Deconstructing (Future) Quantum Computer Design & Use

(Where does the power of Quantum Computing stem from? How will it impact Security/Privacy)

January 24, 2019

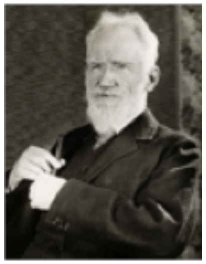
Csaba Andras Moritz,
Founder and Chairman
BlueRISC (www.bluerisc.com),
EPRIVO (www.eprivo.com)
Professor UMASS Amherst

andras@bluerisc.com

A Big *Gedanken Experiment*

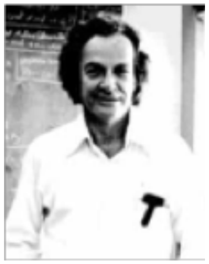
- Large-scale quantum computers
  - Do not exist but not known to violate underlying physics
  - Justified by entirely new (fascinating) form of computation, big challenges

- George Bernard Shaw, 1938
  - *"You have nothing to do but mention the quantum theory and people will take your voice for the voice of science and believe anything"*



- Scott Aaronson, MIT computer scientist, 2011
  - *"Quantum Mechanics, contrary to its reputation, is actually really simple, once you take all the Physics out."*

- Richard Feinman, 1965
  - *"I think I can safely say that nobody understands quantum mechanics."*
  - **What else did he say?**



Outline (Sharp Left Turn)

- **Qubits, Quantum State**
- **Quantum Circuits**

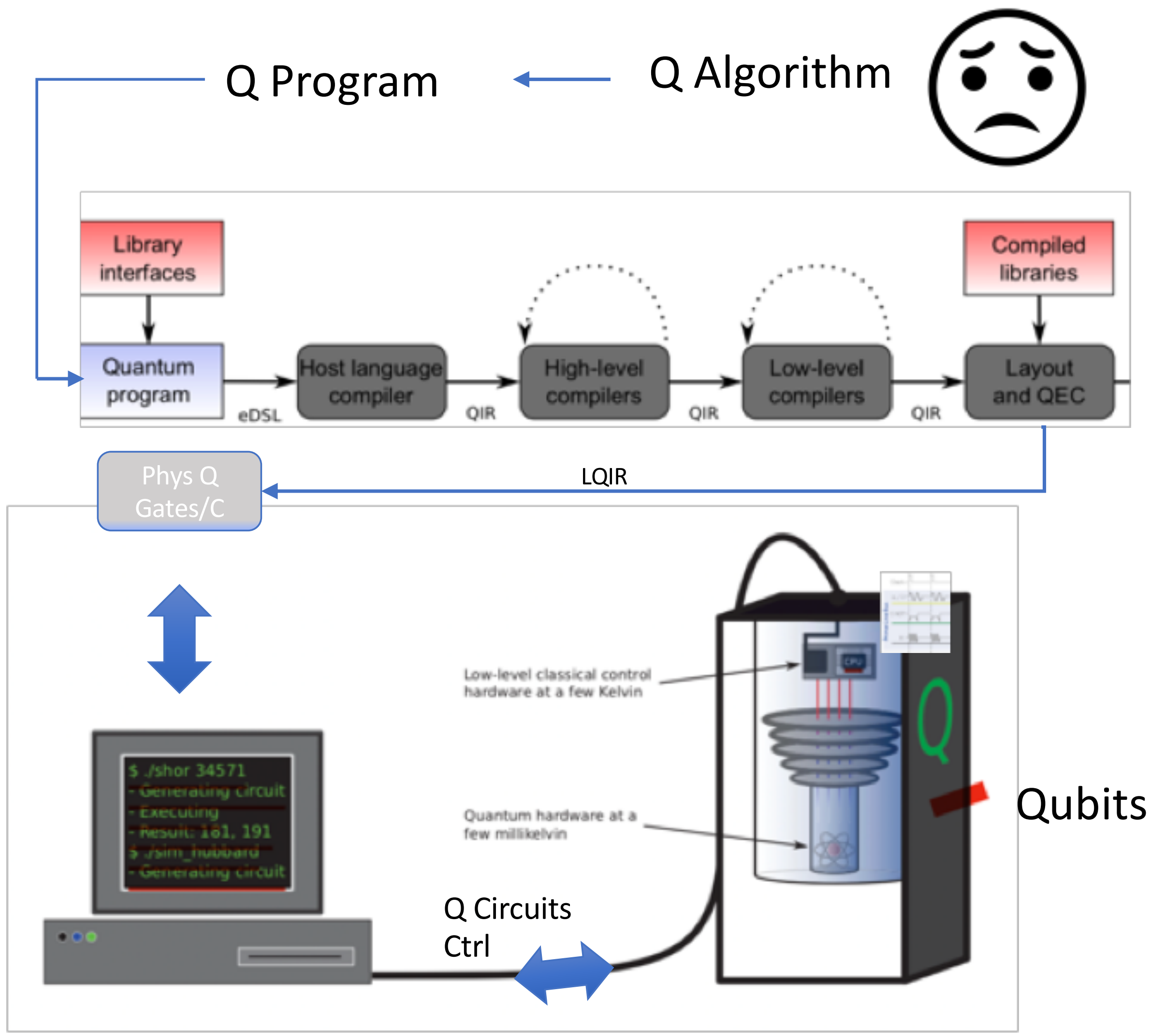
} *Foundations*

- Q Algorithms
- Q Compilers (QEC, Circuit Synth)
- Runtime Models
- C Microarchitecture Ctrl

Thomas Barbey photography

Viewpoint: (Future) Quantum Computer Design (Much) More Than Quantum Hardware Design

- Runs a quantum algorithm on quantum hardware
- Controlled by classical computer - feeds it with things to do (Q circuits)
- Q circuits - Q compiler generated – transforms state of special kind of bits called qubits



Part of Figure from (Haner, 2016)

Qubits, Quantum Information

- A classical bit can be 0 or 1.
- A qubit is a two-level quantum system. Its *quantum state is* a superposition of 0 and 1 states.
  - $N = 1$: $|\Psi\rangle = c_0|0\rangle + c_1|1\rangle$ Dirac "ket" notation
  - $N = 2$: $|\Psi\rangle = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle$
    - c_i - complex coefficients called amplitudes; $|00\rangle$ etc., standard *basis states*

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} c_i \left| b_{i,n-1} b_{i,n-2} \ldots b_{i,0} \right\rangle \quad (1)$$

- Why noteworthy?
  - N qubits can "store" exponentially more info than classical N bits due to superposition => enables *Quantum parallelism*
  - Wave-like properties => non-classical operations like *entanglement, interference*

Superposition: Imagine You Cannot Make Up Your Mind on Deserts, *State.* Which Are You *Inclined* to Get?

Qubits, Quantum Information contd.

- How to implement a qubit?
  - Clockwise and counterclockwise current in a superconducting circuit
  - Up and down electron spin in a uniform magnetic field
  - Horizontal and vertical polarization of a photon
  - Ground and excited state of a trapped ion

- Issues
  - Subject to quantum noise or decoherence => fragile
  - Output is *only* classical – so called *measurements* necessary to read out

Vector Representation of State

- State of a qubit can also be represented as a two dimensional complex vector space

$$\alpha|0\rangle + \beta|1\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\end{pmatrix}$$

  - where $|0\rangle$ and $|1\rangle$ are the basis states of a two-dimensional complex vector space and α, $\beta \in C$
- For two qubits, the basis states are all possible configurations of two classical bits, i.e. $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. A general two-qubit state:

$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$
$$= \alpha\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\\0\\0\end{pmatrix} + \gamma\begin{pmatrix}0\\0\\1\\0\end{pmatrix} + \delta\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\\\gamma\\\delta\end{pmatrix}$$

State of N-qubit System

- The state of a general n-qubit system can be an *arbitrary* superposition over all 2^n computational basis states, i.e.,

Dirac's *vector*

- $$\sum_{q_1 q_2 \ldots q_n \in \{0,1\}^n} c_{q_1 \ldots q_n} |q_1 \ldots q_n\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle \qquad \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \ldots \\ c_{2n\_1} \end{pmatrix}$$

- Complex amplitudes c_i need to satisfy the normalization condition:

$$\sum_i |c_i|^2 = 1$$

- Probability of measuring a given state is equal to lulus of the amplitude associated with that state. **Bohr's rule**.

Joint Quantum State from Individual States

- Example two qubits:
  - $|\alpha\rangle := \alpha_0|0\rangle+\alpha_1|1\rangle$ and $|\beta\rangle := \beta_0|0\rangle+\beta_1|1\rangle$,
  - the state of the entire system is the tensor product of the two individual states
    - which corresponds to the Kronecker product in vector notation

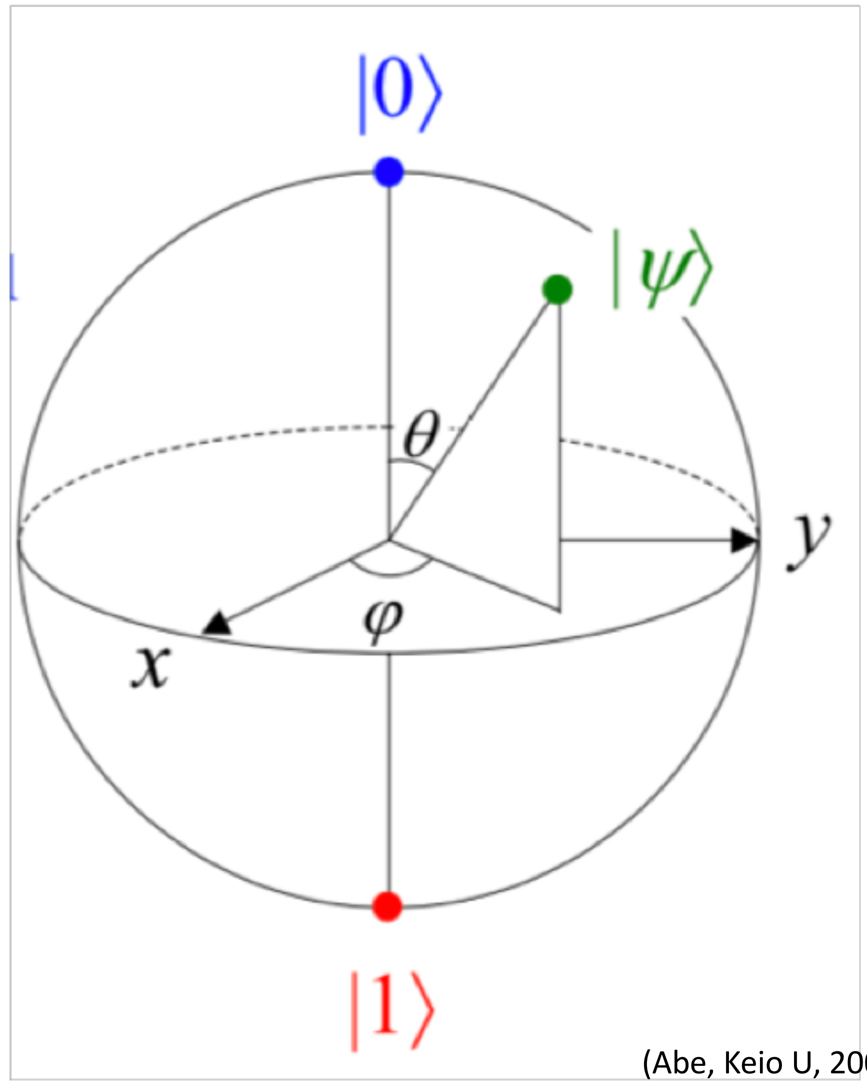
$$\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix} \otimes \begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix} = \begin{pmatrix}\alpha_0\beta_0\\ \alpha_0\beta_1\\ \alpha_1\beta_0\\ \alpha_1\beta_1\end{pmatrix}$$

  - Or,
    - $= \alpha_0\beta_0\ |00\rangle + \alpha_0\beta_1\ |01\rangle + \alpha_1\beta_0\ |10\rangle + \alpha_1\beta_1\ |11\rangle$ *in Dirac's*

Block Sphere Representation of a Qubit

- Intuitive to understand quantum transformations
- $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$

- The state is visualized as a point/vector on the sphere.
- A gate is a transformation to another point/vector.



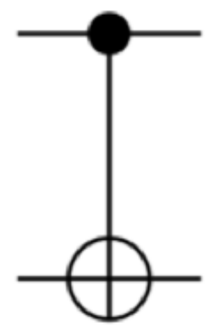
(Abe, Keio U, 2005)

Qubit Gates

- A gate is a unitary transformation (thus reversible) on the quantum state
  - $|\psi\rangle \rightarrow U|\psi\rangle = |\psi'\rangle$
  - U is the unitary operator or matrix

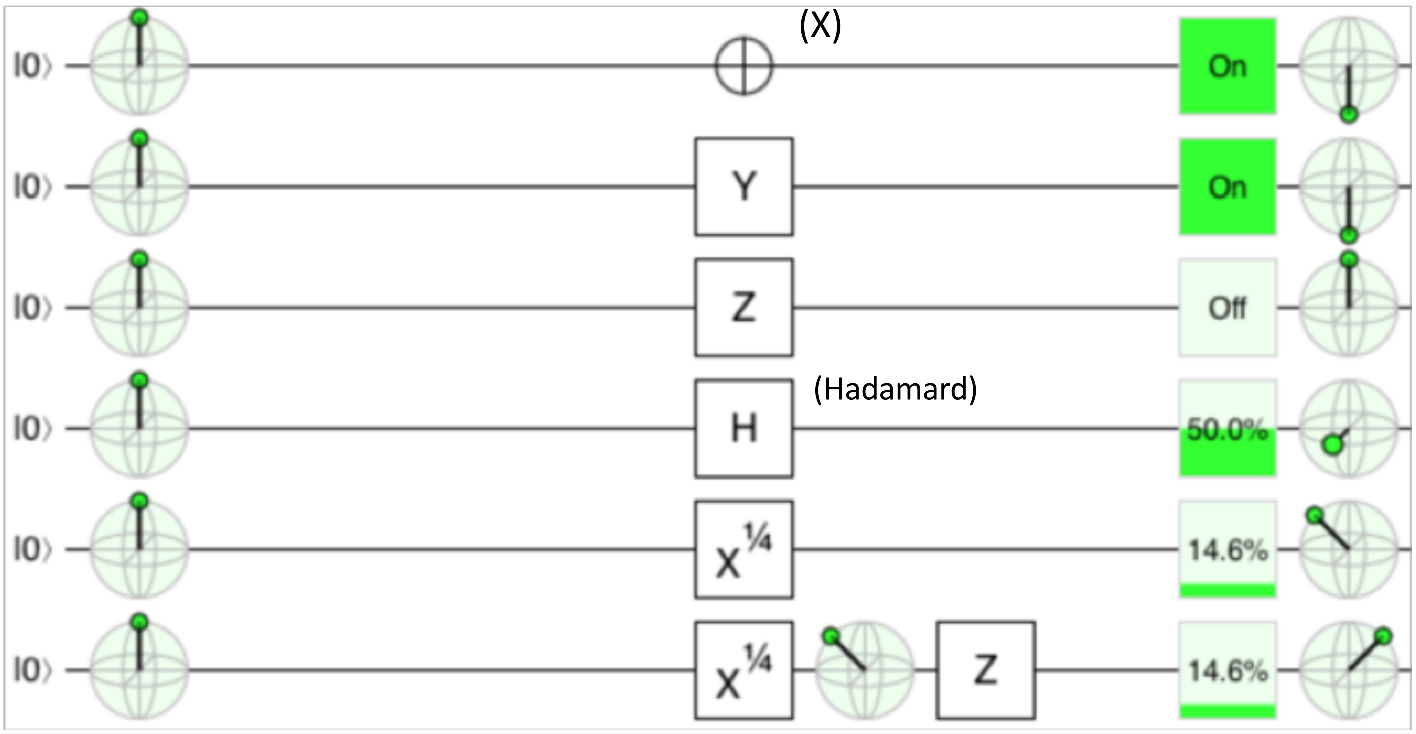
- A unitary quantum operation on n qubits can be written as a matrix of dimension $2^n \times 2^n$.

