International Core Journal of Engineering Volume 12 Issue 1, 2026
ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

Software Control Method of Photonic Quantum Computer

Yuhang Liu
SUZHOU LINK-IC CO., LTD., Shanghai 200100, China

Abstract

The design and operation of software control for a photonic quantum computer are closely intertwined with its hardware implementation. Here, a brief overview of the operation of a photonic quantum computer is provided prior to the discussion of its control software. The control software can be divided into two components: those functioning before and during the execution of quantum circuits and algorithms on the quantum computer. The control software operating in the pre-runtime phase focuses on the calibration of the photonic quantum computer, the compilation of the algorithm to be executed, and the configuration of photonic circuits required for running the algorithm. For any fault-tolerant quantum computer, error correction is essential during the runtime phase.

Keywords

Photonic Quantum Computer; Error Correction; Quantum Calibration; Quantum Compilation; Mach-Zehnder Interferometer.

1. Introduction

According to the DiVincenzo criterion for quantum computers[1], quantum computing devices must meet seven criteria. The most important criteria for quantum computing are a scalable physical system with well-characterized qubits and a broad set of quantum gates. Therefore, in the following discussion, the generation of optical quantum states and the construction of quantum gates are described, respectively.

1.1 Generation of Optical Quantum State

The generation of optical quantum states means the preparation of quantum light sources. The most mainstream single-photon source generation method is to prepare time-correlated photon pairs based on spontaneous nonlinear processes obtained directly by the spontaneous four-wave mixing effect in silicon waveguides[2].

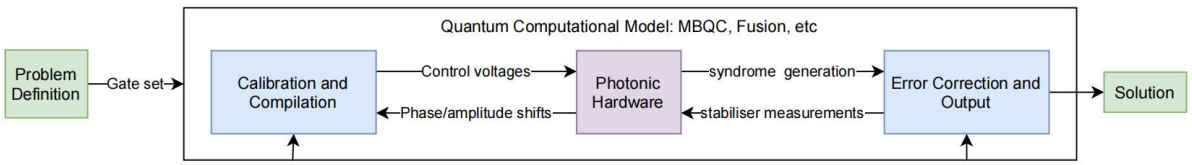

Fig. 1 Block diagram of software controlled photonic quantum computer.

Silicon waveguides have high third-order nonlinear coefficients, and photon pairs can be directly generated by the corresponding spontaneous four-wave mixing (SFWM) process[2]. In this process, two photons originating from the pump laser are converted into two new photons with conservation of momentum and energy between the four photons (as shown in Fig.2).

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

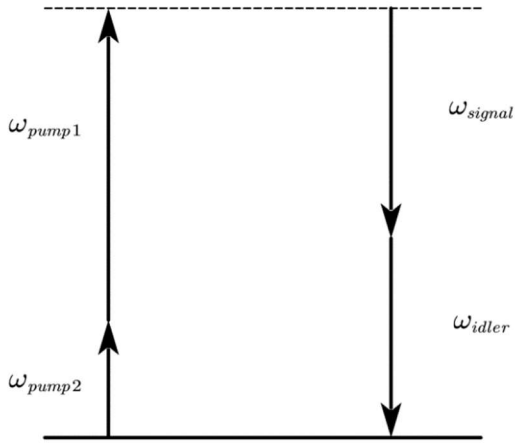

Fig. 2 Schematic diagram of the four-wave mixing process.

The general approach is to prepare photon pairs directly through silicon nanowires or ring cavities like Fig.3. After the photon pair is successfully prepared, the pump laser in the quantum signal needs to be filtered out. In experiments, high isolation wavelength division multiplexers (WDMs) and pre- and post-filters (IF, OF) are generally used to separate signal photons and stray photons. The figure below Fig.4 is a typical quantum light source structure principle.

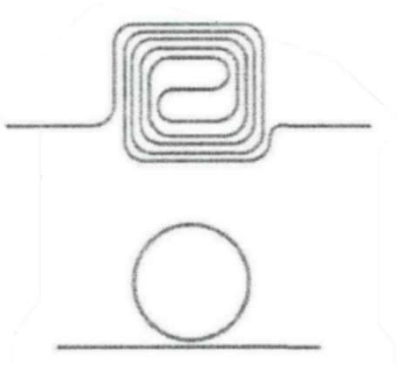

Fig. 3 Quantum light source composed of nano-wires and ring resonators.

