IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 January 2024, accepted 21 January 2024, date of publication 30 January 2024, date of current version 9 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3360034

RESEARCH ARTICLE

Scalable Fully-Coupled Annealing Processing System Implementing 4096 Spins Using 22nm CMOS LSI

TAICHI MEGUMI, AKARI ENDO, AND TAKAYUKI KAWAHARA, (Fellow, IEEE)
Department of Electrical Engineering, Tokyo University of Science, Katsushika, Tokyo 125-8585, Japan

Corresponding author: Takayuki Kawahara (kawahara@ee.kagu.tus.ac.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 22H01559.

ABSTRACT Annealing processors specialized for combinatorial optimization problems are attracting attention. Although a general-purpose fully-coupled type is required for annealing processors in CMOS, the complexity of the coupling has led to scalability issues. Therefore, a scalable structure has been proposed that divides the calculation into multiple chips and has already been verified by using FPGAs. In the proposed system, 4,096 spins are implemented on a single board by 36 22-nm CMOS LSIs (512-spin fully-coupled annealing processor) and 1 FPGA for control. This is the largest fully-coupled annealing processing system implemented on a single board. This paper describes two main techniques used in the proposed system. The first is an approximate halving of the number of chips (from 64 to 36) by reducing the interaction matrix, and the second is an 8-parallel-solution search by parallel operation after reducing the number of chips. We found that a single LSI and a control FPGA can operate as a 512-spin fully-coupled annealing processor. Compared with CPU, the proposed system with 36 LSIs and control FPGAs improves computation speed by a factor of 34.0 and power performance by a factor of 2,344, while searching for 8 solutions in parallel.

INDEX TERMS Annealing processor, simulated annealing, scalable, interaction reduction, Ising machine.

I. INTRODUCTION

A. BACKGROUND AND OUR WORK

Combinatorial optimization problems (COPs) exist in various fields such as transportation, shift scheduling, and drug discovery. COP requires the best combination of solutions to be found among many alternatives under various constraints. COP is generally called NP-hard, and the larger the scale, the more the number of combinations explodes, making it significantly more difficult to find a solution. As the scale increases, COPs become impossible to solve in a realistic time by brute force on a classical computer. For example, the traveling salesman problem [1], which is typical for COP, has 1.27×10^{89} combinations in 64 cities (4096 spin fully-coupled annealing scale). The time to solve it brute-force on a classical computer (64-bit floating-point arithmetic, operating frequency 4 GHz) is on the order of $10^{89}/10^{9} = 10^{79}$ s, which is far from real time. Therefore, annealing processors have been attracting attention in recent years. An annealing processor is a new computer inspired by the natural phenomenon in which metal is heated to a high temperature and then gradually lowered to a low-energy state. In an annealing processor, the COP is represented by a physical model called an Ising model, which maps the combination of upward and downward spins to the solution state of the problem. The combination of spins that minimizes the energy of the entire Ising model is the optimal solution of the COP. The annealing processor searches for a solution by lowering the energy of the Ising model. Even if the optimal solution is not reached, the results obtained as a pseudo-solution can be used in the real world.

Annealing processors have been realized in various forms. There is quantum annealing [2], which applies quantum superposition states, and coherent Ising machines [3], which

The associate editor coordinating the review of this manuscript and approving it for publication was Woorham Bae.

use light. In particular, annealing processors in CMOS [4], which are realized in semiconductor digital circuits, have been the focus of much research because they can be realized at room temperature and without the need for large-scale equipment. As shown in Fig. 1, annealing processors can be broadly classified into two types on the basis of the coupling method between spins: fully-coupled and sparsely-coupled.

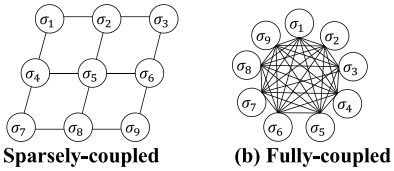


FIGURE 1. Coupling between spins in the Ising model.

The sparsely-coupled type has only partial couplings between spins and can be easily made scalable. However, the disadvantages are that the problem is difficult to map to the hardware and the scale of the problem that can be solved per number of spins is small. Fully-coupled annealing is the opposite: it has all the couplings between spins, which makes the problem easy to map to hardware, but it is not scalable because it has a square number of couplings per spin. In annealing processors in CMOS, the memory to store the couplings as data is large on the scale of the square of the number of spins. According to previous research, a fully-coupled scalable structure has been proposed for annealing processors in CMOS [5]. Although verified on a field-programmable gate array (FPGA), 384 spins are implemented by dividing the core part of the annealing calculation into 16 FPGAs and one more FPGA to control them. While the scalable structure limits the number of spins per chip, the number of spins can theoretically be increased without limit by increasing the number of chips.

The reason for implementing the system in an application specific integrated circuit (ASIC) instead of an FPGA is mainly to improve the scale and power consumption, and also to reduce the unit cost per chip, which will be effective for expanding the number of chips in the future. This paper describes two techniques applied to the proposed system: reducing the number of chips required by interaction reduction and improving solution accuracy by parallel processing of chips.

B. RELATED WORKS

Quantum annealing has high solution accuracy and computational speed, but requires a large cooler due to operation at extremely low temperatures [2]. Another disadvantage is that the implementation is sparsely-coupled. A fully-coupled system has also been proposed for a coherent Ising machine based on light, but the physical implementation scale is large [3].

Much research has been reported on CMOS bases due to their ease of implementation. In the field of analog circuits, Oscillator Ising Machines [6] and bistable latch-based Ising machines [7] have been proposed, but both are sparsely-coupled. In the field of digital circuits, FPGAs have been verified in various ways, including simulated bifurcation machines [8] and Network-on-Chip-based annealing processing [9]. In ASICs, several sparsely-coupled implementations have been proposed [10], [11] and 1.3 M spins [12] have been achieved using 40 nm CMOS. Fully coupled implementations are few, with 512 spins on a single chip using 28nm CMOS [4] and 2K spins on a 40nm CMOS multi-chip [13] being reported.

While fully-coupled implementations are often limited to FPGA verification, our work has the advantage of ASIC implementation of a fully-coupled scalable structure in a 22nm CMOS. In fact, it is the largest implementation on a single board, and can be further scaled up in the future.

II. SCALABLE FULLY-COUPLED ANNEALING PROCESSING SYSTEM (SFCAPS)

A. FULLY-COUPLED ANNEALING PROCESSOR IN CMOS

The annealing processor is based on the Ising model, a physical model of magnetic bodies. As shown in Fig. 2, the Ising model consists of spins that take two values of ± 1 (upward and downward), an external field that serves as a force in the direction of each spin, and an interaction that is the coupling weight between each spin.

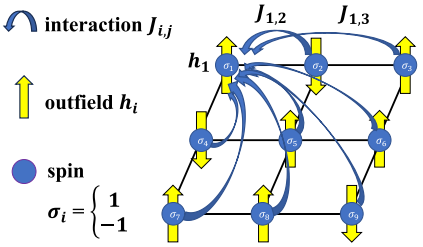


FIGURE 2. Ising model.

By using the i-th spin σ_i, the external magnetic field h_i, and the interaction $J_{i,j}$ between σ_i and σ_j, the energy of the Ising model becomes Eq. (1).

$$E = -\sum_{(i,j)} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i \tag{1}$$

By fitting the evaluation function of the COP to Eq. (1), which is the energy of the Ising model, the values of the interaction and the external field can be determined. σ_i is updated in the direction of lowering this energy.

The energy produced by the state of σ_i is Eq. (2).

$$E_i = -\sigma_i \left(\sum_j J_{ij}\sigma_j + h_i \right) = -\sigma_i \Delta E_i \tag{2}$$

where ΔE_i is set as Eq.(3).

$$\Delta E_i = \sum_j J_{ij}\sigma_j + h_i \tag{3}$$

We can update σ_i in the direction where E_i is always negative and thus with the same sign as ΔE_i. The spin update formula is Eq. (4) using the sign function.

$$\sigma_i' = \text{sign}\,(\Delta E_i) \tag{4}$$

However, if the spin is always renewed in the direction of lower energy, local solution trapping occurs, as shown in Fig. 3.

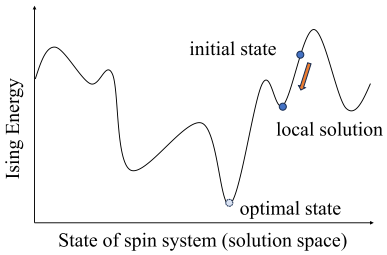


FIGURE 3. Local solution trap of energy.

