# SCIENTIFIC REPORTS

Received: 11 October 2017 Accepted: 18 January 2019 Published online: 01 March 2019

## **OPEN** Dynamic Concatenation of **Quantum Error Correction in** Integrated Quantum Computing Architecture

Ilkwon Sohn<sup>1,3</sup>, Jeongho Bang<sup>2</sup> & Jun Heo<sup>1</sup>

Resource overhead problem caused by concatenation in quantum error correction (QEC) is of significant importance for the realization of fault-tolerant quantum computation (FTQC). To attack this problem, we propose a novel scheme by considering integrated FTQC architecture where the concatenation level is controlled dynamically; i.e., less (or more) concatenation levels are imposed by good (or poor) performance gates—we call this scheme "dynamic concatenation" in this sense. Such a dynamic concatenation is realizable in an integrated structure of FTQC, as the information of the concatenation can be communicated between classical system elements (e.g., compiler and system organizer) and the logical gubits in real-time. We derive the effective lower and upper bounds of the length of gate decomposition in order to achieve the practical advantage, namely of reduction of the overall operation time. By considering two non-trivial examples, it is shown that the aforementioned advantage can indeed be achieved in the presented scheme. Our result also provides an important scientific message, i.e., the interplay between "classical" and "quantum" can be helpful in QEC.

Recently, our forefront agenda of quantum computation is to establish efficient quantum error correction (QEC) schemes<sup>1-4</sup>. QEC is applied to correct various (quantum) errors for faithful quantum computation. To accomplish this, most QEC schemes require additional resources such as, ancillary qubits and gates for extracting and correcting the error without altering quantum properties on main computing qubits<sup>5-7</sup>. Such an overall register of qubits, including those for QEC, is usually called a *logical* qubit<sup>4</sup>. Typically, the ability to correct errors can be improved in a recursive way, particularly to achieve fault tolerance<sup>3</sup>. For example, a bundle of logical qubits can be defined as the second level of a logical qubit, hence any higher level of a logical qubit. This is called concatenation scheme<sup>2</sup>. However, this inevitably increases computational and/or time resources exponentially. Furthermore, non-transversal gates (e.g., for arbitrary phase rotation, or T gates) make the aforementioned problem worse<sup>8</sup>. Thus, it is crucial to solve such a resource-overhead problem toward realization of universal quantum computation (OC).

To solve the problem described above, many studies have been conducted in various perspectives. For example, the quantum circuit for QEC can directly be optimized<sup>6,9-11</sup>. There have been many works on the overhead problem for magic-state distillation<sup>12-18</sup>. Conversion between QEC codes having different transversal gates has also been studied<sup>19,20</sup>. Quite recently, an idea to use different QEC codes in each concatenation has been suggested<sup>21-24</sup>. However, such approaches have focused rather on quantum, although many classical elements should also be casted for universal QC<sup>25</sup>, as described later.

With such background, we start to design an integrated architecture for universal QC<sup>26-31</sup>, consisting of various working layers, i.e., quantum compiler<sup>32-35</sup>, system organizer<sup>25</sup>, logical and physical qubits. Such an architecture that covers both software and hardware has been commonly used in classical computation<sup>36</sup>. Considering this architecture, we suggested a scheme, named "dynamic concatenation (DC)", for lower-overhead. The basic idea is to arrange concatenation level, dynamically, according to gate operation that the computer is trying to use. This scheme is intuitively understandable: i.e., with lower-performance (or higher-performance) gates, more (or

<sup>1</sup>School of Electrical Engineering, Korea University, Seoul, Korea. <sup>2</sup>School of Computational Sciences, Korea Institute for Advanced Study, Seoul, 02455, Korea. <sup>3</sup>Advanced KREONET Center, Korea Institute of Science and Technology Information, Daejeon, Korea. Correspondence and requests for materials should be addressed to J.H. (email: junheo@korea.ac.kr)



