Routing-free quantum RAM via controlled permutations

Simone Faro[1,†], Arianna Pavone[2,‡]

[1]Università di Catania, viale A. Doria n.6, 95125, Catania, Italy
[2]Università di Palermo, via Archirafi n.34, 90123, Palermo, Italy

Abstract

Quantum Random Access Memory (QRAM) is a fundamental primitive in many quantum algorithms, enabling coherent access to large datasets. Existing QRAM architectures implement memory access by routing address information through hierarchical switching networks, introducing structural complexity and sequential dependencies. In this work, we investigate whether routing is an intrinsic requirement for coherent quantum memory access. We answer this question in the negative by introducing a routing-free QRAM architecture based on controlled cyclic permutations of the memory register. Instead of propagating the address, the proposed construction rearranges the memory so that the requested cell is brought to a fixed location for local access. We present an explicit quantum circuit implementation with depth $O(\log n)$ and size $O(n \log n)$ in the standard quantum circuit model. Although it does not improve the asymptotic complexity of existing QRAM constructions, our results show that routing is not an intrinsic computational requirement for coherent memory access, establishing controlled global data movement as an alternative computational paradigm for QRAM and revealing a fundamental tradeoff between locality and parallelism.

Keywords

Quantum Random Access Memory, Quantum Circuits, Circuit Complexity, Data Movement, Parallel Quantum Algorithms

1. Introduction

Efficient access to large datasets is a fundamental requirement in many quantum algorithms, including quantum search, optimization, and quantum machine learning [1, 2]. In the standard computational model, this capability is provided by Quantum Random Access Memory (QRAM), which enables coherent access to data in superposition and plays a central role in many quantum algorithmic frameworks [3].

The most widely studied QRAM architectures, introduced by Giovannetti, Lloyd, and Maccone [4, 5], implement memory access by routing address information through hierarchical networks of quantum switches. A memory query activates a path of length $O(\log n)$ in a binary tree, yielding depth $O(\log n)$ and size $O(n)$ [5]. While these asymptotic bounds are well understood, they do not capture the computational organization induced by routing. Address-

ICTCS 2026: Italian Conference on Theoretical Computer Science, September 7–9, 2026, Udine, Italy

† Supported by University of Catania, progetto PIACERI 2024-2026 - Linea di Intervento 1

‡ Supported by Progetto Eurostart 2026

faro@dmi.unict.it (S. Faro); ariannamaria.pavone@unipa.it (A. Pavone)

0000-0001-5937-5796 (S. Faro); 0000-0002-8840-6157 (A. Pavone)

dependent communication paths introduce sequential dependencies, limit parallelism, and accumulate errors during memory access [6, 7, 8]. Furthermore, fault-tolerant analyses and system-level studies indicate that the cost of memory access may dominate that of the surrounding computation [9, 10, 11], while several quantum speedups based on QRAM assumptions admit classical counterparts under comparable access models [3, 12].

These limitations have motivated numerous alternatives, including circuit-based constructions [13, 14], hardware-oriented proposals [15, 16, 17], approximate and restricted models [18, 19], quantum-walk formulations [20], and algebraic approaches [21, 22]. Despite their differences, these constructions continue to realize memory access through address-dependent routing.

To make this distinction precise, we call a QRAM architecture *routing-based* if memory access is realized by propagating address-dependent information through intermediate communication elements that establish a path to the selected memory cell. Conversely, an architecture is *routing-free* if no such communication path is established and the address controls only global transformations acting directly on the memory register.

This naturally raises a fundamental question: is routing an intrinsic computational requirement for QRAM, or merely one possible architectural choice?

In this work, we answer this question in the negative. Rather than improving the asymptotic complexity of existing QRAM constructions, our goal is to identify the computational principles that are truly necessary for coherent quantum memory access. We show that routing is an implementation strategy rather than an intrinsic computational requirement.

To this end, we introduce a permutation-based QRAM architecture in which memory access is realized through controlled cyclic rotations of the memory register. Instead of propagating the address toward the data, the proposed construction reorganizes the memory so that the requested cell is brought to a fixed location, where it can be accessed by a local operation, yielding the conceptual decomposition

$$\textit{QRAM} = \textit{global permutation} + \textit{local read}.$$

Our construction exploits constant-depth implementations of cyclic rotations based on layers of disjoint SWAP gates [23, 24]. Combining these primitives with a binary decomposition of the address yields a controlled implementation consisting of a logarithmic sequence of highly parallel permutation layers.

The resulting QRAM implementation achieves depth $O(\log n)$ and size $O(n \log n)$ in the standard quantum circuit model without relying on routing structures. Although these bounds do not improve the asymptotic complexity of existing QRAM constructions, they demonstrate that the same QRAM functionality can be realized through a fundamentally different computational organization based on global data movement followed by local access. This organization exposes a tradeoff between locality and parallelism that is not reflected by standard circuit complexity measures: routing-based constructions concentrate computation along localized communication paths, whereas the proposed architecture distributes it across highly regular permutation layers acting uniformly on the memory register.

Overall, our results establish controlled global data movement as a viable alternative computational paradigm for QRAM, broadening the design space of coherent quantum memory access in the standard quantum circuit model.

