SMU Darwin Deason Institute for Cyber Security




Quantum Oracle Synthesis with an Application to QRNG

Mitchell A. Thornton

Darwin Deason Institute for Cyber Security

Southern Methodist University, Dallas, Texas, USA

September 28, 2023

Acknowledgements

Student Investigators

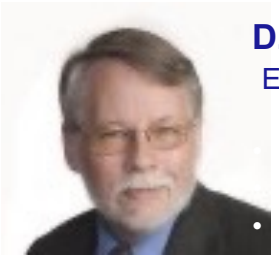
Avi Sinha
Graduate RA
CpE Major (PhD)

Jessie Henderson
Graduate RA
Math Major (MS)
CS/Philosophy (Minor)

Elena Henderson
Graduate RA
Math Major (MS)
CS (Minor)

Faculty Investigators

Eric Larson, Ph.D.
Associate Professor
Researcher in
Darwin Deason
Institute for Cyber Security
Faculty in CS Dept.



D. Michael Miller, Ph.D.
Emeritus Professor
Emeritus Assoc. Provost
University of Victoria
B.C. Canada
Faculty in CS Dept.

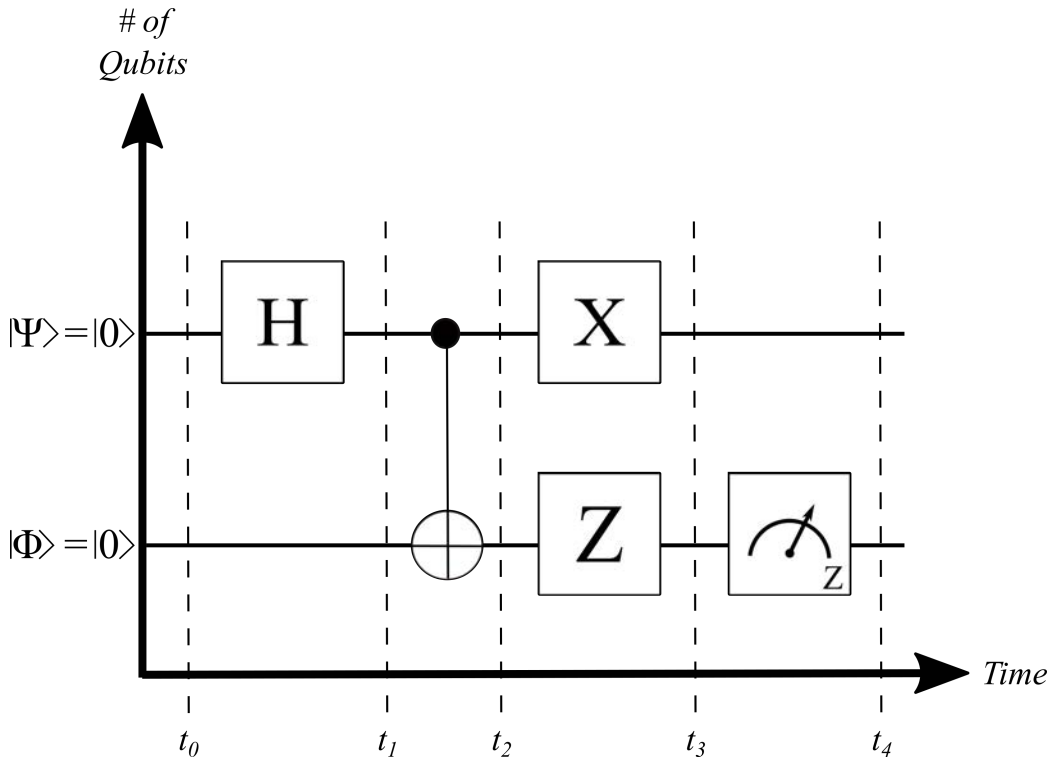
Mitch Thornton, Ph.D.
Green Chair of Engineering
and Professor
Director/Researcher in Darwin
Deason Institute for Cyber
Security
Faculty in ECE Dept.

Presentation Outline

- Quantum Circuits and Oracles
  - What is an Oracle in a Quantum Circuit?
  - Two Oracle Synthesis Methods
  - Experimental Results
- Programmable Weakly Random Source (WRS) for a Quantum Random Number Generator
  - PRNG/TRNG/QRNG: Overview, Architecture and Usage
  - Amplitude Encoding of Probability Mass Function (PMF) as Oracles
  - QRNG Weakly Random Source (WRS) Automated Synthesis
  - QRNG Experimental Results using the 5-qubit IBM QC Noisy Simulator
- Summary and Q&A

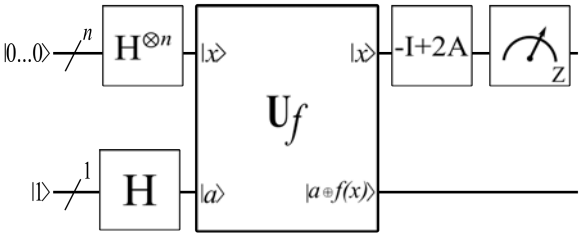
Gate Model Paradigm of Quantum Computing

- Graphical Depiction of Quantum Algorithms
  - "Gate Model" Computational Paradigm
  - Interpretation of Circuit
- Realize quantum algorithms as quantum circuits.
- Low-level gates applied to qubits.

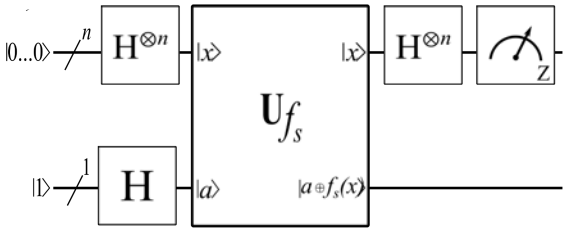


Oracle Examples and Synthesis

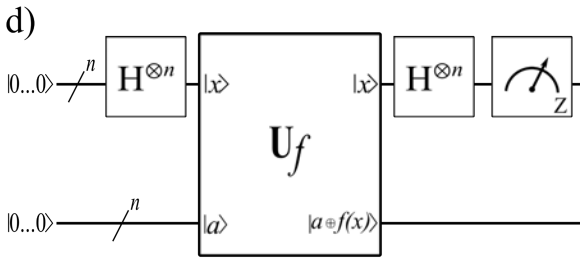
- Two methods for automated oracle synthesis.
- Implemented as part of a tool for automated quantum compilation and optimization.
- Specify irreversible, incomplete functions for oracle synthesis.



