ECE 382V: INTRODUCTION TO QUANTUM COMPUTING SYSTEMS:
FROM A SOFTWARE AND ARCHITECTURE PERSPECTIVE

TEXAS
The University of Texas at Austin

INTRODUCTION

POULAMI DAS
ECE, The University of Texas at Austin

Some Problems Are Hard For Current Computers

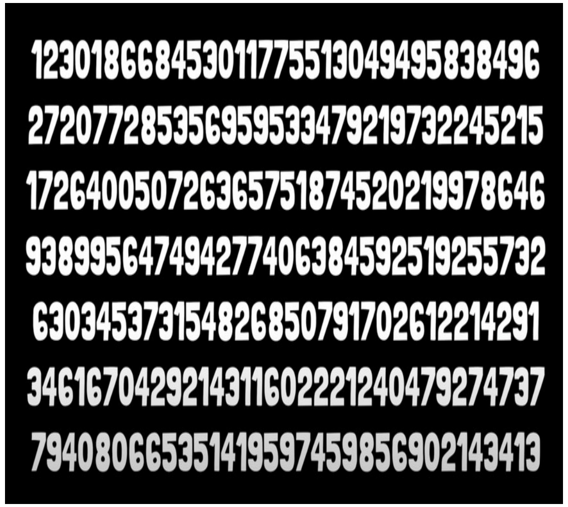
Number crunching
(E.g., Find the sum)

✓ *Easy*

Perception
(E.g., Is this a dog?)

✗ **Hard**
✓ *Solvable*

Factorization
(E.g., Factor N= 2048-bits)



✗ **Too Hard**

How “hard” is factorizing a 2048-bit number?

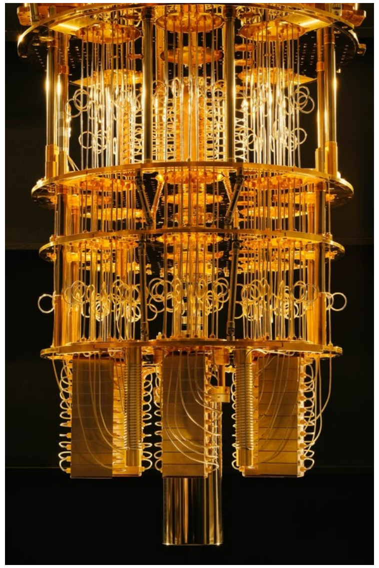
Factorization As An Example

Problem: Compute the prime factors of a 2048-bit number

Supercomputer Farm
(as large as Germany)

All Energy
On Earth

Quantum Computer



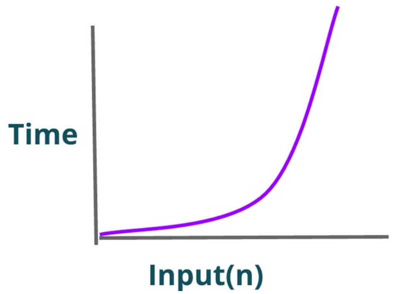
- ***Solution in Years***
- ***Solution in ~8 Hours***

Martinis' Talk @ Google, 2014
Gidney and Ekera, 2019

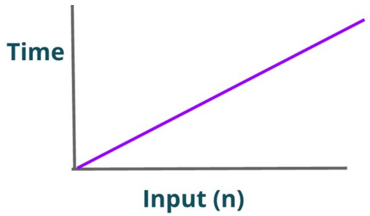
Quantum computers can solve problems that are beyond the scope of current computers

The Limitations Of Classical Machines

- Application
- Algorithm
- Program
- ISA
- µarchitecture
- Circuits
- Electrons



✓ *Faster*



vs

Some problems are fundamentally hard on classical machines due to lack of efficient algorithms

Another Example: Protein Folding

- Proteins are chains of amino acids (N) with an exponential number (2^N) of conformations (or spatial arrangements).

- Many proteins quickly fold to a native conformation –understanding this is crucial for solving many problems in healthcare, agriculture.

- ***O(N^4) on a quantum computer***

IBM Quantum: https://protein-folding-demo.mybluemix.net/
https://arxiv.org/pdf/1908.02163.pdf

Are these the only hard problems for conventional systems?

The True Potential Of Quantum Computing

Quantum computers promise massive speed-up for crucial problems in various domains – chemistry, material science, healthcare, etc.

