

Evaluation of $|Y\rangle$ Magic State Distillation Circuit

Youngchul Kim*, Soo-Cheol Oh*, Sangmin Lee*, Ki-Sung Jin*, Gyuil Cha*

*Future Computing Research Division, ETRI, Daejeon, Republic of Korea

kimyc@etri.re.kr, ponylife@etri.re.kr, sanglee@etri.re.kr, ksjin@etri.re.kr, gicha@etri.re.kr

Abstract— For a universal quantum computer, surface code-protected logical Clifford and non-Clifford gates must be supported fault-tolerantly. However, to implement non-Clifford gates, magic states are required, and since these magic states are faulty, distillation circuits are used to obtain high-fidelity magic states by utilizing multiple low-fidelity states. It is not easy to implement and simulate the operations of a distillation circuit because it requires many resources. This paper presents a resource-efficient implementation and evaluation of a $|Y\rangle$ magic state distillation circuit on a quantum simulator.

Keywords— fault-tolerant quantum computing, quantum error correction, surface code, lattice surgery, magic state distillation

I. INTRODUCTION

Developing a fault-tolerant quantum computer with too noisy qubits and gates on a quantum device is challenging. Much research is on quantum error correction to support fault tolerance with topological codes such as surface code [1][2]. Surface code provides the most prominent quantum error correction method to implement fault-tolerant quantum computation due to its high error threshold of around 1% and simple two-dimensional structure with only nearest-neighbour connectivity [1]. However, it takes many physical qubits to encode a logical qubit with surface code. Also, the surface code should be encoded with more physical qubits to provide a lower error rate. Various techniques, such as braiding [1], lattice surgery [2][3], and twist [4], can implement logical operations between logical qubits protected by surface code. Out of these techniques, lattice surgery can reduce cost and complexity while maintaining two-dimensional layout.

We need to provide a quantum computer that supports universal quantum gates such as Clifford and non-Clifford gates to reap the benefits of quantum computing. Clifford gates, including Pauli gates, can be effectively simulated on a classical computer [7] and implemented by lattice surgery. However, it takes complex work to implement non-Clifford gates logically. In particular, the logical S and T gates require the magic state that can be prepared with the state injection process. Since state injection is not fault-tolerant, the injected state has low fidelity and needs to be distilled by the magic state distillation procedure. However, obtaining a higher-fidelity magic state from multiple lower-fidelity states requires many resources and time. Previous studies [3][6] have proposed several magic state distillation protocols and have calculated their costs numerically.

The common distillation protocols are to implement the 7-qubit Steane code for the magic state $|Y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ consumed at the logical S gate to purify the state and the 15-qubit Reed-Muller code for the magic state $|A\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle)$ consumed at the logical T gate [1].

Using the quantum simulator QPlayer [8][9], we have implemented the $|Y\rangle$ state distillation circuit in two different ways: a multi-target CNOT implementation in lattice surgery and an implementation based on lattice surgery translation [5], evaluated the cost of the resources used in them. Here, we considered the placement of logical qubits to minimize the resources required to perform multi-target CNOT in lattice surgery.

This paper is organized as follows. In section II, we first introduce the multi-target CNOT-based distillation and lattice surgery translation-based distillation process for $|Y\rangle$ state and explain how $|Y\rangle$ state is purified through these processes. Then, in section III, we implemented the above circuits on different placements of logical qubits and evaluated the cost of the resources.

II. DISTILLATION PROCESSES FOR $|Y\rangle$ STATE

We need a high-fidelity magic state $|Y\rangle$ to implement a logical S gate using gate teleportation. A magic state is first created by a process called state injection. However, the injected state can be imperfect, so it is then purified to a high-fidelity state using a distillation process.

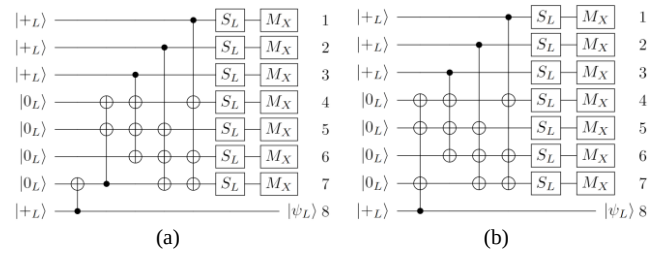


Figure 1. $|Y\rangle$ state distillation circuit implementing 7-qubit Steane code.