To avoid the local solution trap, stochastic optimization methods exist. By stochastically accepting the direction of higher energy in the spin update, the local solution can be avoided. Simulated annealing (SA) is used in the Ising model. According to Eq. (4), SA always accepts when to flip the spin and also accepts the flip with probability P_T when not to flip the spin [14].

$$P_T = \exp\left[-\frac{|\Delta E_i|}{T}\right] \tag{5}$$

In Eq. (5), T represents temperature: the higher the T, the closer P_T is to 1, and the lower the T, the closer P_T is to 0. Theoretically, it is known that an exact solution can be obtained by decreasing T slowly enough to represent stochastic behavior. In one step (referred to as MC step), ΔE_i is calculated for all spins, spin updates are performed, and the temperature is lowered. The basic operation of the annealing processor is to proceed through MC steps until the temperature reaches zero.

In this research, pseudo-annealing (PA) is adopted to simplify Eq. (5) [4], and the spin update formula of PA is the sign function of Eq. (4) plus a temperature random number value.

$$\sigma_i' = \text{sign}\,(\Delta E_i \pm T) \tag{6}$$

Fig. 4 shows the pseudo code of the PA algorithm. In this research, the termination condition of annealing is given by the number of MC steps.

Annealing in CMOS uses a digital circuit to perform the operation shown in Fig. 4. Its circuit structure is mainly the storage of interaction matrices, external fields, and spin values and the calculation of ΔE_i. Fig. 5 shows the internal structure of a fully-coupled annealing processor with eight spins as an example.

As shown in Fig. 5 (b), to calculate ΔE_i, the interaction in row i is read out, summed with the spin, and added to the outer field. σ_i is updated as in Eq. (6), determined by the sign

```
Initializing
Generate initial the spin-state σ and Temperature T
// --------------------Pseudo Annealing----------------------
While MC steps < Loop Num
  For i == 0 to N do  //(N is number of spins)
    ΔE_i ← Σ_j J_ij σ_j + h_i
    p = random number  //(0~1)
    if p < 0.5 then
      T_calc ← T
    else then
      T_calc ← −T
    end if
    σ'_i ← −sign(ΔE_i + T_calc)
  end
  T ← T * temp_coefficient  //temp_coefficient < 0
  MC steps ← MC steps + 1
end While
```

FIGURE 4. Pseudo code of the PA algorithm.

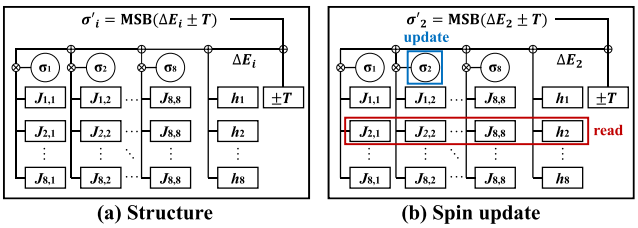


FIGURE 5. Structure of an annealing processor in CMOS.

of ΔE_i plus the temperature random number value. Here, the spin value is treated as a sign bit in the circuit, with +1 being 0 and -1 being 1. Therefore, the sign function is equivalent to taking the Most Significant Bit (MSB). The actual CMOS circuit also includes a controller that controls the calculation, reads out the interaction, lowers the temperature, and so on.

The above is the basic structure of an annealing processor in CMOS (PA implementation).

B. SCALABLE FULLY COUPLED ANNEALING PROCESSING SYSTEM (SFCAPS)

In fully-coupled annealing processor, the interaction in Fig. 5 is proportional to the number of spins, or in other words, to the square of the size of the problem that can be solved, making it difficult to scale up on a single chip. In a scalable fully-coupled annealing processing system (SFCAPS), the interactions are divided into n × n pieces and stored on a chip. Since ΔE_i is divided and calculated, a chip is needed to determine the spin update value by summing them. In previous research [5], 384 spins were implemented in 17 FPGAs: 16 to calculate the divided ΔE_i and 1 for control. Therefore, the 384 × 384 interactions are divided $4 \times 4 = 16$ and stored 96 × 96 (for 96 spins). As an example, Fig. 6 shows the structure of SFCAPS with an 8-spin 4-chip division.

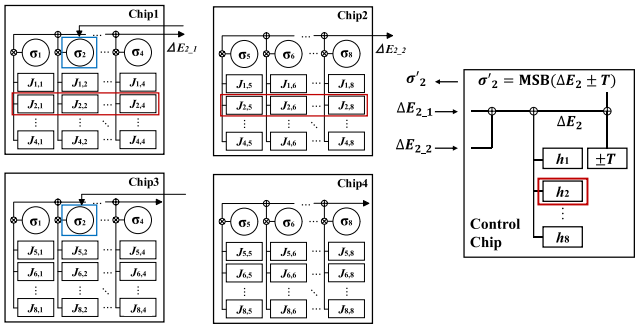


FIGURE 6. Structure of SFCAPS.

Since the interaction is divided in not only the row direction but also the column direction as shown in Fig. 6, ΔE_i is to be divided and calculated. If the calculated value of the Chip k is ΔE_{i_k}, ΔE_i becomes Eq. (7).

$$\Delta E_i = \sum_k \Delta E_{i\_k} + h_i \tag{7}$$

Depending on the spin number i, the behavior of each chip changes, and in the example in Fig. 6, the chip to be calculated changes as shown in Eq. (8).

$$\begin{cases} \sum_k \Delta E_{i\_k} = \Delta E_{i_1} + \Delta E_{i_3} \ (1 \le i \le 4) \\ \sum_k \Delta E_{i\_k} = \Delta E_{i_2} + \Delta E_{i_4} \ (5 \le i \le 8) \end{cases} \tag{8}$$

The spin update value is determined by collecting and summing ΔE_{i_k} by the control chip and adding the outer field and temperature random values. The spin update returns σ'_i for chips 1 and 2 for $(1 \le i \le 4)$ and chips 3 and 4 for $(5 \le i \le 8)$.

All four chips in Fig. 6 have the same circuit structure. Although the scale of a single chip is limited, it can be expanded by increasing the number of chips. This is the scalable structure.

III. PROPOSED SYSTEM

A. SCALABLE STRUCTURE OF THE PROPOSED SYSTEM

In the proposed system, a SFCAPS with 4096 spins is implemented using 36 large-scale integrations (LSIs) (referred to as Achips) designed in a 22-nm process and 1 control FPGA (referred to as CFPGA). SFCAPS requires multiple chips with the same circuitry and thus has a high affinity for ASICs. The major advantages of ASICs are circuit scale expansion and low-power consumption through circuit optimization. In addition, SFCAPS requires a large number of chips, and ASICs are effective in terms of future scaling and practical application because they can reduce the unit cost per chip.

In the previous research, 384 × 384 interactions were divided into 4 × 4 = 16 parts, but in the proposed system, 4096 × 4096 interactions are divided into 8 × 8 = 64 parts. The correspondence between the 8 × 8 division and the Achip

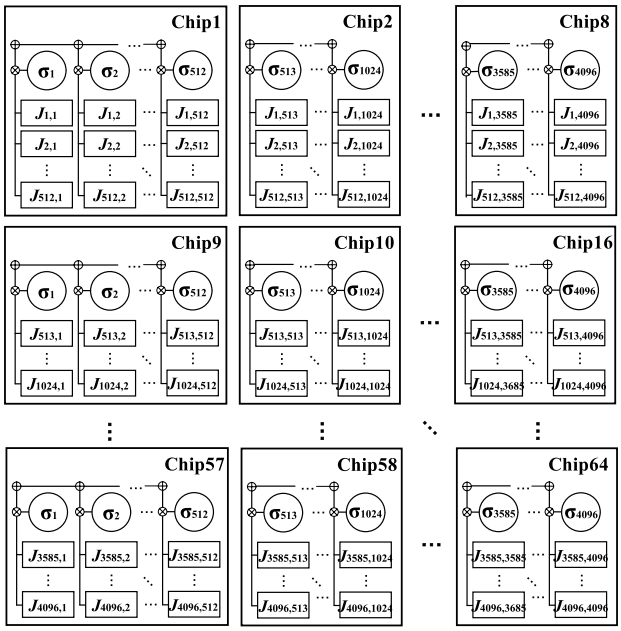


FIGURE 7. 8 × 8 division of the interaction and the corresponding Achip.

is shown in Fig. 7. As explained in Section III-B, the actual number of chips in the proposed system is reduced from 64 to 36 by the interaction reduction method. For simplicity, this section describes the scalable structure before the interaction reduction method is applied.

