

Stab-QRAM: An All-Clifford Quantum Random Access Memory for Special Data

Guangyi Li,^{1,2} Yu Gan,¹ Zeguan Wu,¹ Xueyue Zhang,³ Zheshen Zhang,² and Junyu Liu¹

¹*Department of Computer Science, The University of Pittsburgh, Pittsburgh, PA 15260, USA*

²*Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA*

³*Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, USA*

Quantum random access memories (QRAMs) are pivotal for data-intensive quantum algorithms, but existing general-purpose and domain-specific architectures are hampered by a critical bottleneck: a heavy reliance on non-Clifford gates (e.g., T-gates), which are prohibitively expensive to implement fault-tolerantly. To address this challenge, we introduce the Stabilizer-QRAM (Stab-QRAM), a domain-specific architecture tailored for data with an affine Boolean structure ($f(\mathbf{x}) = \mathbf{Ax} + \mathbf{b}$ over \mathbb{F}_2), a class of functions vital for optimization, time-series analysis, and quantum linear systems algorithms. We demonstrate that the gate interactions required to implement the matrix A form a bipartite graph. By applying König’s edge-coloring theorem to this graph, we prove that Stab-QRAM achieves an optimal logical circuit depth of $O(\log N)$ for N data items, matching its $O(\log N)$ space complexity. Critically, the Stab-QRAM is constructed exclusively from Clifford gates (CNOT and X), resulting in a zero T -count. This design completely circumvents the non-Clifford bottleneck, eliminating the need for costly magic state distillation and making it exceptionally suited for early fault-tolerant quantum computing platforms. We highlight Stab-QRAM’s utility as a resource-efficient oracle for applications in discrete dynamical systems, and as a core component in Quantum Linear Systems Algorithms, providing a practical pathway for executing data-intensive tasks on emerging quantum hardware.

I. INTRODUCTION

The field of quantum computing has transitioned from a theoretical framework to an experimental reality, driven by advancements in physical platforms such as superconducting [1] and trapped-ion [2] systems. These developments have enabled increasingly capable quantum processors. However, a key challenge in realizing the potential of transformative quantum algorithms is the efficient encoding of large classical data into quantum states, as this must be done in a way that preserves the computational advantages of quantum algorithms for big data applications. Analogous to Random Access Memory (RAM) in classical computing [3], Quantum Random Access Memory (QRAM) [4–6] addresses this challenge by performing coherent memory lookups. Specifically, QRAM transforms a superposition of address states $\sum_i \alpha_i |i\rangle |0\rangle^{\otimes m}$ into a superposition of data states $\sum_i \alpha_i |i\rangle |f(i)\rangle$ via the unitary operation U_f . This capability is critical for quantum algorithmic speedups, as it enables simultaneous querying of multiple memory locations in a single coherent operation, providing exponential parallelism in data access unattainable by classical RAM.

Current QRAM architectures can be broadly categorized into two types [7]: *general-purpose (GP)* and *domain-specific (DS)*. A GP architecture is designed for universality, aiming to load arbitrary and unstructured data with perfect fidelity. Conversely, a DS architecture sacrifices this universal capability for highly efficient—in terms of resource cost and/or execution time—loading or representation of data that conforms to a specific structure, distribution, or algorithmic domain. GP architectures [5–12] offer this universality but face significant resource trade-offs; prominent designs either require a prohibitive number of qubits proportional to the memory size, $O(N)$, or suffer from a circuit depth that scales at least linearly with memory size. In line with their design principle, existing DS architectures [13, 14], often based on machine learning models, exemplify this trade-off by typically

providing an *approximate* data loading, which may require computationally intensive training and lack adaptability. Critically, nearly all existing GP and DS architectures rely heavily on non-Clifford gates (e.g., T-gates or Toffoli gates), which exhibit substantially higher error rates and are exceptionally costly to implement fault-tolerantly [15, 16]. This reliance on fragile, resource-intensive gates severely limits the scale of data-intensive algorithms that can be executed.

To address this gap, we introduce the *Stab-QRAM*, a novel DS architecture designed for near- and mid-term quantum devices. Our approach makes a strategic trade-off: we sacrifice universality in favor of exceptional resource efficiency by focusing on data with an *affine Boolean structure*: functions of the form $f(\mathbf{x}) = \mathbf{Ax} + \mathbf{b}$ over the finite field \mathbb{F}_2 . While the implementability of such functions with Clifford circuits is a known consequence of the Gottesman-Knill theorem [17], our work moves beyond this principle to establish a complete and practical quantum memory framework. We formalize this concept into a complete, reconfigurable QRAM architecture, and provide the first rigorous analysis of its optimal performance. The resulting architecture circumvents the non-Clifford gate bottleneck entirely, achieving $O(\log N)$ space and depth complexity, which makes it uniquely suited for data-intensive tasks.

This paper presents a comprehensive blueprint for the Stab-QRAM. In Section II, we present the theoretical foundations of our model and characterize the Stab-QRAM’s properties. In Section III, we provide a comparative analysis with existing architectures. Finally, in Section IV, we explore promising applications and discuss the potential for efficient implementation of the Stab-QRAM on contemporary quantum hardware platforms.

