An Optoelectronic Analog Ising Machine Enabling 2048-Spin and Low-Latency Calculations

Zihao Chen, Zhenhua Li, Zhaoang Deng, Jie Liu* and Siyuan Yu*

State Key Laboratory of Optoelectronic Materials and Technologies, School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou; 510275, China *liujie47@mail.sysu.edu.cn, yusy@mail.sysu.edu.cn

Abstract: An optoelectronic analog Ising machine is experimentally demonstrated. The SpMV algorithm is applied to accomplish two MAX-CUT tasks mapped into 2048-spin Ising networks, taking only 1.68µs per iteration. © 2022 The Author(s)

1. Introduction

Combinatorial optimization problems, which are known to be nondeterministic polynomial time (NP)-hard or NPcomplete problems, are considered challenging to traditional digital computers. Intensive research has been ongoing for more efficient computing concepts, amongst which analog Ising machines (AIM) have emerged as a promising solution. They work based on the concept that the combinatorial optimize problems can be mapped to the Ising model and implemented with artificial spin networks based on various analog physical systems, such as injection-locking lasers [1], degenerate optical parametric oscillator (DOPO) [2], spatial light modulation [3], opto-electronic oscillator (OEO) [4,5], etc. Among these proposed AIM schemes, the OEO-based AIMs [4,5], with their compact and stable setup without any nonlinear optical process or long-distance fiber based loops, have good potential to realize device integration and low-latency calculations.

However, reported OEO-based AIMs often have limited number of spins (e.g. 100 spins as implemented in [4]), which makes it difficult to accomplish large-scale optimization calculations. In addition, the computation time (or latency) of existing demonstrated AIMs, which is no less than 24 μ s for each iteration [2-5] resulted from long-distance feedback loop or hardware limitation, needs to be improved.

In this paper, an OEO-based AIM is experimentally demonstrated utilizing a commercial field-programmable gate array (FPGA) module with digital-to-analog and analog-to-digital converters (DAC/ADC). A sparse matrix–vector multiplication (SpMV) algorithm [6] is applied in the FPGA to calculate the feedback signals, considering that many combinatorial optimization problems can be mapped into Ising model with sparse spin-spin couplings. The MAX-CUT tasks for 2048-spin lattice graph and random graph are both demonstrated in the OEO-based AIM setup. It takes no more than 1.68 µs for each iteration (~0.6 MHz iteration rate), much faster than those of the schemes in [2-5].

2. Experimental Setup

The experimental setup of the OEO-based AIM is shown in Fig. 1(a), which is quite similar with that presented in [4]. The setup includes both optical and electrical pathways. In the optical pathway, light from a distributed feedback (DFB) laser at the wavelength of 1550 nm is modulated by the feedback electrical signal $f_n[k]$ through a Mach–Zehnder modulator (MZM) biased at the DC voltage of $-V_{\pi}/2$, where V_{π} is the half-wave voltage of the MZM. The feedback signal $f_n[k]$ can be expressed as [4]:

$$f_n[k] = \alpha x_n[k] + \beta \sum_m J_{mn} x_m[k] \tag{1}$$

where α and β represent the feedback and coupling strength for the spin $x_n[k]$ during the k_{th} iteration, respectively. J_{mn} is the coupling coefficient between the spins $x_n[k]$ and $x_m[k]$. The calculation of $f_n[k]$ can be mapped into a matrix vector multiplication:

$$F = CX = (\alpha I + \beta J)X \tag{2}$$

where *F* and *X* are the *N*×1 matrices whose n_{th} element is f_n and x_n in Eq. 1, respectively. $C = \alpha I + \beta J$ is the coupling matrix, where *I* is the identity matrix and *J* is the *N*×*N* matrix whose element in the m_{th} row and n_{th} column is J_{mn} . Considering the spins in the Ising network may not be densely coupled, *C* could be a sparse matrix in many combinatorial optimization problems.

The electrical pathway is the key part of the AIM system to realize large-scale sparse matrix calculation as well as low computation latency. It consists of an FPGA (Xilinx KU115), a DAC module with 2.6-GHz update rate and an ADC conversion module with a 2.6-GHz sampling rate. A SpMV algorithm [6] is applied in the $f_n[k]$ calculation of

the FPGA, in which the sparse matrix *C* is transformed into the Compressed Sparse Row (CSR) format. As the examples shown in the insets of Fig. 1(b), three $1 \times N$ matrices are conducted in the CSR format. The non-zero elements in *C* are stored in the first $1 \times N$ matrix *val* row by row, while the corresponding column numbers of the non-zero elements in the matrix *C* are stored in the second matrix *col*. The column numbers of the elements in *val*, which respectively correspond to the first non-zero element of each row in *C*, are stored in the third matrix *ptr*.



Fig. 1. (a) Framework of AIM with FPGA. (b) Application of SpMV algorithm in CSR format with FPGA.

As illustrated in Fig. 1(b), *L* multiplication units (*L* normally < N) are implemented to realize parallel integer multiplication of non-zero elements in each row of *C* with the corresponding elements in the $N \times 1$ matrix *X*. Here noted that the non-zero elements are read from the matrix *val*. *M* non-zero elements [the values of *M* can be decided by the elements in *ptr*, seeing inset in Fig.1(b)] are loaded into the *L* multiplication units ($M \leq L$) for parallel multiplication. The elements in *X* are read from the Block RAM (BRAM) according to the position information stored in the matrix *col* (the column numbers of the non-zero elements in the matrix *C* stored in *col* are corresponding to the row numbers of the desired elements for multiplication. Summarization result for the target row of the matrix *F* in each clock period. The results in each iteration are cached to the off chip DDR4 and the final output after completing all the iterations is transmitted to a computer through a USB interface.