2. Model and Problem formulation

We work in the standard quantum circuit model over qubits, where computations are described by unitary transformations decomposed into a sequence of one- and two-qubit gates. We assume familiarity with the basic notions of quantum computation.

Let $n \in \mathbb{N}$ denote the number of memory cells and let $d \geq 1$ be the number of qubits per cell. The memory is modeled as a quantum register $M = \bigotimes_{j=0}^{n-1} M_j$, where each $M_j \simeq (\mathbb{C}^2)^{\otimes d}$. The total memory register therefore consists of nd qubits. We denote by A the address register, consisting of $\ell = \lceil \log n \rceil$ qubits, and by O an output register of d qubits initialized to $|0\rangle^{\otimes d}$.

We consider the problem of implementing a coherent read-only QRAM access. Formally, we require a unitary transformation U_{QRAM} acting on $A \otimes O \otimes M$ such that, for any basis state $|i\rangle_A$ with $i \in \{0, \ldots, n-1\}$ and any computational basis configuration of the memory,

$$U_{\mathrm{QRAM}} : |i\rangle_A |0\rangle_O |M_0\rangle_{M_0} \cdots |M_{n-1}\rangle_{M_{n-1}} \longmapsto |i\rangle_A |M_i\rangle_O |M_0\rangle_{M_0} \cdots |M_{n-1}\rangle_{M_{n-1}},$$

and extended by linearity to arbitrary superpositions. In particular, the transformation preserves the content of the memory register and coherently correlates each address state with the corresponding cell content.

We introduce a family of unitary operators $\{\mathrm{ROT}_i\}_{i=0}^{n-1}$ acting on M, where ROT_i denotes the cyclic shift by i positions. Formally, for any tensor product basis state $|M_0\rangle \otimes \cdots \otimes |M_{n-1}\rangle$, we define

$$\mathrm{ROT}_i\big(|M_0\rangle \otimes \cdots \otimes |M_{n-1}\rangle\big) = |M_i\rangle \otimes |M_{i+1}\rangle \otimes \cdots \otimes |M_{n-1}\rangle \otimes |M_0\rangle \otimes \cdots \otimes |M_{i-1}\rangle,$$

with indices taken modulo n. Each ROT_i is a permutation unitary over the nd qubits of M.

We define the controlled permutation operator

$$U_{\mathrm{ROT}} = \sum_{i=0}^{n-1} |i\rangle \langle i|_A \otimes \mathrm{ROT}_i,$$

which acts on $A \otimes M$ and maps the content of the addressed cell to the first position of the memory register. For convenience, we identify U_{ROT} with its trivial extension to $A \otimes O \otimes M$, acting as the identity on the output register O. The QRAM read operation can then be implemented as the composition of U_{ROT}, a local extraction of the first cell, and $U_{\mathrm{ROT}}^\dagger$. More precisely, let U_{COPY} be a unitary acting on $O \otimes M_0$ such that $U_{\mathrm{COPY}} : |0\rangle_O |x\rangle_{M_0} \mapsto |x\rangle_O |x\rangle_{M_0}$ for all $x \in \{0,1\}^d$, and extended unitarily to the whole space. Then the unitary

$$U_{\mathrm{QRAM}} = U_{\mathrm{ROT}}^\dagger \cdot (I_A \otimes U_{\mathrm{COPY}} \otimes I_{M_1 \cdots M_{n-1}}) \cdot U_{\mathrm{ROT}}$$

correctly realizes the QRAM read functionality.

We evaluate quantum circuits in terms of size and depth. The size is defined as the total number of two-qubit gates in the circuit, while the depth is the minimum number of sequential layers of such gates required to implement it. We also consider the total number of qubits (circuit width). In addition, for a circuit decomposed into layers $L_1, \ldots, L_D$, we define the active set at layer t as $A_t = \{\text{qubits on which a gate in } L_t \text{ acts}\}$, and the peak activity as $\max_t |A_t|$. In routing-based QRAM constructions, each layer involves $O(\log n)$ active qubits corresponding to a single path in a routing tree, whereas in permutation-based constructions the operations act on $\Theta(n)$ qubits in parallel at each layer. This distinction will be relevant in our analysis.

3. Cyclic rotations via constant-depth circuits

We show that cyclic permutations of a quantum register can be implemented by constant-depth quantum circuits of linear size. The construction is based on a decomposition of cyclic shifts into compositions of involutive permutations, each realizable as a layer of disjoint SWAP gates. For simplicity, in this section we first describe rotations for one-qubit memory cells ($d = 1$); the general case follows by treating each memory cell as an indivisible block.

Let $M = (\mathbb{C}^2)^{\otimes n}$ be an n-qubit register with computational basis indexed by $\{0, \dots, n-1\}$. For $k \in \mathbb{Z}_n$, define the cyclic shift operator ROT_k as the unitary acting on basis states as

$$\mathrm{ROT}_k : |q_0\rangle \otimes \cdots \otimes |q_{n-1}\rangle \;\longmapsto\; |q_k\rangle \otimes |q_{k+1}\rangle \otimes \cdots \otimes |q_{n-1}\rangle \otimes |q_0\rangle \otimes \cdots \otimes |q_{k-1}\rangle\,,$$

where indices are taken modulo n. Equivalently, ROT_k corresponds to the permutation $\pi_k(i) = i + k \pmod n$ on wire indices.

