Quantum Computers will need RAM

Alexandru Paler
Aalto University

Aalto University

HIIT | HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

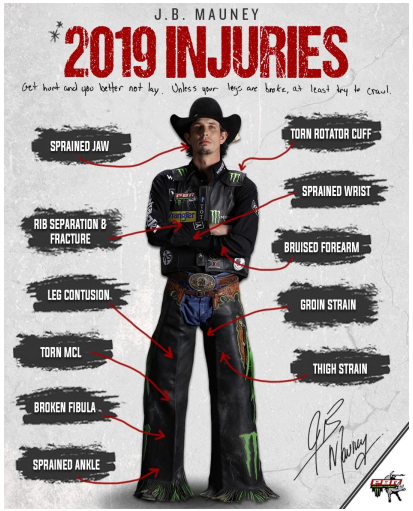
Contents

We build quantum computers because…

Motivation

Physical (raw) qubits - not well behaved

- faulty - affected by environmental noise and manufacturing inconsistencies
- solitary (not many) on a device



JB Mauney, noted for picking the rankest bull when given the choice

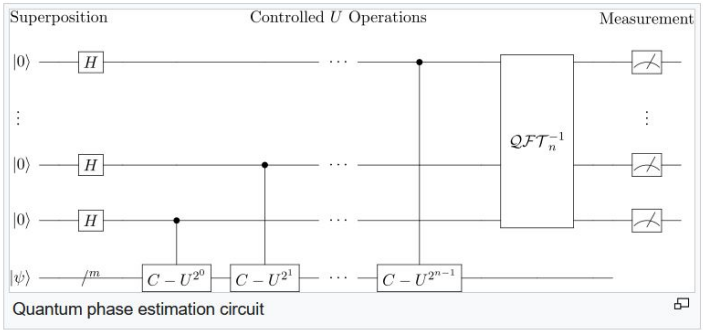
Error-corrected qubits - controlling the risks

- not faulty - or controlled failure rates
- difficult to achieve due to lack of hardware qubits, not scalable classical software etc.

Speed-Ups - Quantum computers are faster …

Exponential - LOG(N)

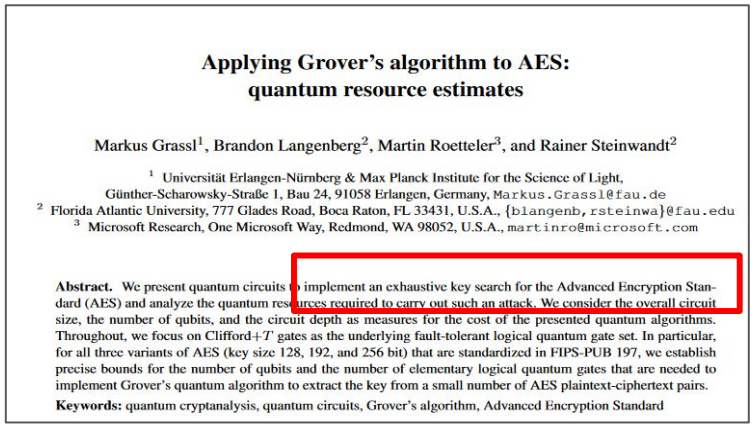
- finding the prime factors of an integer – Shor's algorithm
  - given a composite number N
  - find a non-trivial divisor of N
- solving a set of linear equations – HHL
  - a N × N Hermitian matrix A
  - a unit vector b
  - find the solution vector A **x** = b
  - maintains its logarithmic scaling in N only for sparse or low rank matrices



Quantum phase estimation circuit

Quadratic - SQRT(N)

- unstructured search that finds with high probability the unique input to a black box function – Grover's algorithm
  - cryptanalysis AES
  - combinatorial optimisation
  - travelling salesman



Applying Grover's algorithm to AES: quantum resource estimates

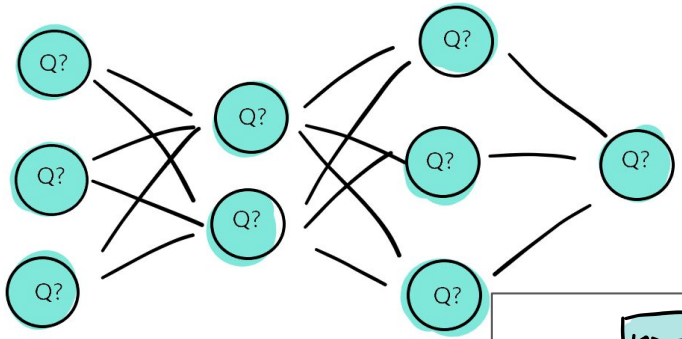
Markus Grassl[1], Brandon Langenberg[2], Martin Roetteler[3], and Rainer Steinwandt[2]

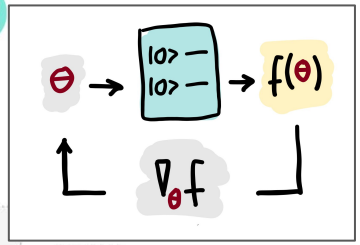
[1] Universität Erlangen-Nürnberg & Max Planck Institute for the Science of Light, Günther-Scharowsky-Straße 1, Bau 24, 91058 Erlangen, Germany, Markus.Grassl@fau.de
[2] Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, U.S.A., {blangenb,rsteinwa}@fau.edu
[3] Microsoft Research, One Microsoft Way, Redmond, WA 98052, U.S.A., martinro@microsoft.com

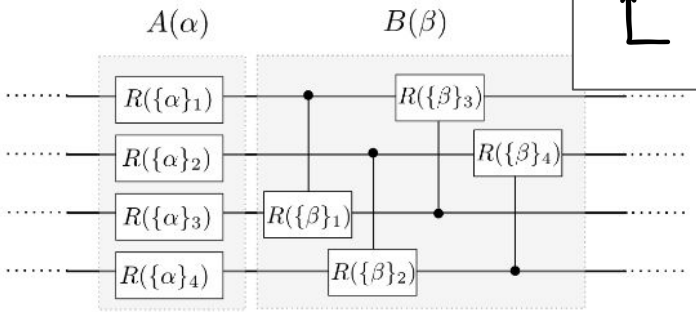
Abstract. We present quantum circuits to implement an exhaustive key search for the Advanced Encryption Standard (AES) and analyze the quantum resources required to carry out such an attack. We consider the overall circuit size, the number of qubits, and the circuit depth as measures for the cost of the presented quantum algorithms. Throughout, we focus on Clifford+T gates as the underlying fault-tolerant logical quantum gate set. In particular, for all three variants of AES (key size 128, 192, and 256 bit) that are standardized in FIPS-PUB 197, we establish precise bounds for the number of qubits and the number of elementary logical quantum gates that are needed to implement Grover's quantum algorithm to extract the key from a small number of AES plaintext-ciphertext pairs.

