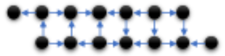
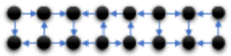
Varies micro-architechtural noise properties based on qubit, control technology, device environment

Basic Building Block: Qubit Operations: Gate

Current QC: 5-20 qubit ops, 49-72 qubits in test

| Machine | Qubits | 2Q Gates | Coherence Time (us) | 1Q Error (%) | 2Q Error (%) | RO Error (%) | Qubit Topology |
|---|---|---|---|---|---|---|---|
| IBM Q5 Tenerife | 5 | 6 | 40 | 0.2 | 4.76 | 6.21 |  |
| IBM Q14 Melbourne | 14 | 18 | 30 | 1.19 | 7.95 | 9.09 |  |
| IBM Q16 Rüschlikon | 16 | 22 | 40 | 0.22 | 7.14 | 4.15 |  |
| Rigetti Agave | 4 | 3 | 15 | 3.68 | 10.8 | 16.37 |  |
| Rigetti Aspen1 | 16 | 18 | 20 | 3.43 | 8.92 | 5.56 |  |
| Rigetti Aspen3 | 16 | 18 | 20 | 3.79 | 5.37 | 6.65 | |
| UMD Trapped Ion (UMDTI) | 5 | 10 | $1.5 \times 10^6$ | 0.2 | 1.00 | 0.6 |  |

# Design Decisions for Architecting a QC System

**Software-visible gate set**
Should SW-gates be unified across implementations or be tailored to the device characteristics? Should we expose native gates or higher-level composite gates?

**Connectivity**
What qubit topology is the best for a set of target applications? How do topology changes affect program success rate?

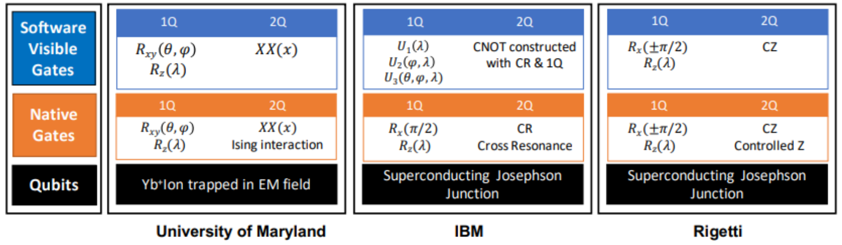
**Noise**
How does variable qubit noise affect program success rate? Are small variations in noise important?

Full-Stack, Real-System Quantum Computer Studies: Architectural Comparisons and Design Insights (International Symposium on Computer Architecture 2019)

presented by Hanhan Zhou

| Machine | Coherence Time (us) | 2Q Error (%) | RO Error (%) | Noise Variation for 2Q ops |
|---|---|---|---|---|
| IBMQ5 | 40 | 4.76 | 6.21 | Up to 9X |
| Rigetti Agave | 15 | 10.8 | 16.37 | Up to 9X |
| UMD Ion Trap | $1.5 \times 10^6$ | 1.00 | 0.6 | Up to 3X |

**Figure 1: Characteristics of the devices used in our study. Each device has different qubit and gate count (higher is better), coherence time (higher is better), error rates (lower is better) and topology (dense connectivity is better). Rigetti Agave has 8 qubits in a ring topology, but only 4 qubits were available during our study.**



