2023 Eleventh International Symposium on Computing and Networking (CANDAR) | 979-8-3503-0670-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/CANDAR60563.2023.00034

Quantum Computer Architecture for Quantum Error Correction with Distributing Process to Multiple Temperature Layers

Ryuji Ukai
Center for Exploratory Research, R&D Group, Hitachi, Ltd.
Tokyo, Japan
ryuji.ukai.gf@hitachi.com

Chihiro Yoshimura
Center for Exploratory Research, R&D Group, Hitachi, Ltd.
Tokyo, Japan
chihiro.yoshimura.ak@hitachi.com

Hiroyuki Mizuno
Center for Exploratory Research, R&D Group, Hitachi, Ltd.
Tokyo, Japan
hiroyuki.mizuno.vp@hitachi.com

***Abstract*—Quantum computers are capable of performing large-scale calculations in a shorter time than conventional classical computers. Because quantum computers are realized in microscopic physical systems, unintended change in the quantum state is unavoidable due to interaction between environment, and it would lead to error in computation. Therefore, quantum error correction is needed to detect and correct errors that have occurred.**

In this paper, we propose quantum computer architecture for quantum error correction by taking account that the components of a quantum computer with quantum dots in silicon are divided into multiple temperature layers inside and outside the dilution refrigerator. Analog signals to control the qubits are precisely generated on a 4 K stage inside the dilution refrigerator, while real-time digital processing is performed outside the dilution refrigerator. We then experimentally demonstrate the digital control sequence for quantum error correction combined with a simulator which simulates quantum states during quantum computation. The real time processing including determination of feed-forward operation and transmission of feed-forward operation command is carried out by an FPGA outside the dilution refrigerator within 0.01 ms for bit-flip error correction. This is a sufficiently short time compared to the assumed relaxation time, which is the approximate time that the quantum state can be preserved, meaning that our proposed architecture is applicable to quantum error correction.

***Index Terms*—quantum computer, quantum computation, architecture, quantum error correction, feed-forward**

I. Introduction

Quantum computer is a new concept of computer that uses quantum mechanical phenomena to realize calculation [1]. Historically, it begins with a proposal of a microscopic quantum mechanical model of computer in 1980 [2], a proposal of simulating nature by quantum mechanical way in 1982 [3], and formularization of quantum Turing machine in 1985 [4]. At the beginning of calculation, information is encoded into a quantum state. Calculation is carried out by performing quantum state manipulations. Calculation result is achieved by measuring the final quantum states.

Quantum computers have attracted attention because of the discovery of efficient algorithms using quantum computers for problems that are considered too time-consuming to solve using conventional classical computers. Well-known examples are the factoring of integers [5] and quantum search algorithm [6].

Although demonstrations of quantum computers are underway in many systems such as silicon quantum dots [7], [8], photons [9]–[11], superconducting circuits [12], [13], and ion trap [14], [15], it is expected that silicon quantum dots have a potential to realize generation of large-scale quantum bits (qubits) and operations of quantum computations by applying semiconductor miniaturization technology.

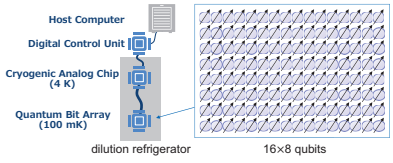


Fig. 1. Concept of large-scale quantum computer using silicon quantum dots.

Recently, a concept of large-scale quantum computer utilizing silicon integrated circuit technologies was proposed (Fig. 1) [16]. Using semiconductor miniaturization techniques, $16 \times 8 = 128$ quantum dots would be generated on a two-dimensional array and a quantum bit is placed on each quantum dot. This quantum bit array (QBA) chip is located at the 100 mK stage of a dilution refrigerator. Each qubit can be controlled by applying the appropriate sequence of analog signals such as high frequency signal and bias voltages. These analog signals are generated by a cryogenic analog chip (CAC) which is located at the 4 K stage of the dilution refrigerator. By placing CAC at a lower temperature (4 K) and closer to QBA, it is expected that the quality of the signal applied to the qubits will be improved. CAC is controlled by digital signals generated by the digital control unit (DCU) which is located outside but close to the dilution refrigerator at room temperature. The host computer, which may be installed away from the dilution refrigerator, controls the entire quantum computer.

Because quantum computers are realized in microscopic

979-8-3503-0670-5/23/$31.00 ©2023 IEEE
DOI 10.1109/CANDAR60563.2023.00034

physical systems in which quantum mechanical effects occur, unintended change in the quantum state is unavoidable due to interaction between environment. It would lead to error in quantum computation. Since a qubit will gradually change even when no quantum operations are performed on the qubit, it is necessary to complete a series of processes within an acceptable time for the state change, called the relaxation time. In the case of spin qubits in silicon, it is currently expected to be the order of 1 ms [17]. In order to achieve large-scale quantum computation, it is necessary to detect the occurrence of error and correct the error before error accumulates. The operation of returning to the quantum state before the unintended change of state occurs is called quantum error correction [1], [18].

