symmetry

Article

AIM: Annealing in Memory for Vision Applications

Zhi Wang [1,*], Xiao Hu [1,*], Jian Zhang [1], Zhao Lv [2] and Yang Guo [1,*]

[1] College of Computer, National University of Defense Technology, Changsha 410073, China; JianZhang@nudt.edu.cn
[2] Center for Assessment and Demonstration Research, Academy of Military Sciences, Beijing 100091, China; zhaolv@nudt.edu.cn
* Correspondence: zhiwang@nudt.edu.cn (Z.W.); xiaohu@nudt.edu.cn (X.H.); guoyang@nudt.edu.cn (Y.G.)

Received: 13 February 2020; Accepted: 12 March 2020; Published: 20 March 2020

Abstract: As the Moore's law era will draw to a close, some domain-specific architectures even non-Von Neumann systems have been presented to keep the progress. This paper proposes novel annealing in memory (AIM) architecture to implement Ising calculation, which is based on Ising model and expected to accelerate solving combinatorial optimization problem. The Ising model has a symmetrical structure and realizes phase transition by symmetry breaking. AIM draws annealing calculation into memory to reduce the cost of information transfer between calculation unit and the memory, improves the ability of parallel processing by enabling each Static Random-Access Memory (SRAM) array to perform calculations. An approximate probability flipping circuit is proposed to avoid the system getting trapped in local optimum. Bit-serial design incurs only an estimated 4.24% area above the SRAM and allows the accuracy to be easily adjusted. Two vision applications are mapped for acceleration and results show that it can speed up Multi-Object Tracking (MOT) by 780× and Multiple People Head Detection (MPHD) by 161× with only 0.0064% and 0.031% energy consumption respectively over approximate algorithms.

Keywords: Ising; annealing; memory; vision applications

1. Introduction

Over the last 50 years, Moore's law has predicted that the performance of integrated circuits approximately doubled every two years [1]. However, it has become more and more difficult to maintain it in recent years. Some domain-specific architectures even non-Von Neumann systems have been presented to keep the progress, such as machine-learning accelerators [2–4], neuromorphic chips [5–7] and quantum computing [8,9]. These new architectures fundamentally based on different theoretical models are redefining new era of computing.

The Ising chip [10–15], based on the Ising model [16], is one of the new architectures, which can efficiently solve combinatorial optimization problem. Combinatorial optimization is a branch of discrete mathematics that seeks to find the best possible solution from a finite or countably set of possibilities [17]. An enormous number of key issues in science and engineering can be classified as combinatorial optimization problems. As many of these problems are known to be NP-hard [18], lots of time and energy will be consumed to solve them under Von Neumann architecture. The Ising model, a symmetrical structure for modeling the behavior of magnetic spin [16], can speed up the ground state search process by enabling each spin to search in parallel. Different from Von Neumann architecture, the Ising model solves combinatorial optimization problem as the following three steps: (1) mapping input problem to the interaction coefficients of the Ising model, (2) performing iterative annealing calculation to approach the ground state, (3) reading final spin state to get the solution of original problem. The self-convergence property of the Ising model makes it a promising method for solving combinatorial optimization problem.

The Ising chip is mainly composed of memory that stores interaction coefficients and calculation logic for iterative annealing [14]. The mapped coefficients of combinatorial optimization problem are firstly stored in the memory, and then the iterative annealing calculation is started to search the ground state. The next state of each spin is determined by its neighboring spin states and their interaction coefficients. According to the connection relationship between spins, the Ising chip can be divided into local-interconnect structure and global-interconnect structure. Each spin is connected to only four or eight spins in local-interconnect structure while connected to (n-1) spins in global-interconnect structure when there are n spins in the system. The oversimplified connection relationship of local-interconnect structure limits its application, and more combinatorial optimization problems can only be mapped to global-interconnect structure. However, the memory capacity for storing coefficients and the iterative annealing logic will increase exponentially with the number of spins due to its complex connection in global-interconnect structure. Hardware-complexity for local-interconnect structure is $O(n)$ while $O(n^2)$ for global-interconnect structure. The current Ising chips are mainly local-interconnect structure. The high hardware cost limits the realization of global-interconnect structure. Based on the characteristics of global-interconnect structure: (1) large-capacity memory is required to store the interaction coefficients, (2) iterative annealing calculation is relatively simple, (3) many calculations in the annealing can be processed in parallel, (4) different calculation accuracy is required for different applications, this paper proposes novel AIM architecture to implement a global-interconnect Ising chip. The advantage of the structure is: (1) it draws annealing into memory to reduce the cost of information-transfer between calculation unit and memory, (2) it improves parallel processing by enabling each SRAM array to perform calculations since each spin needs to update its local search term (LST) in each annealing step, (3) to reduce the influence of annealing calculation logic on the area of SRAM array, the bit-serial design of annealing calculation is adopted in AIM, which incurs only an estimated 4.24% area overhead for the SRAM array. The disadvantage is that it will affect the calculation time due to bitwise calculation. However, on the other hand, bit-serial design allows the computation accuracy easy to be adjusted.

