Introduction to Quantum Computing
量子計算入門

Rod Van Meter
rdv@tera.ics.keio.ac.jp
寺岡研究室

September 28-30, 2004
@ Aizu U.

with help from
伊藤公平
阿部英介
and slides from T. Metodiev, T. Fujisawa

Course Outline

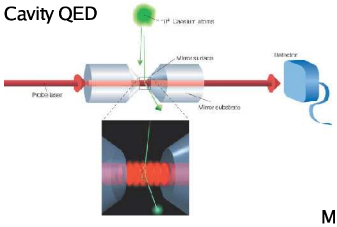
- Lecture 1: Introduction
- Lecture 2: Quantum Algorithms
- Lecture 3: Quantum Computational Complexity Theory
- Lecture 4: Devices and Technologies
- Lecture 5: Quantum Computer Architecture
- Lecture 6: Quantum Networking
- Lecture 7: Wrapup

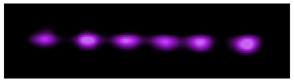
Lecture Outline

- Relationship of Architecture to Technology
- Criteria for Evaluating an Architecture
  - example: layout
- An Advanced Architecture: Scalable Ion Trap
- Architecture in Action: Toward a Quantum Multicomputer
  - Focus on mapping algorithm to architecture

Physical Realization

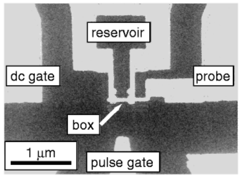
Cavity QED





Ion trap

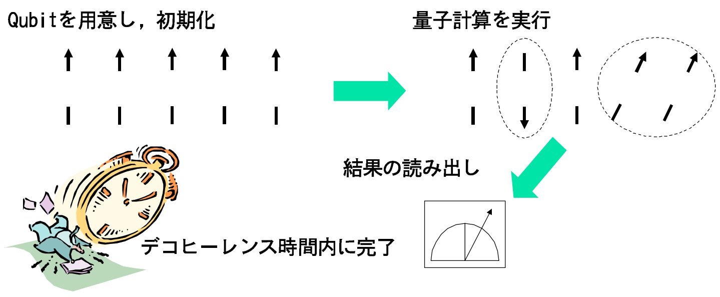
Magnetic resonance



Superconductor

DiVincenzo’s Criteria

1. Well defined extensible qubit array
2. Preparable in the “000…” state
3. Long decoherence time
4. Universal set of gate operations
5. Single quantum measurements



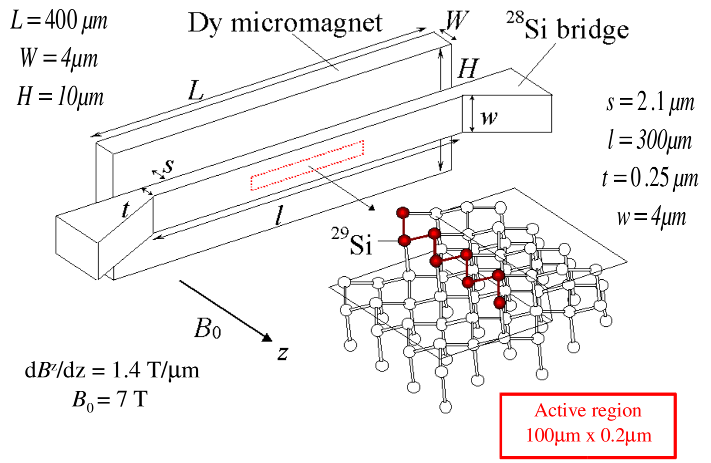
Problems

- Coherence time
  - nanoseconds for quantum dot, superconducting systems
- Gate time
  - NMR-based systems slow (100s of Hz to low kHz)
- Gate quality
  - generally, 60-70% accurate
- Interconnecting qubits
- Scaling number of qubits
  - largest to date 7 qubits, most 1 or 2

Architecture

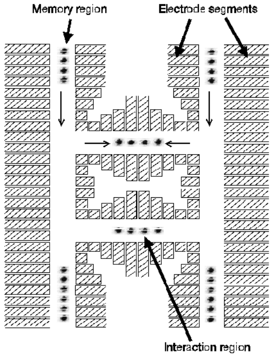
- So far, we have described devices built on specific technologies
- System architecture is the Big Picture
- Technology influences architecture
- Architecture dictates behavior of algorithms
- Common assumption: simple line of qubits

All-Silicon Quantum Computer



Scalable Ion Trap

Qubits are represented in the electron spin of ions; the ions are physically moved around to bring them together to perform two-qubit gates. Gates are laser pulses that cause the qubits to rotate.



