An Analog Clock-free Compute Fabric base on Continuous-Time Dynamical System for Solving Combinatorial Optimization Problems

Muya Chang[1], Xunzhao Yin[2], Zoltan Toroczkai[3], Xiaobo Hu[3], Arijit Raychowdhury[1]

[1]Georgia Institute of Technology, [2]Zhejiang University, [3]University of Notre Dame

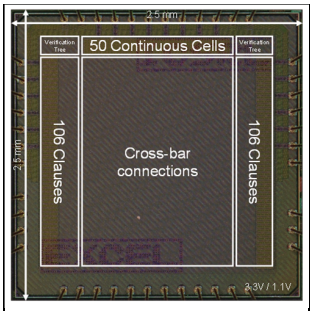
NP-hard combinatorial optimization problems (COPs) are known to be very complex and expensive to solve with traditional computers. COPs can be mapped onto many different kinds of architecture, such as fully digital [1-3], event based [4], oscillators [5-6], Optical LASER [7], Qbits, or pure analog approach [8-9]. We propose a scalable pure analog clock-free continuous-time dynamical system to solve COPs in hardware.

Many well-known real-world applications can be modeled as combinatorial optimization problems (COPs), including autonomous vehicles, EDA tools, supply chain optimization, or flight network scheduling. A recently proposed deterministic, continuous-time dynamical system (CTDS) [8-9] shows promising results to solve COPs as shown in Fig. 1. The mathematical equations behind the system are precisely mapped onto the hardware which minimizes the energy of the system and the final state is a solution. Boolean satisfiability (k-SAT) is canonical NP-complete problems and is the problem of determining if there exists an interpretation that satisfies a given boolean formula. In this paper, we use 3-SAT as our template problem to demonstrate two things through the measurement on a 65nm CMOS hardware. 1) when the constraint density is beyond a certain threshold, the analog trajectories become transiently chaotic. 2) This CTDS shows polynomial analog time-complexity but with the expense of exponential fluctuations in its energy function. This system architecture can map up to 50 variables and 212 clauses and is highly modular and programmable for handling different problem specifications.

Fig. 2 shows both the high-level system architecture as well as the floorplan of the testchip. The system contains four components: (1) 50 continuous cells, which hold the dynamic and analog values of each continuous variable before the system converges. Each continuous cell is constructed with two back-to-back inverters and control gates. (2) 212 clauses, which contribute to the dynamic behavior of the system and respond to the connected continuous cells. (3) 50 x 212 cross-bar connection arrays, each cross point contains a pair of switches. The array controls the connectivity between 212 clauses and 50 continuous cells. (4) Verification tree, which hierarchically checks if all clauses are satisfied.

Fig. 3 shows an example mapping from a classical 3-SAT problem onto the proposed system. In the example problem, clause 1 is the disjunction of S8, S10, and S2; clause 2 is the disjunction of S2, S8, and S1 … etc, and after mapping the corresponding cross points are closed. If we take a closer look at S10 specifically, clauses 1, 41, and 42 are all constrained by either S10 or its negation, therefore while some of them are trying to pull S10 high, some of them will try to pull it low. The comparison between the proposed architecture and a conventional CPU running C-code dynamics solver was also conducted, each problem size N was tested with at least 1000 problems. From the result the proposed architecture not only is comparable when the size of the problems is small, it's at least 10 times faster when the problem size is reasonable large.

Fig. 4 shows the detail circuit of each clause which contains (1) Auxiliary variable circuit (AVC) implementing the dynamics of time-dependent auxiliary variables. If none of the three continuous variables in the clause is satisfied, the auxiliary variable will keep increasing and eventually dominates the clause dynamic. (2) Signal dynamic circuit (SDC) implementing the dynamics of one continuous variable. This block can be further separated into two parts, the first part takes in three continuous variables and try to reach the consensus, the second part takes in the auxiliary variable which would eventually dominates as time elapses if the clause is not satisfied. The first part could be treated as the main operator and the second part as the safeguard to prevent the system from getting stuck at local minima. (3) Digital verification circuit (DVC) simply checking whether the current clause is satisfied.