In this paper, we propose quantum computer architecture for quantum error correction with multi-layers of temperature such as the dilution refrigerator. It is based on the concept of silicon quantum computer shown in Fig. 1 [16]. We show the digital control flow for detecting error and generating control signals for quantum error correction within the relaxation time. We then experimentally demonstrate the principle of the proposed architecture with digital control sequence for quantum error correction combined with a simulator for quantum states and quantum operations by targeting a simple error correction (bit-flip error correction) using a few qubits (three qubits).

II. Quantum Computation and Quantum Error Correction

In this section, we briefly review quantum computation and quantum error correction [1], [18].

Consider a one-qubit system with two computational bases. One example is the spin of an electron, which has two possible states: up and down. These two states are denoted as $|0\rangle$ and $|1\rangle$. In general, an arbitrary one-qubit quantum state with two computational bases is shown by $a|0\rangle+b|1\rangle$, where a and b are complex numbers which satisfy $|a|^2+|b|^2=1$. Alternatively, it can be denoted by a vector $\boldsymbol{v}=\begin{pmatrix}a\\b\end{pmatrix}$ with norm $|\boldsymbol{v}|=1$.

The computational bases of a multi-qubit state are given by the tensor product of the computational bases of one-qubit states. For example, the computational bases of a three-qubit state are $|0\rangle\otimes|0\rangle\otimes|0\rangle$, $|0\rangle\otimes|0\rangle\otimes|1\rangle$, $\cdots$, $|1\rangle\otimes|1\rangle\otimes|1\rangle$, or simply $|000\rangle, |001\rangle, \cdots, |111\rangle$. An arbitrary three-qubit state is denoted by a vector $\boldsymbol{v}=(v_0, \cdots, v_7)^T$ with $|\boldsymbol{v}|=1$, where $\boldsymbol{a}^T$ shows the transpose of the vector $\boldsymbol{a}$.

Quantum computation can be considered as a unitary operation on a quantum state. In matrix notation, it can be denoted as $\boldsymbol{v}'=U\boldsymbol{v}$, where $\boldsymbol{v}$ and $\boldsymbol{v}'$ are the vectors of input and output states, and U is a unitary matrix.

A bit-flip error is a simple error in quantum computation, in which $|0\rangle$ and $|1\rangle$ are inverted:

$$a|0\rangle+b|1\rangle \xrightarrow{\text{bit flip error}} a|1\rangle+b|0\rangle. \tag{1}$$

Fig. 2 shows the error correction circuit for the bit-flip error. In this figure, the subscript represents the identification number of the qubit. Let $|\psi\rangle$ be the initial quantum state of the qubit

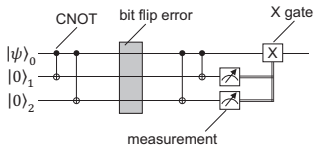


Fig. 2. Error correction circuit for bit-flip error.

that is made tolerant to error. Qubit 1 and 2 are additional qubits which are used for error correction. By applying two-qubit operations named Controlled-NOT (CNOT) to qubits 0 and 1, and then 0 and 2, the information in the initial quantum state is encoded in the three-qubit quantum state. Here, the unitary matrix of the CNOT gate is given by $C_X=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$. Each qubit of the encoded three-qubit state may be subject to the bit-flip error. We assume that the probability of error is low: no error occurs or that the bit-flip error occurs in only one of the three qubits. After exposure to the bit-flip error, CNOT gates are applied to qubits 0 and 2, and then 0 and 1. Qubits 1 and 2 are measured with the measurement bases $|0\rangle$ and $|1\rangle$. If both the measurement results on qubits 1 and 2 are 1, one-qubit operation X is applied to qubit 0, where $X=\begin{pmatrix}0&1\\1&0\end{pmatrix}$. Quantum operations based on measurement results are called feed-forward operations in quantum error correction. This error correction circuit allows the initial state to be restored even if a bit-flip error occurs in any of the qubits.

