

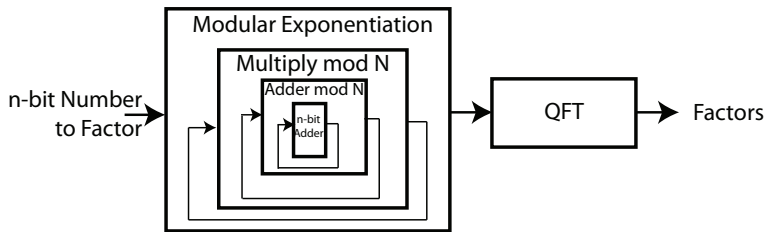
A Fault Tolerant, Area-Efficient Architecture for Shor's Factoring Algorithm

Mark Whitney, Nemanja Isailovic, Yatish Patel,
John Kubiawicz

Univ. of California, Berkeley

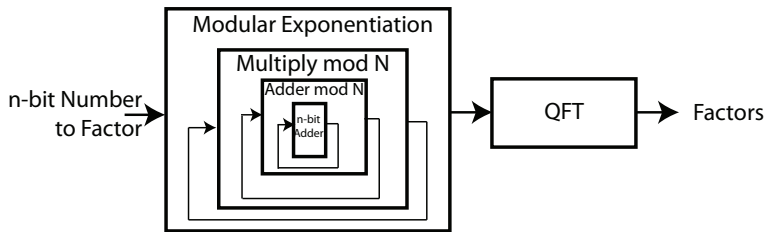
June 23, 2009

Shor's Factoring Algorithm



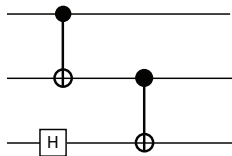
- ▶ Killer app for quantum computing
 - ▶ Runtime polynomial in number of bits
- ▶ Key component to exponentiation: quantum adder
 - ▶ 1024-bit number: billions of operations, 100,000s of qubits
- ▶ High failure rates require strong fault tolerance

Shor's Factoring Algorithm

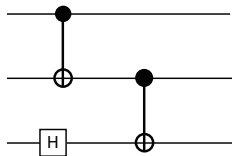


- ▶ Killer app for quantum computing
 - ▶ Runtime polynomial in number of bits
- ▶ Key component to exponentiation: quantum adder
 - ▶ 1024-bit number: billions of operations, 100,000s of qubits
- ▶ High failure rates require strong fault tolerance
 - ▶ Previous area estimates for design were $0.9m^2$
 - ▶ 95% of operations are for fault tolerance

Quantum Fault Tolerance

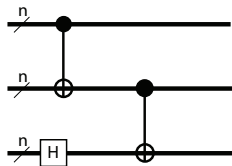


Quantum Fault Tolerance



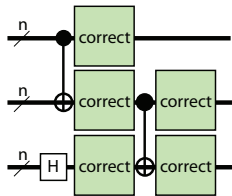
- ▶ Elements of fault tolerance:

Quantum Fault Tolerance



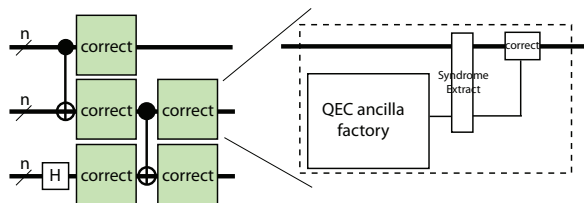
- ▶ Elements of fault tolerance:
 - ▶ Encode qubits in quantum error correcting code (QEC)

Quantum Fault Tolerance



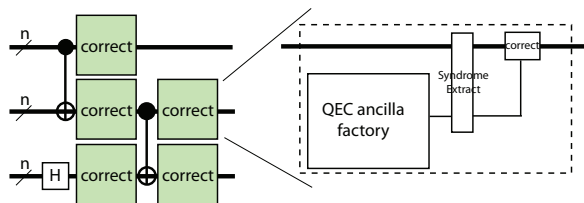
- ▶ Elements of fault tolerance:
 - ▶ Encode qubits in quantum error correcting code (QEC)
 - ▶ Add error correction modules

Quantum Fault Tolerance



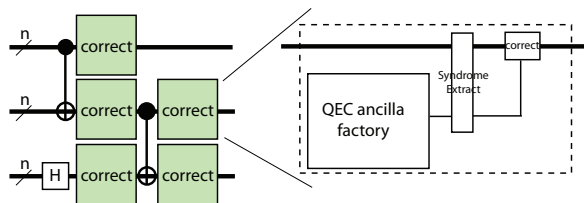
- ▶ Elements of fault tolerance:
 - ▶ Encode qubits in quantum error correcting code (QEC)
 - ▶ Add error correction modules
- ▶ 3 pieces to error correction operation:
 - ▶ Ancilla (helper “parity” bit) production
 - ▶ Error syndrome
 - ▶ Qubit correction