Keywords: quantum cryptanalysis, quantum circuits, Grover's algorithm, Advanced Encryption Standard

Quantum Machine Learning







Quantum Machine Learning - QML

- Using quantum computers like neural networks
- Explore the interplay of ideas from quantum computing and machine learning

Differentiable programming

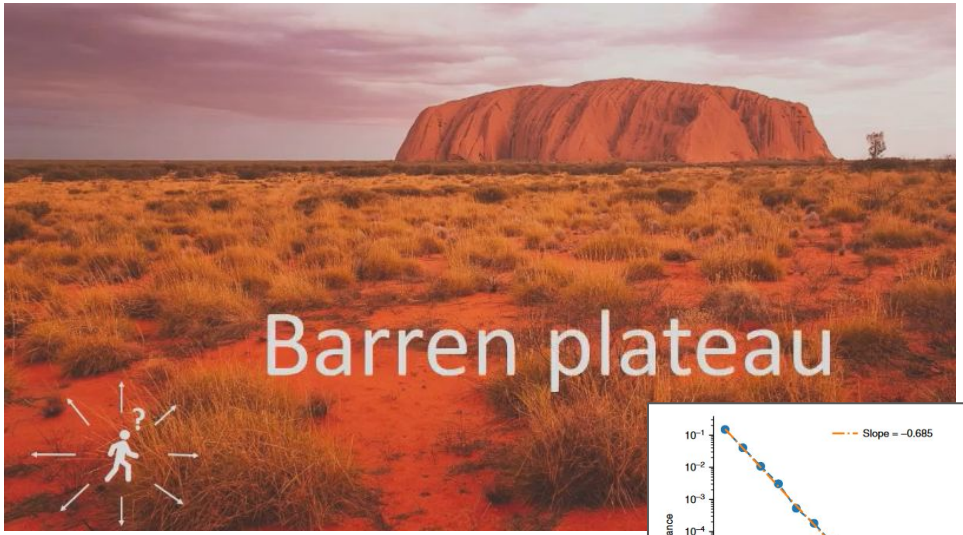
- Paradigm where the algorithms are not hand-coded, but learned
- Implemented in software libraries such as TensorFlow and PyTorch

A quantum gradient

- the vector of partial derivatives of a quantum function
- can be computed by quantum computers
- gradient-based optimization is widely used in QML

With materials from https://pennylane.ai/qml/

Barren plateaus - gradients vanish without additional mitigating steps



Maria Kieferova https://youtu.be/YDiT40If-3U

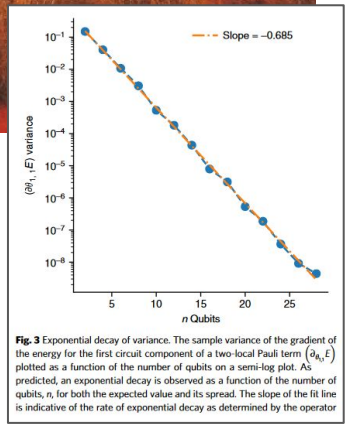


Fig. 3 Exponential decay of variance. The sample variance of the gradient of the energy for the first circuit component of a two-local Pauli term $\left(\partial_{\theta_1} \hat{E}\right)$ plotted as a function of the number of qubits on a semi-log plot. As predicted, an exponential decay is observed as a function of the number of qubits, n, for both the expected value and its spread. The slope of the fit line is indicative of the rate of exponential decay as determined by the operator

Article | Open Access | Published: 16 November 2018

Barren plateaus in quantum neural network training landscapes

Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush & Hartmut Neven

Nature Communications **9**, Article number: 4812 (2018) | Cite this article

Gradient descent without some additional strategy cannot circumvent following exponential challenges on a quantum device in polynomial time

Comparison with classical deep neural network gradients

- the different scaling of the vanishing gradient
- the complexity of computing expected values.

Gradient vanishes exponentially:

- classical deep neural network -> in the number of layers
- quantum circuit -> in the number of qubits

Number of paths:

- In the classical case, exponential in the number of layers
- In the quantum case, exponential in the number of gates

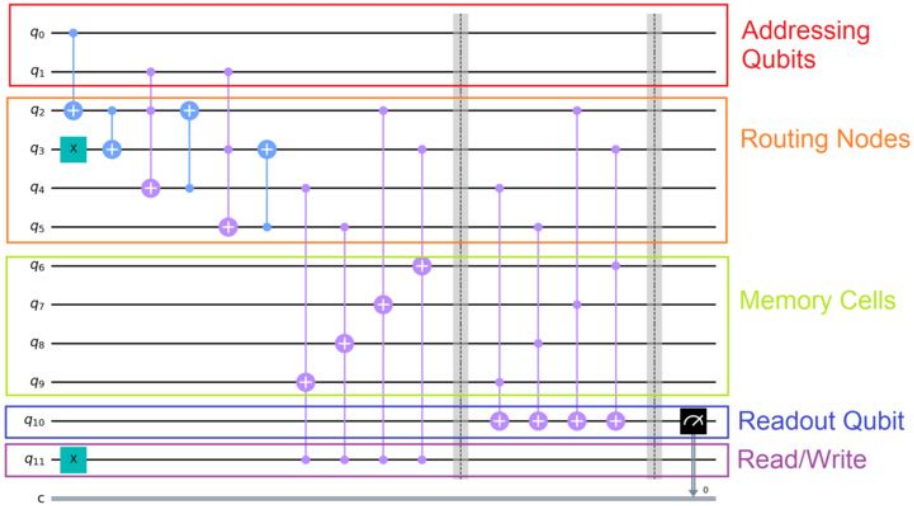
Estimation cost

- classical neural network -> limited by machine precision. Even if the gradient is small, as long as it is consistent enough between batches, the method may eventually succeed.
- On a quantum device, estimation is a random walk with exponentially small probability of exiting the barren plateau

Quantum RAM - QRAM - exponential number of memory cells (physical qubits)

The bucket brigade architecture is a form of QRAM that uses qubits to store classical information. It consists of 3 main parts:

1. Addressing qubits: These qubits hold the address for the memory cell we wish to read/write
2. Routing nodes: These are nodes made up of qubits that find the memory cell based upon the states of the addressing qubits
3. Memory cells: These are made up of qubits that **store classical information** that we wish to read or write to

