Report on QRAM

Varun Ganji

May 5, 2023

Abstract

This paper is an overview of quantum memory and quantum random access memory (qRAM). QRAM is an important ongoing research topic in quantum computing in the hope of building a large physically realizable quantum computer. In this paper we give a primer on quantum memory and classical RAM. Later we discuss the prospective implementations of qRAM, challenges and present work. Especially the work on the paper qRAM by Giovannetti et al, which introduces the bucket brigade architecture of qRAM using exponentially less resources.[3].

1 Introduction

Storing information for using it later is crucial for classical computers. Modern classical computers rely heavily on both volatile and non-volatile memories for computing tasks. Random Access Memory (RAM) is a non-volatile memory that stores the current state of a computation or any data that we are actively working on. Quantum analogous to the RAM is qRAM [3]. It is realized in a similar way to RAM, where we might have input address registers and output data registers. Instead of classical bits, we use quantum bits (qubits). Many quantum computing algorithms which are being developed recently require a stored intermediate state and assume qRAM. For few of them, storing intermediate quantum states help them achieve an exponential speed up. Some examples include Q-Means by Kerenidis et al [6] and quantum active learning algorithm by Casares and Martin-Delgado [1]. This paper is a survey of qRAM and its possible physical realizations, implementations and uses.

2 Preliminaries

2.1 Quantum Memory

Quantum memory cell, which is used to build the qRAM, is a system or medium that stores and retrieves fragile quantum states. A quantum memory cell in general stores an input quantum signal and releases it after some time based on a control mechanism. Loss of information from a system into the environment is called decoherence. Due to high decoherence, memory cells are developed in well-controlled and isolated

System	τ_Q
Nuclear spin	$10^{-2} - 10^{8}$
Electron spin	10^{-3}
Ion trap (In^{+})	10^{-1}
Electron – Au	10^{-8}
Electron – GaAs	10^{-10}
Quantum dot	10^{-6}
Optical cavity	10^{-5}
Microwave cavity	10^{0}

Figure 1: Physical realizations of quantum memory [9]

environment. Figure 1 shows the possible fundamental physical representation of quantum states and their decoherence times. There have been many experiments on the best possible medium. For example, this paper on optical quantum memory [7] lists some possible photonic implementations. Single-qubit quantum memory exceeding ten-minute coherence time [12] uses trapped-ion configuration to achieve high coherence times. Another paper by Rastogi et al [10] is focused on applications of quantum networks, sheds a light on how well-controlled environments are used to achieve the accurate results.

2.2 Classical RAM

Computers store and retrieve data quickly using random access memory units. These are volatile and cheap to produce in terms of cost – an average laptop PC with 8GB of ram can bought for $400. Today we have billions of computers with billions of transistors in each one of them. RAM is constructed using a 2d array of capacitor and transistor pairs. We can store and retrieve classical data from any arbitrary cell. Figure 2 shows how the memory cells are arranged. Each cell is uniquely selected using word and bit lines. RAM interface provides input address registers and output data registers. Similar idea to decoherence is leakage of charge in memory, which is avoided by constantly refreshing memory cells.

3 QRAM

3.1 Motivation

The paper by Giovannetti et al [3] predicts the need of quantum RAM (qRAM) in large scale quantum computers, assuming that we should build them eventually. Similar to the classical RAM, we have input address registers and output data registers. Qubits are used instead of bits. We are not restricting on memory array, it can be either classical or quantum depending on the usage.

Efficient implementation of qRAM can provide exponential speedups for many algorithms. QRAMs are also essential for various algorithms like, quantum searching a

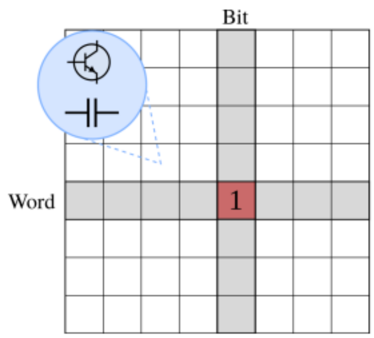

Figure 2: Classical RAM design [8]

classical database [9] and quantum cryptography algorithms.

