PORTLAND QUANTUM LOGIC GROUP

Tutorial 2

Towards Quantum Logic Synthesis

Quantum versus reversible computing

- **Quantum Computing** is a coming revolution – after recent demonstrations of quantum computers, there is no doubt about this fact. They are reversible.
- Top world universities, companies and government institutions are in a race.
- **Reversible computing** is the step-by-step way of scaling current computer technologies and is the path to future computing technologies, which all happen to use reversible logic.
  - DNA
  - biomolecular
  - quantum dot
  - NMR
  - nano-switches
- In addition, Reversible Computing will become mandatory in any technology, because of the necessity to decrease **power consumption**

Quantum Conservative Reversible Gates

Reversible computation:

- The work on classical, reversible computation has laid the foundation for the development of quantum mechanical computers (Feynman)
- On a quantum computer, programs are executed by unitary evolution of an input.
- Evolution of the state of the system.
- All unitary operators **U** are invertible with $\mathbf{U}^{-1} = \mathbf{U}^{\dagger}$
- Thus we can always ``uncompute'' (reverse the computation) on a quantum computer.

Qubits and Quantum Registers

Quantum Bits and Quantum Logic

- Classical bits are either 0 or 1
- Quantum bits "qubits" are in **linear superposition** of **| 0>** and **| 1>**
- Quantum logic "gates" process (i.e. entangle) qubits
  - Manipulate linear superpositions of states
  - Interfere states with other states
- Computation is completely reversible (no information lost), barring measurements and decoherence
  - All quantum logic gates are reversible:
    - so understanding synthesis of reversible circuits is a must to synthesize quantum circuits.

A Qubit

- A quantum-mechanical gate is **strange**.
- The strangeness goes to the very root of the quantum-computational process, to the bits themselves.
- To emphasize the unconventional nature of quantum bits they are called **qubits.**
- **It is not true** that the qubit has some intermediate value between 0 and 1.
- Rather, the ***qubit is in both the 0 state and the 1 state at the same time***, to varying extents.
- **When the state of the qubit is eventually observed or measured, it is invariably either 0 or 1.**

A Qubit: Schroedinger's Cat

- If the cat was dead(zero state) it will always be regarded as such because Quantum Theory does not bring things back to life.
- If the cat was alive (one state) then it will remain that way until it is put back into the box and the device is restarted
  - (and returns to the superposition of states, being both alive and dead).

Qubits and Quantum Registers

Classical bit

Qubit

superposed

0 or 1

0 or 1 or 0 1

Classical register

Qubit register

010

000 or 001 or 010 or 011 or 100

or 101 or 110 or 111

Quantum Superposition and Quantum Parallelism

- Linear superposition of coexisting possibilities in the quantum world
- Every measurement "collapses" these possibilities

States:

A or B

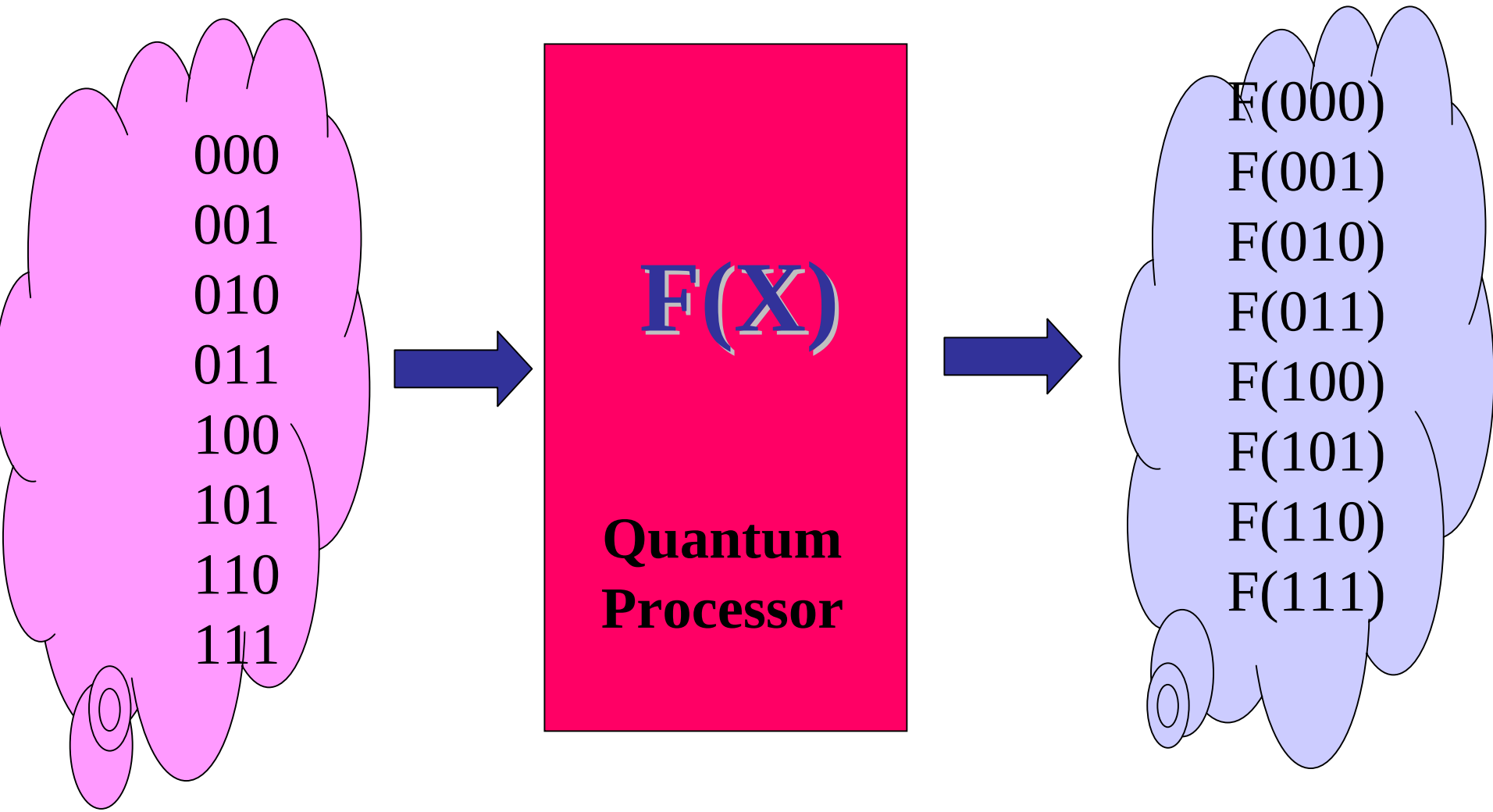
$c_A\,|A\rangle + c_B\,|B\rangle$

- **Measurement** of quantum system:
  - yields state $|A\rangle$ with probability $|c_A|^2$
  - yields state $|B\rangle$ with probability $|c_B|^2$

