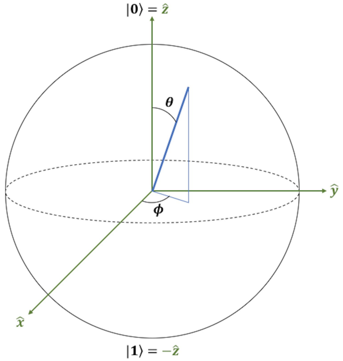
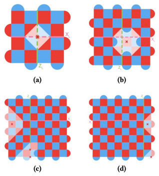
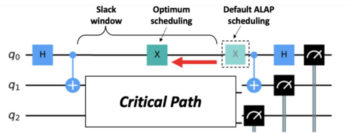
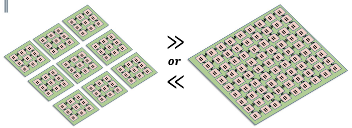
Theory

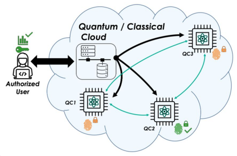


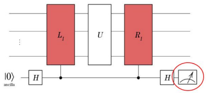
Architecturally-aware Optimization



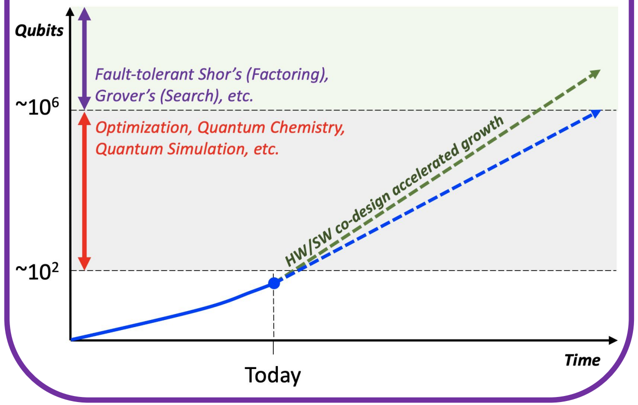








Path to Quantum Utility



Overview of Quantum Software and Compilers

2024 SIAM Quantum Intersections Convening

October 9, 2024

Kaitlin N. Smith
Assistant Professor, Computer Science
Northwestern University

Outline

- Primer on Quantum Software
- The Role of Quantum Compilers
- Future Directions for the Quantum Stack

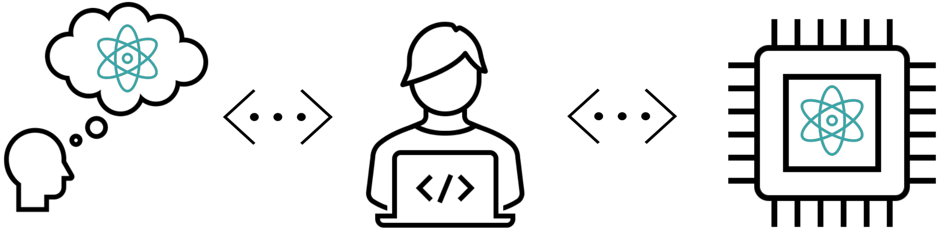
Outline

- **Primer on Quantum Software**
- The Role of Quantum Compilers
- Future Directions for the Quantum Stack

Why Quantum Software?

For wider adoption, tools are required to make quantum algorithms and hardware more accessible. Quantum software helps with both!

- Provides languages and abstractions to reason about algorithms and verify logic
- Translates and optimizes information from high-level specifications to low level representations that are compatible with hardware
- Creates models that improve our understanding of quantum systems
- Accelerates research



Development of Quantum Programming Toolchains

First programming languages from academia, 2000 – mid 2010s

- Quantum Computation Language (QCL),
- Quipper: A scalable quantum programming language
- ScaffCC: a framework for compilation and analysis of quantum computing programs

Software emerges from industry, late 2010s

Qiskit

PENNYLANE

rigetti

Cirq

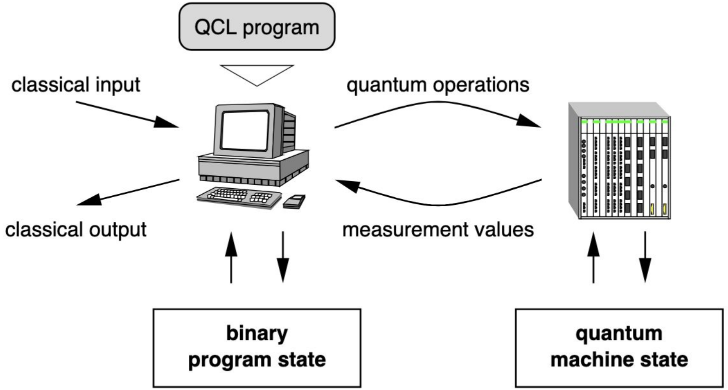


Figure 3.1: The hybrid architecture of QCL

Figure: http://www.itp.tuwien.ac.at/~oemer/doc/quprog.pdf

Quantum Software Explosion, Late 2010s - Today

Many packages have emerged that are highly specialized or focus on points in the compute stack.

- Circuit optimization
- Gate synthesis
- Error mitigation
- Resource estimation
- Circuit / device simulation
- ...and more!

CLASSIQ

Stim

TKET

QuEST

BQSKit

Superstaq

NVIDIA
CUDA QUANTUM

Ocean

Torch Quantum

QCOR
QUANTUM COMPILER

BLOQADE

QuTiP
Quantum Toolbox in Python

mitiq

Q-CTRL

Amazon Braket

QERMIT

QIBO

MUNICH QUANTUM TOOLKIT

While this is great progress, quantum application and device development has outpaced the middleware!

