

Operating Principles and Applications of Probabilistic Computing Based on Probabilistic Bits

YongJin Kim, Ki Hyuk Han, and OukJae Lee*

Center for Semiconductor Technology, Korea Institute of Science and Technology, Seoul 02792, South Korea

(Received 8 July 2026, Received in final form 20 July 2026, Accepted 22 July 2026)

Probabilistic computing has attracted considerable attention as a new computing paradigm that overcomes the limitations of conventional deterministic computing and efficiently solves combinatorial optimization and inference problems. In this review, we examine the operating principles and applications of probabilistic computing based on probabilistic bits (p-bits). We first describe the stochastic state fluctuations of a p-bit, the network dynamics based on Gibbs sampling, and the energy-based computing framework connected to the Ising model. Building on these principles, we explain invertible logic, in which a truth table is encoded into an energy cost function and solutions are obtained through stochastic sampling, using AND and OR gates and a Full-Adder as examples, and introduce its application to integer factorization as a representative inverse problem. We also review how representative combinatorial optimization problems, such as Max-Cut, the Travelling Salesman Problem (TSP), and Number Partitioning, are solved by mapping their objective functions and constraints onto QUBO or Ising model formulations, and discuss the possible extension to multi-state probabilistic computing using recently proposed multi-state probabilistic devices (p-dits). These discussions demonstrate that p-bit networks can effectively perform probabilistic inference and low-energy state searches, suggesting that they can serve as a key technology for inverse computation, combinatorial optimization, and energy-efficient next-generation computing.

Keywords : probabilistic computing, p-bit, invertible logic, Ising model, combinatorial optimization

확률 비트 기반 확률론적 컴퓨팅의 동작 원리와 응용

김용진 · 한기혁 · 이억재*

한국과학기술연구원 반도체기술연구단, 서울특별시 성북구 화랑로 14길 5, 02792

(2026년 7월 8일 받음, 2026년 7월 20일 최종수정본 받음, 2026년 7월 22일 게재확정)

확률론적 컴퓨팅(probabilistic computing)은 기존 결정론적 컴퓨팅의 한계를 극복하고 조합 최적화 및 추론 문제를 효율적으로 해결하기 위한 새로운 계산 패러다임으로 주목받고 있다. 본 해설 논문에서는 확률 비트(probabilistic bit, p-bit) 기반 확률론적 컴퓨팅의 동작 원리와 응용을 고찰하였다. 먼저 p-bit의 확률적 상태 변화, Gibbs sampling 기반 네트워크 동작, 그리고 Ising 모델과 연결되는 에너지 기반 계산 구조를 설명하였다. 이를 바탕으로 진리표(truth table)를 에너지 비용 함수(energy cost function)로 인코딩하고 확률적 샘플링을 수행하는 가역 논리(invertible logic)를 AND, OR 게이트 및 Full-Adder를 예로 들어 설명하고, 이를 대표적인 역연산 문제인 정수 소인수 분해에 적용하는 방법을 소개하였다. 또한 QUBO 및 Ising 모델 기반 매핑을 통해 Max-Cut, Travelling Salesman Problem(TSP), Number Partitioning과 같은 대표적인 조합 최적화 문제의 풀이 과정을 정리하였으며, 최근 제안된 다중 상태 확률 소자(p-dit)를 이용한 다중 상태 확률론적 컴퓨팅으로의 확장 가능성도 함께 논의하였다. 이러한 논의는 p-bit 네트워크가 확률적 추론과 낮은 에너지 상태 탐색을 효과적으로 수행할 수 있음을 보여주며, 향후 역연산, 조합 최적화 및 에너지 효율적인 차세대 컴퓨팅 분야의 핵심 기술로 활용될 수 있음을 시사한다.

주제어 : 확률론적 컴퓨팅, 확률 비트, 가역 논리, 이징 모델, 조합최적화

I. 서 론

최근 인공지능, 빅데이터 처리, 확률적 추론, 조합최적화 문제 등 대규모 연산을 요구하는 응용 분야가 확대됨에 따라 기존 폰 노이만 구조 기반 컴퓨팅의 에너지 소모와 데이터 이동에 따른 지연 시간 문제가 중요한 한계로 부각되고 있다. 기존 디지털 컴퓨팅은 논리 및 산술 연산에서 높은 정확성과 재현성을 제공하지만, 방대한 후보 상태를 탐색해야 하는 최적화 및 추론 문제에서는 계산 복잡도와 에너지 소모가 급격히 증가한다. 이러한 한계를 극복하기 위한 대안으로 확률론적 컴퓨팅(probabilistic computing)이 주목받고 있다[1-8].

확률론적 컴퓨팅은 결정론적으로 0 또는 1의 상태를 갖는 기존 이진 bit와 달리, 시간에 따라 0과 1 (또는 -1과 +1) 사이를 확률적으로 변동하는 확률 비트(probabilistic bit, p-bit)를 기본 연산 단위로 사용한다. P-bit은 외부 입력 신호에 의해 각 상태가 나타날 확률이 조절되며 일반적으로 i 번째 p-bit의 출력 스핀 상태 s_i 는 입력 I_i 에 의해 다음과 같이 표현된다[3,4,9].

$$s_i = \text{sgn}[\tanh(\beta I_i) + \text{rand}(-1, 1)] \quad (1)$$

여기서, β 는 입력에 대한 출력 민감도를 조절하는 역온도(inverse temperature)이며[10,11], $\text{rand}(-1, 1)$ 은 -1과 +1 사이에서 균일하게 분포하는 난수이고, sgn 함수는 입력값의 부호에 따라 -1 또는 +1을 출력한다. 따라서, p-bit은 본질적인 무작위성을 가지면서도 입력 신호를 통해 출력 상태의 확률 분포를 제어할 수 있다.

이러한 p-bit은 다양한 하드웨어 소자를 이용하여 구현될 수 있다. 대표적인 후보로는 CMOS 회로 기반의 난수 발생기[12,13], 메모리스트[14], 자기터널접합(magnetic tunnel junction, MTJ) 기반 소자[2,15,16] 등이 보고되고 있다. 여기서 MTJ 기반 p-bit은 비휘발성, 저전력, CMOS 공정과의 높은 호환성을 가지며, 열적 요동에 의해 자화의 상태가 확률적으로 변동하는 특성을 활용하며 입력 스핀 전류로 출력 확률을 제어할 수 있어 확률론적 컴퓨팅을 위한 유망한 하드웨어 플랫폼으로 주목받고 있다[2,5,16,17]. 최근에는 MTJ 기반 p-bit을 이용하여 가역 논리 연산, 조합최적화 문제 해결 및 확률론적 신경망의 구현을 위한 연구가 활발히 수행되고 있다.

이러한 응용은 다수의 p-bit을 상호 연결하여 구성된 p-bit 네트워크(p-network)를 기반으로 구현되며, 개별 p-bit간의 상호작용은 결합 가중치에 의해 결정된다. 일반적으로 j 번째 p-bit에 인가되는 입력은 다음과 같은 가중합 형태로 표현된다[1,3,4,5,17].

$$I_i = h_i + \sum_j J_{ij} s_j \quad (2)$$

여기서 h_i 는 i 번째 p-bit에 가해지는 국소 바이어스(local bias), J_{ij} 는 j 번째와 i 번째 p-bit 사이의 결합 가중치, 그리고 s_j 는 j 번째 p-bit의 상태이다. 각 p-bit은 주변 p-bit들의 현재 상태로 부터 식(2)로 계산된 입력 I_i 에 따라 자신의 상태를 확률론적으로 결정하고, 일반적으로 비동기(asynchronous)방식으로 순차적으로 갱신된다. 이러한 상태 갱신 과정은 Gibbs sampling과 밀접하게 관련되어 있으며, 반복적인 샘플링을 통해 네트워크는 주어진 확률분포를 따르거나 낮은 에너지 상태를 탐색하게 된다[18-20].