III. Architecture for Error Correction

Design space.—In this section, we show quantum computer architecture with digital control for quantum error correction based on the concept of silicon quantum computer shown in Fig. 1. As mentioned in the Introduction, the components of a quantum computer will be divided into multiple temperature layers inside and outside the dilution refrigerator. The cooling capacity of the dilution refrigerator ($\sim$ 1 W for 4 K stage and $\sim$ 1 mW for 100 mK stage [19]) would limit the performance of chips such as CAC installed in the dilution refrigerator. When distributing process of chips into different temperature layers, it takes a finite amount of time to transfer information between distant components especially inside and outside the dilution refrigerator. Considering quantum error correction, it requires a conditional operations on qubits depending on the results of measurements. A sequence of operations, including transmission of the measurement results, determination of the operation to be performed, and transmission of commands for the operation to be performed, must be carried out within the relaxation time. Therefore, it becomes the design space of quantum computer architecture how and where each processing is performed, and how information and commands are transmitted among each components.

The following is the key parameters in the design space of quantum computer architecture for error correction:

- where is it determined what operation will be performed?

- what level of abstraction should the command of the operation from DCU to CAC be?
- what is the bandwidth or latency of communication between CAC and DCU?
- what is the flow of transmission of measurement results?

The determination of these parameters are not independent of each other. Instead, the determination of one parameter will change the required specifications for the other parameters.

Real-time digital processing for feed-forward operation.—We first consider the number of bits of measurement results and complexity of determination of feed-forward operations. In the case of simple bit-flip error correction (Fig. 2), the number of bits is two and the determination of feed-forward operation is achieved by a simple AND gate for the two bits. Recently, more sophisticated quantum error collections for multi qubits have been proposed [20], [21]. As the size of the error collection increases, the number of measurement results increases and the computational complexity of the determination increases. For example, some methods that use neural networks to perform highly accurate decoding have been proposed [22], [23]. The quantum operations for correcting error must be determined in a sufficiently short time compared to the relaxation time. Several studies have been reported for realization of such real-time digital processing at cryogenic temperatures [24], [25]. However, these studies have predicted that the power consumption of the digital circuitry used for the real-time digital processing would not be negligible compared to the cooling capacity of the dilution refrigerator. In contrast, CAC, an analog circuit, is located at 4K, and it is desirable to increase the power input to CAC in order to improve the performance of CAC. Therefore, we decided to implement the real-time digital processing of decision for feed-forward operation in DCU, which is located outside the dilution refrigerator, in order to maximize the performance of analog signal generation by expending the cooling capacity of the dilution refrigerator for CAC.

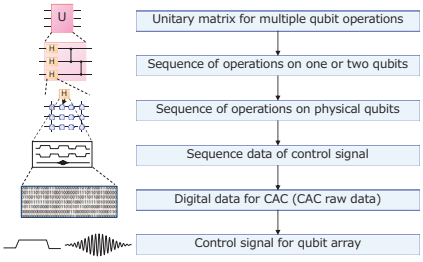


Fig. 3. Data conversion from unitary matrix to control signal for qubit array.

Data transmission.—Next we consider the level of abstraction of commands from DCU to CAC. Fig. 3 shows data conversion from unitary matrix to control signals for QBA. Quantum computations are represented by unitary matrices, whereas the control of the qubits is carried out by the control of bias voltages and high-frequency signals applied to the qubits. Therefore, a conversion from the unitary matrices to control signals is required. The first step of this conversion is that an unitary matrix for multiple qubit operations is decomposed into a sequence of operations on one or two qubits. It is known that an arbitrary multiple qubit operations can be approximated to arbitrary accuracy using the Hadamard gates (H), the $\pi/8$ gate (T), and the CNOT gate, where $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ and $T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}$ [1]. The next step is to assign each logical qubit in the unitary matrix to the physical qubit in QBA. It determines the sequence of unitary operations applied to each physical qubit in QBA. As the third step, the sequence of control signal for QBA is determined. Because the operation and the target qubit are determined, the control sequence to realize it is uniquely determined. Then, the control sequence data is converted once to digital data commands (CAC raw data) which can be stored in and interpreted by CAC. The last step is to execute the digital data commands and generate the control signals for QBA.

In general, computational resources must be consumed for the data conversion from the unitary matrix to control signal for QBA. Thus, in order to reduce the calculation volume and heat generation by CAC at 4K in the dilution refrigerator, the data in the later stages of the data conversion flow should be transmitted from DCU to CAC. However, in terms of the size of data to be transmitted, the later of the data conversion flow has the much size of data. Thus, there is a tradeoff between the amount of computation in CAC and the size of data to be transmitted from DCU to CAC.

In order to perform the feed-forward operation based on the measurement results in error correction within the relaxation time, the transmission time of the command from DCU to CAC must be sufficiently shorter than the relaxation time ($\sim$1 ms). To reduce power consumption at CAC for transmission, we use a simple protocol, serial peripheral interface (SPI), for communication between DCU and CAC. The master and slave of the SPI protocol are DCU and CAC, respectively. Considering the distance between DCU and CAC ($\sim$2 m) and the speed of electrical signal transmission, the maximum feasible clock frequency is about 5 MHz.