Probabilities satisfy standard laws (sum is one)

Quantum Parallel Processing

Uses qubits and superposition, therefore it is naturally parallel



•Why is this practically important?

- **Qualitatively different type of computation.**
- **Different computational complexity**
- ***Some*** **NP complete problems can be solved in polynomial time**
- **More efficient use of physical resources**
- **…**

Wave of probability

- **Uncertainty is described mathematically by a wave of probability which expands to fill the space of all possible states**

- **When the box with Schroedinger's cat is opened this wave of probability collapses into one single state**
  - **Wave of probability and probability amplitudes are useful ways of thinking.**

- *After the box is opened, the cat cannot be returned to its original state.*
- *The cat in the box before it has been opened is our **qubit**, **having both states**, dead **and** alive.*
- *Measurement is like a window between quantum world with probability amplitudes and standard world with probabilities of events.*

Elementary Quantum Notation

Elementary quantum notation:

- A simple quantum system is the two-level **spin-1/2 particle**.
- Its **basis states** are: **spin-down** $|\downarrow\rangle$ and **spin-up** $|\uparrow\rangle$.
- **We** relabel them to represent binary zero and binary one:
  - $|0\rangle$ **will be for 0**
  - $|1\rangle$ **will be for 1**
- The **state of such a single particle (qubit)** is described by the wavefunction

$$\Psi = \alpha\,|0\rangle + \beta\,|1\rangle .$$

- **α and β are amplitudes of probability** .
- They are in general **complex numbers.**

Elementary quantum notation:

α and β are amplitudes of probability .

The amplitude associated with a state determines the *probability that the qubit will be found in that state*.

The squares of the complex coefficients $|\alpha|^2$ and $|\beta|^2$ represent the probabilities for finding the particle in the corresponding states.

Quantum States

- We already represented quantum states and superpositions of quantum states using a **notation called a ket notation**

  "| >"

- In general the amplitudes are **complex numbers** (with both a real and an imaginary part)
  - but in some examples amplitudes are just positive and negative real numbers.

Elementary quantum notation:

• Generalizing this to a set of **k** spin- 1/2 particles there will be 2^k basis states:

  • these are quantum mechanical vectors that span the Hilbert space

  • for instance, they correspond to the 2^k possible bit-strings of length **k**.

• For example, $|13\rangle = |1101\rangle = |\uparrow\uparrow\downarrow\uparrow|$ is such a state for **k=4**.

- The dimensionality of the **Hilbert space** grows exponentially with **k**.
- Quantum computations make use of this enormous size of even the smallest values of k.

Quantum gates and circuits

• Changes occurring to a quantum state vector can be **modeled** using a *quantum circuit*.

•Quantum circuit is like a standard circuit:

- it has wires and elementary gates,
- it processes pairs of complex number instead of bits.

• We describe a basic set of quantum gates.

• Remember that they are only models:

- there are no wires and no logic gates such as EXOR in quantum world,
- it is only our notation to simplify our thinking.

Quantum Gates are Reversible

- Now that we understand qubits, we want to design quantum gates, the simplest processors of qubits to qubits.
- In designing gates for a quantum computer, certain constraints must be satisfied.
  - In particular, the matrix of transition amplitudes must be unitary, which implies, roughly speaking, that it conserves probability:
  - The sum of the probabilities of all possible outcomes must be exactly 1.
- A consequence of this requirement is that any quantum computing operation must be reversible.
- Because of these requirements, the reversible quantum gates must have ***the same number of inputs and outputs.***
  - ***It was not required for optical gates!***

single qubit transformations

- Mathematically, *single qubit transformations* are described by **SU(2)** matrices.
- A continuous range of rotations is possible **in principle.**
- But, for quantum computation, only finitely many rotation angles are necessary.
- It has been shown that a single rotation of *nearly any angle* is sufficient to allow efficient generation of an arbitrary qubit rotation angle to a **precision good enough** for the known quantum algorithms to work.

Logic gates for quantum bits:

- 1. The case of a single quantum bit.
- We represent the states $|\downarrow\rangle$ and $|\uparrow\rangle$ (i.e. $|0\rangle$, and $|1\rangle$) as the vectors $\binom{1}{0}$ and $\binom{0}{1}$, respectively.
- Then the most general unitary transformation corresponds to a 2×2 matrix of the form

$$U_\theta \equiv \begin{bmatrix} e^{i(\delta+\sigma+\tau)}\cos(\theta/2) & e^{-i(\delta+\sigma-\tau)}\sin(\theta/2) \\ -e^{i(\delta-\sigma+\tau)}\sin(\theta/2) & e^{i(\delta-\sigma-\tau)}\cos(\theta/2) \end{bmatrix}$$

where we typically take $\delta = \sigma = \tau = 0$ [14].

14 A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin and H. Weinfurter, ``Elementary gates for quantum computation," submitted to Phys. Rev. A 1995.

