Challenges in Synthesizing Quantum Computers

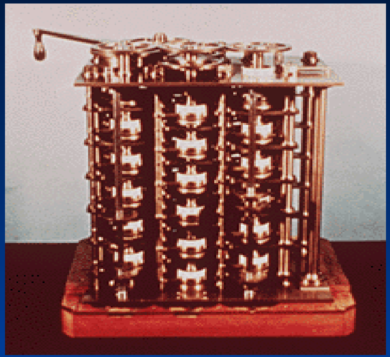
Susmita Sur-Kolay

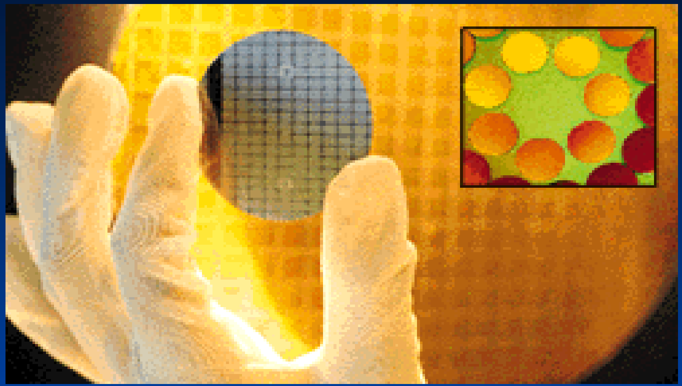
Advanced Computing and Microelectronics Unit
Indian Statistical Institute, Kolkata
http://www.isical.ac.in/~ssk

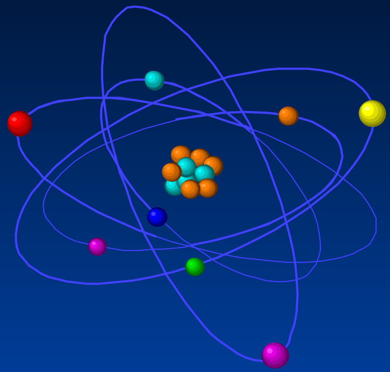
Agenda

- News of the day: 2012 Physics Nobel Prize!
- What is a Quantum Computer?
- Power of Quantum Computers
- Synthesis of QCs
  - Minterm based
  - Fredkin gate based Programmable Quantum Logic block
  - Nearest Neighbour constraints
  - Fault models and Error
- Concluding Remarks

Major drivers of QC







- Computers are getting smaller (and faster) and reaching a point where "classical" physics is no longer a sufficient model for the laws of physics – 32nm chips expected next year!
  - Classical laws for resistors, capacitors also change below 50nm
- Quantum mechanical laws can speed up computational processes by massive parallelism
- Reversible logic can beat switching power dissipation crisis in nanometer Ics
- Key difference with classical architecture: storage & processing

What is a Quantum Computer?

A **quantum computer** is any device for computation that makes *direct use* of distinctively quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data.

Are not present-day computers quantum?

No! Difference lies in storage and processing

Computation & Physical Systems

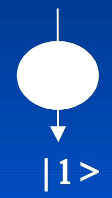
■ Computer	→	■ Physical System
■ Input	→	■ Initial State (Preparation)
■ Rules	→	■ Physical Laws (Laws of Motion)
■ Output	→	■ Final State (Observation with measuring instruments)

Description of a Quantum Physical System

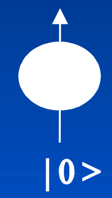
- A State, of which our knowledge may be incomplete.
- A Set of Observables --- features that can be measured.
- Each observable has a spectrum specific to a universe.
- This spectrum is discrete.
- An observable may have different values in different universes in which it is being measured.
- Measurement of observable and their spectra are invariant features of a quantum system.

Qubits

- A qubit, also called qbit, is the basic unit of information in a quantum computer
- Physically may be
  - spin state of an electron may be up (↑) or (↓) down
  - polarization state of a photon may be vertical (↨) or horizontal (↔)
  - Two quantum mechanical states represented by standard quantum mechanics *ket* notation | *0* > and | *1* >



A qubit in the | *1* > state



A qubit in the | *0* > state

- A quantum state |Ψ> can be superposed : $|\Psi\rangle = a\,|0\rangle + b\,|1\rangle$
  - where *a* and *b* are the complex amplitudes representing the probabilities of state |*0* > and |*1* > and $|a|^2 + |b|^2 = 1$

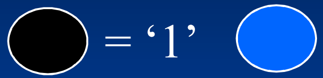
- In matrix notation a qbit is denoted by:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Computation with Qubits

Classical Computation

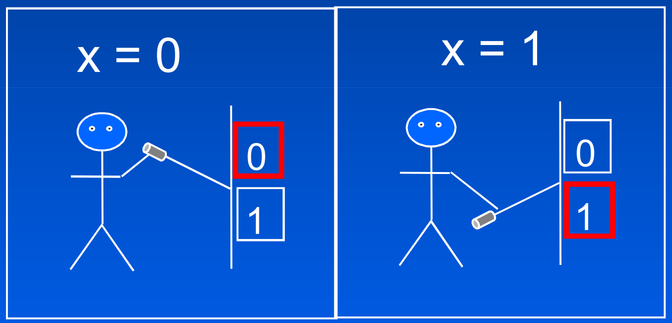
Data unit: bit



$= \text{'1'}$ $= \text{'0'}$

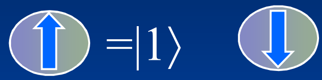
Valid states:

x = ‘0’ or ‘1’



Quantum Computation

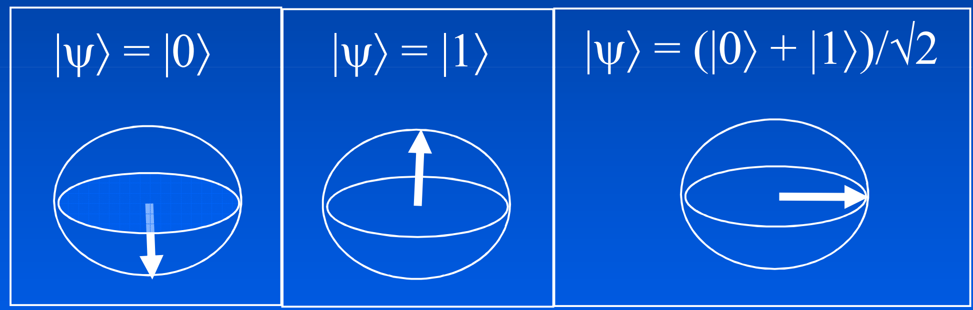
Data unit: qubit



$=|1\rangle$ $=|0\rangle$

Valid states:

$$|\psi\rangle = c_1|0\rangle + c_2|1\rangle$$



How do qubits affect computation?

State	Result: deterministic measurement
x = ‘0’	‘0’
x = ‘1’	‘1’

State	Result: Probabilistic measurement
$\vert\psi\rangle = \vert 0\rangle$	‘0’
$\vert\psi\rangle = \vert 1\rangle$	‘1’
$\vert\psi\rangle = \frac{\vert 0\rangle + \vert 1\rangle}{\sqrt{2}}$	‘0’ 50% ‘1’ 50%

State of a Quantum Computer

- The maximum possible number of states depends on the number of qubits --- n qubits represent 2^n states.

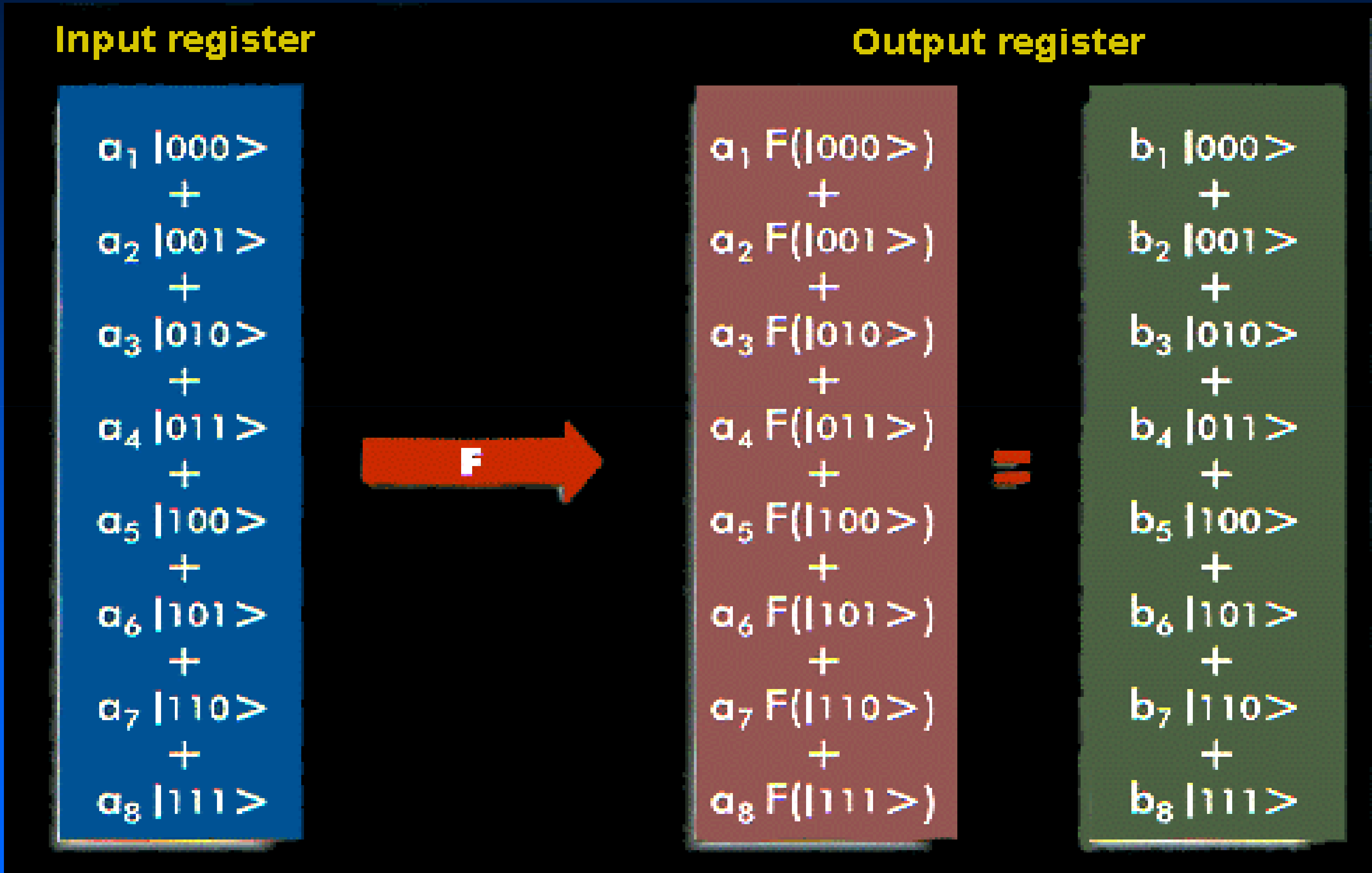
- A 2-qubit vector can simultaneously represent the states 00, 01, 10, 11 and the probability of their occurrence depends on the complex amplitude values.

- 2-qubit State vector (superposition):
  - $\psi = C_0 \, |00\rangle + C_1 \, |01\rangle + C_2 \, |10\rangle + C_3 \, |11\rangle$

  where $\Sigma_i \, |C_i|^2 = 1$

- Hence comes the concept of *quantum register* of M qubits holding 2^M simultaneous values.
  - If we perform an operation on the register, all the values are being operated on simultaneously which leads to quantum parallelism.

