Quantum Computation Primitives (Area Exam Talk)

Michael Frank

MIT AI Laboratory

mpf@ai.mit.edu

Quantum Computation Primitives

Overview

**** Talk will be structured around the 3 papers.*

- Introduction **** Review some background and basic concepts needed to understand the other papers.*
- Shor 1995, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." **** Result that provides a major motivation for QC research, including work on primitives.*
- Barenco *et al.* 1995, "Elementary gates for quantum computation." **** State-of-the-art quantum primitives. (Latest result concerning universal quantum logic gates.)*
- Vedral *et al.* 1995, "Quantum networks for elementary arithmetic operations." **** How primitives are used. (Efficient constructions for reversible arithmetic.)*
- Conclusion. **** Talk about open problems.*

What's Quantum Computation All About?

- A new model of computation. **** New formal model, very different from all earlier, classical models.*
- Based on "superpositions of states." **** Key property is that it takes advantage of the phenomenon in QM where a system can be in a so-called "superposition" of many states at once. WTAML*
- Has advantages over classical computation. **** In ways we will discuss later... But there are also some disadvantages*

Introduction

Fundamental Concepts

- A quantum state or *superposition* is:
  - A function from classical states to complex numbers.
  - A vector of complex numbers indexed by classical states.
- The complex numbers are called *amplitudes.*
- Quantum states change by *unitary transformations:*
  - Operators from functions to functions.
  - Functions mapping vectors to vectors.
  - Linear, represented by matrices.
  - One-to-one, Length preserving.

Fundamental Concepts cont.

- Measurements. **** Interaction that communicates state info to unmodeled external system... Can't measure amplitudes directly, only classical states; prob of a state is given by the square of abs.val. of cpx # assocd w. state... Model changes to be consistent*
  - Interaction with environment**** uncontrolled,unmodeled external*
  - System "chooses" a particular classical state
  - Probability of state is square of absolute value of amplitude
  - Quantum state "collapses" to be consistent w. measured value.
- $|s\rangle$ means a vector where state s has amplitude 1, all other states amplitude 0.

Unitary Operations—Example

"Square root of NOT" gate:

$$
N^{1/2} = \begin{array}{c|cc|} & |0\rangle & |1\rangle \\ |0\rangle & \frac{1}{\sqrt{2}}(1+i) & \frac{1}{\sqrt{2}}(1-i) \\ |1\rangle & \frac{1}{\sqrt{2}}(1-i) & \frac{1}{\sqrt{2}}(1+i) \end{array}
$$

$$
(N^{1/2})^2 = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}
$$

$$
N^{1/2} \times \begin{vmatrix} 1 \\ 0 \end{vmatrix} = \begin{vmatrix} \frac{1}{\sqrt{2}}(1+i) \\ \frac{1}{\sqrt{2}}(1-i) \end{vmatrix}
$$

**** We'll see more examples later when we talk about universal quantum gates.*

Introduction

Historical Development

**** Had its roots perhaps in...*

- Reversible computation (1970's)
  - Bennett, Fredkin, Toffoli

  **** Disc. that RC's as powerful as trad. comps. + could avoid power dissip... But of course people knew that quantum mechanics was reversible, so...*

- Early quantum computation ideas (1980's):
  - Benioff 80, 82 **** Isolated sys evolving by QM can sim a TM*
  - Feynman 82, 86 **** classical can't do QM fast, (ind.) QM can sim rev. dig circuit w. zero dissip.*
  - Deutsch 85 **** Defined formal QTM model, showed an advantage on XOR'd results*

- Recent developments (89-present): **** WTAML, contributors too numerous to list here*
  - Quantum circuits & logic gates
  - Quantum complexity theory **** quantum complexity classes*
  - Physical implementations
  - Error correction codes

**** Field developing very rapidly now.*

Shor '95

Quantum Complexity Theory

- Early developments
  - Deutsch: XOR of 2 function values **** QC could compute in same time & space as 1 value, but only half the time*
  - Deutsch & Jozsa: Exponential speed-up on an Oracle problem **** Not very useful*
- Quantum Complexity Classes **** Some examples*
  - **EQP**— like **P**