Better fertilizers

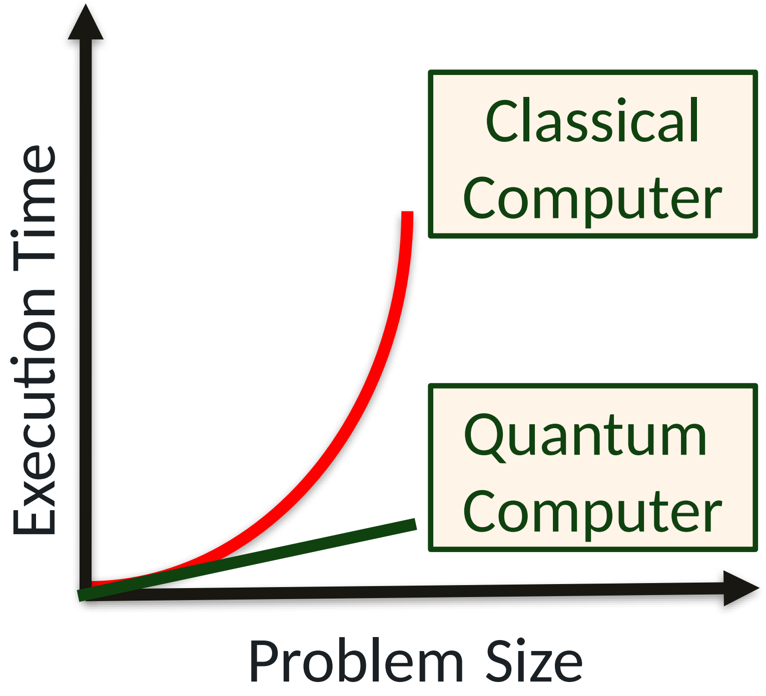
Better medicines

Carbon capture

Efficient batteries

Efficient power-grids

Optimal routes



Quantum computers enable us to solve many important problems

Abstractions & Transformations: Key To Computing

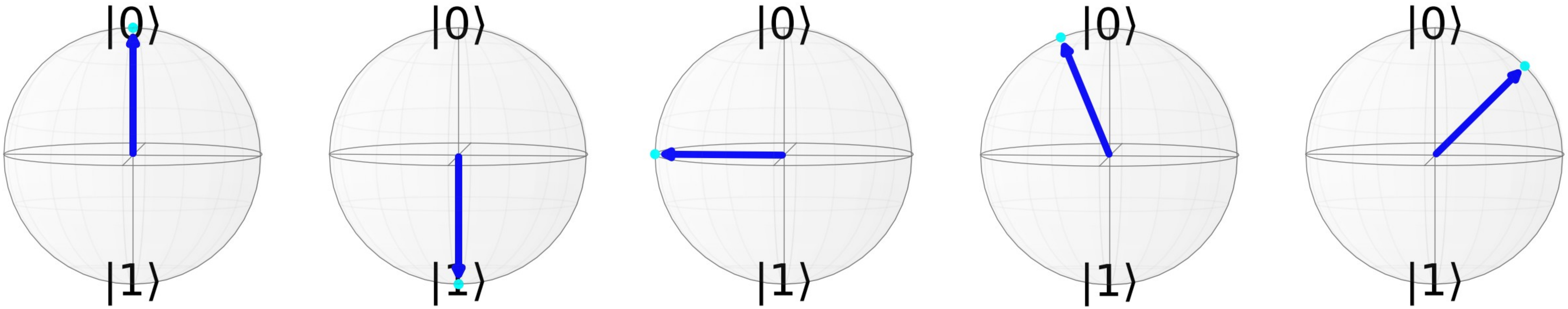
Classical	Quantum	
Application	Q Application	*Math, Quantum Information Theory*
Algorithm	Q Algorithm	
Program	Q Program	*ECE/CSE [Software, Architecture, & Systems]*
ISA	Q ISA	
μarchitecture	Q Ctrl μarch	
Circuits	Q Circuits	
Electrons	Qubits *(atoms, ions)*	*Devices*

This course focuses on the software, architecture, and systems aspect of quantum computing

Quantum Bits (Qubits)

- Classical machines encode information using classical bits
  - Either 0 or 1 at any given time (*only* two points on the sphere)

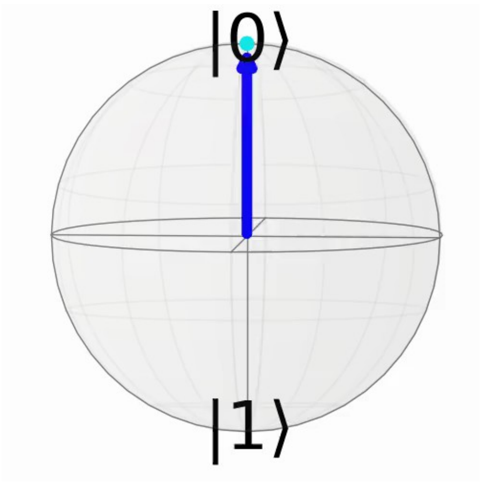
- A qubit is the fundamental unit of information on a quantum computer
  - Any combination of 0 and 1 at the same time (*any* point on the sphere)

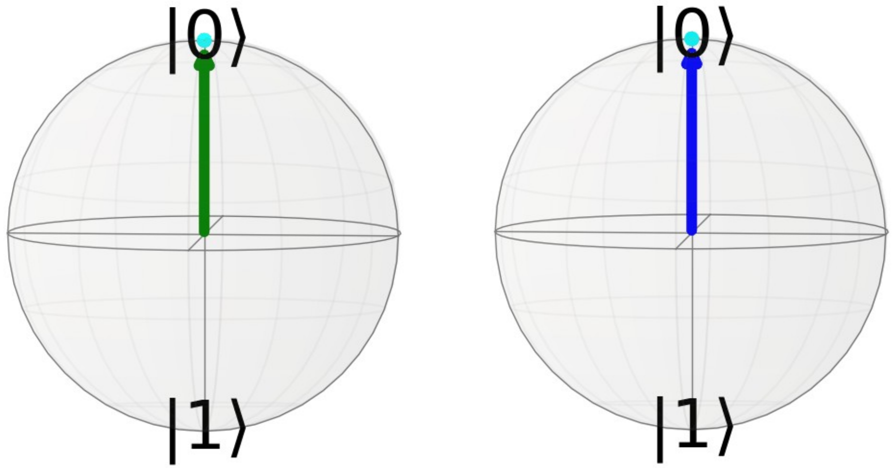


Quantum computers use qubits to encode information

Superposition & Entanglement

- A qubit exists in a ***superposition*** of 0 and 1.
  - A quantum system with ‘n’ qubits exist in ***superposition*** of all possible 2^n states.
- ***Entanglement*** enables us to create highly correlated states.
  - This property allows us to efficiently manage the exponential state space.

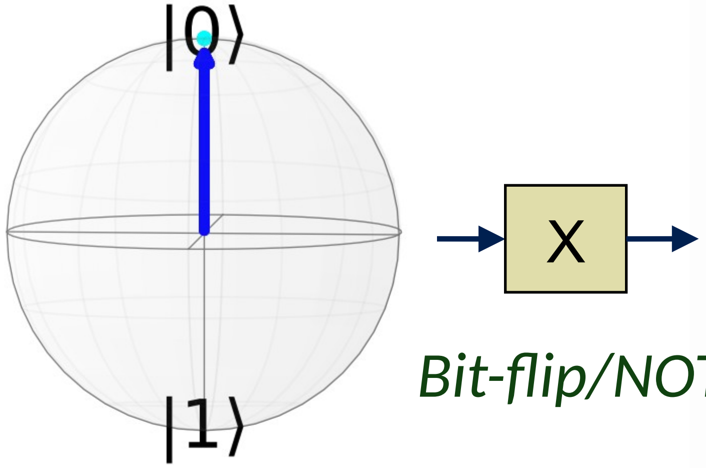




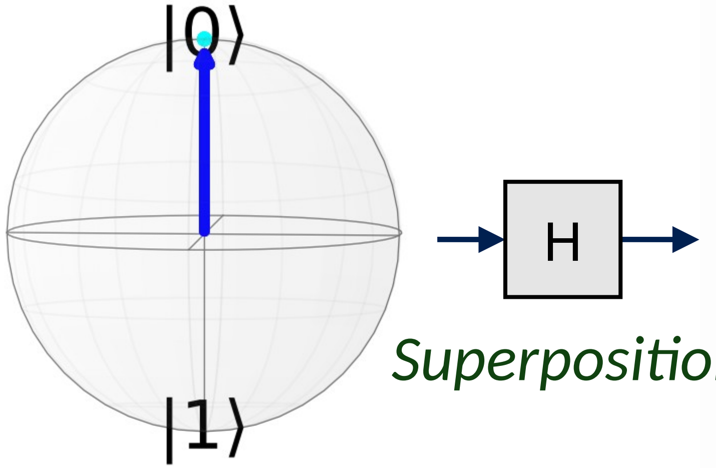
Quantum computers use qubits to encode information