Owing to the special computing paradigm, another challenge for the Ising chip is how to map real-world applications to it for acceleration. Data association, which is a kind of combinatorial optimization problems, exists widely in vision applications since each object does not exist alone in the world [19,20]. Better understanding would be got if association relationship is considered in the algorithm of computer vison. This paper will demonstrate how to map two vision applications to the annealing in memory architecture for acceleration. The results show that the proposed structure can speed up MOT by 780× and MPHD by 161× with only 0.0064% and 0.031% energy consumption respectively over approximate algorithms.

The remainder of this paper is organized as follows. Section 2 reviews the the Ising model. Annealing in memory architecture is introduced in Section 3. Section 4 demonstrates how to map MOT and MPHD to the proposed architecture. Final conclusion is presented in Section 5.

2. Overview of the Ising Model

The Ising model was proposed by Wilhelm Lenz in 1920 to study the ferromagnetism of atomic spins [16]. The model consists of spins that can be one of two-body correlated states to store information. The spins are arranged in a graph and each of them can interact with its neighbors based on the coefficients. The topology of the graph can be local-interconnect or global-interconnect, which depends on the connection relationship between spins. The Hamiltonian of the Ising model is:

$$\mathrm{H}(s) = -\sum_{<i,j>} J_{ij}\, s_i\, s_j - \sum h_i\, s_i \tag{1}$$

where s_i is the state of spin i, J_{ij} is the coefficient between spin i and spin j, h_i is the external magnetic field for spin i. The Hamiltonian represents the total energy of the Ising model and acts as a cost function to be optimized. The goal of ground state search is to find a spin configuration that minimizes the cost function, which corresponds to the ground state of the Ising model.

Each spin simultaneously performs iterative annealing calculation to search for the ground state. The annealing calculation is composed of local search and probability flipping. The local search is defined as follows:

$$Ls_i(t) = \sum_{<i,j>} J_{ij}\, s_j(t) + h_i \tag{2}$$

$$s_i'(t) = \begin{cases} 1 & Ls_i(t) > 0 \\ 0 & others \end{cases} \tag{3}$$

where $Ls_i(t)$ is the LST of spin i, $s_i'(t)$ is the neighboring state of spin i at time t, which makes the local energy lower. To help the system getting out of local optimum, the state transition is defined as a probability behavior, which means that spin i changes its state probabilistically. The flipping probability is shown in (4), which is related to the energy change ΔE in (5) when the state of spin i changes from $s_i(t)$ to $1 - s_i(t)$ and the annealing temperature $T(t)$ at time t.

$$P(s_i'(t)|\, s_i(t)) = \frac{1}{1 + e^{\frac{\Delta E}{T(t)}}} \tag{4}$$

$$\Delta E = (2\, s_i(t) - 1)\, Ls_i(t) \tag{5}$$

The annealing temperature $T(t)$ plays an important influence on the flipping probability, and changes from high to low along the annealing process. It determines the sensitivity of the flipping probability to energy changes. By combining (4) and (5), the next state $s_i(t+1)$ for spin i can be determined by (6).

$$s_i(t+1) = \begin{cases} 1 & with\ probability\ P = \frac{1}{1+e^{\frac{-Ls_i(t)}{T(t)}}} \\ 0 & others \end{cases} \tag{6}$$

The system's energy will continue to decrease as the annealing calculation progresses, and the system finally reaches the ground state or near-ground state when the annealing temperature $T(t)$ arrives at the end of the search.

The local search of each spin can be realized in parallel according to De Gloria algorithm [21]. Firstly, the states of all spins are initialed and the local search term of each spin is precomputed. When one spin is randomly selected to update its state, the probability in (6) is calculated and next state of the spin is determined by comparing probability P with a random r as (7), where r is between 0 and 1. If the spin changes its state, all other spins will update their local search term as (8), which can be done in parallel. After that, another spin is randomly selected to repeat the upon iterative annealing process.

$$s_i(t+1) = \begin{cases} 1 & \frac{1}{1+e^{\frac{-Ls_i(t)}{T(t)}}} > r \\ 0 & others \end{cases} \tag{7}$$

$$Ls_j(t+1) = Ls_j(t) + J_{ij}(2\, s_i(t+1) - 1) \tag{8}$$

3. Annealing in Memory Architecture

Annealing in memory architecture aims at reducing data movement by performing annealing calculation directly in memory. Coefficients between spins are stored in SRAM array and the LSTs are updated in parallel near SRAM. Bit-serial design of annealing calculation incurs little area overhead to SRAM and allows the accuracy easy to be adjusted. To avoid the system getting trapped in local optimum, an approximate probability flipping method is presented.

