

Synthesis and Gate-Level Simulation of Quantum Circuits

University of Michigan

Igor Markov



Recent News and Quotes

- Up-coming DARPA BAA on quantum computing
 - \$\$ to work on *more detailed* proposals
 - Factor 128-bit numbers in 30 sec each
 - No constraints on algorithms or physical space used
 - Loose constraints on resources and development time
- Isaac Chuang (January):
 - “Ion traps are only 10x away from being scalable”
(in terms of error threshold)



Outline

- Q. Circuit Design Automation at Michigan
- Tasks in Design Automation
- Our work on simulation
 - Back-end: data structures and algorithms
 - Compiler front-end and the QuIDDPro distribution
 - A non-quantum application
- Automatic synthesis of quantum circuits
 - Problem formulations & relevant work by others
 - Our results
- Physical synthesis and parallels with VLSI design
- Conclusions



Q.Circuit Design Automation at Michigan

- **Faculty:** John Hayes & Igor Markov
- **Students:** George Viamontes, Smita Krishnaswamy and Vivek Shende
- **Former postdoc:** Ketan Patel
- **Ongoing collaboration:** with Stephen Bullock (NIST)
- **Our background:** computer architecture and algorithms for Electronic Design Automation
 - We have contributed to commercial CAD tools for classical circuit synthesis, layout and testing
 - Our students understand quantum circuits, algorithms, and can develop software



Tasks in Electronic Design Automation

- In non-quantum circuit design
 - Electrical, thermal and mechanical simulation (1960s +)
 - Logic simulation and test (1970s +)
 - Layout: routing (1960s +) and placement (1970s +)
 - Logic synthesis and technology mapping (1980s +)
 - Formal verification & physical verification (1990s +)
- For quantum circuits, same tasks make sense
 - Complexity is much higher
 - Fewer people have the right intuition
 - **Building meaningful, working quantum circuits will likely require extraordinary design automation**



Handling Circuit Complexity

- **Abstraction as an engineering approach**
(ignore inessential details, focus on 1st-order effects)
 - Example: ASIC layout & simulation with standard cells
 - Requires a bottom-up effort: standard-cell libraries
 - Abstraction often fails, requires refinement
- **Separation of concerns** (design flows & tool-chains)
 - Essentially, a divide-and-conquer approach
 - Example: logic design ignored layout issues in the past
 - Such separation often fails, requires merging, e.g., physical synthesis
- **High-performance computing**



Q. Circuit Simulation and Synthesis

- **(Quantum Logic) Simulation:** for a given circuit and a given input, find the output
 - Randomized simulation
 - Estimating the probabilities of outcomes
- **Synthesis:** input a $2^n \times 2^n$ matrix (explicitly or not) and find out an n -qubit circuit that implements it
 - Up to phase or up to a given measurement
 - Up to allowed errors
- **More generic quantum-mechanical simulations**
- **Aaronson: new link between simulation & synthesis**
 - Uses techniques from reversible circuits for q.simul.



Simulation with Data Compression

- **Quantum Information Decision Diagram (QuIDD)**
 - Data structure that can asymptotically compress the matrices and vectors used in quantum-mechanical apps
 - Algorithms for matrix-mult., tensor product, measurement, etc. manipulate the compressed QuIDD directly
- **QuIDDPro Simulator**
 - Implementation of the QuIDD data structure in C++
 - Facilitates high-performance simulation of density matrices, errors, state vectors, communication, etc.
 - Surprising empirical results on Grover's algorithm

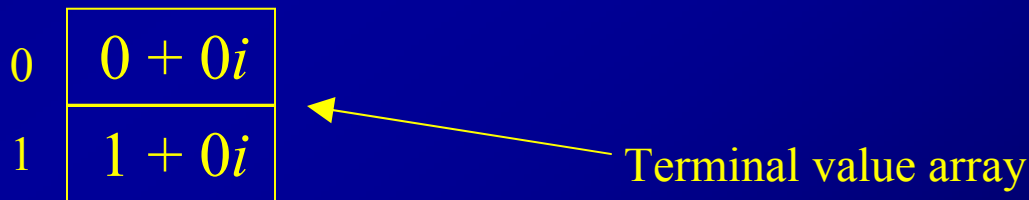
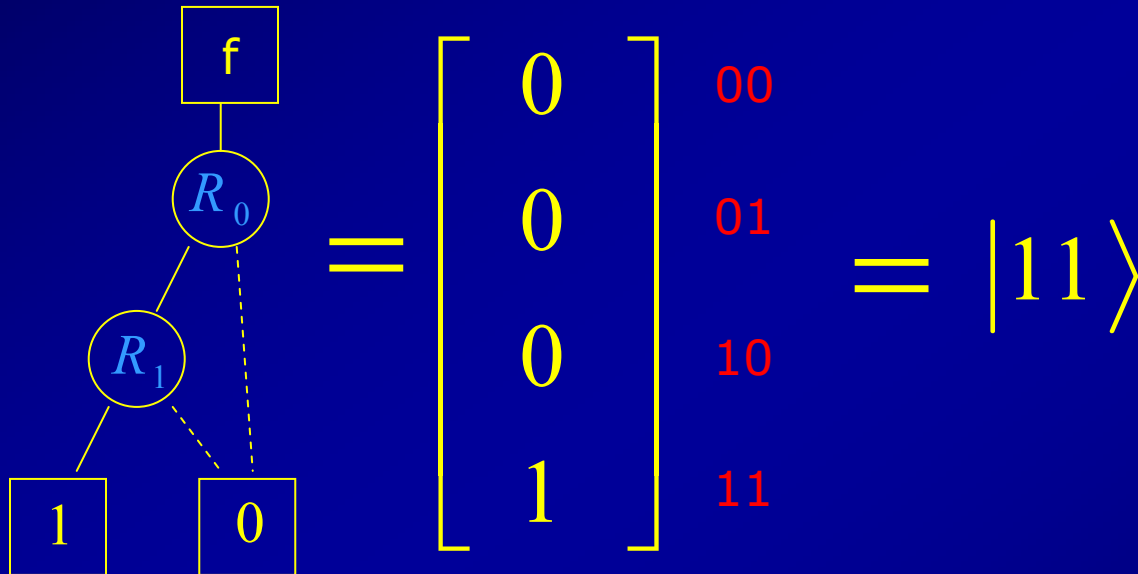


