

New Quantum Algorithms

- Wandering Shor Algorithms
- Continuous Variable Shor Algorithms
- Lifted (Purified) Shor
- Quantum Circle Algorithms

I double dare Yanhua to implement this in optics

- Dual Shor Algorithm
- Functional integral Quantum Algorithms (highly speculative)

A Continuous Variarble Shor Algorithm

This algorithm finds the hidden period of an admissible map

$$f: \mathbb{R} \to \mathbb{R}$$

Lomonaco, Jr., Samuel J., and Louis H. Kauffman, A Continuous Variable Shor Algorithm, arXiv:quant-ph/0210141



A Distributed Quantum Algorithms



Anocha Yimsiriwattana

Yimsiriwattana, Anocha, and Samuel J. Lomonaco, Jr., Generalized GHZ States and Distributed Quantum Computing, arXiv:quant-ph/0402148

Yimsiriwattana, Anocha, and Samuel J. Lomonaco, Jr., Distributed quantum computing: A distributed Shor, algorithm, arXiv:quant-ph/0403146

Distributed Quantum Computing



- Distributed computing is one path to scalable quantum computing
- Provides a mechanism for isolating the problem of decoherence

Overview

The computational cost of transforming a quantum algorithm into a distributed quantum algorithm

- Space Complexity Overhead: The additional space overhead is insignificant, i.e., the number of required additional qubits is 5 and independent of the number of qubits of the non-distributed algorithm
- Time Complexity Overhead: There is a resulting a linear slowdown of the non-distributed algorithm
- But ... the rate of slowdown is bounded above by a constant (= 9) which is independent of the number of qubits in the non-distributed.

Conclusion

It really pays to distribute a quantum algorithm.

- The problem of decoherence is isolated to the individual small computing nodes in the network, where decoherence is most easily handled.
- * As a result, the need for costly quantum error correction is reduced by at least one order of magnitude.

We Will Show How to Use Quantum

Entanglement for Distributed

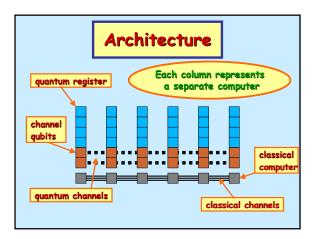
Control of Quantum

Algorithms

Distributed
Quantum
Computing
Architecture

Distributed Quantum Computing

- By a distributed quantum computer, we mean a network of quantum computers interconnected by quantum and classical channels
- The distributed computing paradigm provides an effective way to utilize a number of small quantum computers



Distributed
Quantum
Computing
Primitives

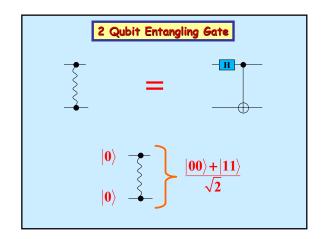
The Application of
Generalized GHZ States
and
Cat-Like States
to
Distributed Quantum Computing

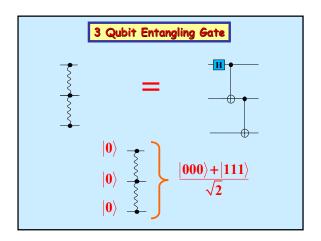
Terminology

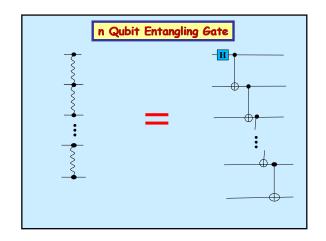
A Generalized GHZ state is a quantum state of the form $\begin{array}{c|cccc}
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline
 & n & 0 \\
\hline
 & n & 1 \\
\hline$

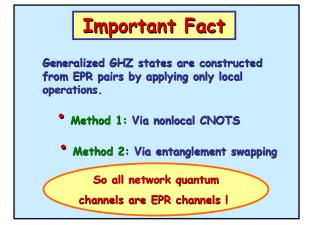
Key Idea
Use Entanglement to Distribute Control

A generalized GHZ state can be used to create a "cat-like" state, which can in turn be used to distribute control.