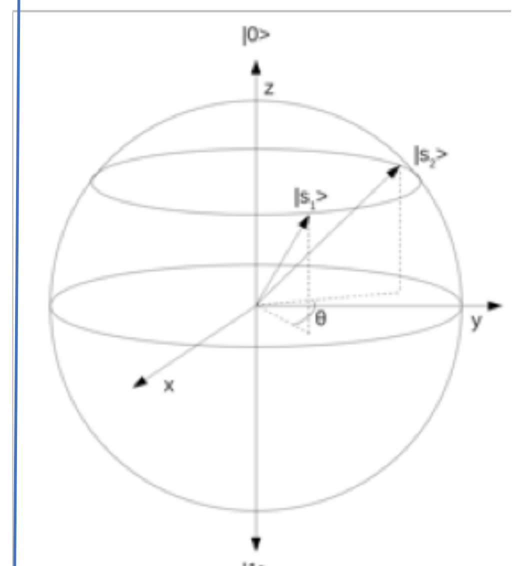
Some Important Q Gates

Gate	Matrix	Symbol
NOT or X gate	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	
Y gate	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Y
Z gate	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Z
Hadamard gate	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	H
S gate	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	S
T gate	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	T
Rotation-Z gate	$\begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$	Rz_θ
Controlled NOT (CNOT)	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	

Gates with *Quirk* Circuit Simulator

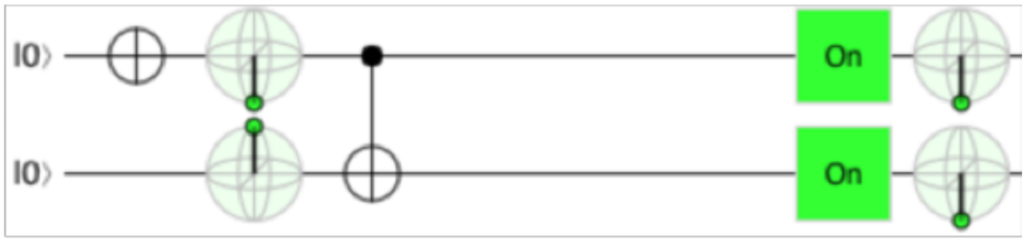
- Created to show effect with Bloch



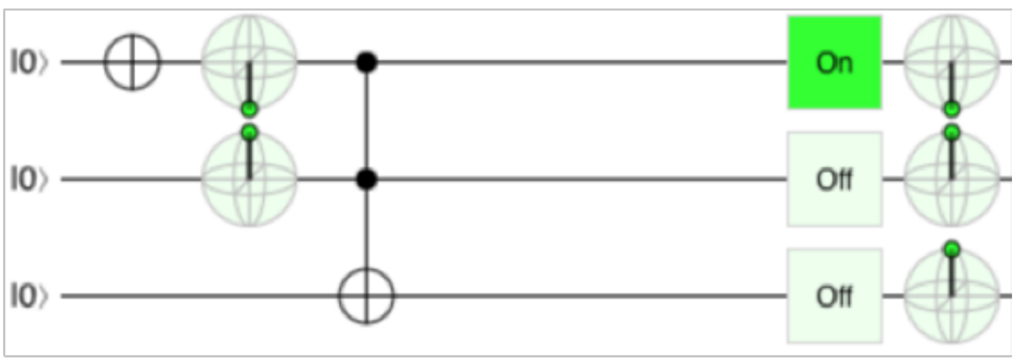


Phase Shift
(Z, S, T)

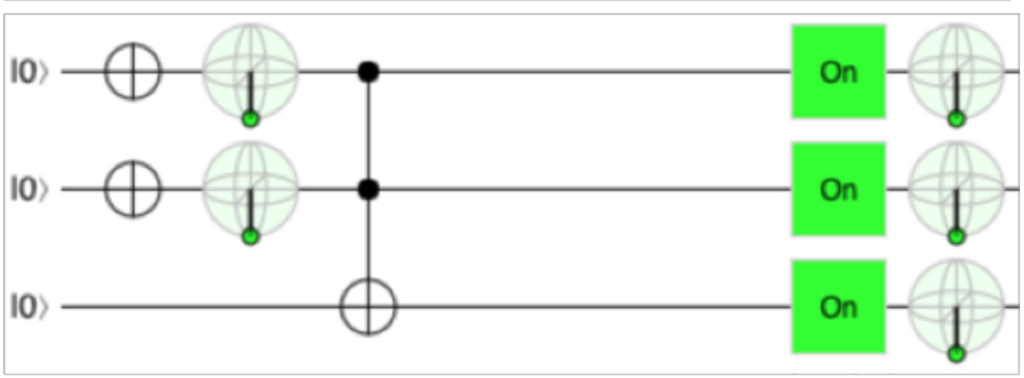
Two-input CNOT



3-input CCNOT or Toffoli

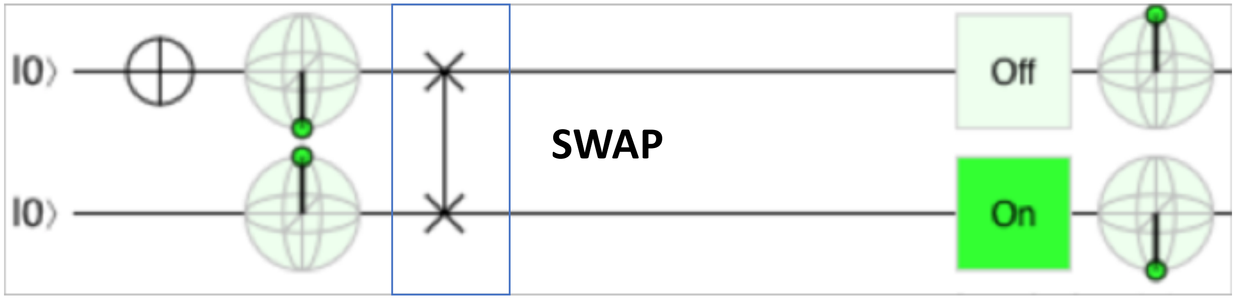


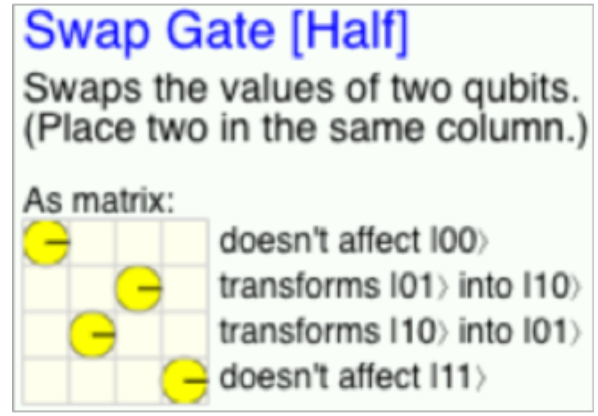
Nothing happens on 3rd

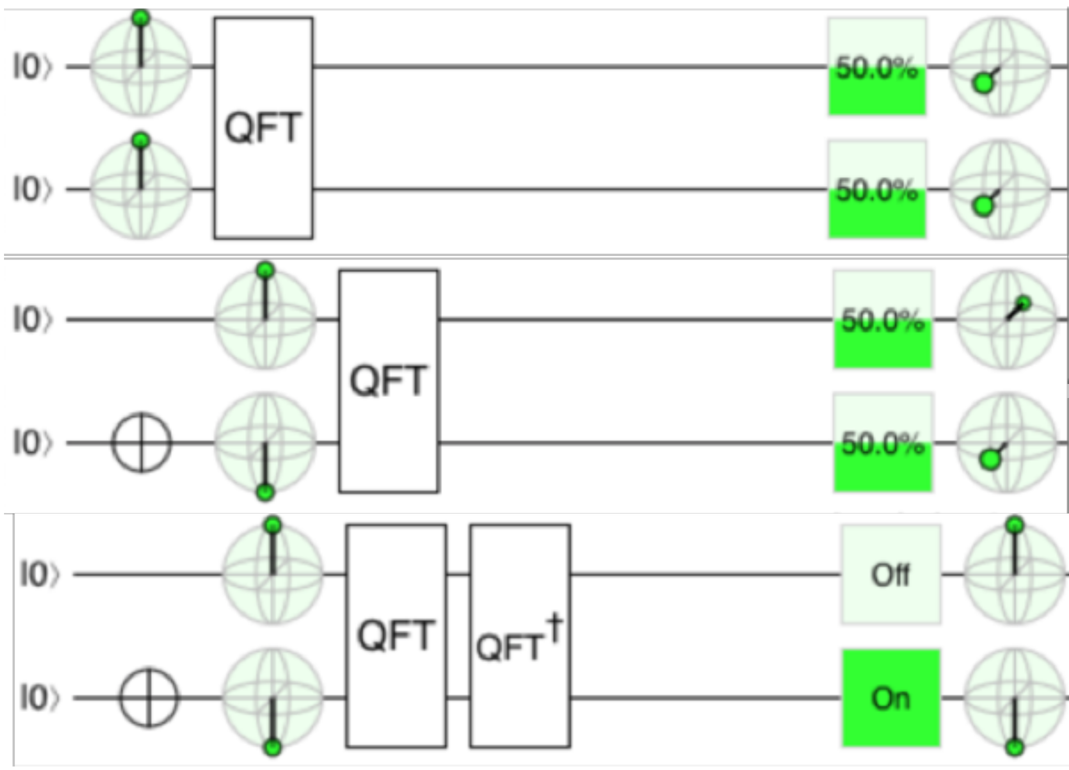


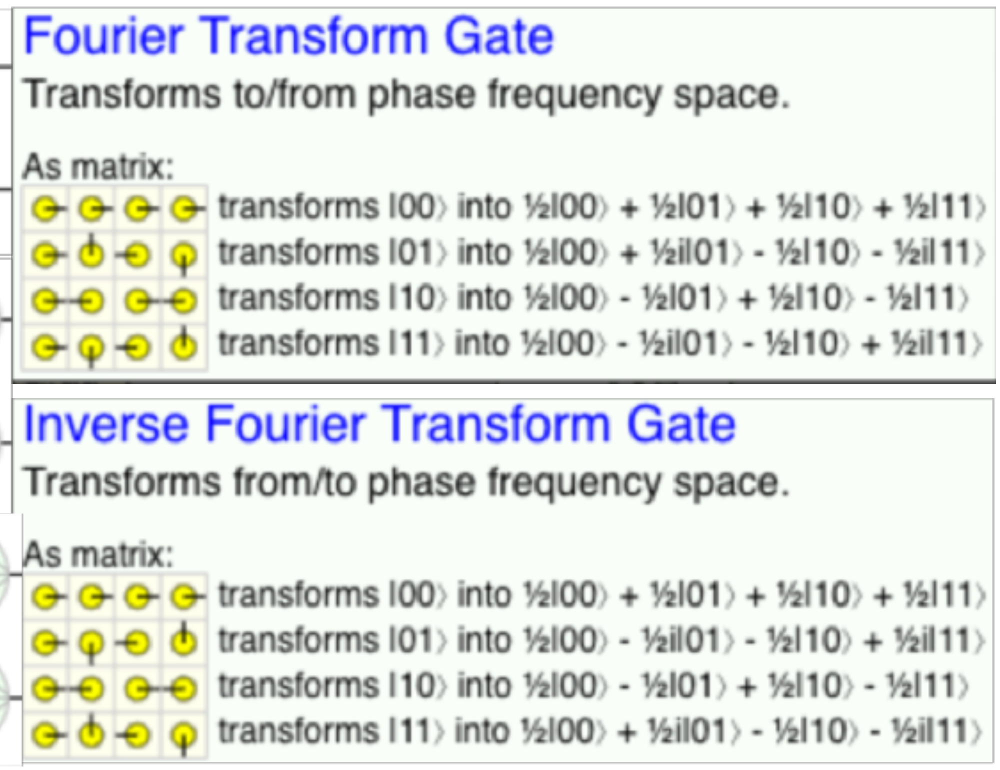
Inverts on 3rd qubit

SWAP, QFT/QFT-1, Reversibility









What is Needed in a Q Computer? Quantum Universality Q Gate Sets

- All you ever need for a quantum computer - could approximate any other quantum gate (unitary operation) at any precision
  - Toffoli and Hadamard already constitute a quantum *universal set*
- Other proven Q universal sets exist:
  - Hadamard, *S*, *T*, and CNOT; Toffoli, Hadamard, and π/4 –gate; CNOT, Hadamard, and π/8 –gate; Three-qubit Deutsch gate,…

- Solovay-Kitaev Theorem, any universal set of gates can simulate any other with *at most a polynomial increase in the # of gates*
  - If we're doing complexity theory, it really doesn't matter which universal gate set.
  - If we are designing a quantum computer it makes a lot of difference!

