

Quantum Science and Technology



PAPER

Multilayered logical qubits and synthesized quantum bits

OPEN ACCESS

RECEIVED
13 December 2022

REVISED
18 April 2023

ACCEPTED FOR PUBLICATION
20 April 2023

PUBLISHED
3 May 2023

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Ki-Sung Jin*  and Gyu-Il Cha

Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

* Author to whom any correspondence should be addressed.

E-mail: ksjin@etri.re.kr

Keywords: quantum simulator, surface code, quantum error, Hilbert-space reduction

Abstract

Quantum error correction is likely to be key in obtaining near term quantum advantage. We propose a novel method for providing multiple logical qubits in the correction of quantum errors using classical computers. The core idea of our work is built upon two main pillars: dividing the Hilbert space into reduced Hilbert spaces with individual logical qubits and synthesizing the reduced Hilbert spaces through a mathematical collaborating between classical bits and logical quantum states. We demonstrate that our method supports at least 20 logical qubits in a surface code with a code distance of 3. Furthermore, we generate entangled states of multiple logical qubits from lattice surgery-based surface codes using only physical qubit operations. This approach enables classical computers to support a larger number of logical qubits using less memory and perform faster simulations.

1. Introduction

The world is on the verge of an imminent quantum revolution whose signs started rising from the qualitative development of quantum devices and the achievement of quantum advantages [1–8]. However, currently available quantum computers are still lacking fault tolerance for diverse quantum errors. In fact, qubits are unsuitable for large-scale algorithms owing to their very short coherence time and unstable gate fidelity. This has driven extensive researches on fault-tolerant quantum computing, which preserves quantum information for a long time and utilizes frequent gate operations without loss of data [9, 10].

The quantum-error correction (QEC) has shifted from a pure theoretical curiosity to practical possibilities with the first proposition of a QEC algorithm by Shor in 1995 [11]. Shor demonstrated the encoding of quantum information through entanglement in an extended qubit system. In fact, a QEC code could be regarded as a mapping of k qubits (the Hilbert space of dimension 2^k) to n qubits (the Hilbert space of dimension 2^n), where $n > k$. The k qubits are logical qubits to be protected against errors. An additional set of $n-k$ qubits store redundant information to prevent an easy corruption of the logical state. Shor's demonstration of the quantum-error resilience has driven the development of numerous QEC codes [12–16], among which the surface code [17–24] stands as the most promising QEC algorithm so far.

Despite its great potential as a breakthrough in QEC, the surface code is still facing inevitable barriers to its application in quantum computers, owing to its requirement of large-scale physical qubits. For example, 20 logical qubits with a code distance of 3 requires a two-dimensional (2D) planar lattice with 388 physical qubits. Currently, the Google Sycamore processor [25] can provide only one or two logical qubits, whereas the IBM Eagle processor [26] cannot apply the surface code owing to connectivity restrictions. Nonetheless, a successful demonstration of the surface code's application was made in 2018 using 1 logical qubit on the 17 qubits ion-trap quantum computer [27]. This constitutes an early experimental verification of the surface code applied on a quantum computer. Meanwhile, an alternative approach to analyze empirically the behaviors of the surface code relies on the use of classical quantum simulators. Typically, the direct evolution approach [28–32] and the graph-based algorithm [33–39] are being studied in this case. The former is a method based on the direct changes of quantum state vectors in a Hilbert space, while the latter is a tensor network approach from a geometric or topological perspective of tensors. However, these two approaches still reveal limitations in supporting large-scale quantum algorithms, mainly due to the exponential increase

of the physical memory for storing quantum states with the number of qubits or circuit depth [40, 41]. The current computing environment cannot afford more than 100 qubits, even with the use of thousands of supercomputers. Therefore, there is a strong need for a novel approach that experimentally verifies the behaviors of the surface code for quantum algorithms with large-scale logical qubits. In this study, we explored new paths to achieve this verification.

Following a careful analysis of the logical qubits' physical states, we realized the possibility of representing large-scale logical qubits using several reduced Hilbert spaces. This is achieved first by dividing the entire Hilbert space into reduced Hilbert spaces, each containing the fully stabilized quantum states of an assigned individual logical qubit. Meanwhile, the information of other logical qubits is preserved in the form of a special qubit that we refer to as the 'magic qubit.' Second, we empirically validated our idea using a classical quantum simulator and successfully demonstrated the generation of the entangled states of 20 logical qubits, using the lattice surgery of the surface code with a code distance of 3. Our contributions in this study are summarized as follows.

- We proposed a novel algorithm supporting large-scale logical qubits in the surface code using a classical computer. Based on a combined approach of the quantum theory and the processing of classical bits, we guaranteed the preservation and evolution of the logical qubit states in reduced Hilbert spaces. We showed that our proposed algorithm supports the behaviors (e.g. error syndrome measurement (ESM), logical operator, and lattice surgery) of the surface code, using only physical gate operations without intentional distortion of logical states.
- We validated our approach experimentally, using a distance 3 surface code. We successfully simulated 20 logical qubits in a square lattice with 388 physical qubits and generated their entangled states. We used QPlayer, which is a classical quantum simulator, on a single server equipped with 512 GB memory to demonstrate the validity of the proposed algorithm. To the best of our knowledge, this work represents the first successful experimental verification of large-scale logical qubits using classical quantum simulator.
- We provided useful means to study QEC algorithms in large-scale logical qubits, beyond the constraints of theoretical or mathematical approaches. Our proposed algorithm enables an easy access to studying various QEC algorithms or error models using multiple logical qubits. The accessibility of the proposed approach is promoted by the low computing resources required for applying the algorithm. Thus, it can be easily applied to general-purpose classical computers in the laboratory.

The remainder of this paper is organized as follows. In section 2, the quantum state patterns of a logical qubit generated by the surface code are analyzed, and a classical quantum simulator for applying our algorithm is introduced. The theoretical basis for the representation of large-scale logical qubits through reduced Hilbert spaces is presented in section 3. The concept of an algorithm, the topology of reduced Hilbert spaces, and the application methods of logical operations are described. In section 4, we show the experimental results obtained using 20 logical qubits in a single computer. Finally, we present the conclusion in section 5.

2. Preliminaries

2.1. Surface code and quantum bits

The surface code belongs to a class of promising low-cost QEC codes, in which logical qubits are encoded in a 2D lattice of physical qubits. It provides an error-correction mechanism with alternate X-type and Z-type stabilizers in a checkerboard-like pattern throughout the lattice. The shape of the distance 3 surface code is shown in figure 1(a). A total of 17 physical qubits are used for the logical qubit. The white circles represent nine data qubits, and the blue (red) faces indicate four X(Z) ancilla qubits. Data qubits encode the states of the physical qubit into logical information, while ancilla qubits continuously collect syndrome measurements from neighboring data qubits. The surface code discretizes the continuously occurring qubit errors into bit-flip (X) and phase-flip (Z) errors. These flips are detected by Z-type and X-type ancilla qubits, respectively. Figure 1(c) illustrates a quantum circuit for measuring the ancilla qubit, which corresponds to a process known as the ESM. Individual X(Z) stabilizers are shown in figure 1(b), where D_i indicates the number of the data qubit used in the ESM circuit.

In the surface code, the stabilizers encode the quantum states of a logical qubit in a special bit string form, referred to as the codeword. For any stabilizer, the codeword of the logical state $|0\rangle_L$ is obtained by projecting it onto the (+) eigenspace of all stabilizers, as follows:

$$|0\rangle_L = \frac{1}{N} \prod_{S_i \in \{S\}} (I^{\otimes n} + S_i) |0^{\otimes n}\rangle, \quad (1)$$

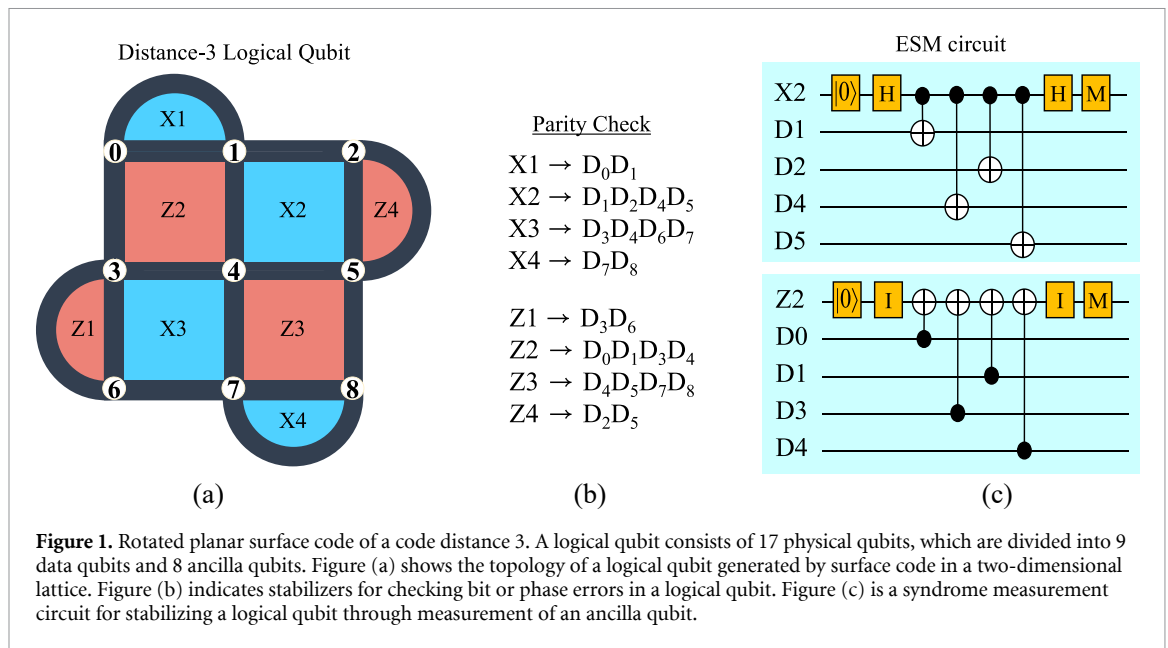
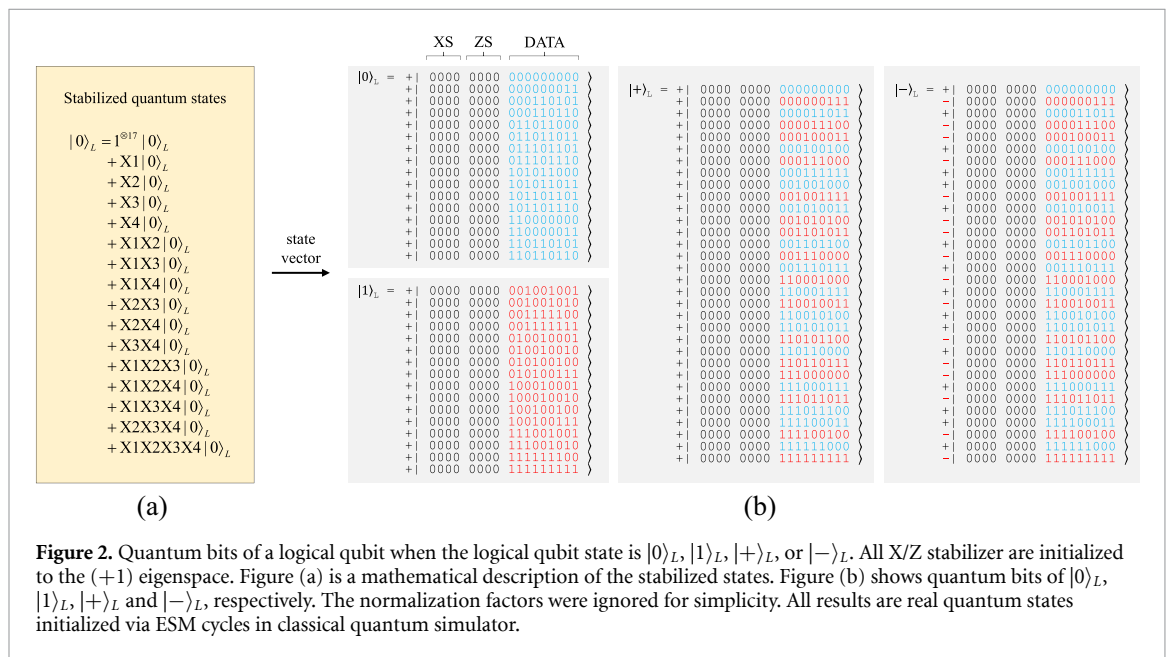
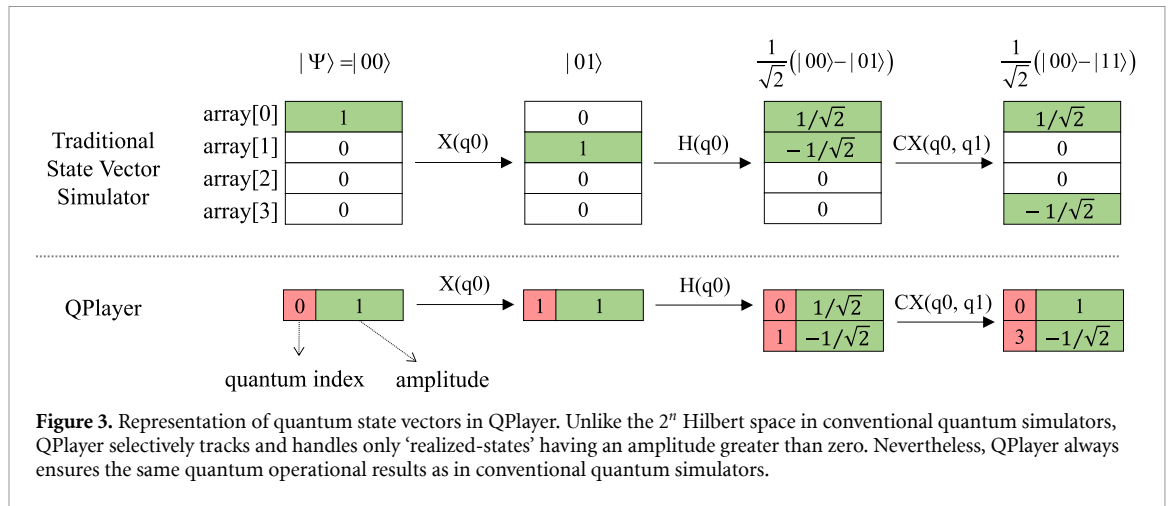