Fig. 8 shows the overall diagram of the proposed system. The proposed system consists of a group of Achips and a control FPGA (CFPGA) that controls them on a single board. The Achip calculates and transmits ΔE_{i_k} and receives σ'_i from the CFPGA. Fig. 9 shows the calculation flow of the proposed system. In proposed system, due to the 8 × 8 division of the

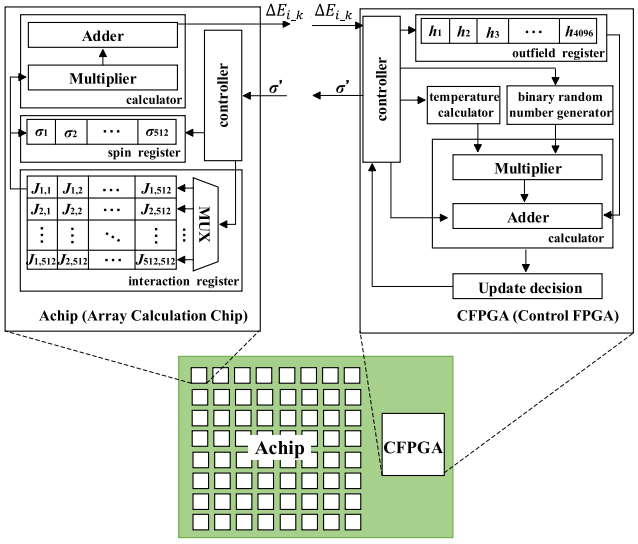


FIGURE 8. Overall diagram of the proposed system.

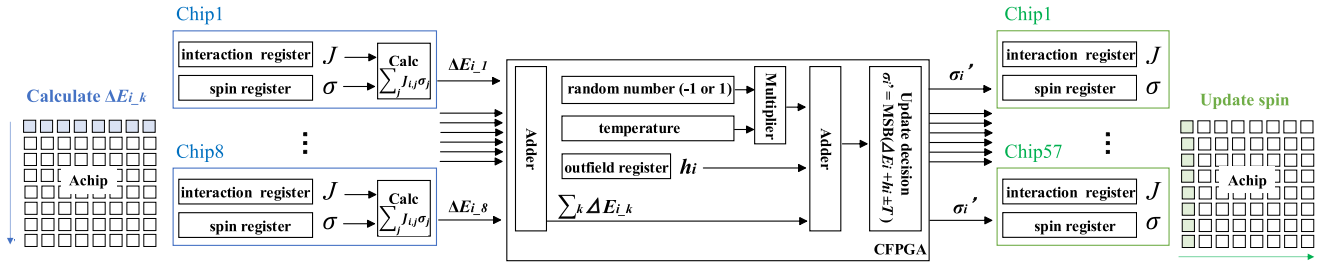


FIGURE 9. Calculation flow of the proposed system.

interaction, $\sum_k \Delta E_{i\_k} (1 \leq i \leq 512)$ is expressed by Eq. (9).

$$\sum_k \Delta E_{i\_k} = \Delta E_{i\_1} + \Delta E_{i\_2} + \Delta E_{i\_3} + \Delta E_{i\_4} \times \Delta E_{i\_5} + \Delta E_{i\_6} + \Delta E_{i\_7} + \Delta E_{i\_8} \quad (9)$$

As shown in Fig. 9, 1 Achip stores 512 spins, so the row of calculation and the column chip of update are switched for every $\boldsymbol{i}$ of 512. CFPGA decides from which Achip to receive calculation results and returns spin update values in accordance with the value of $\boldsymbol{i}$. After all 4096 spins have been updated, the temperature $\boldsymbol{T}$ of the CFPGA is multiplied by the temperature coefficient parameter (≤ 1) and decreased. The proposed system terminates the operation after the specified number of MC steps.

The above is the structure and operation of the proposed system before applying the interaction reduction method.

B. REDUCTION IN THE NUMBER OF CHIPS REQUIRED BY REDUCING INTERACTIONS

In the proposed system, the interaction of 4096 spins is divided into 64 (8 × 8 pieces), which would normally require 64 Achips. In this research, the interaction reduction method described in this section enables the system with 36 chips to be made.

The interaction matrix is the weight matrix between σ_i and σ_j, Eq. (10)

$$\begin{cases} J_{ij} = 0 & (i = j) \\ J_{ij} = J_{ji} & (i \neq j) \end{cases} \quad (10)$$

By using the characteristics of interactions with duplicates and zero elements as in Eq. (10), the interaction matrix can be reduced as shown in Fig. 10.

Such interaction reduction is highly compatible with SFCAPS, and previous research has been reported in FPGAs [15]. The idea is as follows: for a 2 × 2 division, the interactions are divided into (1)∼(4) as shown in Fig. 11.

(1) and (4), which store symmetric matrices, are reduced inside them using the same approach as in Fig. 10. In addition, since (2) and (3) are transposed matrices, the entire interaction block of (3) is reduced. The reduced interactions (1) and (4) are combined into a single chip, and the interaction (2) is stored in a single chip for implementation

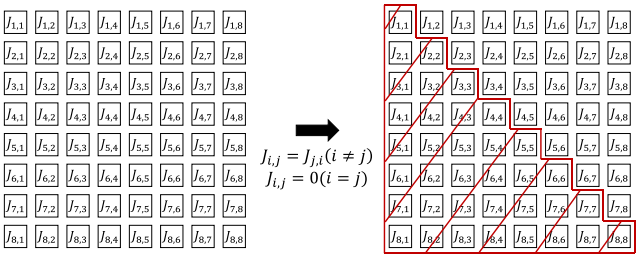


FIGURE 10. Reduction of interaction matrix.

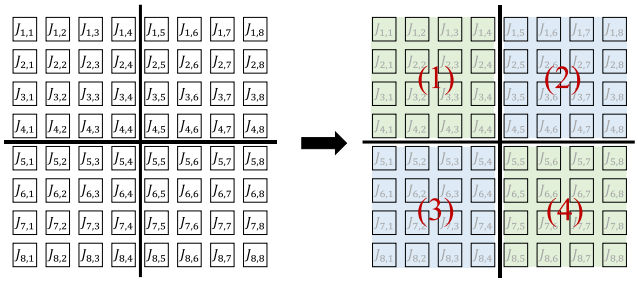


FIGURE 11. Divide and reduce interactions.

on two chips. As described above, the number of chips can be halved by reducing the number of interactions. However, the read control circuit of the interaction differs between the diagonal chips (1) and (4) and the other chips, requiring two types of chips. It is easy to design two different chips for implementation in FPGAs, but not for ASICs.

In the proposed system, the number of chips was reduced while allowing implementation with only one type of chip. First, as shown in Fig. 12, reduction was not applied to diagonal chips. Only chips in the transposed matrix are reduced, and 64 chips are reduced to 36 chips.

The structure was designed so that the interactions can be read out in row and column directions on all chips to compensate for the transposed matrix, including the diagonal chips. Fig. 13 shows the interaction readout structure of the Achip.

Note that Achip needs the spins that the reduced chip would have stored. There is an additional spin σ_{col} that takes a column readout and sums to the product. If we read out the interaction in columns and take the sum of products with

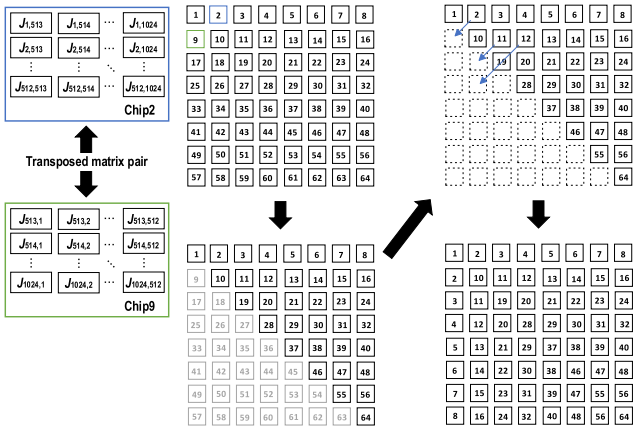
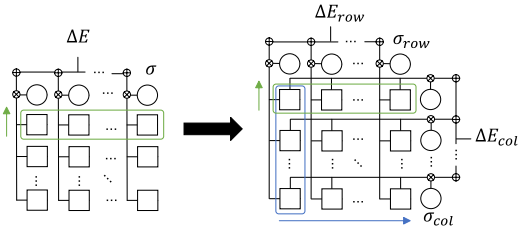


FIGURE 12. Chip reduction through interaction reduction.



(a) Read only in row direction (b) Read in row and column direction

FIGURE 13. Interaction readout structure of Achip.

σ_{col}, we can complement the reduced chip. Although only one calculator is needed, for implementing the spin-threads described in Section III-C, the structure is such that i rows and i columns of the interaction are read out simultaneously and ΔE_{row} and ΔE_{col} are calculated, respectively. Fig. 14 shows the relationship between the spins that each chip has and the ΔE_i it handles.

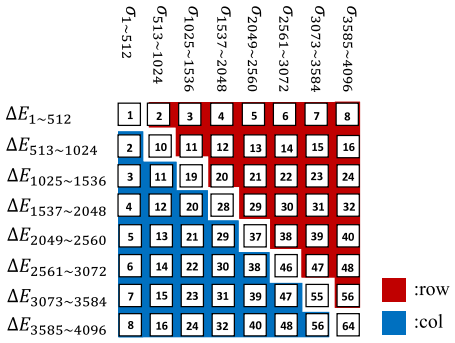


FIGURE 14. Spin and ΔE_i handled by each chip.

