Ongoing Projects on Quantum Circuits and Algorithms

Igor L. Markov and John P. Hayes

Advanced Computer Architecture Laboratory
EECS Department
University of Michigan,
Ann Arbor, MI 48109

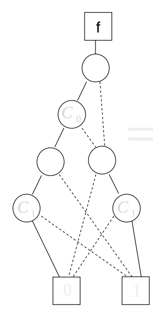
Ongoing Projects

- Simulation of quantum circuits
  - BDD-based QuIDDPro simulator
  - Simulating Grover's algorithm
- Synthesis of two-qubit circuits
  - Bounds for gate counts in two-qubit circuits
- Quantum algos that improve memory usage
  - Quantum counters

Quantum Circuit Simulation Using QuIDDs

- **Motivation**
  - Need for a better way to simulate quantum circuits
- **Quantum Information Decision Diagram (QuIDD)**
  - Novel data representation that uses Binary Decision Diagrams (BDD) widely used in computer-aided circuit design
  - Captures some exponentially-sized matrices and vectors in a form that grows polynomially with the number of qubits
  - <u>Multiplies matrices and vectors in compressed form</u>
- **QuIDDPro Simulator**
  - Our QuIDD-based simulator implemented in C++
  - Experiments with Grover's algorithm demonstrate fast execution and low memory utilization

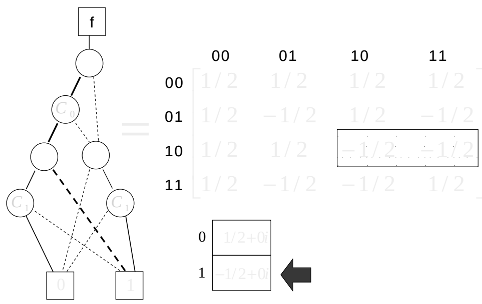
QuIDD Data Representation



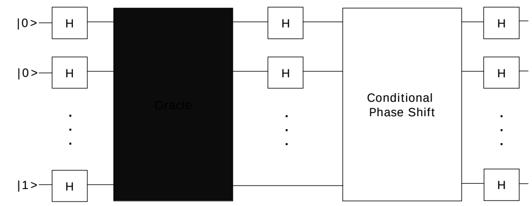
$$
= \begin{array}{c c}
 & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\
\begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} &
\begin{bmatrix}
1/2 & 1/2 & 1/2 & 1/2 \\
1/2 & -1/2 & 1/2 & -1/2 \\
1/2 & 1/2 & -1/2 & -1/2 \\
1/2 & -1/2 & -1/2 & 1/2
\end{bmatrix}
\end{array}
$$

0	$1/2+0i$
1	$-1/2+0i$

QuIDD Data Representation



QuIDDPro Simulation of Grover's Algorithm



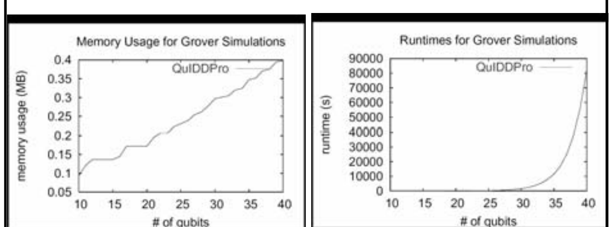
- Search for items in an unstructured database of *N* items
- Contains $n = \log N$ qubits and has runtime

QuIDDPro Simulation Results (Grover's search algorithm)

Oracle 1: Runtime (s)				
n	Oct	MAT	B++	QP
10	89.4	14.0	0.22	0.20
11	2.94e2	45.9	0.72	0.39
12	9.26e2	1.53e2	2.22	0.88
13	3.09e3	5.80e2	6.92	1.94
14	1.36e4	5.90e3	23.09	4.79
15	7.10e4	5.92e4	70.4	9.32
16	TIME-OUT	TIME-OUT	2.13e2	22.2
17	TIME-OUT	TIME-OUT	6.34e2	50.7
18	TIME-OUT	TIME-OUT	1.92e3	1.13e2
19	TIME-OUT	TIME-OUT	5.74e3	2.00e2
20	TIME-OUT	TIME-OUT	1.74e4	3.25e2