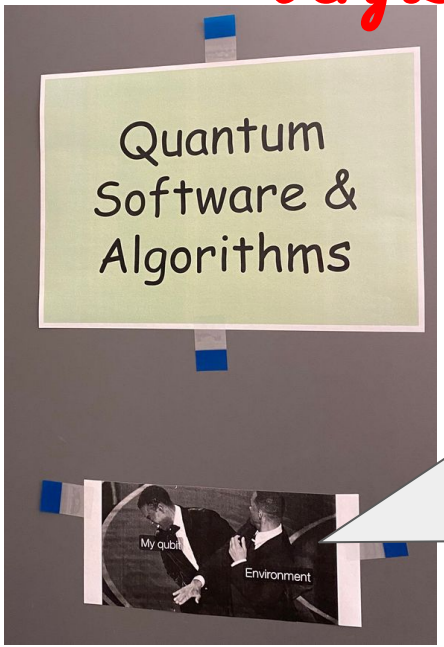


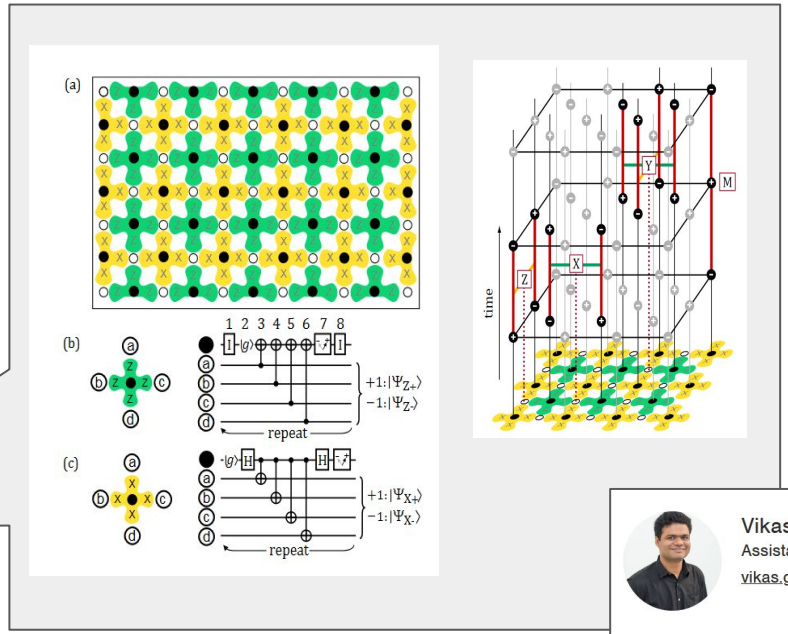
https://quantumcomputinguk.org/tutorials/implementing-qram-in-qiskit-with-code

Parallelizing the queries in a bucket-brigade quantum random access memory

Alexandru Paler, Oumarou Oumarou, and Robert Basmadjian
Phys. Rev. A **102**, 032608 – Published 8 September 2020

Very Large Scale





CS-E4680 - Quantum Machine Learning D, Project, 5.9.2022-8.12.2022

CS-E4700 - Logic and Hard Computational Problems D, Lecture, 6.9.2022-17.11.2022

Vikas Kumar Garg
Assistant Professor
vikas.garg@aalto.fi

Petteri Kaski
Associate Professor
petteri.kaski@aalto.fi

Alexandru Paler
Assistant Professor
alexandru.paler@aalto.fi

Classical Computers and RAM

Operating Systems for quantum computers

- "A set of **interrelated components** working **together** toward some **common** objective"

1960's – IBM System/360

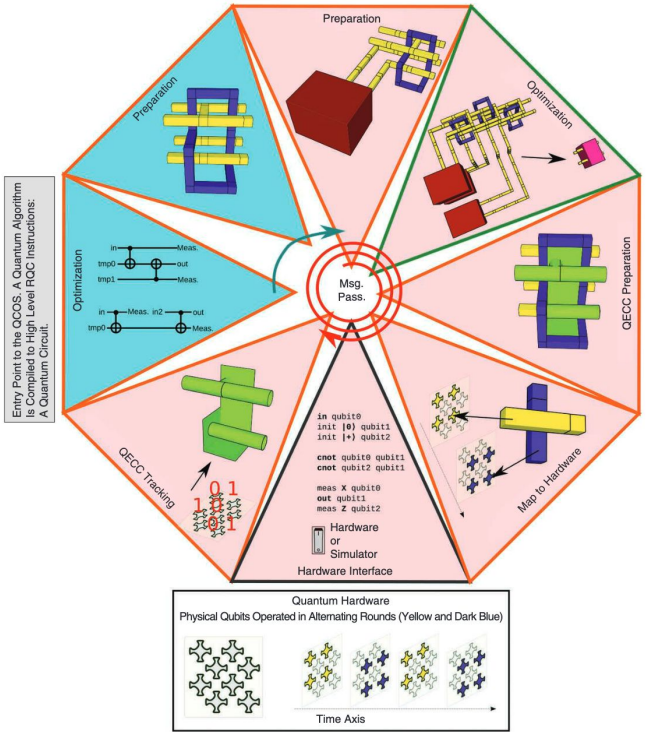
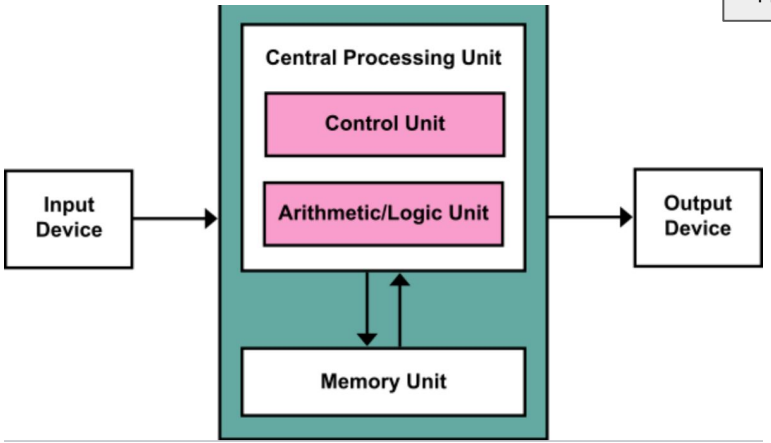


FIGURE 1. The architecture and interaction of the QCOS. Each triangle is a QCOS component. The arrows indicate an arbitrary execution order. To execute an algorithm, the first part of the QCOS is offline (blue arrow and triangles), after which online preparations, optimizations, and other processes are performed in a loop (red spiral and triangles). The component pictograms represent elements of quantum circuits protected by the braided surface quantum error-correcting code (QECC).[12] RQC: reliable quantum computer.