A Quantum Register



Nature of Quantum Operations

Any **linear operation** that takes a quantum state $\alpha_0|0\rangle + \alpha_1|1\rangle$

satisfying $|\alpha_0|^2 + |\alpha_1|^2 = 1$ nd maps to a state $\alpha_0^{'}|0\rangle + \alpha_1^{'}|1\rangle$

satisfying $|\alpha_0^{'}|^2 + |\alpha_1^{'}|^2 = 1$ must be UNITARY

$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix}$$

is unitary if and only if

$$UU^{\dagger} = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix} \begin{bmatrix} u^{*}_{00} & u^{*}_{10} \\ u^{*}_{01} & u^{*}_{11} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Quantum Operators and Quantum Gates

- A quantum logic gate means a closed-system evolution of the n-qubit state space H_n
  - *no* information is gained or lost during this evolution
- If | Ψ > is a state vector in H_n , the logic operation can be represented by
  - $| \Psi > \rightarrow U | \Psi >$ for some unitary $2^n \times 2^n$ matrix U
- Any manipulations on qubits, have to be performed by unitary operations
- A quantum network consists of quantum logic gates whose computational steps are synchronized in time
- A new state can be obtained by operating a unitary operator |U> on the vector space |Ψ>

Single Qubit Gates

NOT Gate ¬

Represented by the matrix U_{NOT}: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$\neg |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \; |1\rangle$

$\neg |1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} . \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$

Hadamard operator defined by the unitary matrix H= $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

H |0> = 1/√2 (|0> + |1>)
H |1> = 1/√2 (|0> - |1>)

Hadamard operator produces a superposition of two qbits

Phase Gate defined by the unitary matrix

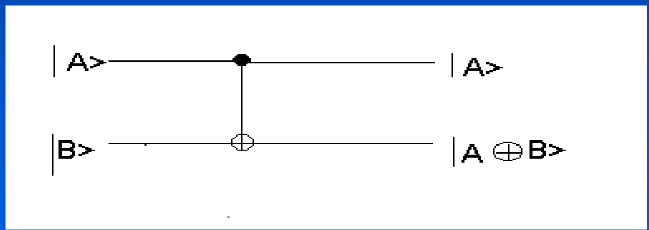
$$U = \begin{bmatrix} e^{i(\alpha - \frac{\beta}{2} - \frac{\delta}{2})} \cos\frac{\gamma}{2} & -e^{i(\alpha - \frac{\beta}{2} + \frac{\delta}{2})} \sin\frac{\gamma}{2} \\ e^{i(\alpha + \frac{\beta}{2} - \frac{\delta}{2})} \sin\frac{\gamma}{2} & e^{i(\alpha + \frac{\beta}{2} + \frac{\delta}{2})} \cos\frac{\gamma}{2} \end{bmatrix}$$

Two & Three Qubit gates

Controlled NOT gate

- Operates on two qbits : control and target qbits
- The target qbit is inverted if the controlling qbit is |1>.
- Logically CNOT(a,b) ↔ (a ,a ⊕ b).

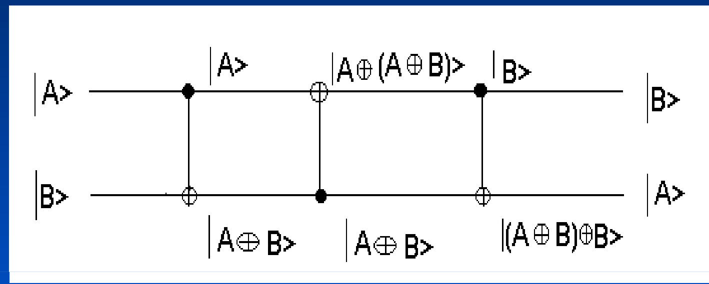
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



$CNOT\ |00> = |00>$
$CNOT\ |01> = |01>$
$CNOT\ |10> = |11>$
$CNOT\ |11> = |10>$

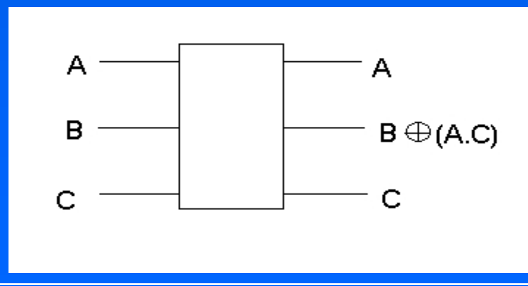
Swap Gate

- Used swapping the two i/p qbits.
- Built with CNOT gates

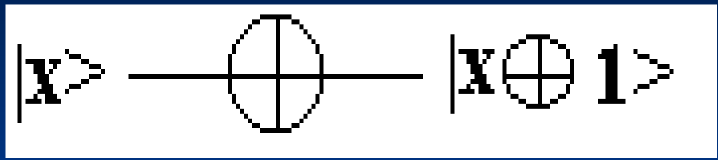


Toffoli Gate

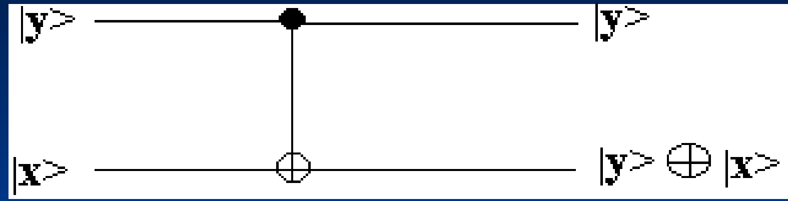
- It is a three qbit universal gate.
- The logical operations of AND, OR and NOT can be performed



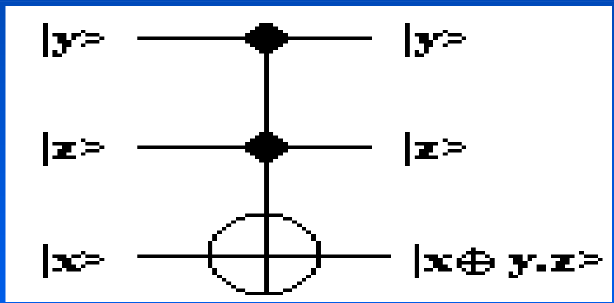
Typical Quantum gates



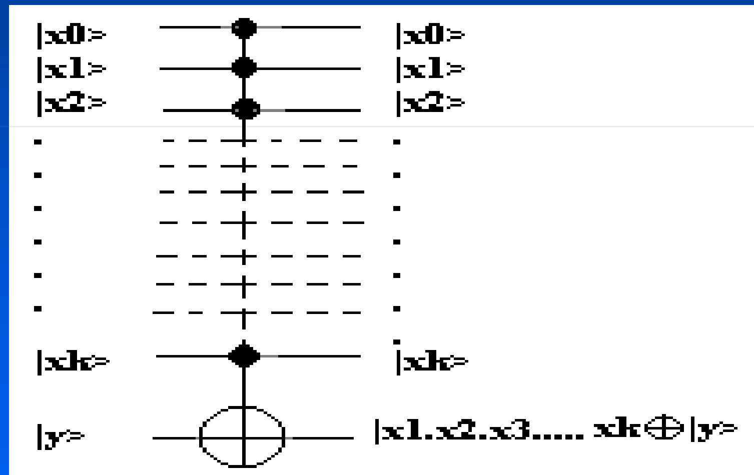
(a) NOT gate



(b) CNOT gate



(c) C^2NOT or Toffoli gate

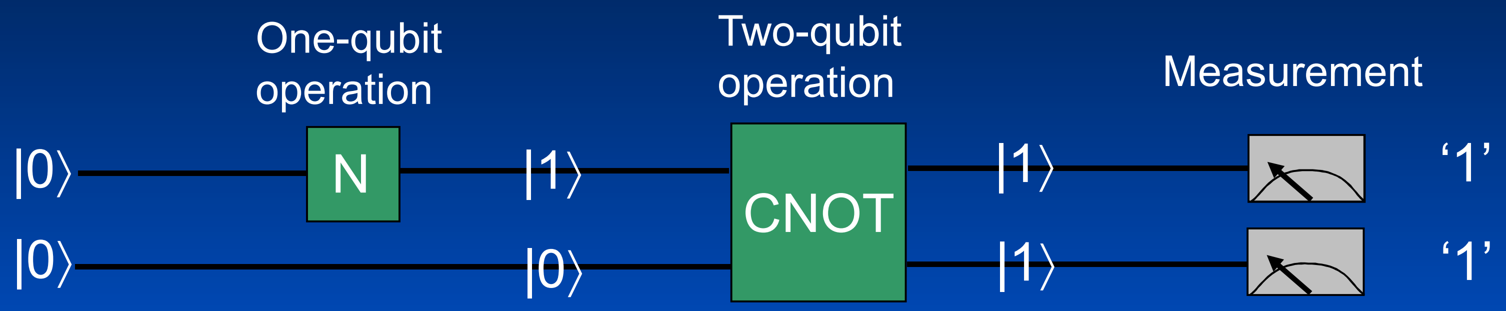


(d) C^kNOT gate

- *Universal set of quantum gates exist.*
- *Quantum Turing machines and even Universal Quantum Turing Machines exist*

Quantum Circuit

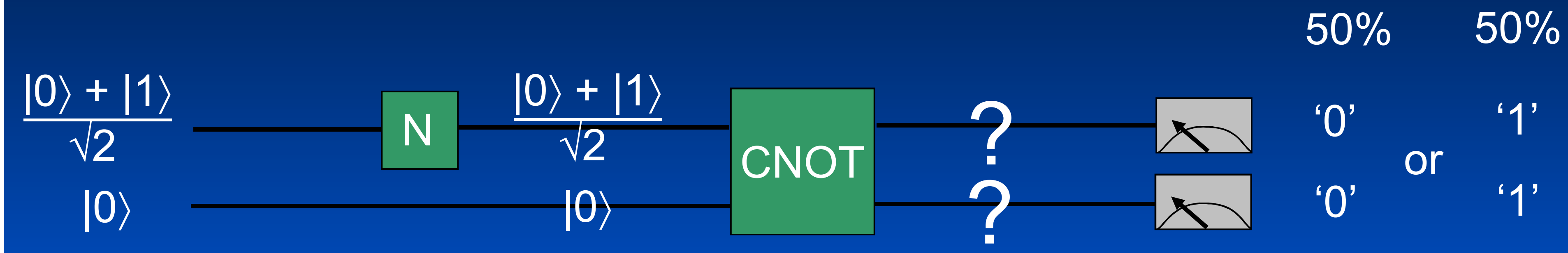
Example Circuit



$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \Rightarrow \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \Rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \Rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

$$\sigma_x \otimes I = \begin{pmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{pmatrix} \qquad \text{CNOT} = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

Example Circuit – contd.



$$\begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ or } \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Separable state:
can be written as
tensor product

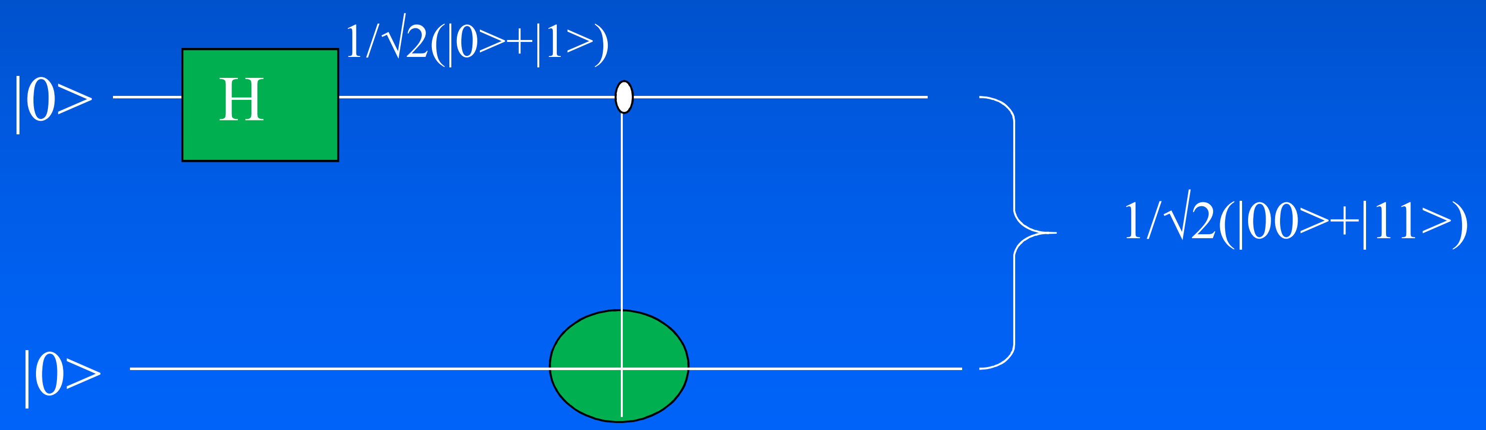
$$|\Psi\rangle = |\phi\rangle \otimes |\chi\rangle$$