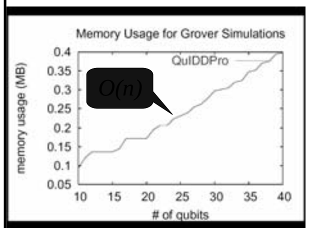
Oracle 1: Peak Memory Usage (MB)				
n	Oct	MAT	B++	QP
10	3.60e-2	2.00e-2	1.95e-2	0.211
11	6.80e-2	4.40e-2	7.03e-2	0.207
12	0.132	9.20e-2	7.42e-2	0.281
13	0.260	0.188	0.129	0.426
14	0.268	0.264	0.250	0.444
15	0.524	0.520	0.500	0.605
16	1.04	1.03	1.00	0.840
17	2.06	2.06	2.00	0.965
18	4.11	4.10	4.00	1.59
19	8.20	8.20	8.00	1.77
20	16.4	16.4	16.0	2.04

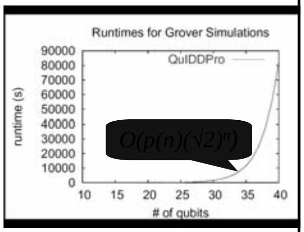
QuIDDPro Simulation of Grover's Algorithm



Same results for any oracle that distinguishes a unique element

QuIDDPro Simulation of Grover's Algorithm





Work in Progress: On The Power of Grover's Algorithm

- Database search with a black-box predicate *p(x)=1*
  - Classical evaluation of *p(x)* on one input (queries)
  - Quantum (parallel) evaluation of *p(x)* facilitates an implementation with fewer queries
- We also assume that *p(x)* is given as a BDD/QuIDD

Non-trivial assumption

  - BDDs are used to represent functions in practical CAD
  - However, a BDD is not really a black-box
  - BDD operations evaluate *p(x)* on multiple inputs at once (no quantum computation is involved)
- Grover on QuIDDs: same query complexity as in the quantum case
  - In practice this simulation is very fast and needs little memory

Quantum Circuit Synthesis

- Synthesis of classical circuits
  - Given a truth table, it is easy to find a circuit
  - Gate-count minimization is trickier, but doable by hand for circuits with several inputs
- Synthesis of *n*-input quantum circuits
  - Given a $2^n x 2^n$ matrix, can find a circuit (known algorithm)
  - Gate-count minimization doable by hand only for one input
  - For two inputs, optimal constructions are less than one year old, involve taking square roots of 4x4 matrices…

Two-qubit Computation with Minimum Resources

1. Some elementary gates have 2 inputs; our work allows to compare gate libraries
2. Most physical implementations of q. computers are currently restricted to 2 qubits
3. Circuits for quantum communication often have 2-3 inputs
4. Given a qantum circuit with >2 inputs, we can look for 2-input subcircuits and re-optimize those (peephole optimization)

Universal Elementary Gates [Barenco et.al. '95]

≠ "basic" gates

- Elementary one-qubit gates:

$$R_y(\theta) = \begin{pmatrix} \cos\theta/2 & \sin\theta/2 \\ -\sin\theta/2 & \cos\theta/2 \end{pmatrix} \quad 0 \le \theta < 2\pi$$

$$R_z(\alpha) = \begin{pmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{pmatrix} \quad 0 \le \alpha < 2\pi$$

- Elementary two-qubit gates: CNOT, conditioned on any line
- Barenco et al.: CNOT, $R_y(\theta)$ and $R_z(\alpha)$ are universal

Technology-Independent Synthesis

- Input: Unitary *4x4*-matrix ***M***
  - Generic quantum computation on 2 qubits
- Output: circuit in terms of elem. gates that implements ***M*** up to a phase
- Minimize: circuit cost
  - E.g., gate count or Σ (gate costs)
- Solutions exist iff the gate library is *universal*

Small Quantum Circuits

- What are the worst-case shortest quantum circuits up to phase?
- One-qubit computation: 3 gates required, suffice
- Technique: matrix decompositions

Phase can be ignored

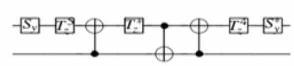
$$U = \begin{pmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{pmatrix} \begin{pmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{pmatrix} \begin{pmatrix} \cos\theta/2 & \sin\theta/2 \\ -\sin\theta/2 & \cos\theta/2 \end{pmatrix} \begin{pmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{pmatrix}$$

Gate 1 Gate 2 Gate 3

Example 5.3 Let $\mathcal{F}$ be the two-qubit Quantum Fourier Transform (QFT) [6]. It is given by the matrix

$$\mathcal{F} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

This operator is at the heart of Shor's factoring algorithm, and is one of the few operators whose implementation on a quantum computer is exponentially faster than any known classical counterpart [6]. Choosing the canonical decomposition appropriately and applying Theorem 3.1, one obtains the following circuit, in which $T_z = R_z(\pi/4)$.



