学位論文

# Quantum random access memory utilizing discrete-time and continuous-time quantum walks

(離散時間及び連続時間量子ウォークを用いた量子ランダムアクセスメモリ)

2024 年 3 月

Ryo Asaka

（浅香　諒）

# Quantum random access memory utilizing discrete-time and continuous-time quantum walks

Ryo Asaka[1]

*Department of Physics, Tokyo University of Science,*
*Kagurazaka 1-3, Shinjuku-ku, Tokyo 162-8601, Japan*

March 1, 2024

[1]E-mail: 1221702@ed.tus.ac.jp

## Abstract

Utilizing discrete-time and continuous-time quantum walks, we propose a quantum random access memory (qRAM) defined on a full binary tree. Our qRAM comprises $2^n$ memory units storing data, which are located at the leaves of the binary tree. This device retrieves $O(2^n)$ data from these units in the form of quantum superposition through the binary tree. First, we formulate a discrete-time quantum walk on the full binary tree to develop the data retrieval algorithm. The walker, responsible for a register retrieving the data, moves along the binary tree in discrete-time intervals through unitary transformations. These transformations involve flipping the walker's coin and then shifting its position based on the outcome of the coinflip. Second, we introduce a continuous-time quantum walk with two-level internal states to physically implement the above algorithm, which is applicable to either bosonic or fermionic particles with internal states. The register is implemented by dual-rail encoding: the qubit is represented by which of the two parallel rails the particle passes through. Coinflipping is realized by applying a localized external field to the internal states of the particle. Meanwhile, position shifting is achieved through scatterings of the particles, which depends on their internal states.

Our qRAM can retrieve $O(2^n)$ $m$-qubit data in the form of a quantum superposition using only $n + m$ qubit resources, which are given by dual-rail encoded particles. Furthermore, our architecture achieves the retrievement with a circuit depth of $O(n \log(n + m))$. These are more efficient than the conventional bucket-brigade qRAM, which requires $O(2^n + m)$ qubit resources and $O(2^n + nm)$ time steps. A crucial aspect of utilizing discrete-time and continuous-time quantum walks is eliminating the need for time-dependent controls; data retrieval is completed simply by the particles passing through the architecture.

# Contents

# Chapter 1

# Introduction

The purpose of this doctoral thesis is to theoretically demonstrate that both discrete-time and continuous-time quantum walks can make significant contributions to quantum random access memory (qRAM: quantum RAM), which combines the insights from three of the author's works [1–3]. In this introduction, we first present an overview of these two types of quantum walks. Second, we provide the concept of the conventional qRAM, highlighting the issues associated with its realization. Finally, we outline the contents of this thesis, briefly explaining how the application of quantum walks can address and resolve the issues of qRAM.

## 1.1 Overview of the discrete-time and continuous-time quantum walks

Quantum walks are foundational frameworks based on quantum mechanical principles, making significant contributions to the field of quantum information. These contributions are mainly to the proposal of quantum algorithms, but also to physical implementations of quantum computing in recent years. Here, researchers broadly classify the quantum walks into two types: the discrete-time quantum walk [4] and the continuous-time quantum walk [5]. Although close relationships between these two quantum walks have been revealed, e.g., the discrete-time quantum walk can imitate the continuous-time one [6, 7], each quantum walk originates from a completely different context.

The discrete-time quantum walk, a quantum counterpart to the classical random walk, is an iteration of two unitary transformations: coinflipping and position shifting. A quantum walker of this model has a *coin state*, which is initially either $|0\rangle_{\mathrm{C}}$ or $|1\rangle_{\mathrm{C}} \in \mathbb{C}^2$. By the coinflipping acting on the walker, the walker's coin state changes to another coin state. Subsequently, by the position shifting, the walker moves to the left $(+1)$ or right $(-1)$ in a one-dimensional space, based on its coin state $|0\rangle_{\mathrm{C}}$ or $|1\rangle_{\mathrm{C}}$, respectively. In contrast to the classical random walk, the quantum walker can shift (take) both $+1$ and $-1$ as a quantum superposition because its coin state can be a quantum superposition of both $|0\rangle_{\mathrm{C}}$ and $|1\rangle_{\mathrm{C}}$.

The faster distribution property compared to the classical quantum walk, arising from the quantum superposition of the walker's coin states and the interference between these states at the same position, results in quantum speed-up for some problems, such as search and element distinctness problems for a desired value [8–12]. Additionally, the emergence of discrete-time quantum walks, defined on various structures, suggests a wide range of applications for this quantum walk: not only in one-dimensional space, but also in, for example, multi-dimensional spaces [13], hypercubes [14, 15], graphs [16], and complex networks [17] (furthermore, in the full binary tree proposed in this thesis).

In contrast, the continuous-time quantum walk is an evolution over a continuous-time $t$ described by $e^{-i\mathcal{A}t}$; the walker moves on a graph whose structure can be represented by an adjacency matrix $\mathcal{A}$ of the graph. Originally, this quantum walk is designed to examine whether a quantum speed-up is possible in a decision problem where decisions and their outcomes are represented by, respectively, edges and vertices of a binary tree graph [5]. Whereas the outcome is negative (as there exists a classical counterpart as fast as the continuous-time quantum walk [5]), using this quantum walk offers an exponential speed-up in black box graph traversal problems [18, 19], and a quadratic speed-up in spatial search problems [20–23].

One can also employ the continuous-time quantum walk as a model of a particle moving and scattering on a graph-like structure with its evolution described by $e^{-i\mathcal{A}t}$, because the adjacency matrix $\mathcal{A}$ representing this structure can be Hermitian. A previous study has then proposed a physical implementation of universal quantum computation [24]. Here, this proposal employs only one quantum walker (particle) and represents the $n$-qubit state, $|0...00\rangle, |0...01\rangle, ..., |1...11\rangle$, by the position of the walker (Namely, the space complexity scales $O(2^n)$ for the $n$-qubit circuit). For the computation, this particle then changes its qubit state by the single-particle scatterings on a graph-based architecture. Inspired by this study, discrete-time quantum walk approaches for universal quantum computation have also been proposed [25–27].

Moreover, a physical implementation for universal quantum computation employing the

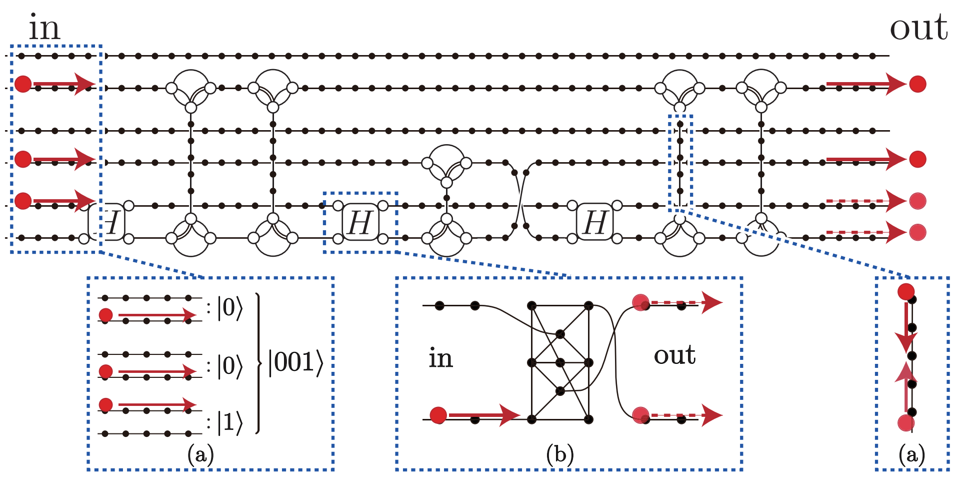
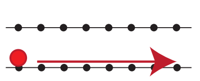
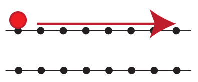


Figure 1.1: A conceptual image of the universal quantum computation using multi-particle continuous-time quantum walk, proposed by A. M. Childs et al. [28]. The computation is completed by the particles passing and scattering through the graph-based architecture. (a) Binary digits, e.g., $|001\rangle$, are represented by the three dual-rail encoded particles (1.1.1). (b) The single-qubit gate, e.g., Hadamard gate $H$, is implemented by single-particle scattering on a subgraph. (c) A two-qubit gate is implemented by utilizing two-particle head-on scattering along a rail.

multi-particle continuous-time quantum walk has been proposed [28], which is one of the main inspirations for this thesis. As shown in Fig. 1.1, this proposal utilizes single- and two-particle scatterings on a graph-based architecture for the computation.

This quantum computation using a multi-particle continuous-time quantum walk results in the following two benefits. The first benefit is the efficient utilization of architecture size; the architecture required for $n$ qubits scales only $O(n)$ in size, whereas the above conventional methods, employing a single continuous-time or discrete-time quantum walker, necessitate an architecture size of $O(2^n)$ [24–27]. The second benefit is that this implementation requires no time-dependent control. Namely, particles simply passing and scattering through the graph-based architecture complete the computation automatically as well as the computation using the single-particle continuous-time quantum walk. Note that, in this computation, each of $n$ particles serves as each of $n$ qubits by the dual-rail encoding. Namely, a binary digit (qubit value) either 0 or 1 is represented based on which rail a particle passes in two parallel rails:

$|0\rangle :=$  , $|1\rangle :=$  , (1.1.1)

(see Fig. 1.1 (a) for an example of three qubits).

Experimental realizations of the quantum walks using photons [29, 30], atoms [31–34], trapped ions [35–37], and superconducting processors [23] suggest the feasibility of realizing such a quantum computation. Note that the above implementation of the universal quantum computation using the multi-particle continuous-time quantum walk can be applicable to either bosonic or fermionic particles [28, 38].

## 1.2 Overview of the quantum random access memory

The purpose of this thesis is to theoretically demonstrate that both discrete-time and multi-particle continuous-time quantum walks can make significant contributions to quantum random access memory (qRAM: quantum RAM) [39, 40] as well, in terms of both developments of an algorithm for qRAM and its physical implementation. Note that the contribution of the quantum walks to the qRAM is detailed in the next part of this introduction.

The qRAM, not yet experimentally demonstrated, is an essential component to achieve quantum speed-ups in various quantum algorithms, such as database search [41–43], machine learning [44, 45], Hamiltonian simulations [46–48], and fast Fourier Transform [49], over their classical counterparts. Concretely, one employs the qRAM to prepare a quantum superposition of $O(2^n)$ data which these algorithms require as a prerequisite input.

The qRAM comprises $2^n$ memory units storing $m$-qubit data classically, which are located at the leaf nodes of the binary tree (see fig. 1.2 for the conceptual image). Through the binary tree, we can retrieve the desired $O(2^n)$ $m$-qubit data from the corresponding memory units as a response to memory access in a superposition:

$$\text{qRAM} : \sum_a |a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \mapsto \sum_a |a\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}}, \tag{1.2.1}$$

where $0 \leq a \leq 2^n - 1$ denotes the memory unit and $x^{(a)}$ is the data recorded in the $a$th unit. The subscripts $A$ and $D$ denote the address and data register allocated $n$ and $m$ qubits, respectively. The address and data are then represented as binary digits: $a = a_{n-1}...a_1a_0$ and $x^{(a)} = x^{(a)}_{m-1}\cdots x^{(a)}_1 x^{(a)}_0$ where $a_q,\ x^{(a)}_q \in \{0, 1\}$ for $q \geq 0$.

By using the qRAM, we can retrieve the superposition of data within time steps of $O(n(n+m))$ in general, and thus resolve the bottleneck for the quantum speed-up associated with the preparation of the superposition of data in the aforementioned quantum algorithms. Through the binary tree, the qRAM retrieves $O(2^n)$ data in parallel (the time steps originate from the number of qubits, $n+m$, and the depth of the binary tree, $n$). As a counter-example of the speed-up, if one constructs the superposition by retrieving $O(2^n)$ data from the memory units not in parallel but one by one, which would require at least $O(2^n)$ time steps, these quantum algorithms can not achieve a speed-up beyond $O(2^n)$ time steps in total.

The main issue for the realization of qRAM is that the implementation requires $2^{n-1} - 1$ ancillary qubit resources and time-dependent controls to $O(2^n)$ of these resources for retrieving $O(2^n)$ data. This is a trade-off for the exponential saving of the time steps compared to the above counter-example by utilizing quantum parallelization. Here, in this thesis, the word *qubit resources* signifies devices and matters that exhibit superposition and entanglement.

Concretely, conventional proposals for the ordinary qRAM require $2^{n-1} - 1$ *quantum switches* as ancillary qubit resources. Each of these switches has three energy levels denoted as $|wait\rangle$, $|left\rangle$, and $|right\rangle \in \mathbb{C}^3$ (or two energy levels: $|left\rangle$, and $|right\rangle \in \mathbb{C}^2$).

As shown in Fig. 1.3, the quantum switches are installed on the nodes other than leaf nodes of the binary tree, $2^{n-1} - 1$ nodes in total. The switches create the routes that direct the data register from the root node to the desired memory units in a superposition. Here, a switch in a quantum superposition of the $|left\rangle$ and $|right\rangle$ passes the data register to both the left and right adjacent nodes as the superposition, leading to the parallelization of the memory access and data retrievement. Note that we will review in detail a common process,

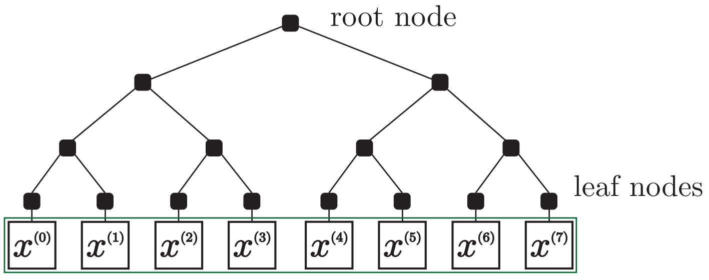


Figure 1.2: Binary tree and memory units. Here, $2^n$ leaf nodes are connected to the $2^n$ memory units, respectively. In the $a$th unit, data $x^{(a)}$ is recorded.

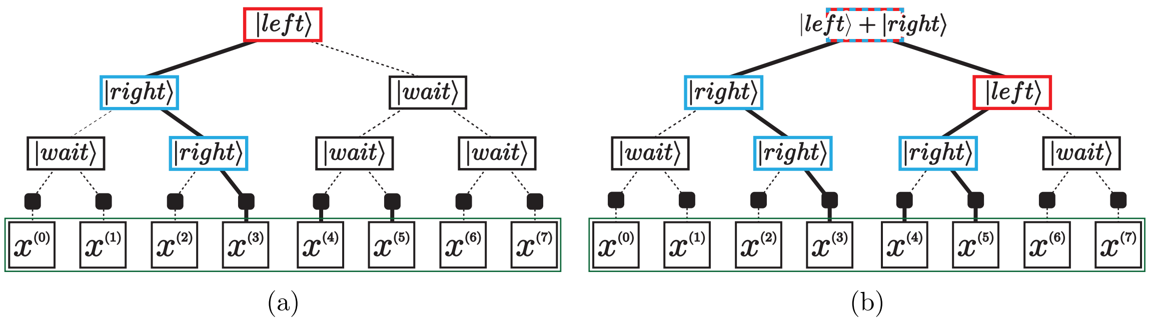


Figure 1.3: Conceptual image of qRAM employing the bucket-brigade process (see Sec 2 for a detailed review of this process). Each node is equipped with the quantum switches that have three energy levels: $|wait\rangle$, $|left\rangle$, and $|right\rangle \in \mathbb{C}^3$. (a) A data qubit accesses the desired unit by the switches directing the qubit to move sequentially, e.g., left node, right, and then right again to the 3rd unit. (b) When certain switches are set to be in a superposition $|left\rangle + |right\rangle$, the data qubit accesses multiple units, e.g., the 3rd and 5th, simultaneously as a spatial superposition.

which is known as the *bucket-brigade* process, for creating the routes and sending the data register along these routes in Chap. 2.

One must maintain the superposition of $O(2^n)$ switches along the created routes to the desired units throughout the retrieval process, because these switches become entangled with the output, i.e., the superposition of $O(2^n)$ data. Additionally, for resolving the entanglements, we need the post-processing to reset the states of all switches to $|wait\rangle$ after the retrieval of the superposition.

Note that, in the bucket-brigade process, the address and data registers become entangled with practically only a logarithmically small number of switches compared to the total number of activated switches, thereby offering substantial resilience to noise affecting the switches. Namely, the address and data registers are partially entangled with only $O(n)$ switches for the retrieving $O(2^n)$ data. Infidelity of the superposition of $O(2^n)$ single qubit data thus scales only with $O(n^2)$, even with arbitrary error channels [50].

However, as reviewed and discussed in Chap. 2, we must dedicate effort to time-dependent controls. Namely, to retrieve the $O(2^n)$ data, we need pre-processing to activate the $O(2^n)$ switches from $|wait\rangle$ to $|left\rangle$ or $|right\rangle$. Additionally, post-processing is required to deactivate the switches and resolve the entanglement between these switches and the output.

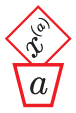
## 1.3 Overview of this thesis

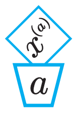
In this thesis, we theoretically propose a qRAM utilizing both discrete-time and continuous-time quantum walks. Characteristically our proposal eliminates the need for $O(2^n)$ qubit resources, specifically all $2^{n-1}-1$ quantum switches, for retrieving a superposition of $O(2^n)$ $m$-qubit data and then any time-dependent control. Pre- or post-processing is not required before or after the retrieval.

Concretely, our qRAM requires only $n+m$ qubit resources and simply passing these qubits through an architecture is sufficient to automatically retrieve the superposition of data. We employ the discrete-time quantum walk to develop an algorithm, an alternative to the bucket-brigade process reviewed in the next chapter, for retrieving data stored in the memory. We then use multi-particle continuous-time quantum walk to propose a physical implementation of qRAM adopting our retrieval algorithm.

As shown in Table. 1.1, although our proposal requires larger space and more quantum gates compared to the bucket-brigade process, we reduce the required qubit resources and time steps to $n+m$ and $O(n\log(n+m))$, respectively. Moreover, the elimination of the need for any time-dependent control is also a significant aspect of our proposal; simply passing the registers through the architecture completes a retrieval process for the superposition of data, automatically.

First, we formulate a discrete-time quantum walk on the full binary tree to propose a data retrieval algorithm. This algorithm uses the walker as the set of address and data registers:

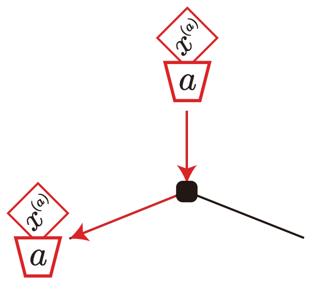
$$|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}} \equiv$$ 

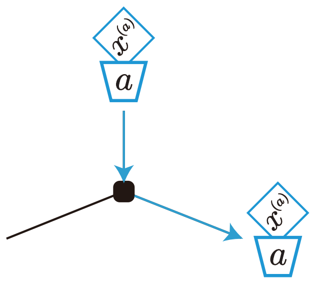
$$, \quad |1\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}} \equiv$$ 

. (1.3.1)

Here, we represent the walker as the bucket with data, where the walker's coin states $|0\rangle_{\mathrm{C}}$ and $|1\rangle_{\mathrm{C}}$ are represented by the colors red and blue, respectively.

In our formulation, the walker (bucket with data) shifts to either the left or right path at the bifurcation point based on its coin states (colors):



,


. (1.3.2)

Although we represent the quantum walker as the bucket with data, our algorithm for the qRAM is completely different from the bucket-brigade method where the registers are carried by the quantum switches.

By utilizing the colors, we eliminate the need for the quantum switches for sending the registers to the desired location on the binary tree. In our process, the bucket with data serving as the registers iteratively changes its color at the bifurcation points of the binary tree to select the left or right path, depending on this color, appropriately. The color change,

| Method | #qubits | #time steps | #quantum gates or spaces |
|---|---|---|---|
| quantum walks | $n+m$ | $O(n\log(n+m))$ | $O(2^n(n^2+nm)\log(n+m))$ |
| bucket-brigade | $O(2^n)+m$ | $O(n(n+m))$ | $O(n2^n)$ |

Table 1.1: Comparison of the number of qubit resources, time steps, quantum gates, and architecture spaces between our proposal, which employs quantum walks, and the bucket-brigade qRAM. Here, in this thesis, the word *qubit resources* signifies devices and matters that exhibit superposition and entanglement.

i.e., coin flipping of the quantum walker, can be implemented by devices installed on each bifurcation point, whereas the conventional process, the bucket-brigade, installs the quantum switch on each bifurcation to guide the registers to the desired location. Note that each of these devices itself does not exhibit quantum superposition and entanglement, and thus is not a qubit resource.

We can implement our algorithm without any need for time-dependent control, partly because our algorithm does not manage the ancillary qubit resources, such as the bucket-brigade process which requires activating the corresponding quantum switches from $|wait\rangle$ to $|left\rangle$ or $|right\rangle$, and maintaining and deactivation these states during and after the data retrieval. Intuitively, simply passing the registers through an architecture is sufficient to complete the data retrieval.

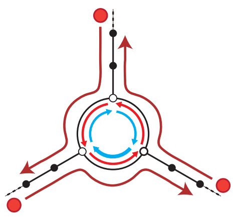
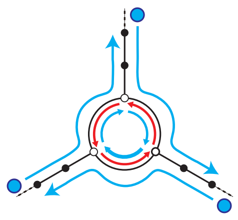
Second, we formulate a multi-particle continuous-time quantum walk with two-level internal states to propose a physical implementation of qRAM, inspired by the aforementioned universal quantum computation [28]. The motivation for introducing the internal states is to implement the colored bucket and data (1.3.1) and the position shifting based on its color (1.3.2). Here, because the internal states correspond to the colors red and blue of the bucket with data, we represent these two states as red and blue. Furthermore, these states are denoted by $|0\rangle_{cq}$ and $|1\rangle c_q \in \mathbb{C}^2$ for the $q$th particle:

$$|0\rangle_{c_q} \equiv \text{🔴}\ , \qquad |1\rangle_{c_q} \equiv \text{🔵}\ . \tag{1.3.3}$$

As with the above universal quantum computation, we combine single- and two-particle scatterings to implement the data retrieval algorithm, thereby solidifying the algorithm's two main aspects. Namely, the retrieval algorithm is completed automatically as the particles simply pass and scatter through a graph-based architecture. The implementation still eliminates the need for any ancillary qubits and time-dependent controls.

In our formulation, which is different from the previous continuous-time quantum walk, a particle on a graph demonstrates the internal-state dependence of single-particle scatterings on a subgraph, as shown in fig 1.4. Concretely, when the particle enters a subgraph, it scatters according to either a scattering matrix or its transpose depending on its internal states. Note that we only utilize a two-particle scattering as a head-on scattering along a rail, where both particles have the same internal state.

The first key concept for utilizing the continuous-time quantum walk to implement our retrieval algorithm is introducing the roundabout gate which is used for the position shifting of the bucket with data. The roundabout gate is a subgraph demonstrating the aforementioned internal-state dependence of a scattering. This graph connects three rails and scatters an incident particle to another rail in either a counter-clockwise or clockwise direction, depending on its internal state:

 ,  . (1.3.4)

See Eq. (4.3.2) and (4.3.3) for concrete images of implementing the shift operator by the roundabout gates.

The second key concept is the employment of dual-rail encoded particles (as proposed by Childs et al. [28]). The dual-rail encoding represents the qubit value 0 or 1 based on which of the two parallel rails a particle passes through as mentioned in Eq. (1.1.1). We thus use $n+m$ particles for the dual-rail encoded qubits to represent the bucket with data (see Eq. (4.1.2) and (4.1.1) for concrete images), where both address and data consist of binary digits as

$$|a\rangle_{\mathrm{A}} = |a_{n-1}\rangle_{\mathrm{A_{n-1}}}|a_{n-2}\rangle_{\mathrm{A_{n-2}}}\cdots|a_0\rangle_{\mathrm{A_0}} \;\in \left(\mathbb{C}^2\right)^{\otimes n}, \tag{1.3.5}$$

$$|x^{(a)}\rangle_{\mathrm{D}} = |x^{(a)}_{m-1}\rangle_{\mathrm{D_{m-1}}}|x^{(a)}_{m-2}\rangle_{\mathrm{D_{m-2}}}\cdots|x^{(a)}_0\rangle_{\mathrm{D_0}} \;\in \left(\mathbb{C}^2\right)^{\otimes m}. \tag{1.3.6}$$

Here, $a_q$, $x^{(a)}_{q'} \in \{0,1\}$, and $|a_q\rangle_{\mathrm{A}_q}, |x^{(a)}_q\rangle_{\mathrm{D}_q} \in \mathbb{C}^2$ for $q \geq 0$.

In summary, simply passing the $n+m$ particles through an architecture automatically completes retrieving the superposition of $O(2^n)$ $m$-qubit data from the memory units. Here, these particles serve as the dual-rail encoded address and data qubits, and the architecture is mostly constructed from a combination of graphs. In this retrieval process, these particles move to the left rail, right rail, or both at a bifurcation point of the binary tree to access the desired memory units as the spatial superposition. This movement can be implemented by the roundabout gate designed by an appropriate graph. Each unit then passes the recorded data to the incident particle as the dual-rail encoding.