3.1. Top-Level Architecture

Figure 1 describes top-level architecture of the annealing in memory design. Annealing Control (Anl_Ctrl) module controls the flow of iterative annealing calculation. It also performs probability flipping to determine next state of the selected spin based on LST and probability flipping term (PFT). Compute Static Random-Access Memory (CSRAM) modules store coefficients and LST. The update of LST is performed in parallel in CSRAM when it receives update command from the Anl_Ctrl module. Randomly Select (Rdm_Slt) module is a random number generator, which generates a random number at regular intervals to select a spin to update. The Probability Flipping (Prob_Flip) module generates the PFT for probability flipping.

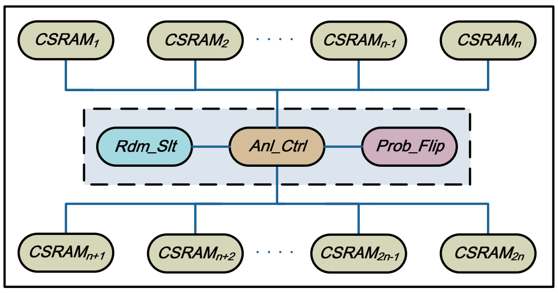


Figure 1. Top-level architecture of annealing in memory design. It is mainly composed of four modules, including Anl_Ctrl, Rdm_Slt, Prob_Flip and CSRAM.

The flow chart of iterative annealing calculation is shown in Figure 2. First of all, the coefficients, LSTs and spins are initialled in the CSRAM module. Next, Anl_Ctrl module selects a spin to update according to the random number generated from Rdm_Slt module. Then, it gets the LST of that spin from the corresponding CSRAM module, and gets PFT from Prob_Flip module at the same time. Next state of the spin is determined based on its LST and the PFT. After that, the corresponding spin and all other LSTs will be updated if the spin has changed its state, and the spin unchanged number n will be reset in the flowing step. Otherwise, the spin unchanged number n will be increased by 1. Another spin will be selected to repeat the annealing process if the spin unchanged number is less than the predefined maximum value. The iteration process will stop when there is no spin updates its state for a certain period and finally the spins are read out to get the solution to the mapped problem.

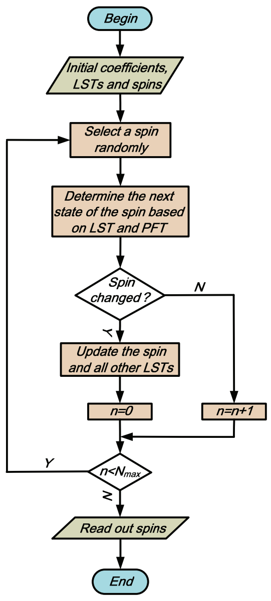

Begin
Initial coefficients, LSTs and spins
Select a spin randomly
Determine the next state of the spin based on LST and PFT
Spin changed ?
N
Y
Update the spin and all other LSTs
n=0
n=n+1
Y
n<N$_{max}$
N
Read out spins
End

Figure 2. The flow chart of iterative annealing calculation process. The iteration process will stop when the spin unchanged number n is equal to the predefined maximum value.

3.2. Local Search in Memory

Local Search is performed in CSRAM module, which is shown in Figure 3. The CSRAM module consists of a SRAM array, adjacent annealing calculation logic and two shift registers. Coefficients between spins and the LST are stored in SRAM array and Lst shift register respectively. The adjacent annealing calculation logic circuit is used to update the LST in each iterative annealing step. In order to reduce hardware cost and make calculation accuracy be adapted easily, bit-serial design is adopted to update the LST.

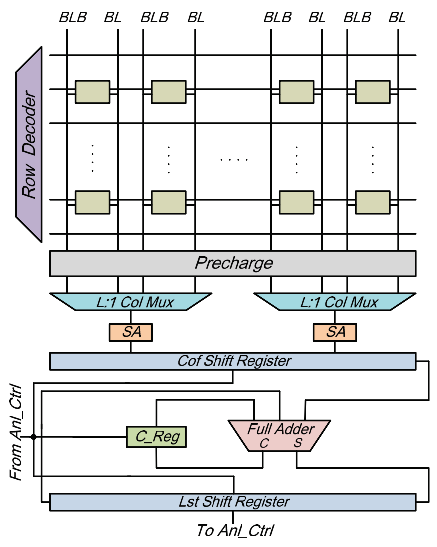


Figure 3. The structure of the CSRAM module. The coefficients are stored in SRAM array and local search term is stored in Lst shift register. Local search is performed by the adjacent calculation logic circuit.