We construct ROT_k as the composition of two involutive permutations. Let σ_1 and $\sigma_2^{(k)}$ be permutations of $\{0, \dots, n-1\}$ defined as follows:

$$\sigma_1(i) = -i \pmod n, \qquad \sigma_2^{(k)}(i) = k - i \pmod n.$$

Both σ_1 and $\sigma_2^{(k)}$ are involutions, i.e., $\sigma_1^2 = \sigma_2^{(k)2} = \mathrm{id}$. Moreover, their composition yields $(\sigma_2^{(k)} \circ \sigma_1)(i) = k + i \pmod n = \pi_k(i)$, which implies that $\mathrm{ROT}_k = U_{\sigma_2^{(k)}} \cdot U_{\sigma_1}$, where U_σ denotes the unitary implementing the permutation σ on the n wires.

Each involution σ can be decomposed into a product of disjoint transpositions. Therefore, the corresponding unitary U_σ can be implemented by a single layer of SWAP gates acting on disjoint pairs of qubits. This yields the following result.

Proposition 3.1. *For every $k \in \mathbb{Z}_n$, the unitary ROT_k can be implemented by a quantum circuit of depth 2 and size $\Theta(n)$ using only SWAP gates.*

Proof. The permutation σ_1 consists of the transpositions $(i, -i)$ for all $i \in \mathbb{Z}_n$ with $i \neq -i$, and possibly fixed points when $2i \equiv 0 \pmod n$. These transpositions are pairwise disjoint and can therefore be implemented in parallel using SWAP gates in a single layer. The same holds for $\sigma_2^{(k)}$, which consists of disjoint transpositions $(i, k-i)$. Since $\mathrm{ROT}_k = \sigma_2^{(k)} \circ \sigma_1$, the circuit consists of two such layers, yielding depth 2. The number of transpositions is $\Theta(n)$, implying linear size. □

We next highlight a structural property of the second layer. Define the two base involutions: $\sigma_2^{(0)}(i) = -i \pmod n$ and $\sigma_2^{(1)}(i) = 1 - i \pmod n$.

For any $k \in \mathbb{Z}_n$, let τ_t denote the cyclic shift $\tau_t(i) = i + t \pmod n$ with $t = \lfloor k/2 \rfloor$. Then it holds that $\sigma_2^{(k)} = \tau_t \circ \sigma_2^{(k \bmod 2)} \circ \tau_t^{-1}$. This shows that all second-layer permutations are obtained from two canonical patterns via conjugation by cyclic shifts. Consequently, the dependence on k reduces to a binary choice together with a cyclic relabeling of indices.

The circuit implementing ROT_k is therefore obtained as follows: a first layer implementing U_{σ_1}, followed by a second layer implementing $U_{\sigma_2^{(k)}}$. Each layer consists of disjoint SWAP gates and can be executed in one time step. The overall circuit has depth 2 and size $\Theta(n)$.

Input: $n \in \mathbb{N}$, $k \in \mathbb{Z}_n$
Output: Circuit C implementing ROT_k
1. $C \leftarrow$ empty circuit on n wires indexed by $\mathbb{Z}_n$
First layer (involution $\sigma_1(i) = -i$)
2. **for each** $i \in \mathbb{Z}_n$ with $i < -i$ **do**
3. add $\mathrm{SWAP}(i, -i)$ to C
Second layer (involution $\sigma_2^{(k)}(i) = k - i$)
4. **for each** $i \in \mathbb{Z}_n$ with $i < k - i$ **do**
5. add $\mathrm{SWAP}(i, k - i)$ to C
6. **return** C

Figure 1: Construction of a constant-depth circuit implementing ROT_k. Each layer corresponds to an involutive permutation decomposed into disjoint transpositions, and can therefore be applied in parallel.

This construction shows that cyclic shifts belong to the class of constant-depth permutation circuits. The high degree of regularity of the construction, together with the fact that all instances differ only by a cyclic relabeling of a constant number of base patterns, makes it particularly suitable for controlled implementations and for integration into larger architectures.

For completeness, we provide an explicit circuit construction implementing ROT_k, obtained by decomposing the underlying permutations into disjoint transpositions, which can be directly translated into parallel layers of SWAP gates (see Figure 1).

4. Controlled rotations and QRAM construction

We now describe an implementation of the controlled cyclic permutation

$$U_{\mathrm{ROT}} = \sum_{i \in \{0,\dots,n-1\}} |i\rangle\langle i|_A \otimes \mathrm{ROT}_i,$$

where A is the address register and ROT_i is the cyclic shift operator defined in the previous section.

Let $\ell = \lceil \log n \rceil$, and write the binary expansion of an address $i \in \{0, \dots, n-1\}$ as

$$i = \sum_{s=0}^{\ell-1} b_s 2^s, \qquad b_s \in \{0, 1\}.$$

For each $s \in \{0, \dots, \ell - 1\}$, let $\delta_s = 2^s \bmod n$.

Since cyclic shifts compose additively modulo n, for every valid address i we have

$$\mathrm{ROT}_i = \prod_{s=0}^{\ell-1} (\mathrm{ROT}_{\delta_s})^{b_s},$$

where the product may be taken in any order, since all cyclic shifts commute.