II. RESULT

Theoretical Model and Construction — We first prove that a universal QRAM, capable of implementing an oracle for an arbitrary Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, cannot be constructed using only Clifford gates. The proof rests on the constrained nature of the Clifford group [18].

A QRAM oracle must perform the transformation $U_f : |\mathbf{x}\rangle|0\rangle^{\otimes m} \mapsto |\mathbf{x}\rangle|f(\mathbf{x})\rangle$. Let us assume this transformation is implemented by a Clifford circuit C . The action of any Clifford circuit on a computational basis state is restricted to the form $C|\mathbf{x}\rangle|0\rangle^{\otimes k} = |g(\mathbf{x})\rangle|\phi(\mathbf{x})\rangle$, where both g and ϕ must be affine functions over the finite field \mathbb{F}_2 [19]. For the circuit C to function as the QRAM oracle U_f , its output must match the required form, which implies we must have $g(\mathbf{x}) = \mathbf{x}$ and $\phi(\mathbf{x}) = f(\mathbf{x})$. This directly forces the function $f(\mathbf{x})$ to be affine.

This constraint makes a pure Clifford-based QRAM for universal data access impossible. The class of affine functions is exponentially smaller than the set of all Boolean functions. For a single output bit ($m = 1$), the number of affine functions is $|\mathcal{A}_n| = 2^{n+1}$, whereas the total number of Boolean functions is $|\mathcal{B}_n| = 2^{2^n}$. Since for any $n \geq 2$ most Boolean functions are non-affine, they cannot be realized by a Clifford circuit. Therefore, a universal QRAM cannot be built from Clifford gates alone.

This fundamental limitation, however, precisely defines the domain where a highly efficient, specialized QRAM can be built. We show that for any function belonging to this affine class,

$$f(\mathbf{x}) = A\mathbf{x} + b, \quad \text{where } A \in \mathbb{F}_2^{m \times n} \text{ and } b \in \mathbb{F}_2^m,$$

an exact, all-Clifford oracle can be constructed. For an n -qubit address register $|\mathbf{x}\rangle$ and an m -qubit data register $|0\rangle^{\otimes m}$ (where often $m = n$), the linear transformation $A\mathbf{x}$ is implemented by applying a CNOT gate from address qubit x_k to data qubit d_j if and only if the matrix entry $A_{j,k} = 1$. The constant offset b is then added by applying an X gate to data qubit d_j if the vector entry $b_j = 1$. The complete unitary operator takes the compact form:

$$C_f = \prod_{j=1}^m \left[\left(\prod_{k: A_{j,k}=1} \text{CNOT}(x_k, d_j) \right) X^{b_j}_{d_j} \right].$$

This construction is reconfigurable and requires only $n + m$ qubits, yielding a space complexity of $O(\log N)$ where $N = 2^n$ denotes the number of memory locations.

Geometric Architecture — The implementation of the affine function $f(x) = Ax + b$ in the Stab-QRAM architecture is modeled by a logical interaction graph $G_A = (V, E)$. The vertices V represent the system's qubits, which are partitioned into two disjoint sets: an n -qubit address register $V_{\text{addr}} = \{x_1, \dots, x_n\}$, and an m -qubit data register $V_{\text{data}} = \{d_1, \dots, d_m\}$.

The graph's edges E represent the required CNOT interactions between these registers. An edge (x_k, d_j) exists if and only if the matrix entry $A_{j,k} = 1$, corresponding to a

CNOT(x_k, d_j) operation. The complete edge set is thus formally defined as $E = \{(x_k, d_j) \in V_{\text{addr}} \times V_{\text{data}} \mid A_{j,k} = 1\}$.

By this construction, every edge connects an address qubit to a data qubit, making G_A an inherently bipartite graph. This structural property is fundamental to the architecture's efficient hardware implementation [20] and it lays the foundation for achieving low circuit depth which will be proved later. For example, consider the case where $n = m = 5$ with the matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

The corresponding bipartite graph is visualized in Figure 1.

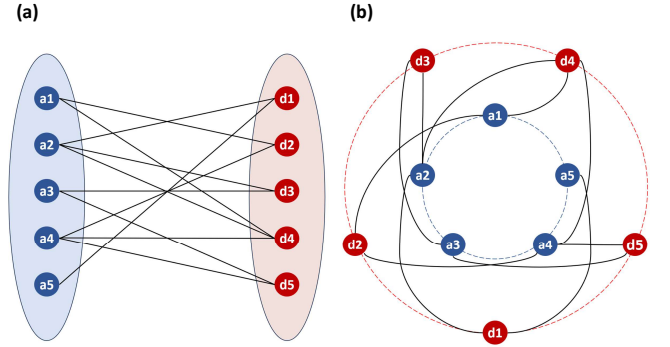


FIG. 1. **Two visualizations of the Stab-QRAM's bipartite graph architecture for a specific 5×5 matrix A .** In both layouts, blue circles represent the 5 address qubits (V_{addr}) and red circles represent the 5 data qubits (V_{data}). An edge between an address and a data qubit indicates a required CNOT gate. (a) A standard linear layout. (b) A circular layout to visualize connectivity.