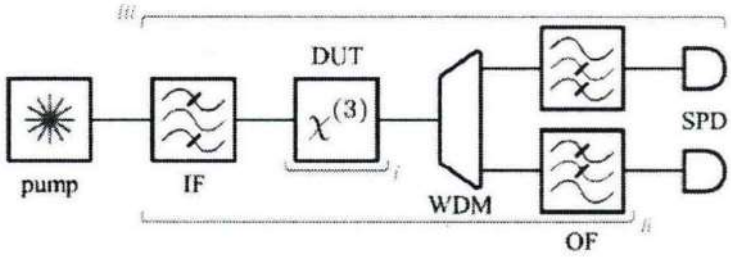

Fig. 4 A typical quantum light source structure principle.

After the quantum light source is prepared and filtered, we need to perform a series of operations on the corresponding quantum state to achieve the expected experimental purpose. The most critical optical components used here are the coupler and the phase controller. The coupler is based on the principle of photon transition coupling when the waveguides are very close together, while the phase controller is based on the thermo-optic and electro-optic effects of the material.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

If two couplers are cascaded and phase control is added in the middle, the simplest MZ interferometer can be realized and widely used in preparing and measuring on-chip quantum states[3], like Fig.5.

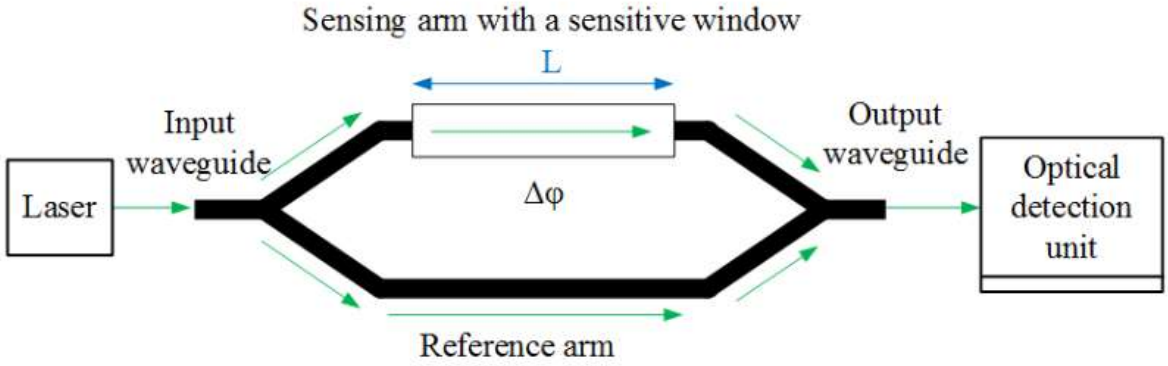

Fig. 5 Schematic diagram of a single-arm controlled thermo-optic MZI.

Any particle itself has many degrees of freedom that can be used to encode quantum information. Simultaneous control of its multiple degrees of freedom is necessary for adequate description and more efficient application of particles. The most commonly used encoding methods are polarization and path encoding among these quantum devices. Since photon polarization is very easy to tune in free space, many proof-of-concept demonstrations of quantum computing schemes are based on polarization degrees of freedom; for path entanglement schemes, which can be extended to higher-dimensional Hilbert spaces, we can use a single photon to encode more information and increase the security of quantum systems. Recently, waveguide modes have been used to encode lateral spatial energy distributions of these waveguide modes in optical fibres and classical integrated devices that are orthogonal to each other, similar to orbital angular momentum modes in spatial light. Therefore, they can be used to encode photonic information without crosstalk between them. Furthermore, to increase the information capacity of qubits, we can simultaneously use multiple degrees of freedom of particles for quantum operations, which has been experimentally achieved by special polarization-dependent mode converters and mode multiplexers[4].

1.2 Transmission and Regulation of Light Quantum States

In quantum computing, a variety of computing models are commonly used. There are four general computing models: quantum gate array, single-vector, adiabatic, and topological quantum computing. These four computational models are equivalent in polynomial time. In our next presentation, we will focus on quantum gate arrays. This model achieves the transport and regulation of light quantum states by combining quantum gates with built-in waveguides.