Quantum Fault Tolerance



- ▶ Elements of fault tolerance:
 - ▶ Encode qubits in quantum error correcting code (QEC)
 - ▶ Add error correction modules
- ▶ 3 pieces to error correction operation:
 - ▶ Ancilla (helper “parity” bit) production
 - ▶ Error syndrome
 - ▶ Qubit correction
- ▶ $\geq 90\%$ of QEC gates are ancilla production
- ▶ $\geq 85\%$ of all gates are ancilla production

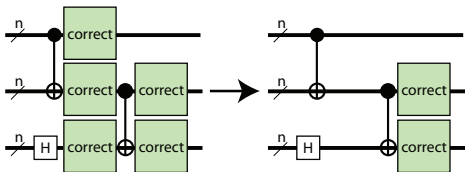
Quantum Fault Tolerance



- ▶ Elements of fault tolerance:
 - ▶ Encode qubits in quantum error correcting code (QEC)
 - ▶ Add error correction modules
- ▶ 3 pieces to error correction operation:
 - ▶ Ancilla (helper “parity” bit) production
 - ▶ Error syndrome
 - ▶ Qubit correction
- ▶ $\geq 90\%$ of QEC gates are ancilla production
- ▶ $\geq 85\%$ of all gates are ancilla production
- ▶ **Efficient design requires managing ancilla production**

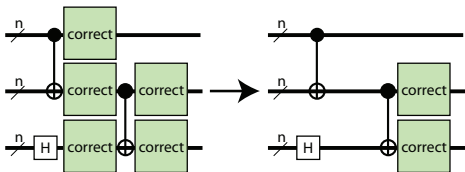
Reducing FT Overhead

- ▶ 2 choices for reducing overhead:
 - ▶ Improve ancilla supply/make more efficient
 - ▶ Decrease demand for ancilla
- ▶ Our contributions:
 - ▶ Qalypso: flexible, efficient ancilla production
 - ▶ Error correction optimization to reduce correct steps



Reducing FT Overhead

- ▶ 2 choices for reducing overhead:
 - ▶ Improve ancilla supply/make more efficient
 - ▶ Decrease demand for ancilla
- ▶ Our contributions:
 - ▶ Qalypso: flexible, efficient ancilla production
 - ▶ Error correction optimization to reduce correct steps

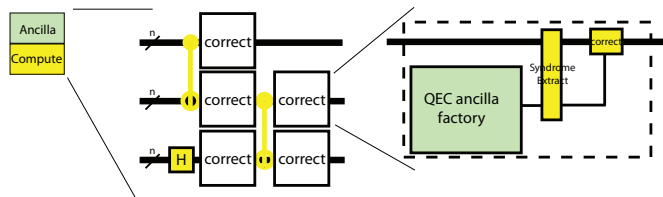


Tiled Quantum Architectures



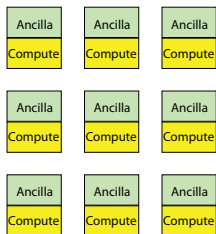
- ▶ Compute and QEC ancilla regions in a single tile

Tiled Quantum Architectures



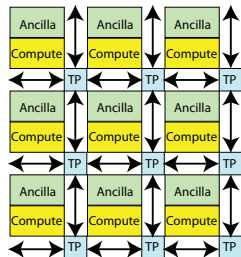
- ▶ Compute and QEC ancilla regions in a single tile

Tiled Quantum Architectures



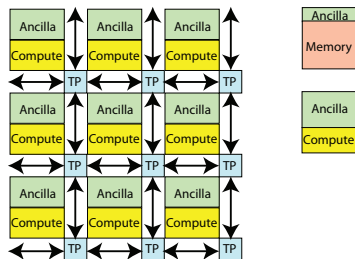
- ▶ Compute and QEC ancilla regions in a single tile

Tiled Quantum Architectures



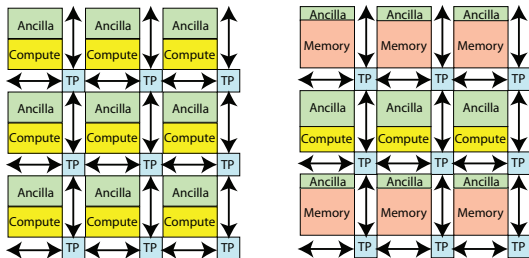
- ▶ Compute and QEC ancilla regions in a single tile
- ▶ Interconnection network connects tiles
 - ▶ Previous work: QLA, LQLA

