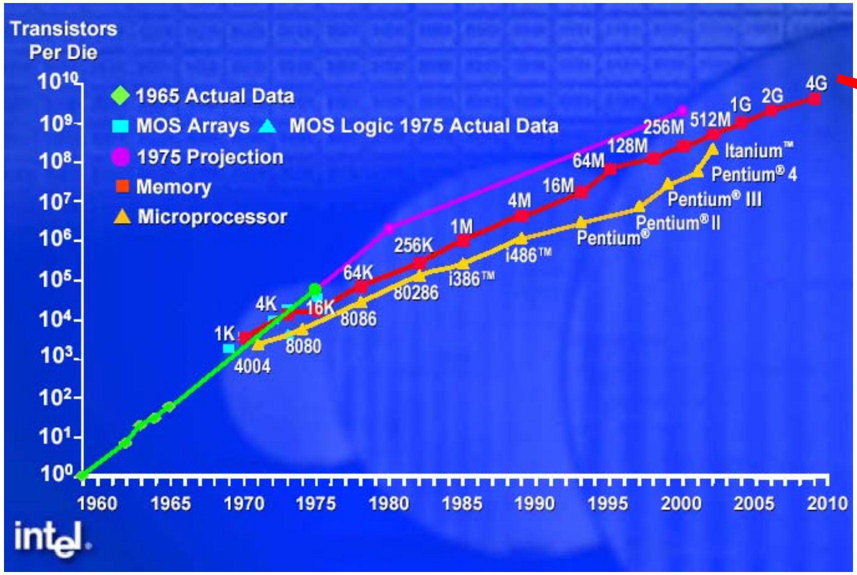
當上帝擲骰子做計算

陳柏中
清大物理系

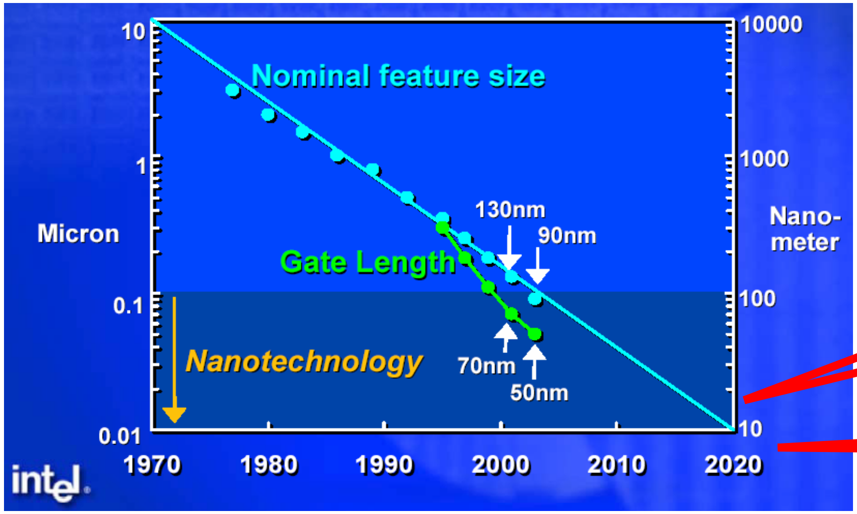
Outline

- Introduction
- Quantum Circuit and Login
- Quantum Fourier Transform (QFT)
- Shor’s algorithm
- Summary

摩爾定律 Moore's law



如何持續成長？



微小尺度=>量子力學效應？

危機？ 轉機？

先知……….

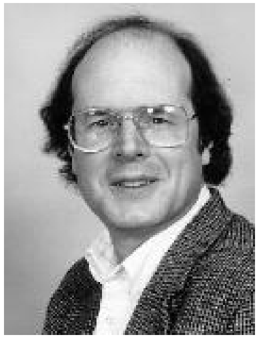
"I think I can safely say that nobody understands quantum mechanics"

There's Plenty of Room at the **Bottom**

Richard P. Feynman, December 29th 1959

When we get to the very, very small world---say circuits of seven atoms---we have a lot of new things that would happen that represent completely new opportunities for design. Atoms on a small scale behave like *nothing* on a large scale, for they satisfy the laws of quantum mechanics. So, as we go down and fiddle around with the atoms down there, we are working with different laws, and we can expect to do different things. We can manufacture in different ways. We can use, not just circuits, but some system involving the quantized energy levels, or the interactions of quantized spins, etc.

八十年代的先行者



C.H. Bennett

G. Brassard

D. Deutsch

C.H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner
"*Quantum Cryptography, or Unforgetable Subway Tokens*" , 1983.

C.H. Bennett and G. Brassard
"*Quantum Cryptography: Public Key Distribution and Coin Tossing*", 1984.

C.H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner,
"*Eavesdrop-Detecting Quantum Communications Channel*", 1985.

D. Deutsch
"*Quantum theory, the Church-Turing principle and the universal quantum computer*", 1985.

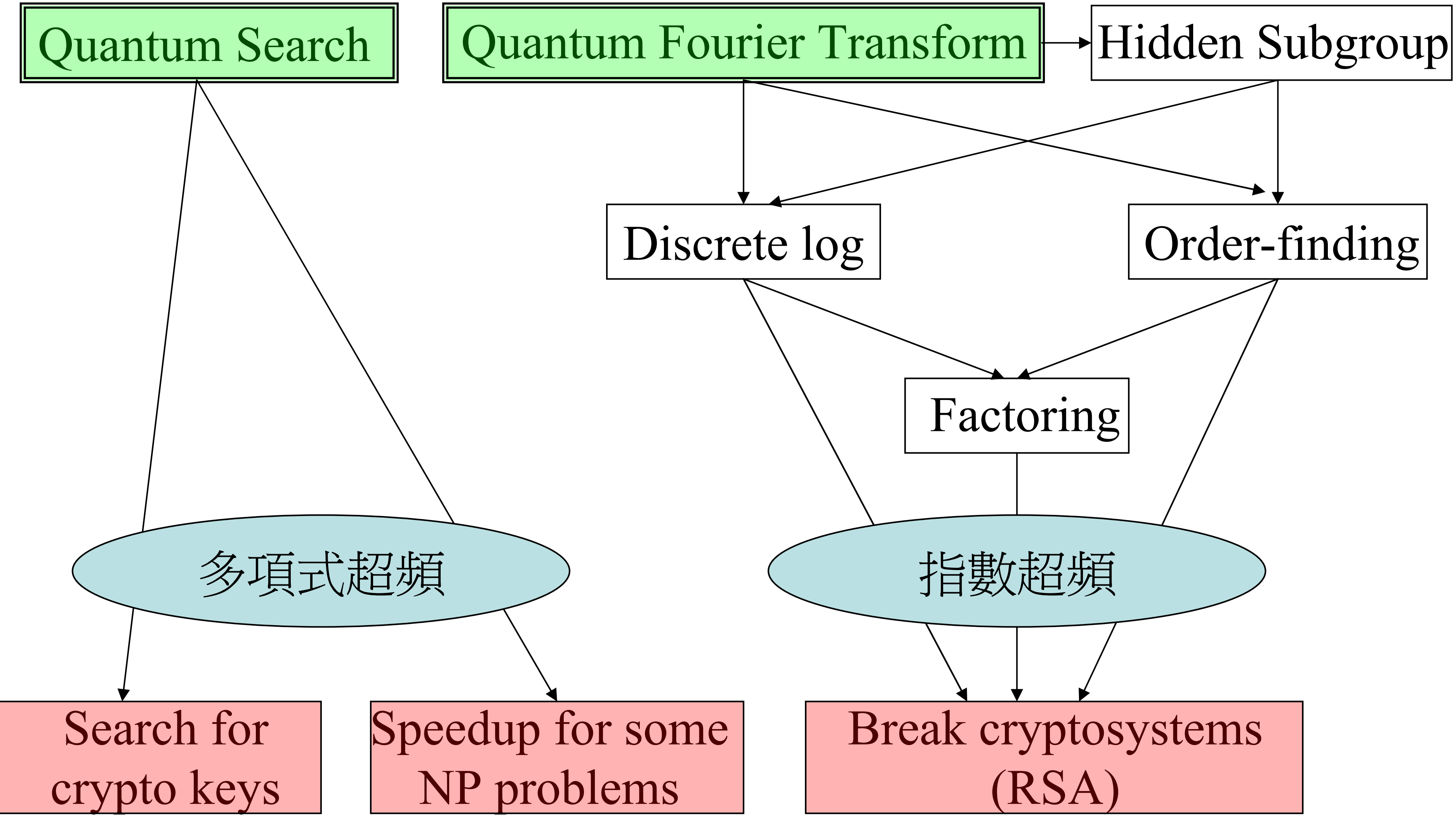
P.W. Shor

九十年代的大突破

利用量子電腦分解大質數

- P. W. Shor,
  - “Algorithms for quantum computation: Discrete logarithms and factoring”
  - *Proc. 35nd Annual Symposium on Foundations of Computer Science*
  - IEEE Computer Society Press (1994), 124-134.