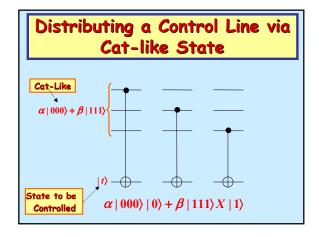


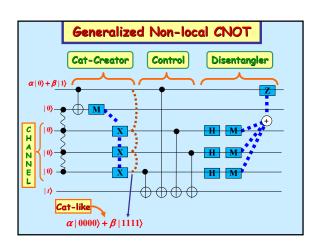


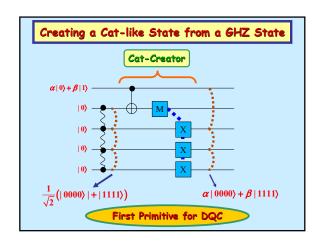


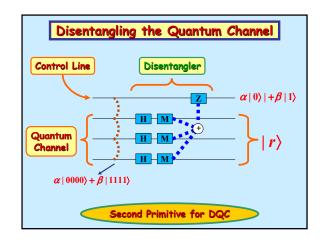
Observations

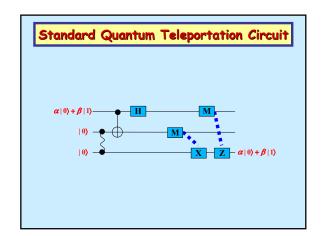
- * The CNOT gate and the set of all one-qubit gates are universal
- If we can implement a non-local CNOT, then a distributed version of any unitary transformation can be implemented

