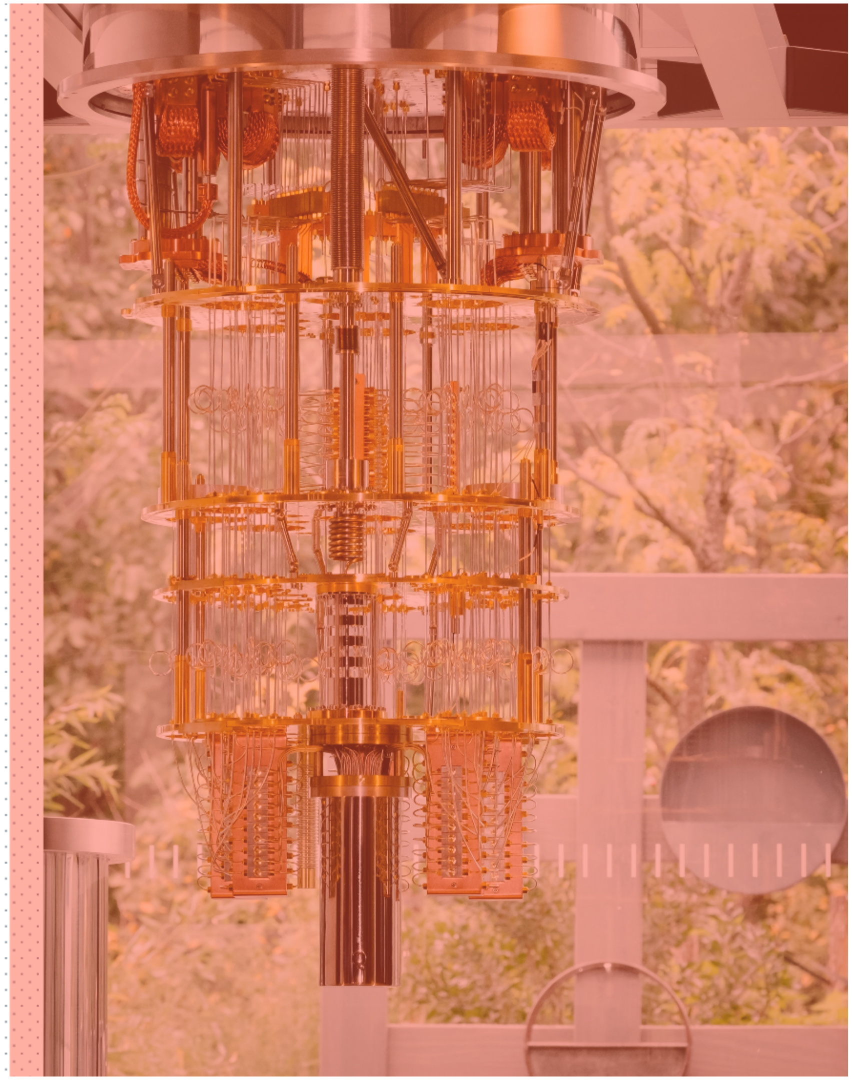
Towards Practical Hybrid Quantum / Classical Computing

M.Sc. Thesis Defense

Marcus Edwards



Conference Participation

MACHINE LEARNING FOR QUANTUM DESIGN

CAM CONFERENCE
Canada • America • Mexico
July 24-27 | Laurentian University | Sudbury, ON

QT
Quantum.Tech
SEPTEMBER 10 - 11, 2019 | BOSTON, MA

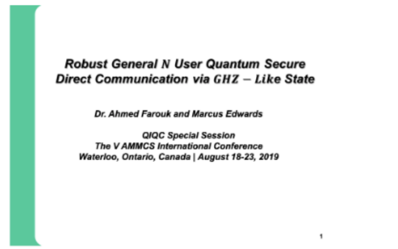
Quantum Summit

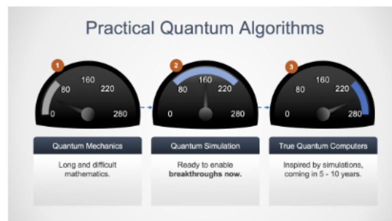
Preparing for Opportunity and Security in the Age of Quantum

by Lazaridis School of Business & Economics

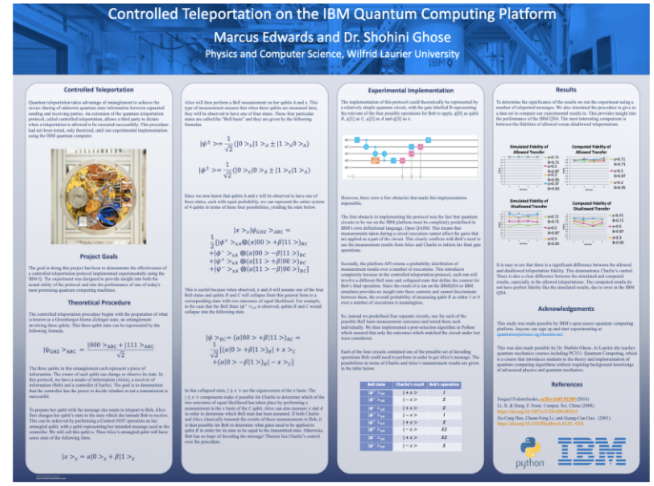
Interdisciplinary AMMCS Conference Series
The V AMMCS International Conference
Waterloo, Ontario, Canada | August 18-23, 2019

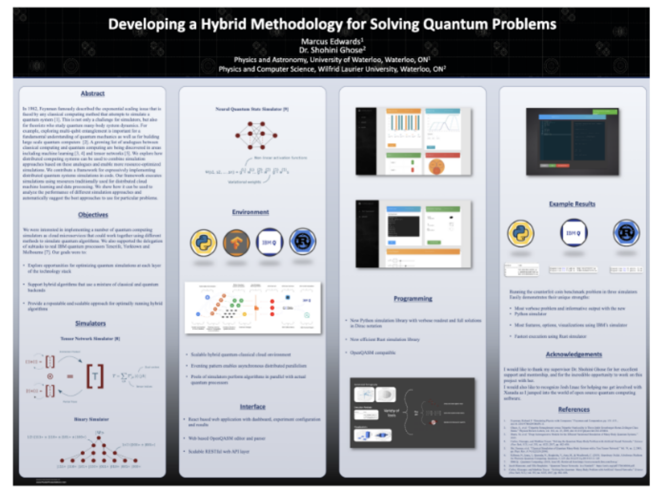

Quantum Annealers as Continuous Testing Automation Backends for Classical Web Code
Marcus Edwards[1], Dr. Atefeh Mashatan[2], Dr. Shohini Ghose[3]
Physics and Astronomy, University of Waterloo, Waterloo, ON[1], Cybersecurity Research Lab, Ted Rogers School of Information Technology Management, Ryerson University[2], Physics and Computer Science, Wilfrid Laurier University, Waterloo, ON[3]
Introduction
We present a methodology and software package for verifying WebAssembly (WASM) web code using quantum annealers. Executing a simulation of a WASM function on a quantum annealer enables fast edge-case detection which can be used to verify the correctness of the function. The method benefits from a speedup due to quantum tunneling in the annealer. This suggests an efficiency improvement over the analogous classical sampling techniques being adopted by web software companies.
Background
Efforts have been made to generalize the classical logic systems that can be optimized on D-Wave's annealers. Most programming tools deal with very low-level logic systems. For example, macro assemblers called QMASM and qbsolv have syntax similar to assembly languages and essentially just directly specify the linear and quadratic coefficients of problems intended for D-Wave. ThreeQ.jl enables the construction of QUBOs within the Julia programming language. In April 2019, Scott Pakin tackled a formidable task and introduced a method of compiling arbitrary Verilog code through a number of steps to QMASM that can be executed on a D-Wave system. This is the state-of-the-art in quantum annealer based simulation of classical programs written in traditional programming languages.
D-Wave also provides their own Python library for programming their annealers. This library has recently introduced some higher level methods that provide developers to ability to easily compose Hamiltonians that correspond to simulations of slightly more complex digital constructs. For example, library methods exist for creating simulations of combinational half and full adders.
Goals
We argue that a specific domain of continuous test automation tasks that is immediately relevant to the verification of the correctness and security of large-scale commercial software development projects could benefit from improvements by delegating a particular class of tests to execution on a quantum annealer using our new method.
We introduce a transpilation technology "QuantEmu" that enables the emulation of algorithms written in practical programming languages including Rust, C/C++, PHP, Python, Ruby, TypeScript and JavaScript on D-Wave's quantum annealer systems to this end. A logical next step for future work would be to provide a language agnostic test automation framework for the validation of web-based software.
TS
wasmer
WA
Figure 1: QuantEmu in the Stack
Methods
While WASM is more hardware agnostic than other assembly languages like x86 and we don't have to deal with assumptions in the language about execution hardware, compiling arbitrary WASM code in its entirety to QMASM is still not a reasonable endeavor. Instead, we classify types of code blocks that can be simulated either in one shot, or as a part of a multi-step validation process. A fundamental difficulty in translating WASM to QMASM is the sequential nature of WASM code. We are limited to 2048 qubits for code and data, so this defines the boundary between what WASM modules will be simulatable and what modules need to be broken up further before being simulated. Hence, our WASM transpilation process involves several intermediate steps.
Figure 2: Transpilation Steps
Methods Continued
Dependency Collapsing for Compatibility with Annealing:
• Collapses all data, name and control dependencies into a single feed-forward data tree
• Each data dependency is mapped to a qubit
• The effectively achieves the simulation of the sequential program by trading the time dimension for additional spatial dimensions
Combinational Element Abstraction:
• Each block of WASM instructions that can be combinationally executed is extracted
• Control structures are normalized to maximize the size and number of these combinational blocks
• These blocks can be compiled individually to QMASM scripts
• Each of these is a candidate for execution on a quantum annealer
Figure 3: Dependency Tracing
Results
A demo WASM module "math.wasm" provides functions that compute the dot products of 2 and 3 dimensional vectors. This module was written for the purposes of this demonstration in the S-expression format. Passing this method to our Rust transpiler is done using the following command.
Figure 4: Transpilation Command
The transpiler then reads the WASM file, categorizing each instruction into one of the categories: function calls, control flow instructions, data flow instructions, data creation instructions, data mutation instructions. All paths through the program are traced out. The user is asked whether to parallelize each function.
Figure 5: Instruction Categorization
The code is normalized using a parallelizing compilation methods inspired by Fortran's Parallelizing Fortran Compiler (PFC), and then each combinational instruction is translated to a Boolean function that captures its behavior. Finally, the user can provide constraints and the resulting data structure is handed off for conversion to a QUBO and Hamiltonian parameters.
Figure 6: Ising Hamiltonian Parameters
Ryerson University
TED ROGERS SCHOOL Information Technology Management
LAURIER Inspiring Lives.
IQC Institute for Quantum Computing
NSERC CRSNG
CYBERSECURITY RESEARCH LAB
www.ryerson.ca/crl


Robust General N User Quantum Secure Direct Communication via GHZ – Like State
Dr. Ahmed Farouk and Marcus Edwards
QIQC Special Session
The V AMMCS International Conference
Waterloo, Ontario, Canada | August 18-23, 2019


Practical Quantum Algorithms
Quantum Mechanics
Long and difficult mathematics.
Quantum Simulation
Ready to enable breakthroughs now.
True Quantum Computers
Inspired by simulations, coming in 5 - 10 years.

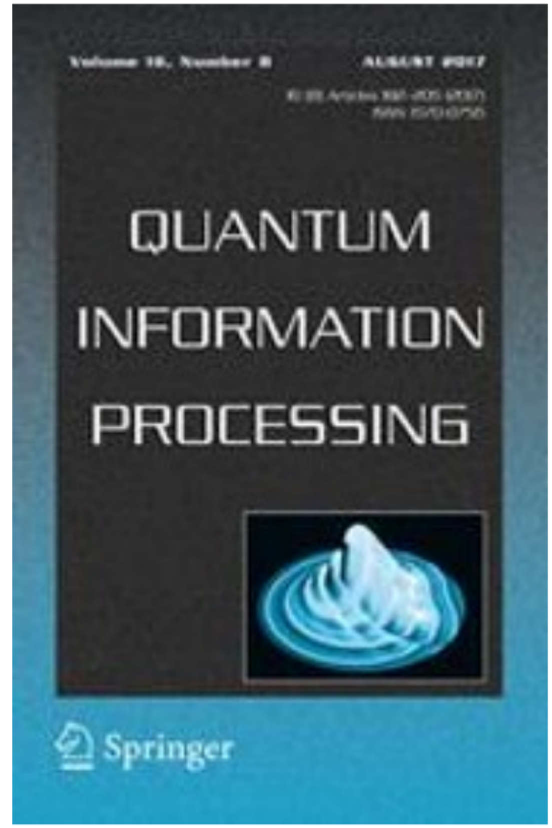

H
0


Controlled Teleportation on the IBM Quantum Computing Platform
Marcus Edwards and Dr. Shohini Ghose
Physics and Computer Science, Wilfrid Laurier University
Controlled Teleportation
Experimental Implementation
Results
Project Goals
Theoretical Procedure
Acknowledgements
References
IBM


Developing a Hybrid Methodology for Solving Quantum Problems
Abstract
Objectives
Environment
Simulators
Programming
Interface
Example Results
Acknowledgements
References

Journal Publication

M. Edwards, A. Mashatan, and S. Ghose.
"A review of quantum and hybrid quantum / classical blockchain protocols".
In: Quantum Information Processing 19.6 (2020).
DOI: 10.1007/s11128-020-02672-y.



Defense Overview

OPEN PROBLEMS

1. Point-to-Point Communication
2. Trusted Node Networks
3. Hybrid / Quantum Networks

LITERATURE REVIEW

1. Blockchain fundamentals
2. Quantum blockchains
3. Hybrid blockchains

METHODOLOGIES

1. Experimental Control Study
2. Experimental Technology Design
3. Theoretical Network Design

RESULTS

1. Experimental Study Results
2. Experimental Demonstration
3. Bounded Scalability

Motivation and Context

Fully Quantum Networks

Status: Basic Physics Research

Switching Quantum Networks

Trusted Quantum Satellite Networks

Status: Engineering and Design

Trusted Quantum Node Networks

Point-to-Point Quantum Communication

Status: Commercially Available

Norbert Lütkenhaus. “Lecture 22 - QKD Networks.” QIC 890 - Applied Quantum Cryptography (Nov. 2020), Waterloo ON, University of Waterloo.

Research Trajectory

Status: Presented in this Thesis!

- Hybrid Blockchain Networks
- Hybrid Quantum / Classical Networks

Status: Basic Physics Research

- Fully Quantum Networks
- Switching Quantum Networks

Open Questions

Hybrid / Quantum Networks	Multi-party applications (not reduceable to point-to-point) Anonymous channels Multi-party arbitrary function evaluation
Trusted Node Networks	Integration of QKD into security architecture Reduction of trust assumptions Authentication or proof of identity
Point-to-Point Communication	Tools to evaluate protocols Abilities to handle imperfections More applications

Norbert Lütkenhaus. “Lecture 22 - QKD Networks.” QIC 890 - Applied Quantum Cryptography (Nov. 2020), Waterloo ON, University of Waterloo.

