Digital Emulation of Oscillator Ising Machines

Shreesha Sreedhara1, Jaijeet Roychowdhury1, Joachim Wabnig2 and Pavan Srinath3
1University of California at Berkeley, US; 2Nokia Bell Labs, GB; 3Nokia Bell Labs, FR

***Abstract*—Ising problem is an NP-hard combinatorial optimization problem. Recently, networks of mutually coupled, nonlinear, self-sustaining oscillators known as Oscillator Ising Machines (OIMs) were shown to heuristically solve Ising problems. The phases of the oscillators in OIMs can be modeled as systems of Ordinary Differential Equations (ODEs) known as Generalized Kuramoto (Gen-K) models. In this paper, we solve Gen-K ODE systems efficiently using cleverly designed fixed point operations. To demonstrate this idea, we fabricated a prototype chip containing 33 spins with programmable all-to-all connectivity. We test this design using Multi-Input Multi-Output decoding problems, and show that the OIM emulator achieves near-optimal Symbol Error Rates (SER).**

***Index Terms*—Ising, Kuramoto, oscillators, OIM, emulation, fixed point, MIMO**

I. Introduction

Ising problem is an NP-hard combinatorial optimization problem [1, 2]. The objective is to minimize a cost function with terms of the form $\mathrm{J}_{i,j}s_is_j$, where $\mathrm{J}_{i,j}$ is a real number, and s_i and s_j are binary variables which can be either -1 or $+1$. Note that the 's_i's are known as spins, and the cost function is known as the Ising Hamiltonian.[1]

Recently, it was shown that networks of mutually coupled, nonlinear, self-sustaining oscillators known as Oscillator Ising Machines (OIMs) heuristically solve Ising problems [3]. OIMs in fact minimize a Lyapunov function that is strongly related to the Ising Hamiltonian of the couplings between the oscillators [3].

A major hurdle to realize well performing Oscillator Ising Machines is implementing programmable couplings that are immune to variations of components. We circumvent this issue by emulating OIMs digitally. It is known that the phases of the oscillators in OIMs can be modeled as systems of Ordinary Differential Equations (ODEs) known as Generalized Kuramoto (Gen-K) models [3–5]. Thus, we emulate OIMs by directly solving the underlying ODEs in hardware.

To demonstrate this idea, we taped-out a prototype integrated circuit with 33 spins and all-to-all programmable connectivity in TSMC-65nm process. We test this chip on Multi-Input Multi-Output (MIMO) decoding problems (which have known mappings to equivalent Ising forms [6]), and show that it achieves near-optimal Symbol Error Rates (SERs) [7].

The rest of the paper is organized as follows. In Section II, we present techniques to efficiently solve Gen-K ODE systems in hardware by exploiting fixed point operations. This is followed by a description of the exact algorithm implemented in the prototype integrated circuit (Section III). The layout and the results of the tests on MIMO problems are provided in Section IV. We conclude the paper in Section V.

[1]The Ising Hamiltonian is of the form $(-1/2)\sum_{i,j}\mathrm{J}_{i,j}s_is_j$, where the variables are the same as defined above. We assume that $\mathrm{J}_{i,i}=0$, and $\mathrm{J}_{i,j}=\mathrm{J}_{j,i}$ for all i,j.

II. Efficiently Solving Gen-K ODE Systems Using Fixed Point Operations

As stated in Section I, the phases of the oscillators in OIMs can be modeled using Gen-K ODE systems. They are of the form ($i \in \{1,2,\ldots,N\}$)

$$\frac{\mathrm{d}\phi_i(t)}{\mathrm{d}t} = -\sum_{j=1}^{N}\mathrm{J}_{i,j}\cdot\mathrm{F_c}\big(\phi_i(t)-\phi_j(t)\big) - \mathrm{F_s}\big(\phi_i(t)\big)\ . \quad (1)$$

Here, $\phi_i(t)$ are the phases of the oscillators, and $\mathrm{J}_{i,j}$ are elements of $\mathbf{J}$, the coupling matrix of the spins.[2] Moreover, $\mathrm{F_c}\big(.\big)$ is a certain 'coupling function' that operates on the differences of the phases, and $\mathrm{F_s}\big(.\big)$ is a suitably chosen 'synchronization function' that forces the phases to be binarized (*i.e.*, forces $2\pi\phi_i$ to be a multiple of π) [3]. For example, $\mathrm{F_c}\big(\psi\big) \triangleq \sin\big(2\pi\cdot\psi\big)$, and $\mathrm{F_s}\big(\phi\big) \triangleq \sin\big(2\pi\cdot 2\phi\big)$.

We solve the above ODE system using a method called as Forward Euler (FE) [8]. Given a time step parameter h, we evaluate

$$\phi_i \leftarrow \phi_i - \sum_{j=1}^{N} h\mathrm{J}_{i,j}\cdot\mathrm{F_c}\big(\phi_i-\phi_j\big) - h\mathrm{F_s}\big(\phi_i\big)\ , \quad (2)$$

for a suitably chosen number of iterations.[3] To efficiently solve the above equation, we define $\mathrm{F_c}\big(\psi\big) \triangleq +1$ if $(\psi \bmod 1) < 0.5$, and -1 if $(\psi \bmod 1) \geq 0.5$. Moreover, we define $\mathrm{F_s}\big(\phi\big) \triangleq -1$ if $(\phi \bmod 0.5) < 0.25$, and $+1$ if $(\phi \bmod 0.5) \geq 0.25$.