Gate Count and Locality — Gate count serves as a key metric for evaluating circuit resource requirements. Our Stab-QRAM circuit contains no non-Clifford gates, with CNOT gates being the primary operations. The total CNOT count equals the Hamming weight of matrix A (number of non-zero entries). As illustrated in Figure 2, this count scales linearly with matrix density p (probability of $A_{j,k} = 1$) and quadratically with dimension n (for $m = n$), matching the expected $E[\text{count}] = m \cdot n \cdot p$. This corresponds to an $O((\log N)^2)$ scaling, where $N = 2^n$ represents the data space size, enabling slow growth and strong scalability for large-scale implementations.

Locality is crucial for hardware realization, as the logical bipartite graph G_A must map onto physical hardware with limited connectivity. To analyze this, we model the hardware as a k -regular graph [21], where each vertex (representing a physical qubit) has exactly k edges connecting it to other vertices. This structure is chosen for its generality, as it abstracts

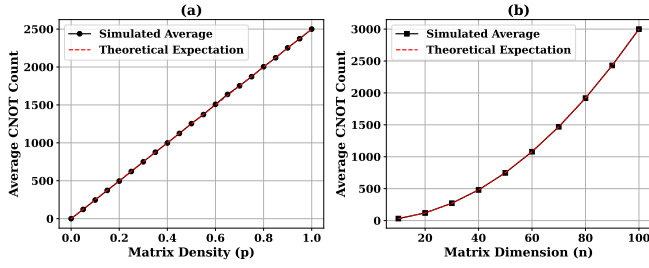


FIG. 2. **CNOT gate count analysis.** (a) Average gate count scales linearly with density p of matrix A ($m = n = 50$). (b) For fixed density, count scales quadratically with matrix dimension n ($m = n$). Simulated data (black) aligns with theoretical expectation (red dashes).

away from specific proprietary architectures while capturing the fundamental connectivity constraints prevalent across various quantum platforms [22, 23]. For concreteness, consider the connectivity in leading platforms: IBM’s 2025 processors, such as the Nighthawk [24], adopt a square-lattice topology with a 4-degree nearest-neighbor connectivity ($k = 4$), enabling denser two-qubit gate implementations in a 2D grid-like arrangement. In contrast, Google’s Willow chip [25] features a more irregular tunable coupler architecture with an average connectivity of 3.47, optimized for error-corrected logical qubits but with boundary effects reducing the effective degree for edge qubits. Moreover, by varying the parameter k , it facilitates the exploration of different connectivity levels, enabling predictions about performance on future hardware with potentially enhanced connectivity without being limited to current designs.

Using a greedy heuristic for mapping—sorting logical qubits by degree and placing them to maximize connections—we assess non-local CNOTs, which require SWAP operations. We quantify this via the maximum required distance d on the hardware graph, necessitating $(d - 1)$ SWAPs per such CNOT, indicating potential bottlenecks.

Heatmaps in Figure 3 highlight Stab-QRAM’s advantages: increasing hardware connectivity k markedly reduces the maximum $(d - 1)$ -locality. Sparse matrices (low p) and smaller dimensions (n) yield significant optimizations, minimizing SWAP overhead and enhancing efficiency on near-term devices.

Circuit Depth — The logical circuit depth, defined as the minimum number of parallel time steps required to execute all gates, is a critical performance metric for quantum algorithms. Our objective is to determine an optimal scheduling strategy for the CNOT and X gates in the Stab-QRAM circuit to minimize this depth. We demonstrate that this scheduling problem can be solved exactly by mapping it to a classic problem in graph theory.

The scheduling problem is elegantly modeled using the bipartite interaction graph $G_A = (V, E)$, where vertices represent qubits and edges represent the required CNOT operations. The fundamental physical constraint is that a single qubit can only participate in one CNOT gate per time step. In the graph-theoretic framework, this constraint translates di-

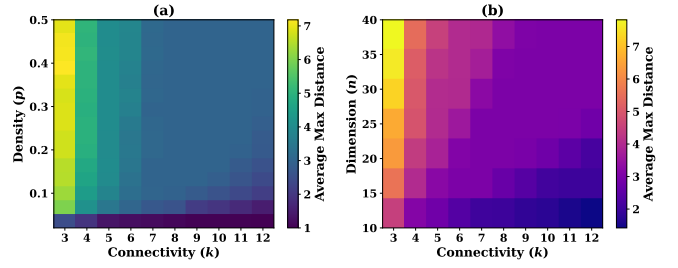


FIG. 3. **Analysis of maximum CNOT locality after mapping.** Color indicates average maximum shortest-path distance d , with darker shades showing better locality (fewer SWAPs). (a) For fixed size ($n = m = 25$), distance rises with density p but falls with connectivity k . (b) For fixed density ($p = 0.25$), distance grows with n , mitigated by higher k .

rectly to an edge-coloring problem: any two edges (CNOTs) that are incident on the same vertex (qubit) must be assigned to different time layers (colors). Consequently, the minimum circuit depth for the CNOT portion of the circuit is precisely the chromatic index of the graph G_A —the minimum number of colors needed for a valid edge-coloring.

To determine this value, we consider the maximum degree of the graph, $\Delta(G_A)$, which is the maximum number of CNOT operations in which any single qubit participates. The maximum degree is formally given by:

$$\Delta = \max \left\{ \max_k \sum_j A_{j,k}, \max_j \sum_k A_{j,k} \right\}$$

A cornerstone result in graph theory, König’s edge-coloring theorem [26], states that the chromatic index of any bipartite graph is exactly equal to its maximum degree. Since the interaction graph G_A is inherently bipartite, this theorem provides a direct construction for the optimal circuit schedule: all CNOT gates corresponding to edges of the same color are assigned to a single parallel execution layer. This allows all CNOT operations to be scheduled in precisely Δ layers.