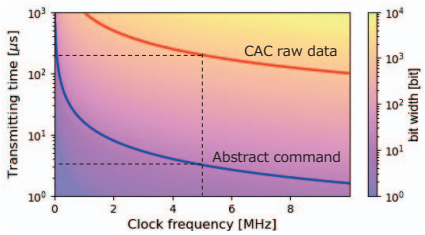
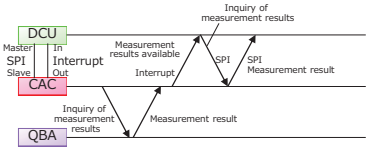


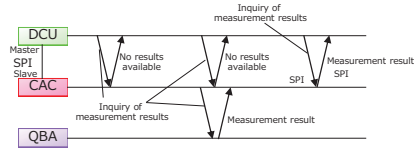
Fig. 4. Estimated time for transmitting command. Horizontal axis shows the clock frequency of SPI, while vertical axis shows the transmitting time for a one qubit operation.

We briefly estimate the time to transmit the digital data commands (CAC raw data) which can be directly interpreted by CAC from DCU to CAC (Fig. 4). For a one qubit operation, it would take about 0.2 ms (the size of data is $\sim$1k bit), which is not negligible compared to the relaxation time. In

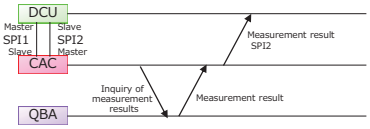
contrast, when transmitting the identifiers of operation and target physical qubit, it would take about 3 μs (the size of data is $\sim$16 bit). As an example of a typical use case of a large-scale QBA, we consider error correction in which a few logical qubits are encoded in about 100 physical qubits. The total time required for transmission of measurement results and feed-forward operation commands is expected to be about 40 μs ($\sim$100 bit measurement results and $\sim$100 bit operation commands), which is sufficiently smaller than the relaxation time. Therefore, we adopt the method of transmitting commands with a high level of abstraction which show the type of operations and the target qubits.



(a) Use of interrupt together.



(b) Polling.



(c) Dedicated line.

Fig. 5. Data flow to transmit measurement results.

Fig. 5 shows several types of data flow to transmit measurement results from CAC to DCU. Fig. 5(a) is a case where an interrupt line is used with SPI protocol. After receiving the measurement results from QBA, CAC interrupts DCU using the interrupt line to handle measurement results. DCU then sends an inquiry of measurement results to CAC, and CAC returns the measurement results to DCU via the SPI protocol. Fig. 5(b) shows a case where DCU repeatedly sends requests to CAC until CAC replies with the measurement results (method by polling). Fig. 5(c) shows a case where CAC transmits the measurement results to DCU with a dedicated SPI line. The master and slave of this SPI line are CAC and DCU, respectively.

In our quantum computer architecture for quantum error corrections, we use the data flow shown in Fig. 5(a) because the upper layer of quantum computer becomes the control master (which is consistent with our architectural philosophy), no extra communication exists during quantum computation, and it can be realized with relatively fewer signal lines.

Sequence diagram and discussion.—Fig. 6 shows an example of sequence diagram of quantum computation using quantum error correction. At the beginning of quantum computation, commands to control CAC are generated by the computer from a unitary matrix for multiple qubit operations. The commands consist of sequence data of operations on physical qubits and a conversion table from sequence data to control signals for QBA. The commands are transmitted to CAC from the computer via DCU. CAC generates control signals from the commands and the control signals are transmitted to QBA. After receiving measurement results from QBA, CAC interrupts DCU to handle measurement results. DCU requests and receives the measurement results from CAC. Based on the measurement results, DCU determines the feed-forward operation and generates commands for it. After receiving the commands, CAC carries out the feed-forward operation. The procedure for acquisition of calculation results is also illustrated in Fig. 6.

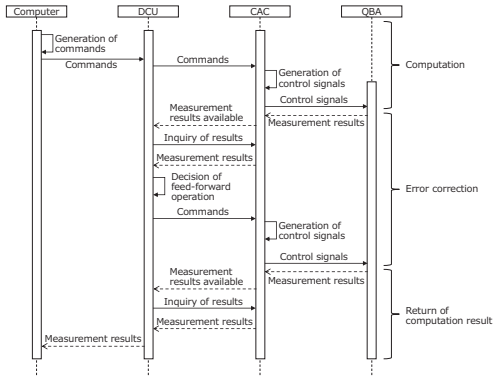


Fig. 6. Sequence Diagram.

Compared with the architecture where real-time digital processing is carried out at cryogenic temperatures, it is possible to reduce the power consumption for digital processing at 4K, and to increase the power input to CAC for better performance of analog signal generation at 4K while providing real-time digital processing necessary to achieve error correction by DCU at room temperature.