Entangled state:
cannot be written
as tensor product

$$|\Psi\rangle \neq |\phi\rangle \otimes |\chi\rangle$$

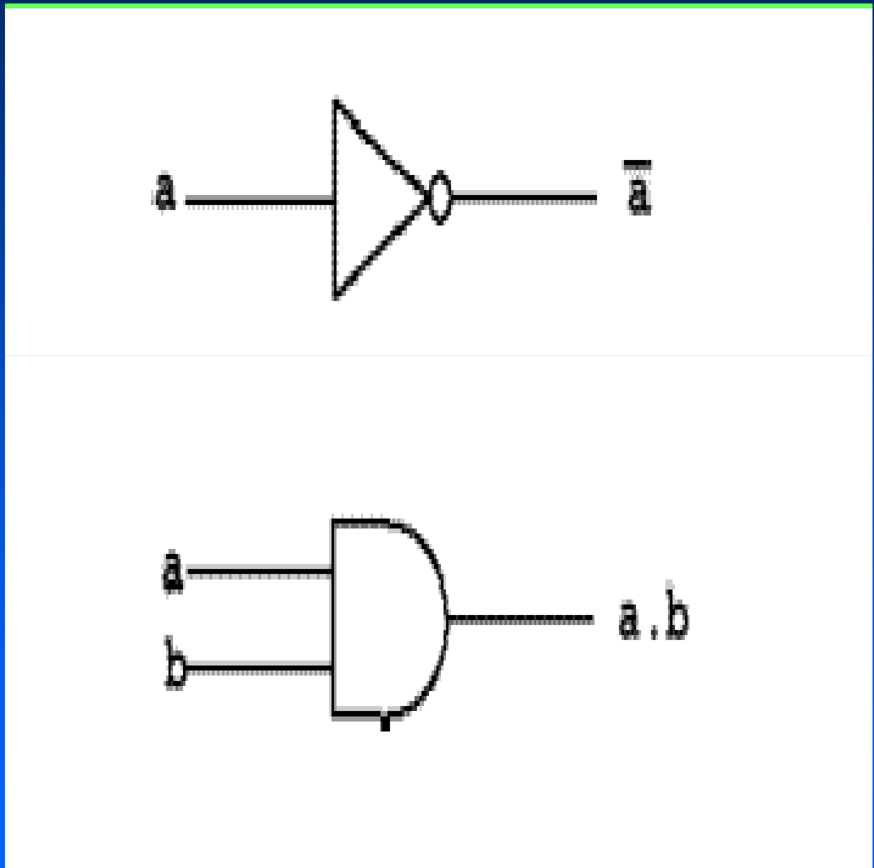
Entanglement

- Strange quantum correlation that does not have a classical analog .
- $\Psi_{entngl} = 1/\sqrt{2}(|0_1 0_2> \pm |1_1 1_2>)$.
- The quantum state cannot be factored.
- When the first qubit is measured to yield either |0> or |1> then the second qubit is determined with no uncertainty.
- Used in quantum communication over a classical channel.



A quantum circuit producing entanglement

Reversible Logic



- The NOT gate is reversible

- The AND gate is irreversible
  – the AND gate erases input information.

Need for Reversible Logic

- Landauer proved that the usage of traditional irreversible circuits leads to power dissipation.
- Bennet showed that a circuit consisting of only reversible gates does not dissipate power.
- Applications like digital signal processing, computer graphics, cryptography, reconfigurable computing, etc. demand the preservation of input data.

Some Interesting Consequences

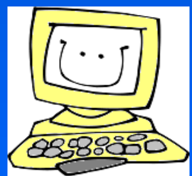
Reversibility

Since quantum mechanics is reversible (dynamics are unitary), quantum computation is reversible.

$|00000000\rangle \rightarrow$  $\rightarrow |\psi\phi\beta\pi\mu\psi\rangle \rightarrow$  $\rightarrow |00000000\rangle$

No cloning theorem

It is impossible to exactly copy an unknown quantum state

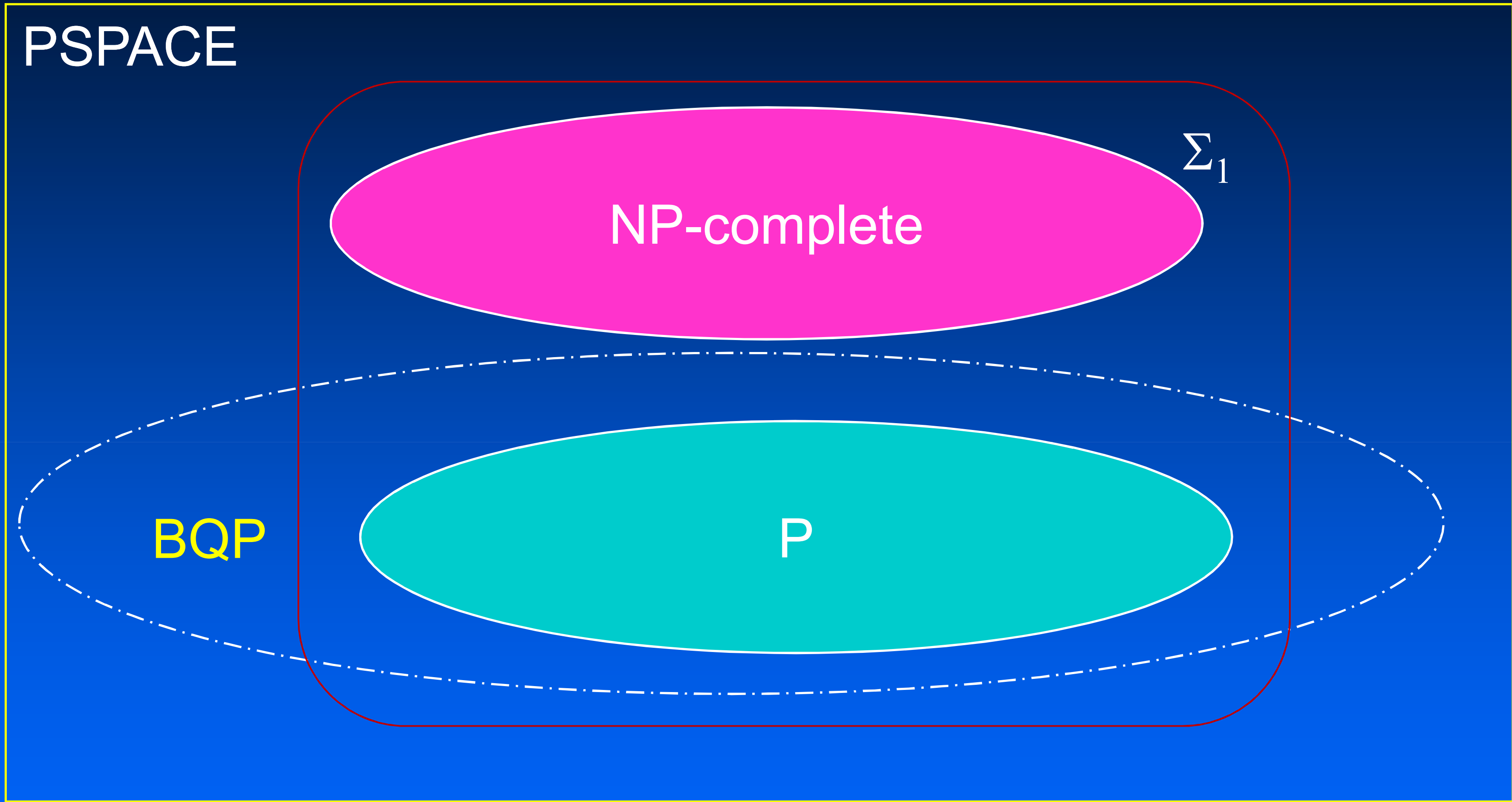
$|\psi\rangle \rightarrow$, $|0\rangle \rightarrow$  $\rightarrow |\psi\rangle$, $\rightarrow |\psi\rangle$

The power of QC

- Deutsch's oracle problem (1985) Given a classical black box function f(x) , can you tell whether f(0) = f(1) evaluating f(x) only once? Yes
- Cryptography
  - Factorization
  - Discrete Logarithm – Shor
  - Secure systems
- Database (unstructured) Search
  - O(sqrt (n)) algorithm for n entries - Grover
- Simulation of dynamic quantum mechanical systems with exponential growth of Hilbert space dimensionality
  - Quantum Chemistry
- Recent spurt in new quantum algorithms: Quantum Algorithm Zoo
  - Algebraic and number-theoretic – mostly superpolynomial speed-up
  - Oracular problems
  - Approximation algorithms

#digits	Ctime/Qctime
50	1s/1hr
100	2.5days
200	19000yrs./42 days
2000	age of universe/0.5 yr.

Computational Complexity Class



BQP: Bounded error quantum polynomial time complexity
Relation with Polynomial Hierarchy and Alternating Turing machines

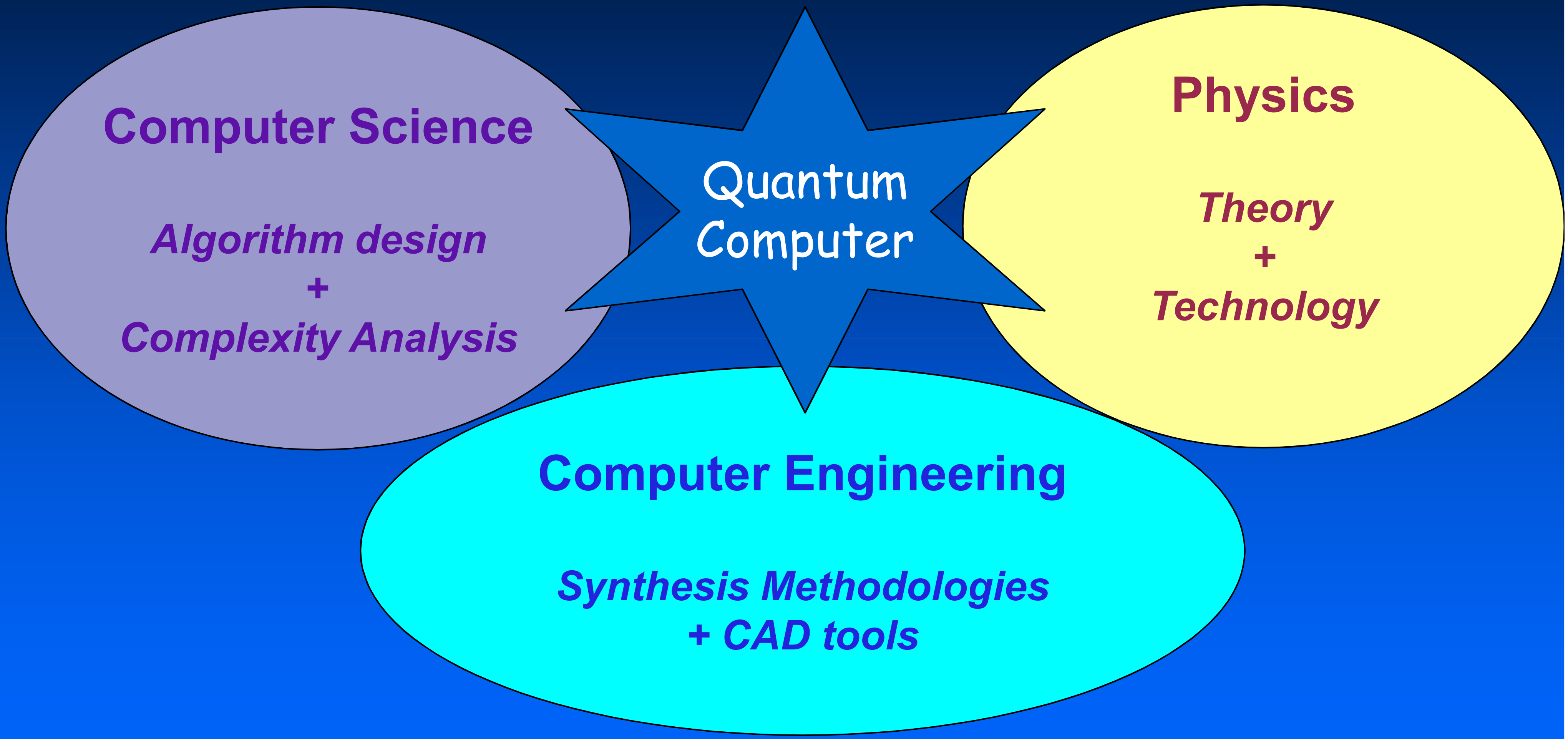
Timeline of Quantum Computing

- 1970s : reversibility, quantum information theory
- 1980s: Feyman's proposal, Toffoli gate and universal set
- 1990s:
  - entanglement based secure communication,
  - Shor's factoring,
  - CNOT with trapped ion,
  - Grover's search
  - 2-qbit NMR, 3-qbit NMR, execution of Grover's
- 2000: 5-qbit NMR, 7 qbit NMR
- 2001: execution of Shor's on 15 with 10^18 identical molecules
- 2002: Roadmap
- 2005: chip with quantum dot
- 2006: 12-qbit QC benchmarked, 2D ion trap
- 2007: 16-qbit QC (Feb)
- 2008: 128-qbit QC being tested
- 2009: Qbits transmitted at meter scale, ms life; Qproc
- 2010: Spintronics
- 2012: Superconductors

Technology as of now

- NMR – not scalable
- Ion trap
- Quantum Dots
- Superconductors
- Photons
- Neutral atoms

Building Quantum Computers



Agenda

- What is a Quantum Computer ?
- Power of Quantum Computers
- Synthesis of QCs
  - Minterm based
  - Nearest Neighbour constraints
  - Fredkin gate based Programmable Quantum Logic block
  - Fault models and Error
- Concluding Remarks

Di Vincenzo’s Criteria

A scalable system with characterized qubits.