Technology is Not Architecture

The behavior of those two systems is completely different! We need a way to describe architecture, the way DiVincenzo describes technology...

Quantum Computer Taxonomy

- flying or sedentary qubits?
- single v. ensemble
- concurrent gate support
- addressing
- natural gates (“instruction set”)
- logical encoding

Quantum Computer Taxonomy (2)

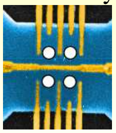
- internal topology
- quantum I/O
- time: clock speed v. decoherence
- timing: jitter and skew control
- programmability
- operating temperature
- measurement time v. gate time

Example: Layout (Internal Interconnect, Measurement)

- Quantum dots as example
- Leads to dots require space
- Double-dot structure limits layout
- Measurement device requires space (fit with every qubit? probably not)

Two-qubit operation for charge qubit

Two-qubit device (preliminary)



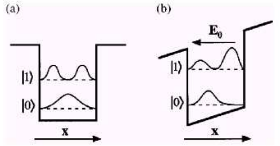


FIG. 1. Charge density in the quantum well in the direction **x** of the applied field. A dipole moment is induced when the electric field is turned on (b), but is zero without the electric field (a).

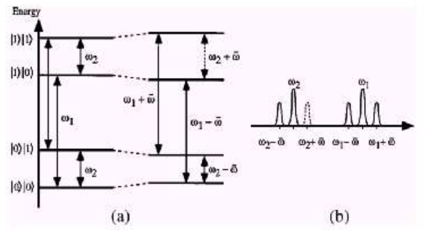
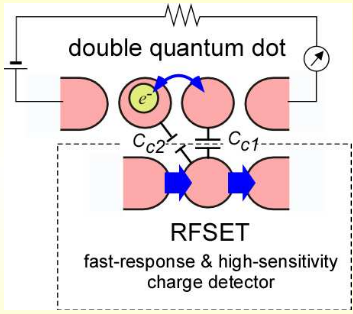


FIG. 2. (a) Energy levels of two quantum dots without and with the coupling induced by the presence of a static electric field $\mathbf{E}_0$. (b) Resonance spectrum of the two quantum dots. The dotted line shows the wavelength for which the two dots act as a controlled-NOT gate, with the first dot being the control qubit and the second the target qubit.

A. Barenco, D. Deutsch, A. Ekert, and R. Jozsa, Phys. Rev. Lett. 74, 4083 (1995).

c.f. Two-qubit CNOT gate in superconducting charge qubit.
T. Yamamoto, Nature 425, 941 (2003).

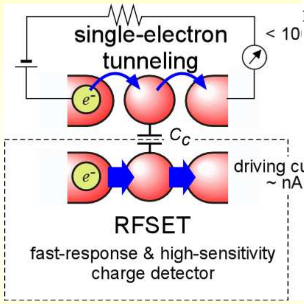
Toward single shot measurement



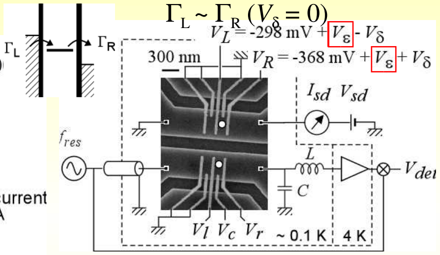
Bandwidth ~ 100 MHz
Sensitivity ~ 3x10^{-6} eHz$^{-1/2}$
$(C_{c1} - C_{c2})/C_\Sigma$ = 0.01 – 0.03
Charge detection time: ~ 10 - 100 ns
(optimistic expectation)
(T_1 = 10 ns – 1 µs – ?)

A. Aassime et al., Phys. Rev. Lett. 86, 3376 (2001).
S. Gardelis et al., Phys. Rev. B 67, 073302 (2003).
J. Elzerman et al., Phys Rev B 67, R161308 (2003).
L. C. L. Hollenberg et al., Phys. Rev. B 69, 113301 (2004).
L. DiCarlo et al., cond-mat/0311308.

Coupled Single-Electron Transistor



$\Gamma_L \sim \Gamma_R$ (V_δ = 0)


Γ_L
Γ_R
V_L = -298 mV + $\boxed{V_\varepsilon}$ - V_δ
V_R = -368 mV + $\boxed{V_\varepsilon}$ + V_δ
300 nm
I_{sd} V_{sd}
f_{res}
L
C
V_{det}
V_l V_c V_r
~ 0.1 K
4 K