A trade off should be made among logic resources, spin number and calculation latency to achieve calculation of a large-scale spin network using limited hardware resources. The low computation latency is mainly resulted from the high synchronism and low transmission delay between the FPGA and ADC/DAC modules, as well as short-distance feedback loop in the OEO-based AIM and good hardware-level parallelism. It only takes 1.68 μ s to finish one self-feedback iteration in our demonstrated OEO-based AIM for a random graph task with 2048 sparsely coupled spins (details can be found in the next section), including 0.5- μ s latency of ADC and DAC, which is much faster than the schemes presented in [2-5].

3. Results

A MAX-CUT task for a 64×32 lattice graph (seeing Fig. 2(a)) with coupling between all adjacent spin pairs is firstly performed by utilizing the OEO-based AIM setup. 200 independent trials each with 1000 iterations are conducted. Here the time-evolution parameters are set with $\alpha = 0.47$, $\beta = 0.53$. The computation time for one iteration is around 1.68 µs (~0.6 MHz iteration rate). As the results shown in Fig. 2(b), 90% ground state (GS) energy can be 100% achieved within 200 iterations in our demonstration. It can also be seen from the results illustrated in Fig. 2(c) that 96% of the calculation trials can finally reach the ground state, while all the trials can achieve 96% ground state energy, showing the good performance of our OEO-based AIM setup in solving the MAX-CUT problem of relatively large-scale lattice graph.

As shown in Fig. 2(d), a random-graph MAX-CUT problem is also constructed to test the capability of the OEObased AIM setup for solving relatively complex combinatorial optimization problem. In this graph with a total of 2048 spins, each spin is randomly coupled with 127 spins. 100 independent trials are carried out, each of which requires 100 iterations. The iteration rate in this calculation is also ~0.6 MHz (~1.68 μ s latency of each iteration), similar with that of MAX-CUT task for the 64×32 lattice graph. The solutions based on other two methods/setups are also calculated for references. The first one is calculated using the simulated annealing (SA) algorithm [7], in which 100 independent trials each with 32,000 sweeps are conducted, and the mean of the obtained solutions is set as the final output. The second reference solution is obtained employing the OEO-based AIM setup presented in [5], in which the high-bandwidth arbitrary waveform generator (AWG) and real-time oscilloscope (RTO) with better performance DAC/ADC but higher transmission latency are utilized to replace the DAC and ADC modules in our scheme. 20 independent trials each with 30 iterations are conducted based on the AIM setup using AWG and RTO. As shown in Fig.2(e) and (f), the Ising Hamiltonian energy found after 100 iterations using our OEO-based AIM system is lower than that calculated with SA by 3%, but higher than that calculated with AWG and RTO by 14%. This may have resulted from the non-ideal effect of the AC coupling components in the DAC used in our scheme, compared with that of the AWG utilized in the reference AIM setup. However, much lower calculation latency can be achieved at the cost of only slightly higher minimum Hamiltonian energy that can be found and acceptable for many practical optimization problems, showing the good overall performance of our AIM scheme.



Fig. 2. (a) A sketch of the 64*32 lattice graph. (b) Time evolution of success rate of solving the square lattice's MAX-CUT for 1000 iterations ($\alpha = 0.47, \beta = 0.53$). (c) The energy distribution for square lattice after 1000 iterations. (d) A sketch of the random-graph. (e) Time evolution of Ising energy for random-graph, including 100 independent trials, ($\alpha = 0.5, \beta = 0.156$), the blue area indicates the range of the energy of AIM solutions. (f) The Ising energy distribution for random-graph with AIM after 100 iterations.

4. Conclusion

An OEO-based AIM employing commercial FPGA with DAC and ADC modules has been experimentally demonstrated. A SpMV algorithm has been applied in the FPGA to accomplish two MAX-CUT tasks (lattice graph and random graph) mapped into Ising network with 2048 spins using limited hardware resources. It takes only 1.68 µs for each iteration, thanks to the high synchronism and low transmission delay between the FPGA and ADC/DAC modules, as well as short-distance feedback loop in the OEO-based AIM and good hardware-level parallelism.

5. Acknowledgement

This work was funded by Innovation Program for Quantum Science and Technology (2021ZD0301400); National Natural Science Foundation of China-Guangdong Joint Fund (U2001601); National Natural Science Foundation of China (61875233), (62101602); The Key Research and Development Program of Guangdong Province (2018B030329001), Guangdong Provincial Pearl River Talents Program (2017BT01X121).

6. References

- 1. Shoko Utsunomiya, *et al.*, Opt. Express **19**, 18091-18108 (2011).
- 2. T. Honjo, et al., Sci Adv. 7, eabh0952 (2021).
- 3. D Pierangeli, *et al.*, Phys Rev Lett. **122**, 213902 (2019).
- 4. F. Böhm, et al., Nat Commun. 10, 3538 (2019).
- 5. Z. Li, et al., OFC, Tu1H.4 (2021).
- 6. Y. Zhang, et al., ICFPT, 255-262 (2009).
- 7. S. Kirkpatrick, *et al.*, Science **220**, 671-680 (1983).