Quantum Gate Operations

Quantum gates manipulate the state of qubits



Bit-flip/NOT



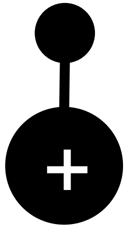
Superposition

Single-qubit gates

X Bit-flip/ NOT

H Superposition

Two-qubit gates

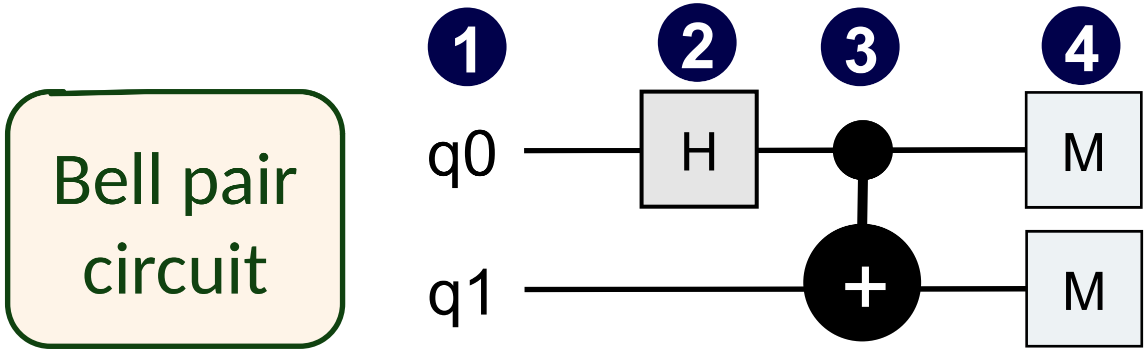


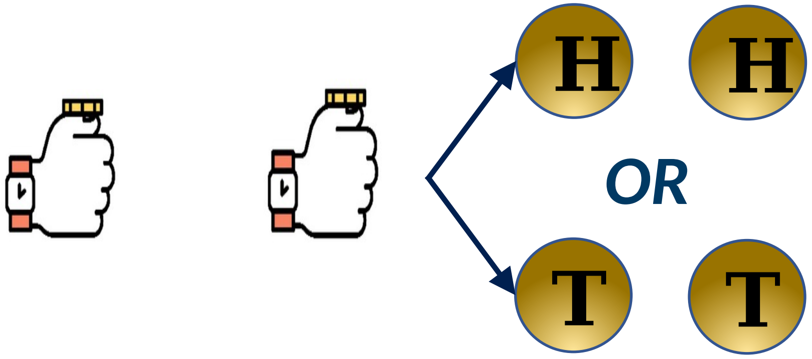
CNOT/ XOR (Entanglement)

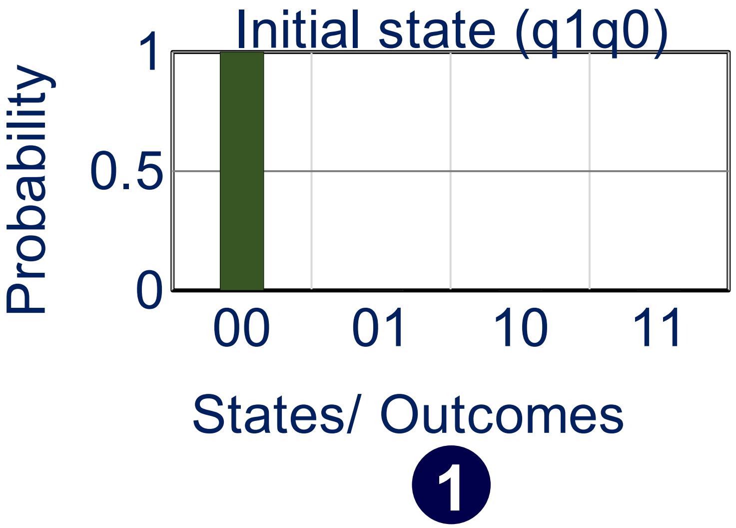
Quantum gate operations rotate the qubits from one point to another

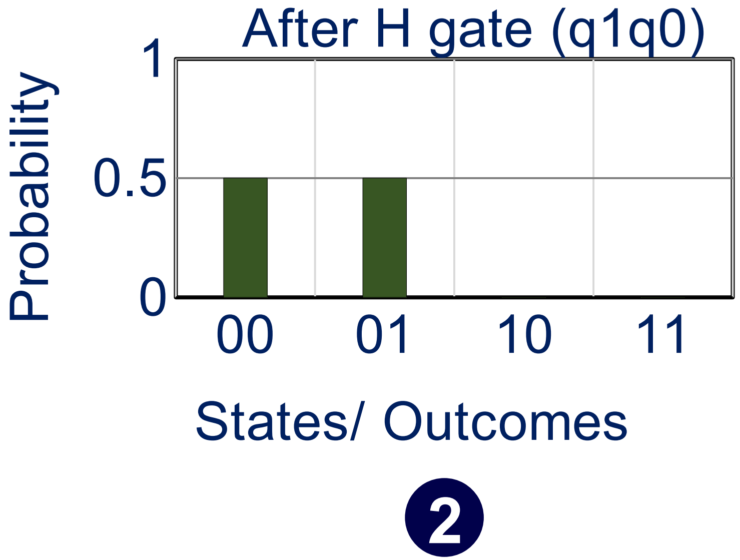
Quantum Programs

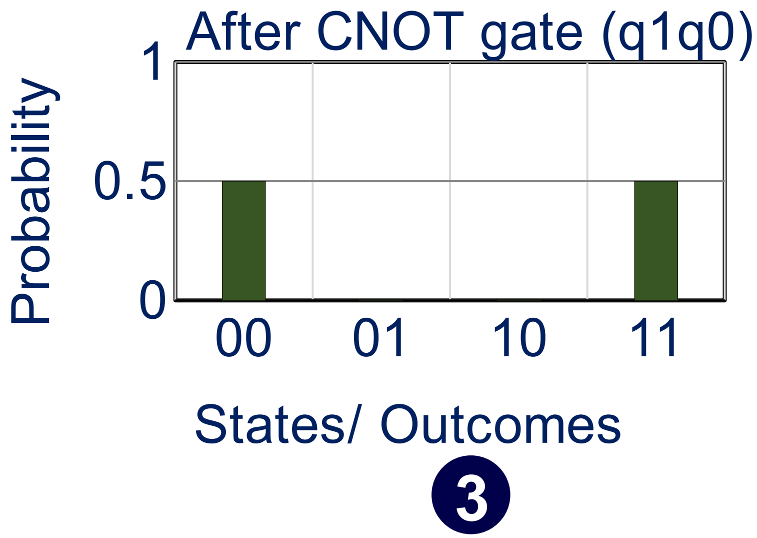
Quantum programs comprise of quantum operations