Tiled Quantum Architectures



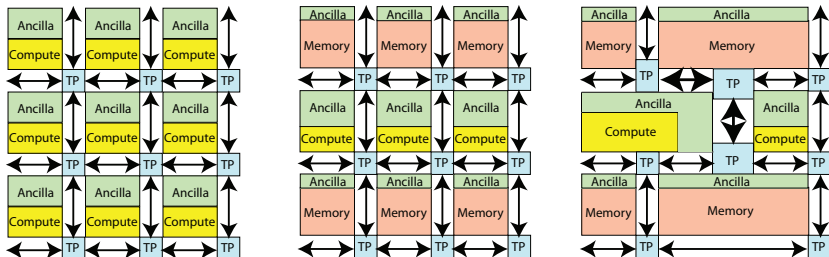
- ▶ Compute and QEC ancilla regions in a single tile
- ▶ Interconnection network connects tiles
 - ▶ Previous work: QLA, LQLA
- ▶ Specialized memory and compute regions
 - ▶ Previous work: CQLA, CQLA+

Tiled Quantum Architectures



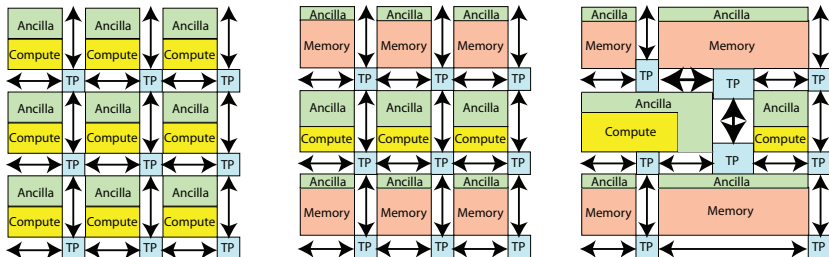
- ▶ Compute and QEC ancilla regions in a single tile
- ▶ Interconnection network connects tiles
 - ▶ Previous work: QLA, LQLA
- ▶ Specialized memory and compute regions
 - ▶ Previous work: CQLA, CQLA+

Tiled Quantum Architectures



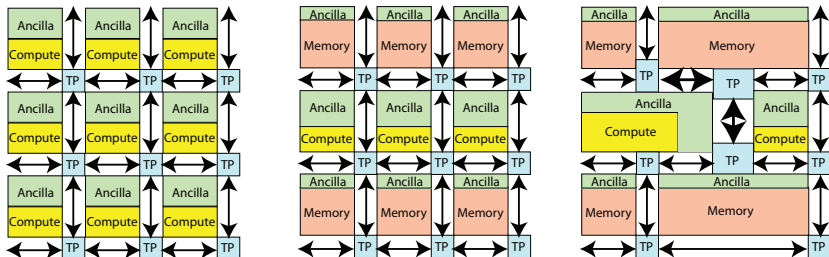
- ▶ Compute and QEC ancilla regions in a single tile
- ▶ Interconnection network connects tiles
 - ▶ Previous work: QLA, LQLA
- ▶ Specialized memory and compute regions
 - ▶ Previous work: CQLA, CQLA+
- ▶ Our contribution: Qalypso: flexible memory, compute, QEC

Tiled Quantum Architectures



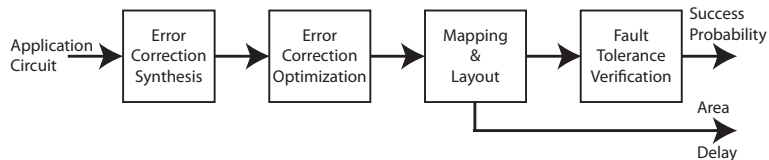
- ▶ Compute and QEC ancilla regions in a single tile
- ▶ Interconnection network connects tiles
 - ▶ Previous work: QLA, **LQLA**
- ▶ Specialized memory and compute regions
 - ▶ Previous work: CQLA, **CQLA+**
- ▶ Our contribution: **Qalypso**: flexible memory, compute, QEC

Tiled Quantum Architectures



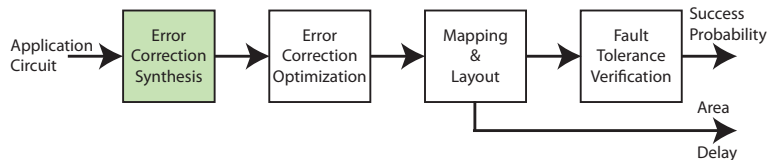
- ▶ Compute and QEC ancilla regions in a single tile
- ▶ Interconnection network connects tiles
 - ▶ Previous work: QLA, **LQLA**
- ▶ Specialized memory and compute regions
 - ▶ Previous work: CQLA, **CQLA+**
- ▶ Our contribution: **Qalypto**: flexible memory, compute, QEC
- ▶ Parameters: number of tiles, compute/memory distribution