WIP: The Quantum Software Stack

Layers involved in classical computing circa 1950s vs current classical computing and quantum tool flows

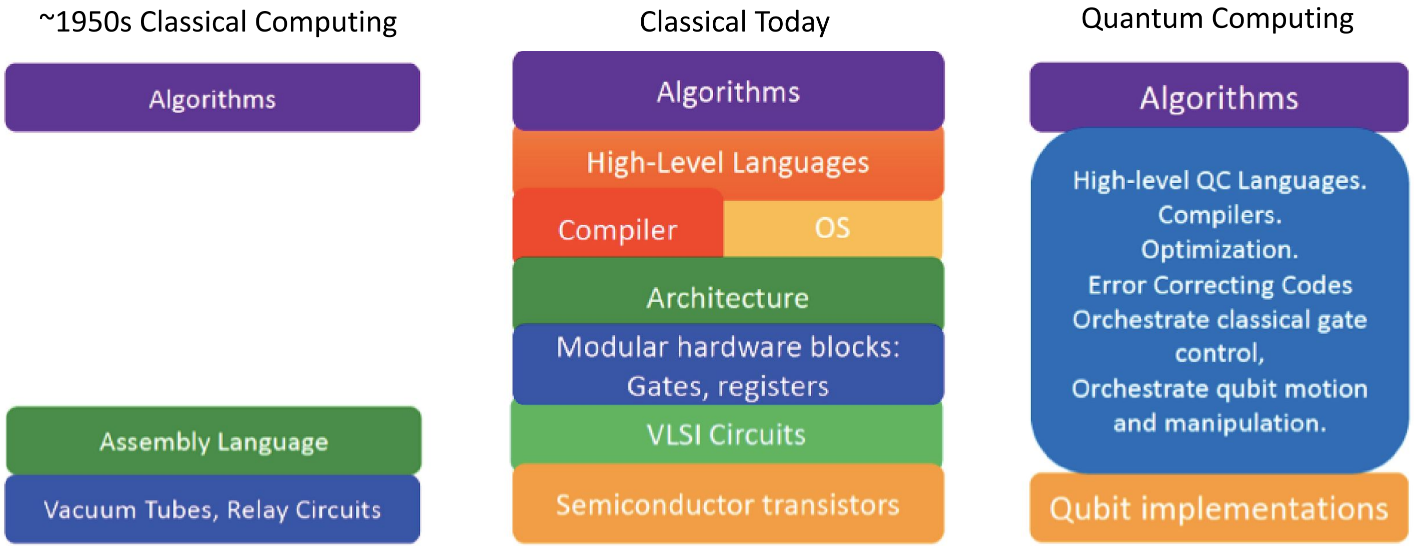


Figure: https://arxiv.org/abs/1903.10541

Software Stack Research

- Device compilation
- Logic and gate synthesis
- Programming languages
- Quantum intermediate representations
- Resource estimation
- Error mitigation
- Optimal control
- Interfaces with classical hardware
- Heterogeneous system interfacing
- Architecture and security in the quantum cloud
- Hybrid HPC ecosystem management

Quantum Computing Stack

Algorithms

High-level QC Languages.
Compilers.
Optimization.
Error Correcting Codes
Orchestrate classical gate control,
Orchestrate qubit motion and manipulation.

Qubit implementations

Additional Examples of Quantum Software

- Circuit simulators
- Device design and analysis tools
- Device characterization
- Benchmarking suites

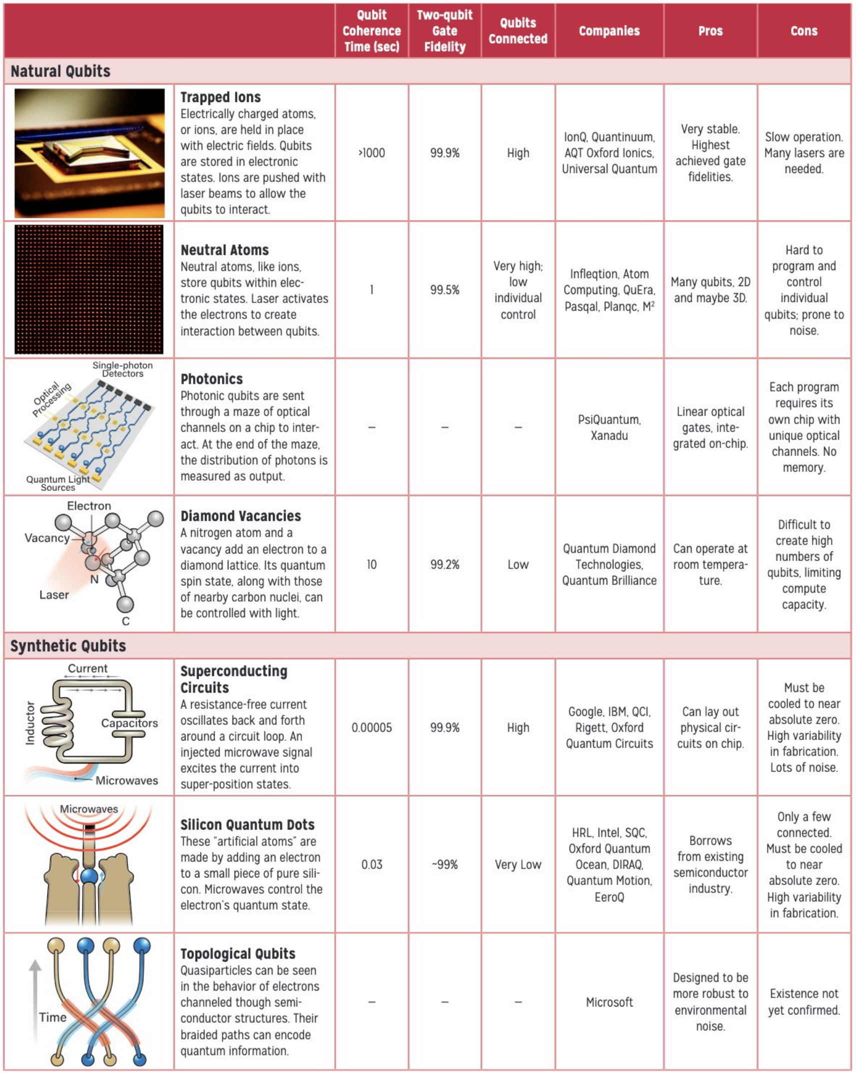
Outline

- Primer on Quantum Software
- **The Role of Quantum Compilers**
- Future Directions for the Quantum Stack

Today’s Quantum Hardware

- **Scale:** < ~1000 qubits per device….millions are needed for fault-tolerance!
    - ***Fault-tolerance*** – quantum computation where many ***physical*** qubits encode a single ***logical*** qubit

- **Architectural constraints:** Error rates, coherence windows, gate sets, operation time, qubit connectivity, etc.
    - High variation between qubit technologies
    - Extremely low temperatures / high isolation often required