已知的量子演算法



Quantum Search
Quantum Fourier Transform
Hidden Subgroup
Discrete log
Order-finding
Factoring
多項式超頻
指數超頻
Search for crypto keys
Speedup for some NP problems
Break cryptosystems (RSA)

資訊＝可區分性

- 想一想,爲什麼用筆把字寫下是一種儲存資訊的方法？

- 因爲我們可以區分不同的字,也就用字的可區分性來儲存資訊。

- 從這樣的觀點來看現在的電腦
    - 所有的資訊可以化約爲位元(O, 1).
    - 所以有的資訊處理過程都可以化約成邏輯閘(NOT,AND)
    - 這樣的原理,讓我們可以用不同的物理系統去製造電腦

$$01010 \neq 10010$$

資訊的物理性
Information is Physical

- 當我們走到微觀的尺度
  - 構成資訊的單元如光子,自旋等,必須遵守量子力學的原理
  - 量子態無法在不受干擾的情況下被量測或拷貝

- 我們需要一套量子資訊學

- 量子資訊學可以化約成
  - 量子位元 Quantum Bit
  - 量子運算 Quantum Gate
  - 萬用量子計算模型 Universal Quantum Computation

非零即一的傳統位元

$\psi=0$ or $\psi=1$

變換莫測的量子位元

Quantum Bit=Qubit

$\psi=a|0>+b|1>$

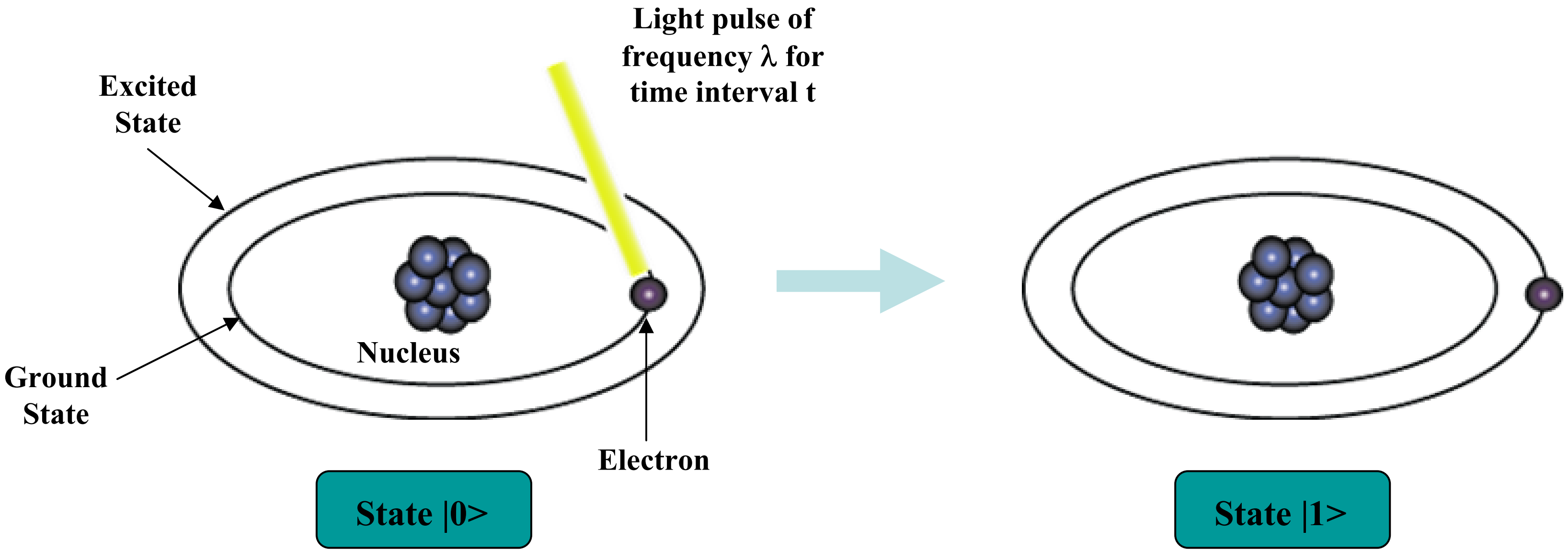
0

Center for Quantum Computation http://www.qubit.org/

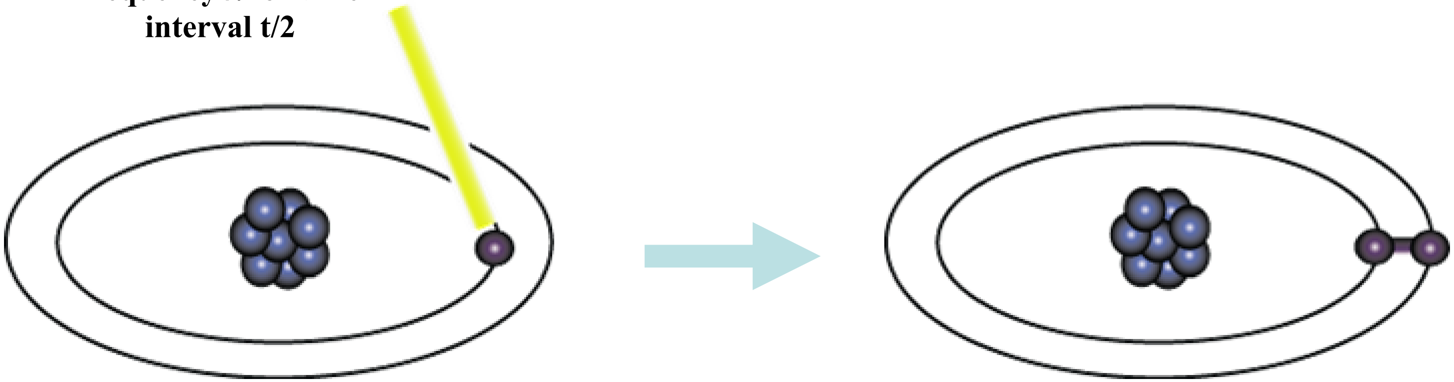
Representation of Data - Qubits

A bit of data is represented by a single atom that is in one of two states denoted by |0> and |1>. A single bit of this form is known as a ***qubit***

A physical implementation of a qubit could use the two energy levels of an atom. An excited state representing |1> and a ground state representing |0>.



Representation of Data - Superposition



Light pulse of
frequency λ for time
interval t/2

State |0>

State |0> + |1>

Data Retrieval

- In general, an n qubit register can represent the numbers 0 through 2^n-1 simultaneously.

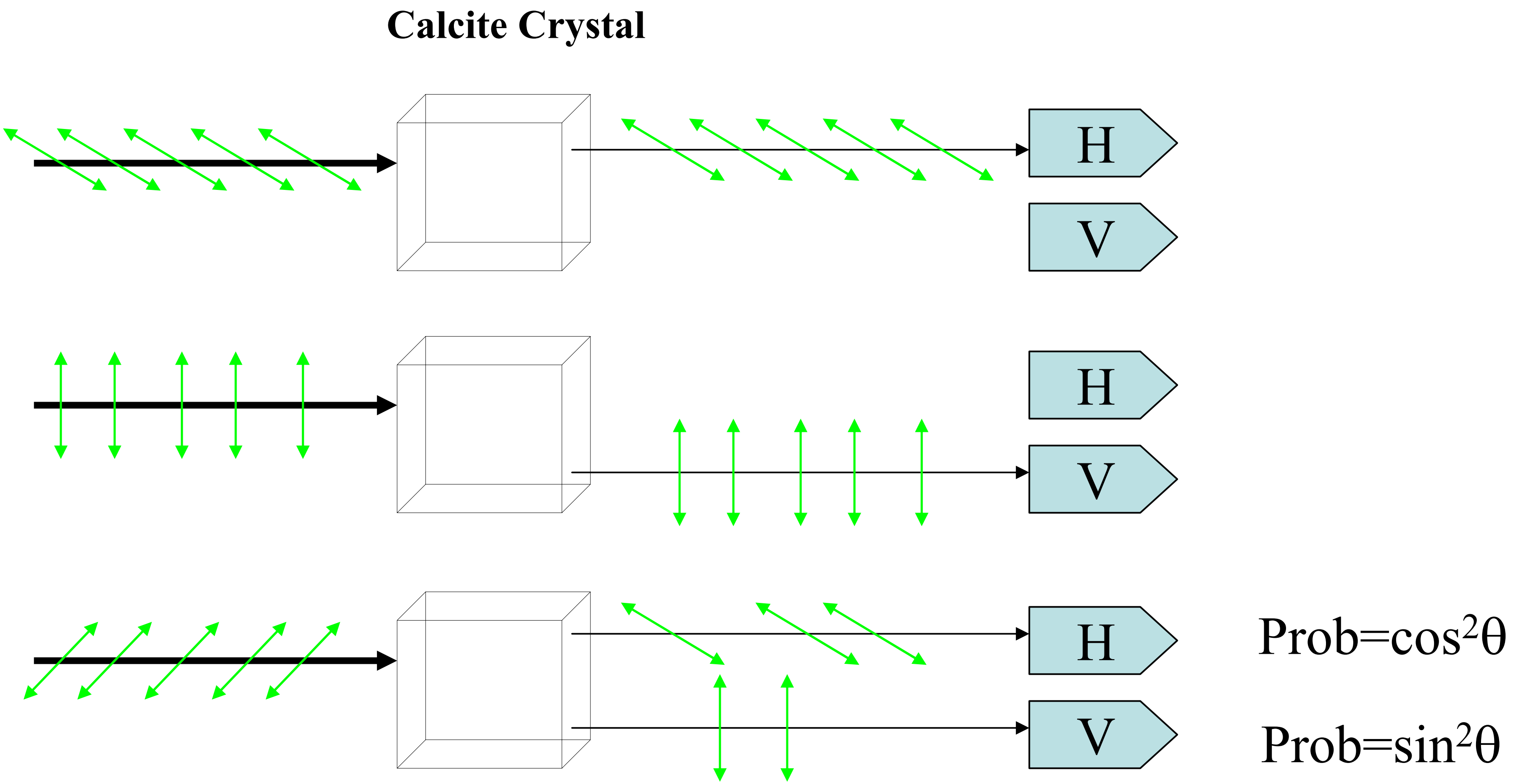
Sound too good to be true?…It is!

- If we attempt to retrieve the values represented within a superposition, the **superposition randomly collapses** to represent just one of the original values.