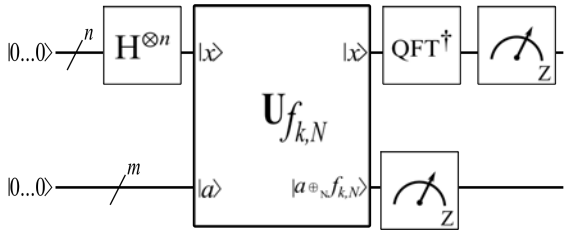
Grover's database search



Bernstein-Vazirani hidden string finder



Simon's periodicity finder



Shor's factorization

Examples of Oracles in "Well Known" Quantum Circuits

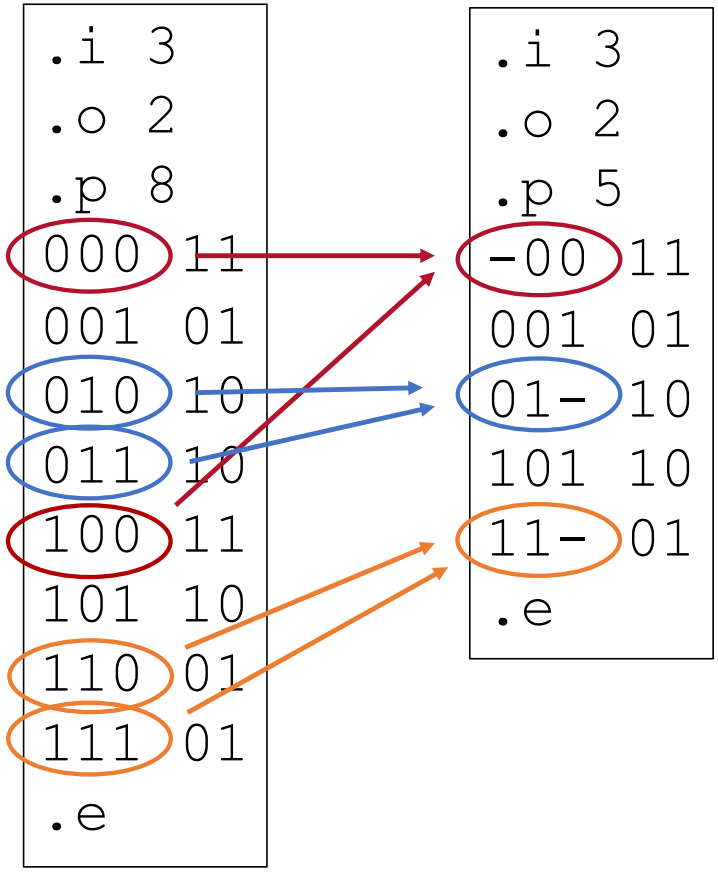
Oracle Properties and Definitions

- Oracle – Subcircuit that represents function of interest f.
  - Switching function representation: f: $\mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$
  - May be irreversible and/or incompletely specified.
- Embedding function f_e: $\mathbb{Z}_2^{n+a} \rightarrow \mathbb{Z}_2^{m+g}$
  - Switching function with the same mappings as f.
  - Ancilla (a) and garbage (g) may be added to make a bijection.
- Oracle function $\boldsymbol{U}_f$: $\mathcal{H}^{2^N} \rightarrow \mathcal{H}^{2^N}$, $N = n + a = m + g$
  - Realization of f comprised of quantum gates.

J.M. Henderson, E.R. Henderson, A. Sinha, M.A. Thornton and D.M. Miller. Automated Quantum Oracle Synthesis with a Minimal Number of Qubits, SPIE 12517, Quantum Information Science, Sensing, and Computation XV, April 30-May 4, 2023. Available via arXiv: https://arxiv.org/pdf/2304.03829.pdf.

Automated Oracle Synthesis with .pla Representation

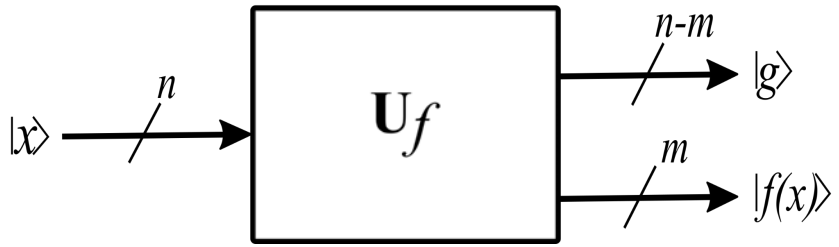
- User provides specification for f.
  - Classical switching function.
  - May be irreversible and/or incompletely specified.
  - In tabular `.pla` format.
  - Standard Sum-of-Products form
  - Other Representations are more Efficient such as QMDD*, Verilog netlists, etc.
- GOAL: Produce an oracle function specification in OpenQASM.



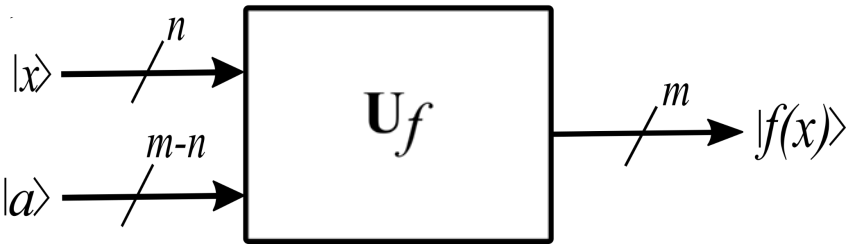
*D.M. Miller and M.A. Thornton, QMDD: A Decision Diagram Structure for Reversible and Quantum Circuits, IEEE International Symposium on Multiple-Valued Logic (ISMVL), May 17-20, 2006, pp. 30-30.

Oracle Synthesis with Minimal Qubits

- Given f: $\mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$, the minimal number of qubits is $max(n, m)$.
- When $n > m$



- When $m < n$



Reversible/Bijective Functions Require the Same Number of Variables and Function Lines

Oracle Synthesis with Minimal Qubits: Embedding Function

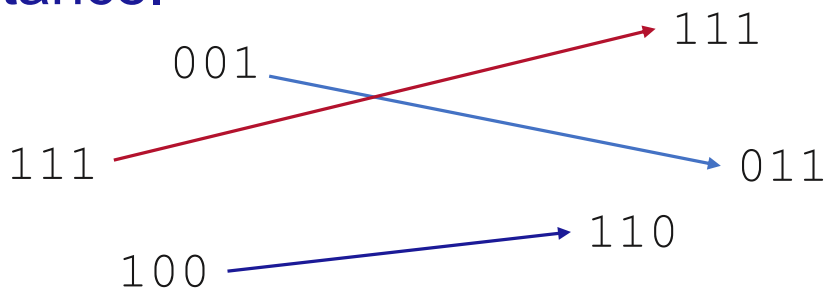
- RTT Method*: embed f in one-to-one function.

1. Determine N_{dup}, the maximum number of times an output bitstring is duplicated.
2. Add ancilla and/or garbage qubits to the function representation.
    1. If $N_{dup} = 0$, then f is already one-to-one, so add nothing.
    2. Otherwise, add $v = log_2(N_{dup})$ garbage outputs and then $v + m - n$ ancilla inputs.
3. Assign values from $\{0, 1\}$ to all ancilla and garbage so that the duplicate output bitstrings are now differentiated.

*E. Gabrielsen and M.A. Thornton, Minimizing Ancilla and Garbage Qubits in Reversible Functions, Southwest Quantum Information and Technology 20th Annual SQuInT Workshop (SQuInT), February 22-24, 2018.