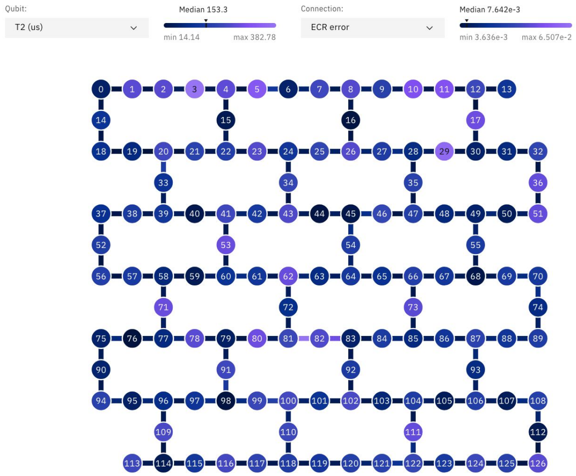
	Qubit Coherence Time (sec)	Two-qubit Gate Fidelity	Qubits Connected	Companies	Pros	Cons
Natural Qubits						
Trapped Ions Electrically charged atoms, or ions, are held in place with electric fields. Qubits are stored in electronic states. Ions are pushed with laser beams to allow the qubits to interact.	>1000	99.9%	High	IonQ, Quantinuum, AQT Oxford Ionics, Universal Quantum	Very stable. Highest achieved gate fidelities.	Slow operation. Many lasers are needed.
Neutral Atoms Neutral atoms, like ions, store qubits within electronic states. Laser activates the electrons to create interaction between qubits.	1	99.5%	Very high; low individual control	Infleqtion, Atom Computing, QuEra, Pasqal, Planqc, M^2	Many qubits, 2D and maybe 3D.	Hard to program and control individual qubits; prone to noise.
Photonics Photonic qubits are sent through a maze of optical channels on a chip to interact. At the end of the maze, the distribution of photons is measured as output.	–	–	–	PsiQuantum, Xanadu	Linear optical gates, integrated on-chip.	Each program requires its own chip with unique optical channels. No memory.
Diamond Vacancies A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.	10	99.2%	Low	Quantum Diamond Technologies, Quantum Brilliance	Can operate at room temperature.	Difficult to create high numbers of qubits, limiting compute capacity.
Synthetic Qubits						
Superconducting Circuits A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into super-position states.	0.00005	99.9%	High	Google, IBM, QCI, Rigett, Oxford Quantum Circuits	Can lay out physical circuits on chip.	Must be cooled to near absolute zero. High variability in fabrication. Lots of noise.
Silicon Quantum Dots These “artificial atoms” are made by adding an electron to a small piece of pure silicon. Microwaves control the electron’s quantum state.	0.03	~99%	Very Low	HRL, Intel, SQC, Oxford Quantum Ocean, DIRAQ, Quantum Motion, EeroQ	Borrows from existing semiconductor industry.	Only a few connected. Must be cooled to near absolute zero. High variability in fabrication.
Topological Qubits Quasiparticles can be seen in the behavior of electrons channeled though semiconductor structures. Their braided paths can encode quantum information.	–	–	–	Microsoft	Designed to be more robust to environmental noise.	Existence not yet confirmed.

Figure: https://arxiv.org/abs/2403.08780

Today’s Quantum Hardware

- **Heterogeneity:** Significant variation between devices, even of same base technology.
- **Quality:** Noise prevents large circuit widths and depths in quantum programs.

IBM Brisbane



Qubit: T2 (us)

Median 153.3

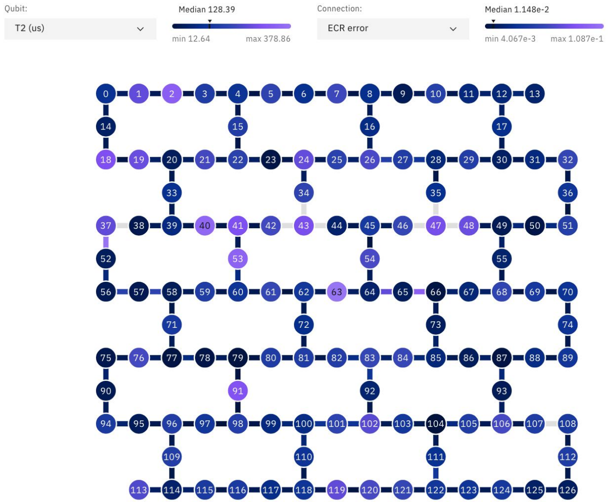
min 14.14 max 382.78

Connection: ECR error

Median 7.642e-3

min 3.636e-3 max 6.507e-2

IBM Nazca



Qubit: T2 (us)

Median 128.39

min 12.64 max 378.86

Connection: ECR error

Median 1.148e-2

min 4.067e-3 max 1.087e-1

The Role of the Compiler

Compilation involves translation of source code into a machine-ready specification in a way that preserves program semantics. Important considerations include:

- Supported hardware instructions
- Rules of the hardware (connectivity, operator time, etc.)
- Optimization rules

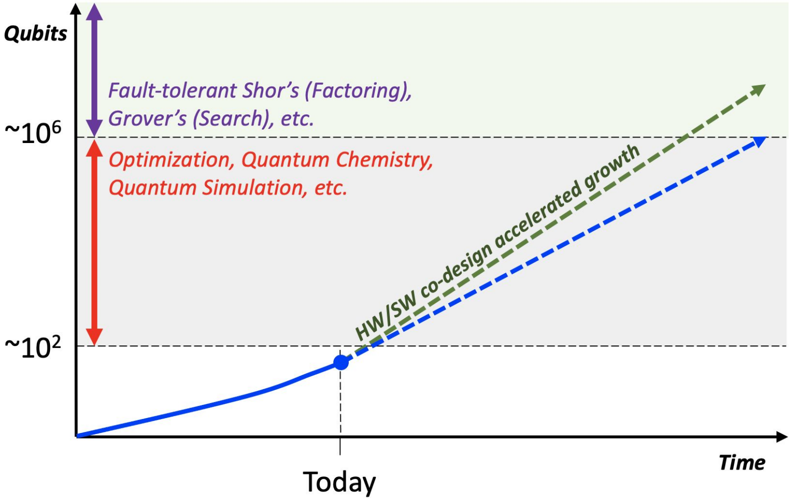
Motivating Quantum Compiler Research

1. Reliability and architectural constraints block widespread adoption of quantum technologies.

   How do we adapt programs to intrinsic QC properties?

2. System improvements via hardware alone will lead to unacceptable scaling timelines.

   Two-qubit (transmon) gate error has lowered by ~0.77x per year,[1] but the rate at which systems improve must be accelerated!