따라서, 확률론적 컴퓨팅에서는 해결하고자 하는 문제에 맞게 J_{ij} 와 h_i 를 설계함으로써 특정 상태가 높은 확률로 나타나도록 네트워크의 확률 분포를 조절한다[1,12]. 또한 p-network는 Hopfield network, Boltzmann machine, Bayesian network과 같은 에너지 기반 모델로 해석될 수 있으며, p-bit의 이진 상태를 Ising spin 변수에 대응시키면 문제의 목적함수, 제약조건, 그리고 논리 연산의 진리표(truth table)을 에너지 비용 함수(energy cost function)으로 매핑할 수 있다[3,21,22]. Energy cost function은 일반적으로 유효한 해 또는 최적해가 최소 에너지 상태에 대응하도록 설계하며, 대표적인 표현 방식으로 QUBO(quadratic unconstrained binary optimization)모델이다. 이는 목적함수와 제약조건을 이진 변수의 선형항과 이차항으로 표현하고, 제약조건을 위반하는 상태에 penalty energy 항을 부여함으로써 최적해가 최소 에너지를 갖도록 한다[23].

본 해설 논문에서는 확률론적 컴퓨팅의 동작 원리를 바탕으로 대표적인 응용을 두 가지 범주로 나누어 설명한다. 첫째, AND, OR, Full-Adder와 같은 기본 논리 게이트(Boolean logic gate)를 대상으로 하는 가역 논리(invertible logic)를 소개한다. 이 접근에서는 논리 게이트의 truth table을 만족하는 상태가 낮은 에너지를 갖도록 energy cost function을 설계하고, 특정 입력 또는 출력을 고정된 상태(clamping)에서 나머지 변수를 샘플링함으로써 순방향 논리 연산뿐만 아니라 역방향 추론을 수행한다. 둘째, 외판원 문제(Traveling Salesman Problem, TSP), 최대컷(Max-Cut), 수분할(Number Partitioning problem)문제와 같은 조합최적화 문제에 대한 Ising model 기반을 설명한다. 이 경우 목적함수와 제약조건을 Ising Hamiltonian 또는 energy cost function으로 매핑하고, p-network가 낮은 에너지 상태를 탐색함으로써 최적해 또는 근사해를 도출한다.

II. 가역 불리언 논리(Invertible Boolean Logic)

확률론적 컴퓨팅에서 불리언 논리(Boolean Logic)은 기존 디지털 회로의 결정론적 입력-출력 함수가 아니라, truth table

을 만족하는 상태를 확률적으로 샘플링하는 방식으로 구현된다. 기존 디지털 회로에서는 입력이 주어지면 출력이 유일하게 결정되는 단방향 연산이 수행되지만, p-bit 기반 네트워크에서는 입력 또는 출력 일부를 고정한 상태에서 나머지 변수를 확률적으로 샘플링할 수 있다. 이처럼 출력으로부터 가능한 입력 조합을 역으로 추론할 수 있는 계산 방식을 역연산 논리라고 한다.

1. Boolean logic gate의 확률적 표현

P-bit을 사용하는 확률론적 컴퓨팅에서는 Fig. 1(a)와 (b)에서처럼 각 불리언 변수를 p-bit으로 표현하고, 논리식을 만족하는 상태들이 높은 확률로 나타나도록 p-network를 구성한다[1,3,13]. 이때, 불리언 변수 $x_i \in \{0,1\}$ 는 Ising spin 형태의 p-bit 상태 $s_i \in \{-1,+1\}$ 로 변환되어 표현된다 [3,14,15].

$$s_i = 2x_i - 1 \quad \text{또는} \quad x_i = \frac{s_i + 1}{2} \quad (3)$$

2. Truth table 기반 energy cost function 설계

역연산 논리 풀이에서 가장 중요한 단계는 Fig. 1(c)와 (d)의 각 논리 게이트의 truth table을 energy cost function으로 변환하는 것이다[1,14,15]. 일반적으로 출력 y 가 입력 변수들의 논리 함수 $f(X)$ 로 주어지는 경우, 논리 게이트의 energy cost function은 다음과 같이 정의할 수 있다[23].

$$E_{logic}(x, y) = \lambda (y - f(X))^2 \quad (4)$$

여기서, X 는 입력 변수들의 집합, y 는 출력 변수, 그리고 λ 는 논리 조건 위반을 위반한 상태에 부여되는 penalty coefficient이다. 이 식은 $y = f(X)$ 를 만족하는 상태에서 최소 에너지 ($E_{logic} = 0$)를 가지며, truth table을 위반하는 상태에는 양의 penalty energy가 부여된다. 즉, truth table을 만족하는 상태는

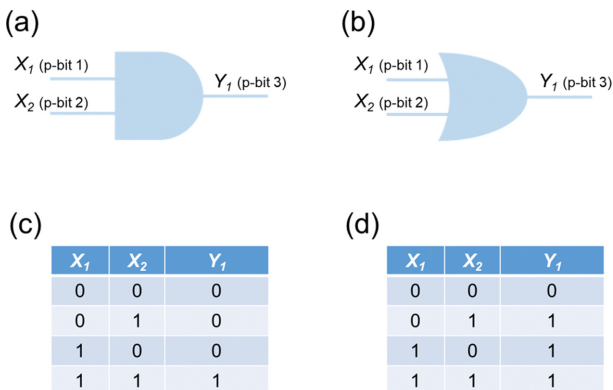


Fig. 1. (Color online) Basic Boolean logic gates used for p-bit-based invertible logic: (a) AND gate and (b) OR gate, with their corresponding truth tables shown in (c) and (d), respectively.

에너지 최소 상태가 되고, truth table을 위반하는 상태는 높은 에너지를 갖는다. 일반적으로 이러한 energy cost function은 논리 관계를 대수적 제약식(algebraic constraint)으로 표현한 후, 이를 penalty function 형태로 변환함으로써 유도할 수 있다 [23]. 예를 들어 AND 게이트는 출력 Y 가 두 입력 X_1 과 X_2 의 논리곱으로 정의되므로, $Y = X_1 X_2$ 의 관계를 만족해야 한다. 이에 energy cost function은 다음과 같이 표현된다[1,2,23].

$$E_{AND} = \lambda (Y - X_1 X_2)^2 \quad (5)$$

식(5)은 $(X_1, X_2, Y) = (0,0,0), (0,1,0), (1,0,0), (1,1,1)$ 과 같이 AND 게이트의 진리표를 만족하는 상태에서는 최소 에너지 ($E_{AND} = 0$)를 가지며, 이를 위반하는 상태에서는 양의 penalty energy를 갖는다. 마찬가지로 OR 게이트는 $Y = X_1 + X_2 - X_1 X_2$ 의 관계를 만족해야 하므로, energy cost function은 다음과 같이 표현될 수 있다[1,23].

$$E_{OR} = \lambda [Y - (X_1 + X_2 - X_1 X_2)]^2 \quad (6)$$

이와 같이 구성된 energy cost function은 truth table을 만족하는 논리 상태를 식(3) binary-to-spin 변환식을 이용하여 최소 에너지 상태로 인코딩하며, 이후 Ising model 또는 p-network의 결합 가중치 J_{ij} 와 local bias h_i 로 매핑되며, 각 p-bit은 확률적으로 상태를 갱신하면서 truth table을 만족하는 낮은 에너지 상태를 높은 확률로 샘플링하게 된다.