As can be seen from Fig. 14, the chips corresponding to n rows and n columns ($1 \leq n \leq 8$) are the same. Before the application of interaction reduction, the chips to be calculated and updated are not identical, but they are all made identical. The specific operation of the system excluding the diagonal chips is described below, focusing on Chip 2.

For $1 \leq i \leq 512$, the ΔE_{row} of the chip in the first row of Fig. 14 is used to calculate $\sum_k \Delta E_{i\_k}$. The spin update is the chip in row 1. Chip 2 has spins 513∼1024 in σ_{row} and 1∼512 in σ_{col}, and the i-th spin in σ_{col} will be updated.

Then for $513 \leq i \leq 1024$, the second row chip in Fig. 14 is used to calculate $\sum_k \Delta E_{i\_k}$. Chip 2 complements Chip 9 in Fig. 12, which stores the transposed matrix, so ΔE_{col} is used. Note that ΔE_{row} is used for all chips except Chip 2 in row 2. The spin is updated in Chip 2 in the second row, and in Chip 2, the i-th chip in σ_{row} is updated.

Chip 2 does not appear if i is 1025 or later.

The other Achip operations are also similar to those of Chip 2, which appears twice with 512 units before all spins are updated. The first 512 times are ΔE_{row} and the next 512 times are ΔE_{col}. In this case, the spin updates are σ_{col} and σ_{row}, in that order.

All above operations are controlled by the CFPGA side. It can control which ΔE_{row} or ΔE_{col} of which chip is acquired at each timing and which σ_{row} or σ_{col} is updated.

For the chips in Fig. 14 diagonal, either ΔE_{row} or ΔE_{col} should be used because they are symmetric matrices. If ΔE_{row} is used, spin update should be applied to σ_{row}.

As described above, it is possible to operate with only one type of Achip. The chip reduction ratio is 36/64, or 43.75%, which is not half, but as shown in Fig. 15, it can be implemented with only one type of Achip, which is close to half as the number of chips increases.

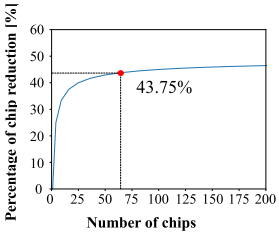


FIGURE 15. Relationship between the number of chips and chip reduction rate due to interaction reduction.

C. IMPROVED SOLUTION ACCURACY THROUGH PARALLEL PROCESSING OF CHIPS

In the proposed system, when the spin updates are performed sequentially starting from the first one, Achip runs for only eight chips of one ΔE group per calculation, as shown in Fig. 16 (ΔE_i and ΔE_{i_k} are used without distinguishing ΔE in Fig. 16 for simplicity of notation).

The remaining chips are idle and do not fully exploit the parallelism provided by the multi-chip. The proposed system employs spin-threads, which take advantage of the parallelism of multi-chip to search for multiple solutions simultaneously [4], [16]. The solution accuracy is expected to be improved by finally adopting the solution with the lowest energy. There is no previous example of introducing spin-threads while reducing the number of chips by interaction reduction in SFCAPS.

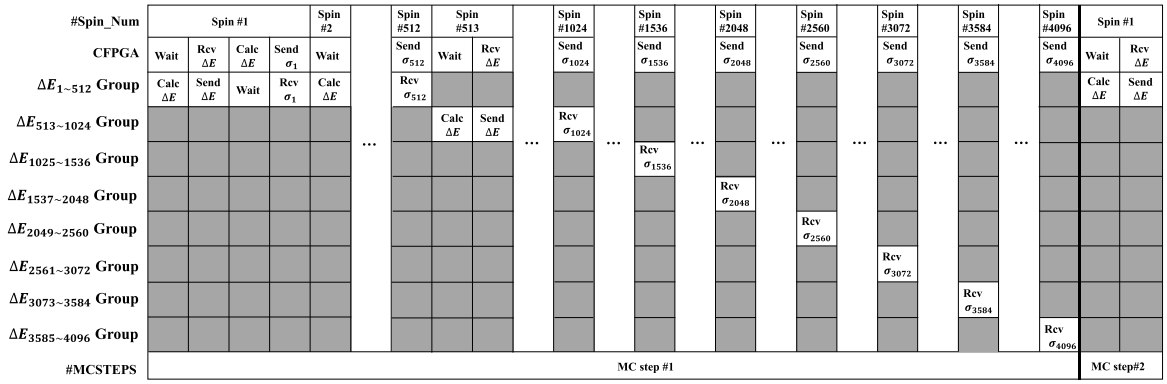


FIGURE 16. Simplified timing chart of the entire calculation.

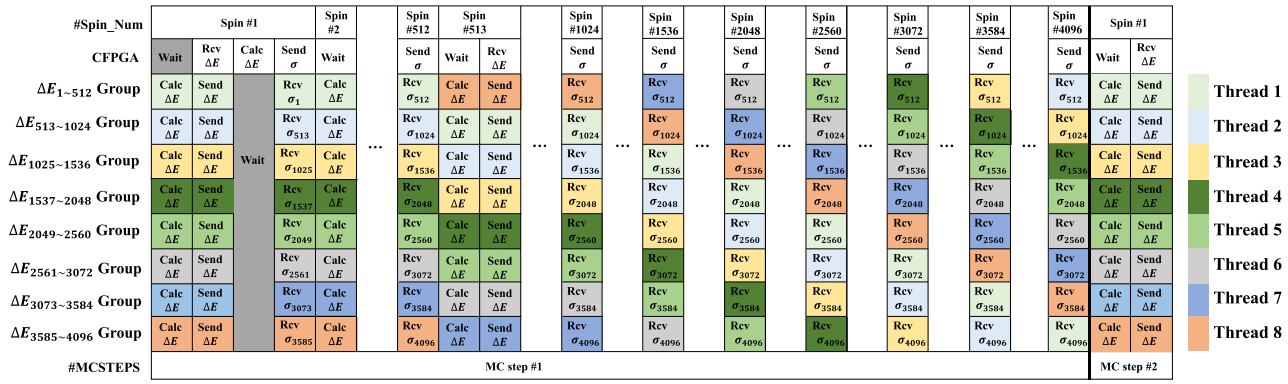


FIGURE 17. Timing chart for parallelization with spin-threads.

In the proposed system, as shown in Fig. 17, 8 color-coded threads are switched every 512 spin updates: thread 1 starting from the σ_1 update, thread 2 starting from the σ_{513} update, and so on. Focusing on thread 2, the spin updates for one cycle are performed in the order $\sigma_{513} \rightarrow \sigma_{4096} \rightarrow \sigma_1 \rightarrow \sigma_{512}$. Below, we explain the principle that the spins of eight threads can be updated at the same time. In SPIN#1, $\sigma_1, \sigma_{513}, \sigma_{1025}, \sigma_{1537}, \sigma_{2049}, \sigma_{2561}, \sigma_{3073}$ and σ_{3585} are updated. In this case, ΔE_1, ΔE_{513}, ΔE_{1025}, ΔE_{1537}, ΔE_{2049}, ΔE_{2561}, ΔE_{3073} and ΔE_{3585} need to be calculated. Therefore, CFPGA collects ΔE_{row} and ΔE_{col} from all chips simultaneously and adds them for each ΔE group. The ΔE_i is calculated by adding the external field to it, and the eight spin update values are determined from the temperature random numbers provided for each thread. The spin update values are returned to σ_{row} and σ_{col} of all Achips at the same time, as shown in the correspondence in Fig. 14.

Until SPIN#512, the spin update values determined from the same ΔE group are returned. For #SPIN#513, thread 1 requires ΔE_{513} and thread 2 requires ΔE_{1025}, and the ΔE groups required by each thread will shift as shown in Fig. 17.

Next, we will explain the operation inside Achip. As shown in Fig. 18, Achip has 8 spin-threads for both σ_{row} and σ_{col}, and internally switches the thread used for every 512 spin updates.

Since Achip complements the chip reduction, different threads will be used in σ_{row} and σ_{col} at the same time.

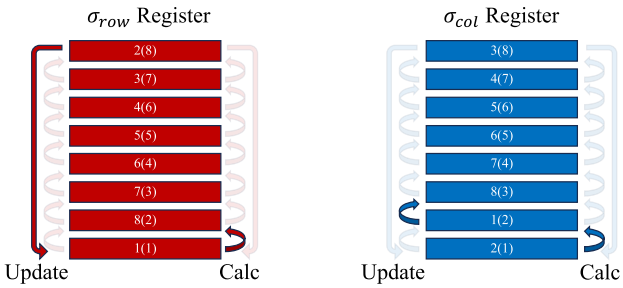


FIGURE 18. Switching threads inside Achip (Chip 2).