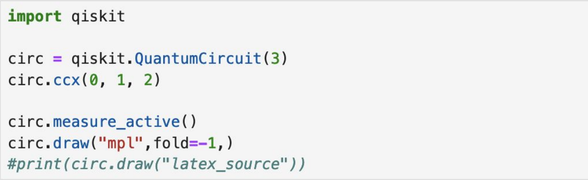


Let's consider a simplified compiler pipeline including ***gate decomposition***, ***placement***, ***routing***, *and* ***low-level optimization****!*

[1] arXiv:2005.02464

Gate Decomposition

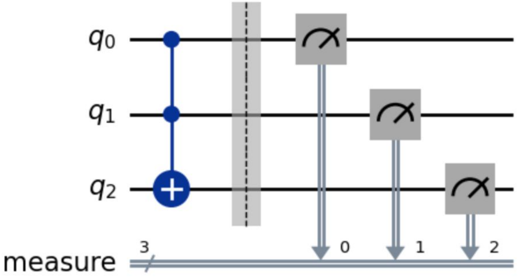
Translate multi-qubit gates into the supported gate set with unitary synthesis

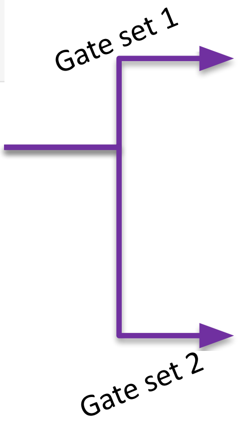


```
import qiskit

circ = qiskit.QuantumCircuit(3)
circ.ccx(0, 1, 2)

circ.measure_active()
circ.draw("mpl",fold=-1,)
#print(circ.draw("latex_source"))
```

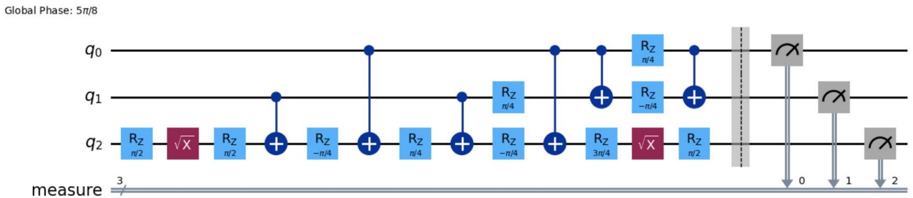




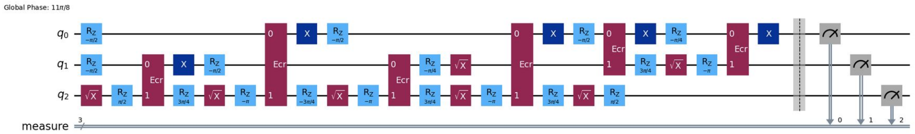
Gate set 1

Gate set 2

```
qiskit.transpile(circ,basis_gates = ["x","sx","rz","cx"]).draw("mpl",fold=-1,)
```



```
qiskit.transpile(circ,basis_gates = ["x","sx","rz","ecr"]).draw("mpl",fold=-1,)
```



Placement and Routing

- Noise aware mapping to assign logical qubits to best regions on-chip
  - Consider: errors, coherence, gate time, etc.
- Inject SWAP operations to make quantum circuit agree with backend connectivity
  - Minimize to prevent additional gate error

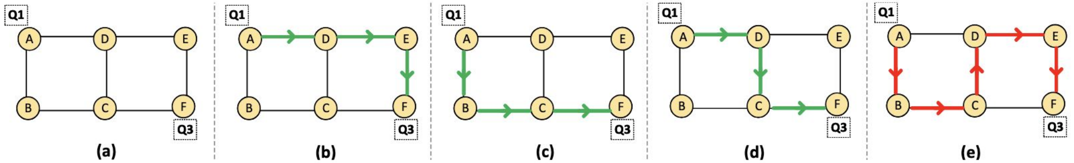


Fig. 4: (a) Layout of a 6-qubit quantum computer, (b)-(e) are possible routes from A to F. Note that options (b)(c)(d) have identical number of swaps and (e) incurs higher swaps. An intelligent policy would choose one from (b)(c)(d).

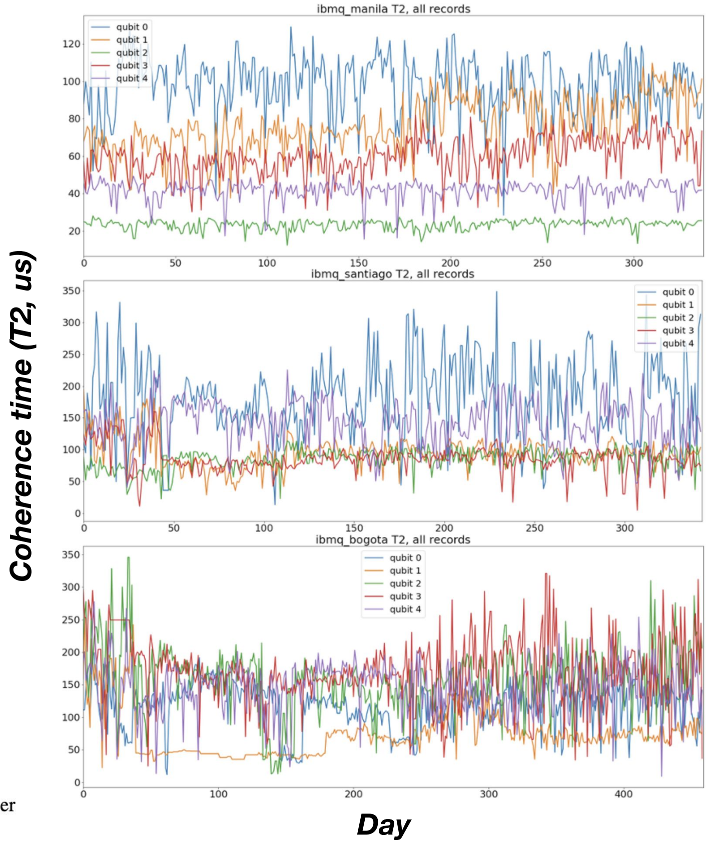
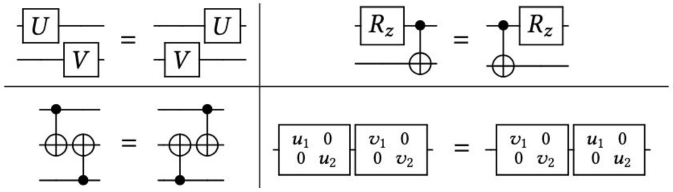