A. Multi-target CNOT-based $|Y\rangle$ state distillation

The distillation circuit implementing 7-qubit Steane code for $|Y\rangle$ magic state is shown in Figure 1. This circuit consists of initializing $|0\rangle$ and $|+\rangle$ states, multi-target CNOTs, logical S gate teleportation, and X -basis measurement. Figure 1(b) circuit is translated from Figure 1(a) according to the commutation relations between different non-commuting

CNOTs to minimize the number of CNOTs and implement the lattice surgery translation-based technique.

For $|Y\rangle$ state distillation circuit, it first creates a logical Bell pair. Then, one logical qubit from the Bell pair with the other six logical qubits is encoded in Steane code to entangle eight logical qubits using the multi-target CNOTs. After that, seven encoded qubits are each rotated with an S gate using $|Y\rangle$ state, which is injected in the first round of distillation or is distilled in a prior round. The logical S gate can be implemented with lattice surgery-based operations as shown in Figure 3(b).

The distillation circuit produces a purified $|Y\rangle$ state from the other logical qubit of the Bell pair. This procedure is performed iteratively to obtain a higher fidelity output state based on the X -basis measurement outcomes of the seven qubits. The output state is interpreted by evaluating the eigenvalues of the Steane code X stabilizers such as $S_1 = M_{X1}M_{X4}M_{X6}M_{X7}$, $S_2 = M_{X2}M_{X5}M_{X6}M_{X7}$, $S_3 = M_{X3}M_{X4}M_{X5}M_{X6}$. If the eigenvalues of all three stabilizers are +1, the output is purified $|Y\rangle$ state and will be kept. Otherwise, the state is discarded, and the subsequent distillation round must be performed. Furthermore, if the product of all measurements is 1, the output state will include a Z -error so that it needs to be tracked.

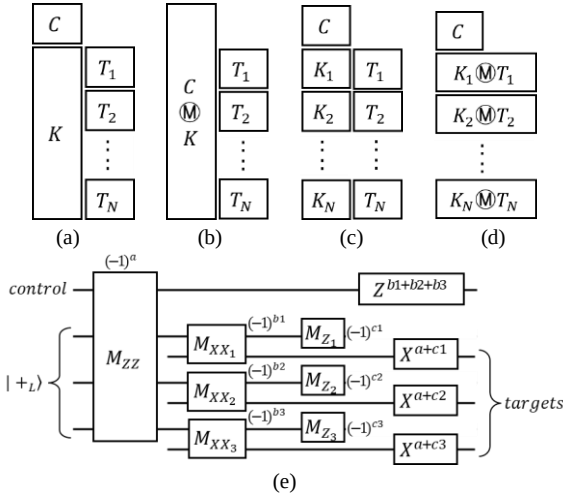


Figure 2. Multi-target CNOT in lattice surgery. C and $T_{1..N}$ refer to control and target qubits. K indicates the vertical rectangular ancilla qubits. (a) The vertical rectangular ancilla qubit is initialized to the $|+\rangle$ state. (b) The control qubit performs logical joint measurements, M_{ZZ} , with the vertical rectangular ancilla qubit. (c)(d) It is then split to perform logical joint measurements, M_{XX} , with the target qubits. (e) Lattice surgery-based multi-target CNOT circuit that implements the previously outlined the multi-target CNOT.

Lattice surgery logical operations between logical qubits allow us to perform state teleportation and gate teleportation of logical qubits, as shown in Figures 3(a) and 3(b), respectively.

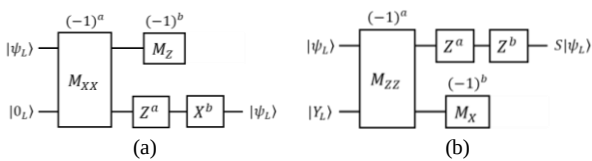


Figure 3. State teleportation and logical S gate in lattice surgery.

B. Lattice surgery translation-based $|Y\rangle$ state distillation

Figure 4 shows that CNOTs with different controls and the same target can be implemented using lattice surgery. It first initializes the two patches corresponding to the control qubits of each CNOT with $|+\rangle$ in Figure 4(b). It then performs a smooth split operation on each patch in Figure 4(b) and a rough merge operation on the two split patches. Eventually, the state of the three patches will be the same as the result in Figure 4(a) [5].