For example, Chip 2 calculates ΔE_{row} for thread 1 and ΔE_{col} for thread 2 in SPIN#1∼SPIN#512. In this case, the spin update is thread 2 for σ_{row} and thread 1 for σ_{col}. In Fig. 18, the threads recognized inside Achip are written in "()" for the threads actually used. Since Achip only knows the internal thread number, Achip needs to be made to recognize which internal thread to use. Eq. (11) is used to obtain the thread to be updated (in fact, because it is a digital circuit, the calculation is based on 0).

$$\begin{cases} Thread_{row}^{update} = mod\left(7 + Thread_{row}^{calc} - x, 8\right) + 1 \\ Thread_{col}^{update} = mod\left(Thread_{col}^{calc} + x - 1, 8\right) + 1 \end{cases} \tag{11}$$

The value of x ($x \leq 7$) in Eq. (11) is different for each chip: in Chip 2, $x = 1$. The update thread for Chip 2 is Eq. (12) when SPIN#1∼#512, which means that the position of the internal thread matches that of the actual thread.

$$\begin{cases} Thread_{row}^{update} = mod\ (7+1-1, 8)+1 = 8 \\ Thread_{col}^{update} = mod\ (1+1-1, 8)+1 = 2 \end{cases} \quad (12)$$

In the proposed system, the value of x is sent to each chip at the time of data loading.

With the above principle, spin-threads can be implemented while applying interaction reduction, and eight solutions can be obtained simultaneously without changing the computation time.

IV. ACHIP CIRCUIT AND OPERATION DETAILS

The block diagram of Achip is shown in Fig. 19.

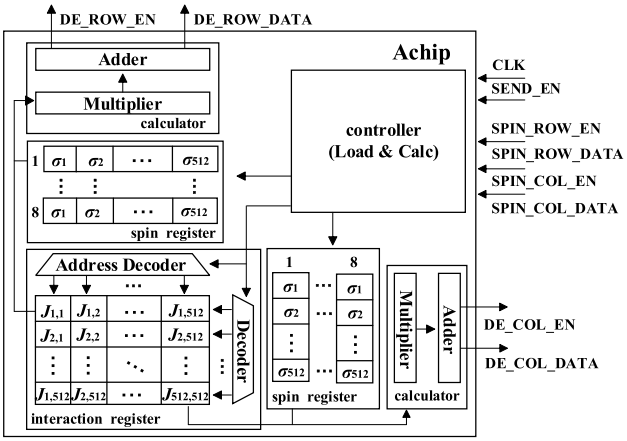


FIGURE 19. The block diagram of Achip.

The Multiplier in Fig. 19 has a simple structure. As shown in Fig. 20, the sign of the interaction $J_{i,j}$ that is passed through the calculator is determined according to the value of the spin σ_j (inverted at $\sigma_j = 1$).

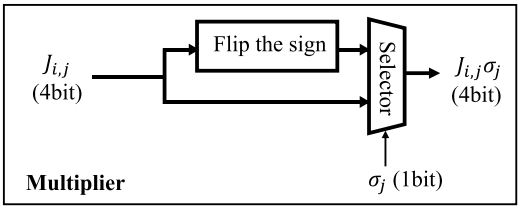


FIGURE 20. Multiplier details.

The Achip circuit has two main features. First, the interactions are read simultaneously in two directions (row and column), and each is summed with the spins. Second, the spin registers are duplicated into eight for the spin-thread implementation. In fact, the interaction registers occupy most of the area, and the spin registers, which are 16 times larger than the original size due to the interaction reduction and spin-threads, account for less than 0.8% of the interaction registers. The controller generates read/write addresses for each register and also controls the entire system.

A simplified timing chart of the Achip is shown in Fig. 21.

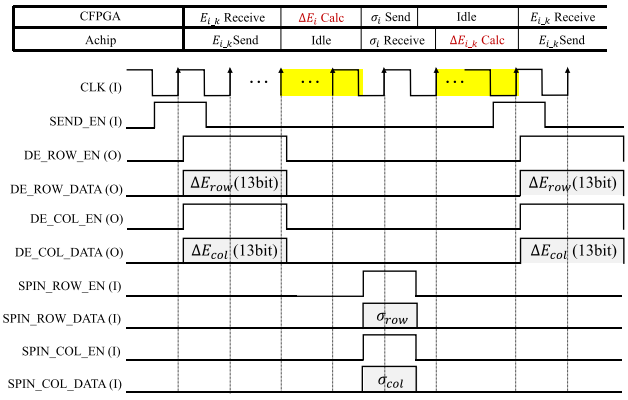


FIGURE 21. Simplified timing chart of the Achip.

All I/Os on the Achip are 1-bit wide signals and communicate serially with the CFPGA. ΔE_{i_k} is the sum of the product of 512 signed 4-bit (-8∼7) interactions and spins {-1,1}, so it is set to 13 bits (can represent -4096∼3584). The internal registers operate on the rising edge of clock, and input data from the CFPGA is sent on the falling edge. The operation of the Achip is simple. It sends ΔE_{row} and ΔE_{col} at the rising edge of SEND_EN. After that, σ_i is received from the CFPGA that calculated ΔE_i. One clock is required to receive σ_i at the controller, and another clock is required to store it in the spin register. Therefore, Achip, which completes the reception operation of σ_i in two clocks, uses an asynchronous calculation system, and calculation starts as soon as the reception operation of σ_i is completed. The calculation time is until SEND_EN rises again. The yellow highlighted area in Fig. 21 indicates that either Achip or CFPGA is performing the calculation. Since the CFPGA can change the timing of sending SEND_EN and the timing of starting ΔE_i calculation, Achips do not need to all run the same operation with the same clock.

Next, the interaction register, which is the most characteristic circuit structure of the Achip, is explained. To optimize the layout, the interaction register is configured by arranging 512 × 512 units, with the circuit shown in Fig. 22 as the smallest unit.

In this research, there are four registers because the interaction is four bits. The write operation to the registers is simple. When both EnW_row and EnW_col generated by the higher-level module are 1, Write Enable is set to 1, and DataW[3:0] is written at the rising edge of the clock.

The interaction register is an asynchronous read structure, and the output always changes with the input signal. The read enables EnR_row and EnR_col to rise simultaneously in row i and column i in Fig. 22. Fig. 23 shows the operation of the $J_{1,2}$ and $J_{2,2}$ registers from $i = 1$ to 4.

The operation is to output the value of its own register in the direction of 1 for read enable, and when 0, it simply

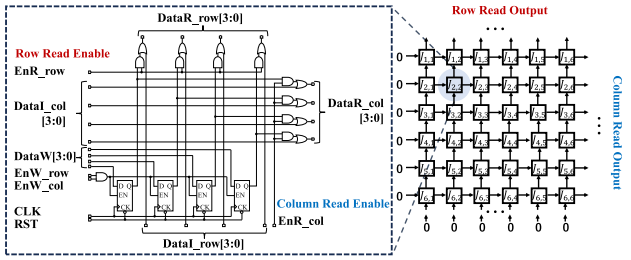


FIGURE 22. Circuit structure of the interaction register.

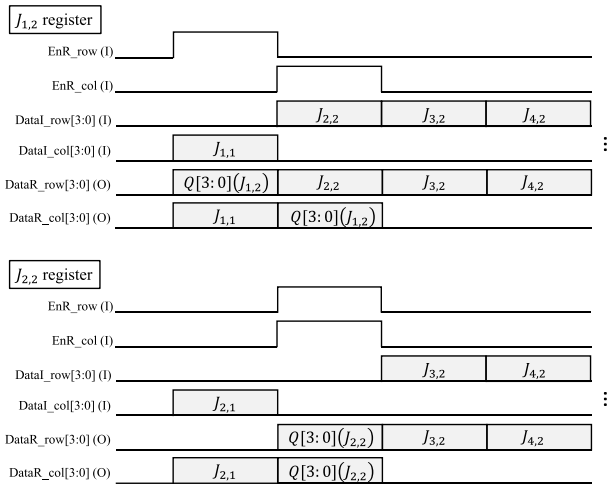


FIGURE 23. Timing chart of $J_{1,2}$, $J_{2,2}$ register ($1 \le i \le 4$).

flows the DataI input from the neighbor (bottom in Fig. 22 for row, left for col). Registers at the ends where DataI does not exist, such as DataI_col in J_1,1, are input 0. For registers other than the diagonal registers, EnR_row and EnR_col do not become 1 simultaneously as in $J_{1,2}$, but the diagonal registers become 1 simultaneously as in $J_{2,2}$ and are read in both directions.

As described above, the interaction register has a structure that can be laid out by simply arranging and coupling 512 × 512 circuits of the smallest unit shown in Fig. 22 in a matrix.