Figure: https://arxiv.org/abs/1805.10224

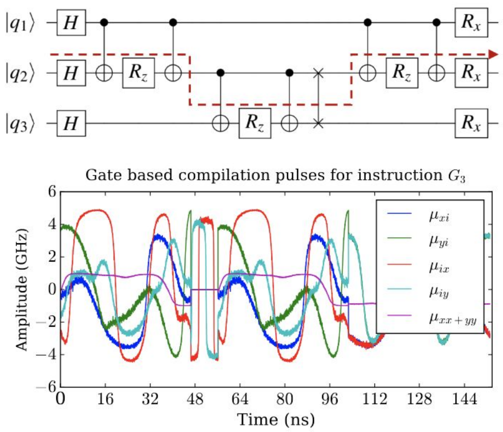
Low-level Optimization

- Gate cancellation with commutativity rules
- Pulse-level optimization

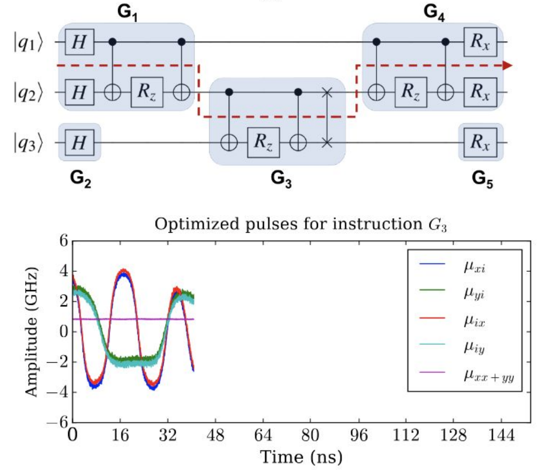
Gate Commutation Relations



Standard QAOA Circuit Compilation



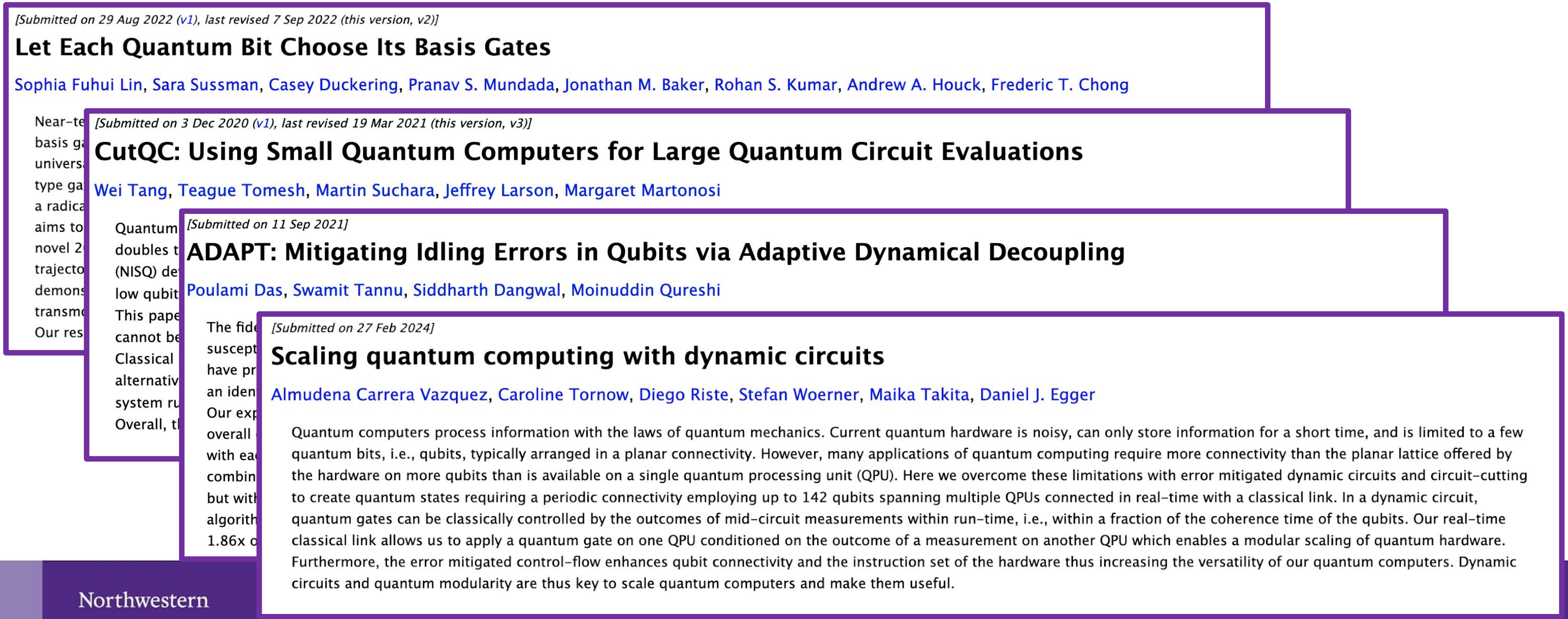
Aggregated Instruction QAOA Circuit Compilation



Figures: https://arxiv.org/abs/1902.01474

Additional Complexity…

- Many stages of compilation are stochastic / heuristic to minimize cost function
- Hardware and applications might need other special considerations



[Submitted on 29 Aug 2022 (v1), last revised 7 Sep 2022 (this version, v2)]

Let Each Quantum Bit Choose Its Basis Gates

Sophia Fuhui Lin, Sara Sussman, Casey Duckering, Pranav S. Mundada, Jonathan M. Baker, Rohan S. Kumar, Andrew A. Houck, Frederic T. Chong

Near-te
basis ga
univers
type ga
a radica
aims to
novel 2
trajecto
demons
transmo
Our res

[Submitted on 3 Dec 2020 (v1), last revised 19 Mar 2021 (this version, v3)]

CutQC: Using Small Quantum Computers for Large Quantum Circuit Evaluations

Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, Margaret Martonosi

Quantum
doubles t
(NISQ) de
low qubit
This pape
cannot be
Classical
alternativ
system ru
Overall, t

[Submitted on 11 Sep 2021]

ADAPT: Mitigating Idling Errors in Qubits via Adaptive Dynamical Decoupling

Poulami Das, Swamit Tannu, Siddharth Dangwal, Moinuddin Qureshi

The fide
suscept
have pr
an iden
Our exp
overall
with ea
combin
but with
algorith
1.86x o

[Submitted on 27 Feb 2024]

Scaling quantum computing with dynamic circuits

Almudena Carrera Vazquez, Caroline Tornow, Diego Riste, Stefan Woerner, Maika Takita, Daniel J. Egger

Quantum computers process information with the laws of quantum mechanics. Current quantum hardware is noisy, can only store information for a short time, and is limited to a few quantum bits, i.e., qubits, typically arranged in a planar connectivity. However, many applications of quantum computing require more connectivity than the planar lattice offered by the hardware on more qubits than is available on a single quantum processing unit (QPU). Here we overcome these limitations with error mitigated dynamic circuits and circuit-cutting to create quantum states requiring a periodic connectivity employing up to 142 qubits spanning multiple QPUs connected in real-time with a classical link. In a dynamic circuit, quantum gates can be classically controlled by the outcomes of mid-circuit measurements within run-time, i.e., within a fraction of the coherence time of the qubits. Our real-time classical link allows us to apply a quantum gate on one QPU conditioned on the outcome of a measurement on another QPU which enables a modular scaling of quantum hardware. Furthermore, the error mitigated control-flow enhances qubit connectivity and the instruction set of the hardware thus increasing the versatility of our quantum computers. Dynamic circuits and quantum modularity are thus key to scale quantum computers and make them useful.