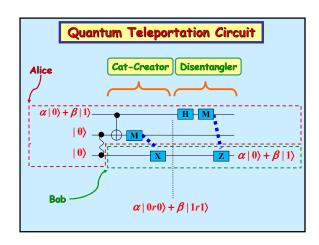


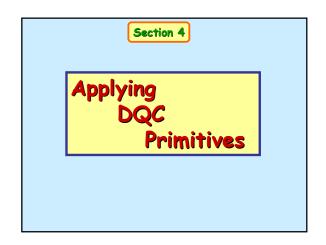


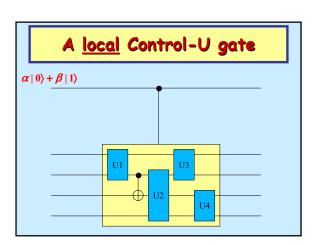


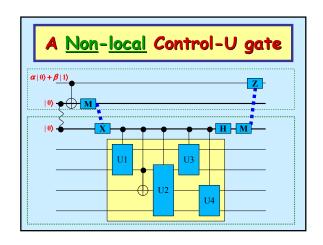


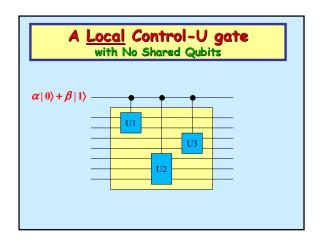


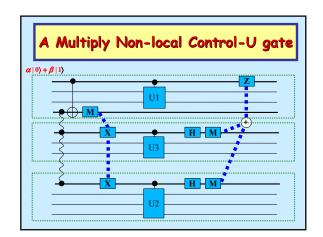


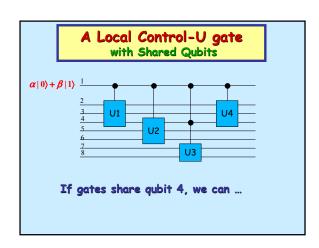


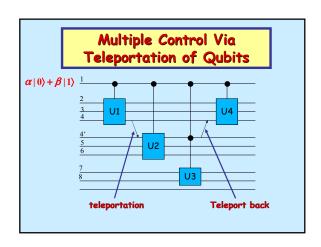


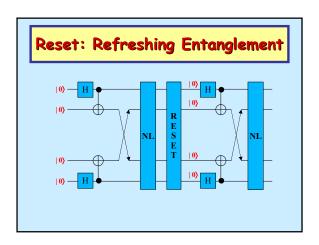


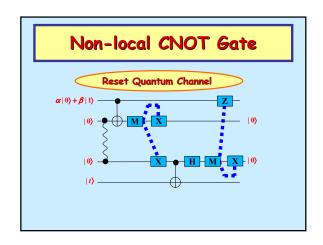


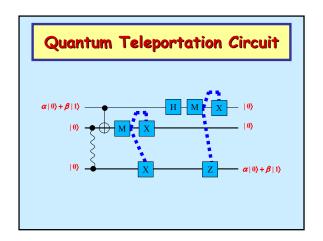


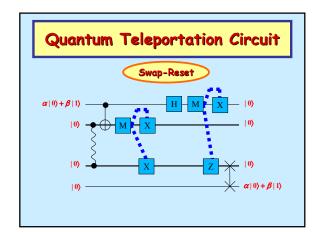


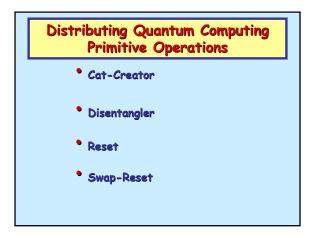


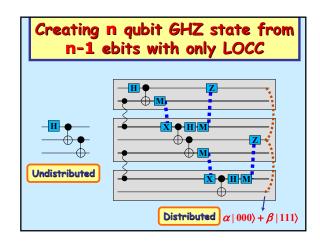


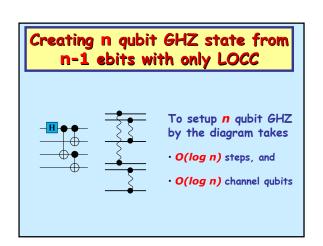




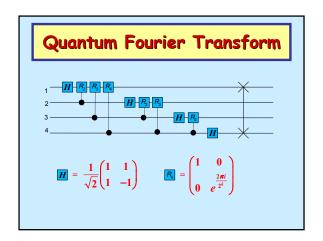


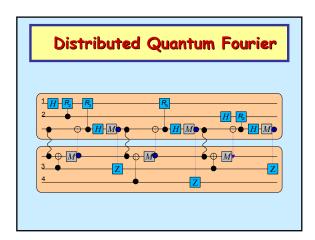


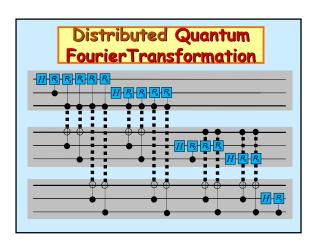




A Distributed
Quantum Fourier
Transform





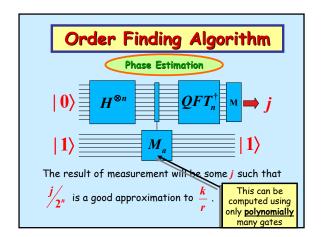


Distributed Shor
Factoring Algorithm

Order Finding Algorithm

- Given N and $a \in \mathbb{Z}_N$ find the order of a. Let r be the order of a.
- Define M_a as follow: let $x \in \mathbb{Z}_N$

$$M_a:|x\rangle \rightarrow |ax \bmod N\rangle$$



Order Finding & Shor's Factoring Alg.

