Quantum Computer Architecture (Co-Designed with Software): Tradeoffs and Breaking Abstractions

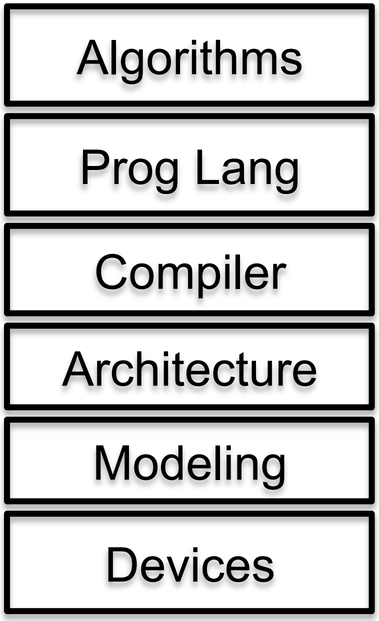
Fred Chong

Seymour Goodman Professor
Department of Computer Science
University of Chicago

Lead PI, the EPiQC Project,
an NSF Expedition in Computing

With Margaret Martonosi, Ken Brown, Peter Shor, Eddie Farhi, Aram Harrow, Diana Franklin, David Schuster, John Reppy, and Danielle Harlow (UChicago, MIT, Princeton, Duke, UCSB)