Computer Aided Design Flow



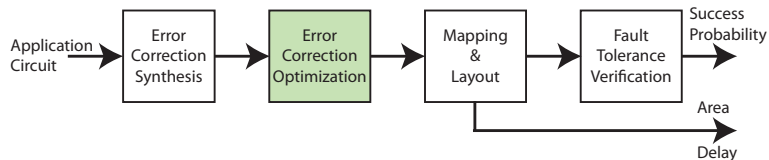
- ▶ Input: Application circuit (Shor's, etc.)
- ▶ Output: Fault tolerant, spatial layout of circuit

Computer Aided Design Flow



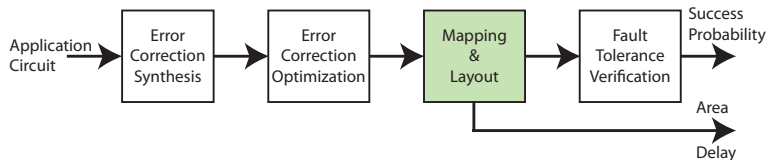
- ▶ Input: Application circuit (Shor's, etc.)
- ▶ Output: Fault tolerant, spatial layout of circuit

Computer Aided Design Flow



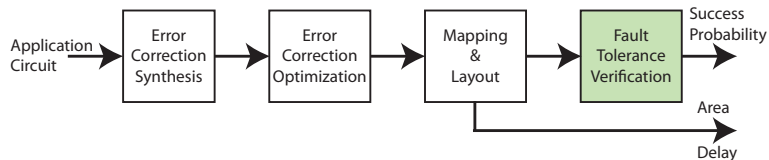
- ▶ Input: Application circuit (Shor's, etc.)
- ▶ Output: Fault tolerant, spatial layout of circuit

Computer Aided Design Flow



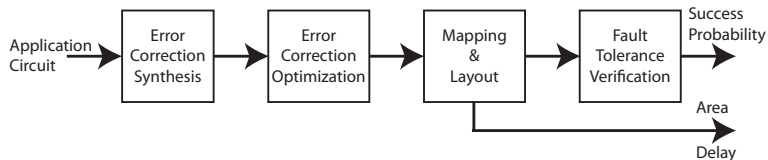
- ▶ Input: Application circuit (Shor's, etc.)
- ▶ Output: Fault tolerant, spatial layout of circuit

Computer Aided Design Flow



- ▶ Input: Application circuit (Shor's, etc.)
- ▶ Output: Fault tolerant, spatial layout of circuit

Computer Aided Design Flow



- ▶ Input: Application circuit (Shor's, etc.)
- ▶ Output: Fault tolerant, spatial layout of circuit
- ▶ Advantages:
 - ▶ Detailed layout allows accurate comparison of designs
 - ▶ Can search large space of configurations

Metrics

- ▶ Basic metrics:
 - ▶ Probability no error in output: $p_{success}$
 - ▶ Area
 - ▶ Delay of single run of circuit: D_{single}

Metrics

- ▶ Basic metrics:
 - ▶ Probability no error in output: $p_{success}$
 - ▶ Area
 - ▶ Delay of single run of circuit: D_{single}
- ▶ Tradeoff area, D_{single} , $p_{success}$ in design

Metrics

- ▶ Basic metrics:
 - ▶ Probability no error in output: $p_{success}$
 - ▶ Area
 - ▶ Delay of single run of circuit: D_{single}
- ▶ Tradeoff area, D_{single} , $p_{success}$ in design
- ▶ ADCR: Area-Delay to Correct Result

Metrics

- ▶ Basic metrics:
 - ▶ Probability no error in output: $p_{success}$
 - ▶ Area
 - ▶ Delay of single run of circuit: D_{single}
- ▶ Tradeoff area, D_{single} , $p_{success}$ in design
- ▶ ADCR: Area-Delay to Correct Result
 - ▶ $E(Delay) = D_{single} \times E_{correct}(runs) = \frac{D_{single}}{p_{success}}$

Metrics

- ▶ Basic metrics:
 - ▶ Probability no error in output: $p_{success}$
 - ▶ Area
 - ▶ Delay of single run of circuit: D_{single}
- ▶ Tradeoff area, D_{single} , $p_{success}$ in design
- ▶ ADCR: Area-Delay to Correct Result
 - ▶ $E(Delay) = D_{single} \times E_{correct}(runs) = \frac{D_{single}}{p_{success}}$
 - ▶ $ADCR = Area \times E(Delay) = \frac{Area \times D_{single}}{p_{success}}$
 - ▶ Area efficiency of probabilistic circuits