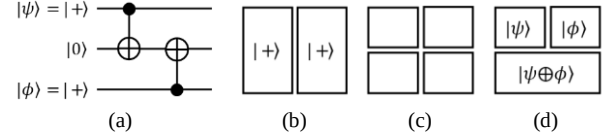


Figure 4. Implementation of CNOTs with the same target and different controls to extend to multi-target CNOTs using lattice surgery.

Lattice surgery-based CNOT with different controls and the same target can be extended to a $|Y\rangle$ state distillation circuit consisting of multi-target CNOTs, as shown in Figure 1(b).

III. EVALUATION OF $|Y\rangle$ STATE DISTILLATION PROCESSES

In this section, we have implemented the $|Y\rangle$ state distillation circuits mentioned in Section II using the QPlayer quantum simulator and evaluated each circuit.

A. Logical qubit architecture

For this work, we have used surface code logical qubits with a distance of 2, as shown in Figure 5(a) [10]. Lattice surgery merging and splitting operations along the $X(Z)$ -boundary between two logical qubits are shown in Figures 5(b) and 5(c).

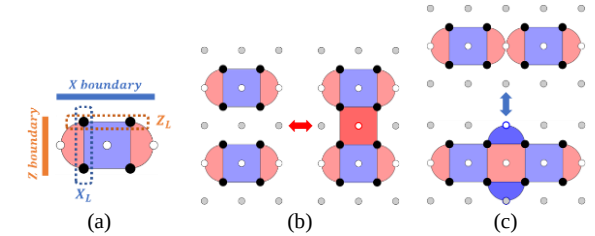


Figure 5. Layout of surface code logical qubit and lattice surgery between two logical qubits. (a) Surface code logical qubit with distance 2. Physical data qubits are represented by black circles, and ancilla qubits by white circles. The purple(pink) plaquettes represent the $X(Z)$ stabilizers, respectively. (b) Logical joint measurements along the X -boundary, M_{ZZ} . (c) Logical joint measurements along the Z -boundary, M_{XX} .

The $|Y\rangle$ state distillation circuits have been implemented in three logical qubit architectures, as shown in Figure 6. The logical qubit architecture with 24 logical qubits in Figure 6(a) and the logical qubit architecture with 18 logical qubits in Figure 6(b) perform a multi-target CNOT-based $|Y\rangle$ state distillation circuit. The logical qubit architecture with 20 logical qubits in Figure 6(c) performs a lattice surgery-based $|Y\rangle$ state distillation circuit. Logical operations of $|Y\rangle$ state distillation circuits do not use some logical qubits in the logical qubit architectures, but we consider a checkerboard form for simplicity.

The two logical qubits must be placed in the nearest neighboring locations to perform lattice surgery merging and splitting operations, which involve multi-target CNOT, state teleportation, and gate teleportation in a logical qubit architecture. Therefore, it is essential to properly arrange the control and target qubits for multi-target CNOT in the $|Y\rangle$ state distillation circuit.

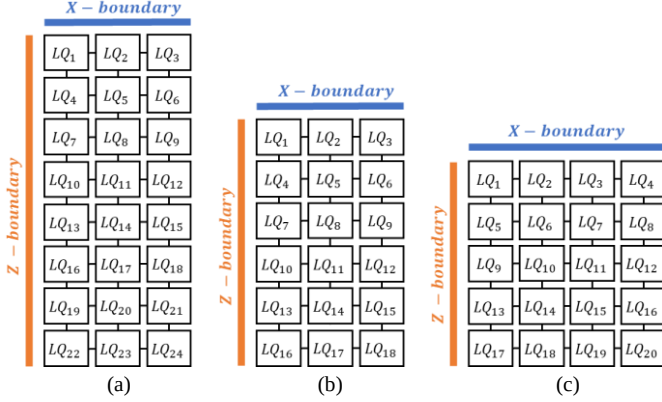


Figure 6. Logical qubit architectures for implementing $|Y\rangle$ state distillation circuits.