An Architects View of the World



Algorithms

Prog Lang

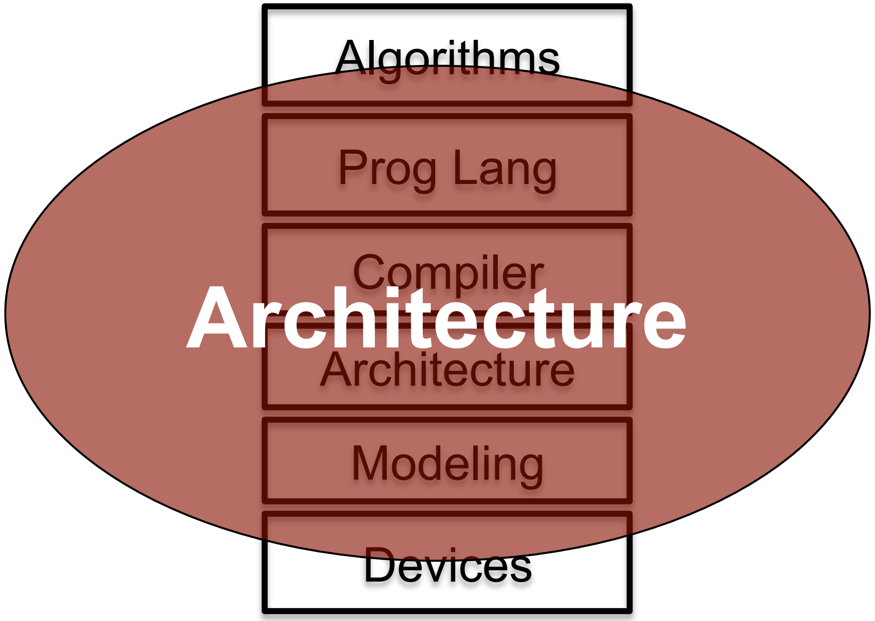
Compiler

Architecture

Modeling

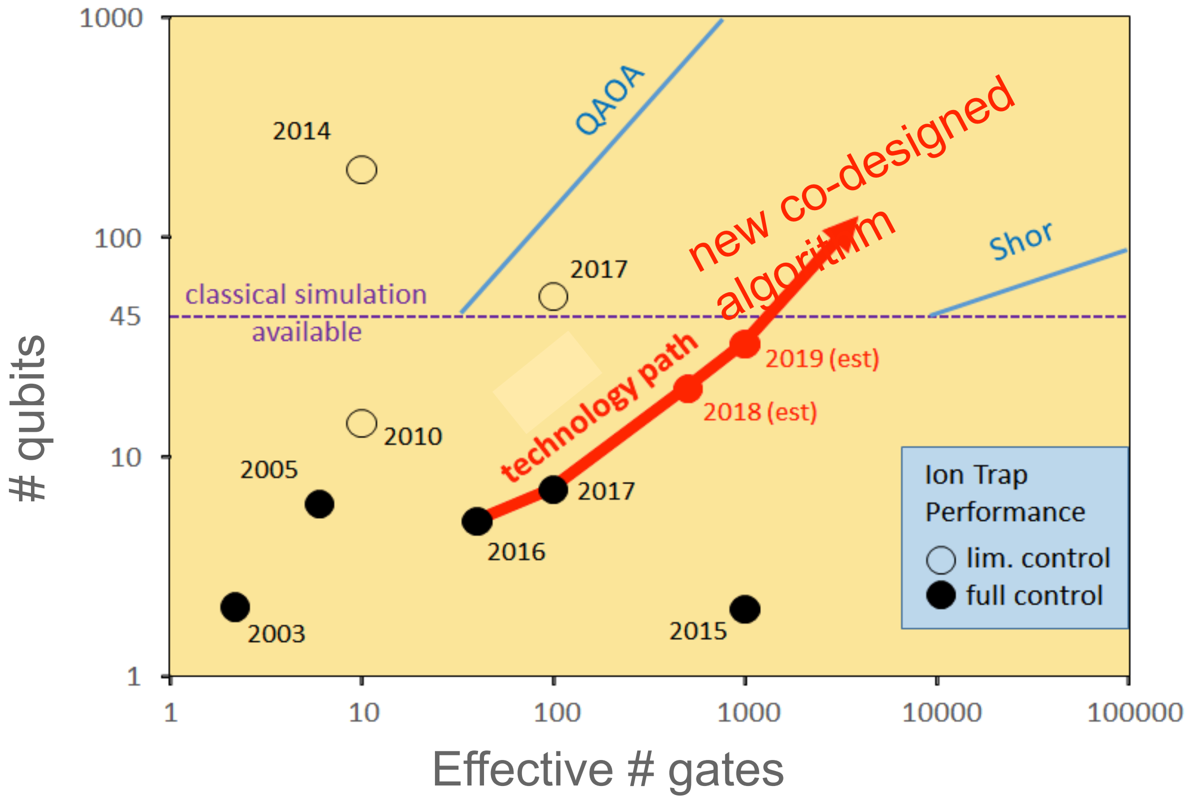
Devices

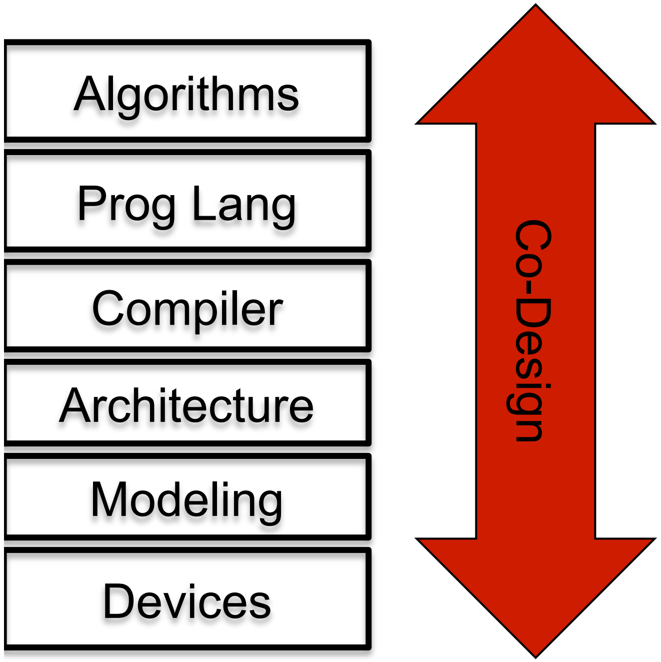
An Architect's View of the World



Goal

Develop co-designed algorithms, SW, and HW to close the gap between algorithms and devices by 100-1000X, accelerating QC by 10-20 years.