1

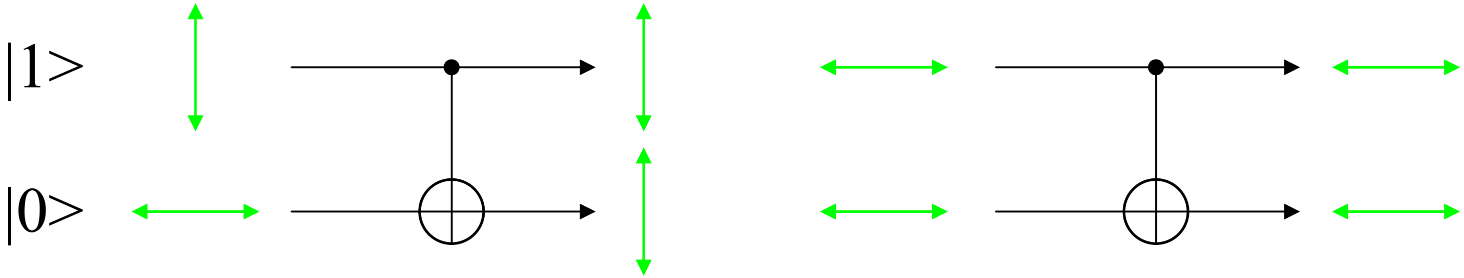
n

量子態疊加原理
Supperposition Principle

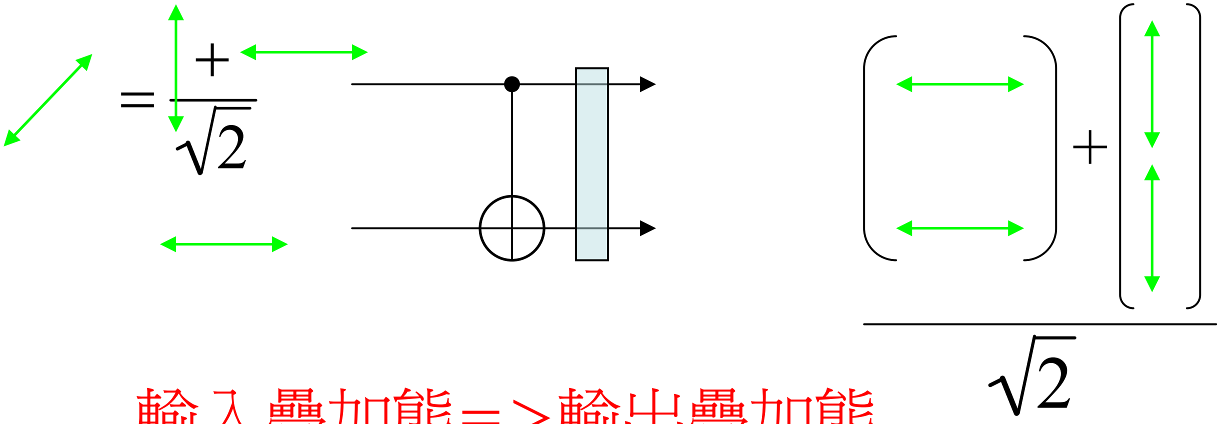


量子平行運算

傳統電腦的XOR=Control Not

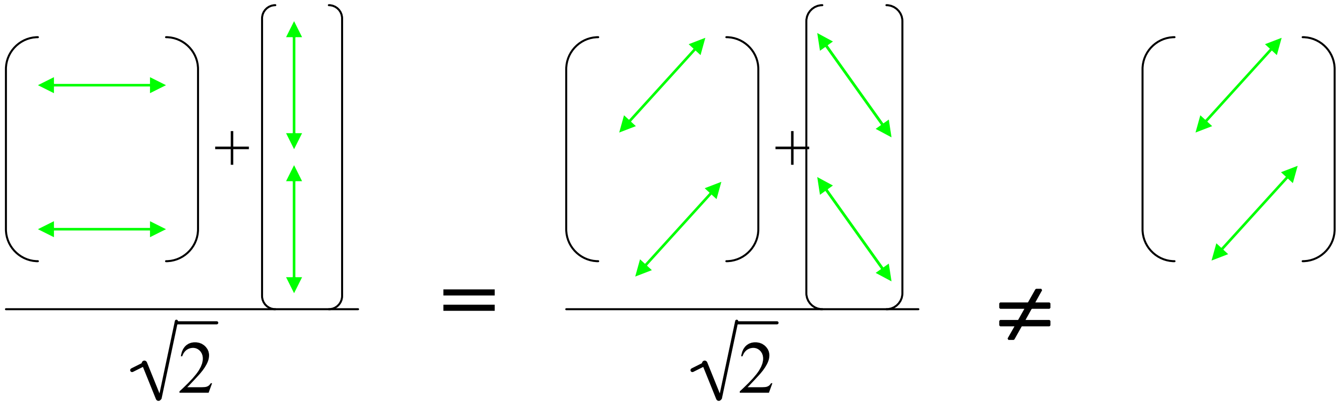


量子版本的Control Not



輸入疊加態=>輸出疊加態

量子糾纏
Quantum Entanglement

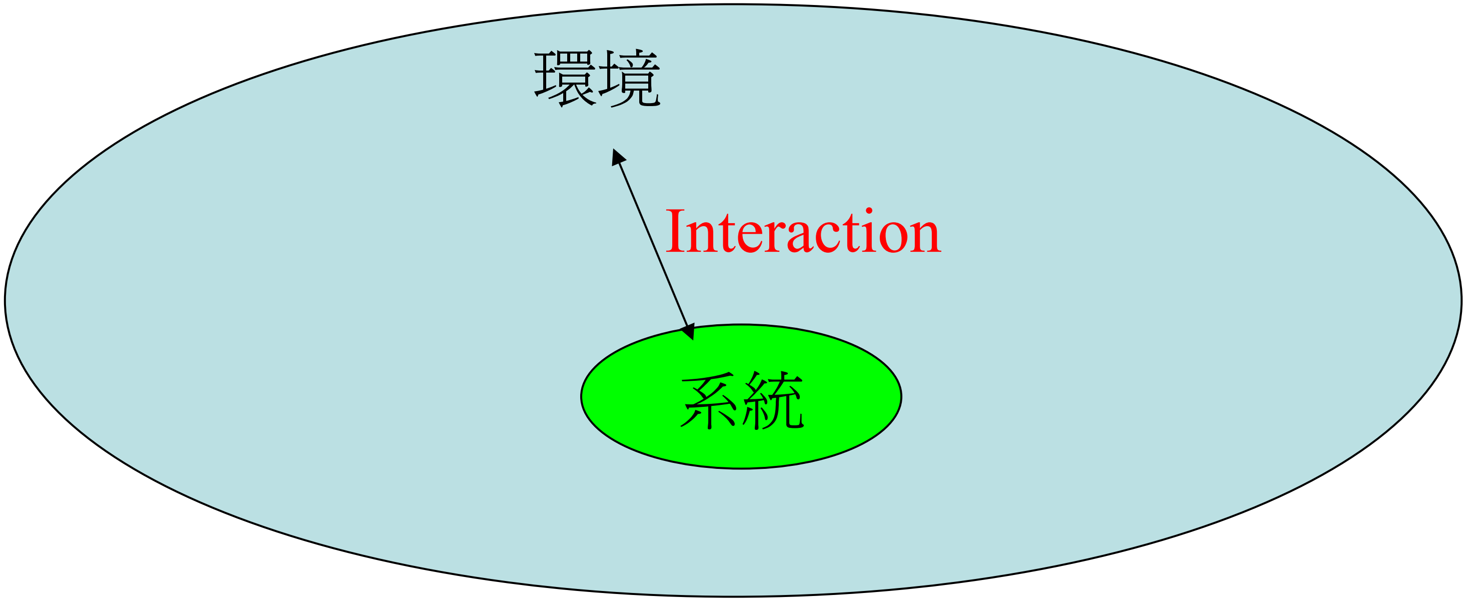


The two-photon may be said to be in a definite state of *sameness* of polarization even though neither photon has a polarization of its own

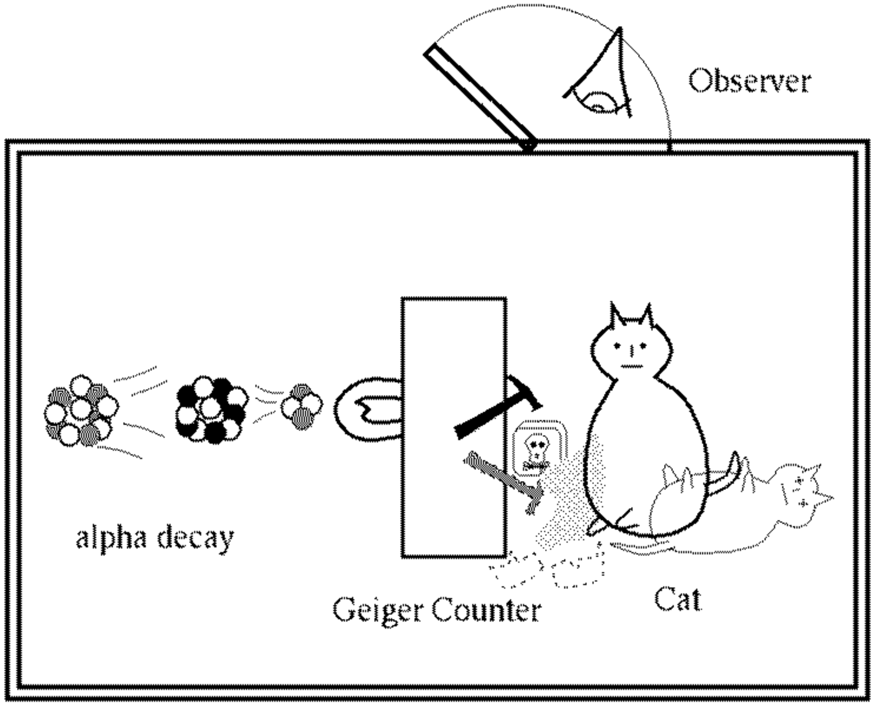
量子電腦的困擾

量子消相干
Quantum De-Coherence

量子糾纏=> 量子消相干過程=> 量子不糾纏=古典物理



Schroedinger's Cat



Before Opening the Box

$$|\text{Whole}\rangle|\text{Alive}\rangle + |\text{Decayed}\rangle|\text{Dead}\rangle$$

After Opening the Box

$$|\text{Whole}\rangle|\text{Alive}\rangle|\text{Buy Kitty Litter}\rangle + |\text{Decayed}\rangle|\text{Dead}\rangle|\text{RIP}\rangle$$

?!?

http://www.lassp.cornell.edu/ardlouis/dissipative/Schrcat.html

量子消相干時間尺度決定了量子資訊處理的可能性

如何對抗量子消相干?

- Active:
  - Quantum error correction
  - Quantum feedback control
  - …….
- Passive:
  - Decoherence-free subspace
  - Bang-bang decoupling
  - …….