3. Clamping과 Gibbs sampling을 이용한 논리 연산

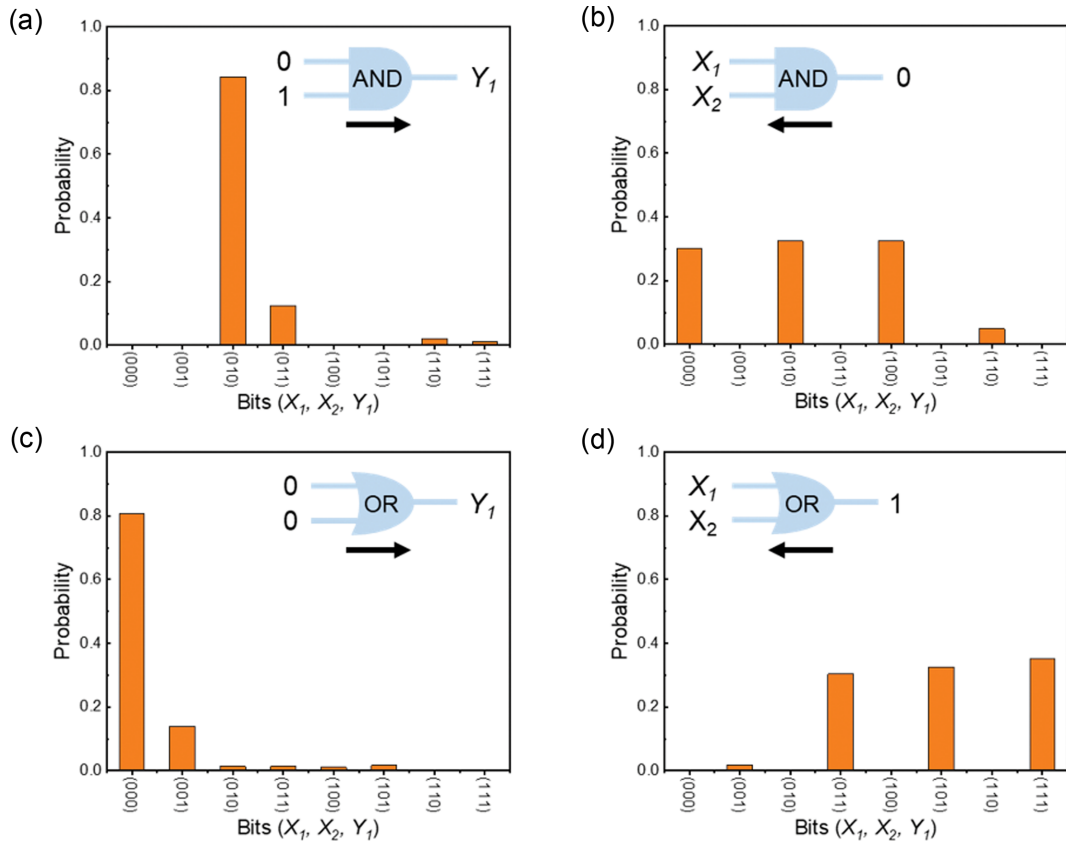
Clamping은 특정 p-bit의 상태를 외부 조건으로 고정하는 과정이다. 이때, 입력 변수를 clamping하면 순방향(forward) 논리 연산이 수행되며, 반대로 출력 변수를 clamping하면 해당 출력을 만족하는 가능한 입력 조합을 역방향(inverted or backward)으로 확률적으로 탐색한다. 예를 들어, AND 게이트에서 입력 X_1, X_2 를 고정하면 출력 Y 가 샘플링이 되고 $(X_1, X_2 \rightarrow Y)$, 반대로 출력 Y 를 고정하면, 가능한 입력 조합 X_1, X_2 가 샘플링된다($Y \rightarrow X_1, X_2$). 이때 각 p-bit은 주변 p-bit들의 상태가 주어졌을 때, 자신의 조건부 확률에 따라 비동기적으로 업데이트된다. 이러한 Gibbs sampling 과정이 반복되면 네트워크는 clamping 조건을 만족하면서 energy cost function이 낮은 상태들을 높은 빈도로 방문한다.

4. AND gate, OR gate 및 Full-Adder

다음으로 대표적인 예인 AND 게이트, OR 게이트 및 Full-Adder 게이트 동작에 대해서 살펴보겠다. 이들 논리 게이트는 모두 동일한 원리로 p-network로 구현될 수 있으며, 대응되는 논리 관계와 energy cost function은 Table I에 정리하였다[1,14,23].

Table I. Logic relations and energy cost functions of basic logic gates for invertible probabilistic computing.

Gate	Logic relation	Energy cost function
NOT	$Y = 1 - X$	$E_{NOT} = \lambda(Y + X - 1)^2$
COPY	$Y = X$	$E_{COPY} = \lambda(Y - X)^2$
AND	$Y = X_1 X_2$	$E_{AND} = \lambda(Y - X_1 X_2)^2$
OR	$Y = X_1 + X_2 - X_1 X_2$	$E_{OR} = \lambda[Y - (X_1 + X_2 - X_1 X_2)]^2$
XOR	$Y = X_1 + X_2 - 2X_1 X_2$	$E_{XOR} = \lambda[Y - (X_1 + X_2 - 2X_1 X_2)]^2$
Half Adder	$S = X_1 + X_2 - 2X_1 X_2$ $C = X_1 X_2$	$E_{HA} = \lambda(S - X_1 - X_2 + 2X_1 X_2)^2$ $+ \lambda(C - X_1 X_2)^2$
Full Adder	$S = X_1 + X_2 - 2X_1 X_2$ $- 2(X_1 X_2 + X_1 C_{in} + X_2 C_{in}) + 4X_1 X_2 C_{in}$ $C_{out} = X_1 X_2 + X_1 C_{in} + X_2 C_{in} - 2X_1 X_2 C_{in}$	$E_{FA} = \lambda[S - X_1 - X_2 - C_{in}$ $+ 2(X_1 X_2 + X_1 C_{in} + X_2 C_{in}) - 4X_1 X_2 C_{in}]^2$ $+ \lambda[C_{out} - X_1 X_2 - X_1 C_{in} - X_2 C_{in} + 2X_1 X_2 C_{in}]^2$

**Fig. 2.** (Color online) Forward and inverted operations of p-bit-based Boolean logic gates. Probability distributions obtained for the AND gate under (a) forward and (b) inverted operations, and for the OR gate under (c) forward and (d) inverted operations.

먼저, AND 게이트에 대해서 살펴보겠다. AND 게이트는 다음의 $Y = X_1 \wedge X_2$ 논리식을 따르며 truth table을 만족하는 상태는 Fig. 1(c)와 같다. AND 게이트의 energy cost function은 Table I에 제시한 바와 같이 truth table을 만족하는 4 가지 상태가 에너지 최소 상태를 갖도록 설계한다. 이때, 입력 X_1, X_2 를 (0,1)로 clamping하면, Fig. 2(a)와 같이 해당 입력에 대응하는 출력 $Y = 0$ 가 높은 확률로 샘플링 된다 [14]. 반대로, $Y = 0$ 을 clamping하면 Fig. 2(b)와 같이 가능

한 입력은 $(X_1, X_2) = (0,0), (0,1), (1,0)$ 이고 이들 조합으로 높은 확률로 샘플링 된다.

OR 게이트는 다음의 $Y = X_1 \vee X_2$ 논리식을 따르며 truth table을 만족하는 상태는 Fig. 1(d)와 같다. OR 게이트 역시 Table I에 정리된 energy cost function을 이용하여 truth table을 만족하는 4 가지 상태가 에너지 최소 상태를 갖도록 구현된다. Fig. 2(c)와 같이 입력 X_1, X_2 를 (0,0)로 clamping하면, 출력 Y 가 OR 연산 결과에 해당하는 상태로 샘플링 된다