Logic gates for quantum bits:

$$U_\theta \equiv \begin{bmatrix} e^{i(\delta+\sigma+\tau)}\cos(\theta/2) & e^{-i(\delta+\sigma-\tau)}\sin(\theta/2) \\ -e^{i(\delta-\sigma+\tau)}\sin(\theta/2) & e^{i(\delta-\sigma-\tau)}\cos(\theta/2) \end{bmatrix}$$

$$\delta = \sigma = \tau = 0$$

leads to:

$$\mathbf{U}_\Pi = \begin{bmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \Bigg|_{\Theta = \Pi}$$

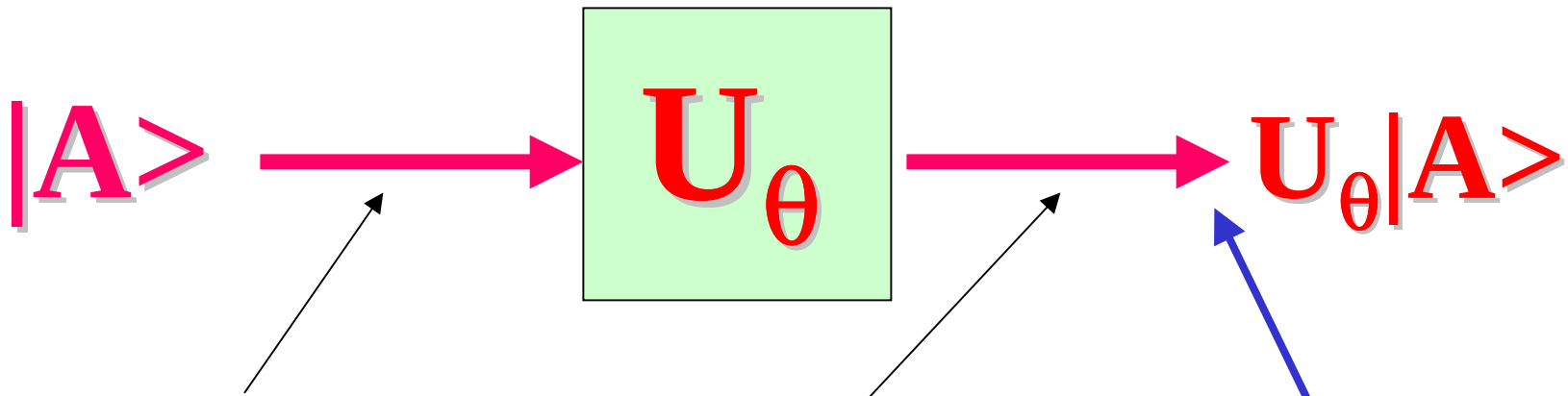
$$\mathbf{U}_\Pi = \begin{bmatrix} \cos(90) & \sin(90) \\ -\sin(90) & \cos(90) \end{bmatrix} \quad \mathbf{U}_\Pi = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Logic gates for quantum bits:

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

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

Schematic of the quantum circuit diagram for a **one-bit gate**.



The lines represent single quantum bits

(such as a **spin-1/2 particle**).

after qubit has ``passed'' through this circuit it comes out in the state $U_\theta|A>$.

Logic gates for quantum bits:

- Using this operator we can **flip bits**:

$$U_{\Pi}|0\rangle = -|1\rangle \qquad U_{\Pi}|1\rangle = |0\rangle$$

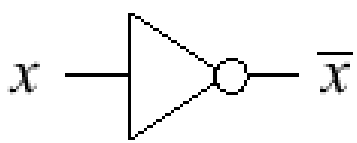
•The extraneous sign represents a phase factor that does not affect the logical operation of the gates and may be removed if we wish, *now or at a later stage*.

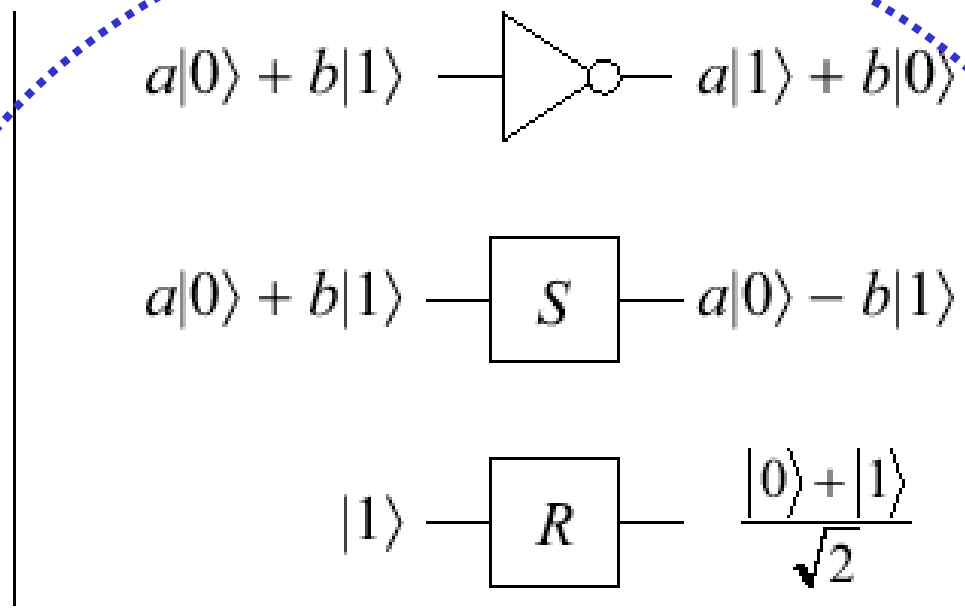
•Such one-bit computations are illustrated schematically as a quantum circuit in the previous slide

D. P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995).

Single qubit gates

- Consider the class of ***single bit gates***. Classically, the only non-trivial member of this class is the not gate, whose operation is defined by its truth table, in which 0 --> 1 and 1 --> 0.





Single bit logic gates.

