Study tradeoffs between area - reliability - performance.

Goals of this research: Reduce overall area of the design.

Leverage conventional architectural techniques to improve performance.

Provide abstractions for further research.
Background and prior work.

Overview of quantum error correction codes.

Specialization into memory and compute regions.

Improving performance.

Results and discussion.
Use ions trapped in electromagnetic fields.

Lasers acting on ions induce quantum gates.

Newer traps are micromachined.

Courtesy: C. Monroe at U. Michigan
Quantum Logic Array

Single logical qubit

Sea of lower level qubits

Repeaters

Compute-anywhere design.

Teleportation based long-distance communication.

Exponential speedup when factoring large numbers.

Unresolved Issue: Size
Sea of qubits design.

Teleportation based long-distance communication.

Exponential speedup when factoring large numbers.

Unresolved Issue: Size
Design Pyramid

- Speed
- Reliability
- Area
- QLA
Background and prior work.

Overview of quantum error correction codes.

Specialization into memory and compute regions.

Improving performance.

Results and discussion.
Comparison with Classical Codes

Classical three bit code

- Single bit encoded as three bits.
- Majority Voting.

Equivalent quantum code

- Nine qubit Shor code
- Protects against *bit*-flips and *phase*-flips.
Comparison with Classical Codes

Classical three bit code

- Single bit encoded as three bits.
- Majority Voting.

Equivalent quantum code

- Nine qubit Shor code
  - Protects against *bit-flips* and *phase-flips*. 
Greater Reliability

Need greater reliability than provided by encoding a single time.

The No cloning theorem and restrictions on measurement require greater reliability.

Cannot use methods like checkpointing or make duplicates.

Solution: Use concatenated codes.
Concatenated Codes

1 logical qubit

Level 1: 7 physical qubits

Level 2: 49 physical qubits

Concatenated Steane Code

Reliability increases doubly exponentially.

Exponentially slower.

Exponentially greater resources.
Outline

Background and prior work.

Overview of quantum error correction codes.

Specialization into memory and compute regions.

Improving performance.

Results and discussion.
Conventional wisdom: \textbf{Max. parallelism} necessary to minimize computation time and reduce prob. of failure.
Shor's quantum algorithm to find factors of very large numbers yields exponential speedup over classical algorithms.

Modular exponentiation is the most compute intensive part of Shor's factoring algorithm.

Primary component: Draper carry-lookahead adder (quantum version of the classical adder).
App. Constrained Parallelism

Create slower but denser memory region and faster but sparse compute region.
Logical data qubits

Logical ancilla qubits

An ion when idle has a lifetime of $\sim 10$ sec
CQLA: Compressed QLA

Compute Block

Memory Block

Compute Block
Area Reduction

![Bar Chart]

Shor’s Alg. Input Size

- 64-bit
- 256-bit
- 512-bit
- 1024-bit

<table>
<thead>
<tr>
<th>Size</th>
<th>Factor of</th>
<th>Perf. Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit</td>
<td>6.4</td>
<td>-20%</td>
</tr>
<tr>
<td>256-bit</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td>512-bit</td>
<td>7.4</td>
<td></td>
</tr>
<tr>
<td>1024-bit</td>
<td>9.1</td>
<td></td>
</tr>
</tbody>
</table>
CQLA: Reduced Size

QLA: 90cm x 90cm

CQLA: 28cm x 28cm
Design Pyramid: CQLA

- Speed
- Reliability
- Area

QLA

CQLA

Area

Reliability

Speed
Outline

Background and prior work.

Overview of quantum error correction codes.

Specialization into memory and compute regions.

**Improving** performance.

Results and discussion.
Concatenated Codes

Reliability increases doubly exponentially.
Exponentially slower.
Exponentially greater resources.

1 logical qubit

Level 1: 7 physical qubits

Level 2: 49 physical qubits

Concatenated Steane Code
### Compute @ Level 1

<table>
<thead>
<tr>
<th>Level 2 Encoding</th>
<th>Memory: Very reliable and slow. (Periodic error-correction)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compute: Very reliable and fast. (49bit quantum operations and error-correction)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 1 Encoding</th>
<th>Cache: Less reliable. (Infrequent Error-correction)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compute: Less reliable, exponentially faster. (7bit quantum operations and error-correction)</td>
</tr>
</tbody>
</table>

Transfer between encoding levels
Faster CQLA
## Overall Results

### Shor’s Alg. Input Size

<table>
<thead>
<tr>
<th></th>
<th>256-bit</th>
<th>512-bit</th>
<th>1024-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Reduced</td>
<td>5.0</td>
<td>6.0</td>
<td>9.1</td>
</tr>
<tr>
<td>L1 SpeedUp</td>
<td>17.4</td>
<td>17.4</td>
<td>18.2</td>
</tr>
<tr>
<td>Total SpeedUp</td>
<td>6.2</td>
<td>6.2</td>
<td>4.9</td>
</tr>
</tbody>
</table>

- **256-bit**
  - Area Reduced: 5.0
  - L1 SpeedUp: 17.4
  - Total SpeedUp: 6.2

- **512-bit**
  - Area Reduced: 6.0
  - L1 SpeedUp: 17.4
  - Total SpeedUp: 6.2

- **1024-bit**
  - Area Reduced: 9.1
  - L1 SpeedUp: 18.2
  - Total SpeedUp: 4.9

The chart above illustrates the factor of improvement in Area Reduced and SpeedUp for different input sizes (256-bit, 512-bit, 1024-bit) using Shor’s algorithm.
Design Pyramid: QLA

QLA

Area

Reliability

Speed
Design Pyramid: CQLA
Parallelism in quantum computing constrained by applications.