    **** Worst-case polytime, no error*
  - **ZQP**— like **ZPP**

    **** "Las Vegas," no error, expected ptime*
  - **BQP**— like **BPP**

    **** pr$>$ .5 correct, expected ptime*
  - $\exists Y : \mathbf{EQP}^Y \supset \mathbf{NP}^Y \cup \mathbf{co}-\mathbf{NP}^Y$

    **** example result from Berthiaume+Brassard 94*
- All results relativized

Shor '95

Outline of Shor's Algorithm

Uses this standard reduction of factoring N to finding the order of x mod N:

1. Pick a random $x < N$.
2. If $\gcd(x, N) \neq 1$ return it.
3. Compute the order of x, mod N.

$$r \leftarrow \min a : x^a \bmod N = 1$$

4. If $\gcd(x^{r/2} - 1, N) \neq 1$, return it.
5. If $\gcd(x^{r/2} + 1, N) \neq 1$, return it.
6. Repeat as needed.

Step 3 uses a quantum computer.

Shor '95

Heart of Shor's Algorithm

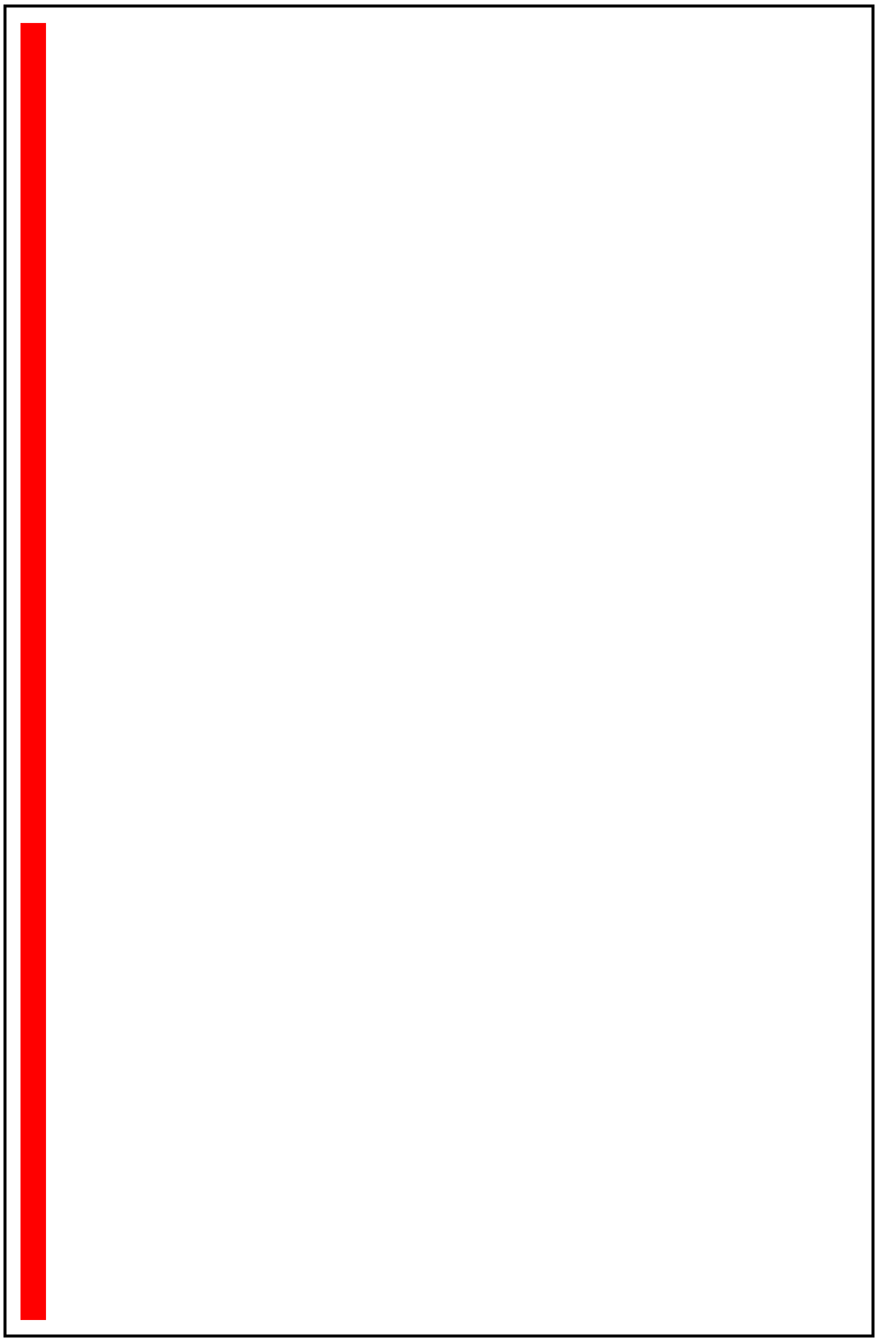
Uses two quantum registers (a, b): $0 \leq a < q, 0 \leq b < N$, where $N^2 \leq q = 2^l < 2N^2$.