Single qubit logic gates.

qubit **not gate**

- Qubit **not gate** is defined by its *unitary operator*

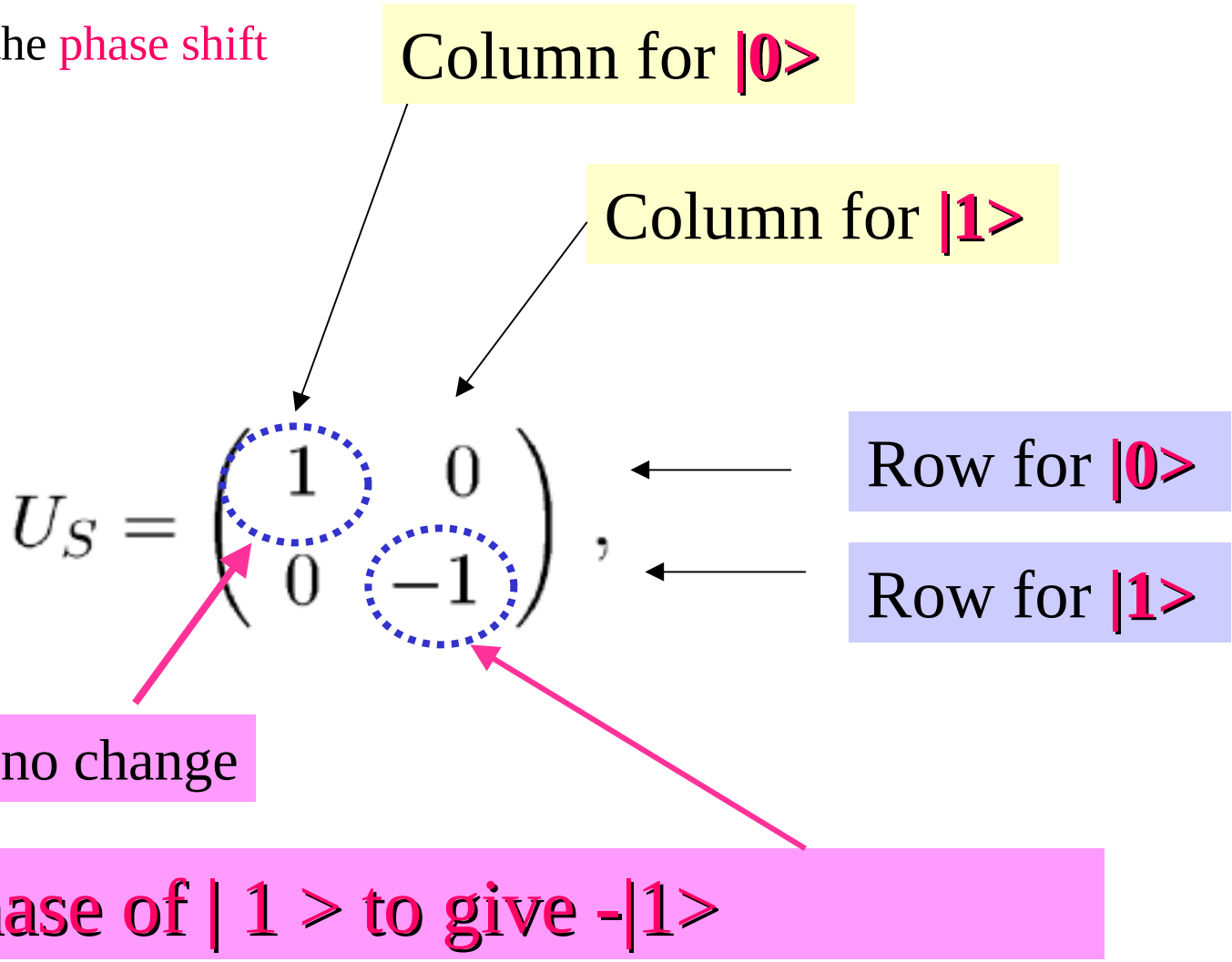
$$U_{not} = \begin{array}{c} \begin{matrix} |0\rangle & |1\rangle \end{matrix} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{matrix} |0\rangle \\ |1\rangle \end{matrix} \end{array} ,$$

where (much like a classical truth table) the two columns refer to the inputs (| 0> and | 1 >) and the two rows the outputs.

- The transform must be unitary to **preserve the norm of the state**.
- Observe, that there are many additional non-trivial single qubit gates.

Phase shift **gate**

- Another *unitary operator*
- **Important *one is*** the phase shift



the Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

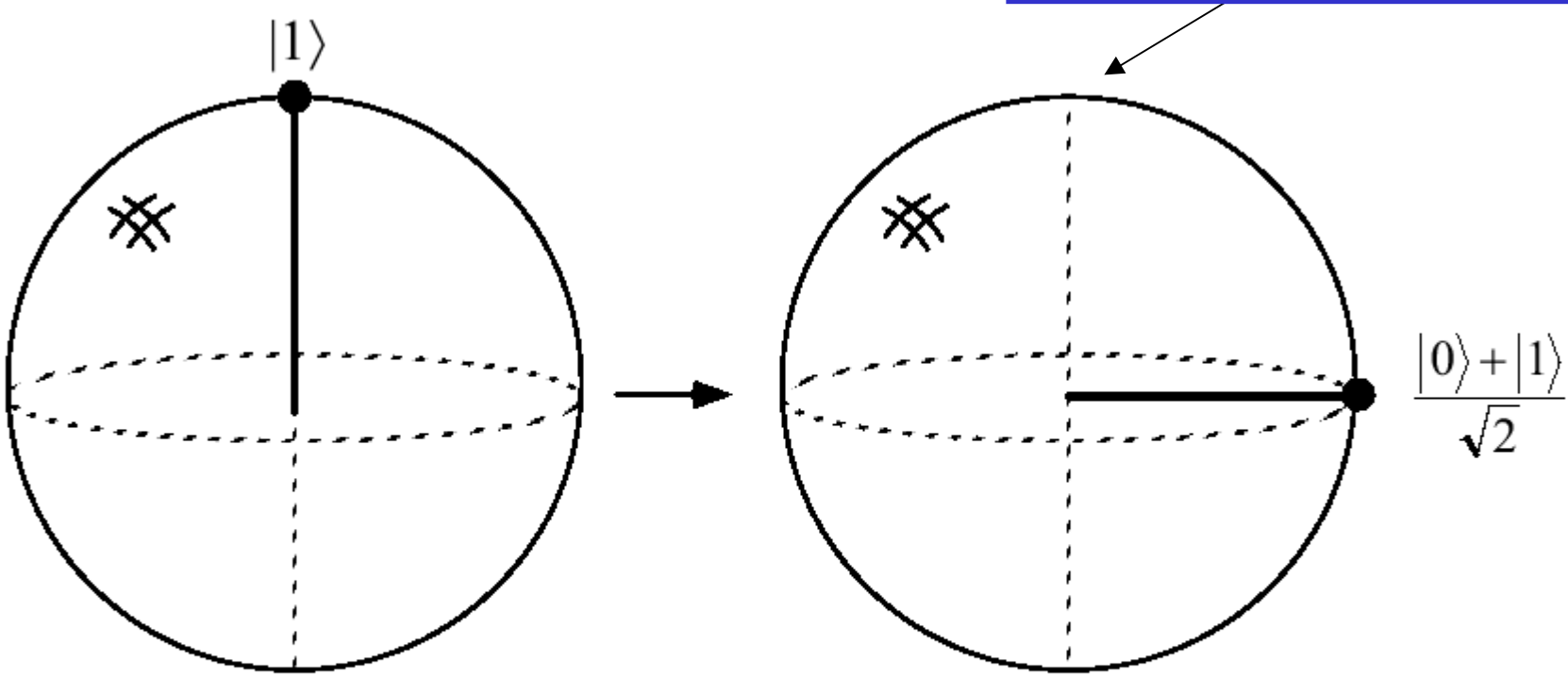
- This gate is also known as the ***"square-root of not "*** gate.
- Its action can be visualized as being similar to rotating the qubit sphere about the **y** axis by **90°**
- This shows how a ***definite state*** like |1> can be transformed by H into the **superposition state**