IV. Experiments

In this section, we show our experiment to demonstrate the principle of our quantum computation architecture for quantum error corrections. The target error correction protocol is the bit-flip error correction. In this experiment, we focus on the digital control of quantum computation architecture, so the processing in DCU and the communication between DCU and CAC are mainly considered. Instead, generation of control signals by CAC is omitted and quantum state operations are simulated by a simulator.

Fig. 7 and Fig. 8 show photographs and a schematic of our experimental setup, respectively. As DCU, we use a Digilent Zybo Z7-20, which is an evaluation board of a Zynq[1] XC7Z020-1CLG400C by Advanced Micro Devices, Inc. A Zynq series chip has a processing system (PS) with a processor and a programmable logic (PL) with a field-programmable

[1]"Zynq" is a trademark of Advanced Micro Devices, Inc.

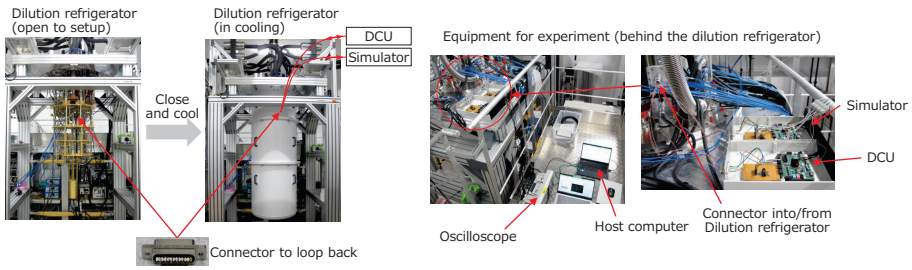
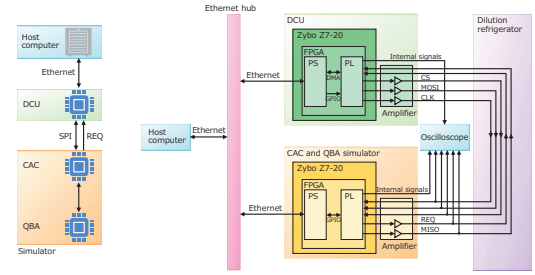
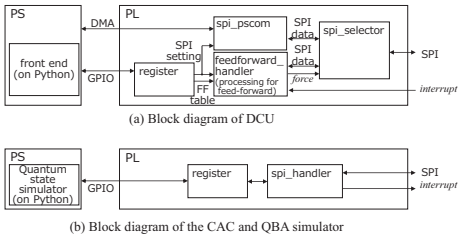


Fig. 7. Experimental setup.



(a) Abstract diagram. (b) Detail diagram.

Fig. 8. A schematic of our experimental setup.



(a) Block diagram of DCU

(b) Block diagram of the CAC and QBA simulator

Fig. 9. Block diagram.

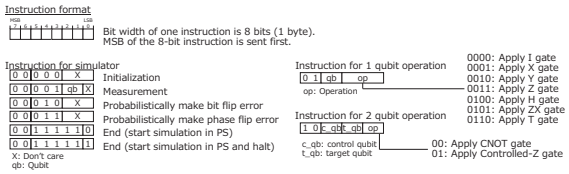


Fig. 10. SPI command format.

gate array (FPGA). We use PYNQ (Python[2] productivity for Zynq) with the Zybo board [26], [27]. PYNQ provides a framework to control PL via Python running on PS.

In actual quantum computation, CAC generates control signals based on commands received from DCU, and quantum state operations are performed by the control signals. Instead, in this experiment, we make and use a CAC and QBA simulator. It traces the quantum states changed from time to time as a result of the operations represented by the commands which CAC receives from DCU. This is achieved by calculating the change in the density matrix of the quantum state due to the operation. Let $|\psi\rangle$ be the initial quantum state. The density matrix ρ of the state $|\psi\rangle$ is given by $\rho = |\psi\rangle\langle\psi|$, where $\langle\psi|$ is the conjugate transpose of $|\psi\rangle$. The density matrix ρ' after the application of unitary operation U on the quantum state with the density matrix ρ is given by $\rho' = U\rho U^\dagger$, where $U^\dagger$ is the conjugate transpose of the unitary matrix U. The probabilistic bit-flip error channel is achieved by determining whether the error is occurred and the target qubit on which the error occurs based on a random number generated in PS.

The simulator in this experiment has two functions: one is to simulate communication with DCU by CAC, and the other is to simulate quantum states during quantum computation. In order to achieve these functions, we use another Zybo Z7-20 on which PYNQ is running. The communication with DCU and the simulation of quantum states are realized by PL and PS, respectively.