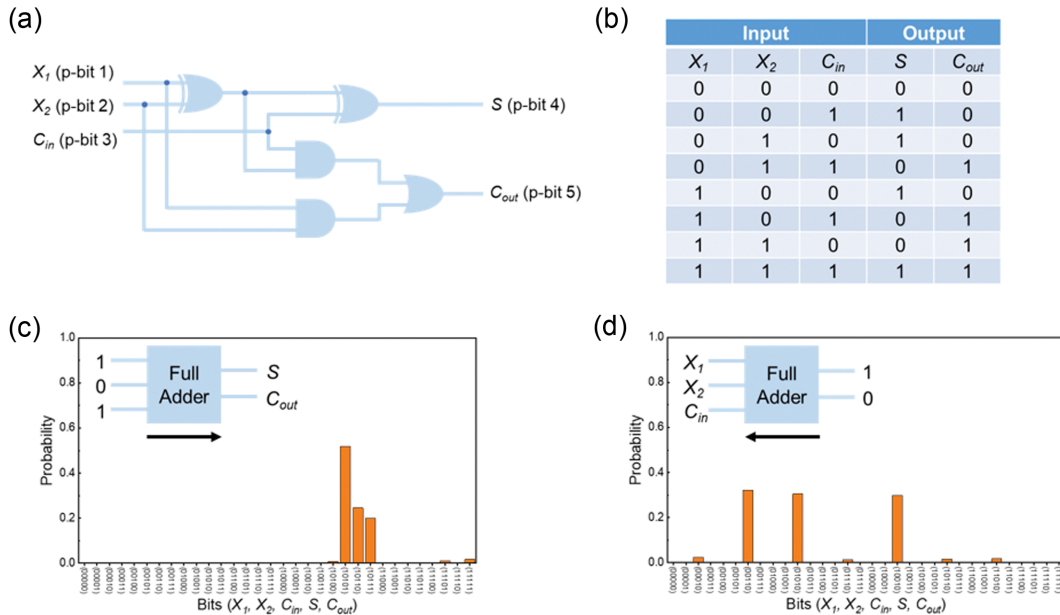


Fig. 3. (Color online) P-bit based probabilistic Full-Adder. (a) Schematic of a Full-Adder composed of XOR, AND, and OR gates. (b) Corresponding truth table. (c) Forward operation with input clamping and (d) inverted operation with output clamping.

[14]. 반대로 $Y = 1$ 을 clamping하면 Fig. 2(d)와 같이 가능한 입력은 $(X_1, X_2) = (0, 1), (1, 0), (1, 1)$ 이 가능한 입력 조합으로 샘플링 된다.

마지막으로, Full-Adder는 두 개의 1-bit 입력 X_1, X_2 와 입력 자리올림 비트 C_{in} 를 이용하여 덧셈 연산을 수행하는 가장 기본적인 산술 회로로 Fig. 3(a)와 같이 표현된다. 여기서, 합 비트(Sum, S)와 출력 자리올림 비트(C_{out})는 각각 $S = X_1 \oplus X_2 \oplus C_{in}$ 및 $C_{out} = X_1 X_2 + C_{in}(X_1 \oplus X_2)$ 의 관계를 만족한다. 대응되는 energy cost function은 Table I와 같이 표현되며, truth table은 Fig. 3(b)와 같다. P-bit 기반 Full-Adder에서도 truth table을 만족하는 상태가 낮은 에너지를 갖도록 설계한다. Fig. 3(c)와 같이 입력 X_1, X_2, C_{in} 을 (1, 0, 1)로 clamping하면 출력 S, C_{out} 가 Full-Adder 연산 결과에 해당하는 상태(0, 1)로 확률적으로 높게 샘플링 된다[1]. 반대로, 출력 S, C_{out} 를 (1, 0)으로 clamping하면 가능한 입력 상태들, (001), (010), (100)이 Fig. 3(d)와 같이 확률적으로 출력된다. 이처럼 Full-Adder는 순방향 연산뿐만 아니라 출력으로부터 가능한 입력 조합을 추론하는 역연산도 동일한 하드웨어에서 수행할 수 있으며, 이러한 특성은 보다 복잡한 산술 회로 및 정수 소인수분해와 같은 역연산 문제 응용의 기반이 된다.

5. 역연산 정수 소인수 분해

역연산은 확률론적 컴퓨팅의 대표적인 특징으로, 이를 이용한 정수 소인수 분해(integer factorization) 풀이 문제에 활발히 연구되고 있다[2, 24-28]. 정수 소인수 분해는 주어진 정수

(N)에 대하여 $X \times Y = N$ 을 만족하는 두 정수 X 와 Y 를 찾는 문제이다. 일반적인 디지털 시스템에서는 입력 값을 이용하여 출력 값을 계산하는 순방향 곱셈은 비교적 간단히 수행이 가능하지만, 출력 값만으로 입력 값을 찾는 역연산은 일반적으로 Shor's algorithm과 같이 별도의 방식을 요구한다. 확률론적 컴퓨팅에서는 정수 소인수 분해를 수행하기 위한 두 가지 접근 방식이 사용될 수 있다. 사용된다. 첫 번째 방법은 AND 게이트, Full-Adder 및 곱셈기(binary multiplier)를 이용하여 invertible multiplier를 구성한 뒤, 출력 값 N 을 clamping하여 입력 피연산자 A 와 B 를 역추론하는 방법이다 [2, 25, 26]. 두 번째 방법은 곱셈 관계 자체를 하나의 global energy cost function으로 정의하고, 에너지가 최소가 되는 상태를 확률적으로 탐색하는 방법이다[24, 27, 28]. 본 절에서는 후자의 접근 방식을 중심으로 설명한다.

정수 소인수 분해를 위한 energy cost function은 다음과 같이 정의될 수 있다[29].

$$E = \lambda (AB - N)^2 \quad (7)$$

여기서 A 와 B 는 소인수 후보이며, N 은 소인수 분해를 수행하고자 하는 정수, λ 는 논리 조건을 만족하지 않는 상태에 부여되는 penalty coefficient이다. 이때, $AB = N$ 을 만족하는 상태에서 최소 에너지($E = 0$)를 가지며, 조건을 만족하지 않는 상태는 양의 penalty energy를 부여한다. 예를 들어, Fig. 4(a)와 같이 2-bit X 2-bit 곱셈의 경우, 두 피연산자는 각각 $A = A_1 A_0$ 와 $B = B_1 B_0$ 로 표현되며, 출력은 $N = N_3 N_2 N_1 N_0$

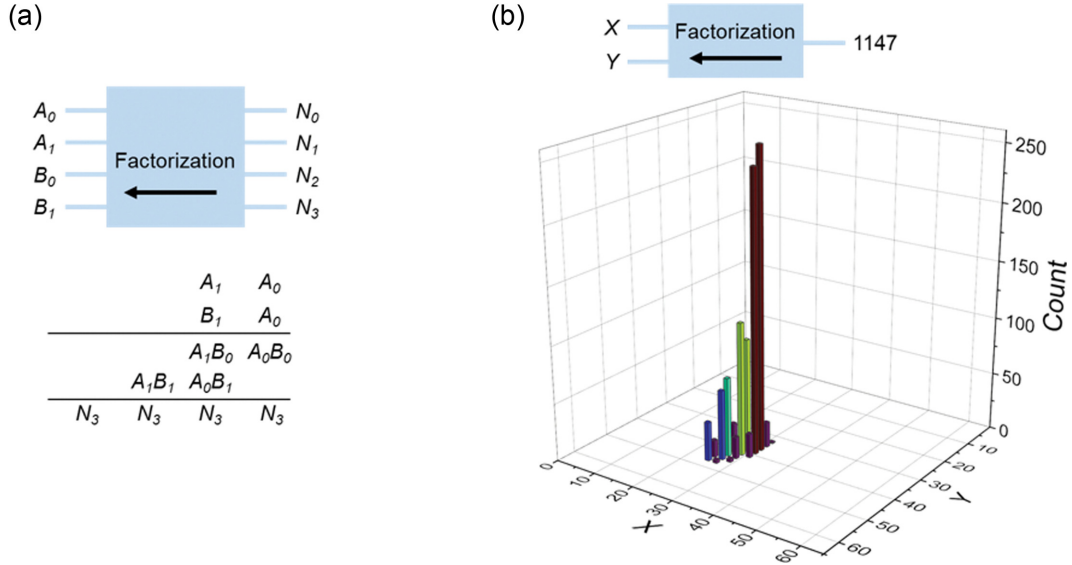


Fig. 4. (Color online) Invertible logic-based integer factorization using p-network. (a) Schematic representation of a 2-bit $\times$ 2-bit binary multiplier and its multiplication procedure. (b) Sampling distribution obtained from the inverted operation of a 6-bit $\times$ 6-bit multiplier with the output clamped to $N = 1147$, showing dominant factor states corresponding to $(X, Y) = (31, 37)$ and $(37, 31)$.

