‡N=20, ‼N=50, ◊N=60, where N is # of variables.

- Table shows comparison of KLIMA with select SAT solvers on 3-SAT problems only.
- Other SAT solvers either lack support for parallel gradient computation (Snap-SAT) or are limited to 3-SAT problems only (Stochastic SAT and VIP-Sat). The latter are competitive on 3-SAT by virtue of multi-variable update heuristics.
- Multi-variable updates can be readily implemented in KLIMA architecture with peripheral modifications [9].
- Leveraging mature CMOS/SRAM technologies, KLIMA is inherently scalable and suitable for immediate deployment in real-time applications.

Scaling Up with KLIMA 2.0



KLIMA 2.0 Architecture



S = Switch Block
C = Connection Block
K = KLIMA Tile

- KLIMA 2.0 is a multi-tile architecture where each tile is a mixed-signal in-memory computing core (equivalent to current version of KLIMA).
- Inter-tile communication is established with FPGA-inspired routing fabric comprising point-to-point interconnect, connection and switch blocks [5].
- Any-to-any connectivity enabled by such architecture allows scaling up to thousands of variables (11K variables in 1 cm^2 at 55 nm) in a single chip without requiring any decomposition [5].
- Performance of KLIMA 2.0 is modeled at 55 nm but using multi-variable update heuristics.

[1] Zhang, Qiaochu, et al, *VLSI,* 2024. [2] Shim, Chaeyun, et al., *ISSCC,* 2024. [3] Cılasun, Hüsrev, et al., *Scientific Reports*, 2024. [4] Xie, Shanshan, et al. *ISSCC*, 2023. [5] Bhattacharya, Tinish, et al. *ISVLSI*, 2024. [6] Sharma, Anshujit et al., *Scientific Reports*, 2023.

[7] Reifenstein, Sam. et al., *Advances in Optics and Photonics,* 2023. [8] Munch, Christian, et al., *arxiv preprint arXiv: 2312.11645*, 2023. [9] Hizzani, M., et al., *ISCAS*, 2024.