The outline of this thesis is as follows. In Chap. 2, we review the conventional retrieval algorithm, the bucket-brigade process. In Chap. 3, we introduce the discrete-time quantum walk on the full binary tree and describe the retrieval algorithm using this model. In Chap. 4, we summarize some key concepts for the implementation including the dual-rail encoding and the roundabout gate. In Chap. 5, we then formulate the continuous-time quantum walk with two-level internal states and the implementation of our retrieval algorithm. In Chap. 6, we summarize and discuss our result, supplemented with the discussion of experimental realization and the error bounds of our proposal. In the appendix, we provide technical aspects of our continuous-time quantum walk.

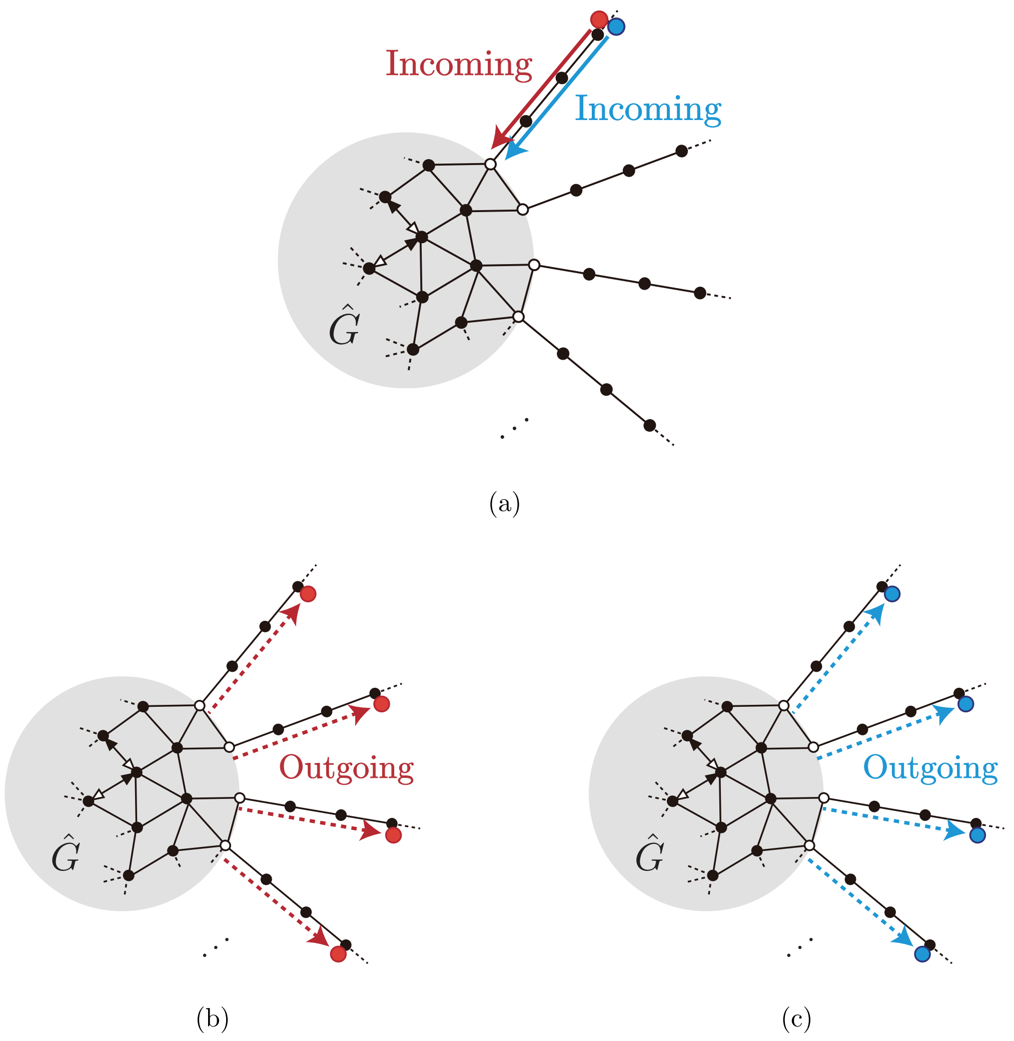


Figure 1.4: Our model, the continuous-time quantum walk with two-level internal states, demonstrates the internal state-dependence of scattering on a graph. Concretely, a single-particle scattering on the graph follows a scattering matrix or its transpose depending on the internal states. Using this dependence, we will derive the roundabout gate in Chap. 5. This gate passes the incoming particles to another rail clockwise or its transpose, i.e., counter-clockwise, depending on the internal states as shown in Eq. (1.3.4). (a) We consider a particle that enters a graph (subgraph) through a rail. Here, the particle's internal state is depicted as either red or blue. (b) (resp. (c)) The single-particle scattering of the red (resp. blue) is described by a scattering matrix (resp. its transpose). See Sec. 5.1 for a concrete discussion.

# Chapter 2

# Bucket-brigade process: the conventional retrieval process

To highlight the uniqueness and novelty of our qRAM, let us review the so-called bucket-brigade process for retrieving a superposition of data (1.2.1). This innovative process, which can be considered a standard and mainstream method in research for qRAM, was proposed by V. Giovannetti, S. Lloyd, and L. Maccone in 2008 [51]. The advantages of this method are few time steps for data retrieval and high resilience to noise. Namely, the process can retrieve a superposition of $O(2^n)$ $m$-qubit data within only $O(2^n+nm)$ time steps. Additionally, while $2^{n-1}-1$ quantum switches (qutrits) are generally required, the address and data registers become practically entangled with only $O(n)$ switches during the process.

An architecture for qRAM that supports the bucket-brigade process consists of $2^n$ memory units and a full binary tree with depth $n$, the leaf nodes of which are attached to the respective memory units (see Fig. 1.3). Through the binary tree, the bucket-brigade process retrieves $O(2^n)$ $m$-qubit data stored in the memory units in the form of quantum superposition in response to the memory call:

$$\text{qRAM} : \sum_a |a\rangle_\mathrm{A}|0\rangle_\mathrm{D} \mapsto \sum_a |a\rangle_\mathrm{A}|x^{(a)}\rangle_\mathrm{D}, \tag{2.0.1}$$

where $a$ is the assigned number to the $a$th unit and $x^{(a)}$ is the data stored in this unit.

Generally, the bucket-brigade process requires $n+m$ and $(2^{n-1}-1)$ qubit resources, each serving different purposes. The $n+m$ resources are allocated to the registers as address and data qubits:

$$|a\rangle_\mathrm{A} =|a_{n-1}\rangle_{\mathrm{A}_{n-1}}|a_{n-2}\rangle_{\mathrm{A}_{n-2}}\cdots|a_0\rangle_{\mathrm{A}_0} \in \left(\mathbb{C}^2\right)^{\otimes n}, \tag{2.0.2}$$

$$|x^{(a)}\rangle_\mathrm{D} = |x^{(a)}_{m-1}\rangle_{\mathrm{D}_{m-1}}|x^{(a)}_{m-2}\rangle_{\mathrm{D}_{m-2}}\cdots|x^{(a)}_0\rangle_{\mathrm{D}_0} \in \left(\mathbb{C}^2\right)^{\otimes m}, \tag{2.0.3}$$

where $a_q$, $x^{(a)}_q \in \{0,1\}$ for $q \geq 0$. The remaining $2^{n-1}-1$ resources are allocated to the *quantum switches*. As shown in Fig. 1.3, these quantum switches, each having three energy levels: $|wait\rangle$, $|left\rangle$, and $|right\rangle \in \mathbb{C}^3$, are installed in the nodes of the binary tree, excluding the leaf nodes, with a total of $2^{n-1}-1$ nodes.

Here, for the experimental realization of this method, the initial proposals have involved employing photons as qubits, where the value 0 or 1 is represented by the polarization, and using atoms with a three-level system as quantum switches [40, 51]. Additionally, researchers have also proposed various approaches utilizing phonons [52], photonic microchips [53], or circuit-based implementations [54–57].

Using the quantum switches, the bucket-brigade process creates a route that allows the data qubits $\{|0\rangle_{\mathrm{D}_i}\}$ to access the desired memory unit and retrieve data stored in this unit. Note that this route corresponds to the binary digits of the address value. For example, to access the 3rd of memory units, which corresponds to the binary digits $a_2a_1a_0 = 011$ $(=3)$, the register moves from the root node toward this unit: first left (corresponding to the 2nd digit “0”), then right (the 1st digit “1”), and finally right again (the 0th digit “1”), as shown in Fig. 1.3 (a).

Initially, all quantum switches are in the $|wait\rangle$ states, and we change the states of corresponding switches by sequentially inputting the $n$ address qubits of the address register (1.3.5) into the binary tree through the root node, starting with the $(n-1)$th address qubit. Assume here that a quantum switch interacts with an incoming qubit either absorbing it or moving it to an adjacent node on the left or right. At a node, if a quantum switch is $|wait\rangle$ and the incoming qubit is $|0\rangle_{\mathrm{A}_q}$ (resp. $|1\rangle_{\mathrm{A}_q}$), the switch absorbs this qubit and activates its state to $|left\rangle$ (resp. $|right\rangle$) through a unitary transformation $U$. In contrast, if a quantum switch is $|left\rangle$ (resp. $|right\rangle$), the switch passes the incoming qubit to the left (resp. right) adjacent node by the same unitary transformation $U$.

For instance, as shown in Fig. 1.3 (a), sequentially inputting address qubits in the state $|011\rangle_\mathrm{A}$, starting from the final (2nd) qubit $A_2$, carves the route to the 3rd memory unit. As another example, which is shown in Fig. 1.3 (b), sequentially inputting address qubits with a superposition of $(|011\rangle + |101\rangle)_\mathrm{A}$ carves a route to 3rd and 5th memory units. Note that sending and absorbing each address qubit requires $O(n)$ time steps due to the depth of the binary tree being $n$, in total $O(n^2)$ time steps for $n$ address qubits.

After establishing the routes, we query the desired memory units by sequentially inputting the $m$ data qubits. In this inputting scheme, by $U$, the quantum switches along a route pass the data qubit to the left, right, or both adjacent switches depending on whether the activated state is $|left\rangle$, $|right\rangle$, or a superposition of both, respectively. Each qubit, upon reaching the desired memory unit, changes its state from $|0\rangle_{\mathrm{D}_q}$ to $|1\rangle_{\mathrm{D}_q}$ by the NOT (Pauli-X) gate if the $q$th digit of data is 1, which is recorded within the unit. Subsequently, the data qubit returns to the root node through the inverse transformation $U^\dagger$ of $U$, and then we input the next data qubit. In summary, the repeatedly inputting, querying, and returning results in the superposition of data as Eq. (2.0.1). Here, $O(nm)$ time steps for the $m$-qubit data are required because a query per bit requires $O(n)$ steps which also originates from the depth of the tree.

Finally, we must execute post-processing to initialize the architecture. Namely, we recover absorbed all address qubits from the quantum switches using $U^\dagger$ to resolve the entanglement between the quantum switches and the register. This process necessitates additional $O(n^2)$ time steps.

Whereas the bucket-brigade process needs to activate $O(2^n)$ switches each from $|wait\rangle$ to $|right\rangle$ or $|left\rangle$ to retrieve $O(2^n)$ data as a superposition, the registers become practically entangled with only $O(n)$ switches during the process. Indeed, some studies have stated and proved the noise resilience of the bucket-brigade process [40, 50, 54, 58]. For example, the previous study [50] has proved that one can achieve a query, i.e., retrieving $O(2^n)$ single qubit data from the memory units as a quantum superposition after the carving routes, with infidelity which is almost given by

$$\sum_{l=1}^{n}(2^{-l})(\varepsilon T 2^{l}) = \varepsilon T n, \tag{2.0.4}$$

within time steps $T = O(n)$.

However, we must dedicate effort to time-dependent controls for the quantum switches. The first is the pre-processing to activate the switches for carving the routes toward the desired memories. The second is the post-processing to recover the absorbed qubits from the switches to resolve the entanglement between the $O(2^n)$ activated switches and the output, i.e., the superposition of $O(2^n)$ data.

Chapter 3

# Data retrieval algorithm via discrete-time QW [1]

In this chapter, we propose a novel retrieval algorithm via a discrete-time quantum walk. First, we formulate the discrete-time quantum walk on a full binary tree. Second, we develop the retrieval algorithm requiring only $n+m$ qubit resources for the $O(2^n)$ $m$-qubit data, where the quantum walker is employed as the address and data registers. This algorithm utilizes the coin states of the walker to eliminate the need for $2^{n-1}-1$ qubit resources (quantum switches) required in the conventional bucket-brigade process. We also show that the algorithm can eliminate the need for any time-dependent control. Namely, simply passing the quantum walker, i.e., the registers, through an architecture for our algorithm is sufficient to retrieve the data as a superposition.

## 3.1 Discrete-time quantum walk on a binary tree

To derive the retrieval algorithm that requires no quantum switches, we formulate a discrete-time quantum walk on a full binary tree. As explained in the next chapter, the walker serves as a set of address and data registers, where the walker with a coin state $|0\rangle_{\mathrm{C}}$ and $|1\rangle_{\mathrm{C}}$ are denoted by the red and blue buckets with data, respectively.

The walker moves on the binary tree toward the desired memory units by a unitary transformation, referred to as *shift operator*. Here, this operator relies on the coin state (color) to guide the set of registers, whereas the unitary transformation $U$ in the bucket-brigade process relies on the quantum switch to guide the registers in the desired direction, This distinction leads to the elimination of the need for $O(2^n)$ qubit resources.

As a preface, let us refer to a path constructing the full binary tree as *bus*. As shown in Fig. 3.1, the symbol $r^{(l,w)}$ indicates the $w$th bus from the left, located on the $l$th level of the binary tree, for $0 \le l \le n-1$ and $0 \le w \le 2^1 - 1$. Here, the term bus is used because it resembles a data bus in computing, where data is transferred through it.

Let us denote the state of the walker with the coin $|c\rangle_{\mathrm{C}}$ ($c \in \{0,1\}$) located on the bus $r^{(l,w)}$ on the binary tree as

$$|r^{(l,w)}\rangle_{\mathrm{B}}|c\rangle_{\mathrm{C}} \ \in V_{\mathrm{B}} \otimes V_{\mathrm{C}},\ c \in \{0,1\}. \tag{3.1.1}$$

Here, $V_{\mathrm{C}}$, *coin space*, is the Hilbert space spanned by $\{|c\rangle_{\mathrm{C}} | c \in \{0,1\}\}$, and $V_{\mathrm{B}}$, *bus space*, is spanned by $\{|r^{(l,w)}\rangle_{\mathrm{B}} | 0 \le l \le n-1, 0 \le w \le 2^l - 1\}$.

Now we introduce the shift operator $\mathcal{S}^{(l-1,w)}_{\mathrm{B,C}}$ that depicts the scattering of the walker incident into the bus, which is at the $(l-1)$th level and $w$th from the left on the binary tree:

$$\mathcal{S}^{(l-1,w)}_{\mathrm{B,C}} := \tilde{\mathcal{S}}^{(l-1,w)}_{\mathrm{B}} \otimes |0\rangle\langle 0|_{\mathrm{C}} + \tilde{\mathcal{S}}^{(l-1,w)\top}_{\mathrm{B}} \otimes |1\rangle\langle 1|_{\mathrm{C}} \ \in \mathrm{End}\left(V_{\mathrm{B}} \otimes V_{\mathrm{C}}\right), \tag{3.1.2}$$

$$\tilde{\mathcal{S}}^{(l-1,w)}_{\mathrm{B}} := |r^{(l,2w)}\rangle\langle r^{(l-1,w)}|_{\mathrm{B}} + |r^{(l,2w+1)}\rangle\langle r^{(l,2w)}|_{\mathrm{B}} + |r^{(l-1,w)}\rangle\langle r^{(l,2w+1)}|_{\mathrm{B}} \ \in \mathrm{End}\left(V_{\mathrm{B}}\right). \tag{3.1.3}$$

By the operator $\mathcal{S}^{(l-1,w)}_{\mathrm{B,C}}$, our walker scatters to the counter-clockwise adjacent bus if its coin is $|0\rangle_{\mathrm{C}}$, following the operator $\tilde{\mathcal{S}}^{(l-1,w)}_{\mathrm{B}}$. In contrast, the walker with $|1\rangle_{\mathrm{C}}$ follows its transpose $\tilde{\mathcal{S}}^{(l-1,w)\top}_{\mathrm{B}}$, scattering to the clockwise adjacent bus.

For example, if the coin state of the walker on the bus $r^{(l-1,w)}$ is the superposition of $|0\rangle_{\mathrm{C}}$ and $|1\rangle_{\mathrm{C}}$, this walker moves to both the left and right adjacent buses on $l$th level, creating a spatial superposition:

$$\mathcal{S}^{(l-1,w)}_{\mathrm{B,C}} : |r^{(l-1,w)}\rangle_{\mathrm{B}} \left(|0\rangle_{\mathrm{C}} + |1\rangle_{\mathrm{C}}\right) \to |r^{(l,2w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}} + |r^{(l,2w+1)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{C}}. \tag{3.1.4}$$

In our algorithm, the walker evolves in discrete time by iteratively flipping its coin state and shifting its position based on the outcome of the coinflip. To achieve this, we introduce a controlled coinflip operator denoted as $\mathcal{X}^{(l-1,w)}_{\mathrm{C,A}}$, and coinflip operator denoted as $\mathcal{X}^{(l,2w+1)}_{\mathrm{C}}$ in the following chapter.

## 3.2 Exponential qubit savings

Applying the discrete-time quantum walk on the binary tree, we propose a novel retrieval algorithm that saves qubit resources exponentially compared to the bucket-brigade process.

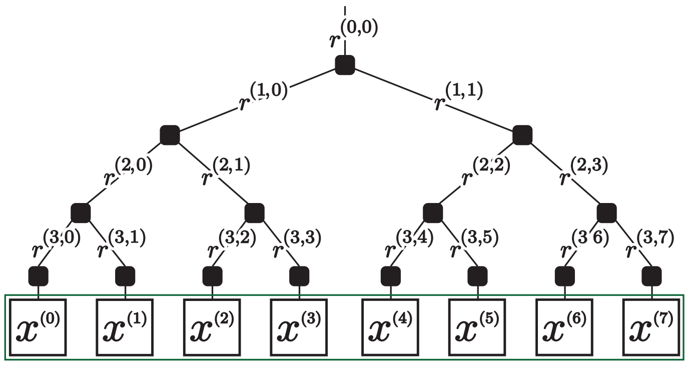
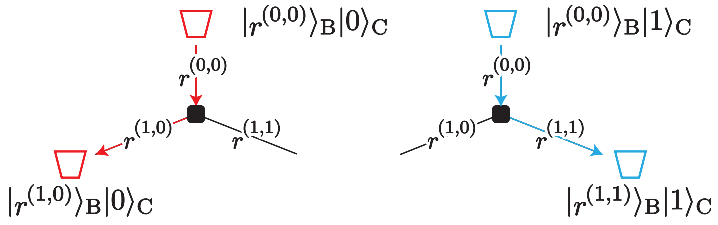


Figure 3.1: Buses $\{r^{(l,w)}\}$. The symbol $r^{(l,w)}$ represents the bus on the $l$th level and is the $w$th from the left on this level. We use the term *bus* in reference to the "memory bus".



$|r^{(0,0)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}$
$r^{(0,0)}$
$r^{(1,0)}$
$r^{(1,1)}$
$|r^{(1,0)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}$
$|r^{(0,0)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{C}}$
$r^{(0,0)}$
$r^{(1,0)}$
$r^{(1,1)}$
$|r^{(1,1)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{C}}$

Figure 3.2: Shift operation. Our walker scatters to another bus either in the counter-clockwise or clockwise direction, based on its coin (color) state, by the shift operator (3.1.2).

We only require $n+m$ qubit resources allocated to the walker (bucket with data) which serves as the set of address and data registers. While the primary motivation for the algorithm is saving qubit resources exponentially, removing the need for any time-dependent control is also a significant aspect. Namely, by the walker, i.e., the set of registers, simply passing through an architecture, the data retrieval process is completed.

Unlike the conventional approach, our algorithm uses an architecture with two binary trees symmetrically connected on both sides of the memory, as shown in Fig. 3.3. A discrete-time quantum walker traverses one side of the trees to access the desired memory units as spatial superposition, which we refer to as *routing*. After reaching the units, the walker queries the data within the units, *querying*. Subsequently, the walker traverses the other side of the trees, and, finally, we obtain the superposition of data from the root bus $r^{(0,0)}$ of this tree, *outputting*.

Specifically, our algorithm consists of three steps: routing $\mathcal{R}$, querying $\mathcal{Q}$, and outputting $\mathcal{R}^\dagger$ as

$$\sum_a |r^{(0,0)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \xrightarrow{\mathcal{R}} \sum_a |r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \tag{3.2.1}$$

$$\xrightarrow{\mathcal{Q}} \sum_a |r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}} \xrightarrow{\mathcal{R}^\dagger} \sum_a |r^{(0,0)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}}, \tag{3.2.2}$$

where the walker, serving as the bucket with data, is defined on the tensor product of the four spaces (the bus, coin, address, and data spaces) $V_{\mathrm{B}} \otimes V_{\mathrm{C}} \otimes V_{\mathrm{A}} \otimes V_{\mathrm{D}}$. Here, the address and data space are the Hilbert spaces spanned by the $n$ and $m$-qubit states, as shown in Eq. (1.3.5) and (1.3.6), i.e., $V_{\mathrm{A}} = (\mathbb{C}^2)^{\otimes n}$ and $V_{\mathrm{D}} = (\mathbb{C}^2)^{\otimes m}$, respectively. The $V_{\mathrm{C}}$ and $V_{\mathrm{B}}$ are the coin and bus spaces, respectively, as described below in Eq. (3.1.1).

First, the routing operation $\mathcal{R}$ is defined as

$$\mathcal{R} := \prod_{l=1}^{n} \left( \sum_{w=0}^{2^{l-1}-1} \mathcal{X}_{\mathrm{C}}^{(l,2w+1)}\ \mathcal{S}_{\mathrm{B,C}}^{(l-1,w)}\ \mathcal{X}_{\mathrm{C,A}}^{(l-1,w)} \right) \in \mathrm{End}(V_{\mathrm{B}} \otimes V_{\mathrm{C}} \otimes V_{\mathrm{A}}), \tag{3.2.3}$$

$$\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)} := |r^{(l-1,w)}\rangle\langle r^{(l-1,w)}|_{\mathrm{B}} \otimes \mathcal{X}_{\mathrm{C}} \otimes |1\rangle\langle 1|_{A_{(n-1)-l}} \in \mathrm{End}(V_{\mathrm{B}} \otimes V_{\mathrm{C}} \otimes V_{\mathrm{A}}), \tag{3.2.4}$$

$$\mathcal{X}_{\mathrm{C}}^{(l,2w+1)} := |r^{(l,2w+1)}\rangle\langle r^{(l,2w+1)}|_{\mathrm{B}} \otimes \mathcal{X}_{\mathrm{C}} \in \mathrm{End}(V_{\mathrm{B}} \otimes V_{\mathrm{C}}), \tag{3.2.5}$$

where $\mathcal{X}_{\mathrm{C}}$ is the Pauli-X acting on the coin, i.e., $\mathcal{X}_{\mathrm{C}} := |0\rangle\langle 1|_{\mathrm{C}} + |1\rangle\langle 0|_{\mathrm{C}} \ \in \mathrm{End}(V_{\mathrm{C}})$. We refer to the two operators $\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}$ and $\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$ as the *coinflip* and *controlled coinflip*, respectively.