1.2.1 Theoretical Analysis of Quantum Gate Arrays

The quantum computing process can be decomposed into a series of small qubit gate operations just like classical computing. For single-bit gates, commonly used gate devices include the Hadamard gate, Pauli $(X(Y,Z))$ gate and phase gate. Let's take Hadamard gate as an example. Under the condition that the basis vector $(|0\rangle = (0,1)^T)$, $(|1\rangle = (1,0)^T)$ is selected, the H gate can be expressed as a matrix:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

In fact, for a single quantum gate, we can directly write its general form. For the above basis vector, the general single quantum gate can be expressed as:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

$$U(\theta,\phi,\lambda)=\begin{bmatrix}\cos\theta & -e^{i\lambda}\sin\theta\\ e^{i\phi}\sin\theta & e^{i\lambda+i\phi}\cos\theta\end{bmatrix}$$

Similar to classical computing, the quantum computing process can be decomposed into a series of small qubit gate operations. For single-qubit gates, commonly used gate devices include the Hadamard gate, Pauli $X(Y,Z)$ gate, and phase gate. Taking the Hadamard gate as an example: when the basis vectors are selected as $|0\rangle=(0,1)^T$ and $|1\rangle=(1,0)^T$, the Hadamard gate (H) can be expressed as a matrix:

$$H=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

In fact, the general form of a single quantum gate can be directly written. For the above basis vectors, the general single quantum gate is expressed as:

$$U(\theta,\phi,\lambda)=\begin{pmatrix}\cos\theta & -e^{i\lambda}\sin\theta\\ e^{i\phi}\sin\theta & e^{i(\lambda+\phi)}\cos\theta\end{pmatrix}$$

When extending to multi-qubit operations, multi-qubit gate operations can be divided into two categories. In the first category, no interaction between qubits is considered, and the quantum gate operation can be directly obtained by taking the tensor product of single-qubit gate operations. In the second category, when qubit interactions are considered, the corresponding operation evolution process cannot be decomposed into single-qubit operations. In this case, special controlled gate operations are required, such as the common CNOT gate structure. The CNOT gate has two input qubits: a control qubit and a target qubit. Its operation principle is: if the control qubit is in the $|0\rangle$ state, the target qubit remains unchanged; if the control qubit is in the $|1\rangle$ state, the target qubit is flipped.

In the two-photon basis vectors:

$$|00\rangle=\begin{pmatrix}1\times\begin{pmatrix}1\\0\end{pmatrix}\\ 0\times\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix},\quad |01\rangle=\begin{pmatrix}1\times\begin{pmatrix}0\\1\end{pmatrix}\\ 0\times\begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix}$$

$$|10\rangle=\begin{pmatrix}0\times\begin{pmatrix}1\\0\end{pmatrix}\\ 1\times\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix},\quad |11\rangle=\begin{pmatrix}0\times\begin{pmatrix}0\\1\end{pmatrix}\\ 1\times\begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix}$$

the CNOT gate can be expressed as:

$$U_{CNOT}=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

It has been proven that arbitrary multi-qubit operations can be constructed using universal single-quantum gates and CNOT gates[2]. Therefore, in the envisioned universal photonic quantum computer, as long as the above two types of quantum gates are constructed, all quantum logic can be implemented through combinations, and a compiler can be developed based on this foundation.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

1.2.2 Experimental Realization of Quantum Gates

All operations can be implemented using the general logic gates described in the previous section for single-bit quantum gates. In SOI, two paths are used to encode ($\{|0\rangle, |1\rangle\}$), which can be achieved by using a combination of a 50/50 beam splitter and a microcontroller, such as the MZ interferometer mentioned earlier. Our focus is on implementing a two-bit CNOT gate. In CNOT gates, selective regulation according to the quantum state of the control bits is required, which requires photon-photon solid nonlinear interactions. Since it is pretty challenging to realize strong nonlinear interactions in SOI directly, the mainstream uses probabilistic CNOT gate operations to achieve scalable quantum computing processes, requiring three non-equilibrium beam splitters for quantum interference and energy compensation.

To facilitate experimental integration and higher-dimensional information storage, path coding and pattern coding are used to realize the function of the CNOT gate. We know that a CNOT gate can be decomposed into sequential operations of a CZ gate and two H gates, and the CZ gate can delay the phase of (π) to the target bit according to the state of the control bit. This can be achieved by quantum interference. After the interference of two identical photons, the optical energy is compensated by two energy attenuators, and a probabilistic CZ gate with a success probability of 1/9 can be realized like Fig.6.