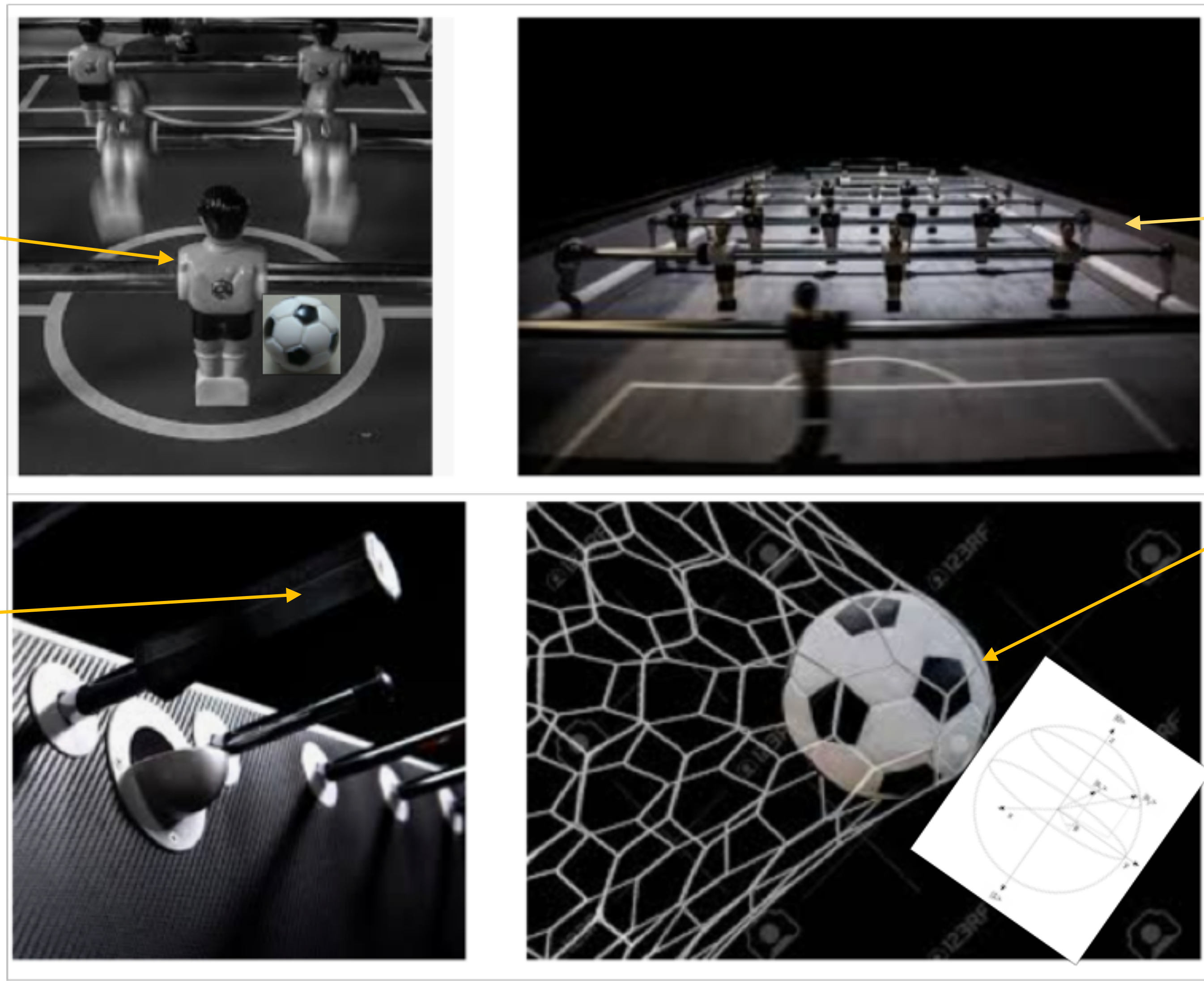
Quantum Circuits

- Sequence of quantum gates, each performing a unitary transformation on the Q state of registers

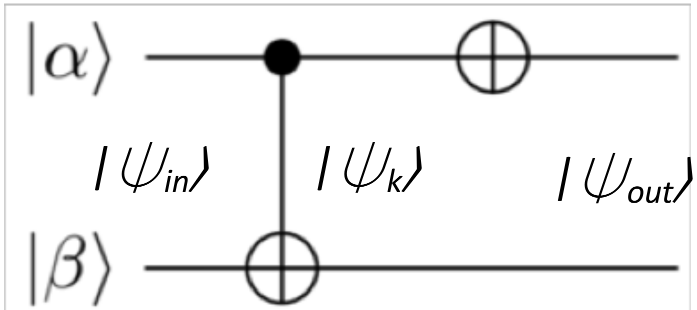
- Same number of inputs as outputs
- No loops! No cloning! But control flow...

- Read left to right; wires are qubits and symbols on each wire are gates; gates can act on one or more qubits
  - Hard to implement many-qubit gates …so often everything is built up from smaller gates (just like classical). E.g., n-QFT $O(n^2)$ in H, Ctrl phase shift gates
  - Connectivity – not all qubits may be used in CNOT
  - QEC – use many qubits as one logical one

Game of Table Soccer (w/ Block Sphere)?



Let's Analyze a 1st Simple Circuit:
(CNOT on joint two-qubit, then NOT on top)



Q Circuit

$|\psi_{in}\rangle = \alpha_0\beta_0\,|00\rangle + \alpha_0\beta_1\,|01\rangle + \alpha_1\beta_0\,|10\rangle + \alpha_1\beta_1\,|11\rangle$

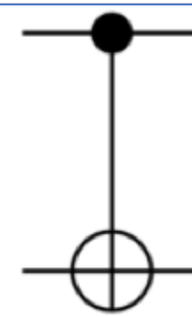
- Applying CNOT: matrix-vector multiplication using the unitary matrix of CNOT and joint input state vector

$$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\cdot\begin{pmatrix}\alpha_0\beta_0\\\alpha_0\beta_1\\\alpha_1\beta_0\\\alpha_1\beta_1\end{pmatrix}=\begin{pmatrix}\alpha_0\beta_0\\\alpha_0\beta_1\\\alpha_1\beta_1\\\alpha_1\beta_0\end{pmatrix}$$

$|\psi_{in}\rangle$ $|\psi_{k}\rangle$

Controlled NOT (CNOT)

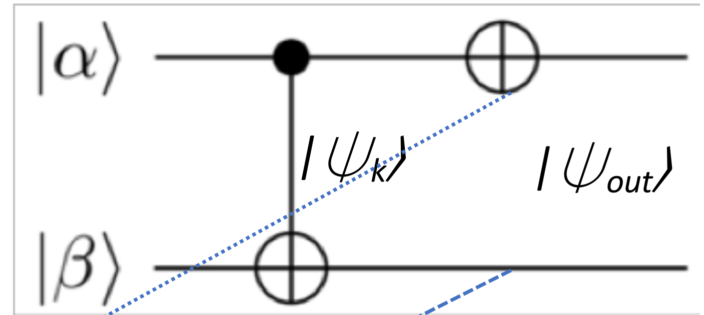
$$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$



- Next, apply NOT on first qubit only

NOT or X gate	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\oplus$

Q Circuit Example Contd.



- NOT or X gate is applied on top qubit, nothing on bottom

$$X \otimes \mathbb{1}_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (1)$$

$$|\psi_{out}\rangle = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_1 \\ \alpha_1\beta_0 \end{pmatrix} = \begin{pmatrix} \alpha_1\beta_1 \\ \alpha_1\beta_0 \\ \alpha_0\beta_0 \\ \alpha_0\beta_1 \end{pmatrix} \quad (2)$$

$|\psi_k\rangle$ $|\psi_{out}\rangle$

Quantum Parallelism - Intuition

- As a consequence of the linearity of quantum mechanics, gates are simultaneously applied to all basis states of a superposition at once
  - Classical SIMD operations, e.g., multimedia instruction set (ISA)
- If, for example, n qubits are in a complete superposition over all 2^n basis states, all possible outputs of a function f(x) can be calculated using only one function call:
- $$f\left(\sum_{i=0}^{2^n-1} c_i \left|x\right\rangle\right) = \sum_{i=0}^{2^n-1} c_i \left|f(x)\right\rangle \qquad (1)$$
- Key challenge: measurement collapses result to single basis state
  - To overcome this, quantum algorithms employ clever reduction schemes, making use of quantum interference effects

Outline (Beyond the Tangible?)

- Qubits, Quantum State
- Quantum Circuits
- *Q Algorithms*
- Q Compilers (Q Circuit Synth)
- Runtime Models
- Classical Microarchitecture

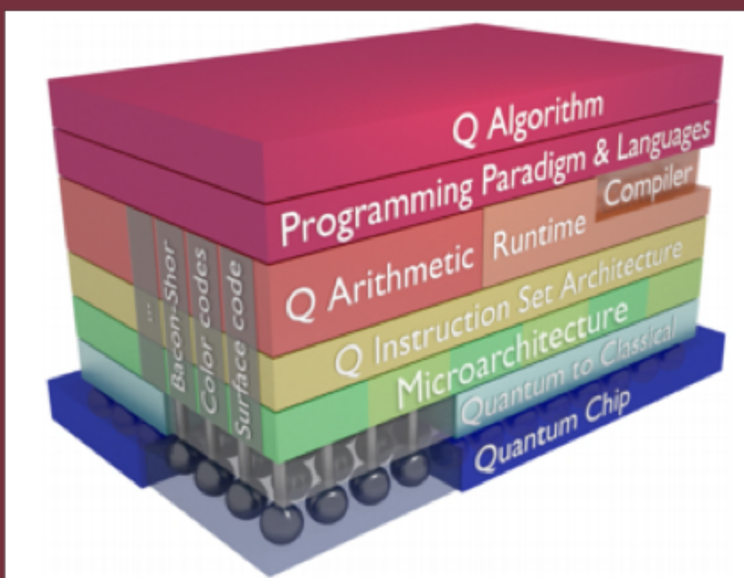
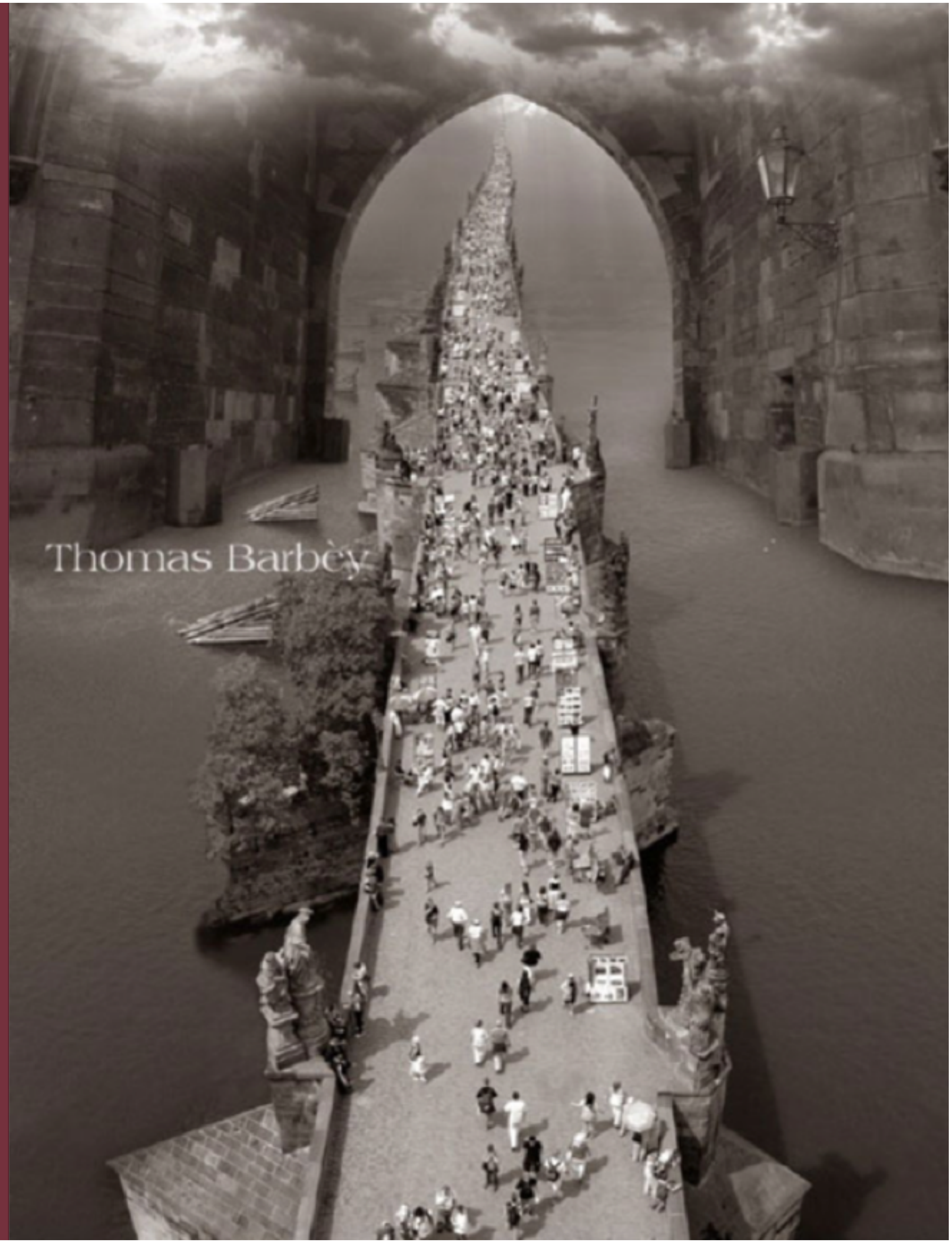
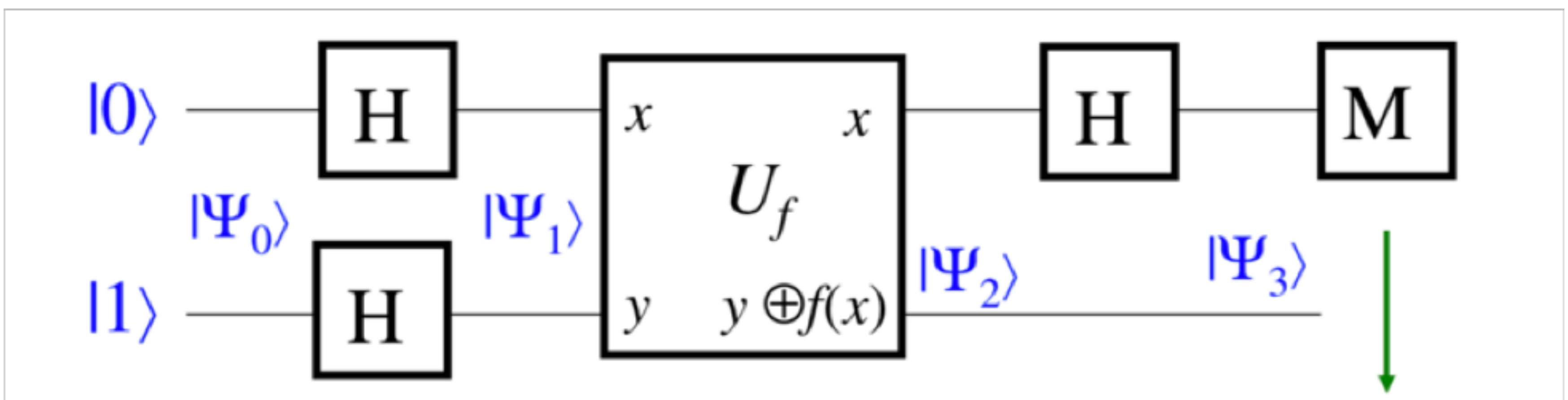