Let us detail the shifting of the walker on $w$th bus on level $l-1$ to the adjacent two bus on $l$, which is described by $\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}\mathcal{S}_{\mathrm{C,B}}^{(l-1,w)}\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$ as in Eq. (3.2.3). By the controlled coinflip $\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$ the walker flips its coin to $|1\rangle_{\mathrm{C}}$ only if the $[(n-1)-l]$th address is $|1\rangle_{\mathrm{A}_{(n-1)-l}}$:

$$\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)} : |r^{(l-1,w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \mapsto |r^{(l-1,w)}\rangle_{\mathrm{B}}|a_{(n-1)-l}\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}. \tag{3.2.6}$$

By the shift operator $\mathcal{S}_{\mathrm{C,B}}^{(l-1,w)}$, the walker moves from the bus $r^{(l-1,w)}$ to $r^{(l,2w+a_{(n-1)-l})}$:

$$\mathcal{S}_{\mathrm{C,B}}^{(l-1,w)} : |r^{(l-1,w)}\rangle_{\mathrm{B}}|a_{(n-1)-l}\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \mapsto |r^{(l,2w+a_{(n-1)-l})}\rangle_{\mathrm{B}}|a_{(n-1)-l}\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}. \tag{3.2.7}$$

By the coinflip $\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}$, the coin state is initialized to $|0\rangle_{\mathrm{C}}$ regardless of whether the walker moves to the left or right adjacent bus:

$$\mathcal{X}_{\mathrm{C}}^{(l,2w+1)} : |r^{(l,2w+a_{(n-1)-l})}\rangle_{\mathrm{B}}|a_{(n-1)-l}\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \mapsto |r^{(l,2w+a_{(n-1)-l})}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}. \tag{3.2.8}$$

The summand in Eq. (3.2.3) thus represents shifting the walker, which exists on buses at level $l-1$ as a spatial superposition, to desired buses on the level $l$:

$$\sum_{w=0}^{2^{l-1}-1} \mathcal{X}_{\mathrm{C}}^{(l,2w+1)}\ \mathcal{S}_{\mathrm{B,C}}^{(l-1,w)}\ \mathcal{X}_{\mathrm{C,A}}^{(l-1,w)} : \sum_{a} |r^{(l,\ \sum_{j=0}^{l-1} 2^{(l-1)-j} a_{(n-1)-j})}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \tag{3.2.9}$$

$$\mapsto \sum_{a} |r^{(l+1,\ \sum_{j=0}^{l} 2^{l-j} a_{(n-1)-j})}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}. \tag{3.2.10}$$

Note that in the routing process $\mathcal{R}$, the walker with an address $0 \leq a \leq 2^n - 1$ passes through the buses $r^{(0,0)}$, $r^{(1,a_{n-1})}$, $r^{(2,\ 2a_{n-1}+a_{n-2})}$, $r^{(3,\ \sum_{i=0}^{2} 2^{2-i} a_{(n-1)-i})}$, ..., $r^{(n,\ \sum_{j=0}^{n-i} 2^{(n-1)-j} a_{(n-1)-j})} = r^{(n,a)}$ in order. Namely, the route, through which the walker moves toward the desired units, corresponds to the binary digits of the address value in the same as the bucket-brigade process. For example, the walker with address $3(=011)$ on the bus $r^{(0,0)}$ moves first to the left bus $r^{(1,0)}$, then to the right $r^{(2,1)}$, and finally to $r^{(3,3)}$.

Second, the querying operation $\mathcal{Q}$ is defined as

$$\mathcal{Q} := \sum_{a=0}^{2^n-1} \left( |r^{(n,a)}\rangle\langle r^{(n,a)}|_{\mathrm{B}} \otimes \left[ \bigotimes_{q=0}^{m-1} (\mathcal{X}_{\mathrm{D}_q})^{x_q^{(a)}} \right] \right) \in \mathrm{End}(V_{\mathrm{B}} \otimes V_{\mathrm{D}}), \tag{3.2.11}$$

where $\mathcal{X}_{\mathrm{D}_i}$ is the Pauli-X acting on $q$th data qubit, i.e., $\mathcal{X}_{\mathrm{D}_i} := |0\rangle\langle 1|_{\mathrm{D}_i} + |1\rangle\langle 0|_{\mathrm{D}_i} \in \mathrm{End}(V_{\mathrm{D}_i})$. The summand represents the querying of a memory unit. Namely, the walker on the bus $r^{(n,a)}$ changes its data from $|0\rangle_{\mathrm{D}}$ to $|x^{(a)}\rangle_{\mathrm{D}}$ as

$$\begin{aligned}
&|r^{(n,a)}\rangle\langle r^{(n,a)}|_{\mathrm{B}} \otimes \left[\bigotimes_{q=0}^{m-1} \left(\mathcal{X}_{\mathrm{D}_q}\right)^{x_q^{(a)}}\right] : |r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}} \\
&\mapsto |r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}} \\
&\quad \Big(= |r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}\left[|x_{m-1}^{(a)}\rangle_{\mathrm{D}_{m-1}} \cdots |x_1^{(a)}\rangle_{\mathrm{D}_1}|x_0^{(a)}\rangle_{\mathrm{D}_0}\right] \\
&\quad\quad = |r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}\left[\left(\mathcal{X}_{D_{m-1}}\right)^{x_{m-1}^{(a)}}|0\rangle_{\mathrm{D}_{m-1}} \cdots \left(\mathcal{X}_{D_1}\right)^{x_1^{(a)}}|0\rangle_{\mathrm{D}_1}\left(\mathcal{X}_{D_0}\right)^{x_0^{(a)}}|0\rangle_{\mathrm{D}_0}\right]\Big)
\end{aligned} \tag{3.2.12}$$

After the query, the walker moves to the bus $r^{(0,0)}$ on the opposite binary tree, which is described by the conjugate transpose $\mathcal{R}^\dagger$ of $\mathcal{R}$ as shown in Eq. (3.2.2).

Whereas the bucket-brigade process inputs $n+m$ qubits sequentially, our process inputs them in parallel. The required steps are then only $O(n)$ in total, as shown in Eq. (3.2.3).

Additionally, no post-processing is required, such as dissolving entanglements between the registers and nodes of the binary tree in the bucket-brigade process. The nodes and walker (address and data registers) are never entangled because each node is equipped only with devices implementing the shift (3.1.2), coinflip (3.2.5), and controlled-coinflip operators (3.2.4) to pass the walker on the left or right bus appropriately. Here, each of these devices itself does not exhibit quantum superposition and entanglement.

In fact, the implementation of the controlled coinflip $\mathcal{X}_{\mathrm{A,C}}^{(l,w)}$ requires $O(\log(n+m))$ circuit depth, as described in the next chapter (see Eq. (4.2.7)). The total circuit depth for the physical implementation of this algorithm thus becomes $O(n\log(n+m))$.

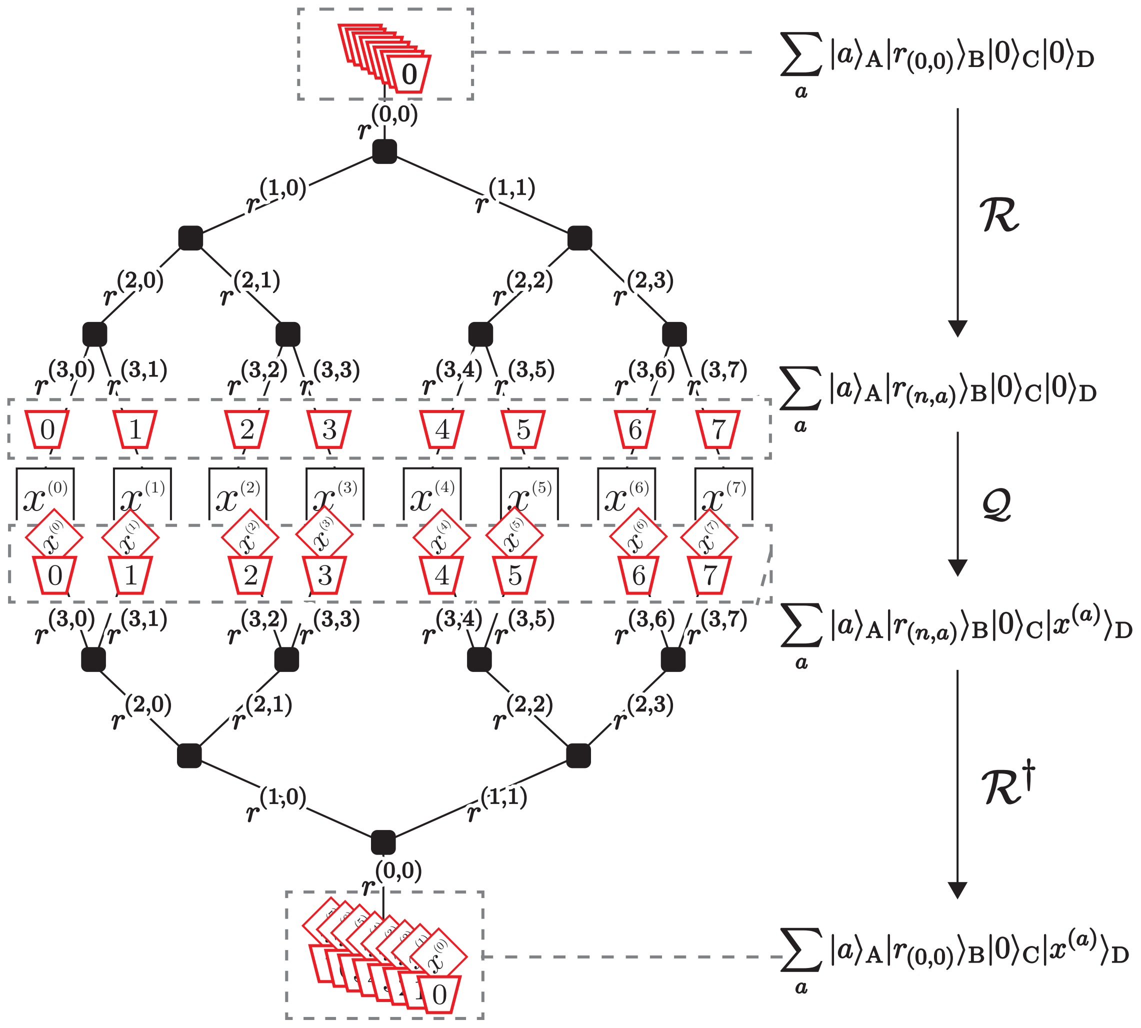


Figure 3.3: Conceptual image of our algorithm. The bucket with data represents the discrete-time quantum walker that serves as the address and data registers. First, the quantum walker accesses the memory units as spatial superposition by the routing operation $\mathcal{R}$ (3.2.3). Subsequently, the walker queries data from the units by the querying operation $\mathcal{Q}$ (3.2.11). Finally, we obtain a superposition of the data from the root bus $r^{(0,0)}$ by the output operation $\mathcal{R}^\dagger$.

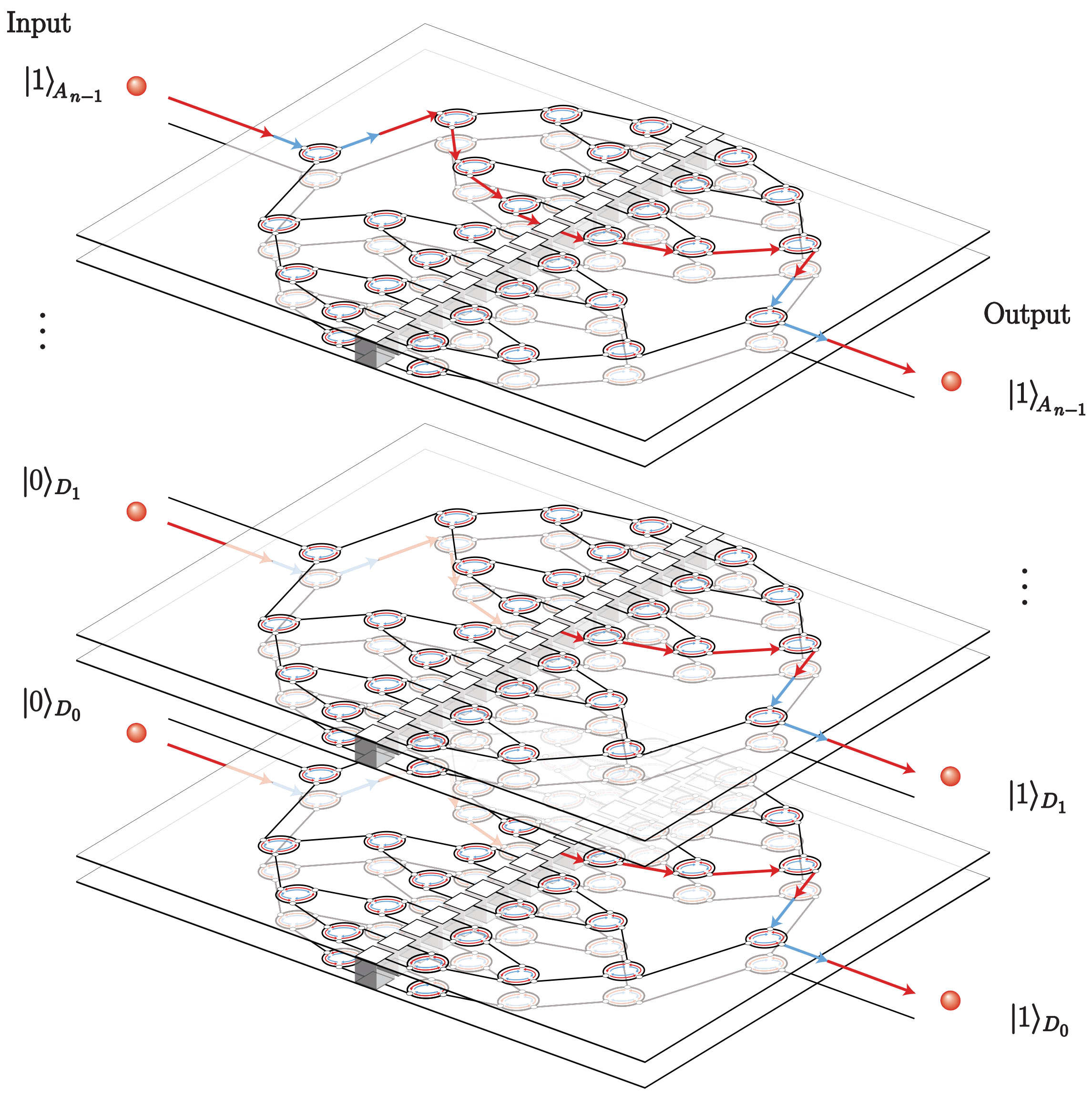


Figure 3.4: Overview of our qRAM. Because we construct each bus from $2(n+m)$ rails for the $n+m$ dual-rail encoded qubits, the architecture features stacked binary trees with a total of $2(n+m)$ trees. At each bifurcation point of the $2(n+m)$ rails, the dual-rail encoded qubits (Eq. (4.1.1) or Eq. (4.1.2)) undergo operations, the controlled coinflip $\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$ (Eq. (4.2.3) or Eq. (4.2.4)), shift operator $\mathcal{S}_{\mathrm{B,C}}^{(l-1,w)}$ (Eq. (4.3.2) or Eq. (4.3.3)), and coinflip $\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}$ (Eq. (4.4.1) or Eq. (4.4.2)).

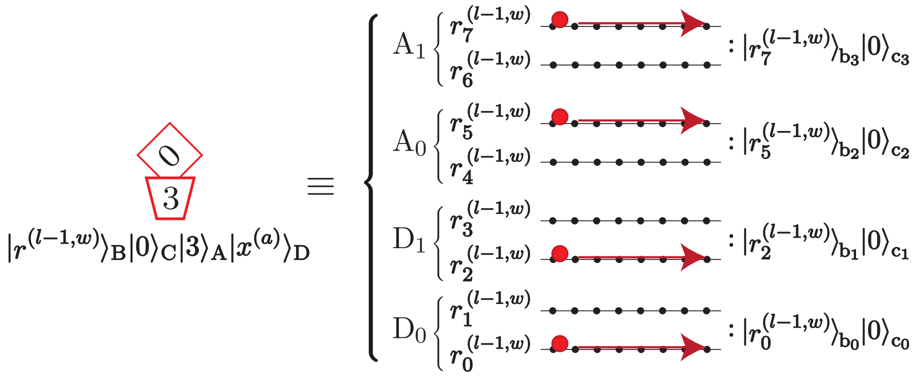
Chapter 4

# Connection between discrete-time and continuous-time QWs [1–3]

This chapter summarizes the key concepts for implementing our retrieval algorithm using only $n + m$ qubit resources, specifically using $n + m$ particles with two-level internal states. We will highlight that the implementation necessitates three gates: the roundabout, NOT, and controlled NOT gates. Note that the implementation utilizing the continuous-time quantum walk is detailed in the next chapter.

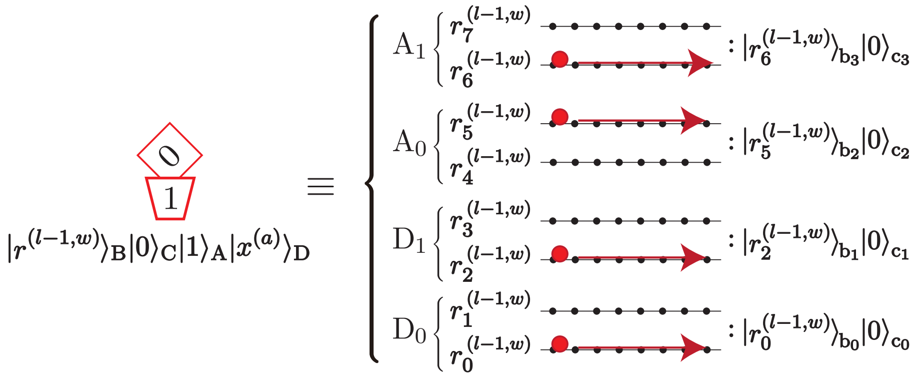
## 4.1 Implementation of the bucket with data (address and data registers)

To implement the set of address and data registers (bucket with data), we employ $n + m$ particles as dual-rail encoded qubits. Assume here that each particle has two-level internal states, which are represented by $|0\rangle_{c_q}$ and $|1\rangle_{c_q}$ for the $q$th of $n+m$ particles (let us represent these internal states as respectively red and blue because they correspond to the red and blue buckets with data). For example, we implement sets of the registers, whose states are $|0\rangle_{\mathrm{C}}|r^{(l-1,w)}\rangle_{\mathrm{B}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$ and $|0\rangle_{\mathrm{C}}|r^{(l-1,w)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$, using four particles:



0
3
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}}$
$\equiv$
$\mathrm{A}_1$
$r_7^{(l-1,w)}$
$r_6^{(l-1,w)}$
$: |r_7^{(l-1,w)}\rangle_{\mathrm{b}_3}|0\rangle_{\mathrm{c}_3}$
$\mathrm{A}_0$
$r_5^{(l-1,w)}$
$r_4^{(l-1,w)}$
$: |r_5^{(l-1,w)}\rangle_{\mathrm{b}_2}|0\rangle_{\mathrm{c}_2}$
$\mathrm{D}_1$
$r_3^{(l-1,w)}$
$r_2^{(l-1,w)}$
$: |r_2^{(l-1,w)}\rangle_{\mathrm{b}_1}|0\rangle_{\mathrm{c}_1}$
$\mathrm{D}_0$
$r_1^{(l-1,w)}$
$r_0^{(l-1,w)}$
$: |r_0^{(l-1,w)}\rangle_{\mathrm{b}_0}|0\rangle_{\mathrm{c}_0}$

(4.1.1)



0
1
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|1\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}}$
$\equiv$
$\mathrm{A}_1$
$r_7^{(l-1,w)}$
$r_6^{(l-1,w)}$
$: |r_6^{(l-1,w)}\rangle_{\mathrm{b}_3}|0\rangle_{\mathrm{c}_3}$
$\mathrm{A}_0$
$r_5^{(l-1,w)}$
$r_4^{(l-1,w)}$
$: |r_5^{(l-1,w)}\rangle_{\mathrm{b}_2}|0\rangle_{\mathrm{c}_2}$
$\mathrm{D}_1$
$r_3^{(l-1,w)}$
$r_2^{(l-1,w)}$
$: |r_2^{(l-1,w)}\rangle_{\mathrm{b}_1}|0\rangle_{\mathrm{c}_1}$
$\mathrm{D}_0$
$r_1^{(l-1,w)}$
$r_0^{(l-1,w)}$
$: |r_0^{(l-1,w)}\rangle_{\mathrm{b}_0}|0\rangle_{\mathrm{c}_0}$

(4.1.2)

where we represent the color $|0\rangle_{\mathrm{C}}$ by aligning all colors of the particles to red (namely, if the color of the bucket with data is blue, we align all colors of the particles to blue). As described in the introduction (see Eq. (1.1.1)), the dual-rail encoding represents the binary value 0 or 1 based on which of the two parallel rails a particle passes through. We then construct the bus $r^{(l-1,w)}$ using $2(n+m)$ rails labeled as $r_0^{(l-1,w)}, r_1^{(l-1,w)}, r_2^{(l-1,w)}, \dots, r_{2(n+m)-1}^{(l-1,w)}$ for the $n+m$ qubits. Here, a set of two rails, $r_{2q}^{(l-1,w)}$ and $r_{2q+1}^{(l-1,w)}$, corresponds to the $q$th data qubit for $0 \le q \le m-1$ or $(q-m)$th address qubit for $m \le q \le n+m-1$.

Explicitly, the dual-rail encoded state, which represents the bucket with data colored red or blue, is written as follows:

$$|r^{(l-1,w)}\rangle_{\mathrm{B}}|c\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|x\rangle_{\mathrm{D}} = |r^{(l-1,w)}\rangle_{\mathrm{B}}|c\rangle_{\mathrm{C}}\left(\bigotimes_{q=0}^{n-1}|a_q\rangle_{\mathrm{A}_q}\right)\left(\bigotimes_{q=0}^{m-1}|x_q\rangle_{\mathrm{D}_q}\right) \tag{4.1.3}$$

$$\equiv \left\{\bigotimes_{q=m}^{n+m-1}\left(\delta_{0,a_{q-m}}|r_{2q}^{(l-1,w)}\rangle_{\mathrm{b}_q} + \delta_{1,q-m}|r_{2q+1}^{(l-1,w)}\rangle_{\mathrm{b}_q}\right)|c\rangle_{\mathrm{c}_q}\right\}$$
$$\otimes\left\{\bigotimes_{q=0}^{m-1}\left(\delta_{0,x_q}|r_{2q}^{(l-1,w)}\rangle_{\mathrm{b}_q} + \delta_{1,x_q}|r_{2q+1}^{(l-1,w)}\rangle_{\mathrm{b}_q}\right)|c_q\rangle_{\mathrm{c}_q}\right\} \in \bigotimes_{q=0}^{n+m-1} V_{\mathrm{b}_q}\otimes V_{\mathrm{c}_q}, \tag{4.1.4}$$

where $c \in \{0,1\}$. Here, the *color space* $V_{\mathrm{c}_q}$ is the Hilbert space spanned by $\{|c\rangle_{\mathrm{c}_q}|c \in \{0,1\}\}$, which represents the internal state of the $q$th particle. Subsequently, the *bus space* $V_{\mathrm{b}_q}$ is the Hilbert space spanned by $\{|r_{2q}^{(l-1,w)}\rangle_{\mathrm{b}_q}, |r_{2q+1}^{(l-1,w)}\rangle_{\mathrm{b}_q}|0 \leq l \leq n-1, 0 \leq w \leq 2^l - 1\}$, representing rails on which the $q$th dual-rail encoded particle exists. Note that these particles are indistinguishable; we assign the label $q$ explicitly to the particle corresponding to the $q$th dual-rail encoded qubit, i.e., passing through the rail $r_{2q}^{(l-1,w)}$ or $r_{2q+1}^{(l-1,w)}$.

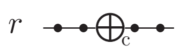
Because each bus consists of $2(n+m)$ rails, the architecture's structure features $2(n+m)$ binary trees stacked as shown in Fig. 3.4.

## 4.2 Implementation of the controlled coinflip $\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$

To implement the controlled coinflip $\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$ (3.2.4) we introduce the NOT gate and controlled NOT gates acting on the internal states (colors). Here, the NOT gate reverses the color of a single particle. The controlled NOT gate acts between two particles, e.g., particles $q$ and $q'$, so that the color of $q$th particle is reversed only if the color of $q'$th particle is $|1\rangle_{\mathrm{c}_{q'}}$.

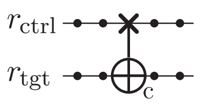
Let us denote the NOT gate installed in any rail $r$, and the controlled NOT gate in any two rails $r_{\mathrm{ctrl}}$ and $r_{\mathrm{tgt}}$ as

$\mathrm{NOT_c}(r) :=$



$$= |r\rangle\langle r|_{\mathrm{b}_q} \otimes \left(|1\rangle\langle 0|_{\mathrm{c}_q} + |0\rangle\langle 1|_{\mathrm{c}_q}\right), \tag{4.2.1}$$

$\mathrm{CNOT_{c,c'}}(r_{\mathrm{ctrl}}, r_{\mathrm{tgt}}) :=$

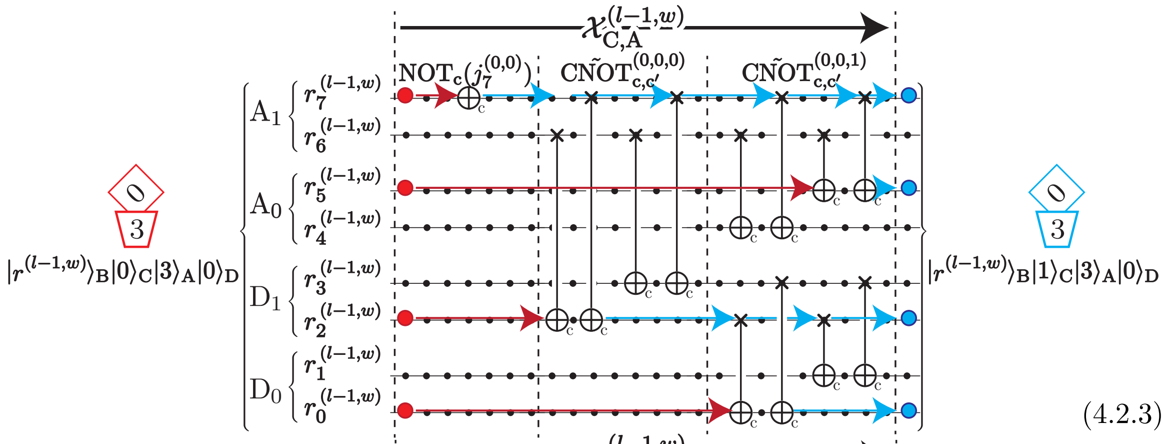


$$= |r_{\mathrm{ctrl}}\rangle\langle r_{\mathrm{ctrl}}|_{\mathrm{b}_{q'}} \otimes |1\rangle\langle 1|_{\mathrm{c}_{q'}} \otimes \mathrm{NOT_c}(r_{\mathrm{tgt}}), \tag{4.2.2}$$

where $\mathrm{NOT_c}(r) \in \mathrm{End}(V_{\mathrm{b}_q} \otimes V_{\mathrm{c}_q})$ and $\mathrm{CNOT_{c,c'}}(r_{\mathrm{ctrl}}, r_{\mathrm{tgt}}) \in \mathrm{End}(V_{\mathrm{b}_q} \otimes V_{\mathrm{c}_q} \otimes V_{\mathrm{b}_{q'}} \otimes V_{\mathrm{c}_{q'}})$. Actual implementations of the NOT and controlled NOT gates are detailed in Sec 5.2.