$$[|0\rangle - |1\rangle] / (\sqrt{2})$$

which gives 0 or 1 with equal probability when measured along the computational basis.

Square Root of NOT

Useful concept, a **cubit sphere**



Visualization of square root of not logic gate on the qubit sphere

infinitely many single qubit gates

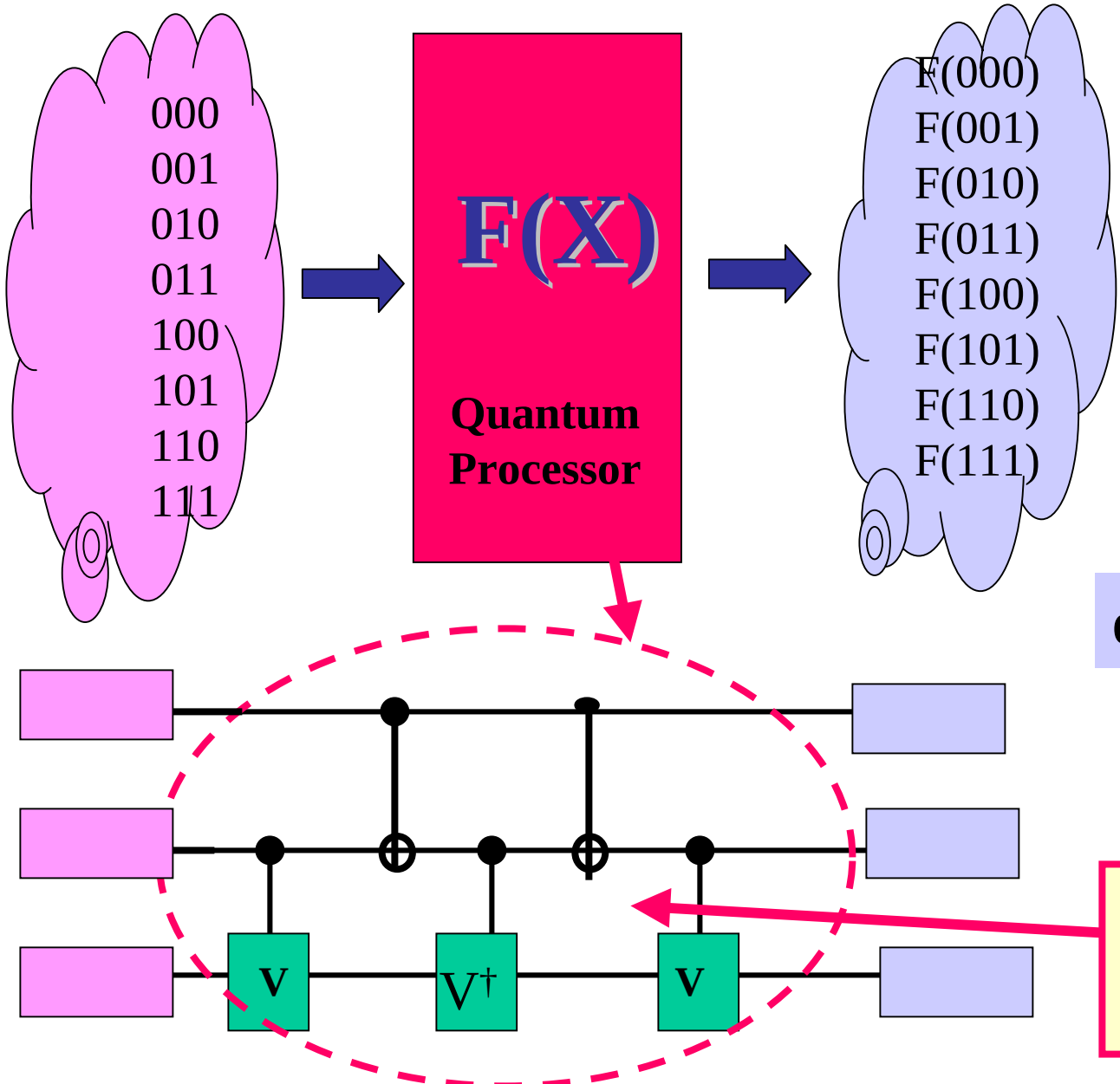
- **All of which can be generated from rotations,**