Metrics

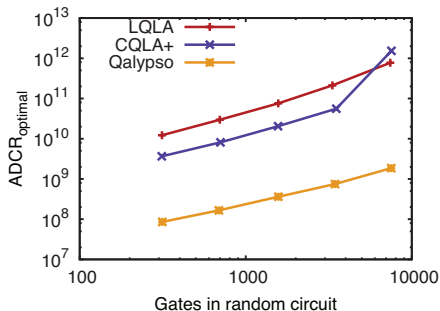
- ▶ Basic metrics:
 - ▶ Probability no error in output: $p_{success}$
 - ▶ Area
 - ▶ Delay of single run of circuit: D_{single}
- ▶ Tradeoff area, D_{single} , $p_{success}$ in design
- ▶ ADCR: Area-Delay to Correct Result
 - ▶ $E(Delay) = D_{single} \times E_{correct}(runs) = \frac{D_{single}}{p_{success}}$
 - ▶ $ADCR = Area \times E(Delay) = \frac{Area \times D_{single}}{p_{success}}$
 - ▶ Area efficiency of probabilistic circuits
- ▶ $ADCR_{optimal}$: best ADCR for over all possible configurations

Architectural Evaluation on Random Circuits

- ▶ $ADCR_{optimal}$ comparison on random circuits
- ▶ Compare best performing archs: Qalypso, LQLA, and CQLA+

Architectural Evaluation on Random Circuits

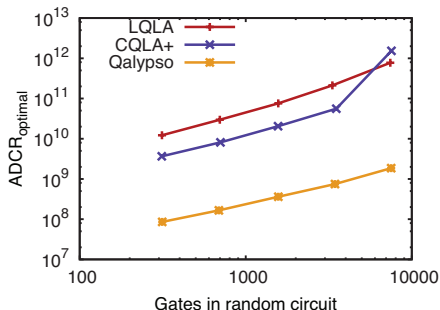
- ▶ $ADCR_{optimal}$ comparison on random circuits
- ▶ Compare best performing archs: Qalypso, LQLA, and CQLA+



- ▶ Qalypso has significantly lower ADCR than previous work

Architectural Evaluation on Random Circuits

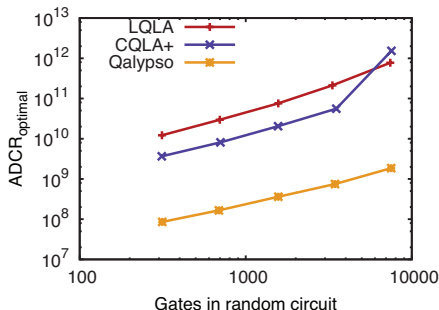
- ▶ $ADCR_{optimal}$ comparison on random circuits
- ▶ Compare best performing archs: Qalypso, LQLA, and CQLA+



- ▶ Qalypso has significantly lower ADCR than previous work
 - ▶ 4x lower latency than LQLA
 - ▶ 2x smaller area than CQLA+

Architectural Evaluation on Random Circuits

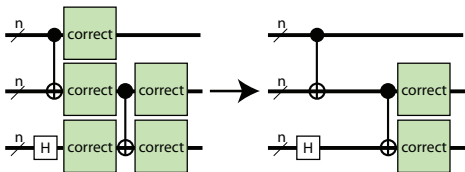
- ▶ $ADCR_{optimal}$ comparison on random circuits
- ▶ Compare best performing archs: Qalypso, LQLA, and CQLA+



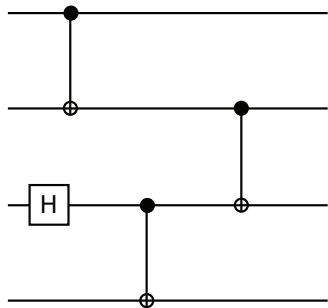
- ▶ Qalypso has significantly lower ADCR than previous work
 - ▶ 4x lower latency than LQLA
 - ▶ 2x smaller area than CQLA+
- ▶ Qalypso targets ancilla production to performance critical tiles

Reducing FT Overhead

- ▶ 2 choices for reducing overhead:
 - ▶ Improve ancilla supply/make more efficient
 - ▶ **Decrease demand for ancilla**
- ▶ Our contributions:
 - ▶ Qalypso: more flexible, efficient ancilla production
 - ▶ **Error correction optimization to reduce correct steps**