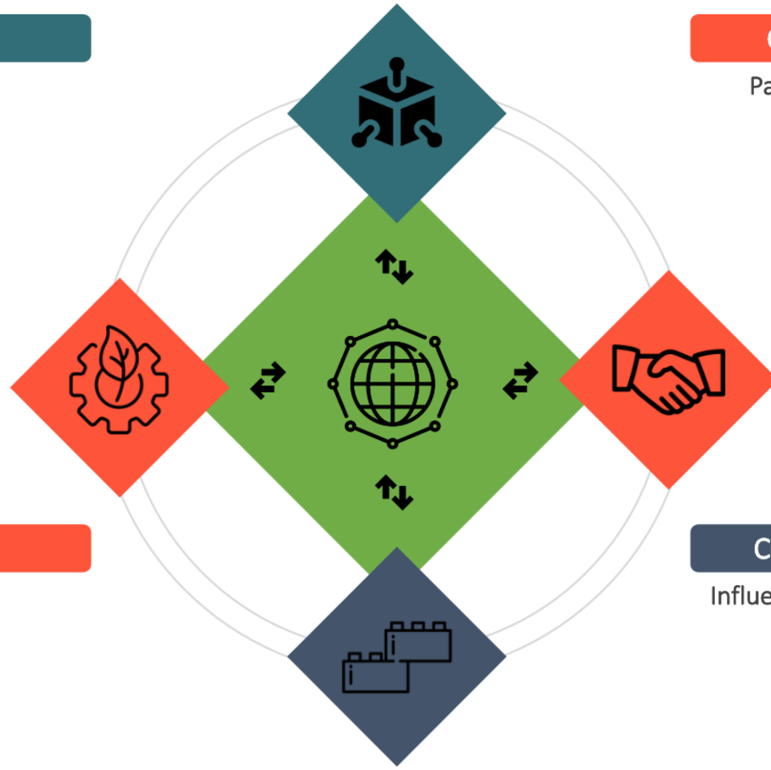
Practical Quantum / Classical Internet Goals

VIABLE TECHNOLOGY

Requires quantum control.

COMPLEMENTARY PARADIGMS

Paradigms need to be complementary.



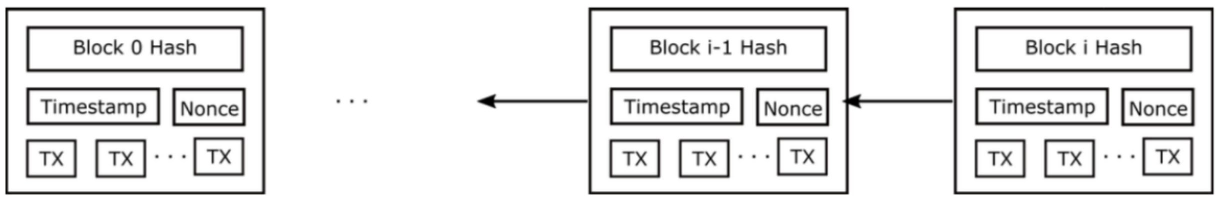
SUSTAINABLE ECOSYSTEM

New tooling and security.

CONSTRUCTIVE COLLABORATION

Influence between peers should be kept level.

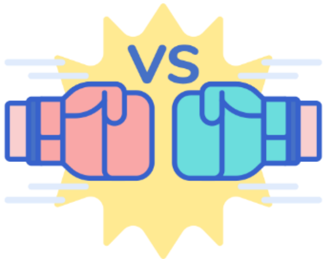
Blockchain – Data Structure

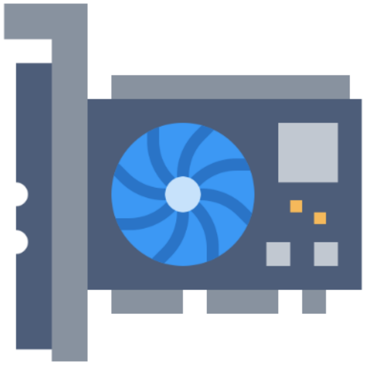


- TX = a transaction record
- Timestamp = the exact time of a block's publication
- Nonce = a unique, non-recurring identifier for a block
- Hash = the "proof of work" for the block

Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. url: https: //bitcoin.org/bitcoin.pdf.

Blockchain – Consensus Algorithms





Proof of Stake

Proof of Work

Dylan J. Yaga et al. Blockchain Technology Overview. Nov. 2018. url: https: //www.nist.gov/publications/blockchain-technology-overview.

Blockchain – Smart Contracts

Vitalik Buterin. Ethereum whitepaper. 2013. url: https : / / whitepaper . io / document/5/ethereum-whitepaper.

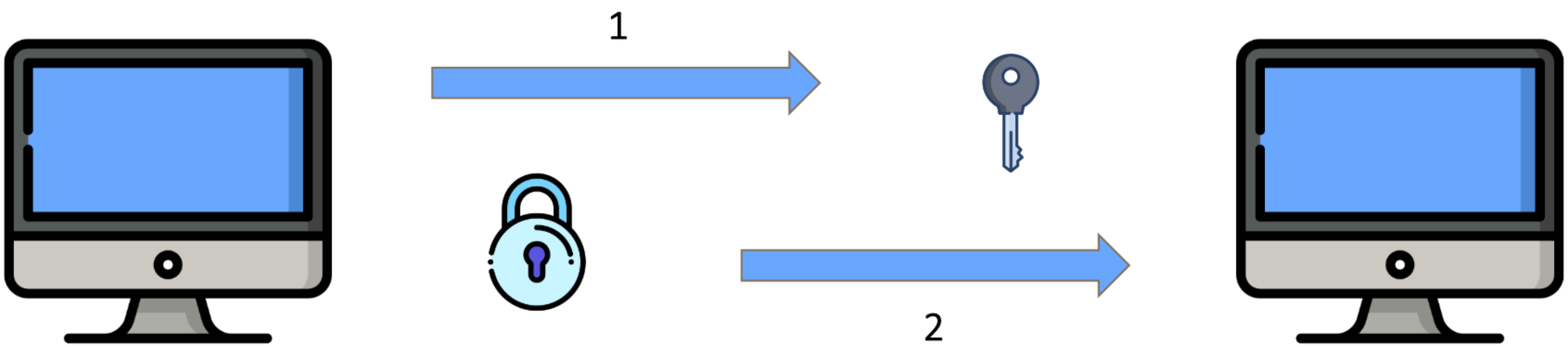
Quantum Blockchain

- Quantum coins

1. Generate two random bit strings M and N of length l
2. Prepare a quantum state $|\$ >= |0 >^{\otimes l}$
3. For each bit $i < l$:
   - If $M_i = 0$ and $N_i = 0$, do nothing to the i^{th} qubit
   - If $M_i = 0$ and $N_i = 1$, rotate the i^{th} qubit state to $|1 >$
   - If $M_i = 1$ and $N_i = 0$, rotate the i^{th} qubit state to $|+ >$
   - If $M_i = 1$ and $N_i = 1$, rotate the i^{th} qubit state to $|- >$

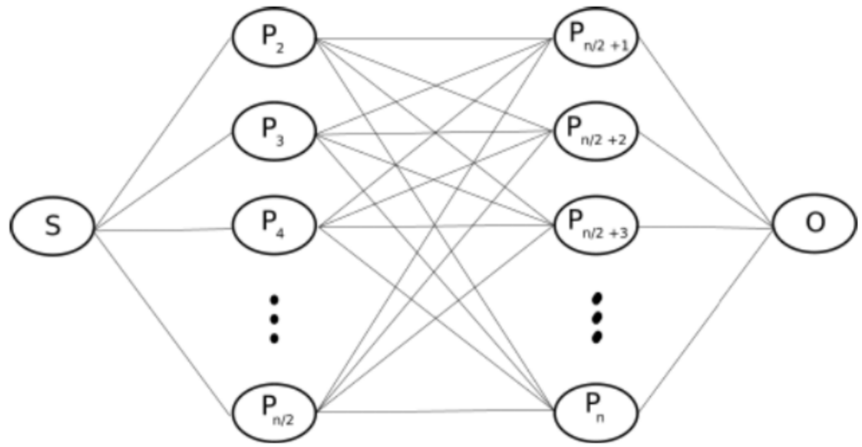
Stephen Wiesner. “Conjugate coding”. In: ACM SIGACT News 15.1 (1983), pp. 78–88. doi: 10.1145/1008908.1008920.

Quantum Binding Commitments



Dominique Unruh. “Collapse-Binding Quantum Commitments Without Random Oracles”. In: Advances in Cryptology - ASIACRYPT 2016 Lecture Notes in Computer Science (2016), pp. 166–195. doi: 10.1007/978-3-662- 53890-6_6.

Quantum Honest Byzantine Agreement



- S = Sender of vote
- P = other network participant
- n = number of participants
- L = verification lists distributed to all P
- b = Boolean vote value
- ID = indexes of b in L

Xin Sun et al. "Quantum-enhanced Logic-based Blockchain I: Quantum Honest- success Byzantine Agreement and Qulogicoin". In: arXiv e-prints, arXiv:1805.06768 (May 2018), arXiv:1805.06768. arXiv: 1805.06768 [quant-ph].

Collision Free Quantum Money

- equal superposition of exponentially many unrelated terms $|\$_l >= \frac{1}{\sqrt{N_l}} \sum_{x s.t. L(x)=l} |x>$
- Valid quantum money states $M^r \doteq \sum_l |\$_l >< \$_l|$ $M = \frac{1}{N}\sum_{i=1}^{N} P_i$
- Verification Kraus operators $(I \otimes \frac{1}{\sqrt{N}}\sum_{i=1}^{N} < i|)U(I \otimes \frac{1}{\sqrt{N}}\sum_{i=1}^{N} |i>) = \frac{1}{N}\sum_{i=1}^{N} P_i = M$
- Approximately verified state $\sum_l |\$_l >< \$_l|$

Andrew Lutomirski et al. “Breaking and making quantum money: toward a new quantum cryptographic protocol”. In: arXiv e-prints, arXiv:0912.3825 (Dec. 2009), arXiv:0912.3825. arXiv: 0912.3825 [quant-ph].

Quantum Lightning

Quantum money

Verification using a classical serial number

Verification does **not tamper** with money

Mark Zhandry. “Quantum Lightning Never Strikes the Same State Twice”. In: Advances in Cryptology - EUROCRYPT 2019 Lecture Notes in Computer Science (2019), pp. 408–438. doi: 10.1007/978-3-030-17659-4_14.

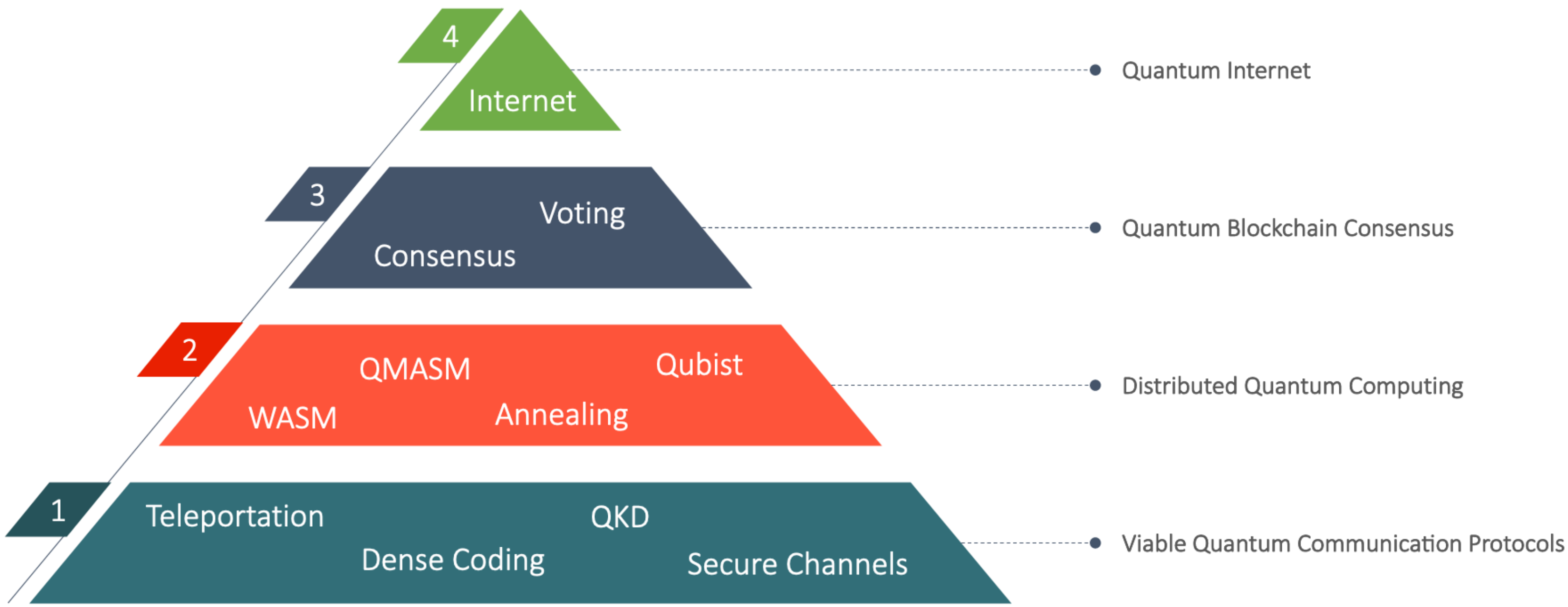
Hybrid Quantum Classical Blockchain

- Payments
  - P sends |$>, a contract id *cid,* and serial number *serial* to P'
  - P' sends a *Retrieve Contract* message to the ledger, retrieving the contract *cid.*
  - P' accepts the payment if *cid* and Verify(|$>) = *serial*

- Coin Recovery
  - *Trigger* message to cause a smart contract to execute a circuit *BanknoteLost*
  - P' has the chance to challenge for a fixed time by demonstrating ownership of the matching classical serial number
  - If not successfully challenged, P is given the new quantum coin

Andrea Coladangelo. "Smart contracts meet quantum cryptography". In: arXiv e-prints, arXiv:1902.05214 (Feb. 2019), arXiv:1902.05214. arXiv: 1902.05214 [quant-ph].

