

## **Superconductor amoeba-inspired problem solvers for combinatorial optimization**

Naoki Takeuchi<sup>1,2,\*</sup>, Masashi Aono<sup>3</sup>, and Nobuyuki Yoshikawa<sup>1,4</sup>

<sup>1</sup> Institute of Advanced Sciences, Yokohama National University, 79-5 Tokiwadai, Hodogaya, Yokohama 240-8501, Japan

<sup>2</sup> PRESTO, Japan Science and Technology Agency, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan

<sup>3</sup> Faculty of Environment and Information Studies, Keio University, 5322 Endo, Fujisawa, Kanagawa 252-0882, Japan

<sup>4</sup> Department of Electrical and Computer Engineering, Yokohama National University, 79-5 Tokiwadai, Hodogaya, Yokohama 240-8501, Japan

E-mail: [takeuchi-naoki-kx@ynu.jp](mailto:takeuchi-naoki-kx@ynu.jp)

Abstract: Adiabatic quantum-flux-parametron (AQFP) logic is an energy-efficient superconductor logic family; the energy dissipation of an AQFP gate can be arbitrarily reduced through adiabatic switching. In addition to high energy efficiency, AQFP logic has the advantage that it can easily introduce stochastic processes by exploiting naturally occurring thermal fluctuations. In this paper, we propose to use AQFP logic to implement an amoeba-inspired problem solver (APS), which is a stochastic local search method to explore solutions to combinatorial optimization problems such as the Boolean satisfiability problem (SAT). We designed a superconductor amoeba-inspired problem solver (SAPS) using AQFP logic, which finds solutions to a simple logical constraint satisfaction problem in the manner of APS, and fabricated it using a Nb integrated circuit fabrication process. Experimental results showed that the probability distribution of the stochastic processes in AQFP logic can be controlled by the

magnitude of bias current and that SAPS finds solutions using a small number of iterations when a moderate bias current is applied. The present results indicate the possibility of using AQFP logic to build hardware dedicated to the implementation of stochastic local search algorithms to solve combinatorial optimization problems such as SAT.

## **I. Introduction**

Superconductor logic can perform logic operations in an energy-efficient manner by taking advantage of its physical features: zero dc resistance, magnetic flux quanta, and the Josephson effect. Over the last several decades, many types of superconductor logic families have been proposed and demonstrated [1,2]. Rapid single flux quantum (RSFQ) logic [3] is one of the most developed logic families; RSFQ microprocessor prototypes have been designed and demonstrated by Tanaka *et al.* [4,5] Moreover, the extensive study of RSFQ logic has contributed to the invention of very energy-efficient logic families, such as energy-efficient RSFQ (ERSFQ) logic [6], reciprocal quantum logic (RQL) [7], and low-voltage RSFQ (LV-RSFQ) logic [8]. The Cryogenic Computing Complexity (C3) project [9] funded by IARPA has recently been developing very low-power microprocessors using ERSFQ logic and RQL while developing submicron fabrication technology [10].

Adiabatic quantum-flux-parametron (AQFP) logic [11], which is one of the energy-efficient superconductor logic families, is adiabatic logic based on the quantum-flux-parametron (QFP) [12]. The switching energy (energy dissipation per switching event) of a single AQFP gate can be arbitrarily reduced [13] through adiabatic switching [14,15], in which the potential energy profile evolves from a single-well shape to a double-well shape such that the logic state can change quasi-statically. Followed by the establishment of a design environment for AQFP logic [16,17], we have demonstrated complex AQFP logic circuits such as 8-bit carry look-ahead adders

[16,18] and register files [19] to develop low-power microprocessors. We have also conducted research on information thermodynamics [20], which studies information processing from a thermodynamics perspective, by the calculation of the heat emitted from and absorbed by AQFP logic. In previous work [21], we demonstrated Landauer's principle [22] numerically by showing that the probability distribution of logic states (logical entropy  $H$ ) in AQFP gates is associated with thermodynamic entropy  $S$  and that the change in logical entropy  $\Delta H$  is accompanied by heat absorption  $Q$  such that  $\Delta H = \beta Q$  in the quasi-static limit, where  $\beta$  is the inverse temperature.

The previous demonstration revealed that stochastic processes can be introduced into AQFP gates through heat absorption, which paves the way for the use of AQFP logic to implement stochastic local search algorithms for combinatorial optimization problems. Simulated (or quantum) annealing [23,24] is one of the most well-known local search algorithms; several types of hardware that implement simulated or quantum annealing have recently been proposed and are attracting much attention [25,26]. The Ising model can be used to apply annealing to many combinatorial optimization problems that find minima of objective functions [27]; however, this requires the careful scheduling of temperature, or the magnitude of fluctuation, because the variables can stabilize even at local minimum states, which do not satisfy all constraints. On the other hand, the amoeba-inspired problem solver (APS) [28,29] is a stochastic local search algorithm that is dedicated to solving the Boolean satisfiability problem (SAT) but does not require the scheduling of fluctuation. APS was formulated by Aono *et al.* and is inspired by the complex spatiotemporal dynamics of a single-celled amoeba of the true slime mold *Physarum polycephalum* [28], which deforms into the optimal shape to maximize favorable nutrient absorption and minimize the risk of being exposed to aversive light stimuli. SAT is a problem to determine if all the given logical constraints (Boolean formula) can be satisfied and is classified as a nondeterministic polynomial time (NP)-complete problem, which implies that all NP

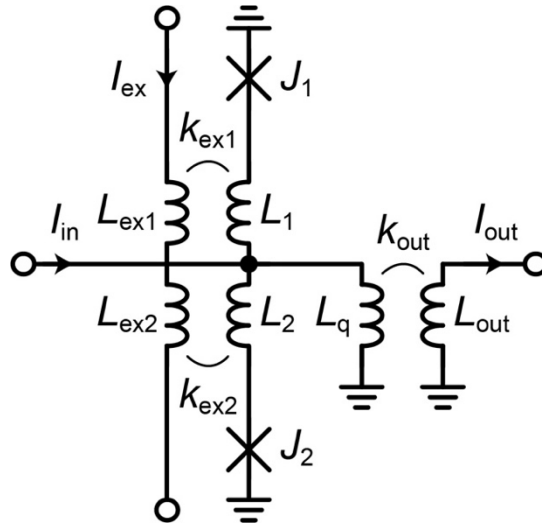
problems, including many practical real-world problems, can be reduced to SAT [30]. In general, APS can find solutions to SAT with a fewer number of iteration steps than conventional algorithms because APS updates multiple variables in parallel during each iteration whereas conventional algorithms change a single variable during each iteration [29]. Therefore, it is meaningful to develop hardware that can find solutions to SAT quickly by implementing APS. The key challenge to achieve hardware for the implementation of APS is to introduce stochastic processes into logic devices, but this is achievable in AQFP logic.

Here, as a proof-of-concept, we demonstrate a superconductor amoeba-inspired problem solver (SAPS) that can find solutions to a simple logical constraint satisfaction problem in the manner of APS. SAPS is composed of basic AQFP logic gates such as buffers and NOR gates. In SAPS, stochastic processes are introduced to AQFP gates via thermal fluctuation, where the amount of heat absorption, or the magnitude of entropy change, is controlled by the bias current applied to the AQFP gates. Followed by a basic explanation of AQFP logic and APS, we show the detail of SAPS and demonstrate it at 4.2 K in liquid He. The measurement results indicate that SAPS can find solutions through stochastic processes and that the solutions are quickly found when moderate bias current is applied.