Both quantum and classical RAMs are computationally expensive. Traditional architectures involve using $O(N^{1/d})$ switches (i.e. two-body interactions) to access one out of the $N = 2^n$ memory slots. This exponential usage of resources imply relatively slow speed and high energy usage in classical RAM and high decoherence rate in qRAM. This paper proposes a new architecture called bucket brigade scheme, which reduces the number of switches required.

3.2 Encoding Data

There are two ways we can encode the classical data in a quantum system.

1. One using the amplitudes of the quantum state.
2. One using the basis states itself.

We look at the one encoding the data into basis state in this paper. QRAM uses quantum superposition to access a superposition of memory cells. A superposition of data registers d is returned that is mapped to the address register a. 'j' is the index of the memory cell in equation 1.

$$\sum_j \psi_j |j\rangle_a \quad \xrightarrow{\text{qRAM}} \quad \sum_j \psi_j |j\rangle_a |D_j\rangle_d \ , \tag{1}$$

3.3 Naive Architecture

The design of qRAM architecture is motivated from the less efficient type of classical RAM architecture called 'fanout' RAM. The fanout model employs a tree structure where the nodes are transistors that can direct incoming current left or right depending on their state. Each level of the tree represents a specific address bit, and the value of the address bit can activate a level-specific switch. This switch instructs all transistors at that level to either send current left or right. As a result, only one complete path is created from the root of the tree to a leaf. This is shown in figure 3. A binary tree with n-levels can have memory contents at its leaves for an n-bit address. All transistors 2^k

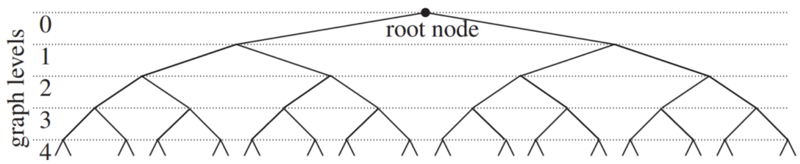

Figure 3: Fanout architecture of classical RAM [3]

within level k of a tree are controlled by a single switch that can enable all of them to route current to left or right. Thus making an exponential number of switches to be turned for each memory call.

Direct Translation of the above fanout scheme into qRAM can lead to a gigantic superposition, which is hard to keep from decoherence and has very high error rates. Suppose now, each node of a tree represent a qubit, where $|0\rangle$ represents to go left and $|1\rangle$ represents to go right .The address register of n qubits controls n quantum control lines that act coherently on each level of the bifurcation graph. A superposition of addresses is entangled with a set of switches that selects a superposition of paths through the graph. This is a precarious superposition shown in equation 2 where j_k is the kth bit of the address register, and s_k is the state of the 2^k switches controlled by it.

$$\sum_j \psi_j \; |j_0 j_1 \cdots j_{n-1}\rangle_a \otimes |j_0\rangle_{s_0} |j_1\rangle_{s_1}^{\otimes 2} \cdots |j_{n-1}\rangle_{s_{n-1}}^{\otimes 2^{n-1}} \tag{2}$$

Quantum bus is a signal that travels back and forth in the bifurcation tree. To complete a quantum memory call the quantum bus is injected into the root node and follows the superposition of paths through the graph [3]. If each memory cell stores one bit of information, the quantum bus is a single qubit that can be copied using a Controlled-NOT gate. For any data register with arbitrary length k we can send the quantum bus back and forth to carry the information. However, implementing such a qRAM scheme for a reasonably sized memory is highly demanding in practice.

3.4 Bucket Brigade Scheme

The term "bucket brigade" was coined in the qRAM paper to describe the process of sending both the address register and the signal through the bifurcation graph. The term is metaphorically similar to passing buckets of water along a line of improvised firefighters, who carve out a path that crosses the entire graph. This path allows for the extraction of information from the memory cells.

In this architecture, the key idea is to use qutrits instead of qubits. A qutrit is a three state quantum system. The qutrit states are described as $|wait\rangle, |left\rangle or |right\rangle$. Initially all the qutrits are initialized to $|wait\rangle$ state. To perform a memory access, input address register qubits are sent one by one from the root of the bifurcation tree.