The remaining single-qubit X gates, which implement the constant offset b , can all be executed in a single additional parallel layer after the CNOT operations are complete. Therefore, the total logical depth of the Stab-QRAM circuit is given by:

$$D_{\text{logic}} = \Delta + 1$$

In the worst-case scenario, the maximum degree Δ is bounded by $\max(n, m)$. Since the number of memory locations is $N = 2^n$, the circuit depth scales as $O(n) = O(\log N)$. This analysis rigorously establishes the optimal logarithmic depth complexity of the Stab-QRAM architecture, a key advantage highlighted in Table I.

Numerical simulations in Figure 4 underscore Stab-QRAM’s application advantages. Sparse matrices yield minimal depths, suiting sparse-data tasks like quantum machine learning or graph algorithms with reduced latency. Dense matrices, while deeper, maintain robustness below bounds,

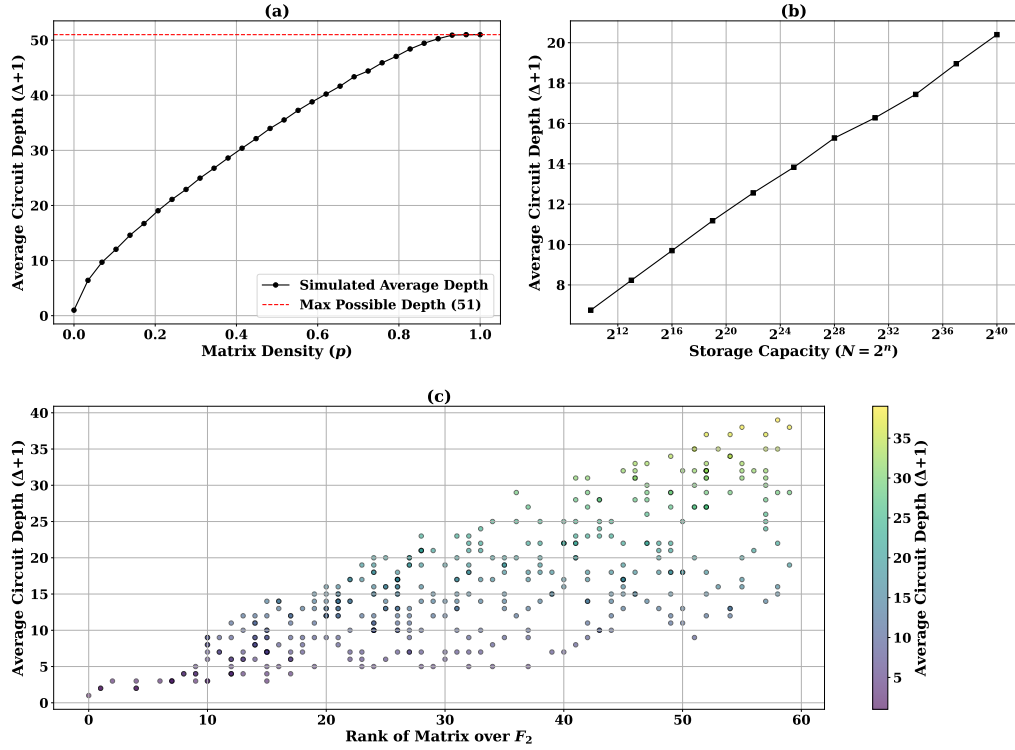


FIG. 4. **Analysis of logical circuit depth.** (a) For a fixed size ($n = m = 50$), the average depth increases with matrix density p , remaining well below the theoretical maximum depth of $n + 1$ (red dashed line). (b) The average depth scales linearly against the storage capacity $N = 2^n$ on a logarithmic axis, visually confirming the $O(n) = O(\log N)$ complexity. (c) A scatter plot for randomly generated matrices reveals no direct correlation between the matrix rank over \mathbb{F}_2 and the circuit depth.

supporting reliable performance in dense-data scenarios. The $O(\log N)$ scaling enables efficient large-scale quantum memory, outperforming alternatives in expansive computations. The analysis of matrix rank versus depth in Figure 4(c) yields a key insight: there is no direct correlation. This is because rank is a global algebraic property, whereas the circuit depth is dictated by the local graph property of maximum qubit degree, Δ . This distinction is critical for algorithm design, as it shows the Stab-QRAM’s performance hinges on the local distribution of CNOTs rather than the matrix’s global structure. Consequently, sparse matrices, regardless of their rank, are ideal for leveraging the architecture’s inherent parallelism.

III. COMPARISON

The Stab-QRAM occupies a unique and highly advantageous position within the landscape of QRAM designs. As a domain-specific architecture, it leverages the structure of affine Boolean functions to achieve exceptional efficiency, setting it apart from both general-purpose and other domain-specific models.