The above functions can be easily evaluated using fixed point formats [9]. Let ϕ_i in (2) be represented as an n-bit number of the form $0 \bullet b_1b_2\ldots b_n$, where the fixed point is placed before the most significant bit (MSB) (*i.e.*, b_1). Note that the decimal equivalent of the above form is $\sum_{i=1}^{n} b_i 2^{-i}$. It ranges from 0 (when $\forall i,\ b_i=0$) to $1-1/2^n$ (when $\forall i,\ b_i=1$). Hence, it is clear that the above fixed point format stores $(\phi_i \bmod 1)$ with n bits of precision.

Let $\psi \triangleq \phi_i - \phi_j$. Note that $(\psi \bmod 1) = ((\phi_i-\phi_j) \bmod 1) = ((\phi_i \bmod 1) - (\phi_j \bmod 1)) \bmod 1$. Thus, $(\psi \bmod 1)$ is merely the fixed point subtraction of the two phases. Moreover, denoting the MSB of ψ as $b_1(\psi)$, it can be easily verified that $\mathrm{F_c}\big(\phi_i-\phi_j\big) = \mathrm{F_c}\big(\psi\big) = +1$ if $b_1(\psi)=0$, and -1 if $b_1(\psi)=1$. The above can be extended to $\mathrm{F_s}\big(.\big)$ as well. Denoting the penultimate MSB of ϕ as $b_2(\phi)$, we have $\mathrm{F_s}\big(\phi\big) = -1$ if $b_2(\phi)=0$, and $+1$ if $b_2(\phi)=1$. We can thus efficiently evaluate the RHS of (2) using a fixed point format.

III. Designing a Custom Integrated Circuit to Solve Gen-K ODE Systems

In this section, we focus on designing an integrated circuit to digitally emulate a 33-spin OIM system with all-to-all

[2]We assume that the diagonal entries of $\mathbf{J}$ are zero.

[3]Note that the RHS in (2) must be calculated for all i before the phases are updated.

978-3-9819263-7-8/DATE23/© 2023 EDAA

Algorithm 1: Euler's method implemented on the prototype chip.

```
// The following pseudocode is run thrice concurrently (once per set).
// Let p ∈ {0, 1, 2} be the unique index of each set.
// Unless stated otherwise, all the variables are local to the set.
φ⃗_set,p ← rand(N/3)   // Initialize the phases of the set to random numbers
φ⃗_set,q ← φ⃗_set,p     // φ⃗_set,q holds the phases of S_q, initialized to φ⃗_set,p
φ⃗_next,p ← φ⃗_set,p    // φ⃗_next,p holds the RHS of (2) for i ∈ S_p
t ← 0                  // Variable to keep track of time
while t < t_stop do    // Repeat until t reaches a predefined parameter t_stop
    for clk ∈ {0, 1, 2} do
        // Note: one iteration of the loop executes in one clock in hardware.
        φ⃗_next,p ← φ⃗_next,p + {Σ^(i)_{j∈S_q} : i ∈ S_p}
        // Note: the above command uses φ⃗_set,p and φ⃗_set,q
        if clk ≠ 2 then // Computation of RHS of (2) is not yet complete
            φ⃗_set,q ← φ⃗_set,q of S_{(p+1) mod 3}
            // Get the 'next' set of phases; it is not local to the p^th set
        else // RHS of (2) has been computed, result is available in φ⃗_next,p
            φ⃗_set,p ← φ⃗_next,p, φ⃗_set,q ← φ⃗_next,p   // Store the phases
            t ← t + h   // h is the time step parameter
```

connectivity. Essentially, we initialize the phases to random numbers, then repeatedly evaluate (2) for many time steps.

Note that we distribute the computation required for one time step over 3 clock cycles to save on die area. First, we divide the N $(= 33)$ phases into three sets S_0, S_1, and S_2 with $N/3$ $(= 11)$ phases each. By abusing the notation, (2) can be rewritten as $\phi_i \leftarrow \phi_i + \Sigma^{(i)}_{j\in S_0} + \Sigma^{(i)}_{j\in S_1} + \Sigma^{(i)}_{j\in S_2}$, where $\Sigma^{(i)}_{j\in S_q}$ are terms that couple ϕ_i to the phases of S_q.[4] Thus, we divide the coupling matrix into 9 blocks as shown in Fig. 1. In each clock cycle, set S_p calculates $\Sigma^{(i)}_{j\in S_q}$ for all $i \in S_p$ as shown in Fig. 1. The above idea is concretized in Alg. 1. We omit its detailed explanation for brevity.

IV. Layout, and Measurement Results

The design described in the previous section was taped-out in TSMC 65-nm process. The layout of the prototype chip is shown in Fig. 2, it occupies an area of $1.5 \times 2.0 = 3\,\mathrm{mm}^2$.