This suggests a decomposition of U_{ROT} into ℓ controlled fixed shifts. Let A_s denote the s-th address qubit, and define $V_s = |0\rangle\langle 0|_{A_s} \otimes I_M + |1\rangle\langle 1|_{A_s} \otimes \mathrm{ROT}_{\delta_s}$.

Then, on the subspace spanned by valid addresses $\{|i\rangle : 0 \le i < n\}$, $U_{\mathrm{ROT}} = \prod_{s=0}^{\ell-1} V_s$.

We next analyze how to implement each unitary V_s efficiently.

Lemma 4.1. *For every $m \ge 1$, the coherent fan-out transformation $F_m : |x\rangle\,|0\rangle^{\otimes m} \mapsto |x\rangle\,|x\rangle^{\otimes m}$, for all $x \in \{0,1\}$, can be implemented by a quantum circuit of depth $O(\log m)$ and size $O(m)$.*

Proof. The transformation is implemented by a balanced binary tree of CNOT gates. Starting from one control qubit, after one layer we obtain two copies, after two layers four copies, and so on. After $O(\log m)$ layers, m target ancillae have been initialized to the value of the input qubit. The total number of CNOT gates is $O(m)$. □

For each s, Proposition 1 provides a circuit for ROT_{δ_s} consisting of two layers of disjoint SWAP gates. Let $\mathrm{ROT}_{\delta_s} = W_{s,2} \cdot W_{s,1}$, where each $W_{s,j}$ is a product of $\Theta(n)$ disjoint SWAP gates. Denote by $E_{s,j}$ the set of transpositions appearing in $W_{s,j}$.

To implement V_s, we replace each SWAP gate in $W_{s,1}$ and $W_{s,2}$ by a controlled-SWAP gate controlled on the address bit A_s. Since a single control qubit cannot be used simultaneously in all these gates, we first distribute A_s coherently to sufficiently many ancillary copies.

Lemma 4.2. *For every $s \in \{0,\dots,\ell-1\}$, the unitary V_s can be implemented by a circuit of depth $O(1)$ and size $O(n)$, assuming the availability of one control ancilla for each transposition in $E_{s,1} \cup E_{s,2}$ initialized to the value of A_s.*

Proof. By Proposition 1, ROT_{δ_s} is the composition of two layers of disjoint SWAP gates. For each transposition in $E_{s,1}$ and $E_{s,2}$, apply a controlled-SWAP gate using a distinct ancillary copy of A_s as control. Since the target pairs are disjoint within each layer and the controls are distinct, all controlled-SWAP gates in a given layer can be applied in parallel. Each controlled-SWAP admits a constant-size, constant-depth decomposition into one- and two-qubit gates. Hence the resulting circuit has constant depth and size $O(n)$. □

We now combine the above ingredients into a construction of U_{ROT}. For each address bit A_s, we allocate $O(n)$ ancillae and use the coherent fan-out circuit of Lemma 2 to prepare one control copy for each transposition in the two-layer decomposition of ROT_{δ_s}. Since the fan-out circuits for different values of s act on disjoint ancilla registers, they can be executed in parallel.

Theorem 4.3. *The controlled cyclic permutation*

$$U_{\mathrm{ROT}} = \sum_{i \in \{0,\dots,n-1\}} |i\rangle\langle i|_A \otimes \mathrm{ROT}_i$$

can be implemented by a quantum circuit of depth $O(\log n)$ and size $O(n \log n)$.

Proof. The circuit is composed of three stages.

First, for each address bit A_s, prepare $O(n)$ coherent copies of its value using Lemma 2. Since all ℓ fan-out circuits act on disjoint ancilla registers, they can be performed in parallel. Their total depth is therefore $O(\log n)$, and their total size is $O(n\ell) = O(n \log n)$.

Second, for each $s = 0,\dots,\ell-1$, apply the controlled fixed shift V_s using the copied ancillae, as described in Lemma 3. Since the unitaries V_s act on the same memory register, they must be

applied sequentially. Each has depth $O(1)$ and size $O(n)$, so the total contribution of this stage is depth $O(\ell) = O(\log n)$ and size $O(n\ell) = O(n \log n)$.

Third, uncompute all ancilla copies by reversing the fan-out circuits. As in the first stage, this contributes depth $O(\log n)$ and size $O(n \log n)$.

Summing the three contributions yields total depth $O(\log n) + O(\log n) + O(\log n) = O(\log n)$, and total size $O(n \log n) + O(n \log n) + O(n \log n) = O(n \log n)$. □

We now use U_{ROT} to implement QRAM read. Let U_{COPY} denote the local extraction unitary defined in Section 2. Then $U_{\mathrm{QRAM}} = U_{\mathrm{ROT}}^{\dagger} \cdot (I_A \otimes U_{\mathrm{COPY}} \otimes I_{M'}) \cdot U_{\mathrm{ROT}}$, where M' denotes the memory register excluding the first cell.

Theorem 4.4. *Assume that each memory cell consists of $d = O(1)$ qubits. Then there exists a permutation-based implementation of QRAM read on n memory cells with circuit depth $O(\log n)$ and size $O(n \log n)$.*

Proof. By the previous theorem, each occurrence of U_{ROT} has depth $O(\log n)$ and size $O(n \log n)$. Since d is constant, the local unitary U_{COPY} has constant depth and size. Therefore the asymptotic complexity of U_{QRAM} is dominated by the two controlled rotation circuits, yielding depth $O(\log n)$ and size $O(n \log n)$. □

This construction provides a coherent implementation of QRAM access entirely through controlled global permutations of the memory register, without routing or path-dependent information propagation.