Complementary Research Areas



Effectiveness of Quantum Control for Networks

Algorithm Selection

- Looked at entanglement based communication
- Chose controlled quantum teleportation
- Chose controlled Dense Coding

Translation to Code

- Converted circuits to low-level specifications in OpenQASM
- Converted OpenQASM to Python programs
- Made compatible with IBM Q application programming interface

Experimentation

- Designed and implemented case-based fidelity measurement circuits
- Designed and implemented tests for the impact of the controller on result fidelity
- Automated batch testing for each algorithm and controller behaviour

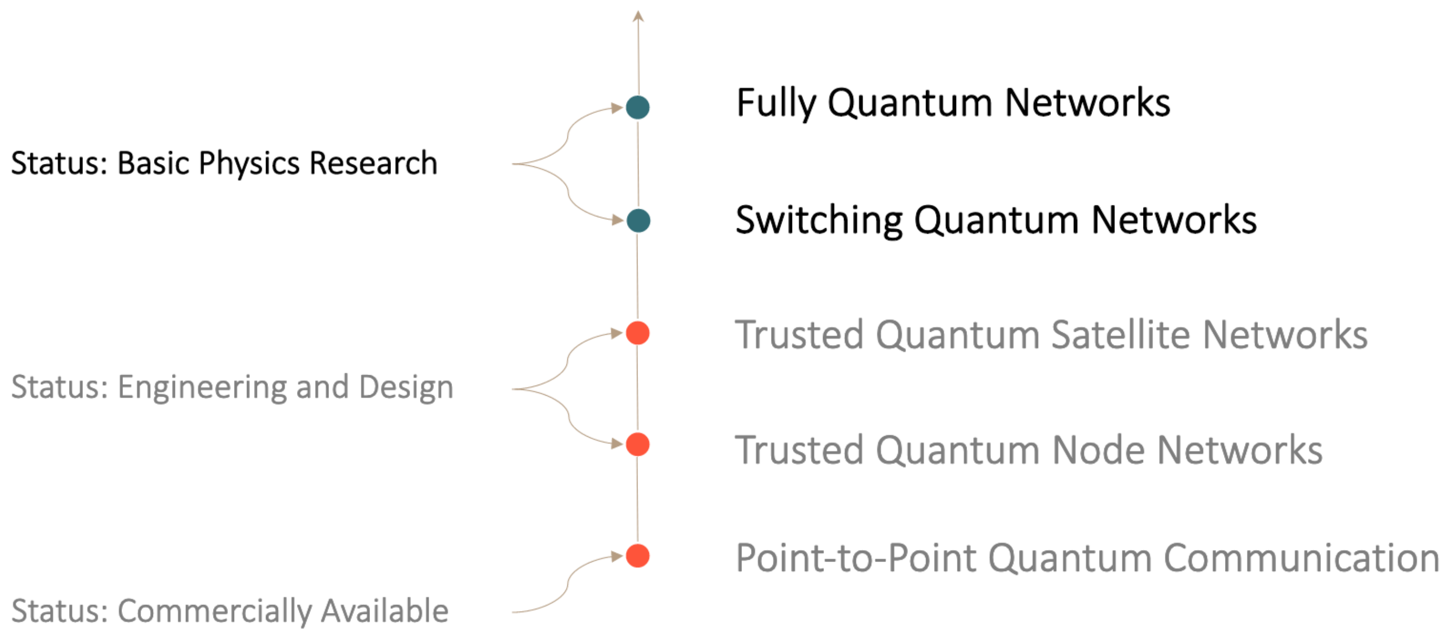
Results Analysis

- Collected statistical results from batches of results
- Performed analysis of statistical significance of the controller
- Performed state tomography to verify fidelity results

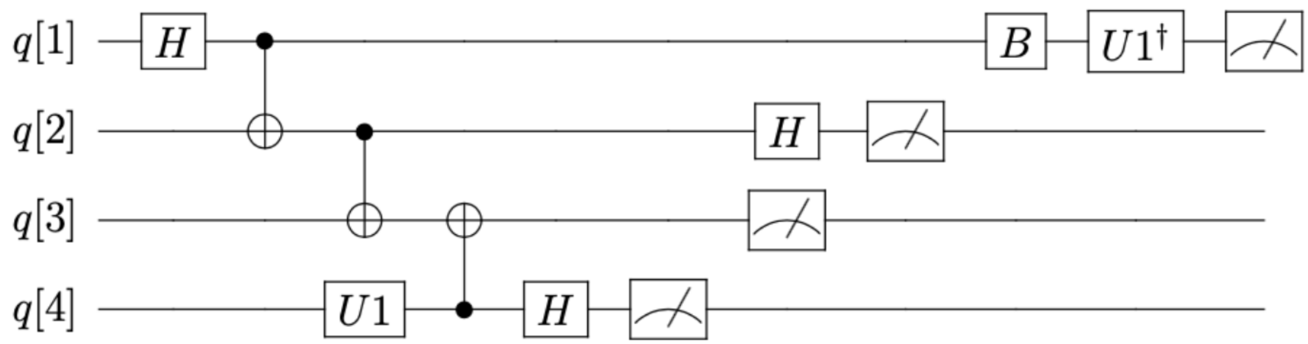
Open Questions Addressed

Hybrid / Quantum Networks	Multi-party applications (not reduceable to point-to-point) Anonymous channels Multi-party arbitrary function evaluation
Trusted Node Networks	Integration of QKD into security architecture Reduction of trust assumptions Authentication or proof of identity
Point-to-Point Communication	Tools to evaluate protocols Abilities to handle imperfections More applications

Motivation and Context



Controlled Teleportation



Xihan Li and Fuguo Deng. “Controlled teleportation”. In: Frontiers of Computer Science in China 2.2 (2008), pp. 147–160. doi: 10.1007/s11704-008-0020-0.

Bob’s Decoding Operations

Bell State	Charlie’s Result	Bob’s Operation
$\vert\phi^+ >_{\mathbf{xA}}$	$\vert+\mathbf{x}>$	I
$\vert\phi^+ >_{\mathbf{xA}}$	$\vert-\mathbf{x}>$	Z
$\vert\phi^- >_{\mathbf{xA}}$	$\vert+\mathbf{x}>$	Z
$\vert\phi^- >_{\mathbf{xA}}$	$\vert-\mathbf{x}>$	I
$\vert\psi^+ >_{\mathbf{xA}}$	$\vert+\mathbf{x}>$	X
$\vert\psi^+ >_{\mathbf{xA}}$	$\vert-\mathbf{x}>$	XZ
$\vert\psi^- >_{\mathbf{xA}}$	$\vert+\mathbf{x}>$	XZ
$\vert\psi^- >_{\mathbf{xA}}$	$\vert-\mathbf{x}>$	X

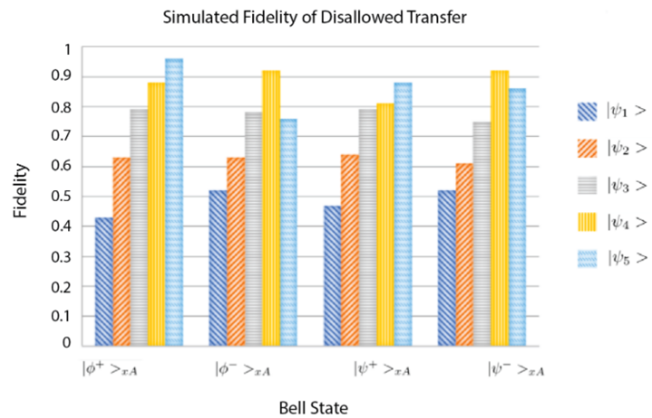
Xihan Li and Fuguo Deng. “Controlled teleportation”. In: Frontiers of Computer Science in China 2.2 (2008), pp. 147–160. doi: 10.1007/s11704-008-0020-0.

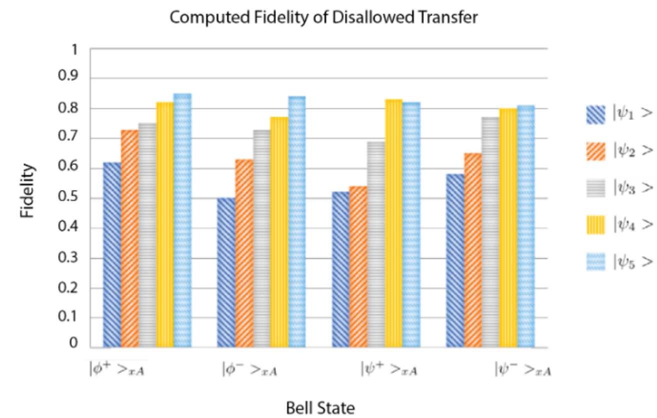
Experimental Methodology

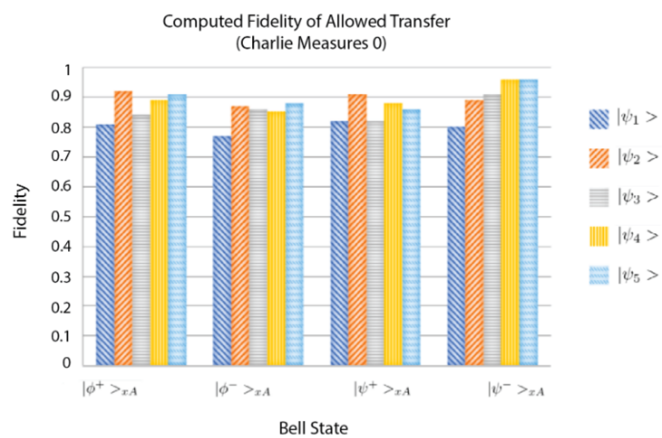
- Limitations
  1. Measurements taken during an execution cannot affect the gates that are applied as a part of the circuit.
  2. API returns a probability distribution of measurement results over a number of executions.
  3. Direct measurement results provide no insight since they can't discriminate between contexts.

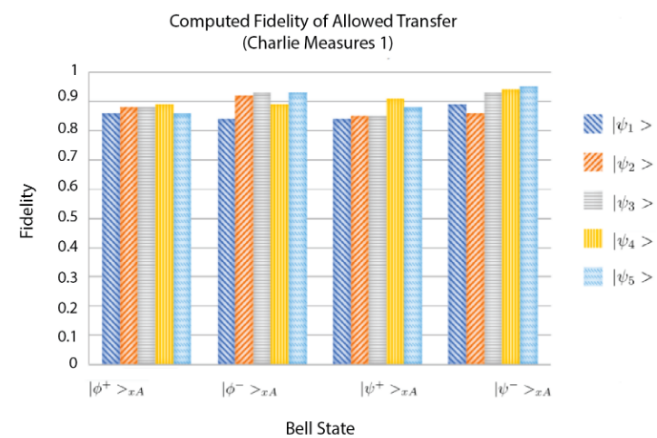
- Strategies
  1. A post-selection algorithm was implemented that filtered out bad choices by Bob.
  2. Each of the four circuits was run in a batch of 1000 executions each.
  3. The dagger of the input state was applied to qubit B at the end of the procedure.

Fidelities

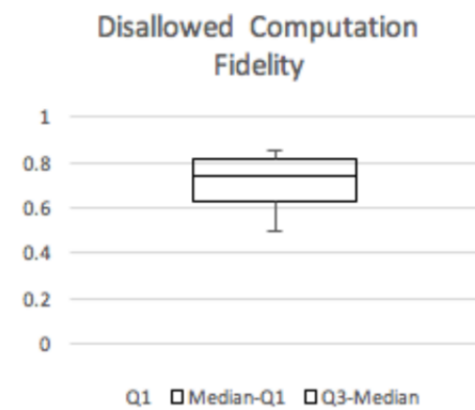

Simulated Fidelity of Disallowed Transfer
Fidelity
Bell State
$|\phi^+>_{xA}$ $|\phi^->_{xA}$ $|\psi^+>_{xA}$ $|\psi^->_{xA}$
$|\psi_1>$ $|\psi_2>$ $|\psi_3>$ $|\psi_4>$ $|\psi_5>$


Computed Fidelity of Disallowed Transfer
Fidelity
Bell State
$|\phi^+>_{xA}$ $|\phi^->_{xA}$ $|\psi^+>_{xA}$ $|\psi^->_{xA}$
$|\psi_1>$ $|\psi_2>$ $|\psi_3>$ $|\psi_4>$ $|\psi_5>$


Computed Fidelity of Allowed Transfer
(Charlie Measures 0)
Fidelity
Bell State
$|\phi^+>_{xA}$ $|\phi^->_{xA}$ $|\psi^+>_{xA}$ $|\psi^->_{xA}$
$|\psi_1>$ $|\psi_2>$ $|\psi_3>$ $|\psi_4>$ $|\psi_5>$

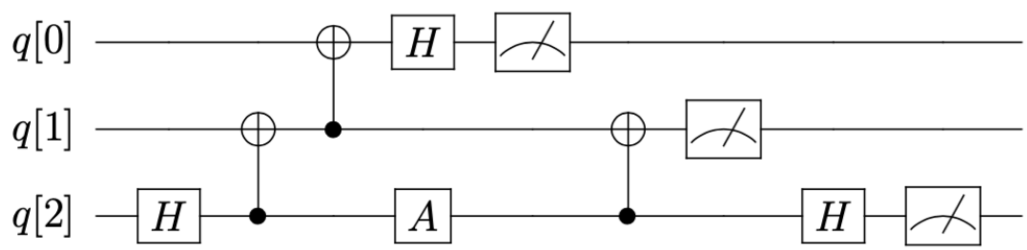

Computed Fidelity of Allowed Transfer
(Charlie Measures 1)
Fidelity
Bell State
$|\phi^+>_{xA}$ $|\phi^->_{xA}$ $|\psi^+>_{xA}$ $|\psi^->_{xA}$
$|\psi_1>$ $|\psi_2>$ $|\psi_3>$ $|\psi_4>$ $|\psi_5>$

Significance



$$\frac{\overline{F}_{allowed} - \overline{F}_{disallowed}}{max(Q3_{allowed}) - min(Q1_{disallowed})} = \frac{0.87 - 0.71}{0.92 - 0.62} = 57\%$$

Superdense Coding



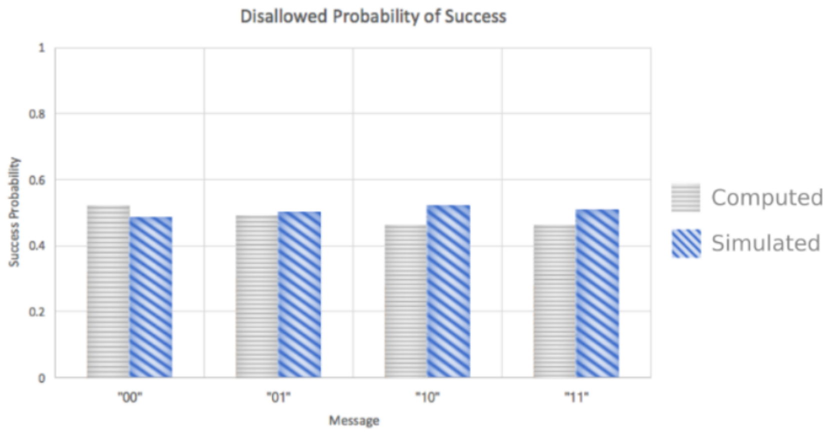
Jiu-Cang Hao, Chuan-Feng Li, and Guang-Can Guo. "Controlled dense coding using the Greenberger-Horne-Zeilinger state". In: Physical Review A 63.5 (Nov. 2001). doi: 10.1103/physreva.63.054301.

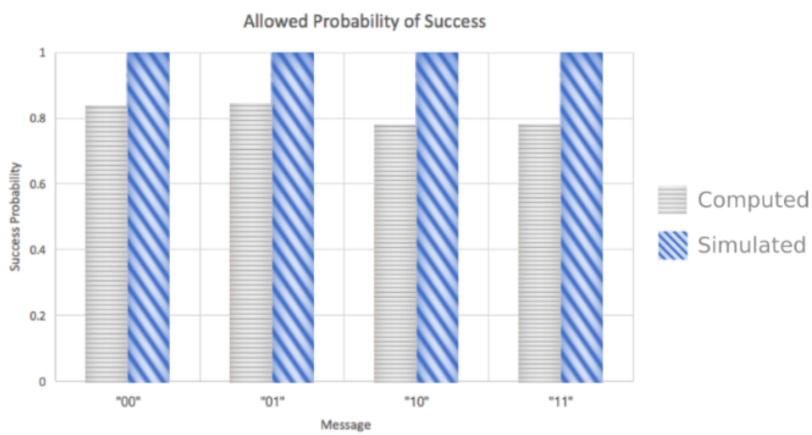
Data Encoding Operations

Message	Gates
00	I
01	Z
10	X
11	Y

Jiu-Cang Hao, Chuan-Feng Li, and Guang-Can Guo. "Controlled dense coding using the Greenberger-Horne-Zeilinger state". In: Physical Review A 63.5 (Nov. 2001). doi: 10.1103/physreva.63.054301.