- Repeat to find different j
- ullet Use <u>continued fraction</u> algorithm to find the period $oldsymbol{r}$

Quantum Factoring



Quantum Factoring Algorithm

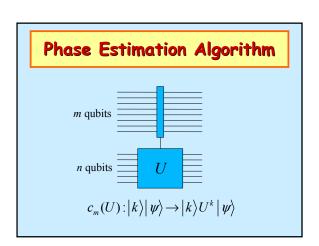
Quantum Factoring

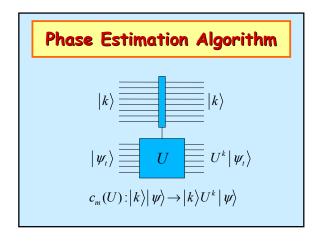
Order Finding

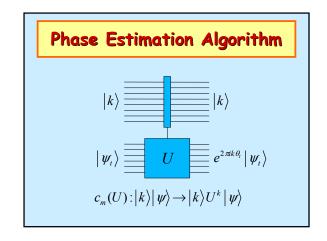
Phase Estimation

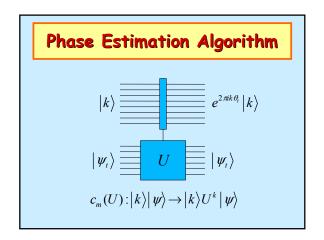
Phase Estimation Algorithm Problem Definition

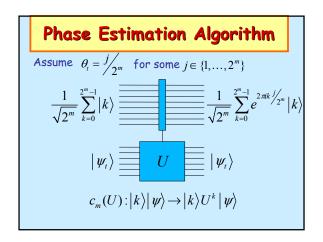
- Input: (1) an *n*-qubit unitary transformation ${\cal U}$ with eigenvectors $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_N\rangle$ and eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ where $N=2^n$ and for each ${\bf k}$ $\lambda_k=e^{2\pi i \theta_k}$
 - (2) an eigenvector $|\psi_t\rangle$: $t \in \{1,...,N\}$
- Output : an estimation value of θ_i

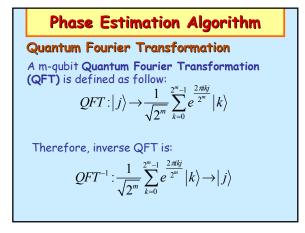


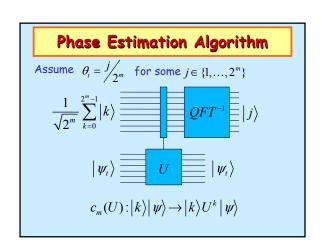


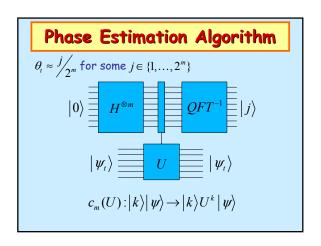


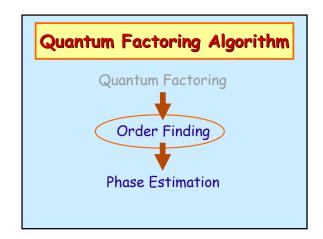












Order Finding Algorithm

Problem Definition

• Input : Integers N and \mathbb{Z}_N

• Output: The order of a, r, such that $a^r \equiv 1 \mod N$

Define a unitary transformation

 $M_a: |x\rangle \rightarrow |ax\rangle$

Order Finding Algorithm

Consider $|\psi_t\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \omega^{-st} |a^s\rangle$

$$|\psi_0\rangle = \frac{1}{\sqrt{r}}(|1\rangle + |a\rangle + \dots + |a^{r-1}\rangle)$$

$$\begin{split} |\psi_{0}\rangle &= \frac{1}{\sqrt{r}} \Big(|1\rangle + |a\rangle + \dots + |a^{r-1}\rangle \Big) \\ |\psi_{1}\rangle &= \frac{1}{\sqrt{r}} \Big(|1\rangle + \omega^{-1}|a\rangle + \dots + \omega^{-(r-1)}|a^{r-1}\rangle \Big) \\ &\vdots \\ &\vdots \\ \text{where } \omega = e^{\frac{2\pi i}{r}} \end{split}$$