Figure 1. Rotated planar surface code of a code distance 3. A logical qubit consists of 17 physical qubits, which are divided into 9 data qubits and 8 ancilla qubits. Figure (a) shows the topology of a logical qubit generated by surface code in a two-dimensional lattice. Figure (b) indicates stabilizers for checking bit or phase errors in a logical qubit. Figure (c) is a syndrome measurement circuit for stabilizing a logical qubit through measurement of an ancilla qubit.





state vectors of the $|0\rangle_L$, $|1\rangle_L$, $|+\rangle_L$, and $|-\rangle_L$ are shown in figure 2. All the states were observed in this case using the classical quantum simulator. The difference between the order of the qubit number and the topology of the real lattice will be addressed later in the section related to the qubit mapping. Meanwhile, the difference between $|0\rangle_L$ and $|1\rangle_L$ depends on the number of bits 1 in the bit string of the data qubits, i.e. an even and an odd number of 1s for $|0\rangle_L$ and $|1\rangle_L$, respectively. The odd number of bits 1 for the state vector $|1\rangle_L$ is associated with the operation of the logical Pauli-X, which requires bit flips for three physical qubits. Specifically, the $|+\rangle_L$ and $|-\rangle_L$ state vectors have the same bit string, but a different sign pattern. The (+) and (-) signs refer to the $|0\rangle_L$ and $|1\rangle_L$, respectively.

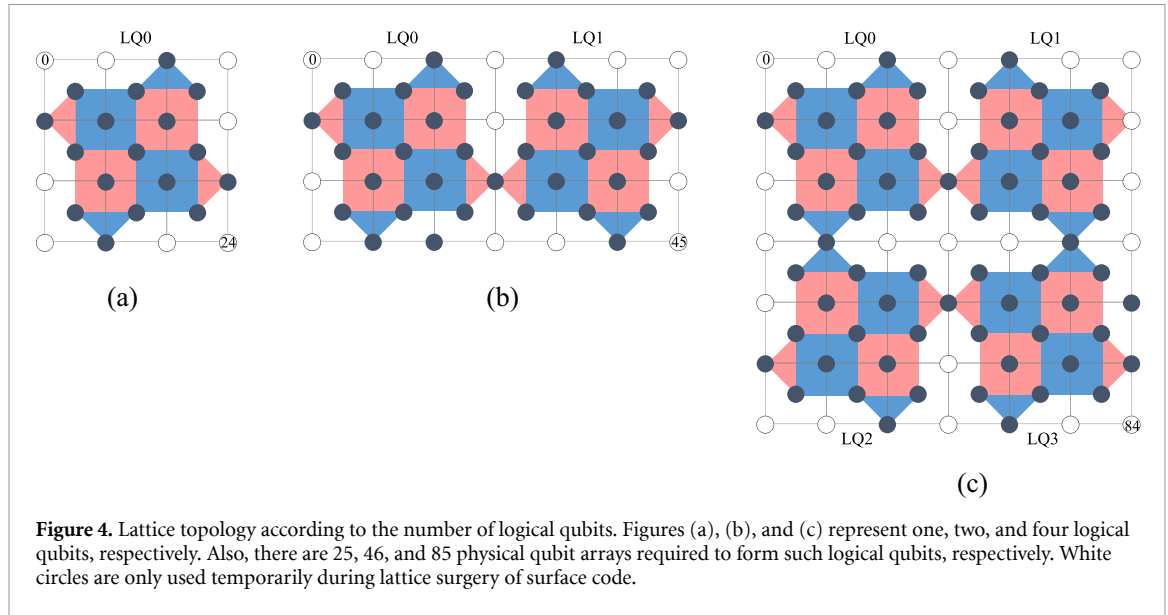
Most importantly, the principle of the stabilization code features a unique preservation of the original stabilized states. Even when temporarily corrupted due to errors, the logical state is always restored to its initial state pattern via the ESM and error-correction cycles. This implies that the states of the logical qubit exist only in a fixed stabilized space, not in the entire Hilbert space. Consequently, the state of the logical qubit could be represented in a reduced Hilbert space of 2^4 (for $|0\rangle_L$, $|1\rangle_L$) or 2^5 (for $|+\rangle_L$, $|-\rangle_L$), instead of 2^{17} . Since the amplitudes of the remaining physical states in the Hilbert space are always zero, they have no physical effect on the logical qubit. It follows that, if the states of the logical qubits are represented only with a reduced Hilbert space, a larger number of logical qubits will become possible even with relatively small computing resources.

2.2. QPlayer: a specialized classical quantum simulator

QPlayer [43, 44] is a new type of state vector quantum simulator developed to overcome the exponential explosion of memory space. QPlayer classifies an individual quantum state either as a 'realized-state' or an 'unrealized-state.' The former refers to a quantum state with an amplitude greater than zero, whereas the latter refers to a state with a zero amplitude. QPlayer stores only the 'realized-state' in computer memory, and manages the sequence of state vectors for their quantum computation. As a result, QPlayer simulates more qubits with a less memory and supports faster operations than ever before.

Figure 3 shows an example of a quantum state's evolution in QPlayer. For the sake of clarity, we assume the case of only two qubits. In traditional quantum simulators, 2^N quantum states are stored in memory, while 2^{N-1} matrix calculations for every gate operation are running, where N is the number of physical qubits. Therefore, as the number of qubits increases, an exponential increase in the memory consumption and computational cost is inevitable. In contrast, a much smaller amount of available memory is required when using QPlayer as only the 'realized-states' are stored. Moreover, the position of the state vector is provided in the form of a 'quantum index' to identify the state pairs from the reduced Hilbert space for the calculation with the quantum matrix. Accordingly, the number of matrix operations is reduced ($<2^{N-1}$). Despite using a reduced Hilbert space, QPlayer yields the same quantum operation results as conventional quantum simulators. [43] shows that QPlayer exhibits 9 times faster ESM and 11 773 times faster teleport operations between two logical qubits than does the QuEST simulator [32], which is a representative quantum simulator.

Figures 4(a)–(c) illustrate the topologies of lattices supporting one, two, and four logical qubits, respectively. The required number of physical qubits is 25, 46, and 85, respectively. The white circles represent qubits that temporarily operate during the lattice surgery of the surface code. These qubits remain unused in



normal state. Owing to the use of a reduced Hilbert space and the resulting low memory consumption, QPlayer demonstrated surface code behaviors using five logical qubits in a single computing server. However, despite its superiority relative to the traditional quantum simulators (i.e. two logical qubits max), QPlayer still cannot surpass the limit of five logical qubits. This limitation is attributed to the size of the quantum memory space, which increases in a multiple of 2^5 proportionally to the number of logical qubits as

$$\# \text{of physical states} = \prod_{i=1}^{\text{NLQ}} \text{NPS}_{\text{LQ}_i}, \quad (2)$$

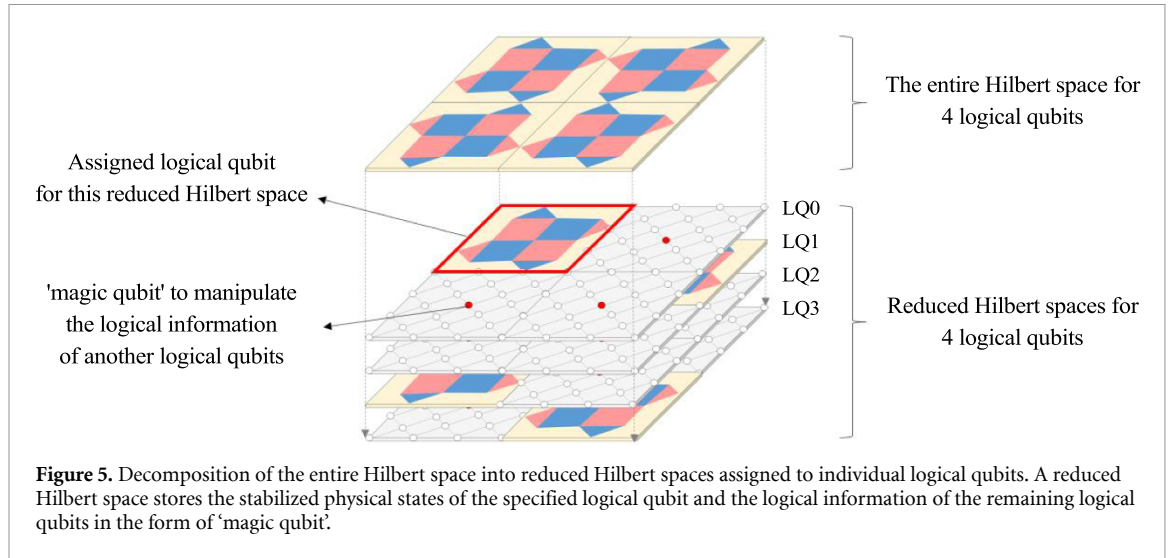
where NLQ and NPS_{LQ_i} refer to the number of the logical qubits and the number of physical states in LQ_i , respectively. Equation (2) helps in estimating the number of physical states generated as the number of LQs increases. Since individual LQs have 2^4 (for $|0\rangle_L, |1\rangle_L$) or 2^5 (for $|+\rangle_L, |-\rangle_L$) physical states, their tensor product yields the total number of physical states, as in equation (2). Therefore, if all LQs are superposed (e.g. $|+\rangle_L, |-\rangle_L$), the total number of physical states reaches $2^{5 \times \text{NLQ}}$.