Redundancy in Quantum Mechanics

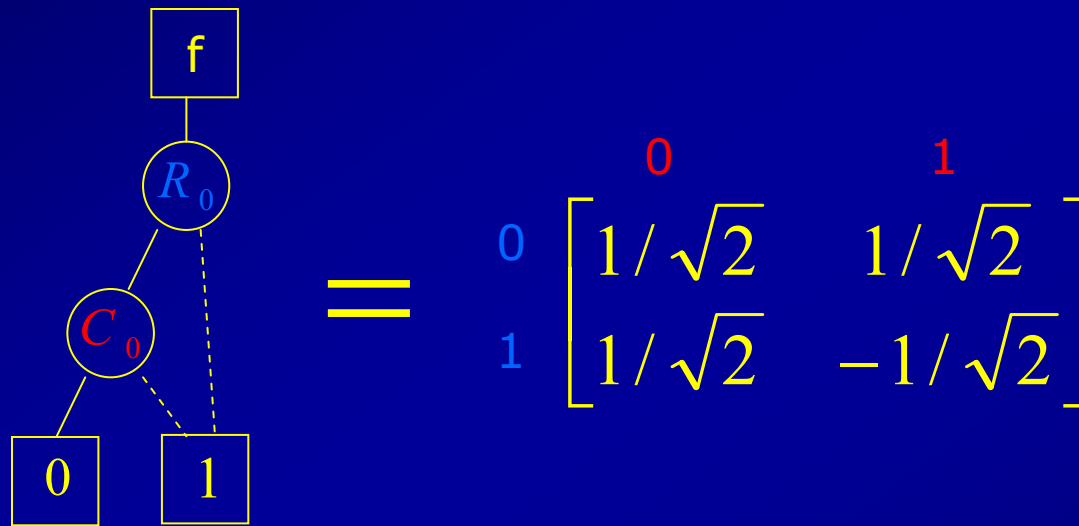
$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$$



QuIDD Vector Example



QuIDD Matrix Example 2x2

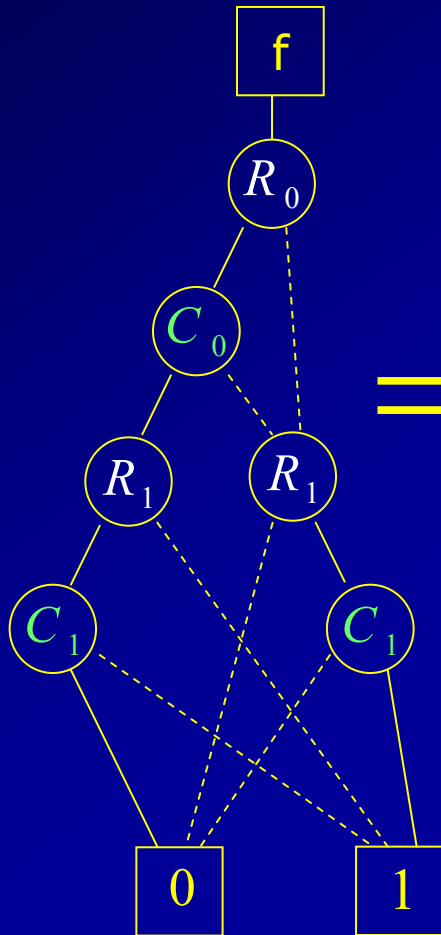


0	$-1/\sqrt{2}$
1	$1/\sqrt{2}$

1-Qubit Hadamard
Operator



QuIDD Matrix Example 4x4



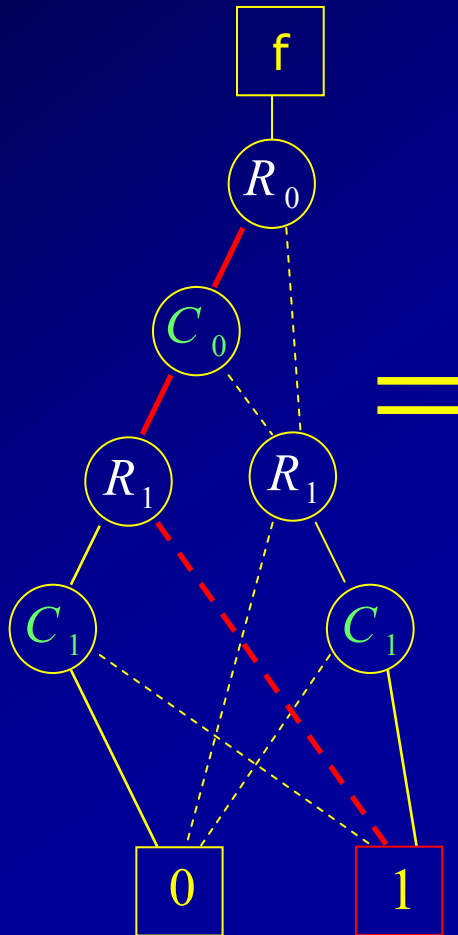
=

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

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



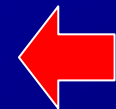
QuIDD Matrix Example 4x4



=

	00	01	10	11
00	1/2	1/2	1/2	1/2
01	1/2	-1/2	1/2	-1/2
10	1/2	1/2	-1/2	-1/2
11	1/2	-1/2	-1/2	1/2

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



QuIDD Operations

- Operations are performed on compressed matrices and vectors w/o decompressing them
- Graph traversal algorithms based on *Apply*
 - Inner/outer product, scalar operations
 - Tensor product, matrix multiplication, matrix addition
 - Measurement, partial trace, etc
- Runtime and memory usage of all operations depend on the size of QuIDD operands (i.e. # of nodes)
 - Polynomial-sized QuIDDs lead to polynomial runtime and memory in simulations