L = 100 nH
C ~ 0.6 pF
f_{res} ~ 650 MHz

C_c/C_Σ ~ 0.01

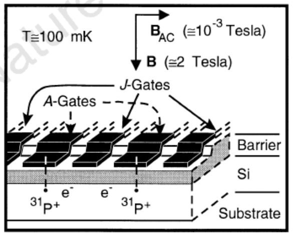
RFSET:
radio-frequency single-electron transistor

charge detection limit (@~10kHz)
~ 5x10^{-5} e/Hz$^{1/2}$ for RFSET
~ 5x10^{-3} e/Hz$^{1/2}$ for the upper QD
(BW ~ 40 kHz)

R. Schoelkopf et al., Science 280, 1238 (1998).
H. D. Cheong et al., Appl. Phys. Lett. 81, 3257 (2002).

Kane Solid-State NMR

Qubits are stored in the spin of the nucleus of phosphorus atoms embedded in a zero-spin silicon substrate. Standard VLSI gates on top control electric field, allowing electrons to read nuclear state and transfer that state to another P atom.

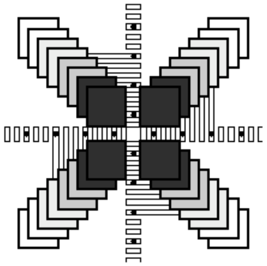


Kane, Nature, 393(133), 1998

Kane/Oskin Lattice

Black dots are location of P atoms. Small rectangles are quantum-scale leads. Large squares are standard-size VLSI leads.

Fitting it all in is tough! This is the role of system architecture...



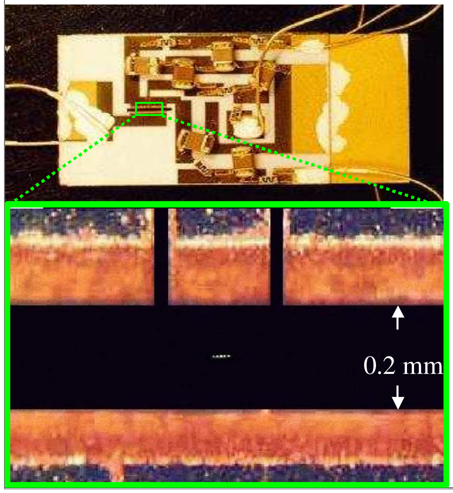
Oskin et al., ISCA, 2003

Advanced Architecture: Scalable Ion Trap

One of the few architectures that separates storage space from action space; that is, memory and CPU.

Main group is Wineland group at NIST (USA).

Trapped-Ion QIP



- Accomplishments:
  - Deutsch-Josza algorithm
    - Blatt group
    - Guide, Nature 421, 48 (2003)
  - 4 qubit entanglement
    - Wineland group
    - Monroe, AIP Conf. Proc. 551 (2001)
  - Ballistic transport
    - Wineland group
    - Rowe, Quantum Information and Computation 2, 257 (2002)

Scalable Ion Trap QC: Architecture?

- Scaling: mictrotraps

(Wineland/NIST)

- **Large-scale QC?**
  - **Teleportation can be used for wiring & code conversion**
  - **Gate errors ~ O(10^{-4}) possible**

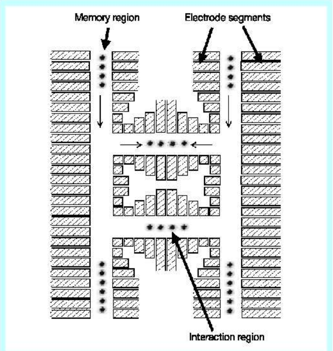
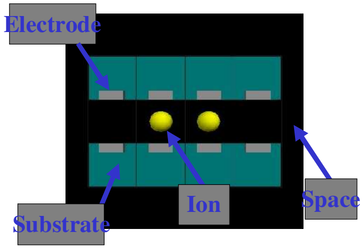


Figure 1 Diagram of the quantum charge-coupled device (QCCD). Ions are stored in the memory region and moved to the interaction region for logic operations. Thin arrows show transport and confinement along the local trap axis.

Kielpinski et al, Nature v417, p 709, 2002

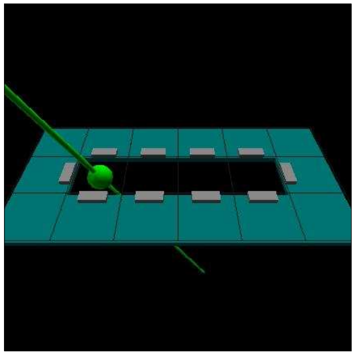
Ion trap essentials:



- RF Paul Trap Segments
  - Substrates with attached electrodes for ion trapping and control
- Ions in linear chains
  - Qubits are hyperfine states
  - Qubits are coupled through collective vibrations
- Lasers implement logic gates and measurement

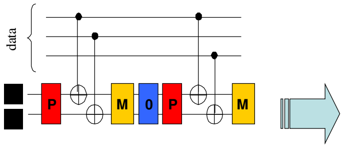
Gates and Measurement

- Quantum gates are laser pulses
- Two-qubit gates couple qubits via ion chain vibration
- Measurement uses a laser pulse and a detector