Remark. The depth bound above is obtained by using $O(n \log n)$ ancillary qubits to distribute all address bits in parallel. If one instead reuses ancillae separately for each stage V_s, the width can be reduced to $O(n)$ at the cost of increasing the depth to $O(\log^2 n)$. Thus, the construction admits an explicit width-depth tradeoff.

5. Complexity and comparison

In this section, we analyze the complexity of the proposed QRAM construction and compare it with routing-based architectures. Our goal is to characterize both its asymptotic performance and its structural properties.

5.1. Depth and Size

The core primitive underlying our construction is the controlled cyclic permutation

$$U_{\mathrm{ROT}} = \sum_{i \in \mathbb{Z}_n} |i\rangle \langle i|_A \otimes \mathrm{ROT}_i.$$

As shown in Section 4, each index $i \in \mathbb{Z}_n$ can be written in binary as $i = \sum_{s=0}^{\ell-1} b_s 2^s$, with $\ell = \lceil \log n \rceil$, and the corresponding rotation decomposes as

$$\mathrm{ROT}_i = \prod_{s=0}^{\ell-1} (\mathrm{ROT}_{\delta_s})^{b_s}, \qquad \delta_s = 2^s \bmod n.$$

Since cyclic rotations commute, the order of the product is irrelevant.

This yields a decomposition of U_{ROT} into $\ell = \Theta(\log n)$ controlled fixed shifts, each conditioned on a single address bit. Each unitary ROT_{δ_s} admits a constant-depth implementation using two layers of disjoint SWAP gates, which are promoted to controlled-SWAP gates.

To enable parallel execution of these gates, each address bit is coherently distributed to $\Theta(n)$ ancillary qubits using a fan-out circuit. This introduces a logarithmic-depth overhead.

The resulting circuit consists of three stages: (i) a fan-out stage distributing the ℓ address bits, (ii) ℓ controlled permutation layers applied sequentially, and (iii) an uncomputation stage removing the ancillary information.

The fan-out and uncomputation stages each have depth $O(\log n)$, while each controlled permutation layer has constant depth. Since $\ell = \Theta(\log n)$ such layers must be applied sequentially, the overall depth is $O(\log n)$.

The size is dominated by the controlled permutation layers. Each layer consists of $\Theta(n)$ controlled-SWAP gates acting on disjoint pairs of qubits, and $\Theta(\log n)$ such layers are required. Therefore, the total size is $O(n \log n)$.

When memory cells contain d qubits, each SWAP acts on d qubits, yielding size $O(nd \log n)$ and depth $O(\log n + d)$.

We emphasize that the logarithmic overhead in both depth and size arises from the binary decomposition of the address and the sequential application of $\Theta(\log n)$ global permutation layers, while each individual permutation remains constant-depth and linear-size.

5.2. Comparison with routing-based QRAM

We compare the proposed construction with standard routing-based QRAM architectures, such as the fan-out and bucket-brigade models [5, 4].

Routing-based QRAM implements memory access by propagating the address through a hierarchical network, activating a path of length $O(\log n)$ that selects the target cell. This yields depth $O(\log n)$ and size $O(n)$.

In contrast, the proposed construction eliminates routing and instead applies controlled global permutations of the memory register. Rather than propagating the address, the data is rearranged so that the selected cell is brought to a fixed location.

From an asymptotic perspective, both approaches achieve logarithmic depth. However, the permutation-based construction incurs an additional logarithmic factor in size, yielding $O(n \log n)$, due to the need to apply $\Theta(\log n)$ global permutation layers.

More significantly, the two approaches differ in their computational structure. Routing-based architectures are inherently path-based: at each layer, only $O(\log n)$ qubits are active, corresponding to a routing path. In contrast, the permutation-based construction applies uniform transformations to $\Theta(n)$ qubits in parallel at each layer.

Table 1 summarizes the main characteristics of representative QRAM architectures. Besides the standard complexity measures of depth and size, we also compare their structural organization. In particular, the *regularity* of a construction refers qualitatively to the uniformity and repetitiveness of its circuit structure, namely whether memory access is realized by repeatedly applying the same pattern of gates across the entire memory register or by address-dependent routing operations whose location varies with the access path. While this notion is not intended

Table 1
Comparison of representative QRAM architectures.

Model	Depth	Size	Routing	Global Ops	Regularity	Active Qubits
Fan-out QRAM [5]	$O(\log n)$	$O(n)$	Yes	No	Low	$O(\log n)$
Bucket-brigade [4]	$O(\log n)$	$O(n)$	Yes	No	Medium	$O(\log n)$
FF-QRAM [13]	$O(\log n)$	$O(n)$	Yes	No	Medium	$O(\log n)$
Circuit QRAM [14]	$O(\log n)$	$O(n)$	Yes	No	Medium	$O(\log n)$
RF-QRAM (this work)	$O(\log n)$	$O(n \log n)$	No	Yes	High	$\Theta(n)$

as a formal complexity measure, it highlights architectural differences that are not reflected by asymptotic depth and size alone.