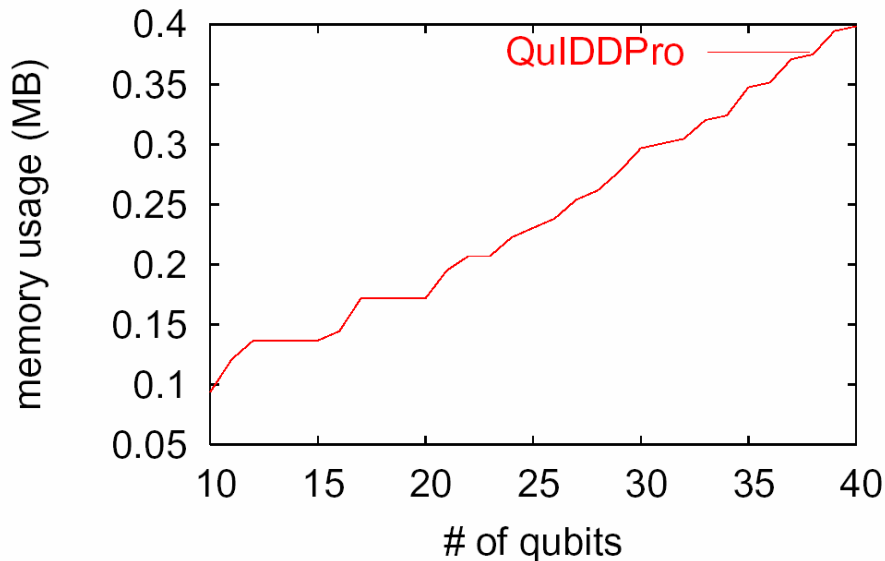
Useful Polynomial-Sized QuIDDs

- Arbitrary superpositions of several basis states (linear in n)
 - Equal superpositions of all basis states except for several
 - n -qubit Pauli operators
 - n -qubit Hadamard operators
 - Conditional phase-shift operator
 - Oracle originally considered by Grover
 - *And others*
-
- QuIDDs for QFT exhibit exponential scaling with current data structures

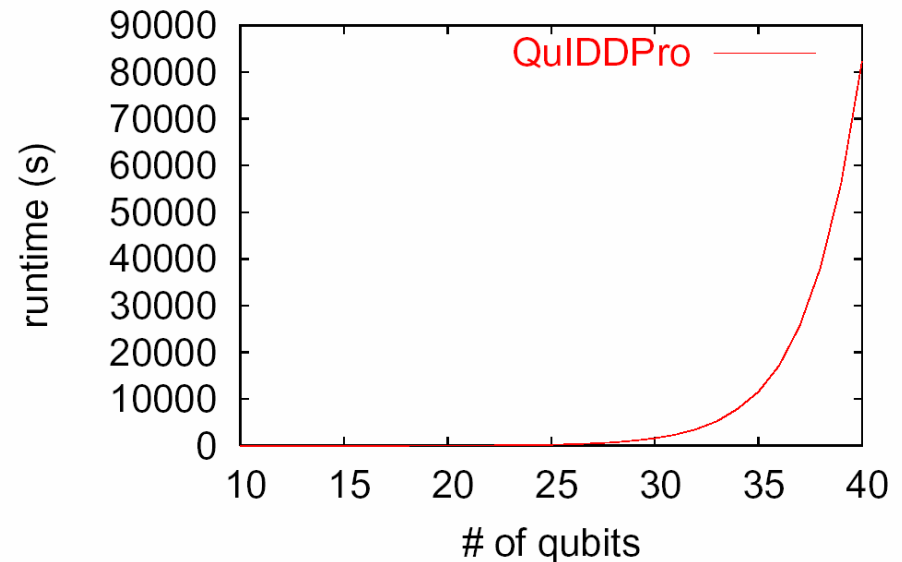


QuIDDDPro Simulation of Grover's Algorithm

Memory Usage for Grover Simulations



Runtimes for Grover Simulations

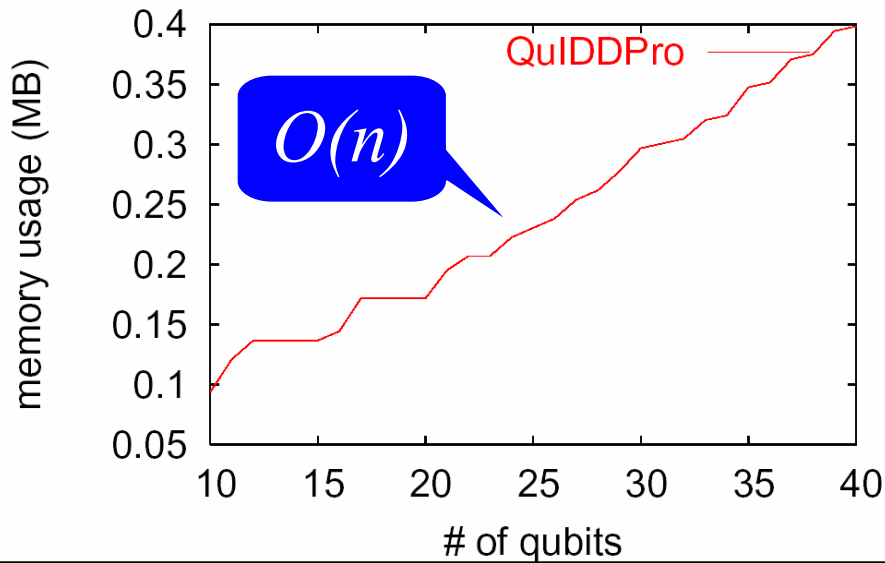


Same results for any oracle that distinguishes a unique element

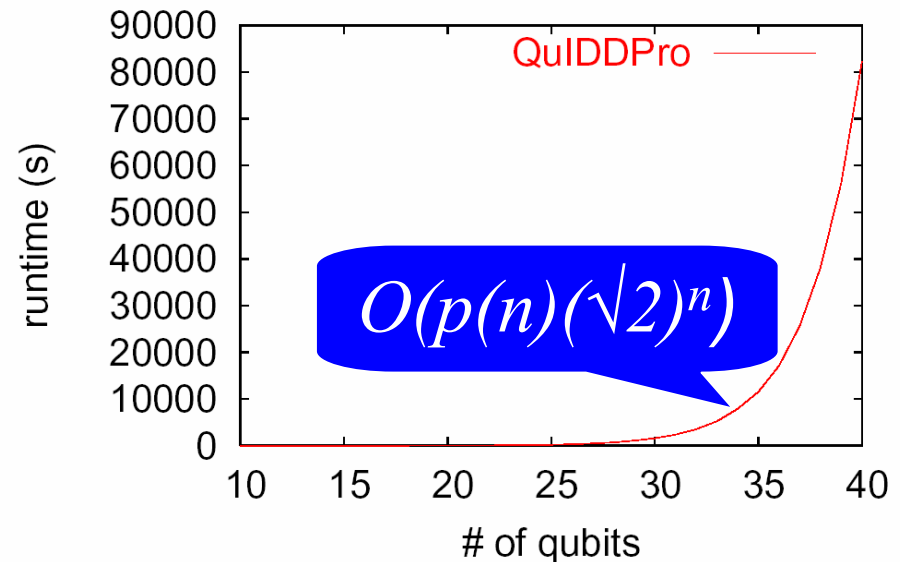


QuIDDDPro Simulation of Grover's Algorithm

Memory Usage for Grover Simulations



Runtimes for Grover Simulations



Same results for any oracle that distinguishes a unique element



Density-Matrix Simulations

- Even for pure (noiseless) states, density matrices lead to a huge increase in memory usage
 - Take a **16-qubit** state
 - State-vector representation: 2^{16} complex amplitudes, 16 bytes per amplitude (two doubles) → **1GB RAM**
 - Density-matrix representation: 2^{32} complex amplitudes → **64TB RAM**
- Sample QuIDDPro/D simulation
 - **16-qubit** reversible full adder (w Toffoli gates), **24 gates**
 - runtime: **0.44 sec**, peak memory: **62.5 KB**
- More empirical data in [quant-ph/0403114](https://github.com/quant-ph/0403114)