3. Methods

3.1. Multi-layered Hilbert spaces

Dividing the entire Hilbert space into multiple reduced spaces containing individual LQs, as shown in figure 5, forms the key concept of our proposed approach. Considering an example of 4 LQs in a square lattice with 85 physical qubits (figure 4(c)), the same number of reduced Hilbert spaces is formed in a layered structure. In the frame of our proposed approach, the logical information always evolves equally in all reduced Hilbert spaces, even though the layers are mutually isolated. To achieve this, we associated two types of quantum information with each reduced Hilbert space, as follows. The first type consists of the entire physical states of the LQ assigned to the reduced Hilbert space. These states represent the $|0\rangle_L, |1\rangle_L, |+\rangle_L$, or $|-\rangle_L$ generated by the distance 3 surface code. Therefore, the number of physical states for each LQ in one reduced Hilbert space is 2^4 or 2^5 , with patterns similar to those shown in figure 2. The second type corresponds to the logical information of other LQs within the same reduced Hilbert space, which is preserved using the magic qubit (MQ). The MQ is defined by any physical qubit within the corresponding LQ, as long as it does not overlap with the neighboring LQ region. For convenience, we select the MQ as the physical qubit located in the middle of the logical topology. The number of MQ in each reduced Hilbert space is equal to $\text{NLQ}-1$, where NLQ is the number of LQ. Since the MQ has only up to two physical states, it is expected that the number of physical states in a reduced Hilbert space will decrease, as described by the following equations

$$|\Psi\rangle_{\text{Logical}} = |\text{LQ}^3\rangle \otimes |\text{LQ}^2\rangle \otimes |\text{LQ}^1\rangle \otimes |\text{LQ}^0\rangle \quad (3)$$

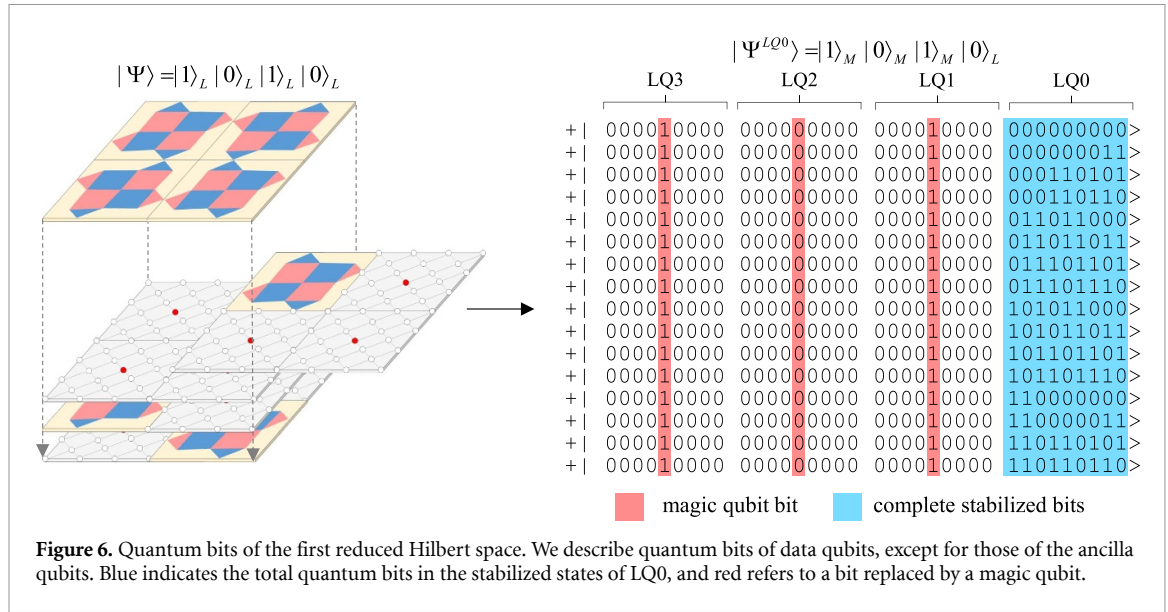


$$\begin{aligned}
 |\Psi^{LQ^0}\rangle_{\text{Logical}} &= |\text{MQ}^3\rangle \otimes |\text{MQ}^2\rangle \otimes |\text{MQ}^1\rangle \otimes |\text{LQ}^0\rangle \\
 |\Psi^{LQ^1}\rangle_{\text{Logical}} &= |\text{MQ}^3\rangle \otimes |\text{MQ}^2\rangle \otimes |\text{LQ}^1\rangle \otimes |\text{MQ}^0\rangle \\
 |\Psi^{LQ^2}\rangle_{\text{Logical}} &= |\text{MQ}^3\rangle \otimes |\text{LQ}^2\rangle \otimes |\text{MQ}^1\rangle \otimes |\text{MQ}^0\rangle \\
 |\Psi^{LQ^3}\rangle_{\text{Logical}} &= |\text{LQ}^3\rangle \otimes |\text{MQ}^2\rangle \otimes |\text{MQ}^1\rangle \otimes |\text{MQ}^0\rangle.
 \end{aligned} \tag{4}$$

Equation (3) defines the logical quantum state $|\Psi\rangle_{\text{Logical}}$ in the original Hilbert space consisting of four LQs and represented by a linear combination of four logical states. Since each LQ can assume two logical states (i.e. $|0\rangle_L$ or $|1\rangle_L$), $|\Psi\rangle_{\text{Logical}}$ can generate 2^4 logical states (i.e. from $|0000\rangle_L$ to $|1111\rangle_L$). Nonetheless, the logical state is a mere abstraction of the physical states. Since each stabilized [LQ] can have a linear combination of 2^5 physical states, the actual number of physical quantum states $|\Psi\rangle_{\text{Physical}}$ becomes 2^{20} ($2^{5 \times \text{NLQ}}$). In contrast, equation (4) indicates another form for the logical states reconstructed in a reduced Hilbert space, in which the original quantum space $|\Psi\rangle_{\text{Logical}}$ extends to four reduced Hilbert spaces $|\Psi^{LQ^i}\rangle_{\text{Logical}}$. Here, $|\Psi^{LQ^i}\rangle_{\text{Logical}}$ denotes a reduced Hilbert space generated for the i th LQ. We formalize the assigned logical qubit and the remaining logical qubits as [LQ] and [MQ], as shown in equation (4). The interchanging of [LQ] with [MQ] leads to a reduced number of physical states in the reduced Hilbert space because [LQ] is a linear combination of 2^5 physical states, whereas [MQ] is a linear combination of 2^1 physical states.

Although the combinations of physical states for $|\Psi\rangle_{\text{Logical}}$ and $|\Psi^{LQ^i}\rangle_{\text{Logical}}$ are different, they all preserve the same logical information in the frame of our proposed approach. This is supported by equations (5) and (6), showing that the logical information in the separate reduced Hilbert spaces is equivalent. This could be explained by assuming that the four LQs are entangled on a Bell basis. The first line in equation (5) indicates clearly that the original Hilbert space is composed by entangled logical states of $|0000\rangle_L + |1111\rangle_L$ using the complete physical states. However, the remaining lines in equation (5) describe the four reduced Hilbert spaces showing different patterns of physical states, which is due the replacement of some LQs with MQs. Nevertheless, their logical information guarantees the same entangled logical states $|0000\rangle_L + |1111\rangle_L$, as the original Hilbert space

$$\begin{aligned}
 |\Psi\rangle_{\text{Physical}} &= \sum S_i |\text{LQ}^3\rangle \otimes \sum S_j |\text{LQ}^2\rangle \otimes \sum S_k |\text{LQ}^1\rangle \otimes \sum S_l |\text{LQ}^0\rangle \\
 &\quad + \sum S_j |\text{LQ}^3\rangle \otimes \sum S_i |\text{LQ}^2\rangle \otimes \sum S_k |\text{LQ}^1\rangle \otimes \sum S_l |\text{LQ}^3\rangle \\
 |\Psi^{LQ^0}\rangle_{\text{Physical}} &= |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes \sum S_i |\text{LQ}^0\rangle + |1\rangle \otimes |1\rangle \otimes |1\rangle \otimes \sum S_j |\text{LQ}^0\rangle \\
 |\Psi^{LQ^1}\rangle_{\text{Physical}} &= |0\rangle \otimes |0\rangle \otimes \sum S_i |\text{LQ}^1\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle \otimes \sum S_j |\text{LQ}^1\rangle \otimes |1\rangle \\
 |\Psi^{LQ^2}\rangle_{\text{Physical}} &= |0\rangle \otimes \sum S_i |\text{LQ}^2\rangle \otimes |0\rangle \otimes |0\rangle + |1\rangle \otimes \sum S_j |\text{LQ}^2\rangle \otimes |1\rangle \otimes |1\rangle \\
 |\Psi^{LQ^3}\rangle_{\text{Physical}} &= \sum S_i |\text{LQ}^3\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle + \sum S_j |\text{LQ}^3\rangle \otimes |1\rangle \otimes |1\rangle \otimes |1\rangle
 \end{aligned} \tag{5}$$



$$\begin{aligned}
 |\Psi\rangle_{\text{Logical}} &= |0\rangle_L |0\rangle_L |0\rangle_L |0\rangle_L + |1\rangle_L |1\rangle_L |1\rangle_L |1\rangle_L \\
 |\Psi^{LQ0}\rangle_{\text{Logical}} &= |0\rangle_M |0\rangle_M |0\rangle_M |0\rangle_L + |1\rangle_M |1\rangle_M |1\rangle_M |1\rangle_L \\
 |\Psi^{LQ1}\rangle_{\text{Logical}} &= |0\rangle_M |0\rangle_M |0\rangle_L |0\rangle_M + |1\rangle_M |1\rangle_M |1\rangle_L |1\rangle_M \\
 |\Psi^{LQ2}\rangle_{\text{Logical}} &= |0\rangle_M |0\rangle_L |0\rangle_M |0\rangle_M + |1\rangle_M |1\rangle_L |1\rangle_M |1\rangle_M \\
 |\Psi^{LQ3}\rangle_{\text{Logical}} &= |0\rangle_L |0\rangle_M |0\rangle_M |0\rangle_M + |1\rangle_L |1\rangle_M |1\rangle_M |1\rangle_M.
 \end{aligned} \tag{6}$$

3.2. Quantum bits in a reduced Hilbert space

Figure 6 illustrates the process of representing the physical states of $|\Psi\rangle = |1\rangle_L |0\rangle_L |1\rangle_L |0\rangle_L$ in a reduced Hilbert space. For demonstration purposes, we consider the first reduced Hilbert space as an example. The quantum bits of all the ancilla (X/Z) qubits are zero since they all have (+1) eigenvalues. Therefore, we specify only the quantum bits of nine data qubits, excluding the bits of the ancilla qubits. The blue color highlight indicates 2^4 (i.e. 16) stabilized states of LQ0. Since the logical state of LQ0 is $|0\rangle_L$, all bit strings have an even number of bit 1. If LQ0 is in the $|+\rangle_L$ state, it would have 2^5 (i.e. 32) states. These state patterns correspond to those described previously in figure 2. The remaining LQs show logical information through magic qubits. They are not subjected to an ESM, but rather to physical gate operations of magic qubits changing their quantum states. Accordingly, $|MQ\rangle$ have only physical states of $|000000000\rangle$ or $|000010000\rangle$. Thus, while $|\Psi\rangle$ requires $2^{4 \times 4}$ physical states to represent the logical information $|1\rangle_L |0\rangle_L |1\rangle_L |0\rangle_L$, the reduced Hilbert space of $|\Psi^{LQ0}\rangle$ uses only 2^4 ($2^0 \times 2^0 \times 2^0 \times 2^4$) physical states of $|1\rangle_M |0\rangle_M |1\rangle_M |0\rangle_L$ to represent the same logical state.

3.3. Synthesizing reduced Hilbert spaces by replacing quantum bits

The reduced Hilbert space is a closed system, mutually isolated from other reduced Hilbert spaces. Since single qubit logical operations have no interaction between logical qubits, the quantum state of a logical qubit evolves independently while ensuring locality within the reduced Hilbert space. In other words, logical operations such as the Pauli operator or the Hadamard can be applied independently using either $|LQ\rangle$ or $|MQ\rangle$ within each reduced Hilbert space. However, multiqubit logical operations such as the CNOT gate involves the interaction of many reduced Hilbert spaces. To support such multiqubit logical operations, we propose a synthesis method of reduced Hilbert spaces. From a quantum mechanical perspective, the no-cloning theorem fundamentally forbids the copying of quantum states between qubits. In addition, it is also clear that the entangled state of qubits cannot be decomposed into an individual tensor product. This conceptual strangeness challenges the description of quantum states in classical computers. In this study, we propose an alternative approach to represent quantum states in classical computers using the replacement of classical bits while fully preserving the logical information of the original states.