- Quantum Error Correction
  - Knowing the error without knowing the state
  - Encoded qubit, Ex: $|0>_L=|000>, |1>_L=|111>$
  - Error detection
  - Error correction

- Threshold for resilient quantum computation
  - $P_{th}\approx 10^{-5}$

Di Vincenzo’s Criteria

- Be a scalable physical system with well-defined qubits
- Be initializable to a simple fiducial state such as |000...>
- Have much longer decoherence times
- Have a universal set of quantum gates
- Permit high quantum efficiency measurements

A Scalable Physical System with Well-defined Qubits

- Well-defined qubits
  - Parameter known, (in principle), for each qubit
  - Coupling to environment and other qubit known
  - Tunable inter qubit interactions (can be relaxed)
  - Higher levels far away (to reduce decoherence)

- Scalable
  - Many qubit can be read out, manupulated individaully
  - Selectivity

Basic Components

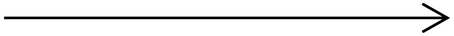
- Get familiar with the basic components in quantum circuit

- Understand the physical and operation meaning of quantum circuit

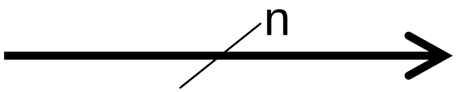
- Be able to read the more complicated quantum circuit and

Basic Components

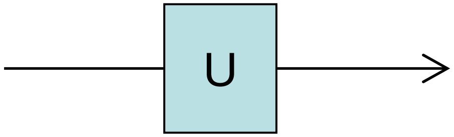
- Qubit
  - Single qubit



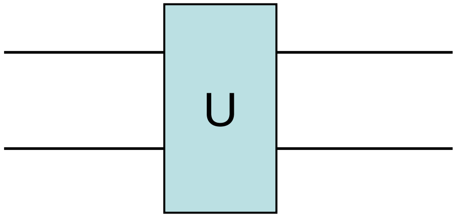
  - Multiple qubit



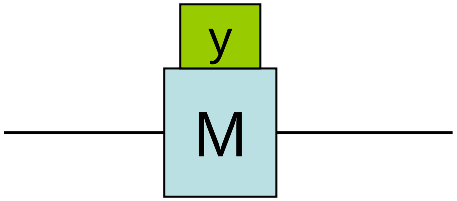
- Quantum operation
  - Single qubit operation



  - Multiple qubit operation



- Quantum measurement



Basic Components

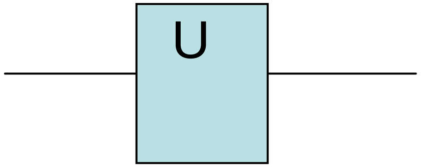
- Doing nothing

$$|\varphi\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

$$|\varphi'\rangle = |\varphi\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

- Single qubit gate

$$|\varphi\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

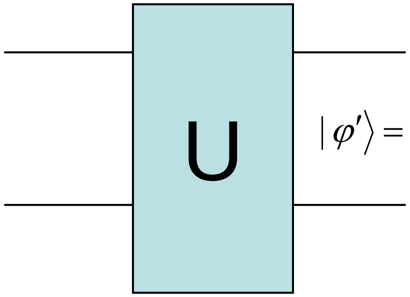


$$|\varphi'\rangle = U|\varphi\rangle = \begin{bmatrix}U_{00} & U_{01}\\ U_{10} & U_{11}\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

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

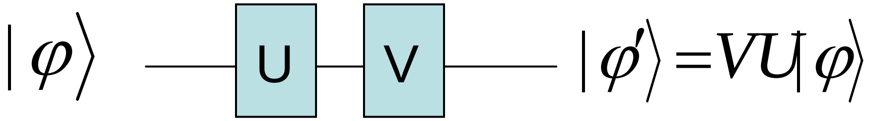
- Two qubits gate

$$|\varphi\rangle = \begin{bmatrix}a_{00}\\ a_{01}\\ a_{10}\\ a_{11}\end{bmatrix}$$

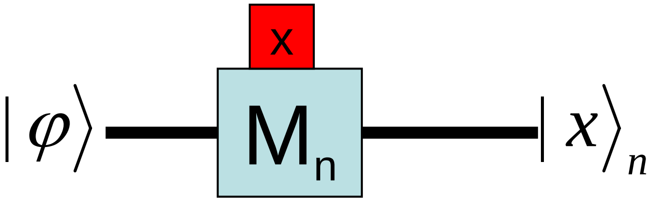


$$|\varphi'\rangle = U\,|\varphi\rangle = \begin{bmatrix}U_{00,00} & U_{00,01} & U_{00,10} & U_{00,11}\\ U_{01,00} & U_{01,01} & U_{01,10} & U_{01,11}\\ U_{10,00} & U_{10,01} & U_{10,10} & U_{10,11}\\ U_{11,00} & U_{11,01} & U_{11,10} & U_{11,11}\end{bmatrix}\begin{bmatrix}a_{00}\\ a_{01}\\ a_{10}\\ a_{11}\end{bmatrix}$$

Basic Components



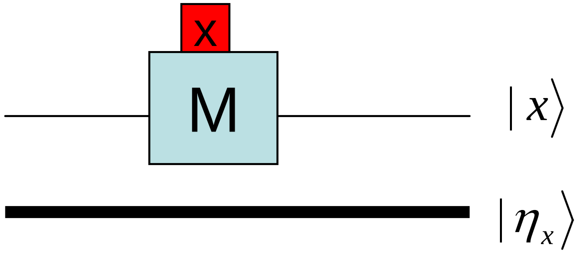
$$|\varphi\rangle \qquad |\varphi'\rangle = VU|\varphi\rangle$$



$$|\varphi\rangle \qquad |x\rangle_n$$

$$|\varphi\rangle = \sum_{0 \le x_n \le 2^n} a_x \, |x\rangle_n \to p(x) = |a_x|^2$$

$$|\varphi\rangle = \begin{array}{l} a_0 \, |0\rangle |\eta_0\rangle \\ + \quad a_1 \, |1\rangle |\eta_1\rangle \end{array}$$

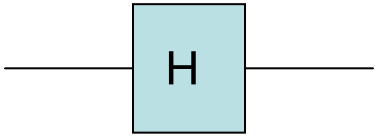


$$|x\rangle \qquad |\eta_x\rangle$$

$$p(x) = |a_x|^2$$

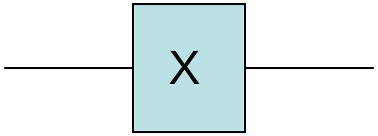
Frequently Used Gates

- Hadamard:



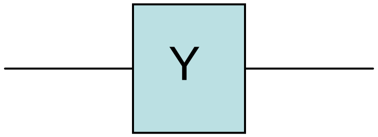
$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Pauli-X:



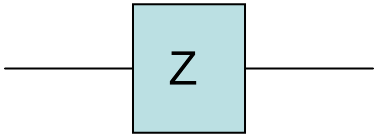
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- Pauli-Y:



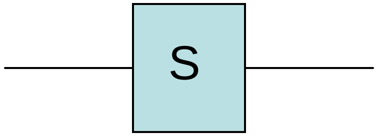
$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

- Pauli-Z:



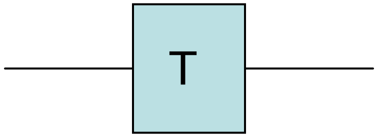
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Phase:



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

- π /8:

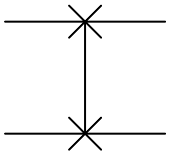


$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

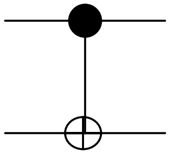
Frequently Used Gates

- SWAP

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



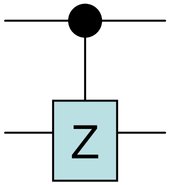
- Controlled-NOT



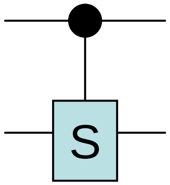
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- Controlled-Z

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



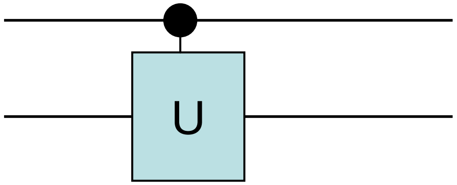
- Controlled-Phase



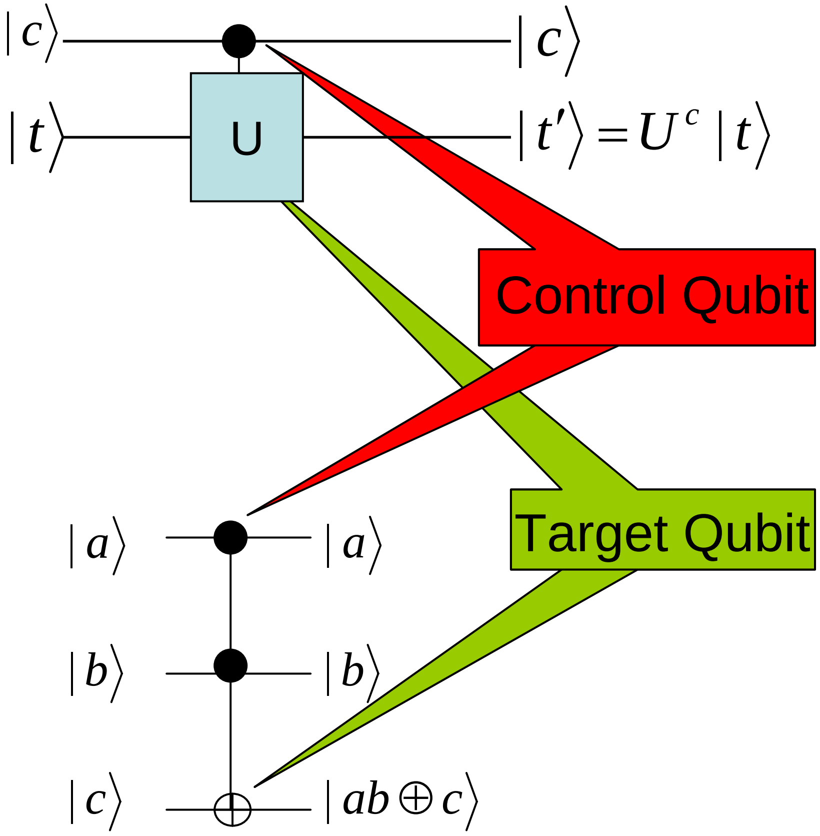
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