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9153285

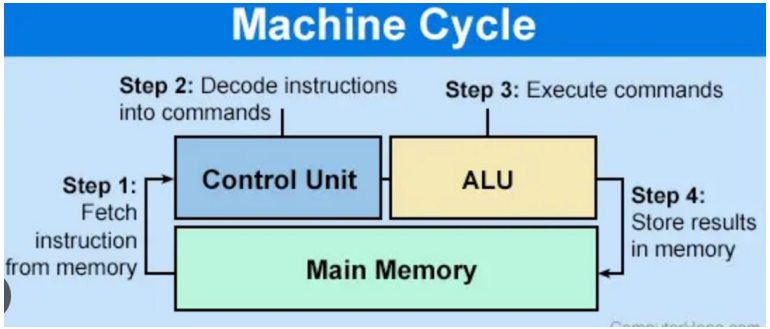
von Neumann Architectures

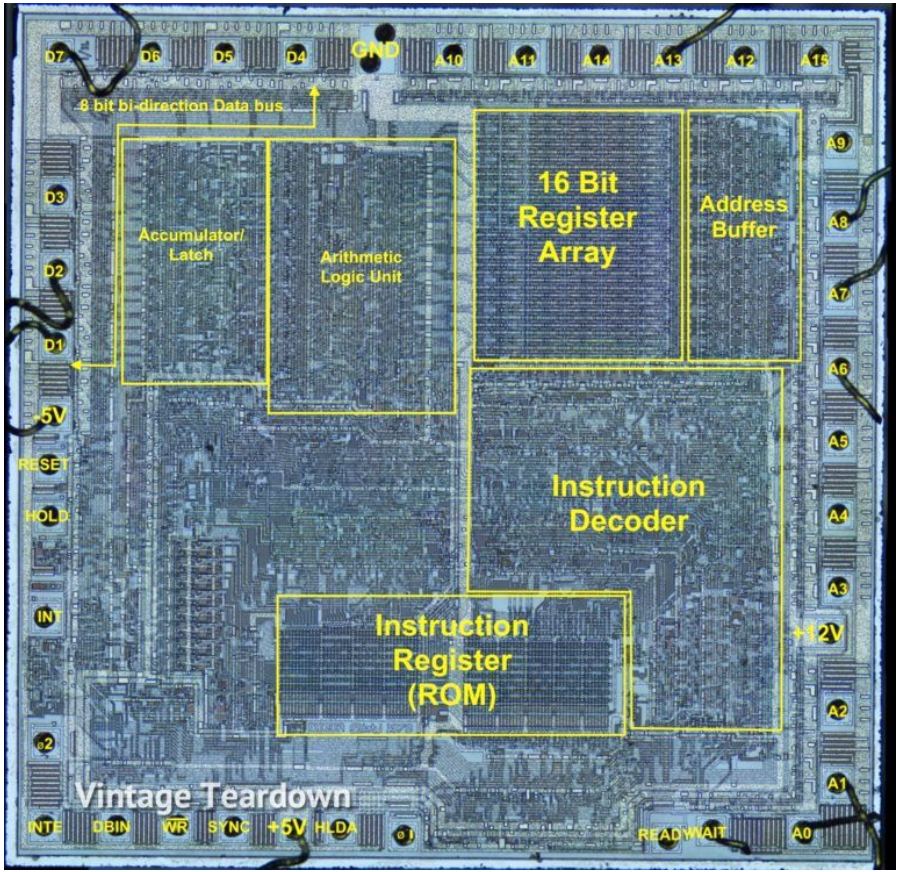
- A processing unit with both an arithmetic logic unit and processor registers
- A control unit that includes an instruction register and a program counter
- **Memory that stores data and instructions**
- External mass storage
- Input and output mechanisms



Processor Registers

- User-visible registers
  - available to all user programs or applications
  - data registers and address registers
- Control and status registers
  - control the operation of the processor
  - usually visible only to the operating system
- Program Counter (PC): the address of the next instruction to execute
- Instruction register (IR): the mostly recently fetched instruction





Intel 8080A https://newbedev.com/where-are-registers-and-what-do-they-look-like

Memory hierarchy

- Typically there are several levels
  - Data path itself (latches, etc., not visible to programmer)
  - Registers (visible to the programmer/compiler)
  - Cache (not visible to the programmer/compiler, but OS handles)
    - L1 (often zero-wait, usually per core)
      - Often separate for code and data
    - L2 (shared)
    - L3 (can be found off-chip)
  - Main memory (usually DRAM)
  - Disk memory
- Differences in sizes and speeds are significant
- Note that the same data can reside in several places

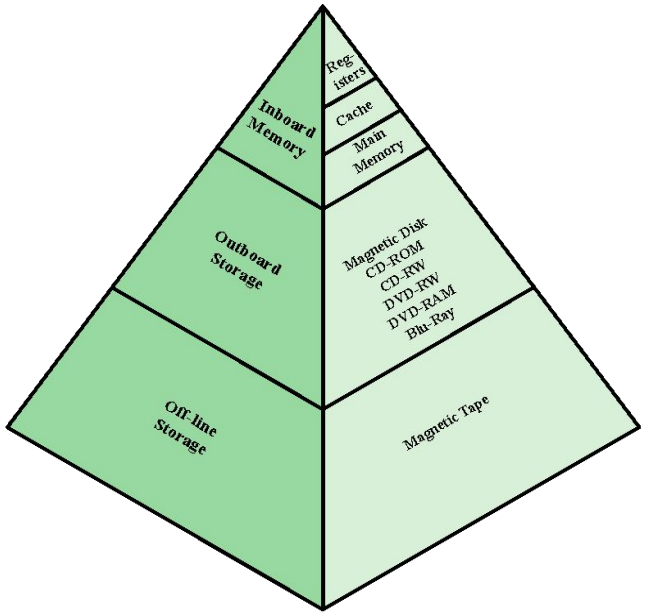


Figure 1.14 The Memory Hierarchy

QRAM

Original idea

Quantum random access memory

Vittorio Giovannetti[1], Seth Lloyd[2], Lorenzo Maccone[3]
[1] *NEST-CNR-INFM & Scuola Normale Superiore, Piazza dei Cavalieri 7, I-56126, Pisa, Italy.*
[2] *MIT, RLE and Dept. of Mech. Engin. MIT 3-160, 77 Mass. Av., Cambridge, MA 02139, USA.*
[3] *QUIT, Dip. Fisica "A. Volta", Univ. Pavia, via Bassi 6, I-27100 Pavia, Italy.*

A random access memory (RAM) uses n bits to randomly address $N = 2^n$ distinct memory cells. A quantum random access memory (qRAM) uses n qubits to address any quantum superposition of N memory cells. We present an architecture that exponentially reduces the requirements for a memory call: $O(\log N)$ switches need be thrown instead of the N used in conventional (classical or quantum) RAM designs. This yields a more robust qRAM algorithm, as it in general requires entanglement among exponentially less gates, and leads to an exponential decrease in the power needed for addressing. A quantum optical implementation is presented.

PACS numbers: 03.67.Lx 03.65.Ud 03.67.-a

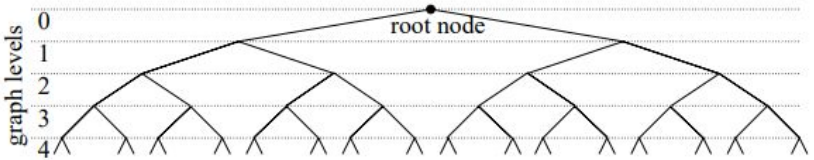


FIG. 1: Bifurcation graph of the RAM addressing.