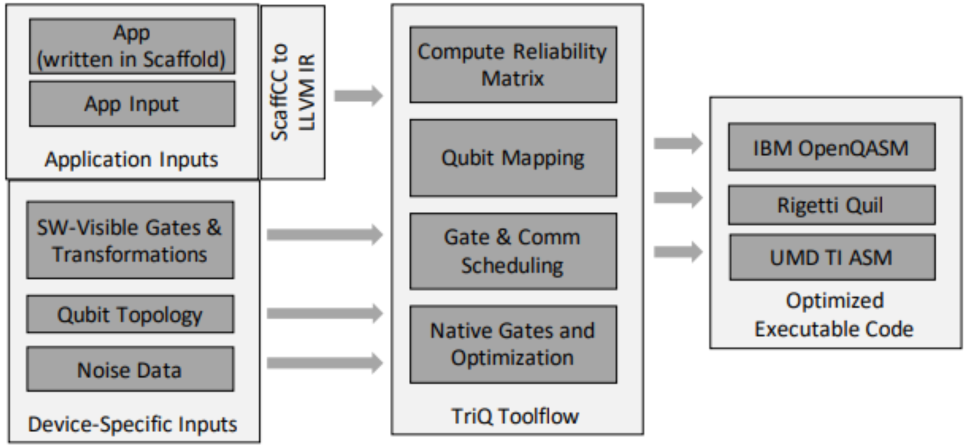




**Figure 4: Overview of the TriQ toolflow. Input to TriQ consists of high-level Scaffold programs and their inputs, as well as device-specific QC system properties. Output is optimized code in one of three vendor-specific executable formats.**

- Result1

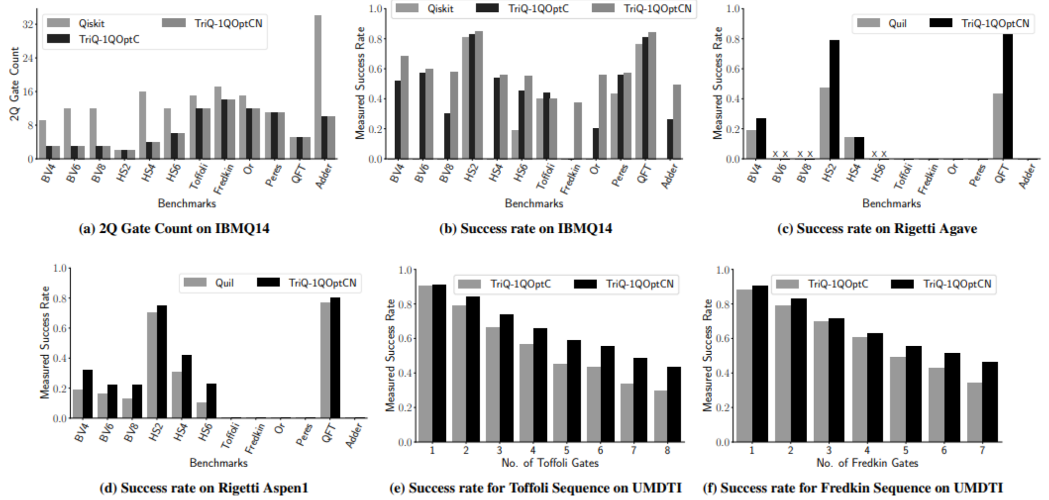



**Figure 11: Importance of noise-adaptivity. (a) and (b) compare TriQ-1QOptC, TriQ-1QOptCN and IBM Qiskit compiler for IBMQ14. By optimizing gate errors, communication and single qubit gates simultaneously, TriQ-1QOptCN obtains up to 28x improvement over Qiskit and 2.8x over TriQ-1QOptC. (c) and (d) compare Rigetti's Quil compiler and TriQ-1QOptCN on Rigetti Agave and Aspen1. TriQ-1QOptCN obtains up to 2.3x improvement over Quil. (e) and (f) compare TriQ-1QOptC and TriQ-1QOptCN on UMDTI, where noise-adaptivity provides up to 1.47x improvement. In (b), (c) and (d) runs with zero height bars correspond to failed runs where the correct answer did not dominate in the output distribution.**

Conclusion

NISQ QC prototypes are now available for experiments

To study key system design questions, the work built TriQ, a top-to-bottom toolflow which compiles high-level language programs for multiple target systems.

TriQ show several examples of how leveraging hardware details in the compiler can provide a significant boost in program success rates.

Insights:

Expose native gates to software better for optimization

QC is not yet ready for device-independent ISA abstraction

Machine topology matters

Compilers can mitigate hardware connectivity limitation

# My understanding

# Make use of Noisy Intermediate-Scale Quantum

# All stack?

5, 14 and 16 qubits superconducting, 5-qubit trapped ion

# Security Problems?