- Controlled-U

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{bmatrix}$$



Controlled Operation



$|c\rangle$ $|c\rangle$

$|t\rangle$ U $|t'\rangle = U^c\,|t\rangle$

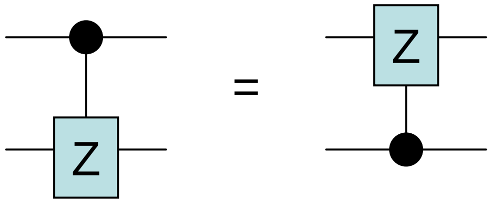
Control Qubit

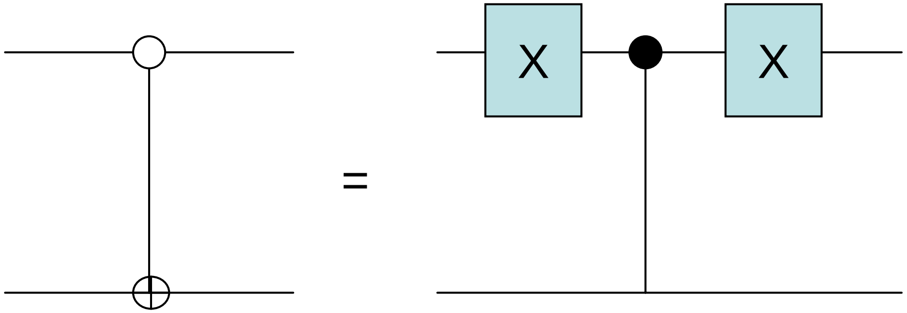
Target Qubit

$|a\rangle$ $|a\rangle$

$|b\rangle$ $|b\rangle$

$|c\rangle$ $|ab \oplus c\rangle$





Apply “NOT” when control=0

Connection to Classical Computation

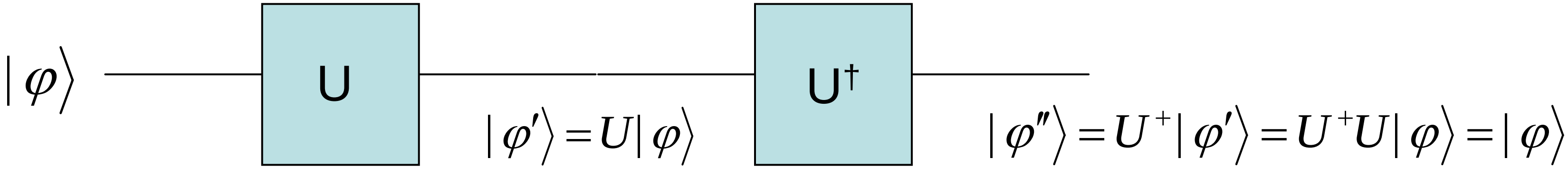
- Classical computation is (in principle) irreversible
- Quantum computation is (in principle) reversible
- Connection? Reversibility? Dissipation?

Energy and Computations

- *Laudauer’s* principle:
  - Suppose a computer erases a single bit of information. The entropy of theenvironment increases by ***at least*** kB*ln2, here kB is Boltzmann’s constant.
- *Reversibly* computation*:*
  - If all computer could be done ***reversibly***, then Landauer’s principle imply no lower bound on the amount of energy dissipated by the computer!

Quantum Gates are Reversible

- For any unitary matrix U, we have U†U=I



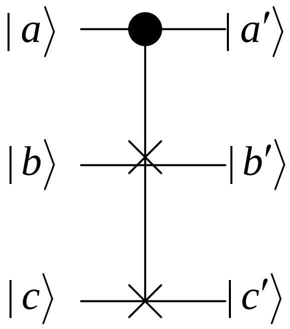
$|\varphi\rangle$

$|\varphi'\rangle = U|\varphi\rangle$

$|\varphi''\rangle = U^{+}|\varphi'\rangle = U^{+}U|\varphi\rangle = |\varphi\rangle$

- Is it possible to simulate classical gate by quantum gate?
  - The answer is, of course, *yes*.

Fredkin Gate

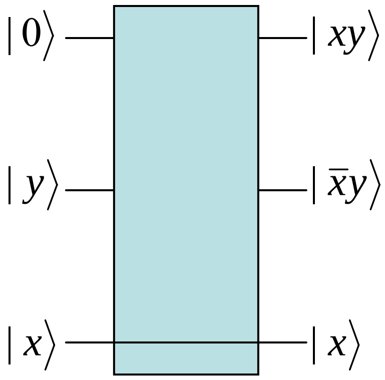


Unitary Matrix

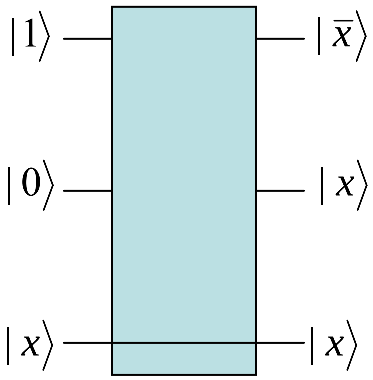
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Truth Table

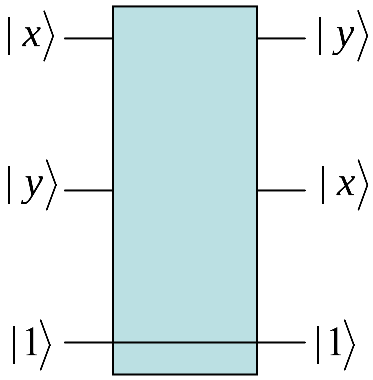
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	1	1



AND

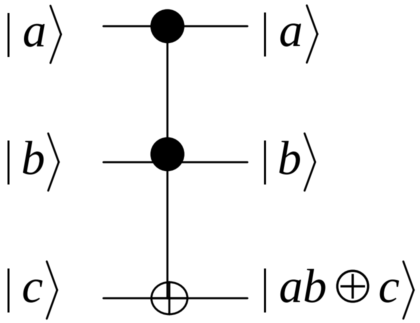


NOT



CROSSOVER

Toffoli gate

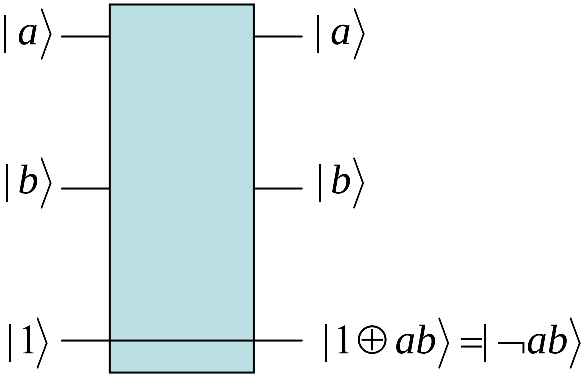


Unitary Matrix

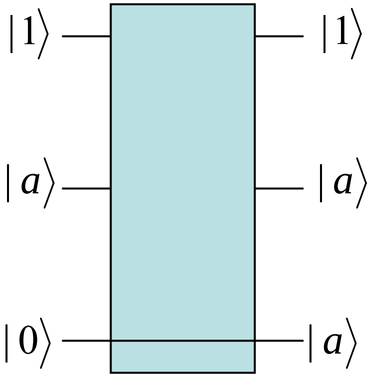
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Truth Table

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0



NAND



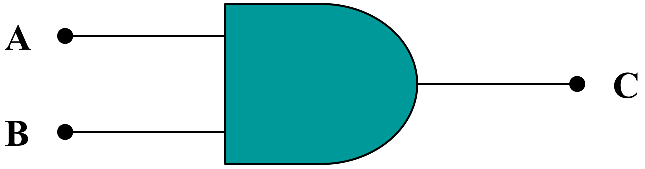
FANOUT

Operations on Qubits - Reversible Logic

- Due to the nature of quantum physics, the destruction of information in a gate will cause heat to be evolved which can destroy the superposition of qubits.

Ex.

The AND Gate



Input		Output
A	**B**	**C**
0	0	0
0	1	0
1	0	0
1	1	1

In these 3 cases, information is being destroyed

- This type of gate cannot be used. We must use ***Quantum Gates***.

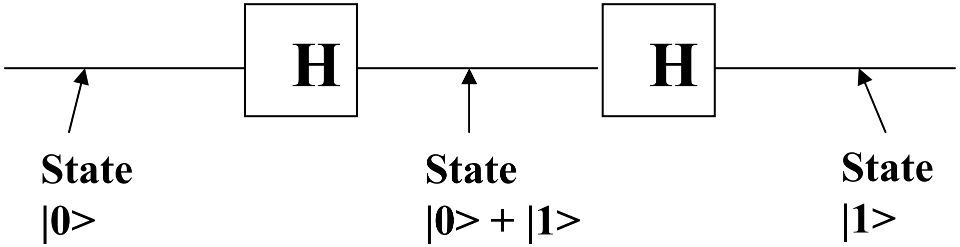
Quantum Gates

- Quantum Gates are similar to classical gates, but do not have a degenerate output. i.e. their original input state can be derived from their output state, uniquely. ***They must be reversible.***

- This means that a deterministic computation can be performed on a quantum computer only if it is reversible. Luckily, it has been shown that any deterministic computation can be made reversible.(Charles Bennet, 1973)

Quantum Gates - Hadamard

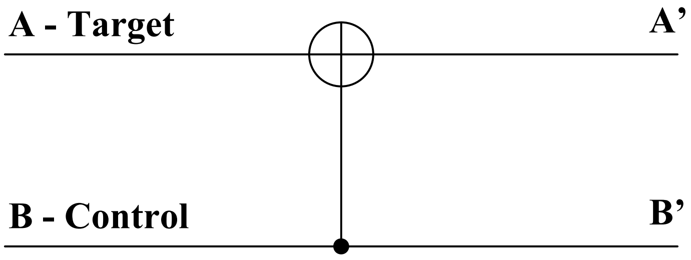
- Simplest gate involves one qubit and is called a ***Hadamard Gate (***also known as a square-root of NOT gate.) Used to put qubits into superposition.