$$U_R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

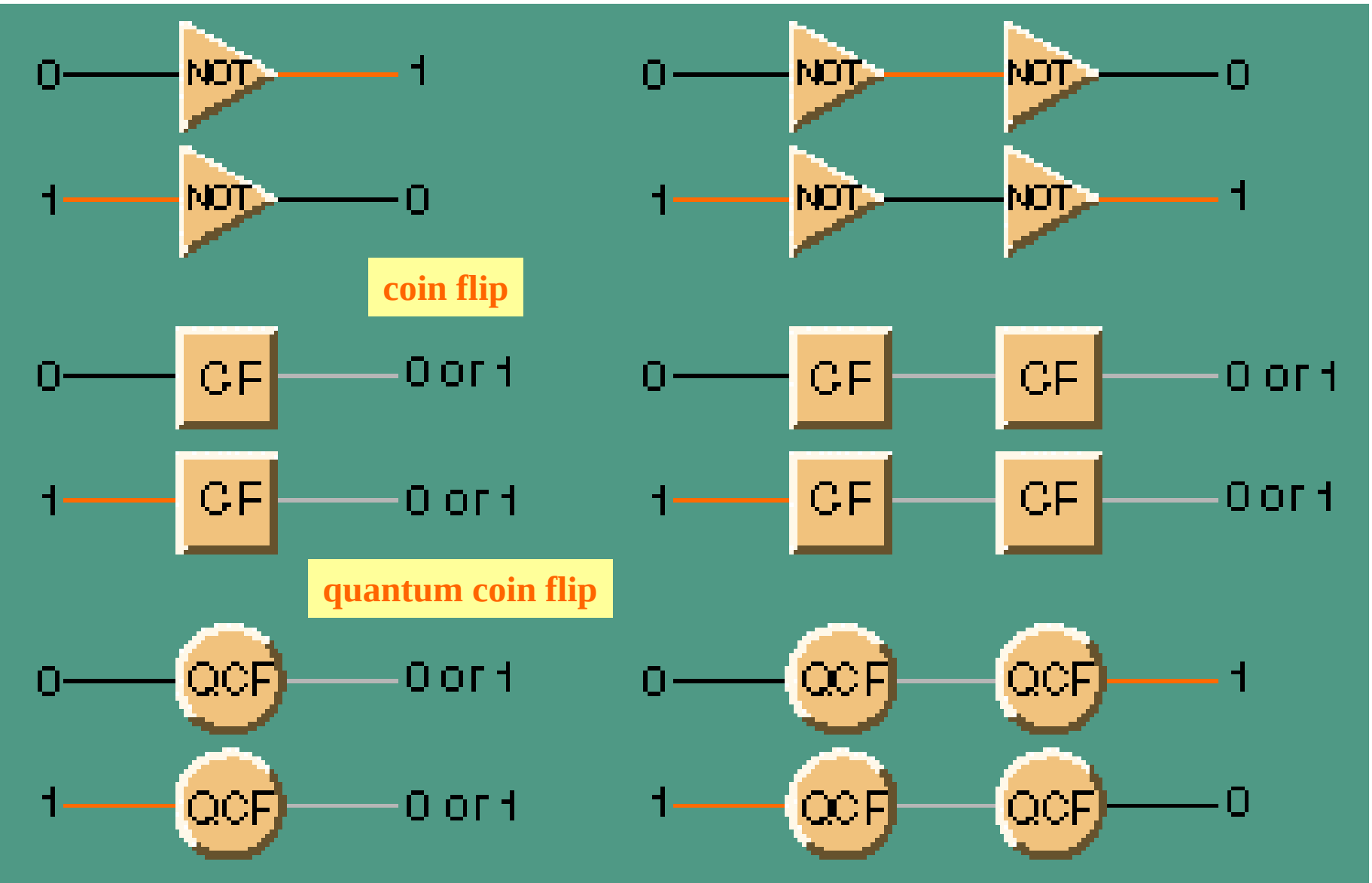
and phase shifts,

$$U_P(\phi_1, \phi_2) = \begin{pmatrix} e^{i\phi_1} & 0 \\ 0 & e^{i\phi_2} \end{pmatrix}.$$

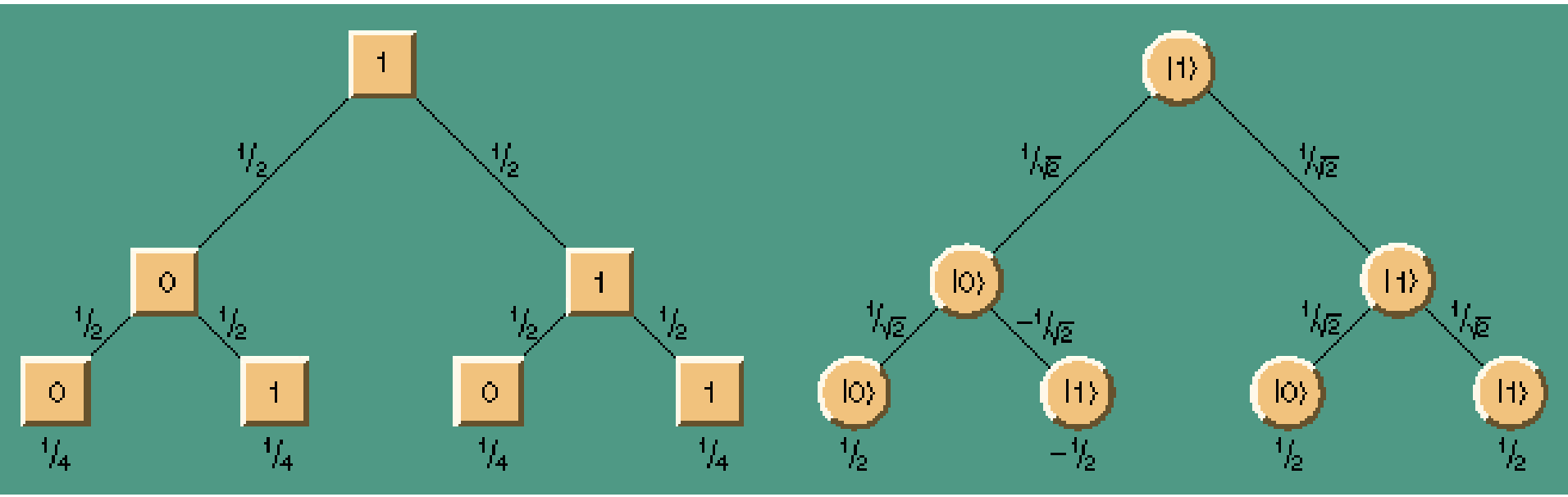
Quantum Networks



Quantum Gates: Not, Quantum Coin Flip



Quantum Interference



Quantum Interference

- For **Quantum Coin Flip** (QCF) gates the analysis is in terms of amplitudes instead of probabilities.
- The first QCF gate transforms the initial |1> state into a **|0>** state with an amplitude of $1/\sqrt{2}$
  - (we assume for simplification that amplitude is a real, not complex number)
- The second QCF gate produces a final **|0>** state with amplitude of $1/\sqrt{2}$
- Multiplying these component amplitudes (just as one would multiply probabilities) yields an overall amplitude of 1/2 for the computational path **|1> --> |0> --> |0>.**