1. Initial quantum state is $|0, 0\rangle$, *i.e.* $(a = 0, b = 0)$.
2. Go to superposition
$$\sum_{a=0}^{q-1} |a, 0\rangle / \sqrt{q}.$$
3. Transform $|a, 0\rangle \to |a, x^a \bmod N\rangle$.
4. Perform quantum Fourier transform over first register:
$$\psi'(c, b) = \sum_{a=0}^{q-1} \exp(2\pi iac/q)\psi(a, b)/\sqrt{q}$$
5. Measure c, compute c/q, round to nearest fraction d/r with $r < N$.
6. The order of $x \bmod N$ is usually this r or a small multiple of it.

Example of Shor Algorithm

$n = 33$, $q = 256$

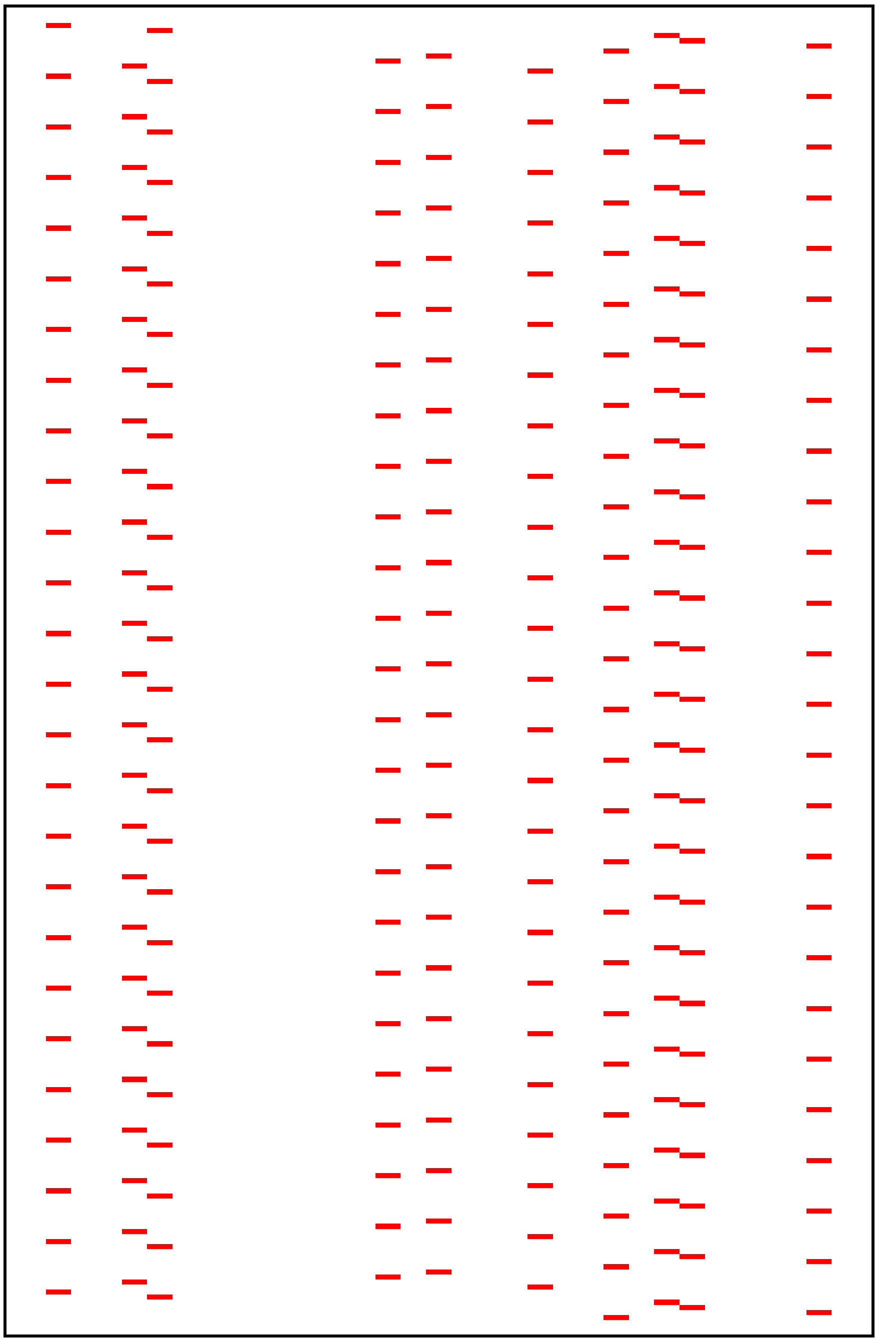
Initial Superposition



Example of Shor Algorithm

$n = 33$, $q = 256$, $x = 5$

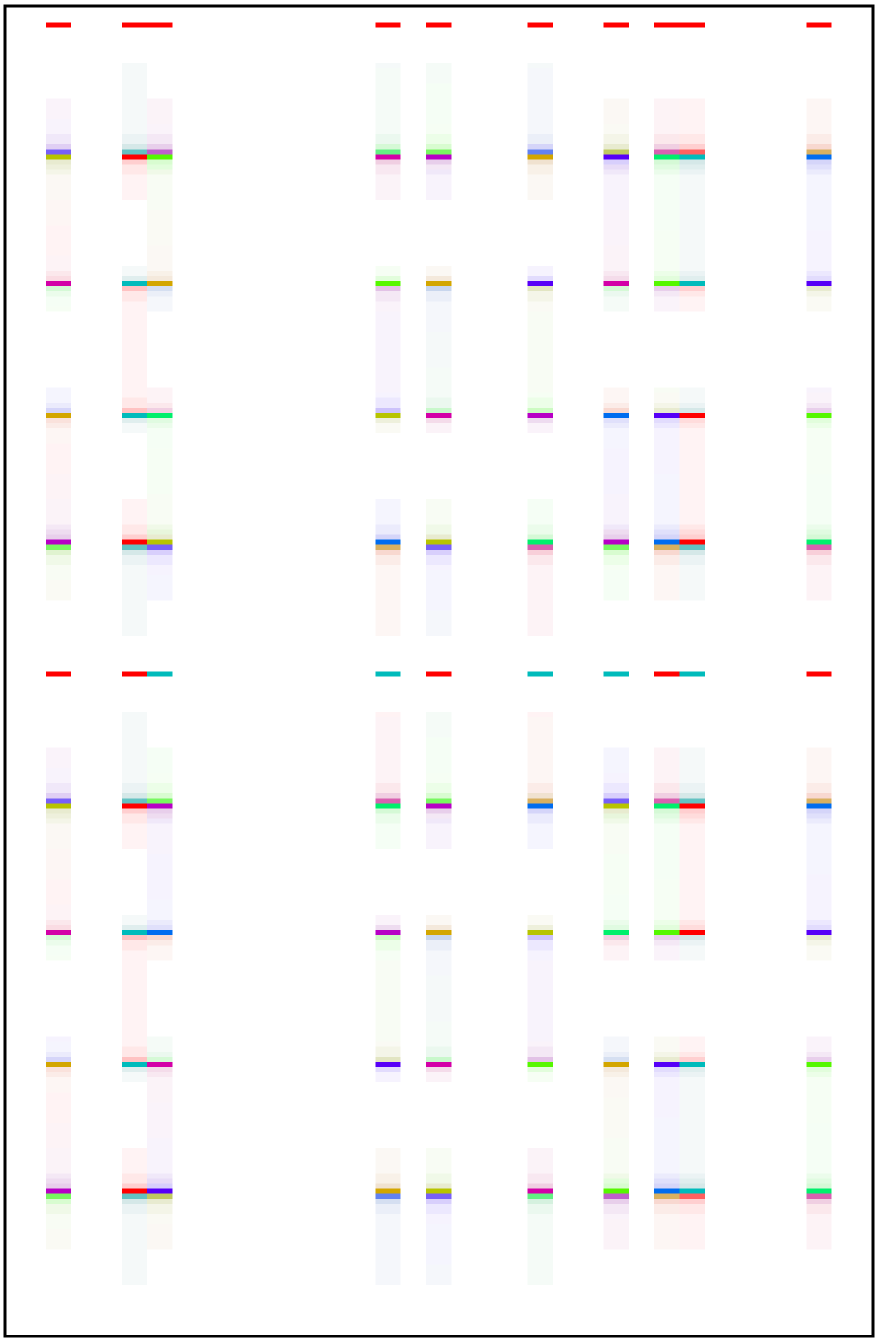
After computing $x^a \bmod n$



Example of Shor Algorithm

$n = 33$, $q = 256$, $x = 5$

After Fourier transform



Shor ′95

Other Important Points Regarding Shor′s Method

- High precision transforms not required. (Coppersmith ′94)