The ability to initialize qubits in a simple state like |00….>.

Qubits must be sufficiently isolated from the environment to avoid decoherence.

Long relevance decoherence time >> gate operation time.

Universal set of quantum gates.

Measurement of qubits would be reliable.

Design Flow for
Application Specific Quantum Processor (ASQP)

Quantum Algorithm

↓

Partitioning into Quantum Macro Blocks

↓

Synthesis of Quantum Macro Blocks using high level logic blocks
(C^KNOT, Fredkin, Swap etc.)

↓

Synthesis in terms of Basic gate operations
(Toffoli, CNOT, H, NOT, CPhase, Phase)

Synthesis in terms of quantum operations available in the physical machine descriptions (PMD)

Addition of Fault-Tolerant Quantum Circuit blocks

Error correcting encoding and
Physical Realization - programming

Cost Metrics for Quantum Logic Synthesis

- Gate complexity
  - Number
- Depends on # inputs and technology:
  - Eg. for NMR based QCs:
    - 1 qbit - 2,
    - 2-qbit – 5,
    - 3 qbit – 25
- Fault model and Error performance

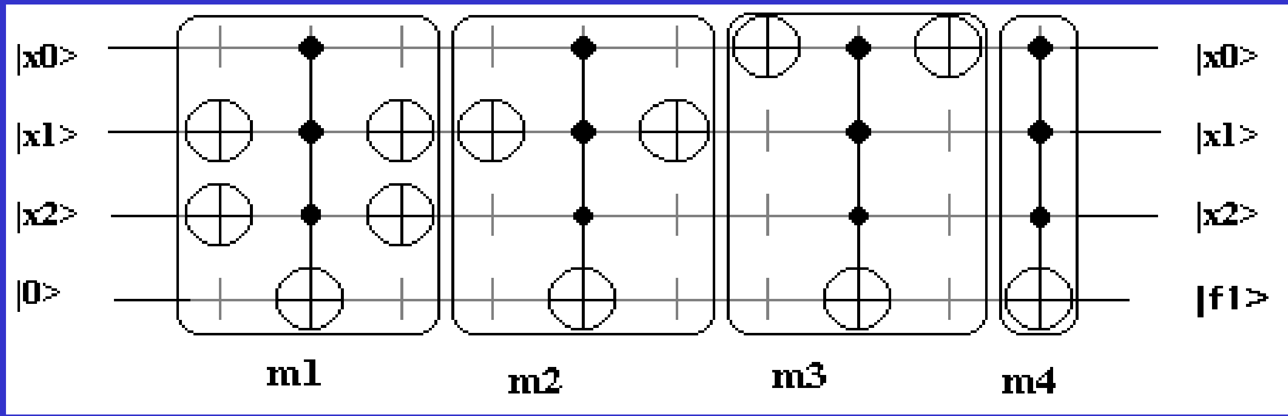
Methods of Synthesis

- Bottom-up
  - Binary logic decision diagram based
  - Quantum gate/operator identities
    - Reduction of gate complexity by using quantum logic identities
  - Reed-Muller decomposition of reversible logic
    - » Effect of Polarity on gate count
    - » Nearest neighbour constraints between the control and the target qbits

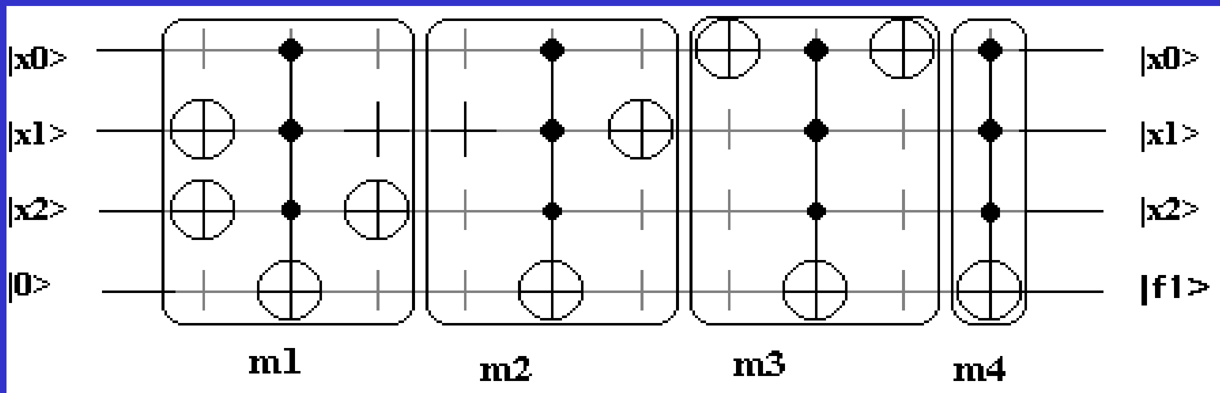
Methods of Synthesis

- Top-down
  - – Hierarchical synthesis with building blocks
  - – Decomposition into unitary matrices
    - » Eg. Perkowski's genetic algorithm
      - » Template based moves

Minterm based Quantum Boolean Circuit (QBC) Synthesis

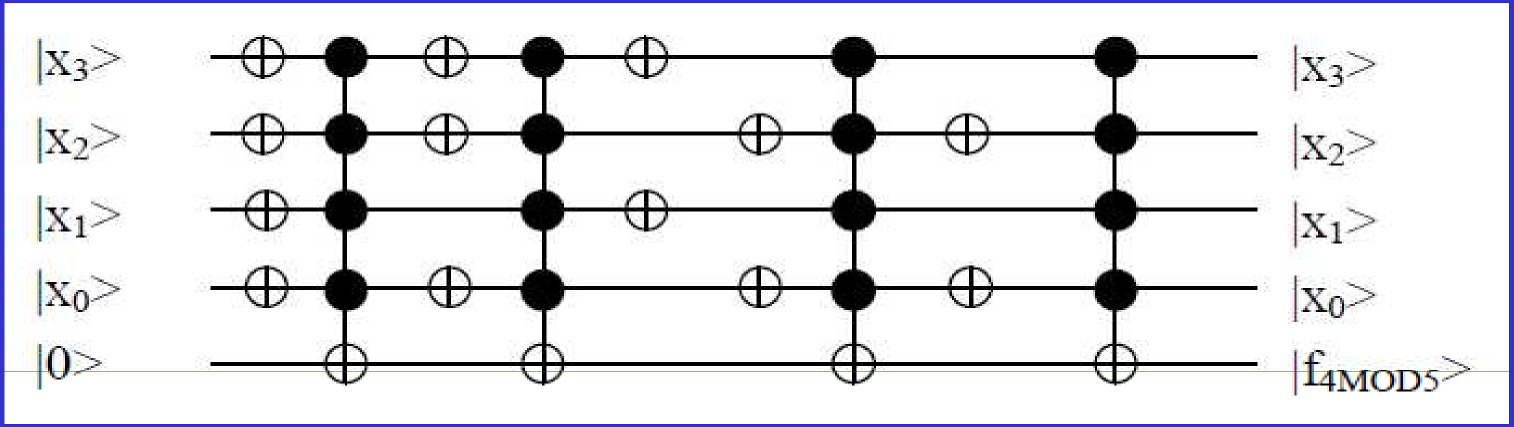


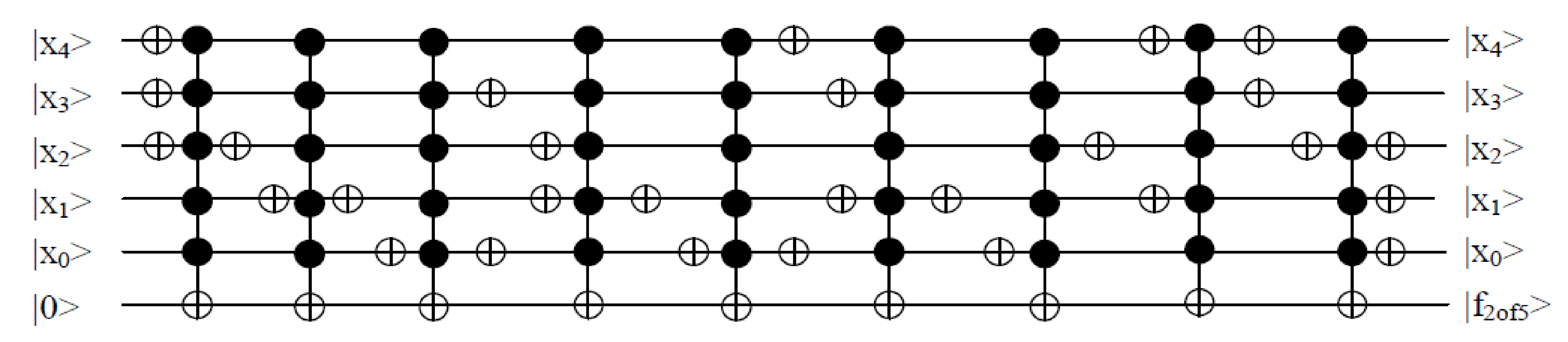
QBC for $f_1(x_0,x_1,x_2) = \overline{x}_2\overline{x}_1 x_0 + x_2\overline{x}_1 x_0 + x_2 x_1\overline{x}_0 + x_2 x_1 x_0$
using *minterm* gates *m1*,*m2*,*m3* and *m4*



Minimized circuit for $f_1(x_0,x_1,x_2)$.

Minterm Based Synthesis of Benchmarks





Reed Muller Representation

A boolean function *f* of *n* variables can be expressed in the Reed Muller form using XOR, AND and NOT (Akers 1959):

$$f(x_0', \ldots x_{n-1}') = \oplus \sum_{i=0}^{2^n-1} b_i.\varphi_i \quad ;$$

where b_i ε (0,1) and $\Phi_i = \prod_{k=0}^{n-1} \dot{x}_k$ and $\dot{x}_k = x_k$ or $\bar{x}_k$

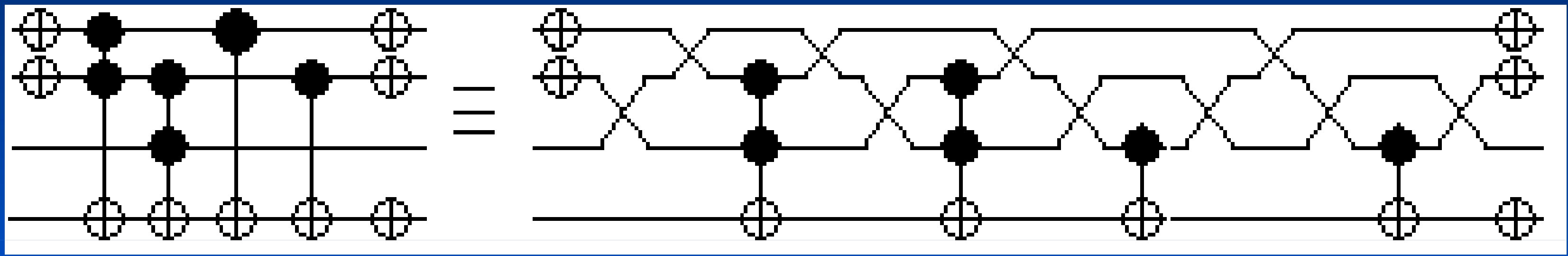
- Φ_i are known as product terms and b_i determines whether a product term is present or not .
- The XOR operation is indicated by ⊕ and multiplication is assumed to be the AND operation.
- The canonical Reed Muller expression can be classified as **Positive Polarity Reed Muller (PPRM),** where *all the variables* are un-complemented.
- In a **Fixed Polarity Reed Muller (FPRM)** expression, for each variable in $\dot{x}_k$, either x_k or $\bar{x}_k$ (not both) appear throughout.
- If both x_k and $\bar{x}_k$ appear, then it is called a Generalized Reed Muller (GRM) expression.