Quantum Interference

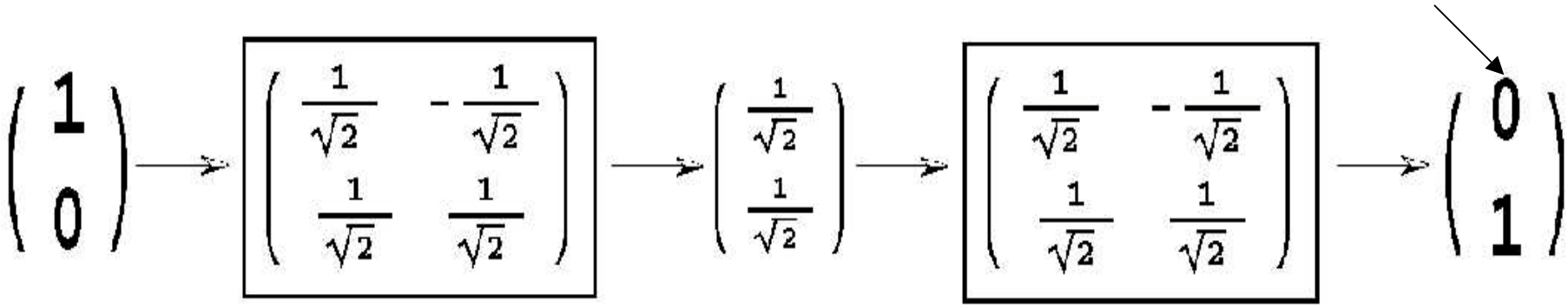
- The amplitude is the same for the paths |1> --> |1> --> |0> and |1> --> |1> --> |1>.
- In the case of the path |1> --> |0> --> |1>, however, the result is different.
- This is because the transition from |0> to |1> has an amplitude of **$-1/\sqrt{2}$**
- The total amplitude for this path is **-1/2.**

Quantum Interference

- In the absence of interference, this change of sign would still have no effect on the outcome of an experiment:
  - **Squaring the absolute value of each amplitude would yield four path probabilities of 1/4, which would sum to a probability of 1/2 for the |0> final state and 1/2 for the |1> final state.**
- Because of interference, however, the two paths leading to the |1> state, with amplitudes of 1/2 and -1/2, cancel each other out, whereas the |0> paths, both with amplitudes of 1/2, sum to yield a total amplitude (and also a total probability) of 1.

The “Square Root of NOT”

Negated input!!



Thus the operation of two QCF gates can be described as above

Because the square of QCF is a NOT, this gate is called Square Root of NOT

- Random bit if measured after one pass
- NOT operation if measured after second pass
- **THIS IS STRANGE!**

Let us check

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

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

Let us check

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

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

Let us check

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

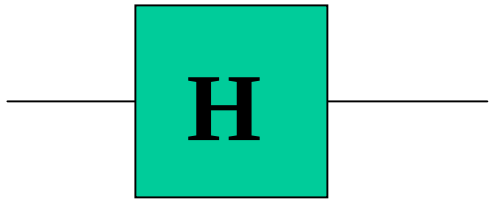
The square root of NOT.

- There is something decidedly counterintuitive about these results.
- Passing a signal through one QCF gate randomizes it, yet putting two QCF gates in a row yields a deterministic result.
- It is as if we had invented a machine that first *scrambles eggs and then unscrambles them*.
- There is **no analogue** of this machine in the more familiar world of **classical physics**.

Most important Quantum Gates and their Matrices

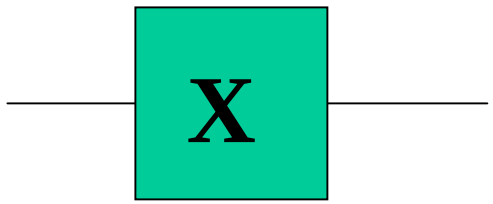
Other 1*1 unitary gates (quantum)

Hadamard



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

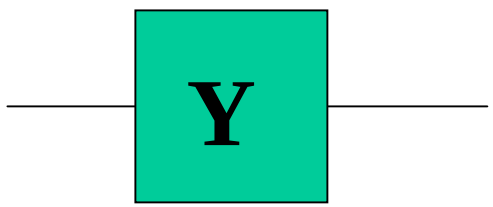
Pauli-X



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

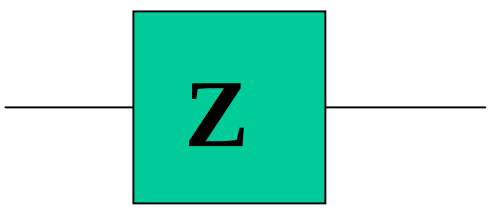
Classical inverter

Pauli-Y



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

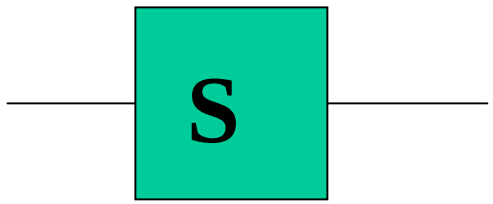
Pauli-Z



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

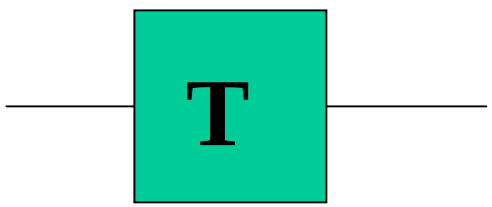
Other 1*1 unitary gates (quantum)

phase



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

Π/8



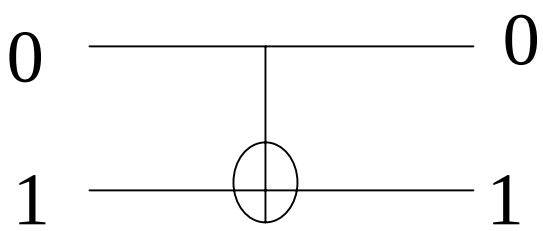
$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Quantum Circuits