Outline

- Primer on Quantum Software
- The Role of Quantum Compilers
- **Future Directions for the Quantum Stack**

The Future of Quantum Systems

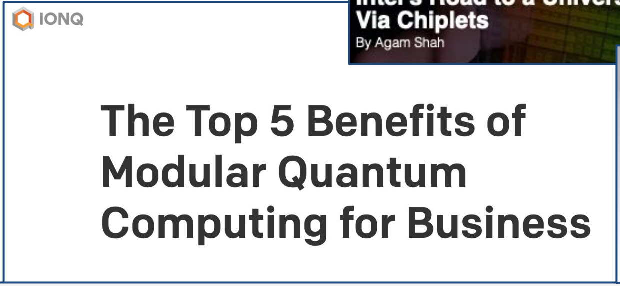
- Modular architectures
- Many technologies
- Quantum clouds
- QEC on the horizon

Continued opportunity for scaling with HW / SW co-design:

1. ***Make the most of existing architecture***
2. ***Guide the design of future infrastructure***

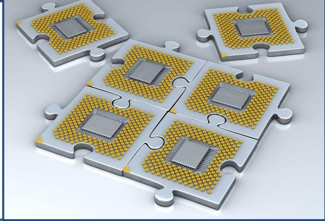
Intel's Road to a Universal Quantum Computer Is Via Chiplets

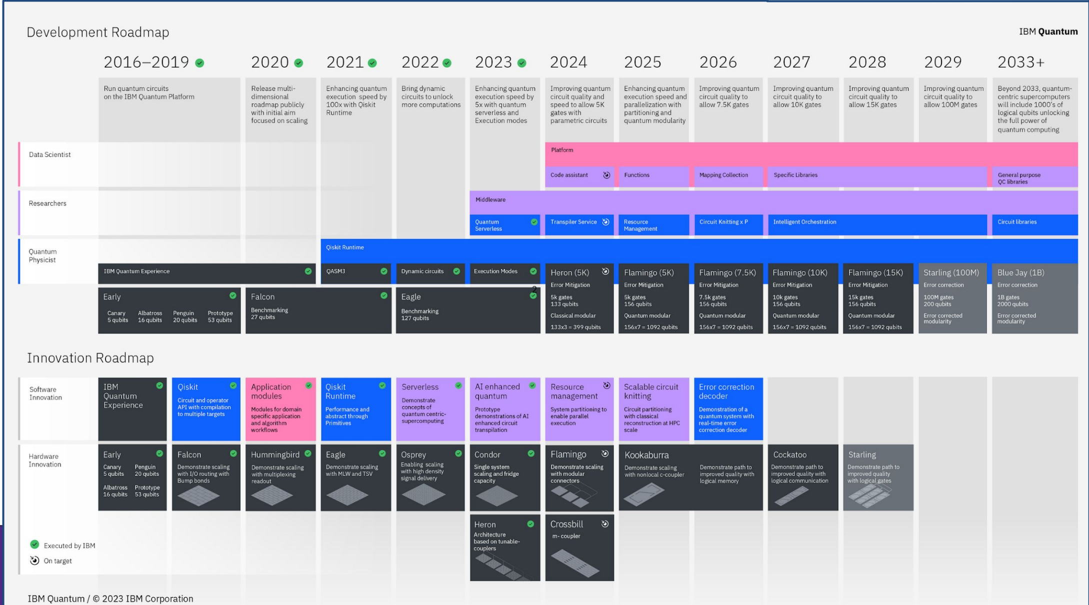
By Agam Shah



IONQ

The Top 5 Benefits of Modular Quantum Computing for Business





Development Roadmap

IBM **Quantum**

2016–2019 ✓	2020 ✓	2021 ✓	2022 ✓	2023 ✓	2024	2025	2026	2027	2028	2029	2033+
Run quantum circuits on the IBM Quantum Platform	Release multi-dimensional roadmap publicly with initial aim focused on scaling	Enhancing quantum execution speed by 100x with Qiskit Runtime	Bring dynamic circuits to unlock more computations	Enhancing quantum execution speed by 5x with quantum serverless and Execution modes	Improving quantum circuit quality and speed to allow 5K gates with parametric circuits	Enhancing quantum execution speed and parallelization with partitioning and quantum modularity	Improving quantum circuit quality to allow 7.5K gates	Improving quantum circuit quality to allow 10K gates	Improving quantum circuit quality to allow 15K gates	Improving quantum circuit quality to allow 100M gates	Beyond 2033, quantum-centric supercomputers will include 1000's of logical qubits unlocking the full power of quantum computing

Data Scientist

Platform

Code assistant — Functions — Mapping Collection — Specific Libraries — General purpose QC libraries

Researchers

Middleware

Quantum Serverless ✓ — Transpiler Service — Resource Management — Circuit Knitting x P — Intelligent Orchestration — Circuit libraries

Quantum Physicist

Qiskit Runtime

IBM Quantum Experience ✓ — QASM3 ✓ — Dynamic circuits ✓ — Execution Modes ✓

Early ✓
Canary 5 qubits
Albatross 16 qubits
Penguin 20 qubits
Prototype 53 qubits

Falcon ✓
Benchmarking 27 qubits

Eagle ✓
Benchmarking 127 qubits

Heron (5K)	Flamingo (5K)	Flamingo (7.5K)	Flamingo (10K)	Flamingo (15K)	Starling (100M)	Blue Jay (1B)
Error Mitigation	Error Mitigation	Error Mitigation	Error Mitigation	Error Mitigation	Error correction	Error correction
5k gates 133 qubits	5k gates 156 qubits	7.5k gates 156 qubits	10k gates 156 qubits	15k gates 156 qubits	100M gates 200 qubits	1B gates 2000 qubits
Classical modular	Quantum modular	Quantum modular	Quantum modular	Quantum modular	Error corrected modularity	Error corrected modularity
133x3 = 399 qubits	156x7 = 1092 qubits	156x7 = 1092 qubits	156x7 = 1092 qubits	156x7 = 1092 qubits		

