A Time-Division Multiplexing Ising Machine on FPGAs

Kasho Yamamoto, Weiqiang Huang,
Shinya Takamaeda-Yamazaki, Masayuki Ikebe, Tetsuya Asai, Masato Motomura
Hokkaido University
{yamamoto, huang}@lalsie.ist.hokudai.ac.jp,{takamaeda, ikebe, asai, motomura}@ist.hokudai.ac.jp

ABSTRACT

Annealing machines based on the Ising model which can solve combinatorial optimization problems is an emerging solution to overcome the performance limit of von Neumann architecture. However, it is difficult to solve practical combinatorial optimization problems by existing approaches of FPGA-based annealing machines, due to the small number of implementable spins. In this paper, we propose the time-division multiplexing Ising machine architecture that efficiently utilizes on-chip memory resources in an FPGA, in order to address large scale combinatorial optimization problems. The evaluation result shows that it is possible to increase the spin number by 64 times compared to the conventional annealing machine. In addition, the time-multiplxing architecture liberates a logical Ising structure representing problem constraints from the physical hardware structure on an FPGA. It provides the ability to implement more complex topologies corresponding to practical of problems on FPGAs.

1. INTRODUCTION

Combinational optimization is a fundamental and practical method to describe and solve various social problems in our daily lives, such as transportation cost optimization. The difficult point to solve such optimization problems is that they are known as NP, so that they can be resolved in a polynomial time. Even if approximate solutions via heuristic approaches are allowed, it takes certainly long computing time to solve them due to their iterative searches of optimal solution points.

In order to overcome inefficiency of the modern von Neumann computers for such optimization problems, an Ising computer has been proposed based on the "natural computing" paradigm which maps a target problem onto a physical model in nature and observes the obtained status of the physical matters as its computing result. Especially, Yamaoka et al. has proposed a CMOS annealing LSI [1] to accelerate combinational optimization problems by utilizing artificial Ising model as electric circuits. Additionally, FPGA-based Ising machines are attractive alternatives for easy development of the systems instead of custom LSIs. One of the main problems of existing FPGA-based Ising machines is the limitation of simulated spin count coming from available hardware resource amount.

In this paper, we explore a novel FPGA-based Isign machine architecture for increasing the simulated spin count.

This work was presented in part at the international symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART2017) Bochum, DE, June 7-9, 2017.

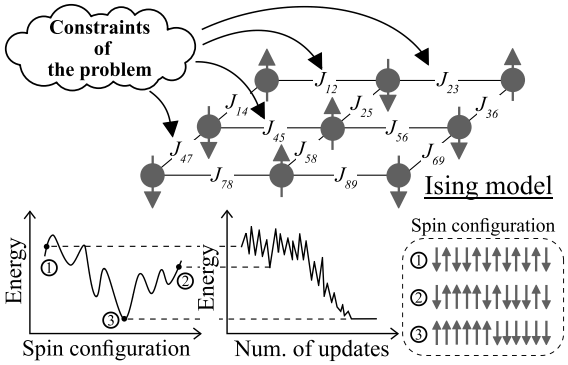

Figure 1: Ising model

We found that the prior FPGA-based implementations did not utilize on-chip memory blocks as known as " Block RAM " (or " BRAM " in short) in Xilinx FPGAs) on an FPGA effectively. Thus we focus on employing the on-chip memory block to increase the spin count.

The contributions of this papers are described as follows:

1. We present a time-division multiplexing architecture of Ising machine on an FPGA that utilizes on-chip memory blocks for the spin count increase. It enables to flexibly expand the spin count independently of the available logic circuit resources of an FPGA, such LUTs and registers.
2. We evaluated the efficiency of our proposal by using commercial FPGA synthesis tools. The evaluation results show that our architecture can handle 64 times more spin than previous one.
3. Based on our results, we discuss the another possibility of time-division architecture other than the spin count increase. By utilizing the time-division architecture, an Ising machine with a complex topology can be supported easily compared to the prior naive FPGA-based Ising machines.