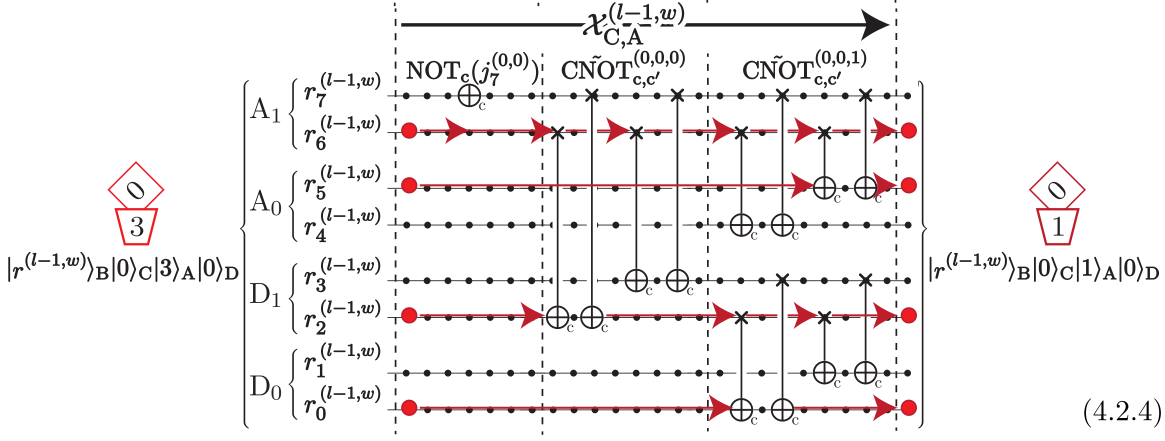
Recall that the controlled coinflip, represented by the operator $\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$, flips the coin (color) of the bucket with data only if its $[(n-1)-l]$th address qubit is $|1\rangle_{\mathrm{A}_{(n-1)-l}}$. For implementing this operator, we combine NOT and controlled NOT gates to construct a corresponding circuit. First, we use the NOT gate to reverse the particle's color passing through the rail $r_{2((n+m-1)-l)+1}^{(l-1,w)}$, which corresponds to the address qubit $|1\rangle_{\mathrm{A}_{(n-1)-l}}$. Subsequently, we use the controlled NOT gates to align the internal states of all particles. Namely, the colors of the remaining particles match the particle's color changed in the first step.

For example, a four-qubit circuit for the operator $\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$ changes (resp. does not change) the color of the four-qubit state (4.1.1) (resp. (4.1.2)) as follows:



$\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$
$\mathrm{NOT_c}(j_7^{(0,0)})$
$\tilde{\mathrm{CNOT}}_{\mathrm{c,c'}}^{(0,0,0)}$
$\tilde{\mathrm{CNOT}}_{\mathrm{c,c'}}^{(0,0,1)}$
$\mathrm{A}_1$
$\mathrm{A}_0$
$\mathrm{D}_1$
$\mathrm{D}_0$
$r_7^{(l-1,w)}$
$r_6^{(l-1,w)}$
$r_5^{(l-1,w)}$
$r_4^{(l-1,w)}$
$r_3^{(l-1,w)}$
$r_2^{(l-1,w)}$
$r_1^{(l-1,w)}$
$r_0^{(l-1,w)}$
0
3
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$
0
3
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$

(4.2.3)



$\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)}$
$\mathrm{NOT_c}(j_7^{(0,0)})$
$\tilde{\mathrm{CNOT}}_{\mathrm{c,c'}}^{(0,0,0)}$
$\tilde{\mathrm{CNOT}}_{\mathrm{c,c'}}^{(0,0,1)}$
$\mathrm{A}_1$
$\mathrm{A}_0$
$\mathrm{D}_1$
$\mathrm{D}_0$
$r_7^{(l-1,w)}$
$r_6^{(l-1,w)}$
$r_5^{(l-1,w)}$
$r_4^{(l-1,w)}$
$r_3^{(l-1,w)}$
$r_2^{(l-1,w)}$
$r_1^{(l-1,w)}$
$r_0^{(l-1,w)}$
0
3
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$
0
1
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|1\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$

(4.2.4)

Here, the operator $\tilde{\mathrm{CNOT}}_{\mathrm{c,c'}}^{(l-1,w,t)}$ is defined as

$$\tilde{\mathrm{CNOT}}_{\mathrm{c,c'}}^{(l-1,w,t)} := \bigotimes_{k=0}^{2^t-1} \prod_{s,s'\in\{0,1\}} \mathrm{CNOT}_{\mathrm{c,c'}}\left(r_{2q_{l-1,t,k}+s}^{(l-1,w)}, r_{2q_{l-1,t,k+1/2}+s'}^{(l-1,w)}\right) \tag{4.2.5}$$

$$\in \mathrm{End}\left(\bigotimes_{k=0}^{2^t-1}(V_{\mathrm{c}_{q_{l,t,k}}} \otimes V_{\mathrm{b}_{q_{l,t,k}}}) \otimes (V_{\mathrm{c}_{q_{l,t,k+1/2}}} \otimes V_{\mathrm{b}_{q_{l,t,k+1/2}}})\right). \tag{4.2.6}$$

Here, the two rails $r_{2q_{l-1,t,k}}^{(l-1,w)}$ and $r_{2q_{l-1,t,k}+1}^{(l-1,w)}$ correspond to the $q_{l-1,t,k}$th of $n+m$ qubits, where $q_{l-1,t,k} := ((n+m-l-1)-(N/2^t)k) \bmod (n+m)$.

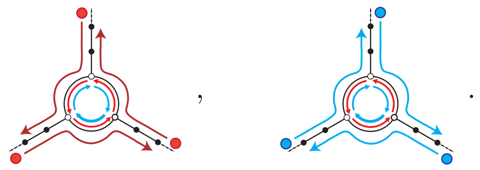
Explicitly, the circuit which implements the controlled coinflips is written as

$$\mathcal{X}_{\mathrm{C,A}}^{(l-1,w)} \equiv \left(\prod_{t=0}^{\log(n+m)-1} \tilde{\mathrm{CNOT}}_{\mathrm{c,c'}}^{(l-1,w,t)}\right) \mathrm{NOT_c}\left(r_{2q_{l-1,0,0}+1}^{(l-1,w)}\right) \in \mathrm{End}\left(\bigotimes_{q=0}^{n+m-1} V_{\mathrm{c}_q} \otimes V_{\mathrm{b}_q}\right). \tag{4.2.7}$$

The depth of the circuit implementing the controlled coinflip is then $O(\log(n+m))$.

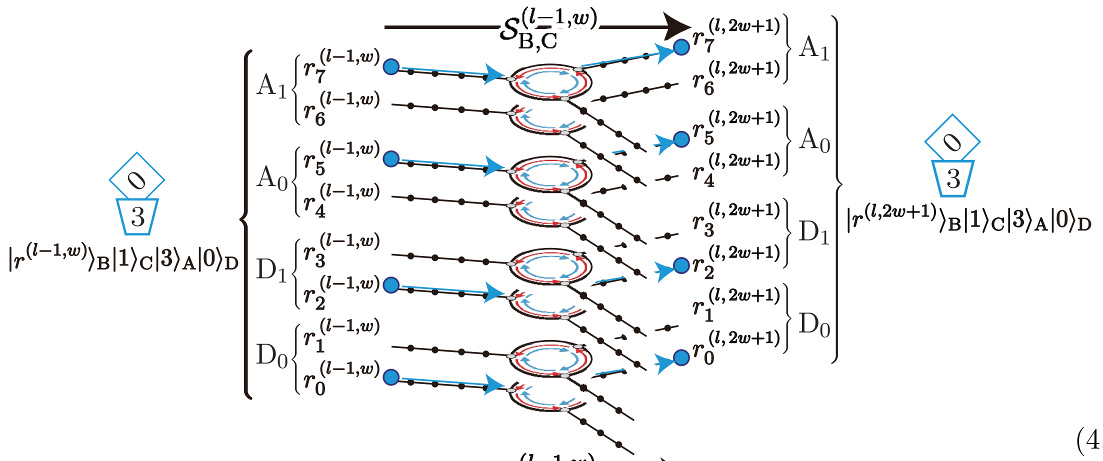
## 4.3 Implementation of the shift operator $S^{(l-1,w)}_{\mathrm{B,C}}$

To implement the shift coin operator $S^{(l-1,w)}_{\mathrm{B,C}}$ (3.1.2), we branch the $2(n+m)$ rails of the bus $r^{(l-1,w)}$ into left and right rails using $2(n+m)$ roundabout gates. Here, as mentioned in the introduction (see Eq. (1.3.4)), the roundabout gate scatters an incident particle to another rail either counter-clockwise or clockwise depending on its internal states:


(4.3.1)

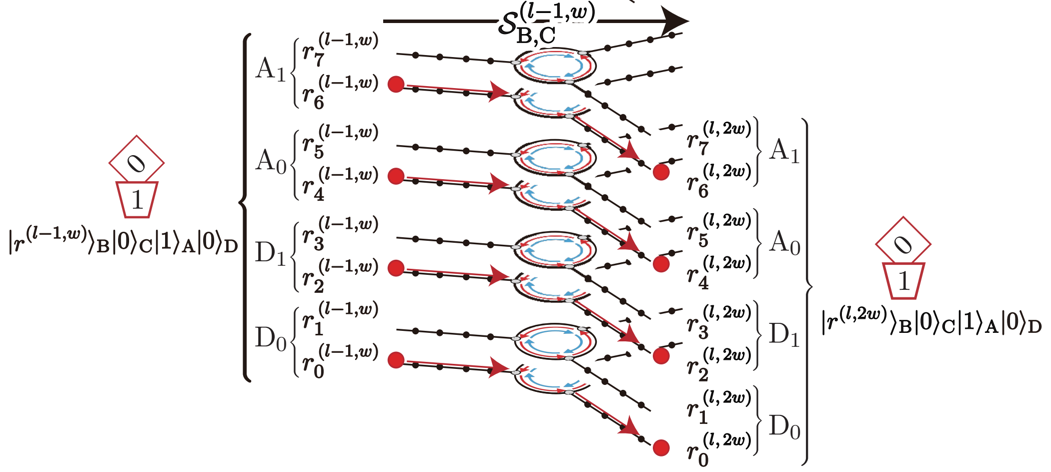
We will derive a subgraph demonstrating the functionality of the roundabout gate in Sec. 5.2.

As an example of the implementation of the shift operator, let us consider the four-qubit dual-rail encoded particles. These particles, whose colors are changed to red or blue by the controlled coinflip, scatter to respectively the left or right rails depending on these colors by the roundabout gates:



$\mathcal{S}^{(l-1,w)}_{\mathrm{B,C}}$
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$
0
3
$\mathrm{A}_1$ $r^{(l-1,w)}_7$ $r^{(l-1,w)}_6$
$\mathrm{A}_0$ $r^{(l-1,w)}_5$ $r^{(l-1,w)}_4$
$\mathrm{D}_1$ $r^{(l-1,w)}_3$ $r^{(l-1,w)}_2$
$\mathrm{D}_0$ $r^{(l-1,w)}_1$ $r^{(l-1,w)}_0$
$r^{(l,2w+1)}_7$ $r^{(l,2w+1)}_6$ $\mathrm{A}_1$
$r^{(l,2w+1)}_5$ $r^{(l,2w+1)}_4$ $\mathrm{A}_0$
$r^{(l,2w+1)}_3$ $r^{(l,2w+1)}_2$ $\mathrm{D}_1$
$r^{(l,2w+1)}_1$ $r^{(l,2w+1)}_0$ $\mathrm{D}_0$
$|r^{(l,2w+1)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$

(4.3.2)



$\mathcal{S}^{(l-1,w)}_{\mathrm{B,C}}$
$|r^{(l-1,w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|1\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$
0
1
$\mathrm{A}_1$ $r^{(l-1,w)}_7$ $r^{(l-1,w)}_6$
$\mathrm{A}_0$ $r^{(l-1,w)}_5$ $r^{(l-1,w)}_4$
$\mathrm{D}_1$ $r^{(l-1,w)}_3$ $r^{(l-1,w)}_2$
$\mathrm{D}_0$ $r^{(l-1,w)}_1$ $r^{(l-1,w)}_0$
$r^{(l,2w)}_7$ $r^{(l,2w)}_6$ $\mathrm{A}_1$
$r^{(l,2w)}_5$ $r^{(l,2w)}_4$ $\mathrm{A}_0$
$r^{(l,2w)}_3$ $r^{(l,2w)}_2$ $\mathrm{D}_1$
$r^{(l,2w)}_1$ $r^{(l,2w)}_0$ $\mathrm{D}_0$
$|r^{(l,2w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|1\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$

(4.3.3)

Here, each of the roundabout gates connects three rails: $r^{(l-1,w)}_i$, $r^{(l,2w)}_i$, and $r^{(l,2w+1)}_i$.

Explicitly, the implementation of the shift operator is written as

$$\mathcal{S}_{\mathrm{B,C}}^{(l-1,w)} \equiv \sum_{q=0}^{n+m-1} \sum_{s\in\{0,1\}} R\left(r_{2q+s}^{(l-1,w)}, r_{2q+s}^{(l,2w)}, r_{2q+s}^{(l,2w+1)}\right) \in \mathrm{End}\left(\bigotimes_{q=0}^{n+m-1} V_{\mathrm{b}_q} \otimes V_{\mathrm{c}_q}\right). \tag{4.3.4}$$

We denote here the functionality of the roundabout gate as

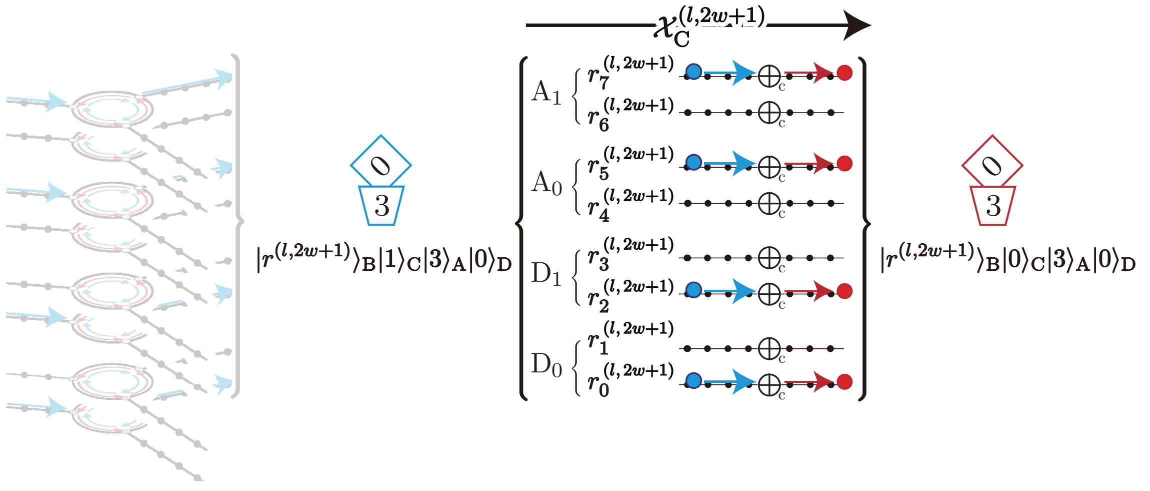
$$R\left(r, r', r''\right) := \tilde{R}\left(r, r', r''\right) \otimes |0\rangle\langle 0|_{\mathrm{c}_q} + \tilde{R}^{\top}\left(r, r', r''\right) \otimes |1\rangle\langle 1|_{\mathrm{c}_q} \in \mathrm{End}(V_{\mathrm{b}_q} \otimes V_{\mathrm{c}_q}), \tag{4.3.5}$$

$$\tilde{R}\left(r, r', r''\right) := |r'\rangle\langle r|_{\mathrm{b}_q} + |r''\rangle\langle r'|_{\mathrm{b}_q} + |r\rangle\langle r''|_{\mathrm{b}_q} \in \mathrm{End}(V_{\mathrm{b}_q}), \tag{4.3.6}$$

where the $r$, $r'$ and $r''$ are any rails connected to the roundabout gate. Recall that we assign the label $q$ to the particle corresponding to the $q$th dual-rail encoded qubit.

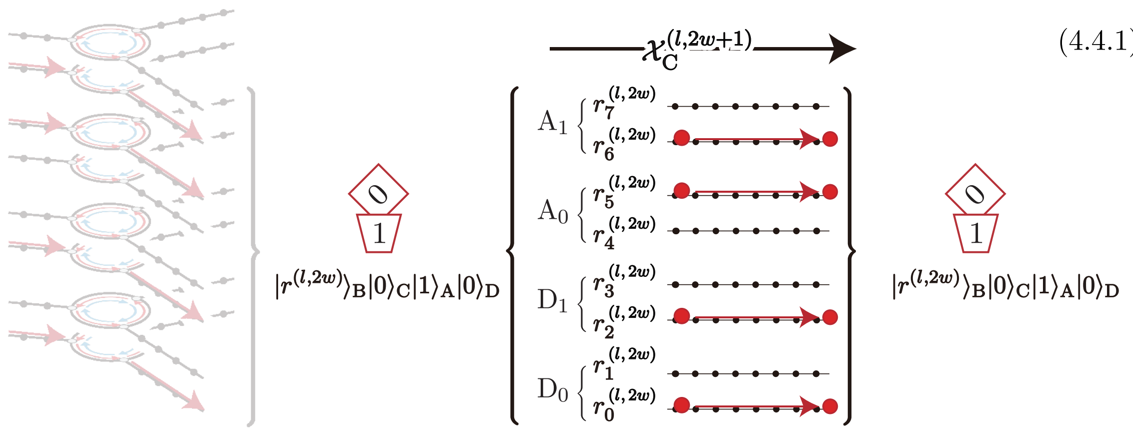
## 4.4 Implementation of the coinflip $\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}$

To implement the operator $\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}$ (3.2.5), we installed the NOT gates exclusively on the right-side rails $\{r_q^{(l,2w+1)}\}$, components of the bus $r^{(l,2w,+1)}$. Here, this operator resets the color of the bucket with data to red after the shift. Namely, as following examples, all colors of the four particles become red regardless of whether these particles scatter left or right:



$\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}$
0
3
$|r^{(l,2w+1)}\rangle_{\mathrm{B}}|1\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$
A$_1$
$r_7^{(l,2w+1)}$
$r_6^{(l,2w+1)}$
A$_0$
$r_5^{(l,2w+1)}$
$r_4^{(l,2w+1)}$
D$_1$
$r_3^{(l,2w+1)}$
$r_2^{(l,2w+1)}$
D$_0$
$r_1^{(l,2w+1)}$
$r_0^{(l,2w+1)}$
0
3
$|r^{(l,2w+1)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$

(4.4.1)



$\mathcal{X}_{\mathrm{C}}^{(l,2w+1)}$
0
1
$|r^{(l,2w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|1\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$
A$_1$
$r_7^{(l,2w)}$
$r_6^{(l,2w)}$
A$_0$
$r_5^{(l,2w)}$
$r_4^{(l,2w)}$
D$_1$
$r_3^{(l,2w)}$
$r_2^{(l,2w)}$
D$_0$
$r_1^{(l,2w)}$
$r_0^{(l,2w)}$
0
1
$|r^{(l,2w)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|1\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$

(4.4.2)

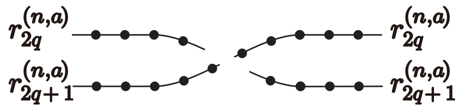
Explicitly, the implementation of this operator is written as

$$\mathcal{X}_{\mathrm{C}}^{(l,2w+1)} \equiv \sum_{q=0}^{n+m-1} \left(\mathrm{NOT}_{\mathrm{c}}\left(r_{2q}^{(l,2w+1)}\right) + \mathrm{NOT}_{\mathrm{c}}\left(r_{2q+1}^{(l,2w+1)}\right)\right) \in \mathrm{End}\left(\bigotimes_{q=0}^{n+m-1} V_{\mathrm{c}_q} \otimes V_{\mathrm{b}_q}\right). \tag{4.4.3}$$

## 4.5 Implementation of the query operator $\mathcal{X}_{\mathrm{D}_i}$

Finally, we implement the query operator $\mathcal{X}_{\mathrm{D}_i}$ in (3.2.11) by crossing the two rails if the corresponding digit of the data is 1 (See Fig. 4.1 (a) for example of $n = 2$ and $m = 2$):

$$\mathcal{X}_{\mathrm{D}_q} \equiv$$

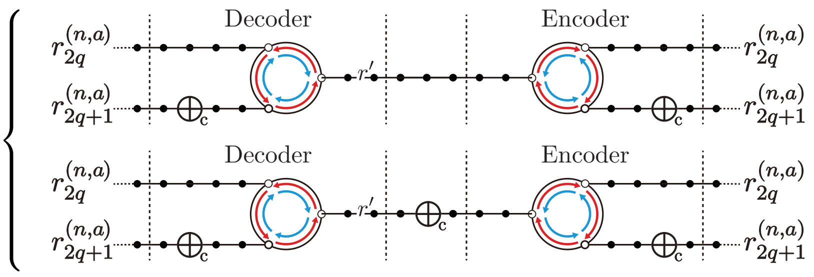


$$. \in \mathrm{End}(V_{\mathrm{b}_q}) \tag{4.5.1}$$

In another way, we can implement the query operator using an encoder and decoder. First, the decoder changes the particle's state from $|0\rangle_{\mathrm{c}_q}|r_{2q}^{(a,w)}\rangle_{\mathrm{b}_q}$ to $|0\rangle_{\mathrm{c}_q}|r'\rangle_{\mathrm{b}_q}$. Subsequently, we selectively write binary data 0 or 1 into the internal state by either applying or not applying the NOT gate on a rail $r'$. Finally, the binary data is dual-rail encoded by the encoder. The implementation is explicitly written as

$$(\mathcal{X}_{\mathrm{D}_i})^{x_q^{(a)}} \equiv \mathrm{Encoder} \cdot \left(\mathrm{NOT}_{\mathrm{c}}(r')\right)^{x_q^{(a)}} \cdot \mathrm{Decoder} \tag{4.5.2}$$

$=$



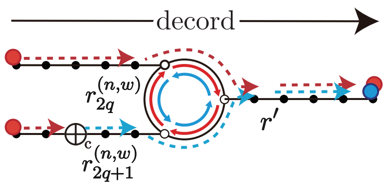
if $x_q^{(a)} = 0$

if $x_q^{(a)} = 1$

(4.5.3)