Note: Two Hadamard gates used in succession can be used as a NOT gate

Quantum Gates - Controlled NOT

- A gate which operates on two qubits is called a ***Controlled-NOT (CN) Gate.*** If the bit on the control line is 1, invert the bit on the target line.



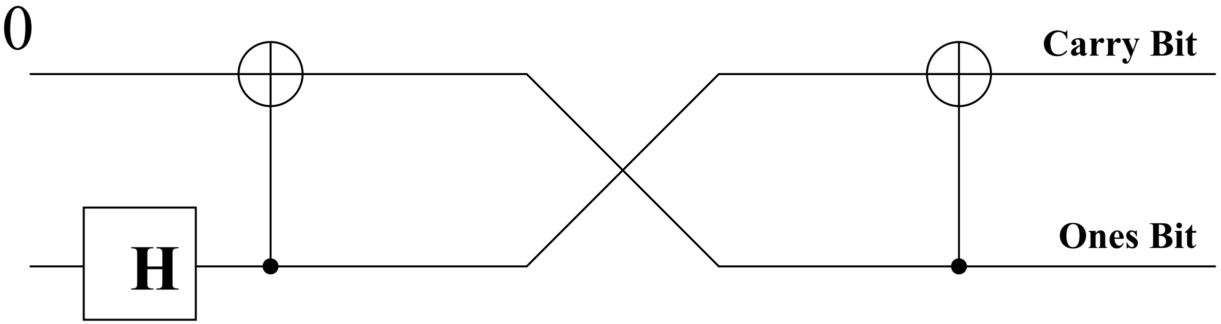
Input		Output	
A	B	A’	B’
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

Note: The CN gate has a similar behavior to the XOR gate with some extra information to make it reversible.

Example Operation - Multiplication By 2

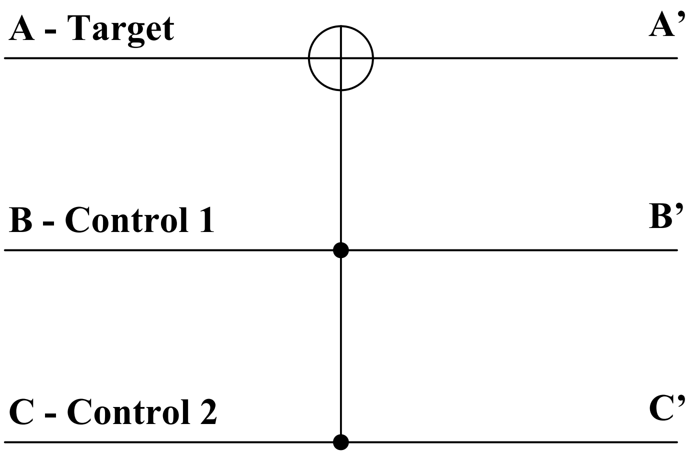
- We can build a reversible logic circuit to calculate multiplication by 2 using CN gates arranged in the following manner:

Input		Output	
Carry Bit	Ones	Carry Bit	Ones Bit
0	0	0	0
0	1	1	0



Quantum Gates - Controlled Controlled NOT (CCN)

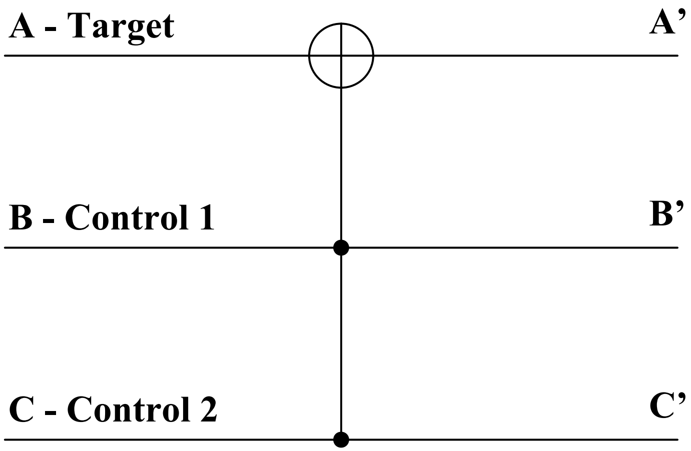
- A gate which operates on three qubits is called a ***Controlled Controlled NOT (CCN) Gate.*** Iff the bits on both of the control lines is 1,then the target bit is inverted.



Input			Output		
A	B	C	A’	B’	C’
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

A Universal Quantum Computer

- The CCN gate has been shown to be a ***universal*** reversible logic gate as it can be used as a NAND gate.



Input			Output		
A	B	C	A’	B’	C’
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

When our target input is 1, our target output is a result of a NAND of B and C.

Quantum Fourier Transform

- Discrete Fourier transform:

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i jk/N}$$

- Quantum Fourier transform:

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle$$

$$\sum_{j=0}^{N-1} x_j |j\rangle \to \sum_{k=0}^{N-1} y_k |k\rangle$$

QFT: Product Representation

$$2^{n/2}\,| j_1,\mathrm{K}\,, j_n\rangle$$

$$= \sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n}\,|k\rangle$$

$$= \sum_{k_1=0}^{1}\Lambda\sum_{k_n=0}^{1} e^{2\pi ij\left(\sum_{l=1}^{n} k_l 2^{-l}\right)}\,|k_l\rangle$$

$$= \sum_{k_1=0}^{1}\Lambda\sum_{k_n=0}^{1} \otimes_{l=1}^{n} e^{2\pi ijk_l 2^{-l}}\,|k_l\rangle$$

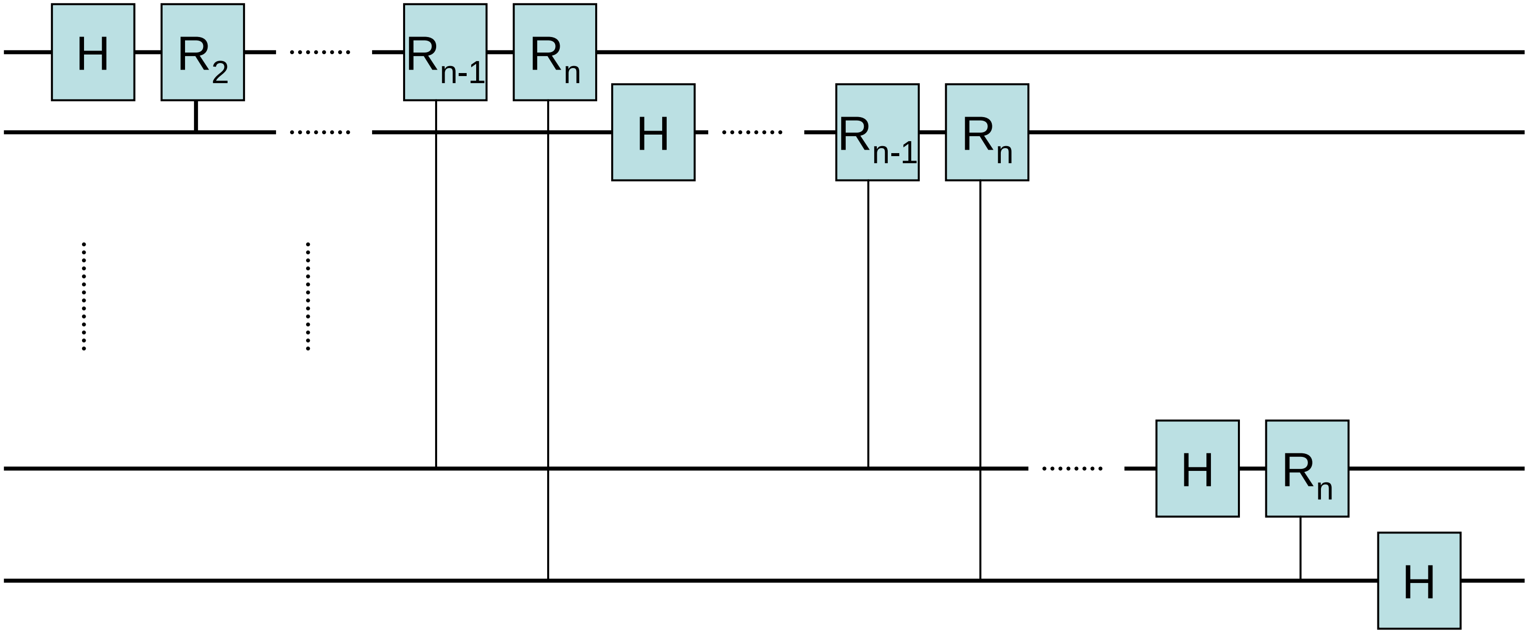
$$= \otimes_{l=1}^{n}\left[\sum_{k_1=0}^{1} e^{2\pi ijk_l 2^{-l}}\,|k_l\rangle\right] = \otimes_{l=1}^{n}\left[|0\rangle + e^{2\pi ij2^{-l}}\,|1\rangle\right]$$

$$\to \left(|0\rangle + e^{2\pi i0.j_n}\,|1\rangle\right)\left(|0\rangle + e^{2\pi i0.j_{n-1}j_n}\,|1\rangle\right)\Lambda\left(|0\rangle + e^{2\pi i0.j_1j_2\mathrm{K}\,j_n}\,|1\rangle\right)$$