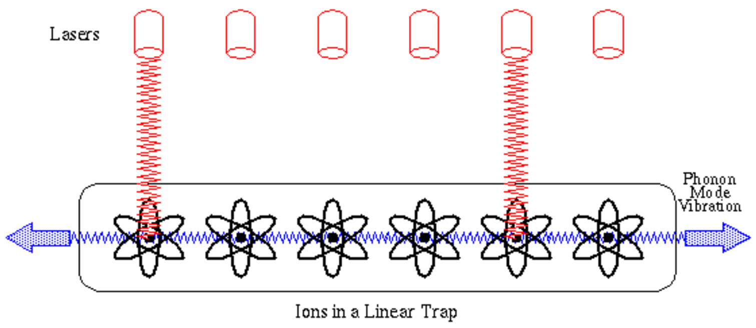




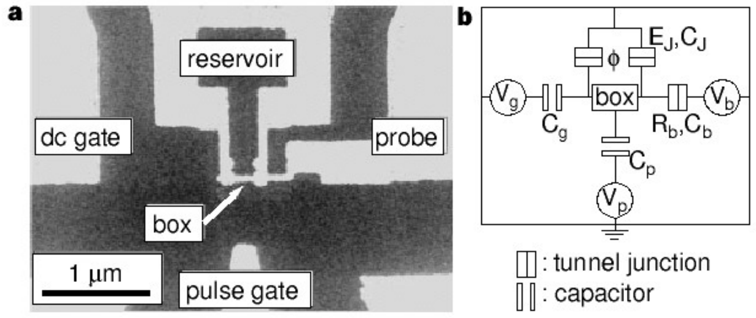
Quantum programs manipulate qubits to amplify the probabilities of the correct output(s)

Qubit Candidate Technologies

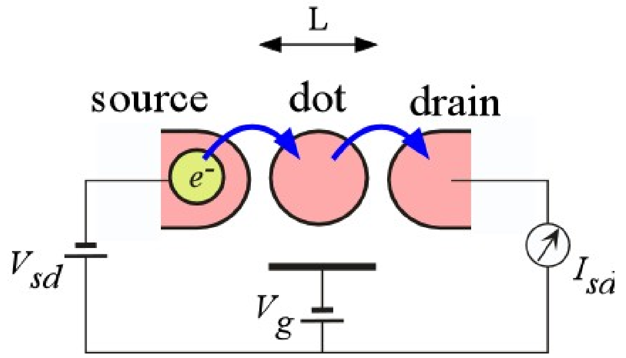
- A (physical) qubit can be any two-level system that can exist in any quantum superposition of two independent (physically distinguishable) quantum states