## II. Adiabatic quantum-flux-parametron (AQFP) logic

Figure 1(a) shows a schematic of an AQFP gate. When the excitation current  $I_{\text{ex}}$  ramps up and a magnetic flux  $\Phi_{\text{ex}}$  is applied to the gate, either of the two Josephson junctions  $J_1$  and  $J_2$  switches, depending on the polarity of the input current  $I_{\text{in}}$ . As a result of the junction switching, a flux quantum  $\Phi_0 \approx 2.07 \times 10^{-15}$  Wb is stored in the superconductor loop composed of  $L_q$ ,  $L_1$ , and  $J_1$  ( $L_q$ ,  $L_2$ , and  $J_2$ ) for positive  $I_{\text{in}}$  (negative  $I_{\text{in}}$ ). The output current  $I_{\text{out}}$  is generated through the mutual inductance  $M_{\text{out}} = k_{\text{out}}(L_q L_{\text{out}})^{0.5}$ . The direction of  $I_{\text{out}}$  shows the logic state of the gate:

positive  $I_{\text{out}}$  (negative  $I_{\text{out}}$ ) represents a logic 1 (a logic 0). Figure 1(b) shows the time evolution of the potential energy of the AQFP gate for a positive  $I_{\text{in}}$  while switching, where  $\Phi_{\text{ex}} = M_{\text{ex}}I_{\text{ex}}$  and  $M_{\text{ex}} = k_{\text{ex}1}(L_{\text{q}}L_1)^{0.5} + k_{\text{ex}2}(L_{\text{q}}L_2)^{0.5}$ . AQFP gates are generally symmetrical; therefore, we assume that  $L_1 = L_2$  and  $k_{\text{ex}1} = k_{\text{ex}2}$ . The figure shows that the profile of the potential energy evolves from a single-well shape into a double-well shape as  $\Phi_{\text{ex}}$  increases. Positive  $I_{\text{in}}$  tilts the potential energy toward a logic 1, so that the state of the gate (the blue particle in the figure) switches gradually to a logic 1. This switching process is almost deterministic if the magnitude of  $I_{\text{in}}$  is sufficiently large. On the other hand, if  $I_{\text{in}}$  is not large, then the state could switch to a logic 0 due to thermal fluctuation, as shown in Fig. 1(c), where the color strength of the particles represents the probability of being in each state. In this case, the AQFP gate absorbs heat  $Q$  from the thermal bath, and the logical entropy of the gate changes by  $\Delta H = \beta Q (> 0)$  [21]. In this way, stochastic processes can be introduced to AQFP gates through heat absorption. It is noteworthy that the magnitude of  $Q$ , or  $\Delta H$ , can be controlled by the magnitude of  $I_{\text{in}}$  because it determines the shape of the potential energy during a switching process.



(a)

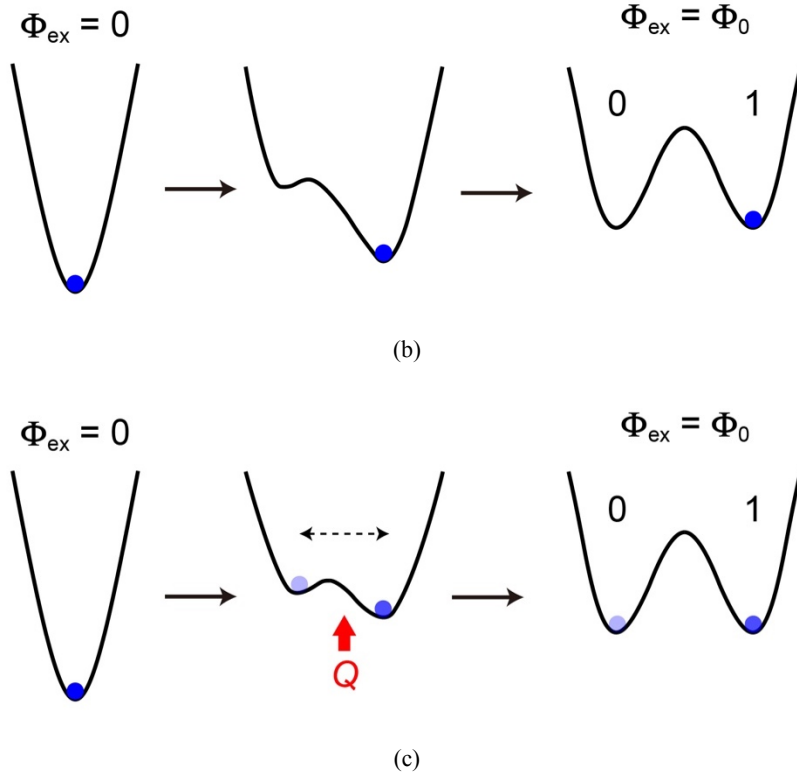


Fig. 1. (a) Schematic of the adiabatic quantum-flux-parametron gate. When the excitation current  $I_{\text{ex}}$  is applied, either of the two Josephson junctions  $J_1$  and  $J_2$  switches, depending on the polarity of the input current  $I_{\text{in}}$ , and a flux quantum is stored. (b) Time evolution of the potential energy with large  $I_{\text{in}}$  during a switching process. The potential energy is significantly tilted, so that the switching process is almost deterministic. (c) Time evolution of the potential energy with small  $I_{\text{in}}$  during a switching process. In this case, the AQFP gate can switch stochastically to both logic states 0 and 1 because of thermal fluctuation; the AQFP gate absorbs heat  $Q$  and increases entropy, thereby changing the probability distribution inside the potential well.

### III. Amoeba-inspired problem solver (APS)

There are multiple versions of APS; however, an iteration of APS generally includes the following three procedures. (1) The logic states of all the variables are observed. (2) The variables are simultaneously updated such that the variables that satisfy given constraints hold their values, and those that do not satisfy constraints are flipped; this is the reason why APS does not require the

scheduling of fluctuation. (3) The update of the variables fails stochastically (i.e., the variables are stochastically flipped, regardless of given constraints), which is required to avoid deadlocked states, where variables keep evolving but never reach a solution [31]. The variables eventually stop changing as procedures 1 through 3 are iterated, which ensures that the states of the variables correspond to a solution, that satisfies all the constraints [29]. Here, we treat a simple logical constraint satisfaction problem that we call the NOR problem [31]: find a vector  $\mathbf{x} = (x_1, x_2, x_3, \dots, x_N)$  such that the variables satisfy  $x_i = \text{NOR}(x_{i-1}, x_{i+1})$ , where  $x_i \in \{0, 1\}$  and  $i \in \{1, 2, 3, \dots, N\}$ . In APS, the solutions to the NOR problem can be found by updating the variables in accordance with the following equations:

$$X_i(t+1) = \overline{x_{i-1}(t) + x_{i+1}(t)}, \quad (1)$$

$$x_i(t+1) = \begin{cases} \overline{X_i(t+1)} & \text{with a probability } p_1 \text{ if } x_i(t) = 1 \text{ and } X_i(t+1) = 0 \\ \overline{X_i(t+1)} & \text{with a probability } p_2 \text{ if } x_i(t) = 0 \text{ and } X_i(t+1) = 0 \\ X_i(t+1) & \text{otherwise,} \end{cases} \quad (2)$$

where  $x_i(t)$  shows the state of  $x_i$  at the current iteration  $t$ ,  $X_i(t+1)$  shows the intermediate state for updating  $x_i$ , and  $x_i(t+1)$  shows the state of  $x_i$  at the next iteration  $t+1$ . Equation 1 corresponds to procedure 2 in APS:  $x_i(t)$  holds its value when it satisfies the given constraint and is flipped otherwise. For instance,  $X_i(t+1) = 1$  if  $x_{i-1}(t) = 0$ ,  $x_i(t) = 1$ , and  $x_{i+1}(t) = 0$ ;  $X_i(t+1) = 0$  if  $x_{i-1}(t) = 1$ ,  $x_i(t) = 1$ , and  $x_{i+1}(t) = 0$ ; and  $X_i(t+1) = 1$  if  $x_{i-1}(t) = 0$ ,  $x_i(t) = 0$ , and  $x_{i+1}(t) = 0$ . Equation 2 corresponds to procedure 3 in APS: the change from 1 to 0 fails with a probability  $p_1$ , and the conservation of 0 fails with a probability  $p_2$ . Figures 2(a) and (b) show some examples of the time evolution of the variables for  $N = 6$ , where the initial state is  $\mathbf{x} = (0, 0, 0, 0, 0, 0)$ . Figure 2(a) shows the case for  $p_1 = 0$  and  $p_2 = 0$ . Stochastic processes are not introduced, so that the variables alternate between the two states  $\mathbf{x} = (0, 0, 0, 0, 0, 0)$  and  $\mathbf{x} = (1, 1, 1, 1, 1, 1)$ , i.e., the variables are deadlocked and cannot reach a solution. Figure 2(b) shows the case for  $p_1 > 0$  and  $p_2 > 0$ . Unlike the case of Fig. 2(a), the change from 1 to 0 and the conservation of 0 fail stochastically. These