2. CMOS ANNEALING MACHINE

2.1 Ising Model

$$H = -\sum_{<i,j>} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i \quad (1)$$

The Ising model is a statistical model representing the behavior of the spins of a magnetic material. The energy function of the Ising model is represented by the equation (1), where σ_i are individual spin states, j_{ij} are the interaction

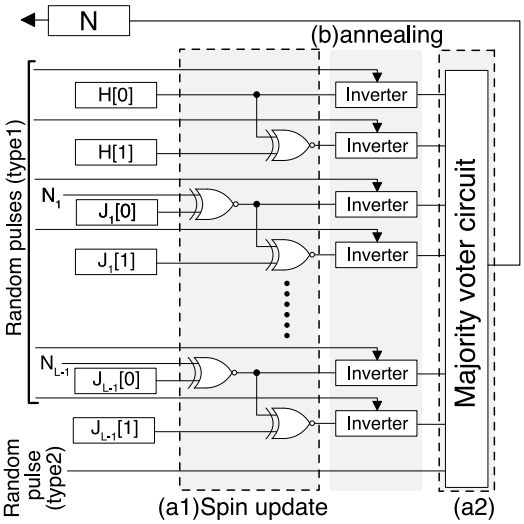

Figure 2: operator unit

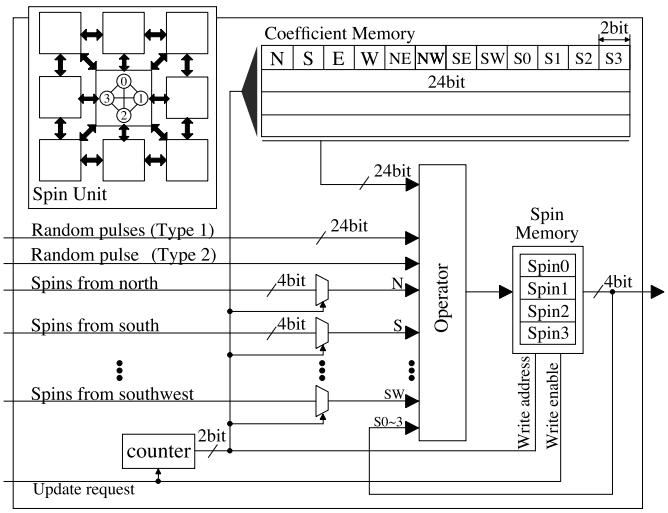

Figure 3: spin unit

coefficients that represent the strength of the interactions between different pairs of spin states, and h_i is the external magnetic coefficients. The Ising model has the property that the state of each spin is updated so that its energy function is minimized. Therefore, when a combinatorial optimization problem is mapped on to the Ising model, the combination of solution parameters that minimizes its energy can be observed naturally obtained after enough updates of spins as shown in figure 1.

2.2 Hardware Architecture

CMOS annealing is a technique to simulate the Ising model by CMOS curcuit. In CMOS annealing, the spin directions and interaction coefficients are held in binary $(+1, -1)$ and ternary $(+1, 0, -1)$, respectively. The interaction coefficient is stored with 2bits: the upper bit represents the sign, and the lower bit represent the absolute value. Using these values, the interaction effect between neighbor spins are simulated on the digital circuit.

Figure 2 shows the unit named operator for update states of spins. This unit receives the state of adjacent spins and interaction coefficients for spin updates. With these values, calculations are performed in each spin-to-spin connection through XNOR gates on figure 2 (a1). After that, the results are tabulated by the majority voter circuit in figure 2 (a2).In this way, the state of the next spin is determined. States of multiple spins can be updated at the same time as long as the spins to be updated are not connected. Therefore, even if the number of spins included in the Ising model increases, the number of updated spins which processed at the same time increases. In short, problem size (number of spins) has little influence on one update time.

Although the energy decreases according to the interaction operation described above, there is a possibility that the model is trapped in a local minimum which is not the overall minimum. To escape from the local minimum, the state of spins are randomly destroyed with a thermal fluctuation. This operation is performed by the inverters of (b) in the figure 2. The value obtained by the XNOR operation is inverted by the random pulses (depicted as "type 1"). The temperature decreases with each update and the inversion probability decreases as the temperature decreases. A random pulse (depicted as "type 2") is used when the number of 0 and 1 are equal. For details of both random pulses, see original papers [2]. This enables to escape from the local minimum by randomly transitioning the spin state and it is possible to find a state close to the ground state.

The optimal network depends problem. There are some structure for well known optimization problems such as a three-dimensional mesh structure, a chimera topology, and so on. In this paper, we utilize the chimeric topology [3]. In the chimera topology, four complete spins are contained within one spin unit. Each internal spin is coupled to the same position spin inside the adjacent spin units. In short, one spin is coupled with the other three spins in the spin unit and eight spins at the same position inside the adjacent spin unit as shown in the upper left of the figure 3.