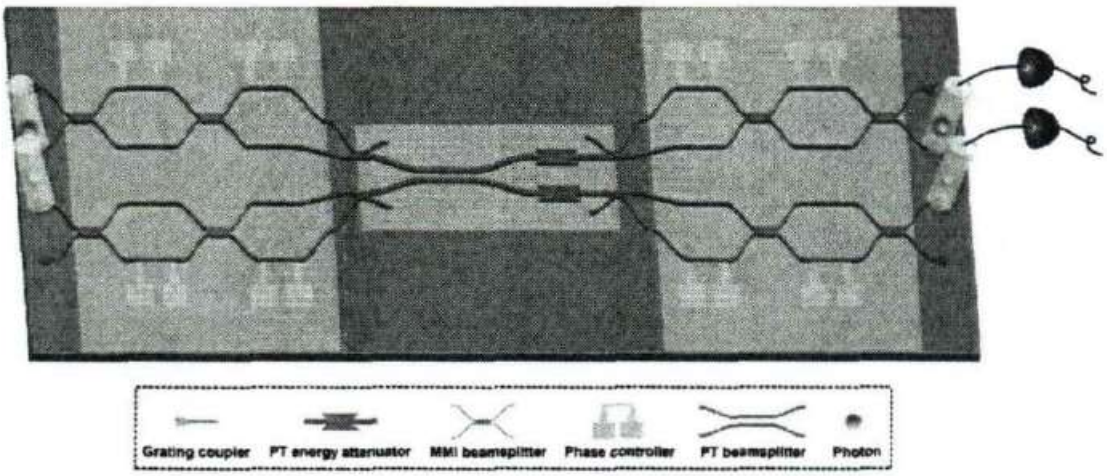

Fig. 6 Schematic diagram of a probabilistic CNOT gate using path coding and pattern coding.

After inserting single-bit H gates before and after, the whole system forms a probabilistic CNOT gate[5]. To determine the success of the CNOT gate construction, projection measurements of two-photon states can be achieved using an MZ structure and a phase controller after the second H gate. Similarly, we can get all quantum gate operations, which lays a sufficient low-level foundation for implementing the compiler.

2. Photonic Quantum Computer Software Control

2.1 Calibration

The process of building the quantum operating system begins with the quantum hardware system. Once a fault-tolerant quantum gate is realized, an instruction set for operation purpose could be realized. New research about the fault-tolerant universal quantum gate operation is discussed by researchers from University of Innsbruck. They realized a fault-tolerant logical T gate by injecting the magic state by teleportation from one logic qubit onto the other[6]. A practical quantum instruction set architecture technology is introduced in detail in a paper from Cornell University. They describe an abstract machine with a classical/quantum feedback loop, which is applicable to the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

optical quantum computer. Meanwhile, they proposed an instruction language for such kind of quantum computers called Quil and analyzed its feasibility for compilation.

Using the methods of University of Innsbruck, the optical quantum computer could keep steady, then a Quil-like instruction set is necessary for this computer. A general quantum computing algorithm is basically constructed with four single qubit operations, which are Phase gate, Hadamard gate, Pauli gate and $(\pi/8)$ gate. Apparently, these quantum computations should be combined for universal unitary gates. From the derivation about the two-level unitary gates, single qubit and CNOT gates we can see that these operations are universal[7].

The first step to build up an instruction set is to define those quantum gates as complex matrices with optional parameters. Then defining quantum circuits as a combination of bits and qubits with orders and expanding the quantum circuits is the way to address the quantum computer hardware. Finally, the system also needs to measure the qubits and record the measurements into a classical memory. A tentative quantum software framework is feasible with these fundamental elements.

2.2 Compilation

A typical quantum algorithm workflow with a gate-model quantum computer is shown in Fig.7. Many quantum compilers are absorbed into full-stack libraries or closed-source. One specific quantum assembly (QASM) compiler is proposed by Princeton University, and Fig.8 shows the internal structure of the ScaffCC compiler. The structure of this compiler could be separated into CTQG models, QASM code generation and quantum program analysis[7].