Ion-trap qubits



Superconducting qubits

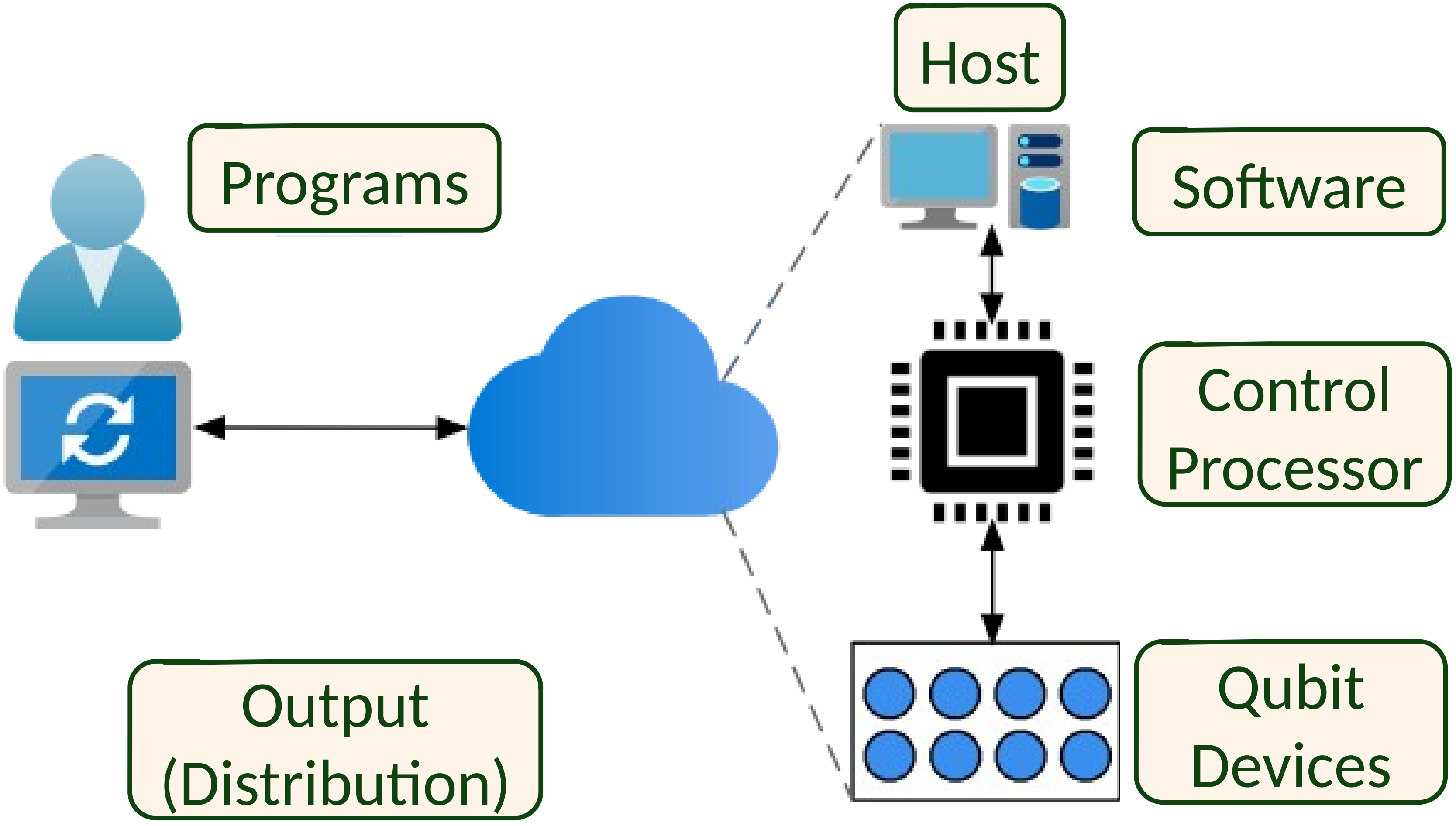


Quantum-Dot qubits

- And others: Photonic qubits, topological qubits, neutral atom qubits

Qubits are only a part of the quantum computing systems

Quantum Computer: Organization



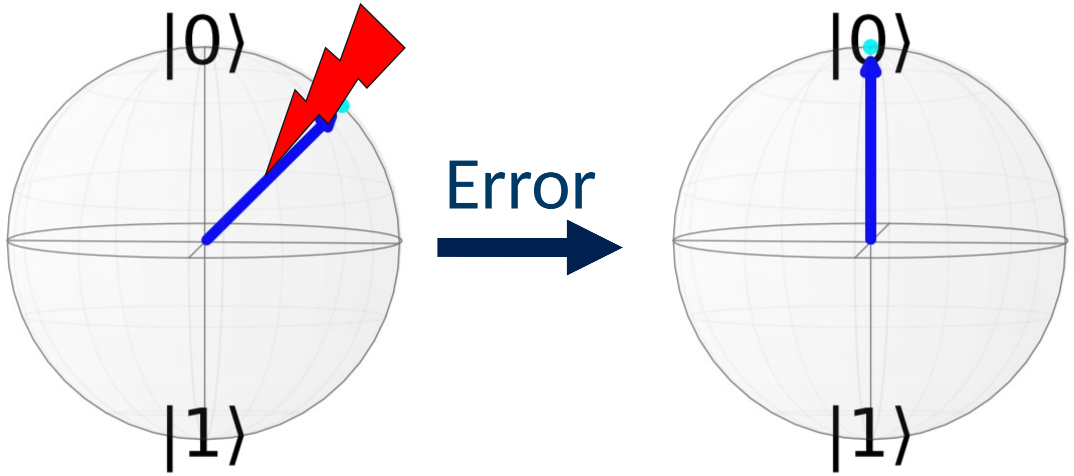
- ***Interface for access***
- ***Compilation***
- ***Output processing***

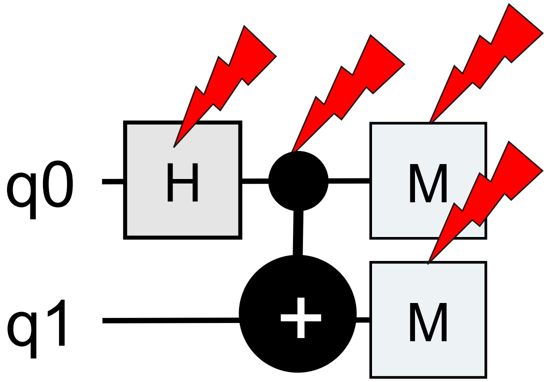
- ***Precise control to perform operations***

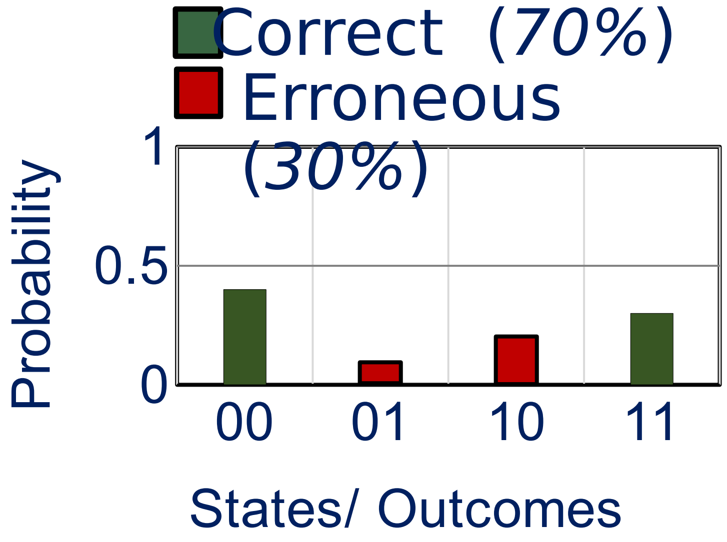
Quantum computer = Qubits + Classical Control Processor + Software

Key Challenge In Quantum Computing: Qubits Are Noisy

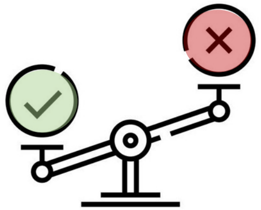
- Qubit devices naturally lose their state even when left idle (***decoherence***)
- Imperfect quantum operations lead to computational errors