Oracle Synthesis with Minimal Qubits: Embedding Function

- Embed one-to-one function into onto f_e.
- Pair the input and output bitstrings that are not explicitly specified.
- Make the input and output of each pair as similar as possible: Minimize their Hamming distance.



Oracle Synthesis with Minimal Qubits: TBS Method 1

- Basic unidirectional Transformation Based Synthesis (TBS) method.
  - Process the embedding function in order of ascending input bitstrings.
  - Produce a cascade of quantum gates that will comprise the oracle function.

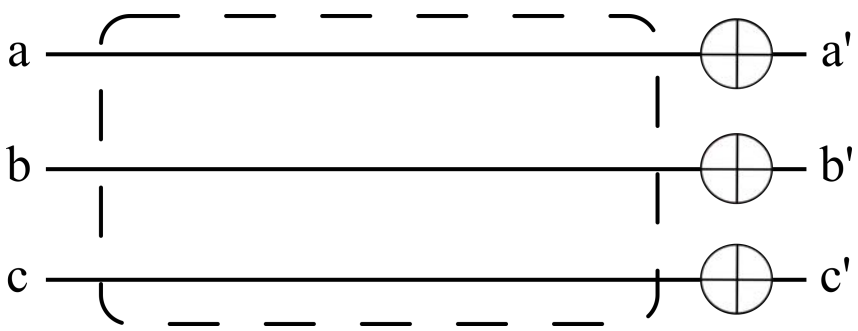
- For each input-output pair
  1. Determine a sequence of quantum gates that transform the output bitstring into the input bitstring.
  2. Add these gates to the input side of the gate cascade.
  3. Apply these gates to the function, producing an intermediate function representation that is then used as the function to be synthesized.

E.R. Henderson, J.M. Henderson, A. Sinha, E.C. Larson, D.M. Miller and M.A. Thornton. Automated Synthesis of Quantum Subcircuits, September 2, 2023. Available via arXiv: https://arxiv.org/pdf/2309.01028.pdf.

TBS Method 1 Example

```
.i 3
.o 3
.p 8
000 111
001 001
010 100
011 011
100 000
101 010
110 110
111 101
.e
```

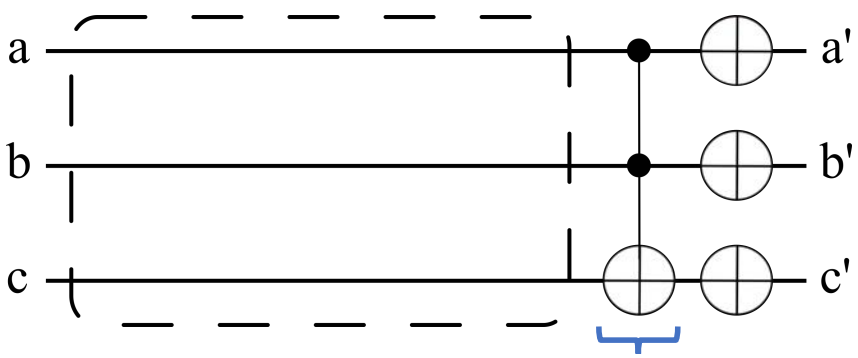
```
.i 3
.o 3
.p 8
000 000
001 110
010 011
011 100
100 111
101 101
110 001
111 010
.e
```



TBS Method 1 Example

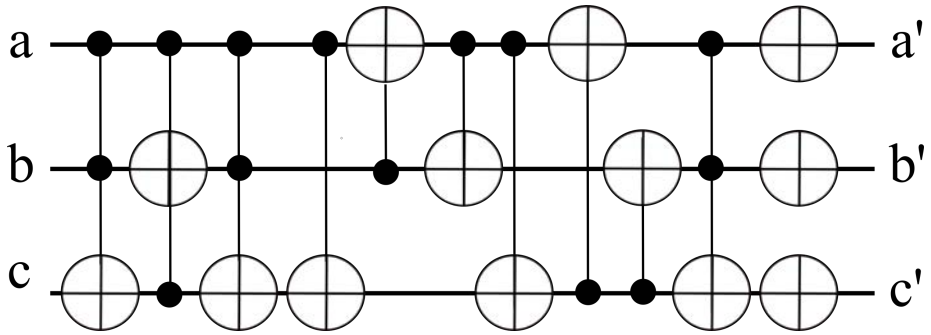
```
.i 3
.o 3
.p 8
000 000
001 110
010 011
011 100
100 111
101 101
110 001
111 010
.e
```

```
.i 3
.o 3
.p 8
000 000
001 111
010 011
011 100
100 110
101 101
110 001
111 010
.e
```



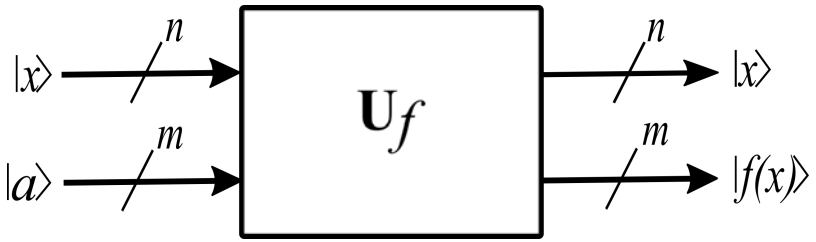
TBS Method 1 Example

```
.i 3
.o 3
.p 8
000 111
001 001
010 100
011 011
100 000
101 010
110 110
111 101
.e
```



ESOP Method 2: Oracle Synthesis with Preserved Domain Qubits

- Keep domain values when range values produced.
- $f\colon \mathbb{Z}_2^n \to \mathbb{Z}_2^m$



- Implicitly formulate f_e.
  - Interpret each domain (n qubits) and range (m qubits) pair of f as an $n+m$ qubit output.

Oracle Synthesis with Preserved Domain Qubits: ESOP Method 2

- Exclusive-Or Sum-of-Products (ESOP) Method.
- Represent f as an ESOP.
  - Input variables are combined with AND operators to make product terms.
  - Product terms are summed together with exclusive-OR operators.
- EXORCISM-4.
  - Creates minimized ESOP form of f.
  - Represented in `.pla` file format, where each input bitstring is a product term of the ESOP representation of f.