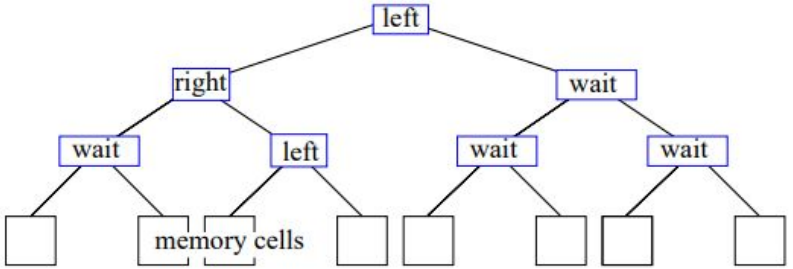


FIG. 2: Bifurcation graph of the bucket-brigade architecture. Here the third memory cell is addressed (address register 010).

Circuit implementation

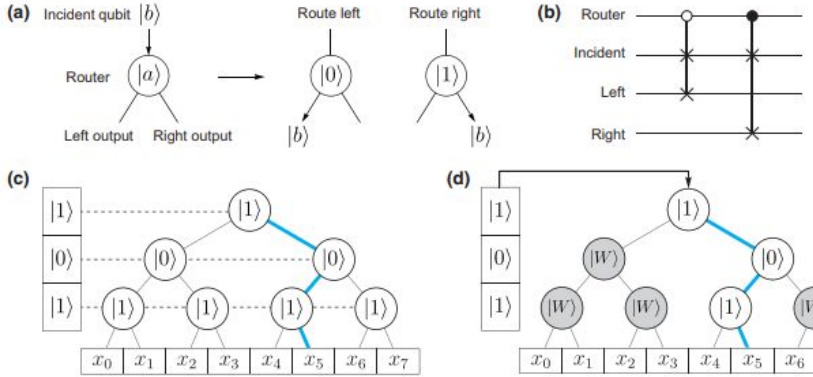
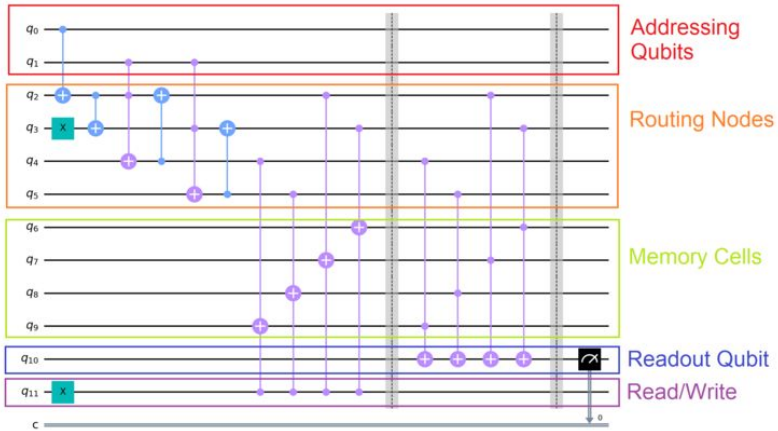


FIG. 1. QRAM implementations. (a) Quantum router. The router directs an incident qubit $|b\rangle$ at its top port out of either the left or right output port conditioned on the state $|a\rangle$ of the router. When $|a\rangle = |0\rangle$ ($|1\rangle$), the incident qubit leaves out of the left (right) port. (b) Example of a quantum circuit that implements the routing operation using two controlled-SWAP gates, one conditioned on the control being $|0\rangle$ (open circle) and the other conditioned on the control being $|1\rangle$ (filled circle). (c) Fanout QRAM. Each address qubit controls the states of all routers within the corresponding level of the binary tree. A bus qubit injected at the top node then follows the path (blue) to the specified memory element. (d) Bucket-brigade QRAM, utilizing routers with three sates: wait $|W\rangle$, route left $|0\rangle$, and route right $|1\rangle$. The address qubits themselves are routed into the tree, carving out a path to the memory.



https://quantumcomputinguk.org/tutorials/implementing-qram-in-qiskit-with-code

Circuit implementation

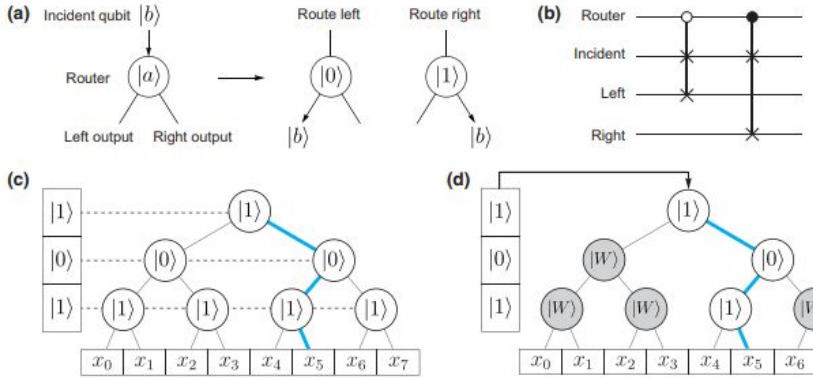


FIG. 1. QRAM implementations. (a) Quantum router. The router directs an incident qubit $|b\rangle$ at its top port out of either the left or right output port conditioned on the state $|a\rangle$ of the router. When $|a\rangle = |0\rangle$ ($|1\rangle$), the incident qubit leaves out of the left (right) port. (b) Example of a quantum circuit that implements the routing operation using two controlled-SWAP gates, one conditioned on the control being $|0\rangle$ (open circle) and the other conditioned on the control being $|1\rangle$ (filled circle). (c) Fanout QRAM. Each address qubit controls the states of all routers within the corresponding level of the binary tree. A bus qubit injected at the top node then follows the path (blue) to the specified memory element. (d) Bucket-brigade QRAM, utilizing routers with three sates: wait $|W\rangle$, route left $|0\rangle$, and route right $|1\rangle$. The address qubits themselves are routed into the tree, carving out a path to the memory.

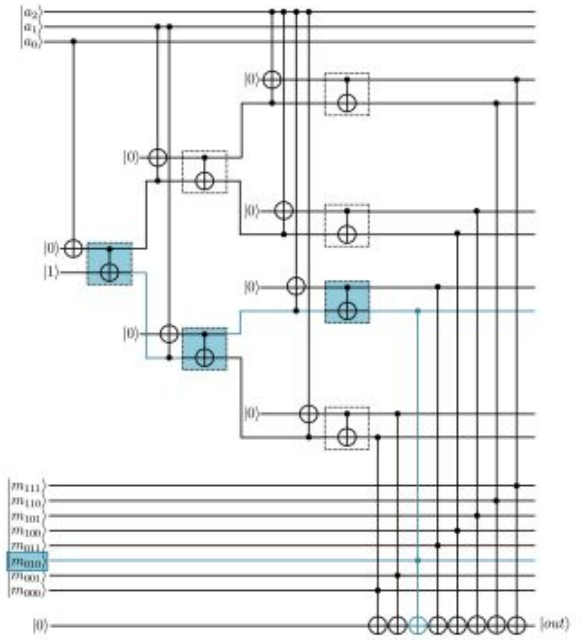


FIG. 2. One method of constructing a bucket brigade style qRAM circuit. Original image taken from [13]. The circuit is implemented using only CNOTs and Toffolis, which we decompose over Clifford+T. The circuit is independent of the contents of the memory, which is initialized separately in the lower register. For a fully reversible qRAM query we must run this circuit as written, and then uncompute the fanout of the address bits