Different scheduling mechanisms of quantum operations.

Introduced a memory hierarchy for quantum computers.

Area reduced factor of 9 and speedup of factor of 4.
Even better results using the Bacon-Shor quantum error-correction code.

Area reduced by a factor of 13.
Speedup of factor of 8.

Details of transfer networks to enable change in encodings.
Limited control signals: Incorporate studies of laser resources and laser power.

Incorporating fault tolerance into compiler optimization: Compiler techniques to reduce error-correction costs.
¿Questions?

Project webpage: http://aar.cs.ucdavis.edu/qarc

Your questions...
## Overall Results

<table>
<thead>
<tr>
<th>Par Xfer</th>
<th>Adder Size</th>
<th>L1 SpeedUp</th>
<th>L2 SpeedUp</th>
<th>Adder SpeedUp</th>
<th>Area Reduced</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Steane [[7, 1, 3]] Code</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>256</td>
<td>17.4</td>
<td>0.9</td>
<td>6.2</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>17.4</td>
<td>0.9</td>
<td>6.3</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>18.2</td>
<td>0.8</td>
<td><strong>4.9</strong></td>
<td><strong>9.1</strong></td>
</tr>
<tr>
<td>5</td>
<td>256</td>
<td>10.4</td>
<td>0.9</td>
<td>4.0</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>10.4</td>
<td>0.9</td>
<td>4.0</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>10.9</td>
<td>0.8</td>
<td><strong>2.9</strong></td>
<td><strong>9.1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bacon-Shor [[9, 1, 3]] Code</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>256</td>
<td>9.6</td>
<td>1.5</td>
<td>5.9</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>9.6</td>
<td>2.3</td>
<td>8.8</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>10.1</td>
<td>2.0</td>
<td><strong>8.1</strong></td>
<td><strong>13.4</strong></td>
</tr>
<tr>
<td>5</td>
<td>256</td>
<td>5.2</td>
<td>1.5</td>
<td>3.7</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>5.2</td>
<td>2.3</td>
<td>5.4</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>5.4</td>
<td>2.0</td>
<td><strong>4.9</strong></td>
<td><strong>13.4</strong></td>
</tr>
</tbody>
</table>
Concatenated Codes

- 1 logical qubit
  - Level 1: 9 physical qubits
  - Level 2: 81 physical qubits

Reliability increases doubly exponentially.
Exponentially slower.
Exponentially greater resources.
Let memory remain at Level 2 encoding.

**Compute** at **Level 1** encoding.

**Drawbacks:** Reliability **degrades.**

**Transfer** between Level 1 and Level 2 is **very expensive.**

Use a **cache** to alleviate transfer costs.

**Improve Performance**
## Size Reduction

<table>
<thead>
<tr>
<th>Input Size</th>
<th>Compute Blocks</th>
<th>Area Reduced (Factor of)</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit</td>
<td>9</td>
<td>6.4</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.7</td>
<td>0.9</td>
</tr>
<tr>
<td>256-bit</td>
<td>36</td>
<td>6.6</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>49</td>
<td>5.0</td>
<td>0.9</td>
</tr>
<tr>
<td>512-bit</td>
<td>64</td>
<td>7.4</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>6.0</td>
<td>0.9</td>
</tr>
<tr>
<td>1024-bit</td>
<td>100</td>
<td>9.1</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>121</td>
<td>7.8</td>
<td>0.9</td>
</tr>
</tbody>
</table>
## Overall Results

<table>
<thead>
<tr>
<th>Parallel Transfers</th>
<th>Input Size</th>
<th>L1 Speed Up</th>
<th>L2 Speed Up</th>
<th>Total Speed Up</th>
<th>Area Reduced (Factor of)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>256</td>
<td>10.4</td>
<td>0.9</td>
<td>4.0</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>10.4</td>
<td>0.9</td>
<td>4.0</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>10.9</td>
<td>0.8</td>
<td>2.9</td>
<td>9.1</td>
</tr>
<tr>
<td>10</td>
<td>256</td>
<td>17.4</td>
<td>0.9</td>
<td>6.2</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>17.4</td>
<td>0.9</td>
<td>6.3</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td><strong>18.2</strong></td>
<td><strong>0.8</strong></td>
<td><strong>4.9</strong></td>
<td><strong>9.1</strong></td>
</tr>
</tbody>
</table>