Ion Trap Array Layout

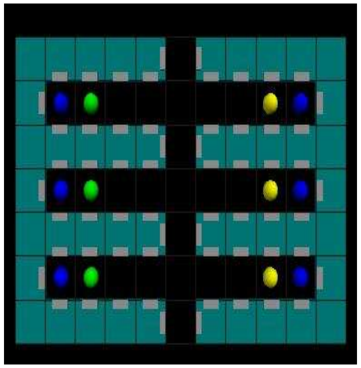
Quantum Circuit



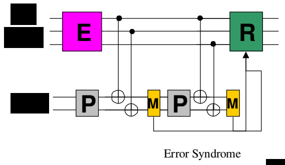
P = Prepare M = Measure

0 = Initialize

Layout and Physical Operations

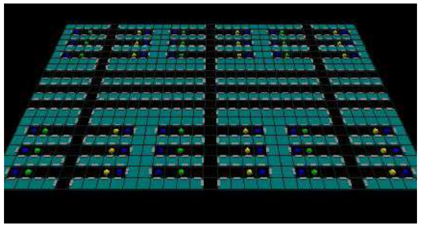


Quantum TMR



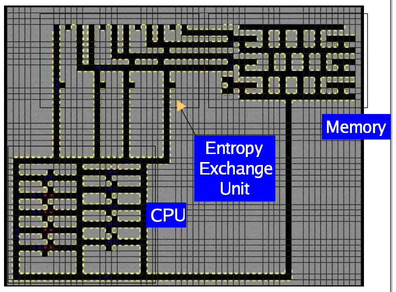
[redacted]

Layout to Any Recursion Depth



General Quantum Architecture

- Processing Units and Memory
- Preparation and Initialization Units
- Communication Strategies
  - Quantum Teleportation Channels
  - Swap Channels



Architecture in Action: Designing a Quantum Multicomputer

- Focus: Designing a Quantum Multicomputer
- Down a Level: Architectural Algorithmic Analysis
- Down a Level: Fast Quantum Arithmetic

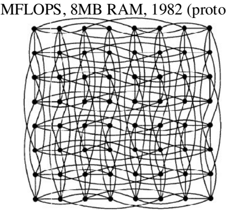
Goal: Design the Fastest, Most Scalable Quantum Computer Possible

Taking a page from the design of classical computers...

Two Paths to Scalability

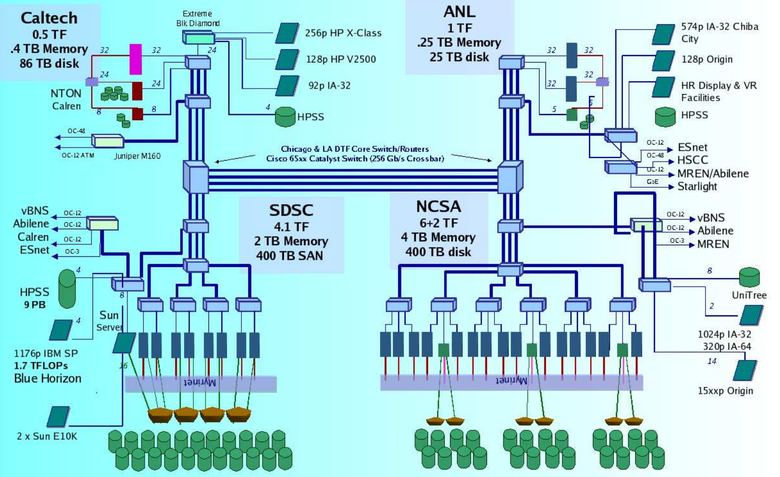
Cray 1, 80MFLOPS, 8MB RAM, $9M, 1976

Caltech Cosmic Cube, 64 processors (8086/7)
3MFLOPS, 8MB RAM, 1982 (prototype)



Two choices:
Make it bigger, or figure out how to connect more than one smaller unit hopefully achieving both *speed* and *storage capacity* increases

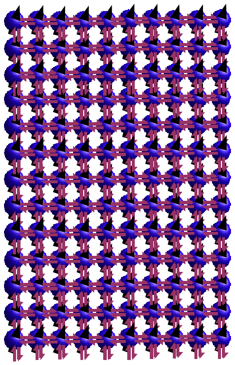
TeraGrid (Aug. 2004):
13.6 TFLOPS, 6.8 TB memory, 900 TB network disk, 10 PB archive
(courtesy Reagan Moore, SDSC)