stochastic processes are highlighted in red. At iterations 2 and 4, the change from 1 to 0 fails, which breaks the deadlocked state and leads to a satisfied state, or a solution  $\mathbf{x} = (1, 0, 1, 0, 1, 0)$ , at iteration 5. The solution satisfies all the constraints; for instance,  $x_3$  satisfies  $x_3 = \text{NOR}(x_2, x_4)$ . The satisfied state continues for a while because Eq. 1 does not change the variables. However, the conservation of 0 fails stochastically in accordance with Eq. 2, so that the satisfied state is eventually broken, thereby starting a search for another solution. In Fig. 2(b), the conservation of 0 fails at iteration 8, and another solution  $\mathbf{x} = (0, 1, 0, 0, 1, 0)$  is found at iteration 9. In this way, APS can find multiple solutions during a time evolution, which is important in some applications such as the simulation of chemical reactions [32]. Figure 2(c) shows the simulated performance of APS solving NOR problems: the average iteration number to find a solution over 500 trials as a function of the problem size  $N$ , where the initial state is  $\mathbf{x} = (0, 0, 0, \dots, 0)$ , and the probabilities of stochastic processes are  $p_1 = 0.2$  and  $p_2 = 1/(2N)$ . Note that  $p_1$  and  $p_2$  are kept constant during a solution search because APS does not require the scheduling of fluctuation.  $p_2$  works only to break satisfied states and to find multiple solutions; therefore,  $p_2$  is set to a small value  $1/(2N)$  in this simulation so that it does not affect the search speed significantly. The fitting curve in Fig. 2(c) is  $5.47\ln N + 2.88$ , which indicates that APS can find solutions quickly using simple procedures. Although the procedures shown in Eqs. 1 and 2 are applicable only to the NOR problem, APS has undergone several modifications to solve general SAT, the details of which are given in the literature [28,29,31].



Iteration	0	1	2	3	4	5	6	7	8	9	10
$x_1$	0	1	0	1	0	1	0	1	0	1	0
$x_2$	0	1	0	1	0	1	0	1	0	1	0
$x_3$	0	1	0	1	0	1	0	1	0	1	0
$x_4$	0	1	0	1	0	1	0	1	0	1	0
$x_5$	0	1	0	1	0	1	0	1	0	1	0
$x_6$	0	1	0	1	0	1	0	1	0	1	0

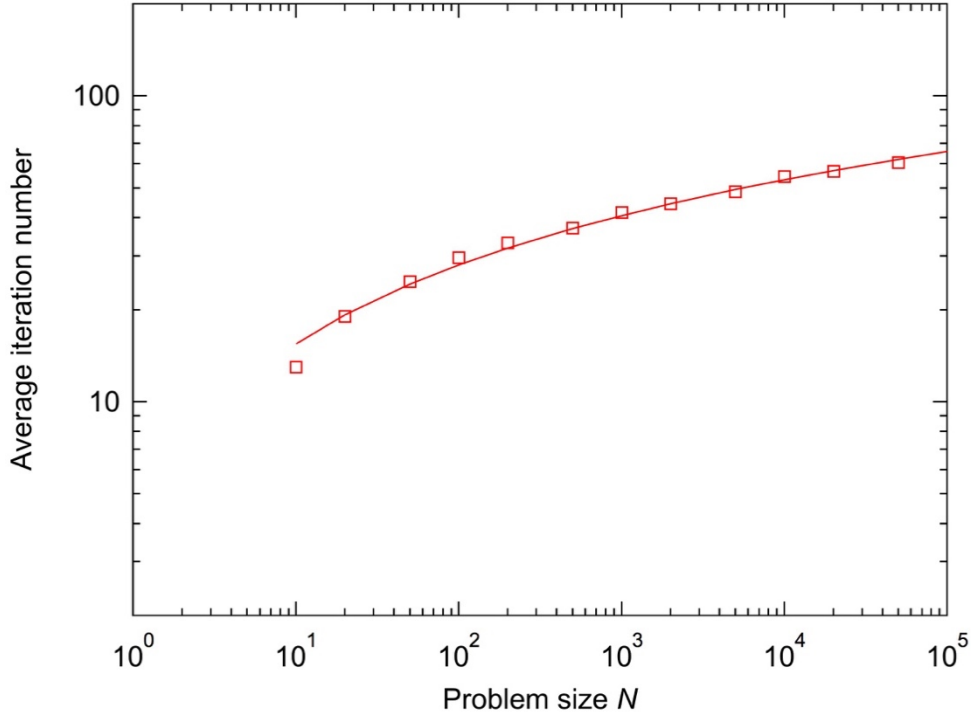
Deadlocked

(a)

Iteration	0	1	2	3	4	5	6	7	8	9	10
$x_1$	0	1	0	1	1	1	1	1	1	0	0
$x_2$	0	1	0	1	0	0	0	0	1	1	1
$x_3$	0	1	0	1	0	1	1	1	1	0	0
$x_4$	0	1	0	0	0	0	0	0	0	0	0
$x_5$	0	1	1	1	1	1	1	1	1	1	1
$x_6$	0	1	0	0	0	0	0	0	0	0	0

Satisfied      Satisfied

(b)



(c)

Fig. 2. Solving NOR problems using APS. (a) Time evolution for  $N = 6$  without stochastic processes. Every process is deterministic; therefore, the variables cannot find a solution, or become deadlocked. (b) Time evolution for  $N = 6$  with stochastic processes. The variables can break the deadlocked state because of stochastic processes, which leads to multiple solutions. (c) Simulated performance of APS solving NOR problems for  $p_1 = 0.2$  and  $p_2 = 1/(2N)$ . The average iteration number to find a solution was calculated over 500 trials for each problem size  $N$ . The fitting curve is  $5.47\ln N + 2.88$ .

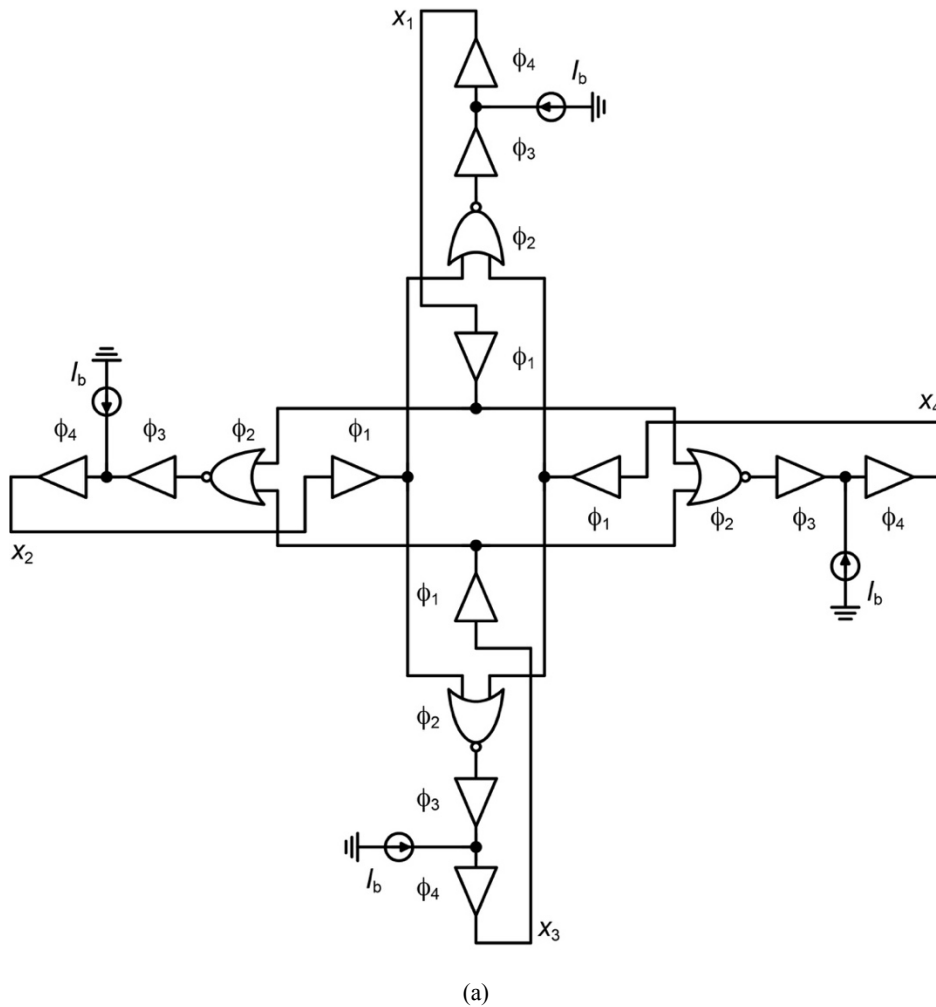
#### IV. Superconductor amoeba-inspired problem solver (SAPS)