When the selected spin changes its state in an iterative annealing step, all spins need to update their LSTs in the corresponding CSRAM module. The related coefficient is firstly read to the Cof shift register as the update command arrives, and then the LST is updated by adding the related coefficient and original LST based on the changed spin. C_reg is a register that stores carry in the bit-serial calculation.

3.3. Approximate Probability Flipping Method

To avoid the system getting trapped in local optimum, the state transition is defined as a probability behavior, which means the flipping of spin state occurs probabilistically. The flipping probability is related to energy change and annealing temperature. The following state of the spin is determined by comparing the probability P with a random r as (7), which can be transformed into the following:

$$s_i(t+1) = \begin{cases} 1 & T(t) * \ln(\frac{1}{r} - 1) + Ls_i(t) > 0 \\ 0 & others \end{cases} \tag{9}$$

$$Pf(t) = T(t) * \ln(\frac{1}{r} - 1) \tag{10}$$

where $Pf(t)$ is the PFT, which is the product of annealing temperature $T(t)$ and $\ln(1/r - 1)$. Then, the state of the spin can be determined according to the sign bit of the sum, which is obtained by adding LST $Ls_i(t)$ with PFT $Pf(t)$. Due to the complexity of the PFT, a digital hardware circuit is proposed to implement it approximately.

As it can be seen in Figure 4, the curve of $\ln(1/r - 1)$ is approximated by three lines, where r is a random number between 0 and 1, and two of which have the same slope. The linear interpolation of $\ln(1/r-1)$ is explained via (11). (11) can be approximatively replaced by (12) using the similar technique that was proposed in [22].

$$\ln(\frac{1}{r}-1) \approx \begin{cases} -29.711 * r + 4.718 & 0 < r < 0.125 \\ -4.74 * r + 2.371 & 0.125 \le r \le 0.875 \\ -29.711 * r + 24.892 & 0.875 < r < 1 \end{cases} \tag{11}$$

$$\ln(\frac{1}{r}-1) \approx \begin{cases} -32 * r + 4.875 & 0 < r < 0.125 \\ -4 * r + 2 & 0.125 \le r \le 0.875 \\ -32 * r + 27.125 & 0.875 < r < 1 \end{cases} \tag{12}$$

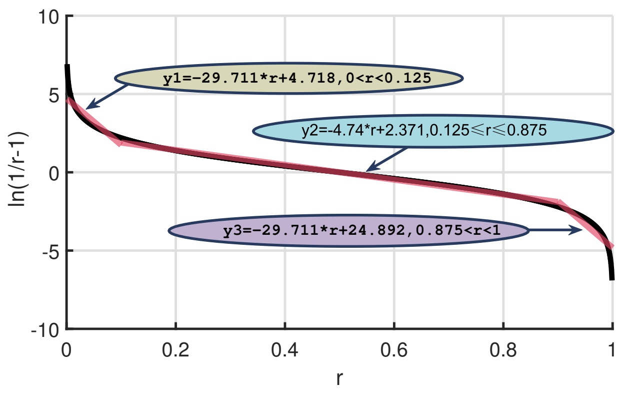


Figure 4. The curve of $\ln(1/r-1)$, where r is a random number between 0 and 1. The curve is approximated by three lines.

The fixed constants can be represented by a finite and shorter number of bits and the product of different constants with r can be realized by shifting, so the approximation in (12) is more convenient for digital logic hardware implementation. The selection of different fixed constants and different products, which is based on the value of r, can be achieved by multiplexers. The proposed structure to approximate $\ln(1/r-1)$ is shown in Figure 5, where the final approximate value is stored in register R2. The annealing Temperature T is set to some fixed values, which is powers of 2, and will keep a fixed time for each value along the annealing process. Therefore, the PFT $Pf(t)$ can be obtained by shifting register R2 according to the value of T.

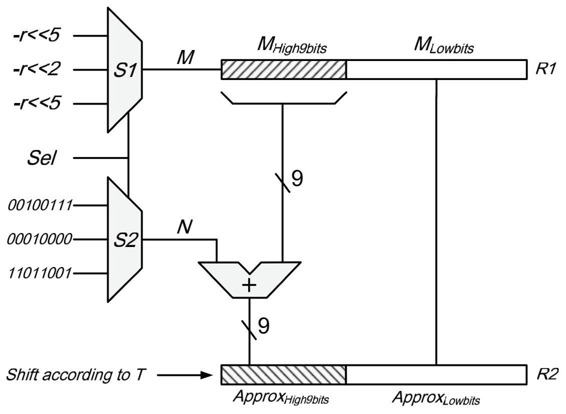


Figure 5. Proposed three-region linear approximation architecture. The multiplication with powers of 2 can be achieved by shifting. The selection of different fixed constants and different products, which is based on the value of r, can be achieved by multiplexers.