Cannot use
Gottesman-Knill



QuIDDPro Simulator

- Front-end ASCII interface: interactive & batch modes
- Input language similar to Matlab
 - Built-in support for popular gates and operations
 - Key control constructs such as looping and conditional execution
- **Support for state-vector simulations and density-matrices**
- Implementation details: C++, CUDD, GMP, flex, bison
- Future extensions to the input language
 - User-defined functions
 - Technology-specific operators
 - Dirac notation?
- Development of new data structures
- User's guide



Recent References

- * G. F. Viamontes, I. L. Markov and J. P. Hayes, ``Improving Gate-Level Simulation of Quantum Circuits,’’ [quant-ph/0309060](https://arxiv.org/abs/quant-ph/0309060), to appear in *Quantum Information Processing* (shorter version in *Proc. Design Automation & Test in Europe*, Paris, France, February 2004)
- * G. F. Viamontes, I. L. Markov and J. P. Hayes, ``Graph-based Simulation of Quantum Computation in the State-vector and Density-matrix Representation,’’ [quant-ph/0403114](https://arxiv.org/abs/quant-ph/0403114) to appear in *Proc. SPIE Conf. on Quantum Information and Computation*, Orlando, Florida, April 2004.

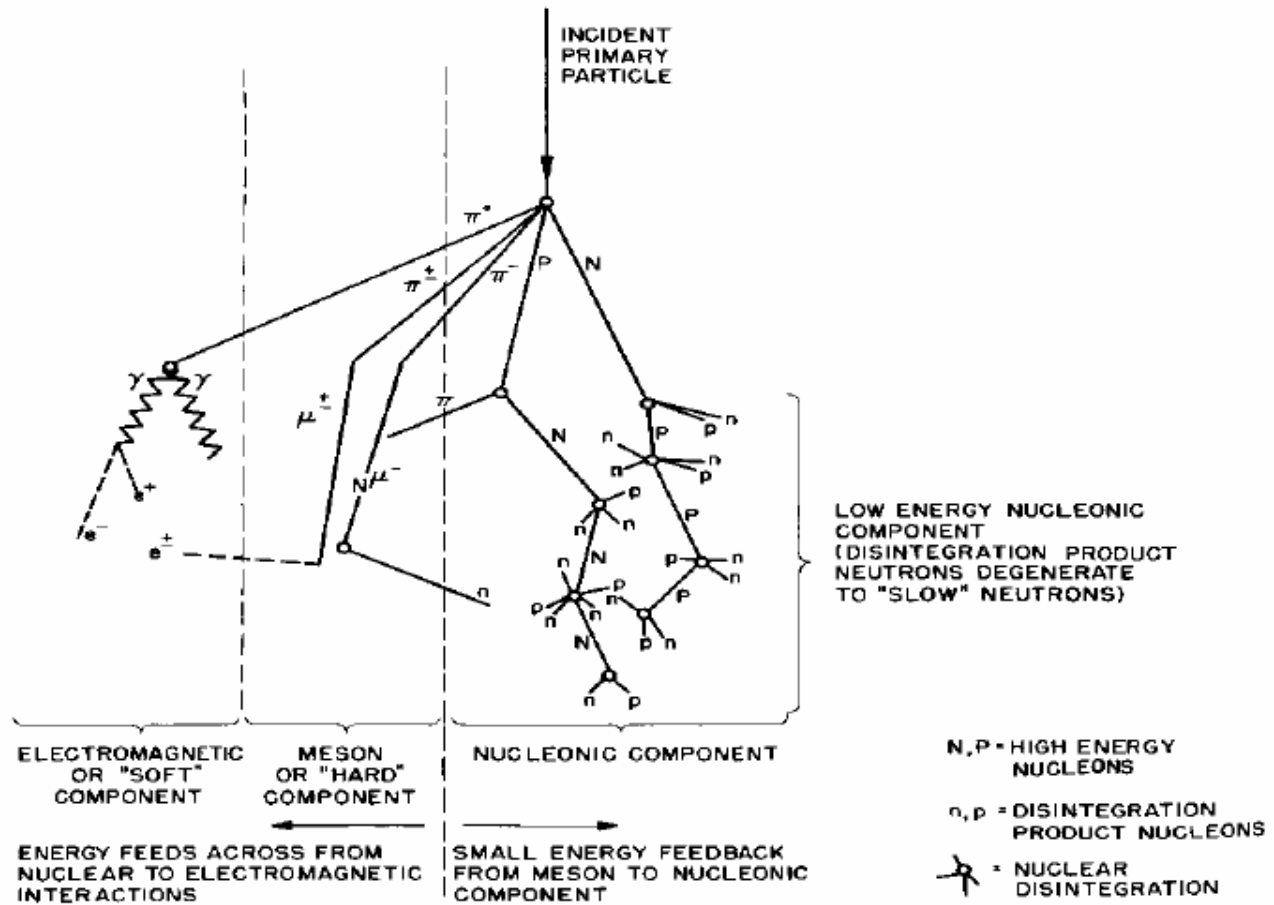


A Non-Quantum Application: Modeling Soft Errors in VLSI Circuits

- Transient faults
 - Mostly caused by neutron hits
 - Neutrons generated by cosmic particles in atmosphere
 - Became noticeable as transistors became smaller
- Incidence varies significantly with altitude
- Much more pronounced in aerospace applications
 - Especially during solar flares
- Folklore:
 - Sun Micro didn't turn on ECC on L2 cache by mistake
 - Lost customers