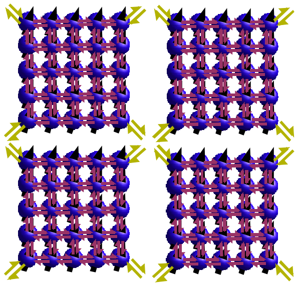


Two Paths to Scalability

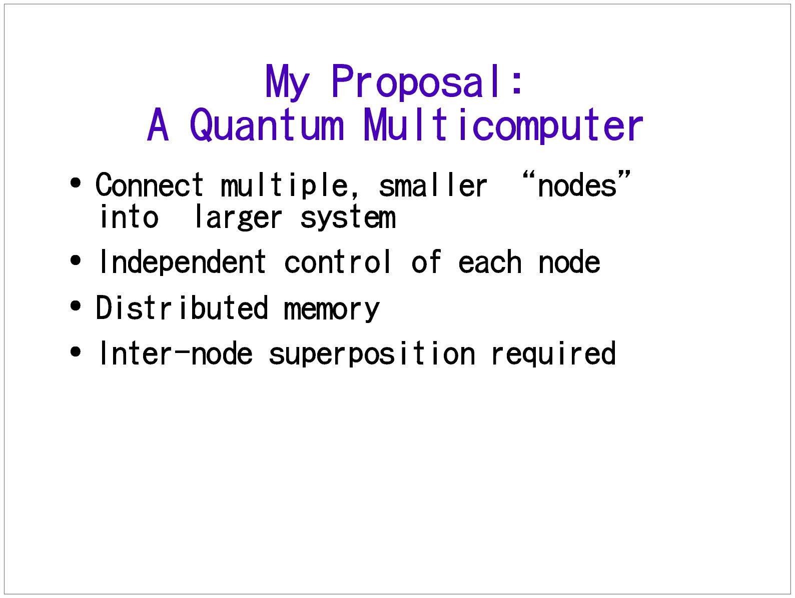
Parallel/distributed computing has clearly “won”.
But the real lesson is not One Big v. Lots of Little;
the lesson is that a group of computers can solve problems that an individual computer cannot.
Even the biggest individual computer can be combined in a network to create a larger *system.*

Two Paths to Scalability



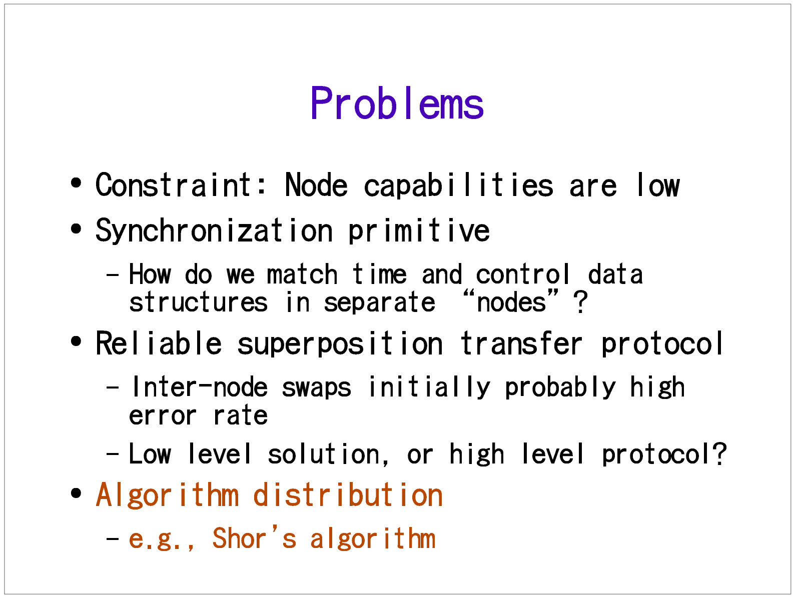


Two choices:
Make it bigger, or figure out how to connect more than one smaller unit hopefully achieving both *speed* and *storage capacity* increases