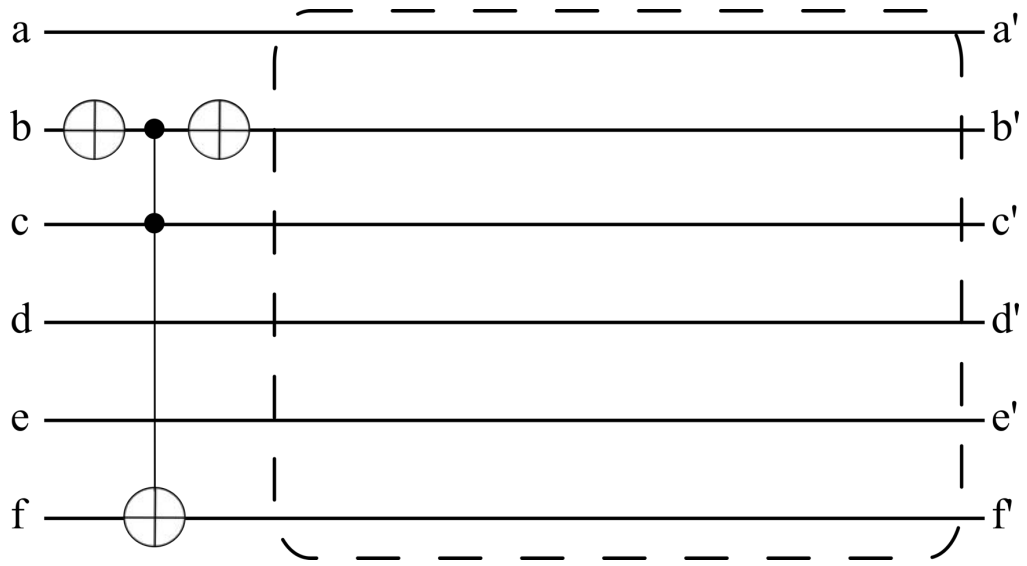
K. Fazel, M.A. Thornton and J.E. Rice, ESOP-based Toffoli Gate Cascade Generation, Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, August 22-24, 2007, pp. 206-209.

Oracle Synthesis with Preserved Domain Qubits: ESOP Method 2

- Produce a cascade of gates that comprise the oracle function.
- For each input-output pair (each product term)
  - For each output variable value of 1, add a Toffoli gate to the output side of the cascade.
    - Toffoli gate target is on the corresponding output qubit.
    - Toffoli gate controls are determined by the input variable values.
      - Each value of 1 maps to a control on the corresponding input qubit.
      - Each value of 0 maps to a control sandwiched between two Pauli-X gates on the corresponding input qubit.
- The cascade is the oracle function.

ESOP Method 2 Example

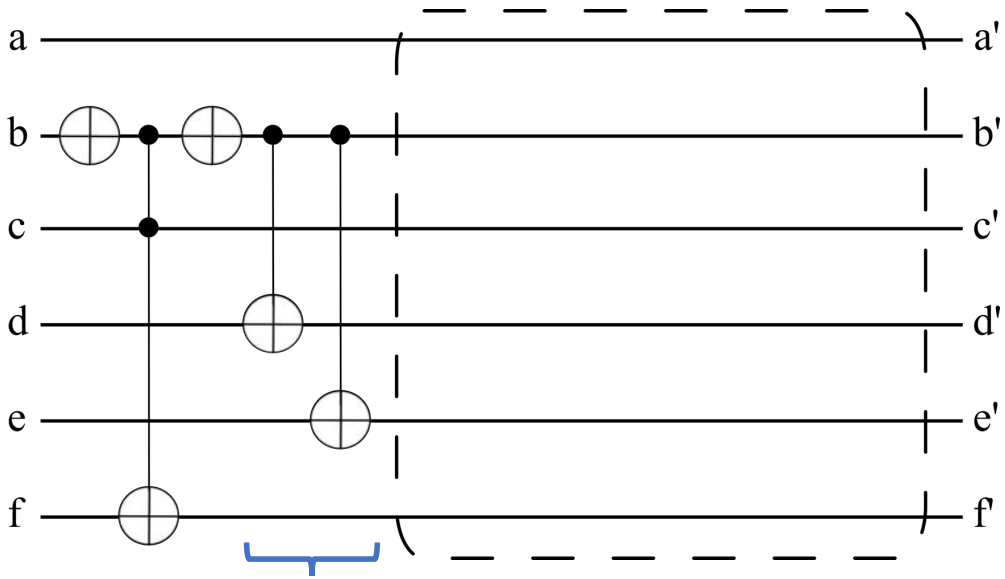
```
.i 3
.o 3
.p 5
-01 001
-1- 110
0-0 111
01- 101
1-1 011
.e
```



Must convert standard SOP .pla into ESOP: use exclusive-OR rather than inclusive-OR

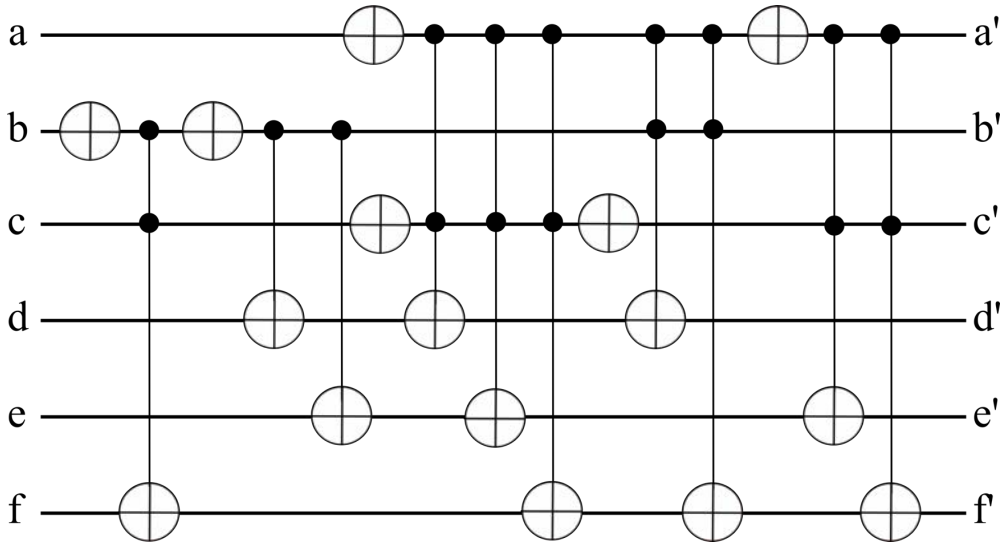
ESOP Method 2 Example

```
.i 3
.o 3
.p 5
-01 001
-1- 110
0-0 111
01- 101
1-1 011
.e
```



ESOP Method 2 Example

```
.i 3
.o 3
.p 5
-01 001
-1- 110
0-0 111
01- 101
1-1 011
.e
```