Fidelities


Disallowed Probability of Success
1
0.8
0.6
0.4
0.2
0
Success Probability
"00"
"01"
"10"
"11"
Message
Computed
Simulated


Allowed Probability of Success
1
0.8
0.6
0.4
0.2
0
Success Probability
"00"
"01"
"10"
"11"
Message
Computed
Simulated

State Tomography Verification

- Relative phases

$$phase_j = \frac{2\pi j}{phases - 1}$$

- Axis rotations

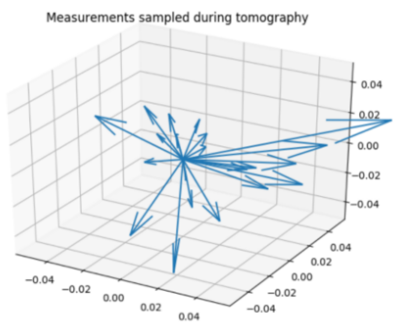
Axis Rotations

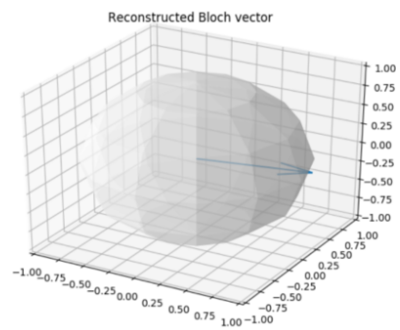
X-Axis	**Y-Axis**	**Z-Axis**
M H	**M H S†**	**M**

- Bloch vector reconstruction

```
# sum all observed vectors
x, y, z = zip(*bloch_vectors)
x = functools.reduce(lambda pre, curr: pre + curr, x)
y = functools.reduce(lambda pre, curr: pre + curr, y)
z = functools.reduce(lambda pre, curr: pre + curr, z)
```

- Results (Alice sends Z)





Shukla, Abhishek, et al. "Complete Characterization of the Directly Implementable Quantum Gates Used in the IBM Quantum Processors." 2018.

Hybrid Streaming Web Code Execution

Language Selection

- Looked for the most compatible and interoperable source and target programming languages
- Selected WebAssembly for the source language since it is a ubiquitous transpilation target
- Selected QMASM for the target language since it supports composable macros

Transpiler Design

- Designed source transformation with techniques from Fortran Parallelizing Compiler
- Designed compilation to Ising Hamiltonians using Quadratic Unconstranined Boolean Optimizations (QUBOs)
- Optimized compiler for minimal passes through source code

Implementation

- Implemented extension of Mozilla's Rust based WASM parser
- Integrated with PyQUBO for automatic conversion of generated QUBOs to Ising model parameters
- Exported Ising model parameters in tabular D-Wave Qubist compatible format

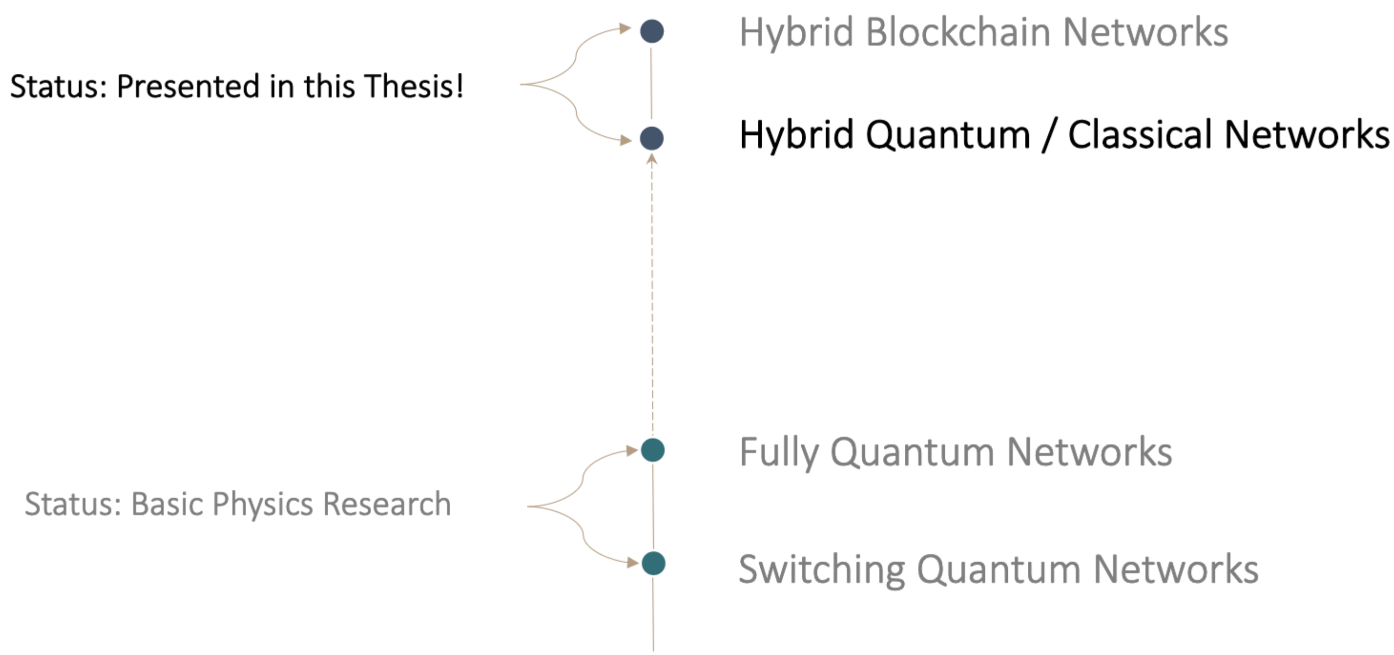
Demonstration

- Demonstrated basic WASM code parallelization
- Demonstrated Conversion of WASM vector multiplication program to QUBOs
- Demonstrated conversion of vector multiplication QUBO to tabular Ising model parameters

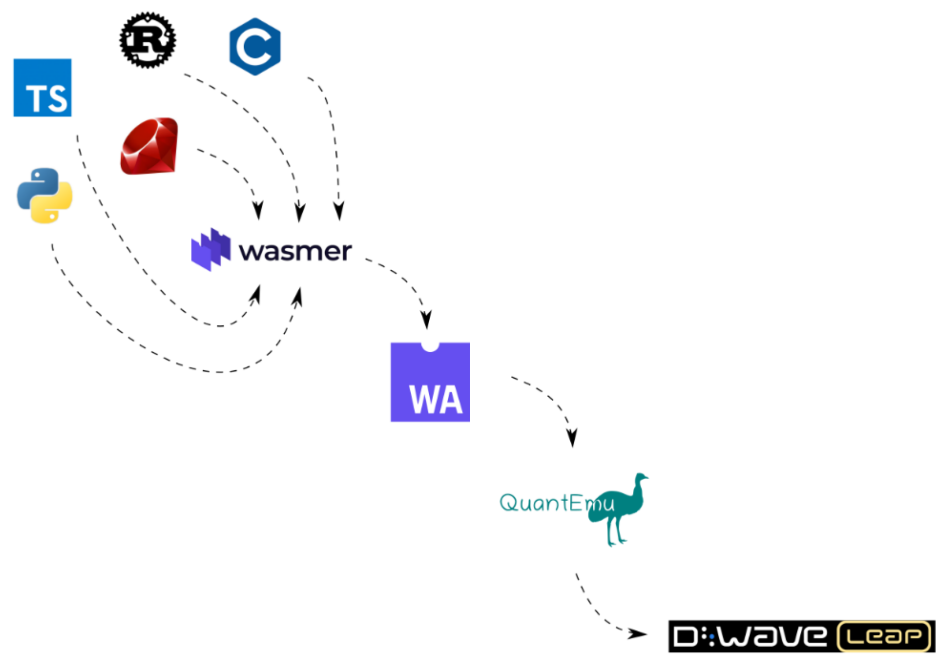
Open Questions Addressed

Hybrid / Quantum Networks	Multi-party applications (not reduceable to point-to-point) Anonymous channels Multi-party arbitrary function evaluation
Trusted Node Networks	Integration of QKD into security architecture Reduction of trust assumptions Authentication or proof of identity
Point-to-Point Communication	Tools to evaluate protocols Abilities to handle imperfections More applications

Motivation and Context



WebAssembly



WebAssembly Text Format

```
function accel(vi, vf, t){
    return (vi - vf)/t;
}
```

```
(func $accel (param $vi i32) (param $vf i32) (param $t i32) (result i32)
  (i32.div_u
    (i32.sub
      (get_local $vf)
      (get_local $vi)
    )
    (get_local $t)
  )
)
```

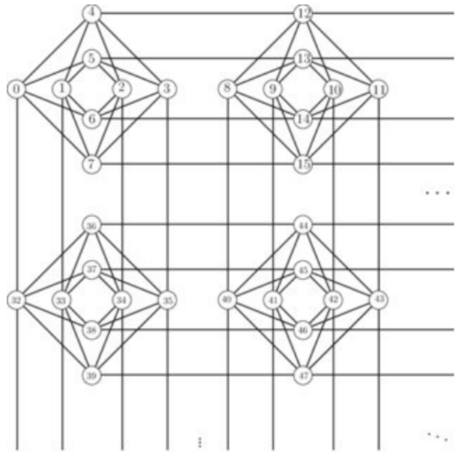
MDN Web Docs. Understanding WebAssembly text format. url: https : / / developer.mozilla.org/en-US/docs/WebAssembly/Understanding_the_text_ format#S-expressions.

Quantum Annealing

- D-Wave's annealers are capable of solving problems of a form

$$\mathbf{H}(\hat{\sigma}) = \sum_{i=0}^{N-1} h_i \sigma_i + \sum_{i=0}^{N-2} \sum_{j=i+1}^{N-1} J_{i,j} \sigma_i \sigma_j$$

- Qubit connectivity "Chimera" graph



D-Wave. url: https://www.dwavesys.com/p-ress-releases/d-wave-previ-ews- next-generation-quantum-computing-platform.

Quadratic Unconstrained Optimization

- A quadratic unconstrained Boolean expression $-x - z + 2xz$

- Unpenalized solution example $\mathbf{P} = 2xz - x - z + 1 = 2 \times 0 \times 1 - 0 - 1 + 1 = -1 + 1 = 0$

- Penalized solution example $\mathbf{P} = 2xz - x - z + 1 = 2 \times 0 \times 0 - 0 - 0 + 1 = 1$

D-Wave. Wave Ocean Software Documentation. url: http : / / docs . ocean . dwavesys.com/.

QMASM Macros

Cell	Logic	Quadratic pseudo-Boolean function representation
NOT	Y=$\neg A$	$\mathbb{H}_{\neg}(\sigma_\gamma, \sigma_A) = \sigma_A\sigma_\gamma$
AND	Y=A $\wedge B$	$\mathbb{H}_{\wedge}(\sigma_\gamma, \sigma_A, \sigma_B) = -\frac{1}{2}\sigma_A - \frac{1}{2}\sigma_B + \sigma_\gamma + \frac{1}{2}\sigma_A\sigma_B - \sigma_A\sigma_\gamma - \sigma_B\sigma_\gamma$
OR	Y=A $\vee B$	$\mathbb{H}_{\vee}(\sigma_\gamma, \sigma_A, \sigma_B) = -\frac{1}{2}\sigma_A - \frac{1}{2}\sigma_B - \sigma_\gamma + \frac{1}{2}\sigma_A\sigma_B - \sigma_A\sigma_\gamma - \sigma_B\sigma_\gamma$
NAND	Y=A↑B	$\mathbb{H}_{\uparrow}(\sigma_\gamma, \sigma_A, \sigma_B) = -\frac{1}{2}\sigma_A - \frac{1}{2}\sigma_B - \sigma_\gamma + \frac{1}{2}\sigma_A\sigma_B + \sigma_A\sigma_\gamma + \sigma_B\sigma_\gamma$
NOR	Y=A↓B	$\mathbb{H}_{\downarrow}(\sigma_\gamma, \sigma_A, \sigma_B) = \frac{1}{2}\sigma_A + \frac{1}{2}\sigma_B + \sigma_\gamma + \frac{1}{2}\sigma_A\sigma_B + \sigma_A\sigma_\gamma + \sigma_B\sigma_\gamma$
XOR	Y=A $\oplus B$	$\mathbb{H}_{\oplus}(\sigma_\gamma, \sigma_A, \sigma_B, \sigma_a) = \frac{1}{2}\sigma_A - \frac{1}{2}\sigma_B - \frac{1}{2}\sigma_\gamma + \sigma_a - \frac{1}{2}\sigma_A\sigma_B$ $-\frac{1}{2}\sigma_A\sigma_\gamma + \sigma_A\sigma_a + \frac{1}{2}\sigma_B\sigma_\gamma - \sigma_B\sigma_a - \sigma_\gamma\sigma_a$
XNOR	Y=A $\Leftrightarrow B$	$\mathbb{H}_{\Leftrightarrow}(\sigma_\gamma, \sigma_A, \sigma_B, \sigma_a) = \frac{1}{2}\sigma_A - \frac{1}{2}\sigma_B + \frac{1}{2}\sigma_\gamma + \sigma_a - \frac{1}{2}\sigma_A\sigma_B$ $+\frac{1}{2}\sigma_A\sigma_\gamma + \sigma_A\sigma_a - \frac{1}{2}\sigma_B\sigma_\gamma - \sigma_B\sigma_a + \sigma_\gamma\sigma_a$
2:1 MUX	Y $= (\mathrm{S} \wedge B) \vee (\neg S \wedge A)$	$H_{MUX}(\sigma_\gamma, \sigma_S, \sigma_A, \sigma_B, \sigma_a) = \frac{1}{2}\sigma_S + \frac{1}{4}\sigma_A - \frac{1}{4}\sigma_B$ $+\frac{1}{2}\sigma_\gamma + \sigma_a + \frac{1}{4}\sigma_S\sigma_A - \frac{1}{4}\sigma_S\sigma_B + \frac{1}{2}\sigma_S\sigma_\gamma + \sigma_S\sigma_a$ $+\frac{1}{2}\sigma_A\sigma_B - \frac{1}{2}\sigma_A\sigma_\gamma + \frac{1}{2}\sigma_A\sigma_a - \sigma_B\sigma_\gamma - \frac{1}{2}\sigma_B\sigma_a + \sigma_\gamma\sigma_a$

Scott Pakin. "A quantum macro assembler". In: 2016 IEEE High Performance Extreme Computing Conference (HPEC) (2016). doi: 10.1109/hpec.2016. 7761637.

QMASM TypeScript

QMASM, Los Alamos National Laboratory's macro assembler for D-wave's quantum annealer, implemented in TypeScript.

Language documentation is provided by Scott Pakin **here**.