Fault-Tolerance

Fault tolerant resource estimation of quantum random-access memories

Olivia Di Matteo,[1,2,3,*] Vlad Gheorghiu,[3,4,†] and Michele Mosca[3,4,5,6,‡]
[1]*TRIUMF, Vancouver, BC, V6T 2A3, Canada*
[2]*Department of Physics & Astronomy, University of Waterloo, Waterloo, ON, N2L 3G1, Canada*
[3]*Institute for Quantum Computing, University of Waterloo, Waterloo, ON, N2L 3G1, Canada*
[4]*Department of Combinatorics & Optimization, University of Waterloo, Waterloo, ON, N2L 3G1, Canada*
[5]*Perimeter Institute for Theoretical Physics, Waterloo, ON, N2L 6B9, Canada*
[6]*Canadian Institute for Advanced Research, Toronto, ON, M5G 1Z8, Canada*
(Dated: Version of January 24, 2020)

Quantum random-access look-up of a string of classical bits is a necessary ingredient in several important quantum algorithms. In some cases, the cost of such quantum random-access memory (qRAM) is the limiting factor in the implementation of the algorithm. In this paper we study the cost of fault-tolerantly implementing a qRAM. We construct and analyze generic families of circuits that function as a qRAM, discuss opportunities for qubit-time tradeoffs, and estimate their resource costs when embedded in a surface code.

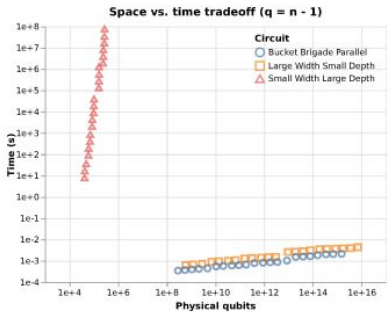


FIG. 7. Space (physical qubits) vs. time tradeoff for basic circuits. Values are calculated assuming a surface code cycle time of 200ns, state injection error rate of 10^{-4}, and intrinsic gate error rate of 10^{-5}. We note that these values are quite optimistic given the current state of quantum hardware. Each point corresponds to a different memory size 2^n from $n = 15$ to $n = 36$, with smaller n using fewer resources in each case. Memory fullness q is set to $n - 1$ for each n for the large width/depth circuits.

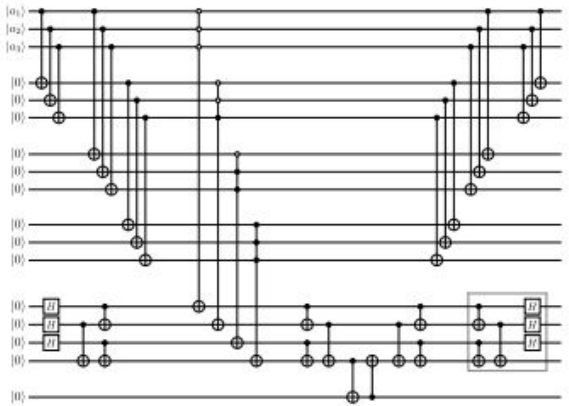


FIG. 4. A circuit with small depth but a large number of qubits, where the implementation of the MPMCTs is performed in parallel. Each MPMCT requires $n - 1$ ancillae, significantly increasing the number of qubits. The register containing Hadamards prepares the superposition over all even-parity states, which eliminates the need to uncompute the MPMCTs when making the query fully reversible. When multiple queries are performed, we do not need to implement the portion of the circuit in the dotted box, as the uncomputation of the parity following the CNOT from the output bit leaves us in the even-parity superposition; the dotted box serves to return these qubits to $|0\rangle$. Resources for the creation and destruction of the even-parity superposition are neglected from our analysis, as this need only be performed once, and can then be used for a large number of queries.

QROM

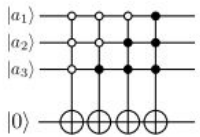
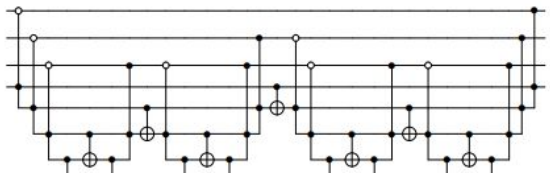


FIG. 3. A qRAM circuit with few qubits but large depth. The addresses of all 2^q locations known to contain a 1 are implicitly stored in the circuit as mixed-polarity multiple control Toffoli gates.

The read-only aspect of QROM makes it distinctly different from QRAM in that one can read from QROM but cannot write to it during the course of a computation



Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity

Ryan Babbush,[1,*] Craig Gidney,[2] Dominic W. Berry,[3] Nathan Wiebe,[4]
Jarrod McClean,[1] Alexandru Paler,[5] Austin Fowler,[2] and Hartmut Neven[1]
[1]*Google Inc., Venice, CA 90291, United States*
[2]*Google Inc., Santa Barbara, CA 93117, United States*
[3]*Department of Physics and Astronomy, Macquarie University, Sydney, NSW 2109, Australia*
[4]*Microsoft Research, Redmond, WA 98052, United States*
[5]*Institute for Integrated Circuits, Linz Institute of Technology, 4040 Linz, Austria*
(Dated: September 20, 2018)

We construct quantum circuits which exactly encode the spectra of correlated electron models up to errors from rotation synthesis. By invoking these circuits as oracles within the recently introduced "qubitization" framework, one can use quantum phase estimation to sample states in the Hamiltonian eigenbasis with optimal query complexity $\mathcal{O}(\lambda/\epsilon)$ where λ is an absolute sum of Hamiltonian coefficients and ϵ is target precision. For both the Hubbard model and electronic structure Hamiltonian in a second quantized basis diagonalizing the Coulomb operator, our circuits have T gate complexity $\mathcal{O}(N + \log(1/\epsilon))$ where N is number of orbitals in the basis. This enables sampling in the eigenbasis of electronic structure Hamiltonians with T complexity $\mathcal{O}(N^3/\epsilon + N^2\log(1/\epsilon)/\epsilon)$. Compared to prior approaches, our algorithms are asymptotically more efficient in gate complexity and require fewer T gates near the classically intractable regime. Compiling to surface code fault-tolerant gates and assuming per gate error rates of one part in a thousand reveals that one can error correct phase estimation on interesting instances of these problems beyond the current capabilities of classical methods using only about a million superconducting qubits in a matter of hours.