The energy histories of iterative annealing calculation process by using different probability flipping methods are depicted in Figure 6. Coefficients of the Ising model are randomly generated. State of the system will fall into local optimum soon if there is no probability flipping, which means the next state of the spin is only determined by local search. Although simplified for hardware implementation, approximate probability flipping method is able to find a state with similar energy to theoretical method at the end of the annealing process.

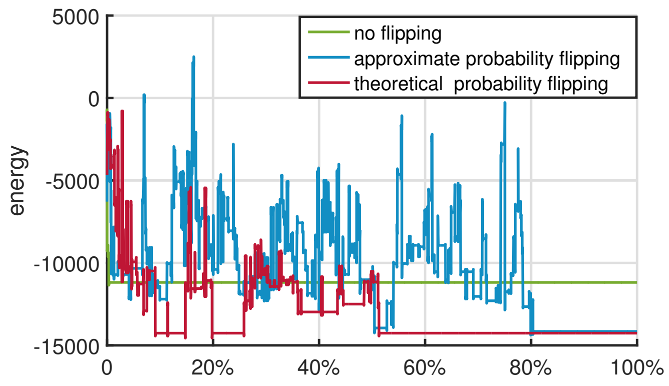


Figure 6. Energy history of iterative annealing calculation process by using different probability flipping methods. Coefficients of the the Ising model are randomly generated.

3.4. Hardware Performance

The annealing in memory architecture is synthesized using Design Compiler in 28 nm process, where the SRAM array is obtained using memory compiler with a 28 nm library. The target clock frequency is 1 GHz, and the size of the SRAM array is 1024*64. The entire chip consists of 4096 SRAM arrays, so that it can support up to 4096 global interconnect spins if the bit-width of the coefficient is configured to 16-bits.

We conclude the hardware performance in Table 1. Area of CSRAM and the entire chip are 20.74 µm^2 and 85 mm^2 respectively. The local search logic circuit incurs only 4.24% area overhead to CSRAM module. The chip power is 3.26 W.

Table 1. Hardware performance of AIM.

Items	**Values**
Clock frequency	1 GHz
Chip area	85 mm^2
Chip power	3.26 W
Number of SRAM array	4096
Area of SRAM array	19.86 µm^2
Area of CSRAM	20.74 µm^2

4. Application

Vision applications mainly focus on the understanding of useful information from images. Data association exists widely in vision applications since each object does not exist alone in the world. Better understanding would be got if association relationship is considered in the algorithm of computer vison. Data association is a kind of combinatorial optimization problems, which can be accelerated by the Ising model. In this section, we demonstrate how to map two vision applications to the annealing in memory architecture for acceleration.

4.1. Mapping MOT to AIM

MOT is the problem of simultaneously tracking multiple moving objects in a sequence of images and is a key component of many vision tasks. Tracking-by-detection is widely accepted MOT method. The core issue is how to effectively associate hypotheses and build complete trajectories.

Figure 7a shows the process of mapping MOT to AIM. Firstly, hypotheses for each frame are obtained from the detector and the combinations of trackers with hypotheses are regarded as spins. The unary item in (1) is defined by the affinity between tracker and hypothesis, the pairwise item is defined by the cost of violating interactions [19]. In the next step, coefficients are passed to AIM and the annealing process starts. We get spin states at the end of the annealing process and map it to the solution of MOT problem. The combination of tracker and hypothesis is successful if the spin state is 1, otherwise the hypothesis does not belong to the tracker.