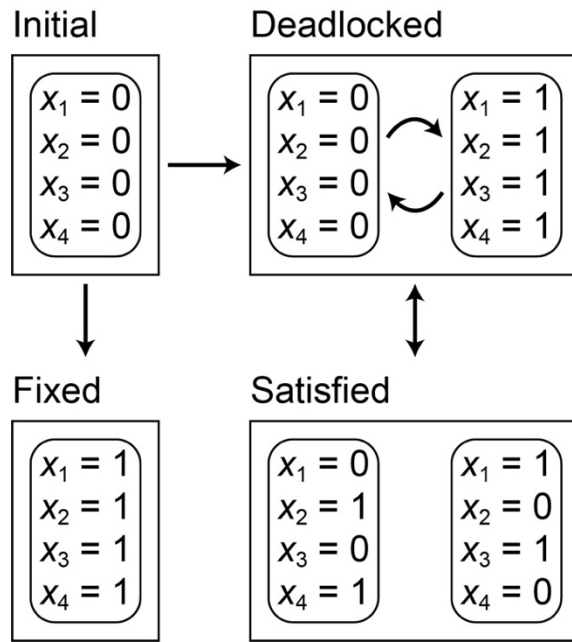
We design SAPS using AQFP logic that can solve a small-scale NOR problem ( $N = 4$ ). Figure 3(a) shows a schematic of SAPS, which is based on the amoeba-inspired semiconductor circuit demonstrated by Kasai *et al.* [33] The AQFP logic gates in Fig. 3(a) are based on the buffer shown in Fig. 1(a) and are driven by a four-phase excitation mode using a pair of ac excitation currents and a dc-offset current. The details of AQFP logic gate design and excitation methods are given

in the literature [16,34]. Logic operations are performed along the excitation phases  $\phi_1$  through  $\phi_4$  with a phase separation of  $90^\circ$ . The logic states of the buffers in phase  $\phi_4$  represent the states of the variables  $x_1$  through  $x_4$ . SAPS implements the three procedures of APS as follows. (1) The buffers in phase  $\phi_1$  observe the variables  $x_1$  through  $x_4$ . (2) The NOR gates update all the variables simultaneously in accordance with Eq. 1,  $X_i = \text{NOR}(x_{i-1}, x_{i+1})$ . (3) Stochastic processes are introduced by the bias current  $I_b (> 0)$ , which applies an offset input current to the buffers in phase  $\phi_4$ . Both  $I_b$  and the signal current that represents a logic 1 (positive  $I_{\text{out}}$  in Fig. 1(a)) have the same polarity; therefore, logic 1s propagate deterministically from the buffers in phase  $\phi_3$  to those in phase  $\phi_4$ . On the other hand, logic 0s (negative  $I_{\text{out}}$  in Fig. 1(a)) propagate stochastically from the buffers in phase  $\phi_3$  to those in phase  $\phi_4$  because  $I_b$  reduces the magnitude of the signal current that represents logic 0s between phases  $\phi_3$  and  $\phi_4$ . The buffers in phase  $\phi_4$  absorb heat and increase logical entropy, as shown in Fig. 1(c); therefore,  $x_i$  is updated in accordance with Eq. 2. For simplicity,  $p_1 = p_2$  in this circuit configuration because both  $p_1$  and  $p_2$  are determined by the amplitude of  $I_b$ . In this way, the variables  $x_1$  through  $x_4$  evolve in the manner of APS. SAPS is scalable because the problem size can be increased by repeating the same circuit structure, as shown in Fig. 3(d), which illustrates SAPS for  $N = 8$ . While the previously reported amoeba-inspired circuit [33] used circuit parameter variations to avoid deadlocked states, SAPS utilizes thermal fluctuation to avoid deadlocked states, which enables multiple solutions independent of initial states to be found, as will be shown later. SAPS includes 12 Josephson junctions per variable, excluding the peripheral circuits for readout. Assuming the average energy dissipation is 5 zJ per junction [16], the power consumption of SAPS for  $N$  variables is only  $0.3N$  (nW) for a 5 GHz operation.

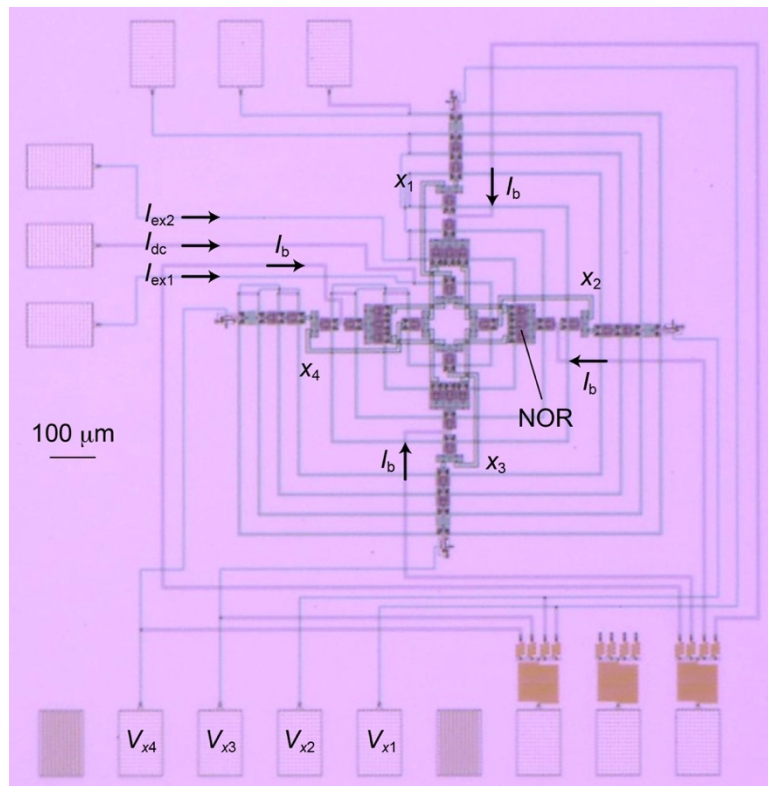
Figure 3(b) shows the possible states of SAPS. In the experiment, SAPS starts searching for solutions from the initial state  $\mathbf{x} = (0, 0, 0, 0)$ . SAPS undergoes state transition from the initial

state to the deadlocked state, where the variables alternate between the two combinations  $\mathbf{x} = (0, 0, 0, 0)$  and  $\mathbf{x} = (1, 1, 1, 1)$  due to the logical constraint  $x_i = \text{NOR}(x_{i-1}, x_{i+1})$  represented by the NOR gates. If the update of a variable fails due to thermal fluctuation, then the variables can break the deadlocked state and switch to one of the satisfied states  $\mathbf{x} = (0, 1, 0, 1)$  and  $\mathbf{x} = (1, 0, 1, 0)$ , which are the solutions to the NOR problem. The given logical constraints are satisfied, so that the variables keep the same logical values for a while after reaching a satisfied state. The variables can return to the deadlocked state again due to thermal fluctuation, and therefore multiple solutions can appear without initialization of the variables, which is one of the features in APS [31].





(b)



(c)