Nearest Neighbor Synthesis

- Younnes and Miller (2003) has mentioned about the interaction between the adjacent only qubits for the practical implementation of QBC.
- J-coupling force is required to perform multi-qubit logic operations and this works effectively only between the adjacent qubits.
- **Nearest neighbor relationship** between the control and the target qubits is truly justified due to the limitation of the J-coupling force.
- Bringing the control and the target qubits of any quantum gate on adjacent lines in a quantum gate network which is called the **nearest neighbor configuration**.
- *SWAP* gates play a key role .

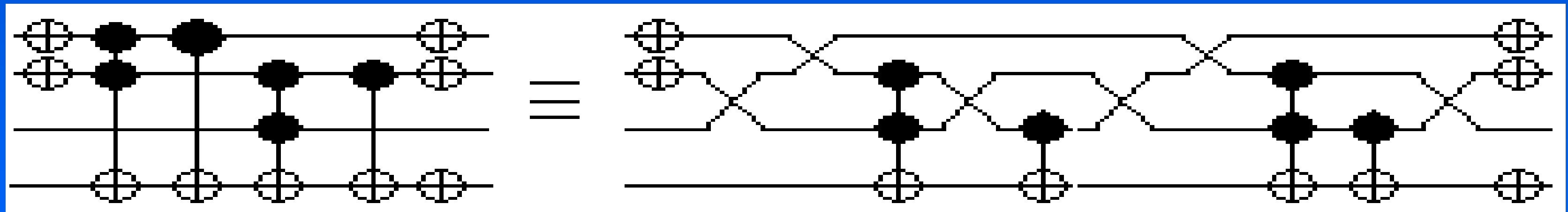
Placement Rule for Nearest Neighbor Configuration

Placement Rule: *The C^2NOT and CNOT gates which work on the same target qbit and have at least one control qbit common, should be adjacent. SWAP gates are used.*



(a) Nearest neighbor circuit using

individual templates of C^2NOT (4,3,1), CNOT(4,1) and CNOT (3,1) gates



(b) Minimized nearest neighbor circuit using
the templates of C^2NOT(4,3,1) + CNOT(4,1) and CNOT (3,1)

Minimize overhead (# SWAP gates): Ordering input qubits, position of target qubit line

Nearest Neighbor Synthesis Rules

- **Rule 1:** *For a C^2NOT gate with indices of its input lines* (*ctrl1*,*ctrl2*,*target*) *in a QBC, the number of pairs of SWAP gates required is $s_t + s_b$, where s_t is max{(ctrl1 – target-2),0} and s_b is max{(ctrl2 – target - 1),0}.*

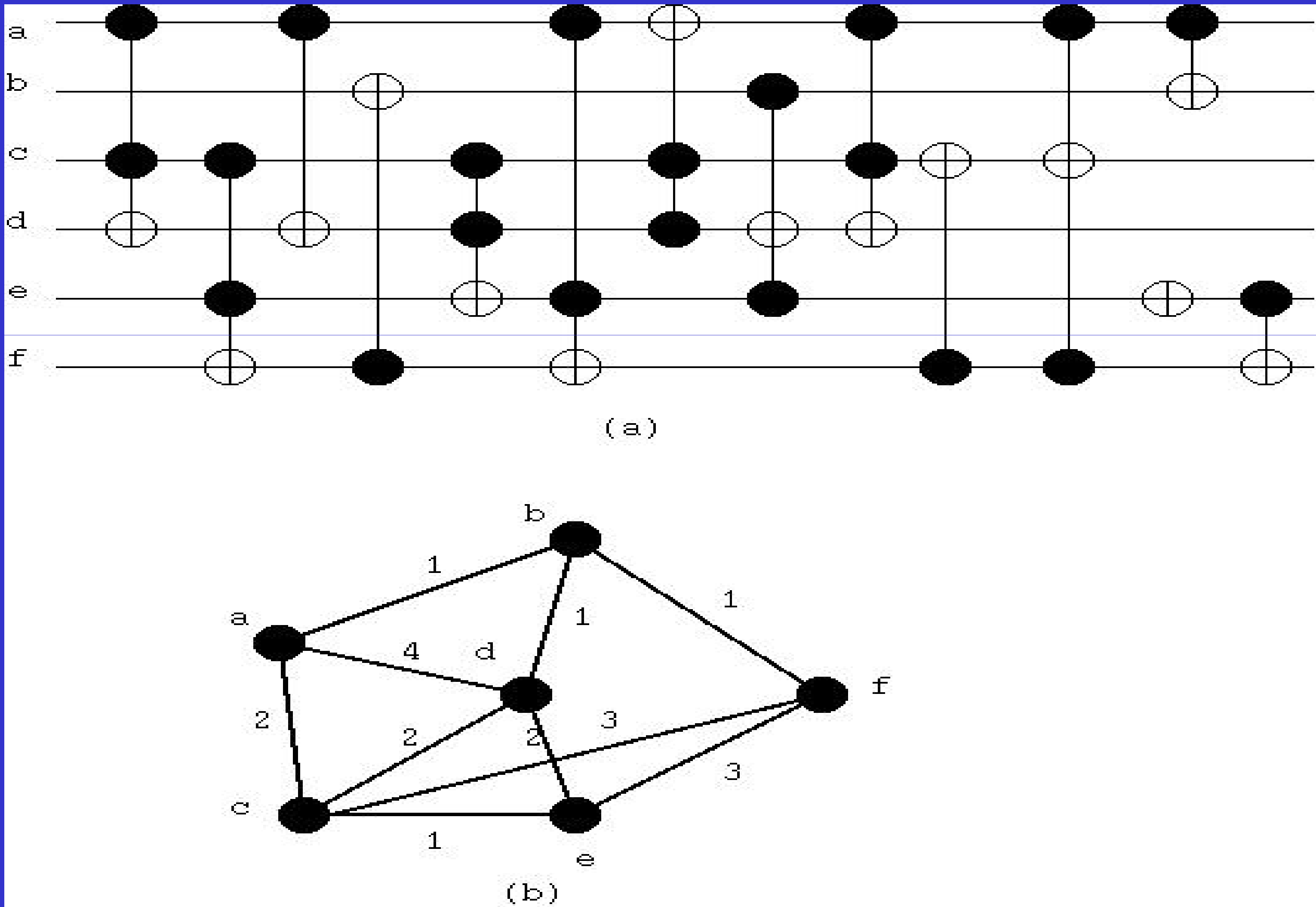
- **Rule 2:** *For a CNOT gate with indices of its input lines* (*ctrl*, *target*) *in a QBC, the number of pairs of SWAP gates required is s_c where s_c is max{(ctrl – target - 1),0}.*

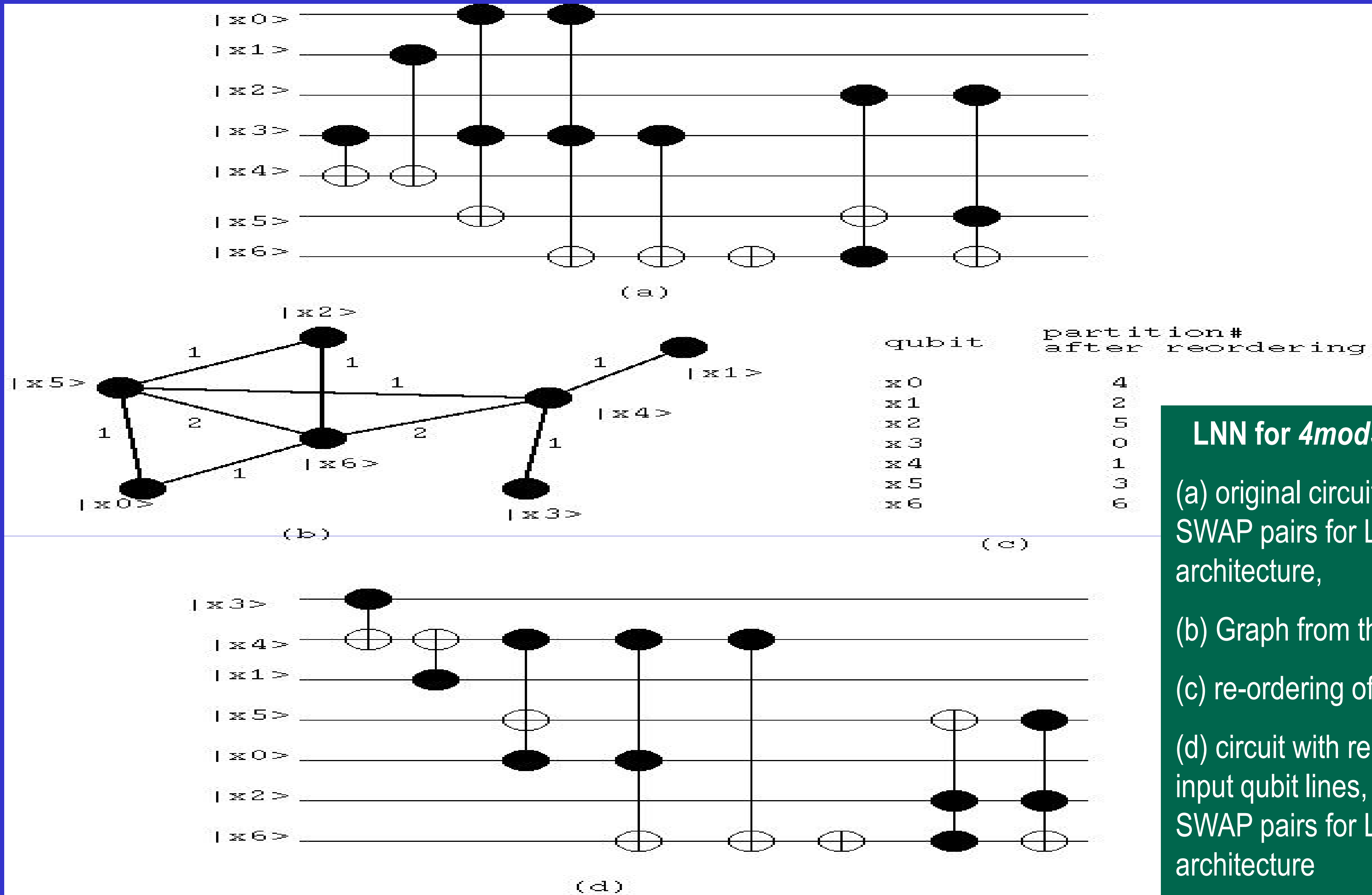
- **Rule 3:** *If the top-control qubit of a C^2NOT gate and the control qubit of a CNOT gate are on the same qubit line in a QBC, then the number of pairs of SWAP gates required is one more than that required for the C^2NOT gate alone.*

- **Rule 4:** *If the bottom-control qubit of a C^2NOT gate and the control qubit of a CNOT gate in a QBC are on the same line then the total SWAP gate requirement is same as that of the C^2NOT gate alone.*

- **Rule 5:** *The C^2NOT and CNOT gates which work on the same target qubit and have at least one control qubit common, should be adjacent.*