Overall, the proposed architecture should be viewed not as an asymptotic improvement, but as an alternative paradigm that replaces address propagation with data movement, exposing a tradeoff between locality and parallelism.

5.3. Structural and space-time tradeoffs

Standard complexity measures such as size and depth do not fully capture the structural differences between routing-based and permutation-based QRAM constructions. To make this distinction precise, we introduce additional measures that quantify the distribution of computational activity.

Peak activity. Let a circuit be decomposed into layers $L_1, \ldots, L_D$ of disjoint two-qubit gates. For each layer L_t, let $A_t = \{\text{qubits acted upon by gates in } L_t\}$. The *peak activity* is defined as $\max_{1 \le t \le D} |A_t|$.

Proposition 5.1. *Consider a QRAM access circuit on n memory cells. Routing-based architectures have peak activity $O(\log n)$, whereas the permutation-based construction has peak activity $\Theta(n)$.*

Proof. In routing-based architectures, memory access proceeds along a root-to-leaf path in a binary tree of depth $O(\log n)$. At each layer, only qubits corresponding to nodes on this path are involved, yielding $|A_t| = O(\log n)$. In contrast, in the permutation-based construction, each permutation layer consists of $\Theta(n)$ disjoint SWAP or controlled-SWAP gates acting on the memory register, covering all memory cells, and therefore $|A_t| = \Theta(n)$. □

Space-time volume. Let W and D denote the width and depth of a circuit. The space-time volume is defined as $\mathrm{Vol} = W \cdot D$.

Proposition 5.2. *A single QRAM access has space-time volume $O(n \log^2 n)$ in the depth-optimized permutation-based construction, and $O(n \log n)$ in routing-based architectures.*

Proof. In routing-based architectures, the width is $\Theta(n)$ and the depth is $O(\log n)$, yielding volume $O(n \log n)$. In the permutation-based construction, the depth is $O(\log n)$, but the width is dominated by the ancillary qubits required to distribute the address bits, which is $O(n \log n)$ as discussed in Section 4. Therefore, the volume is $O(n \log n) \cdot O(\log n) = O(n \log^2 n)$. □

While routing-based architectures achieve lower space-time volume, the distribution of computational effort differs significantly. Routing-based constructions realize memory access through a sequence of localized operations along a path, resulting in low peak activity and temporally distributed computation. In contrast, the permutation-based construction concentrates the computation into $\Theta(\log n)$ layers, each involving $\Theta(n)$ qubits in parallel. These observations highlight a fundamental tradeoff between locality and parallelism: routing-based architectures achieve spatial locality at the cost of sequential dependence, whereas permutation-based constructions eliminate routing dependencies by distributing computation globally.

5.4. Write operation

The permutation-based framework naturally extends to support coherent write operations. Let W be a d-qubit register containing the value to be written, and let $M = \bigotimes_{j=0}^{n-1} M_j$ be the memory register. The write operation is defined as a unitary mapping

$$|i\rangle_A |w\rangle_W |M_0\rangle \cdots |M_{n-1}\rangle \longmapsto |i\rangle_A |w\rangle_W |M_0\rangle \cdots |w\rangle_{M_i} \cdots |M_{n-1}\rangle ,$$

extended by linearity.

Let U_{ROT} denote the controlled cyclic permutation. The write operation is implemented as $U_{\mathrm{WRITE}} = U_{\mathrm{ROT}}^{\dagger} \cdot (I_A \otimes U_{\mathrm{UPD}} \otimes I_{M'}) \cdot U_{\mathrm{ROT}}$, where U_{UPD} is a reversible update acting on $W \otimes M_0$. A standard realization of U_{UPD} is given by $|w\rangle_W |x\rangle_{M_0} \mapsto |w\rangle_W |x \oplus w\rangle_{M_0}$, implemented using d parallel CNOT gates. Correctness follows from the fact that U_{ROT} maps the addressed cell M_i to position M_0, where the update is applied, and $U_{\mathrm{ROT}}^{\dagger}$ restores the original ordering. The construction is unitary and preserves coherence.

The complexity is dominated by the controlled permutation, yielding depth $O(\log n + d)$, and size $O(n \log n + d)$. Thus, the permutation-based approach supports both read and write operations within a unified unitary framework, where memory access is realized via global data movement followed by local operations.

5.5. Noise considerations

Beyond asymptotic complexity, the structural properties of the proposed construction suggest qualitatively different noise behavior compared to routing-based QRAM architectures. Routing-based constructions implement memory access through a sequence of operations along a path of length $O(\log n)$, leading to error accumulation across dependent layers [6, 7]. In contrast, the permutation-based construction replaces sequential routing with a small number of highly parallel layers acting on the entire memory register. As a result, errors are not confined to a specific path, but may instead be distributed across $\Theta(n)$ qubits involved in each access. From a circuit-level perspective, this suggests a tradeoff between temporal and spatial error accumulation: routing-based architectures concentrate errors along sequential operations, whereas permutation-based constructions expose a larger number of qubits to noise simultaneously but over fewer steps. We emphasize that this observation is qualitative, and that the overall robustness depends on physical parameters such as error rates, connectivity constraints, and the availability of error correction. A precise quantitative analysis, particularly in fault-tolerant settings, remains an open problem.