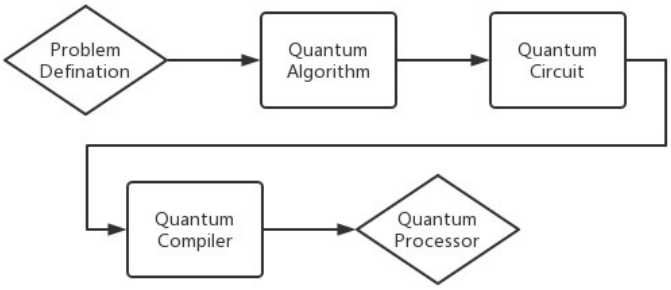

Fig. 7 Typical workflow for Quantum Computer.

The compilation process proceeds as follows:

1) Identify different modules in the program marked as CTQG.

2) Translate the output of the CTQG modules into QASM code, which is linked to the optical quantum computer hardware.

3) Use a QASM-to-IR (Intermediate Representation) translator to convert the entire program for subsequent analysis.

4) After analyzing the entire program, the compiler generates a specific mapping for the quantum computing system.

Quantum program analysis primarily focuses on qubit entanglement, resource consumption, and time consumption. Disentanglement identification is achieved by recording timestamps for each gate: when a disentangled qubit is detected, the system can track discarded qubits. Meanwhile, the wave function of auxiliary (ancilla) qubits must be aligned with that of data qubits to avoid incorrect outputs. Additionally, a final state check is required for each module to ensure all qubits are either uncomputed or measured. Analyzing the sources of qubits and operations is crucial for loading the program into the system.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

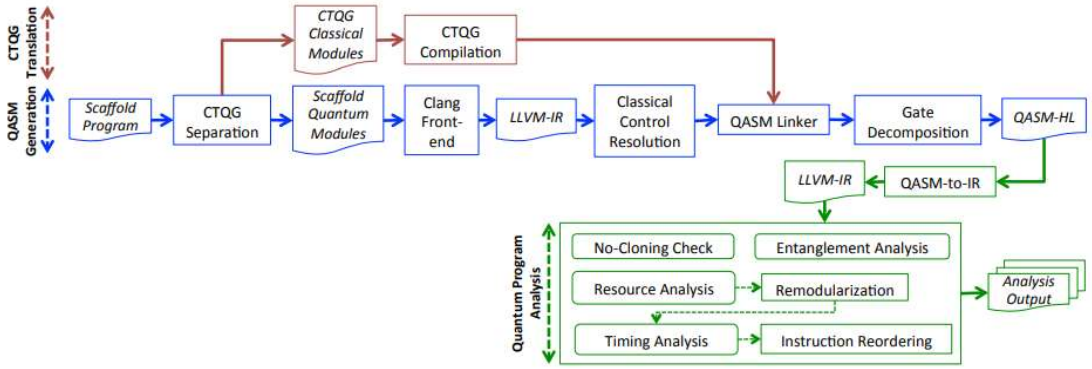

Fig. 8 Compiler structure for Quil.

Increasing the number of gates requires more qubits for error correction and computations. Due to the trillions of operations in quantum algorithms, storing each individual operation is infeasible. Similar to classical computer design, memory is used to avoid repeated calls to the same module. Another challenge is estimating the critical path length for time scheduling: since qubits cannot be replicated (due to the no-cloning theorem), the reading and measurement processes are identical in this system, so the timing of these instructions is scheduled in an "as-soon-as-possible" order.

2.3 Configuration

Performing tasks that cannot be done efficiently by classical computing is an important application of quantum computers. For example, in classical computing, we can find the prime factor of an n-bit number using the number field sieve method, an operation that is exponentially related to the size of the number, so this is a difficult problem in classical computing. While in quantum computing, we can use the $(O(n^2))$ operation to achieve a fast finding of its prime factor. With this result we can imagine that there are also other problems that cannot be solved by classical computing that can be accelerated exponentially on quantum computing.

The quantum Fourier transform is a quantum implementation of the Discrete Fourier Transform (DFT) in terms of wave function amplitudes and is the basis for algorithms such as quantum prime factorisation and quantum phase estimation. The Quantum Fourier Transform was first applied by Shor in the large number prime factorisation algorithm. As the algorithm runs much faster with the Quantum Fourier Transform, the Quantum Fourier Transform is widely used as one of the most critical subroutines in quantum algorithms after quantum computers.