Linear Nearest Neighbor Synthesis for multi-target QBC by Graph Partitioning





LNN for *4mod5-bdd _287:*

(a) original circuit needs 15 SWAP pairs for LNN architecture,

(b) Graph from this circuit,

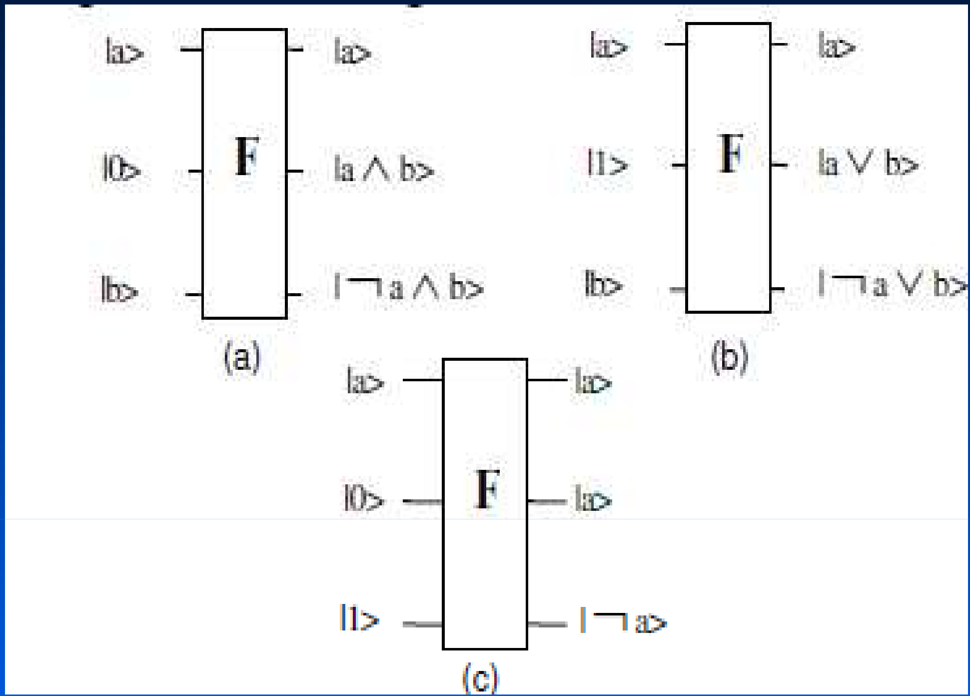
(c) re-ordering of qubit lines

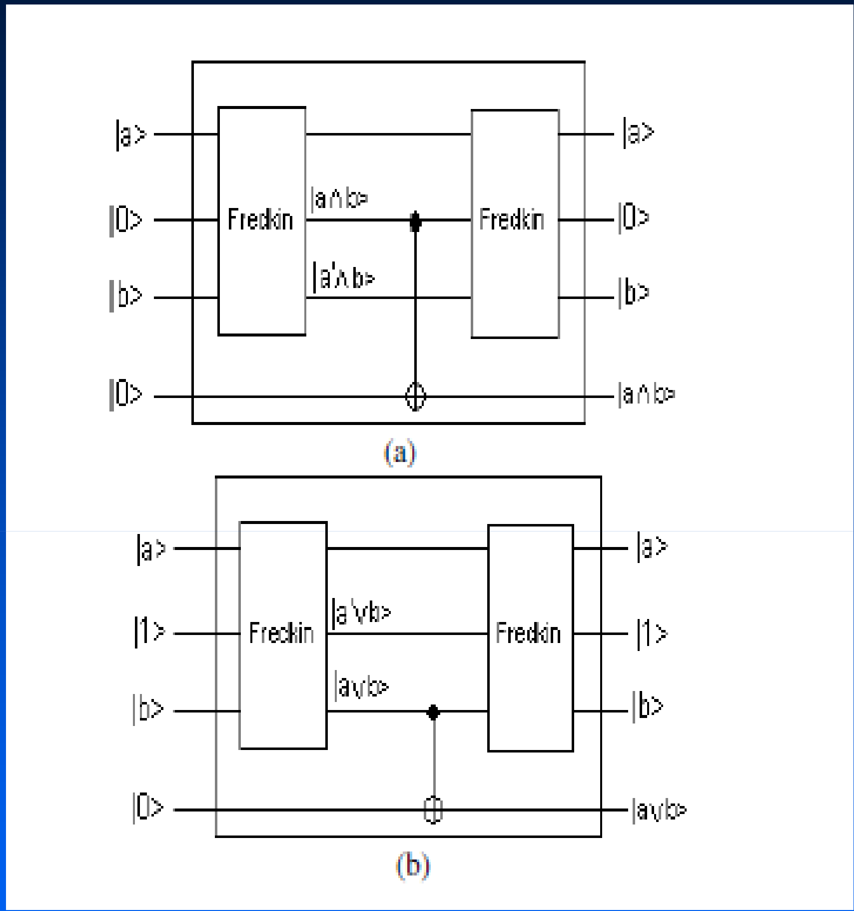
(d) circuit with re-ordered input qubit lines, with only 12 SWAP pairs for LNN architecture

Over *Revlib benchmarks*, mean reduction in QC is 46.6%,and 17.5% over method by Wille *et al.*

Amlan Chakrabarti, Susmita Sur-Kolay, Ayan Chaudhury: Linear Nearest Neighbor Synthesis of Reversible Circuits by Graph Partitioning CoRR abs/1112.0564: (2011)

Fredkin gate based synthesis

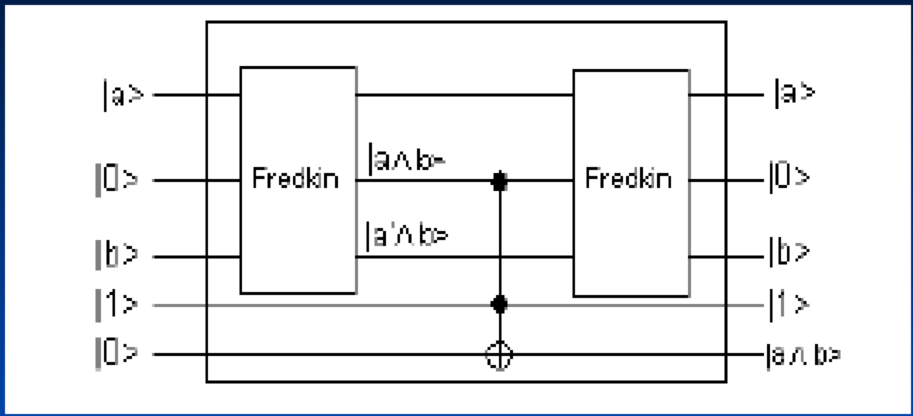




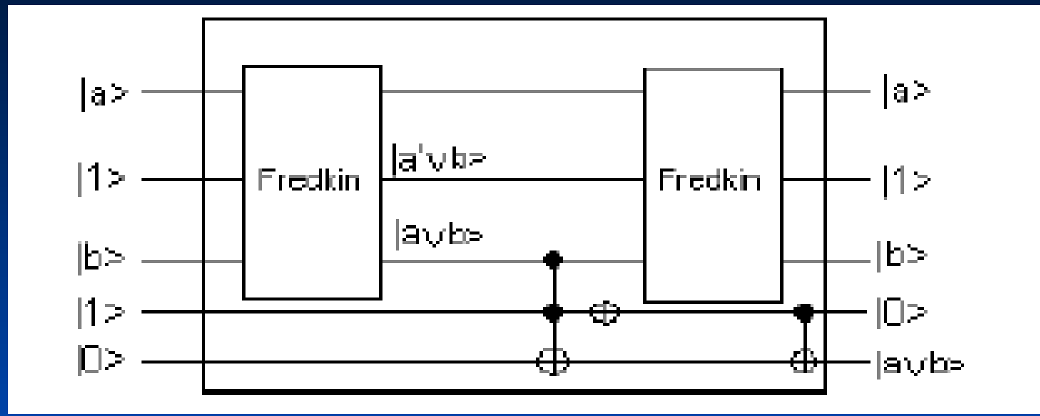
Can realize

- AND, OR, NOT,
- 2-to-1 multiplexer
- Multiplexer network
- garbage-free versions

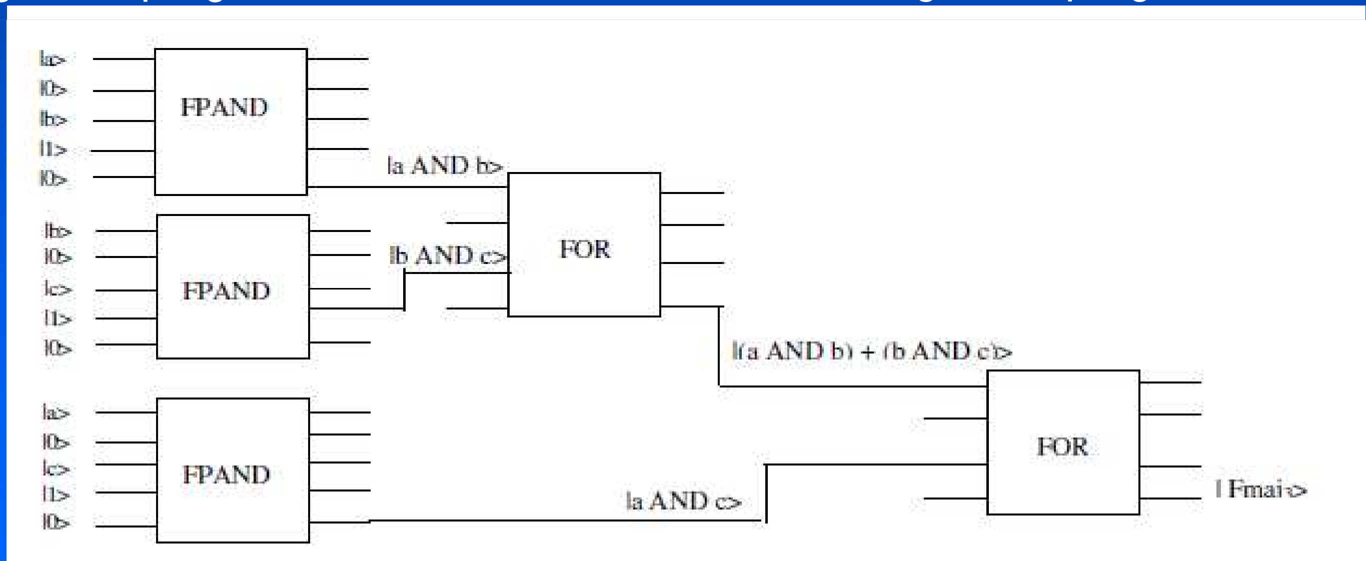
Fredkin gate based
Programmable Quantum Logic Block or *PLAQ*



FPAND: Garbage-free programmable AND



FPOR: Garbage-free programmable OR



Garbage-free programmable QPLA for *Majority3* with FPAND plane and FPOR plane

Chakrabarti and Sur-Kolay: IJCSES'09

Gate-Level Error Model

Types of Errors: Bit flip or phase flip or both

NOT:

Output ↓ \ Input →	0	1
0	p	1-p
1	1-p	p

HADAMARD:

Output \ Input →	0	1
$\frac{1}{\sqrt{2}}(\|0> - \|1>)$	p	1-p
$\frac{1}{\sqrt{2}}(\|0> + \|1>)$	1-p	p

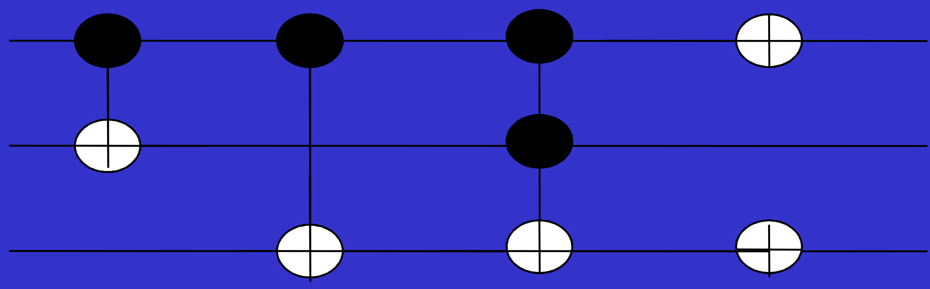
TOP-CNOT (top-control):

	00	01	10	11
00	1-p	p	0	0
01	p	1-p	0	0
10	0	0	p	1-p
11	0	0	1-p	p

BOT-CNOT (bottom-control):