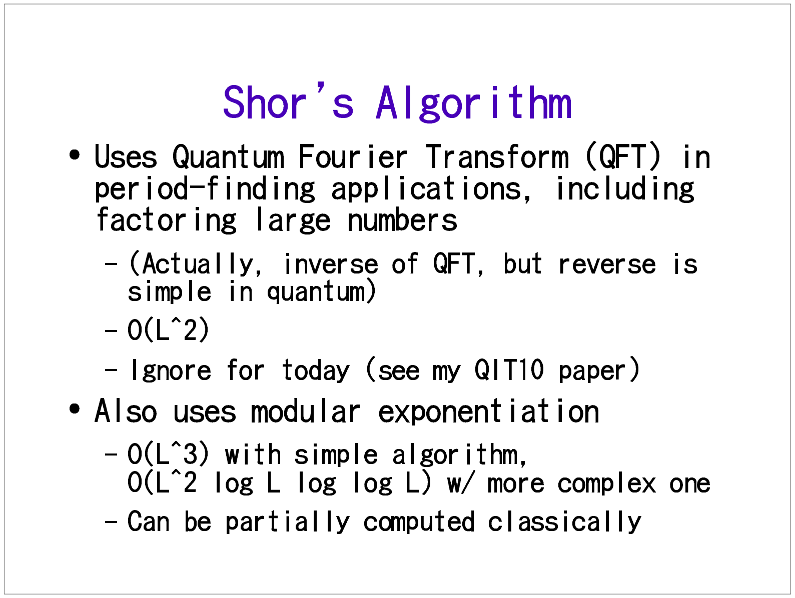
My Proposal: A Quantum Multicomputer

- Connect multiple, smaller “nodes” into larger system
- Independent control of each node
- Distributed memory
- Inter-node superposition required



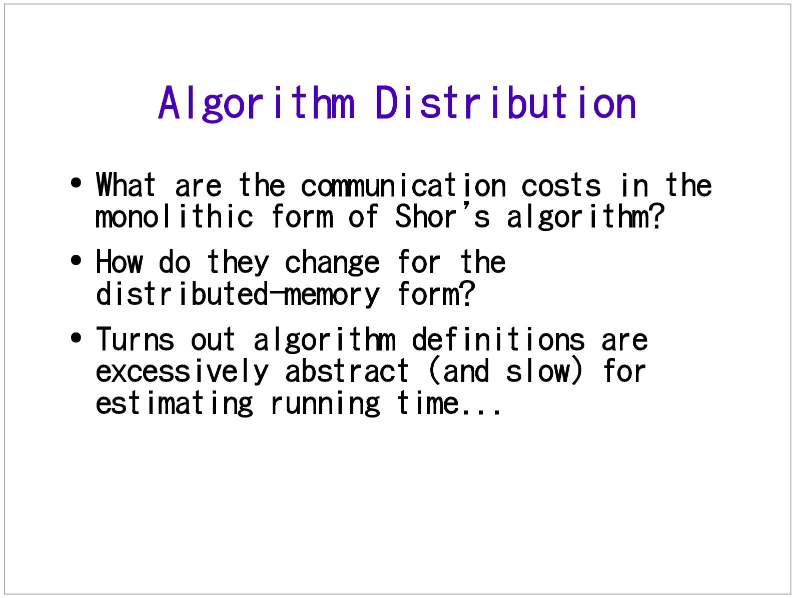
Problems

- Constraint: Node capabilities are low
- Synchronization primitive
  - How do we match time and control data structures in separate “nodes” ?
- Reliable superposition transfer protocol
  - Inter-node swaps initially probably high error rate
  - Low level solution, or high level protocol?
- Algorithm distribution
  - e.g., Shor's algorithm



Shor's Algorithm

- Uses Quantum Fourier Transform (QFT) in period-finding applications, including factoring large numbers
  - (Actually, inverse of QFT, but reverse is simple in quantum)
  - O(L^2)
  - Ignore for today (see my QIT10 paper)
- Also uses modular exponentiation
  - O(L^3) with simple algorithm,
    O(L^2 log L log log L) w/ more complex one
  - Can be partially computed classically



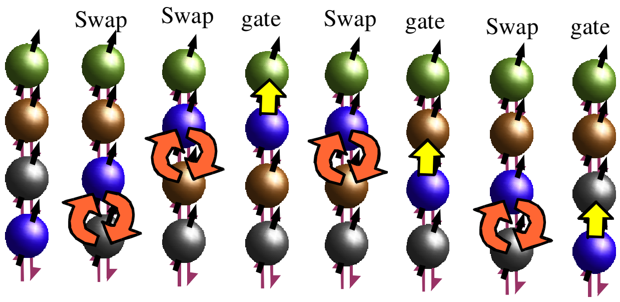
Algorithm Distribution

- What are the communication costs in the monolithic form of Shor's algorithm?
- How do they change for the distributed-memory form?
- Turns out algorithm definitions are excessively abstract (and slow) for estimating running time...

Architectural Algorithm Analysis

- Modular exponentiation
  - Vedral, Beckman, Gossett, Zalka...
- Many optimizations can be done
  - faster modulo, parallel multiplication, better adders, hand optimization
- VERY dependent on architecture
  - AC: abstract, w/ Toffoli
  - TC: two-qubit gates only
  - NTC: two-qubit, neighbors only
- Space and concurrency are critical!