The figure 3 shows the spin unit of the previous study. Since the spin unit holds for fully-coupled spins and the adjacent spins can not be updated at the same time, each spin unit use one operator while switching input to reduce resources. A 2bit counter decides the update spin and control unit and its value is used as the address of spin memory and coefficient memory and control signal of selectors. After that, the update spin is written back to the address indicated by the counter.

The update speed does not change even if the problem size is larger and that update of the whole model is completed with 4 sequential updates. Small FPGAs, however, can not contain a large scale problems, due to the hardware resource limitations. Since the degree of spins implemented in the hardware is 11, spin duplications based on the idea of graph minor [6] are required to implement a more complex problem that requires a larger degrees between spins. Such duplications unfortunately occupy the hardware spins redundantly, so that the size of implementable problems is diminished.

2.3 Motivation

The original implementation approach consumes a huge hardware resources. Especially the consumption of LUTs (Look Up Table) is linearly increased, when the target problem size is increased. We found that the original implementation does not utilize on-chip memory blocks (as known as "Block RAM" in Xilinx FPGAs). In this work, we explore another architecture that employs on-chip memory blocks

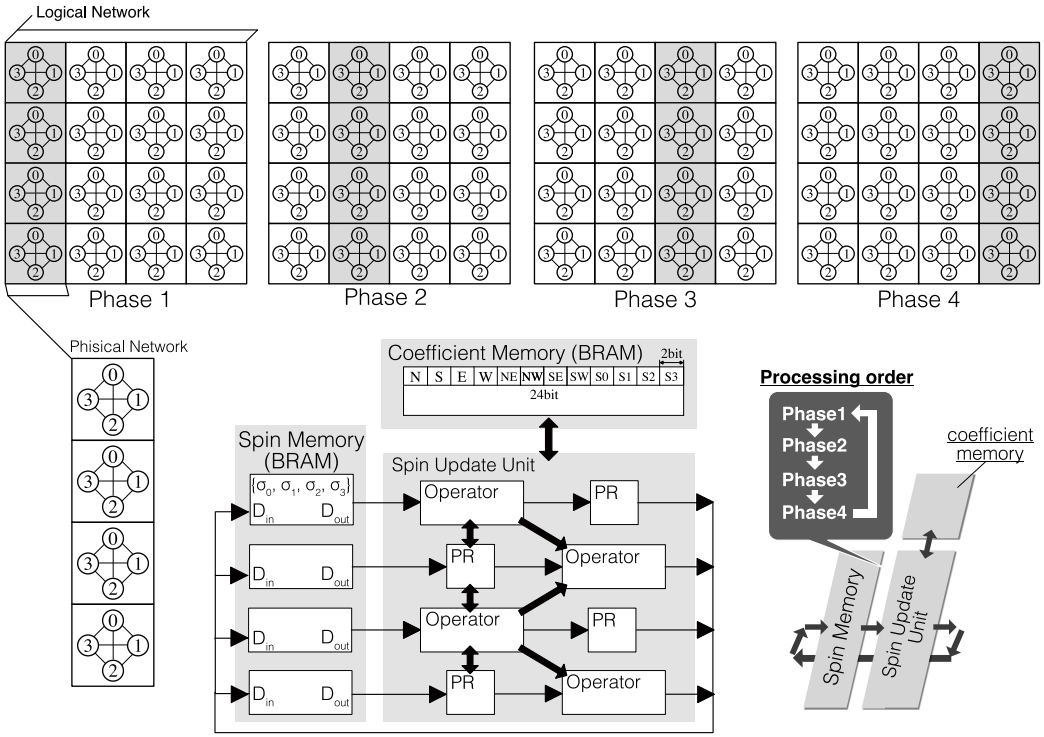

Figure 4: Time-Division Multiplexing Architecture

for increasing the spin count without additional hardware resource pressures. Especially, we investigate a flexible mechanism that decouples the logical Ising structure of a given optimization problem and the physical Ising structure realized on an FPGA as hardware circuit.

3. TIME-DIVISION MULTIPLEXING ARCHITECTURE

3.1 Architecture Overview

We propose a time-multiplexing architecture that processes large combinational optimization problems on limited hardware resources by separating spins of a target problem into both spacial and temporal directions.