New in Version 1.0.0

- Support for the following QMASM features:
  - Specifying 2-local Ising Hamiltonian parameters via: qubit weights, coupling strengths
  - Relating qubits via: chains, anti-chains, equivalences
  - Relating qubits to classical values via pins
  - Importable and parameterizable macro system via: !begin_macro, include, !use_macro
  - Macro chaining via: !next
  - Assertions
  - Logical and mathematical expressions involving the following elements: +, -, *, /, **, =, /=, ||, |, &&, &, ~, !, ^, %, <, >, <<, >>, <=, >=
  - For loops
  - If/else conditionals
  - Support for the following classical types: Iterator, Range, Int, Float, Bool
  - Support for the following quantum data types: Qubit, Ancilliary, QubitArray, Register

Install

```
> npm i qmasm-ts
```

Weekly Downloads

16

Version

1.0.5

License

BSD-3-Clause-Att…

Unpacked Size

88.6 kB

Total Files

11

Issues

0

Pull Requests

0

Homepage

github.com/MackEdweise/qmasm-ts#re…

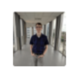
Repository

github.com/MackEdweise/qmasm-ts

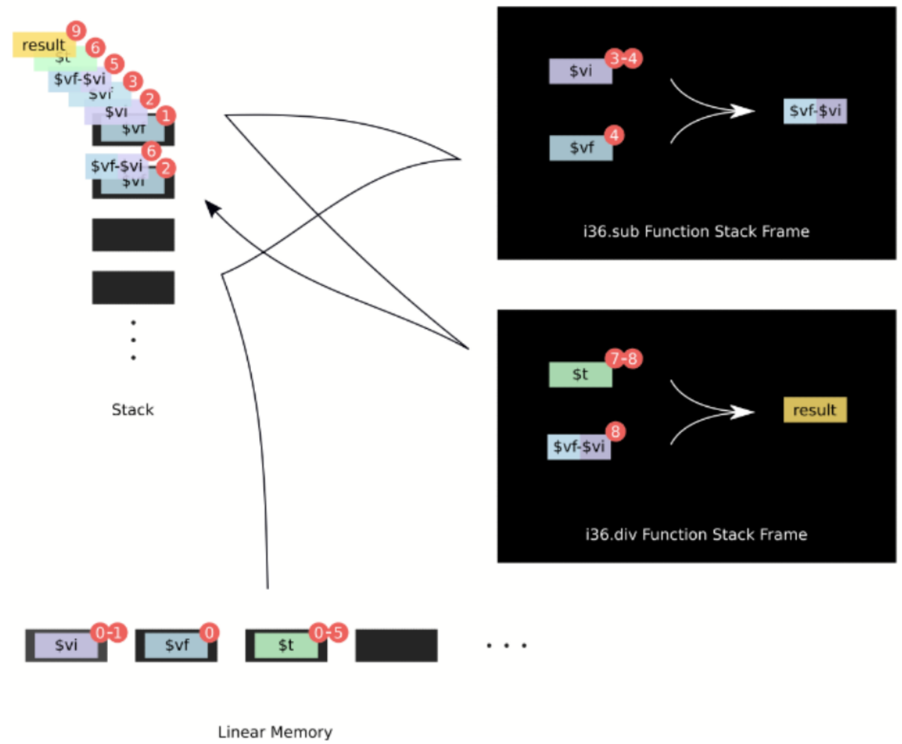
Last publish

7 months ago

Collaborators



Dependency Tracing



Interactive Transpiling

```
BeginWasm { version: 1 }
BeginSection { code: Type, range: Range { start: 10, end: 29 } }
TypeSectionEntry(FuncType { form: Func, params: [I32, I32, I32, I32, I32, I32], returns: [I32] })
TypeSectionEntry(FuncType { form: Func, params: [I32, I32, I32, I32], returns: [I32] })
EndSection
BeginSection { code: Function, range: Range { start: 31, end: 34 } }
EndSection
BeginSection { code: Export, range: Range { start: 36, end: 59 } }
ExportSectionEntry { field: "dot_three", kind: Function, index: 0 }
ExportSectionEntry { field: "dot_two", kind: Function, index: 1 }
EndSection
BeginSection { code: Code, range: Range { start: 61, end: 95 } }
BeginFunctionBody { range: Range { start: 63, end: 81 } }
1. Unreachable
2. GetLocal { local_index: 0 }
3. GetLocal { local_index: 3 }
4. I32Mul
5. GetLocal { local_index: 1 }
6. GetLocal { local_index: 2 }
7. GetLocal { local_index: 4 }
8. GetLocal { local_index: 5 }
9. Call { function_index: 1 }
10. I32Add
11. End
BeginFunctionBody { range: Range { start: 82, end: 95 } }
1. Unreachable
2. GetLocal { local_index: 0 }
3. GetLocal { local_index: 2 }
4. I32Mul
5. GetLocal { local_index: 1 }
6. GetLocal { local_index: 3 }
7. I32Mul
8. I32Add
9. End
EndSection
First pass found 2 functions:
[0, 1]
```

```
Parallelize function 1 (yes/no)?
y
Analyzing function 1...
Found 0 blocks in function 1
Found 0 calls to other functions from function 1
Parallelize function 0 (yes/no)?
y
Analyzing function 0...
Found 0 blocks in function 0
Found 1 calls to other functions from function 0
Registering call to function 1 from function 0
Found 0 blocks in function 1
Found 0 calls to other functions from function 1
```

Lowering Expressions to Constraints

```
/// The physical expression enum represents the valid
/// operations and data types that can be understood by PyQUBO.
#[derive(Clone, Debug)]
pub enum PhysicalExpression {
    Not{ operand: Box<PhysicalExpression> },
    Add{ operand_one: Box<PhysicalExpression>, operand_two:
        Box<PhysicalExpression> },
    Mul{ operand_one: Box<PhysicalExpression>, operand_two:
        Box<PhysicalExpression> },
    Spin{ val: bool }, // 0 represents -1
    Num{ val: usize },
    Binary{ val: bool }
}
```

```
/// The abstract operation enum represents logical operations
/// that can be compiled to simulatable transfer functions
/// for quantum annealers.
#[derive(Clone, Debug)]
pub enum AbstractExpression {
    Spin { id: usize },
    Num { val: usize },
    Add { ty: Type },
    Mul { ty: Type }
}
```

8-bit Multiplier Circuit - Constraints

```
(Spin("A5") * Spin("B5"))

(Not(Spin("A4") * Spin("B5")) + Not(Spin("A5") * Spin("B4")))

(Not(Spin("A3") * Spin("B5")) + Spin("A4") * Spin("B4") + Not(Spin("A5") *
    Spin("B3")))

(Not(Spin("A2") * Spin("B5")) + Spin("A3") * Spin("B4") + Spin("A4") * Spin("B3")
    + Not(Spin("A5") * Spin("B2")))

(Num(1) + Not(Spin("A1") * Spin("B5")) + Spin("A2") * Spin("B4") + Spin("A3") *
    Spin("B3") + Spin("A4") * Spin("B2") + Not(Spin("A5") * Spin("B1")))

(Not(Spin("A0") * Spin("B5")) + Spin("A1") * Spin("B4") + Spin("A2") * Spin("B3")
    + Spin("A3") * Spin("B2") + Spin("A4") * Spin("B1") + Not(Spin("A5") *
    Spin("B0")))

(Spin("A0") * Spin("B4") + Spin("A1") * Spin("B3") + Spin("A2") * Spin("B2") +
    Spin("A3") * Spin("B1") + Spin("A4") * Spin("B0"))

(Spin("A0") * Spin("B4") + Spin("A1") * Spin("B3") + Spin("A2") * Spin("B2") +
    Spin("A3") * Spin("B1") + Spin("A4") * Spin("B0"))

(Spin("A0") * Spin("B3") + Spin("A1") * Spin("B2") + Spin("A2") * Spin("B1") +
    Spin("A3") * Spin("B0"))

(Spin("A0") * Spin("B2") + Spin("A1") * Spin("B1") + Spin("A2") * Spin("B0"))

(Spin("A0") * Spin("B1") + Spin("A1") * Spin("B0"))

(Spin("A0") * Spin("B0"))
```

8-bit Multiplier Circuit – Ising Parameters

```
({('A2', 'A2'): -6.0
('A0*B2', 'A0*B2'): 27.0
('A0', 'A0*B2'): -10.0
('A1*B1', 'B0'): -16.0
('A1', 'B2'): 8.0,
('A0', 'B0'): 8.0
('B2', 'B2'): -6.0
('A1*B1', 'B1'): -10.0
('A2', 'A2*B0'): -10.0
('A0*B2', 'B1'): -16.0,
('A1', 'B1'): 5.0
('A2*B0', 'A2*B0'): 27.0
('A1', 'A2'): 8.0
('B1', 'B2'): 8.0
('A0*B2', 'A1*B1'): 32.0,
('A1*B1', 'A2'): -16.0
('A1', 'A2*B0'): -16.0
('A1', 'A1'): -6.0
('A0', 'A2'): 8.0
('A0', 'A0'): -6.0,
('A2*B0', 'B2'): -16.0
('A1', 'A1*B1'): -10.0
('A0', 'B1'): 8.0
('A1*B1', 'B2'): -16.0
('A2', 'B1'): 8.0,
('A0*B2', 'A2*B0'): 32.0
('A2*B0', 'B1'): -16.0
('B0', 'B2'): 8.0
('A2*B0', 'B0'): -10.0
('B1', 'B1'): -6.0,
('A0*B2', 'A1'): -16.0
('A0', 'A1'): 8.0
('A1*B1', 'A2*B0'): 32.0
('A0*B2', 'A2'): -16.0
('A0', 'A2*B0'): -16.0,
('A2', 'B0'): 5.0
('A0', 'B2'): 5.0
('A0*B2', 'B2'): -10.0
('B0', 'B1'): 8.0
('A1*B1', 'A1*B1'): 27.0,
('A0', 'A1*B1'): -16.0
('A2', 'B2'): 8.0
('A1', 'B0'): 8.0
('A0*B2', 'B0'): -16.0
('B0', 'B0'): -6.0}, 7.0)
```

A Scalable Hybrid Consensus Network Architecture

Problem Selection

- Chose to address the complexity of scalability issue with p2p consensus
- Chose to address the issue of fairness within blockchain
- Chose to address the issue of dishonest collusion within blockchain networks

Algorithm Formulation

- Designed a voting scheme that benefits from a quadratic quantum speedup
- Designed for a sum-of-squares distribution of influence like that used in quadratic voting
- Based consensus on a novel combination of distributed QML and QHBA

Architecture Design

- Designed to be demonstrable now with limited quantum resources
- Designed to gradually incorporate powerful quantum resources fairly
- Designed for a hybrid cloud environment

Bounding

- Determined order of quantum speedup
- Determined scaling limitations based on current quantum device characteristics
- Determined scaling limitations based on future projections for quantum devices

Open Questions Addressed

Hybrid / Quantum Networks	Multi-party applications (not reduceable to point-to-point) Anonymous channels Multi-party arbitrary function evaluation
Trusted Node Networks	Integration of QKD into security architecture Reduction of trust assumptions Authentication or proof of identity
Point-to-Point Communication	Tools to evaluate protocols Abilities to handle imperfections More applications

Research Trajectory

Hybrid Blockchain Networks

Status: Presented in this Thesis!

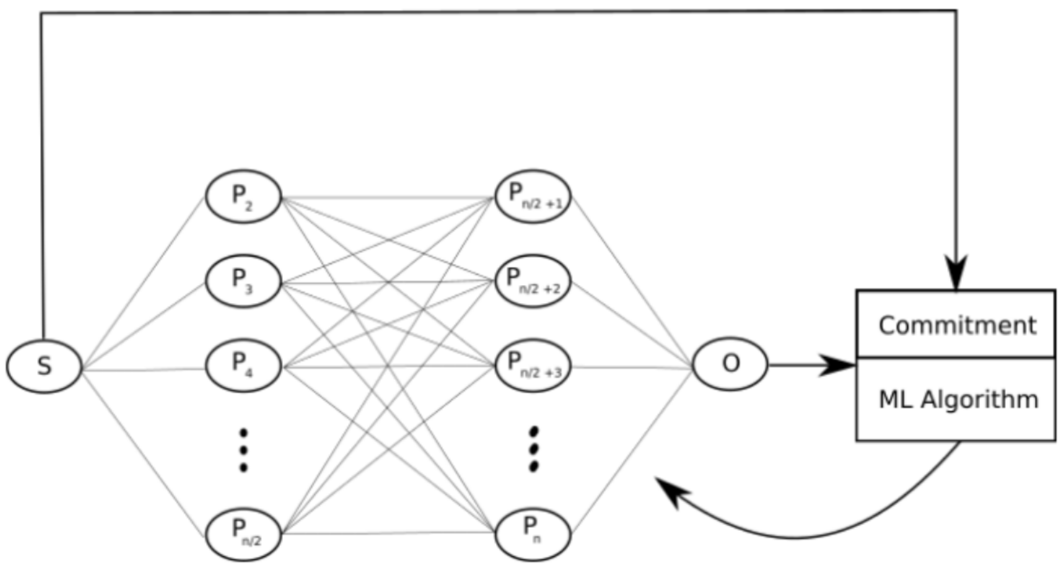
Hybrid Quantum / Classical Networks

Fully Quantum Networks

Status: Basic Physics Research

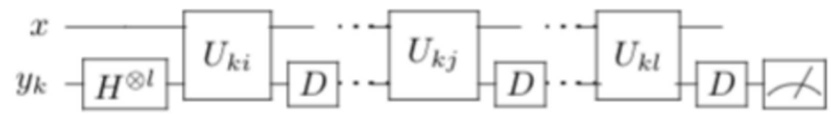
Switching Quantum Networks

Augmented QHBA

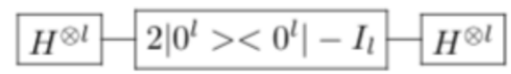


Associative Measuring Neurons

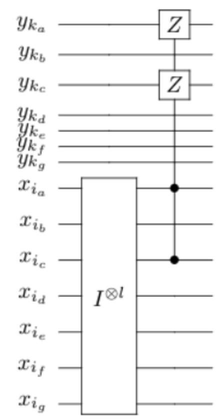
- High-dimensional threshold classifier

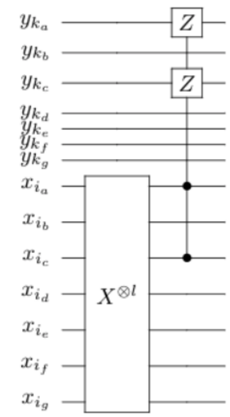


- Uses competing Grover's searches with diffusion op D



- Uses classically parameterized oracles Uki

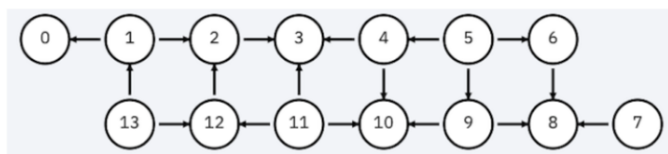




Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC 96 (1996). doi: 10.1145/237814.237866.