Oracles for Benchmark Functions

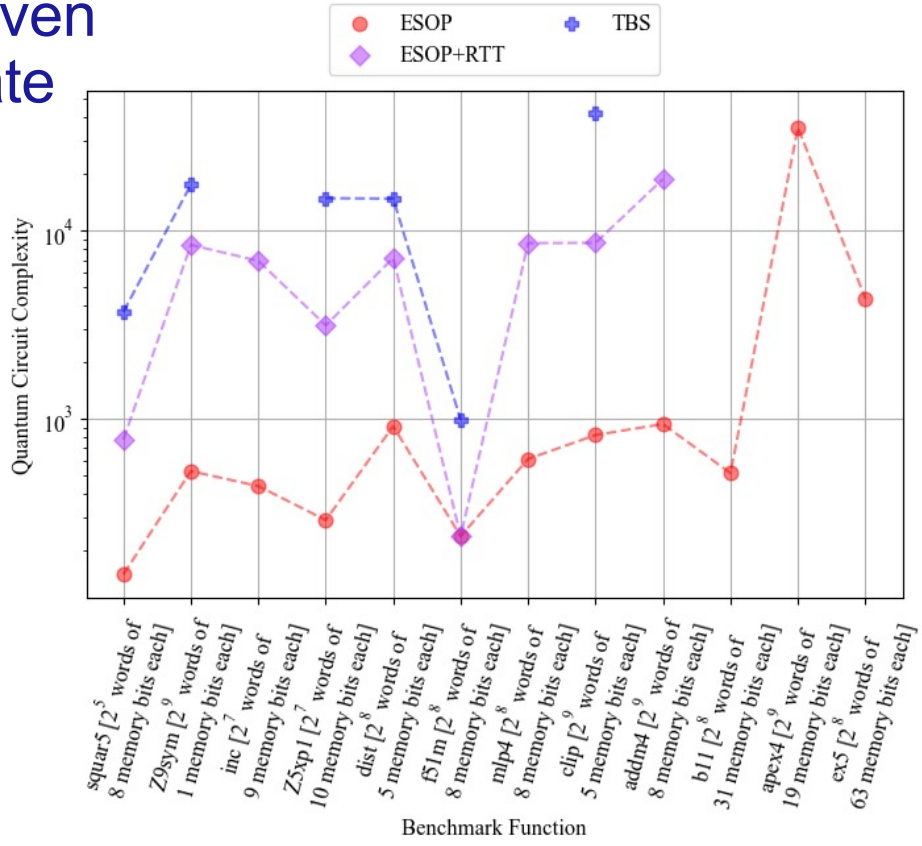
- Synthesized functions of interest from a `.pla` benchmark set as oracles.
  1. ESOP method.
  2. ESOP method following the RTT method.
  3. TBS method (which requires the RTT method).
- Compare by
  1. Circuit complexity.
  2. Time-to-synthesis.
  3. Qubit count.

Oracles for Benchmark Functions: Circuit Complexity

- The sum of the costs of each gate, given by the number of qubits on which a gate acts.
- Circuit complexity, on average.

ESOP No RTT	ESOP with RTT	TBS
3,769.5	6,981.6	15,698.5

- ESOP method alone had the lowest.

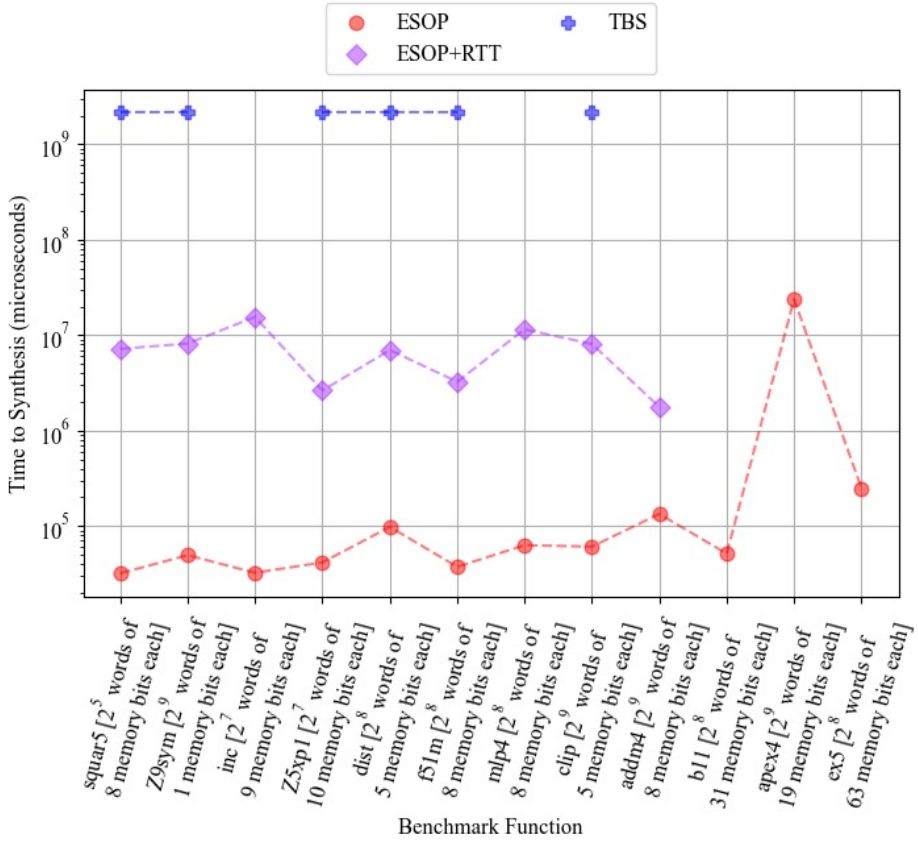


Oracles for Benchmark Functions

- Time-to-synthesis, on average.

ESOP No RTT	ESOP with RTT	TBS
77.2 ms	7,230 ms	2,150,000 ms

- ESOP method alone had the lowest.

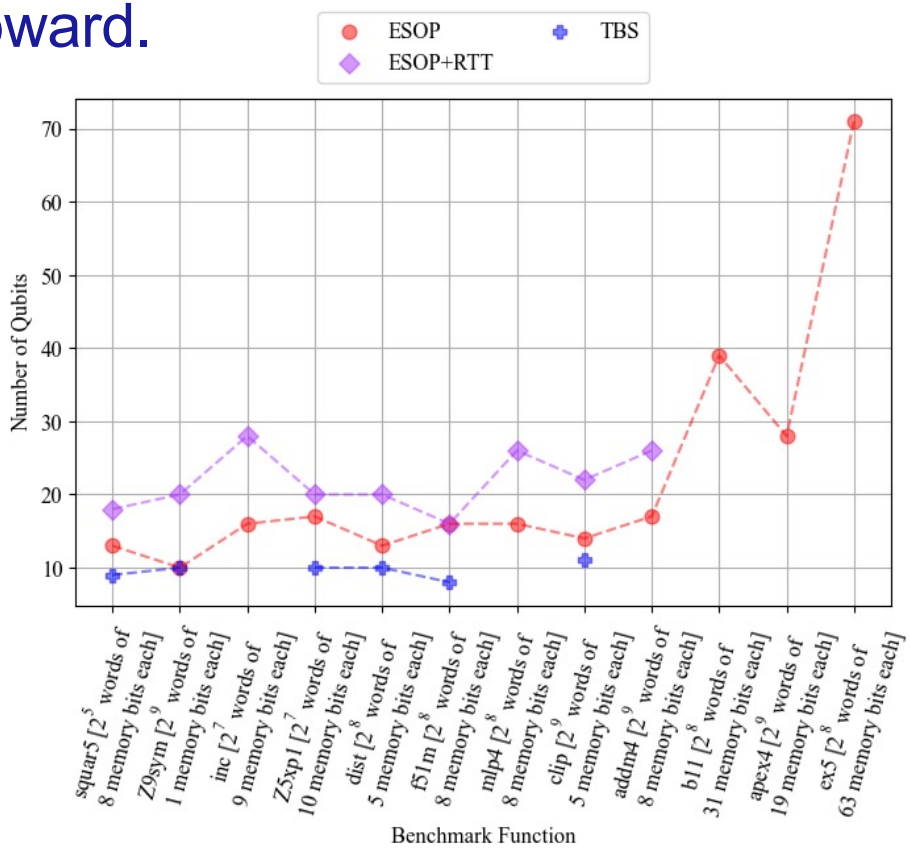


Oracles for Benchmark Functions

- Qubit count, on average, rounding upward.