Die micrograph.

Fig. 5 shows a measured transient behavior of an example problem. In the waveform three example continuous variables, system convergence signal, and reset signal are plotted from top to bottom respectively. After the hardware is programmed, we release the system and wait for the convergence, we can see from the figure that once the system converges, all the continuous variables are converged to either zero or one.

Even with the same problem size, some problems are intrinsically more difficult than the others, thus we are particularly interested in how the continuous-time dynamic system behaves visually in the 3D space. To do so we conducted a batch of measurements on the hardest benchmark k-SAT and sort the transient data based on their convergence time. We then plot four different level of convergence time in the 3D space, as shown in Fig. 5. For each 3D figure, the axis are the values of each of the three continuous variables, and the color represents normalized time stamp throughout the transient behavior. We can see that while for easy problems the system quickly reaches the convergence point, for medium-level problems the system searches in the space for a while. For some difficult problems the system takes a long time searching through the space but eventually reaches the convergence point. However, for some extreme cases the system doesn't reach the convergence and instead randomly search in the space chaotically.

Fig. 6 demonstrates the measured data over 1000 problems from the standard benchmark k-SAT for each problem size. First plot demonstrates the relationship between the chaotic level and the problem difficulty, it can been seen that the trend of the chaotic level follows the trend of the convergence time throughout different problem difficulty. Second plot shows the relationship between the ratio of clause/variable (M/N) and the convergence time, we can see that as the ratio of M/N increases from 0.2 to 4.2, the convergence time dramatically increases, which is expected as problems with constraint density M/N = 4.25 are proved to be the hardest problems. The third plot demonstrates the solvability for each difference problem sizes. While we notice a dramatic solvability degradation after N=30, the main reason causing this is that once the auxiliary variable saturates, the system enters the phase where it starts randomly searching in the solution space and since it is not strictly obeying the mathematical equations, it is not guaranteed to find the solution. To prove this, we show that by increasing the core voltage from 1 volt to 1.2 volts, since the physical limitation of the auxiliary variable is slightly relieved, some very hard problems become solvable and thus the solvability slightly improved.

We compare prior works with our design in Fig. 6. This is the first design that relies of a system of coupled differential equations to solve the canonical SAT problem.

Acknowledgements:

MC and AR were supported by the Semiconductor Research Corporation under the Center for Brain-Inspired Computing (C-BRIC) under Grant 2777.005 and 2777.006.

References:

[1] K. Yamamoto et al., ISSCC, 2020.

[2] Y. Su et al., ISSCC, 2020.

[3] T. Takemoto et al., ISSCC, 2021.

[4] H. Mostafa et al., Nat Commun, 2015.

[5] I. Ahmed et al., VLSI, 2020.

[6] S. Dutta et al., IEDM, 2019.

[7] D. Pierangeli et al., PRL, 2019.

[8] M. Ercsey-Ravasz et al., Nature Phys, 2011

[9] X. Yin el al., TVLS, 2018.

2022 IEEE Custom Integrated Circuits Conference (CICC) | 978-1-6654-0756-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/CICC53496.2022.9772850