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \Lambda + j_n 2^0$$

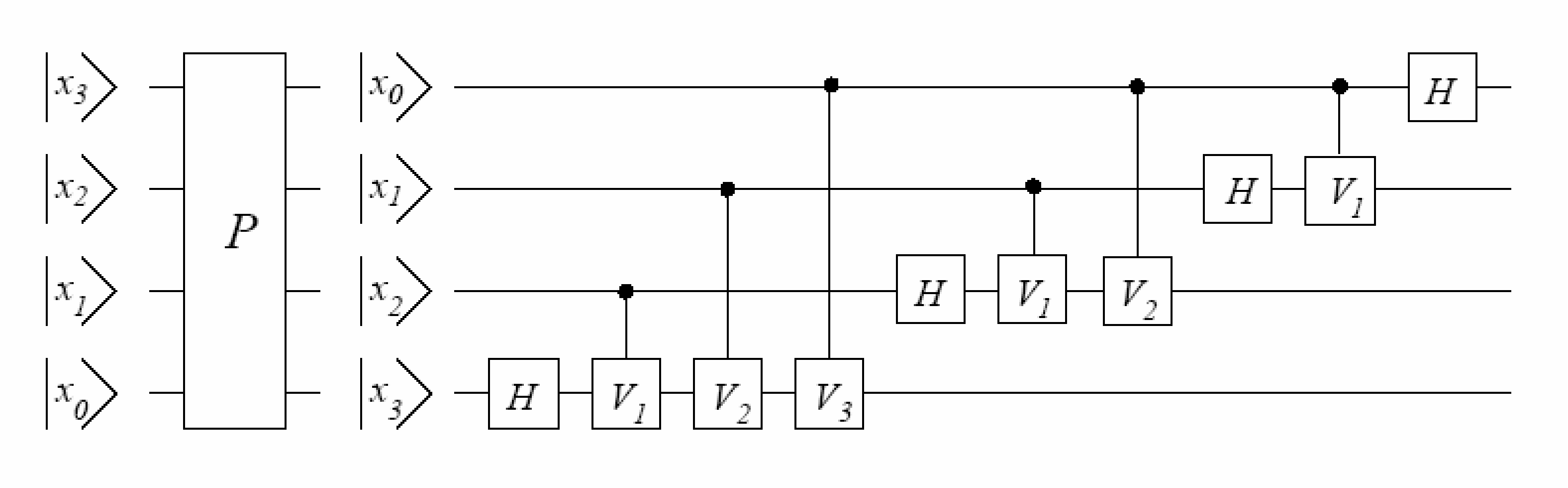
$$0.j_l j_{l+1}\mathrm{K}\,j_m = j_l/2 + j_{l+1}/4 + \mathrm{K} + j_m/2^{m-l+1}$$

Efficient Circuit for QFT



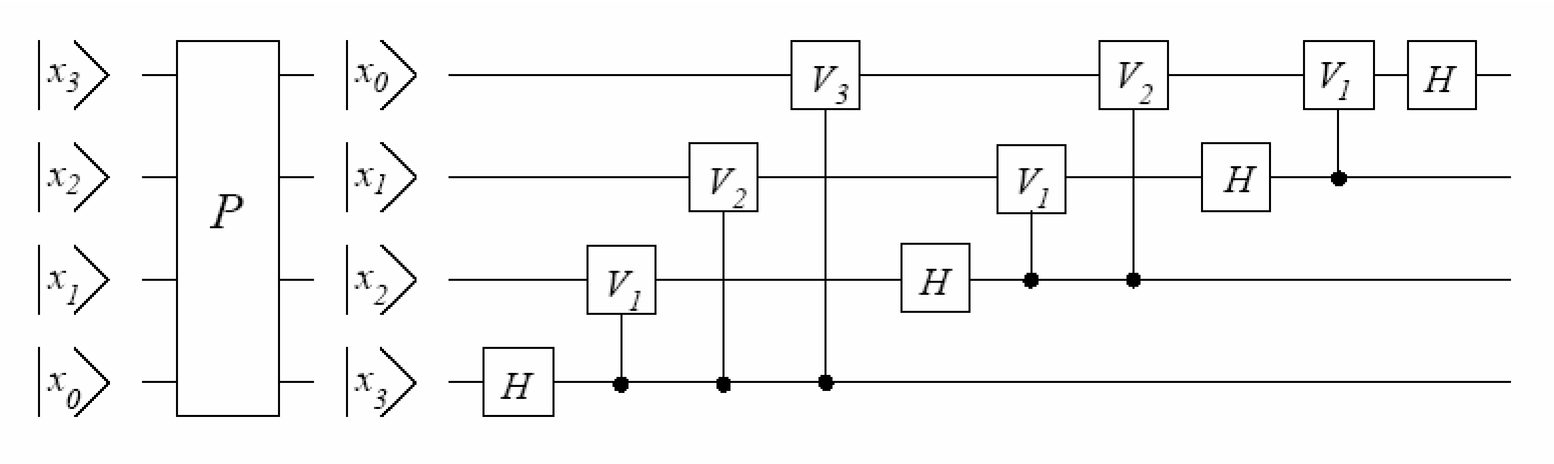
$$R_n = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$$

4-Qubit QFT-1

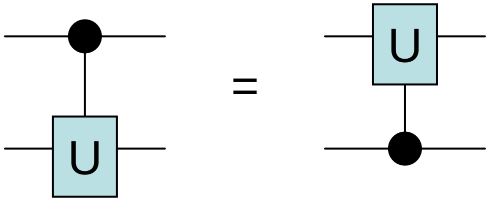


- Qubits representing more significant bits are represented by higher lines

4-Qubit QFT-2

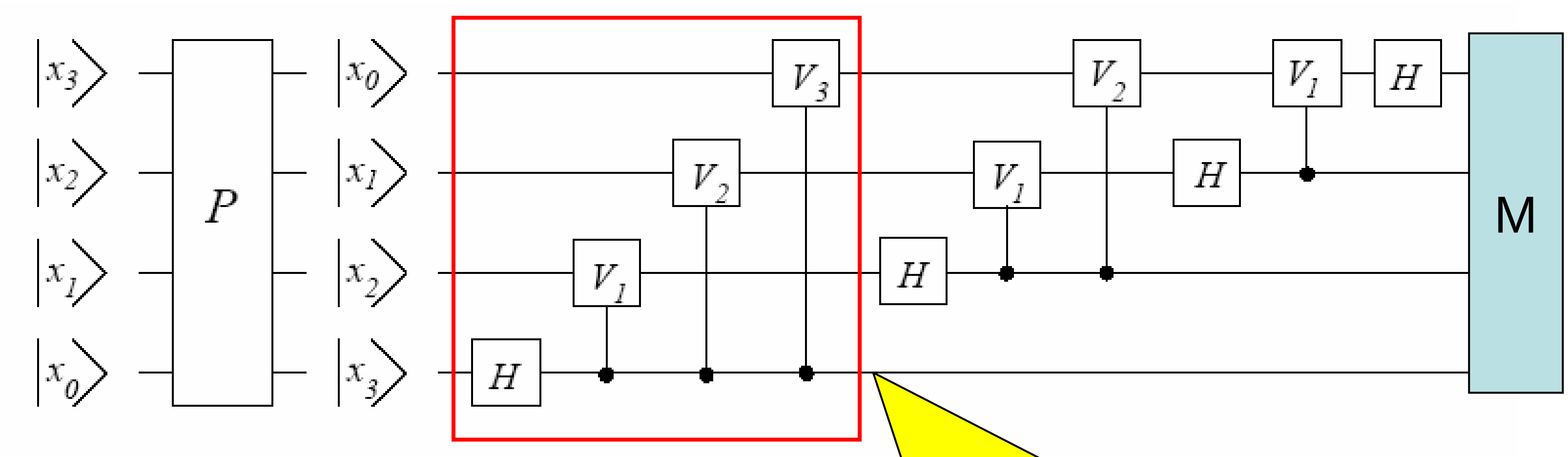


- Use the symmetric property of controlled gate



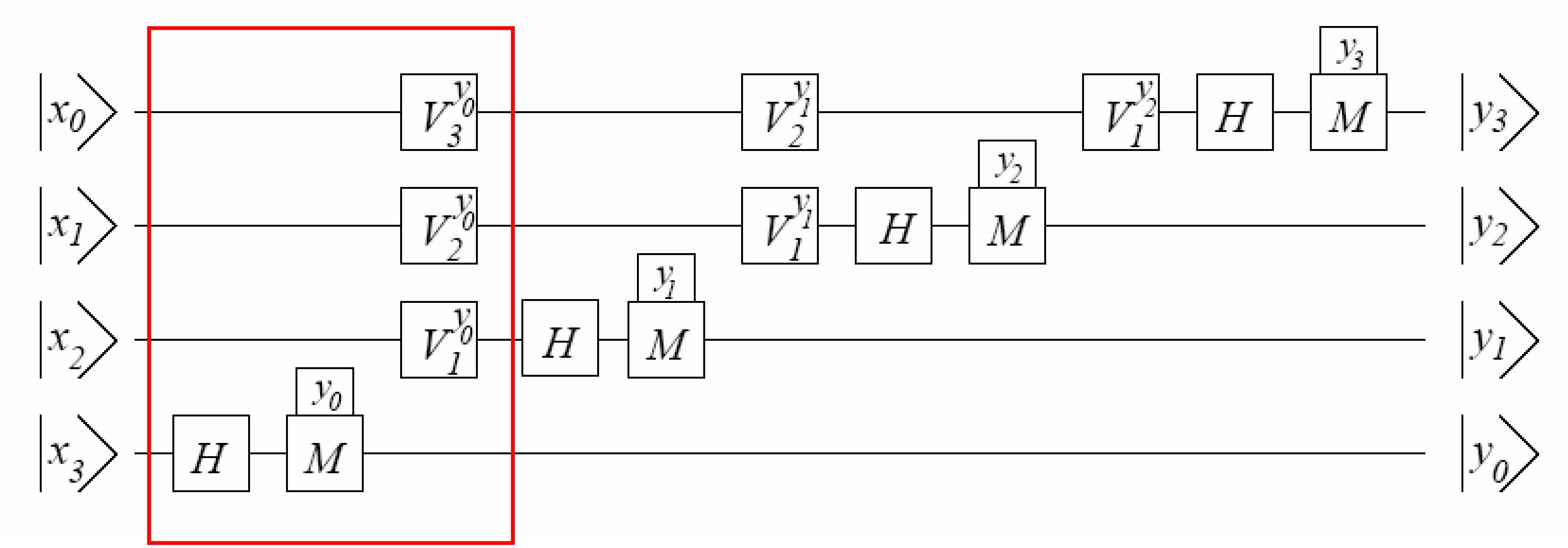
- Now if we do the measurement right after the QFT circuit (period finding)