Figure 4 presents the proposed architecture and processing mechanism. The architecture consists of a spin memory (Spin Memory) that stores the state of all spins, an interaction coefficient memory (Coefficient Memory) that stores the interaction coefficient between adjacent spins, and Spin Update Unit wihch updates the state of spins row by row including Operator / Pipeline Register (PR) array. In the Spin Update Unit , there is a constraint that adjacent spins can not be updated simultaneously [1]. therefore the pipeline registers are sandwiched between Operators.

The behavior of the architecture is as follows. As mentioned above, four complete spins (local spin) are contained within one spin unit in the chimera topology. This architecture updates sequentially from the local spin 0 in the chimera topology. A target Ising network is separated into 4 sections; We hereinafter call the section "Phase". First, spins in phase1 shown at the top of the figure 4 are read and transferred to the Spin Update Unit. the interaction coefficients of Phase 0 and Local spin 0 of Phase 2 are read from Spin Memory and Coefficient Memory, respectively. Then, they are transferred to the Spin Update Unit. Based on the read value, the Spin Update Unit updates the local spin 0 of phase 1. The back operator whose PR is inserted in the preceding stage receives the updated spin state from the front operator. This process is similarly performed up to Phase 4. After that, processing for local spin 1 is performed from phase 1 and it is repeated until all the local spins are updated. The architecture repeats this until the Ising model converges.

3.2 Scalability

It is possible to increase the spin count by dividing the Ising structure as long as it can be stored in BRAM. However, the more the process is divided, the more time it takes to update once. Thus, in order to improve the processing speed, it is necessary to allocate many Spin Units. Since the Ising model can update using information of adjacent spins, parallelism can be increased by arranging Operator / PR arrays in the column direction as long as the memory bandwidth of BRAM and circuit resources allows.

In this implementation, the processing unit is assumed to be one line, but depending on the problem it may be divided into rectangles. In such a case, it is necessary to consider the sharing of the boundary part of the processing unit. For example, when processing 8×4 networks on this architecture, First, the upper 4 rows are processed in the horizontal direction, then the lower 4 rows are processed in the same way. In processing the 4th and 5th rows, the states of spins of the 5th and 4th rows are required, respectively. Therefore, processing can not be performed because there is no mechanism sharing the boundary part of the vertical processing in the architecture described above. This problem can be solved by connecting the top and bottom (torus) operators in the Spin Update Unit so that the data at the boundary can be shared.

Table 1: Resource Utilization of Entire Design

	#. Spin Unit	LUT	FF	BRAM
Available		303600	607200	1030
conventional CMOS annealing	4 (2×2)	682 (0.22%)	833 (0.13%)	0.5
	16 (4×4)	1305 (0.42%)	905(0.14%)	0.5
	64 (8×8)	3897 (1.28%)	1193(0.19%)	0.5
time division CMOS annealing	4×4	663 (0.22%)	825 (0.14%)	4 (0.39%)
	8×8	832 (0.28%)	831 (0.14%)	4 (0.39%)

Table 2: Resource Utilization of each block

	random pulse generator				Spin Unit			
	#. Spin Unit	LUT	FF	BRAM	#. Spin Unit	LUT	FF	BRAM
conventional CMOS annealing	4 (2×2)	501	809	0.5	4 (2×2)	45	6	0
	16 (4×4)	597	809	0.5	16 (4×4)	42	6	0
	64 (8×8)	981	809	0.5	64 (8×8)	42	6	0
time division CMOS annealing	4×4	509	809	0.5	4×4	33	0	1.5
	8×8	547	809	0.5	8×8	34	0	1.5