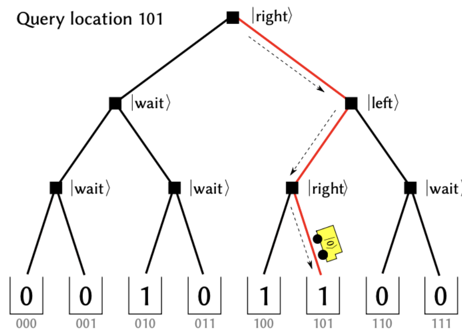

Figure 4: Bucket brigade scheme with query 101 [8]

Each time the following unitary operations 3 and 4 are applied at some level k. Here, $|s\rangle$ refers to some arbitrary state after unitary. As U is reversible, state $|s\rangle$ is unconcerned and will be undone later on [8].

$$U(|0\rangle\,|\text{wait}\rangle) \quad \rightarrow \quad |s\rangle\,|\text{left}\rangle \tag{3}$$

$$U(|1\rangle\,|\text{wait}\rangle) \quad \rightarrow \quad |s\rangle\,|\text{right}\rangle \tag{4}$$

The key difference between the fanout scheme and the bucket brigade scheme is that the later operates in two phases. It initializes the path before sending in a quantum bus. In bucket brigade, the coherence complexity of accessing a single memory cell is $O(logN)$. Since, only $O(logN)$ qutrits are in state other than $|wait\rangle$. Overall query complexity is $O(klogN)$ for some k length address register. In this scheme, even if all qutrits are in superposition, the state remains highly resistant to noise. If a fraction ϵ of the gates are decohered, the resulting state has an average fidelity of $O(1 - \epsilon logN)$ [3]. Figure 4 shows operation of the bucket brigade architecture.

3.5 Circuit Implementation with Qiskit

Quantum circuit implementation is referred from [11], contains three sections.

- **Address qubits**: Holds address register bits
- **Routing nodes**: Routes the address qubits to find the memory cell based on states of address qubits.
- **Memory cells**: Qubits that store classical information.

In the figure 5 the readout qubit is measured to read the memory cell state. Read/write enable qubit is used to switch between reads and writes. This is just a brief overview of the complex circuit. In fact, the bucket brigade model uses both qubits and qutrits, but most quantum processors today only support qubits. Synthesizing qutrits will be challenging.

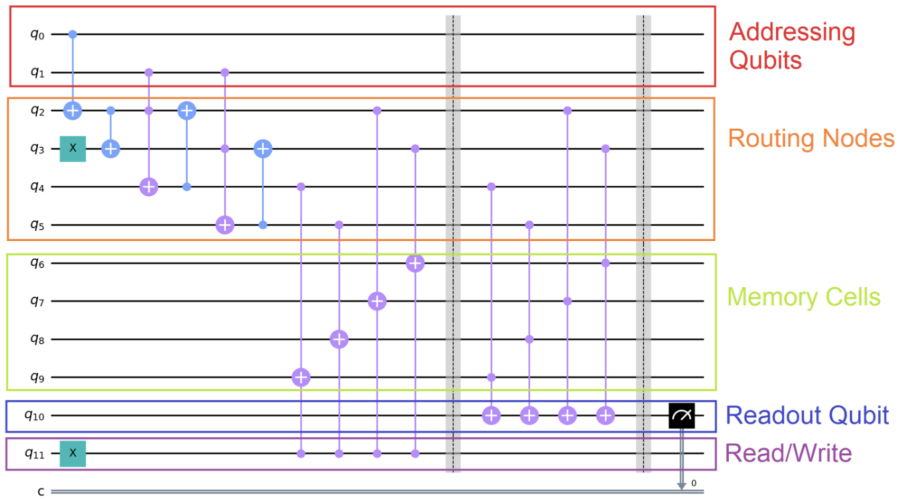

Figure 5: Circuit implementation using qiskit
[11]

3.6 Current State and Experiments

Current state of building a qRAM is only on paper. Though lots of progress is being made through experiments and prototypes, we are not at the point of having a general purpose, scalable qRAM. Many technical hurdles remain. One recent paper on qRAM [2] by Chen et al proposed possible experimental implementations of constructing the bifurcation tree. Prof. Seth Llyod who was co-author of the original QRAM paper is involved in this latest experimental analysis at MIT. They used photons for theoretical analysis of qRAM efficiency and query fidelity. The numerical simulations in the paper show that the architecture can achieve > 0.99 fidelity with $>$ kHz success rate for a qRAM containing 102 memory cells.