~ 10- 100 microseconds



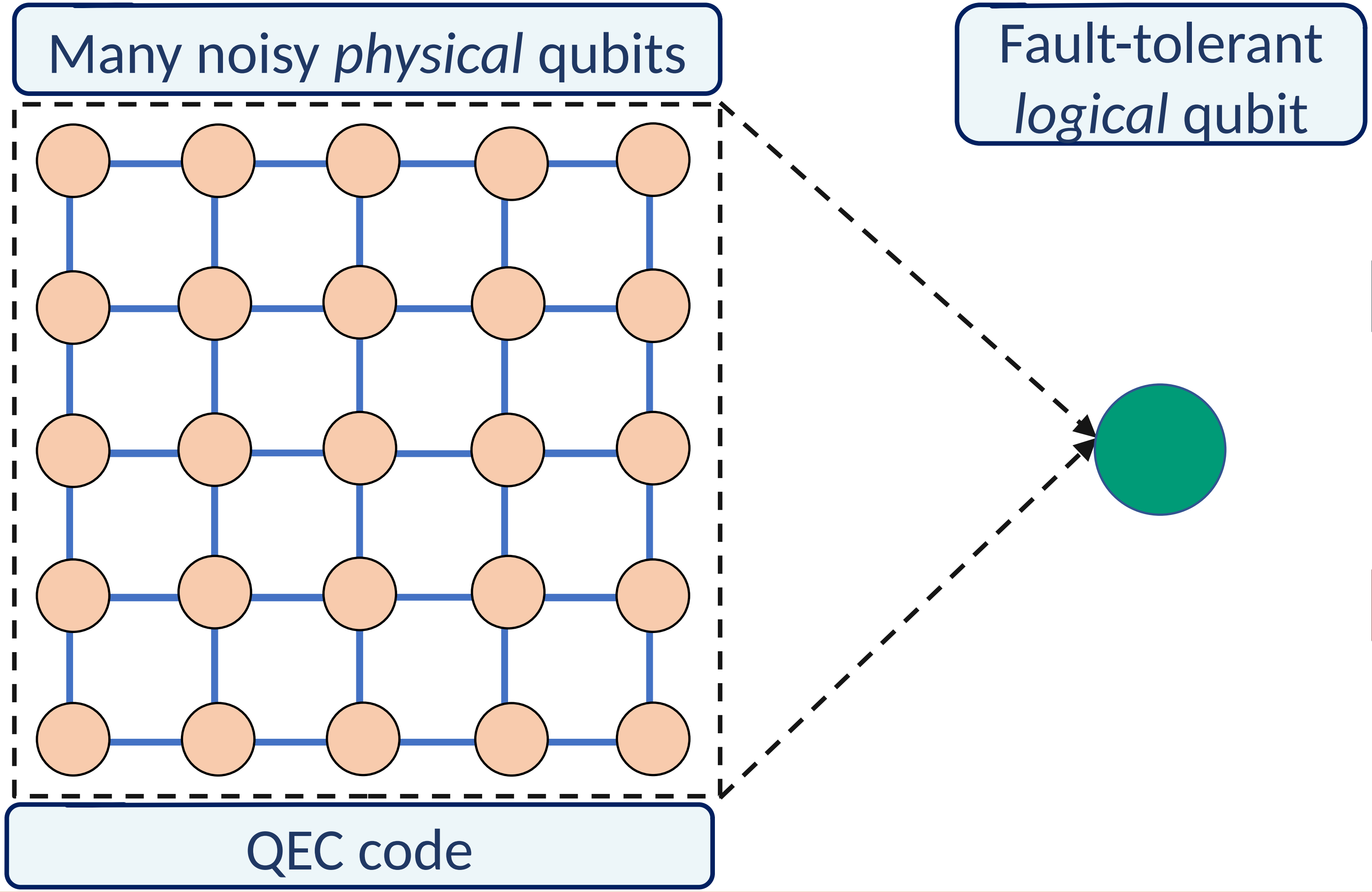
Gate errors ~1 %

Measurement/Readout errors ~1-4 %

Classical bit error-rate ~10^{-18}

Quantum hardware errors lead to incorrect outputs during program execution

Quantum Error Correction (QEC)



Logical error rate < physical error rate

50-1000 physical qubits -> 1 logical qubit

QEC enables fault-tolerance but is expensive

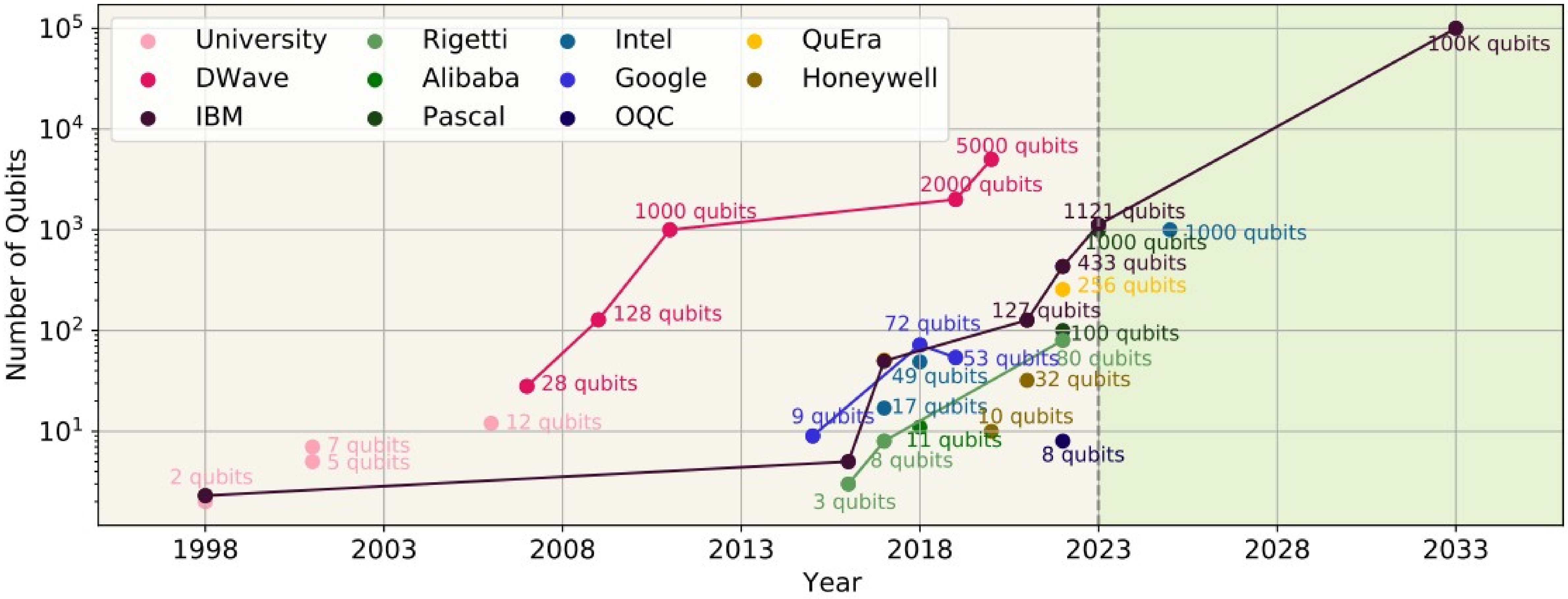
Evaluating Qubit Device Technologies

- Two vectors: Qubit lifetimes and operational fidelities + Scalability

Qubit Technology	Advantages	Challenges	Example
Superconducting	High gate fidelities	Cryogenic operations Short qubit lifetimes	Google, IBM Rigetti
Trapped ions	Extremely high gate fidelities Very long qubit lifetimes	Poor laser scalability Ultra-high vacuums	Honeywell IonQ
Photonic	Promising gate fidelities No cryogenics or vacuums	Two-qubit gates hard Short qubit lifetimes	PsiQuantum Xanadu
Neutral atoms	Promising gate fidelities Long qubit lifetimes	Poor laser scalability Ultra-high vacuums	QuEra, Pascal ColdQuanta
Silicon/Spin	Promising gate fidelities Good qubit lifetimes	Cryogenic operations High interference/crosstalk	Intel
Topological	Extremely high gate fidelities Extremely long qubit lifetimes	Extremely hard to build * No demonstration yet	Microsoft

It is too early now to bet on a single qubit device technology

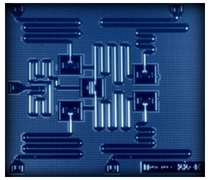
Available Quantum Systems + Future Roadmap

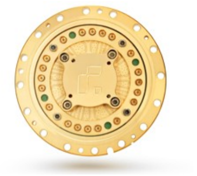


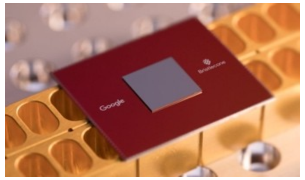
Objective is to scale up the systems while improving device error rates

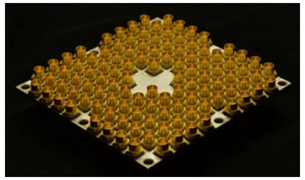
Noisy Intermediate Scale Quantum (NISQ)

Near-term quantum computers with few hundreds of qubits: NISQ machines









Today
(up-to 433 qubits)