**Figure 1.** Schematic picture of our integrated QC architecture and dynamic concatenation. (a) An integrated QC architecture. The quantum compiler decomposes a quantum algorithm into an assembly code. The assembly code and the maximum tolerable error rate  $\varepsilon_{\tau}$  are stored in a classical memory. The system organizer manages and controls logical qubits. In particular, it evaluates the concatenation level required to run the algorithm. The block of the logical qubits performs QC according to a set of logical gate operations. The physical-qubit block is responsible for the control of native gates at physical level. (b) A simple illustration of our dynamic concatenation (DC) scheme. By using our DC scheme, we can reduce concatenation level of a (logical) qubit from l to l-s. Of course, the additional processes (denoted as red boxes) for decoding and encoding should be adopted to complete the DC and it imposes the extra cost. Nevertheless, it is expected to achieve practical advantage, reduction of the overall operation time, for a sequence within a length, say N, of single (logical) qubits. Here, the effective lower and upper bound of N, which enable us to achieve the aforementioned advantage, is derived theoretically (see the main text).

*less*) *requirements of concatenation level would be imposed*. Since the gate-performance would strongly depend on the operation to be realized, such an idea is quite natural and reasonable. For example, it is generally harder to implement multi-qubit gates, i.e., correlating gates, than single-qubit gates<sup>37-40</sup>. We derive the effective lower and upper bounds of the gate-decomposition length that allow us to achieve the practical advantage, namely of reduction of the overall operation time. We then apply our DC scheme to the nontrivial examples: quantum Fourier transformation and ground state estimation algorithm on five (logical) qubits. By using Steane code<sup>41</sup>, we show that the aforementioned advantage can indeed be achieved. We believe that our approach will bring forward practical advantages and provide intuition on how classical-quantum interplay can improve the QEC.

#### Results

Integrated QC architecture. We will briefly describe our integrated architecture for QC consisting of the following working layers: (1) quantum compiler, (2) system organizer, (3) logical qubits, and (4) physical qubits (See Fig. 1a). In such an architecture, the process of QC runs as follows. Firstly, let us start with an operation or kernel (e.g., quantum Fourier transform) in a program. The information of the operation is delivered to the first working layer, i.e., quantum compiler. The main role of the quantum compiler is to decompose the delivered operation into a proper set of universal (logical) gates. The sequence of the decomposed gates is called "assembly code." The quantum compiler also computes the so-called maximum tolerable error rate (MTER)  $\varepsilon_{\tau}^{-34}$ . Here, note that the error of every decomposed gate should be lower than  $\varepsilon_{\tau}$ . The evaluated  $\varepsilon_{\tau}$  and assembly code are stored in a classical memory M. Then, the system organizer manages the whole system and controls logical qubits using the aforementioned information. The concatenation level required to complete the operation is typically evaluated in this working layer, system organizer. Conventionally, the evaluation of the required concatenation level is made, particularly depending on the lowest-performance gates in the whole decomposed set. However, the evaluation is done group-by-group in the set, i.e., dynamically, in our scheme (as described later). The block of logical qubits is responsible for implementing the logical gates assigned by the system organizer. This logical-qubit block also decomposes each logical gate again into native gates implemented at physical hardware level. Here, by "native gate" we mean that the gate is native to the physical hardware<sup>31</sup>. We note that every information used in each working layer are communicated through classical channel C, such as, I/O bus. Such an architecture is quite common in classical (and also quantum) computation<sup>36</sup>.

**Dynamic concatenation (DC) scheme.** In a typical scheme of QEC, the level of concatenation would be evaluated excessively large sometimes, because the evaluation is made based on the lowest-performance gates in the entire process. This necessarily causes a waste of computational resources. For example, let us consider that we meet a long series of single-qubit (decomposed) gates and a few two-qubit gates in a compilation. Here, noting that in general the single-qubit gates have lower error rates compared to the multi-qubit gates, it seems unreasonable to fix the concatenation levels based on the lowest-performance (mostly, two-qubit) gates. Thus, here we suggest a novel scheme, named dynamic concatenation (DC), that is to apply the concatenation group-by-group of the gates in the middle of the process. We expect that this DC scheme allows us to achieve a practical advantage and can be realized without spending too much extra cost, particularly when a long series, say *N*, of single-qubit gates is encountered in the assembly code.