Previous Work

- Proof of universality is constructive
  [Barenco et. al `95] in *Phys. Rev. A*
  - Can be interpreted as a synthesis algorithm
  - However, no attempt to minimize #gates
- Can be viewed as matrix factorization
  - [Cybenko `01]
  - ***M*=*QR*** with unitary ***Q*** & upper-triangular ***R***
    (***M*** unitary $\Rightarrow$ ***R*** diagonal)
  - We count gates, and the answer is 61

Our Results

- New synthesis procedures for 2 qubits
  - Can implement any operator in **18** gates or less, at most 3 of them are CNOTs
  - Lower bounds: sometimes **18** gates and **3** CNOTs are required
  - For a specific operator, we can tell when **0,1,2 or 3** CNOTs are required

Gate libraries	Lower and Upper Bounds			
	CNOT	Overall	CNOT	Overall
{CNOT, R_y, R_z}	3	18	3	18
{CNOT, R_y, R_x}	3	18	3	18
{CNOT, R_x, R_z}	3	**18**	3	**19**
{CNOT, R_x, R_y, R_z}	3	18	3	18
Basic gates	3	**9**	3	**10**

Table 1. Constructive upper bounds on gate counts for generic circuits using several gate libraries. Each bound given for controlled-not (CNOT) gates is compatible with the respective overall bound. These bounds are tighter than those from [2, 8] in all relevant cases. In particular, we never use input-independent rotation gates. Bounds that may potentially be tightened are shown in bold.

Our Work (2)

- Lower bounds
  - There exist two-qubit computations (*most of them*) that require at least ***17*** elementary gates
    - At least ***15*** non-const gates
    - At least ***2*** CNOTs
  - Bounds are not constructive and not tight, except for "15 input-dependent gates"
- We never use "temporary storage" qubits but that could lead to smaller gate counts

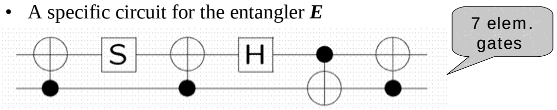
The Entangler and Disentangler

- "Computational basis"
  - **|00>**,**|01>**,**|10>** and **|11>**
- The "entangler" computation maps **|00>** to **(|00>+|11>)/√2,** etc.
- The "disentangler" is ***E***$^{-1}$***=E****
- Key lemma
  - If ***U=A⊗B***, then ***EUE**** has only real entries
  - **An efficient way to recognize tensor products**

$$E = \frac{\sqrt{2}}{2}\begin{pmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{pmatrix}$$

Circuits For *E* and *E**

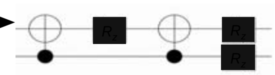
- A specific circuit for the entangler *E*



- ***S=diag(1,i)*** counts as one elementary gate
- The Hadamard gate ***H*** counts as two
- ***E**** is implemented by reversing the diagram
  - Change ***S*** to $S^{-1}=diag(1,-i)$

Our (Key) Synthesis Procedure

- The "canonical decomposition" for 2-qubit computations:
  - $\forall U \;\; \exists K_1, K_2$ and Δ such that $U=K_1\Delta K_2$
  - $E\Delta E^*$ is diagonal (5 gates)

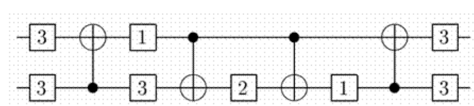


  - K_1,K_2 have only real entries
- The terms K_1, K_2 and Δ can be found explicitly
  - Numerical analysis: polar and spectral decompositions
- Reduce K_1 and K_2 to tensor products using entanglers
  - $EUE^*=\underline{E(A\otimes B)\cancel{E^*E}}\,\Delta\,\underline{\cancel{E^*E}(C\otimes D)E^*}$
  - ***A,B,C*** and ***D*** are one-qubit computations: 3 gates each
- Note that ***E*** and ***E**** are the same for any input