→ **IBM Condor** → **IBM Kookaburra**

2023
1000+ qubits

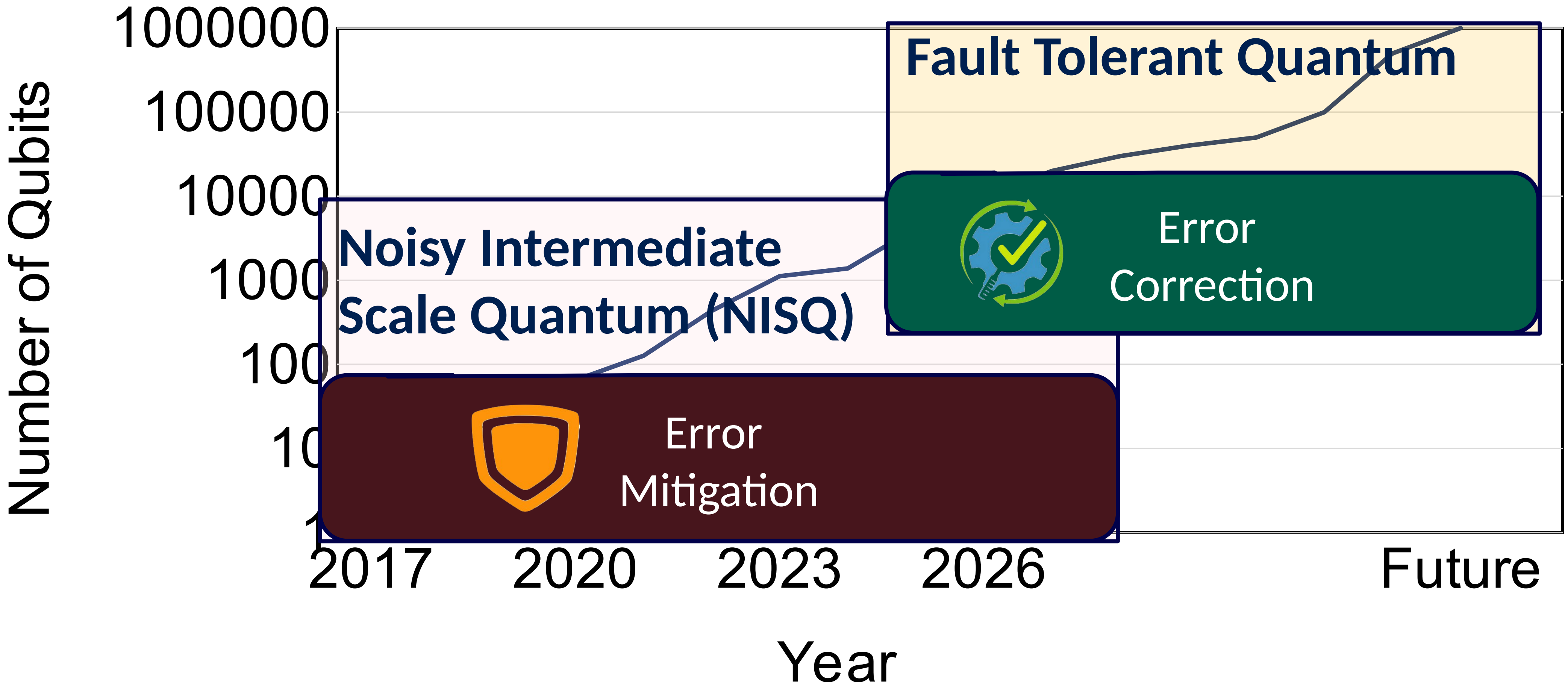
2025
4000+ qubits

} Too small for QEC at application level

Preskill, "Quantum Computing in the NISQ Era and beyond", 2018

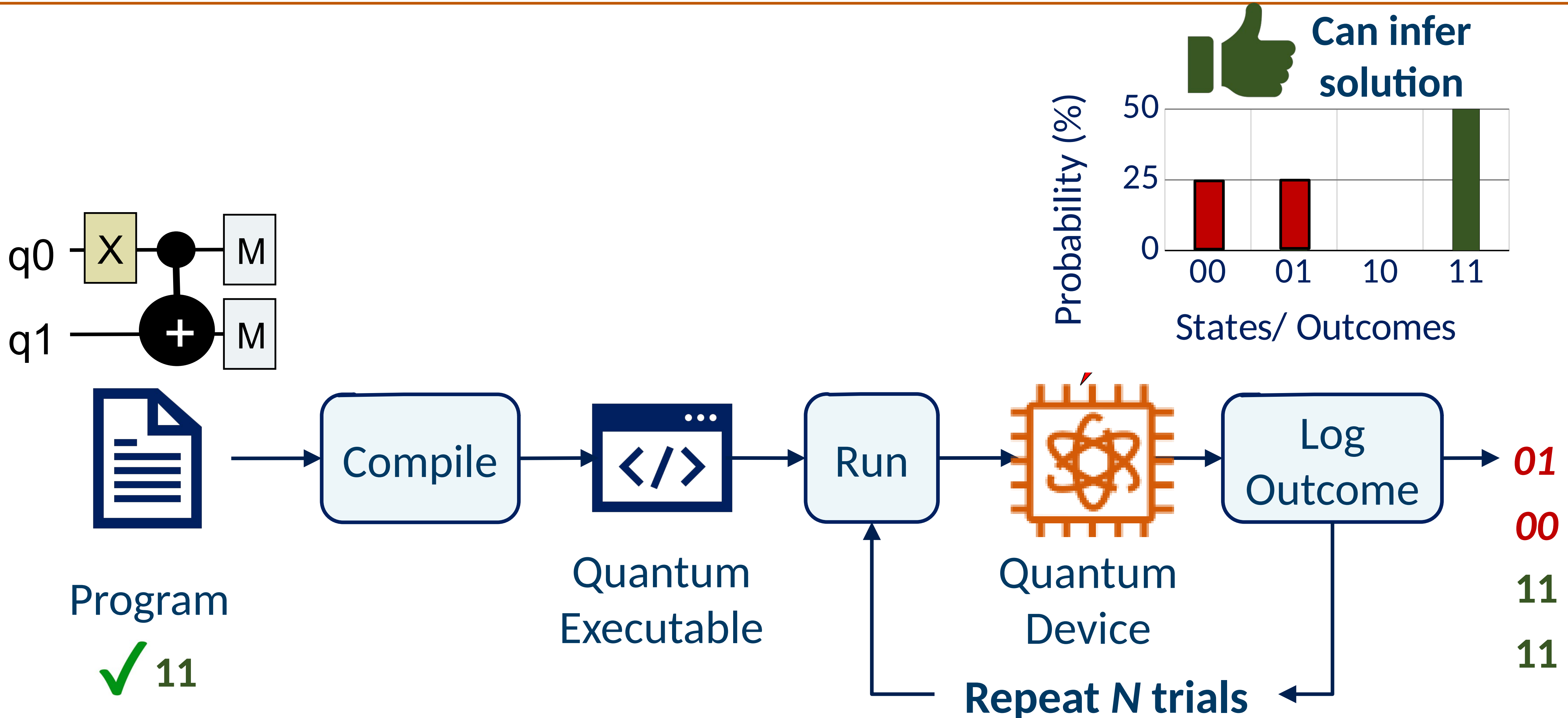
NISQ systems run programs in the presence of errors