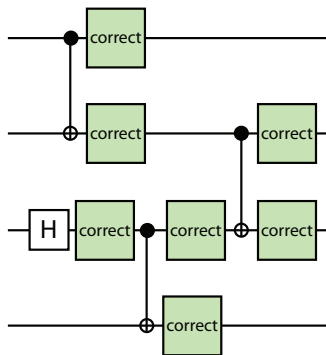


Reducing QEC Overhead



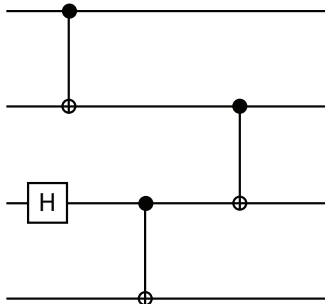
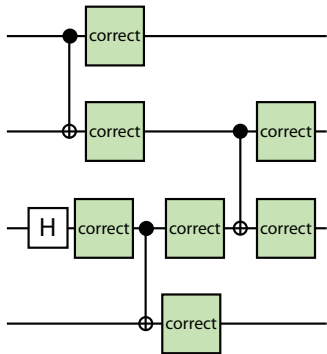
Reducing QEC Overhead

- ▶ Standard approach: insert error correction steps everywhere



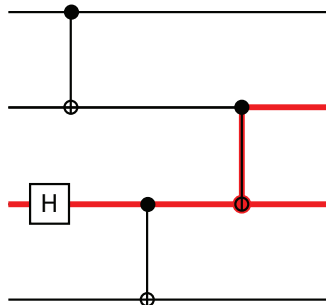
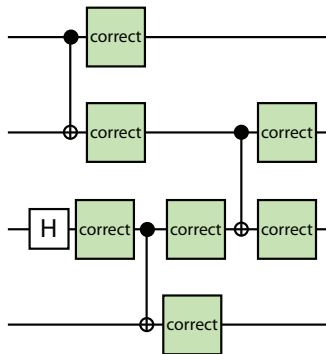
Reducing QEC Overhead

- ▶ Standard approach: insert error correction steps everywhere
- ▶ Our approach:



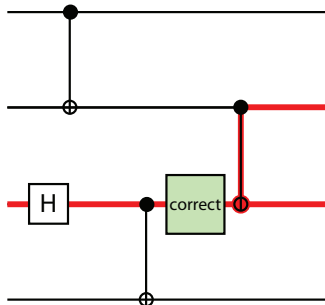
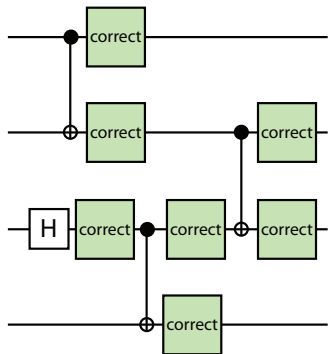
Reducing QEC Overhead

- ▶ Standard approach: insert error correction steps everywhere
- ▶ Our approach:
 - ▶ Identify critical error paths



Reducing QEC Overhead

- ▶ Standard approach: insert error correction steps everywhere
- ▶ Our approach:
 - ▶ Identify critical error paths
 - ▶ Add correction steps where “most important”

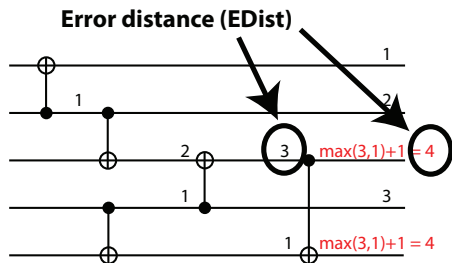


Simple Error Model for FT Circuits

- ▶ Identifying critical error paths

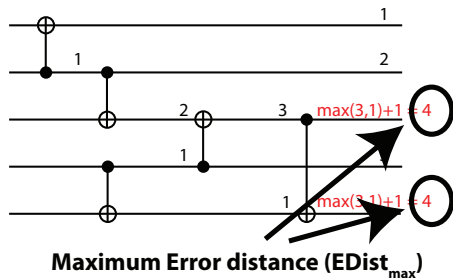
Simple Error Model for FT Circuits

- ▶ Identifying critical error paths
 - ▶ Gates propagate max input error distance to all outputs
 - ▶ Gates add 1 to error distance



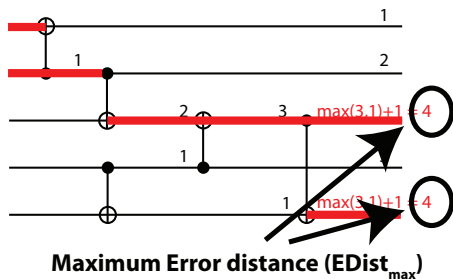
Simple Error Model for FT Circuits

- ▶ Identifying critical error paths
 - ▶ Gates propagate max input error distance to all outputs
 - ▶ Gates add 1 to error distance