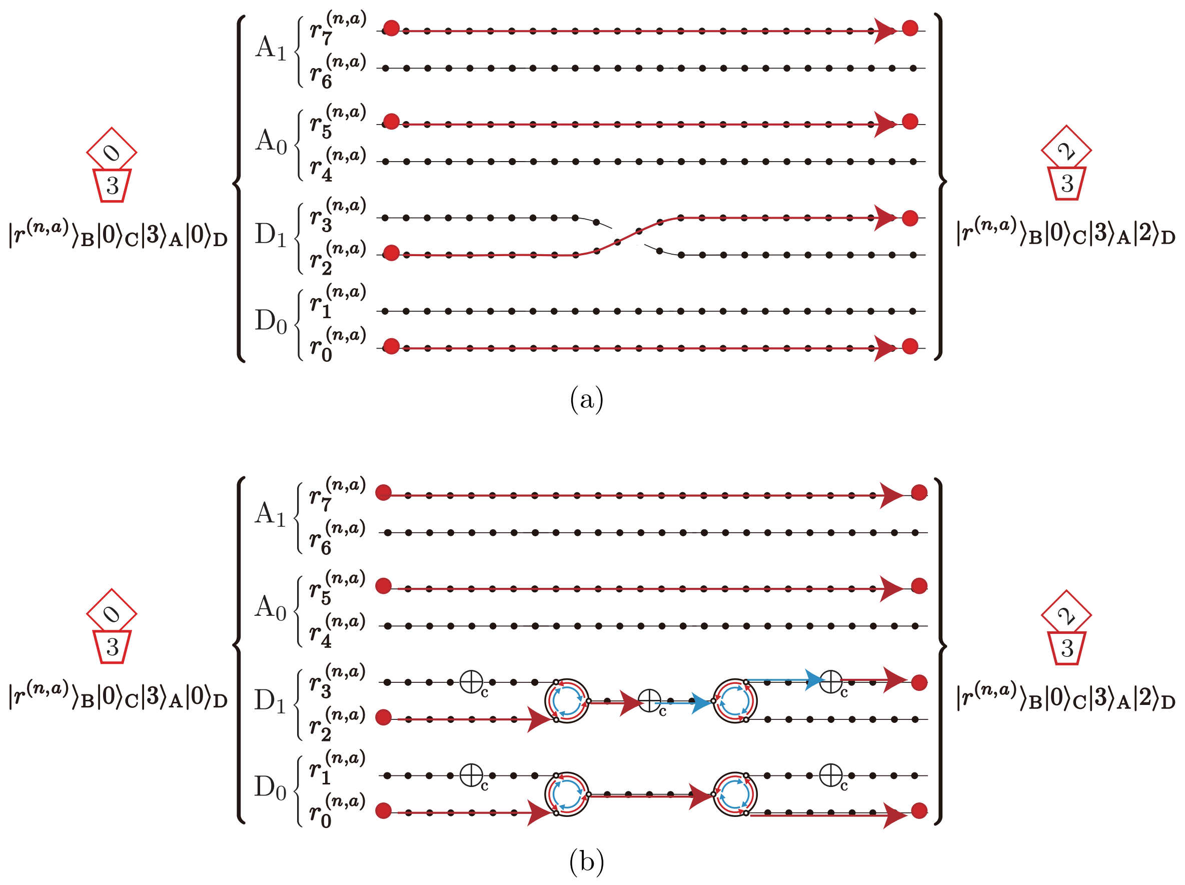
Here, the decoder, which is a combination of the roundabout and NOT gates, converts the dual-rail encoded qubit value into the internal states of the particle as

$$\left(\alpha|r_{2q}^{(a,w)}\rangle_{\mathrm{b}_q} + \beta|r_{2q+1}^{(a,w)}\rangle_{\mathrm{b}_q}\right)|0\rangle_{\mathrm{c}_q} :$$



$$: |r'\rangle_{\mathrm{b}_q}\left(\alpha|0\rangle_{\mathrm{c}_q} + \beta|1\rangle_{\mathrm{c}_q}\right), \tag{4.5.4}$$

where Encode $:= \mathrm{NOT}_{\mathrm{c}}(r_{2q+1}^{(a,w)})R(r', r_{2q}^{(a,w)}, r_{2q+1}^{(a,w)}) \in \mathrm{End}(V_{\mathrm{c}_q} \otimes V_{\mathrm{b}_q})$ (in the implementation of the qRAM, the decoder is used for only converting the qubit value 0 into the internal

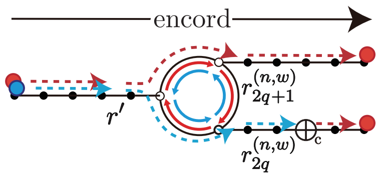


0
3
$|r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|0\rangle_{\mathrm{D}}$
A$_1$
A$_0$
D$_1$
D$_0$
$r_7^{(n,a)}$
$r_6^{(n,a)}$
$r_5^{(n,a)}$
$r_4^{(n,a)}$
$r_3^{(n,a)}$
$r_2^{(n,a)}$
$r_1^{(n,a)}$
$r_0^{(n,a)}$
2
3
$|r^{(n,a)}\rangle_{\mathrm{B}}|0\rangle_{\mathrm{C}}|3\rangle_{\mathrm{A}}|2\rangle_{\mathrm{D}}$
(a)
c
(b)

Figure 4.1: Circuits from querying from the 3rd memory unit, in which the data $x^{(n)}$ $(=2)$ is recorded. Both of these circuits have equivalent functionality. (a) The query operator $\mathcal{X}_{\mathrm{D}_q}$ is implemented by crossing the two parallel rails if the $q$th binary digit is 1. (b) The query operator can also be implemented by combining the decoder, encoder, and NOT gates. Depending on whether the binary digit 0 or 1, a NOT gate is either placed or not on a rail between the decoder and encoder.

states referred to as red). Additionally, the encoder is defined as the inverse operation of the decoder as

$$|r'\rangle_{\mathrm{b}_q}\left(\alpha|0\rangle_{\mathrm{c}_q}+\beta|1\rangle_{\mathrm{c}_q}\right):$$



$$:\left(\alpha|r_{2q}^{(a,w)}\rangle_{\mathrm{b}_q}+\beta|r_{2q+1}^{(a,w)}\rangle_{\mathrm{b}_q}\right)|0\rangle_{\mathrm{c}_q}$$

(4.5.5)

(See Fig. 4.1 (b) for example of $n=2$ and $m=2$).

Chapter 5

# Physical implementation by continuous-time QW [2, 3]

In this chapter, we discuss a physical implementation of qRAM that adopts our retrieval algorithm, which solidifies the algorithm's two advantages; the implementation requires neither ancillary qubit resources nor any time-dependent control. First, we formulate a two-level multi-particle continuous-time quantum walk applicable to either bosonic or fermionic particles with two-level internal states. Second, we propose implementations of the three key gates introduced in Chap. 4, i.e., roundabout, NOT, and CNOT gates. Third, we discuss the circuit construction and estimate an error caused by a finite-size effect. Finally, we propose a universal quantum computer compatible with our qRAM.

## 5.1 Continuous-time quantum walks with two-level internal states

To detail the implementation in the next chapter, we formulate either a bosonic or fermionic multi-particle continuous-time quantum walk with two-level internal states. In this model, the particles evolve on a graph $G$ in continuous time according to the following Hamiltonian:

$$H_G = K_G + U_G, \tag{5.1.1}$$

where $K_G$ is a kinetic term and $U_G$ is an interaction term defined in the following. Note that, in the next chapter, we construct an architecture adopting our retrieval algorithm as the graph $G$, where the particles traverse as the dual-rail encoded qubits.

Notably, our formulation, the kinetic term more specifically, exhibits internal-state dependence of the single-particle scattering on the subgraph, which leads to the roundabout gate in the following chapter. Here, consider that this subgraph, denoted as $\hat{G}$, connects $m$ semi-infinite rails $\{r_j | 0 \leq j \leq n-1\}$ as shown in Fig. 5.1. As described below, the incoming particle into the subgraph through a rail scatters according to a scattering matrix (S-matrix) or its transpose, depending on its internal states. Note that, in the next chapter, we derive a subgraph connecting three rails, which demonstrates the internal-state dependence of single-particle scattering, analogous to the roundabout gate (4.3.5).

To detail the single-particle scattering on the subgraph, let us consider the following scattering plane wave of the incoming particle through a semi-infinite rail $r_{j_{\text{in}}}$ $(0 \leq j_{\text{in}} \leq m-1)$ with momentum $k \in \mathbb{R}$ as

$$\begin{aligned} |\text{sc}_{r_{j_{\text{in}}}}(k)\rangle := \sum_x \Bigg[ &\sum_{j=0}^{m-1} \left( \delta_{r_j, r_{j_{\text{in}}}} e^{-ikx} + S_{0,r_j,r_{j_{\text{in}}}}(k) e^{ikx} \right) |v_x^{r_j}\rangle_{\text{v}_q} |0\rangle_{\text{c}_q} \\ &+ \sum_{j=0}^{m-1} \left( \delta_{r_j, r_{j_{\text{in}}}} e^{-ikx} + S_{1,r_j,r_{j_{\text{in}}}}(k) e^{ikx} \right) |v_x^{r_j}\rangle_{\text{v}_q} |1\rangle_{\text{c}_q} \Bigg] \in \text{End}\left( V_{\text{v}_q} \otimes V_{\text{c}_q} \right), \end{aligned} \tag{5.1.2}$$

$$\text{where } S_{1,r_j,r_{j_{\text{in}}}}(k) = S_{0,r_{j_{\text{in}}},r_j}(k) \text{ for } 0 \leq j, j_{\text{in}} \leq m-1. \tag{5.1.3}$$

Here, $|v_x^{r_j}\rangle_{\text{v}_q}$ denotes the particle being at the $x$th vertex along the rail $r_j$ counted from the subgraph, i.e., $v_{x=0}^{r_j}$ is the leading vertex attached to the subgraph. The space $V_{\text{v}_q}$ is spanned by $\{|v\rangle_{\text{v}_q} | v \in \mathbb{V}(G)\}$ (as described in Eq. (5.2.1), this space includes the bus space $V_{\text{b}_i}$). We denote the scattering matrix for the particle with $|c\rangle_{\text{c}_i}$ for $c \in \{0,1\}$ as $S_c(k) \in \mathbb{C}^{m \times m}$, where $S_{c,r_j,r_{j_{\text{in}}}}(k) \in \mathbb{C}$ is the element at the $j$th row and $j_{\text{in}}$th column of this matrix. We prove the transposition (5.1.3) in the paragraph whose conclusion is (5.1.16).

Note that in the next chapter, we will construct the architecture of the qRAM as the graph $G$, a combination of subgraphs and rails. Additionally, we will introduce a wave packet as the dual-rail encoded qubit, which is constructed by a superposition of the plane waves and whose scatterings are dominated by a plane wave with specific momentum.

Now, let us explicitly define the kinetic and interaction terms in order. Specifically, we will first discuss the single-particle scattering on the subgraph using the kinetic, and then address a two-particle scattering using the interaction term (the implementation is achieved by combining the single- and two-particle scatterings). Note that we assume all rails, which are components of the entire graph $G$, have semi-infinite length to discuss each scattering independently.

**Kinetic term** The kinetic term of the $n$ particles on a graph $G$, which demonstrates the internal-state dependence, is defined as follows (note that focusing solely on the single particle, we can alter the kinetic term to (5.1.7)):

$$K_G := \sum_{c=0,1} \sum_{(v,v')\in\mathbb{E}(G)} \left( e^{i(-1)^c \theta_{v,v'}} a^\dagger_{v,c} a_{v',c} + e^{-i(-1)^c \theta^*_{v,v'}} a^\dagger_{v',c} a_{v,c} \right), \tag{5.1.4}$$

where $K_G \in \mathrm{End}(\bigotimes_q V_{\mathrm{v}_q} \otimes V_{\mathrm{c}_q})$. The graph $G$ consists of a set of edges $\mathbb{E}(G)$ and vertices $\mathbb{V}(G)$. Thus, $(v, v')$ represents an edge between two vertices $v$ and $v'$. Note that we do not distinguish $(v, v')$ and $(v', v)$ so that the sum is taken over either $(v, v')$ or $(v', v)$. The two terms inside the summation of $K_G$ represent the walks, i.e., switching the positions of a particle, between two vertices $v$ and $v'$ connected by the edge $(v, v')$. Namely, $a^\dagger_{v,c}$ (resp. $a_{v,c}$) is the creation (resp. annihilation) operator of a particle on the vertices $v$ with the internal state $c \in \{0, 1\}$, and $\theta_{v,v'} \in \mathbb{R}$ is a factor of a weight (coefficient) associated with the walk from $v$ to $v'$. Here, the creation operators $\{a^\dagger_{v,c} | v \in \mathbb{V}(G), c \in \{0, 1\}\}$ are generators of the Hilbert space $\bigotimes_q (V_{\mathrm{v}_q} \otimes V_{\mathrm{c}_q})$ so that these operators correspond to the bases spanning this space as

$$\left\{ \bigotimes_{q=0}^{n-1} \left( |v_q\rangle_{\mathrm{v}_q} |c_q\rangle_{\mathrm{c}_q} \right) = \prod_{q=0}^{n-1} a^\dagger_{v_q,c_q} |0\rangle \mid v_q \in \mathbb{V}(G), c_q \in \{0, 1\} \right\} \tag{5.1.5}$$

where $|0\rangle$ is the no-particle (vacuum) state defined by $a_{v_q,c_q}|0\rangle = 0$. Finally, to ensure the symmetry of bosonic particles (resp. the anti-symmetry and the Pauli exclusion principle for fermionic particles), we require the following commutation relations (resp. anti-commutation relations):

$$[a_{v_q,c_q}, a^\dagger_{v_{q'},c_{q'}}]_\pm = \delta_{v_q,v_{q'}} \delta_{c_q,c_{q'}}, \; [a_{v_q,c_q}, a_{v_{q'},c_{q'}}]_\pm = [a^\dagger_{v_q,c_q}, a^\dagger_{v_{q'},c_{q'}}]_\pm = 0, \tag{5.1.6}$$

where $[\cdot,\cdot]_-$ denotes the commutator ($[\cdot,\cdot]_+$ denotes the anti-commutator).

To show the transposition between the S-matrices of the red and blue particles (5.1.3), we introduce the kinetic term for the single-particle incident into the subgraph $\hat{G}$ as $K^{(1)}_G$. Here, as mentioned above, we assume each rail attached to the subgraph has semi-infinite length allowing us to treat the single-particle scattering on the subgraph independently. Focusing only on a single particle on the subgraph $\hat{G}$ attached $m$ semi-infinite rails, we obtain the following kinetic term for the single-particle from (5.1.4) and (5.1.5):

$$K^{(1)}_G := A_G \otimes |0\rangle\langle 0|_{\mathrm{c}_q} + A^*_G \otimes |1\rangle\langle 1|_{\mathrm{c}_q} \in \mathrm{End}(V_{\mathrm{v}_q} \otimes V_{\mathrm{c}_q}), \tag{5.1.7}$$

$$A(G) := A(\hat{G}) + \sum_{j=0}^{m-1} A(r_j) \in \mathrm{End}(V_{\mathrm{v}_q}). \tag{5.1.8}$$

Here, $A(\hat{G})$ and $A(r_j)$ are adjacency matrices of the subgraph $\hat{G}$ and the semi-infinite rail $r_j$, which contain walks the particle can take, e.g., $|v_x\rangle\langle v_{x'}|$ for $v_x, v_{x'} \in \mathbb{V}(\hat{G})$, and weights (coefficients) each associated with the walk, e.g., $e^{i\theta_{v_x,v_{x'}}}$ and $e^{-i\theta_{v_x,v_{x'}}}$, as

$$A(r_j) := \sum_x \left(|v_x^{r_j}\rangle\langle v_{x+1}^{r_j}|_{\mathrm{v}_q} + |v_{x+1}^{r_j}\rangle\langle v_x^{r_j}|_{\mathrm{v}_q}\right) \text{ for } 0 \leq j \leq m-1,\ x \geq 0. \tag{5.1.9}$$

$$A(\hat{G}) := \sum_{(v_x,v_{x'})\in\mathbb{E}(\hat{G})} \left(e^{i\theta_{v_x,v_{x'}}}|v_x\rangle\langle v_{x'}|_{\mathrm{v}_q} + e^{-i\theta_{v_x,v_{x'}}}|v_{x'}\rangle\langle v_x|_{\mathrm{v}_q}\right) \text{ for } 0 \leq x, x' \leq M-1 \tag{5.1.10}$$

where $\mathbb{E}(\hat{G})$ and $\mathbb{V}(\hat{G})$ represent sets of edges and vertices of the subgraph. Here, we assume that the subgraph $\hat{G}$ has $M$ vertices, and the first $m$ vertices are identical to the leading vertices of the $m$ rails, respectively:

$$v_0 = v_0^{r_0},\ v_1 = v_0^{r_1},\ v_2 = v_0^{r_2},\ \cdots,\ v_{m-1} = v_{x=0}^{r_{m-1}}. \tag{5.1.11}$$

Note that, as shown in Eq. (5.1.9), the rails consist of undirected edges whose weights are 1. However, as shown in Eq. (5.1.10), the subgraph can consist of directed edges whose weights depend on the direction of the walk (the experimental realization of the directed edge is discussed in Chap. 6).

Now let us show the transposition between the S-matrices of the red and blue particles (5.1.3). Here, the S-matrices are determined by the following Schrödinger equation:

$$K_G^{(1)}\left(|\mathrm{sc}_{j_{\mathrm{in}}}(k)\rangle + |\mathrm{bd}_{j_{\mathrm{in}}}(k)\rangle\right) = E_k\left(|\mathrm{sc}_{j_{\mathrm{in}}}(k)\rangle + |\mathrm{bd}_{j_{\mathrm{in}}}(k)\rangle\right) \tag{5.1.12}$$

$$\text{where } E_k = 2\cos k \ \because \langle v_{x\geq 1}^{j_r}|A_{j_r}|\mathrm{sc}_{j_{\mathrm{in}}}(k)\rangle = 2\cos k\langle v_{x\geq 1}^{j_r}|\mathrm{sc}_{j_{\mathrm{in}}}(k)\rangle. \tag{5.1.13}$$

where $|\mathrm{bd}_{j_{\mathrm{in}}}(k)\rangle \in \mathbb{C}^{M-m}$ is a bound state localized in the subgraph (see Eq. (A.1.7)). As described in Appendix A.1, using the Schrödinger equation and the conservation laws of momentum $k$ and energy $E_k$, we derive the following equation to determine the S-matrix from the adjacency matrix of the subgraph:

$$S_0(k) = -e^{2ik}Q^{-1}(k)Q(-k),\ S_1(k) = -e^{2ik}(Q^*(-k))^{-1}Q^*(k) \tag{5.1.14}$$

$$Q(k) := \left\{1 - e^{ik}\left(B_{\hat{G}} + C_{\hat{G}}^{\dagger}\frac{1}{2\cos k - D_{\hat{G}}}C_{\hat{G}}\right)\right\}. \tag{5.1.15}$$

Here, $B_{\hat{G}} \in \mathbb{C}^{m\times m}$ and $D_{\hat{G}} \in \mathbb{C}^{(M-m)\times(M-m)}$ are adjacency matrices, submatrices of $A_{\hat{G}}$ more specifically, consisting of vertices $\{v_x|0 \leq x \leq m-1\}$ and $\{v_x|m \leq x \leq M-1\}$, respectively (see Fig. 5.2 for concrete image). Additionally, $C_{\hat{G}} \in \mathbb{C}^{m\times(M-m)}$ is also a submatrix of $A_{\hat{G}}$, which represents the connections between the $m$ and $M-m$ vertices. In conclusion, we obtain the transposition (5.1.3) as

$$S_1(k) = (S_0(-k))^* = \left(S_0^{\dagger}(k)\right)^* = S_0(k)^T. \tag{5.1.16}$$

Here, we use $S_c(k) = S_c(-k)^{\dagger}$ obtained from (5.1.14) for $c \in \{0,1\}$.

In the implementation, we treat a wave packet as the particle's state, which serves as the dual-rail encoded qubit. The single particle scattering is characterized by the S-matrix $S_c(k = -\pi/2)$ for $c \in \{0,1\}$; we construct the wave packet by a superposition of plane waves, whose Fourier coefficient has a sharp peak at $k = -\pi/2$ as described in (5.2.2) and (5.2.3).

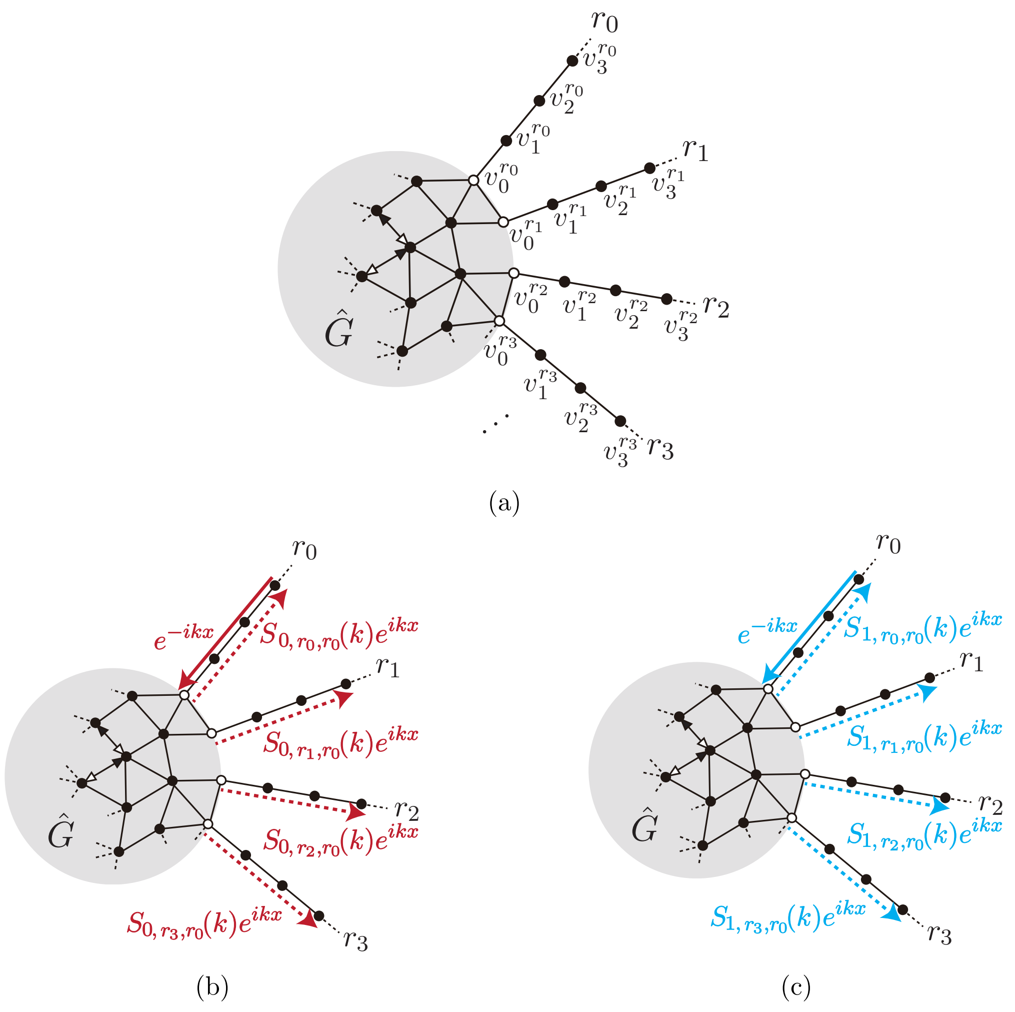


Figure 5.1: (a) A subgraph connecting rails. (b) Conceptual images of the scattering plane waves. The scattering coefficients for the red and blue particles are described by two S-matrices, which are transposed of each other, as shown in Fig. (5.1.3).

**Interaction term** We introduce the interaction term of the particles to consider the two-particle scattering, which is utilized for the implementation. Concretely, we consider the on-site (resp. nearest-neighbor) interactions for bosonic (resp. fermionic) particles:

$$U_G = \begin{cases} \frac{u}{2}\sum_{v\in\mathbb{V}(G)} n_v(n_v-1) \text{ for the bosonic particles,} \\ u\sum_{(v,v')\in\mathbb{E}(G)} n_v n_{v'} \text{ for the fermionic particles,} \end{cases} \tag{5.1.17}$$

where $n_v := \sum_{c=0}^{1} a_{v,c}^\dagger a_{v,c}$ is the number operator, and $u \in \mathbb{R}$ is the interaction strength.

This interaction term describes that two particles, whose states are written as the wave packets characterized by the plane waves with $k_0 \in (-\pi, 0)$ and $k_1 \in (0, \pi)$, acquire the following phase factor after the head-on scattering:

$$P_+(k_q, k_{q'}) := \frac{2(\sin k_q - \sin k_{q'}) + iu}{2(\sin k_q - \sin k_{q'}) - iu} \text{ for the bosonic,} \tag{5.1.18}$$

$$P_-(k_q, k_{q'}) := \frac{1 + e^{i(k_q+k_{q'})} - e^{ik_{q'}}u}{1 + e^{i(k_q+k_{q'})} - e^{ik_q}u} \text{ for the fermionic.} \tag{5.1.19}$$

See Appendix 2 for the details. The derivation of the phase shift is not pivotal to the discourse in the following chapter, unlike the discussion on the kinetic term used for the roundabout gate.

We assume that all rails have a semi-infinite length to discuss each scattering independently of other scatterings. However, in Sec. 5.3, we discuss the architecture construction using finite rails with an estimation of the error caused by finite-size effects. Note that this error estimation includes the position deviation of the wave packets due to single- and two-particle scatterings.

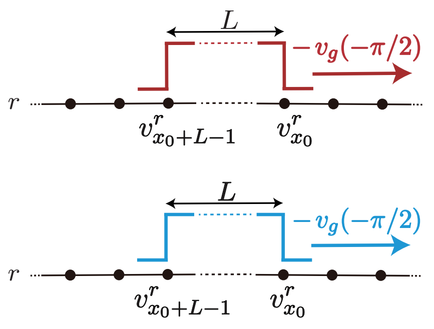
## 5.2 Physical implementation of circuit components

Now, we illustrate how continuous-time quantum walk with two-level internal states implements our data retrieval algorithm. Specifically, under this model, we detail the four key concepts introduced in Sec. 5.2 for the implementation: the dual-rail encoded qubit, NOT gate, controlled NOT gate, and roundabout gate. The above three gates require no ancillary qubit resource and time-dependent control, which indicates that the dual-rail encoded qubits (particles) simply passing the gates are sufficient to automatically complete the retrieval algorithm.