Then $M_a | \psi_t \rangle = \omega | \psi_t \rangle$

Order Finding Algorithm

$$t_{r}^{\prime} \approx \frac{j}{2^{m}} \text{ for some } j \in \{1, \dots, 2^{m}\}$$

$$|0\rangle \qquad H^{\otimes m} \qquad QFT^{-1} \qquad |j\rangle$$

$$|\psi_{t}\rangle \qquad M_{a} \qquad |\psi_{t}\rangle$$

$$c_{m}(M_{a}): |k\rangle |\psi_{t}\rangle \rightarrow |k\rangle M_{a}^{k} |\psi_{t}\rangle$$

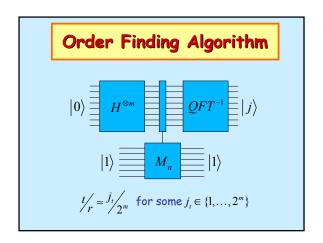
Order Finding Algorithm

Two obstacles:

- · How to construct $|\psi_{\iota}\rangle$?
- How to implement $c_m(M_a)$?

Order Finding Algorithm

Observer that $|1\rangle = \frac{1}{\sqrt{r}} \sum_{t=0}^{r-1} |\psi_t\rangle$ $|0\rangle|1\rangle \stackrel{_{H\otimes I}}{\Rightarrow} \frac{1}{\sqrt{r}} \sum_{\iota=0}^{r-1} \frac{1}{\sqrt{2^m}} \sum_{k}^{2^m-1} |k\rangle |\psi_{\iota}\rangle$ $\overset{c_m(M_a)}{\Rightarrow} \frac{1}{\sqrt{r}} \sum_{t=0}^{r-1} \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} |k\rangle \omega^{kt} |\psi_t\rangle$ $\stackrel{QFT^{-1}}{\Rightarrow} \frac{1}{\sqrt{r}} \sum_{t=0}^{r-1} |j_t\rangle |\psi_t\rangle$



Order Finding Algorithm

Two obstacles:

- · How to construct $|\psi_t\rangle$?
- How to implement $c_m(M_a)$?

Modular Exponentiation

Erasing Garbage

Let f be functions. We define unitary transformation F as follow:

 $F:|x\rangle|0\rangle|0\rangle\rightarrow|x\rangle|f(x)\rangle|g(x)\rangle$ where g is "garbage" function.

The garbage can be erased as follow:
$$|x\rangle|0\rangle|0\rangle|0\rangle \mathop{\Rightarrow}_{COPY}^{F\otimes I}|x\rangle|f(x)\rangle|g(x)\rangle|0\rangle \\ \mathop{\Rightarrow}_{COPY}|x\rangle|f(x)\rangle|g(x)\rangle|f(x)\rangle \\ \mathop{\Rightarrow}_{F^{R}\otimes I}|x\rangle|0\rangle|0\rangle|f(x)\rangle$$

$$XF = F^{-1} \cdot COPY \cdot F$$

The Overwriting Invertible Function

Let f be invertible functions.

$$F: |x\rangle |0\rangle \to |x\rangle |f(x)\rangle$$

$$FI: |x\rangle |0\rangle \to |x\rangle |f^{-1}(x)\rangle$$

Overwriting Invertible operation is

$$|x\rangle|0\rangle \underset{SWAP}{\Rightarrow} |x\rangle|f(x)\rangle$$

$$\Rightarrow |f(x)\rangle|x\rangle$$

$$\Rightarrow |f(x)\rangle|0\rangle$$

$$OF = FI^R \cdot SWAP \cdot F$$

Modular Exponentiation Let $k_0 k_1 \cdots k_{m-1}$ be the binary representation of k, $M_a^k = \prod_{i=1}^{m-1} M_a^{k_s 2^s}$ $|k\rangle$ $M_{a}^{2^{2}}$ M_a^2