The derivation of the quantum Fourier transform is an important reference for the configuration of optical quantum computers and is derived as follows.

First, the discrete Fourier transform is applied to the vector $((x_0, \ldots, x_{N-1}))$ and maps it onto $((y_0, \ldots, y_{N-1}))$. The transformation is defined as:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \, w_N^{jk}$$

where $(w_N = e^{-2\pi i/N})$.

Similarly, the quantum Fourier transform is applied to the state and maps it to:

$$|Y\rangle = \sum_{k=0}^{N-1} y_k \, |k\rangle$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

Substituting the first equation into the second:

$$|Y\rangle = \sum_{k=0}^{N-1} y_k |k\rangle$$
$$= \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j w_N^{jk} |k\rangle$$
$$= \sum_{j=0}^{N-1} x_j \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} w_N^{jk} |k\rangle$$

Therefore, it can also be understood that the quantum Fourier transform transforms the basis $(|j\rangle)$ into $(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} w_N^{jk} |k\rangle)$, i.e.:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} w_N^{jk} |k\rangle$$

It can also be understood as a Unitary Matrix:$U_{QFT} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} w_N^{jk} |k\rangle\langle j|$

This is due to the orthonormality of the computing base,$(\langle j|i\rangle = \delta_{ij})$, therefore, $(\sum_{j=0}^{N-1} \langle j|k\rangle = 1)$.

An alternative equivalent representation is derived as follows. Assume that $(N = 2^n)$ (n is an integer) and $(|0\rangle, \ldots, |2^n - 1\rangle)$ is the computational base of an n-quantum computer. The calculation base $(|j\rangle)$ can be rewritten to the binary representation $(j = j_1 j_2 \ldots j_n)$, which satisfies $(j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0)$. Also, the binary decimal $(0.j_l j_{l-1} \cdots j_m)$ represents $(j_l/2 + j_{l+1}/4 + \cdots + j_m/2^{m-l+1})$. Thus there is the following transformation, and we can write the quantum Fourier transform above in the form of the product below. This product representation of the quantum Fourier transform is more convenient and can be considered directly as the definition of the quantum Fourier transform. Using this product representation, it is easy to construct circuits for the quantum Fourier transform.

2.4 Continuous Running and Error Correction

Linear-optical quantum computation, like other quantum computational platforms, is highly sensitive to noise and must rely on continuous error correction for fault-tolerant operation. In discrete-variable quantum computation, errors occur when single-photons are lost, from noise in phase-shifters, or when probabilistic gates and fusions fail. In continuous-variable schemes, bright sources can compensate for optical loss (through absorption or scattering), but other error mechanisms still exist.

The quantum threshold theorem states that for a constant error rate lower than a threshold rate, quantum computation can be made entirely fault-tolerant with sufficient error-correction in place. The choice of photonic quantum computation method implemented (MBQC, Fusion, or others) will greatly affect the physical hardware and choice of error correcting codes.

Broadly speaking, all error correcting codes operate on a highly entangled large graph-state in which many physical qubits are used to form a single, logical qubit. Stabilizer codes are a family of quantum codes in which the use of ancillary qubits (or photons in the case of a photonic system) are used to error correct. The Calderbank-Shor-Steane (CSS) family of stabiliser codes were designed using classical codes as a basis and the Steane code is able to correct for arbitrary single qubit errors.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

Fig.9 shows the circular design choices of selecting a computation scheme, designing hardware and developing a novel error correction method. Fig.10 shows the general scheme adopted for error correcting; here fault-tolerance is applied to a single qubit ($|\Psi\rangle$) but the general figure is applicable to additional data qubits. Software running in an error correcting scheme will predominately be decoding the syndromes returned from the photonic hardware under a given error correcting scheme. Decoding algorithms have previously used minimum weight perfect matching where today we have open source software for decoding syndromes using this algorithm[8]. More advanced machine learning (ML) algorithms are being developed to further optimise the decoding process.