6. Block-decomposed permutation-based access

We describe a variant of the permutation-based QRAM construction based on a decomposition of the address into block and intra-block components. This construction does not improve the complexity, but provides additional insight into the structure of permutation-based access.

Block decomposition. Let the memory register be partitioned into $B = \frac{n}{k}$ blocks of size k. The address $i \in \mathbb{Z}_n$ is decomposed as $i = b \cdot k + \ell$, where $b \in \{0, \dots, B-1\}$ identifies the block and $\ell \in \{0, \dots, k-1\}$ identifies the position within the block.

Access procedure. The access operation is implemented in two stages.

First, we apply, in parallel to all blocks, a controlled cyclic rotation conditioned on the intra-block address ℓ. Formally, for each block $M^{(j)}$, we apply the transformation $M^{(j)} \mapsto \mathrm{ROT}_\ell(M^{(j)})$, so that in every block the element at position ℓ is moved to the first position.

Second, we consider the register consisting of the first cell of each block, which contains B candidate elements. We apply a controlled cyclic rotation on this register, conditioned on the block index b, bringing the desired element to the first position of the global memory.

Correctness. After the first stage, for each block j, the element originally at position ℓ within that block is located at its first position. In particular, the target element M_i is located at the first position of block b. The second stage rotates the first elements of all blocks so that the first element of block b is moved to the first position of the global memory. Therefore, the overall transformation correctly implements the desired memory access.

Complexity. The first stage consists of $B = n/k$ controlled rotations on blocks of size k, applied in parallel. Each controlled rotation on k elements can be implemented, using the construction of Section 4, with size $O(k \log k)$ and depth $O(\log k)$. Therefore, the total size of the first stage is $O\big(\frac{n}{k} \cdot k \log k\big) = O(n \log k)$, while its depth is $O(\log k)$. The second stage is a controlled rotation on $B = n/k$ elements, which can be implemented with size $O\big(\frac{n}{k} \log \frac{n}{k}\big)$ and depth $O\big(\log \frac{n}{k}\big)$. Therefore, the total size of the construction is $O\big(n \log k + \frac{n}{k} \log \frac{n}{k}\big)$, while the total depth is $O\big(\log k + \log \frac{n}{k}\big) = O(\log n)$.

Discussion. For all choices of k, the asymptotic complexity remains $O(n \log n)$ in size and $O(\log n)$ in depth. However, the construction reveals a structural decomposition of the access operation into a local phase, acting independently on each block, and a global phase, acting on a reduced representation of the memory. This decomposition provides an intermediate point between fully global permutation-based access and localized routing-based schemes. In particular, it shows that different components of the address can be processed at different spatial scales, inducing a hierarchy of data-movement operations.

While this variant does not improve asymptotic bounds, it may be relevant in settings where local operations are significantly cheaper than global ones, or where memory is physically organized in modular blocks. A detailed investigation of such scenarios is left for future work.

The above construction suggests a more general perspective on quantum memory access based on hierarchical decompositions of the address space. In this view, access operations can be structured as compositions of permutation primitives acting at multiple spatial scales, interpolating between fully global transformations and localized routing mechanisms.

7. Discussion and conclusion

We have presented a permutation-based approach to coherent quantum memory access that replaces address routing with controlled global transformations of the memory register. Rather than propagating address information through a routing network, the proposed construction reorganizes the memory so that the addressed cell is brought to a fixed location, where it can be accessed by a local operation.

From a complexity-theoretic perspective, the resulting QRAM implementation achieves depth $O(\log n)$ and size $O(n \log n)$ in the standard quantum circuit model. Although this does not improve the asymptotic complexity of existing routing-based architectures, it demonstrates that routing is not an intrinsic computational requirement for implementing coherent QRAM access. The same functionality can instead be realized through controlled global data movement followed by local extraction.

The key distinction therefore lies not in asymptotic complexity, but in the organization of the computation. Routing-based constructions realize memory access by establishing address-dependent communication paths through intermediate routing elements, leading to localized operations with relatively low activity per circuit layer. In contrast, the proposed construction deliberately replaces address routing with global data movement, applying a sequence of highly regular permutation layers that act uniformly across the memory register. This organization exposes a tradeoff between locality and parallelism that is not reflected by standard circuit complexity measures.

Whether this tradeoff is advantageous ultimately depends on the underlying hardware architecture and communication model. Our complexity analysis is carried out in the standard circuit model, where SWAP and controlled-SWAP gates are treated as elementary operations. Additional costs arising from restricted qubit connectivity, architecture-specific communication constraints, or fault-tolerant implementations are outside the scope of the present work. Conversely, the high regularity of the proposed construction may simplify compilation, scheduling, and optimization in architectures that efficiently support uniform parallel operations. A quantitative comparison under realistic hardware assumptions remains an interesting direction for future work.

On the theoretical side, the logarithmic overhead originates from the binary decomposition of the address register, suggesting the investigation of alternative encodings or computational models that could further reduce the complexity of controlled global permutations. More generally, viewing quantum memory access as the composition of global data movement and local operations provides a framework for exploring intermediate organizations between fully global permutation-based constructions and localized routing-based architectures.