V. IMPLEMENTATION EVALUATION OF THE PROPOSED SYSTEM

A. COPs USED IN THE EVALUATION

In this research, the operation of a single Achip was verified and the proposed system with 36 Achips was evaluated. One Achip was verified by implementing 512 spins with 1 Achip and CFPGA on a test board. The Max Cut problem was adopted for each evaluation. As an addition, the proposed system also solved the Vertex Cover problem. Each problem is briefly described below.

In the Max Cut problem, an undirected graph $G = (V, E)$ and a weight $w_{u,v}$ for each edge are given. We divide the vertices into two subsets, V_1, $V_2 (V = V_1 \cup V_2)$, and cut an edge when the vertices at both ends of the edge belong to different subsets. Maximizing the sum of these cut values (the weights $w_{u,v}$ of the edges to cut) is the Max Cut problem.

We define a variable, σ_v, that represents to which subset the vertex $v \in V$ belongs.

$$\sigma_v = \begin{cases} 1 & (v \in V_1) \\ -1 & (v \in V_2) \end{cases} \tag{13}$$

The cut value C is expressed in Eq. (14).

$$C = \frac{1}{2} \sum_{(u,v)\in E} w_{u,v} (1 - \sigma_u \sigma_v) \tag{14}$$

The annealing processor defines the energy by multiplying Eq. (14) by -1 to calculate the minimum energy.

$$E = -\frac{1}{2} \sum_{(u,v)\in E} w_{u,v} (1 - \sigma_u \sigma_v) \tag{15}$$

In the Vertex Cover problem, given an undirected graph $G = (V, E)$, a subset of V, the problem is to find, for every edge, at least one vertex on each side of the edge that is included in that subset and has the smallest number of elements. We define a binary variable, σ_v, that represents to which subset the vertex $v \in V$ belongs. When the variable x_v is introduced, the energy is expressed in Eq. (16) with A and B as positive constants.

$$E = A \sum_{uv\in E} (1 - x_u)(1 - x_v) + B \sum_{v\in V} x_v \tag{16}$$

The first term in Eq. (16) represents the constraint that for all edges, at least one of the two vertices at each end of the edge must be 1. The second term is an objective function that reduces the number of x_v that are 1. Although x_v is treated here as a binary variable, it must be transformed from $x_i = (\sigma_i + 1)/2$ because it is represented by a spin that takes ± 1 in an annealing processor.

In this research, we used "Rudy," a random graph generator for Gset, which is commonly used as a benchmark for Max Cut problems, for the 512-spin problem of evaluating a single Achip with an undirected graph $G = (V, E)$. The number of vertices was set to 512, the probability that each vertex had an edge on another was set to 4%, and the seed value was set to 42 (number of edges: 2616). For the 4096-spin problem, the same parameters were set to 4096, 2%, and 42, respectively (number of edges: 167,731). Also, all weights used in Max Cut were set to 1.

B. EVALUATION OF ACHIP

In this research, Achip was designed in a 22-nm CMOS process. Fig. 24 shows a micrograph of the Achip and its layout.

The interaction registers arranged in matrix form occupy most of the layout area of 2,960μm in height and 1,562μm in width. In this research, registers are used only for the placement and wiring of standard cells, but if SRAM is used, area and power consumption are expected to be improved.

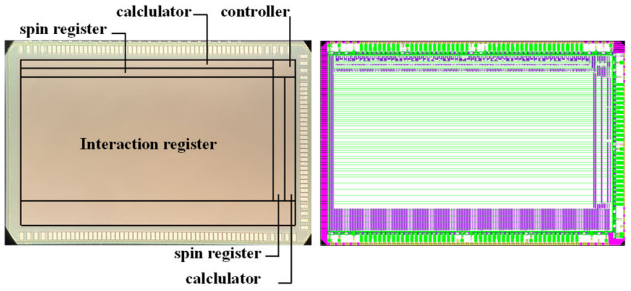
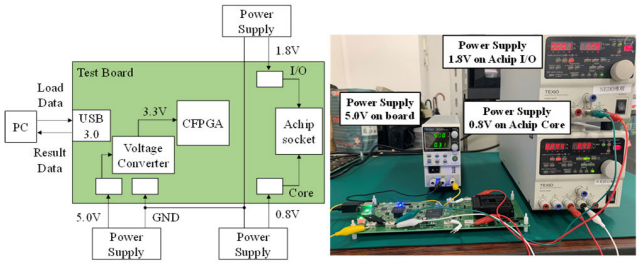


FIGURE 24. Achip micrograph and layout.

The interaction registers are read out in rows and columns, with spin registers and calculators in each direction. The spin registers and calculators are designed to be the size of the interaction rows and columns, simplifying the wiring on the layout. As explained so far, the spin register has eight threads in each direction.

Next, the evaluation environment of the Achip is shown in Fig. 25 and Achip's specifications in Table 1.



(a) Schematic diagram (b) Photograph

FIGURE 25. Evaluation environment of Achip.

TABLE 1. Specifications of Achip.

Technology	22nm CMOS
Chip Size	3mm × 2mm
Op. Freq [MHz]	10MHz
V_{DD} [V] (I/O / Core)	1.8 / 0.8
Op. Current [mA] (I/O / Core)	44 / 1
Op. Power [mW] (I/O / Core)	35.2 / 1.8

As shown in Fig. 25, 1 Achip and CFPGA are implemented on the test board, able to perform 512 spins of annealing. USB 3.0 is used for communication with the PC to load parameters such as interactions and to output the spin values held by the CFPGA (the same applies to the proposed system). The core power consumption was 35.2 mW when solving the fully-coupled Max Cut problem at 10 MHz. Fig. 26 shows the energy transition diagram for the Ising model.

Fig. 26 compares the results of emulating a PA under the same conditions on a CPU using MATLAB R2022a with the actual machine (8 threads). Although the CPU is

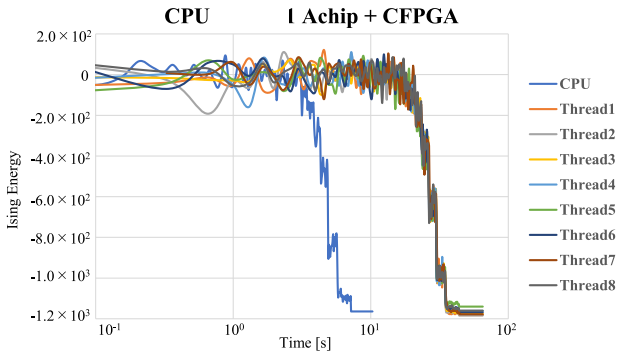


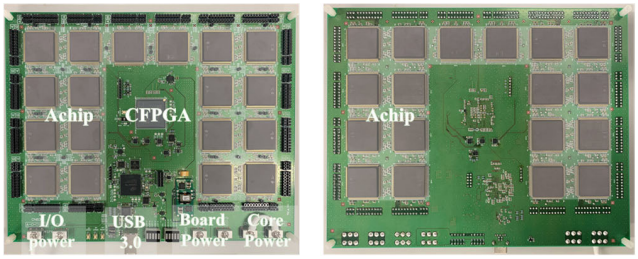
FIGURE 26. Energy transition diagram when solving 512-node Max Cut problem with one Achip and CFPGA (CPU comparison: Intel Core i7-4790) Number of plots: 1/50.

64-bit floating point and the interaction of the proposed system is 4-bit integer, exactly the same calculations are performed. The reason for this is that the interaction of the solved Max-cut problem is binary {0,1}. The argument is similar for the Max-cut problem used in the benchmark and the vertex covering problem, which can also be represented by a signed 4-bit.

The annealing conditions were set to an initial temperature of 512, 10,000 MC steps, and temperature coefficient of 0.9995 for each MC step. The temperature coefficient is the same value used in the results to be presented hereafter. Achip cannot operate only one thread due to its internal switching operation of eight spin-threads as shown in Fig. 18. Therefore, the 512-spin problem simply takes 8 times longer to obtain 8 solutions. The Achip behavior of the verification here is exactly the same as that in the proposed system. In the proposed system, all Achips have this 512-spin scale annealing operation.

C. EVALUATION OF THE PROPOSED SYSTEM

A board photo of the proposed system is shown in Fig. 27.



(a) Front side (b) Back side

FIGURE 27. Photograph of the board of the proposed system.

In the proposed system, 18 Achips and CFPGA are placed on the front side of the board, and 18 Achips are placed on the backside. The CFPGA is also connected to the Achip on the back side and exchanges signals with each chip during calculation as shown in Fig. 21.

The evaluation environment and specifications of the proposed system are shown in Fig. 28 and Table 2, respectively.