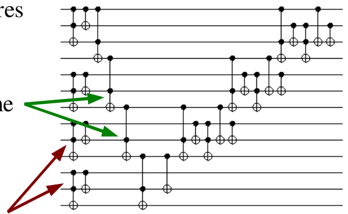
NTC: 1D Layout, Neighbors Only



Position first, then do action gates
Only 1 new neighbor after each swap
What's the performance penalty?

Carry-Ripple Adder

O(n) latency on all architectures



Carry ripples one bit at a time

(Can use concurrent
gates here, but limited above)

(Variants used in both major exponentiation algorithms,
VBE (quant-ph/9511018) and BCDP (quant-ph/9602016).)

Carry-Ripple Adder

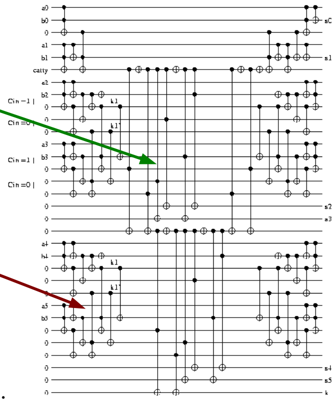
O(n) latency on all architectures

Carry ripples one bit at a time

(Can use concurrent
gates in first time slot, then limited)

(Variants used in both major exponentiation algorithms,
VBE (quant-ph/9511018) and BCDP (quant-ph/9602016).)

Conditional-Sum Adder



O(log n) latency when long-distance gates are easy. O(n) when swap required (NTC architecture) -- with a big constant!

Better use of concurrent gates (total still O(n) or larger).

(Carry-save and carry-lookahead are other types that reach O(log n). See quant-ph/9808061, quant-ph/0406142.)

Conditional-Sum Adder (AC)

O(log n) latency when long-distance gates are easy. O(n) when swap required (NTC architecture) -- with a big constant!

Better use of concurrent gates (total still O(n) or larger).

(Carry-save and carry-lookahead are other types that reach O(log n). See quant-ph/9808061, quant-ph/0406142.)

Conditional-Sum Adder (NTC)

O(log n) latency when long-distance gates are free. O(n) when swap required (NTC architecture) -- with a big constant!

Better use of concurrent gates (total still O(n) or larger).

(Carry-save and carry-lookahead are other types that reach O(log n). See quant-ph/9808061, quant-ph/0406142.)

Latency for Mod Exp (128 bits)

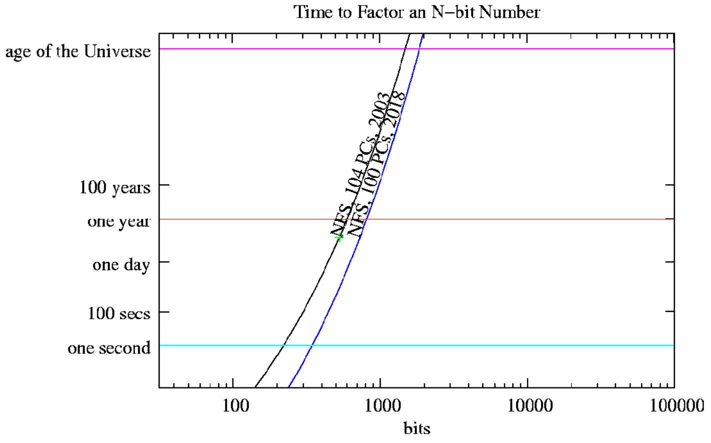
	AC gates	perf	TC gates	perf	NTC gates	perf
VBE	1.25E+08	1	4.99E+08	1	8.32E+08	1
BCDP	4.96E+07	2.5	1.32E+08	3.7	4.64E+08	1.8
VBE (100n)	7.56E+06	16	3.03E+07	16	5.05E+07	17
BCDP (100n)	2.53E+06	49	6.71E+06	74	2.36E+07	35
A	2.65E+07	4.7	1.07E+08	4.7	1.77E+08	4.7
B	3.71E+05	336	1.38E+06	360	1.71E+07	49
C	1.22E+06	102	4.89E+06	102	8.11E+06	103
D	2.19E+05	570	N/A		N/A	
E	1.71E+05	727	N/A		N/A	

B, C, D, E, VBE(100n), BCDP(100n) use 100n storage;
others use 5n-7n
gates for AC are CCNOT, others are CNOT