Innovation Roadmap

Software Innovation

- IBM Quantum Experience ✓
- Qiskit ✓ — Circuit and operator API with compilation to multiple targets
- Application modules ✓ — Modules for domain specific application and algorithm workflows
- Qiskit Runtime ✓ — Performance and abstract through Primitives
- Serverless ✓ — Demonstrate concepts of quantum centric-supercomputing
- AI enhanced quantum ✓ — Prototype demonstrations of AI enhanced circuit transpilation
- Resource management — System partitioning to enable parallel execution
- Scalable circuit knitting — Circuit partitioning with classical reconstruction at HPC scale
- Error correction decoder — Demonstration of a quantum system with real-time error correction decoder

Hardware Innovation

- Early ✓ — Canary 5 qubits, Penguin 20 qubits, Albatross 16 qubits, Prototype 53 qubits
- Falcon ✓ — Demonstrate scaling with I/O routing with Bump bonds
- Hummingbird ✓ — Demonstrate scaling with multiplexing readout
- Eagle ✓ — Demonstrate scaling with MLW and TSV
- Osprey ✓ — Enabling scaling with high density signal delivery
- Condor ✓ — Single system scaling and fridge capacity
- Heron ✓ — Architecture based on tunable-couplers
- Flamingo — Demonstrate scaling with modular connectors
- Crossbill — m- coupler
- Kookaburra — Demonstrate scaling with nonlocal c-coupler — Demonstrate path to improved quality with logical memory
- Cockatoo — Demonstrate path to improved quality with logical communication
- Starling — Demonstrate path to improved quality with logical gates

✓ Executed by IBM

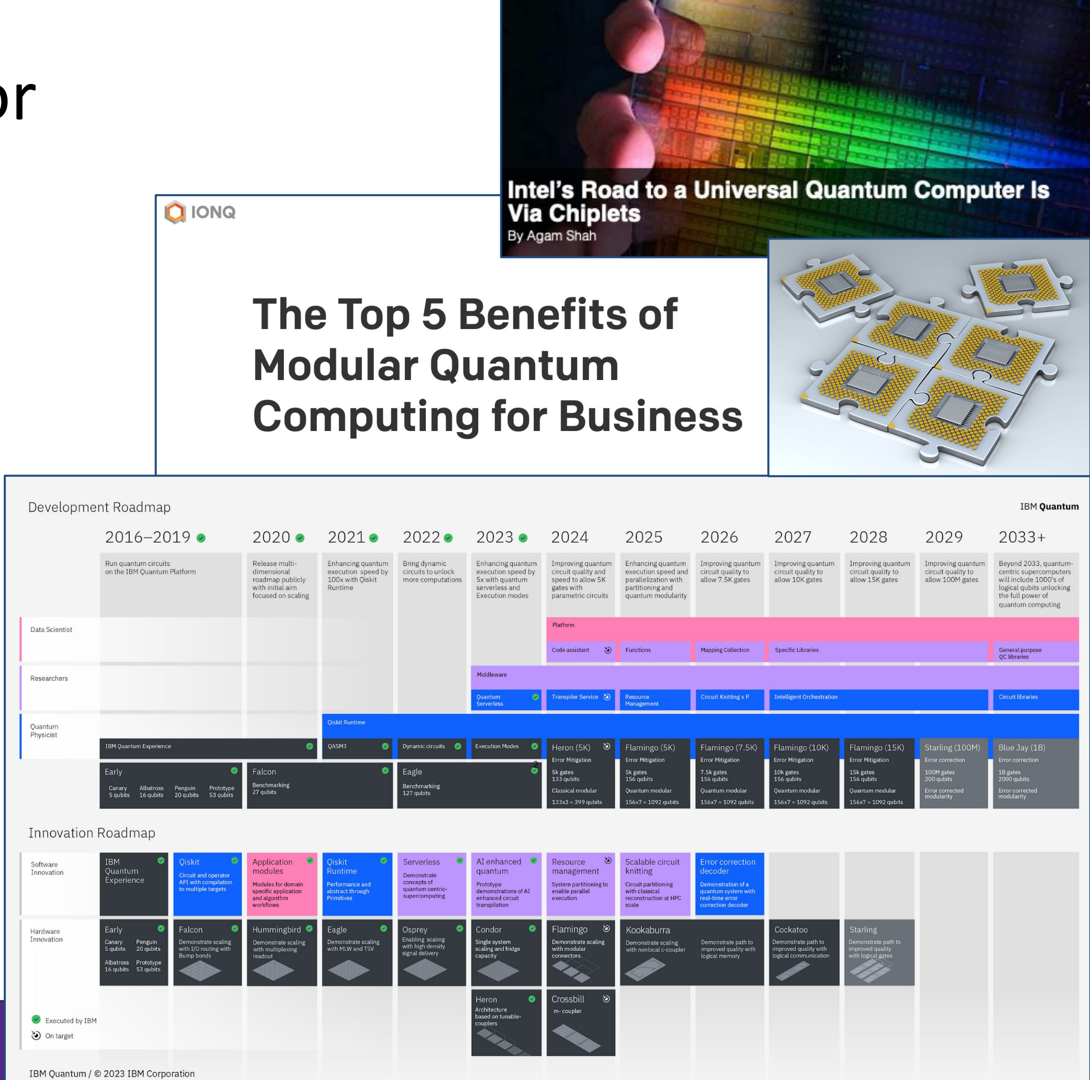
On target

IBM Quantum / © 2023 IBM Corporation

Image: https://newsroom.ibm.com/2023-12-04-IBM-Debuts-Next-Generation-Quantum-Processor-IBM-Quantum-System-Two,-Extends-Roadmap-to-Advance-Era-of-Quantum-Utility, https://www.hpcwire.com/2022/10/11/intels-road-to-a-universal-quantum-computer-is-via-chiplets/, https://ionq.com/posts/the-top-5-benefits-of-modular-quantum-computing-for-business

Open-ended Problems for Quantum Software

- System characterization and benchmarking
- Architecturally-dependent quantum error mitigation (QEM)
- Synergistic implementations of QEM and QEC
- Heterogeneous system interfacing
- Modularity-aware algorithm mapping and compilation
- Scalable software stacks
- Architecture and security in the quantum cloud
- Hybrid HPC ecosystem management