- Also useful for finding discrete logarithms.

Barenco *et al.* ’95

Quantum Logic Gates

- Background
  - Invented by Deutsch (1989).
  - Analogous to classical logic gate arrays (acyclic circuits).
  - Similar to reversible gate arrays (Fredkin, Toffoli).
- Classical state of system is divided into bits.
- Quantum logic gates: local unitary transforms that operate on only a few state bits at a time.
- Computation via predetermined sequence of gate applications to selected bits.
- Existence of universal gates.

Barenco *et al.* ′95

Brief History of Universal Quantum Gates and Adequate Gate Sets

- Deutsch ′89
  - universal 3-bit Toffoli-like gate
- DiVincenzo ′95
  - adequate set of 2-bit gates
- Barenco ′95
  - universal 2-bit gate
- Deutsch *et al.* ′95
  - almost all 2-bit gates are universal
- Barenco *et al.* ′95
  - 2-bit XOR + set of 1-bit gates is adequate

Universal Quantum Gates

Deutch's Generalized 3-bit Toffoli Gate:

$$\begin{vmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & a & b \\ & & & & & & b & a \end{vmatrix}$$

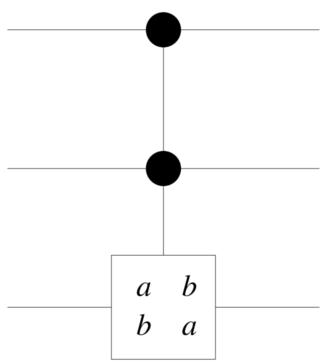
where

$$a = ie^{i\pi\alpha/2}(1 + e^{i\pi\alpha})/2$$

and

$$b = ie^{i\pi\alpha/2}(1 - e^{i\pi\alpha})/2$$

and α is irrational.



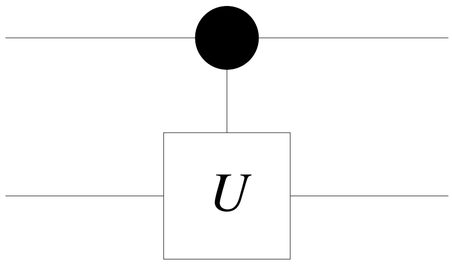
Universal Quantum Gates

Barenco's Generalized 2-bit Toffoli-like gate:

$$A(\phi, \alpha, \theta) = \left| \begin{matrix} 1 & & & \\ & 1 & & \\ & & e^{i\alpha} \cos\theta & -ie^{i(\alpha-\phi)} \sin\theta \\ & & -ie^{i(\alpha+\phi)} \sin\theta & e^{i\alpha} \cos\theta \end{matrix} \right|$$

where ψ, α, θ, and π are relatively irrational. Also works *e.g.* for $\psi = \pi, \alpha = \pi/2$:

$$A(\pi, \pi/2, \theta) = \left| \begin{matrix} 1 & & & \\ & 1 & & \\ & & i\cos\theta & -\sin\theta \\ & & -\sin\theta & i\cos\theta \end{matrix} \right|$$