Using parallelised Toffoli gates

Parallelising the Queries in Bucket Brigade Quantum RAM

Alexandru Paler[+,*], Oumarou Oumarou[π], Robert Basmadjian[π]
[+] *University of Transilvania, B-dul Eroilor 29, 500036, Brașov, România*
[*] *Johannes Kepler University, Altenberger Str. 69, 4040, Linz, Austria and*
[π] *Department of Informatics, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany* [*]

Quantum algorithms often use quantum RAMs (QRAM) for accessing information stored in a database-like manner. QRAMs have to be fast, resource efficient and fault-tolerant. The latter is often influenced by access speeds, because shorter times introduce less exposure of the stored information to noise. The total execution time of an algorithm depends on the QRAM access time which includes: 1) address translation time, and 2) effective query time. The bucket brigade QRAMs were proposed to achieve faster addressing at the cost of exponentially many ancillae. We illustrate a systematic method to significantly reduce the effective query time by using Clifford+T gate parallelism. The method does not introduce any ancillae qubits. Our parallelisation method is compatible with the surface code quantum error correction. We show that parallelisation is a result of advantageous Toffoli gate decomposition in terms of Clifford+T gates, and after addresses have been translated, we achieve theoretical $\mathcal{O}(1)$ parallelism for the effective queries. We conclude that, in theory: 1) fault-tolerant bucket brigade quantum RAM queries can be performed approximately with the speed of classical RAM; 2) the exponentially many ancillae from the bucket brigade addressing scheme are a trade-off cost for achieving exponential query speedup compared to quantum read-only memories whose queries are sequential by design. The methods to compile, parallelise and analyse the presented QRAM circuits were implemented in software which is available online.

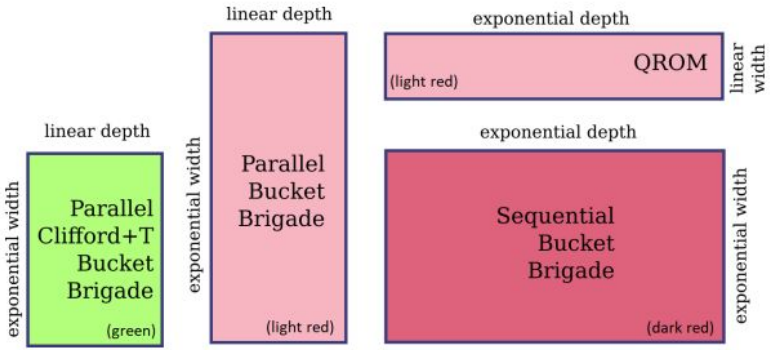


FIG. 2. Comparison between different versions of QRAM. The proposed QRAM is marked green. The parallel bucket brigade (light red) [7] introduces an exponential number of ancillae to achieve parallelism. This is not the case with our decomposition. The original (sequential) bucket brigade circuit (dark red) is sequential and has the same number of wires like the one proposed herein, but requires an exponential number of control signals. The QROM [2] circuit was specifically designed with sequential queries, but does not require the control signals.

Using parallelised Toffoli gates

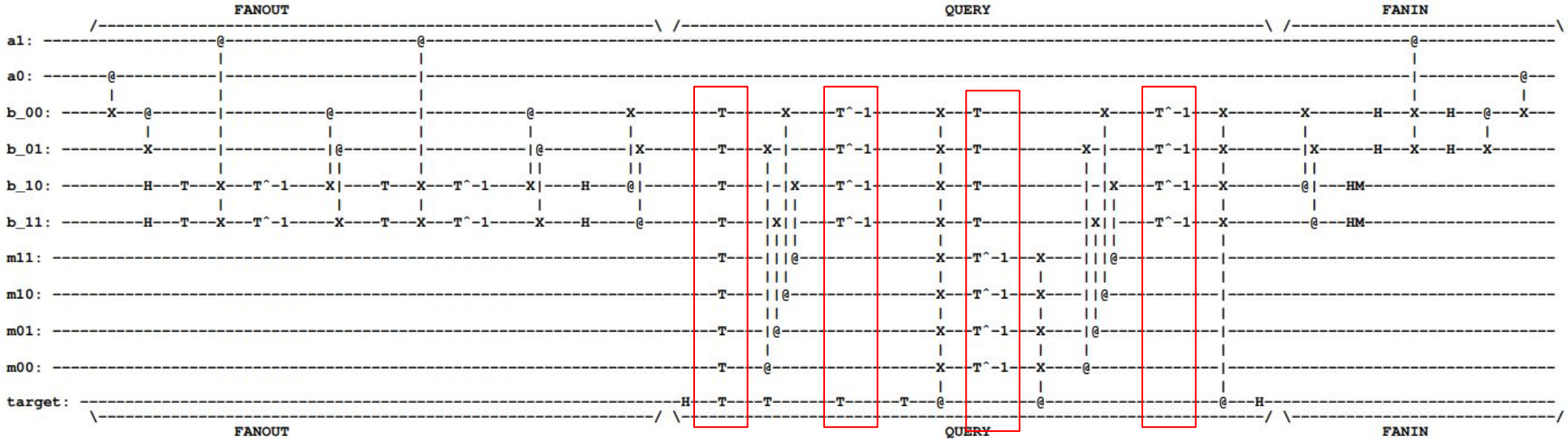


FIG. 5. The equivalent Clifford+T representation of the circuit from Fig. 1. The depth of the circuit is constant for a fixed q by FANIN and FANOUT, while QUERY has a constant depth irrespective of q or n. The circuit operates on 2^q memory cells and executes 2^n queries with $n \leq q$. The sequence of four T gates on the target qubit will be cancelled.

Using Fredkin gates

Quantum Fan-out: Circuit Optimizations and Technology Modeling

Pranav Gokhale*
University of Chicago

Samantha Koretsky
University of Chicago

Shilin Huang
Duke University

Swarnadeep Majumder
Duke University

Andrew Drucker
University of Chicago

Kenneth R. Brown
Duke University

Frederic T. Chong
University of Chicago

ABSTRACT

Instruction scheduling is a key compiler optimization in quantum computing, just as it is for classical computing. Current schedulers optimize for data parallelism by allowing simultaneous execution of instructions, as long as their qubits do not overlap. However, on many quantum hardware platforms, instructions on overlapping qubits can be executed simultaneously through *global interactions*. For example, while fan-out in traditional quantum circuits can only be implemented sequentially when viewed at the logical level, global interactions at the physical level allow fan-out to be achieved in one step. We leverage this simultaneous fan-out primitive to optimize circuit synthesis for NISQ (Noisy Intermediate-Scale Quantum) workloads. In addition, we introduce novel quantum memory architectures based on fan-out.

Our work also addresses hardware implementation of the fan-out primitive. We perform realistic simulations for trapped ion quantum computers. We also demonstrate experimental proof-of-concept of fan-out with superconducting qubits. We perform depth (runtime) and fidelity estimation for NISQ application circuits and quantum memory architectures under realistic noise models. Our simulations indicate promising results with an asymptotic advantage in runtime, as well as 7–24% reduction in error.

is argued to likely require days [72] on a supercomputer. A core aspect of the Supremacy result is a coupler activation schedule that maximizes simultaneous quantum resource utilization.