The Multiplier

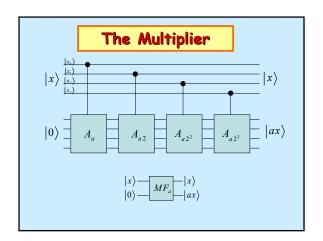
$$|x\rangle - M_a - |ax\rangle$$
 $M_a: |x\rangle \rightarrow |ax\rangle$

$$M_a: |x\rangle \rightarrow |ax\rangle$$

Observe that $M_a^k |x\rangle = M_a \cdots M_a |x\rangle = |a^k x\rangle$ $M_a^k = M_{ak}$

Let $x_0x_1\cdots x_{n-1}$ be the binary representation of x,

$$ax = \sum_{s=0}^{n-1} ax_s 2^s$$



Overwrite Multiplier

$$MF_a: |x\rangle |0\rangle \rightarrow |x\rangle |ax\rangle$$

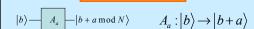
$$MFI_a: |x\rangle |0\rangle \rightarrow |x\rangle |a^{-1}x\rangle$$

 $MFI_a = MF_{a^{-1}}$

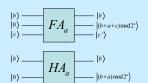
Define $M_a = MFI_a^R \cdot SWAP \cdot MF_a$

$$|x\rangle$$
 MF_a $|x\rangle$ MFI_a^R $|ax\rangle$

The Adder



Assume we have Full Adder (FA) and Half Adder (HA):



The Adder

Case 1:
$$b + a < N$$

Case 1:
$$b+a < N$$

 $(b+a) \mod N = (b+a) \mod 2^n$

Case 2:
$$b + a \ge N$$

Case
$$2 \cdot b + a \ge N$$

 $(b+a) \mod N = (b+a+2^n-N) \mod 2^n$

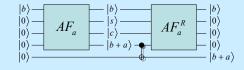
$$\begin{array}{c|c} |b\rangle & & & & |b\rangle \\ |0\rangle & & & & |a\rangle \\ |0\rangle & & & & |a\rangle \\ |0\rangle & & & & |c\rangle \\ |0\rangle & & & & |c\rangle \\ \hline & & & & |a\rangle \\ AF_a: |b\rangle |0\rangle |0\rangle |0\rangle \rightarrow |b\rangle |s\rangle |c\rangle |b+a\rangle \\ \end{array}$$

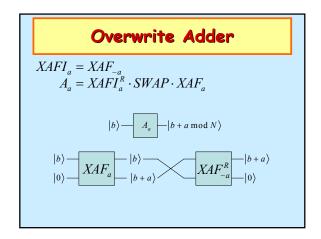
The Adder

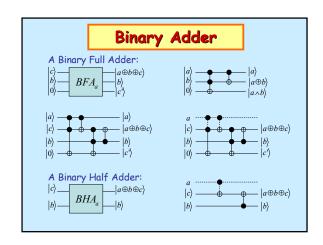
$$AF_a: |b\rangle|0\rangle|0\rangle|0\rangle \rightarrow |b\rangle|s\rangle|c\rangle|b+a\rangle$$

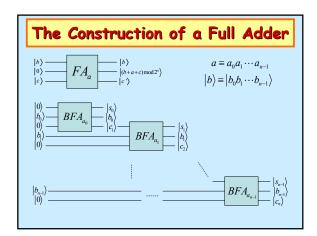
$$XAF_a = AF_a^R \cdot COPY \cdot AF_a$$

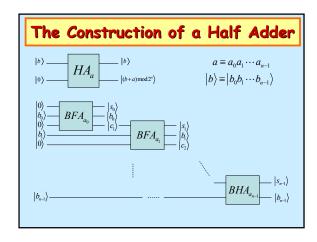
$$XAF_a: |b\rangle|0\rangle|0\rangle|0\rangle \rightarrow |b\rangle|0\rangle|0\rangle|b+a\rangle$$