Figure 7 describes our proposed idea of building a reduced Hilbert space by synthesizing two different reduced Hilbert spaces. We selected the reduced Hilbert spaces of LQ0 ($|\Psi^{LQ0}\rangle$) and LQ1 ($|\Psi^{LQ1}\rangle$) in figure 6

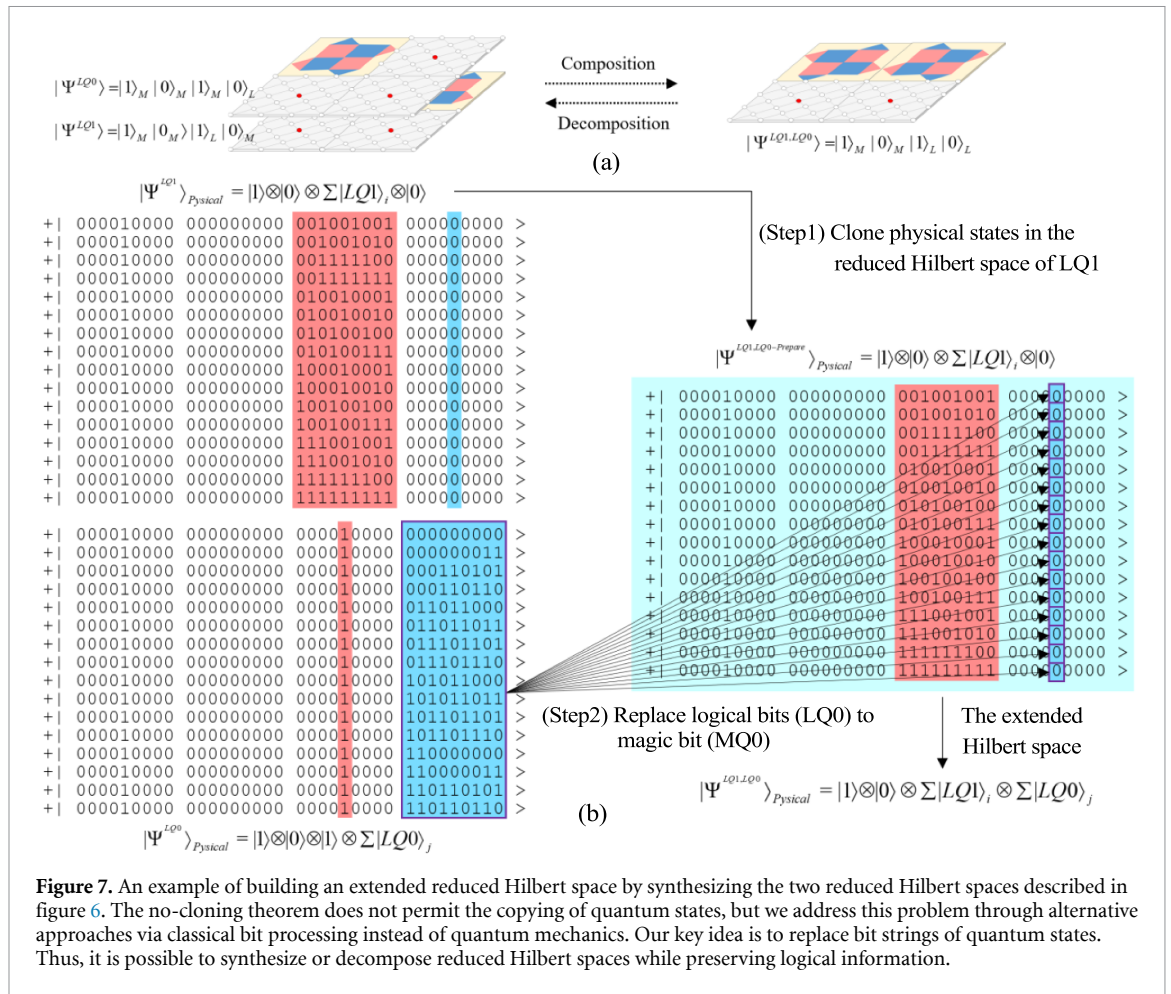


Figure 7. An example of building an extended reduced Hilbert space by synthesizing the two reduced Hilbert spaces described in figure 6. The no-cloning theorem does not permit the copying of quantum states, but we address this problem through alternative approaches via classical bit processing instead of quantum mechanics. Our key idea is to replace bit strings of quantum states. Thus, it is possible to synthesize or decompose reduced Hilbert spaces while preserving logical information.

as an example to illustrate the process. Figure 7(a) presents the concept of synthesizing two reduced Hilbert spaces while ensuring the same logical information. The individual logical states of $|\Psi^{LQ^0}\rangle$ and $|\Psi^{LQ^1}\rangle$ are $|1\rangle_M |0\rangle_M |1\rangle_L |0\rangle_L$ and $|1\rangle_M |0\rangle_M |1\rangle_L |0\rangle_M$, respectively, whereas the synthesized reduced Hilbert space ($|\Psi^{LQ^1, LQ^0}\rangle$) has the logical states of $|1\rangle_M |0\rangle_M |1\rangle_L |0\rangle_L$. It is worth noting that the magic qubit component ($|MQ\rangle$) is replaced with the logical qubit component ($|LQ\rangle$). Since $|LQ\rangle$ and $|MQ\rangle$ can preserve the same logical information, then the logical states of $|\Psi^{LQ^0}\rangle$, $|\Psi^{LQ^1}\rangle$, and $|\Psi^{LQ^1, LQ^0}\rangle$ are all equivalent. Figure 7(b) describes these concepts in terms of the physical states of the reduced Hilbert space of $|\Psi^{LQ^0}\rangle$ and $|\Psi^{LQ^1}\rangle$ (i.e. left side) and the physical states of the synthesized reduced Hilbert space (i.e. right side).

The first step in the synthesis process is to prepare a new reduced Hilbert space ($|\Psi^{LQ^1, LQ^0-Prepare}\rangle$), which is possible in three different ways: (1) initialize all logical qubits to $|0\rangle_L$, (2) initialize the logical information of all logical qubits to magic qubits, and (3) replicate one of the associated reduced Hilbert spaces. However, since (1) and (2) require additional procedures for embedding logical information to $|\Psi^{LQ^1, LQ^0-Prepare}\rangle$, we choose (3) to maintain a high computing efficiency. The replication of $|\Psi^{LQ^1}\rangle$ is completed by copying state vectors stored in the computer’s memory. Although quantum mechanics does not allow the copying of quantum states, classical computing enables building the same Hilbert space through memory copying of state vectors without deformation of quantum information. As a result, the physical states of $|\Psi^{LQ^1}\rangle$ and $|\Psi^{LQ^1, LQ^0-Prepare}\rangle$ after the first step are exactly the same.

In the second step, the $|MQ0\rangle$ in $|\Psi^{LQ^1, LQ^0-Prepare}\rangle$ is substituted with the logical information of $|LQ0\rangle$ in $|\Psi^{LQ^0}\rangle$. Here, we apply the bit operation technique of classical computing for such quantum information transfer. That is, the ‘000000000’ bit of $|MQ0\rangle$ in $|\Psi^{LQ^1, LQ^0-Prepare}\rangle$ is replaced by individual bits of $|0\rangle_L$ in $|\Psi^{LQ^0}\rangle$. This is done by performing a bitwise-OR operation of the masking bit of $|MQ0\rangle$ and each physical bits of $|0\rangle_L$. This bit operation results in a $|MQ0\rangle$ bit extending by the number of 16 quantum bits of $|LQ0\rangle$. Thus, it can be seen that the finally synthesized $|\Psi^{LQ^1, LQ^0}\rangle$ has 16^2 quantum states, indicating that it represents the entire physical states of both $|LQ0\rangle$ and $|LQ1\rangle$.

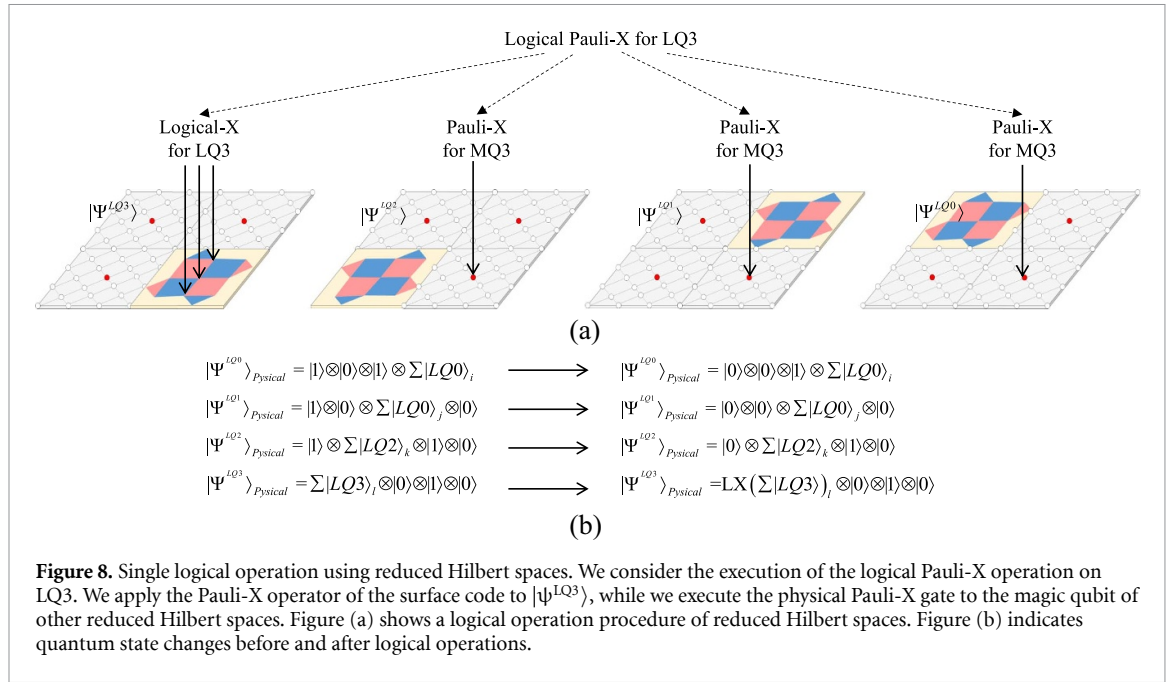


Figure 8. Single logical operation using reduced Hilbert spaces. We consider the execution of the logical Pauli-X operation on LQ3. We apply the Pauli-X operator of the surface code to $|\psi^{LQ3}\rangle$, while we execute the physical Pauli-X gate to the magic qubit of other reduced Hilbert spaces. Figure (a) shows a logical operation procedure of reduced Hilbert spaces. Figure (b) indicates quantum state changes before and after logical operations.

Meanwhile, decomposing the synthesized reduced Hilbert space back into the original reduced Hilbert spaces can be achieved similarly through quantum bit replacement.

3.4. Logical quantum operation

The surface code defines a set of fundamental logical operations for fault-tolerant universal quantum computations, including logical-X, Z, H, S, and CNOT as Clifford gates and logical-T as a non-Clifford gate. To support these logical operations, ensuring that all the reduced Hilbert spaces preserve the same logical information is an important supporting factor of our approach. Its importance stems from the need to guarantee that a synthesized reduced Hilbert space has all the physical states of the associated LQs when performing lattice surgery of the surface code. To this end, it is necessary to revise logical operations of the surface code, so that it can operate in the reduced Hilbert spaces.

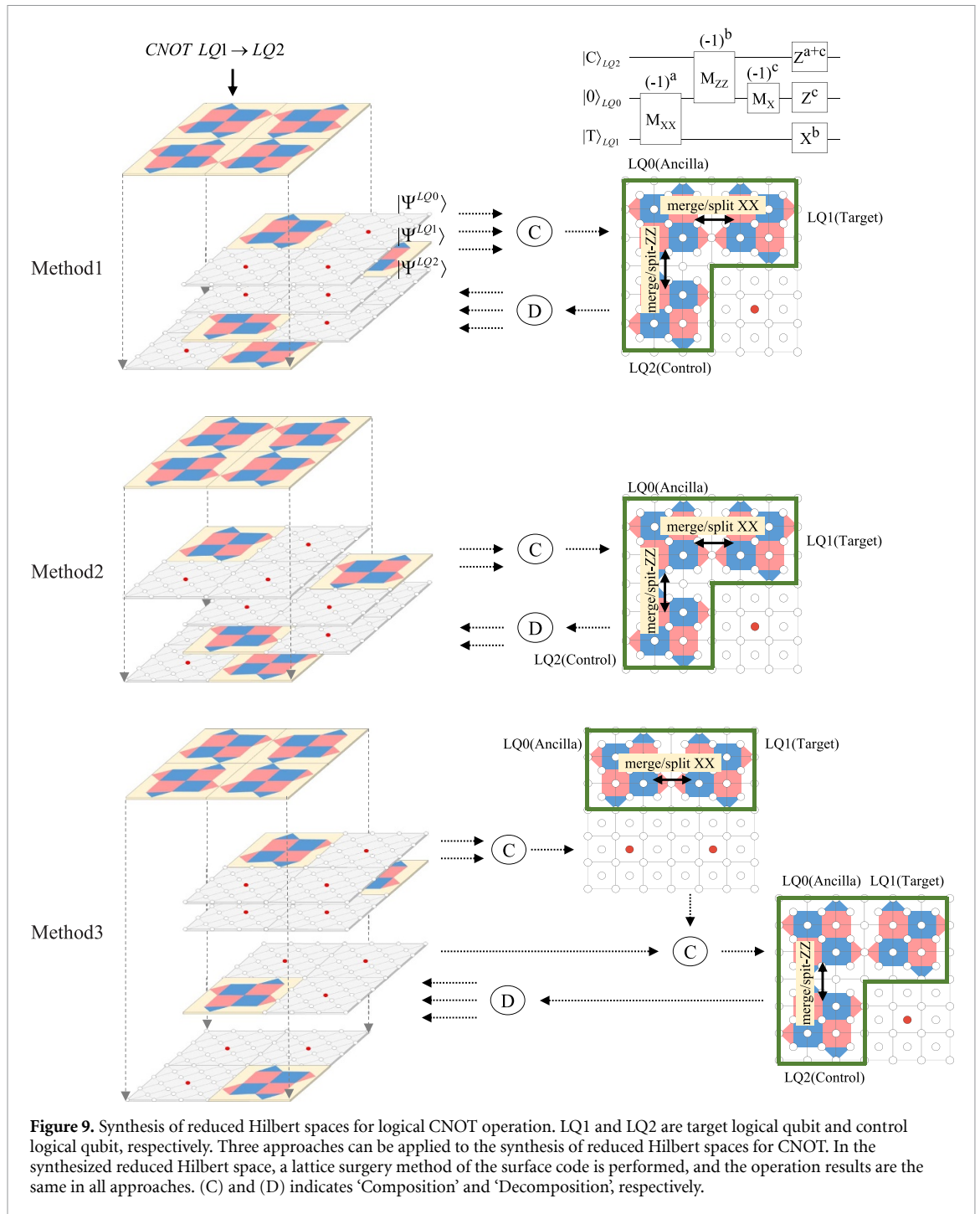
3.4.1. Single qubit operation

Figure 8 shows a single logical operation in reduced Hilbert spaces. Such logical operations include Pauli-X, Pauli-Z, Hadamard, and ESM. Figure 8(a) shows an example of a logical Pauli-X operation on LQ3. Basically, all logical operations are applied to all reduced Hilbert spaces. Since $|\Psi^{LQ3}\rangle$ has the entire stabilized states of LQ3, we apply the logical operator defined in the surface code to $|\Psi^{LQ3}\rangle$. However, in other reduced Hilbert spaces ($|\Psi^{LQ2}\rangle$, $|\Psi^{LQ1}\rangle$, and $|\Psi^{LQ0}\rangle$), we run a physical Pauli-X gate to MQ3, which represents a magic qubit for LQ3. The resulting changes are shown in figure 8(b). It can be seen that the logical state of LQ3 in all reduced Hilbert spaces changed from $|1\rangle_L$ to $|0\rangle_L$.