[2]"Python" is a registered trademark of the Python Software Foundation.

Fig. 9(a) and Fig. 9(b) show the block diagrams of DCU and the simulator, respectively. The command data of quantum operation sequence are prepared by a host computer in advance in the designated SPI format shown in Fig. 10. By the direction from the host computer, the command data are transmitted from a memory in DCU to PL by direct memory access (DMA). The spi_pscom block in DCU generates the SPI signals and then DCU sends the SPI data to the simulator via the SPI signal lines: clock (CLK), master out slave in (MOSI), and chip select (CS). The spi_selector block works as an interrupt handler: DCU normally performs SPI communication between the spi_pscom block and simulator, while DCU performs SPI communication between the feedforward_handler block and the simulator when interrupted by the signal *interrupt* from the simulator. Internally, the operation of the spi_selector block is determined by the GPIO line labeled *force*.

When PL in the simulator receives the command data from DCU by the spi_handler block, it temporarily stores the command data in a register. PS in the simulator receives the command data from PL via GPIO, then track the quantum states based on the command data. Once the simulation until the measurement in error correction is complete, PS transmits the measurement results to PL. Then PL transmits a signal

for request of SPI communication to send the measurement results from the simulator to DCU via the designated GPIO line: *interrupt*.

As soon as DCU receives a request via *interrupt*, it immediately starts one-byte SPI transmission, which is handled by the feedforward_handler block. By activating the GPIO line labeled *force*, the spi_selector block transmits SPI data between the SPI lines and the feedforward_handler block during this transmission. Within this one-byte transmission, DCU receives the measurement results, determines the feedforward operation, and sends the command for the feedforward operation. In MOSI, DCU sends 01 in binary at the first two bits of the one-byte SPI transmission because the required feed-forward is a one-qubit operation (See Fig. 10). The simulator sends the two-bit measurement results to DCU in master in slave out (MISO) line of SPI at the first two bits of the one-byte SPI transmission. After DCU receives the two-bit measurement results, DCU determines the feedforward operation and the command for it by referring to the table prepared and stored in a register in advance, and sends that command with the remaining 6 bits of the one-byte SPI transmission. The simulator calculates the quantum state change based on the command.

Each transmission line (CLK, MOSI, CS, MISO, *interrupt*) has an amplifier (digital IC 74HC541) before the output of DCU or the simulator. The wires in the actual dilution refrigerator are used as the transmission lines between DCU and the simulator. In order to operate the simulator at room temperature outside the dilution refrigerator, the transmission lines from the DCU into the dilution refrigerator is looped back at the 4K plate where the CAC would be located. Therefore, the distance between DCU and the simulator is about twice the actual distance between DCU and CAC.

Fig. 11 shows an experimental result of digital control for bit-flip error correction. The initial state was set to $|\psi\rangle_i = TH|0\rangle$. The whole waveform data of one trial of the error correction is shown in Fig. 11(a). In addition to the signals of SPI and *interrupt*, internal signals (FF_determined, cmd[0:3], and read_done) are also captured by the digital input channels of a mixed signal oscilloscope (Keysight MSO-X 6004A). Here, FF_determined becomes high when the feed-forward command is determined in DCU. The cmd[0:3] shows a four-bit indicator which command is processed in PS of CAC and QBA simulator. The read_done shows that the process in PS of CAC and QBA simulator is finished. The sequence of the waveform data consists of four steps: (1) transmission of command data for initialization of quantum states and quantum state operations from DCU to the simulator, which are performed at time 0 ms in Fig. 11(a); (2) simulation of quantum state operations at time from 0 ms to 27 ms; (3) transmission of measurement results from the simulator to DCU, followed by transmission of command data for feed-forward operation from DCU to simulator at time 27 ms; and (4) simulation of feed-forward operation at time from 27 ms to 32 ms.

Fig. 11(b) shows the waveform of command data transmission for initialization of quantum states and quantum state operations from DCU to the simulator. Corresponding to the initialization of quantum state to $TH|0\rangle$ and the bit-flip error correction, the command data are 00 44 46 84 88 10 88 84 0A 0C 3E in hexadecimal, which show initialization to $|000\rangle$, H_0, T_0, $CNOT_{0,1}$, $CNOT_{0,2}$, probabilistic bit-flip error, $CNOT_{0,2}$, $CNOT_{0,1}$, measurement on qubit 1, measurement on qubit 2, and the end command.