A number of papers [37, 38, 47, 63] in the architecture community have studied quantum scheduling, inspired by techniques from the classical setting. One principle underlying these papers is exclusive activation: a qubit can be involved in at most one operation per timestep [38]. In architectural terms, this is a *structural hazard* [44]. Under exclusive activation, schedulers optimize for data parallelism by simultaneously executing instructions on disjoint qubits. However, there are natural limits to such schedulers, since instructions on overlapping qubits must be serialized.

Our work begins with a simple but consequential observation: the structural hazard of exclusive activation is not actually enforced by most quantum hardware. In fact, it can be *more* natural for a quantum processor to simultaneously execute multiple operations on shared qubits through *global interactions*. The building block of our work is the fan-out operation depicted in Figure 1. This operation can be understood purely classically. The depiction on the left in Figure 1 has four CNOT (Controlled-NOT) gates, each comprising a control (•) and a target (⊕). The target is flipped iff the control qubit is 1. This operation performs fan-out for classical

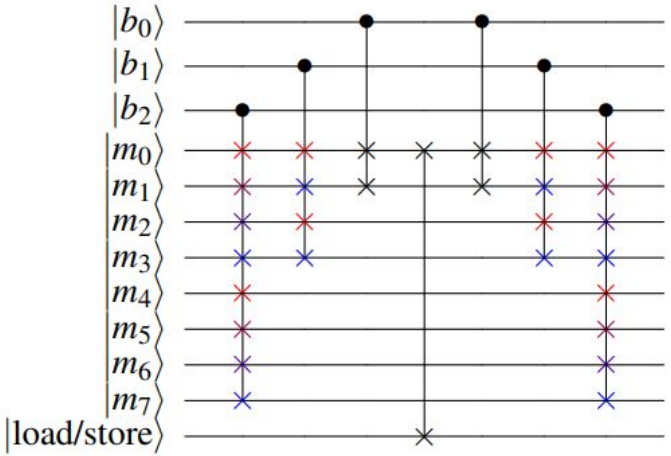


Figure 10: Architecture for an explicit quantum memory with $n = 3$ index qubits and $2^n = 8$ memory cells of bitwidth $W = 1$. Quirk demo.

… and more fault-tolerance

PAPER • OPEN ACCESS

On the robustness of bucket brigade quantum RAM

Srinivasan Arunachalam[1,2], Vlad Gheorghiu[10,2,3], Tomas Jochym-O'Connor[2,4], Michele Mosca[2,3,5,6] and Priyaa Varshinee Srinivasan[7,8,9]

Published 7 December 2015 • © 2015 IOP Publishing Ltd and Deutsche Physikalische Gesellschaft

New Journal of Physics, Volume 17, December 2015

Citation Srinivasan Arunachalam *et al* 2015 *New J. Phys.* **17** 123010

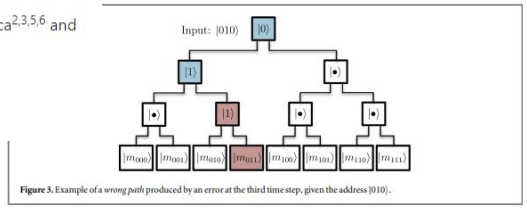


Figure 3. Example of a *wrong path* produced by an error at the third time step, given the address |010⟩.

The parameters of interest:

- are the error probability per gate, denoted by ϵ,
- the overall fidelity of the addressing circuit (i.e. the probability of a right-path), denoted by prp,
- and the number of levels in the qRAM addressing binary tree denoted by n

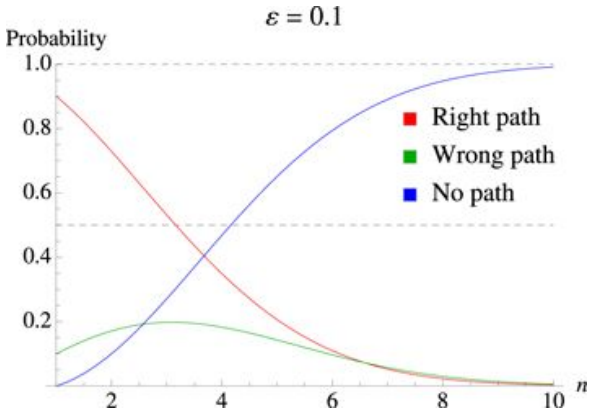


FIG. 2. Conceptual picture of noise resilience. Each ket represents the state of the QRAM when a different memory element is queried, with the superposition of kets representing a superposition of queries to different elements. When a router r suffers an error (red lightning bolt), it corrupts only the subset of queries where r is active (indicated by thick red kets); other queries in the superposition succeed regardless. Because most routers are only active in a small fraction of queries, most queries succeed and the total infidelity is low.

Resilience of Quantum Random Access Memory to Generic Noise

Connor T. Hann,[1,2,*] Gideon Lee,[1,2,3] S.M. Girvin,[1,2] and Liang Jiang[1,2,4]

[1] *Departments of Applied Physics and Physics, Yale University, New Haven, Connecticut 06520, USA*

[2] *Yale Quantum Institute, New Haven, Connecticut 06520, USA*

[3] *Yale-NUS College, 16 College Avenue West, 138527, Singapore*

[4] *Pritzker School of Molecular Engineering, The University of Chicago, Chicago, Illinois 60637, USA*

(Received 21 December 2020; accepted 13 April 2021; published 29 April 2021)

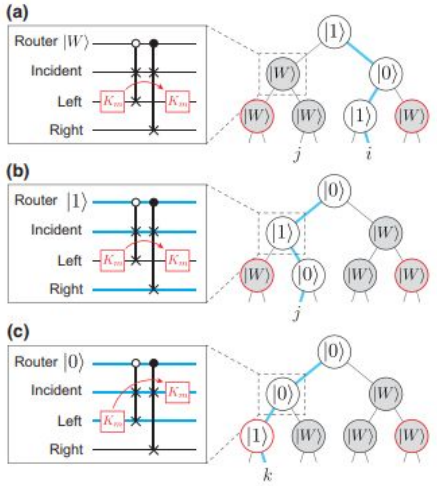


FIG. 4. Error propagation. (a),(b) Constrained propagation during queries to elements $\in g(c)$. The error in the leftmost router can propagate upward into the left output of the router indicated by the dashed box. The circuits on the left show that the error does not propagate further, regardless of whether the router is inactive (a) or active (b). In the circuit diagrams, red boxes denote errors $K_{m>0}$, and the red arrows indicate how the error propagates (i.e., how the error transforms under conjugation by the routing operation). (c) Error propagation is not constrained during queries to elements $\notin g(c)$. Note that the state of the router dictates how the error propagates in these examples.