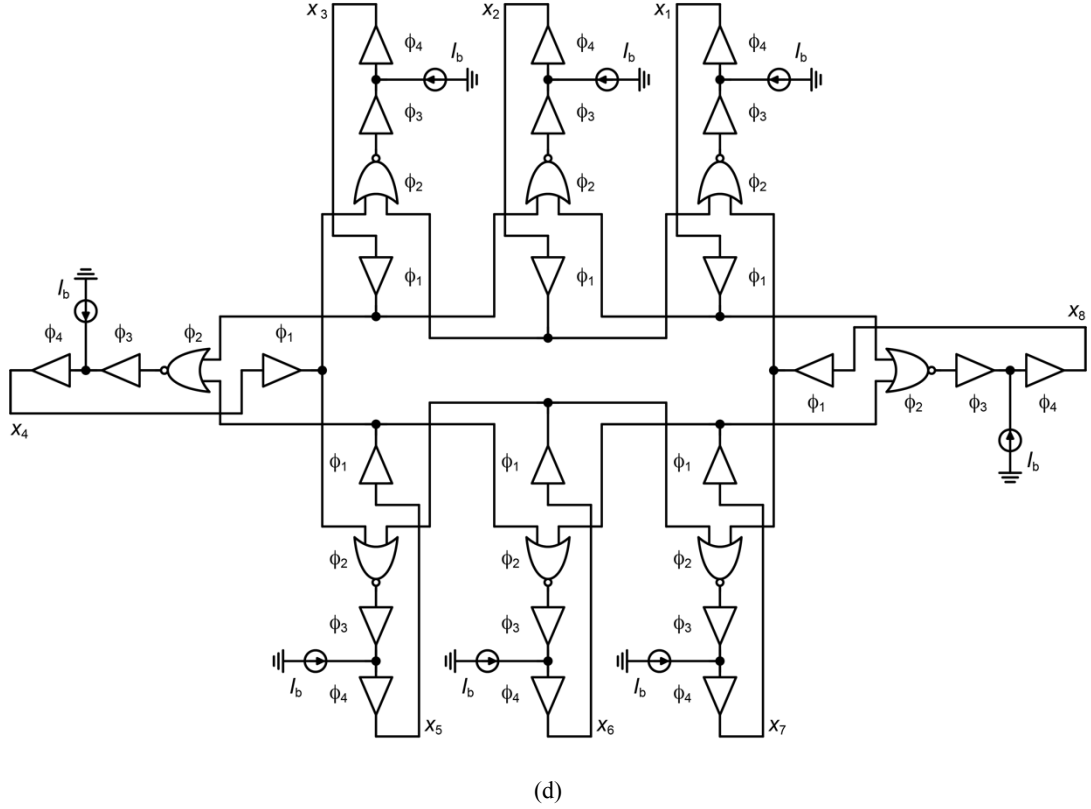


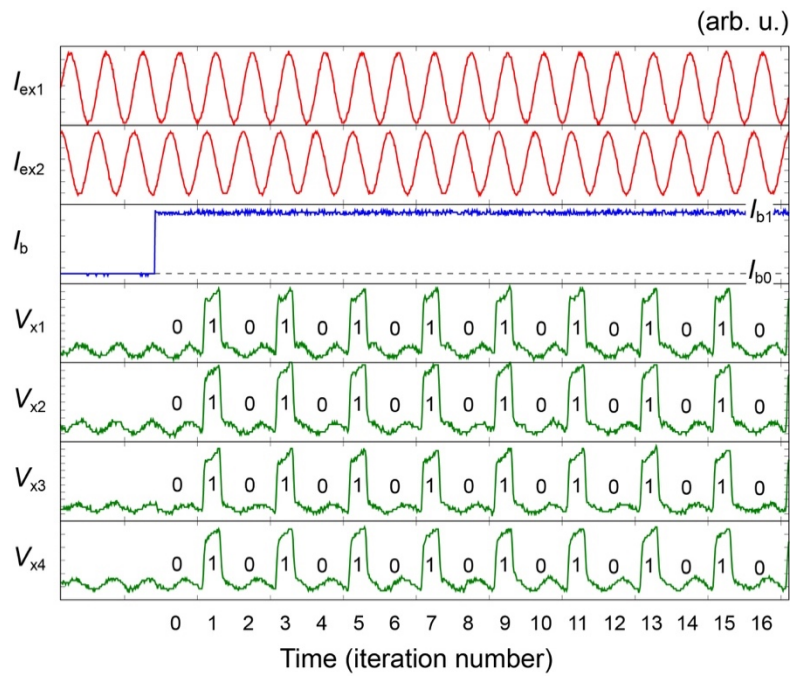
Fig. 3. (a) Schematic of a superconductor amoeba-inspired problem solver (SAPS) that can solve a NOR problem for  $N = 4$ . The buffers in  $\phi_1$  observe the variables  $x_1$  through  $x_4$ . The NOR gates in  $\phi_2$  update the variables in parallel. The bias current  $I_b$  induces stochastic processes, where logic 1s propagate deterministically, whereas logic 0s are stochastically flipped. (b) Possible states of the variables  $x_1$  through  $x_4$ . In the deadlocked state, the variables alternate between  $\mathbf{x} = (0, 0, 0, 0)$  and  $\mathbf{x} = (1, 1, 1, 1)$  and do not find solutions. With stochastic processes, the deadlocked state is stochastically broken, and the variables can switch to the satisfied states where the solutions are found. (c) Micrograph of SAPS, which was fabricated using the Nb integrated circuit fabrication process. (d) SAPS for  $N = 8$ .

## V. Stochastic solution search using SAPS

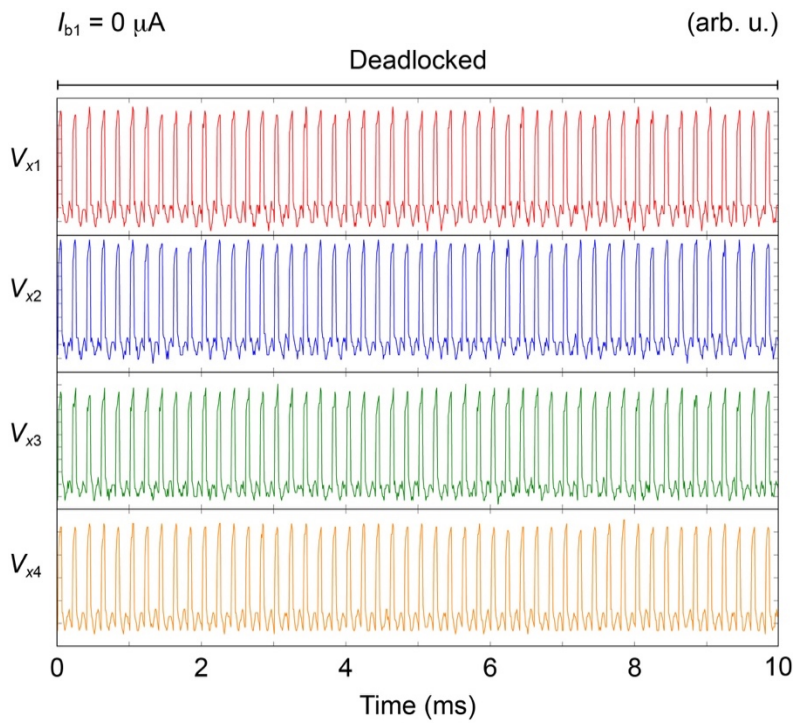
We fabricate SAPS and operate it to demonstrate that the solutions to the NOR problem are found through the time evolution shown in Fig. 3(b). Figure 3(c) shows a micrograph of SAPS, which was fabricated using the National Institute of Advanced Industrial Science and Technology (AIST) 2.5 kA/cm<sup>2</sup> Nb standard process (STP2) [35]. A pair of excitation currents  $I_{ex1}$  and  $I_{ex2}$  are

provided by a function generator, and a dc-offset current  $I_{dc}$  is provided by a voltage source. The bias current  $I_b$  is provided to each buffer in phase  $\phi_4$  by a common voltage source. The logic states of the variables  $x_1$  through  $x_4$  are converted into voltage signals  $V_{x1}$  through  $V_{x4}$  using dc superconducting quantum interference devices (dc-SQUIDs) [11] that are magnetically coupled to the buffers in phase  $\phi_4$ . The SAPS chip was placed inside a liquid He dewar in the experiment. Figure 4(a) shows the measurement sequences.  $I_{ex1}$  and  $I_{ex2}$  are sinusoidal currents with a phase separation of  $90^\circ$  that power and clock the AQFP gates in SAPS. At the beginning,  $I_b$  is set to a negative value  $I_{b0}$  ( $= -27 \mu\text{A}$ ) to set the variables  $x_1$  through  $x_4$  to the initial state  $\mathbf{x} = (0, 0, 0, 0)$ . After the initialization,  $I_b$  increases to a positive value  $I_{b1}$  to introduce stochastic processes, and the time evolution of  $V_{x1}$  through  $V_{x4}$  ( $x_1$  through  $x_4$ ) is then observed. In synchronization with  $I_{ex2}$ ,  $x_1$  through  $x_4$  are iteratively updated, and their corresponding voltage signals  $V_{x1}$  through  $V_{x4}$  appear in the form of unipolar return-to-zero (RZ) encoding. Figures 4(b) through 4(d) show typical measurement waveforms for the three different values of  $I_{b1}$ , where the frequency of  $I_{ex1}$  and  $I_{ex2}$  is 100 kHz. Figure 4(b) shows the time evolution of  $V_{x1}$  through  $V_{x4}$  for  $I_{b1} = 0 \mu\text{A}$ . No positive  $I_b$  is applied; therefore, every logic operation in SAPS is deterministic, and thus the variables stay in the deadlocked state. Figure 4(c) shows waveforms for  $I_{b1} = 18.24 \mu\text{A}$ . SAPS escapes from the deadlocked state and switches to satisfied states because a positive  $I_b$  flips logic 0s stochastically between  $\phi_3$  and  $\phi_4$ . The variables keep the same logic values for a while after reaching a satisfied state, which ensures that the variables form a solution in APS [28]. Both the satisfied states  $\mathbf{x} = (0, 1, 0, 1)$  and  $\mathbf{x} = (1, 0, 1, 0)$  appear stochastically via the deadlocked state. These indicate that stochastic processes due to thermal fluctuation help SAPS break the deadlocked state and find solutions, as expected in Fig. 3(b), and that both solutions are found from the same initial state. Figure 4(d) shows waveforms for  $I_{b1} = 27.55 \mu\text{A}$ . In this case,  $I_{b1}$  is too large, so the variables are fixed to  $\mathbf{x} = (1, 1, 1, 1)$ , which corresponds to the fixed state in Fig.