Figure from (Fu, 2017)



David Deutsch Algorithm and Circuit, 1985

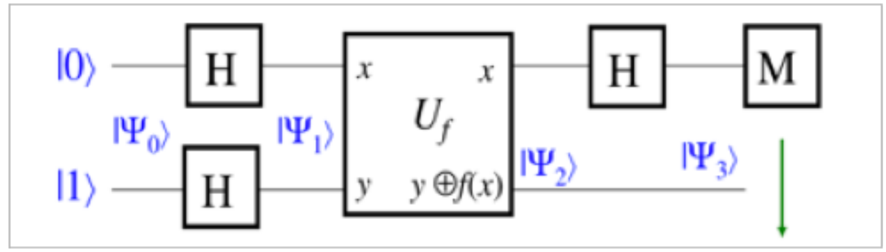
- Determine if a single-variable Boolean function f(x) is *constant* (f(0)=f(1)) or *balanced* (f(0) ≠ f(1)).
  - Classical version requires TWO runs of the algorithm
- Q Algorithm can evaluate f(0) and f(1) simultaneously
  - IDEA: quantum computer could extract the value of f(0)⊕f(1) at once (note this is 0 if f(x) constant and 1 if balanced)



Balanced = |1⟩
Constant = |0⟩

(Hayes, 2003)

Deutsch Algorithm Contd



Balanced = $|1\rangle$
Constant = $|0\rangle$

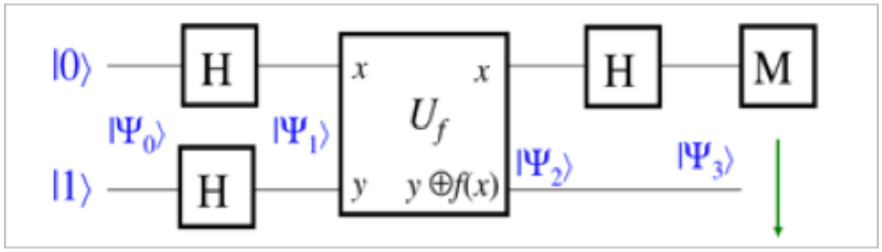
- Hadamards to create a *superposition of states*
  - After that a measurement would yield 50% likely one of the basis states

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

$$|0\rangle \text{ to } \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |1\rangle \text{ to } \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

  - Creates joint state of $|\psi_1\rangle$ ($|y\rangle=H|1\rangle$, $|x\rangle=H|0\rangle$)
- We then utilize a custom unitary transformation U_f to compute f(x)
  - Input is top qubit, bottom output has y xor f(x)
- Hadamard is used again to interfere $|\psi_2\rangle$ states, yielding $|\psi_3\rangle$
- Measure top qubit (aka control qubit) for result

Deutsch Algorithm Contd



Balanced = |1⟩
Constant = |0⟩

- Recall what inputs |x⟩ and |y⟩ were after initial Hadamards

|y⟩

$$|0\rangle \text{ to } \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |1\rangle \text{ to } \frac{|0\rangle - |1\rangle}{\sqrt{2}} \qquad (1)$$

$$U_f|0\rangle|y\rangle = \frac{U_f|0\rangle|0\rangle - U_f|0\rangle|1\rangle}{\sqrt{2}} = \frac{|0\rangle|f(0)\rangle - |0\rangle|f(0)\oplus 1\rangle}{\sqrt{2}} = (-1)^{f(0)}|0\rangle|y\rangle;$$

$$U_f|1\rangle|y\rangle = \frac{U_f|1\rangle|0\rangle - U_f|1\rangle|1\rangle}{\sqrt{2}} = \frac{|1\rangle|f(1)\rangle - |1\rangle|f(1)\oplus 1\rangle}{\sqrt{2}} = (-1)^{f(1)}|1\rangle|y\rangle. \qquad (2)$$

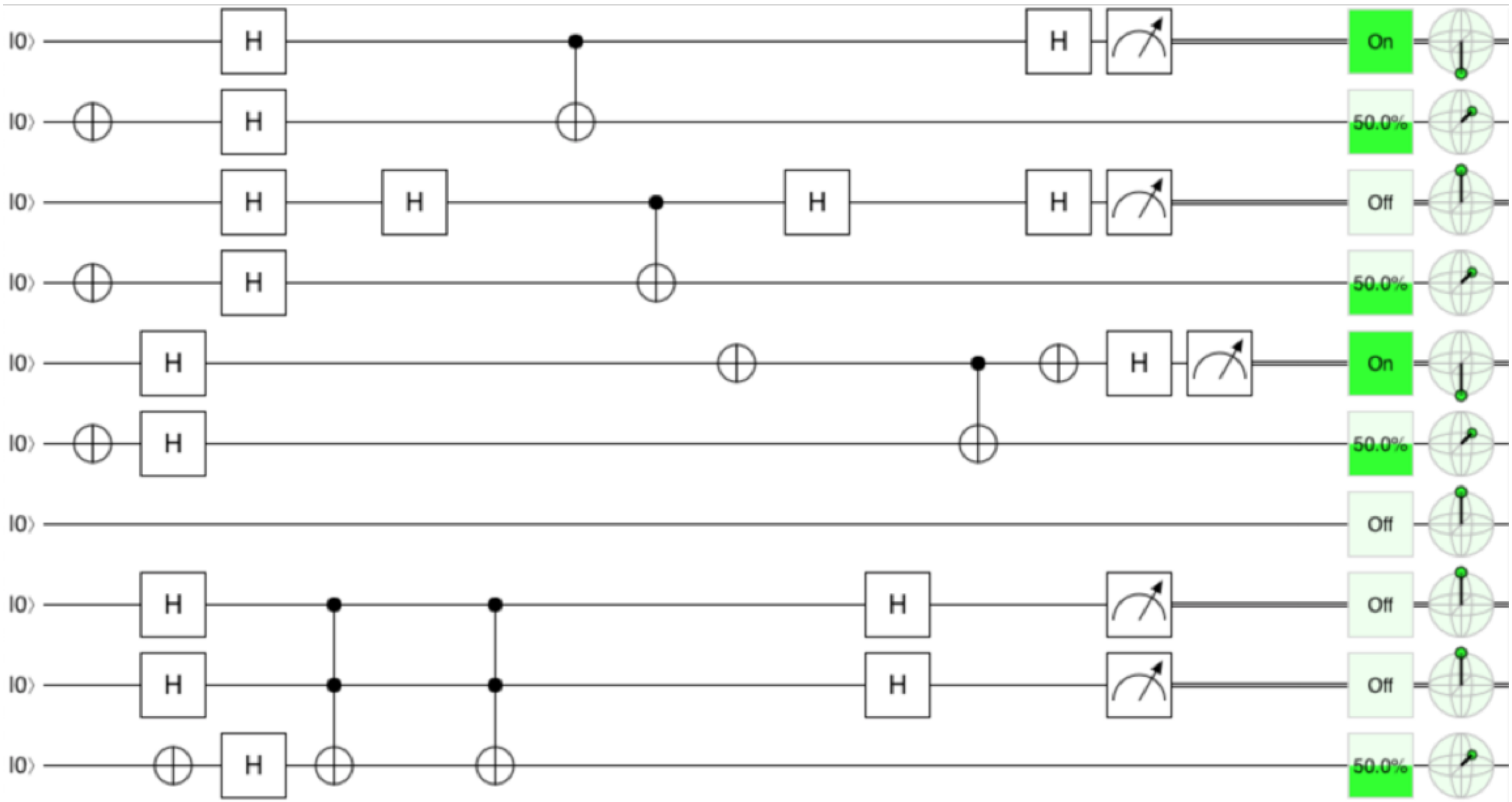
$$U_f|x\rangle|y\rangle = \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}|y\rangle = (-1)^{f(0)}\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}|y\rangle. \qquad (3)$$

$$(H\otimes I)U_f|x\rangle|y\rangle = \begin{cases} |0\rangle|y\rangle & , \quad \text{if } f(0) = f(1); \\ |1\rangle|y\rangle & , \quad \text{if } f(0) \neq f(1). \end{cases} \qquad (4)$$

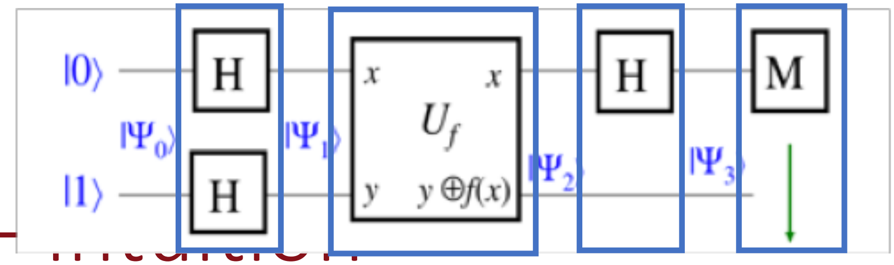
When measure to get result!

What we need to know

Deutsch Simulations



Deconstructing Deutsch –



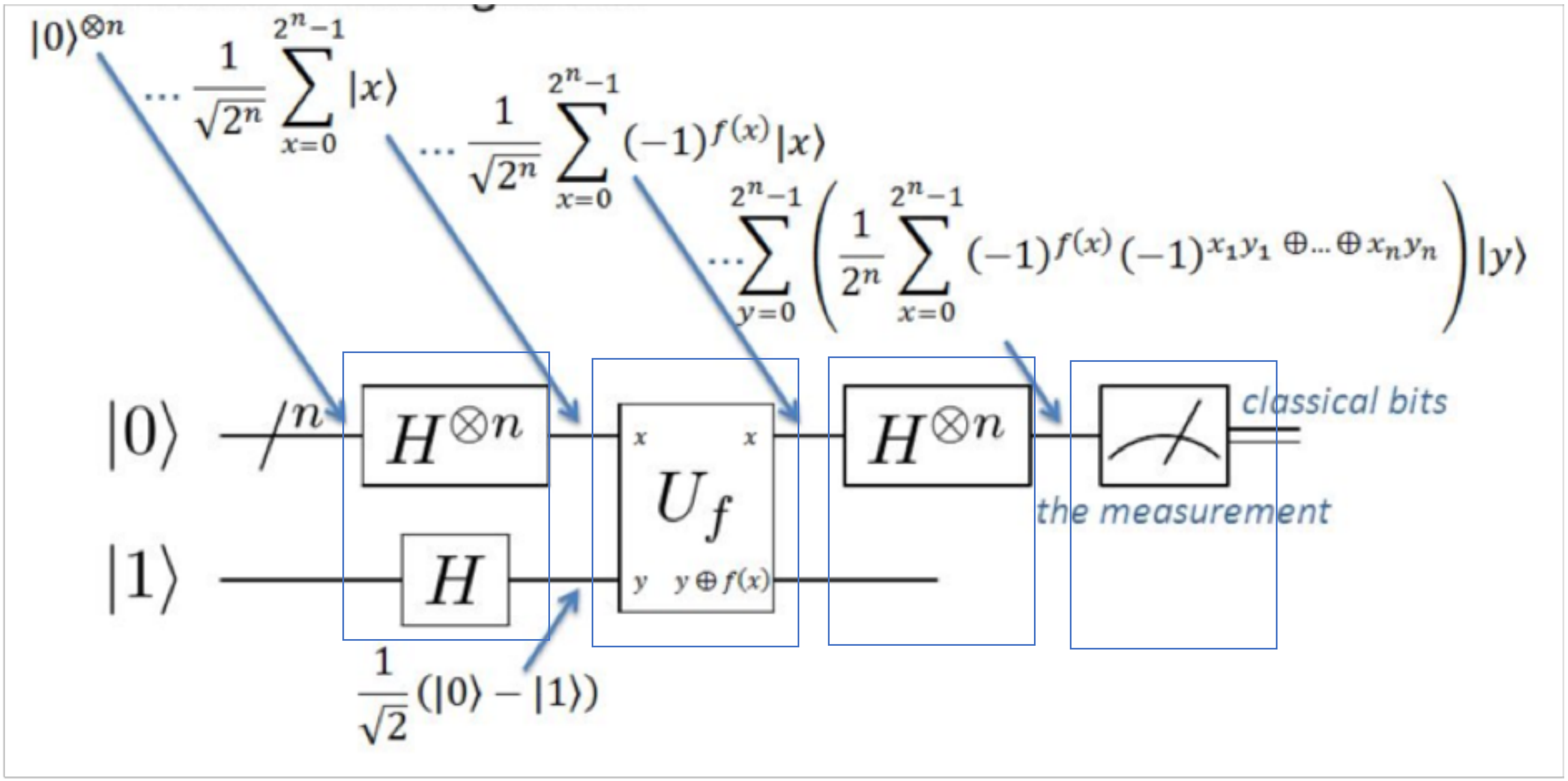
Balanced = |1⟩
Constant = |0⟩

- What did it take?

1. Algorithm/idea:
   - f(0) xor f(1) gives the result of whether f(x) is *balanced* or not
2. Cleverly using quantum computer:
   - Quantum parallelism – compute on superposition of basis states (prepared by/after the Hadamard gates) – SIMD like
     - Yielded result (U_f) had both f(0) and f(1) in it; in fact it had f(0) xor f(1)!
   - U_f is black box/oracle - f(x) but made reversible
   - Interfere/Bias - Why Hadamard at the end? Recognizing that |x〉 after U_f step is either H|1〉 or H|0〉 based on the result of f(0) xor f(1). Can map back to basis (preparing for a measurement) by H again: to |0⟩ if f is constant and |1⟩ balanced.

- Lesson: we need to think in terms of quantum parallelism & reduce result to global property combining simultaneous evaluations of f.