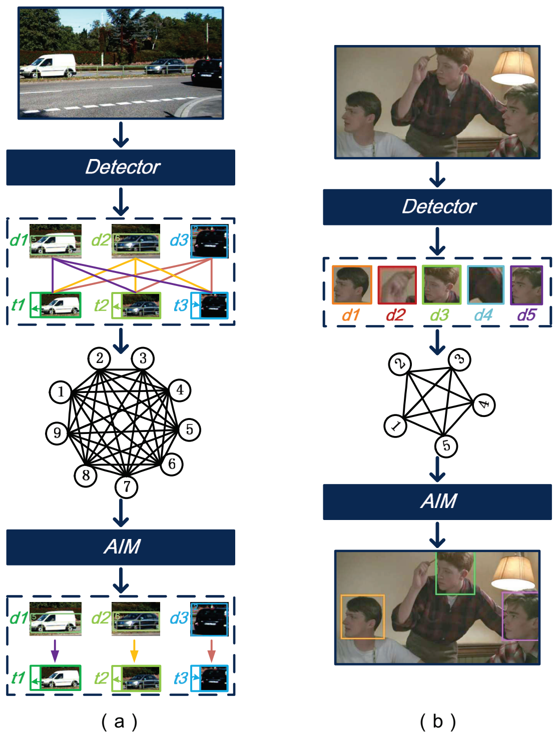


Figure 7. (**a**) The process of mapping MOT to AIM, where the unary item is defined by the affinity between tracker and hypothesis, the pairwise item is defined by the cost of violating interactions. (**b**) The process of mapping MPHD to AIM, where the unary item is defined by the response of local object detector at corresponding locations and the pairwise item depends on the image data.

4.2. Mapping MPHD to AIM

Data association is also considered in MPHD. Since each object does not exist alone in the world, the model in [20] models spatial association between objects which provide complementary contextual cues for detection, and can get better result than others.

The process of mapping MPHD to AIM is shown in Figure 7b. At first, objects of the image are obtained from the detector and each object is regarded as a spin. The unary item in (1) is defined by the response of local object detector at corresponding locations and the pairwise item depends on the image data. After that, coefficients are passed to AIM, and then we get spin states at the end of the annealing process, which corresponds to the solution of the MHPD problem. If the spin state is 1, the object is a head, otherwise it is not.

4.3. Evaluation

To evaluate the performance, we develop a cycle-accurate simulator for AIM, and run the baseline of multi-branch algorithm for MOT [19], Quadratic Pseudo-Boolean Optimization (QPBO) and sequential tree-reweighted message passing (TRWS) algorithm for MPHD [20] on Core i7-8450 processor. The evaluation for MOT is based on the KITTI Vision Benchmark Suite [23], which is a widely used benchmark to evaluate MOT, and the CLEAR MOT metrics [24]. Dataset used for MPHD evaluation is HollywoodHeads dataset [20] and Casablanca dataset [25], which are widely used datasets to evaluate MPHD.

Table 2 shows the MOT results of evaluating multi-branch algorithm and AIM on KITTI Vision Benchmark. The upper three rows show the evaluation results for pedestrians and the following three rows show the evaluation results for cars. The specific metrics include Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), ID-switches (ID), Fragmentations (Frag), Mostly Tracked (MT), Partly Tracked (PT), Mostly Lost (ML). For Pedestrians, AIM outperforms multi-branch in MOTA and MOTP. However, AIM achieves better MOTA but worse MOTP and Frag for cars. One of qualitative tracking results by using AIM is shown in Figure 8a. Figure 9 reports precision-recall (PR) curve for MPHD by using QBPO, TRWS and AIM. All the three methods can get similar results for multiple people head detection. One of qualitative head detection results by using AIM is shown in Figure 8b.

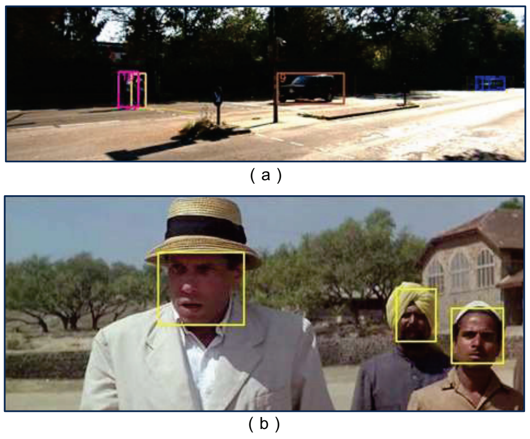


Figure 8. Examples of qualitative tracking (**a**) and detecting (**b**) results by using AIM, which are highlighted by rectangular box.