Algorithms

Prog Lang

Compiler

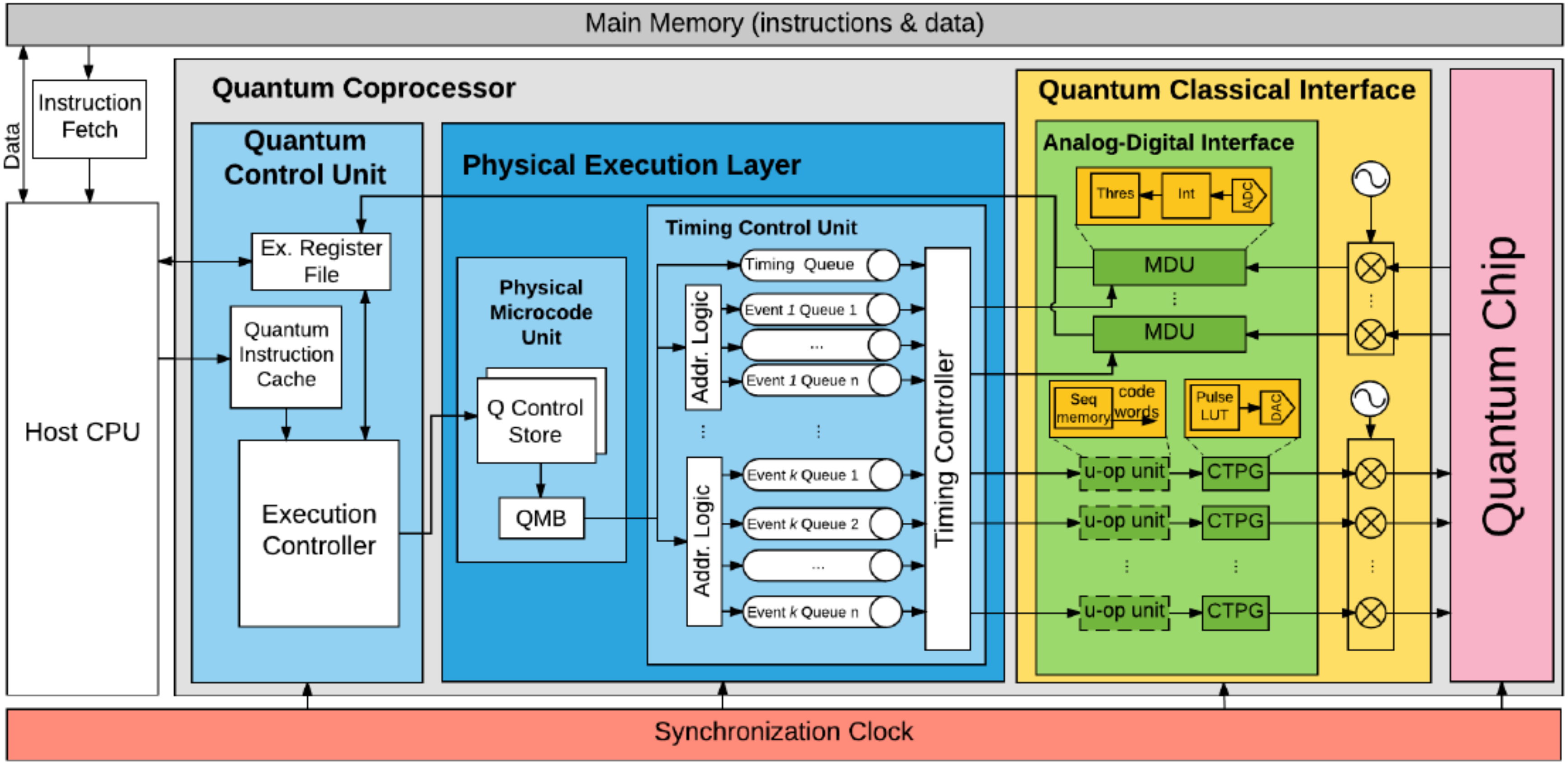
Architecture

Modeling

Devices

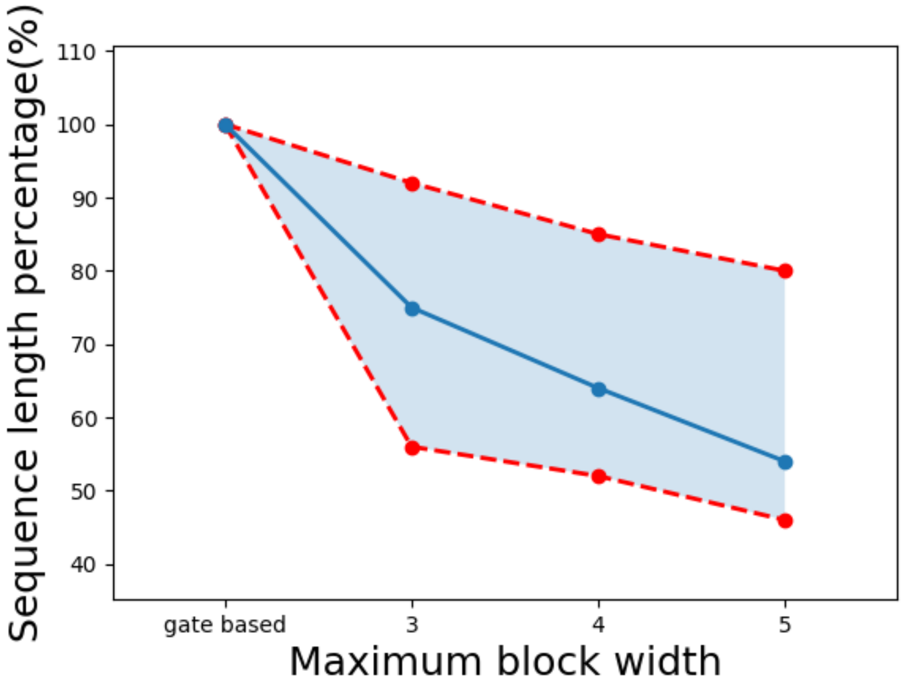
Co-Design

Microarchitecture



[Fu+ Micro 2017 Best Paper]