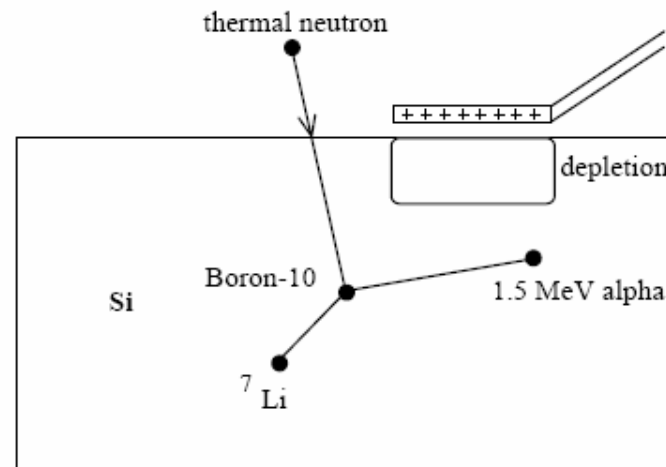
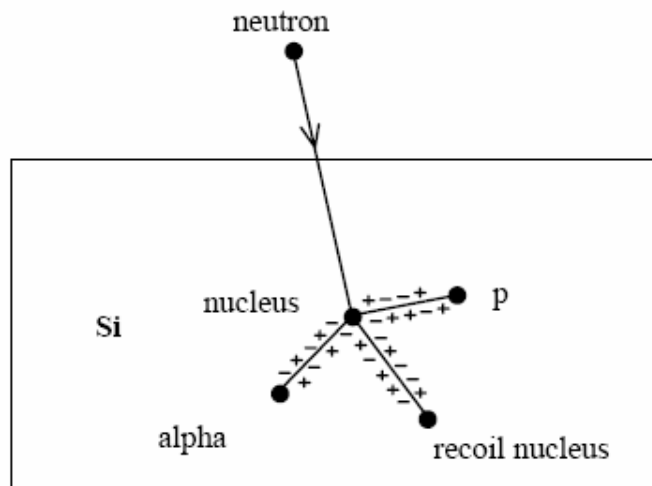


Where Neutrons Come From



Schematic Diagram of Cosmic Ray Shower

How Neutrons Can Generate Charged Particles in any IC

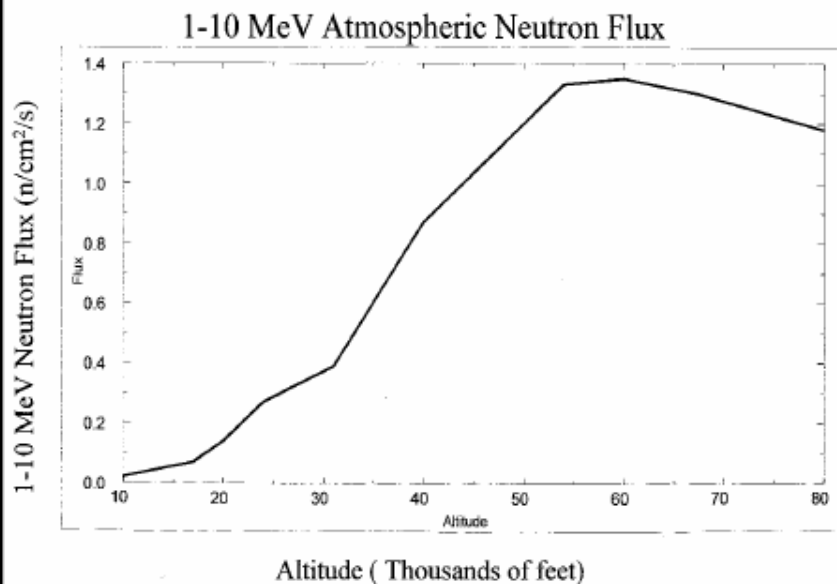


note: These alpha and other charged particles are being generated right in the silicon itself. Unlike packaging induced alphas, they do not have to penetrate the top metalization. They can be generated right where they can do the most harm.

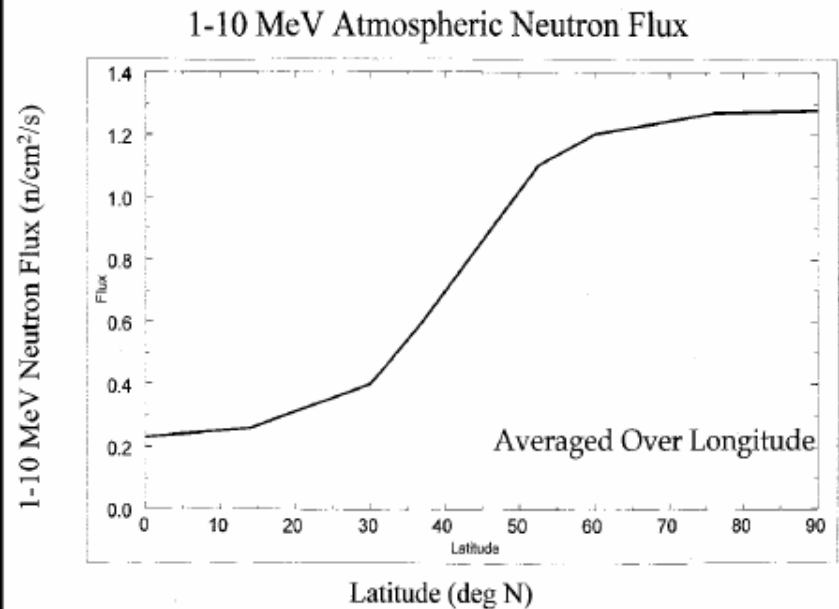
How Neutron Flux Varies

with altitude
with latitude

Neutron Models: Flux vs. Altitude



Neutron Model: Flux vs. Latitude



What NSEU testing was done by Xilinx

- **Proton Cross Sections**
 - Taken at Crocker (Davis) and Texas A&M
 - Correlation with neutron data was disappointing
- **Neutron Cross Sections**
 - Taken at the LANSCE facility at Los Alamos
 - Evaluating contribution of energy spectrum models
- **Atmospheric Neutron Testing (Rosetta)**
 - Large population of parts
 - Tested at three altitudes
 - Correlated with LANSCE results

Simulating Transient Faults

- Fault models used in CAD tools today are geared towards manufacturing defects
 - Need inherently probabilistic fault models
- Example: probabilistic AND gate