B. Multi-target CNOT-based $|Y\rangle$ state distillation

We have implemented the multi-target CNOT-based $|Y\rangle$ state distillation circuit on the logical qubit architecture in Figures 6(a) and 6(b), arranged in three forms: MT-CNOT-1, MT-CNOT-2, and MT-CNOT-3, respectively, as shown in Figure 7. In Figure 7, the patch numbers denote the logical qubits in the circuit, and K indicates the vertical rectangular ancilla qubits required to perform the lattice surgery-based multi-target CNOTs of Figure 2.

In Figure 7(MT-CNOT-1), the vertical rectangular ancilla qubit is initialized to the $|+\rangle$ state first and performs logical joint measurements, M_{ZZ} , with the control qubits 1, 2, 3, and 8. It is then split and shrunk to perform logical joint measurements, M_{XX} , with the target qubits 4, 5, 6, and 7. After the multi-target CNOTs, we inject the $|Y\rangle$ state into the neighboring qubits and perform a logical S gate, followed by an X -basis measurement, as shown in Figure 1(b). However, since a vertical rectangular ancilla qubit consisting of up to six patches is required to perform multi-target CNOTs and too much memory is required to organize it, this paper excludes it from the list of possible implementations and only considers logical qubit placement and computation.

Therefore, in Figure 7(MT-CNOT-2), the size of the vertical rectangular ancilla qubit is reduced to four patches to minimize the amount of memory required to construct the vertical rectangular ancilla qubit. To this end, the control qubits 1 and 8 locations are adjusted, as shown in Figure 7(MT-CNOT-2(a)). In this placement, the distillation circuit's first and fourth multi-target CNOTs are performed by decomposing them into single-target CNOTs and multi-target CNOTs, respectively. For the first multi-target CNOT, the ancilla qubit H_8 is first used to perform the CNOT with target qubit 7, and the right vertical rectangular ancilla qubit K is

used to perform the CNOT with target qubits 4 and 5. For the fourth multi-target CNOT, the ancilla qubit H_1 is used to perform the CNOT with target qubit 4, and the left vertical rectangular qubit K is used to perform the CNOT with target qubits 6 and 7.

In Figure 7(MT-CNOT-3), the logical qubit architecture of Figure 6(b) is used to optimize space for ancilla qubits for multi-target CNOT operations and $|Y\rangle$ state injected qubit for logical S gates. In Figure 7(MT-CNOT-3(a)), control qubits 1, 2, 3, and 8 perform CNOT with target qubits 4, 5, 6, and 7 using the left and right vertical rectangular ancilla qubits K . The vertical rectangular ancilla qubit, split to perform logical joint measurement M_{XX} with the target qubits, is used as a space to inject the $|Y\rangle$ state for logical S gates. Logical qubits 5 and 6 move to perform the logical joint measurement, M_{ZZ} , with the neighboring qubit where the $|Y\rangle$ state was injected.

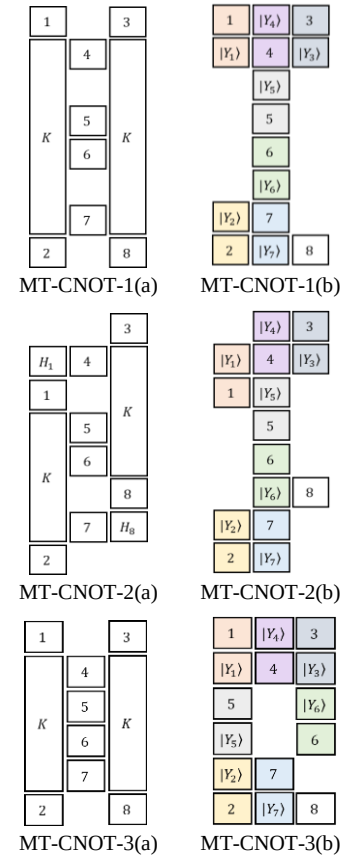


Figure 7. This illustrates how $|Y\rangle$ state distillation circuits are implemented with multi-target CNOTs.