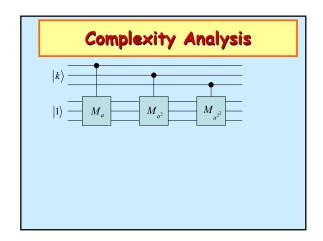


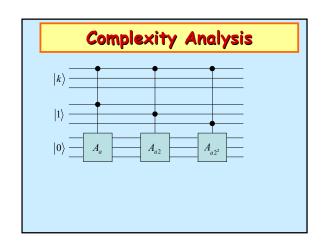


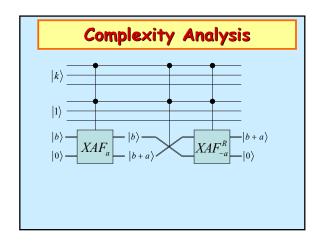


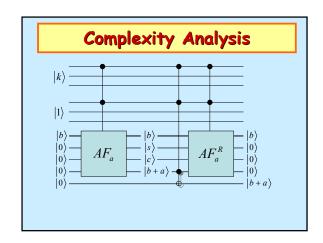


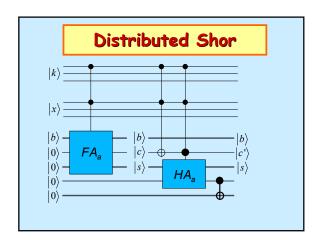


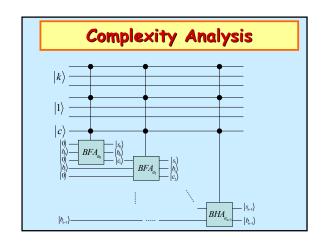


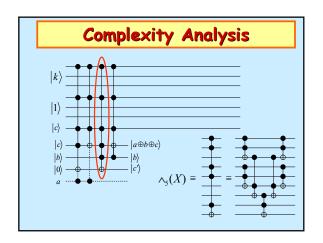












```
Complexity Analysis

The number of gates

Let G(F) be the number of gates in circuit F.

G(SHOR) = G(H^{\otimes n}) + G(c_m(M_a)) + G(QFT^R) = O(mn^2)

G(c_m(M_a)) = m(G(M_a) + 1) = 70mn^2 + 3mn + m

G(M_a) = G(MF_a) + G(SWAP) + G(MF_a^R) = 70n^2 + 3n

G(MF_a) = n(G(A_a) + 1) = 35n^2 + n

G(A_a) = G(XAF_a) + G(SWAP) + G(XAF_a^R) = 35n

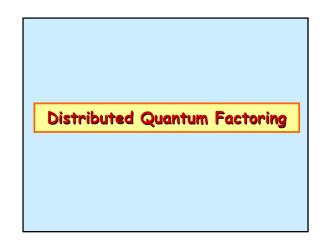
G(XAF_a) = G(AF_a) + G(COPY) + G(AF_a^R) = 17n

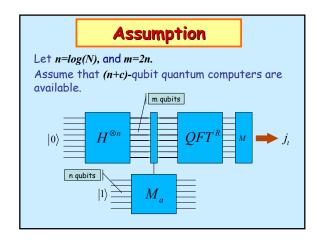
G(AF_a) = G(FA_a) + G(HA_a) + 2 = 8n

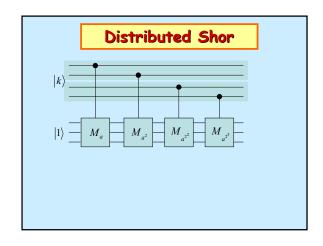
G(FA_a) = nG(FFA_a) = 4n

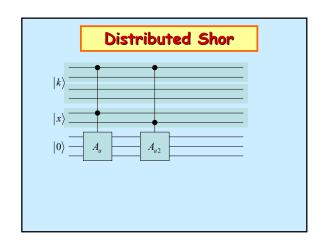
G(HA_a) = (n-1)G(FFA_a) + G(FFA_a) = 4n-2
```