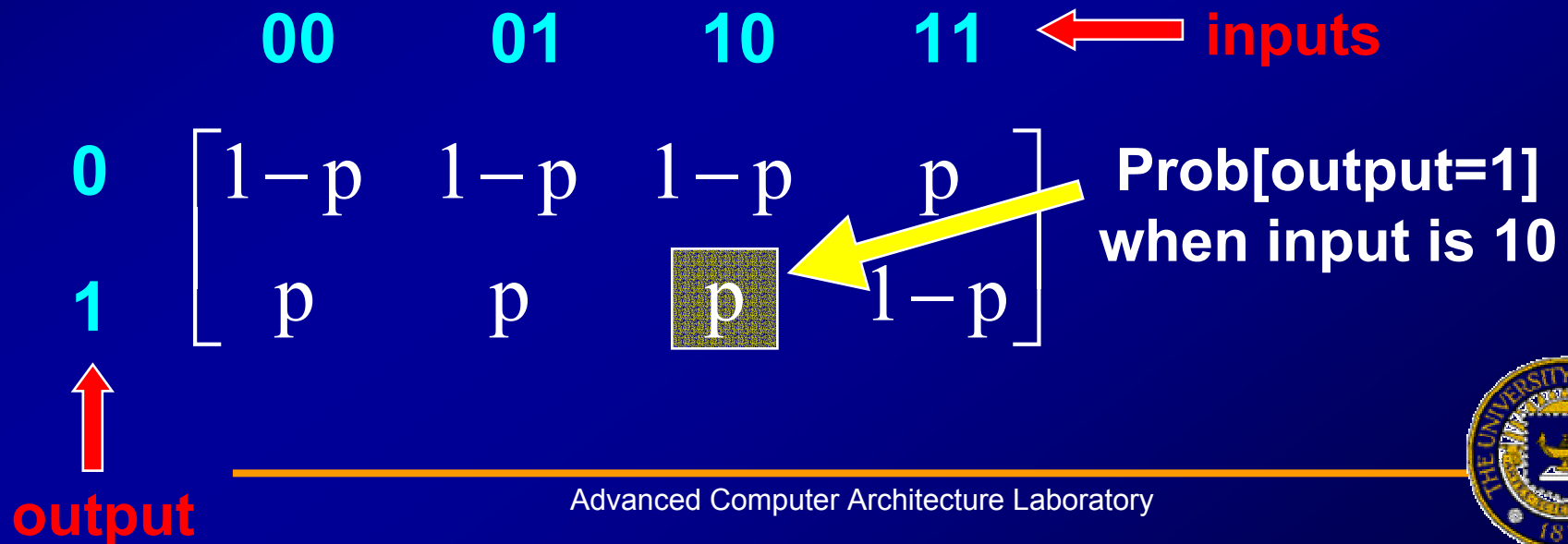


- Faultiness of a circuit depends on circuit structure and gate fault models
 - Wire faults can be modeled by fake faulty buffers



Probabilistic Transfer Matrix (PTM)

- Row indices represent outputs values
- Column indices represent inputs values
- Matrix elements capture pairwise transition probabilities



Estimating Reliability Based on PTM

circuit error probability = $\sum_i p_i \cdot \sum_{j: C(j,i)=0} P_{[j,i]}$

Annotations:

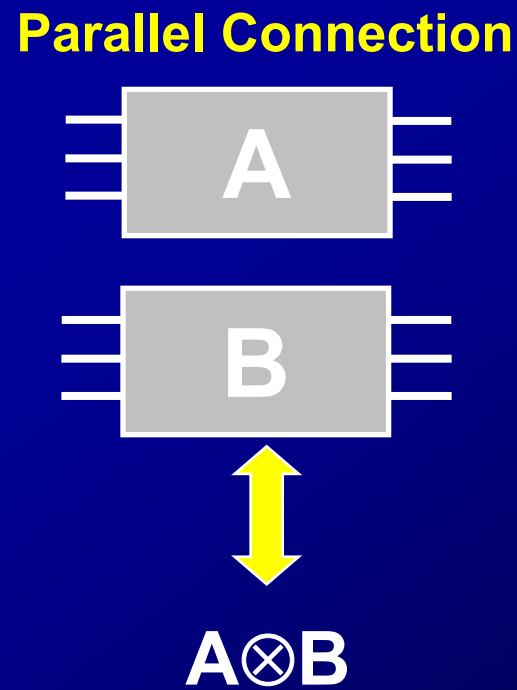
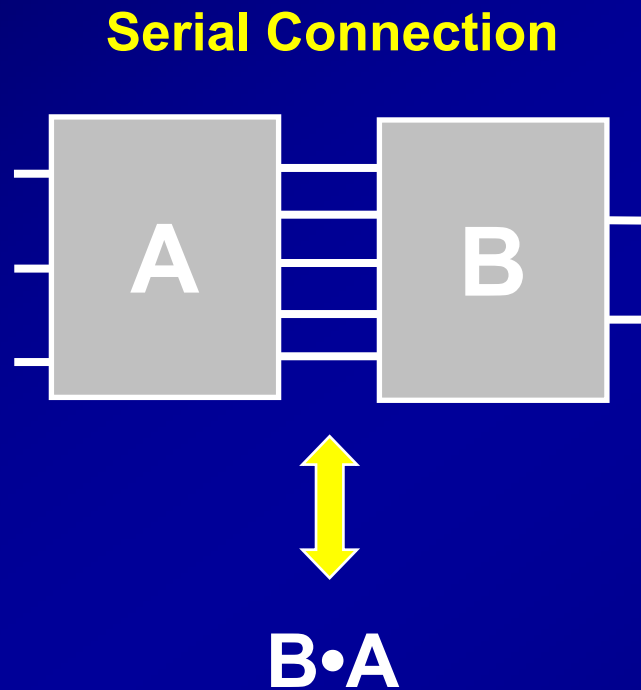
- circuit input probabilities** (points to p_i)
- (j,i)-th entry of probabilistic transfer matrix** (points to $P_{[j,i]}$)
- ideal transfer matrix (i.e., no errors)** (points to $C(j,i)=0$)

- PTM for a circuit can be computed from PTM of its gates
 - three basic methods of gate composition: serial, parallel and fanout



Computing Circuit PTMs

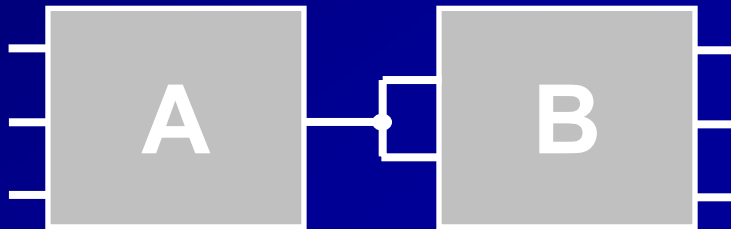
- Similar to simulating quantum circuits