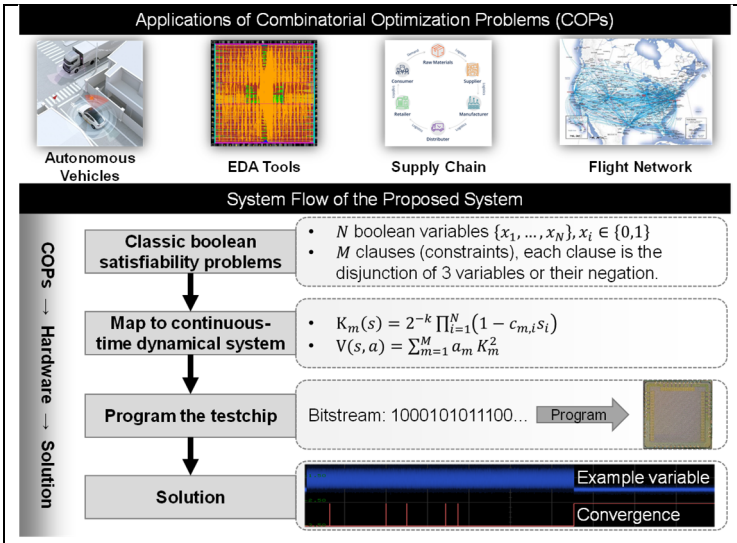


Fig. 1. Applications of COPs and system flow of the proposed system

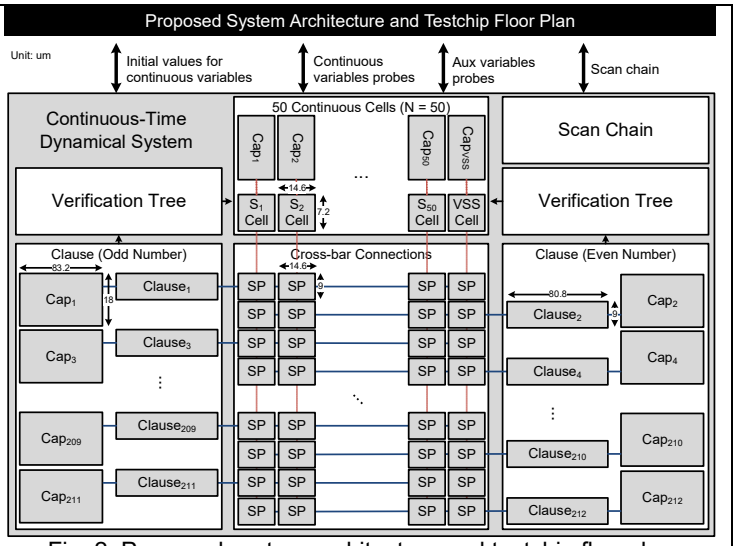


Fig. 2. Proposed system architecture and testchip floorplan

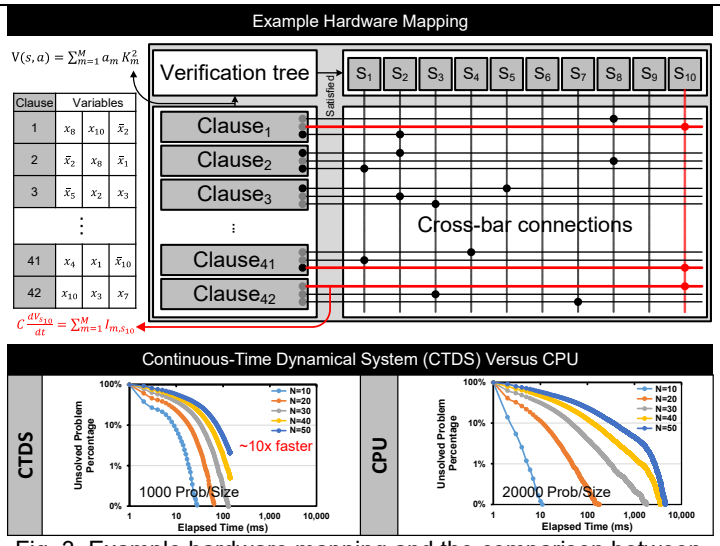


Fig. 3. Example hardware mapping and the comparison between CTDS and CPU.