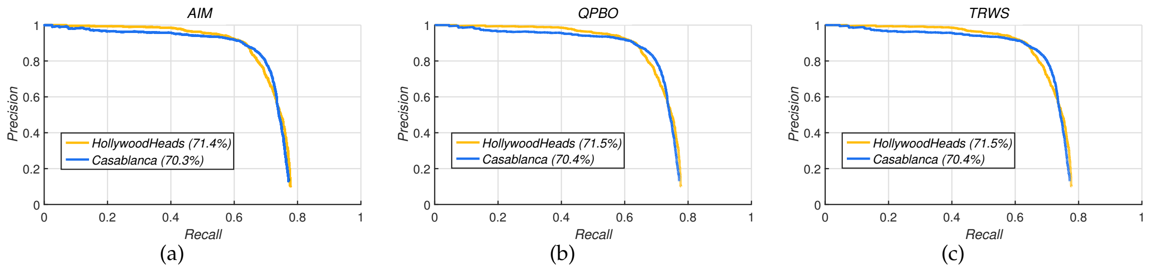


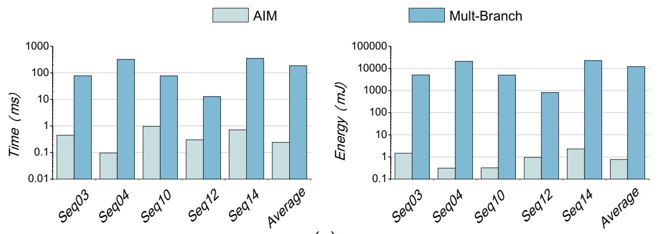
Figure 9. Precision-recall curve for MPHD on the HollywoodHeads [20] and Casablanca datasets [25] by using AIM, QPBO and TRWS. All the three methods can get similar results for MPHD.

The run time and energy consumption of all methods are shown in Figure 10. AIM achieves 780× speedup in latency compared to multi-branch algorithm for MOT, 161× and 195× speedup compared to QPBO and TRWS for MPHD respectively. The significant speedup can be attributed to the highly parallel processing for enabling each SRAM array to carry out local search. AIM achieves energy efficiency that is 0.0064% of the multi-branch algorithm for MOT, 0.031% and 0.025% of QPBO and

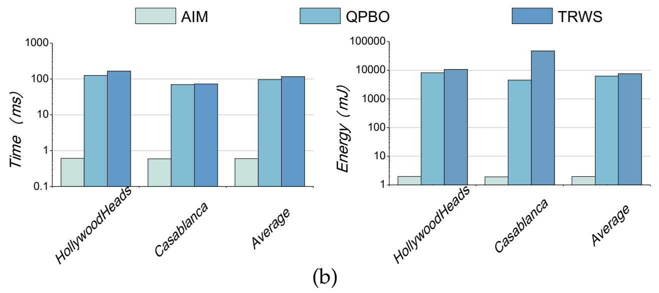
TRWS for MHPD respectively. The energy efficiency improvement can be explained by annealing in memory architecture that reduces data movement between calculation unit and the memory.

Table 2. Results for MOT, where the upper three rows show the evaluation results for pedestrians and the following three rows show the evaluation results for cars.

Pedestrians	**MOTA**	**MOTP**	**ID**	**Frag**	**MT**	**PT**	**ML**
Multi-branch	46.76%	75.98%	1	12	60.00%	30.00%	10.00%
AIM	47.12%	76.03%	1	12	60.00%	30.00%	10.00%
Cars	**MOTA**	**MOTP**	**ID**	**Frag**	**MT**	**PT**	**ML**
Multi-branch	69.79%	83.86%	4	19	60.34%	29.31%	10.34%
AIM	70.00%	83.84%	4	21	60.34%	29.31%	10.34%



(a)



(b)

Figure 10. (**a**) The run time and energy consumption comparison of AIM and multi-branch algorithm for MOT. (**b**) The run time and energy consumption comparison of AIM, QPBO and TRWS for MOT.

5. Conclusions

This paper presents annealing in memory architecture to realize the Ising calculation, which is expected to accelerate solving combinatorial optimization problem. Based on the characteristics of the Ising chip mainly including large-capacity memory and simple iterative calculation, it draws annealing into memory to reduce the cost of information transfer between calculation unit and the memory, improves the ability of parallel processing by enabling each SRAM array to perform calculation. An approximate probability flipping circuit is proposed to avoid getting trapped in local optimum. This paper also demonstrates how to map two vision applications to the proposed architecture for acceleration. The results show that it can speed up multi-object tracking by 780× and multiple people head detection by 161× with only 0.0064% and 0.031% energy consumption respectively over approximate algorithms.