3(b).

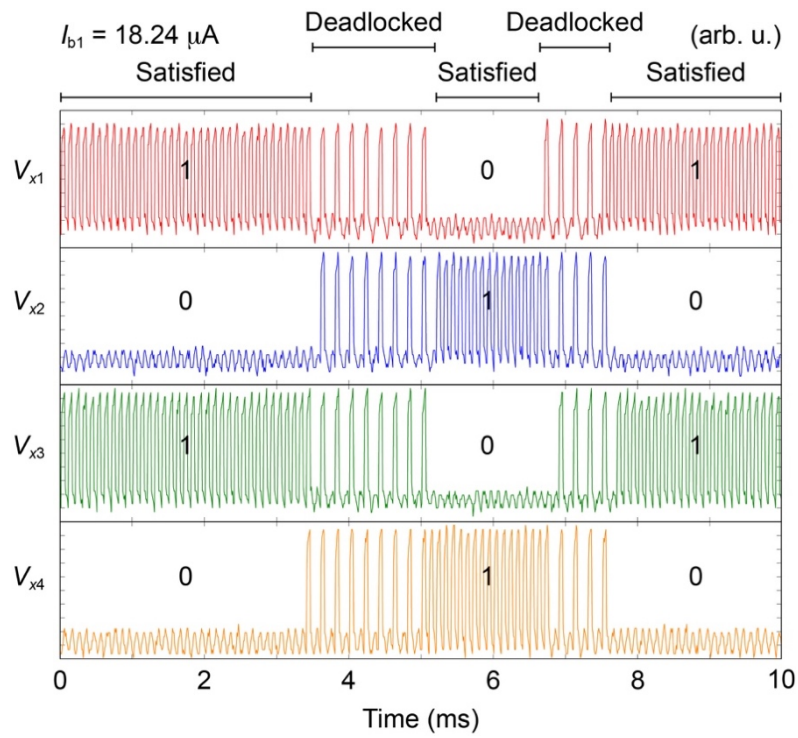


(a)

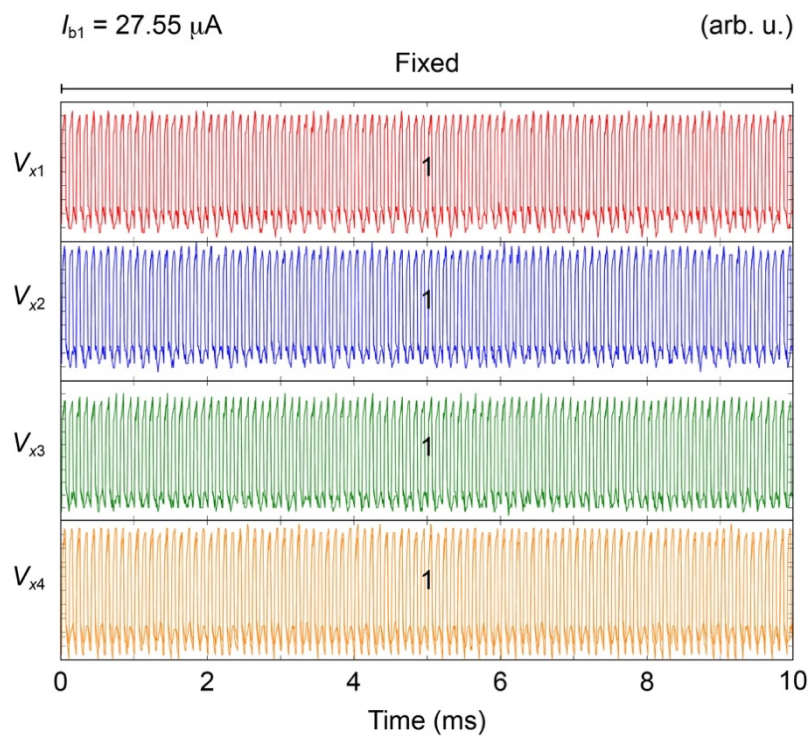


(b)





(c)



(d)

Fig. 4. (a) Sequence of the measurement waveforms of SAPS.  $I_{ex1}$  and  $I_{ex2}$  power and clock the AQFP gates in SAPS.  $I_b$  initially has a negative value  $I_{b0} = -27 \mu\text{A}$  to initialize the variables to  $(0, 0, 0, 0)$  and increases to a positive value  $I_{b1}$  to introduce stochastic processes. The logic states of the variables  $x_1$  through  $x_4$  are converted into the voltages signals  $V_{x1}$  through  $V_{x4}$  for readout. (b) Waveforms for  $I_{b1} = 0 \mu\text{A}$ . Every logic operation is deterministic, so that the variables stay in the deadlocked state. (c) Waveform for  $I_{b1} = 18.24 \mu\text{A}$ . The moderate bias current introduces stochastic processes, so that the variables can switch to the satisfied states stochastically. (c) Waveform for  $I_{b1} = 27.55 \mu\text{A}$ . The logic states of the variables are fixed to  $(1, 1, 1, 1)$  because the bias current is too large.

The measurement sequence shown in Fig. 4(a) was repeated 200 times for each  $I_{b1}$ , where one measurement sequence includes 1000 iteration steps, to evaluate how fast solutions are found depending on the value of  $I_{b1}$ . Figure 5(a) shows the measured probability of being in satisfied states  $P_{\text{sat}}$  as a function of the iteration number and  $I_{b1}$ .  $I_{b1}$  determines how much heat SAPS absorbs, or how often logic 0s are flipped between phases  $\phi_3$  and  $\phi_4$ ; therefore, the time evolution of  $P_{\text{sat}}$  is strongly dependent on  $I_{b1}$ . The figure shows that  $P_{\text{sat}}$  increases quickly for  $I_{b1}$  around  $18.40 \mu\text{A}$ , which indicates that there is an appropriate amount of heat absorption to find solutions. Figure 5(b) shows the time evolution for the probability distribution of the variables  $x_1$  through  $x_4$  for  $I_{b1} = 18.4 \mu\text{A}$ . While the probabilities of being in  $\mathbf{x} = (0, 0, 0, 0)$  and  $\mathbf{x} = (1, 1, 1, 1)$  drop quickly, those of being in the satisfied states  $\mathbf{x} = (0, 1, 0, 1)$  and  $\mathbf{x} = (1, 0, 1, 0)$  increase rapidly, which indicates that stochastic processes introduced by  $I_{b1}$  help SAPS find solutions quickly. Moreover, both the solutions appear with similar probabilities, although the probabilities of being in  $\mathbf{x} = (0, 1, 0, 1)$  and  $\mathbf{x} = (1, 0, 1, 0)$  are not exactly the same due to circuit parameter variation. In future work, we will demonstrate a large-scale SAPS to study the search speed more comprehensively because the size of the problem treated in this experiment is too small for such discussions.