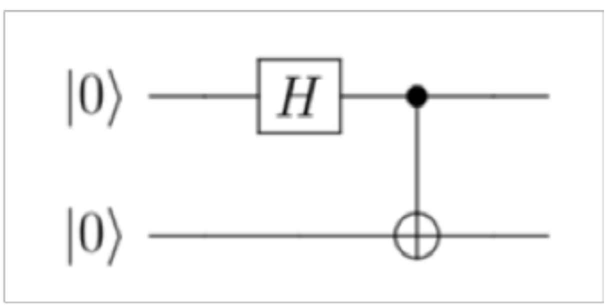
Deutsch-Jozsa (1996)



No Cloning, Power of Entanglement

- <u>No cloning theorem</u> – no ways to create a copy of $|\psi\rangle$ - this is a disadvantage vs classical computing
    - There is, however, a way to <u>assign a state to another qubit; needs entanglement</u> ... only one version can exist at the time

- <u>Entanglement</u> – state of qubits where they are correlated; one cannot express/decompose to tensor product of individual states (recall we used the tensor product for the joint state of two qubits)

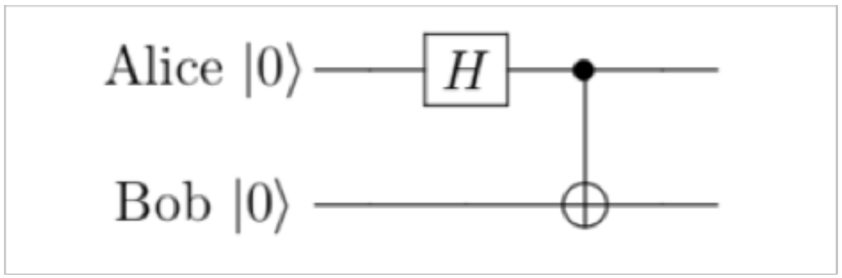


$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

- EPR pair – after Einsten, Podolsky, and Rosen

Secret Sharing, State Assignment, Teleportation

- One qubit is Alice's and one Bob's. They entangle them as shown.

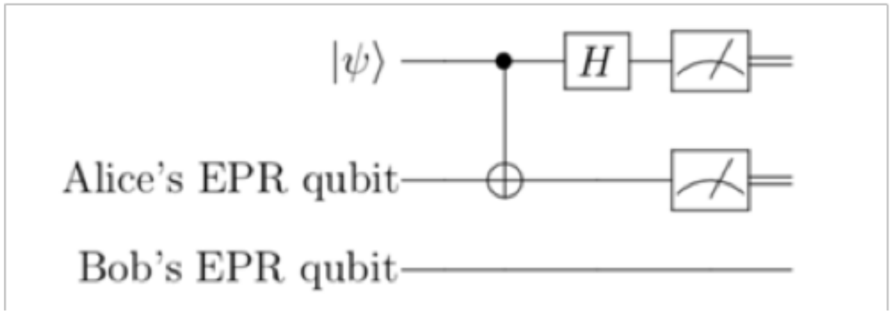


- They take them each home ☺
- Alice has another secret qubit $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, wants to give it to Bob
  - Problem:
    - α_0 , α_1 cannot be extracted – measurement would destroy $|\psi\rangle$
    - Even if possible, at what precision? Complex numbers…many many bits.
  - Is it possible?

(O'Donnel, 2015)

Secret Sharing, State Assignment, Teleportation

- Alice can send |ψ⟩ with following circuit



- Let us see joint state before the CNOT

$$|\psi\rangle \otimes \left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right) = \frac{\alpha_0}{\sqrt{2}}|011\rangle + \frac{\alpha_0}{\sqrt{2}}|000\rangle + \frac{\alpha_1}{\sqrt{2}}|100\rangle + \frac{\alpha_1}{\sqrt{2}}|111\rangle. \quad (1)$$

- Then passes through the CNOT + H gates; joint three-qubit state:

$$\frac{\alpha_0}{2}|000\rangle + \frac{\alpha_1}{2}|001\rangle + \frac{\alpha_1}{2}|010\rangle + \frac{\alpha_0}{2}|011\rangle + \frac{\alpha_0}{2}|100\rangle - \frac{\alpha_1}{2}|101\rangle - \frac{\alpha_1}{2}|110\rangle + \frac{\alpha_0}{2}|111\rangle. \quad (2)$$

- Now ALICE measures her two qubits (top two in figure)

Secret Sharing, State Assignment, Teleportation

- Note that independent of ALICE's measurement Bob's state is equal or close to Alice's $|\psi\rangle$!

Alice's

Alice's measurement	Prob. of meas.	Collapsed state
$\|00\rangle$	$\frac{\|\alpha_0\|^2}{4} + \frac{\|\alpha_1\|^2}{4} = \frac{1}{4}$	$\|00\rangle \otimes (\alpha_0\|0\rangle + \alpha_1\|1\rangle)$
$\|01\rangle$	$\frac{\|\alpha_1\|^2}{4} + \frac{\|\alpha_0\|^2}{4} = \frac{1}{4}$	$\|01\rangle \otimes (\alpha_1\|0\rangle + \alpha_0\|1\rangle)$
$\|10\rangle$	$\frac{\|\alpha_0\|^2}{4} + \frac{\|-\alpha_1\|^2}{4} = \frac{1}{4}$	$\|10\rangle \otimes (\alpha_0\|0\rangle - \alpha_1\|1\rangle)$
$\|11\rangle$	$\frac{\|-\alpha_1\|^2}{4} + \frac{\|\alpha_0\|^2}{4} = \frac{1}{4}$	$\|11\rangle \otimes (-\alpha_1\|0\rangle + \alpha_0\|1\rangle)$

Bob's

- There is a unitary transformation (gate) for each case to $|\psi\rangle$
  - E.g., second row needs a NOT gate
  - third one a $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ fourth one a $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

- Note this is a **secret sharing approach. Alice needs to call the result of her measurement and Bob applies corresponding U**
  - First verified in 1992 by Bennett. In 2012, Ma, et. al., performed quantum teleportation at a distance of 143 kilometers.

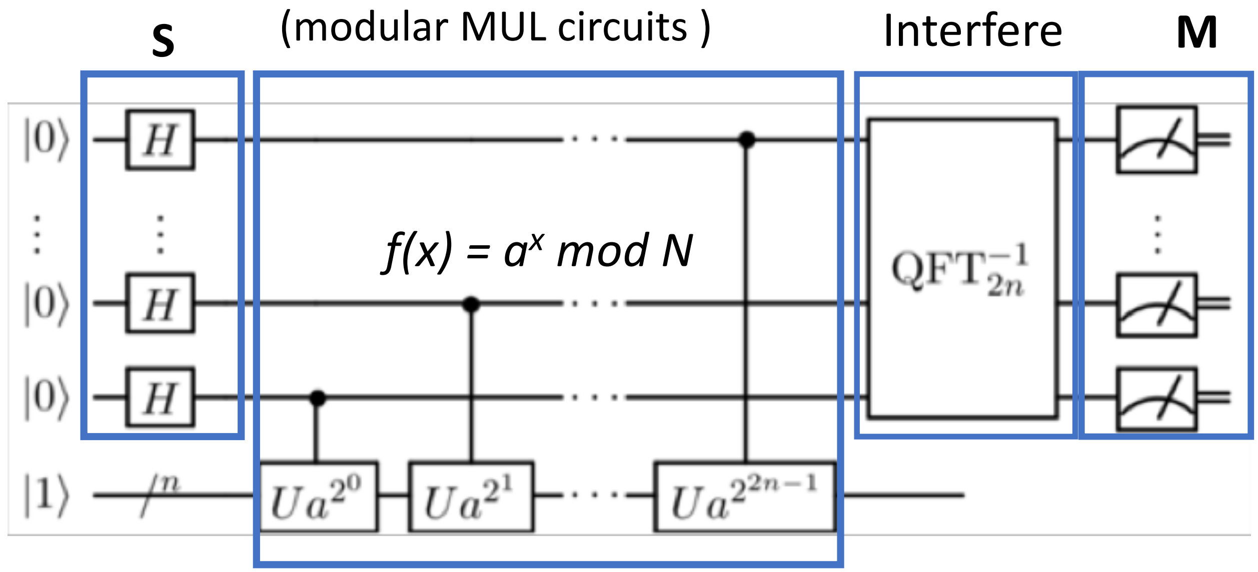
(O'Donnel, 2015)

Quantum Entanglement?

(Photography of Thomas Barbey)

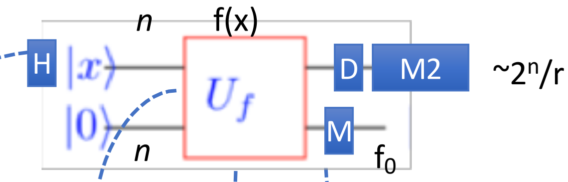
Shor's Algorithm 1994

Given an integer N,
find its prime factors



- Alg. Idea: calculate instead the *period r* of modular exponentiation.
- Math (simplified):
  - $f(x) = a^x \; mod \; N, a \; random$; period *r* smallest *int*. such as $f(x+r) = f(x)$
  - Classically compute *prime factors* as $\gcd\left(a^{\frac{r}{2}} + 1, N\right)$ and $\gcd(a^{\frac{r}{2}} - 1, N)$
  - How to calculate *r*?
- Quantum circuit **(see pattern S-Uf-I-M)** custom for each *N, a*
  - First part superposition, then next calculates f(x) with quantum parallelism
  - QFT gets you a number that is multiple of the inverse of r … (2^{2n}/r) => rest classically

Shor's Derivation (n top register assumed). Note I show for any periodic f(x)



$$|\Psi\rangle = H^{\otimes n}|0^n\rangle = \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle. \quad (1)$$

$$U_f|\Psi\rangle|0^n\rangle = \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle. \quad (2)$$

$$\left(\frac{1}{\sqrt{m}}\sum_{k=0}^{m-1}|x_0+kr\rangle\right)|f_0\rangle.$$

(collapsed joint state)

$f(x_0) = f_0$ $\quad m = \lfloor \frac{2^n}{r} \rfloor.$ $\quad$ *r is period* (3)

(after measurement of f(x))

$$D\left(\frac{1}{\sqrt{m}}\sum_{k=0}^{m-1}|x_0+kr\rangle\right)$$
$$= \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}\frac{1}{\sqrt{m}}\sum_{k=0}^{m-1}\omega^{(x_0+kr)y}|y\rangle$$
$$= \sum_{y=0}^{2^n-1}\omega^{x_0 y}\frac{1}{2^{n/2}\sqrt{m}}\left(\sum_{k=0}^{m-1}\omega^{kry}\right)|y\rangle. \quad (4)$$

$$\omega = e^{2\pi i/2^n}.$$

(after applying DQFT on top register)

$$\frac{1}{2^n m}\left|\sum_{k=0}^{m-1}\omega^{kry}\right|^2. \quad (5)$$

Maxed if $ry/2^n$ is an integer (6)

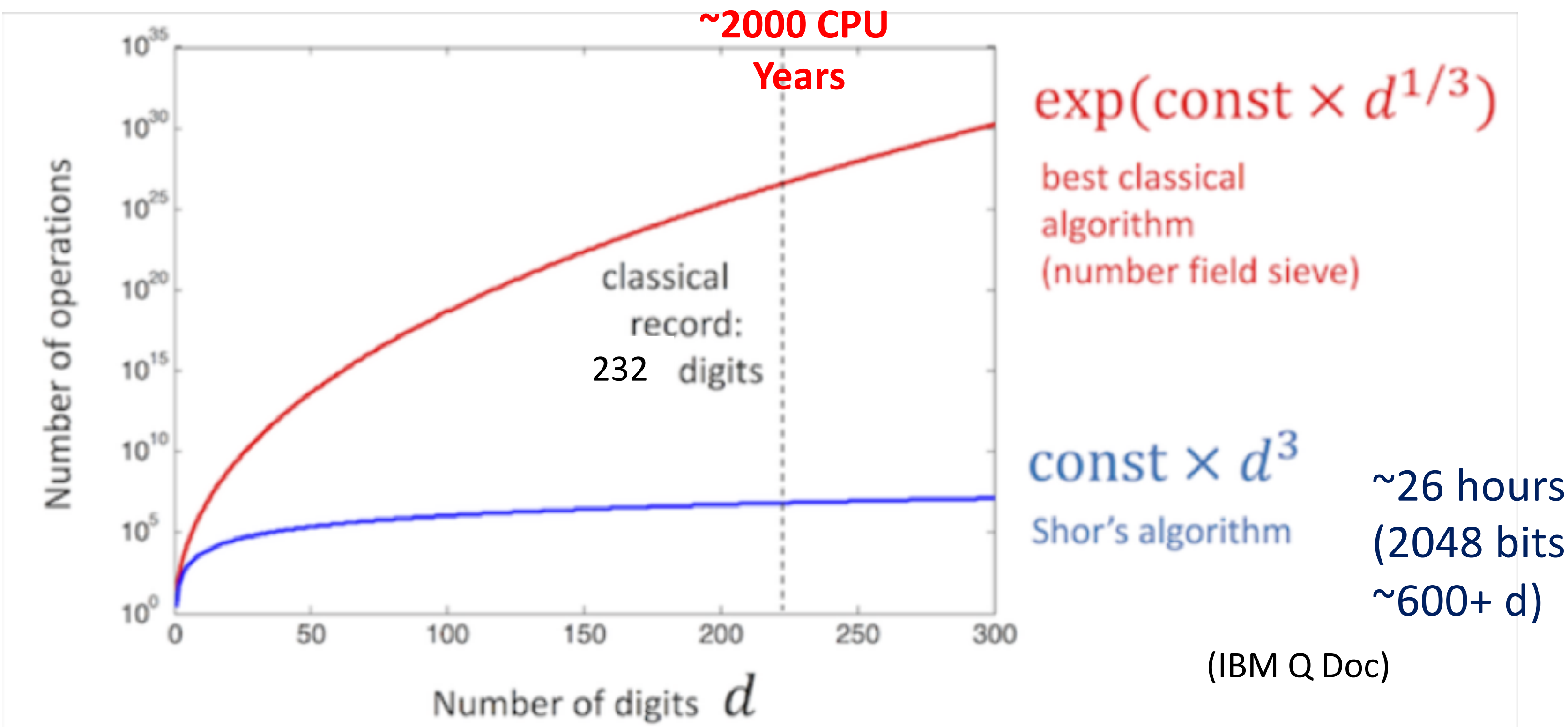
i.e., result will be multiple of $2^n/r$ (7)

(after measurement of top register)

Derivation follows (A. Dawar, Cambridge U)

Shor’s Algorithm Q Circuit

- Alexey Kitaev implementation: ~10d qubits (d is digits in N) and ~d^3 in gates



$$O\left(e^{1.9(\log N)^{1/3}(\log\log N)^{2/3}}\right) \rightarrow O\left((\log N)^2(\log\log N)(\log\log\log N)\right)$$

Impact on Today's Privacy, Security

- It will take around 26.7 hours for 2048 bits RSA (~600 digits) to be broken. Without fault tolerance it needs ~200M gates and ~6000 qubits.
- Also, a derivative of Shor's can be used to break ECC elliptic curve cryptography by computing discrete logarithms on a hypothetical quantum computer.
- The latest estimates for breaking a curve with a 256-bit modulus with 128-bit security level are 2330 qubits.
- Fault tolerance needs can significantly increase these numbers.

So, How Do You Design a Q Algorithm?