GP architectures offer universality at the cost of significant resource overhead. The seminal Bucket-Brigade QRAM (BB-QRAM) [5] is notable for its logarithmic $O(\log N)$ query time, but its $O(N)$ space complexity makes it prohibitive for large datasets. We select BB-QRAM as the representative

router-based QRAM, as subsequent developments based on BB-QRAM do not offer significant improvements in asymptotic complexity for key metrics and continue to rely heavily on non-Clifford gates. In contrast, Quantum Read-Only Memory (QROM) [10] achieves an optimal $O(\log N)$ space complexity but suffers from a $O(N \log N)$ circuit depth. Crucially, both architectures rely on a high number of non-Clifford gates, leading to at least an $O(N)$ T-gate count, which is a primary bottleneck for fault-tolerant implementation.

Other DS architectures sacrifice universality for efficiency, a philosophy our work shares, but they do so with different trade-offs. Prominent examples like Parametric Quantum Circuit based QRAM (PQC-based QRAM) [13] uses variational or machine learning techniques to “learn” a data-loading function. Though the PQC-based QRAM work claims the ability to access arbitrary data, its design differs significantly from a general-purpose QRAM, and its ability to extend to arbitrary datasets is relatively weak. So far, its functionality has been demonstrated primarily in quantum machine learning and binary storage tasks, which is why we regard PQC-based QRAM as a DS QRAM in this work. While PQC-based QRAM can achieve impressive $O(1)$ logical circuit depth for specific tasks, this paradigm introduces distinct challenges. The data loading is often approximate, the training process itself can be computationally intensive, and the underlying circuits still require non-Clifford gates, resulting in a non-zero T-gate complexity.

QRAM design	QRAM Types	Space complexity	Logical Circuit depth	T gate count
Stab-QRAM (this work)	Domain-specific (DS)	$O(\log N)$	$O(\log N)$	0
PQC-based QRAM	Domain-specific (DS)	$O(\log N)$	$O(1)$	$O((\log N)^2)$
Bucket-Brigade QRAM	General-purpose (GP)	$O(N)$	$O(\log N)$	$O(N)$
QROM	General-purpose (GP)	$O(\log N)$	$O(N \log N)$	$O(N(\log N)^2)$

TABLE I. Comparison of various QRAM architectures based on their type, space complexity, circuit depth, and T gate count. $N = 2^n$ is the total number of memory locations, where n is the number of address qubits.

The Stab-QRAM’s advantage lies in its unique combination of strengths. Unlike trainable models, its construction is deterministic, providing exact data loading for its specified domain. Most importantly, by restricting its operation to a class of functions implementable with linear algebra over \mathbb{F}_2 , our architecture is constructed exclusively using Clifford gates. This design choice completely eliminates the non-Clifford gate bottleneck, resulting in a zero T-gate count. As summarized in Table I, the Stab-QRAM combines $O(\log N)$ space and depth complexity with unparalleled T-gate efficiency, making it a uniquely powerful and practical solution for a significant class of quantum algorithms.

IV. DISCUSSION

In this work, we introduced the Stab-QRAM, a domain-specific architecture designed to overcome a critical bottleneck in quantum computing by enabling highly efficient loading of structured classical data. By deliberately trading the universality of general-purpose designs for exceptional resource efficiency on affine Boolean problems ($f(x) = Ax + b$), Stab-QRAM attains $O(\log N)$ space and depth complexity with zero T-gates. Its Clifford-only construction is inherently compatible with fault-tolerant quantum computing (FTQC) [27], completely eliminating the costly magic state distillation [28] required by non-Clifford QRAMs. This not only slashes the qubit and runtime overheads in FTQC implementations but also lowers the practical threshold for achieving quantum advantage, positioning Stab-QRAM as a powerful enabler for bringing its target applications into the near-future fault-tolerant era.

One of the most promising applications of Stab-QRAM lies in time-series analysis [29], where data can often be modeled as discrete affine dynamical systems in binary state spaces. For instance, linear feedback shift registers [30] (LFSRs) underpins many real-world systems, including digital communications [31], gene regulatory networks [32], and simplified financial models. In this context, Stab-QRAM serves as a quantum co-processor for simulating system evolutions $\mathbf{x}_{t+1} = A\mathbf{x}_t + b$, enabling parallel exploration of superimposed trajectories. This capability directly supports probabilistic forecasting by sampling from evolved quantum states, as well as structured search tasks via integration with Grover’s algorithm [33] for pattern matching or cryptanalysis of LFSR-

based stream ciphers.

Beyond time-series, Stab-QRAM holds potential for optimization problems involving affine constraints, such as binary linear programming or solving systems of linear equations [34] over \mathbb{F}_2 . Its reconfigurable nature—where the circuit is dynamically defined by the matrix A and vector b —allows for flexible adaptation to different problem instances without hardware redesign, making it suitable for hybrid quantum-classical workflows in quantum machine learning. This capability is particularly consequential for Quantum Linear Systems Algorithms (QLSAs), including variants of the HHL algorithm [35]. Stab-QRAM provides a direct, resource-efficient construction of the primary oracle encoding the problem data, which is often a bottleneck in practical QLSA implementations. As QLSAs are a core subroutine in more advanced frameworks like Quantum Interior Point Methods (QIPMs) [36] and Mod2VQLS [37], the profound efficiency gains offered by Stab-QRAM propagate enables a viable path toward applying these powerful algorithms to challenging optimization tasks in fields such as logistics, power flow, and network design.