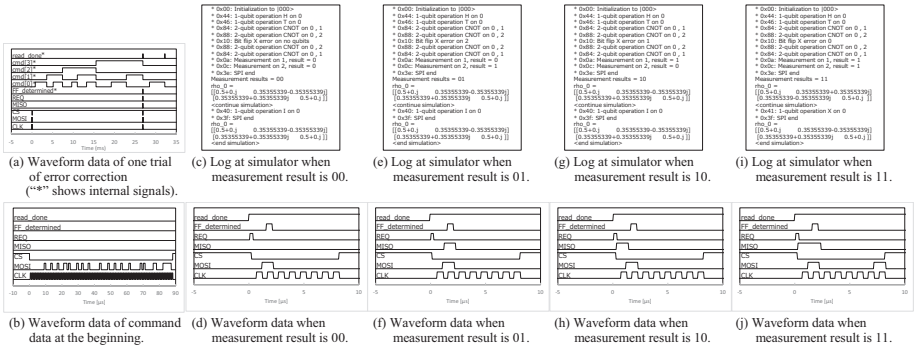
Fig. 11(c), (e), (g), and (i) show the log data of quantum state simulation at CAC and QBA simulator when measurement results are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Each corresponds to the case when bit-flip error occurs on no qubits, on qubit 2, on qubit 1, and on qubit 0, respectively. While the density matrix in the initial state is $\rho_i = |\psi\rangle_i\,{}_i\langle\psi| = \frac{1}{2}\begin{pmatrix} 1 & e^{-i\pi/4} \\ e^{i\pi/4} & 1 \end{pmatrix}$, the density matrices after the measurement are ρ_i, ρ_i, ρ_i, and $\rho_e = \frac{1}{2}\begin{pmatrix} 1 & e^{i\pi/4} \\ e^{-i\pi/4} & 1 \end{pmatrix}$.

Fig. 11(d), (f), (h), and (j) show the waveform data for transmission of measurement results from the simulator to DCU, and command data for feed-forward operation from DCU to simulator when measurement results are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Once the simulation at PS in the simulator was finished (read_done became high), the simulator made *interrupt* high. Triggered by this, DCU started SPI communication. The first two bits in MISO show the measurement results. Feedforward command was determined (FF_determined became high) as soon as the DCU received 2 bits. If they are $(1,1)$, feed-forward command became 41 in hexadecimal, which shows X_0. In any other case, feed-forward command became 40 in hexadecimal, which shows I_0 (identity operation with the unitary matrix $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$). By performing the feed-forward operations, the density matrix of the quantum state became ρ_i for all cases (Fig. 11(c), (e), (g), and (i)). This means that even if a bit-flip error occurred during the calculation process, the error has been corrected.

As Fig. 11(d), (f), (h), and (j) shows, the transmission of the measurement results from the simulator to DCU and the command transmission of the feed-forward operation from DCU to the simulator are realized during a one-byte transmission of the SPI communication. In particular, it takes less than one clock cycle of SPI communication from receiving the measurement results to determining the feed-forward operation. Transmission and determination were achieved within about 10 μs including transmission time, which is sufficiently shorter than the relaxation time.

V. Conclusion

We proposed quantum computer architecture with digital control for quantum error correction. Taking account that the components of a quantum computer with quantum dots in silicon are divided into multiple temperature layers inside and outside the dilution refrigerator, we determined the process of error correction that the measurement results are sent to DCU outside the dilution refrigerator, the feed-forward operation is determined by DCU, and the command for the feed-forward



(a) Waveform data of one trial of error correction ("*" shows internal signals).
(c) Log at simulator when measurement result is 00.
(e) Log at simulator when measurement result is 01.
(g) Log at simulator when measurement result is 10.
(i) Log at simulator when measurement result is 11.
(b) Waveform data of command data at the beginning.
(d) Waveform data when measurement result is 00.
(f) Waveform data when measurement result is 01.
(h) Waveform data when measurement result is 10.
(j) Waveform data when measurement result is 11.

Fig. 11. Experimental result of digital control for bit-flip error correction with the initial state $TH|0\rangle$.

operation is sent to CAC inside the dilution refrigerator. This would avoid the need for CAC at 4 K to be equipped with an FPGA or other high processing power, and would also make it possible to realize complex processing including for feed-forward operations with equipment at room temperature in future large-scale quantum computation.

We then experimentally demonstrated the principle of our quantum computer architecture for quantum error correction combined with a simulator which simulates quantum states during quantum computation. In particular, the processing in DCU and the communication between DCU and CAC were demonstrated. The real time processing including determination of feed-forward operations based on measurement results was carried out by an FPGA in DCU. It shows that our proposed quantum computer architecture with digital control has the performance to achieve error correction within the relaxation time. Our results will contribute to the realization of quantum computer systems using a large number of qubits.

ACKNOWLEDGMENT

This work is supported by JST Grant No. JPMJMS2065.