ESOP No RTT	ESOP with RTT	TBS
23	22	10

- TBS method had the lowest.



Oracle Synthesis Summary

- The TBS and ESOP methods provide different advantages.
  - TBS improves qubit count.
  - ESOP improves circuit complexity and time-to-synthesis.
- Explore modifications to TBS method so that it can take advantage of incompletely specified functions.
- Other ways to explore improvements in time-to-synthesis, qubit count, circuit complexity, etcetera.
  - Different ways of formulating embedding functions.
  - Alternative forms of data encoding.

A. Sinha, E.R. Henderson, J.M. Henderson, and M.A. Thornton. Automated Quantum Oracle Synthesis with a Minimal Number of Qubits, IEEE/ACM Third International Workshop on Quantum Computing Software (QCS), Dallas, TX, 2022. https://ieeexplore.ieee.org/document/10025532. Available via arXiv: https://arxiv.org/abs/2211.09860.

Quantum Random Number Generator (QRNG)

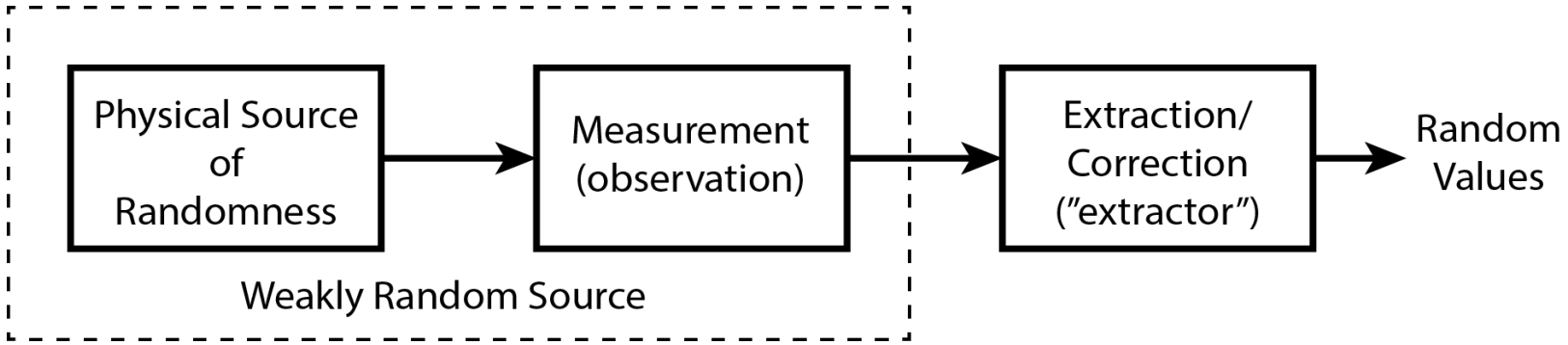
- Random Number Generators are Crucial Elements in Applications such as:
  - Cryptographic Systems and Implementations
  - Use in Monte Carlo Simulation Methods
  - Temporary Key and Password Generators
  - Nonce Generators
  - Prior Distributions to Enhance Performance of Quantum Algorithms such as Grover's Search
- Generating High Quality Random Numbers is a Difficult Problem
- Quantum State Measurement Observes a Variate of a Random Variable
- Quantum Circuits can be Synthesized for Arbitrary Distributions

True versus Pseudo-RNGs

- TRNGs comprise a weakly random source (WRS) and an extractor function
  - WRS is from a natural process
  - Extractor function removes deterministic error from the WRS
  - We focus on the quality and flexibility of generating a weakly random source
- Pseudo-random number generators(PRNGs) are created from deterministic calculations, which can be recreated
  - Thus, in theory, PRNGs are not cryptographically secure for temporary passwords, encryption keys, and cryptographic seed, salt, and nonce values.
- TRNG for non-uniform distributions
  - Generating arbitrary(user-defined and non-parametric) PMFs efficiently
  - Simulating Monte Carlo models
  - Uses in PQC cryptography
  - Prior Distributions in Common QC Algorithms (like Grover's Search)

True Random Number Generator Architecture

- Typically, TRNGs Produce Equally-likely Random Bit Streams



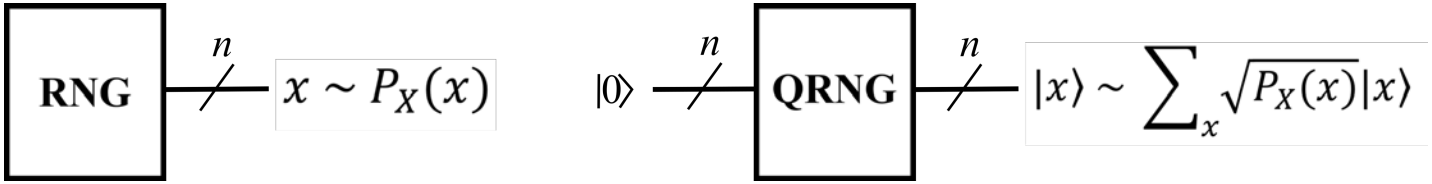
- We use a Quantum Computer as a WRS
  - Enables Programmable PMF, not just Equally-likely (uniform PMF) bits
- Extractor Removes Determinism and Bias to Increase Generated Entropy
  - Can be Viewed as a Random Variable Transformation Analytically

Acronyms

- QC – Quantum Computing
- TRNG – True random number generator
- WRS – Weakly Random Source
- PRNG – Pseudo-random number generator
- PMF – Probability Mass Function
- PQC – Post-quantum cryptography
- QRNG – Quantum random number generator
- OpenQASM – Open Quantum Assembly Language
- KL-Divergence – Kullback-Leibler divergence
- JS-Divergence – Jenson-Shannon divergence

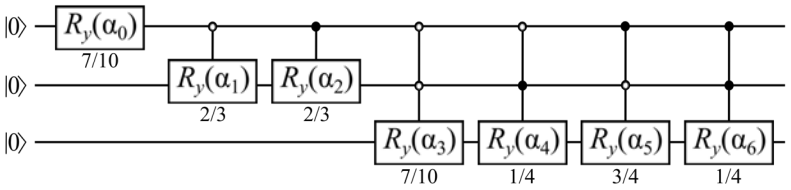
Background: Quantum RNGS

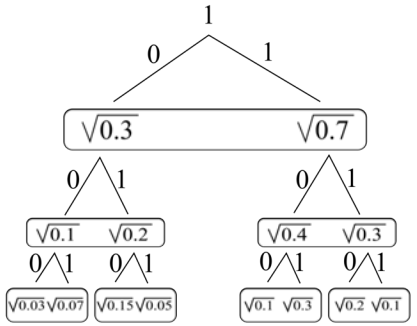
- In QRNGs, randomness is quantum mechanical in nature
- Quantum WRS examples
  - Sources include radioactive decay, semiconductor device noise, photodetector dark counts, and photon paths
- Programmable Gate-model WRS/QRNGs (Our contribution)
  - The QCs are programmable and allow many PMF types
  - Can apply optimizations the QRNG circuit
  - Can map QRNG to a specified quantum hardware