From a hardware perspective, Stab-QRAM’s practicality stems from its specialized design. Compared to a general-purpose QRAM, it relies solely on CNOT gates with specific connectivity requirements, a feature that enables its implementation on widely used quantum platforms, including trapped ions [2], Rydberg atom arrays [38], photonics [39, 40], and superconducting qubits coupled with microwave or mechanical modes [40–46]. For superconducting circuits in particular, Stab-QRAM circumvents the need for complex controlled-iSWAP gates, which demand native support on specially engineered hardware [41] or quantum routers [42–44]. Instead, a general-purpose processor equipped with optimized CNOT gates can readily support it. Moreover, Stab-QRAM stands to benefit from recent advances in superconducting architectures that enable enhanced connectivity, either through a shared bus [47, 48] or across separate modules [49–51]. In photonic measurement-based quantum computing (MBQC) platforms [52, 53], Stab-QRAM can be implemented by encoding time-bin or dual-rail address and data qubits in cluster states. The oracle’s all-Clifford layers are executed as parallel measurements to realize the CNOT and X gates based on fixed bases with Pauli-frame tracking—no adaptive angles are needed. The photonic implementation only requires short optical delays based on fiber

loops [54, 55] or on-chip low-loss waveguides [56] without resorting to no long-lived quantum storage, while photonic switching and multiplexing supplies the in-round parallelism. This hardware-friendly, T-gate-free construction is not only suitable for demonstration on current Noisy intermediate-scale quantum computing (NISQ) [57] devices but also ensures that its performance will scale with hardware maturity.

Despite its applications and hardware ease, Stab-QRAM’s domain-specific nature limits it to affine functions, potentially hindering broader adoption. However, it forms an efficient base for structured QRAMs. Future work could add minimal non-Clifford gates to introduce non-linearity, enabling non-affine functions like Quadratic unconstrained binary optimization (QUBO) [58], non-linear machine learning features [59], or piecewise affine cryptography and simulations. The T-count would scale with non-linearity, not data size N , offering expansion potential with hardware advances. This bridges specialized efficiency and general-purpose costs, enabling more near-term algorithms.

In conclusion, Stab-QRAM exemplifies a design phi-

losophy rarely seen in the QRAM literature: a focused, data-structure-specific quantum memory engineered for maximal efficiency, rather than universal applicability. This targeted approach demonstrates how specialized quantum accelerators can unlock significant real-world performance gains long before fully universal, fault-tolerant quantum computers become available.

Acknowledgment — We thank Frederic T. Chong, Yongshan Ding, Steve Girvin, Allen Mi, Kate Smith and Youtao Zhang for helpful inputs. GL, YG, ZW, and JL are supported in part by the University of Pittsburgh, School of Computing and Information, Department of Computer Science, Pitt Cyber, PQI Community Collaboration Awards, John C. Mascaró Faculty Scholar in Sustainability, NASA under award number 80NSSC25M7057, and Fluor Marine Propulsion LLC (U.S. Naval Nuclear Laboratory) under award number 140449-R08. ZZ acknowledges NSF CAREER Award No. 2317471. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