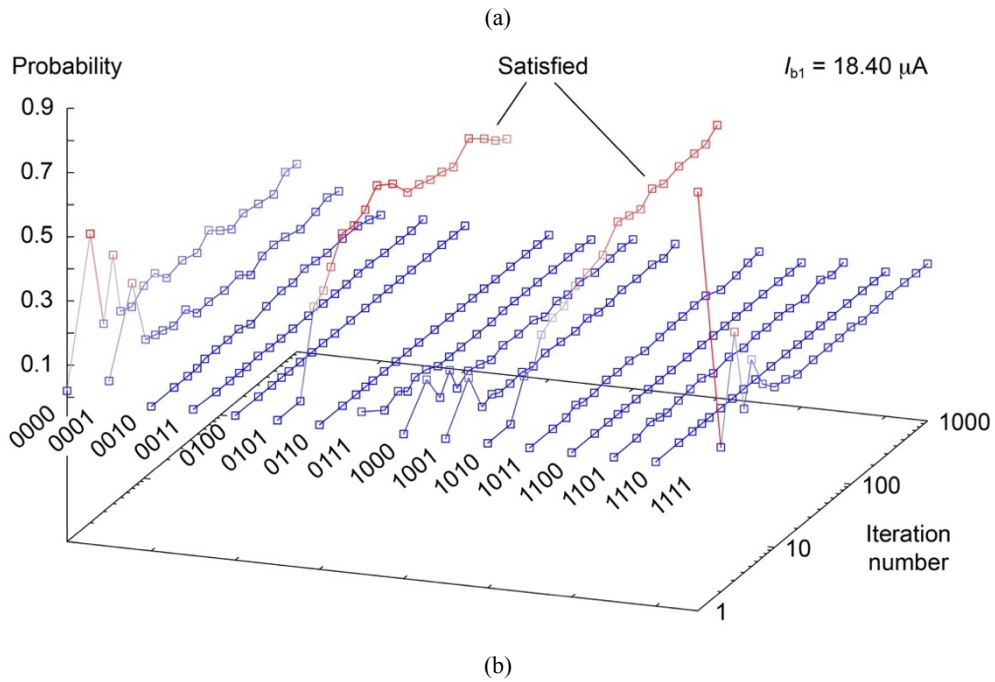
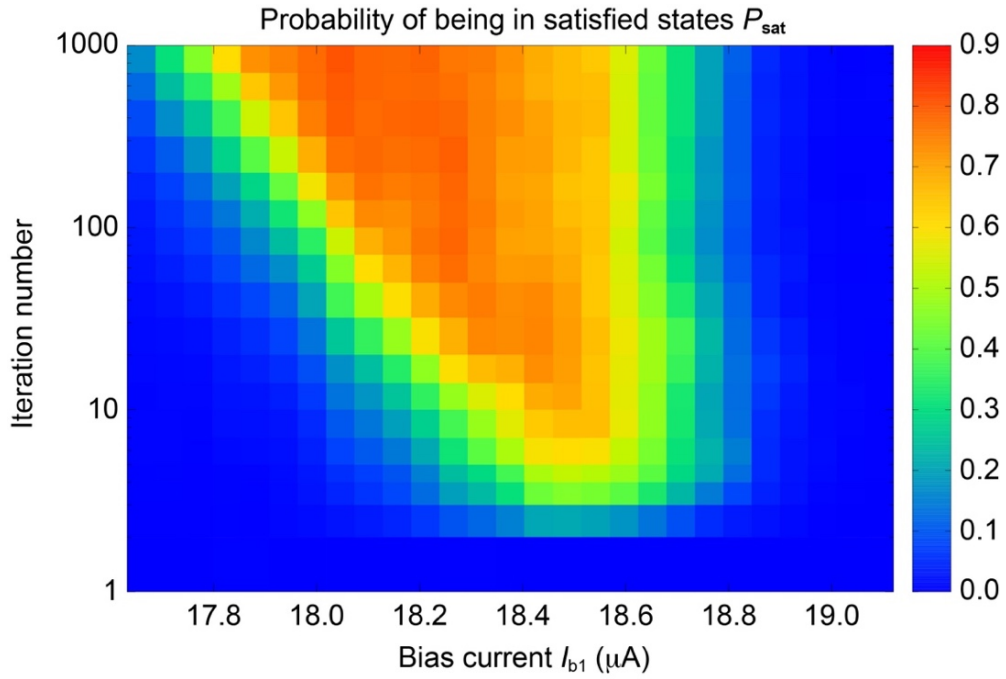


Fig. 5. Time evolution of the variables in the experiments. (a) Evolution of the probability of being in satisfied states  $P_{\text{sat}}$ . With small  $I_{b1}$ ,  $P_{\text{sat}}$  requires many iterations to achieve high  $P_{\text{sat}}$ . However, with  $I_{b1}$  of approximately 18.4 μA,  $P_{\text{sat}}$  reaches 0.5 with less than 10 iterations. (b) Evolution of the probability distribution of each state for  $I_{b1} = 18.40 \mu\text{A}$ . The probabilities of being in  $\mathbf{x} = (0, 0, 0, 0)$  and  $\mathbf{x} = (1, 1, 1, 1)$  drop quickly, and those of being in  $\mathbf{x} = (0, 1, 0, 1)$  and

$\mathbf{x} = (1, 0, 1, 0)$  increase rapidly. Both the solutions  $\mathbf{x} = (0, 1, 0, 1)$  and  $\mathbf{x} = (1, 0, 1, 0)$  appear with similar probabilities.

## VI. Conclusions

In this study, we proposed to use AQFP logic to implement APS by taking advantage of the ease of introducing stochastic processes into the circuit implementation. We designed SAPS using basic AQFP logic gates and fabricated it using a Nb integrated circuit fabrication process. SAPS was operated at 4.2 K, and it was demonstrated that it can find solutions to a simple logical constraint satisfaction problem through the use of stochastic processes induced by thermal fluctuation. The experimental results showed that the probability distribution of the logic states of AQFP gates can be controlled by the magnitude of the bias current and that SAPS can find solutions with a small number of iterations when a moderate bias current is applied. The estimated power consumption of SAPS was only 2.4 nW for a 5 GHz operation. These results reveal the possibility of very low-power superconductor circuits dedicated to the performance of stochastic solution-search algorithms.

For our next step, we will modify SAPS such that it can solve not only NOR problems but also general SAT. In a previous study [36], we proposed a circuit model called circuit-level AmoebaSAT (CL-AmbSAT), which can find solutions to general SAT in the manner of APS. This model is designed to search for solutions to 3-SAT [37], which is a NP-complete problem to determine the satisfiability of a formula in the conjunctive normal form (CNF) where each clause includes at most three literals. Arbitrary SAT instances can be reduced to 3-SAT instances, so that CL-AmbSAT can solve general SAT. In CL-AmbSAT, each variable  $x_i$  is represented by a small circuit unit called a variable cell, and the variable cells are interconnected in accordance with the given 3-SAT instance. Figure 6 illustrates a schematic of a variable cell representing  $x_i(t)$ , which is composed of basic logic gates, flip-flops (FFs), a majority (MAJ) gate, and stochastic gates

(SGs).  $l_{j,\alpha}$ ,  $l_{k,\beta}$  and others represent the literals that share clauses (logical constraints) in the given 3-SAT instance with  $x_i$ , where  $l_{j,\alpha} = x_j$  ( $l_{j,\alpha} = \neg x_j$ ) for  $\alpha = 0$  ( $\alpha = 1$ ). SGs introduce stochastic processes; for instance, SG1 always passes a logic 1 but flips a logic 0 with a probability  $p_1$ . Our numerical simulation showed that CL-AmbSAT can find solutions to 3-SAT in the manner of APS; more details of CL-AmbSAT and simulation results can be found in the literature [36]. Therefore, we will modify SAPS on the basis of CL-AmbSAT. Importantly, all the circuit components in CL-AmbSAT can be implemented by AQFP logic. SGs can be implemented by buffers with bias current that controls probability distribution, and other logic gates are available in our AQFP cell library [16,34].

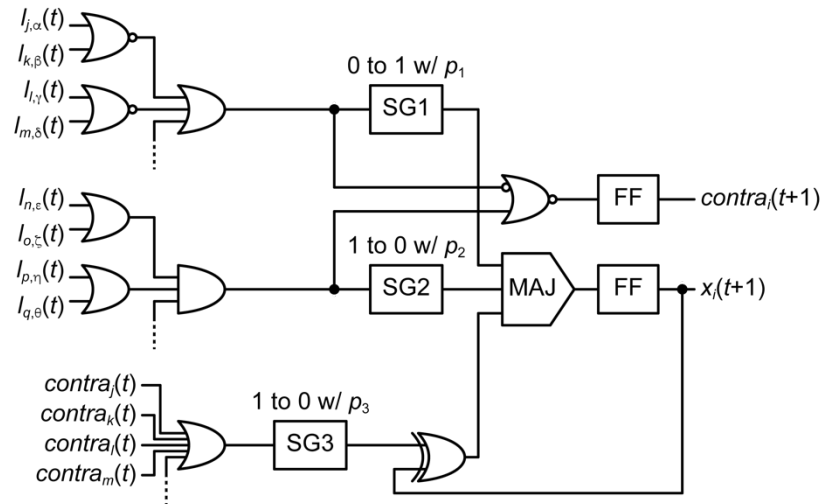


Fig. 6. Variable cell representing  $x_i$  in a circuit model called CL-AmbSAT. The variable cells are interconnected in accordance with the given 3-SAT instance. Stochastic processes are introduced by SGs.