Then, we derive the range of N in which the DC effectively works. Firstly, let us assume that a concatenation level is reduced from l to l-s. Here, s is the number of concatenations lowered by the DC. Then, by using the condition that the gate errors should be less then  $\varepsilon_{\tau}$ , we can write the upper bound of N, such that

$$\left(\prod_{i=1}^{s} F_D^{(l-i)} F_E^{(l-i)}\right) F_{1Q}^{(l-s)^N} \ge 1 - \frac{\varepsilon_{\tau}}{\gamma}$$

$$\tag{1}$$

where  $F_{1Q}^{(l-s)}$  is the fidelity of the single-qubit gates at the reduced, i.e., l-s, concatenation level and the factor  $\gamma$  is adopted for more tight condition.  $F_D^{(l-i)}$  and  $F_E^{(l-i)}$  are respectively the fidelity of the decoding and encoding, which have to be taken into account in order to analyze the extra cost of the DC (in terms of the fidelity degradation). This extra cost imposed by the imperfection, i.e.,  $F_{D,E}^{(l-i)} < 1$ , of the encoding and/or decoding should not violate the condition related to MTER. To proceed, we then adopt the approximation as

$$\prod_{i=1}^{s} F_D^{(l-i)} F_E^{(l-i)} \simeq F_{\text{CNOT}}^{(l-i)}^{(1-i)^{53s}},$$
(2)

where  $F_{\text{CNOT}}^{(l-1)}$  is the fidelity of CNOT gates at l-1 concatenation level. Such an adoption is acceptable, as a product of encoding and decoding fidelities is bounded by  $F_{\text{CNOT}}^{(l-1)^{53}}$ . Here, the exponent 53 is made by counting the effective propagation of the error in the whole encoding and decoding circuits (Please see supplementary information). Then, by assuming that  $F_{1Q}^{(l-s)} \leq F_{\text{CNOT}}^{(l-1)}$ , we can characterize the upper bound of N as the following form:

$$N \leq \left| \frac{\log\left(1 - \frac{\varepsilon_{\tau}}{\gamma}\right)}{\log F_{1Q}^{(l-s)}} - 53s \right|,\tag{3}$$

where  $\lfloor \cdot \rfloor$  denotes the flooring operation.

Now, we turn to the lower bound of *N*. This lower bound can be obtained by assuming the reduction of the time to complete *N* series of single-qubit gates; i.e.,

$$\frac{1}{\prod_{i=1}^{s} F_{D}^{(l-i)} F_{E}^{(l-i)} F_{1Q}^{(l-s)^{N}}} \left( \sum_{i=1}^{s} \left( T_{D}^{(l-i)} + T_{E}^{(l-i)} \right) + N T_{1Q}^{(l-s)} \right) < N \frac{T_{1Q}^{(l)}}{F_{1Q}^{(l)^{N}}}, \tag{4}$$

where  $T_{1Q}$  is the required time for single-qubit gate operation;  $T_D$  and  $T_E$  are the time to complete the additional decoding and encoding, which are also adopted for the analysis of the extra cost of the DC (in terms of the time delay), together with  $F_D^{(1-i)}$  and  $F_E^{(1-i)}$ . Then, by using Eq. (2) and assuming

$$\sum_{i=1}^{s} \left( T_D^{(l-i)} + T_E^{(l-i)} \right) \simeq s(T_D^{(l-1)} + T_E^{(l-1)}),$$
(5)

we can arrive at the lower bound of N as

$$N \ge \left[ \left( \frac{s(T_D^{(l-1)} + T_E^{(l-1)})}{T_{1Q}^{(l)}} \right) \frac{1}{F_{\text{CNOT}}^{(l-1)^{53s}}} \right], \tag{6}$$

where  $[\cdot]$  denotes the ceiling operation. Here, we note that the first term  $s(T_D^{(l-1)} + T_E^{(l-1)})T_{1Q}^{(l)^{-1}}$  is dominant, because  $1/F_{CNOT}^{(l-1)^{53s}}$  is close to 1. Consequently, we can derive the effective range of *N* enabling the practical advantage, namely of the fidelity enhancement and/or reduction of the operation time:

$$\left| \frac{\left( s(T_D^{(l-1)} + T_E^{(l-1)}) }{T_{1Q}^{(l)}} \right) \frac{1}{F_{\text{CNOT}}^{(l-1)^{53s}}} \right| \le N \le \left| \frac{\log \left( 1 - \frac{\varepsilon_{\tau}}{\gamma} \right)}{\log F_{1Q}^{(l-s)}} - 53s \right|.$$
(7)