**Dual-rail encoded qubit** Because the architecture is based on a graph, specifically a combination of subgraphs and rails, the particles that serve as the dual-rail encoded qubits (4.1.4) must evolve over continuous time $t$ according to the Hamiltonian $H_{\mathrm{G}}$(5.1.1), or more specifically $e^{-iH_{\mathrm{G}}t}$. As described in the following, we then represent the particles' states using the wave packets each constructed by a superposition of the plane waves, the eigenstate of this Hamiltonian.

For convenience, we consider the following wave packet (sinc pulse) as the particle's state, which will serve as the dual-rail encoded qubit in Eq. (4.1.4), traveling along any semi-infinite rail $r$:

$$|r\rangle_{\mathrm{b}_q}|c\rangle_{\mathrm{c}_q} \equiv \frac{1}{\sqrt{L}}\sum_{x=x_0}^{x_0+L-1} e^{-i\frac{\pi}{2}x}|v_x^r\rangle_{\mathrm{v}_q}|c\rangle_{\mathrm{c}_q} = \left\{\vphantom{\sum_{x}^{x}}\right.$$



$(c = 0)$

$(c = 1)$

(5.2.1)

where $x_0, L \in \mathbb{Z}$ and $c \in \{0,1\}$ ($L$ denotes the wavelength). Here, this wave packet can be decomposed into the following superposition of the plane waves:

$$\frac{1}{\sqrt{L}} \sum_{x=x_0}^{x_0+L-1} e^{-i\frac{\pi}{2}x} |v_x^r\rangle_{\mathrm{v}_q} |c\rangle_{\mathrm{c}_i} = \sum_x \frac{1}{\sqrt{2\pi}} \left( \int dk f(k) e^{-ik(x-x_0)} \right) |v_x^r\rangle_{\mathrm{v}_q} |c\rangle_{\mathrm{c}_q} \tag{5.2.2}$$

where Fourier coefficient is defined as

$$f(k) = \sqrt{\frac{2}{L\pi}} \frac{\sin\left[\frac{L}{2}\left(k + \frac{\pi}{2}\right)\right]}{k + \frac{\pi}{2}}. \tag{5.2.3}$$

Let us show that this wave packet travels the rail with a group velocity defined as

$$v_g\left(-\frac{\pi}{2}\right) := E'\left(-\frac{\pi}{2}\right) = 2. \tag{5.2.4}$$

where $E(k) = 2\cos k$ from Eq. (5.1.13). The reason for this group velocity is that the wave packet (5.2.2) evolves over a time interval $T = O(L)$ as

$$e^{-iA(r)T} \left( \frac{1}{\sqrt{L}} \sum_{x=x_0}^{x_0+L-1} e^{-i\frac{\pi}{2}x} |v_x^r\rangle_{\mathrm{v}_q} |c\rangle_{\mathrm{c}_q} \right) \tag{5.2.5}$$

$$= \sum_x \left( \frac{1}{\sqrt{2\pi}} \int dk f(k) e^{-ikx - iE(k)T} \right) |v_x^r\rangle_{\mathrm{v}_q} |c\rangle_{\mathrm{c}_q} \tag{5.2.6}$$

$$= \sum_x \left( \frac{1}{\sqrt{2\pi}} \int dk f(k) e^{-ikx - iv_g(-\frac{\pi}{2})(k+\frac{\pi}{2})T + O(L^{-\frac{1}{2}})} \right) |v_x^r\rangle_{\mathrm{v}_q} |c\rangle_{\mathrm{c}_q} \tag{5.2.7}$$

$$= e^{-iv_g(-\frac{\pi}{2})\frac{\pi}{2}T} \left( \frac{1}{\sqrt{L}} \sum_{x=x_0 - v_g(-\frac{\pi}{2})T}^{x_0+L-1-v_g(-\frac{\pi}{2})T} e^{-i\frac{\pi}{2}x} |v_x^r\rangle_{\mathrm{v}_q} |c\rangle_{\mathrm{c}_q} \right) + O(L^{-\frac{1}{2}}) \tag{5.2.8}$$

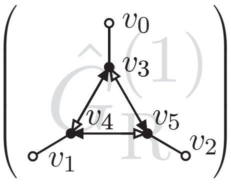
Here, we apply Taylor expansion to the second line as

$$E(k) = E\left(-\frac{\pi}{2}\right) + E'\left(-\frac{\pi}{2}\right)\left(k + \frac{\pi}{2}\right) + \frac{E''\left(-\frac{\pi}{2}\right)}{2!}\left(k + \frac{\pi}{2}\right)^2 + \frac{E'''\left(-\frac{\pi}{2}\right)}{3!}\left(k + \frac{\pi}{2}\right)^3 + \cdots \tag{5.2.9}$$

$$= v_g\left(-\frac{\pi}{2}\right)\left(k + \frac{\pi}{2}\right) + O\left(L^{-\frac{3}{2}}\right) \tag{5.2.10}$$

with approximation the momentum spread as $O(1/\sqrt{L})$ based on the Fourier coefficient $f(k)$. Note that we can ignore the global phase factor in (5.2.8) because we construct the circuit to synchronize the propagations of the $n + m$ wave packets as described in Chap. 5.3.

The Fourier coefficient (5.2.3) has a sharp peak at $-\pi/2$. As mentioned above, the roundabout gate supports such a wave packet as describe the next part; the scattering of the wave packet impinging on the subgraph is dominated by the S-matrix $\left[S_{r_j, r_{j_{\mathrm{in}}}, c}(k = -\pi/2)\right]$.

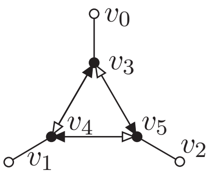
$\mathcal{A}\Bigg($ 

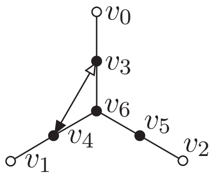
$\Bigg) =$

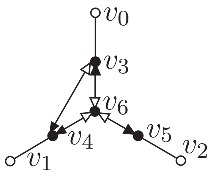
$$\left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & -i & i \\ 0 & 1 & 0 & i & 0 & -i \\ 0 & 0 & 1 & -i & i & 0 \end{array}\right) \quad \begin{array}{cc} B & C^{\dagger} \\ C & D \end{array}$$

Figure 5.2: Adjacency matrix of $\hat{G}_{\mathrm{R}_0}$ in Eq. (5.2.11). As described below Eq. (5.1.14), the adjacency matrix comprises four submatrices: $B$, $C$, $C^{\dagger}$, and $D$.

**Roundabout gate** We show that each of the following subgraphs demonstrates the functionality of the roundabout gate (see fig. 5.4 for concrete image):

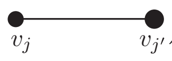
$\hat{G}_{\mathrm{R}_0} :=$ 

$, \hat{G}_{\mathrm{R}_1} :=$ 

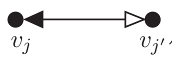
$, \hat{G}_{\mathrm{R}_2} :=$ 

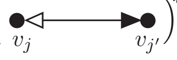
. (5.2.11)

Here, consider that each of the three vertices $v_0$, $v_1$, and $v_2$, which are marked white circles, is connected to a rail through which the wave packet impinges on the subgraph. The incoming wave packet scatters to another rail, either counter-clockwise or clockwise depending on its internal states, as described in the following paragraph. Additionally, two types of edges, components of these subgraphs, represent directed and undirected edges, whose weights associated with the walk are respectively 1 and $i$ (or $-i$). For example, adjacency matrices of a graph consisting of two vertices $v_j$ and $v_{j'}$ and one of the two edges are as follows:

$\mathcal{A}\Big($ 

$\Big) = |v_{x'}\rangle\langle v_x| + |v_x\rangle\langle v_{x'}|,$ (5.2.12)

$\mathcal{A}\Big($ 

$\Big) = \mathcal{A}\Big($ 

$\Big)^* = i|v_{x'}\rangle\langle v_x| - i|v_x\rangle\langle v_{x'}|.$ (5.2.13)

Now, we determine the S-matrices for the single-particle scattering on the subgraphs in (5.2.11) to describe that each subgraph functions as the roundabout gate for the wave packet (5.2.1). We obtain these S-matrices by substituting the adjacency matrices $A(\hat{G}_{\mathrm{R}_0})$, $A(\hat{G}_{\mathrm{R}_1})$, or $A(\hat{G}_{\mathrm{R}_2})$ into Eq. (5.1.14). Here, the adjacency matrix of $\hat{G}_{\mathrm{R}_0}$ for example can be represented as shown in Fig. 5.2.

Specifically, elements of the S-matrices, i.e., scattering coefficients in Eq.(5.1.2), for the particle with $|0\rangle_{\mathrm{c}_i}$ on the subgraph $\hat{G}_{\mathrm{R}_0}$ are given as follows:

$$S_{0,r_0,r_0}(k) = S_{0,r_1,r_1}(k) = S_{0,r_2,r_2}(k) = \frac{-e^{4ik}}{i + 2i\tan k}, \tag{5.2.14}$$

$$S_{0,r_1,r_0}(k) = S_{0,r_2,r_1}(k) = S_{0,r_0,r_2}(k) = \frac{-e^{3ik}(e^{ik} - i)}{2 - i\cot[k]}, \tag{5.2.15}$$

$$S_{0,r_0,r_1}(k) = S_{0,r_1,r_2}(k) = S_{0,r_2,r_0}(k) = \frac{-e^{3ik}(e^{ik} + i)}{2 - i\cot[k]}, \tag{5.2.16}$$

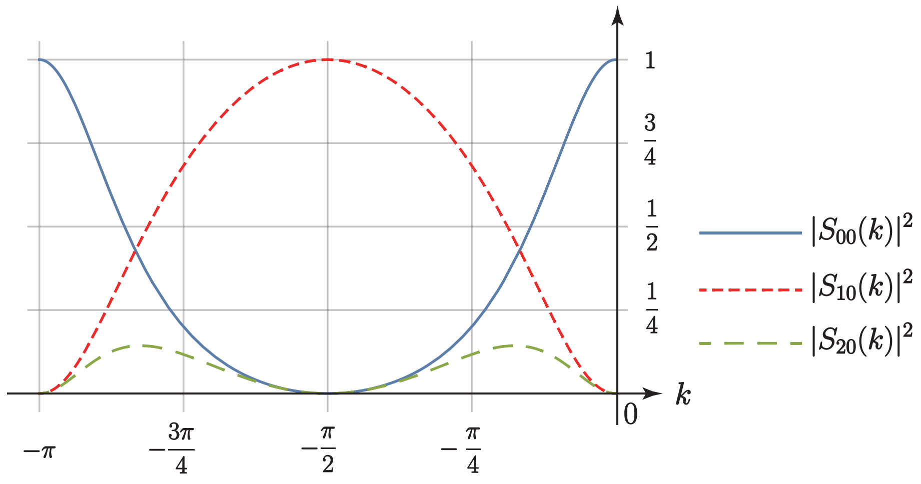


Figure 5.3: Graphical representation for the momentum dependence of the S-matrix of $\hat{G}_{\mathrm{R}_0}$.

where, $r_0$, $r_1$, and $r_2$ denote rails attached to the vertices respectively $v_0$, $v_1$, and $v_3$ of the subgraph $\hat{G}_{\mathrm{R}_0}$. We depict the momentum dependence of these scattering coefficients in Fig. 5.3. Note that the S-matrix for the particle $|1\rangle_{\mathrm{c}_i}$ is given by its transposition as proved in Eq. (5.1.16). Similarly, we obtain elements of the S-matrices for the subgraphs $\hat{G}_{\mathrm{R}_1}$ and $\hat{G}_{\mathrm{R}_2}$ as follows:

$$S_{0,r_0,r_0}(k) = S_{0,r_1,r_1}(k) = \frac{-e^{4ik}}{i + 2i\tan k}, S_{r_2,r_2,0}(k) = e^{2ik} S_{r_0,r_0,0}(k) \tag{5.2.17}$$

$$S_{0,r_1,r_0}(k) = \frac{-e^{3ik}(e^{ik} - i)}{2 - i\cot[k]}, S_{0,r_2,r_1}(k) = S_{0,r_0,r_2}(k) = ie^{ik} S_{0,r_1,r_0}(k), \tag{5.2.18}$$

$$S_{0,r_0,r_1}(k) = \frac{-e^{3ik}(e^{ik} + i)}{2 - i\cot[k]}, S_{0,r_1,r_2}(k) = S_{0,r_2,r_0}(k) = -ie^{ik} S_{0,r_0,r_1}(k). \tag{5.2.19}$$

The S-matrices indicate that the subgraphs can pass the wave packet to another rail in a counter-clockwise or clockwise direction with some accuracy, depending on the internal states. Specifically, an incoming wave packet, located at the distance $L$ on a semi-infinite rail $r_0$ from the roundabout gate, exits to either $r_1$ or $r_2$ depending on the internal states respectively $|0\rangle_{\mathrm{c}_q}$ or $|1\rangle_{\mathrm{c}_q}$ after a time interval $T = O(L)$:

$$\frac{1}{\sqrt{L}} \sum_{x=L}^{2L-1} e^{-i\frac{\pi}{2}x} |v_x^{r_0}\rangle_{\mathrm{v}_q} |c\rangle_{\mathrm{c}_q} \xrightarrow{e^{-iK_{\mathrm{G}}^{(1)}T}} \begin{cases} \frac{1}{\sqrt{L}} \sum\limits_{x=v_g(-\frac{\pi}{2})T}^{L-1+v_g(-\frac{\pi}{2})T} e^{i\frac{\pi}{2}x} |v_x^{r_1}\rangle_{\mathrm{v}_q} |0\rangle_{\mathrm{c}_q} + O(L^{-1/4}) \text{ if } c = 0 \\ \frac{1}{\sqrt{L}} \sum\limits_{x=v_g(-\frac{\pi}{2})T}^{L-1+v_g(-\frac{\pi}{2})T} |v_x^{r_2}\rangle_{\mathrm{v}_q} |1\rangle_{\mathrm{c}_q} + O(L^{-1/4}) \text{ if } c = 1 \end{cases}. \tag{5.2.20}$$

Here, as in Eq. (5.1.7), the kinetic term $K_{\mathrm{G}}^{(1)}$ for the particle is defined as

$$K_{\mathrm{G}}^{(1)} := A_{\mathrm{G}} \otimes |0\rangle\langle 0|_{\mathrm{c}_q} + A_{\mathrm{G}}^* \otimes |1\rangle\langle 1|_{\mathrm{c}_q},\ A_{\mathrm{G}} := A\left(G_{\mathrm{R}_\tau}\right) + \sum_{j=0}^{2} A(r_j). \tag{5.2.21}$$

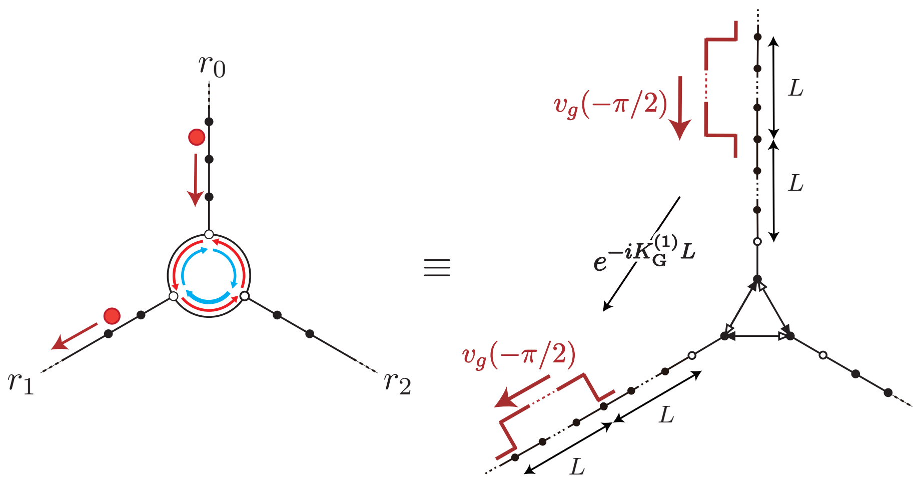


Figure 5.4: Equivalence between the subgraph $G_{\mathrm{R}_0}$ and the roundabout gate. The S-matrix of $G_{\mathrm{R}_0}$ indicates that the counter-clockwise or clockwise scattering, analogous of the roundabout gate.

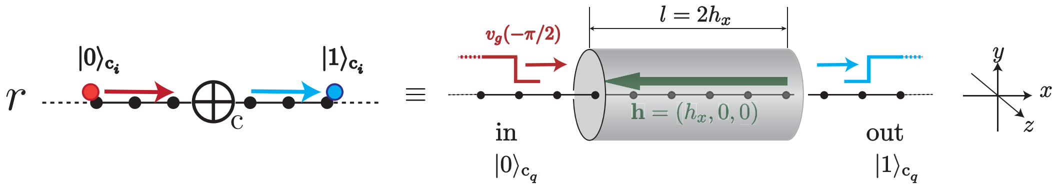


Figure 5.5: Conceptual image of the implementation for the NOT gate. After passing through the tunnel, the particle's internal states change according to (5.2.23).

for $\tau \in \{0, 1, 2\}$. Note that the time evolution of this wave packet, which is constructed from the superposition of the scattering plane waves (5.1.2), over a continuous time interval $t$ is obtained by

$$e^{-iK_{\mathrm{G}}^{(1)}t}\left(\frac{1}{\sqrt{L}}\sum_{x=L}^{2L-1} e^{-i\frac{\pi}{2}x}|v_x^{r_0}\rangle_{\mathrm{v}_q}|c\rangle_{\mathrm{c}_q}\right) = e^{-iK_{\mathrm{G}}^{(1)}t}\left(\int dk e^{ikL} f(k)|\mathrm{sc}_{r_0}(k)\rangle\right). \tag{5.2.22}$$

Additionally, the second term $O(L^{-1/4})$ in Eq. (5.2.20) is an error caused by the fact that the incoming wave packet is given by the superposition of the scattering plane waves with momenta closest to $k = -\pi/2$ (see Appendix in [28] for the estimation of this error bound).

**NOT gate** The NOT gate acting on the internal states may be realized by, for example, a model resembling a tunnel that generates an external localized field. This tunnel, as depicted in Fig. 5.5, is set on any rail $r$ and is parametrized by its length $l$ and an external field $\mathbf{h} = (h_x, h_y, h_z) \in \mathbb{R}^3$.

We can estimate that the evolution of the internal states after passing through the tunnel is described by $e^{-i(\sigma\cdot\mathbf{h})\frac{l}{2}}$ with an error of $O(L^{-1})$, because the wave packet propagates along the rail with a velocity of $2 + O(L^{-1})$ (see (5.2.7) and (5.2.9)). Here, $\sigma := (\sigma_x, \sigma_y, \sigma_z)$ is the Pauli matrices, with each acting on the internal states. Note that the above error will be absorbed in the second term in Eq. (5.2.8), the evolution by $e^{-iA(r)T}$ within a time interval $T = O(L)$.

For example, let us consider the implementation of the NOT gate by selecting parameters to satisfy $(\sigma \cdot \mathbf{h})l/2 = \sigma_x(4n+1)\pi/2 \in \mathrm{End}(V_{\mathrm{c}_q})$:

$$\mathrm{NOT}_{\mathrm{c}}(r) \equiv |r\rangle\langle r|_{\mathrm{b}_q} \otimes e^{-i\sigma_x \frac{4n+1}{2}\pi} = |r\rangle\langle r|_{\mathrm{b}_q} \otimes (-i)\left(|1\rangle\langle 1|_{\mathrm{c}_q} + |0\rangle\langle 0|_{\mathrm{c}_q}\right) \tag{5.2.23}$$

where $\mathrm{NOT}_{\mathrm{c}}(r) \in \mathrm{End}\left(v_{\mathrm{b}_q} \otimes v_{\mathrm{c}_q}\right)$. Here, due to the characteristic of the retrieval algorithm, we can ignore the phase factor $-i$. Namely, whereas the $n+m$ particles (wave packets) serving as the bucket with data acquire a phase factor that depends on the address, i.e., $(-i)^{\sum_{q=0}^{n-1} a_q}$, after the routing process $\mathcal{R}$, this phase factor is eliminated in the subsequent output process $\mathcal{R}^\dagger$.

**Controlled NOT gate** To implement the CNOT gate (4.2.2), we utilize the head-on scattering along a rail. Here, we choose the interaction strength of the interaction term (5.1.17) as $U = \pm 4$ for bosonic particles (reps. $U = \pm 2$ for fermionic) so that the two particles acquire a phase factor $P_+(-\frac{\pi}{2}, \frac{\pi}{2}) = \pm i$ (5.1.18) (resp. $P_-(-\frac{\pi}{2}, \frac{\pi}{2}) = \pm i$ (5.1.19)) after the scattering, as detailed in Sec. 5.1.

To execute the $\mathrm{CNOT}_{\mathrm{c,c'}}(r_{\mathrm{ctrl}}, r_{\mathrm{tgt}})$ (4.2.2) between the $i$th and $i'$th particles, we use two tunnels and head-on scatterings as

$$\left(|r_{\mathrm{ctrl}}\rangle_{\mathrm{b}_{q'}} |c'\rangle_{\mathrm{c}_{q'}}\right) \otimes \left(|r_{\mathrm{tgt}}\rangle_{\mathrm{b}_q} |c\rangle_{\mathrm{c}_q}\right) \tag{5.2.24}$$

$$\xrightarrow{H_{\mathrm{c}}(r_{\mathrm{tgt}})} \left(|r_{\mathrm{ctrl}}\rangle_{\mathrm{b}_{q'}} |c'\rangle_{\mathrm{c}_{q'}}\right) \otimes \frac{-i}{\sqrt{2}} \left\{|r_{\mathrm{tgt}}\rangle_{\mathrm{b}_q} \left(|0\rangle_{\mathrm{c}_q} + (-1)^c |1\rangle_{\mathrm{c}_q}\right)\right\} + O(L^{-1}) \tag{5.2.25}$$

$$\xrightarrow{\text{Scat.}} \left(|r_{\mathrm{ctrl}}\rangle_{\mathrm{b}_{q'}} |c'\rangle_{\mathrm{c}_{q'}}\right) \otimes \frac{-i}{\sqrt{2}} \left\{|r_{\mathrm{tgt}}\rangle_{\mathrm{b}_q} \left(|0\rangle_{\mathrm{c}_q} + (-1)^{c \oplus c'} |1\rangle_{\mathrm{c}_q}\right)\right\} + O(L^{-1/4}) \tag{5.2.26}$$

$$\xrightarrow{H_{\mathrm{c}}^\dagger(r_{\mathrm{tgt}})} \left(|r_{\mathrm{ctrl}}\rangle_{\mathrm{b}_{q'}} |c'\rangle_{\mathrm{c}_{q'}}\right) \otimes \left(|r_{\mathrm{tgt}}\rangle_{\mathrm{b}_q} |c \oplus c'\rangle_{\mathrm{c}_q}\right) + O(L^{-1/4}) \tag{5.2.27}$$

where $c, c' \in \{0, 1\}$ and $\oplus$ denotes exclusive disjunction, i.e., $c \oplus c' \in \{0, 1\}$. Here, we define the Hadamard operation $H_{\mathrm{c}}(r_{\mathrm{tgt}})$ as

$$H_{\mathrm{c}}(r_{\mathrm{tgt}}) := |r_{\mathrm{tgt}}\rangle\langle r_{\mathrm{tgt}}|_{\mathrm{b}_q} \otimes \frac{-i}{\sqrt{2}} \left[\left(|0\rangle + |1\rangle\right)\langle 0|_{\mathrm{c}_q} + \left(|0\rangle - |1\rangle\right)\langle 1|_{\mathrm{c}_q}\right] \in \mathrm{End}\left(V_{\mathrm{b}_q} \otimes V_{\mathrm{c}_q}\right) \tag{5.2.28}$$

which may be implemented by the tunnel whose parameters are adjusted to satisfy $(\sigma \cdot \mathbf{h})l/2 = \pi(4n+1)(\sigma_x + \sigma_y)/2\sqrt{2}$. The second term of Eq. (5.2.25) represents the error arising from higher-order contributions to the group velocity as described in the previous part which discusses the implementation of the NOT gate. Additionally, the term Scat. in Eq. (5.2.26) denotes head-on scatterings that occur twice, but only when the particles share identical internal states, specifically $|1\rangle_{\mathrm{c}_q}$ and $|1\rangle_{\mathrm{c}_{q'}}$. This interaction gives a $-1$ phase shift. To achieve these selective scatterings depending on internal states, we utilize the roundabout gates, as illustrated in Fig. 5.6.