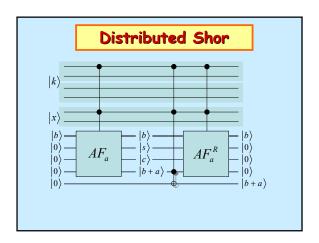
Complexity Analysis The number of qubits Let Q(F) be the number of qubits in circuit F. $Q(SHOR) = \max\{Q(H^{\otimes n}), Q(c_m(M_a)), Q(QFT^R)\} = 5n+m+1$ $Q(c_m(M_a)) = Q(M_a) + m = 5n+m+1$ $Q(M_a) = \max\{Q(MF_a), Q(SWAP), Q(MF_a^R)\} = 5n+1$ $Q(MF_a) = Q(A_a) + n = 5n+1$ $Q(A_a) = \max\{Q(XAF_a), Q(SWAP), Q(XAF_a^R)\} = 4n+1$ $G(XAF_a) = 4n+1$

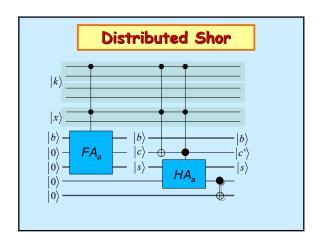


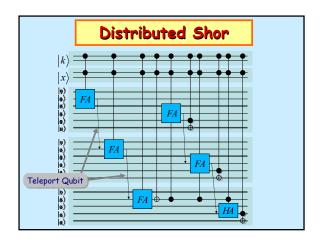












Assumptions

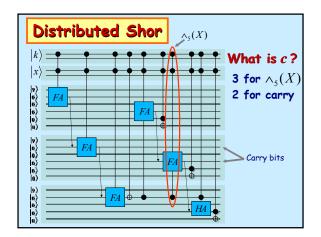
Let n=log(N), and m=2n.

Assume that (n+c)-qubit quantum computers are available.

What is c?

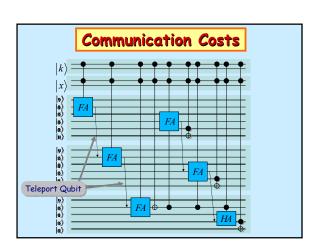
The number c is dependent on

- . How many channel qubits are needed?
- . How many extra carry qubits are needed?
- The assumptions made for the elementary gates?



Communication Cost

- · Communication Cost:
 - Distributed Control
 - Teleportation
- \cdot Over estimate: Every gate is non-local control gate. Then the significant control is a second control of the second control of
- · Communication cost is lower:
 - not all controlled gate are non-local,
 - the control qubit can be shared.



Communication Costs

The Number of Full Adder Circuit

Let A#(F) be the number of full adder in circuit F.

$$\begin{array}{lll} A\#(C_m(M_a)) = mA\#(M_a) & = 16mn \\ A\#(M_a) = A\#(MF_a) + A\#(SWAP) + A\#(MF_a^R) & = 16n \\ A\#(MF_a) = nA\#(A_a) & = 8n \\ A\#(A_a) = A\#(XAF_a) + A\#(SWAP) + A\#(XAF_a^R) & = 8 \\ A\#(XAF_a) = A\#(AF_a) + A\#(COPY) + A\#(AF_a^R) & = 4 \\ A\#(AF_a) = A\#(FA_a) + A\#(HA_a) & = 2 \end{array}$$

Communication Costs

Let D(F) be the number of distributed control in circuit F. If m=2n,

$$D(SHOR) = 4A\#(c_m(M_a)) + D(QFT^R)$$
$$= O(mn) + D(QFT^R)$$
$$= O(n^2)$$

Let T(F) be the number of teleportation in circuit F. If m=2n,

$$T(SHOR) = 3A \# (c_m(M_a)) = O(n^2)$$