IBM Q Melbourne

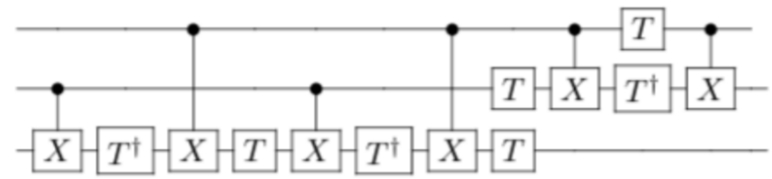
qubit	multi_qb_gate_error	T1 (us)	T2 (us)	Frequency (GHz)	readout_error	gate_error
Q0		73.32348273	23.48828043	5.100090141	0.0215	0.004031062
Q1	CX1_0: 0.03, CX1_2: 0.04	63.23181621	116.7289054	5.238609742	0.054	0.012242205
Q2	CX2_3: 0.04	46.13953307	74.56571753	5.032644087	0.1864	0.010450744
Q3		81.05055849	74.78940464	4.896205701	0.047	0.002494886
Q4	CX4_3: 0.03, CX4_10: 0.04	55.43102145	27.63898146	5.028667392	0.1226	0.002551687
Q5	CX5_4: 0.05, CX5_6: 0.05, CX5_9: 0.07	27.79450766	50.71953989	5.06718735	0.0568	0.004714312
Q6	CX6_8: 0.04	56.16840169	56.0630866	4.923906934	0.0478	0.004816689
Q7	CX7_8: 0.03	32.50641909	45.28966051	4.974534967	0.0598	0.004438222
Q8		47.68062524	71.45643335	4.739563654	0.0389	0.004361702
Q9	CX9_8: 0.04, CX9_10: 0.05	38.43726664	79.71232612	4.963421912	0.0443	0.006372041
Q10		56.99362705	69.83941723	4.945065458	0.037	0.003278348
Q11	CX11_3: 0.05, CX11_10: 0.05, CX11_12: 0.06	57.53451171	71.43323367	5.004981691	0.0357	0.0044898
Q12	CX12_2: 0.06	78.13277541	117.4664528	4.760047973	0.0918	0.007732648
Q13	CX13_1: 0.12, CX13_12: 0.1	21.39891833	41.28178002	4.968495889	0.0498	0.011006778



CX Gate	GF Gate Time (ns)
CX1_0	239
CX1_2	174
CX2_3	261
CX4_3	266
CX5_4	300
CX5_6	300
CX7_8	220
CX9_8	434
CX9_10	300
CX11_10	261
CX11_12	261
CX13_12	300
CX13_1	652
CX12_2	1043
CX11_3	286
CX4_10	261
CX5_9	348
CX6_8	348

Circuit Compilation for IBM Q

- CZ gate compilation



- Available single- and double-param 1-qubit gates

$$u1(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{\lambda i} \end{bmatrix} \qquad u2(\phi, \lambda) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -e^{\lambda i} \\ e^{\phi i} & e^{(\phi i + \lambda i)} \end{bmatrix}$$

- T and T† compilation

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{\pi}{4} i} \end{bmatrix} = u1(\frac{\pi}{4}) \qquad T^{\dagger} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{\pi}{4} i} \end{bmatrix} = u1(-\frac{\pi}{4})$$

- Hadamard compilation

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = u2(2\pi, 3\pi)$$

Vivek V. Shende and Igor L. Markov. “On the CNOT-cost of TOFFOLI gates”. In: arXiv e-prints, arXiv:0803.2316 (Mar. 2008), arXiv:0803.2316. arXiv: 0803. 2316 [quant-ph].

Time Penalties

- Penalty for Uki $$T(U_{ki}) \doteq 2 \cdot 0ns + 6 \cdot 350ns = 2100ns$$
- Penalty for D $$T(D) \doteq 2 \cdot 0ns + 4 \cdot 70ns + 1 \cdot 350ns = 640ns$$
- Penalty per repetition $$T_{rep} = T(U_{ki}) + T(D) = 2740ns$$
- Associative measuring neuron penalty $$T_{assoc} = \sum_{i} \lfloor b_k + N_i \rfloor \cdot 2740ns$$
- Bounded neuron penalty < T2 $$T2 = max(T_{assoc} | \, |P|) = \sum_{i=0}^{\frac{|P|}{2}} \lfloor b_k + max(N_i) \rfloor \cdot 2740ns$$

Scaling in Network Nodes

$$22.40\mu s = \sum_{i=0}^{\frac{|P|}{2}} \lfloor b_k + max(N_i) \rfloor \cdot 2740ns$$

$$\frac{22.40\mu s}{2740ns} = \sum_{i=0}^{\frac{|P|}{2}} \lfloor b_k + max(N_i) \rfloor$$

$$\frac{22.40\mu s}{2740ns} - \sum_{i=0}^{\frac{|P|}{2}} b_k \doteq \sum_{i=0}^{\frac{|P|}{2}} max(N_i)$$

$$\frac{22.40\mu s}{2740ns} - \sum_{i=0}^{\frac{|P|}{2}} b_k \doteq \frac{|P|}{2} max(N_i)$$

$$max(N_i|\ |P|) \doteq \frac{2}{|P|} \cdot \frac{22.40\mu s}{2740ns} - \frac{2}{|P|} \cdot \sum_{i=0}^{\frac{|P|}{2}} b_k$$

In the worst case, $\sum_{i=0}^{\frac{|P|}{2}} b_k \to \frac{|P|}{2}$ since $0 \leq b_k \leq 1$.

$$max(N_i|\ |P|) + 1 \doteq \frac{2}{|P|} \cdot \frac{22.40\mu s}{2740ns}$$

$$\frac{2}{|P|} \cdot \frac{22.40\mu s}{2740ns} \geq 1$$

Recap: Open Questions Addressed

Hybrid / Quantum Networks	Multi-party applications (not reduceable to point-to-point) Anonymous channels Multi-party arbitrary function evaluation
Trusted Node Networks	Integration of QKD into security architecture Reduction of trust assumptions Authentication or proof of identity
Point-to-Point Communication	Tools to evaluate protocols Abilities to handle imperfections More applications

Discussion

In each case we demonstrated that different aspects of the quantum internet are useable in a limited sense today, and bounded this usefulness.

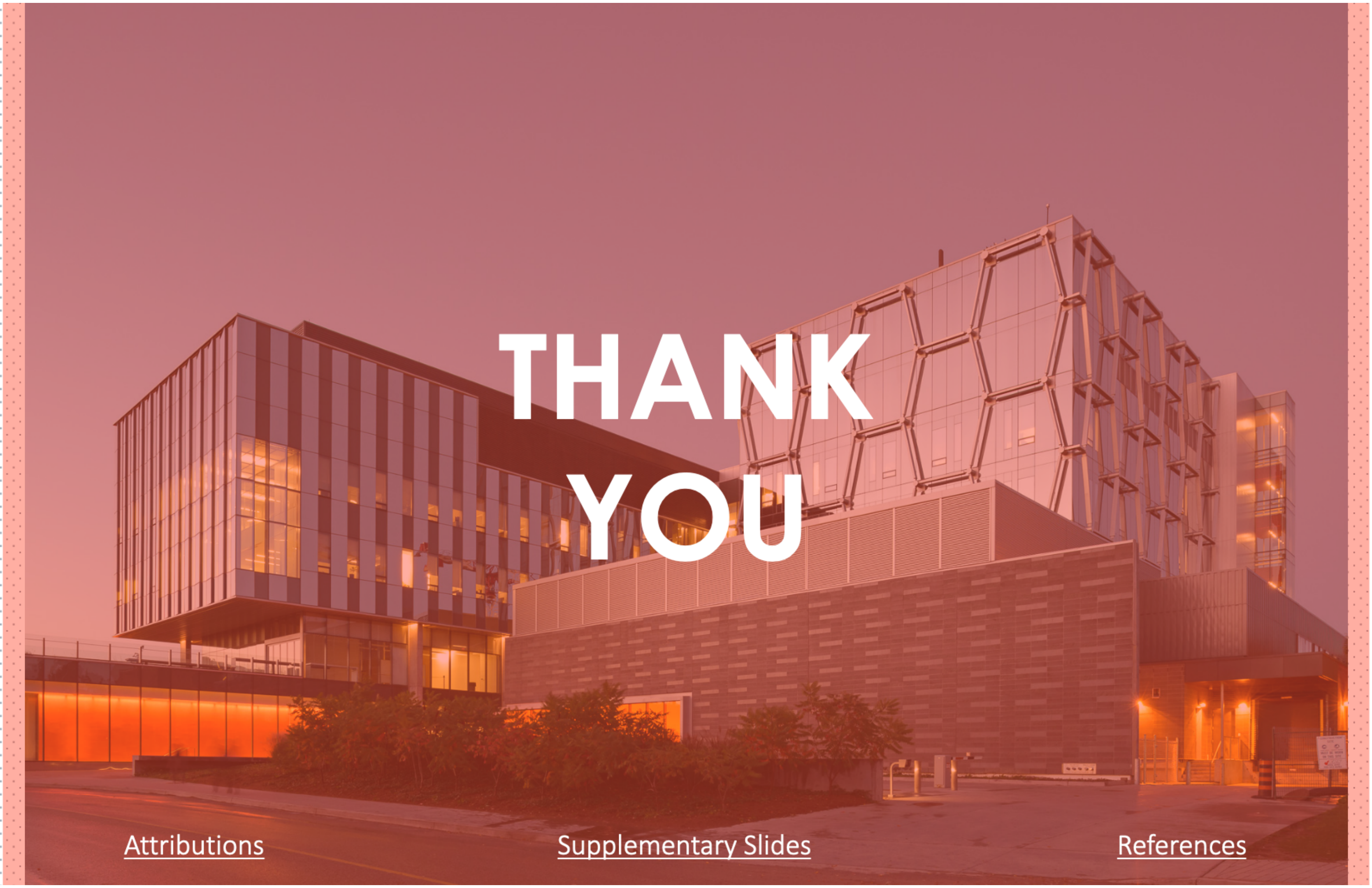
We also provided several novel algorithms and technologies that aim to accelerate the internet through a fair, non-wasteful and stable transition into the quantum age.

It remains to be seen if IBM will meet with their projected success of a quantum “Moore’s law”.

There are currently no known degree-2 hash functions that are proven to be non-collapsing.

It remains for someone to develop a true testing framework on top of our WASM transpiler.

Blockchain technology may be yet to stand the test of time.


THANK
YOU
Attributions
Supplementary Slides
References

Background / Literature Review

Quantum Physics

- Quantum fundamentals
- Energy Distributions
- Schrodinger’s Equation

Quantum Computing

- The Qubit model
- Quantum circuits
- Useful operators

Quantum Blockchain

- Blockchain fundamentals
- Quantum blockchains
- Hybrid blockchains

Quantum Information

- Energy distributions or “complexes” $$R = \frac{(N+P)^{N+P}}{N^N \cdot P^P}$$
- Entropy of a quantum system $$S_N = klog(R) = k\{(N+P)log(N+P) - NlogN - PlogP\}$$
- Entropy of a particle $$S = k\{(1+\frac{U}{\epsilon})log(1+\frac{U}{\epsilon}) - \frac{U}{\epsilon}log\frac{U}{\epsilon}\}$$
- Quantum Harmonic Oscillator $$E_n = \hbar\omega(n+\frac{1}{2}) = (2n+1)\frac{\hbar}{2}\omega$$
- ”Complexes" become "quantum states" ψr $$\rho = \sum_{r=0}^{R} \alpha_r |\psi_r >< \psi_r|$$
- Pure states are complex-weighted sums $$|\psi >= \sum_{r=0}^{R} \alpha_r |r >$$

M. Planck. “On the Theory of the Energy Distribution Law of the Normal Spectrum”. In: Annalen der Physik (1901), p. 553.

Quantum Physics

- Einstein's photons
- De Broglie's energy and momentum / frequency and wave number equivalence
- Momentum is proportional to reduced Planck's constant

$$E = hf \ for \ photons$$

$$c = f\lambda \ for \ waves$$

$$E^2 = p^2c^2 + (mc^2)^2$$

$$m \to 0 \ for \ photons$$

$$\therefore E = pc \to p = \frac{E}{c}$$

$$\therefore p = \frac{h}{\lambda} = \hbar k$$

D. J. Griffiths and D. F. Schroeter, "Introduction to Quantum Mechanics," 2018.

Quantum System Energy

- From here we can describe the kinetic energy of a particle

$$KE = \frac{1}{2}mv^2$$

$$p = mv \rightarrow KE = \frac{p^2}{2m}$$

$$= \frac{h^2}{2\lambda^2 m} \quad = \frac{h^2 k^2}{4\pi^2 2m}$$

$$\therefore KE = \frac{\hbar k^2}{2m}$$

- Schrodinger's equation describes system energy

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + U(x,t)\psi(x,t) = \frac{i\hbar\partial\psi(x,t)}{\partial t}$$

E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. Physical Review, 28(6):1049-1070, Jan 1926.

Differential Values and Operators

- Momentum from kinetic energy $p^2 = \frac{-\hbar^2 \partial^2}{\partial x^2}$ since $mv^2 = \frac{p^2}{2m}$

- Momentum operator $$p = \frac{-i\hbar\partial}{\partial x}$$

- Observable value probability $$\int_a^b |\psi(x,t)|^2 dx$$

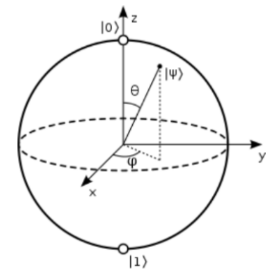
- Expectation values $$< Q >= \int_{-\infty}^{\infty} \psi^*(x,t)\ \hat{Q}\ \psi(x,t)\ dU$$

- i.e. $$< p >= \int_{-\infty}^{\infty} \psi^*(x,t) \frac{\hbar}{i} \frac{\partial}{\partial x} \psi(x,t) dx$$

D. J. Griffiths and D. F. Schroeter, "Introduction to Quantum Mechanics," 2018.

Quantum Computing – Qubits

- Quantum state as a vector $$|\psi\rangle = \alpha_0 \, |0\rangle + \alpha_1 \, |1\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$$
- Polar qubit equations $$\alpha_0 = cos(\frac{\theta}{2})e^{i0} \qquad \alpha_1 = sin(\frac{\theta}{2})e^{i\phi}$$
- Natural normalization $$|\alpha_0|^2 + |\alpha_1|^2 = 1$$
- Pure multi-qubit systems $$|\Psi >= \sum_{i=0}^{N} \alpha_i |i > \qquad \sum_{i=1}^{N} |\alpha_i|^2 = 1$$
- Tensor representation $$|\Phi\rangle = |\phi_0\rangle \otimes |\phi_1\rangle \otimes \cdots \otimes |\phi_n\rangle$$



N. David Mermin. Quantum Computer Science: An Introduction. Cambridge University Press, 2007. doi: 10.1017/CBO9780511813870.

Quantum Computing – Representations

- Pauli operators

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Bloch vectors

$$U = x\sigma_x + y\sigma_y + z\sigma_z \qquad v = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Quantum circuits

$$q[1] \;\; —\boxed{X}—\boxed{U}—$$

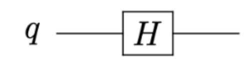
$$q[2] \;\; —\boxed{Y}—\boxed{Z}—$$

N. David Mermin. Quantum Computer Science: An Introduction. Cambridge University Press, 2007. doi: 10.1017/CBO9780511813870.

Quantum Computing – Operations

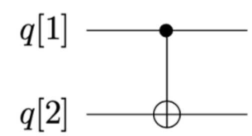
- Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



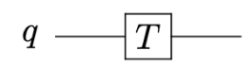
- Controlled-Not gate

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



- T-Gate

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$



- General form

$$U_3(\theta, \phi, \lambda) = \begin{bmatrix} cos(\frac{\theta}{2}) & -e^{\lambda i} sin(\frac{\theta}{2}) \\ e^{\phi i} sin(\frac{\theta}{2}) & e^{\lambda i + \phi i} cos(\frac{\theta}{2}) \end{bmatrix}$$

N. David Mermin. Quantum Computer Science: An Introduction. Cambridge University Press, 2007. doi: 10.1017/CBO9780511813870.