Computing Circuit PTMs

- Seems harder than simulating quantum circuits

Fanout



$B' \cdot A$

B' is the PTM of B with columns corresponding to invalid inputs removed

- This is easy with QuIDDs



Reference

- * K.N.Patel, I.L.Markov and J.P.Hayes, “Evaluating Circuit Reliability Under Probabilistic Gate-Level Fault Models ,” *Intl. Workshop on Logic and Synthesis*, 2003



Quantum Logic Synthesis

- Assume a gate library (CNOTs, one-qubit ops)
- Input a $2^n \times 2^n$ matrix U and find an n -qubit circuit
 - Typically U is implemented up to phase
 - We show how to implement up to a given measurement
- Difficult problem even for small n , even when U is given explicitly (matrix elements)
- Otherwise U may be given by a circuit which should be simplified
 - We can extract sub-circuits and simplify them



Recent Results in Synthesis (1)

- **2-qubit case**: motivated by existing physical implementations, peephole optimization, quantum communication
- Two 2-qubit operators U, V are equivalent up to local unitaries iff $\chi(U\varepsilon U^t\varepsilon) = \chi(\pm V\varepsilon V^t\varepsilon)$ where χ denotes the characteristic polynomial and $\varepsilon = \sigma^y \otimes \sigma^y$
- Related techniques allow finding one-qubit operators and simplify the synthesis problem by normalization



Recent Results in Synthesis (2)

- If CNOT is the only multi-qubit gate in library, then **3 CNOTs are required** for most 2-qubit ops
 - 3 CNOTs are also sufficient
 - Additionally, at least 6 basic gates are required (6+3)
 - 7 additional basic gates are sufficient (7+3)
 - 15 additional one-parameter gates necessary & sufficient
- **B gate** from Berkeley
 - A replacement for CNOT
 - Two B gates are sufficient for any 2-qubit operator (no asymptotic improvement though)



Recent Results in Synthesis (3)

- Some n -qubit ops require $(4^n - 3n - 1)/4$ CNOTs
 - For $n = 2$, this yields 3
 - Reference: [quant-ph/0308033v2](https://arxiv.org/abs/quant-ph/0308033v2)
- This lower bound was matched asymptotically by Vartiainen et al with a synthesis algorithm
 - Remaining gap is $\sim 70x$
- Note that this applies to worst-case optimality
- In particular, for diagonal operators $\Theta(2^n)$ gates are necessary and sufficient
 - Reference: [quant-ph/0303039](https://arxiv.org/abs/quant-ph/0303039)



2-Qubit Universal Circuits

- The following circuits can implement an arbitrary 2-qubit op; They use 15 elementary gates and 3 CNOT gates

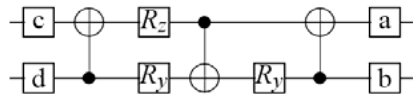


FIG. 1: A universal two-qubit circuit with **three** CNOT gates. It requires **10** basic gates [3] or **18** gates from $\{\text{CNOT}, R_y, R_z\}$.

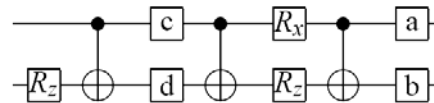


FIG. 2: Another universal two-qubit circuit with **three** CNOT gates. It requires **10** basic gates [3] or **18** gates from $\{\text{CNOT}, R_x, R_z\}$.

CNOT Counting

- For $U \in SU(4)$ an arbitrary 2-qubit operator, we found closed-form tests for the number of required CNOT gates
- Then U can be implemented with
 - 0 CNOT gates iff $\chi(U\epsilon U^t\epsilon) = (x \pm 1)^4$
 - 1 CNOT gate iff $\chi(U\epsilon U^t\epsilon) = (x+i)^2(x-i)^2$
 - 2 CNOT gates iff $\text{tr}(U\epsilon U^t\epsilon)$ is real
($\epsilon = \sigma^y \otimes \sigma^y$, χ is the characteristic polynomial)
- Reference: [quant-ph/0308045v3](https://arxiv.org/abs/quant-ph/0308045v3)



Technology-mapping for CNOT gates

- Given a 2-qubit Hamiltonian H , find τ such that $\exp(i\tau H)$ is equivalent to CNOT up to one-qubit operators
- Set $\chi(e^{iH\tau}\epsilon(e^{iH\tau})^t\epsilon) = (x+i)^2(x-i)^2$
 - Numerically solve for τ
 - Solutions may or may not exist for specific H
- This may help implementing CNOT for new technologies
- Open question: is there an analytical solution?



Synthesis with Measurement

- Quantum computations are typically followed by a known measurement
 - Some information is erased
 - Additional flexibility during logic synthesis
- **We propose a new synthesis problem**
 - **Fix a projective measurement** (space decomposition) to be performed at the end of computation
 - **Consider two operators equivalent iff** for all inputs, probabilities of outcomes are the same



Synthesis with Measurement

- Suppose all subspaces are spanned by computational basis vectors
- Then at most two CNOTs are required to implement an arbitrary 2-qubit operator
- 2 CNOTs are necessary and sufficient if
 - No subspace is 3-dim and the subspaces are not $\text{span}(|0\rangle, |3\rangle)$ or $\text{span}(|1\rangle, |2\rangle)$
- 1 CNOT is necessary and sufficient if one of the subspaces is 3-dimensional
- Reference: [quant-ph/0401162](https://arxiv.org/abs/quant-ph/0401162)



Recent References

- [1] S. S. Bullock and I. L. Markov: “An Arbitrary Two-Qubit Computation in 23 Elementary Gates,” [PRA 68 012318](#) (also in *Proc. ACM/IEE Design Automation Conference 2003*)
- [2] S. S. Bullock and I. L. Markov: “Asymptotically Optimal Circuits for Diagonal Computations”, [quant-ph/0303039](#) *Quantum Information and Computation*, vol. 4, no. 1, January 2004, pp. 27-47
- [3] V. Shende, I. L. Markov and S. S. Bullock: “Minimal Universal Two-qubit Quantum Circuits,” [quant-ph/0308033v2](#), to appear in PRA (also in *Proc. ACM/IEEE Design Automation and Test in Europe 2004*)
- [4] V. Shende, S. S. Bullock and I. L. Markov: “Recognizing Small-Circuit Structure in Two-Qubit Operators, ” [quant-ph/0308045v3](#)
- [5] V. Shende and I. L. Markov, “Measurement Saves CNOT Gates in Optimal 2-Qubit Circuits”, [quant-ph/0401162](#)



None of Those Results Account for Layout!