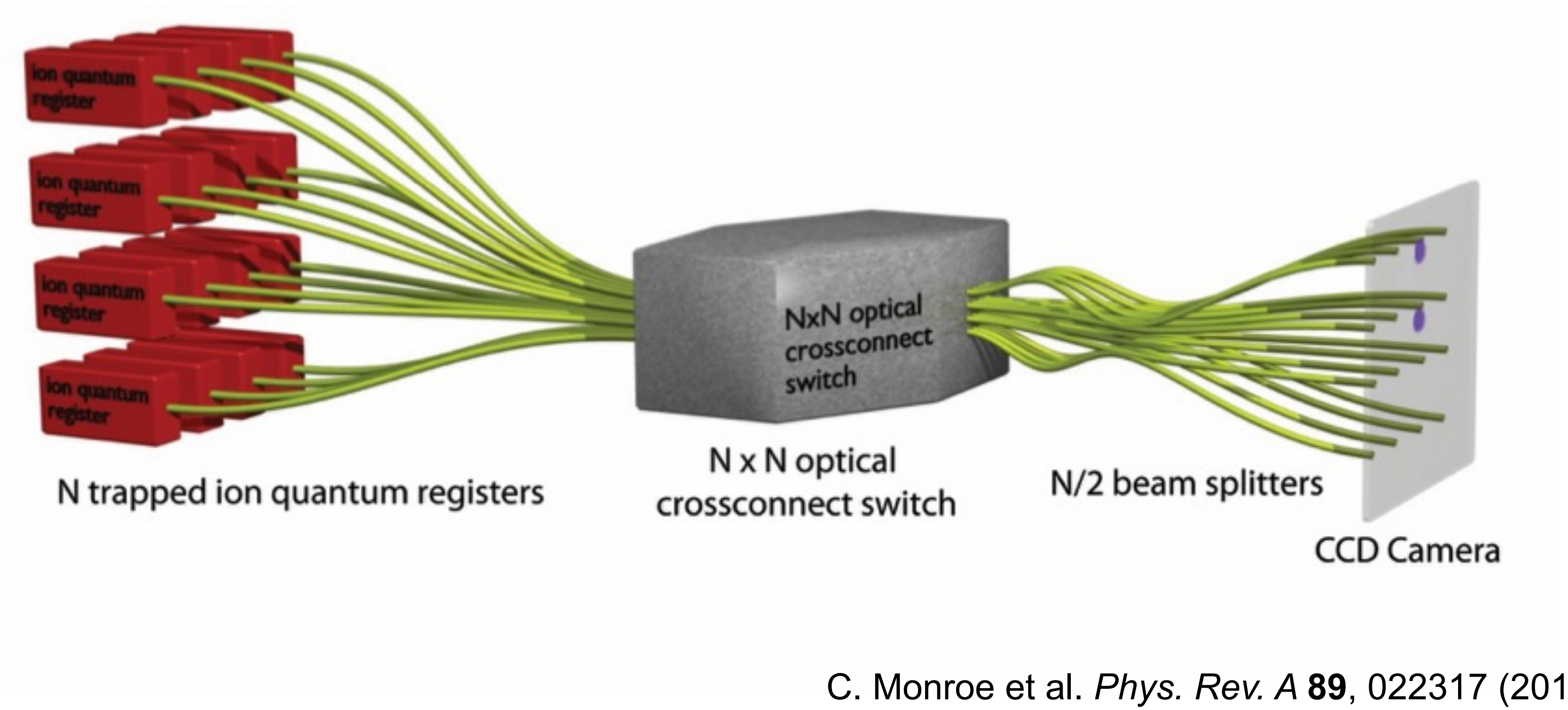
Breaking ISA Abstraction



- Multi-Qubit Operators for QAOA
  - Direct translation from compiler to control pulses

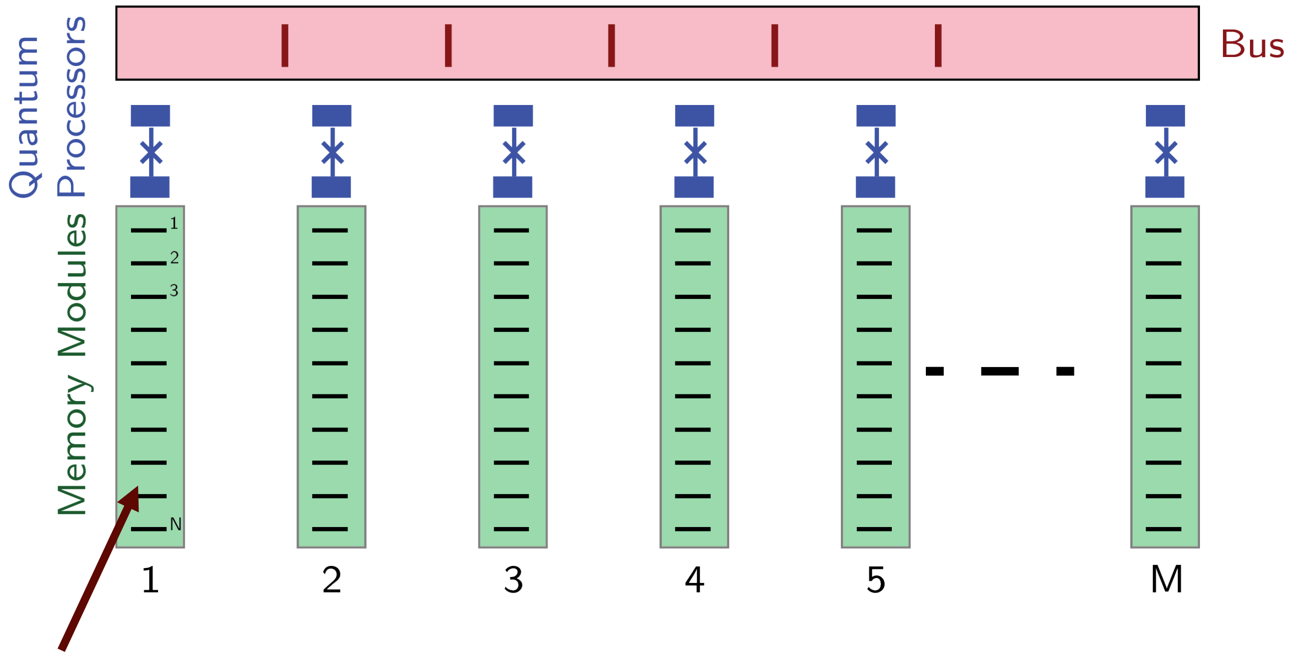
[Joint work with David Schuster]