**Figure 2.** A density-plot of the effective N with respect to  $F_{1Q}^{(l-1)}$  and  $\varepsilon_{\tau}$  (on a log-log scale). A colored region indicates the length N that allows us to lower the concatenation level, satisfying Eq. (6). This graph also provides a useful intuition about how our DC scheme works. For instance, consider two points:  $F_{1Q}^{(l)} \simeq 10^{-13}$  (blue, denoted B) and  $F_{CNOT}^{(l)} \simeq 10^{-11}$  (red, denoted A) in the line of  $\varepsilon_{\tau} \simeq 10^{-8}$ . The concatenation level is conventionally determined by the lowest performance gates, i.e., the point A. However, in our DC scheme, the concatenation can be controlled dynamically between from A to B, depending on the number N of decomposed gates, providing the advantage (see the main text).



**Figure 3.** (a) n = 5 QFT circuit. (Initial states ( $|0\rangle$ ) are omitted. Time progresses from left to right.) A total of six controlled-*T* gates and controlled-*R<sub>Z</sub>* gates in this circuit, which generate 10 sequences of the single-qubit gates that cannot be processed in parallel. (b) Decomposed QFT circuit.  $R_Z(\varphi) \left(\varphi = \pm \frac{\pi}{8}, \pm \frac{\pi}{16}, \pm \frac{\pi}{32}\right)$  will be decomposed into 250 single-qubit gates each in this circuit. Thus there are 18 sequences  $R_Z(\varphi)$  of the single-qubit gates. However, eight of them are processed in parallel and 10 sequences affect the total processing time.

Analyzing further, we draw the graph of  $\log N$  with respect to  $\log \varepsilon_{\tau}$  and  $\log F_{1Q}^{(l-s)}$  by using Eq. (7) (see Fig. 2). In particular, we specify the region of  $(\log \varepsilon_{\tau}, \log F_{1Q}^{(l-1)})$  which can bring the above-described advantages by the DC. Such a specification can offer a better intuition of how our DC works. For example, let us consider the hypothetical values of single-qubit and two-qubit gate performances at l-1 concatenation level:  $F_{1Q}^{(l)} \simeq 10^{-13}$  (blue point, denoted *B*) and  $F_{CNOT}^{(l)} \simeq 10^{-11}$  (red point, denoted *A*). Here, let  $\varepsilon_{\tau}$  to be  $\simeq 10^{-8}$ . In a conventional scheme, the level *l* of the concatenations required to complete a QC has been determined only for the lowest-performance gates, i.e., the point *A*. However, in our DC scheme, the system organizer controls the concatenations dynamically, when a series *N* of single-qubit gates is met in an assembly code and its length *N* is satisfied with Eq. (7). In this case, the concatenation is determined dynamically, moving between the points A and B. This allows the gain *s* to be achieved by our DC scheme.

**Analysis for five-qubit quantum Fourier transform.** As an example, we consider five-qubit quantum Fourier transform (QFT). Firstly, we draw a circuit to run QFT at the program level (see Fig. 3a), where the Hadamard *H* and the conditional phase-rotation (i.e.,  $\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}$  and  $\frac{\pi}{16}$ ) gates are employed. The last two gates are SWAP<sup>42</sup>. Here, we omit the parts of initial states  $|0\rangle^{\otimes 5}$  and measurements at the end. These operations would be programmed as a set of commands by a user. The quantum compiler decompose these operations into the logical gates, i.e., *H*, *S*, *T*,  $R_Z(\varphi) \left(\varphi = \pm \frac{\pi}{8}, \pm \frac{\pi}{16}, \pm \frac{\pi}{32}\right)$ , and controlled-NOT (CNOT) gates (as in Fig. 3b), where  $R_Z(\varphi)$ 

	Level-2 Concatenation		Level-3 Concatenation	
	Operating Time (sec)	Error Rate	Operating Time (sec)	Error Rate
Single-Qubit Gate	${\simeq}1.9977 \times 10^{-7}$	${\simeq}4.8665 \times 10^{-15}$	$\simeq\!1.4009\times10^{-5}$	${\simeq}8.6481 \times 10^{-25}$
Two-Qubit Gate	${\simeq}1.0063 \times 10^{-7}$	${\simeq}4.8665 \times 10^{-11}$	$\simeq\!1.4010\times10^{-5}$	${\simeq}8.6481 \times 10^{-17}$

**Table 1.** Characteristics of logical gates at a concatenation level 2 and 3. Whether a one-qubit gate or a twoqubit gate, there is no significant difference in terms of operating time due to the common error correction process. However, the error rate of the single-qubit gate is about  $10^{-13}$  times lower than  $\varepsilon_{\tau}$ . We can reduce the operating time of the single-qubit gates sequence by reducing this gap through the DC.