3.4.2. CNOT

In the surface code, the CNOT operation is performed by lattice surgery which is a key principle that transfers logical information or causes entanglement through merging and splitting of LQs. In figure 9, the upper right circuit shows the implementation of CNOT between two logical qubits by using lattice surgery [45]. This requires two logical qubits (Control and Target) and one additional logical qubit (Ancilla). All processes are achieved through joint measurement of Control and Target using Ancilla as an intermediary. The joint-measurement has the effect of quantum information transfer between Control and Target, and the logical CNOT can be completed through post-corrections based on the measurement outcomes. Mathematically explaining the effect of logical CNOT, it can be seen that it changes to $X \otimes I \rightarrow X \otimes X$ and $I \otimes Z \rightarrow Z \otimes Z$, respectively. The former indicates that the logical bit of Control is transferred to the Target, and the latter means that the logical phase of the Target is transferred to the Control.

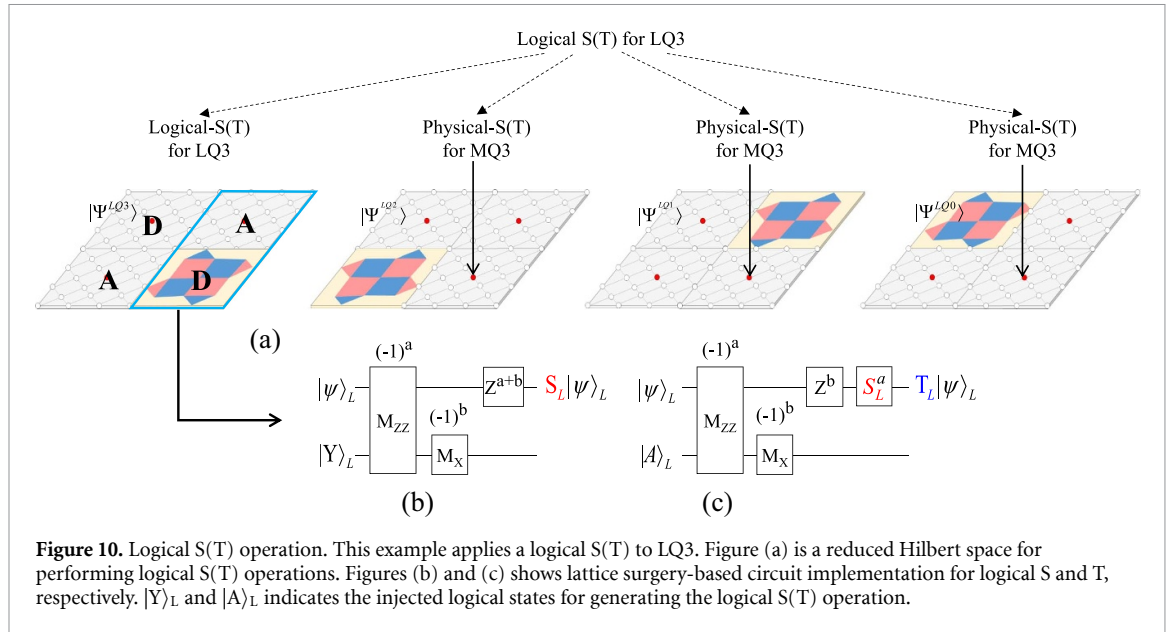
The logical CNOT in the surface code is originally designed to be performed in a single Hilbert space where logical information is encoded by all physical qubits. However, since the reduced Hilbert spaces are mutually isolated, our idea needs to generate one Hilbert space by synthesizing related LQs for lattice surgery



(see appendix A). For this, we propose the following three steps: (1) preparing a single reduced Hilbert space by synthesizing the associated reduced Hilbert spaces, (2) performing a logical CNOT operation by lattice surgery in the synthesized reduced Hilbert space, (3) restoring the logical information after lattice surgery from the synthesized reduced Hilbert space to the original reduced Hilbert spaces. That is, this process can be summarized as a series of synthesis-CNOT-decomposition steps. After all the processes are completed, the temporarily synthesized Hilbert space is removed from the computer memory.

The synthesis of reduced Hilbert spaces for the CNOT operation is possible using three different methods. Since they all provide the same operational results, one can selectively utilize them according to the software efficiency. We consider an example where LQ0, LQ1, and LQ2 correspond to an ancilla logical qubit, a target logical qubit, and a control logical qubit, respectively.

Method 1 is based on the synthesis of three reduced Hilbert spaces at a time. Using the bit operation described above, the logical information of LQ0 (in $|\Psi^{LQ0}\rangle$), LQ1 (in $|\Psi^{LQ1}\rangle$), and LQ2 (in $|\Psi^{LQ2}\rangle$) are transmitted to the synthesized reduced Hilbert space, which has the entire physical states of three logical



qubits. That is, since each of the LQs has 2^4 or 2^5 quantum states, the synthesized quantum states may be extended to $2^{5 \times 3}$. Then, the lattice surgery of the surface code is performed in the synthesized reduced Hilbert space, only through the physical qubit operation. When the lattice surgery is completed, the updated logical information is restored to the original reduced Hilbert space. Even if LQ1 and LQ2 are entangled by lattice surgery, the restored original reduced Hilbert space forms entanglement relation by $|LQ\rangle$ and $|MQ\rangle$.

Method 2 consists of two steps, as follows. In the first step, the target logical qubit (LQ1) and the ancilla logical qubit (LQ0) are synthesized to perform a merge/split XX. Then, the control logical qubit (LQ2) is synthesized into the reduced Hilbert space, and a merge/split ZZ is performed. Similar to Method 1, when the lattice surgery is completed, the logical information is restored to the original reduced Hilbert spaces.

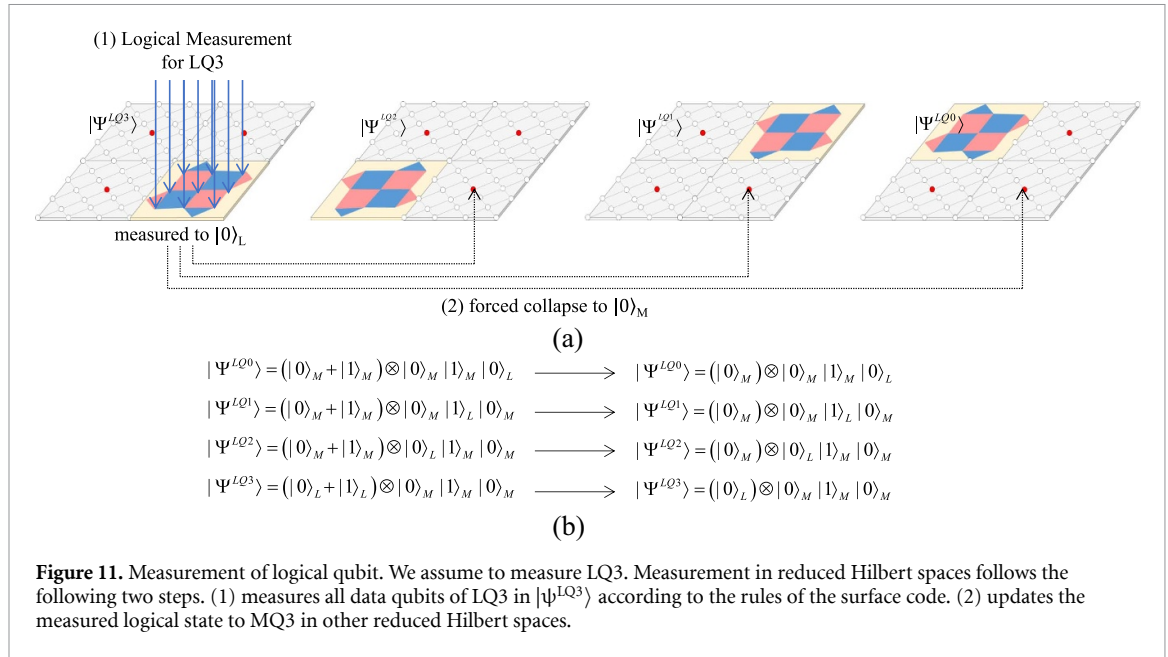
Method 3 is based on synthesizing the control logical qubit (LQ2) and the target logical qubit (LQ1) while excluding the ancilla logical qubit. Interestingly in this case, the logical information is not distorted during the lattice surgery. This is mainly because the ancilla logical 0020 qubit is always initialized by an ESM during lattice surgery. Thus, Method 3 is the most efficient in terms of computational cost. The process after synthesis is similar to Methods 1 and 2.

3.4.3. S, T

In quantum computing, the S and T are single qubit operators that rotate the qubit's state by $\pi/2$ and $\pi/4$ radians, respectively, around the z-axis of the Bloch sphere. However, it should be noted that the logical S and T in surface code cannot be applied independently in a single logical qubit system, rather requires lattice surgery with an additional ancilla logical qubit. Figure 10 shows how to implement the logical S and T in a reduced Hilbert space. For explanation, we assume that the logical S(T) is applied to LQ3. Similar to the description of the previous sections, we perform a physical S(T) gate for MQs in the reduced Hilbert spaces ($|\Psi^{LQ0}\rangle$, $|\Psi^{LQ1}\rangle$, and $|\Psi^{LQ2}\rangle$), where LQ3 is not assigned. However, $|\Psi^{LQ3}\rangle$, which includes the full stabilized states of LQ3, requires lattice surgery between two logical qubits. To this end, we utilize a logical S(T) with the nearby logical ancilla qubit (logical data and ancilla qubit placement can be found in appendix A). At this time, since the logical ancilla qubit operate only temporarily during the lattice surgery, their state changes do not affect the quantum information of other logical data qubits. The logical S(T) operators can be implemented by consuming the magic states of $|Y\rangle_L$ and $|A\rangle_L$, which can be generated using a state injection scheme [24]. $|Y\rangle_L$ and $|A\rangle_L$ indicates $1/\sqrt{2}(|0\rangle_L + |1\rangle_L)$ and $1/\sqrt{2}(|0\rangle_L + e^{i\pi/4}|1\rangle_L)$, respectively. Figures 10(b) and (c) refer to circuits that generate logical S and T operators, respectively. Several methods for generating logical S and T are provided, but we apply an approach using X-boundary joint measurement of two logical qubits [18, 46, 47].