### Acknowledgements

The present study was supported by PRESTO of the Japan Science and Technology Agency (JST) (Grant Nos. JPMJPR1528 and JPMJPR1321). The circuits were fabricated in the Clean Room for Analog-digital superconductIVITY (CRAVITY) of the National Institute of Advanced Industrial Science and Technology (AIST) with the standard process (STP2). The authors would like to

thank C. J. Fourie for providing the 3D inductance extractor (InductEx), Y. Abe for supporting measurements, and C. L. Ayala for proofreading the manuscript.

## References

- [1] T. Van Duzer, Superconductor digital electronics past, present, and future, *IEICE Trans. Electron.* **E91–C**, 260 (2008).
- [2] D. S. Holmes, A. M. Kadin, and M. W. Johnson, Superconducting computing in large-scale hybrid systems, *Computer (Long. Beach. Calif.)* **48**, 34 (2015).
- [3] K. K. Likharev and V. K. Semenov, RSFQ logic/memory family: a new Josephson-junction technology for sub-terahertz-clock-frequency digital systems, *IEEE Trans. Appl. Supercond.* **1**, 3 (1991).
- [4] M. Tanaka, F. Matsuzaki, T. Kondo, N. Nakajima, Y. Yamanashi, A. Fujimaki, H. Hayakawa, N. Yoshikawa, H. Terai, and S. Yorozu, A single-flux-quantum logic prototype microprocessor, in *2004 IEEE Int. Solid-State Circuits Conf. (IEEE Cat. No.04CH37519)* (IEEE, 2004), pp. 298–529.
- [5] Y. Ando, R. Sato, M. Tanaka, K. Takagi, N. Takagi, and A. Fujimaki, Design and demonstration of an 8-bit bit-serial RSFQ microprocessor: CORE e4, *IEEE Trans. Appl. Supercond.* **26**, 1301205 (2016).
- [6] O. A. Mukhanov, Energy-efficient single flux quantum technology, *IEEE Trans. Appl. Supercond.* **21**, 760 (2011).
- [7] Q. P. Herr, A. Y. Herr, O. T. Oberg, and A. G. Ioannidis, Ultra-low-power superconductor logic, *J. Appl. Phys.* **109**, 103903 (2011).
- [8] M. Tanaka, M. Ito, A. Kitayama, T. Kouketsu, and A. Fujimaki, 18-GHz, 4.0-aJ/bit operation of ultra-low-energy rapid single-flux-quantum shift registers, *Jpn. J. Appl. Phys.* **51**, 053102 (2012).
- [9] M. A. Manheimer, Cryogenic computing complexity program: phase 1 introduction, *IEEE Trans. Appl. Supercond.* **25**, 1301704 (2015).
- [10] S. Tolpygo, V. Bolkhovsky, T. Weir, A. Wynn, D. Oates, L. Johnson, and M. Gouker, Advanced fabrication processes for superconducting very large scale integrated circuits, *IEEE Trans. Appl. Supercond.* **26**, 1100110 (2016).
- [11] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, An adiabatic quantum flux parametron as an ultra-low-power logic device, *Supercond. Sci. Technol.* **26**, 035010 (2013).
- [12] M. Hosoya, W. Hioe, J. Casas, R. Kamikawai, Y. Harada, Y. Wada, H. Nakane, R. Suda, and E. Goto, Quantum flux parametron: a single quantum flux device for Josephson

- supercomputer, *IEEE Trans. Appl. Supercond.* **1**, 77 (1991).
- [13] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, Thermodynamic study of energy dissipation in adiabatic superconductor logic, *Phys. Rev. Appl.* **4**, 034007 (2015).
- [14] K. Likharev, Dynamics of some single flux quantum devices: I. Parametric quntron, *IEEE Trans. Magn.* **13**, 242 (1977).
- [15] J. G. Koller and W. C. Athas, Adiabatic switching, low energy computing, and the physics of storing and erasing information, in *Work. Phys. Comput.* (IEEE, 1992), pp. 267–270.
- [16] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, Adiabatic quantum-flux-parametron cell library adopting minimalist design, *J. Appl. Phys.* **117**, 173912 (2015).
- [17] Q. Xu, C. L. Ayala, N. Takeuchi, Y. Murai, Y. Yamanashi, and N. Yoshikawa, Synthesis Flow for Cell-based adiabatic quantum-flux-parametron structural circuit generation with HDL back-end verification, *IEEE Trans. Appl. Supercond.* **27**, 1301905 (2017).
- [18] C. L. Ayala, N. Takeuchi, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa, Majority-logic-optimized parallel prefix carry look-ahead adder families using adiabatic quantum-flux-parametron logic, *IEEE Trans. Appl. Supercond.* **27**, 1300407 (2017).
- [19] N. Tsuji, C. L. Ayala, N. Takeuchi, T. Ortlepp, Y. Yamanashi, and N. Yoshikawa, Design and implementation of a 16-word by 1-bit register file using adiabatic quantum flux parametron logic, *IEEE Trans. Appl. Supercond.* **27**, 1300904 (2017).
- [20] J. M. R. Parrondo, J. M. Horowitz, and T. Sagawa, Thermodynamics of information, *Nat. Phys.* **11**, 131 (2015).
- [21] N. Takeuchi and N. Yoshikawa, Minimum energy dissipation required for a logically irreversible operation, *Phys. Rev. E* **97**, 012124 (2018).
- [22] R. Landauer, Irreversibility and heat generation in the computing process, *IBM J. Res. Dev.* **5**, 183 (1961).
- [23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, *Science* **220**, 671 (1983).
- [24] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E* **58**, 5355 (1998).
- [25] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, Quantum annealing with manufactured spins, *Nature* **473**, 194 (2011).
- [26] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing, *IEEE J. Solid-State Circuits* **51**, 303 (2016).

- [27] A. Lucas, Ising formulations of many NP problems, *Front. Phys.* **2**, 1 (2014).
- [28] M. Aono, M. Naruse, S.-J. Kim, M. Wakabayashi, H. Hori, M. Ohtsu, and M. Hara, Amoeba-inspired nanoarchitectonic computing: solving intractable computational problems using nanoscale photoexcitation transfer dynamics, *Langmuir* **29**, 7557 (2013).
- [29] M. Aono, S. Kasai, S. Kim, M. Wakabayashi, H. Miwa, and M. Naruse, Amoeba-inspired nanoarchitectonic computing implemented using electrical Brownian ratchets, *Nanotechnology* **26**, 234001 (2015).
- [30] A. Biere, M. Heule, H. Van Maaren, and T. Walsh, *Handbook of Satisfiability* (IOS Press, 2009).
- [31] M. Aono, M. Hara, and K. Aihara, Amoeba-based neurocomputing with chaotic dynamics, *Commun. ACM* **50**, 69 (2007).
- [32] M. Aono and M. Wakabayashi, Amoeba-inspired heuristic search dynamics for exploring chemical reaction paths, *Orig. Life Evol. Biosph.* **45**, 339 (2015).
- [33] S. Kasai, M. Aono, and M. Naruse, Amoeba-inspired computing architecture implemented using charge dynamics in parallel capacitance network, *Appl. Phys. Lett.* **103**, 163703 (2013).
- [34] N. Takeuchi, S. Nagasawa, F. China, T. Ando, M. Hidaka, Y. Yamanashi, and N. Yoshikawa, Adiabatic quantum-flux-parametron cell library designed using a 10 kA cm<sup>-2</sup> niobium fabrication process, *Supercond. Sci. Technol.* **30**, 035002 (2017).
- [35] S. Nagasawa, Y. Hashimoto, H. Numata, and S. Tahara, A 380 ps, 9.5 mW Josephson 4-Kbit RAM operated at a high bit yield, *IEEE Trans. Appl. Supercond.* **5**, 2447 (1995).
- [36] N. Takeuchi, M. Aono, Y. Hara-Azumi, and C. L. Ayala, A circuit-level amoeba-inspired SAT solver, *ArXiv:1812.11792* (2018).
- [37] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co. New York, 1979.