-
- [1] A. Megrant and Y. Chen, “Scaling up superconducting quantum computers,” *Nature Electronics* **8**(6), 549–551 (2025).
 - [2] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, “Trapped-ion quantum computing: Progress and challenges,” *Applied Physics Reviews* **6**(2), 021314 (2019).
 - [3] R. C. Jaeger, T. N. Blalock, and B. J. Blalock, *Microelectronic circuit design* (McGraw-Hill, New York, NY, 1997).
 - [4] S. Jaques, A. G. Rattew, “QRAM: A Survey and Critique,” arXiv:2305.10310 [quant-ph] (2023).
 - [5] V. Giovannetti, S. Lloyd, and L. Maccone, “Quantum random access memory,” *Physical Review Letters* **100**(16), 160501 (2008).
 - [6] V. Giovannetti, S. Lloyd, L. Maccone, “Architectures for a quantum random access memory,” *Phys. Rev. A* **78**(5), 052310 (2008).
 - [7] S. Xu, C. T. Hann, B. Foxman, S. M. Girvin, and Y. Ding, “Systems architecture for quantum random access memory,” arXiv:2306.03242 [quant-ph] (2023).
 - [8] S. Xu, A. Lu, and Y. Ding, “Fat-Tree QRAM: A high-bandwidth shared quantum random access memory for parallel queries,” in *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS ’25)* (Association for Computing Machinery, New York, NY, USA), 390–406 (2025).
 - [9] D. K. Park, F. Petruccione, and J. K. K. Rhee, “Circuit-based quantum random access memory for classical data,” *Scientific Reports* **9**(1), 3949 (2019).
 - [10] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, “Encoding electronic spectra in quantum circuits with linear T complexity,” *Physical Review X* **8**(4), 041015 (2018).
 - [11] M. A. Nielsen, I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2002).
 - [12] G. H. Low, V. Kliuchnikov, L. Schaeffer, “Trading T gates for dirty qubits in state preparation and unitary synthesis,” *Quantum* **8**, 1375 (2024).
 - [13] K. Phalak, J. Li, and S. Ghosh, “Approximate quantum random access memory architectures,” arXiv:2210.14804 [quant-ph] (2022).
 - [14] M. Y. Niu, A. Zlokapa, M. Broughton, S. Boixo, M. Mohseni, V. Smelyanskiy, and H. Neven, “Entangling quantum generative adversarial networks,” arXiv:2105.00080 [quant-ph] (2021).
 - [15] D. Honciuc Menendez, A. Ray, and M. Vasmer, “Implementing fault-tolerant non-Clifford gates using the $[[8,3,2]]$ color code,” *Physical Review A* **109**(6), 062438 (2024).
 - [16] A. Singal, K. N. Smith, “Heterogeneously error-corrected QRAMs,” arXiv:2504.21687 [quant-ph] (2025).
 - [17] D. Gottesman, “The Heisenberg Representation of Quantum Computers,” arXiv:quant-ph/9807006 (1998).
 - [18] J. Tolar, “On Clifford groups in quantum computing,” *Journal of Physics: Conference Series* **1071**(1), 012022 (2018).
 - [19] S. Aaronson and D. Gottesman, “Improved simulation of stabilizer circuits,” *Physical Review A* **70**(5), 052328 (2004).
 - [20] A. B. Khesin, “Quantum computing from graphs,” arXiv:2501.17959 [quant-ph] (2025).
 - [21] G. D. Scholes, “Large coherent states formed from disordered k-regular random graphs,” *Entropy* **25**(11), 1519 (2023).
 - [22] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, “Quantum computing with Qiskit,” arXiv:2405.08810 [quant-ph] (2024).
 - [23] Google Quantum AI and Collaborators, “Quantum error correction below the surface code threshold,” *Nature* **638**(8044), 920–926 (2025).
 - [24] IBM Quantum Computing Blog, “How IBM will build the world’s first large-scale, fault-tolerant quantum computer,” IBM (2025).
 - [25] K. D., “Google Willow Chip, A Closer Look At The Tech Giant’s Push into Quantum Computing,” *Quantum Zeitgeist* (2025).
 - [26] C. Mummert, “The logical strength of König’s edge coloring theorem,” arXiv:2008.12694 [math.LO] (2020).
 - [27] A. Paler and S. J. Devitt, “An introduction to fault-tolerant