	00	01	10	11
00	1-p	0	p	0
01	0	p	0	1-p
10	p	0	1-p	0
11	0	1-p	0	p

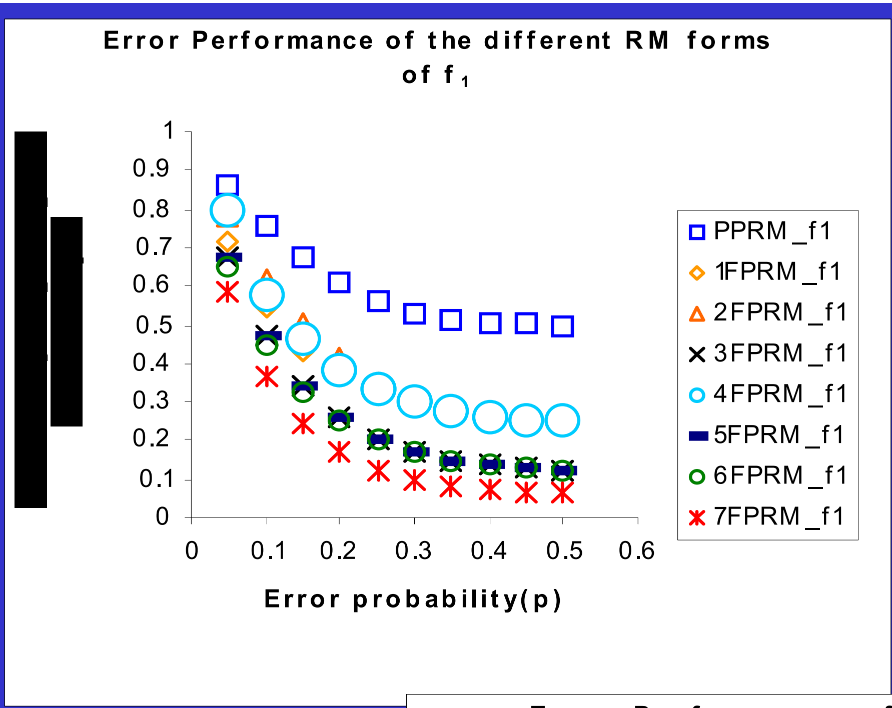
Error Computation Method



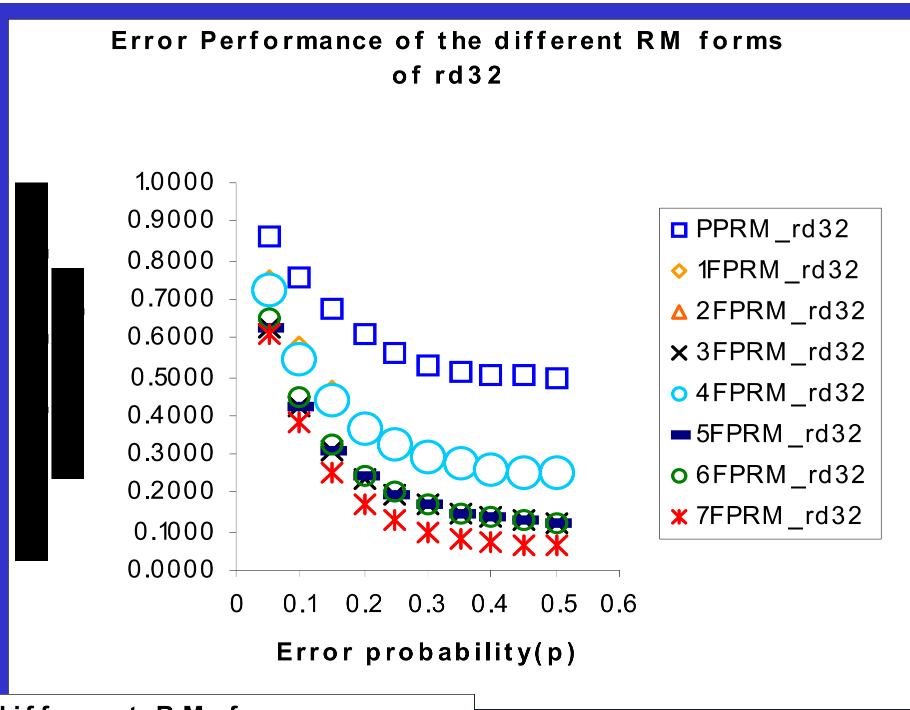
A Quantum circuit with 4 gate levels

Matlab Program for this Quantum Circuit:

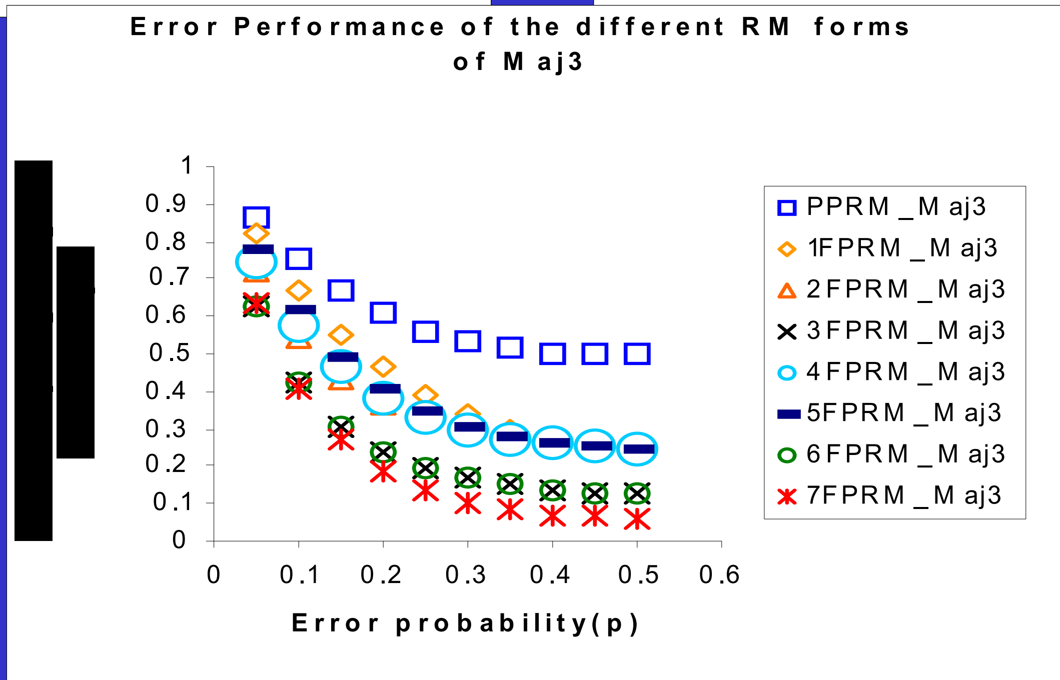
```
function example_array = example(p)
L1= kron (topcnot(p),eye(2,2));
L2= CNOT3_1(p);
L3= toffoli(p));
L4=kron (kron(my_not(p),eye(2,2)), my_not(p));
example_array = L4*L3*L2*L1;
```



Error Performance of the different RM forms of f_1
[redacted]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6
Error probability(p)
PPRM_f1
1FPRM_f1
2FPRM_f1
3FPRM_f1
4FPRM_f1
5FPRM_f1
6FPRM_f1
7FPRM_f1

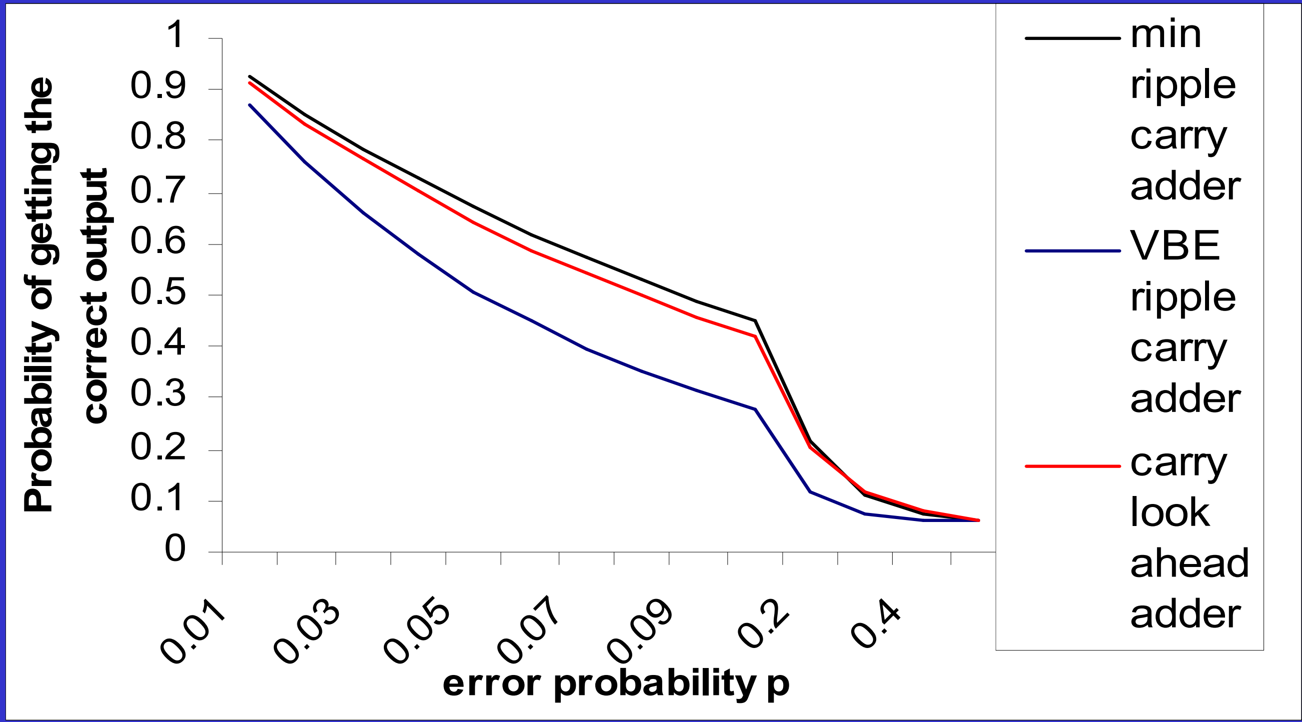


Error Performance of the different RM forms of rd32
[redacted]
1.0000
0.9000
0.8000
0.7000
0.6000
0.5000
0.4000
0.3000
0.2000
0.1000
0.0000
0 0.1 0.2 0.3 0.4 0.5 0.6
Error probability(p)
PPRM_rd32
1FPRM_rd32
2FPRM_rd32
3FPRM_rd32
4FPRM_rd32
5FPRM_rd32
6FPRM_rd32
7FPRM_rd32



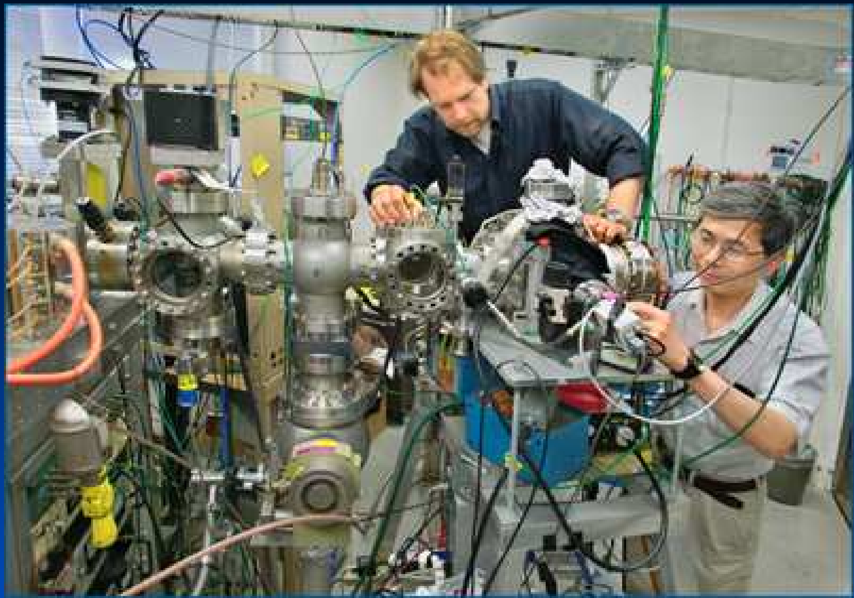
Error Performance of the different RM forms of Maj3
[redacted]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6
Error probability(p)
PPRM_Maj3
1FPRM_Maj3
2FPRM_Maj3
3FPRM_Maj3
4FPRM_Maj3
5FPRM_Maj3
6FPRM_Maj3
7FPRM_Maj3

Error performance of the adder circuits

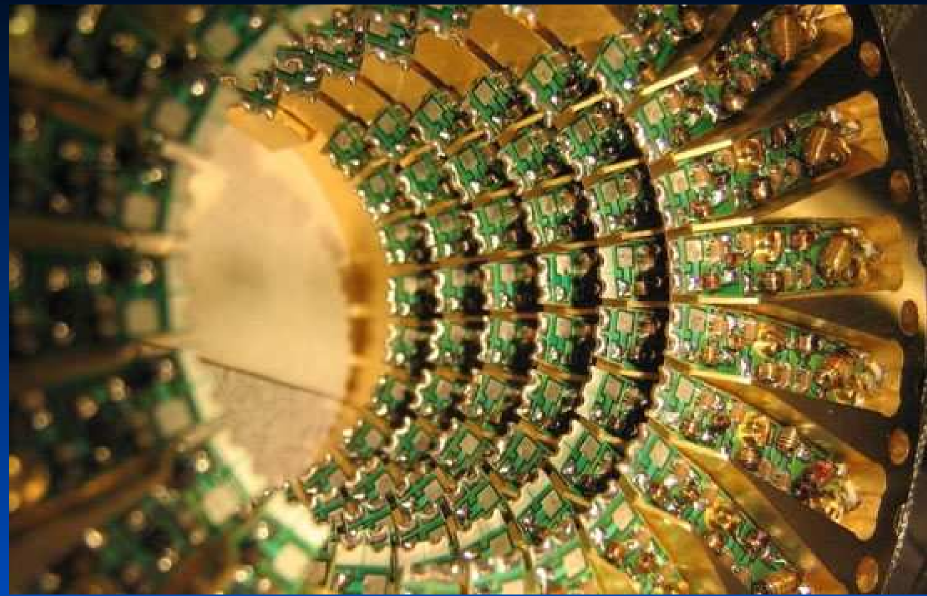


A. Chakrabarti and S. Sur-Kolay, "Designing Quantum Adder Circuits and Evaluating Their Error Performance", Proc. IEEE International Conference on Electronic Design 08, Penang, Malyasia, December 2008.