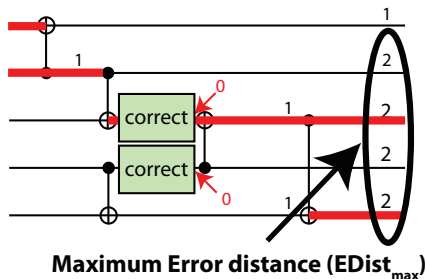
Simple Error Model for FT Circuits

- ▶ Identifying critical error paths
 - ▶ Gates propagate max input error distance to all outputs
 - ▶ Gates add 1 to error distance
 - ▶ $EDist_{max}$ corresponds to critical paths



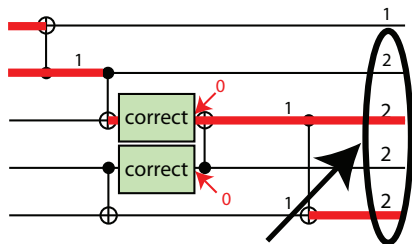
Simple Error Model for FT Circuits

- ▶ Identifying critical error paths
 - ▶ Gates propagate max input error distance to all outputs
 - ▶ Gates add 1 to error distance
 - ▶ $EDist_{max}$ corresponds to critical paths
- ▶ Corrections reduce error counts



Simple Error Model for FT Circuits

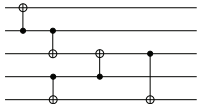
- ▶ Identifying critical error paths
 - ▶ Gates propagate max input error distance to all outputs
 - ▶ Gates add 1 to error distance
 - ▶ $EDist_{max}$ corresponds to critical paths
- ▶ Corrections reduce error counts



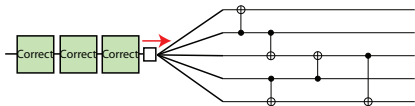
Maximum Error distance ($EDist_{max}$)

- ▶ Similar to delay in synchronous circuits
- ▶ Apply classical retiming approach: recorection
 - ▶ Corrections balance $EDist$ → Synch registers balance delay

QEC Optimization Through Circuit Recorrection

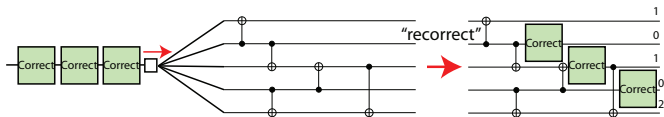


QEC Optimization Through Circuit Recorrection



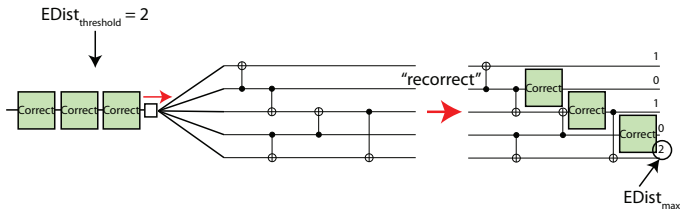
- ▶ Add correct ops at front of circuit

QEC Optimization Through Circuit Recorrection



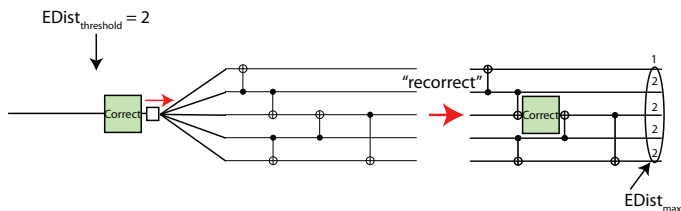
- ▶ Add correct ops at front of circuit

QEC Optimization Through Circuit Recorrection



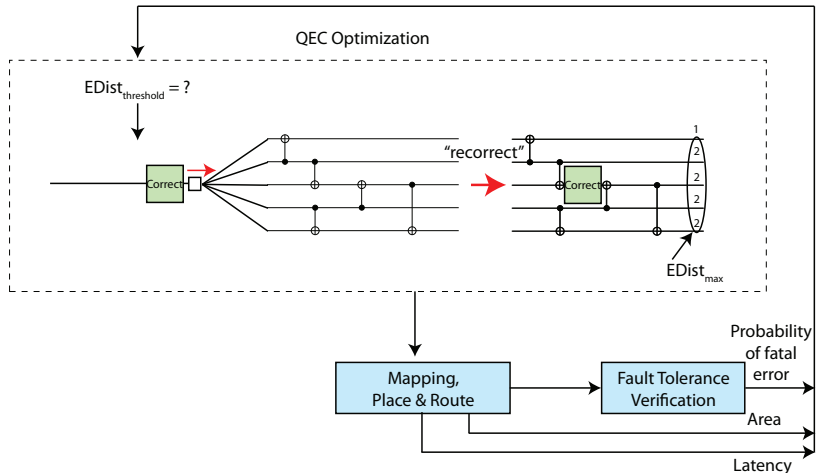
- ▶ Add correct ops at front of circuit
 - ▶ Add enough corrects to reach a target $EDist_{\text{threshold}}$