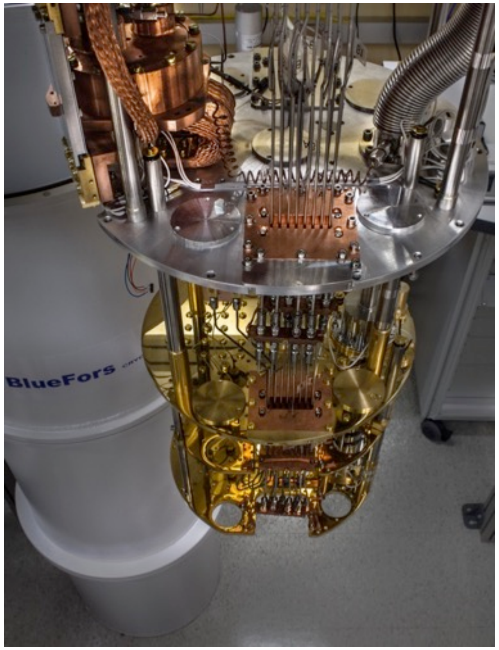
Modularity



C. Monroe et al. *Phys. Rev. A* **89**, 022317 (2014)

Modular Chicago QC Hardware architecture (Schuster)





Each memory mode an hold a qudit with up to 10 states

Advantages:

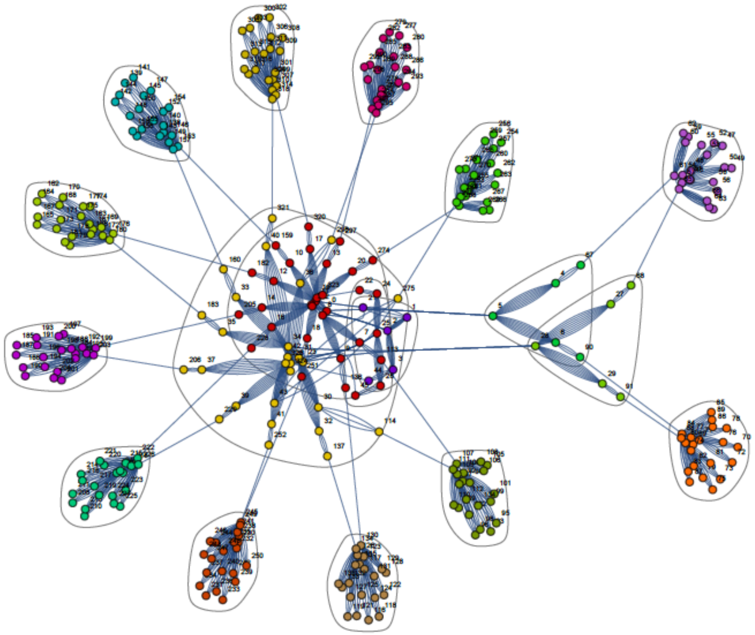
- 10 qubits per module, made in the machine shop, not the cleanroom
- 10x fewer transmons, 10x less classical hardware

Local vs Non-Local Communication

- Maybe 10X bandwidth difference?
- Not that unusual in the classical world
- How does this affect quantum algorithms?

Static vs Dynamic: Mapping Data

- Static spectral and graph partitioners
- Map for clustering
  - Probably necessary to get to 1000 qubits
- Map for irregular physical constraints
  - Qubit couplings, hardware defects
- Granularity of mappings
- Interaction with qubit reuse