- Magic a la Ramanujan?
- Possible way to think (~ the 3Bs of Eagalman):

  - **Find** an f(x) as part of your problem, its **global property**, preprocess in classical

  1. **Setup superposition state(s) on input register**
  2. **Calculate simultaneously (SIMD) w/ quantum parallelism** f(x)
  3. **Bias/Interfere, Initial Measurement** - Bias state. Find way to expose metric across function outputs, a <u>global property</u> of f(x), to help solve what you need.
     - Measure f(x) … typical entanglement (f(x), x), f(x) property *reveals* itself in x.
  4. **Ready to Measure Result**
     - *The basis state that is most likely, must indicate your global property*

  - Finish classical

Design Requirements for a Q Computer

- The DiVincenzo Criteria for Q Hardware:
  - (i) scalable physical system with well-characterized qubits,
  - (ii) ability to initialize the state of the qubits,
  - (iii) "universal" set of quantum gates,
  - (iv) long relevant decoherence times, much longer than the gate-operation time,
  - (v) a qubit-specific measurement.

- Additionally, for overall system:
  - Q Compiler to generate from Q Program/Classical - sequence of gates w/ QEC
  - Classical CPU + Co-Processor w/ Controller to manage Q Hardware IO
  - Adequate fault tolerance - the *threshold theorem*:
    - Arbitrarily long Q computation with arbitrary reliability can be executed, if the error rates of Q gates are under a *critical accuracy threshold*. If decoherence only source, then robust computation requires decoherence 10^4 times longer than 1 Q gate delay

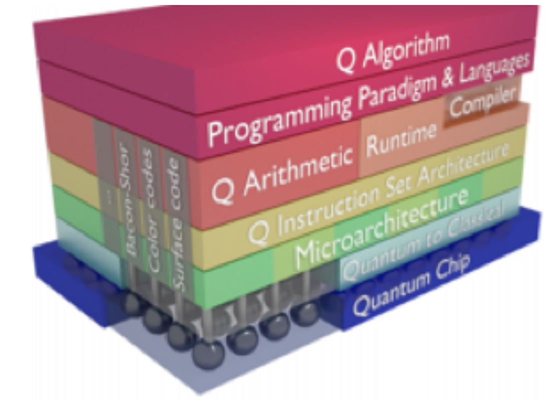
(E. Dennis, arxiv)

Why Use Q Tools/Compiler?

- Why not design algorithm and then create circuit manually?

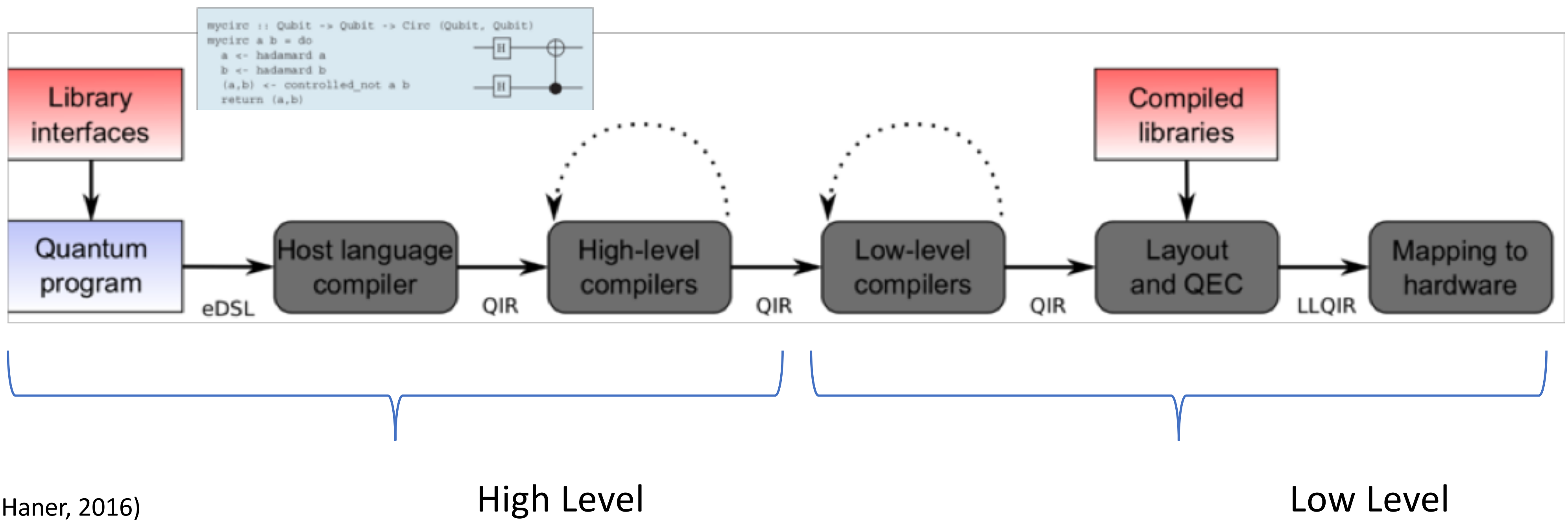
- Does not scale

- Beyond <u>convenience</u> (complexity), 3 primary problems to optimize
  1. Space – # of qubits needed
     - Gate sequences, Q oracles, QEC, manage ancillas/uncomputes; classical
  2. Time – decoherence, sequence of gates, precision
  3. Errors – how much error correction and where?
     - Maximize probability of correct interpretation



Q Compilers

- Classical code and embedded quantum program (EDSL)
- After the high-level compilation stage, the code consists of quantum gates, inlined library functions, and library calls to be resolved later
- Low- level compiler is to translate all quantum gates into sequences of gates from a discrete, technology-dependent set., & add QEC

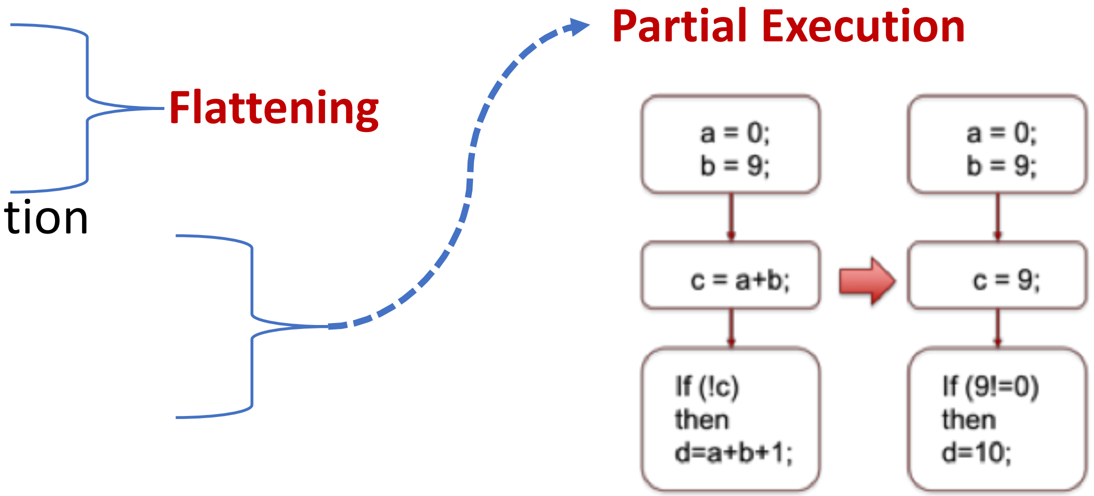


(Haner, 2016)

Prepare for Q Circuit Instantiation and Q Program Analysis

- <u>Prepare circuits</u>: have *inputs and oracle gates* <u>specified, +QEC</u>
  - Q Libraries can be ready but oracle needs to be generated (like in RTL)
  - May need to *generate multiple circuits at different sizes, inputs*

- Function inlining
- Loop unrolling
- Constant folding
- Constant propagation

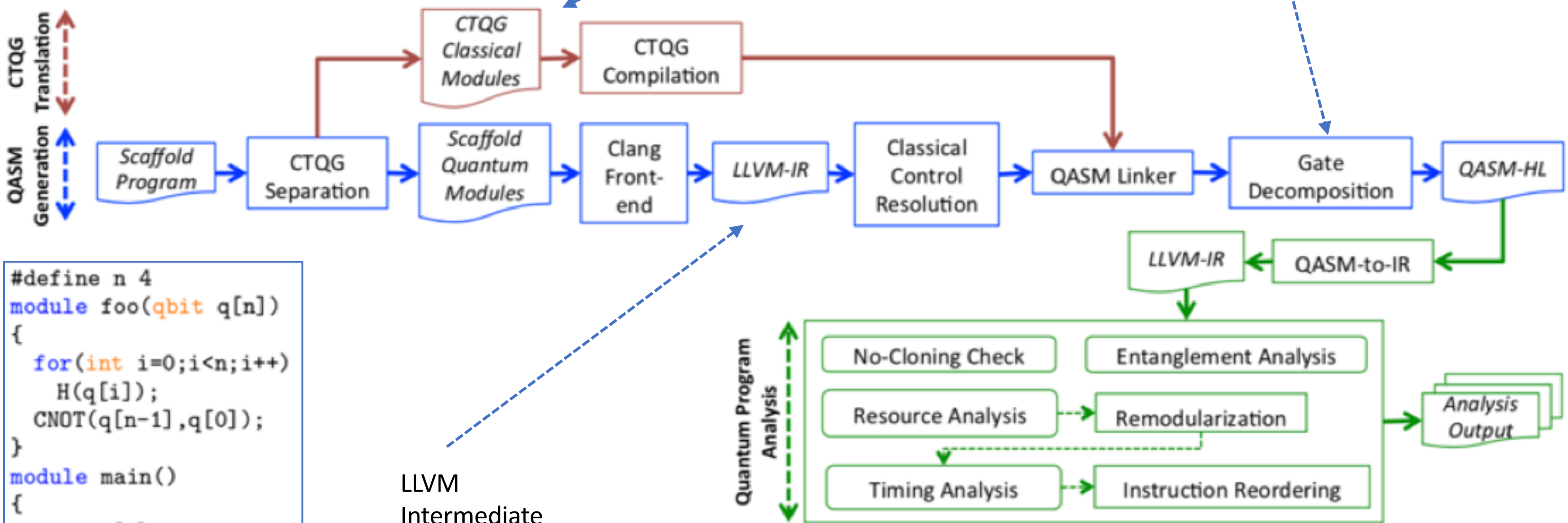


- <u>Quantum Program Analysis</u>: data-flow to check on entangled states, verify incorrect copying/assignment, resource utilization (incl ancillas), check uncomputes, estimate critical path

ScaffCC (C Front)

Allows using of classical gates

Toffoli, rotations replaced w/ subcircuits




```
#define n 4
module foo(qbit q[n])
{
  for(int i=0;i<n;i++)
    H(q[i]);
  CNOT(q[n-1],q[0]);
}
module main()
{
  qbit b[n];
  foo(b);
}
```

LLVM Intermediate Format

Q Program Analyses on LLVM

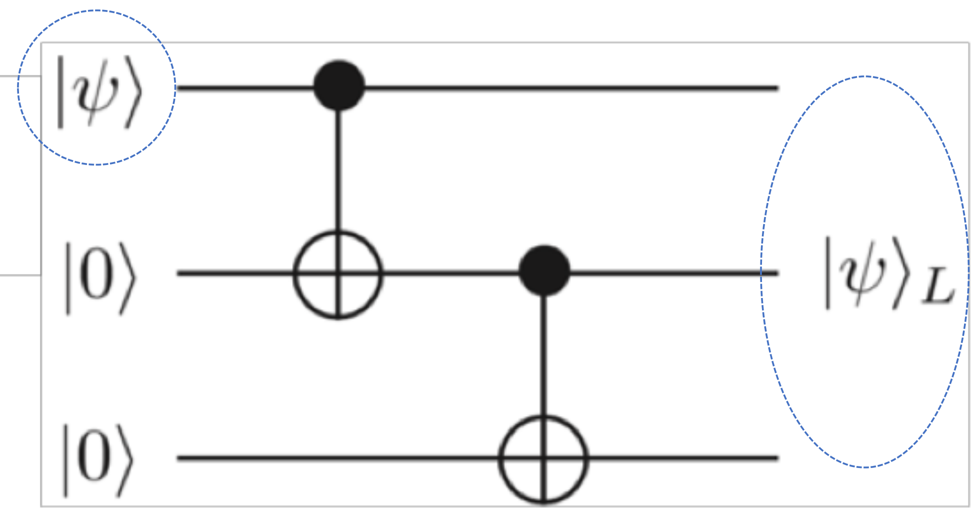
Scaffold code

(Abhary, 2014)

Low-level Optimizations, Error Handling

- Logical qubits to physical qubits (e.g. for QASM or hardware)
- Need to add redundancy/fault tolerance for state errors
  - Success – correct “interpretation” of results
  - Quantum Error Correction codes – bitflips and phase, QEC Logical qubits
    - Challenge – no state copies can be kept, hard to judge/detect due to measurement; need to use ancillas and *syndrome* information

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle_L + \beta|1\rangle_L = \alpha|000\rangle + \beta|111\rangle = |\psi\rangle_L .$$

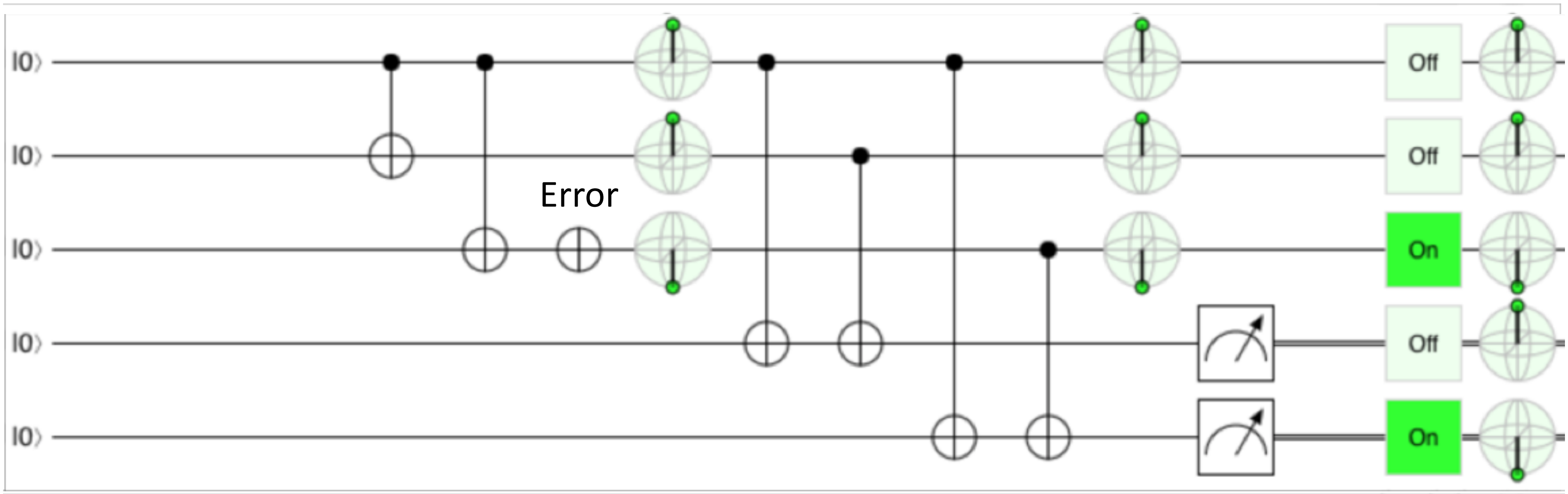


(Nemeto, 2013)