.....

is the arbitrary  $\varphi$ -rotation gate;  $S = R_Z(\pi/2)$  and  $T = R_Z(\pi/4)$ . Here, we note that  $R_Z(\varphi)$  is decomposed into, approximately, more than 250 of H, T, and S gates again<sup>35,43</sup>. For simplicity of analysis, we do not consider the decomposition of the last two SWAPs. Actually, SWAP operation would be more primitive, e.g., in a quantum-dot system. The quantum compiler should also evaluate MTER  $\varepsilon_{\tau}$ . In the evaluation, we assume  $\varepsilon_{\tau} \simeq 10^{-12}$  and the threshold value  $p_{th} \simeq 2.7390 \times 10^{-5}$  of Steane code<sup>34,44</sup>. We assume further that the performance of the single-qubit gate is better than that of the two-qubit controlled gate<sup>45,46</sup>. Lastly, we do not take into account the measurement and magic-state usage to calculate the gate operating time. Thus, our calculations has no influence on the generality of the results, because high-level gates require a longer time to create a magic-state<sup>17</sup> and a higher level of logical qubit also requires more syndrome measurements.

On the basis of the analysis, it is found that three concatenation levels are required to complete the 5-qubit QFT in a conventional QEC. However, in the case of using our DC scheme, only two concatenations are sufficient for  $0 < N \le 91$ . In particular, we can prove that the overall operation time can be reduced (more than 20 times), such that

$$T_{\rm CC} \simeq 3.5331 \times 10^{-2} \ge T_{\rm DC} \simeq 1.4766 \times 10^{-3},$$
 (8)

where  $T_{CC}$  and  $T_{DC}$  are the operation time for conventional scheme and our DC scheme, respectively. This result is intuitively understandable, i.e., *more (or less) concatenations are imposed for lower (or higher) performance gates, such as two-qubit controlled gates (or single-qubit gates)*. Therefore, we expect that such a speed-up will be more conspicuous for large-qubit QFT, which is decomposed into a huge number of single-qubit gates (H, T, and S).

Analysis for quantum ground state estimation. We then consider the ground state estimation (GSE), which is an algorithm for finding the lowest energy (or state) of a Hamiltonian. The polynomial time quantum GSE runs based on the QFT and quantum phase estimation, where the energies are usually estimated for the number *m* of wave functions (called molecular weight) with *b*-bit of precision. Here, we consider a molecule having m = 40 with b = 3 bit of precision. Subsequently, we have  $\varepsilon_{\tau} \simeq 7.0353 \times 10^{-12}$ , which is evaluated by ScaffCC compiler<sup>34</sup>. In such a setting, the (logical) gate performance can be calculated on each level of concatenation. The results are listed in Table 1. Based on these results, we can infer that, in a conventional QEC scheme, three concatenations are required to complete quantum GSE, because the error rate of the two-qubit gate cannot reach  $\varepsilon_{\tau} \simeq 7.0353 \times 10^{-12}$  until level-3 of concatenation. However, if we consider only single-qubit gate, it is sufficient to adopt level-2 of concatenation. Thus, our DC scheme will bring speed-up, preventing waste of the computational time and resource. Actually, our results reveal that we could reduce one-level of concatenation.

#### Discussion

We have suggested a novel QEC concatenation scheme to reduce the overall operation time. Our main idea was to evaluate the concatenation, dynamically. The presented scheme was named "dynamic concatenation (DC)" in this sense. The presented scheme was expected to work effectively for a series *N* of decomposed single-qubit (logical) gates, providing the aforementioned advantage. The effective range of *N* was derived theoretically. We then applied our DC scheme to the computation of quantum Fourier transform and quantum ground state estimation on five (logical) qubits. As a result, we explicitly showed the expected advantage. Indeed, such an advantage could be utilized and enabled by the integrated QC architecture that consists of quantum-classical hybridized working layers. In this sense, our work also implies an important scientific message, that is, a proper interplay between "classical" and "quantum" would be very important for the realization of the universal QC. We believe that the presented scheme could be improved more by incorporating other useful schemes.