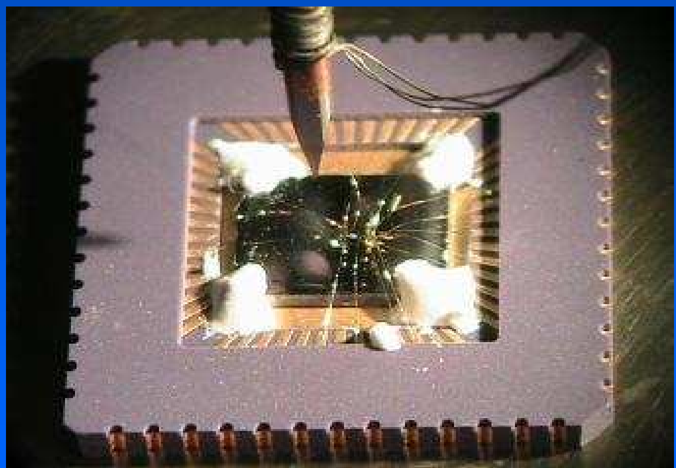
Need efficient methods



Ion trap

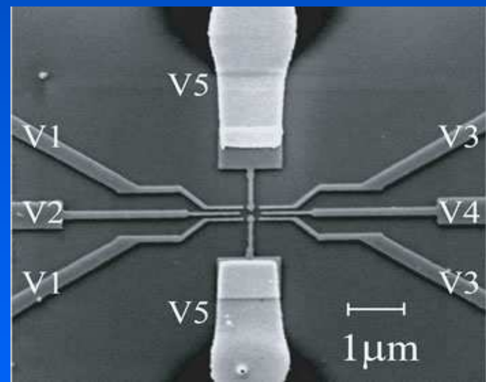


NMR Spin



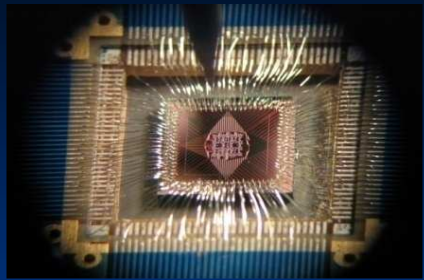
Qchip

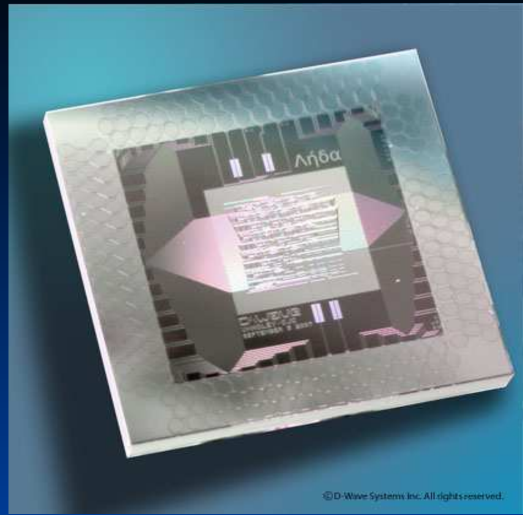
Qchip bus



Quatum dot switches

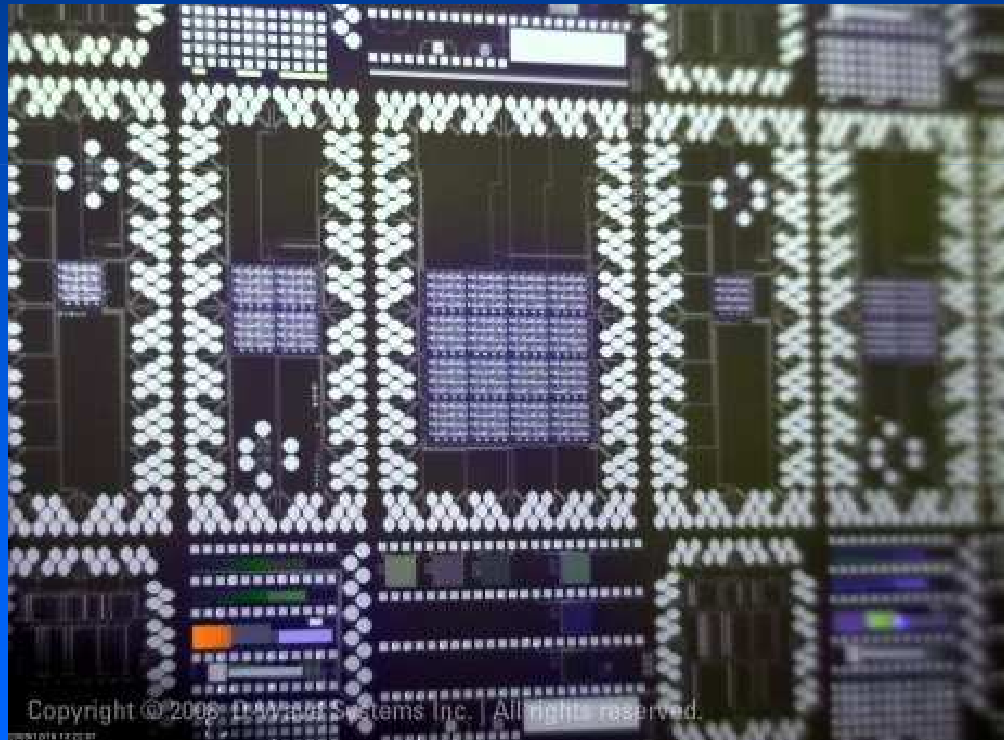
D-Wave





28 qubit chip

Superconducting processor



128 qubit chip

Our Recent Contributions*

- Synthesis of quantum computers with reduced gate cost and delay
  - Quantum Boolean circuits
    - Quantum registers,adders, multiplexers
    - *Programmable* Quantum Boolean circuits with Fredkin gates
  - Reversible logic
    - Top-down: Decomposition approach
    - Bottom-up: Reed-Muller approach
- Physical constraints
  - Nearest Neighbour Placement Rules for Quantum gates
- Methodology for computation of error
- Synthesis of Multi-valued Quantum Logic
- Efficient Simulation of QCs

* Joint work with Amlan Chakrabarti, Sudhindu Mandal

Upcoming Research Issues

- Physical Machine Description specific quantum library generation
  - Each of the PMDs have a variability in terms of basic quantum operations they support, so PMD specific quantum operator library needs to be created.
- Quantum error coding and correction schemes
  - Efficient schemes are needed having less overhead, absolute need for quantum computing in reality.
- Quantum control
  - Efficient control on the quantum operations is required for successful quantum computing.
- Model of a quantum finite state machine
  - Halting problem for QTMs !?

Concluding remarks

- Need efficient synthesis methods
- Efficient simulators
- Programming models
- Testing and verification
- More real-life applications
  - – eg. Quantum cryptography in Swiss ballot boxes
- Technological progress and Scalability

The Ultimate Computer:

Mass 1 kg, Volume: 1 litre

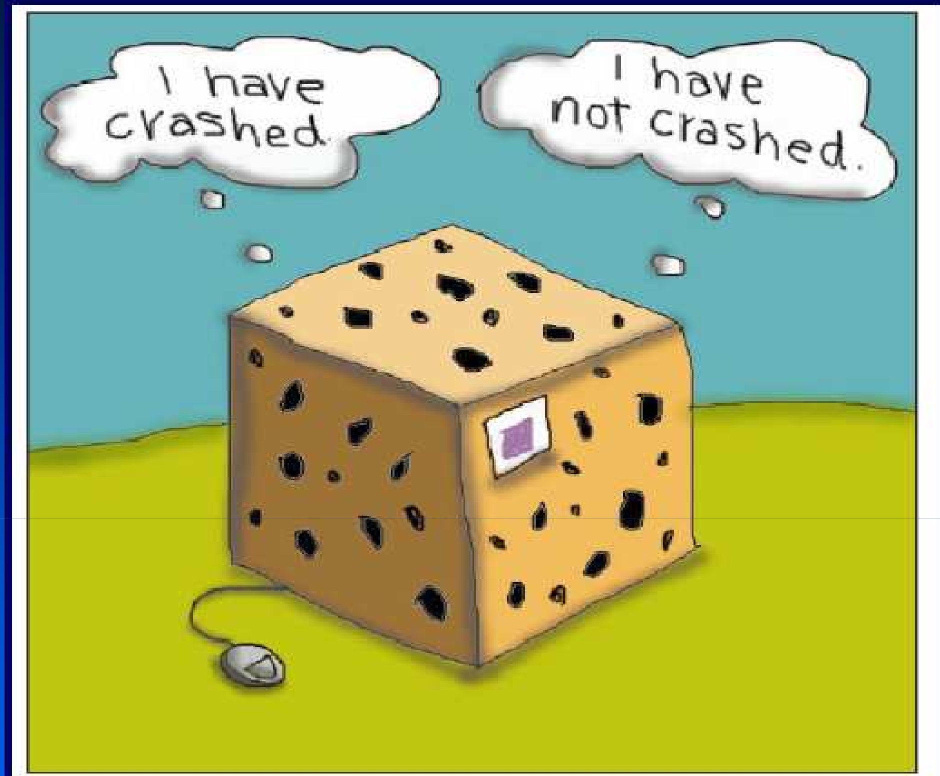
Max. $E = mc^2 = 9 \times 10^{16}$J

Max. Speed (ops) = 5×10^{50}

Error rate: 10^{-8} at 6×10^{8} K temp.

Courtesy: Stolze, Suter

QWINDOWS 2098



Schrödinger's computer. —Sally O. Lee

Thank you

References

- R. P. Feynman, (1982). "Simulating Physics with Computers". *Intl. J. of Theoretical Physics* **21**: 467–488.
- D. Deutsch, (1985). "Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer". *Proc. Roy. Soc. Lond. A400*, 97–117.
- R. P. Feynman, (1986). Quantum Mechanical Computers, *Foundations of Physics*, Vol.16, 507-531.
- P. Shor, (1994). "Algorithms for quantum computation: discrete logarithms and factoring". *Proc. 35th FOCS, Santa Fe, NM, USA,* 124–134.. arXiv:quant-ph/9508027
- L. K. Grover, (1996). "A Fast Quantum Mechanical Algorithm for Database Search". *Proc. 28th Annual ACM STOC,* 212–219.. arXiv:quant-ph/9605043
- P. W. Shor, (1998) Quantum Computing, *Documenta Mathematica* - Extra Vol. ICM'98, 1-1000.
- N. Gershenfeld and I. L. Chuang, (1998) *Quantum Computing with Molecules*, Scientific American - June.
- J. Preskill, (1998) Quantum Computing: Pro and Con, *Proc. Roy. Soc. Lond*. A454, 469-486, arXiv:quant-ph/9705032.
- H.Buhrman, R. Cleve , and A. Wigderson, (1998) Quantum vs. classical communication and computation, *Proc. 30th Annual ACM STOC*, 63-68.
- M. A. Nielsen and I. L. Chuang, (2002) *Quantum Computation and Quantum Information*, Cambridge University Press.
- J. Gruska, Quantum Computing
- I. Oliveira . NMR Quantum information Processing
- D. MacMohan (2007). *Quantum Computing Explained.* Wiley Interscience.