Background: Amplitude Encoding

- Each controlled rotation affects the ratio of the left and right densities of the PMF
- The rotations are recursively calculated by finding the ratio of the left and right subareas of the PMF
- Phases of the states are irrelevant





$$\sqrt{0.03}|000\rangle + \sqrt{0.07}|001\rangle + \sqrt{0.15}|010\rangle + \sqrt{0.05}|011\rangle + \sqrt{0.1}|100\rangle + \sqrt{0.3}|101\rangle + \sqrt{0.2}|110\rangle + \sqrt{0.1}|111\rangle$$

Quantum State Representing a PMF

A. Sinha, E.R. Henderson, J.M. Henderson and M.A. Thornton,Automated Quantum Memory Compilation with Improved Dynamic Range, International Conference for High Performance Computing, Networking, Storage, and Analysis (SC22), International Workshop on Quantum Computing software (QSC22), November 13, 2022, 14 pp.,Available via arXiv, https://arxiv.org/pdf/2211.09860.pdf.

Contribution: Flex-QRNG

- Flex-QRNG is a programmable tool to output a user-defined non-parametric or parametric distribution as a quantum state
- Automate the procedure of creating QRNG circuits
  - Specify our PMF in a tabular form containing fixed point values
  - N qubits correspond to 2^N PMF bins
  - N qubits also means the numbers will have a word size of N
- Automate mapping to hardware and optimizations
  - Makes it easy for a QC developer to obtain QRNG results without manual hardware mapping and optimization

A. Sinha, E.R. Henderson, J.M. Henderson, E.C. Larson and M.A. Thornton, A Programmable True Random Number Generator using Commercial Quantum Computers, SPIE 12517, Quantum Information Science, Sensing, and Computation XV, April 30-May 4, 2023, 1251705, Available via arXiv: https://arxiv.org/pdf/2304.03830.pdf

Contribution: Flex-QRNG Design Overview

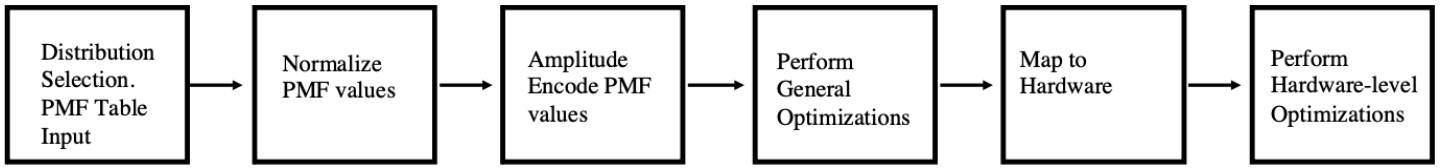
- Frontend – receives specification of the PMF function and number of qubits to generate the circuit with hardware-independent optimizations

- Backend – maps circuit to hardware-specific gate libraries of a commercially available gate model QC, then applies hardware-dependent optimizations. Can easily be extended to target other platforms such as transmon, ion-trap, or photonic computers.

- Final Output – output OpenQASM file containing native gate operations that represent the WRS/QRNG

Contribution: Flex-QRNG Compilation Process

1. User specifies distribution (PMF table and number of qubits)
2. Perform Amplitude Normalization on the PMF bins
3. Amplitude encode the PMF bins into a quantum state
4. Perform High-level optimizations using a compiler
5. Map to a hardware backend
6. Perform Hardware level optimizations
7. Run Measurements on QRNG



Experimental Results: Synthesis of Binomial PMF

- Create amplitudes for a binomial PMF take coefficients of polynomial $(x+y)^N$ where $N=5$

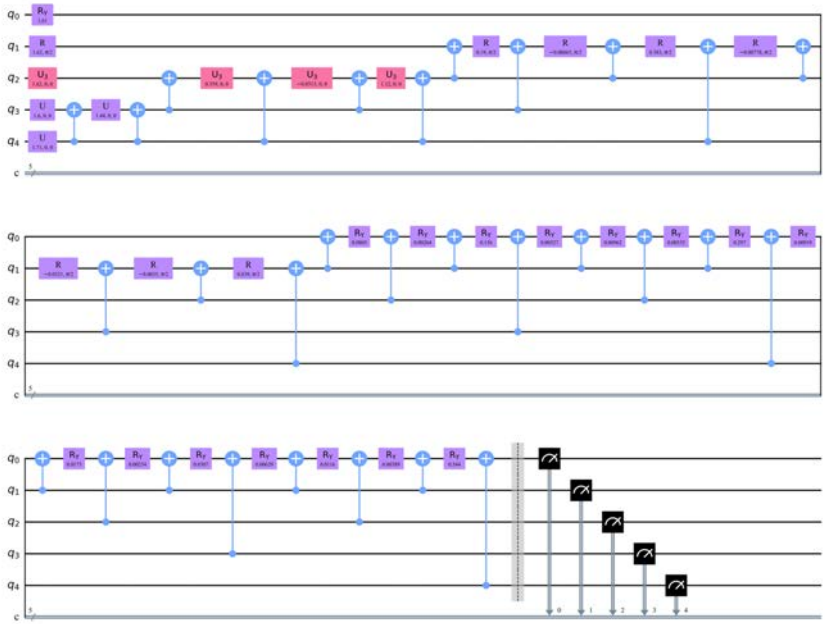
$$\text{binomial} = [1,\ 31,\ 465,\ 4495,\ 31465,\ 169911,\ 736281,\ 2629575,\ 7888725,\ 20160075,\ 44352165,\ 84672315,\ 141120525,\ 206253075,\ 265182525,\ 300540195,\ 300540195,\ 265182525,\ 206253075,\ 141120525,\ 84672315,\ 44352165,\ 20160075,\ 7888725,\ 2629575,\ 736281,\ 169911,\ 31465,\ 4495,\ 465,\ 31,\ 1]$$

- Normalize amplitudes (divide by Euclidian norm)

$$\text{binomial normalized} = [1.4658E-09,\ 4.54397E-08,\ 6.81595E-07,\ 6.58875E-06,\ 4.61213E-05,\ 0.000249055,\ 0.001079238,\ 0.003854421,\ 0.011563263,\ 0.029550561,\ 0.065011233,\ 0.124112355,\ 0.206853925,\ 0.302324967,\ 0.388703529,\ 0.440530666,\ 0.440530666,\ 0.388703529,\ 0.302324967,\ 0.206853925,\ 0.124112355,\ 0.065011233,\ 0.029550561,\ 0.011563263,\ 0.003854421,\ 0.001079238,\ 0.000249055,\ 4.61213E-05,\ 6.58875E-06,\ 6.81595E-07,\ 4.54397E-08,\ 1.4658E-09]$$