Details (1)

- After the initial "divide-and-conquer" many gate cancellations can be made
- This brings down max #gates to **28**
  - Only **15** of them depend on input, which matches an *a priori* lower bound
- Further reductions based on the analysis of $E(A\otimes B)E^*$ and $E(C\otimes D)E^*$
  - Max no. of gates reduced to **23**
  - However, **19** gates depend on the input

Details (2)



The structure of our generic 23-gate circuit

- For additional details, see
  - ***Physical Review A* 68(1), July 2003, 012318**
    quant-ph/0211002

Validation of Our Synthesis Algorithm

- Implementation in C++
  - We plan to put it up on the Web as an ASP
- Can capture structure
  - Several examples in quant-ph/0211002
  - Optimal results for any $\boldsymbol{A \otimes B}$ circuit (QR decomposition $\rightarrow$ typically **61** gates)
  - For 2-qubit Fourier transform: a circuit with minimal # of CNOT gates

Summary

algorithm	decomp.	# elem. gates	# CNOTs	# var 1-qubit gates
Cybenko 2000	*QR*	61	18	39
Our #1	u. *KAK*	23	4	19
Our #2	u. *KAK*	28	8	15 (sharp)
Our lower bounds		17	2	15

- First generic synthesis algorithm to capture circuit structure, e.g., $\boldsymbol{A \otimes B}$
- Recent work (1)
  - Lower and upper bounds of 18 gates (almost done)
  - Solved the synthesis of ***n***-qubit diagonal computations

  quant-ph/0303039 (asymptotically optimal circuits)

Recent Work (2)

Gate libraries	Lower and Upper Bounds			
	CNOT	Overall	CNOT	Overall
{CNOT, R_y, R_z}	3	18	3	18
{CNOT, R_y, R_x}	3	18	3	18
{CNOT, R_x, R_z}	3	**18**	3	**19**
{CNOT, R_x, R_y, R_z}	3	18	3	18
Basic gates	3	**9**	3	**10**

Table 1. Constructive upper bounds on gate counts for generic circuits using several gate libraries. Each bound given for controlled-not (CNOT) gates is compatible with the respective overall bound. These bounds are tighter than those from [2, 8] in all relevant cases. In particular, we never use input-independent rotation gates. Bounds that may potentially be tightened are shown in bold.

Circuit identities	Descriptions
$C_j^k C_j^k = 1$	CNOT-gate cancellation
$\chi^{jk}\chi^{jk} = 1$	SWAP-gate cancellation
$C_j^k C_k^j = \chi^{jk} C_j^k$	CNOT-pair elimination
$C_k^j R_x^j(\theta) = R_x^j(\theta) C_k^j,\ C_k^j S_x^j = S_x^j C_k^j$	move R_x, S_x via CNOT $\oplus$
$C_k^j R_z^k(\theta) = R_z^k(\theta) C_k^j,\ C_k^j S_z^k = S_z^k C_k^j$	move R_z, S_z via CNOT $\bullet$
$C_j^k \chi^{jk} = \chi^{jk} C_k^j$	move CNOT via SWAP
$V^j \chi^{jk} = \chi^{jk} V^k$	move 1-bit gate via SWAP
$R_n(\theta) R_n(\phi) = R_n(\theta + \phi)$	merging R_n gates
$\vec{n} \perp \vec{m} \implies S_n R_m(\theta) = R_{n \times m}(\theta) S_n$	changing axis of rotation

Table 2. Circuit identities used in our work.

Recent Work

- V. V. Shende, I. L. Markov and S. S. Bullock, ``On Universal Gate Libraries and Generic Minimal Two-qubit Circuits," quant-ph/0308033
- V. V. Shende, S. S. Bullock and I. L. Markov, ``Recognizing Small-Circuit Structure in Two-Qubit Operators," quant-ph/0308045
- George F. Viamontes, Igor L. Markov and John P. Hayes, ``Improving Gate-Level Simulation of Quantum Circuits," quant-ph/0309060