Barenco *et al.* '95

Main Results

- Universality of XOR + 1-bit quantum gates
  - 2-bit Barenco gate already known universal
  - 4 one-bit gates + 2 XORs suffice to build it
- Construction of generalized Toffoli gates
  - 3-bit version via five 2-bit gates
  - n-bit version via $O(n^2)$ 2-bit gates
  - No auxilliary bits needed for above
  - n-bit version via $O(n)$ 2-bit gates given 1 work bit
- Arbitrary n-bit unitary transforms
  - Appear to require exponential # of primitives*** *worst case*
  - Not yet proven*** *suggested by dimensional analysis*
  - Are exponential ones interesting?*** *Sims of quantum systems*

Vedral *et al.* '95

- Goal: explicitly construct modular exponentiation out of primitives.*** *for Shor's alg*
- Approach: Use classical reversible network of Toffoli gates.
  - addition (in-place)
  - addition modulo N
  - multiplication mod N
  - exponentiation mod N
- Results:*** *nice, but unsurprising. O(n) add, O(nn) mult, nnn exp... Shor hinted could do better multiplies.*
  - $O(n^3)$ running time
  - $O(n)$ temporary storage
- Errors:
  - Carry op defined wrong
  - Misstatement of Barenco *et al.* result

**** Together with other papers this gives a pretty complete recipe for a how to build up factoring from simple 1- and 2-bit primitive quantum logic gates, if we just knew how to implement them & correct for errors we'd be all set.*

Conclusion
Important Open Problems

- $\mathbf{BQP} \supseteq \mathbf{NP}$?
  - Seems unlikely.

**** algs use periodicity*

  - Bennett *et al.*: $\mathbf{NP}^R \supset \mathbf{BQP}^R$.
  - Proof of $\mathbf{BQP} \subset \mathbf{NP}$ unlikely.
- $\mathbf{BQP} \supset \mathbf{BPP}$?
  - Probably, but unproven.**** Shor's alg evidence*
  - If untrue, helpful to physics.
- Can quantum computers simulate physics in polynomial time/space?**** Open as far as I know.*
- Can error correction problem be solved?
  - Errors of imprecision.
  - Interactions w. environment.

**** exp.decoher*

- Physical Implementation
  - Many proposals
  - Rapid progress
  - "Just an engineering problem"

Conclusion

- An intriguing new model for computation
  - In accordance with physics
  - Potentially more powerful
- Field is developing rapidly
  - Quantum complexity theory
  - Universal primitives
  - Error correction
  - Physical implementation
- Potential big impact
  - Fast atomic-scale computers

    **** Having a model that's close to the physics may help in the design of atom-scale computers even if they don't take advantage of superpositions all that much*
  - Applications to cryptography
  - Simulating physics
- *It will happen if it's useful.*

Introduction

Fundamental Concepts

(Stated more mathematically)

- Quantum vs. classical state spaces:*** *space that system evolves in*
  - S a classical state-space.*** *for a system*
  - Hilbert space $\mathcal{H} = \mathbb{C}^S$.*** *Set of functions from S to C, or vector of C's indexed by S. Note this is much much larger*
  - $\mathcal{H}$ continuous, $\dim(\mathcal{H}) = |S|$.
- Amplitudes & probabilities: *** *resulting from 'measurement'*
  - Let $s \in S$ and $\psi \in \mathcal{H}$.
  - $\Pr(s|\psi) = |\psi(s)|^2/c$. *** *c a norm.const*
- Unitary transformations: *** *all quantum evolution*
  - ψ considered as a vector, $\mathcal{H}$ as a vector space
  - U a transformation matrix $U\psi \in \mathcal{H}$*** *ie a linear func from H to itself*
  - Unitary $U = ((U^{-1})^T)^*$.*** *explain notation. 1-to-1 and length-preserving*

Correcting Decoherence

Key is to design encoding where a single bit measurement yields *no information* about encoded bits.

- Calderbank & Shor ’95
- Chuang & Laflamme ’95
- Steane ’96