- Amplitude encode

Amplitude Encoded PMF bins



Experimental Method: Simulation Setup

- Parametric distributions: uniform, triangle, and binomial PMFs
- Non-parametric PMFs: bimodal and arbitrary distributions

- Hardware: 5 qubits used IBM Washington Backend Simulator
- Synthesis efficiency metrics:
  - Gate Count
  - Quantum Depth

- PMF Distribution Quality Metrics (next slide)

Experimental Methods: Quality Metrics

- We measure representation quality by comparing our generated PMF to the ideal PMF
  - The ideal distribution (P) is originally defined in the fixed-point values per PMF bin
  - The sampled distribution (Q) contains counts of each measured quantum state normalized by the number of measurements

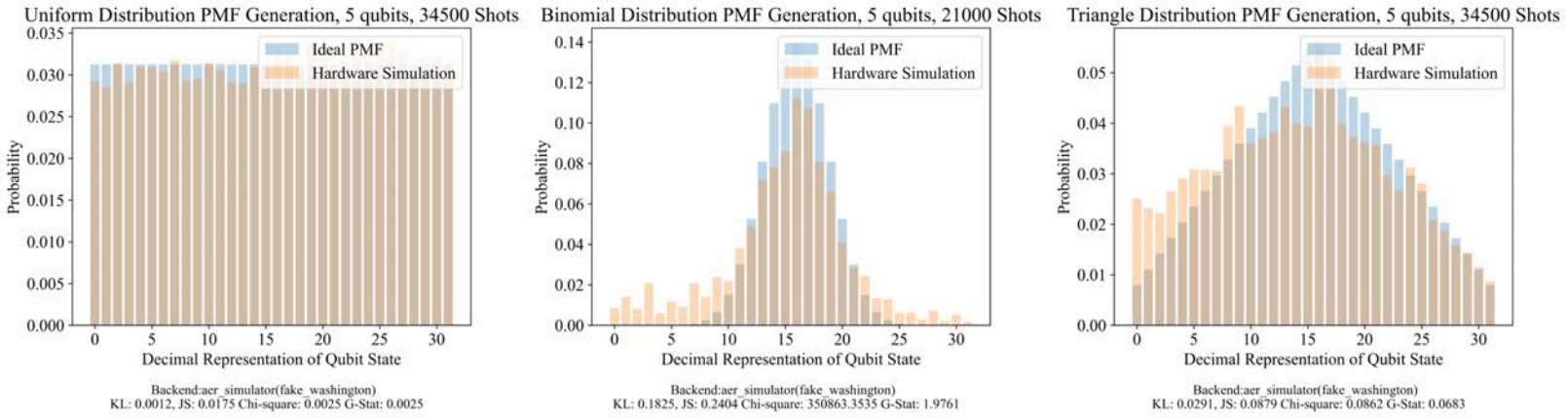
- Quality Metrics
  - KL divergence
  - G-stat
  - JS divergence

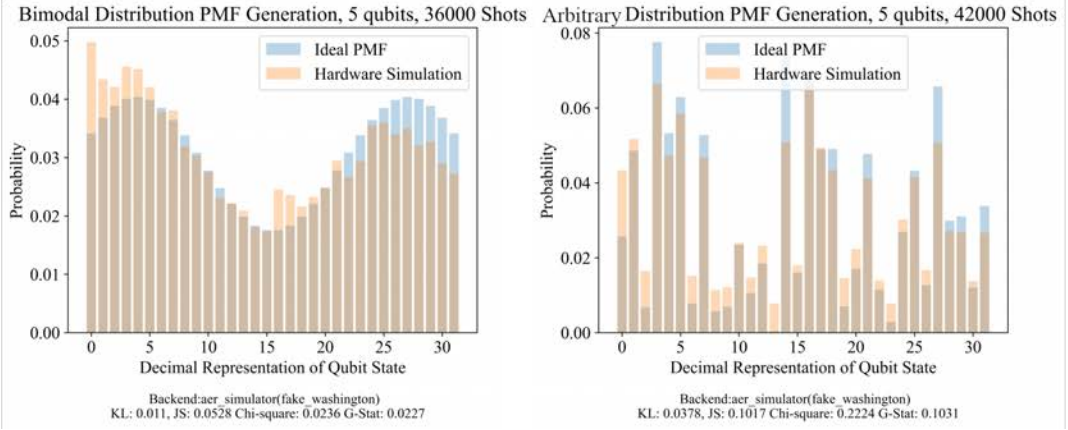
$$\mathrm{KL}(P\|Q) = \sum_i P(i) \ln\left(\frac{P(i)}{Q(i)}\right)$$

$$G = 2n \times KL(P\|Q)$$

$$\mathrm{JSD}(P\|Q) = \frac{1}{2}KL(P\|M) + \frac{1}{2}KL(Q\|M) \text{ where } M = \frac{1}{2}(P+Q)$$

Experimental Results: Visualizing Noise





- Using a noisy simulator of the IBM-Q Washington.
- The blue histogram illustrates the ideal PMF input
- The orange histogram illustrates sampled PMF distribution

Experiment Results: Metric Comparison

- Distributions with small tapering probabilities are less accurate
  - Noisy binomial and triangle distributions have the lowest p-values
  - Bimodal and arbitrary distributions have higher p-values.
- This can be caused by reduced precision of rotations and control of probabilities in real hardware simulations.

	KL-divergence	JS	G-stat	G-stat P-Value	Gates	Depth
Uniform	0.0012	0.0175	0.0025	0.9601224	5	1
Binomial	0.1825	0.2404	1.9761	0.1598019	61	57
Triangle	0.0291	0.0879	0.0683	0.7938283	61	57
Bimodal Non-Parametric	0.011	0.0528	0.0227	0.8802398	61	57
Arbitrary Non-Parametric	0.0378	0.1017	0.1031	0.7481408	61	57

Table 2. Statistical comparison of Flex-QRNG circuits and resulting distributions for five different PMFs when simulated on IBM-Q's Washington device.

SMU has Two Faculty Openings for Quantum Informatics

- Both are Tenure-track; CS or ECE or Joint Appointments
- Endowed Position and Intro. Asst. Prof. Position

https://www.smu.edu/Lyle/Departments/Faculty-Openings

- Contact me for more Information: mitch@smu.edu
- Rhines Professor of Quantum Informatics (Assoc./Full)
- TT Asst. Prof. Specializing in Quantum Engineering
- Launching M.S. in Quantum Engineering in Fall'24 Semester

Please Consider Joining Our Research Group at SMU in Dallas, Texas

Summary and Q&A

- As the Number of Available Qubits Increases, QC Users will need to Rely upon Automated Tools for Algorithm Development
- Automated Oracle Synthesis Enables Arbitrary Functions to be Implemented as Reversible and Unitary Transformations
- An Example Use-Case for a Programmable QRNG is Described

Thanks for your Attention

Questions????