QEC Optimization Through Circuit Recorrection



- ▶ Add correct ops at front of circuit
 - ▶ Add enough corrects to reach a target $EDist_{\text{threshold}}$
- ▶ Find minimal number of corrects so $EDist_{\text{max}} \leq EDist_{\text{threshold}}$

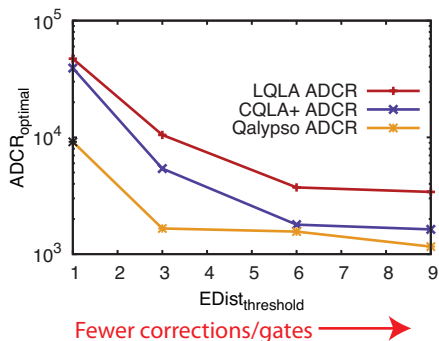
QEC Optimization Through Circuit Recorrection



- ▶ Add correct ops at front of circuit
 - ▶ Add enough corrects to reach a target $EDist_{threshold}$
- ▶ Find minimal number of corrects so $EDist_{max} \leq EDist_{threshold}$
- ▶ Find $EDist_{threshold}$ for desired $p_{success}/area/D_{single}/ADCR$

Optimization and Adder Performance

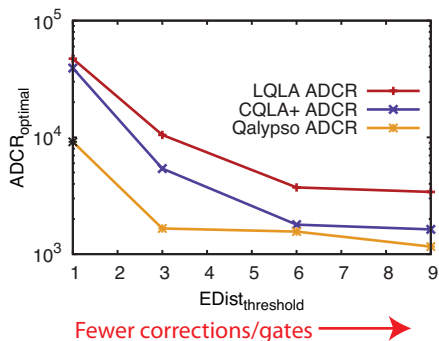
► 1024-bit Quantum Carry Lookahead Adder



► Higher $EDist_{threshold}$ = fewer corrections, more optimization

Optimization and Adder Performance

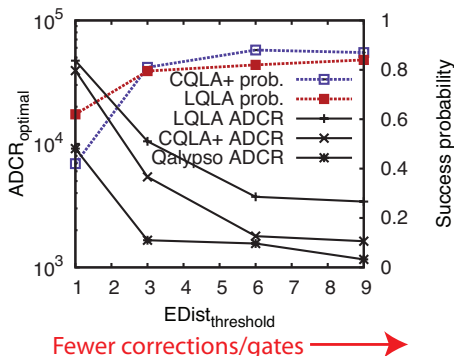
► 1024-bit Quantum Carry Lookahead Adder



- Higher $EDist_{threshold}$ = fewer corrections, more optimization
- $\geq 10x$ improvement for CQLA+ and LQLA
- $6x$ improvement for Qalypso

Optimization and Adder Performance

▶ 1024-bit Quantum Carry Lookahead Adder



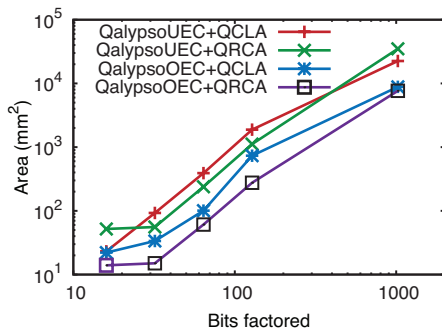
- ▶ Higher $EDist_{threshold}$ = fewer corrections, more optimization
- ▶ $\geq 10\times$ improvement for CQLA+ and LQLA
- ▶ $6\times$ improvement for Qalypto

Shor's Performance

- ▶ Full Shor's factorization - billions of ops for 1024
- ▶ Include ripple carry adder design
- ▶ 2x reduction in latency for optimization (QCLA opt best)

Shor's Performance

- ▶ Full Shor's factorization - billions of ops for 1024
- ▶ Include ripple carry adder design
- ▶ 2x reduction in latency for optimization (QCLA opt best)



- ▶ 5x reduction in area for optimization (QRCA opt best)
- ▶ Best optimized Qalypso 1024-bit design is approx $0.01m^2$

Conclusion

- ▶ ADCR: New metric for evaluating FT quantum circuits
 - ▶ Area efficiency metric including reliability, area, delay
- ▶ Qalypso outperforms other QC architectures
 - ▶ Detailed layout and simulation allows accurate comparison
 - ▶ CAD flow enables automated search of configuration space
- ▶ Error correction optimization reduces area and latency
 - ▶ Minimal impact to reliability
- ▶ Together orders of magnitude area improvement for Shor's factoring