로 표현된다[24]. 따라서, 각 이진수는 $A = 2A_1 + A_0$, $B = 2B_1 + B_0$, $N = 2^3 N_3 + 2^2 N_2 + 2N_1 + N_0$ 와 같이 나타낼 수 있다. 이를 식(7)에 대입하면, energy cost function은 다음과 같이 전개된다.

$$E = \lambda \left[(2A_1 + A_0)(2B_1 + B_0) - (2^3 N_3 + 2^2 N_2 + 2N_1 + N_0) \right]^2 \quad (8)$$

여기서, $A_1, A_0, B_1, B_0, N_3, N_2, N_1, N_0$ 는 각각 A, B, N 을 구성하는 이진 비트이며, p-network에서는 각 비트가 하나의 p-bit으로 표현된다. 이후 각 p-bit의 입력 함수는 식(8)로 정의된 energy cost function으로부터 유도되며, 각 p-bit은 자신의 상태 변화에 따른 에너지 차이에 기반하여 확률적으로 상태를 갱신함으로써 낮은 에너지 상태를 탐색한다.

이를 피연산자의 비트 수를 증가시킴으로써, 보다 큰 정수에 대해서도 확장할 수 있다. Fig. 4(b)는 6-bit $\times$ 6-bit 곱셈 관계로 확장된 p-network를 이용하여 출력값 $N = 1147$ 의 소인수 분해를 수행한 결과를 나타낸다. 출력값을 clamping 한 상태에서 두 피연산자 X 와 Y 를 확률적으로 갱신하면, p-network는 energy cost function을 최소화하는 입력 상태를 탐색한다. 그 결과 $XY = 1147$ 을 만족하는 $(X, Y) = (37, 31)$ 과 $(31, 37)$ 상태가 높은 확률로 샘플링되었으며, 이를 통해 p-bit 기반 역연산을 이용한 정수 소인수 분해가 수행될 수 있음을 알 수 있다.

III. Ising 모델 기반 조합최적화 문제 풀이

Ising 모델 기반 조합최적화 탐색은 주어진 목적함수와 제

약조건을 반영한 energy cost function이 최소가 되는 상태를 탐색하는 데 초점을 둔다[3,4,20,22,23]. 앞선 진리표 기반의 접근법과 목적은 다르지만, 문제를 energy cost function으로 표현하고 p-bit 네트워크의 Gibbs sampling을 통해 낮은 에너지 상태를 탐색한다는 공통점을 갖는다.

1. 조합최적화 문제와 Ising/QUBO 매핑

조합최적화 문제는 주어진 목적함수를 최소화하거나 최대화하는 해를 찾는 문제이다. 대표적인 문제로는 Max-Cut, Traveling salesman problem(TSP), graph coloring, knapsack 문제 등이 있다. 이러한 문제들은 일반적으로 변수의 수가 증가함에 따라 경우의 수가 지수적으로 증가하기에 직접 탐색하여 해를 찾는 방식은 비효율적이다. 확률론적 컴퓨팅에서는 조합최적화 문제를 energy minimization 문제로 변환하여 해결한다. 즉, 문제의 목적함수와 제약조건을 하나의 energy cost function으로 표현하고, 이 함수의 값이 낮은 상태가 최적해 또는 근사 최적해에 해당하도록 설계한다. 이를 위해 널리 사용되는 표현이 Ising 모델과 QUBO(quadratic unconstrained binary optimization) 모델이다[3].

조합최적화 문제를 Ising 모델로 표현하기 위해, 먼저 Ising 모델은 -1 또는 $+1$ 값을 갖는 이진 spin 변수 S_i 를 사용하여 다음과 같은 Hamiltonian으로 표현된다[3,20,23].

$$H_{\text{Ising}} = -\sum_{i < j} J_{ij} S_i S_j - \sum_i h_i S_i \quad (9)$$

여기서 J_{ij} 는 i 번째 spin과 j 번째 spin 사이의 상호작용 계수

이며, h_i 는 i 번째 spin에 작용하는 local field이다. QUBO 모델은 0 또는 1 값을 갖는 binary 변수 x_i 를 사용하며, 일반적으로 다음과 같이 표현된다[23,30].

$$E_{QUBO}(X) = \sum_i Q_{ii} x_i + \sum_{i < j} Q_{ij} x_i x_j \quad (10)$$

여기서 Q_{ii} 는 각 변수에 대한 선형 계수, Q_{ij} 는 변수 간 상호작용을 나타내는 이차 계수이다. Ising모델과 QUBO모델은 서로 다른 변수 표현을 사용하지만, 식(3)을 통해 상호 변환될 수 있다. 따라서, 0/1 변수로 정의된 조합최적화 문제는 QUBO 형태로 표현한 뒤 Ising Hamiltonian으로 변환할 수 있으며, 반대로 Ising spin 기반 문제도 QUBO 형태로 나타낼 수 있다[3].

2. 최대컷(Max-Cut) 문제

Max-Cut 문제는 대표적인 NP-hard 조합최적화 문제로 회로 분할, 통신 네트워크, 머신 러닝, 물류 및 스케줄링 등의 문제를 Ising 모델 기반으로 가장 직접적으로 표현될 수 있는 대표적인 NP-hard 조합최적화 문제이다. 여기서 Fig. 5(a)와 같이 그래프는 정점(vertex)과 이를 연결하는 간선(edge)으로 구성되며, Max-Cut 문제는 주어진 그래프 $G = (V, E)$ 에서 각 정점을 두 개의 집합으로 분할하였을 때, 서로 다른 집합에 속한 정점 사이를 연결하는 edge weight의 합이 최대가 되도록 하는 최적의 분할 문제로 정의된다[20,23,31].

각 정점 i 에 Ising spin 변수 $s_i = \pm 1$ 을 할당하면, $s_i = +1$ 과 $s_i = -1$ 은 각각 서로 다른 두 집합을 의미한다. 두 정점 i 와 j 가 같은 집합에 속하면 $s_i s_j = +1$ 이고, 서로 다른 집합에 속하면 $s_i s_j = -1$ 이다. 따라서 edge (i, j) 가 cut에 포함되는지 여부는 다음 항으로 표현할 수 있다.

$$\frac{1 - s_i s_j}{2} \quad (11)$$

이 값은 두 정점이 서로 다른 집합에 속할 때는 +1이 되고,

같은 집합에 속할 때는 0이 된다. 따라서 Max-Cut의 목적함수는 다음과 같이 표현된다[20,23,32].

$$C_{MaxCut} = \frac{1}{2} \sum_{(i,j) \in E} w_{ij} (1 - s_i s_j) \quad (12)$$

여기서 w_{ij} 는 edge (i, j) 의 weight이다. Max-Cut은 C_{MaxCut} 을 최대화하는 문제이며, 이를 energy cost function을 최소화하는 형태의 energy minimization 문제로 바꾸면 다음과 같이 표현된다.

$$E_{MaxCut} = \frac{1}{2} \sum_{(i,j) \in E} w_{ij} s_i s_j \quad (13)$$

이 식에서 연결된 두 정점이 서로 다른 spin 상태를 가질 때 에너지가 낮아지므로, p-network는 edge로 연결된 정점들이 서로 다른 상태를 갖는 구성을 높은 확률로 샘플링하게 된다.

P-network에서 각 정점(vertex)은 하나의 p-bit에 대응되며, w_{ij} 는 두 p-bit 사이의 coupling weight로 반영된다. 따라서 Max-Cut 문제는 그래프의 정점과 edge weight 정보를 p-network의 결합 구조로 매핑하고, 샘플링을 통해 C_{MaxCut} 이 큰 spin configuration을 확률적으로 탐색하는 energy minimization 문제로 구현될 수 있다. Fig. 5(b)는 25개의 p-bit과 250개의 edge로 구성된 Max-Cut 문제에 대해 annealing 및 샘플링 과정에서의 energy 변화를 나타낸다. 초기에는 다양한 spin configuration이 샘플링되면서 energy가 크게 변동하지만, annealing이 진행됨에 따라 낮은 energy 상태가 점차 높은 확률로 샘플링 된다. 이후 최소 에너지 상태의 E_{MaxCut} 에 도달하며, annealing 이후의 샘플링 과정에서는 해당 최소 에너지 상태가 지배적으로 샘플링되고, 이에 대응하는 C_{MaxCut} 의 spin configuration이 Max-Cut 문제의 해로 선택된다[20].