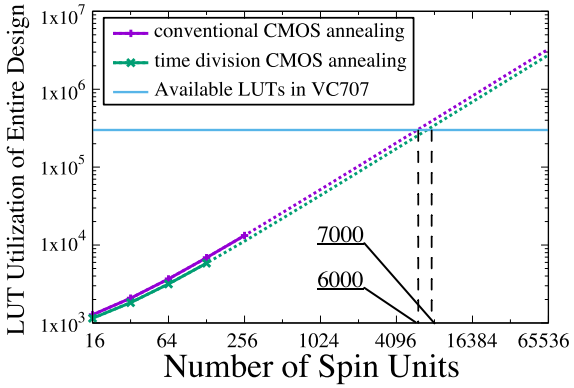

Figure 5: Estimation of LUT consumption

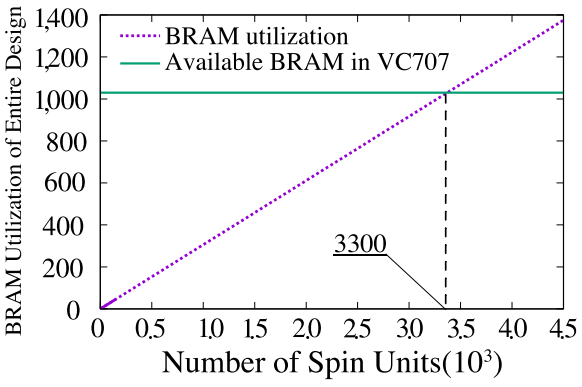

Figure 6: Estimation of BRAM consumption

4. EVALUATION

4.1 Setup

We evaluated the resource utilization of the conventional architecture and our architecture. Our target borad is Xilinx Virtex-7 FPGA VC707 evaluation kit (FPGA : Virtex-7 XC7VX485T). Both architectures were synthesized in verilog HDL and Vivado Design Suite 2016.2.

The dotted line in the figure 5 and 6 is extrapolated based on a small scale implementation. All spin spin units update at each clock. In the conventional CMOS annealing machine, all spins are updated in 4 clocks, whereas in the proposed architecture, it takes 4 clocks for one phase.

4.2 Resource Utilization

Tables 1 and 2 show the resource usage of the conventional architecture and our proposed architecture. Although the resource amount in the paper of the conventional architecture differs from the resource amount shown in this paper, it is thought that the minimum implementation was done in order to focus on the resource usage amount in this evaluation. As shown in the table, when the conventional architecture and the proposed architecture for the same problem size are compared, the number of Spin Units required for processing is reduced by the time division processing. Therefore, the total resource consumption of the proposed architecture is smaller than that of the conventional architecture Even when Spin Units are of the same size, the proposed architecture has less LUT and FF. it is considered that LUT and FF consumption per Spin Unit of our architecture are reduced from the resource consumption amount of each module in the table. One of the factors to reduce the resource consumption of the Spin Unit is to store the spin state and interaction coefficient in the BRAM. The other is the difference of the method of supplying the spin states and the interaction coefficients to the Operator. In the conventional architecture, data related to the spin to be updated is selected by the selector for supplying data to the Operator. On the other hand, since our architecture enables to acquire necessary data by just accessing the BRAM, selectors are unnecessary in front of each Operator in the proposed architecture.

4.3 Spin Count

Based on these results of syntheses, we estimated the number of Spin Units that can be deployed on two architectures on the target board. The results for the LUT are shown in figure 5, and the results for the BRAM are shown in figure 6. Note that the horizontal axis of figures is not the number

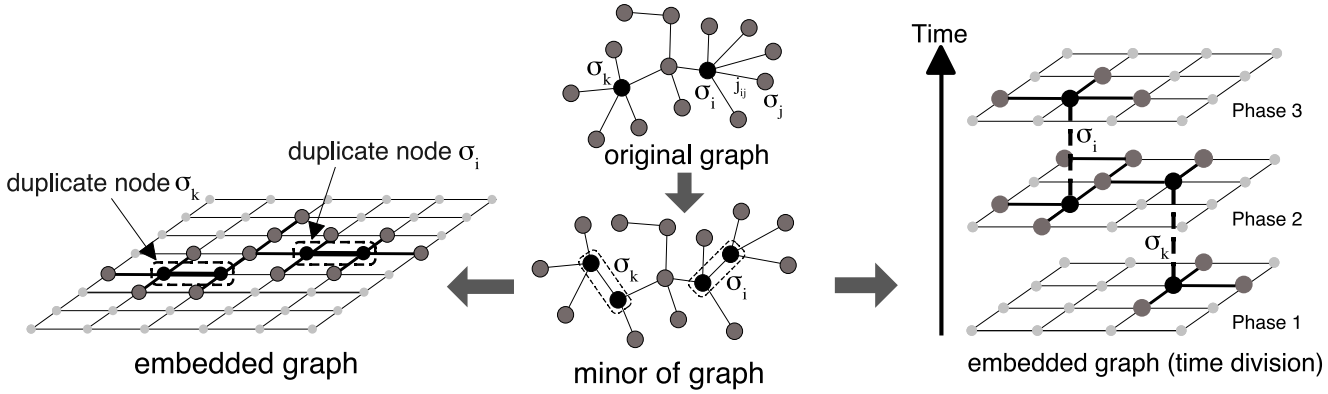

Figure 7: graph minor

of spins, but the Spin Unit (4 spins).