Academic History

Bachelor of Science Double Major Computer Science and Physics	2014-2018
Research Assistantship Laurier Neuromechanics Lab	2015-2017
Master of Physics Quantum Information Science	2018-2020
Research Assistantship Institute for Quantum Computing	2018-2020
Highly Qualified Personnel Ryerson Cybersecurity Research Lab	2019

Industry Participation

SIGMA Development
Partnership
Software and hardware development agency

2016-2019

11 Clients

---

rQadium
Sole Proprietorship (MOU with Business Partner)
Advanced research and development agency

2019-2020

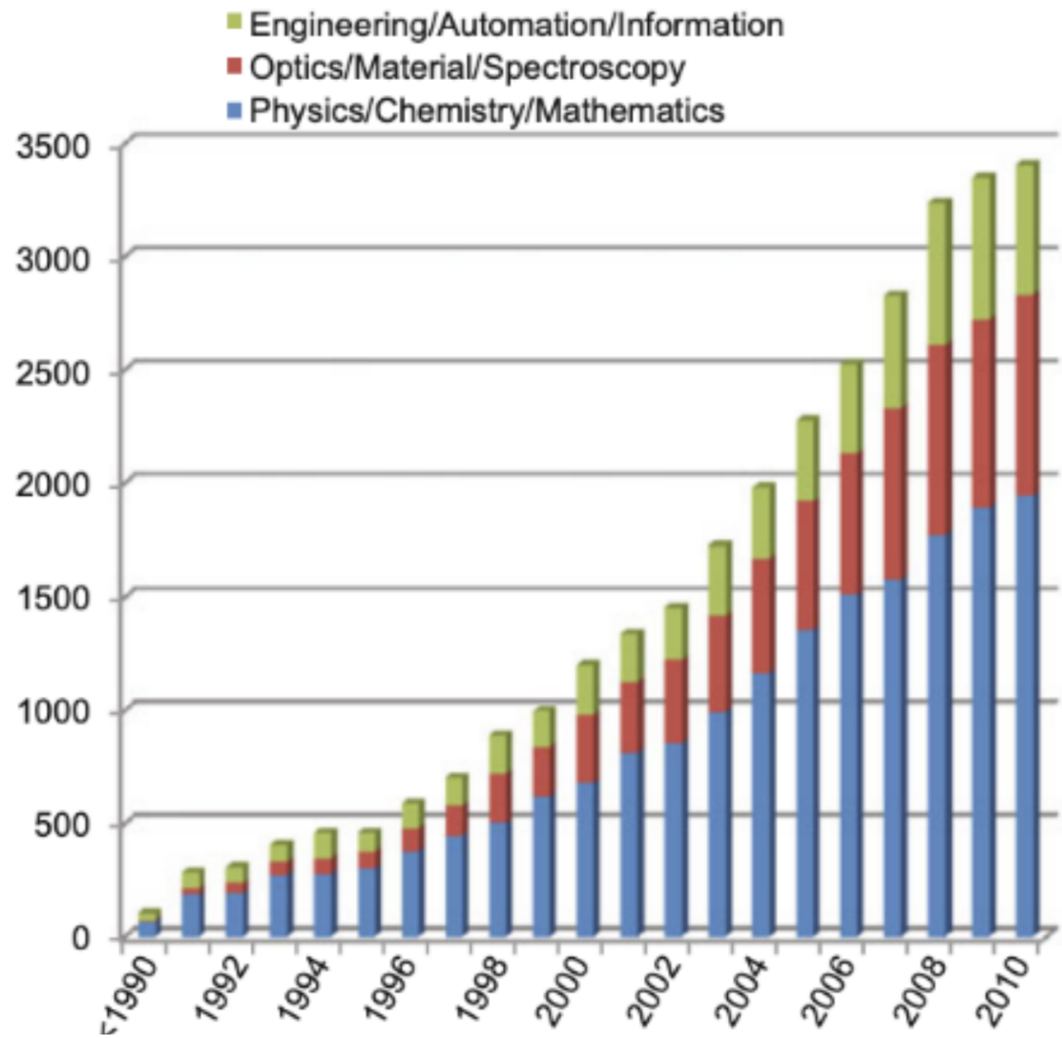
4 Clients

4000+ Downloads

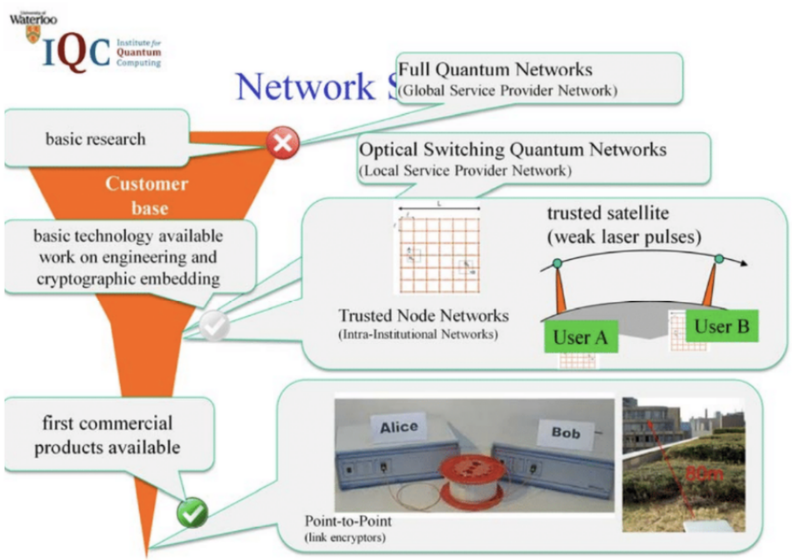
3 Patents*

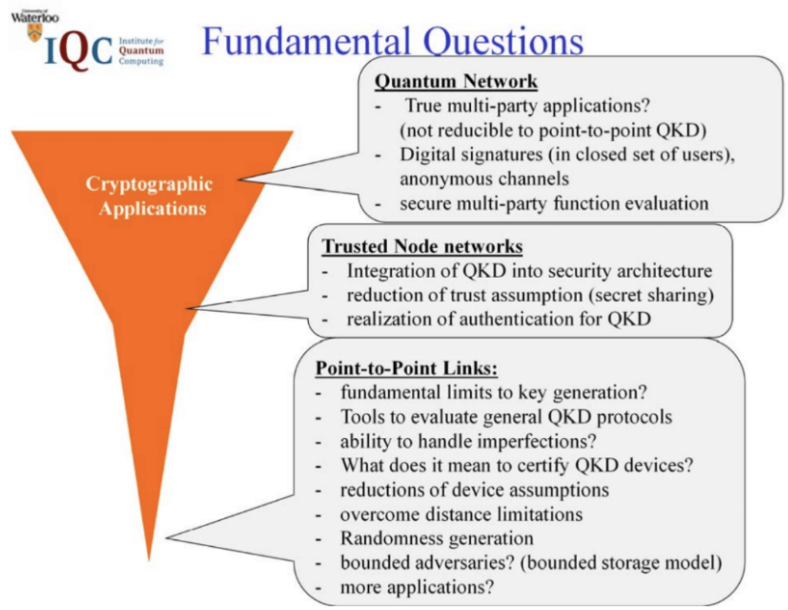
Quantum Control Publications

Quantum Networks ∈ Quantum Control



QIC890 Slides - Open Questions





Norbert Lütkenhaus. "Lecture 22 - QKD Networks." QIC 890 - Applied Quantum Cryptography (Nov. 2020), Waterloo ON, University of Waterloo.

Controlled Teleportation Equations

- Initial state

$$|\psi_{GHZ}>_{ABC}= \frac{|000>_{ABC}+|111>_{ABC}}{\sqrt{2}}$$

- Alice's preparation

$$|x>_x|\psi_{GHZ}>_{ABC}=$$

$$\frac{1}{2}[|\phi^+>_{xA}\otimes(\alpha|00>+\beta|11>)_{BC}$$

$$+|\phi^->_{xA}\otimes(\alpha|00>-\beta|11>)_{BC}$$

$$+|\psi^+>_{xA}\otimes(\alpha|11>+\beta|00>)_{BC}$$

$$+|\psi^->_{xA}\otimes(\alpha|11>-\beta|00>)_{BC}]$$

- Intermediate state example (after Alice's Bell measurement)

$$|\psi>_{BC}=(\alpha|00>+\beta|11>)_{BC}=$$

$$\frac{1}{\sqrt{2}}[(\alpha|0>+\beta|11>)_B|+x>_C$$

$$+(\alpha|0>-\beta|11>)_B|-x>_C]$$

Controlled Teleportation Inputs

α	β	Input State
0.71	0.71	$\|\psi_1>$
0.5	0.87	$\|\psi_2>$
0.3	0.95	$\|\psi_3>$
0.37	0.93	$\|\psi_4>$
0.17	0.98	$\|\psi_5>$

IBM Q Physical Operators

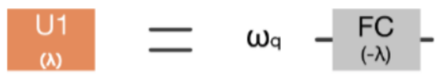


FIG. 11. u1 Frame Change Physical Gate

FIG. 12. u2 Frame Change Physical Gate

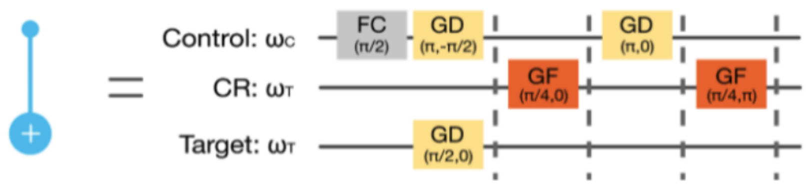


FIG. 13. CX Physical Gate

Quantum Binding Commitments

- Quantum binding commitments

Definition 1 (Classical-style binding) *No algorithm A can output a commitment c and two signatures s, s' that open c to two different messages m and m'.*

Definition 2 (Collapsing hash function - informal) *H is a collapsing hash function iff no quantum polynomial time algorithm B can distinguish between $Game_1$ and $Game_2$. An adversary is valid if A outputs a classical value c and a register M where H(m) = c.*

$Game_1$	$Game_2$
$(S, M, U, c) \leftarrow A(1^\gamma)$	$(S, M, U, c) \leftarrow A(1^\gamma)$
$ok \leftarrow V_c(M, U)$	$ok \leftarrow V_c(M, U)$
$m \leftarrow M_{ok}(M)$	$b \leftarrow B(1^\gamma, S, M, U)$
$b \leftarrow B(1^\gamma, S, M, U)$	

Definition 3 (Collapsing hash function - formal) *A function H $\mathbb{X} \rightarrow \mathbb{Y}$ is $\in$(q)-collapsing if*

$$cAdv[H](q) := \sup_{SMCU} \delta_q(M, \overline{M}|\overline{C}U) \leq \in (q)$$

for all q. The supremum is over all states $SMCU = S\ H(M)\ CU$ with complexity $\leq q$.

Quantum Lightning

- "Bolt" generation

1. Randomly choose n random upper-triangular matrices $A_i \in \{0,1\}^{m\times m}$, and set $\mathbb{A} = \{A_i\}_i$. $\mathbb{A}$ is the public key. Let the hash function $f_{\mathbb{A}} : \{0,1\}^m \to \{0,1\}^n$ be $f_{\mathbb{A}}(x) = (x^T \cdot A_i \cdot x)_i$. If we let operations be taken mod 2, this captures general degree 2 functions over $\mathbb{F}_2$.

2. Begin with a state $|\phi_0 > \quad \sum_{\Delta} \sum_{x \in S_\Delta} \frac{1}{2^{kn/2}\sqrt{|S_\Delta|}} |\Delta, x>$

3. Δ is defined such that we can run a computation which maps $\Delta = (\Delta_1, ..., \Delta_k)$ to an affine space S_Δ s.t. $\forall x \in S, f_{\mathbb{A}}(x) = f_{\mathbb{A}}(x + \Delta_j) \forall j$. Then we construct a uniform superposition of elements in S_Δ to yield:

$$\sum_{\Delta} \sum_{x \in S_\Delta} \frac{1}{2^{kn/2}\sqrt{|S_\Delta|}} |\Delta, x>$$

5. Compute the maps $(x, \Delta_1, ..., \Delta_k)$ to $(x, x - \Delta_1, ..., x - \Delta_k)$ in superposition. The final state is a bolt

A Quantum Voting Protocol

- Ballot commitment

1. For each $i \in \{1, ..., n\}$ voter V_i generates the i^{th} row of an $n \times n$ matrix of integers $r_{i,1}, ..., r_{i,n}$ such that $\sum_j r_{io,j} = 0(mod\, n+1)$.
2. For each i, j voter V_i sends $r_{i,j}$ to voter V_j via a quantum secure communication.
3. Then each voter V_i knows the i^{th} column $r_{1,i}, ..., r_{n,i}$. V_i computes his/her masked ballot $\hat{v}_i = v_i + \sum_j r_{j,i}(mod\, n+1)$. V_i commits $\hat{v}_i$ to every tallier of the blockchain via a quantum commitment protocol.

- Ballot tallying

1. Each voter V_i reveals $\hat{v}_i$ to every tallier of the blockchain by opening his/her commitment.
2. The talliers each run the Quantum Honest Success Byzantine Agreement Protocol to reach a consensus on the value of the masked ballot $\hat{v}_1, ..., \hat{v}_n$.
3. The result of the vote is $\sum_i \hat{v}_i = \sum_i v_i(mod\, n+1)$.

Attributions

Icons were made by Pixel Perfect, Freepik, Icongeek26, Kiranshastry, Good Ware, surang, Flat Icons, Pixel perfect from www.flaticon.com.

References

J. Abhijith et al. “Quantum Algorithm Implementations for Beginners”. In: arXiv e-prints, arXiv:1804.03719 (Apr. 2018), arXiv:1804.03719. arXiv: 1804. 03719 [cs.ET].

Allen et al. PFC: A Program to Convert Fortran to Parallel Form. Mar. 1982. url: https://hdl.handle.net/1911/101547.

J. R. Allen et al. “Conversion of control dependence to data dependence”. In: Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages- POPL 83 (1983). doi: 10.1145/567067.567085.

Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. “Quantum Attacks on Classical Proof Systems: The Hardness of Quantum Rewinding”. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science (2014). doi: 10.1109/focs.2014.57.

Marcella Atzori. “Blockchain Technology and Decentralized Governance: Is the State Still Necessary?” In: SSRN Electronic Journal (2015). doi: 10.2139/ssrn. 2709713.

Pavan Balaji. “OpenMP”. In: Programming Models for Parallel Computing (2015). doi: 10.7551/mitpress/9486.003.0014.

Barenco et al. “Elementary gates for quantum computation”. eng. In: Physical review. A, Atomic, molecular, and optical physics 52.5 (1995), pp. 3457–3467. issn: 1050-2947.

Charles H. Bennett and Gilles Brassard. “Quantum cryptography: Public key distribution and coin tossing”. In: arXiv e-prints, arXiv:2003.06557 (Mar. 2020), arXiv:2003.06557. arXiv: 2003.06557 [quant-ph].

Charles Bennett and Gilles Brassard. “Quantum public key distribution reinvented”. In: ACM SIGACT News 18 (July 1987), pp. 51–53. doi: 10.1145/ 36068.36070.

Charles H. Bennett et al. “Strengths and Weaknesses of Quantum Computing”. In: SIAM Journal on Computing 26.5 (1997), pp. 1510–1523. doi: 10.1137/ s0097539796300933.

Guido Bertoni et al. “On the Indifferentiability of the Sponge Construction”. In: Advances in Cryptology - EUROCRYPT 2008 Lecture Notes in Computer Science (2008), pp. 181–197. doi: 10.1007/978-3-540-78967-3_11.

Eli Biham and Tal Mor. “Bounds on Information and the Security of Quantum Cryptography”. In: Physical Review Letters 79.20 (1997), pp. 4034–4037. doi: 10.1103/physrevlett.79.4034.

Anatolij G. Butkovskij and Jurij I. Samojlenko. Control of quantum mechanical processes and systems. Kluwer, 1990.

Vitalik Buterin. Ethereum whitepaper. 2013. url: https : / / whitepaper . io / document/5/ethereum-whitepaper.