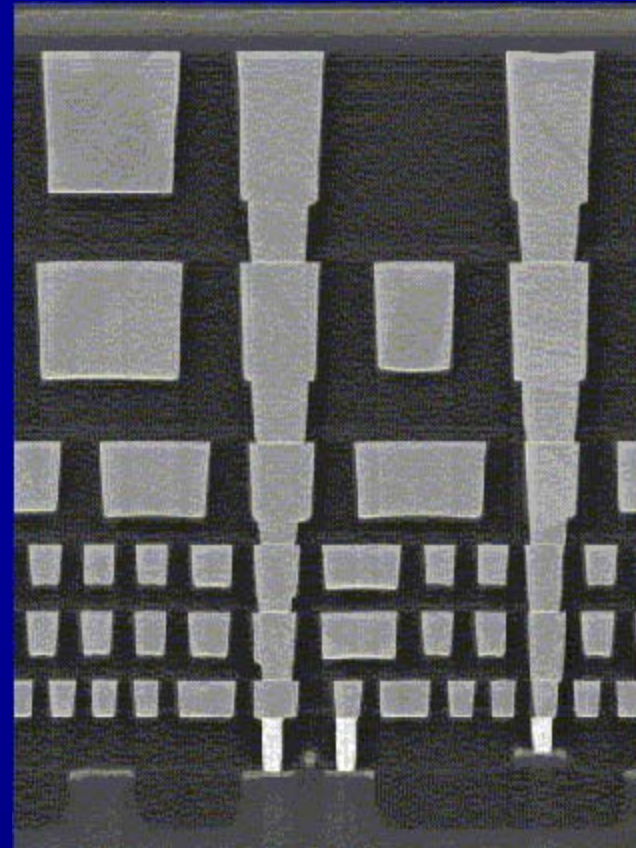
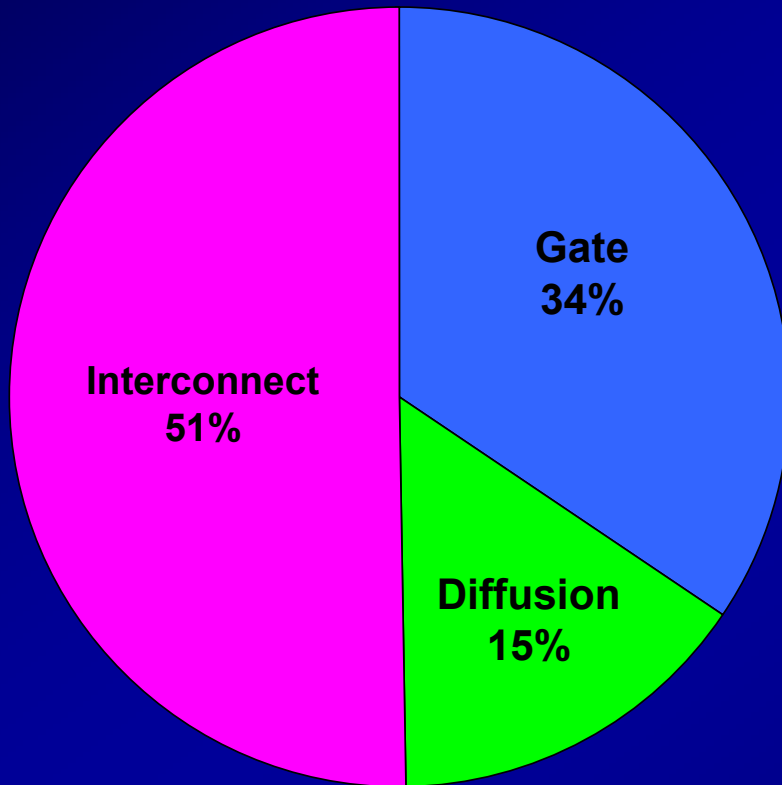
- Layout is technology-dependent
(synthesis is just a cosy abstraction)
- Layout tends to limit scalability
 - Interactions often limited to nearest neighbors
 - Related considerations in VLSI w.r.t. buffer insertion
- Earlier, synthesis of non-quantum VLSI circuits has been performed w/o accounting for layout
 - Circuit delay was measured in #gates on critical paths, assuming zero delay in wires
 - However, due to the scaling of R s and C s wires now account for $>80\%$ of circuit delay



Total Dynamic Power Breakdown (Intel Centrino)

Global clock included

Source: *Intel*, Feb 2004



Physical Synthesis in VLSI

- To synthesize commercially-feasible VLSI logic, one must compute circuit delays correctly
 - Wire delays depend on distances between gates
 - To find gate locations, one must perform placement
 - ... but placement is performed after synthesis!
- Synthesis and placement must be simultaneous!
 - Such CAD tools appeared in the last 3 years
 - Mostly based on independently developed synthesis and placement programs



Why Physical Synthesis is Difficult

- Classical synthesis can be performed w/o technology details, to some extent
 - “Technology mapping” is the interface to real world
- Accounting for layout in synthesis requires a lot more technology information
 - This breaks abstraction
 - However, existing algorithms can be extended (finding good placements is still a part of the problem)
- Quantum Physical Synthesis is justified
 - Layout is more critical for quantum circuits
 - More difficult too



Toward Quantum Physical Synthesis(1)

- Where reasonable, follow VLSI design principles
- For a given technology, regularize layout (λ -grid)
- **Try to describe all geometry with rectangles**
 - May require moving parts, paths of motion
- Define design rules: minimum spacing, etc
 - Dynamic constraints?
- **Define standard cells:** hide difficult physical effects
 - This is very likely to cost performance!



Toward Quantum Physical Synthesis(2)

- Formally map qubits into geometry (grid-aligned rectangles)
- **Formulate placement and routing**
 - Study complexity (classical VLSI layout is harder than number-factoring, but that's okay)
 - Come up with algorithms for irregular quantum circuits (those may include QECCs)
- **Define new objective functions for synthesis**



Conclusions

- Working with non-trivial quantum circuits will soon require circuit design automation
 - Decent progress in simulation
(in the worst case, use parallel computers)
 - Decent progress in automatic synthesis
 - Layout problems are not even formalized yet
- Layout promises to be a major obstacle to the scalability of quantum circuits
 - Some experience from VLSI design may apply
 - Need work on quantum circuit layout