As can be seen from the figure, since the LUT consumption of the proposed architecutre is smaller than that of conventional method, the proposed architecture can allocate many Spin Units. However, even with the maximum use of 2000 BRAMs of 18 kb (1000 in the case of using 36 kb), there are about 3,300 Spin Units that can be deployed considering BRAM consumption in figure 6. In other words, compared to the conventional architecture, our architecture has the same number of spins by dividing the process into two, and it can handle the spin number over the conventional architecture with division of three or more. Further, if the Ising network is devided by 128, our architecture can hold about 211,000 spins 64 times larger than the conventional architecture.

5. DISCUSSION

5.1 How to Efficiently Utilize On-chip Memory Blocks

In this section, based on the results obtained up to the previous section, we will discuss future strategies of the FPGA Ising machine. The following is the knowledge obtained by this research.

- It is possible to increase the number of spins by using BRAM.
- Increasing the degree of division of the Ising network makes it enables to use BRAM effectively and to drastically increase the number of spins, however, the update speed decreases.
- When the degree of division is reduced and the degree of parallelization is increased, more BRAM blocks are consumed than the memory amount actually required. Therefore, the number of spins does not increase.

If the number of spins in the Ising network exceeds the number of spins can be deployed in conventional architecture, It is necessary to adopt BRAM based architecture. In that case, it is necessary to increase the degree of parallelism as much as the LUT permits with the minimum degree of division that can effectively utilize BRAM in order to achieve both the number of spins and the update speed. Considering that point, when researching the Ising model in a single FPGA board in the future, research on a more efficient BRAM base architecture or a method for canceling the overhead of time division may be important. In the next subsection, we discuss the latter.

5.2 Flexibility to Implementable Problems

Although the overhead caused by time-division multiplexing is a disadvantage of this architecture as described above, the advantage of the BRAM based architecture is that it has the stream processing architecture and the data are concentrated in memory. therefore, it is easy to observe the result of annealing.

Further, in the conventional architecture, the target Ising network (logical network) and the network of the Spin Unit on the FPGA (physical network) always have to correspond one to one. On the other hand, it enables to liberate a logical Ising structure from the physical hardware structure on an FPGA by using BRAM. It is necessary to think about a scheme that cancels disadvantages with this advantage.

First, we explain the case where performance such as processing speed and number of spins is degraded due to constraint of physical network in conventional architecture. In the case of processing the logical network (original graph at the center top of the figure 7) with hardware having a spin unit array of two dimensional lattices (physical network) as shown on the left, It is impossible to solve this problem directly with this hardware. It is because the degree of the spin i and j in the original graph is 5 while the maximum degree of the physical network is 4. Therefore, it is necessary to perform a graph minor that increases the degree of the vertex by duplicating the vertex (spin) as shown in the center lower part of the figure 7. In that case, the spin number of that minor is larger than the spin of the original graph. In addition, the duplicated spins must always be in the same state. Therefore, when one spin is reversed, it is necessary to set the interaction coefficient between replication spins to a large value so that the other spin also reverses. In such a case, there is a possibility that the possible value of the interaction coefficient of the conventional architecture may be insufficient or the convergence time and result may be affected. This is a disadvantage in the case where the logical network is restricted by the physical network.

On the other hand, in the case of the proposed architecture capable of separating the logical network from the physical network, connecting the duplicate nodes at the same

positions in different phases as shown in our architecture on the right side of the figure 7 make it possible to avoid lowering accuracy and convergence time. In this figure, 1 Phase corresponds to the physical network. When the process is switched from Phase 1 to Phase 2 after the processing in phase 1, in the duplicated spins connected by dotted lines, the intermediate results in phase 1 are retained and the majority decision processing is continued. Although it is impossible to prevent increase in spin due to duplication, it is not necessary to add new interaction, therefore it is possible to perform almost the same processing as the processing without minor operation on original graph by this mechanism.

If minor operation is not performed on the original graph, the degree of the physical network needs to be increased. Therefore, the complexity of the hardware increases. In addition, the constraint that adjacent spins can not be updated at the same time becomes a cause of a decrease in processing speed because increasing the order is equal to increasing the number of adjacent spins. From this fact, when solving the combinatorial optimization problem, firstly, the number of spins required for processing the problem network is obtained, Next, it is necessary to set the complexity of the physical network based on the degree of the graph and finally explore how to set the parallelism based on the depth of the BRAM and the LUT.