Part of 9-qubit code, Shor’s, 1995

EC-1-bit flip error

Fragment of Shor's Syndrome Based QEC- with Q Circuit Simulator



Ancilla Measurement: $\|00\rangle$,	∴ Clean State
Ancilla Measurement: $\|01\rangle$,	∴ Bit Flip on Qubit 3
Ancilla Measurement: $\|10\rangle$,	∴ Bit Flip on Qubit 2
Ancilla Measurement: $\|11\rangle$,	∴ Bit Flip on Qubit 1

Error Detection w/ Post-Select, Reset, Rerun

- Detect but not correct (00 state = no errors); no info. to correct

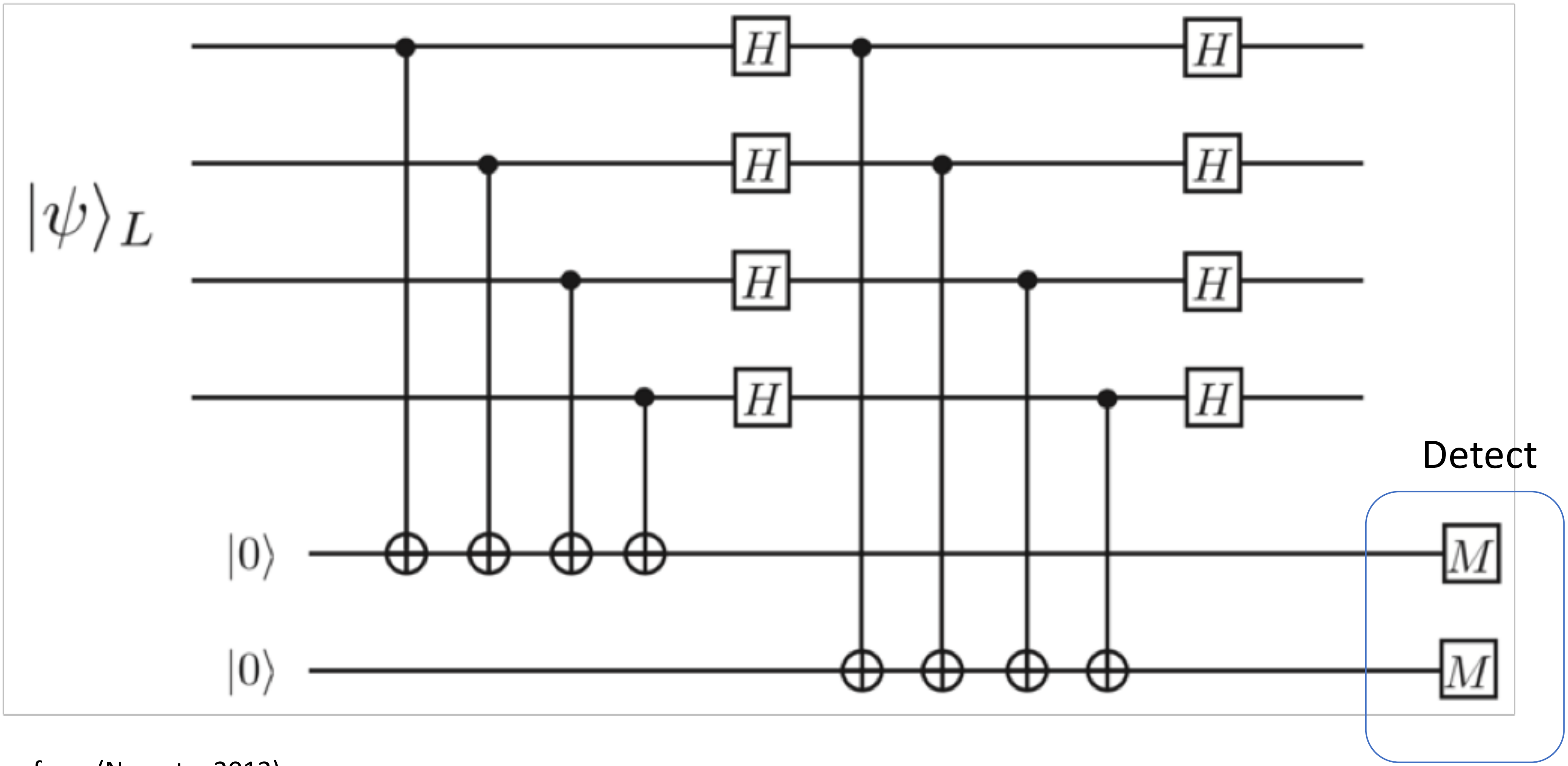


Figure from (Nemeto, 2013)

Stabilizer Formalism – Stabilizer Codes

- Enables easy synthesis of correction circuits/logical operations in Q compilers. E.g., 7 physical qubits for one logical. 6 Stabilizers are measured for single X (bit flip) or Z (phase flip) errors.

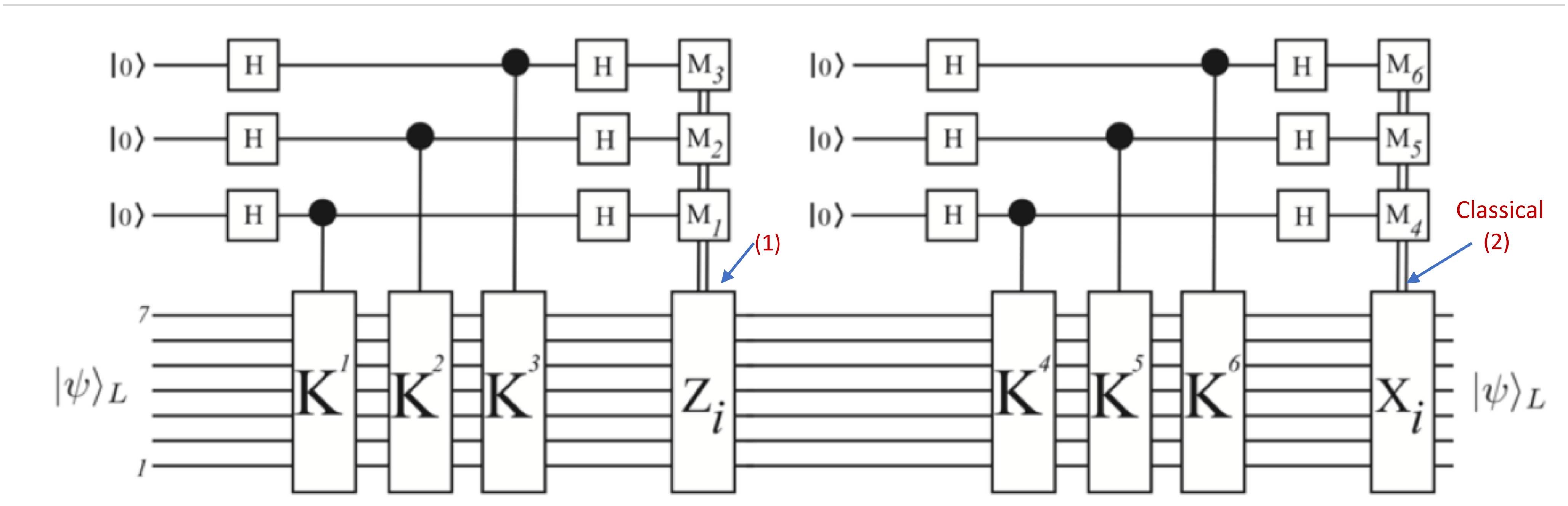


Figure from (Nemeto, 2013)

7-qubit *Steane* code, 1996

Envisioned Runtime Models

- Batch (Static)
  - Set up and run Q algorithm interleaved with regular codes statically
  - Trace based and w/ constant propagation – need circuit specified out
  - TBytes of code
- Dynamic Execution
  - Dynamically choose what to run - Interleave classical + Q arbitrarily
    - E.g., Post-selected error correction; which bit to add the X gate on out of 3
- *Dynamic Compilation*
  - Generate new circuit based on result of Q subroutine (measurement)

Is Dynamic Compilation a Must?

- Static version not always possible or best
- Phase estimation relying algorithms – phase needed to be extracted for next part of Q algorithm
  - Like in solving Linear Systems, Shortest Vector Algorithm
  - **<u>Angle of rotation depends on a measurement</u>**
- Note that even in Shor's; assumed that a static approach would work as phase rotations can be pre-generated (again code size! TBs of code, 1000s of high precision rotations)

$$\left\{R_z(\frac{\pi}{2^1}), R_z(\frac{\pi}{2^2}), R_z(\frac{\pi}{2^3}), ..., R_z(\frac{\pi}{2^k})\right\}$$

- Tight coupling of QHW-CHW in modern QEC like Surface codes
  - Adjust next phase of computation, compensate for errors detected

Code Size, Runtime Cost of Precision & QEC

- Ground State Estimation algorithm for Fe_2S_2 for example requires 10^{14} rotations, each approximated with 10^5 gates. That is 10^{19}!!!
- Dynamic compilation for a given phase precision based on static Solovay-Kitaev algorithm too slow
  - Initial work by Kudrow 5X improvement by classical optimizations
  - FP precision may not be adequate…

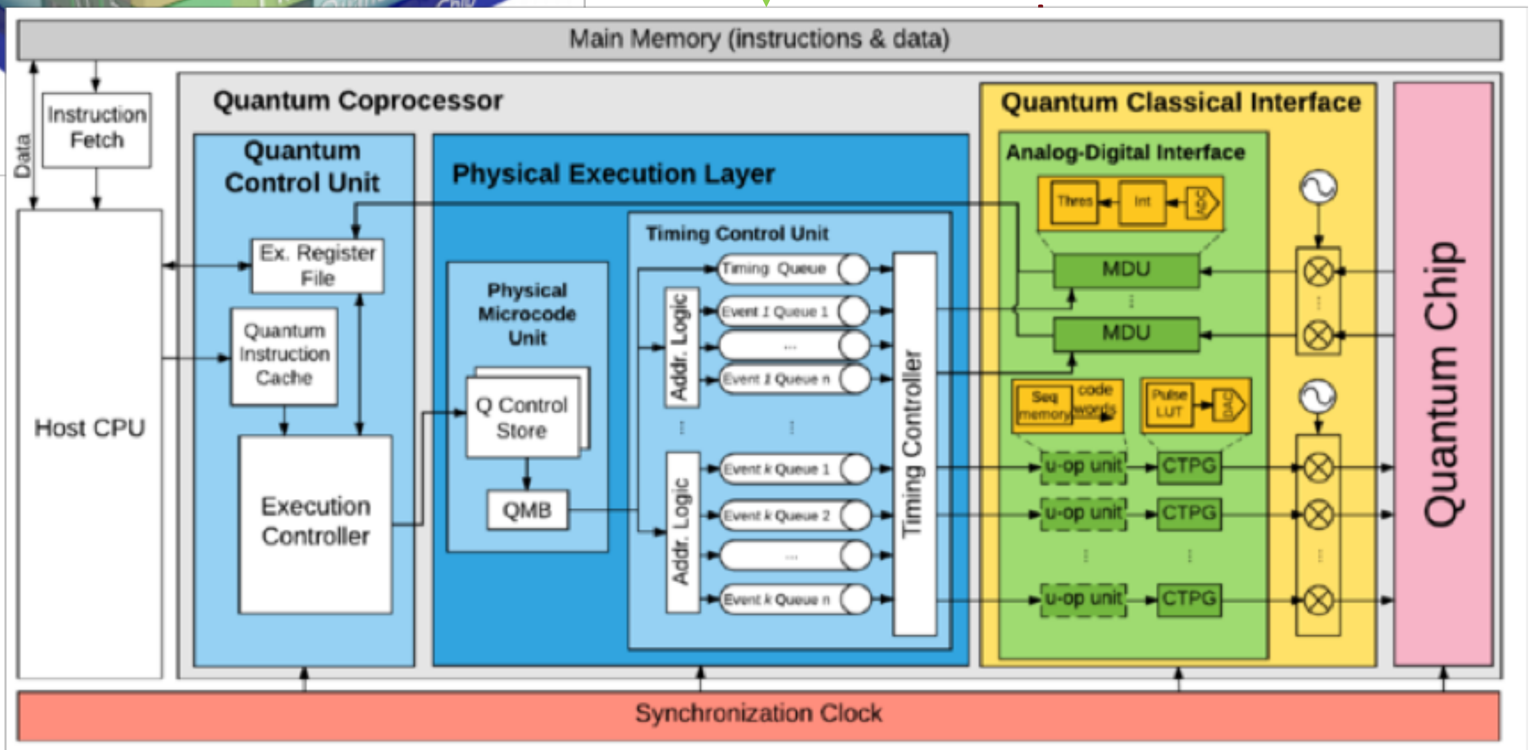
Technology	Experimental	Calculation
Ion Trap	10 μs [5]	1.0 μ s
Neutral Atom	31 μs [28]	0.915 μ s
Quantum Dot	2.6 ns [3]	1 ns
Superconductor	20 ns [23]	1 ns
Photons	15 ps [31]	1 ns

T Gate time; Bacon-Shor L2 QEC - 2 + orders slowdown

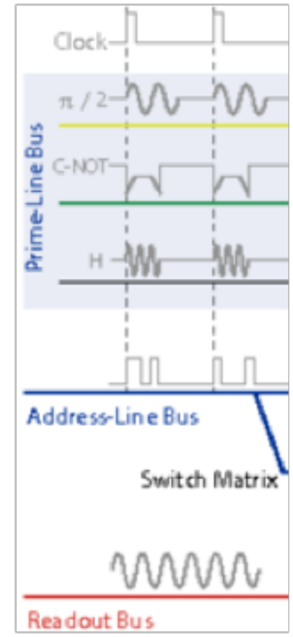
"… the high error-correction overhead of quantum computers can make the crossover point between polynomial and exponential performance occur at 100 years of computation"

(Kudrow et al, 2013 ISCA)