3. 외판원 문제(Travelling Salesman Problem, TSP)

TSP는 대표적인 NP-hard 조합최적화 문제로, 물류 및 배송 경로 최적화, 반도체 회로 설계, 로봇 및 자율주행, 생산

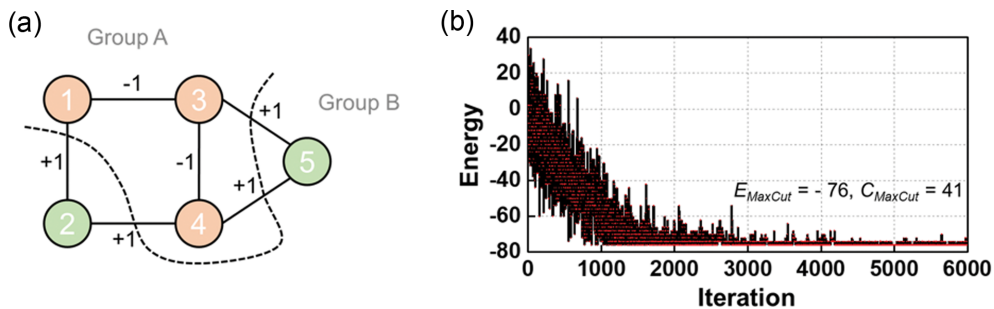


Fig. 5. (Color online) Example of a p-bit-based Max-Cut optimization. (a) Graph partition corresponding to the maximum-cut solution. (b) Evolution of the energy during annealing and sampling for a Max-Cut problem with 25 p-bits and 250 edges.

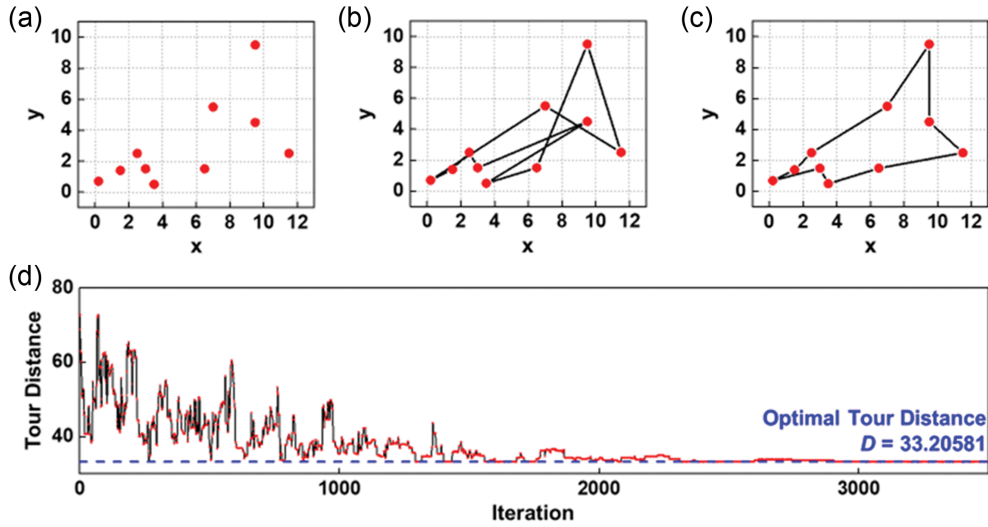


Fig. 6. (Color online) P-bit-based optimization of the travelling salesman problem (TSP). (a) City distribution for a 10-city TSP distance. (b) Randomly sampled tour under unconstrained p-bit fluctuations. (c) Optimized tour obtained after probabilistic annealing. (d) Evolution of the tour distance during sampling iterations, demonstrating convergence toward a low-energy solution corresponding to the optimal tour.

스케줄링 등 다양한 분야에 응용된다. TSP는 Fig. 6(a)와 같이 N 개의 도시가 주어졌을 때 각 도시를 정확히 한 번씩 방문하고 다시 출발 도시로 돌아오는 경로 중 전체 이동거리가 최소가 되는 최적의 경로를 찾는 문제이며, 가능한 경로의 수가 도시 수의 증가에 따라 기하급수적으로 증가하기 때문에 계산 복잡도가 매우 높은 문제로 알려져 있다[33-35]. TSP문제를 Fig. 6(a)에 나타난 바와 같이 QUBO 또는 Ising모델로 표현하기 위해서는, 각 도시와 방문 순서를 동시에 나타내는 binary 변수를 도입한다[34]. 일반적으로 $N \times N$ 행렬 $x_{i,t}$ 를 다음과 같이 정의한다[23,34].

$$x_{i,t} = \begin{cases} 1, & \text{도시 } i \text{가 방문 순서 } t \text{에 선택되는 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases} \quad (14)$$

여기서 i 는 도시의 index, t 는 방문 순서를 의미한다. TSP 문제에서는 두 가지 주요 제약조건이 필요하다. 첫째, 각 도시는 한 번 방문되어야 한다[23].

$$\sum_i x_{i,t} = 1 \quad (15)$$

둘째, 각 방문 순서에는 정확히 하나의 도시만 배정되어야 한다[23].

$$\sum_t x_{i,t} = 1 \quad (16)$$

이러한 제약조건을 만족하지 않는 상태에는 penalty energy가 부여되며, 유효하지 않은 경로가 낮은 에너지 상태로 선택되지 않도록 한다. 또한 실제 최적화 목표는 인접한 방문 순서 사이의 이동 거리 합을 최소화하는 것이다. 따라서 거리 항

은 다음과 같이 표현할 수 있다[23,34].

$$E_{dist} = \sum_{i,j} d_{ij} x_{i,t} x_{j,t+1} \quad (17)$$

여기서 d_{ij} 는 도시 i 에서 도시 j 로 이동하는 거리이다. 전체 TSP energy cost function은 제약조건 penalty와 거리 항을 포함하여 다음과 같이 표현된다[34].

$$E_{TSP} = A \sum_i \left(1 - \sum_t x_{i,t}\right)^2 + B \sum_t \left(1 - \sum_i x_{i,t}\right)^2 + C \sum_{i,j} d_{ij} x_{i,t} x_{j,t+1} \quad (18)$$

여기서 A 와 B 는 제약조건 위반에 대한 penalty coefficient이며, C 는 이동 거리 항의 가중치이다. 첫 번째 항은 각 도시가 한 번만 방문되도록 강제하고, 두 번째 항은 각 방문 순서에 하나의 도시만 배정되도록 강제한다. 세 번째 항은 전체 경로 길이를 최소화하는 목적함수에 해당한다.

P-network에서 각 binary 변수 $x_{i,t}$ 는 식(3) 변환을 통해 하나의 p-bit 또는 Ising 변수로 대응될 수 있다. 전체 TSP energy cost function을 QUBO 형태로 정리한 뒤, 식(2) 통해 Ising Hamiltonian으로 매핑하면, 해당 계수들은 p-network의 coupling weight와 local bias로 구현된다. Fig. 6(b)에서와 같이, 초기에는 p-bit들이 외부 입력 없이 무작위적 상태를 변화시키며 다양한 경로를 탐색한다. 이후 annealing 과정이 진행됨에 따라 p-network는 Fig. 6(c)에 나타난 것처럼 제약조건을 만족하면서 이동 거리가 짧은 경로를 탐색한다. 또한 Fig. 6(d)에서 확인할 수 있듯이, 샘플링이 반복됨에 따라 tour distance는 점진적으로 감소하며 최적해에 수렴한다[34]. 최종적으로 샘플링된 경로 중 energy cost가 가장 낮은 상태가 TSP의 최적해 또는 근사 최적해로 선택된다.