3.4.4. Measurement

The measurement in a reduced Hilbert space is shown in figure 11. It is described through the example of measuring the logical qubit LQ3 in the logical state $(|0\rangle_L + |1\rangle_L) \otimes |0\rangle_L |1\rangle_L |0\rangle_L$. As shown in the figure, $|\Psi^{LQ3}\rangle$ represents the superposed logical state $(|0\rangle_L + |1\rangle_L)$ having all physical states of LQ3, while the remaining reduced Hilbert spaces represent the states $(|0\rangle_M + |1\rangle_M)$ through the magic qubit. The



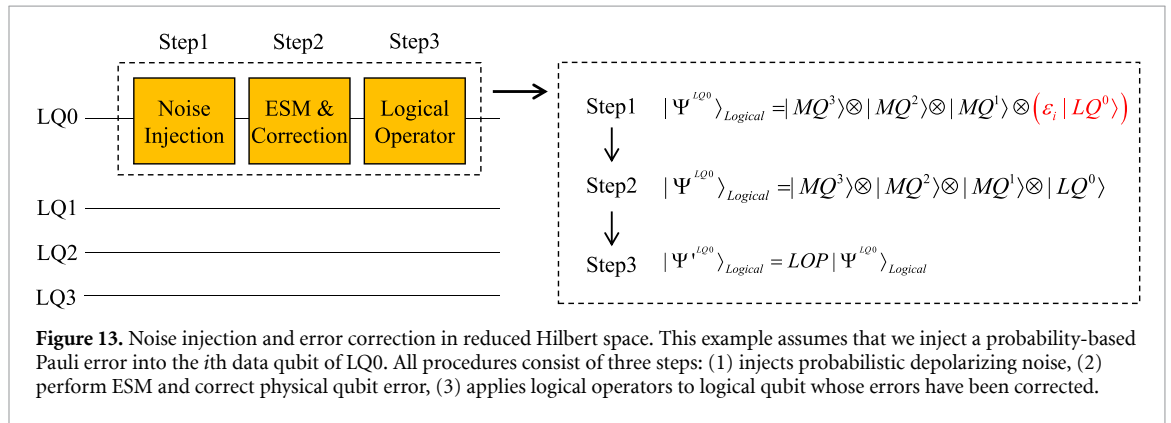
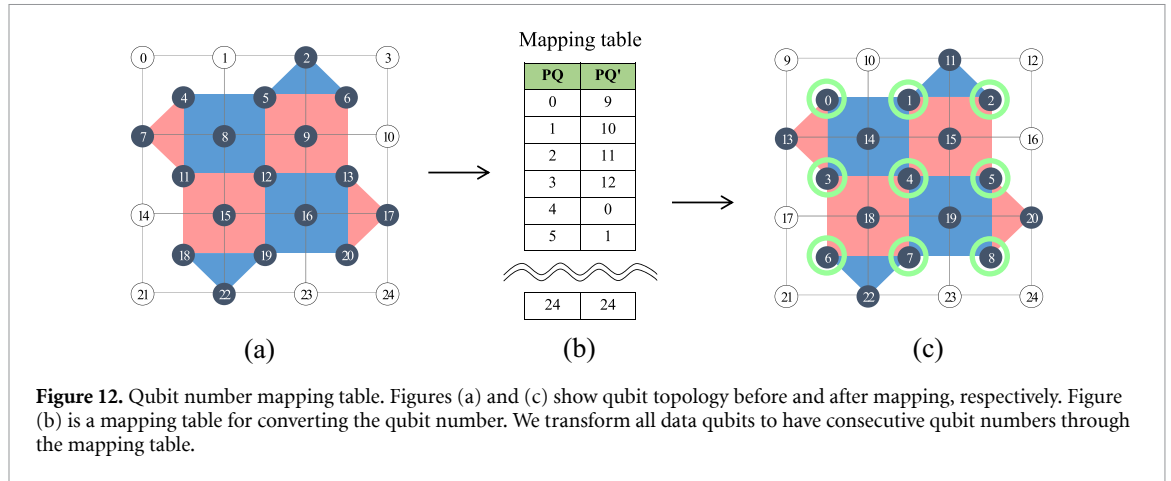
measurement is performed in two steps, as shown in figure 11(a). In the first step, LQ3 is measured following the measurement policy of the surface code, which is done at $|\Psi^{LQ^3}\rangle$ where LQ3 is assigned. After measuring nine data qubits of LQ3 according to the probability amplitude, the logical measured state is determined based on whether the number of measured (+1) eigenvalue is odd or even. In the example considered here, we assume that LQ3 is measured in the logical state of $|0\rangle_L$. The second step is based on updating the magic qubit state for LQ3 in other reduced Hilbert spaces according to the measured logical state. For this, we forcibly collapse each magic qubit into the $|0\rangle_M$ state. Although it is quantum mechanically impossible, we can collapse a qubit to a certain state by adjusting the probability amplitude of the qubit in classical computers. Figure 11(b) shows the state of reduced Hilbert spaces before and after the measurement of LQ3. We can confirm that the logical information of LQ3 was changed from $|0\rangle + |1\rangle$ to $|0\rangle$ in all reduced Hilbert spaces.

3.5. Mapping the qubit index

The transfer of logical information through bit-wise operations between reduced Hilbert spaces is a key principle in the foundation of our approach. However, if we consider the example of the physical lattice topology shown in figure 12(a), the direct application of our approach would require a significant computing overhead for bit computation. This high demand for computing power results from the need for repeating the bit operations for each data qubit, as bit streams of data qubits cannot be delivered at once. Therefore, we associate continuous qubit numbers to all data qubits by establishing a special mapping table, as shown in figure 12(b). However, it should be noted that no modifications to the quantum algorithm or breaking of the surface code operation's integrity are required when changing the qubit number. In fact, gate operations are called in an algorithm based on a given qubit number in the real lattice topology. Thus, changing the number of the qubit in the mapping table is a mere change of the qubit numbering within the computer's memory. However, breaking the connectivity of the physical qubits can be a potential problems, because most quantum computers (e.g. Google Sycamore) allow the CNOT operation only between adjacent qubits. For example, in figure 12(a), qubit 8 can apply the CNOT operation only with qubits 4, 5, 11, and 12. We check the qubit connectivity through the mapping table when the CNOT operation is called.

3.6. Noise injection and error correction

In quantum systems, noise can broadly be classified into two types: coherent and incoherent errors. The coherent error may be viewed as unitary rotations about a particular axis, and can be more damaging, since they can accumulate coherently over time. Several schemes for mitigating coherent errors have been proposed, including averaging over random gate sequences [48–50] and optimization of the decoding algorithm [51]. In contrast, incoherent error means stochastic noise, where an error operation is applied with some classical probability. A typical method is to use a depolarization channel [18, 52]. Simulating surface codes with depolarizing channels only requires Clifford operations and Pauli measurements on stabilizer states, allowing for efficient simulation on classical computers via the Gottesman–Knill theorem [53, 54].



Of these two, our study focuses specifically on emulating incoherent errors in the reduced Hilbert space. The main reason is that the coherence error rotates the state of physical qubits in specific direction, which results in expanding the size of the reduced Hilbert space. For example, even if nine physical data qubits in a distance 3 logical qubit are slightly rotated about XY axis, the size of its reduced Hilbert space will increase from 2^4 to 2^9 . This ultimately increases the computational cost of classical computing and undermines the purpose of this study to support more logical qubits with fewer resources in the reduced Hilbert space. To inject the incoherent error to the reduced Hilbert space, we utilize the well-known depolarization channel in quantum information theory as

$$N_i(\rho) = (1 - \varepsilon_i\rho) + \varepsilon_i^x X\rho X + \varepsilon_i^y Y\rho Y + \varepsilon_i^z Z\rho Z, \tag{7}$$

where suitable probabilities $\varepsilon_i^x, \varepsilon_i^y, \varepsilon_i^z$ and $\varepsilon_i = 1 - \varepsilon_i^x - \varepsilon_i^y - \varepsilon_i^z$. i indicates a physical qubit experiencing the Pauli error. The effect of Pauli noise also is efficiently simulable depending on the Gottesman–Knill theorem [18].

Figure 13 shows the noise injection and correction in the reduced Hilbert space. To simplify the example, we assume that an error occurred only in the i th physical data qubit of LQ0. The whole process consists of three stages. Step 1 injects a Pauli error generated by the probability-based depolarization channel in equation (7). In this case, the logical state of LQ0 is changed in the type of error applied to the i th data qubit. If it is an X error, the logical bit is reversed, and if it is a Z error, the logical phase is reversed. At this point, the error in a specific data qubit only affects the corresponding logical qubit, so the quantum state of the reduced Hilbert space can be expressed as $|\Psi^{LQ0}\rangle_{Logical} = |MQ^3\rangle \otimes |MQ^2\rangle \otimes |MQ^1\rangle \otimes (\varepsilon_i \otimes |LQ^0\rangle)$ in this stage. If the number of errors does not exceed the error-threshold which is $(d - 1) / 2$, such errors guarantee a locality that affects only the associated logical qubits. Thus, we can correct the error of the corresponding logical qubit through the ESM step without affecting the state of the other logical qubit. In step 2, an error can be detected or corrected according to the ESM circuit in figure 1. Depending on the type of syndrome, we correct the logical error via physical Pauli-X or -Z. However, if no noise is injected, then all stabilizers will be measured as (+) eigenvalue, indicating that no error occurred on the logical qubit. Step 3 is to apply logical operators to the logical qubit. Since the already injected error has been corrected, the operation can be performed in a noiseless logical state.

Table 1. Estimation of computing resources for reduced Hilbert states.

# Physical qubits	# Logical qubits	# Ideal physical states	# Realized-states	Total memory
85	4	2^{85}	2^8	16 KB
181	9	2^{181}	2^{13}	1 MB
313	16	2^{313}	2^{20}	256 MB
388	20	2^{388}	2^{24}	5 GB
481	25	2^{481}	2^{29}	200 GB

3.7. Cost estimation

In classical quantum simulators, memory space and computational costs exponentially saturate proportional to the number of qubits. Even in QuEST, moving logical information between two logical qubits takes more than 1 h. These problems are unlikely to be solved by using thousands of high-performance computers. The reason is that the Hilbert space for many qubits is too large to simulate using traditional computing resources. In contrast, our approach enables faster quantum operations at a lower cost by providing a reduced Hilbert space. In the following, we estimate the benefits of our idea in terms of memory requirements and computational costs.

3.7.1. Memory consumption

As described above, the reduced Hilbert space consists of the stabilized states of the assigned logical qubit and the magic states of the remaining magic qubits. Since the maximum number of states of the $|LQ\rangle$ and $|MQ\rangle$ are 2^5 and 2^1 (for $|+\rangle_L$ or $|-\rangle_L$), the total number of physical states of a reduced Hilbert space is given by

$$\#PhysicalStates = NSLQ \times NSMQ = 2^5 \times 2^{NLQ-1}, \quad (8)$$

where NSLQ is the number of physical states for LQ, NSMQ is the number of physical states for MQ, and NLQ is the number of logical qubits. #PhysicalStates refers to the number of the ‘realized-states’ stored in memory using QPlayer. Since the ‘realized-state’ is defined as a complex number variable in computing language, the memory usage for simulating such physical states on a computer is given by

$$\text{Memory(bytes)} = \{(\#PhysicalStates) \times \text{AMPSZ}\} \times \text{NSRHS} = \{(2^5 \times 2^{NLQ-1}) \times 2^4\} \times \text{NLQ}, \quad (9)$$

where AMPSZ refers to the size of the complex number in the computer memory and NSRHS corresponds to the number of reduced Hilbert spaces. Because a complex number that requires two 8-byte double data types representing a real part and an imaginary part, the memory size for each physical state is 2^4 bytes.

Table 1 summarizes the predicted values according to the number of logical qubits when applying our approach to a classical computer. Only 200 GB is required to simulate 25 logical qubits. However, the table is merely a conversion of the physical state into bytes. Considering the memory used internally by the QPlayer process, it is expected that a single computing server can support a little less than 25 logical qubits. Nevertheless, it may be very encouraging to be able to run quantum algorithms using approximately 20 logical qubits on a single computing server.