Author Contributions: Conceptualization, Z.W. and J.Z.; methodology, Z.W. and X.H.; formal analysis, X.H.; investigation, Z.L.; writing–original draft preparation, Z.W.; writing–review and editing, Z.W. and Y.G.; project administration, Y.G. All authors have read and agree to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moore, G.E.. Progress in digital integrated electronics. In Proceedings of the Electron Devices Meeting, Washington, D.C., USA 1–3 December 1975; Volume 21, pp. 11–13.
2. Chen, Y.; Luo, T.; Liu, S.; Zhang, S.; He, L.; Wang, J.; Li, L.; Chen, T.; Xu, Z.; Sun, N.; et al. Dadiannao: A machine-learning supercomputer. In Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, Cambridge, UK, 13–17 December 2014; IEEE Computer Society: Washington, DC, USA 2014, pp. 609–622.
3. Chen, Y.H.; Krishna, T.; Emer, J.S.; Sze, V. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE J. Solid-State Circuits* **2016**, *52*, 127–138. [CrossRef]
4. Gao, M.; Pu, J.; Yang, X.; Horowitz, M.; Kozyrakis, C. Tetris: Scalable and efficient neural network acceleration with 3d memory. In Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, Xi'an, China, 8–12 April 2017; pp. 751–764.
5. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673. [CrossRef] [PubMed]
6. Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Merolla, P.; Imam, N.; Nakamura, Y.; Datta, P.; Nam, G.J.; et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2015**, *34*, 1537–1557. [CrossRef]
7. Lin, C.K.; Wild, A.; Chinya, G.N.; Davies, M.; Wang, H. Programming Spiking Neural Networks on Intel Loihi. *Computer* **2018**, *51*, 52–61. [CrossRef]
8. Johnson, M.W.; Amin, M.H.; Gildert, S.; Lanting, T.; Hamze, F.; Dickson, N.; Harris, R.; Berkley, A.J.; Johansson, J.; Bunyk, P.; et al. Quantum annealing with manufactured spins. *Nature* **2011**, *473*, 194–198. [CrossRef]
9. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
10. Yamaoka, M.; Yoshimura, C.; Hayashi, M.; Okuyama, T.; Aoki, H.; Mizuno, H. 24.3 20k-spin Ising chip for combinational optimization problem with CMOS annealing. In Proceedings of the 2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers, San Francisco, CA, USA, 11–15 February 2015; pp. 1–3.
11. Zhang, J.; Chen, S.; Wang, Y. Advancing CMOS-Type Ising Arithmetic Unit into the Domain of Real-World Applications. *IEEE Trans. Comput.* **2017**, *67*, 604–616. [CrossRef]
12. Matsubara, S.; Tamura, H.; Takatsu, M.; Yoo, D.; Vatankhahghadim, B.; Yamasaki, H.; Miyazawa, T.; Tsukamoto, S.; Watanabe, Y.; Takemoto, K.; et al. Ising-model optimizer with parallel-trial bit-sieve engine. In Proceedings of the Conference on Complex, Intelligent, and Software Intensive Systems, Turin, Italy, 10–13 July 2017; Springer: Berlin, Germany, 2017; pp. 432–438.
13. Zhang, J.; Chen, S.; Wang, Z.; Wang, L.; Lv, L.; Wang, Y. Pre-Calculating Ising Memory: Low Cost Method to Enhance Traditional Memory with Ising Ability. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Firenze Fiera Spa, Florence, Italy, 27–30 May 2018; pp. 1–5.
14. Takemoto, T.; Hayashi, M.; Yoshimura, C.; Yamaoka, M. 2.6 A 2× 30k-Spin Multichip Scalable Annealing Processor Based on a Processing-In-Memory Approach for Solving Large-Scale Combinatorial Optimization Problems. In Proceedings of the 2019 IEEE International Solid-State Circuits Conference-(ISSCC), San Francisco, CA, USA, 17–21 February 2019; pp. 52–54.
15. Skubiszewski, M. An exact hardware implementation of the Boltzmann machine. In Proceedings of the Fourth IEEE Symposium on Parallel and Distributed Processing, Arlington, TX, USA, 1–4 December 1992; pp. 107–110.
16. Brush, S.G. History of the Lenz-Ising model. *Rev. Mod. Phys.* **1967**, *39*, 883. [CrossRef]
17. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization*; Prentice Hall: Englewood Cliffs, NJ, USA, 1982; Volume 24.
18. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Springer: Berlin, Germany, 1972; pp. 85–103.

19. Osep, A.; Mehner, W.; Mathias, M.; Leibe, B. Combined image-and world-space tracking in traffic scenes. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1988–1995.
20. Vu, T.H.; Osokin, A.; Laptev, I. Context-aware CNNs for person head detection. In Proceedings of the IEEE International Conference on Computer Vision, Araucano Park, Las Condes, Chile, 11–13 December 2015.
21. De Gloria, A.; Faraboschi, P. Application specific parallel architectures. In Proceedings of the International Conference on Massively Parallel Computing Systems, Ischia, Italy, 2–6 May 1994.
22. Liu, C.W.; Ou, S.H.; Chang, K.C.; Lin, T.C.; Chen, S.K. A Low-Error, Cost-Efficient Design Procedure for Evaluating Logarithms to Be Used in a Logarithmic Arithmetic Processor. *IEEE Trans. Comput.* **2016**, *65*, 1158–1164. [CrossRef]
23. Geiger, A. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
24. Bernardin, K.; Stiefelhagen, R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *Eurasip J. Image Video Process.* **2008**, *2008*, 246309. [CrossRef]
25. Ren, X. Finding people in archive films through tracking. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Miami, FL, USA, 20–25 June 2009.