Spectral communities for 2-level Bravyi-Haah magic-state factory

Static vs Dynamic: Compilation

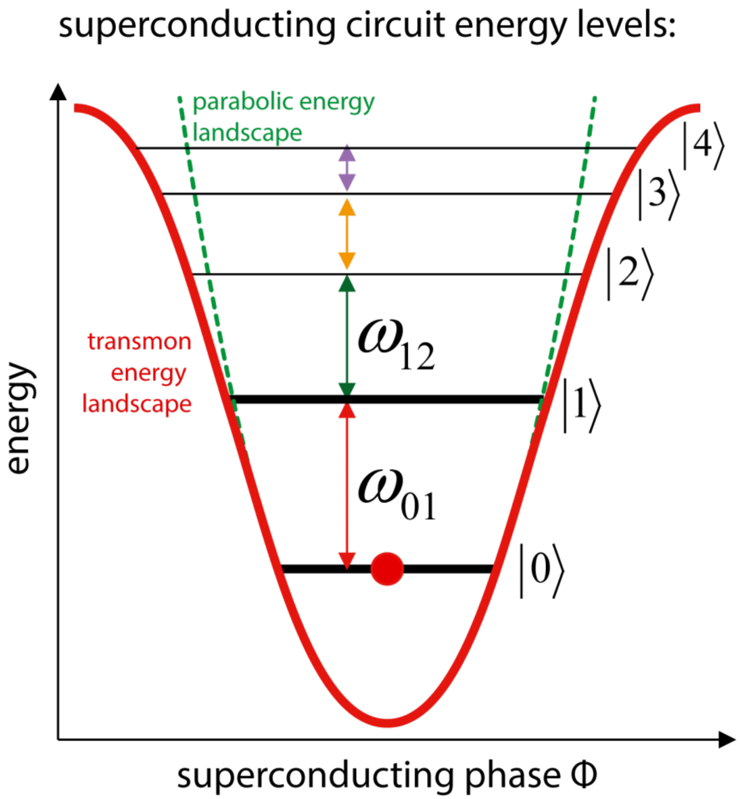
- Many applications static
- But quantum-classical co-processing may require dynamic parameters
- How to get a high level of optimization without complete re-compilation?
  - Eg hours for optimal control pulse generation, but how to adapt to changing rotation angles?
  - Similar to partial compilation for FPGAs

Multiple Tech vs Comm Overhead

- Classical architectures composed of multiple technologies: logic, SRAM, DRAM, interconnect
- With optical transduction, we can have:
  - Ions for high connectivity
  - Superconductors for high speed
  - Neutral atoms for storage

N-ary Logic vs Errors

- Use more than 2 qubit states per device
- Good for swap gates
- But higher modes have higher error probability

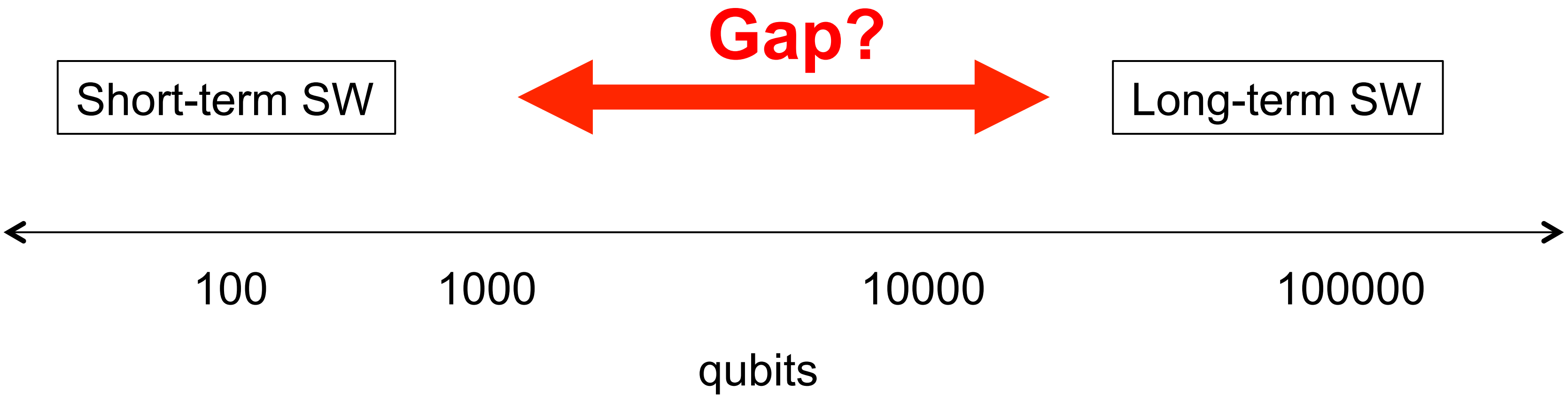


[image credit: qutech blog]

Classical Control and Computation

- Temperature boundaries and interconnect constraints [Tannu+ Micro17]
  - Cryo-cmos: high power, but lower cost to cool 4k
  - Superconducting: expensive memory, low power, but expensive to cool to 10mk
- Real-time control: hard for GHz speeds
  - Adaptive algorithms, ML
- Error decoding
  - Fast, simple decoder in superconducting logic
    - Trade frequency of decoding for quality

Specialization vs Abstraction



Gap?

Short-term SW

Long-term SW

100

1000

10000

100000

qubits