REFERENCES

[1] M. A. Nielsen and I. L. Chang, "Quantum Computation and Quantum Information," Cambridge University Press, Cambridge, England, 2020.
[2] Paul Benioff, "The Computer as a Physical System: Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines," Journal of Statistical Physics, vol. 22, pp. 563-591, 1980.
[3] Richard P. Feynman, "Simulating Physics with Computers," International Journal of Theoretical Physics, vol. 21, pp. 467-488, 1982.
[4] D. David, "Quantum theory, the Church-Turing principle and the universal quantum computer," Proceedings of the Royal Society of London. Series A, vol. 400, pp. 97-117, 1985.
[5] P.W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," Proceedings of 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, 1994.
[6] L.K. Grover, "A fast quantum mechanical algorithm for database search," Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, pp. 212-219, 1996.
[7] L. Steffen, Y. Salathe, M. Oppliger, P. Kurpiers, M. Baur, C. Lang, C. Eichler, G. Puebla-Hellmann, A. Fedorov, and A. Wallraff, "Deterministic quantum teleportation with feedforward in a solid state system," Nature, vol. 500, pp. 319-322, 2013.
[8] K. Takeda, A. Noiri, T. Nakajima, T. Kobayashi, and S. Tarucha, "Quantum error correction with silicon spin qubits," Nature, vol. 608, pp. 682-686, 2022.
[9] S. Yokoyama, R. Ukai, S. C. Armstrong, C. Sornphiphatphong, T. Kaji, S. Suzuki, J. Yoshikawa, H. Yonezawa, N. C. Menicucci, and A. Furusawa, "Ultra-Large-Scale continuous-variable cluster states multiplexed in the time domain," Nature Photonics, vol. 7, pp. 982-986, 2013.
[10] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, "Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement," Phys. Rev. Lett., vol. 99, 250505, 2007.
[11] T. Aoki, G. Takahashi, T. Kajiya, J. Yoshikawa, S. L. Braunstein, P. van Loock, and A. Furusawa, "Quantum error correction beyond qubits," Nature Physics, vol. 5, pp. 541-546, 2009.
[12] U. Skosana and M. Tame, "Demonstration of Shor's factoring algorithm for N = 21 on IBM quantum processors," Scientific Reports, vol. 11, p. 16599, 2021.
[13] M. Gong et al., "Experimental exploration of five-qubit quantum error-correcting code with superconducting qubits," National Science Review, vol. 9, nwab011, 2022.
[14] Daniel Barredo, Sylvain de Léséleuc, Vincent Lienhard, Thierry Lahaye, and Antoine Browaeys, "An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays," Science, vol. 354, 1021-1023, 2016.
[15] T. M. Graham et. al., "Multi-qubit entanglement and algorithms on a neutral-atom quantum computer," Nature, vol. 604, pp457-462, 2022.
[16] H. Mizuno, "Quantum Computing from Hype to Game Changer," 2023 Symposium on VLSI Technology and Circuits, PL2-1, 2023.
[17] A. M. J. Zwerver et al., "Qubits made by advanced semiconductor manufacturing," Nature Electronics vol. 5, pp. 184–190, 2022.
[18] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," Phys. Rev. A, vol. 52, R2493, 1995.
[19] G. Batey, A. J. Matthews and M. Patton, "A new ultra-low-temperature cryogen-free experimental platform, J. Phys. Conf. Ser., vol. 568, p. 032014, 2014.
[20] Emanuel Knill, "Quantum computing with realistically noisy devices," Nature, vol. 434, pp. 39–44, 2005.
[21] J. Ghosh, A. G. Fowler, and M. R. Geller, "Surface code with decoherence: An analysis of three superconducting architectures," Phys. Rev. A, vol. 86, p. 062318, 2012.
[22] G. Torlai and R. G. Melko, "Neural Decoder for Topological Codes," Phys. Rev. Lett. vol. 119, p. 030501, 2017.
[23] S. Gicev, L. C. L. Hollenberg, and M. Usman, "A scalable and fast artificial neural network syndrome decoder for surface codes," Quantum vol. 7, p. 1058, 2023.
[24] H. Homulle, S. Visser, B. Patra, G. Ferrari, E. Prati, F. Sebastiano, and E. Charbon, "A reconfigurable cryogenic platform for the classical control of quantum processors," Review of Scientific Instruments, vol. 88, p. 045103, 2017.
[25] Y. Suzuki, Y. Ueno, W. Liao, M. Tanaka, and T. Tanimoto, "Circuit designs for practical-scale fault-tolerant quantum computing," 2023 Symposium on VLSI Technology and Circuits, JFS1-2, 2023.
[26] http://www.pynq.io/.
[27] https://pynq.readthedocs.io/en/latest/.