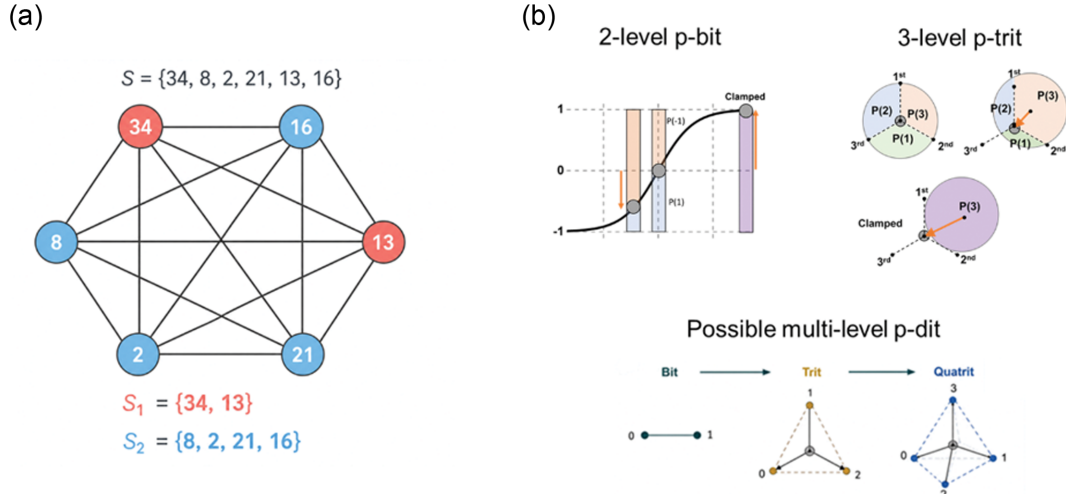


Fig. 7. (Color online) (a) Schematic representation of the number partitioning problem mapped onto an Ising graph. Each integer is represented by a binary spin variable and the optimal partition is obtained by minimizing the corresponding Ising energy function. (b) Conceptual extension from conventional binary p-bits to multi-level p-bits (p-dits), enabling the implementation of multi-way NPPs through probabilistic computing.

4. 수분할 문제(Number Partitioning Problem, NPP)

NPP는 주어진 정수 집합을 두 개의 부분집합으로 분할할 때, 각 부분집합의 원소 합의 차이가 최소가 되도록 하는 대표적인 NP 조합최적화 문제로[22,23,36], 작업 스케줄링, 부하 분산, 병렬 컴퓨팅, 자원 배분 등 다양한 최적화 분야에 응용되며, 최근에는 Ising 모델 기반 확률론적 컴퓨팅 하드웨어의 대표적인 벤치마크 문제로 활용되고 있다. 예를 들어, 집합 $A = \{a_1, a_2, \dots, a_N\}$ 가 주어졌을 때, 각 원소를 두 그룹 중 하나에 할당하여, 다음의 식과 같이

$$\left| \sum_{i \in S_1} a_i - \sum_{i \in S_2} a_i \right| \quad (19)$$

최소화하는 분할을 찾는 것이 목표이다[22,23]. 각 원소 a_i 에 대해 Ising spin 변수 $s_i \in \{-1, +1\}$ 를 정의하면, $s_i = +1$ 은 첫 번째 부분집합 S_1 , $s_i = -1$ 은 두 번째 부분집합 S_2 에 포함되는 것을 의미한다. 그러면 두 부분집합의 합의 차이는 다음과 같이 표현된다[22,23].

$$D = \sum_{i=1}^N a_i s_i \quad (20)$$

이를 최소화하기 위한 energy cost function은 $E = D^2$ 이며, 이는 다음과 같이 전개된다.

$$E = \left(\sum_{i=1}^N a_i s_i \right)^2 = \sum_{i=1}^N a_i^2 + 2 \sum_{i < j} a_i a_j s_i s_j \quad (21)$$

여기서, 첫번째 term은 상수항으로 energy minimization 과정에 영향을 미치지 않는다. 따라서 NPP 문제는 다음과 같은 Ising Hamiltonian으로 표현될 수 있다[22,23].

$$H = 2 \sum_{i < j} a_i a_j s_i s_j \quad (22)$$

즉, p-network에서 coupling weight는 $J_{ij} = 2a_i a_j$ 로 설정된다. 각 p-bit은 주변 p-bit들과의 coupling 관계에 따라 확률적으로 상태를 갱신하며, sampling 과정을 통해 시스템은 점차 낮은 에너지 상태로 수렴한다. 이때, 충분한 sampling 이후 가장 높은 빈도를 갖는 상태는 두 부분집합의 합의 차이를 최소화하는 최적 또는 근사 최적해에 대응된다.

Fig. 7(a)는 NPP 문제를 Ising energy function으로 변환한 예를 나타낸 것이다[37]. 각 정수는 하나의 spin 변수에 대응되며, 두 부분집합으로의 할당은 spin의 상태로 표현된다. 노드 간의 coupling weight는 $J_{ij} = 2a_i a_j$ 에 비례하여 정의되며, 에너지 최소화 과정을 통해 두 부분집합의 합의 차이가 최소가 되는 최적의 분할을 얻을 수 있다.

최근에는 Fig. 7(b)와 같이 기존의 binary p-bit을 다중 상태를 갖는 multi-level p-bit (p-dit)으로 확장하여, 수분할 문제를 2개 이상 여러 개의 부분집합으로 일반화한 Multi-way (or K-) NPP에 적용하려는 연구가 보고되고 있다[22,38,39]. 하나의 p-dit은 두 개 이상의 상태를 표현할 수 있으므로 필요한 변수의 수를 감소시키고, 보다 복잡한 조합최적화 문제를 효율적으로 해결할 수 있다. 이는 기존의 이진 p-bit 기반 확률론적 컴퓨팅이 다양한 다중 상태 조합최적화 문제로 확장될 수 있음을 시사한다.

IV. 전망 및 결론

차세대 컴퓨팅 기술은 목표로 하는 연산과 구현 방식에 따라 서로 다른 특성을 갖는다. 딥러닝 기반의 패턴 인식 및

생성형 인공지능은 높은 학습 및 추론 성능을 제공하지만, 대규모 행렬 연산과 메모리 접근으로 많은 연산 자원과 에너지를 필요하며, 뉴로모픽 컴퓨팅은 신경계의 병렬적 동작을 모사하여 이를 저전력에 유리하지만, 알고리즘과 하드웨어의 공동 설계가 필요하며 범용 연산으로의 확장에 어려움이 있다 [40-42]. 양자 컴퓨팅은 양자 중첩과 얽힘 현상을 이용하여 특정 문제에서 계산 가속의 가능성을 제공하지만, 큐비트 (qubit)의 잡음과 결맞음 및 대규모 집적에 여전히 한계가 존재한다 [43]. 이에 비해 확률론적 컴퓨팅은 상온 동작과 기존 CMOS 공정과의 호환성을 바탕으로 학습이 되지 않는 문제 풀이에 확률적 상태 변화와 에너지 최소화를 연산 자원으로 활용하여, 다양한 계산학적 난제 문제들을 하나의 에너지 기반 연산 구조에서 다룰 수 있다는 점에서 차별성을 가진다.

본 논문에서는 p-bit 기반 확률론적 컴퓨팅의 기본 원리와 대표적인 응용 사례를 통해 그 동작 방식을 고찰하였다. 먼저 p-bit의 확률적 상태 변화와 Gibbs sampling 기반 동작을 설명하고, p-network가 energy cost function을 기반으로 다양한 상태를 확률적으로 탐색하는 과정과 함께, 이것이 Ising model과 연결되는 에너지 기반 계산 구조임을 기술하였다. 이를 바탕으로 두 가지 응용 범주를 살펴보았다. 첫째, AND, OR 게이트 및 Full-Adder를 대상으로 하는 가역 논리 (invertible logic)에서는 truth table을 energy cost function으로 인코딩함으로써 순방향 논리 연산뿐만 아니라 출력으로부터 입력을 추론하는 역방향 연산이 가능함을 보였으며, 이러한 특성이 정수 소인수 분해와 같은 역연산 문제에 응용될 수 있음을 소개하였다. 둘째, Max-Cut, TSP, Number Partitioning과 같은 조합 최적화 문제에서는 목적함수와 제약조건을 Ising Hamiltonian 또는 QUBO 형태로 매핑하여 low-energy state를 탐색하는 과정을 기술하였다. 이러한 사례들 외에도 p-bit 기반 확률론적 컴퓨팅은 Boolean satisfiability (SAT), graph coloring, knapsack, job-shop scheduling 및 vehicle routing problem (VRP)과 같은 다양한 NP-hard 문제로의 적용 가능성이 활발히 연구되고 있다.