C. Lattice surgery translation-based $|Y\rangle$ state distillation

The lattice surgery translation-based distillation process [5] is implemented in a logical qubit architecture consisting of the 20 logical qubits of Figure 6(c), as illustrated in Figure 8. The lattice surgery translation-based distillation process is performed as follows. Firstly, for each multi-target CNOT in the circuit in Figure 1(b), the vertical rectangular patches are initialized with the $|+\rangle$ state in Figure 8(a). Each vertical rectangular patch is then smoothly split in Figure 8(b). The

numbers in Figure 8(b) indicate which patch contributes to which qubit of the circuit in Figure 1(b). In Figure 8(c), patches with different control qubits and the identical target qubits are moved to neighboring locations, and rough merge operations are performed. And then, the rough-merged patches are reduced to a single patch size. We inject $|Y\rangle$ state into the patches adjacent to the logical qubit patch to perform the logical S gate. We merge smoothly with the $|Y\rangle$ state-injected patch and take X-basis measurements of logical qubits 1 through 7. Therefore, according to the measurement outcomes, logical qubit 8 has the distilled state.

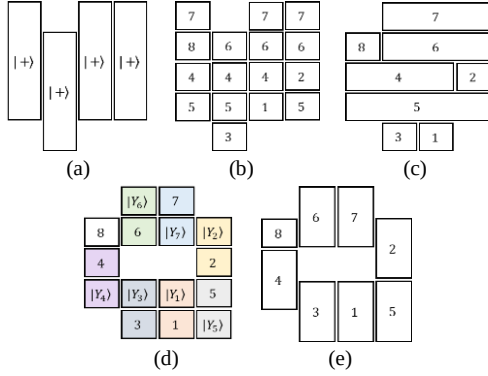


Figure 8. This illustrates how $|Y\rangle$ state distillation circuits are implemented using lattice surgery translation [5].

We have evaluated the space-time cost of the implementation methods for $|Y\rangle$ state distillation circuit in Table 1. Since the surface code patch with distance-2 used in this paper requires d^2 data qubits and $(d^2 - 1)$ ancilla qubits, the total physical qubits are $2d^2$ to the leading order. Thus, the space cost requirements are equal to the number of logical qubits in the logical qubit architecture multiplied by the number of physical qubits for one patch.

The time cost requirements are given by code-cycles. The implementations based on multi-target CNOTs have the following time costs.

- [2d] Initialization of the vertical rectangular ancilla qubits with states $|+\rangle$ for smooth merging and splitting of control qubits and vertical rectangular ancilla qubits, twice each.
- [2d] Smooth merging and splitting for CNOT(1, 4) and CNOT(8, 7) (for MT-CNOT-2 only).
- [2d] Rough merging and splitting for CNOT(1, 4) and CNOT(8, 7) (for MT-CNOT-2 only).
- [4d] Smooth merging and splitting of control qubits and vertical rectangular ancilla qubits.
- [2d] Shrinking the split ancilla qubits.
- [2d] Rough merging and splitting of target qubits and split ancilla qubits.
- [1d] Movement of qubits 5 and 6 (for MT-CNOT-3 only).
- [1d] Injection of $|Y\rangle$ states.
- [1d] Smooth merging with $|Y\rangle$ state injected qubits.

The lattice surgery translation-based implementation has the following time costs [5].

- [1d] Initialization of the vertical rectangular qubits with states $|+\rangle$.
- [1d] Smooth splitting of the vertical rectangular qubits.
- [1d] Movement of qubits 1 and 7.
- [1d] Rough merging of the same target qubits.
- [1d] Shrinking the merged qubits.
- [1d] Injection of $|Y\rangle$ states.
- [1d] Smooth merging with $|Y\rangle$ state injected qubits.

Although the time cost of the lattice surgery translation-based implementation is less than the multi-target CNOT-based one, we can simulate the $|Y\rangle$ state distillation circuit using the multi-target CNOT-based implementation with fewer logical qubits.

TABLE 1. EVALUATION OF THE SPACE-TIME COST OF THE IMPLEMENTATION METHODS FOR $|Y\rangle$ STATE DISTILLATION CIRCUIT

Cost	Implementation methods for $ Y\rangle$ state distillation		
	MT-CNOT-2	MT-CNOT-3	LST
Space	$24 \times 2 \times d^2$	$18 \times 2 \times d^2$	$20 \times 2 \times d^2$
Time	$16d$	$13d$	$7d$
Total	$768d^3$	$468d^3$	$280d^3$

IV. CONCLUSIONS

We have implemented and evaluated the $|Y\rangle$ state distillation circuits using the QPlayer quantum simulator on logical qubit architectures. The logical qubits are encoded in a surface code with distance 2. The logical qubits are arranged to use fewer resources to perform the multi-target CNOTs comprising the $|Y\rangle$ state distillation circuit. We also have implemented and compared how CNOT operations with different controls and the same targets can achieve the same results using lattice surgery-based methods and extend them to multi-target CNOT. The $|Y\rangle$ state distillation circuit requires many logical qubits and, therefore, many physical qubits, which is challenging to implement in practice and complex to verify by simulation. This paper verifies $|Y\rangle$ state distillation using more efficient resources. As a next step, we will evaluate $|A\rangle$ state distillation circuits for logical T gate.

ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00014), A Technology Development of Quantum OS for Fault-tolerant Logical Qubit Computing Environment).

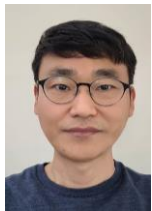
REFERENCES

- [1] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, 86, 2012.
- [2] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, "Surface code quantum computing by lattice surgery," *New Journal of Physics*, 14, 2012.

- [3] D. Litinski, "A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery," *Quantum* 3, 128, 2019.
- [4] D. Litinski and F. V. Oppen, "Lattice Surgery with a Twist: Simplifying Clifford Gates of Surface Codes," *Quantum* 2, 62, 2017.
- [5] D. Herr, F. Nori, and S. Devitt, "Lattice surgery translation for quantum computation," *New Journal of Physics*, 19, 2017.
- [6] S. Bravyi and A. Kitaev, "Universal quantum computation with ideal Clifford gates and noisy ancillas," *Physical Review A*, 71, 2005.
- [7] D. Gottesman, "The Heisenberg Representation of Quantum computers," *arXiv:quant-ph/9807006*, 1998.
- [8] K. S. Jin and G. I. Cha, "QPlayer: Lightweight, scalable, and fast quantum simulator," *ETRI Journal*, 45, (2023), 304–317.
- [9] K. S. Jin and G. I. Cha, "Multilayered logical qubits and synthesized quantum bits," *Quantum Science and Technology*, 8 035008, 2023.
- [10] A. Erhard, H. P. Nautrup, M. Meth, L. Postler, R. Stricker, M. Stadler, V. Negnevitsky, M. Ringbauer, P. Schindler, H. J. Briegel, R. Blatt, N. Friis, and T. Monz, "Entangling logical qubits with lattice surgery," *Nature*, 589, 2021.



Youngchul Kim is currently a principal researcher at ETRI, Daejeon, Republic of Korea since 2000. He received his BS and MS degrees in Computer Science from the Kangwon National University, Chuncheon, Republic of Korea in 1996 and 1999, respectively. His research interests include fault-tolerant quantum computing, distributed systems, and cloud systems.



Soo-Cheol Oh received his BS, MS, and PhD degrees in Computer Engineering in 1995, 1997, and 2003, respectively, from Pusan National University, Pusan, Republic of Korea. From 1997 to 1998, he worked as a research engineer at the LG Multimedia Research Laboratory. Since 2005, he has been working as a principal researcher at the ETRI, Daejeon, Republic of Korea. His current research interests are in quantum computing and cloud systems.



Sangmin Lee received her BS degree in Computer Engineering at the Inha University, Incheon, Republic of Korea in 1991. She has been with the ETRI, Daejeon, Republic of Korea since 1991, where she has worked on developing the SCSI and FC RAID system, distributed parallel file system, dual-mode big data platform, and simulation technology for the digital twin. Currently, she is working as a principal researcher. Her current research interests include distributed systems, extreme storage systems, and quantum operating systems.



Ki-Sung Jin received his BS and MS degrees in Computer Engineering from Jeonbuk National University, Jeonju, Republic of Korea, in 1999 and 2001, respectively. Since 2001, he has been with the ETRI, Daejeon, Republic of Korea, where he has worked on developing the cluster database, distributed parallel filesystem, dual-mode big data platform and simulation technology for the digital twin. He is currently a principal researcher. His current research interests include distributed systems, extreme storage systems, and quantum operating systems.



Gyuil Cha received his BS and MS degrees in Computer Science from Korea University, Seoul, Republic of Korea, in 1998 and 2000, respectively. Since 2000, he has been with the ETRI, Daejeon, Republic of Korea, and is currently a principal researcher. His research interest is a quantum operating system for fault-tolerant quantum computing. He has been involved in the technology development of operating systems, memory virtualization, supercomputing, microservice architectures, and extreme storage systems.