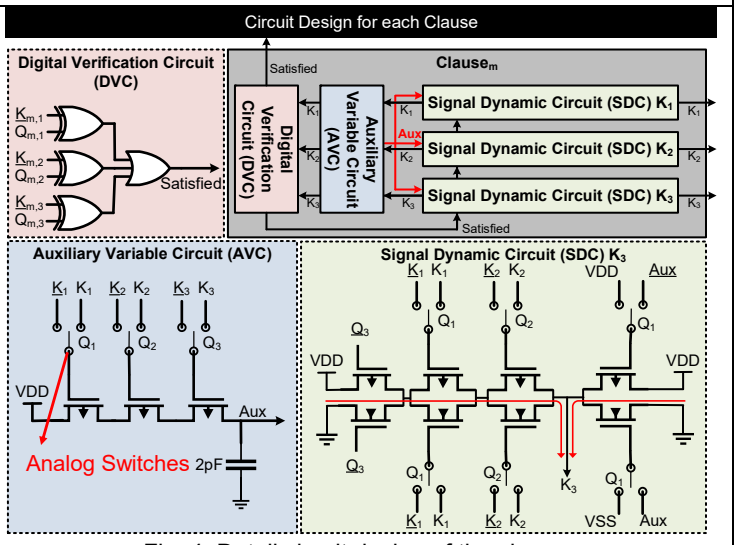


Fig. 4. Detail circuit design of the clauses

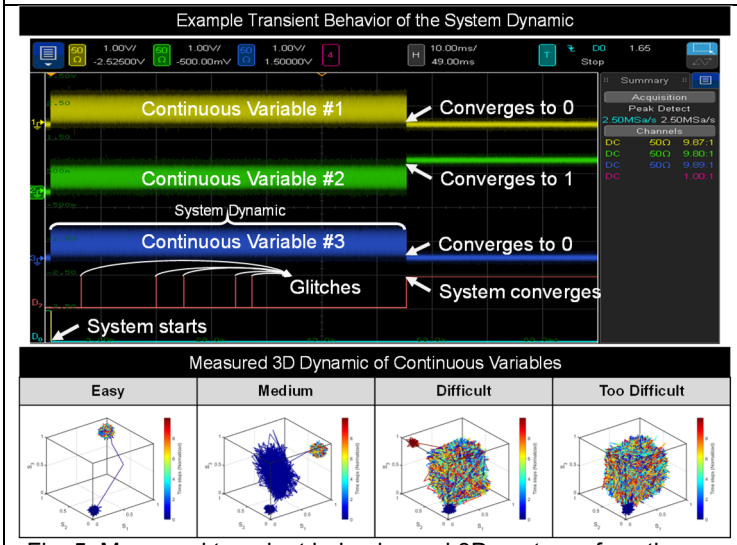
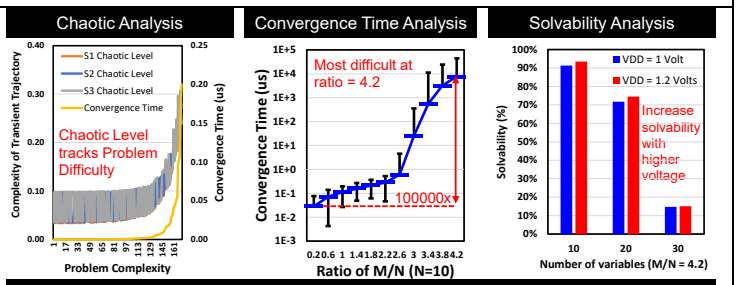


Fig. 5. Measured transient behavior and 3D contour of continuous variables



Comparison Table

	This work	VLSI '20 [5]	IEDM '19 [6]	PRL '19 [7]	D-Wave
Architecture	Cross-bar	Ring oscillators	IMT Oscillators	Optical LASER	Qbits
Special process	No	No	Yes	Yes	Yes
Annealing method	Continuous-time dynamic	Oscillator interaction	IMT stochasticity	Spatial light modulation	Quantum interaction
Technology	65nm CMOS	65nm CMOS	IMT device	Photonics	Superconductor
Operating temperature	Room temp.	Room temp.	Room temp.	Room temp.	-273.14°C
Measured solutions	200	1000	1	16	N/A
Peak power	27mW	23mW	N/A	N/A	25KW
Measured accuracy	93.5%	Easy: 98-100%* Difficult: 82-100%*	97.6% (4 spins)	>95% (87% cases)	N/A

Fig. 6. Measured chaotic analysis, convergence time analysis, solvability analysis, and comparison table.