확률론적 컴퓨팅은 확률적 샘플링과 에너지 최소화 과정 자체를 계산 자원으로 활용한다는 점에서 기존의 결정론적 디지털 컴퓨팅과 근본적으로 차별화된다. 특히 p-bit 기반 확률론적 컴퓨팅은 CMOS 회로와의 높은 호환성과 하드웨어 구현 용이성을 바탕으로, 향후 조합 최적화, 확률적 추론, 역연산 컴퓨팅, 인공지능 등 에너지 효율이 요구되는 차세대 컴퓨팅 분야에서 중요한 계산 패러다임으로 발전할 것으로 기대된다. 아울러 최근 제안되고 있는 multi-level p-bit과 같은 확장된 확률 변수 표현 방식은 보다 복잡한 조합 최적화 문제를 효율적으로 다룰 수 있는 새로운 연구 방향을 제시할 것이다.

감사의 글

This work was supported by the KIST Institutional Program, and the National Research Council of Science & Technology (NST) grant by the Korea government (MSIT) (No. GTL24042-000).

References

- [1] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, *Phys. Rev. X* **7**, 031014 (2017).
- [2] W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, *Nature* **573**, 390 (2019).
- [3] K. H. Han, Y. Kim, H. C. Koo, and O. J. Lee, *J. Korean Magn. Soc.* **35**, 97 (2025).
- [4] K. H. Han, Y. J. Kim, H. C. Koo, O. J. Lee, and S. Hong, *Appl. Phys. Lett.* **125**, 142402 (2024).
- [5] J. Kaiser and S. Datta, *Appl. Phys. Lett.* **119**, 150503 (2021).
- [6] N. A. Aadit, A. Grimaldi, M. Carpentieri, L. Theogarajan, J. M. Martinis, G. Finocchio, and K. Y. Camsari, *Nat. Electron.* **5**, 460 (2022).
- [7] S. Chowdhury, A. Grimaldi, N. A. Aadit, S. Niazi, M. Mohseni, S. Kanai, H. Ohno, S. Fukami, L. Theogarajan, G. Finocchio, S. Datta, and K. Y. Camsari, *IEEE J. Explor. Solid-State Comput. Devices Circuits* **9**, 1 (2023).
- [8] S. Misra, L. C. Bland, S. G. Cardwell, J. A. C. Incorvia, C. D. James, A. D. Kent, C. D. Schuman, J. D. Smith, and J. B. AIM-one, *Adv. Mater.* **35**, e2204569 (2023).
- [9] K. Y. Camsari, S. Salahuddin, and S. Datta, *IEEE Electron Device Lett.* **38**, 1767 (2017).
- [10] R. A. Rutenbar, *IEEE Circuits and Devices Mag.* **5**, 19 (1989).
- [11] D. Bertsimass and J. Tsitsiklis, *Stat. Sci.* **8**, 10 (1993).
- [12] A. Z. Pervaiz, B. M. Sutton, L. A. Ghasasala, and K. Y. Camsari, *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 1920 (2019).
- [13] W. Lee, H. Kim, H. Jung, Y. Choi, J. Woo, and C. Kim, *Sci. Rep.* **15**, 8018 (2025).
- [14] K. S. Woo, J. Kim, J. Han, W. Kim, Y. H. Jang, and C. S. Hwang, *Nat. Commun.* **13**, 5762 (2022).
- [15] J. Daniel, Z. Sun, X. Zhang, Y. Tan, N. Dilley, Z. Chen, and J. Appenzeller, *Nat. Commun.* **15**, 4098 (2024).
- [16] N. S. Singh, K. Kobayashi, Q. Cao, K. Selcuk, T. Hu, S. Niazi, N. A. Aadit, S. Kanai, H. Ohno, S. Fukami, and K. Y. Camsari, *Nat. Commun.* **15**, 2685 (2024).
- [17] K. Y. Camsari, B. M. Sutton, and S. Datta, *Appl. Phys. Rev.* **6**, 011305 (2019).
- [18] S. Geman and D. Geman, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721 (1984).
- [19] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, *Cog. Sci.* **9**, 147 (1985).
- [20] N. Onizawa, and T. Hanyu, *Sci. Rep.* **14**, 1339 (2024).
- [21] J. J. Hopfield, *PNAS* **79**, 2554 (1982).

- [22] K. H. Han, G. Park, J. U. Ahn, Y. J. Kim, D. H. Yun, S.-H. Jang, Y.-J. Nah, M.-G. Kang, H. C. Koo, and O. J. Lee, *Sci. Rep.* **15**, 28881 (2025).
- [23] A. Lucas, *Front. Phys.* **2**, (2014).
- [24] J.-K. Han, J.-Y. Park, S. Rehman, M. F. Khan, M.-S. Kim, and S. Kim, *InfoMat* **7**, e70018 (2025).
- [25] A. Z. Pervaiz, L. A. Ghantasala, K. Y. Camsari, and S. Datta, *Sci. Rep.* **7**, 10994 (2017).
- [26] Y. Liu, Q. Hu, Q. Wu, X. Liu, Y. Zhao, D. Zhang, Z. Han, J. Cheng, Q. Ding, Y. Han, B. Peng, H. Jiang, X. Xue, H. Lv, and J. Yang, *Micromachines* **13**, 924 (2022).
- [27] H. Jung, H. Kim, W. Lee, J. Jeon, Y. Choi, T. Park, and C. Kim, *Sci. Rep.* **13**, 16186 (2023).
- [28] J. Jeon and C. Lim, *Electronics* **15**, 2510 (2026).
- [29] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Suter, and J. Du, *Phys. Rev. Lett.* **101**, 220405 (2008).
- [30] F. Glover, G. Kochenberger, and Y. Du, *arXiv:1811.11538* (2018).
- [31] M. X. Goemans and D. P. Williamson, *J. ACM* **42**, 1115 (1995).
- [32] F. Barahona, *J. Phys. A: Math. Gen.* **15**, 3241 (1982).
- [33] X. Geng, Z. Chen, W. Yang, D. Shi, and K. Zhao, *Appl. Soft Comput.* **11**, 3680 (2011).
- [34] J. Si, S. Yang, Y. Cen, J. Chen, Y. Huang, Z. Yao, D.-J. Kim, K. Cai, J. Yoo, X. Fong, and H. Yang, *Nat. Commun.* **15**, 3457 (2024).
- [35] R. Zhang, X. Li, C. Wan, R. Hoffmann, M. Hindenberg, Y. Xu, S. Liu, D. Kong, S. Xiong, S. He, A. Vardar, Q. Dai, J. Hong, Y. Sun, Z. Zheng, T. Kämpfe, G. Yu, and X. Han, *Nat. Commun.* **17**, 189 (2026).
- [36] N. Al-Kayed, C. St-Arnault, H. Morison, A. Aadhi, C. Huang, A. N. Tait, D. V. Plant, and B. J. Shastri, *Nature* **648**, 576 (2025).
- [37] S. Mertens, *arXiv:cond-mat/0310317* (2003).
- [38] Y. He, S. Luo, C. Fang, and G. Liang, *Sci. Rep.* **14**, 15076 (2024).
- [39] C. Duffee, J. Athas, A. Grimaldi, D. Volpe, G. Finocchio, E. Wei, and P. K. Amiri, *Phys. Rev. Appl.* **24**, 044077 (2025).
- [40] Y. LeCun, Y. Bengio, and G. Hinton, *Nature* **521**, 436 (2015).
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Adv. Neural Inf. Process. Syst.* **30**, 5998 (2017).
- [42] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, *Nat. Comput. Sci.* **2**, 10 (2022).
- [43] J. Preskill, *Quantum* **2**, 79 (2018).