4-Qubit QFT-2



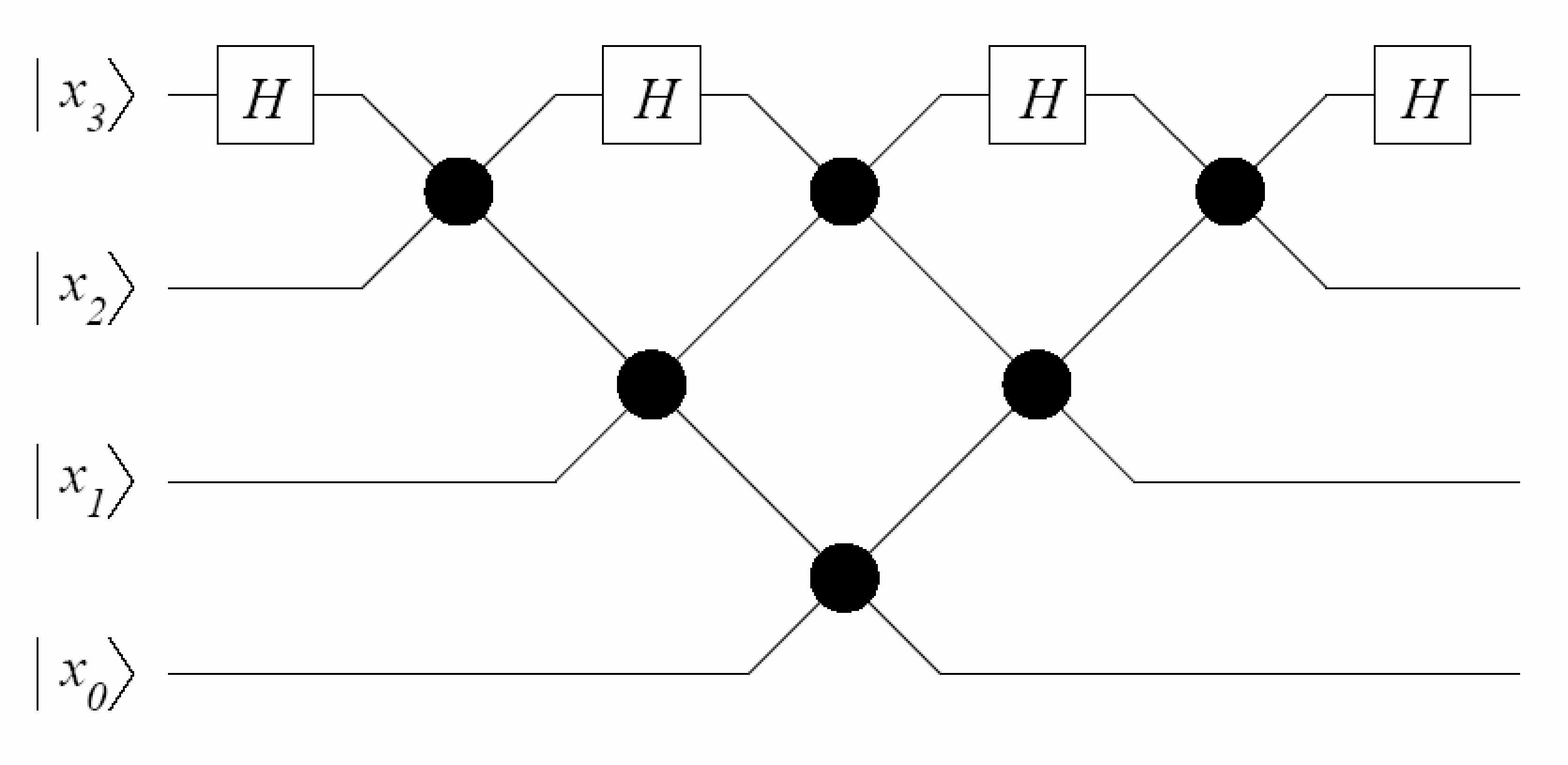
- Nothing further happens until the final measurement

4-Qubit QFT-3



- If y_0=1, apply V_1, V_2, V_3
- If y_0=0, do nothing
- Produce the same state after the red box
- PRL 76, 3228 (1996)

4-Qubit QFT-4



- PRL 76, 3228 (1996)
- Symmetric between control and target

Shor’s Algorithm

- Shor’s algorithm shows (in principle,) that a quantum computer is capable of factoring very large numbers in polynomial time.

The algorithm is dependant on

  - Modular Arithmetic
  - Quantum Parallelism
  - Quantum Order Finding
  - Quantum Fourier Transform

Shor's Algorithm - Periodicity

- An important result from Number Theory:

  $F(a) = x^a \mod N$ is a periodic function

- Choose N = 15 and x = 7 and we get the following:

$$7^0 \mod 15 = 1$$

$$7^1 \mod 15 = 7$$

$$7^2 \mod 15 = 4$$

$$7^3 \mod 15 = 13$$

$$7^4 \mod 15 = 1$$

⋮

- Order of **$x \mod N$** is the least positive integer **r** s.t. **$x^r=1 \mod N$**

Shor's Algorithm - In Depth Analysis

To Factor an odd integer N (Let's choose 15) :

1. Choose an integer *q* such that $N^{2\ 2} < q < 2N^{2}$ **let's pick q=256**
2. Choose a random integer *x* such that GCD(*x*, N) = 1 **let's pick x=7**
3. Create two quantum registers (these registers must also be entangled so that the collapse of the input register corresponds to the collapse of the output register)
   - Input register: must contain enough qubits to represent numbers as large as q-1. **up to 255, so we need 8 qubits**
   - Output register: must contain enough qubits to represent numbers as large as N-1. **up to 14, so we need 4 qubits**

Shor’s Algorithm - Preparing Data

4. Load the input register with an equally weighted superposition of all integers from 0 to *q*-1. **0 to 255**

5. Load the output register with all zeros.

The total state of the system at this point will be:

$$\underbrace{\frac{1}{\sqrt{256}}\sum_{a=0}^{255}}_{\text{Input Register}} |a, \underbrace{000}_{\text{Output Register}}>$$

Note: the comma here denotes that the registers are entangled

Shor’s Algorithm - Modular Arithmetic

6. Apply the transformation x^a mod N to each number in the input register, storing the result of each computation in the output register.

Note that we are using decimal numbers here only for simplicity.

Input Register	7^a Mod 15	Output Register
|0>	7^0 Mod 15	1
|1>	7^1 Mod 15	7
|2>	7^2 Mod 15	4
|3>	7^3 Mod 15	13
|4>	7^4 Mod 15	1
|5>	7^5 Mod 15	7
|6>	7^6 Mod 15	4
|7>	7^7 Mod 15	13

Shor’s Algorithm - Superposition Collapse

7. Now take a measurement on the output register. This will collapse the superposition to represent ***just one*** of the results of the transformation, let’s call this value *c*.

$$|\phi>=|0>|1>+|1>|7>+|2>|4>+|3>|13>+|4>|1>+|5>|7>+|6>|4>+|7>|13>+..$$

$$=(|0>+|4>+..)|1>+(|1>+|5>+..)|7>+(|2>+|6>+..)|4>+(|3>+|7>+..)|13>$$

Our output register will collapse to represent one of the following:

$$|1>, |4>, |7>, \text{ or } |13>$$

For sake of example, lets choose |1>

Shor’s Algorithm - Entanglement

Now things really get interesting !

8. Since the two registers are entangled, measuring the output register will have the effect of partially collapsing the input register into an **equal superposition** of each state between 0 and q-1 that yielded c (the value of the collapsed output register.)

Since the output register collapsed to |1>, the input register will partially collapse to:

$$\frac{\mathbf{1}}{\sqrt{\mathbf{\mathit{64}}}}\ |\mathbf{0}> + \frac{\mathbf{1}}{\sqrt{\mathbf{\mathit{64}}}}\ |\mathbf{4}> + \frac{\mathbf{1}}{\sqrt{\mathbf{\mathit{64}}}}\ |\mathbf{8}> + \frac{\mathbf{1}}{\sqrt{\mathbf{\mathit{64}}}}\ |\mathbf{12}>, \ldots$$

The probabilities in this case are $\frac{\mathbf{1}}{\sqrt{\mathbf{\mathit{64}}}}$ since our register is now in an equal superposition of 64 values (0, 4, 8, . . . 252)

Shor's Algorithm - QFT

$$\left|\Psi\right\rangle = \frac{1}{\sqrt{64}} \sum_{a\in\{0,4,6,\Lambda\}} \left|a\right\rangle\left|1\right\rangle$$

Note: A is the set of all values that 7^{a} mod 15 yielded 1. In our case A = {0, 4, 8, …, 252}

So the final state of the input register after the QFT is:

$$\left|\Psi\right\rangle = \frac{1}{\sqrt{64}} \sum_{a\in\{0,4,6,\Lambda\}} \left|a\right\rangle\left|1\right\rangle \Rightarrow \left|\Psi\right\rangle = \frac{1}{\sqrt{64}}\frac{1}{\sqrt{256}} \sum_{k=0}^{255} e^{\frac{2\pi i * a * k}{256}} \left|k\right\rangle\left|1\right\rangle$$

Shor’s Algorithm - QFT

The QFT will essentially peak the probability amplitudes at integer multiples of *q*/4 in our case 256/4, or 64.

|0>, |64>, |128>, |192>, …

So we no longer have an equal superposition of states, the probability amplitudes of the above states are now higher than the other states in our register. We measure the register, and it will collapse with high probability to one of these multiples of 64, let’s call this value p.

With our knowledge of q, and p, there are methods of calculating the period (one method is the continuous fraction expansion of the ratio between q and p.)

Shor's Algorithm - The Factors :)

10. Now that we have the period, the factors of N can be determined by taking the greatest common divisor of N with respect to $x^{(P/2)} + 1$ and $x^{(P/2)} - 1$. The idea here is that this computation will be done on a classical computer.

We compute:

Gcd($7^{4/2} + 1$, 15) = **5**

Gcd($7^{4/2} - 1$, 15) = **3**

We have successfully factored 15!

Shor’s Algorithm - Problems

- The QFT comes up short and reveals the wrong period. This probability is actually dependant on your choice of *q*. The larger the q, the higher the probability of finding the correct probability.
- The period of the series ends up being odd

If either of these cases occur, we go back to the beginning and pick a new x.

Summary

- Quantum computer is powerful
  - – In some applications
- Quantum computer is not powerful
  - – Hybrid computation needed
- Quantum computation is difficult
  - – Decoherence, decoherence, decoherence
  - – Fight the decoherence
- Quantum computation is fun !
  - – Deeper understanding of QM