Overall, our results broaden the design space of QRAM implementations by showing that coherent memory access in the standard quantum circuit model does not inherently require routing. Rather than replacing existing routing-based architectures, permutation-based constructions complement them by demonstrating that the same computational functionality admits fundamentally different organizations with comparable asymptotic complexity. We hope that this perspective will stimulate further investigation into alternative computational models and architectures for quantum memory access.

Declaration on generative AI

Generative AI was used exclusively to assist with the linguistic refinement of the manuscript, including the rephrasing of selected sentences and paragraphs to improve clarity, conciseness, and readability.

References

[1] L. K. Grover, T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions (2002). arXiv:quant-ph/0208112.

[2] I. Kerenidis, A. Prakash, Quantum recommendation systems, in: ITCS, 2017.

[3] S. Aaronson, Read the fine print, Nature Physics 11 (2015) 291–293. doi:10.1038/nphys3272.

[4] V. Giovannetti, S. Lloyd, L. Maccone, Quantum random access memory, Physical Review Letters 100 (2008) 160501. doi:10.1103/PhysRevLett.100.160501.

[5] V. Giovannetti, S. Lloyd, L. Maccone, Architectures for a quantum random access memory, Physical Review A 78 (2008) 052310. doi:10.1103/PhysRevA.78.052310.

[6] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, P. V. Srinivasan, On the robustness of bucket brigade quantum ram (2015). arXiv:1502.03450.

[7] C. T. Hann, G. Lee, S. M. Girvin, L. Jiang, Resilience of quantum random access memory to generic noise, PRX Quantum 2 (2021) 020311. doi:10.1103/PRXQuantum.2.020311.

[8] R. Mehta, G. Lee, L. Jiang, Analysis and suppression of errors in quantum random access memory under extended noise models (2024). arXiv:2412.10318.

[9] O. D. Matteo, V. Gheorghiu, M. Mosca, Fault-tolerant resource estimation of quantum random access memories, IEEE Transactions on Quantum Engineering 1 (2020) 1–13. doi:10.1109/TQE.2020.2965993. arXiv:1902.01329.

[10] S. Jaques, A. Rattew, Quantum random access memory: A survey and critique, Quantum 9 (2025) 1922. doi:10.22331/q-2025-12-02-1922.

[11] X. Xu, B. Peng, S. Sinha, et al., Systems architecture for quantum random access memory, in: Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2023, pp. 1–15.

[12] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019), ACM, 2019, pp. 217–228. doi:10.1145/3313276.3316310.

[13] D. K. Park, F. Petruccione, J.-K. K. Rhee, Circuit-based quantum random access memory for classical data, Scientific Reports 9 (2019) 3949. doi:10.1038/s41598-019-40439-3.

[14] P. A. Moreno Casares, Circuit implementation of bucket brigade qram for quantum state preparation, arXiv preprint arXiv:2006.11761 (2020).

[15] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. M. Girvin, L. Jiang, Hardware-efficient quantum random access memory with hybrid quantum acoustic systems, Physical Review Letters 123 (2019) 250501. doi:10.1103/PhysRevLett.123.250501.

[16] D. K. Weiss, S. Puri, S. M. Girvin, Quantum random access memory architectures using 3d

superconducting cavities, PRX Quantum 5 (2024) 020312. doi:10.1103/PRXQuantum.5.020312.

[17] Z. Wang, H. Qiao, A. N. Cleland, L. Jiang, Quantum random access memory with transmon-controlled phonon routing, Physical Review Letters 134 (2025) 210601. doi:10.1103/PhysRevLett.134.210601.

[18] K. Phalak, X.-C. Li, S. Ghosh, Approximate quantum random access memory architectures (2022). arXiv:2210.14804.

[19] Z.-Y. Chen, C. Xue, Y.-J. Wang, T.-P. Sun, H.-Y. Liu, X.-N. Zhuang, M.-H. Dou, T.-R. Zou, Y. Fang, Y.-C. Wu, G.-P. Guo, Efficient and error-resilient data access protocols for a limited-sized quantum random access memory, arXiv preprint arXiv:2303.05207 (2023). doi:10.48550/arXiv.2303.05207.

[20] R. Asaka, K. Sakai, R. Yahagi, Two-level quantum walkers on directed graphs ii: An application to qram, arXiv preprint arXiv:2204.08709 (2022). doi:10.48550/arXiv.2204.08709.

[21] P. Mukhopadhyay, A new architecture for quantum random access memory (2024). arXiv:2408.16794.

[22] D. Bohac, Universal quantum random access memory (2025). arXiv:2512.12999.

[23] A. Pavone, C. Viola, The quantum cyclic rotation gate, SN Comput. Sci. 5 (2024) 830. URL: https://doi.org/10.1007/s42979-024-03141-4. doi:10.1007/S42979-024-03141-4.

[24] S. Faro, A. Pavone, C. Viola, Families of constant-depth quantum circuits for rotations and permutations, in: U. de'Liguoro, M. Palazzo, L. Roversi (Eds.), Proceedings of the 25th Italian Conference on Theoretical Computer Science, Torino, Italy, September 11-13, 2024, CEUR Workshop Proceedings, CEUR-WS.org, 2024, pp. 29–41. URL: https://ceur-ws.org/Vol-3811/paper270.pdf.