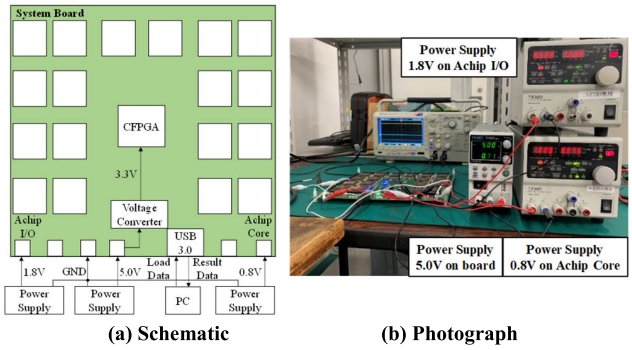


FIGURE 28. Evaluation environment of the proposed system.

TABLE 2. Specification of proposed system.

Entire System	Achip36 + The others (Include CFPGA)
Device (Achip / CFPGA)	22nm CMOS / XC7S100-2FGGA676I
Op.Frequency [MHz]	10
Voltage [V] (Achip I/O / Achip Core / The others)	1.8 / 0.8 / 5.0
Op.Current [A] (Achip I/O / Achip Core / The others)	1.608 / 0.033 / 0.31
Op.Power [W] (Achip I/O / Achip Core / The others / Entire System)	1.29 / 0.059 / 1.55 / 2.90

In the proposed system, the Achip core power at 10 MHz was 1.29 W, and the total system power was 2.90 W. In Fig. 28, 1.830 V is applied to the Achip I/O and 0.975 V to the core. This is due to a voltage drop caused by problems with the board and measurement environment, and actual 1.8V and 0.8V are applied to the Achip, respectively.

Fig. 29 shows the energy transition diagram for the Max Cut problem solved for 4096-node.

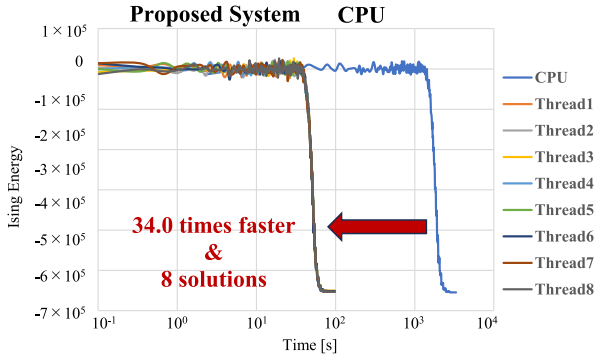


FIGURE 29. Energy transition diagram when solving 4096-node Max Cut problem with proposed system (CPU comparison: Intel Corei7-4790) Number of plots: 1/50.

The annealing conditions were initial temperature of 512 and MC step of 15,000 cycles. The proposed system was able to obtain 8 solutions in 34.0 times faster computation time than the CPU. Although 1 Achip took longer than the CPU, the proposed system was able to solve the 4096-spin problem in 8 parallel spins in the same computation time as 1 Achip. In the proposed system, Achip alone solves a 512-spin problem, but the system as a whole can solve a 4096-spin problem. This is the advantage of scalability.

Fig. 30 shows the energy transition diagram for the Vertex Cover problem.

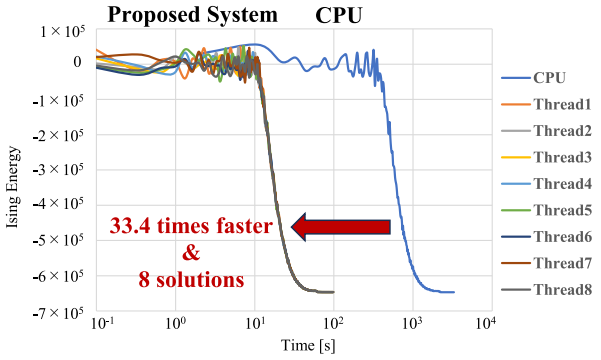


FIGURE 30. Energy transition diagram when solving 4096-node Vertex Cover problem with proposed system (CPU comparison: Intel Corei7-4790) Number of plots: 1/50.

The annealing conditions were initial temperature of 1024 and MC step of 15,000 cycles. The proposed system was able to obtain 8 solutions in 33.4 times faster computation time than the CPU.

Although the operating frequency is limited to 10 MHz due to the board specifications, the computation speed can be significantly improved by increasing the operating frequency.

At 10 MHz, the number of clocks required for ΔE_{i_k} communication is longer than that for calculation, but this can be improved by increasing the operating frequency.

To demonstrate the effect of the spin-thread, Table 3 shows eight solutions to the Max Cut and Vertex Cover problems obtained by the proposed system. For the Max Cut problem, the more cuts, the better the solution, and for the Vertex Cover problem, the fewer vertices (spins) that become 1, the better the solution.

As shown in Table 3, the proposed system can search for eight solutions in parallel, showing that it can select a better solution than when only thread 1 is used as the solution.

TABLE 3. Improvement of solution accuracy by spin-threads.

	thread 1	thread 2	thread 3	thread 4	thread 5	thread 6	thread 7	thread 8
1. 4096-node Max Cut (Cut value)	97640	97669	97616	97698	97659	97646	97696	97685
2. 4096-node Vertex Cover (Number of 1's)	3804	3795	3800	3801	3795	3800	3796	3798

1. The bigger the better 2. The smaller the better

Next, Fig. 31 shows the results of the power comparison with the CPU, representing the case when the Max Cut problem was solved.

In Fig. 31, the proposed system and the entire PC are compared for the same subject. Also, in core part, the power of CPU only and 36 Achips only (Core + I/O) are compared. The proposed system shows 2344 times higher power performance ratio than CPU and can search for 8 solutions in parallel. A 2223 times higher power performance ratio is also observed in the core part comparison.

	CPU PA Simulation	This work
Op. Frequency [MHz]	3600	10
Op. Power [W] (Core part)	200 (85)	2.9 (1.3)
Time [s]	3341.5	98.3
Power consumption [J] (Core part)	657444 (279414)	285 (128)

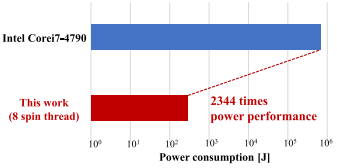


FIGURE 31. Power comparison between CPU and proposed system solving 4096-node Max Cut problem.

These are the results of 4096 spin operations. For comparison with other research [6], [9], Table 4 shows the results of solving the Max-cut problem (G1, G25, G46, G48) for the 800, 1000, 2000, and 3000 nodes of the Gset benchmark. The problem was solved by setting the 4096-spin interaction matrix to 0 except in the region where it is used. With an initial temperature of 512 and 15000 MC steps, the computation time is 98.3s. Since the proposed system provides 8 solutions, the best solution is selected. In Table 4, the average and maximum values of the results of solving the problem for 10 times are shown for the theoretical and actual machine, respectively.

TABLE 4. Theoretical and actual benchmark results of the proposed system (average and maximum of 10 times).

Max-cut Problem	Spin Number	Best Known Result	This Work Theorical / Measured		Dynamic Simulated Annealing [9] (Accuracy [%])
			Mean (Accuracy [%])	Max (Accuracy [%])	
G1	800	11624	11623.9 / 11619.1 (99.99 / 99.96)	11624 / 11621 (100 / 99.97)	11624 (100)
G25	2000	13340	13323.2 / 13305.1 (99.87 / 99.73)	13326 / 13314 (99.90 / 99.81)	13313 (99.80)
G46	1000	6649	6642.5 / 6633.8 (99.90 / 99.77)	6646 / 6642 (99.95 / 99.89)	6646 (99.95)
G48	3000	6000	6000 / 5995.2 (100 / 99.92)	6000 / 5996 (100 / 99.93)	6000 (100)

In the actual machine, the accuracy was reduced due to defects in some circuits, but accuracy in the 99th percentile was obtained for all benchmarks. Other work [9] computed with parallel spin updates and had comparable solution accuracies. In addition, although [8] is not an annealing method, it is capable of parallel operations and solves benchmarks faster on GPUs. Therefore, we would like to consider parallel updates or other speedups appropriate to the structure of the proposed system.

Based on the above discussion, we once again describe the strengths of the proposed system. The proposed system is the first multi-chip scalable structure implemented in ASICs. It is clear that ASIC implementations consume less power than GPUs and can reduce power consumption per chip compared to FPGAs. As a result, even using 36 chips, the core part consumed as low as 1.3W.

Lastly, this work is compared with the latest annealing processors in CMOS in Table 5. The proposed system has the largest number of fully-coupled spins implemented on the same board.

TABLE 5. Comparison of this work and the latest annealing processors in CMOS.