Intel's Road to a Universal Quantum Computer Is Via Chiplets

By Agam Shah

IONQ

The Top 5 Benefits of Modular Quantum Computing for Business

Development Roadmap

IBM **Quantum**

	2016–2019 ✓	2020 ✓	2021 ✓	2022 ✓	2023 ✓	2024	2025	2026	2027	2028	2029	2033+
	Run quantum circuits on the IBM Quantum Platform	Release multi-dimensional roadmap publicly with initial aim focused on scaling	Enhancing quantum execution speed by 100x with Qiskit Runtime	Bring dynamic circuits to unlock more computations	Enhancing quantum execution speed by 5x with quantum serverless and Execution modes	Improving quantum circuit quality and speed to allow 5K gates with parametric circuits	Enhancing quantum execution speed and parallelization with partitioning and quantum modularity	Improving quantum circuit quality to allow 7.5K gates	Improving quantum circuit quality to allow 10K gates	Improving quantum circuit quality to allow 15K gates	Improving quantum circuit quality to allow 100M gates	Beyond 2033, quantum-centric supercomputers will include 1000's of logical qubits unlocking the full power of quantum computing
Data Scientist						Platform Code assistant ◎	Functions	Mapping Collection	Specific Libraries			General purpose QC libraries
Researchers					Middleware Quantum Serverless ✓	Transpiler Service ◎	Resource Management	Circuit Knitting x P	Intelligent Orchestration			Circuit libraries
Quantum Physicist	IBM Quantum Experience ✓		Qiskit Runtime QASM3 ✓	Dynamic circuits ✓	Execution Modes ✓	Heron (5K) ◎ Error Mitigation 5k gates 133 qubits Classical modular 133x3 = 399 qubits	Flamingo (5K) Error Mitigation 5k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Flamingo (7.5K) Error Mitigation 7.5k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Flamingo (10K) Error Mitigation 10k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Flamingo (15K) Error Mitigation 15k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Starling (100M) Error correction 100M gates 200 qubits Error corrected modularity	Blue Jay (1B) Error correction 1B gates 2000 qubits Error corrected modularity
	Early ✓ Canary 5 qubits Albatross 16 qubits Penguin 20 qubits Prototype 53 qubits	Falcon ✓ Benchmarking 27 qubits		Eagle ✓ Benchmarking 127 qubits								

Innovation Roadmap

Software Innovation	IBM Quantum Experience ✓	Qiskit ✓ Circuit and operator API with compilation to multiple targets	Application modules ✓ Modules for domain specific application and algorithm workflows	Qiskit Runtime ✓ Performance and abstract through Primitives	Serverless ✓ Demonstrate concepts of quantum centric-supercomputing	AI enhanced quantum ✓ Prototype demonstrations of AI enhanced circuit transpilation	Resource management ◎ System partitioning to enable parallel execution	Scalable circuit knitting Circuit partitioning with classical reconstruction at HPC scale	Error correction decoder Demonstration of a quantum system with real-time error correction decoder				
Hardware Innovation	Early ✓ Canary 5 qubits Penguin 20 qubits Albatross 16 qubits Prototype 53 qubits	Falcon ✓ Demonstrate scaling with I/O routing with Bump bonds	Hummingbird ✓ Demonstrate scaling with multiplexing readout	Eagle ✓ Demonstrate scaling with MLW and TSV	Osprey ✓ Enabling scaling with high density signal delivery	Condor ✓ Single system scaling and fridge capacity	Flamingo ◎ Demonstrate scaling with modular connectors	Kookaburra Demonstrate scaling with nonlocal c-coupler	Demonstrate path to improved quality with logical memory	Cockatoo Demonstrate path to improved quality with logical communication	Starling Demonstrate path to improved quality with logical gates		
						Heron ✓ Architecture based on tunable-couplers	Crossbill ◎ m- coupler						

✓ Executed by IBM

◎ On target

IBM Quantum / © 2023 IBM Corporation

Image: https://newsroom.ibm.com/2023-12-04-IBM-Debuts-Next-Generation-Quantum-Processor-IBM-Quantum-System-Two,-Extends-Roadmap-to-Advance-Era-of-Quantum-Utility, https://www.hpcwire.com/2022/10/11/intels-road-to-a-universal-quantum-computer-is-via-chiplets/, https://ionq.com/posts/the-top-5-benefits-of-modular-quantum-computing-for-business

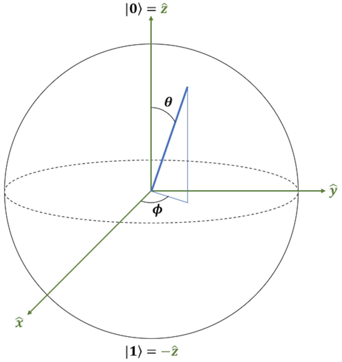
Opportunities for Multidisciplinary Engagement

Much space for applied mathematicians to contribute!

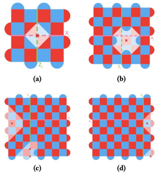
- Optimization everywhere within the software stack
- New techniques for gate synthesis / optimal control
- Formal methods for validation / verification
- New QEC codes

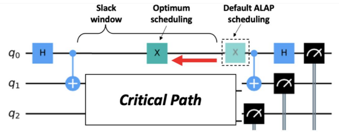
Thank you!

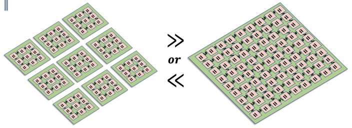
Theory

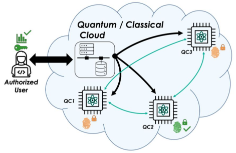


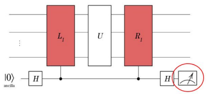
Architecturally-aware Optimization



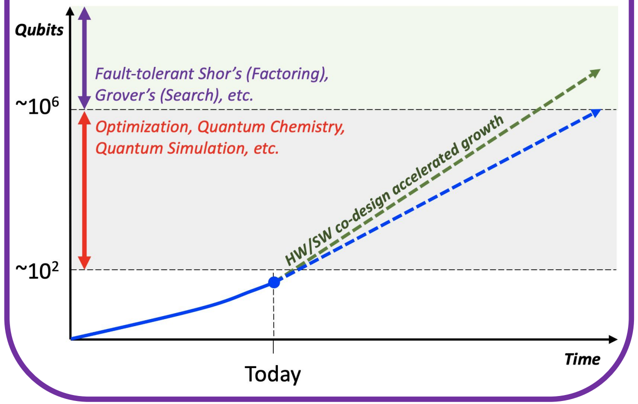








Path to Quantum Utility



Overview of Quantum Software and Compilers

Kaitlin N. Smith
Assistant Professor, Computer Science
Northwestern University