3.7.2. Matrix calculations

Given N qubits, traditional quantum simulators always calculate 2^{N-1} of 2×2 matrix operations for every gate operation. By contrast, a relatively small number of calculations is required in our approach because matrix calculation is applied only for the realized-states within the reduced Hilbert space. Therefore, assuming that all logical qubits are superposed, the maximum number of matrix calculations can be predicted as

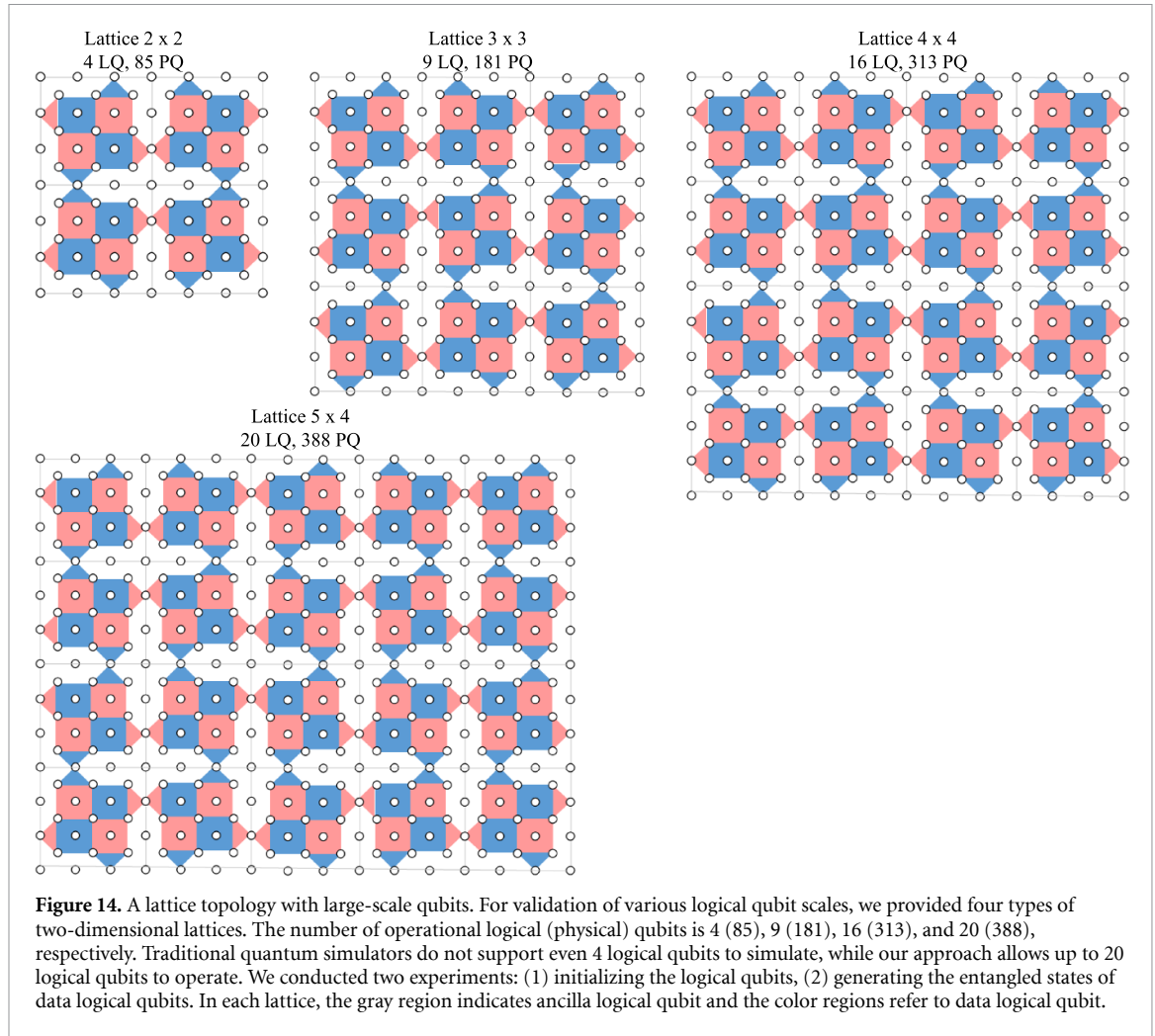
$$\text{MatrixCalculations} = NSLQ \times NSMQ \times \text{NSRHS} = 2^5 \times 2^{NLQ-1} \times \text{NLQ}. \quad (10)$$

4. Results and discussion

4.1. Experimental setup

We first simulated large-scale logical qubits using a classical quantum simulator (QPlayer)

supporting a reduced Hilbert space. Then, we analyzed the results from various perspectives. The experiment was conducted on a single Dell PowerEdge T640 server installed Ubuntu 20.04.2 LTS with two Intel Xeon Gold 6132 CPUs (56 cores total) and 512 GB of memory. We did not use any hardware accelerators such as GPGPU but used only the OpenMP library provided by Linux for simulation parallelization.



In addition, for the experiments on various logical qubit scales, we defined four types of 2D lattice topologies, as shown in figure 14. The color and gray areas correspond to the data logical qubits and the ancilla logical qubits, respectively. In this lattice topology, we conducted two experiments: (1) initializing all logical qubits and (2) generating the entangled states of data logical qubits.

4.2. Initializing logical qubits

Table 2 shows the time and memory consumptions required to initialize all logical qubits depending on their number. We initialized the logical qubits using the ESM of the surface code, to ensure that all the X/Z stabilizers have a (+1) eigenvalue. The experimental results include the number of stabilized physical states, initialization time, and required memory. First, we consider initializing logical qubits to $|+\rangle_L$ on a 2×2 lattice. As predicted in equation (8), there are 256 physical states in one reduced Hilbert space. Since the number of reduced Hilbert spaces is 4, the total number of physical quantum states is 1024. The initialization process took 334 ms and consumed 5.3 MB. These values differ slightly from those listed in table 2 because the QPlayer process uses meta-information such as quantum index as well as quantum states. Meanwhile, in the case of a 5×4 lattice, the total number of physical states generated for $|+\rangle_L$ is 268 435 456 states using 276 GB. In this case, it took ~ 1475 s to complete the initialization process of all logical qubits. This relatively long operation time accounts for the initialization of both the data logical qubit and the ancilla logical qubit. It is worth recalling that the ancilla logical qubit is temporarily used only during the lattice surgery when applying real quantum algorithms.

4.3. Building entangled states for all logical qubits

Table 3 shows the entangled states of data logical qubits. We applied Method 3 in figure 9 for the logical CNOT operation. All operations were conducted only with lattice surgery using physical qubits. In the preparation step, we initialized a control logical qubit to $|+\rangle_L$ and the remaining logical qubits to $|0\rangle_L$. Then, we applied a continuous logical CNOT operation in a daisy chain starting from the control logical qubit to the remaining logical qubits (refer to appendix A). Meanwhile, the logical CNOT operation between two

Table 2. Initializing data logical qubits for four types of initial logical states.

Lattice topology	Initial states of all logical qubit	# Physical states	Elapsed time	Memory
2×2 (4 LQs)	$ 0\rangle_L$	$64 (16 \times 4)$	146 ms	4.6 MB
	$ 1\rangle_L$	$64 (16 \times 4)$	162 ms	4.6 MB
	$ +\rangle_L$	$1024 (256 \times 4)$	334 ms	5.3 MB
	$ -\rangle_L$	$1024 (256 \times 4)$	329 ms	5.3 MB
3×3 (9 LQs)	$ 0\rangle_L$	$144 (16 \times 9)$	254 ms	4.8 MB
	$ 1\rangle_L$	$144 (16 \times 9)$	269 ms	4.8 MB
	$ +\rangle_L$	$73\,728 (8192 \times 9)$	829 ms	118 MB
	$ -\rangle_L$	$73\,728 (8192 \times 9)$	817 ms	118 MB
4×4 (16 LQs)	$ 0\rangle_L$	$256 (16 \times 16)$	297 ms	6.4 MB
	$ 1\rangle_L$	$256 (16 \times 16)$	282 ms	6.4 MB
	$ +\rangle_L$	$16\,777\,216 (1048\,576 \times 16)$	81 s	16 GB
	$ -\rangle_L$	$16\,777\,216 (1048\,576 \times 16)$	82 s	16 GB
5×4 (20 LQs)	$ 0\rangle_L$	$320 (16 \times 20)$	350 ms	7.2 MB
	$ 1\rangle_L$	$320 (16 \times 20)$	348 ms	7.2 MB
	$ +\rangle_L$	$268\,435\,456 (16\,777\,216 \times 20)$	1475 s	276 GB
	$ -\rangle_L$	$268\,435\,456 (16\,777\,216 \times 20)$	1499 s	276 GB

Table 3. Entanglement of all data logical qubits generated by lattice surgery.

Lattice topology	Logical states after CNOT operations	Elapsed time	Memory
2×2 (4 LQs)	$ 0000\rangle_L + 1111\rangle_L$	551 ms	37 MB
3×3 (9 LQs)	$ 000000000\rangle_L + 111111111\rangle_L$	1.7 s	40 MB
4×4 (16 LQs)	$ 0000000000000000\rangle_L + 1111111111111111\rangle_L$	2.8 s	41 MB
5×4 (20 LQs)	$ 00000000000000000000\rangle_L + 11111111111111111111\rangle_L$	6.9 s	42 MB

logical qubits follows five steps: (1) it synthesizes the reduced Hilbert spaces of the control logical qubit and the target logical qubit, (2) it executes a merge/split XX of the ancilla logical qubit and the target logical qubit, (3) it executes a merge/split ZZ of the ancilla logical qubit and the control logical qubit, (4) it applies the post-operation according to the syndrome measured in the merge/split stage, and (5) it restores the logical information to the original reduced Hilbert spaces. As shown in table 3, the logical qubits are entangled with $|0\dots 0\rangle_L + |1\dots 1\rangle_L$ after all CNOT operations are completed. The physical states of the logically entangled states are described in appendix B. The memory consumption can also be inferred from the physical quantum states described in appendix B.

5. Conclusion

In this study, we proposed a novel method for simulating large-scale logical qubits based on the surface code running on a classical computer. The main limitations experienced in conventional quantum simulators are related to the number of qubits as these simulators try to apply all quantum operations in one Hilbert space. We presented fundamental aspects to overcome these limitations by dividing one Hilbert space into reduced Hilbert spaces for each logical qubit. We successfully showed that our proposed approach enables quantum operations on a larger number of logical qubits while consuming less memory resources and performing faster quantum computation than ever before. We experimentally demonstrated that our idea can support up to 20 logical qubits on a single computing server. Furthermore, we found that lattice surgery of surface codes works well even in reduced Hilbert spaces while conducting entanglement generation experiments of multiple logical qubits.

Importantly, we demonstrated that the stabilized states of the logical qubits remained fully preserved, despite the use of reduced Hilbert spaces. We showed that all behaviors (such as ESM, logical operator, and lattice surgery) of the surface code are possible only with physical qubit operations. We carefully expect that our ideas would enable further research possibilities in the future, including their combination with various quantum-error models, or the use of multiple logical qubits in error-tolerant quantum algorithms. Finally, it is worth stressing that our proposed approach functions on a general-purpose computing server in the

laboratory. Since practical quantum computers are not to become available in the near future, our study offers means to advance the research on QEC and quantum algorithms in the noisy intermediate-scale quantum era.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

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).

Appendix A. Execution flow of the logical CNOT operation for generating entanglement

Figure 15 shows the sequence of the logical CNOT operation for generating entanglement of logical qubits in each 2D lattice. However, since the basic rules are the same for all lattices, we considered a 3×3 lattice as an example. In reduced Hilbert spaces, the detailed procedure of the logical CNOT using the lattice surgery is as follows. We applied Method 3 in figure 9 for the synthesis of reduced Hilbert spaces.

Step1: initialize data logical qubits

- initialize LQ4 to $|+\rangle_L$
- initialize LQ0 to $|0\rangle_L$
- initialize LQ2 to $|0\rangle_L$
- initialize LQ6 to $|0\rangle_L$
- initialize LQ8 to $|0\rangle_L$

Step2: logical CNOT LQ4 \rightarrow LQ0

- synthesize the reduced Hilbert space with LQ4 and LQ0
- merge/split XX of LQ1 (Ancilla) and LQ0 (Target) in the synthesized Hilbert space
- merge/split ZZ of LQ1 (Ancilla) and LQ4 (Control) in the synthesized Hilbert space
- restore logical information to the original Hilbert spaces of LQ4, LQ0
- remove the synthesized Hilbert space from the memory