- quantum computing,” in *Proceedings of the 52nd Annual Design Automation Conference (DAC '15)* (Association for Computing Machinery, New York, NY, USA), Article No. 60 (2015).
- [28] P. S. Rodriguez, J. M. Robinson, P. N. Jepsen, Z. He, C. Duckering, C. Zhao, K.-H. Wu, J. Campo, K. Bagnall, M. Kwon, T. Karolyshyn, P. Weinberg, M. Cain, S. J. Evered, A. A. Geim, M. Kalinowski, S. H. Li, T. Manovitz, J. Amato-Grill, J. I. Basham, *et al.*, ”Experimental demonstration of logical magic state distillation,” arXiv:2412.15165 [quant-ph] (2024).
 - [29] A. Daskin, ”A walk through of time series analysis on quantum computers,” arXiv:2205.00986 [quant-ph] (2022).
 - [30] H.-I. Kim and J.-C. Jeon, ”Quantum LFSR structure for random number generation using QCA multilayered shift register for cryptographic purposes,” *Sensors* **22**(9), 3541 (2022).
 - [31] X. Zhang, ”A low-power parallel architecture for linear feedback shift registers,” *IEEE Transactions on Circuits and Systems II: Express Briefs* **66**(3), 412–416 (2019).
 - [32] M. Mircea, D. Garlaschelli, and S. Semrau, ”Inference of dynamical gene regulatory networks from single-cell data with physics informed neural networks,” arXiv:2401.07379 [q-bio.MN] (2024).
 - [33] L. K. Grover, ”A fast quantum mechanical algorithm for database search,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 212–219 (1996).
 - [34] M. R. Perelshtein, A. I. Pakhomchik, A. A. Melnikov, A. A. Novikov, A. Glatz, G. S. Paraoanu, V. M. Vinokur, and G. B. Lesovik, ”Large-scale quantum hybrid solution for linear systems of equations,” *Annalen der Physik* **534**(6), 2200082 (2022).
 - [35] A. W. Harrow, A. Hassidim, and S. Lloyd, ”Quantum algorithm for linear systems of equations,” *Physical Review Letters* **103**(15), 150502 (2009).
 - [36] M. Mohammadhossein, Z. Wu, B. Augustino, A. Carr, T. Terlaky, ”Improvements to Quantum Interior Point Method for Linear Optimization,” *ACM Transactions on Quantum Computing* **6**(1), 8 (2025).
 - [37] W. Aboumrada and D. Widdows, ”Mod2VQLS: A variational quantum algorithm for solving systems of linear equations modulo 2,” arXiv:2311.12771 [quant-ph] (2023).
 - [38] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, *et al.*, ”Logical quantum processor based on reconfigurable atom arrays,” *Nature* **626**(7997), 58–65 (2024).
 - [39] S. Zhang, J. Shi, Z. Cui, Y. Wang, Y. Wu, L. Duan, Y. Pu, ”Realization of a programmable multi-purpose photonic quantum memory with over-thousand qubit manipulations,” *Phys. Rev. X* **14**(2), 021018 (2024).
 - [40] Z. Wang, H. Qiao, A. N. Cleland, L. Jiang, ”Quantum random access memory with transmon-controlled phonon routing,” *Phys. Rev. Lett.* **134**(21), 210601 (2024).
 - [41] Z. Li, E. Gupta, F. Zhao, R. Banerjee, Y. Lu, T. Roy, A. Oriani, A. Vrajitoarea, S. Chakram, D. I. Schuster, ”A Cascaded Random Access Quantum Memory,” arXiv:2503.13953 [quant-ph] (2025).
 - [42] C. Miao, S. Léger, Z. Li, G. Lee, L. Jiang, and D. I. Schuster, ”Implementation of a quantum addressable router using superconducting qubits,” arXiv:2503.04295 [quant-ph] (2025).
 - [43] F. Shen, Y. Ji, D. Xiang, Y. Wang, K. Wang, C. Zhang, A. Zhang, Y. Zou, Y. Gao, Z. Cui, *et al.*, ”Experimental realization of the bucket-brigade quantum random access memory,” arXiv:2506.16682 [quant-ph] (2025).
 - [44] S. Zhang, Y.-J. Wang, P. Wang, R.-Z. Zhao, X.-Y. Yang, Z.-A. Zhao, T.-L. Wang, H.-F. Zhang, Z.-F. Li, Y. Wu, *et al.*, ”Demonstrating Coherent Quantum Routers for Bucket-Brigade Quantum Random Access Memory on a Superconducting Processor,” arXiv:2505.13958 [quant-ph] (2025).
 - [45] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang, ”Hardware-efficient quantum random access memory with hybrid quantum acoustic systems,” *Phys. Rev. Lett.* **123**, 250501 (2019).
 - [46] C. T. Hann, G. Lee, S. M. Girvin, and L. Jiang, ”Resilience of quantum random access memory to generic noise,” *PRX Quantum* **2**, 020311 (2021).
 - [47] X. Zhang, E. Kim, D. K. Mark, S. Choi, and O. Painter, ”A superconducting quantum simulator based on a photonic-bandgap metamaterial,” *Science* **379**(6629), 278–283 (2023).
 - [48] C. Song, K. Xu, W. Liu, C.-p. Yang, S.-B. Zheng, H. Deng, Q. Xie, K. Huang, Q. Guo, L. Zhang, *et al.*, ”10-qubit entanglement and parallel logic operations with a superconducting circuit,” *Physical Review Letters* **119**(18), 180511 (2017).
 - [49] X. Wu, H. Yan, G. Andersson, A. Anferov, M.-H. Chou, C. R. Conner, J. Grebel, Y. J. Joshi, S. Li, J. M. Miller, *et al.*, ”Modular quantum processor with an all-to-all reconfigurable router,” *Physical Review X* **14**(4), 041030 (2024).
 - [50] K. Heya, T. Phung, M. Malekakhlagh, R. Steiner, M. Turchetti, W. Shanks, J. Mamin, W.-S. Lu, Y. P. Kandel, N. Sundaresan, *et al.*, ”Randomized benchmarking of a high-fidelity remote CNOT gate over a meter-scale microwave interconnect,” arXiv:2502.15034 [quant-ph] (2025).
 - [51] C. Zhou, P. Lu, M. Praquin, T.-C. Chien, R. Kaufman, X. Cao, M. Xia, R. S. K. Mong, W. Pfaff, D. Pekker, *et al.*, ”Realizing all-to-all couplings among detachable quantum modules using a microwave quantum state router,” *npj Quantum Information* **9**(1), 54 (2023).
 - [52] R. Raussendorf, D. E. Browne, and H. J. Briegel, ”Measurement-based quantum computation on cluster states,” *Phys. Rev. A* **68**, 022312 (2003).
 - [53] B.-H. Wu, R. N. Alexander, S. Liu, and Z. Zhang, ”Quantum computing with multidimensional continuous-variable cluster states in a scalable photonic platform,” *Phys. Rev. Research* **2**, 023138 (2020).
 - [54] W. Asavanant, Y. Shiozawa, S. Yokoyama, B. Charoensombutamon, H. Emura, R. N. Alexander, S. Takeda, J.-I. Yoshikawa, N. C. Menicucci, H. Yonezawa, and A. Furusawa, ”Generation of time-domain-multiplexed two-dimensional cluster state,” *Science* **366**, 373–376 (2019).
 - [55] M. V. Larsen, X. Guo, C. R. Breum, J. S. Neergaard-Nielsen, U. L. Andersen, ”Deterministic generation of a two-dimensional cluster state,” *Science* **366**, 369–372 (2019).
 - [56] X. Jia *et al.*, ”Continuous-variable multipartite entanglement in an integrated microcomb,” *Nature* **639**, 329–336 (2025).
 - [57] J. Preskill, ”Quantum computing in the NISQ era and beyond,” *Quantum* **2**, 79 (2018).
 - [58] D. De Santis, S. Tirone, S. Marmi, and V. Giovannetti, ”Optimized QUBO formulation methods for quantum computing,” arXiv:2406.07681 [quant-ph] (2024).
 - [59] M. Boneberg, F. Carollo, and I. Lesanovsky, ”Non-linear classification capability of quantum neural networks due to emergent quantum metastability,” arXiv:2408.10765 [quant-ph] (2024).