#### References

- 1. Shor, P. W. Scheme for reducing decoherence in quantum computer memory. Physical review A 52, R2493 (1995).
- Knill, E., Laflamme, R. & Zurek, W. H. Resilient quantum computation: error models and thresholds. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 454, 365–384 (1998).
- 3. Knill, E. Quantum computing with realistically noisy devices. *Nature* **434**, 39–44 (2005).
- 4. Terhal, B. M. Quantum error correction for quantum memories. Rev. Mod. Phys. 87, 307–346 (2015).
- 5. Suchara, M. et al. Comparing the Overhead of Topological and Concatenated Quantum Error Correction. ArXiv e-prints arXiv:1312.2316 (2013).
- 6. Goto, H. Minimizing resource overheads for fault-tolerant preparation of encoded states of the steane code. *Scientific Reports* 6, 19578 (2016).
- 7. Weinstein, Y. S. & Buchbinder, S. D. Use of shor states for the [7, 1, 3] quantum error-correcting code. Phys. Rev. A 86, 052336 (2012).
- 8. Bravyi, S. & Kitaev, A. Universal quantum computation with ideal clifford gates and noisy ancillas. Phys. Rev. A 71, 022316 (2005).
- 9. Chamberland, C. & Beverland, M. E. Flag fault-tolerant error correction with arbitrary distance codes. *Quantum* 2, 53 (2018).