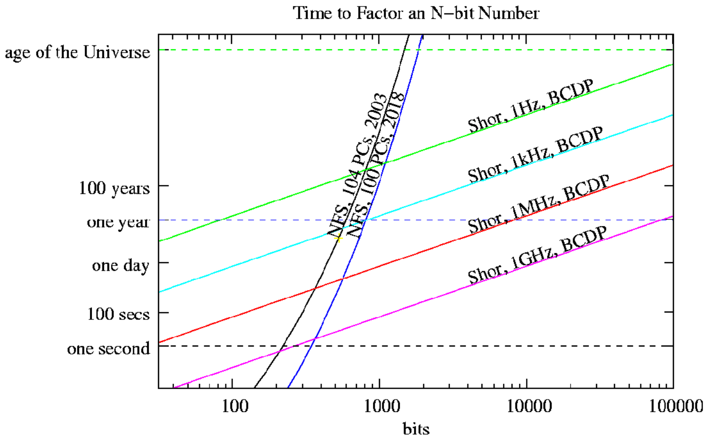
Factoring Larger Numbers

- 576 bits in a month current world record
  - previous record, 512 bits using 104 PCs, one month
- 512 bits in one month requires
  - using 5n space, no concurrency (original BCDP algorithm):
    - AC arch: 2800 Hz logical CCNOT
    - NTC arch: 78kHz logical 2-qubit gate
  - using 100n space, high concurrency (rdv algorithms E, C):
    - AC arch: 1.13 Hz logical CCNOT
    - NTC arch: 130 Hz logical 2-qubit gate

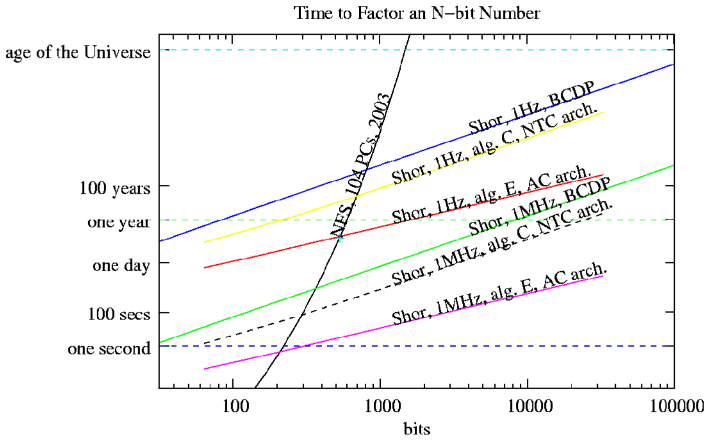
Factoring Larger Numbers



Factoring Larger Numbers



Factoring Larger Numbers

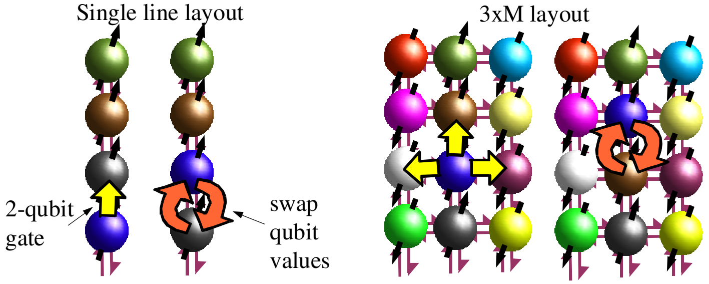


Arithmetic Summary

- Basic exponentiation algorithms O(n^3)
- Constant factors are critical!!!!
- Architecture impacts constant factors and asymptotic performance
  - Can reach O(log^3 n) w/ unlimited space on some architectures
  - Other archs O(n^2 log n)
  - Communications, concurrency important
- full paper at quant-ph/0408006
- JJAP letter in progress
- see also Fowler 0402196 & Devitt 0408081

Open Question: 2D Layout Efficiency

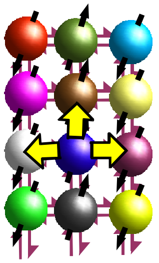
Does it give us an asymptotic change in O(.)?



Neighbor only operations require swapping qubits

3xM layout roughly twice as efficient
3 new neighbors after each swap
= 2/3 reduction in swaps

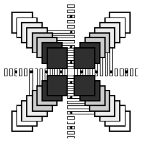
Algorithms on Other Architectures

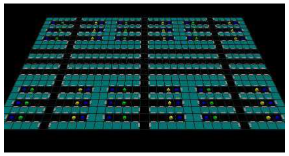


A lot is known about the computational complexity of circuits on arbitrary-interconnect machines; a little is known about circuit depth.

We and a few others (Fowler et al.) have investigated circuits on linear nearest neighbor (LNN or NTC) architectures.

Almost nothing is known about efficient circuits for 2D layout, lattices, scalable ion trap, etc.





Conclusions

- Architecture impacts constant factors and asymptotic performance
  - Key factors are concurrency, # of qubits, and interconnect topology
- Basic algorithms, including arithmetic, still need much work!
- Quantum multicomputer is ultimate goal
  - Solve larger problems, maybe faster