In 2019, a team from Tsinghua University built an experimental random access memory using 210 memory cells based on a dual-rail encoding of qubits [5] [4]. It reports an experiment that realizes a random access quantum memory of 105 qubits carried by 210 memory cells in a macroscopic atomic ensemble. It uses a different architecture and not based on bucket brigade. Here, they input an optical qubit into a cell pair and then retrieved and read out the stored qubit after a storage time of 1.38 microseconds. A 1.38 microsecond storage time corresponds to a coherency time of 725 nanoseconds. Measured state fidelities of the retrieved optical qubits after storage in the 210-cell quantum memory are shown in the figure 6.

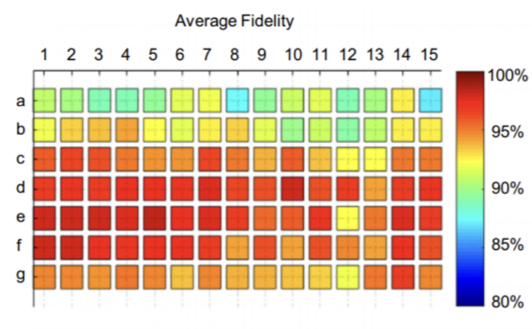

Figure 6: Measured state fidelities of the retrieved optical qubits [5]

4 Conclusion

There are still many open questions regarding how to best design and engineer qRAMs. Features like fan-out and bucket brigade architectures seem promising in theory but need to be realized in practice. It is clear that qRAMs are essential for advancement in quantum computing and the problems they can solve. Their implications depend on overcoming current limitations and engineering challenges. Significant progress is still needed. With each experiment, report, and analysis, we expand our knowledge about how to make qRAMs work at a large, practical scale. But a "quantum RAM" remains aspirational until we reach that point. However, a steady progress is being made.

References

[1] P A M Casares and M A Martin-Delgado. A quantum active learning algorithm for sampling against adversarial attacks. *New Journal of Physics*, 22(7):073026, jul 2020.

[2] K. C. Chen, W. Dai, C. Errando-Herranz, S. Lloyd, and D. Englund. Scalable and high-fidelity quantum random access memory in spin-photon networks. *PRX Quantum*, 2(3), aug 2021.

[3] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical Review Letters*, 100(16), apr 2008.

[4] Aaron Green and Emily Kaplitz. Quantum random access memory. https://www.cs.umd.edu/class/fall2019/cmsc657/projects/group_11.pdf, aug 2019.

[5] N Jiang, Y-F Pu, W Chang, C Li, S Zhang, and L-M Duan. Experimental realization of 105-qubit random access quantum memory. *npj Quantum Information*, 5(1):28, 2019.

[6] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning, 2018.

[7] Alexander I Lvovsky, Barry C Sanders, and Wolfgang Tittel. Optical quantum memory. *Nature photonics*, 3(12):706–714, 2009.

[8] Olivia Di Matteo. A primer on quantum ram. https://github.com/qsharp-community/qram, sep 2020.

[9] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[10] Anindya Rastogi, Erhan Saglamyurek, Taras Hrushevskyi, and Lindsay J. LeBlanc. Superradiance-mediated photon storage for broadband quantum memory. *Physical Review Letters*, 129(12), sep 2022.

[11] Quantum Computin UK. Implementing qram in qiskit with code. https://quantumcomputinguk.org/tutorials/implementing-qram-in-qiskit-with-code, aug 2021.

[12] Ye Wang, Mark Um, Junhua Zhang, Shuoming An, Ming Lyu, Jing-Ning Zhang, L-M Duan, Dahyun Yum, and Kihwan Kim. Single-qubit quantum memory exceeding ten-minute coherence time. *Nature Photonics*, 11(10):646–650, 2017.