C.r. Baugh and B.a. Wooley. “ A Twos Complement Parallel Array Multiplication Algorithm”. In: IEEE Transactions on Computers C-22.12 (1973), pp. 1045–1047. doi: 10.1109/t-c.1973.223648.

Andrea Coladangelo. “Smart contracts meet quantum cryptography”. In: arXiv e-prints, arXiv:1902.05214 (Feb. 2019), arXiv:1902.05214. arXiv: 1902.05214 [quant-ph].

Recruit Communications. PyQUBO. url: https : / / github . com / recruit - communications/pyqubo.
ConsenSys. Smart Dubai. url: https://consensys.net/blockchain-use-cases/ government-and-the-public-sector/smart-dubai/.

ConsenSys. Get to Know the ConsenSys Mesh. Aug. 2019. url: https://media. consensys.net/get-to-know-the-47-projects-that-make-up-the-consensys- mesh-478b7d3028c1.

Crandall. “Nanotechnology: Molecular Speculations on Global Abundance”. In: Precision Engineering 20.2 (1997), p. 148. doi: 10.1016/0141-6359(97)90052-0.

Giuseppe Carleo and Matthias Troyer. “Solving the quantum many-body problem with artificial neural networks”. In: Science 355.6325 (Sept. 2017), pp. 602–606. doi: 10.1126/science.aag2302.

Yury Delendik. yurydelendik/wasmparser.rs. url: https : / / github . com / yurydelendik/wasmparser.rs.

MDN Web Docs. Understanding WebAssembly text format. url: https : / / developer.mozilla.org/en-US/docs/WebAssembly/Understanding_the_text_ format#S-expressions.

D-Wave. url: https://www.dwavesys.com/p-ress-releases/d-wave-previ-ews- next-generation-quantum-computing-platform.

D-Wave. D-Wave Customers. url: https://www.dwavesys.com/our-company/ customers.

D-Wave. The D-Wave 2000QTM System. url: https://www.dwavesys.com/d- wave-two-system.

D-Wave. Wave Ocean Software Documentation. url: http : / / docs . ocean . dwavesys.com/.

D-Wave. dwavesystems/qbsolv. Feb. 2020. url: https : / / github . com / dwavesystems/qbsolv.

AccessScience Editors. Reaching the 50-qubit milestone in quantum computing. 2020/5/11/ 2017. url: https://www.accessscience.com/content/reaching-the- 50-qubit-milestone-in-quantum-computing/BR1120171.

OSTechNix Editor. Blockchain 2.0 - What Is Ethereum [Part 9]. May 2019. url: https://www.ostechnix.com/blockchain-2-0-what-is-ethereum.

Chris Fisher and Eric Abenojar. IBM: Quantum Computing. Apr. 2009. url: https://www.research.ibm.com/ibm-q/.

Serguei Fedortchenko. “A quantum teleportation experiment for undergraduate students”. In: (20160708).

Serge Fehr. “Quantum Cryptography”. eng. In: Foundations of Physics 40.5 (2010), pp. 494–531. issn: 0015-9018.

Serge Fehr. “Classical Proofs for the Quantum Collapsing Property of Classical Hash Functions”. In: Theory of Cryptography Lecture Notes in Computer Science (2018), pp. 315–338. doi: 10.1007/978-3-030-03810-6_12.

Richard Finney and Daoud Meerzaman. “Chromatic: WebAssembly-Based Cancer Genome Viewer”. In: Cancer Informatics 17 (2018). doi: 10.1177/ 1176935118771972.

WebAssembly Foundation. WebAssembly Binary Toolkit. May 2019. url: github.com/WebAssembly/wabt.

Jake Frankenfield. Proof of Stake (PoS). Jan. 2020. url: https : / / www . investopedia.com/terms/p/proof-stake-pos.asp.

Brenda Goh. China wants to ban bitcoin mining. Apr. 2019. url: http://www. reuters . com / article / us- china-cryptocurrency / china- wants- to- ban- Bitcoin- mining-idUSKCN1RL0C4.

Ilya Grigorenko and Herschel Rabitz. “Optimal control of the local electromagnetic response of nanostructured materials: Optimal detectors and quantum disguises.” In: Applied Physics Letters 94.25 (2009), p. 253107. doi: 10.1063/1.3159879.

Goren Gordon and Gustavo Rigolin. “Generalized Quantum State Sharing”. In: 73.6 (20060323). issn: 10502947.

Lov K. Grover. “A fast quantum mechanical algorithm for database search”. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC 96 (1996). doi: 10.1145/237814.237866.

Jay Gambetta and Sarah Sheldon. Cramming More Power Into a Quantum Device. Mar. 2019. url: http://www.ibm.com/blogs/research/2019/03/power- quantum-device/.

Andreas Haas et al. “Bringing the web up to speed with WebAssembly”. In: Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation - PLDI 2017 (2017). doi: 10.1145/3062341. 3062363.

P.o.a. Haikonen. “A modular neural system for machine cognition”. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (2000). doi: 10.1109/ijcnn.2000.857812.

Jiu-Cang Hao, Chuan-Feng Li, and Guang-Can Guo. “Controlled dense coding using the Greenberger-Horne-Zeilinger state”. In: Physical Review A 63.5 (Nov. 2001). doi: 10.1103/physreva.63.054301.

Matthew Iles. The Civil White Paper. Oct. 2018. url: https://blog.joincivil. com/the-civil-white-paper-3e6c6f72dd9e.

NXM Labs Inc. NXM Labs Announces Breakthrough in Quantum-Safe Security for Existing Computers and IoT Devices. Apr. 2019. url: http://www. newswire . ca / news - releases / nxm - labs - announces - breakthrough - in - quantum - safe-security-for-existing-computers-and-iot-devices-890174168.html.

Wasmer Inc. wasmerio/wasmer. Apr. 2020. url: https : / / github . com / wasmerio/wasmer.

A. Karlsson and M. Bourennane. “ Quantum Teleportation using Three Particle Entanglement”. In: Technical Digest. 1998 EQEC. European Quantum Electronics Conference (Cat. No.98TH8326) (). doi: 10.1109/eqec.1998.714867.

J. A. Kirkham. OpenMP . Issue 97 . WebAssembly/threads. url: https : / / github.com/WebAssembly/threads/issues/97.

A Yu Kitaev. “Quantum computations: algorithms and error correction”. eng. In: Russian Mathematical Surveys 52.6 (1997), pp. 1191–1249. issn: 0036-0279.

Xihan Li and Fuguo Deng. “Controlled teleportation”. In: Frontiers of Computer Science in China 2.2 (2008), pp. 147–160. doi: 10.1007/s11704-008-0020-0.

Sylvestre Ledru. Making the Building of Firefox Faster for You with Clever- Commit from Ubisoft. Feb. 2019. url: https://blog.mozilla.org/futurereleases/ 2019/02/12/making-the-building-of-firefox-faster-for-you-with-clever-commit- from-ubisoft/.

Scott Logic. Writing WebAssembly By Hand. url: https://blog.scottlogic.com/ 2018/04/26/webassembly-by-hand.html.

Andrew Lutomirski et al. “Breaking and making quantum money: toward a new quantum cryptographic protocol”. In: arXiv e-prints, arXiv:0912.3825 (Dec. 2009), arXiv:0912.3825. arXiv: 0912.3825 [quant-ph].

Wing K. Luk and Jean Vuillemin. “Recursive implementation of optimal time VLSi integer multipliers”. In: 1984.

Steven P. Lalley and E. Glen Weyl. “Quadratic Voting: How Mechanism Design Can Radicalize Democracy”. In: AEA Papers and Proceedings 108 (2018), pp. 33–37. doi: 10.1257/pandp.20181002.

Tanaya Macheel. Banks Will Start Actually Using Blockchain Next Year: IBM Report. Jan. 2017. url: https://www.americanbanker.com/news/banks-will- start-actually-using-blockchain-next-year-ibm-report.

E. Megidish et al. “Entanglement Swapping between Photons that have Never Coexisted”. In: Physical Review Letters 110.21 (2013). doi: 10.1103/ physrevlett.110.210403.

E. Megidish et al. “Quantum tomography of inductively created multiphoton states”. In: Cleo: 2013 (2013). doi: 10.1364/cleo_qels.2013.qf2b.6.

N. David Mermin. Quantum Computer Science: An Introduction. Cambridge University Press, 2007. doi: 10.1017/CBO9780511813870.

Jack Matier and Pete Waterland. Quantum Resistant Ledger (QRL). url: https : / / raw . githubusercontent . com / theQRL / Whitepaper / master / QRL _ whitepaper.pdf.

Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. url: https: //bitcoin.org/bitcoin.pdf.

Naoki Nakatani. “Matrix Product States and Density Matrix Renormalization Group Algorithm”. In: Reference Module in Chemistry, Molecular Sciences and Chemical Engineering (2018). doi: 10.1016/b978-0-12-409547-2.11473-8.

ETH News. ConsenSys Releases Whitepaper At Dubai's World Government Summit. Feb. 2017. url: www.ethnews.com/consensys-releases-whitepaper- at-dubais-world-government-summit.

Michael Nofer et al. "Blockchain". In: Business & Information Systems Engineering 59.3 (2017), pp. 183–187. doi: 10.1007/s12599-017-0467-3.

Michael Oved and Don Mosites. Swap Protocol Whitepaper. May 2017. url: https://swap.tech/whitepaper/.

Daniel Omalley and Velimir V. Vesselinov. "ToQ.jl: A high-level programming language for D-Wave machines based on Julia". In: 2016 IEEE High Performance Extreme Computing Conference (HPEC) (2016). doi: 10.1109/ hpec.2016.7761616.

Scott Pakin. qb2qmasm. url: https : / / github . com / lanl / qmasm / wiki / qb2qmasm.

Scott Pakin. "A quantum macro assembler". In: 2016 IEEE High Performance Extreme Computing Conference (HPEC) (2016). doi: 10.1109/hpec.2016. 7761637.

Scott Pakin. "Targeting Classical Code to a Quantum Annealer". In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (Apr. 2019). doi: 10.1145/ 3297858.3304071.

M. Planck. "On the Theory of the Energy Distribution Law of the Normal Spectrum". In: Annalen der Physik (1901), p. 553.

Wolfgang Platz and Wolfgang Platz. 3 biggest roadblocks to continuous testing. July 2018. url: https://www.infoworld.com/article/3294197/3-biggest- roadblocks-to-continuous-testing.html.

IBM Q. Melbourne Gate Specifications. June 2018. url: github . com / Qiskit / ibmq-device-information/blob/master/backends/melbourne/V1/version%5C_ log.md%5C#gate-specification.

IBM Q. Quantum Devices & Simulators. June 2018. url: www.research.ibm. com/ibm-q/technology/devices/%5C#ibmq%5C_16%5C_melbourne.

Chad Rigetti. Home. url: https://www.rigetti.com/.

Ronald L. Rivest. S-Expressions. May 1994. url: http://people.csail.mit.edu/ rivest/Sexp.txt.

Del Rajan and Matt Visser. “Quantum Blockchain Using Entanglement in Time”. In: Quantum Reports 1.1 (2019), pp. 3–11. doi: 10 . 3390 / quantum1010002.

Paul Schaus. Blockchain Projects Will Pay Off - 10 Years from Now. Dec. 2016. url: http://www.americanbanker.com/opinion/blockchain-projects-will-pay- off-10-years-from-now.

Sankar Das Sarma, Dong-Ling Deng, and Lu-Ming Duan. “Machine learning meets quantum physics”. In: Physics Today 72.3 (2019), pp. 48–54. doi: 10. 1063/pt.3.4164.

Vivek V. Shende and Igor L. Markov. “On the CNOT-cost of TOFFOLI gates”. In: arXiv e-prints, arXiv:0803.2316 (Mar. 2008), arXiv:0803.2316. arXiv: 0803. 2316 [quant-ph].

British Standards. “electronic design interchange format”. In: (July 2001). doi: 10.3403/bsen61690.

Xin Sun et al. “A Simple Voting Protocol on Quantum Blockchain”. In: International Journal of Theoretical Physics 58.1 (2018), pp. 275–281. doi: 10.1007/s10773-018-3929-6.

Xin Sun et al. “Quantum-enhanced Logic-based Blockchain I: Quantum Honest- success Byzantine Agreement and Qulogicoin”. In: arXiv e-prints, arXiv:1805.06768 (May 2018), arXiv:1805.06768. arXiv: 1805.06768 [quant-ph].

Dmitriy Tsvettsikh and WebAssembly Foundation. wasmerio/vscode-wasm. Feb. 2020. url: https://github.com/wasmerio/vscode-wasm.

T.J Tarn, Garng Huang, and John W Clark. “Modelling of quantum mechanical control systems”. eng. In: Mathematical Modelling 1.1 (1980), pp. 109–121. issn: 0270-0255.

Treum. Verified Organic and ConsenSys-backed Treum launch Ethereum blockchain solution to track and trace the first commercial hemp crop planted in Arizona. June 2019. url: https://itsupplychain.com/verified-organic-and- consensys-backed-treum-launch-ethereum-blockchain-solution-to-track-and- trace-the-first-commercial-hemp-crop-planted-in-arizona/.

Dominique Unruh. “Collapse-Binding Quantum Commitments Without Random Oracles”. In: Advances in Cryptology- ASIACRYPT 2016 Lecture Notes in Computer Science (2016), pp. 166–195. doi: 10.1007/978-3-662- 53890-6_6.

Dominique Unruh. “Computationally Binding Quantum Commitments”. In: Advances in Cryptology- EUROCRYPT 2016 Lecture Notes in Computer Science (2016), pp. 497–527. doi: 10.1007/978-3-662-49896-5_18.

L.M.K. Vandersypen and I.L. Chuang. “NMR techniques for quantum control and computation.(nuclear magnetic resonance)”. English. In: Reviews of Modern Physics 76.4 (2004), pp. 1037–1069. issn: 0034-6861.

Dejan Vujicic, Dijana Jagodic, and Sinisa Randic. “Blockchain technology, bitcoin, and Ethereum: A brief overview”. In: 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH) (2018). doi: 10.1109/infoteh. 2018.8345547.

Stephen Wiesner. “Conjugate coding”. In: ACM SIGACT News 15.1 (1983), pp. 78–88. doi: 10.1145/1008908.1008920.

Aaron Wright and David Roon. A free legal repository. url: http : / / www . openlaw.io/.

ReBing Wu et al. “Control problems in quantum systems”. In: Chinese Science Bulletin 57.18 (2012), pp. 2194–2199. doi: 10.1007/s11434-012-5193-0. url: https://doi.org/10.1007/s11434-012-5193-0.

Dylan J. Yaga et al. Blockchain Technology Overview. Nov. 2018. url: https: //www.nist.gov/publications/blockchain-technology-overview.

Christine Yen, Adam Stacoviak, and Jerod Santo. Observability Is for Your Unknown Unknowns with Christine Yen. Aug. 2019. url: https://changelog. com/podcast/356.

Mark Zhandry. “Quantum Lightning Never Strikes the Same State Twice”. In: Advances in Cryptology-EUROCRYPT 2019 Lecture Notes in Computer Science (2019), pp. 408–438. doi: 10.1007/978-3-030-17659-4_14.

E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. Physical Review, 28(6):1049-1070, Jan 1926.

D. J. Griffiths and D. F. Schroeter, “Introduction to Quantum Mechanics,” 2018.

Shukla, Abhishek, et al. “Complete Characterization of the Directly Implementable Quantum Gates Used in the IBM Quantum Processors.” 2018.