We evaluate the prototype chip using MIMO decoding problems, generated as explained in [10]. Here, channels of closely spaced users are assumed to be correlated; this is considered as a better approximation in real-world scenarios than conventional Rayleigh Fading models [7].

[4]There are in fact $(N-1)$ coupling terms (since $\forall i, \mathrm{J}_{i,i} = 0$) and 1 synchronization term in (2). However, we merely consider the synchronization term to be the $(i,i)^{\text{th}}$ coupling term for the ease of exposition.

$\mathbf{J}$	$\mathbf{S_0} = S_p$	$\mathbf{S_1} = S_q$	$\mathbf{S_2} = S_q$	
$S_p = \mathbf{S_0}$	clk_0	clk_1	clk_2	$\frac{N}{3}$
$S_p = \mathbf{S_1}$	clk_2	clk_0	clk_1	$\frac{N}{3}$
$S_p = \mathbf{S_2}$	clk_1	clk_2	clk_0	$\frac{N}{3}$
	$N/3$	$N/3$	$N/3$	

Fig. 1: A map of the blocks of $\mathbf{J}$ used in various clock cycles.

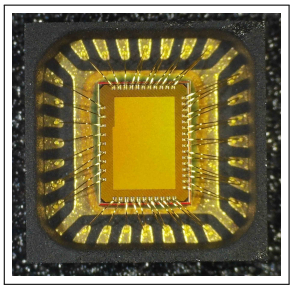

Fig. 2: The layout of the chip.

Fig. 3 shows SER vs Signal to Noise Ratio (SNR) plots of the OIM-emulator, as well as other decoders such as Zero-Forcing Equalization (ZF), Linear Minimum Mean Squared Error (LMMSE), and sphere decoder (an exact algorithm) [7, 11].[5] It is evident that the OIM-emulator achieves near-optimal SERs. Note that the chip consumes approximately $300\,\mathrm{mW}$ (at $120\,\mathrm{MHz}$ clock) when it is busy, and we spend about $120\,\mu\mathrm{J}$ to 'solve' a given MIMO decoding problem.

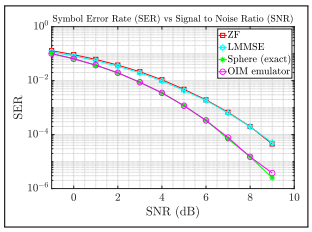

Fig. 3: SER vs SNR plots of many MIMO decoders.

V. Conclusion

We presented a novel approach that digitally emulates a programmable OIM using fixed point operations. Unlike analog OIMs, this emulator is nearly immune to variation of components and performs optimally. This prototype acts as a baseline for future analog/digital OIM designs that might trade off attributes such as quality of solutions, number of bits of programmability, energy per solution, *etc.*

References

[1] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, 1925.

[2] F. Barahona, "On the computational complexity of Ising spin glass models," *J. of Phys. A: Math. and Gen.*, vol. 15, no. 10, p. 3241, 1982.

[3] T. Wang, L. Wu, P. Nobel, and J. Roychowdhury, "Solving combinatorial optimisation problems using oscillator based Ising machines," *Natural Computing*, vol. 20, no. 2, pp. 287–306, 2021, DOI link.

[4] Y. Kuramoto, "Self-entrainment of a population of coupled non-linear oscillators," in *International Symposium on Mathematical Problems in Theoretical Physics*. Springer, 1975, pp. 420–422.

[5] P. Bhansali and J. Roychowdhury, "Gen-Adler: The generalized Adler's equation for injection locking analysis in oscillators," in *Proc. IEEE ASP-DAC*, January 2009, pp. 522–227.

[6] M. Kim, D. Venturelli, and K. Jamieson, "Leveraging quantum annealing for large MIMO processing in centralized radio access networks," in *Proc. of the ACM Spc. Int. Grp. on Data Comm.*, Aug 2019, DOI Link.

[7] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. Boston: McGraw Hill, 2008.

[8] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. New York: John Wiley & Sons, 1989.

[9] C. Kormanyos, *Real-Time C++: Efficient Object-Oriented and Template Microcontroller Programming*. Springer, 2018, DOI Link.

[10] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep Hypernetwork-based MIMO Detection," in *21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2020, pp. 1–5.

[11] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," in *Conf. Record of Thirty-Fifth Asilomar Conf. on Signals, Systems and Computers*, vol. 2. IEEE, 2001, pp. 1051–1055.

[5]A few important parameters of the emulator are: (1) $N = 33$, (2) 'ϕ's and '$\mathrm{J}_{i,j}$'s are 24 bit registers (3) $h = 1/2^6$—assuming $\max_{i,j} \mathrm{J}_{i,j} = 1$, (4) number of iterations of the while loop in Alg. 1 is 1024. Note that we initially set $\mathrm{F_s}(\phi) \triangleq 0$ for the first $(7/8) \times 1024 = 896$ iterations of the while loop. Moreover, we repeat the emulation of Alg. 1 by reusing the final phases of previous emulations as initial conditions; this whole process itself is repeated multiple times with random initial conditions.