- 10. Paetznick, A. Resource optimization for fault-tolerant quantum computing. Ph.D. thesis, University of Waterloo (2014).
- 11. Chao, R. & Reichardt, B. W. Fault-tolerant quantum computation with few qubits. ArXiv e-prints arXiv:1705.05365 (2017).
  - 12. Chamberland, C. & Cross, A. W. Fault-tolerant magic state preparation with flag qubits. *arXiv e-prints* arXiv:1811.00566 (2018).
- 13. Haah, J. & Hastings, M. B. Codes and Protocols for Distilling T, controlled-S, and Toffoli Gates. *Quantum* 2, 71 (2018).
- 14. Hastings, M. B. & Haah, J. Distillation with sublogarithmic overhead. *Phys. Rev. Lett.* **120**, 050504 (2018).
- 15. Haah, J., Hastings, M. B., Poulin, D. & Wecker, D. Magic state distillation at intermediate size. *Quantum Info. Comput.* **18**, 97–165 (2018).
- 16. Bravyi, S. & Haah, J. Magic-state distillation with low overhead. Phys. Rev. A 86, 052329 (2012).
- 17. Goto, H. Step-by-step magic state encoding for efficient fault-tolerant quantum computation. Scientific Reports 4, 7501 (2014).
- Haah, J., Hastings, M. B., Poulin, D. & Wecker, D. Magic state distillation with low space overhead and optimal asymptotic input count. *Quantum* 1, 31 (2017).
- 19. Anderson, J. T., Duclos-Cianci, G. & Poulin, D. Fault-tolerant conversion between the steane and reed-muller quantum codes. *Phys. Rev. Lett.* **113**, 080501 (2014).
- Hwang, Y., Choi, B.-S., Ko, Y.-c. & Heo, J. Fault-tolerant conversion between stabilizer codes by Clifford operations. ArXiv e-prints arXiv:1511.02596 (2015).
- Chamberland, C. & Jochym-O'Connor, T. Error suppression via complementary gauge choices in reed-muller codes. *Quantum Science and Technology* 2, 035008 (2017).
- 22. Chamberland, C., Jochym-O'Connor, T. & Laflamme, R. Thresholds for universal concatenated quantum codes. *Phys. Rev. Lett.* 117, 010501 (2016).
- 23. Chamberland, C., Jochym-O'Connor, T. & Laflamme, R. Overhead analysis of universal concatenated quantum codes. *Phys. Rev. A* **95**, 022313 (2017).
- 24. Nikahd, E., Sedighi, M. & Saheb Zamani, M. Nonuniform code concatenation for universal fault-tolerant quantum computing. *Phys. Rev. A* 96, 032337 (2017).
- 25. Metodi, T. S. & Chong, F. T. Quantum Computing for Computer Architects. (Morgan & Claypool, 2006).
- 26. Oskin, M., Chong, F. T. & Chuang, I. L. A practical architecture for reliable quantum computers. Computer 35, 79–87 (2002).
- Svore, K. M., Aho, A. V., Cross, A. W., Chuang, I. & Markov, I. L. A layered software architecture for quantum computing design tools. *Computer* 39, 74–83 (2006).
- 28. Jones, N. C. et al. Layered architecture for quantum computing. Physical Review X 2, 031007 (2012).
- 29. Van Meter, R. & Horsman, C. A blueprint for building a quantum computer. Communications of the ACM 56, 84-93 (2013).
- Monroe, C. et al. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. Physical Review A 89, 022317 (2014).
- 31. Debnath, S. et al. Demonstration of a small programmable quantum computer with atomic qubits. Nature 536, 63 (2016).
- 32. Tucci, R. R. A rudimentary quantum compiler (2cnd ed.). ArXiv preprint quant-ph/9902062 (1999).
- Shende, V. V., Bullock, S. S. & Markov, I. L. Synthesis of quantum-logic circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25, 1000–1010 (2006).
- 34. JavadiAbhari, A. et al. Scaffcc: Scalable compilation and analysis of quantum programs. Parallel Computing 45, 2-17 (2015).
- 35. Häner, T., Steiger, D. S., Svore, K. M. & Troyer, M. A software methodology for compiling quantum programs. CoRR abs/1604.01401 (2016).
- 36. Null, L. & Lobur, J. *The Essentials of Computer Organization and Architecture*, 4th edn (Jones and Bartlett Publishers, Inc., USA, 2014).
- 37. Gong, B., Wang, L., Tu, T., Li, C.-F. & Guo, G.-C. Robust universal gates for quantum-dot spin qubits using tunable adiabatic passages. *Physical Review A* 94, 032311 (2016).
- 38. Nichol, J. M. et al. High-fidelity entangling gate for double-quantum-dot spin qubits. ArXiv preprint arXiv:1608.04258 (2016).
- 39. Harty, T. P. et al. High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit. Phys. Rev. Lett. 113, 220501 (2014).
- Ballance, C. J., Harty, T. P., Linke, N. M., Sepiol, M. A. & Lucas, D. M. High-fidelity quantum logic gates using trapped-ion hyperfine qubits. *Phys. Rev. Lett.* 117, 060504 (2016).
- 41. Steane, A. M. Error correcting codes in quantum theory. Phys. Rev. Lett. 77, 793-797 (1996).
- 42. Barenco, A., Deutsch, D., Ekert, A. & Jozsa, R. Conditional quantum dynamics and logic gates. *Phys. Rev. Lett.* 74, 4083–4086 (1995). 43. Kim, T. & Choi, B.-S. Efficient decomposition methods for controlled-rn using a single ancillary qubit. *Scientific Reports* 8, 5445
- (2018).
   44. Aliferis, P., Gottesman, D. & Preskill, J. Quantum accuracy threshold for concatenated distance-3 codes. ArXiv preprint quant-
- 44. Ameris, r., Gottesman, D. & Preskiii, J. Quantum accuracy threshold for concatenated distance-3 codes. ArXiv preprint quantph/0504218 (2005).
- 45. Kim, D. et al. High fidelity resonant gating of a silicon based quantum dot hybrid qubit. npj Quantum Inf. 1, 15004 (2015).
- 46. Petta, J. R. et al. Coherent manipulation of coupled electron spins in semiconductor quantum dots. Science 309, 2180–2184 (2005).

### Acknowledgements

We are grateful to Jaewan Kim and Seung-Woo Lee for helpful comments. J.B. also thanks to Marcin Wieśniak for discussion. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2015R1A2A2A01006301). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2018-0-01402) supervised by the IITP (Institute for Information & communications Technology Promotion)

### **Author Contributions**

I.S. and J.H. developed the theoretical idea. I.S. performed the main calculations and should be regarded as the main author. J.B. provided some technical support. I.S. wrote the manuscript with J.B. All authors contributed to analysis of the results.

### Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-019-39439-0.

Competing Interests: The authors declare no competing interests.

**Publisher's note:** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2019