For the convenience of the discussion in the next chapter, let us explicitly describe the time evolution of the head-on scattering between two wave packets, which occur on a semi-infinite rail $r$ of the CNOT gate

$$\left(\sum_{x=-L}^{-1} e^{-i\frac{\pi}{2}x}|v_x^r\rangle_{\mathrm{v}_{q'}}|1\rangle_{\mathrm{c}_{q'}}\right)\left(\sum_{x=0}^{L-1} e^{i\frac{\pi}{2}x}|v_x^r\rangle_{\mathrm{v}_q}|1\rangle_{\mathrm{c}_q}\right) \tag{5.2.29}$$

$$\xrightarrow{e^{-iH_r^{(2)}T/2}} \pm i\left(\sum_{x=-L}^{-(2L+1)} e^{i\frac{\pi}{2}x}|v_x^r\rangle_{\mathrm{v}_{q'}}|1\rangle_{\mathrm{c}_{q'}}\right)\left(\sum_{x=L}^{2L+1} e^{-i\frac{\pi}{2}x}|v_x^r\rangle_{\mathrm{v}_q}|1\rangle_{\mathrm{c}_q}\right) + O(L^{-1/4}) \tag{5.2.30}$$

where $H_r^{(2)}$, defined in Eq. (A.2.4), is the Hamiltonian for the two particles on the semi-infinite rail. Here, the time evolution of the two wave packets, which are constructed from the superposition of the scattering plane waves (A.2.1), over a continuous time interval $t$ is obtained by

$$e^{-iH_{\mathrm{G}}^{(2)}t}\left\{\left(\sum_{x=-L}^{-1} e^{-i\frac{\pi}{2}x}|v_x^r\rangle_{\mathrm{v}_{q'}}|1\rangle_{\mathrm{c}_{q'}}\right)\left(\sum_{x=0}^{L-1} e^{i\frac{\pi}{2}x}|v_x^r\rangle_{\mathrm{v}_q}|1\rangle_{\mathrm{c}_q}\right)\right\} \tag{5.2.31}$$

$$= e^{-iH_{\mathrm{G}}^{(2)}t}\left(\int\int dk_q dk_{q'} e^{-ik_qL} f(k_q) f(k_{q'})|\mathrm{sc}^{(2)}(k_q,k_{q'})\rangle\right), \tag{5.2.32}$$

where $k_q$ (resp. $k_{q'}$) and $f(k_q)$ (resp. $f(k_{q'})$) are the momentum and the Fourier coefficient (5.2.3) for the $q$th (resp. $q'$ th) particle. The second term in Eq. (5.2.30) represents an error from the scattering (see Appendix in [28] for the estimation of this error bound).

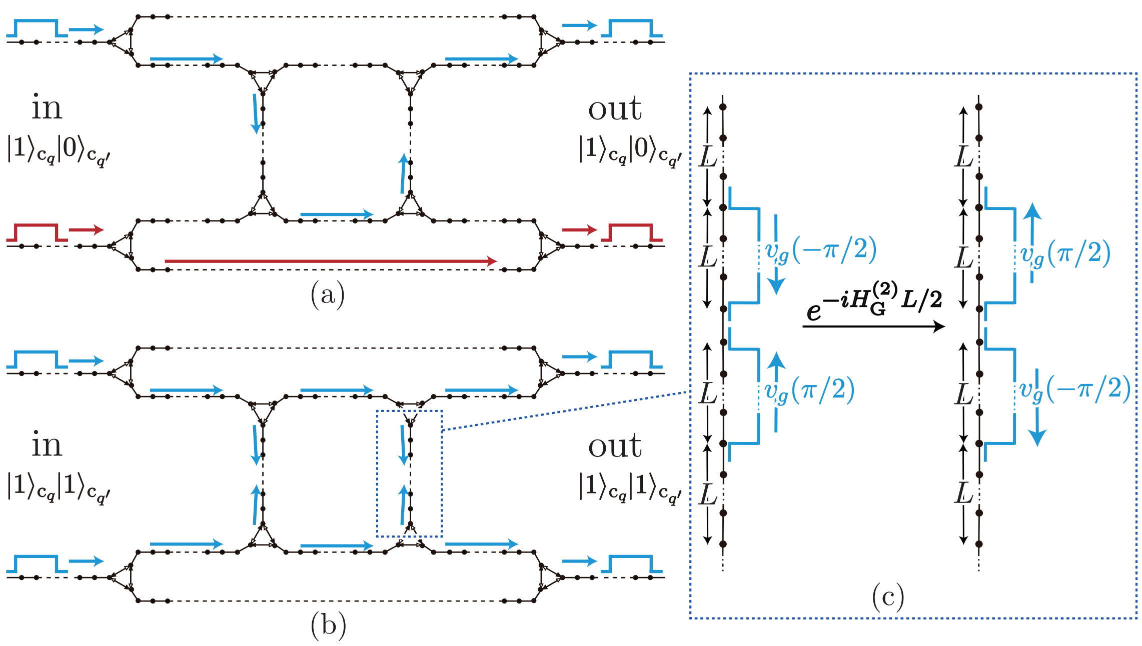

in
$|1\rangle_{c_q}|0\rangle_{c_{q'}}$
out
$|1\rangle_{c_q}|0\rangle_{c_{q'}}$
(a)
in
$|1\rangle_{c_q}|1\rangle_{c_{q'}}$
out
$|1\rangle_{c_q}|1\rangle_{c_{q'}}$
(b)
$L$
$v_g(-\pi/2)$
$v_g(\pi/2)$
$e^{-iH_G^{(2)}L/2}$
(c)

Figure 5.6: Implementation of the selective scatterings Scat. (5.2.26). Roundabout gates direct blue particles to the vertical rails. As a result, head-on scattering occurs twice only when both particles have blue internal states.

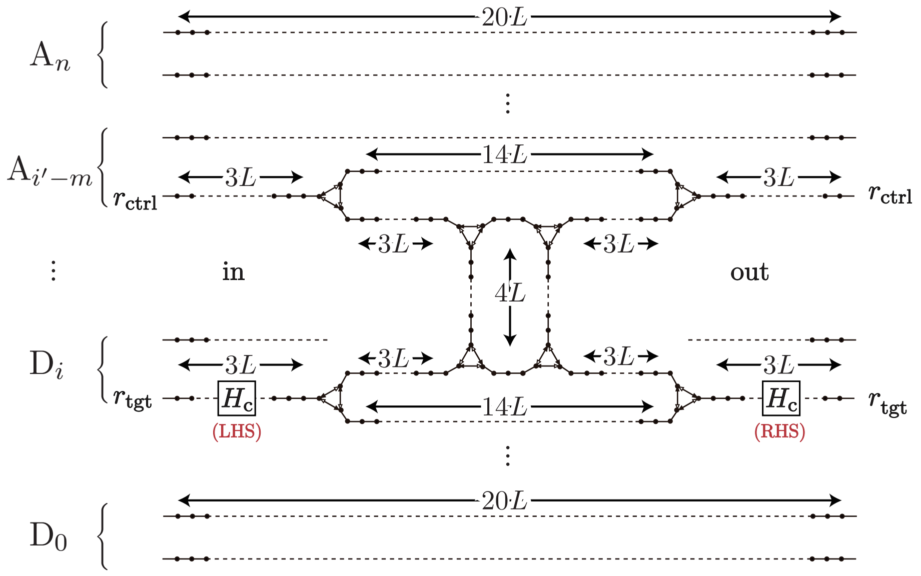


Figure 5.7: Implementation of the CNOT$_{c,c'}(j_{\rm ctrl}, j_{\rm tgt})$ (4.2.2). Two wave packets incident from the left through the rail $r_{\rm ctrl}$ and $r_{\rm tgt}$ undergo operations according to Eq. (5.2.25)-(5.2.27). As described in Sec. 5.3, we adjust the lengths of rails to synchronize the propagations of all the wave packets, where $L$ is the length of the wave packet.

## 5.3 Constructing a circuit with finite rails

In the previous section, we discussed the implementation of the qRAM assuming semi-infinite rails to consider each scattering independently. However, in this section, we discuss a circuit construction using finite rails to synchronize the propagation of the $n+m$ wave packets. The main motivation for the synchronization is to ensure the normal operation of the CNOT gates, which requires the following two contexts. Note that we give an estimation of the error due to finite-size effects at the end of this section.

In the first context, the right-hand side of the Hadamard gate (marked with (RHS) in Fig. 5.7) functions normally only if the wave packet corresponding to the target qubit enters this Hadamard gate at a specific time, regardless of its color. Namely, if the wave packet, which is in a superposition of red and blue, enters the Hadamard gate at a specific time regardless of its color, this wave packet settles into a single state by this Hadamard gate because its preceding state is written as:

$$\sum_x e^{-i\frac{\pi}{2}x}|v_x^{r_{\mathrm{tgt}}}\rangle_{\mathrm{v}_q}\left(\frac{|0\rangle_{\mathrm{c}_q}+|1\rangle_{\mathrm{c}_q}}{\sqrt{2}}\right). \tag{5.3.1}$$

Here, this wave packet has propagated along different routes as a spatial superposition, depending on its color One state passes through only horizontal rails, while the other state passes through both horizontal and vertical rails.

As shown in Fig. 5.7, we can satisfy the first context by adjusting the lengths of rails. Concretely, we set the length of the horizontal rail, through which the red wave packet passes, as $14L$. Here, we assume that the length of the horizontal rails and vertical rails, through which the blue wave packet passes, as $3L$ and $4L$, respectively. These two lengths assumed here originate from the single-particle scattering Eq. (5.2.20) (see Fig. 5.4) and two-particle scattering Eq. (5.2.30) (see Fig. 5.6) (c), respectively (let us discuss the effect of truncating the infinite rails to such finite rails at the end of this section). Note that the lengths of the rails, through which the wave packet corresponding to the target qubit passes, are also adjusted to the same to satisfy the following second context.

In the second context, two wave packets normally obtain the phase factor $-1$, undergoing the two-particle head-on scattering on each of the two vertical rails, only if they enter the vertical rails of the CNOT gate nearly simultaneously. We can satisfy this context by setting the length of some rails to synchronize the propagation of the $n+m$ wave packets per depth (step). Concretely, as shown in Fig. 5.7, we set the length of rails corresponding to the qubits not acted upon by the CNOT gate as $20L$. Because the $n+m$ wave packets proceed to the next depth in the circuit simultaneously, pairs of two wave packets definitely enter the CNOT gates simultaneously.

Finally, let us estimate an error due to finite-size effects, i.e., the effect of truncating the infinite rails to the finite, by using the truncation lemma proved by Childs et al. [28]. This lemma gives the following difference between the two cases of evolution, namely, the time evolution under the assumption of finite rails and that under semi-infinite rails, within a time interval $T=O(L)$:

$$\|\left(e^{-iH_{\mathrm{tot}}T}-e^{-i\tilde{H}T}\right)|\psi\rangle\| = O((n+m)^2L^{-1/4}). \tag{5.3.2}$$

Here, $H_{\mathrm{tot}}$ is a total Hamiltonian for the $n+m$ particles on the circuit constructed using finite rails. Additionally, $\tilde{H}$ is a Hamiltonian under the assumption of the semi-infinite rails.

Namely, $\tilde{H}$ can be $K^{(1)}_{\hat{G}}$ (5.2.21) or $H^{(2)}_{\mathrm{r}}$ (A.2.4). Depending on the Hamiltonian, the initial state $|\psi\rangle$ should be replaced with the initial state denoted in Eq.(5.2.20) or Eq.(5.2.29).

As a consequence, our implementation of the qRAM can control the error bound arbitrarily by choosing lengths of the wave packet and rails according to the circuit depth $(n\log(n+m))$ and number of qubits $(n+m)$. Namely, using Eq.(5.3.2) gives us an error bound of the circuit implementing the retrieval algorithm (as discussed in Chap. 6, this estimation is almost certainly not optimal):

$$\left\||\psi_{\mathrm{final}}\rangle - \sum_a |r^{(0,0)}\rangle_B|0\rangle_{\mathrm{C}}|a\rangle_{\mathrm{A}}|x^{(a)}\rangle_{\mathrm{D}}\rangle\right\| = O((n+m)^3 n\log(n+m)L^{-1/4}) \tag{5.3.3}$$

where $|\Psi_{\mathrm{final}}\rangle$ is the final state of the $n+m$ wave packets obtained from the end of the architecture. Here, based on Eq.(5.3.2), we replace the output of the CNOT gate (5.2.27) as

$$(|r_{\mathrm{ctrl}}\rangle_{\mathrm{b}_{q'}}|c'\rangle_{\mathrm{c}_{q'}}) \otimes (|r_{\mathrm{tgt}}\rangle_{\mathrm{b}_q}|c\oplus c'\rangle_{\mathrm{c}_q}) + O((n+m)^2L^{-1/4}). \tag{5.3.4}$$

Recall that these $n+m$ qubits pass through $O(n\log(n+m))$ CNOT gates, which are installed for operators of the controlled coinflippings $\{\mathcal{X}^{(l-1,w)}_{\mathrm{C,A}}\}$ (4.2.7) (the above error bound in (5.3.3) includes the one caused by the roundabout gates in the shift operators $\{S^{(l-1,w)}_{\mathrm{B,C}}\}$ (4.3.4), and NOT gates in the coinflips $\{\mathcal{X}^{(l,2w+1)}_{\mathrm{C}}\}$ (4.4.3), which can also be estimated by (5.3.2)).

## 5.4 Universal quantum computer compatible with our qRAM

Let us describe the physical implementation of the universal quantum computation, which employs the two-level continuous-time quantum walk, making this computation compatible with our qRAM. Here, we denote the two parallel rails as $r_{2q}$ and $r_{2q+1}$ through which the $q$th of dual-rail encoded qubit (particle) passes.

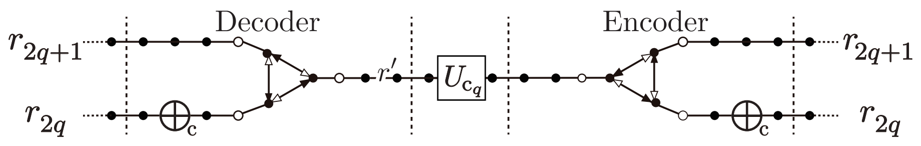
We implement the computation by combining two types of blocks: a block type 1 and type 2 (see fig. 5.8). The type 1 is a circuit implementing single-qubit gates and the type 2 is a circuit implementing controlled phase (CP) gates, each acting on qubits dual-rail encoded into the wave packets. Here, the CP-gate is defined as

$$\mathrm{CP}_{\mathrm{d,d'}}(r_{2q+1}, r_{2q'+1}) = -|r_{2q+1}\rangle\langle r_{2q+1}|_{\mathrm{b}_q} \otimes |r_{2q'+1}\rangle\langle r_{2q'+1}|_{\mathrm{b}_{q'}} \in \mathrm{End}(V_{\mathrm{b}_q}\otimes V_{\mathrm{b}_{q'}}). \tag{5.4.1}$$

Namely, the CP-gate gives a phase factor $-1$ between the $q$ and $q'$th particles only if both qubit values are 1, not 0. Note that we adjust the length of the rails incorporated into these blocks to synchronize propagations of wave packets serving as the dual-rail encoded qubits.

As in Eq. (4.5.3), each of the single-qubit gates in the type 1 is a combination of the encoder and decoder, where any unitary transformation acting on the internal states is also included if this gate is not an identity. The implementation of this gate is explicitly written as

$$(\mathcal{X}_{\mathrm{D}_q})^{x^{(a)}_q} \equiv \mathrm{Encoder}\cdot U_{\mathrm{c}_q}\cdot \mathrm{Decoder} \tag{5.4.2}$$

$=$ 

(5.4.3)

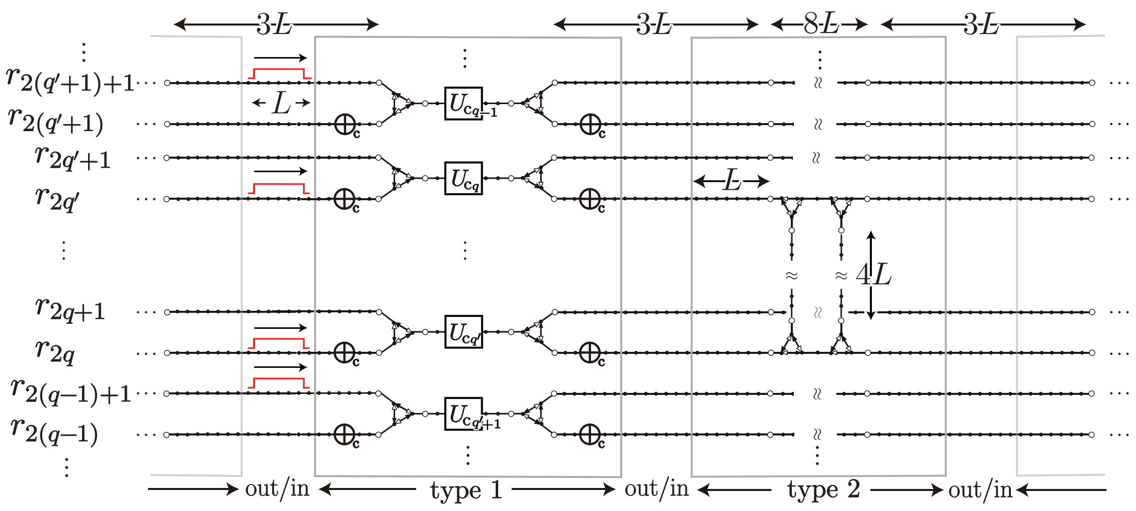
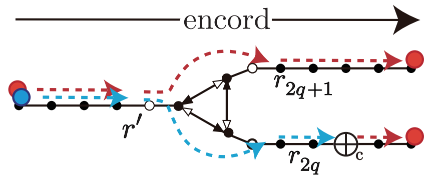


Figure 5.8: Circuit for universal quantum computation compatible with our qRAM.

where the unitary transformation $U_{c_q} \in \mathrm{End}(V_{c_q})$ may be realized by a tunnel whose length $l$ and the field $\mathbf{h}$ are parametrized properly. First, the decoder converts the dual-rail encoded qubit into the color state as shown in Eq. (4.5.4). The unitary gate $U_{c_q} \in \mathrm{End}(V_{c_q})$ then changes the color state into another color state, and finally this color state is converted to the dual-rail encoding qubit by the encoder:

$$|r'\rangle_{\mathrm{b}_q} \left(\alpha'|0\rangle_{\mathrm{c}_q} + \beta'|1\rangle_{\mathrm{c}_q}\right) : \qquad : \left(\alpha'|r_{2q}\rangle_{\mathrm{b}_q} + \beta'|r_{2q+1}\rangle_{\mathrm{b}_q}\right)|0\rangle_{\mathrm{c}_q} \tag{5.4.4}$$



where we denote the color states changed by $U_{\mathrm{c}}$ as $\alpha'|0\rangle_{\mathrm{c}_q} + \beta'|1\rangle_{\mathrm{c}_q}$ ($\alpha'$, $\beta' \in \mathbb{C}$) from $\alpha'|0\rangle_{\mathrm{c}_q} + \beta'|1\rangle_{\mathrm{c}_q}$ in Eq. (4.5.4).

As shown in Fig. 5.8, the CP-gate in the type 2 block, e.g., acting on the $q$- and $q'$th dual-rail encoded qubit, is implemented by attaching two vertical rails between the rails $2q+1$ and $2q'+1$ for $0 \leq q,\ q' \leq m-1$ using four roundabout gates. In this context, the wave packets whose qubit states are both 1 scatter on the two vertical rails respectively, acquiring the $-1$ phase shift. Here, to synchronize all wave packets, we adjust the length of the vertical rails, horizontal rails where these vertical rails are attached, and remaining horizontal rails in the blocks to $4L$, $L$, and $10L$, respectively.

We can also control the error bound arbitrary by choosing length of the wave packet and rail according to the circuit depth and number of the qubits. Namely, applying Eq. (5.3.2), we obtain the error bound of the quantum computation by finite size effect as

$$\||\psi_{\mathrm{final}}\rangle - |\psi_{\mathrm{ideal}}\rangle\| = O((n+m)^3 g L^{-1/4}) \tag{5.4.5}$$

where $g$ is the circuit depth (or number of blocks), $|\Psi_{\mathrm{final}}\rangle$ is the final state obtained from the right-hand side on the circuit, i.e., combination of the blocks, and $|\psi_{\mathrm{ideal}}\rangle$ is the desired output following the implemented quantum computation (algorithm).

# Chapter 6

# Summary and discussion

In this thesis, we have shown an implementation of qRAM along with introducing two kinds of quantum walks: the discrete-time quantum walk on the full binary tree, and the continuous-time multi-particle quantum walk with two-level internal states, applicable to either bosonic or fermionic particles.

Our discrete-time quantum walker, which is utilized for the data retrieval algorithm, moves to another path (bus) in a counter-clockwise or clockwise direction based on its coin states (colors) when the walker enters a bus of the tree. The algorithm retrieves a superposition of $O(2^n)$ $m$-qubit data, requiring only $n + m$ qubit resources allocated to the discrete-time quantum walker, which serves as a set of address and data registers. By guiding the registers to the desired memory units via the walker's coin states, we have eliminated the need for $2^{n-1} - 1$ quantum switches and then any time-dependent control in contrast to the bucket-brigade process. Namely, our algorithm requires no pre- or post-processing before or after the retrieval.

Our continuous-time quantum walker, which is utilized for the physical implementation, has two-level internal states and illustrates the internal state dependence of the scattering on a subgraph. Using this model, we have derived subgraphs that pass the incident quantum walker to adjacent rails in the counter-clockwise or clockwise direction, depending on this walker's internal states. We have then shown that utilizing these subgraphs, referred to as the roundabout gates, can implement a qRAM adopting our retrieval algorithm, with combining operations acting on the internal states of the continuous-time quantum walkers.

Notably, simply passing the $n+m$ wave packets as the continuous-time quantum walkers through a graph-like architecture automatically completes our retrieval algorithm and outputs the superposition of $O(2^n)$ $m$-qubit data. Note that the outputs are not entangled with the architecture. Recall that these $n + m$ wave packets serve as dual-rail encoded qubits, with $n$ qubits forming the address register and $m$ qubits forming the data register. Additionally, the architecture is a combination of the rails, subgraphs each demonstrating the functionality of the roundabout gate, and external localized fields acting on the internal states.

Moreover, we have estimated an error caused by finite-size effects as Eq. (5.3.3) and stated that our implementation can control such errors with arbitrary precision, potentially up to $O(1)$, in exchange for the space complexity of the architecture and time complexity for the data retrieval. Here, this estimation has been based on the discussion presented by A. M. Childs et al. in the context of universal quantum computation using multi-particle continuous-time quantum walk [28]. As they stated in the paper, their discussion and thus our estimation are almost certainly not as optimal. Namely, we can expect that the trade-off of both space and time complexities for the error precision would be mitigated compared to the present estimation. Note that as mentioned in the paragraph for the implementation of the NOT gate in Sec. 5.2, our estimation includes an error caused by an operation acting on the internal states.

Additionally, it should be noted that the particles enter a quantum superposition of two internal states, referred to as the colors red and blue in this thesis, and entangle each other only in each interval between before and after passing the $n + m$ wave packets to another rail through the roundabout gate. Namely, after the passing, these internal states are reset to a specific state corresponding to red. For example, we can see such a process of passing through the roundabout gates with changing color in Eq.(4.2.3), Eq.(4.3.2), and Eq.(4.4.1), in order. A similar statement can be made for the universal quantum computation in Sec. 5.4; the internal state of a particle changes to blue or a superposition of both red and blue only in each interval between before and after passing the particle through the single-qubit gate, a combination of the encoder, decoder, and operators acting on the internal states.

The main challenge for the experimental realization of our proposal is the establishment of directed edges; the subgraphs, which are analogous to the roundabout gate, consist of the directed edges (5.2.13) in addition to the undirected edges (5.2.12). Recall that the internal-state dependence of scattering on such subgraphs arises because a particle with the internal state $|0\rangle_{\mathrm{c}_q}$ evolves in continuous time $t$ according to $e^{-iA(G)t}$ (5.1.7), whereas a particle with $|1\rangle_{\mathrm{c}_q}$ evolves according to $e^{-iA^*(G)t}$ ($A(G)$ is the adjacency matrix of the whole graph $G$). In this context, a particle must obtain a coefficient (weight) of $i$ or $-i$ associated with a walk on the directed edge, depending on its internal states and direction of the walk, whereas the particle obtains no weight on the undirected edge. This challenge may be overcome by the Aharonov-Casher effect [59, 60].

# Acknowledgments

First, I would like to express my gratitude to my supervisor, Professor Kazumitsu Sakai. I appreciate his research instructions and friendly consultations since my laboratory assignment in Apr. 2018 until my Doctoral degree attainment in Mar. 2024. I would also like to express my gratitude to my assistant supervisor, Ryoko Yahagi. I appreciate her research instructions and kind support since the assignment in Apr. 2018 until her transfer effective as of Apr. 2023. Additionally, I would like to express my gratitude to all the colleagues in Professor Kazumitsu Sakai's laboratory who became my peers at any time between Apr. 2018 and Mar. 2024. I thank their cooperation, discussions, and enjoyable moments shared together.