Step3: logical CNOT LQ4 \rightarrow LQ2

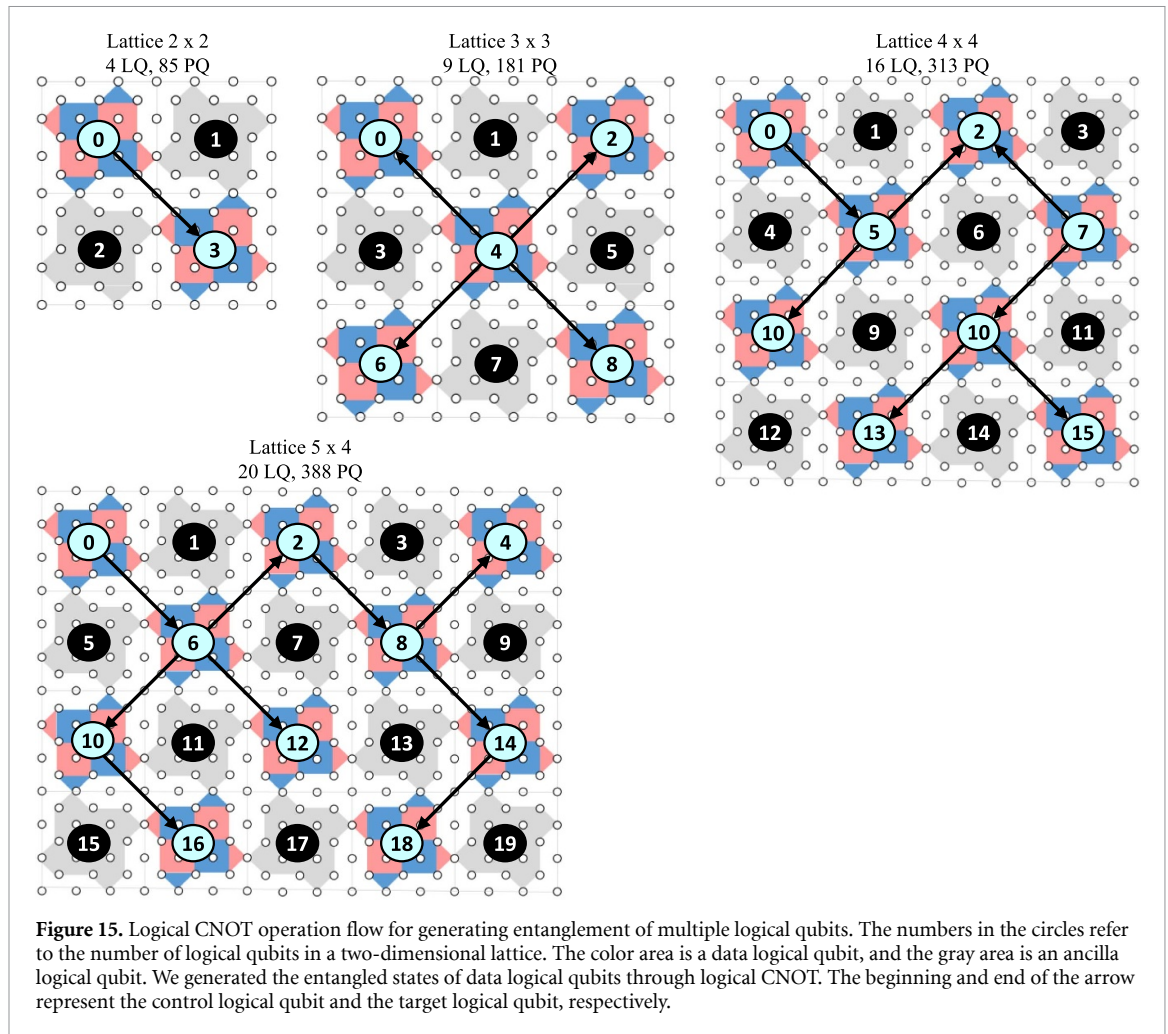
- synthesize the reduced Hilbert space with LQ4 and LQ2
- merge/split XX of LQ1 (Ancilla) and LQ2 (Target) in the synthesized Hilbert space
- merge/split ZZ of LQ1 (Ancilla) and LQ4 (Control) in the synthesized Hilbert space
- restore logical information to the original Hilbert spaces of LQ4, LQ2
- remove the synthesized Hilbert space from the memory

Step4: logical CNOT LQ4 \rightarrow LQ6

- synthesize the reduced Hilbert space with LQ4 and LQ6
- merge/split XX of LQ7 (Ancilla) and LQ6 (Target) in the synthesized Hilbert space
- merge/split ZZ of LQ7 (Ancilla) and LQ4 (Control) in the synthesized Hilbert space
- restore logical information to the original Hilbert spaces of LQ4, LQ6

Step5: logical CNOT LQ4 \rightarrow LQ8

- synthesize the reduced Hilbert space with LQ4 and LQ8
- merge/split XX of LQ7 (Ancilla) and LQ8 (Target) in the synthesized Hilbert space
- merge/split ZZ of LQ7 (Ancilla) and LQ4 (Control) in the synthesized Hilbert space
- restore logical information to the original Hilbert spaces of LQ4, LQ8
- remove the synthesized Hilbert space from the memory



Appendix B. The physical states after entanglement of logical qubits

We provide physical quantum states in the reduced Hilbert space after entanglement, considering the 3×3 lattice in appendix A, as an example. Furthermore, for simple explanation, we describe the physical quantum states for two reduced Hilbert spaces assigned to $|\Psi^{LQ0}\rangle$ and $|\Psi^{LQ4}\rangle$. The physical states of the remaining reduced Hilbert spaces can be inferred from the descriptions in this appendix. All physical states were actually observed through a classical quantum simulator (QPlayer). Figures 16(a) and (b) show the physical quantum states of reduced Hilbert spaces for LQ0 and LQ4, respectively. Blue refers to $|0\rangle_L$ and red to $|1\rangle_L$. The numerical values to the left are probability amplitudes, which encompasses a real part and an imaginary part. As described in section 3, since the state of all logical qubits after entanglement should be $|0..0\rangle_L + |1..1\rangle_L$, it is necessary to observe that this relationship is accurately represented in the reduced Hilbert space. In the case of the first Hilbert space, it can be seen that all stabilized states are described for LQ0. On the other hand, LQ2, LQ4, LQ6, and LQ8 represent the same logical state with a magic qubit. The second Hilbert space is also similar. Only LQ4 has the entire stabilized states, whereas the remaining data logical qubits show entanglement with magic qubits. Meanwhile, we can also confirm that the logical states of the two reduced Hilbert spaces are equivalent by comparing (a) and (b).

- [6] IBM Research 2016 IBM quantum experience (available at: <https://ibm.com/quantum-computing>)
- [7] Rigetti 2017 Rigetti computing (available at: <https://rigetti.com>)
- [8] Intel Corporation 2017 Intel quantum computing (available at: <https://intel.com/content/www/us/en/research/quantum-computing.html>)
- [9] Tzitrin I, Bourassa J E, Menicucci N C and Sabapathy K K 2020 Progress towards practical qubit computation using approximate Gottesman-Kitaev-Preskill codes *Phys. Rev. A* **101** 032315
- [10] Matsumoto R and Hagiwara M 2021 A survey of quantum error correction *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **104** 1654–64
- [11] Shor P W 1995 Scheme for reducing decoherence in quantum computer memory *Phys. Rev. A* **52** R2493
- [12] Steane A 1996 Error correcting codes in quantum theory *Phys. Rev. Lett.* **77** 793
- [13] Steane A 1996 Multiple-particle interference and quantum error correction *Proc. R. Soc. A* **452** 2551–77
- [14] Devitt S J, Munro W J and Nemoto K 2013 Quantum error correction for beginners *Rep. Prog. Phys.* **76** 076001
- [15] Aharonov D and Ben-Or M 1997 Fault-tolerant quantum computation with constant error *Proc. 29th Annual ACM Symp. on Theory of Computing* pp 176–88
- [16] Kitaev A Y 1997 Quantum computations: algorithms and error correction *Russ. Math. Surv.* **52** 1191
- [17] Freedman M H and Meyer D A 2001 Projective plane and planar quantum codes *Found. Comput. Math.* **1** 325–32
- [18] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 Surface codes: towards practical large-scale quantum computation *Phys. Rev. A* **86** 032324
- [19] Raussendorf R and Harrington J 2007 Fault-tolerant quantum computation with high threshold in two dimensions *Phys. Rev. Lett.* **98** 190504
- [20] Javadi-Abhari A, Gokhale P, Holmes A, Franklin D, Brown K R, Martonosi M and Chong F T 2017 Optimized surface code communication in superconducting quantum computers *Proc. 50th Annual IEEE/ACM Int. Symp. on Microarchitecture* pp 692–705
- [21] Andersen C K, Remm A, Lazar S, Krinner S, Lacroix N, Norris G J and Wallraff A 2020 Repeated quantum error detection in a surface code *Nat. Phys.* **16** 875–80
- [22] Bonilla Ataides J P, Tuckett D K, Bartlett S D, Flammia S T and Brown B J 2021 The XZZX surface code *Nat. Commun.* **12** 2172
- [23] Yoder T J and Kim I H 2017 The surface code with a twist *Quantum* **1** 2
- [24] Horsman C, Fowler A G, Devitt S and Van Meter R 2012 Surface code quantum computing by lattice surgery *New J. Phys.* **14** 123011
- [25] Arute F, Arya K, Babbush R, Bacon D, Bardin J C, Barends R and Martinis J M 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
- [26] Chow J, Oliver D and Jay G 2021 IBM Quantum breaks the 100-qubit processor barrier IBM Research Blog (available at: <https://research.ibm.com/blog/127-qubit-quantum-processor-eagle>)
- [27] Trout C J, Li M, Gutiérrez M, Wu Y, Wang S T, Duan L and Brown K R 2018 Simulating the performance of a distance-3 surface code in a linear ion trap *New J. Phys.* **20** 043038
- [28] Häner T and Steiger D S 2017 5 petabyte simulation of a 45-qubit quantum circuit *Proc. Int. Conf. for High Performance Computing Networking Storage and Analysis* pp 1–10
- [29] De Raedt K, Michielsen K, De Raedt H, Trieu B, Arnold G, Richter M and Ito N 2007 Massively parallel quantum computer simulator *Comput. Phys. Commun.* **176** 121–36
- [30] Smelyanskiy M, Sawaya N P and Aspuru-Guzik A 2016 qHiPSTER: the quantum high performance software testing environment (arXiv:1601.07195)
- [31] Wu X C, Di S, Dasgupta E M, Cappello F, Finkel H, Alexeev Y and Chong F T 2019 Full-state quantum circuit simulation by using data compression *Proc. Int. Conf. for High Performance Computing Networking Storage and Analysis* pp 1–24
- [32] Jones T, Brown A, Bush I and Benjamin S C 2019 QuEST and high performance simulation of quantum computers *Sci. Rep.* **9** 1–11
- [33] Shi Y Y, Duan L M and Vidal G 2006 Classical simulation of quantum many-body systems with a tree tensor network *Phys. Rev. A* **74** 022320
- [34] Viamontes G F, Markov I L and Hayes J P 2004 Graph-based simulation of quantum computation in the density matrix representation *Quantum Information and Computation II* vol 5436 pp 285–96
- [35] Markov I L and Shi Y 2008 Simulating quantum computation by contracting tensor networks *SIAM J. Comput.* **38** 963–81
- [36] Biamonte J and Bergholm V 2017 Tensor networks in a nutshell (arXiv:1708.00006)
- [37] Li R, Wu B, Ying M, Sun X and Yang G 2019 Quantum supremacy circuit simulation on Sunway TaihuLight *IEEE Trans. Parallel Distrib. Syst.* **31** 805–16
- [38] Boixo S, Isakov S V, Smelyanskiy V N and Neven H 2017 Simulation of low-depth quantum circuits as complex undirected graphical models (arXiv:1712.05384)
- [39] Chen Z Y, Zhou Q, Xue C, Yang X, Guo G C and Guo G P 2018 64-qubit quantum circuit simulation *Sci. Bull.* **63** 964–71
- [40] Doi J, Takahashi H, Raymond R, Imamichi T and Horii H 2019 Quantum computing simulator on a heterogeneous HPC system *Proc. 16th ACM Int. Conf. on Computing Frontiers* pp 85–93
- [41] Feynman Richard P 2018 *Simulating physics with computers Feynman and Computation* (Boca Raton, FL: CRC Press) pp 133–53
- [42] Roffe J 2019 Quantum error correction: an introductory guide *Contemp. Phys.* **60** 226–45
- [43] Jin K and Cha G 2022 QPlayer: lightweight, scalable, and fast quantum simulator *ETRI J.* **45** 304–317
- [44] Jin K 2023 *Quantum Simulator* (available at: <https://github.com/eQuantumOS/QPlayer>)
- [45] Lao L, van Wee B, Ashraf I, van Someren J, Khammassi N, Bertels K and Almudever C G 2018 Mapping of lattice surgery-based quantum circuits on surface code architectures *Quantum Sci. Technol.* **4** 015005
- [46] Fowler A G and Gidney C 2018 Low overhead quantum computation using lattice surgery (arXiv:1808.06709)
- [47] Lee J, Kang Y, Ha J and Heo J 2022 Lattice surgery-based Surface Code architecture using remote logical CNOT operation *Quantum Inf. Process.* **21** 217
- [48] Knee G C and Munro W J 2015 Optimal Trotterization in universal quantum simulators under faulty control *Phys. Rev. A* **91** 052327
- [49] Wallman J J and Emerson J 2016 Noise tailoring for scalable quantum computation via randomized compiling *Phys. Rev. A* **94** 052325
- [50] Campbell E 2017 Shorter gate sequences for quantum computing by mixing unitaries *Phys. Rev. A* **95** 042306
- [51] Chamberland C, Wallman J, Beale S and Laflamme R 2017 Hard decoding algorithm for optimizing thresholds under general Markovian noise *Phys. Rev. A* **95** 042332
- [52] Wang D S, Fowler A G and Hollenberg L C 2011 Surface code quantum computing with error rates over 1% *Phys. Rev. A* **83** 020302
- [53] Aaronson S and Gottesman D 2004 Improved simulation of stabilizer circuits *Phys. Rev. A* **70** 052328
- [54] Gottesman D 1998 The Heisenberg representation of quantum computers (arXiv:quant-ph/9807006)