6. RELATED WORK

In recent years, several dedicated computers for Ising model base combinatorial optimization problem have been proposed. Some approaches that utilize natural phenomena such as quantum superposition [8] and optical parametric amplification [9]. Although these approaches are very fast in principle, there is a necessity to lower the temperature of the computer near absolute zero and a problem of miniaturization of the optical system.

In hardware architecture research, an Ising Chip [1] that simulates the behavior of the Ising model of a two-layer mesh structure with a CMOS circuit has been proposed. Research on the chimeric topology on the FPGA [3] (which is the comparison target in this paper) and the accelerating the simulated annealing on the FPGA [5] have been studied Further, studies on algorithms for embedding ising network onto such hardware in order to solve arbitrary problems [6], and research on random numbers to improve accuracy [2] have been made. Since these approaches introduce thermal fluctuation rather than quantum fluctuations, there are cases where accuracy is inferior to the approach using the above-mentioned natural phenomenon, but it is sufficiently fast and power efficient. In addition, Monte Carlo simulation of the Ising model [7] on FPGAs have also been proposed.

There are some FPGA-based accelerator implementations for stencil computations that utilize similar ideas and hardware structures as our time-division architecture for Ising model. Compared with stencil calculation [4] , stencil calculation uses only data at time t-1 for processing at time t. On the other hand, the processing in the Ising model annealing imposes not only time t-1 data but also time t data on the processing at time t, and constraints that adjacent spins can not be updated at the same time. In the time division multiplexing process, additionally, it is necessary to consider connections with different phases.

7. CONCLUSION

In this study, we proposed an annealing machine of Ising model using BRAM dedicated to combinatorial optimization problem. Compared with the architecture of the previous research, the number of spins that can be processed at the same time is halved because the read bit width of the BRAM is limited. however, by maximizing the use of BRAM, it is possible to deploy 64 times the spin units of the previous research.

In addition, we discussed efficient use of BRAM and advantage of time division multiplexing with BRAM. In the case where there is a need to solve the problem that can not be handled by the conventional architecture, the BRAM based architecture is indispensable. Therefore, research that alleviates the overhead due to time division processing is necessary.

In the future, we will study hardware topology and partitioning method required for actual applications. We would like to improve our proposed architecture as we presented in the discussion.

8. REFERENCES

[1] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, “ 20k-spin Ising chip for combinational optimization problem with CMOS annealing, ”ISSCC Dig. Tech. Papers, pp. 432 - 433, 2015.

[2] T. Okuyama, C. Yoshimura, M. Hayashi, and M. Yamaoka, “ Computing architecture to perform approximated simulated annealing for Ising models, ”in IEEE International Conference on Rebooting Computing, October 2016.

[3] C. Yoshimura, M. Hayashi, T. Okuyama, and M. Yamaoka, ”FPGA-based Annealing Processor for Ising Model”, International Symposium on Computing and Networking (CANDAR), 2016.

[4] K. Sano, Y. Hatsuda, and S. Yamamoto, “ Scalable streaming-array of simple soft-processors for stencil computations with constant memory-bandwidth, ” Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 234 - 241, May 2011.

[5] ”Fujitsu Laboratories Develops New Architecture that Rivals Quantum Computers in Utility”, http://www.fujitsu.com/global/about/resources/news/press-releases/2016/1020-02.html

[6] J. Cai, W.G. Macready, and A. Roy, ”A practical heuristic for finding graph minors,” arXiv prepring arXiv:1406.2741, pp.1-16, 2014

[7] Y. Lin, F. Wang, X. Zheng, H. Gao, and L. Zhang, “ Monte Carlo simulation of the Ising model on FPGA, ”Journal of Computational Physics, vol. 237, pp. 224 − 234, 2013.

[8] M. Johnson, M. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. Berkley, J. Johansson, P. Bunyk et al., “ Quantum annealing with manufactured spins, ”Nature, vol. 473, no. 7346, pp. 194 - 198, 2011.

[9] S. Utsunomiya, K. Takata, and Y. Yamamoto, “ Mapping of Ising models onto injection-locked laser systems, ” Optics express, vol. 19, no. 19, pp. 18 091 - 18 108, 2011.