Financial supports are also gratefully acknowledged. I would like to express my gratitude to my parents and grandparents, Yoshio Asaka, Takako Asaka, Mitsuji Asaka, and Fumi Asaka. I am grateful for their financial support from my first year of undergraduate studies in Apr. 2015 until the second year of my Master's program in Mar. 2021. I would also like to express my gratitude to Japan Student Services Organization. I appreciate their scholarship from my second year of undergraduate studies in Apr. 2017 until the second year of my Doctoral program in Mar. 2023. Subsequently, I would like to express my gratitude to Pascalia, Inc. I value their employment of me as a full-time worker from Apr. 2021 until Jul. 2022, despite being enrolled in the Doctoral program. Additionally, I would like to express my gratitude to the former Vice Chancellor of the Tokyo University of Science, Kazuyuki Watanabe. I am grateful for his scholarship, *Watanabe Kazuyuki Shogakukin*, in Sep. 2022. Further, I would like to express my gratitude to Hello Work. I am grateful for their unemployment insurance from Sep. 2022 until Mar. 2023. Finally, I would like to express my gratitude to Japan Society for the Promotion of Science (JSPS). I appreciate their fellowship, *Research Fellowship for Young Scientists (DC2)*, and their grant for scientific research, *Grant-in-Aid for JSPS Fellows No. 23KJ1962*, from Apr. 2023 until Mar. 2024.

In addition, my appreciation goes to the committee members, Tetsuro Nikuni, Akifumi Sako, Jaw-Shen Tsai, Fumiki Yoshihara, and again, Kazumitsu Sakai. The doctoral examinations, marked by their candid opinions and frank questions, have remained a precious experience.

Finally, I extend my deepest thanks to the Tokyo University of Science, particularly to the Faculty of Science Division II, for providing an excellent academic environment, essential experience, and the opportunity to engage in the field of physics.

# Appendix A

# Single- and two- particle scattering

We discuss the single-particle scattering on a subgraph and two-particle head-on scattering along a semi-infinite rail.

## A.1 Scattering matrix for a single particle on a subgraph

In this appendix, we derive the following matrix-form equation; solving this equation gives the relationship (5.1.14) (see [61, 62] for the detailed calculation):

$$\begin{pmatrix} B(\hat{G}) & C^\dagger(\hat{G}) \\ C(\hat{G}) & D(\hat{G}) \end{pmatrix} \begin{pmatrix} S_m + I_m \\ \Psi_{M-m} \end{pmatrix} + \begin{pmatrix} e^{ik} S_m + e^{-ik} I_m \\ 0_{M-m} \end{pmatrix} = 2\cos k_q \begin{pmatrix} S_m + I_m \\ \Psi_{M-m} \end{pmatrix}, \tag{A.1.1}$$

where the subgraph $\hat{G}$ has $M$ vertices and the first $m$ vertices are attached to $m$ rails, as shown in Fig. 5.1. Here, $0_{M-m}$ is $(M-m)\times(M-m)$ zero matrix and $I_m$, $S_m$, and $\Psi_{M-m}$ are matrices defined using the $M$ vectors

$$|v_0\rangle_{\mathrm{v}_q} (= |v_0^{r_0}\rangle_{\mathrm{v}_q}),\ |v_1\rangle_{\mathrm{v}_q} (= |v_0^{r_1}\rangle_{\mathrm{v}_q}),\ \cdots, |v_{m-1}\rangle_{\mathrm{v}_q} (= |v_0^{r_{m-1}}\rangle_{\mathrm{v}_q}), \tag{A.1.2}$$

$$|v_m\rangle_{\mathrm{v}_q},\ |v_{m+1}\rangle_{\mathrm{v}_q},\ \cdots,\ |v_{M-1}\rangle_{\mathrm{v}_q} \tag{A.1.3}$$

as

$$I_m := \sum_{j_{\mathrm{in}}=0}^{m-1} |v_{j_{\mathrm{in}}}\rangle\langle v_{j_{\mathrm{in}}}|_{\mathrm{v}_q} \ \in \mathbb{C}^{m\times m}, \tag{A.1.4}$$

$$S_m := S_0(k) = \sum_{j_{\mathrm{in}}=0}^{m-1} \sum_{j=0}^{m-1} S_{0,r_j,r_{j_{\mathrm{in}}}}(k) |v_j\rangle\langle v_{j_{\mathrm{in}}}|_{\mathrm{v}_q} \ \in \mathbb{C}^{m\times m}, \tag{A.1.5}$$

$$\Psi_{M-m} := \Psi_0(k) = \sum_{j_{\mathrm{in}}=0}^{m-1} \sum_{j=m}^{M-1} \Psi_{0,r_j,r_{j_{\mathrm{in}}}}(k) |v_j\rangle\langle v_{j_{\mathrm{in}}}|_{\mathrm{v}_q} \ \in \mathbb{C}^{(M-m)\times m}, \tag{A.1.6}$$

where $\Psi_0(k) \in \mathbb{C}^{(M-m)\times m}$ contains amplitudes of the bound state in Eq. (5.1.12), which can be written as

$$|\mathrm{bd}_{r_{j_{\mathrm{in}}}(k)}\rangle := \sum_{j=m}^{M-1} \left( \Psi_{0,v_j,r_{j_{\mathrm{in}}}} |v_j\rangle_{\mathrm{v}_q} |0\rangle_{\mathrm{c}_q} + \Psi_{1,v_j,r_{j_{\mathrm{in}}}} |v_j\rangle_{\mathrm{v}_q} |1\rangle_{\mathrm{c}_q} \right) \in \mathbb{C}^{M-m}. \tag{A.1.7}$$

Additionally, the elements $B(\hat{G})$, $C(\hat{G})$, and $D(\hat{G})$ are submatrices of the adjacency matrix $A(\hat{G})$, which can be written as

$$B(\hat{G}) := \sum_{j_{\mathrm{in}}=0}^{m-1} \sum_{j=0}^{m-1} \left( e^{i\theta_{v_j, v_{j_{\mathrm{in}}}}} |v_j\rangle\langle v_{j_{\mathrm{in}}}|_{\mathrm{v}_q} + e^{-i\theta_{v_j, v_{j_{\mathrm{in}}}}} |v_{r_{j_{\mathrm{in}}}}\rangle\langle v_j|_{\mathrm{v}_q} \right) \in \mathrm{End}(\mathbb{C}^{m\times m}), \tag{A.1.8}$$

$$C(\hat{G}) := \sum_{j_{\mathrm{in}}=0}^{m-1} \sum_{j=m}^{M-1} \left( e^{i\theta_{v_j, v_{j_{\mathrm{in}}}}} |v_j\rangle\langle v_{j_{\mathrm{in}}}|_{\mathrm{v}_q} \right) \in \mathrm{End}(\mathbb{C}^{(M-m)\times m}), \tag{A.1.9}$$

$$D(\hat{G}) := \sum_{i=m}^{M-1} \sum_{i'=m}^{M-1} \left( e^{i\theta_{v_{i'}, v_i}} |v_{i'}\rangle\langle v_i|_{\mathrm{v}_q} + e^{-i\theta_{v_{i'}, v_i}} |v_i\rangle\langle v_{i'}|_{\mathrm{v}_q} \right) \in \mathrm{End}(\mathbb{C}^{(M-m)\times(M-m)}). \tag{A.1.10}$$

Now, let us derive the matrix-form equation (A.1.1). From (5.1.12), (5.1.2) and (5.1.7), we obtain

$$\begin{aligned} &\left\{ A(\hat{G}) + \sum_{j=0}^{m-1} \sum_{x\geq 0} \left( |v_{x+1}^{r_j}\rangle\langle v_x^{r_j}|_{\mathrm{v}_q} + |v_x^{r_j}\rangle\langle v_{x+1}^{r_j}|_{\mathrm{v}_q} \right) \right\} \\ &\quad \left\{ \sum_{j=0}^{m-1} \sum_{x\geq 0} \left( \delta_{r_j, r_{j_{\mathrm{in}}}} e^{-ikx} + \sum_{j_{\mathrm{in}}=0}^{m-1} S_{0, r_j, r_{j_{\mathrm{in}}}}(k) e^{ikx} \right) |v_x^{r_j}\rangle_{\mathrm{v}_q} + \sum_{i=m}^{M-1} \Psi_{0, r_j, r_{j_{\mathrm{in}}}}(k) |v_i\rangle_{\mathrm{v}_q} \right\} \\ &= 2\cos k_q \left\{ \sum_{j=0}^{m-1} \sum_{x\geq 0} \left( \delta_{r_j, r_{j_{\mathrm{in}}}} e^{-ikx} + \sum_{j_{\mathrm{in}}=0}^{m-1} S_{0, r_j, r_{j_{\mathrm{in}}}}(k) e^{ikx} \right) |v_x^{r_j}\rangle_{\mathrm{v}_q} + \sum_{j=m}^{M-1} \Psi_{0, r_j, r_{j_{\mathrm{in}}}}(k) |v_j\rangle_{\mathrm{v}_q} \right\} \end{aligned} \tag{A.1.11}$$

where $|0\rangle_{\mathrm{c}_q}$ is omitted. Focusing on the states in Eq. (A.1.2) and Eq. (A.1.2) we obtain

$$\begin{aligned} &A(\hat{G}) \left\{ \sum_{j=0}^{m-1} \left( \delta_{r_j, r_{j_{\mathrm{in}}}} + S_{0, r_j, r_{j_{\mathrm{in}}}}(k) \right) |v_j\rangle_{\mathrm{v}_q} + \sum_{j=m}^{M-1} \Psi_{0, v_j, r_{j_{\mathrm{in}}}}(k) |v_j\rangle_{\mathrm{v}_q} \right\} \\ &\quad + \sum_{j=0}^{m-1} \left( \delta_{r_j, r_{j_{\mathrm{in}}}} e^{-ik} + S_{0, r_j, r_{j_{\mathrm{in}}}}(k) e^{ik} \right) |v_j\rangle_{\mathrm{v}_q} \\ &= 2\cos k_q \left\{ \sum_{j=0}^{m-1} \left( \delta_{r_j, r_{j_{\mathrm{in}}}} + S_{0, r_j, r_{j_{\mathrm{in}}}}(k) \right) |v_j\rangle_{\mathrm{v}_q} + \sum_{j=m}^{M-1} \Psi_{0, v_j, r_{j_{\mathrm{in}}}}(k) |v_j\rangle_{\mathrm{v}_q} \right\}. \end{aligned} \tag{A.1.12}$$

In conclusion, multiplying $\langle v_{j_{\mathrm{in}}}|_{\mathrm{v}_q}$ from right and taking the sum over the range $0 \leq j_{\mathrm{in}} \leq m-1$ gives Eq. (A.1.1).

## A.2 Phase shift through a two-particle head-on scattering along a rail

In this appendix, we describe that two wave packets characterized by plane waves with momentum $k_q \in (-\pi, 0)$ and $k_{q'} \in (0, \pi)$ acquire a phase factor by the head-on scattering with the same internal state along a semi-infinite rail. The bosonic or fermionic pairs acquire the phase factor in (5.1.18) or (5.1.19). Concretely, we show that the scattering coefficients $S^{(2)}_{\pm}(k_q,k_{q'})$ of the following plane wave are equal to these phase factors:

$$|\mathrm{sc}^{(2)}(k_q,k_{q'})\rangle := \sum_{x_q,x_{q'}} \psi^{(2)}_{\pm}(x_q,x_{q'};k_q,k_{q'})|v_{x_q}\rangle_{\mathrm{v}_q}|v_{x_{q'}}\rangle_{\mathrm{v}_{q'}}, \tag{A.2.1}$$

$$\psi^{(2)}_{\pm}(x_q,x_{q'};k_q,k_{q'}) := \begin{cases} e^{ik_qx_q+ik_{q'}x_{q'}} \pm S^{(2)}_{\pm}(k_q,k_{q'})e^{ik_{q'}x_q+ik_qx_{q'}} & (x_q<x_{q'}) \\ \frac{1}{2}(1\pm 1)\left(e^{ik_qx+ik_{q'}x}+S^{(2)}_{+}(k_q,k_{q'})e^{ik_{q'}x+ik_qx}\right) & (x_q=x_{q'}=x) \\ S^{(2)}_{\pm}(k_q,k_{q'})e^{ik_{q'}x_q+ik_qx_{q'}} \pm e^{ik_qx_q+ik_{q'}x_{q'}} & (x_q>x_{q'}) \end{cases}. \tag{A.2.2}$$

Here, the $+$ (resp. $-$) sign corresponds to the bosonic (resp. fermionic) pairs, which are symmetric (resp. antisymmetric) under the particle exchange. Note that the above plane wave satisfies the following two facts. The first is the conservation total of momentum $k_q+k_{q'}$ and energy $2(\cos k_q+\cos k_{q'})$. The second is the limitation of the interactions to the on-site (resp. nearest-neighbor) for the bosonic (resp. fermionic).

Concretely, we can determine the scattering coefficient $S^{(2)}_{\pm}(k_q,k_{q'})$ by solving the following Schrödinger equation:

$$\langle v_{x_q}|_{\mathrm{v}_q}\langle v_{x_{q'}}|_{\mathrm{v}_{q'}}H_G|\mathrm{sc}^{(2)}(k_q,k_{q'})\rangle = 2(\cos k_q+\cos k_{q'})\langle v_{x_q}|_{\mathrm{v}_q}\langle v_{x_{q'}}|_{\mathrm{v}_{q'}}|\mathrm{sc}^{(2)}(k_q,k_{q'})\rangle. \tag{A.2.3}$$

Focusing only on the two particles along the semi-infinite rail, we can alter the Hamiltonian $H_{\mathrm{G}}$ as

$$\begin{aligned} H^{(2)}_r = &\sum_x \left(|v_{x+1}\rangle\langle v_x|_{\mathrm{v}_q}+|v_x\rangle\langle v_{x+1}|_{\mathrm{v}_q}\right)\otimes I_{\mathrm{v}_{q'}} \\ &+ I_{\mathrm{v}_q}\otimes\sum_x\left(|v_{x+1}\rangle\langle v_x|_{\mathrm{v}_{q'}}+|v_x\rangle\langle v_{x+1}|_{\mathrm{v}_{q'}}\right)+U_G \end{aligned} \tag{A.2.4}$$

where $I_{\mathrm{v}_q}$ and $I_{\mathrm{v}_{q'}}$ are the identities. From (A.2.2) and (A.2.4), we obtain the following equations

$$\begin{aligned} &\psi^{(2)}_{+}(x,x+1;k_q,k_{q'})+\psi^{(2)}_{+}(x-1,x;k_q,k_{q'}) \\ &\quad+\psi^{(2)}_{+}(x,x+1;k_q,k_{q'})+\psi^{(2)}_{+}(x,x-1;k_q,k_{q'})+u\psi^{(2)}_{+}(x,x;k_q,k_{q'}) \\ &\quad\quad=2(\cos k_q+\cos k_{q'})\psi^{(2)}_{+}(x,x;k_q,k_{q'}) \end{aligned} \tag{A.2.5}$$

for the bosonic pairs and

$$\begin{aligned} &\psi^{(2)}_{\pm}(x-1,x+1;k_q,k_{q'})+\psi^{(2)}_{\pm}(x,x+2;k_q,k_{q'})+u\psi^{(2)}_{\pm}(x,x+1;k_q,k_{q'}) \\ &\quad=2(\cos k_q+\cos k_{q'})\psi^{(2)}_{+}(x,x+1;k_q,k_{q'}) \end{aligned} \tag{A.2.6}$$

for the fermionic pairs.

# Bibliography

[1] Ryo Asaka, Kazumitsu Sakai, and Ryoko Yahagi. Quantum random access memory via quantum walk. *Quantum Science and Technology*, 6(3):035004, 2021.

[2] Ryo Asaka, Kazumitsu Sakai, and Ryoko Yahagi. Two-level quantum walkers on directed graphs. i. universal quantum computing. *Physical Review A*, 107(2):022415, 2023.

[3] Ryo Asaka, Kazumitsu Sakai, and Ryoko Yahagi. Two-level quantum walkers on directed graphs. ii. application to quantum random access memory. *Physical Review A*, 107(2): 022416, 2023.

[4] Yakir Aharonov, Luiz Davidovich, and Nicim Zagury. Quantum random walks. *Physical Review A*, 48(2):1687, 1993.

[5] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Physical Review A*, 58(2):915, 1998.

[6] Andrew M Childs. On the relationship between continuous-and discrete-time quantum walk. *Communications in Mathematical Physics*, 294:581–603, 2010.

[7] Domenico D ’ Alessandro. Connection between continuous and discrete time quantum walks. from d-dimensional lattices to general graphs. *Reports on Mathematical Physics*, 66(1):85–102, 2010.

[8] Neil Shenvi, Julia Kempe, and K Birgitta Whaley. Quantum random-walk search algorithm. *Physical Review A*, 67(5):052307, 2003.

[9] Andris Ambainis, Julia Kempe, and Alexander Rivosh. Coins make quantum walks faster. *arXiv preprint quant-ph/0402107*, 2004.

[10] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

[11] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM (JACM)*, 51(4):595–605, 2004.

[12] Miklos Santha. Quantum walk based search algorithms. In *International Conference on Theory and Applications of Models of Computation*, pages 31–46. Springer, 2008.

[13] Manami Yamagishi, Naomichi Hatano, Ken-Ichiro Imura, and Hideaki Obuse. Proposal of multidimensional quantum walks to explore dirac and schrödinger systems. *Physical Review A*, 107(4):042206, 2023.

[14] Cristopher Moore and Alexander Russell. Quantum walks on the hypercube. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 164–178. Springer, 2002.

[15] Julia Kempe. Discrete quantum walks hit exponentially faster. *Probability theory and related fields*, 133(2):215–235, 2005.

[16] Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 50–59, 2001.

[17] Kanae Mukai and Naomichi Hatano. Discrete-time quantum walk on complex networks for community detection. *Physical Review Research*, 2(2):023378, 2020.

[18] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68, 2003.

[19] Shankar Balasubramanian, Tongyang Li, and Aram Harrow. Exponential speedups for quantum walks in random hierarchical graphs. *arXiv preprint arXiv:2307.15062*, 2023.

[20] Andrew M Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Physical Review A*, 70(2):022314, 2004.

[21] Shantanav Chakraborty, Leonardo Novo, and Jérémie Roland. Finding a marked node on any graph via continuous-time quantum walks. *Physical Review A*, 102(2):022227, 2020.

[22] Simon Apers, Shantanav Chakraborty, Leonardo Novo, and Jérémie Roland. Quadratic speedup for spatial search by continuous-time quantum walk. *Physical review letters*, 129(16):160502, 2022.

[23] Zhiguang Yan, Yu-Ran Zhang, Ming Gong, Yulin Wu, Yarui Zheng, Shaowei Li, Can Wang, Futian Liang, Jin Lin, Yu Xu, et al. Strongly correlated quantum walks with a 12-qubit superconducting processor. *Science*, 364(6442):753–756, 2019.

[24] Andrew M Childs. Universal computation by quantum walk. *Physical review letters*, 102 (18):180501, 2009.

[25] Neil B Lovett, Sally Cooper, Matthew Everitt, Matthew Trevers, and Viv Kendon. Universal quantum computation using the discrete-time quantum walk. *Physical Review A*, 81(4):042330, 2010.

[26] Michael S Underwood and David L Feder. Universal quantum computation by discontinuous quantum walk. *Physical Review A*, 82(4):042304, 2010.

[27] Shivani Singh, Prateek Chawla, Anupam Sarkar, and CM Chandrashekar. Universal quantum computing using single-particle discrete-time quantum walk. *Scientific Reports*, 11(1):11551, 2021.

[28] Andrew M Childs, David Gosset, and Zak Webb. Universal computation by multiparticle quantum walk. *Science*, 339(6121):791–794, 2013.

[29] Alberto Peruzzo, Mirko Lobino, Jonathan CF Matthews, Nobuyuki Matsuda, Alberto Politi, Konstantinos Poulios, Xiao-Qi Zhou, Yoav Lahini, Nur Ismail, Kerstin Wörhoff, et al. Quantum walks of correlated photons. *Science*, 329(5998):1500–1503, 2010.

[30] Kunkun Wang, Yuhao Shi, Lei Xiao, Jingbo Wang, Yogesh N Joglekar, and Peng Xue. Experimental realization of continuous-time quantum walks on directed graphs and their application in pagerank. *Optica*, 7(11):1524–1530, 2020.

[31] Michal Karski, Leonid Förster, Jai-Min Choi, Andreas Steffen, Wolfgang Alt, Dieter Meschede, and Artur Widera. Quantum walk in position space with single optically trapped atoms. *Science*, 325(5937):174–177, 2009.

[32] Maximilian Genske, Wolfgang Alt, Andreas Steffen, Albert H Werner, Reinhard F Werner, Dieter Meschede, and Andrea Alberti. Electric quantum walks with individual atoms. *Physical review letters*, 110(19):190601, 2013.

[33] Philipp M Preiss, Ruichao Ma, M Eric Tai, Alexander Lukin, Matthew Rispoli, Philip Zupancic, Yoav Lahini, Rajibul Islam, and Markus Greiner. Strongly correlated quantum walks in optical lattices. *Science*, 347(6227):1229–1233, 2015.

[34] Christof Weitenberg, Manuel Endres, Jacob F Sherson, Marc Cheneau, Peter Schauß, Takeshi Fukuhara, Immanuel Bloch, and Stefan Kuhr. Single-spin addressing in an atomic mott insulator. *Nature*, 471(7338):319–324, 2011.

[35] Hector Schmitz, Robert Matjeschk, Ch Schneider, Jan Glueckert, Martin Enderlein, Thomas Huber, and Tobias Schaetz. Quantum walk of a trapped ion in phase space. *Physical review letters*, 103(9):090504, 2009.

[36] Franziska Zähringer, Gerhard Kirchmair, Rene Gerritsma, Eenrique Solano, Rainer Blatt, and Christian F Roos. Realization of a quantum walk with one and two trapped ions. *Physical review letters*, 104(10):100503, 2010.

[37] Masaya Tamura, Takashi Mukaiyama, and Kenji Toyoda. Quantum walks of a phonon in trapped ions. *Physical Review Letters*, 124(20):200501, 2020.

[38] Ning Bao, Patrick Hayden, Grant Salton, and Nathaniel Thomas. Universal quantum computation by scattering in the fermi–hubbard model. *New Journal of Physics*, 17(9): 093028, 2015.

[39] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information.* Cambridge university press, 2010.

[40] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.

[41] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[42] Christoph Durr and Peter Hoyer. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*, 1996.

[43] Anton S Albino, Lucas Q Galvão, Ethan Hansen, Mauro Q Nooblath Neto, and Clebson Cruz. Quantum algorithm for finding minimum values in a quantum random access memory. *arXiv preprint arXiv:2301.05122*, 2023.

[44] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[45] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[46] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[47] Dominic W Berry and Andrew M Childs. Black-box hamiltonian simulation and unitary implementation. *arXiv preprint arXiv:0910.4157*, 2009.

[48] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Physical review letters*, 114(9):090502, 2015.

[49] Ryo Asaka, Kazumitsu Sakai, and Ryoko Yahagi. Quantum circuit for the fast fourier transform. *Quantum Information Processing*, 19:1–20, 2020.

[50] Connor T Hann, Gideon Lee, SM Girvin, and Liang Jiang. Resilience of quantum random access memory to generic noise. *Prx Quantum*, 2(2):020311, 2021.

[51] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Physical Review A*, 78(5):052310, 2008.

[52] Connor T Hann, Chang-Ling Zou, Yaxing Zhang, Yiwen Chu, Robert J Schoelkopf, Steven M Girvin, and Liang Jiang. Hardware-efficient quantum random access memory with hybrid quantum acoustic systems. *Physical Review Letters*, 123(25):250501, 2019.

[53] Kevin C Chen, Wenhan Dai, Carlos Errando-Herranz, Seth Lloyd, and Dirk Englund. Scalable and high-fidelity quantum random access memory in spin-photon networks. *PRX Quantum*, 2(3):030319, 2021.

[54] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-O ’ Connor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum ram. *New Journal of Physics*, 17(12):123010, 2015.

[55] Alexandru Paler, Oumarou Oumarou, and Robert Basmadjian. Parallelizing the queries in a bucket-brigade quantum random access memory. *Physical Review A*, 102(3):032608, 2020.

[56] Daniel K Park, Francesco Petruccione, and June-Koo Kevin Rhee. Circuit-based quantum random access memory for classical data. *Scientific reports*, 9(1):3949, 2019.

[57] Mohammed Zidan, Abdel-Haleem Abdel-Aty, Ashraf Khalil, Mahmoud Abdel-Aty, and Hichem Eleuch. A novel efficient quantum random access memory. *IEEE Access*, 9: 151775–151780, 2021.

[58] Zhao-Yun Chen, Cheng Xue, Tai-Ping Sun, Huan-Yu Liu, Xi-Ning Zhuang, Meng-Han Dou, Tian-Rui Zou, Yuan Fang, Yu-Chun Wu, and Guo-Ping Guo. An efficient and error-resilient protocol for quantum random access memory with generalized data size. *arXiv preprint arXiv:2303.05207*, 2023.

[59] Yakir Aharonov and Aharon Casher. Topological quantum effects for neutral particles. *Physical Review Letters*, 53(4):319, 1984.

[60] AA Zvyagin and IV Krive. Aharonov-casher effect in the hubbard model with repulsion. *Soviet physics, JETP*, 75(4):745–747, 1992.

[61] Andrew M Childs and David Gosset. Levinson's theorem for graphs ii. *Journal of Mathematical Physics*, 53(10), 2012.

[62] Andrew M Childs and DJ Strouse. Levinson's theorem for graphs. *Journal of mathematical physics*, 52(8), 2011.