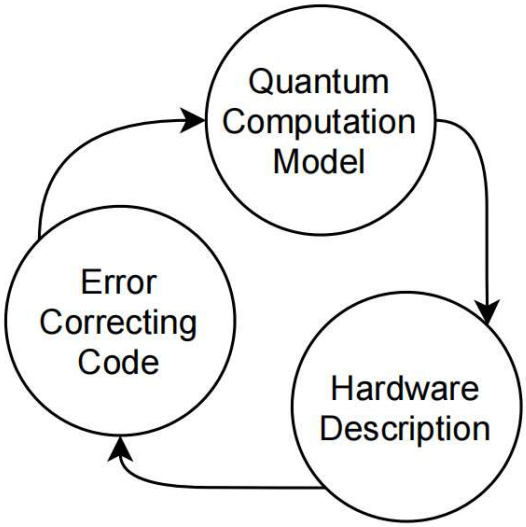

Fig. 9 Circular design choices of selecting computation scheme, designing hardware and developing a novel error correction method.

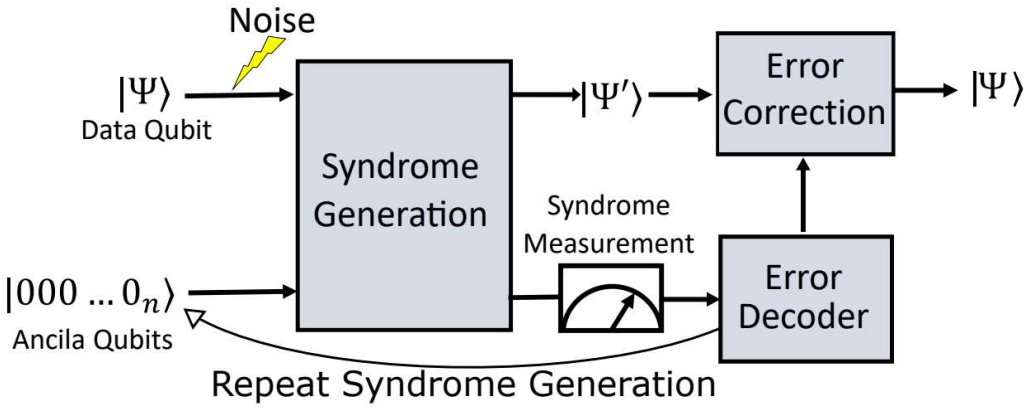

Fig. 10 Error correction of a single data qubit.

Error correction for a LOQC platform shares many problems with other quantum computation platforms, such as superconducting qubit architectures. Bandwidth and latency bottlenecks exist between the classical CPU controlling and running error correction for the quantum processing unit (QPU). Latency, the time it takes between an action from the CPU to affect the QPU, is of key interest in photonic quantum computers where storage of quantum information for any reasonable amount of time in a photon requires long delay lines that require large areas on chip and increase optical losses. As such, moving as much software control of the QPU to become closer to the quantum layer (Fig.12) as possible will help to reduce the control latency. This 'local computation' approach is more suitable

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

to photonic systems in which the operating temperature of the quantum layer and the cooling power available are orders of magnitude higher than in superconducting platforms.

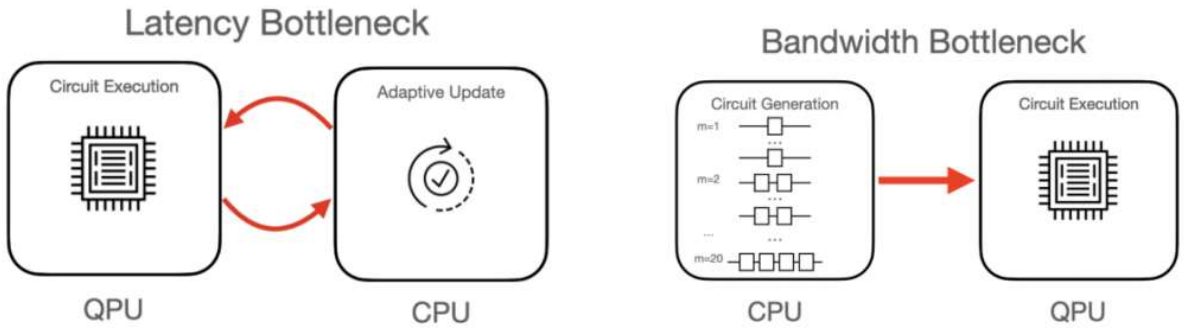

Fig. 11 Depiction of latency and bandwidth bottleneck in a quantum computer.