The Quantum Roadmap



Quantum computers must enable fault-tolerance in the long-run, NISQ-regime until then

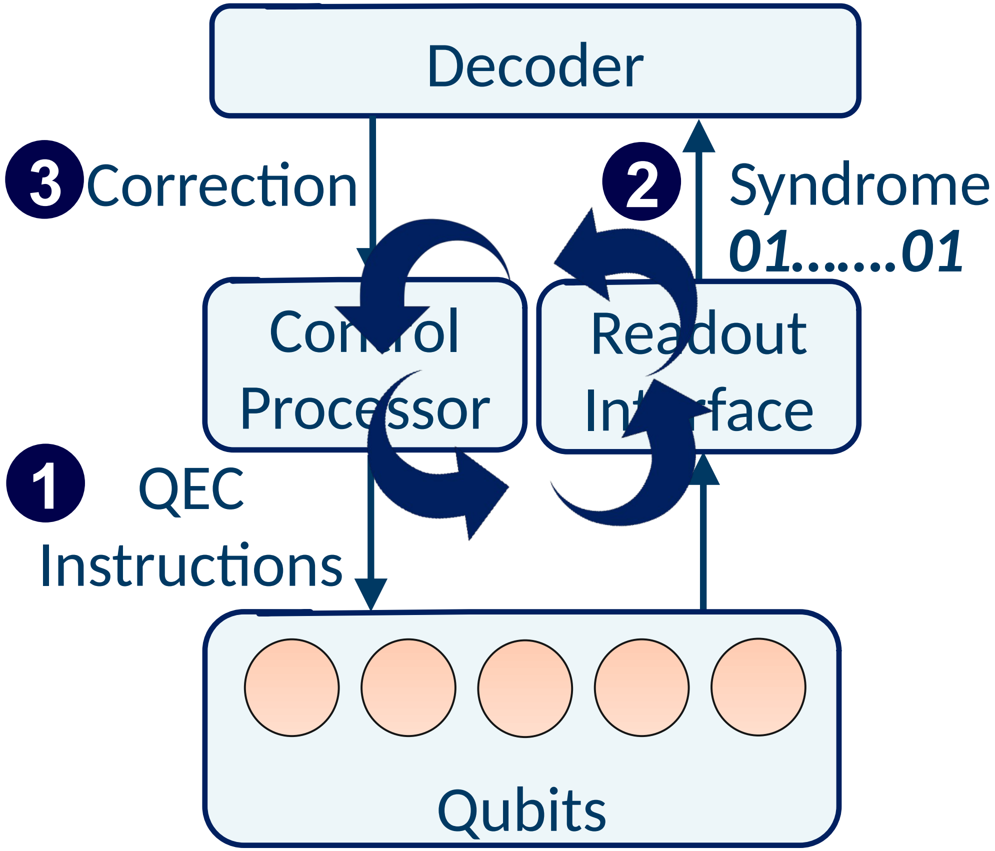
NISQ Computing Model



Goal: Maximize probability of successful execution or trial (program success rate)

Fault-Tolerant Quantum Computing

QEC involves syndrome (parity) extraction and decoding to identify errors



1. Instructions to generate syndrome
2. Send syndrome to decoder
   - *Did an error occur?*
   - *Where did the error occur?*
   - *What was the type of error?*
3. Send correction to control processor

FTQCs extract information about errors and corrects them in real-time

Why This Course?

ECE/CS UG/Grad

Highly motivated

- What is quantum computing?
- What was that recent blog from IBM/Google?
- How do quantum computers work?
- What is the primary role of software/architecture in quantum?
- Little bit from all the layers (algorithms, compilation, micro-arch, qubit devices)

About Instructor & Teaching Assistant

Instructor

- *PhD from Georgia Tech*
  - *Software and Architecture for Improving Fidelity of Emerging Quantum Computers*
- *Traditionally into computer architecture and hardware design*
- *Work experience and collaborations with industry*
- *Office Hours: Tuesday 3-4 pm / By appointment*
- *Email: poulami.das@utexas.edu*

Extra OH on August 24th (Thursday) 3-4 pm

Teaching Assistant

- *Hiring in progress*

We are here to help you succeed

Tentative Course Schedule

Week	Tues	Thurs	Notes
1 (8/21-)	Why QC? Course outline, Sup + Entanglement	Linear Algebra	
2 (8/28-)	Quantum gates and circuits	Basic algorithms (GHZ, Deutsch-Jozsa, BV)	
3 (9/04-)	Setup + learn basics of Qiskit	Basic algo contd. (quantum key distribution)	
4 (9/11-)	Advanced Algorithms [Shor, Grover]	Advanced Algorithms Cont.	HW-1 Due
5 (9/18-)	Errors, NISQ, Fault-Tolerance + Recap	**MIDTERM-1**	
6 (9/25-)	NISQ Applications, NISQ Compilation	Error mitigation for gate errors	Lab-1 Due
7 (10/2-)	Error mitigation for measurement errors	Error mitigation for idle/crosstalk errors	
8 (10/9-)	Variational quantum algorithms (QAOA)	Other topics in NISQ (neutral atom etc.)	HW-2 Due
9 (10/16-)	Quantum cloud services	**MIDTERM-2**	
10 (10/23-)	*Program verification in NISQ [Recorded]*	Quantum Error Correction: Intro and Basics	
11 (10/30-)	Current landscape of QEC, unique challenges	Control challenges for QEC	
12 (11/6-)	Error decoding (MWPM, Union-Find)	Hardware decoders	Lab-2 Due
13 (11/13-)	Hardware decoders contd.	Magic state distillation + other topics in FTQCs	
14 (11/20-)	Fall/Thanksgiving break		
15 (11/27-)	Leakage errors in QEC	Course Recap, Open Problems, Exam Review	

Course Evaluations & Other Logistics

Component	Scoring Policy	Total
2 Homework Assignments	5% x 2	10%
2 Lab Assignments	15% x 2	30%
2 Midterms	20% x 2	40%
1 Final exam / Project	20%	20%

Mainly papers from ISCA, MICRO, ASPLOS, HPCA

- *Quantum Computation and Quantum Information by Nielsen and Chuang*
- *Quantum Computer Systems Research for Noisy Intermediate-Scale Quantum Computers by Yongshan Ding and Frederic T. Chong*
- *Quantum Computing: Progress and Prospects [NAE Report]*
- *If you decide to take the course, please fill out the student information sheet and submit by the end of the class on August 24th*