	This work		2022 [4]	2022 [16]	2023[13]	2023[9]	2021[12]
Technology	22nm CMOS		28nm CMOS	FPGA	40nm CMOS	FPGA	40nm CMOS
Annealing Method	Pseudo Annealing				Metamorphic Annealing	Dynamic Simulated Annealing	Metropolis SA
Topology	Fully coupled						Sparse (King's graph)
Number of Spins*	512×8 (8 Threads)	4096×8 (8 Threads)	512×8 (8 Threads)	384×4 (4 Threads)	2K	4k/16 SPUs	1.3M (16k/chip)
Precision [bit]	4	4	4	6	8	24	5
Op. Freq. [MHz]	10	10	1	10	336	200	100
Op. Power	35.2mW	2.90W (Proposed system)	12mW	3W	151.6-479mW @1.1V, 320MHz 28.97-95.13mW @0.8V, 120MHz	4.7	N/A

*Note: The number of spins in fully-coupled is equivalent to the square of the number of spins in nearest neighbor coupled.
That is 1k spins @ fully-coupled ≒ 1M spins @ nearest neighbor coupled

VI. CONCLUSION

We designed a 22-nm CMOS LSI chip (512-spin annealing processor) and implemented a 4096-spin scalable annealing processing system with 36 of them and 1 control FPGA on 1 board. We evaluated the operation of the proposed system by solving the Max Cut and Vertex Cover problems.

In the proposed system, reduction was applied from the symmetry of the interaction matrix, resulting in a reduction in the number of chips from 64 to 36. In addition, the proposed system can search for eight solutions in parallel by using spin-threads. The results show that the proposed system, while searching for 8 parallel solutions, improves computation speed by a factor of 34.0 and power performance by a factor of 2344 compared with the CPU.

In the future, the system should be scaled up by using multiple boards to solve larger and more complex combinatorial optimization problems.

ACKNOWLEDGMENT

The authors would like to thank Cui Dong and Shinjiro Kitahara for their help in chip layout design.

REFERENCES

[1] A. F. Iskandar, A. F. Sani, R. Riyadi, S. Febriani, and N. R. Syambas, "Fast heuristic algorithm optimization for travelling salesman problem," in *Proc. 7th Int. Conf. Wireless Telematics (ICWT)*, Bandung, Indonesia, Aug. 2021, pp. 1–6, doi: 10.1109/ICWT52862.2021.9678454.

[2] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J. Przybysz, and J. Whittaker, "Architectural considerations in the design of a superconducting quantum annealing processor," *IEEE Trans. Appl. Supercond.*, vol. 24, no. 4, pp. 1–10, Aug. 2014.

[3] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. Mcmahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-I. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, "A coherent Ising machine for 2000-node optimization problems," *Science*, vol. 354, no. 6312, pp. 603–606, Nov. 2016.

[4] R. Iimura, S. Kitamura, and T. Kawahara, "Annealing processing architecture of 28-nm CMOS chip for Ising model with 512 fully connected spins," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 12, pp. 5061–5071, Dec. 2021, doi: 10.1109/TCSI.2021.3114422.

[5] K. Yamamoto and T. Kawahara, "Scalable fully coupled annealing processing system and multi-chip FPGA implementation," *Microprocess. Microsyst.*, vol. 95, Nov. 2022, Art. no. 104674, doi: 10.1016/j.micpro.2022.104674.

[6] I. Ahmed, P.-W. Chiu, W. Moy, and C. H. Kim, "A probabilistic compute fabric based on coupled ring oscillators for solving combinatorial optimization problems," *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2870–2880, Sep. 2021.

[7] A. Mallick, Z. Zhao, M. K. Bashar, S. Alam, M. M. Islam, Y. Xiao, Y. Xu, A. Aziz, V. Narayanan, K. Ni, and N. Shukla, "CMOS-compatible Ising machines built using bistable latches coupled through ferroelectric transistor arrays," *Sci. Rep.*, vol. 13, no. 1, p. 1515, Jan. 2023, doi: 10.1038/S41598-023-28217-8.

[8] K. Tatsumura, A. R. Dixon, and H. Goto, "FPGA-based simulated bifurcation machine," in *Proc. 29th Int. Conf. Field Program. Log. Appl. (FPL)*, Barcelona, Spain, Sep. 2019, pp. 59–66, doi: 10.1109/FPL.2019.00019.

[9] D. Jiang, X. Wang, Z. Huang, Y. Yang, and E. Yao, "A network-on-chip-based annealing processing architecture for large-scale fully connected Ising model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 7, pp. 2868–2880, Jul. 2023, doi: 10.1109/TCSI.2023.3263923.

[10] Y. Su, J. Mu, H. Kim, and B. Kim, "A scalable CMOS Ising computer featuring sparse and reconfigurable spin interconnects for solving combinatorial optimization problems," *IEEE J. Solid-State Circuits*, vol. 57, no. 3, pp. 858–868, Mar. 2022, doi: 10.1109/JSSC.2022.3142896.

[11] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, "A 2 × 30 k-spin multi-chip scalable CMOS annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 145–156, Jan. 2020, doi: 10.1109/JSSC.2019.2949230.

[12] K. Yamamoto, T. Takemoto, C. Yoshimura, M. Mashimo, and M. Yamaoka, "A 1.3-Mbit annealing system composed of fully-synchronized 9-board × 9-chip × 16-kbit annealing processor chips for large-scale combinatorial optimization problems," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2021, pp. 1–3, doi: 10.1109/A-SSCC53895.2021.9634769.

[13] K. Kawamura, J. Yu, D. Okonogi, S. Jimbo, G. Inoue, A. Hyodo, Á. L. García-Anas, K. Ando, B. H. Fukushima-Kimura, R. Yasudo, T. Van Chu, and M. Motomura, "Amorphica: 4-replica 512 fully connected spin 336 MHz metamorphic annealer with programmable optimization strategy and compressed-spin-transfer multi-chip extension," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 42–44 doi: 10.1109/ISSCC42615.2023.10067504.

[14] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[15] S. Kitahara, A. Endo, T. Megumi, and T. Kawahara, "Method of halved interaction elements with regularity arrangement that achieves independent double systems for scalable fully coupled annealing processing," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Taipei, Taiwan, Nov. 2022, pp. 1–3, doi: 10.1109/A-SSCC56115.2022.9980631.

[16] K. Yamamoto and T. Kawahara, "Implemented parallel annealing in scalable fully coupled annealing processing system," in *Proc. IEEE 20th Jubilee World Symp. Appl. Mach. Intell. Informat. (SAMI)*, Poprad, Slovakia, Mar. 2022, pp. 000225–000230, doi: 10.1109/SAMI54271.2022.9780844.

TAICHI MEGUMI received the B.E. degree in electrical engineering from the Tokyo University of Science, Tokyo, Japan, in 2022, where he is currently pursuing the M.E. degree in electrical engineering. His research interest includes information processing circuits.

AKARI ENDO received the B.E. degree in electrical engineering from the Tokyo University of Science, Tokyo, Japan, in 2022, where she is currently pursuing the M.E. degree in electrical engineering. Her research interest includes information processing circuits.

TAKAYUKI KAWAHARA (Fellow, IEEE) received the B.S. and M.S. degrees in physics and the Ph.D. degree in electronics from Kyushu University, Fukuoka, Japan, in 1983, 1985, and 1993, respectively.

In 1985, he joined the Central Research Laboratory (CRL), Hitachi Ltd., where he made fundamental contributions in many areas in the field of low-power memories. In the field of DRAM circuits, his major contributions related to low-voltage circuits, including subthreshold-current reduction by gate-source self-reverse biasing, in 1993 and an over-drive sense-amplifier scheme coupled with direct sensing. He also pioneered the charge-recycling scheme, reported in 1993, which is currently widely applied in various circuits. In the field of flash memory, he and his team developed a 128-Mb chip, in 1996, with a bit-line clamped sensing scheme for fast sensing and a high-voltage generator scheme under a low-voltage supply. In addition, he led the Ultra-Low-Power System LSI Project with CRL. From 1997 to 1998, he was a Visiting Researcher with the Electronics Laboratory (LEG), Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland. In the field of emerging memory and devices, he reported the world's first fully functional 2-Mb STT-RAM proto chip, in 2007, and his team developed FD-SOI SRAM circuitry with back-gate control. After that, he was engaged in the development of circuitry for DNA sequencers (statistical nano-pore and fast ISFET arrays). Sustainable electronics is the focus of his laboratory, which includes low-power artificial intelligence (AI) devices and circuits, sensors and AI signal processing, spin current applications, and quantum computing techniques. In particular, the achievements of the fully-coupled Ising machine LSI have been highly regarded. In 2014, he became a Professor with the Department of Electrical Engineering, Tokyo University of Science, Tokyo, Japan.

Dr. Kawahara was a recipient of the 9th Yamazaki-Teiichi Prize, in 2009, the 2014 IEICE Electronics Society Award, and the Prize for Science and Technology (Development Category) in the FY2017 Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology of Japan. He was the Secretary/Publicity Officer of the 2006/2007 JFE Committee of Symposium on VLSI Circuits, an IEEE SSCS Distinguished Lecturer, in 2008 and 2009, and the FE Regional Chair of IEEE ISSCC 2009/2010.