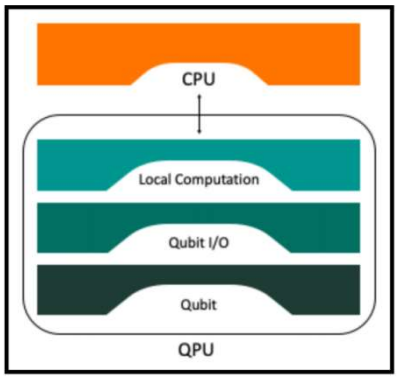

Fig. 12 Hardware and control stack of a quantum computer.

Error correcting schemes have been developed and demonstrated for smaller scale photonic systems[8] but fault-tolerant operation of a photonic quantum computer has yet to be demonstrated. Given the intertwined nature of the error correcting code and the exact nature of the hardware platform it is difficult to say what level of additional research is required before an error correction scheme suitable for a fault-tolerant photonic quantum computer is required. The author would hesitantly provide a technology readiness level of 4 to 5 given the demonstration of all aspects of error-correction on smaller scale devices.

3. Conclusion

Software control of a photonic quantum computer is interwoven with the exact hardware implementation of said quantum computer. Much work has been undertaken in the development of hardware operating under two distinct operating paradigms; discrete-variable and continuous-variable quantum computation. The software must be able to first calibrate the photonic circuit and then compile the quantum circuit, or other problem description, into the correct phases for the Mach-Zehnder interferometers in the photonic circuit. To achieve fault-tolerance in any quantum computer, error-correction is required. The classical CPU controlling the photonic quantum computer must be able to run the error correcting scheme; receiving error syndrome data, decoding errors from the syndrome data and then deciding on what correcting measurements to apply to the graph-state. Under the development category, would be assigned given each element of software control has been demonstrated and validated at smaller scales. Companies looking at designing control software or operating systems for quantum computers have tended to invest development time into schemes focused primarily around superconducting qubit based quantum computers due to the platform being

ISSN: 2414-1895 DOI: 10.6919/ICJE.202601_12(1).0010

more developed. The software control of a photonic quantum computer is unlikely to be feasible within the next 5 years.

References

[1] Gerald Bronstrup, Norbert Jahr, Christian Leiterer, Andrea Cs´aki, Wolfgang Fritzsche, and Silke Christiansen. Optical properties of individual silicon nanowires for photonic devices. ACS nano,4(12):7113–7122, 2010.

[2] Jun Chen, Zachary H Levine, Jingyun Fan, and Alan L Migdall. Frequency-bin entangled comb of photon pairs from a silicon-on-insulator micro-resonator. Optics Express, 19(2):1470–1483,2011.

[3] Erman Engin, Damien Bonneau, Chandra M Natarajan, Alex S Clark, Michael G Tanner, Robert H Hadfield, Sanders N Dorenbos, Val Zwiller, Kazuya Ohira, Nobuo Suzuki, et al. Photon pair generation in a silicon micro-ring resonator with reverse bias enhancement. Optics express, 21(23):27826–27834, 2013.

[4] Lan-Tian Feng, Ming Zhang, Zhi-Yuan Zhou, Ming Li, Xiao Xiong, Le Yu, Bao-Sen Shi, Guo-Ping Guo, Dao-Xin Dai, Xi-Feng Ren, et al. On-chip coherent conversion of photonic quantum entanglement between different degrees of freedom. Nature communications, 7(1):1–7, 2016.

[5] Pisek Kultavewuti, Eric Y Zhu, Xingxing Xing, Li Qian, Vincenzo Pusino, Marc Sorel, and J Stewart Aitchison. Polarization-entangled photon pair sources based on spontaneous four wave mixing assisted by polarization mode dispersion. Scientific reports, 7(1):1–10, 2017.

[6] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy Lloyd O'Brien. Quantum computers. Nature, 464(7285):45–53, 2010.

[7] Youn Seok Lee, Mengyu Xie, Ramy Tannous, and Thomas Jennewein. Sagnac-type entangled photonsource using only conventional polarization optics. Quantum Science and Technology, 6(2):025004,2021.

[8] CS Rizal and B Niraula. Compact si-based asymmetric mzi waveguide on soi as a thermo-optical switch. Optics Communications, 410:947–955, 2018.