Microarchitecture Support – QuMA - Adds



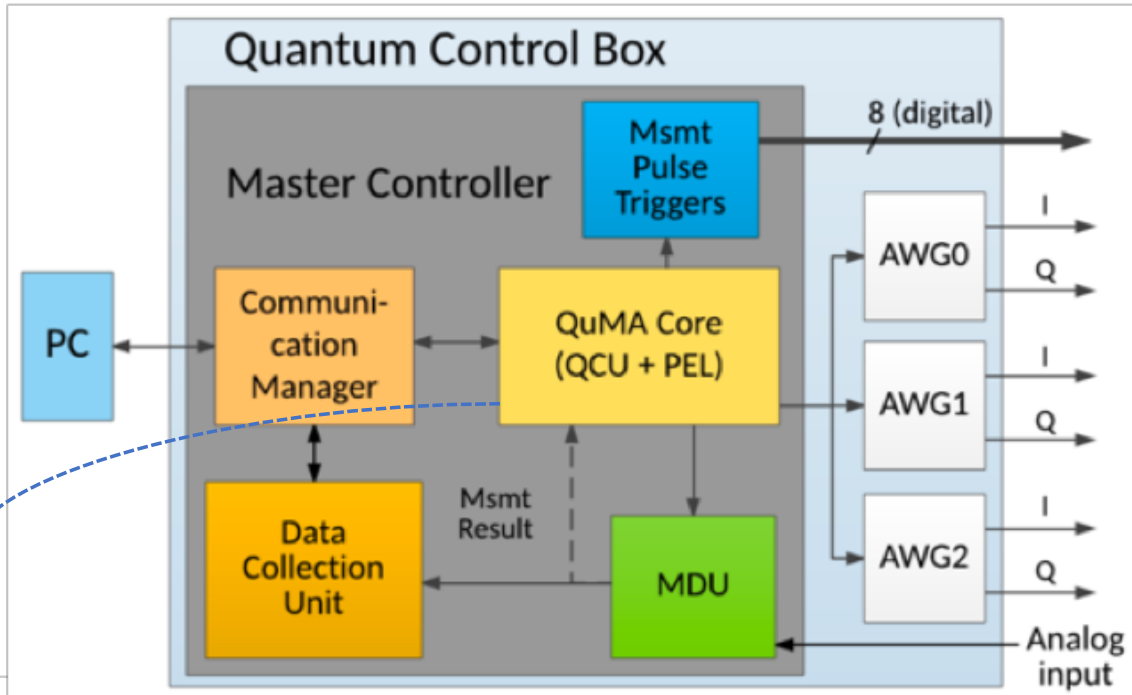
(Fu, 2017)



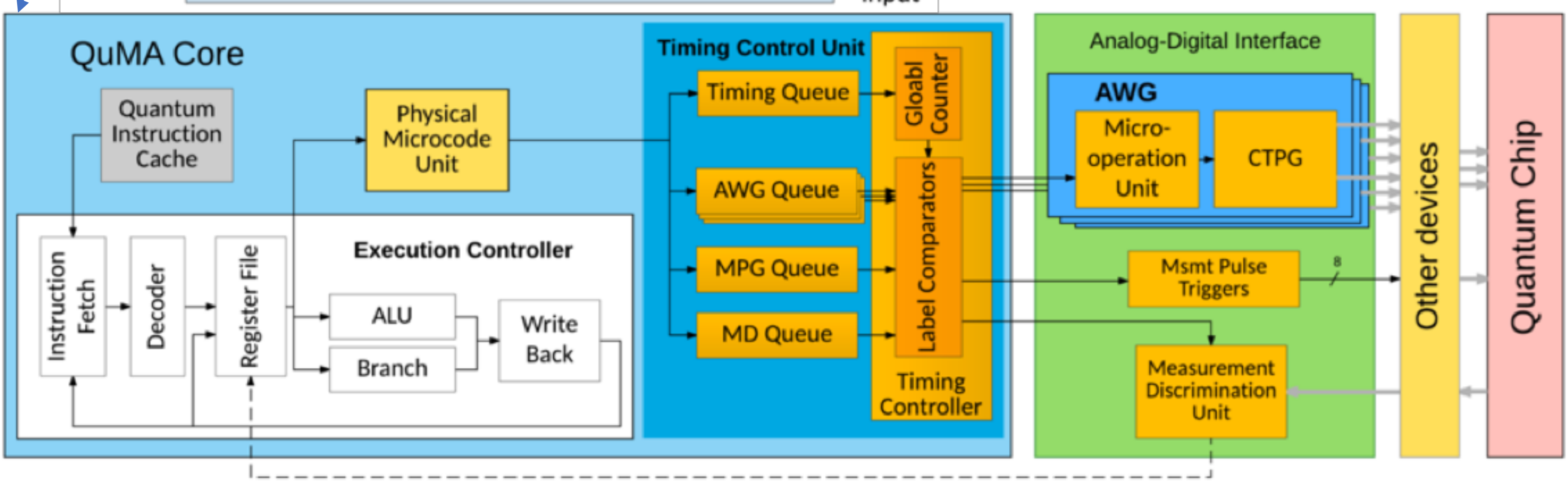
(Hornibrook, 2015)

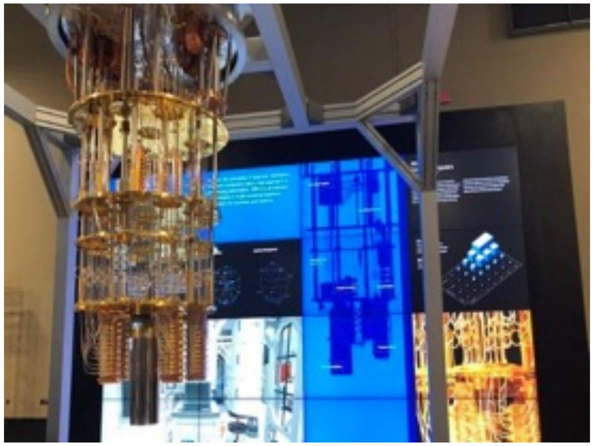
- Four concepts: (i) code-words (machine code), (ii) queue-based based timing; (iii) quantum instructions -> microcode -> code-words; (iv) QuMIS is a quantum microinstruction set. Validated on one super-cond Qb.
  - Manage events/timing, create analog control signals (pulses, wafeforms, coded in code-words)

Implementation w/ 4 FPGA Boards (Fu, 2017)



- Master w/ Altera Cyclone V 5CEFA9 FPGA chip, 8-bit ADC (for data from Q chip)
- 3 X boards - two-channel arbitrary waveform generator (AWG) each on Terasic DE0-Nano w/ Altera Cyclone IV EP4CE22F; 2X 14-bit DAC for qubit pulses
- 20 ns from codeword to ctrl





State-of-the-Art (Dec 2018)

- Simulator (best 56 qubits)
  - Microsoft's with Intel's AVX extensions. Plans "Brainwave" FPGA-based AI accelerator retargeted for Q. ATOS builds custom acceleration.
  - IBM 56-qubit general-purpose q system on a supercomputer. Harvard-MIT and the CALTECH simulated a 51-qubit quantum computer, but was not general-purpose. European researchers from Jülich Supercomputing Centre, Wuhan University, and the University of Groningen simulated a 46-qubits.
- Quantum Hardware (best 50 qubits at IBM, 72 at Google)
  - IBM 20-qubit chip in late 2017, internally tested a 50-qubit chip
  - Intel showed a 49-qubit chip at 2018 at the Consumer Electronics Show (CES).
  - Rigetti has19-qubit chip available for cloud access
  - Google Bristlecone 72 qubit in 2018 March
- Tool chain (widely available)
  - IBM Q Network QISKit API (1500 Univ, 35 papers), University tools based on LLVM (Scaffold, Haskell), NVIDIA QUDA, Microsoft Visual Studio IDE

Summary (+ Answer to Title Question)

Classical Computer	Quantum Computer (Future)
Bits (N bits “store” one of 2^N)	Qubits (Linear combination 2^N basis st)
Universal gate sets, Turing Machine, von Neumann, many-cores; Boolean	Quantum Universal gate sets, Quantum Turing Machine; Hilbert space
Classical inputs, outputs. I/O digital	Same nr of classical inputs, outputs. No loops, no copies. Reversible. I/O analog
Limited Data (SIMD), Instruction (ILP) and Task/Function Parallelism (TLP/FLP) Pipelining, caching.	(Almost) unlimited 2^n SIMD **Quantum Parallelism** (Tricks w/ entanglement, interference. Result: just global property)
Reliability: Perfect (almost)	High Error Rate. Surface QEC: 10^3-10^4 X overhead. 95%+ of work for errors. Skepticism on large QC **w/ 10M-1B Qb.**
Advanced Compilers	Initial flows – must deal w/ **code size explosion, dataflow likely hard**. Q Co-processors to manage *tight* dynamic.
Easy to write code, design algorithms	Can be mastered 😎! Do we need to find killer applications for small QC?

Summary on Security, Privacy

- RSA, ECC easily broken would large scale quantum computer materialize.
  - All email and messaging apps that rely on encryption alone.
  - Financial transactions, defense related communications.
  - All internet traffic.
- New methods are needed to encrypt or by utilizing ideas that do not fully rely on digital solutions alone
  - see EPRIVO physical separation approach.
  - Post quantum encryption algorithms.

Q U E S T I O N S

Thomas Barbey photography

References

1. Ian Glendinning, Notes on Rotations on the Bloch Sphere, Univ. of Texas, 2010
2. Anuj Dawar, Quantum Computing, Lecture Series, Cambridge University, UK.
3. Thomas Haner, Damian S. Steiger,Krysta Svore, and Matthias Troyer, A Software Methodology for Compiling Quantum Programs, arxiv: 1604.01401v2, 2016
4. John Hayes, Tutorial on Quantum Computing, DAC 2003.
5. Eisuke Abe, Quantum Circuits, School on Quantum Computing, Department of Applied Physics and Physico-Informatics, and CREST-JST, Keio University
6. Ryan O'Donnel, Quantum Computation, Lecture 3: The Power of Entanglement, CMU
7. Ali Javadi Abhari, Shruti Patil, Daniel Kudrow, Jeff Heckey, Alexey Lvov, Frederic T. Chong, Margaret Martonosi, ScaffCC: A Framework for Compilation and Analysis of Quantum Computing Programs, ACM CF'14, Italy.
8. Patrick J. Coles, Stephan Eidenbenz, Scott Pakin, Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, Nathan Lemons, Shizeng Lin, Andrey Lokhov, Alexander Malyzhenkov, David Mascarenas, Susan Mniszewski, Balu Nadiga, Dan O'Malley, Diane Oyen, Lakshman Prasad, Randy Roberts, Phil Romero, Nandakishore Santhi, Nikolai Sinitsyn, Pieter Swart, Marc Vuffray, Jim Wendelberger, Boram Yoon, Richard Zamora, and Wei Zhu, Quantum Algorithm Implementations,
   Los Alamos National Laboratory, Los Alamos, New Mexico, USA, Apr. 2018.
9. Nemeto, Quantum Error Correction for Beginners, Reports on Progress in Physics · June 2013
10. http://www.scholarpedia.org/article/Quantum_Computation#Deutsch.27s_Algorithm
11. R J Renka, Quantum Circuits, Univ of Texas, 2018,
12. Programming languages and compiler design for realistic quantum hardware, Frederic T. Chong, Diana Franklin & Margaret Martonosi, Nature Insight, doi:10.1038/nature23459.
13. Jaden Pieper and Manuel E. Lladser Quantum Computation, Scholarpedia, 13(2):52499. doi:10.4249/scholarpedia.52499 revision #186567, University of Colorado, Boulder, CO, USA, 2018.
14. Darshan D. Thaker Tzvetan S. Metodi Andrew W. Cross Isaac L. Chuang Frederic T. Chong, Quantum Memory Hierarchies: Efficient Designs to Match Available Parallelism in Quantum Computing, ISCA 2006

References contd.

15. BENOÎT VALIRON, NEIL J. ROSS, PETER SELINGER, D. SCOTT ALEXANDER, AND JONATHAN M. SMITH, Programming the Quantum Future, Communications of the ACM, August 2015, Vol. 58 No. 8, Pages 52-61
16. Daniel Kudrow, Kenneth Bier, Zhaoxia Deng, Diana Franklin, Yu Tomita, Kenneth R. Brown, and Frederic T. Chong, Quantum Rotations: A Case Study in Static and Dynamic Machine-Code Generation for Quantum Computers, in ACM SIGARCH Computer Architecture News 41(3):166 · July 2013
17. Google thinks it's close to "quantum supremacy." Here's what that really means. It's not the number of qubits; it's what you do with them that counts. by Martin Giles and Will Knight, March 9, 2018
18. Yaoyun Shi, Both Toffoli and Controlled-NOT need little help to do universal quantum computation, https://arxiv.org/abs/quant-ph/0205115v2
19. Guest Lecture by Tom Wong, Quantum Computing, Universal Gate Sets, https://www.scottaaronson.com/qclec/16.pdf
20. Quantum Turing machine, https://en.wikipedia.org/wiki/Quantum_Turing_machine
21. Sevag Gharibian, Introduction to Quantum Computation, Deutsch's Algorithm, VCU, 2015
22. C191 - Lectures 8 and 9 - Measurement in Quantum Mechanics, https://inst.eecs.berkeley.edu/~cs191/fa14/lectures/lecture89.pdf
23. Josef Gruzka, Quantum Computing, Advanced topics in computer science series, 1999.
24. Scott Aaronson, Quantum Computing, Lectures PHYS771, https://www.scottaaronson.com/democritus/lec10.html
25. Quantum Computing Enters 2018 Like It Is 1968, https://www.nextplatform.com/2018/01/10/quantum-computing-enters-2018-like-1968/
26. Quirk circuit simulator, http://algassert.com/quirk
27. X. Fu, M. A. Rol, C. C. Bultink, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudever, L. DiCarlo, K. Bertels, An Experimental Microarchitecture for a Superconducting Quantum Processor, ACM Micro 2017.
28. IBM Q Experience Documentation, Shor's algorithm.
29. Austin G. Fowler et al, Surface codes: Towards practical large-scale quantum computation, https://arxiv.org/pdf/1208.0928.pdf
30. E Dennis, et al, Topological quantum memory, https://arxiv.org/pdf/quant-ph/0110143.pdf
31. Kudrow et al., Quantum Rotations: A Case Study in Static and Dynamic Machine-Code Generation for Quantum Computers, ISCA, 2013.
32. Hornibrook, J. M. et al.Cryogenic Control Architecture for Large-Scale Quantum Computing - Phys.Rev.Applied 3 (2015) no.2, 024010, Phys.Rev.Applied. 3 (2015) 024010 arXiv:1409.2202 [cond-mat.mes-hall]

Acknowledgements and Contact Info

- Extensively used references to create these slides, including figures and derivations.

- Contact at

  - Csaba Andras Moritz, andras@bluerisc.com
  - Or through EPRIVO service