- It will be useful in our work that quantum circuits are natural extensions of classical circuits.
- Quantum Circuits consist of quantum gates interconnected **without fanout or feedback** , by quantum wires.
- Each wire represents the path of a single qubit (in time or **space**, forward from left to right).
- It is described by a state in a **two-dimensional Hilbert space** with basis |0> and |1>.

•To understand how unitary operators are constructed from elementary operators we consider the XOR gate.

•Writing the two-particle basis states as the vectors



$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

the XOR gate becomes the following unitary operator

$$U_{\mathrm{XOR}} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

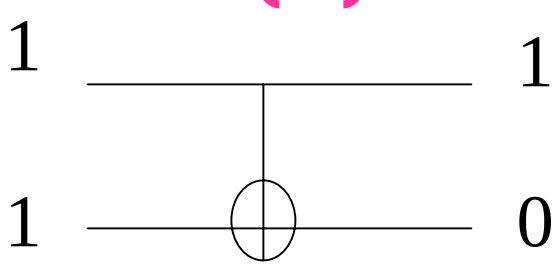
Let us check

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

00 01 10 11

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$



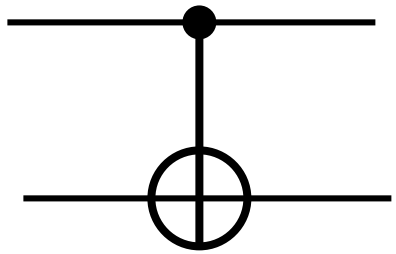
Which means |11> transformed to |10>

What did we learn?

- Matrices with only zeros and ones correspond to classical logic.
- Every Matrix is like a multi-output truth table or multi-output BDD
- Cascade circuit composition from gates is described by matrix multiplication
- In general Quantum Logic, we have matrices and vectors of complex numbers.

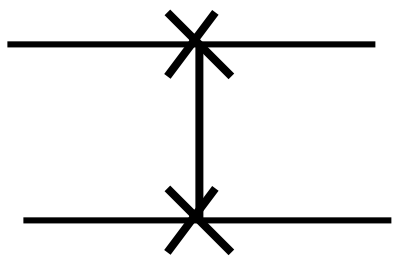
2*2 unitary gates

Controlled
-Not
(Feynman)



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

swap

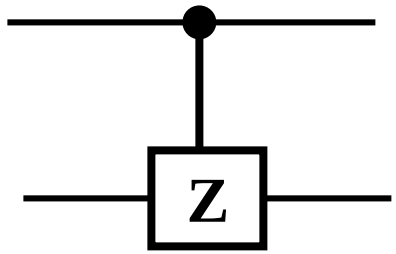


$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

These are counterparts of standard logic because all entries in arrays are 0,1

2*2 unitary gates

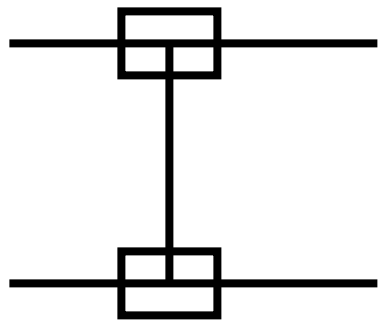
These are truly quantum logic gates because not all entries in arrays are 0,1



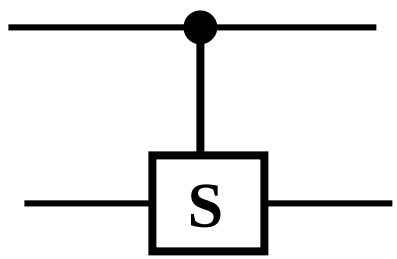
Controlled-Z

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

Another symbol

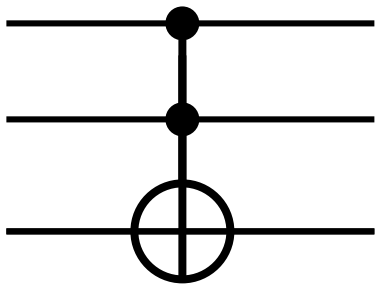


Controlled-phase



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

3*3 unitary gates

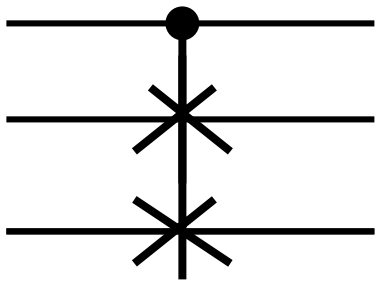


This is a counterpart of standard logic because all entries in arrays are 0,1

Toffoli

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

3*3 unitary gates



This is a counterpart of standard logic because all entries in arrays are 0,1

Fredkin

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

Very Good News

- Fortunately, the Toffoli gate may be constructed by two-particle scattering processes alone.

D. P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995).
D. Deutsch, Proc. Roy. Soc. Lond. A **425**, 73 (1989).
A. Barenco, D. Deutsch and A. Ekert, Phys. Rev. Lett. **74**, 4083 (1995).
T. Sleator and H. Weinfurter, Phys. Rev. Lett. **74**, 4087 (1995).
D. Deutsch, A. Barenco and A. Ekert, Proc. Roy. Soc. Lond. A **449**, 669 (1995).
S. Lloyd, ``Almost any quantum logic gate is universal," Los Alamos National Laboratory preprint.

In particular, we show a construction here involving the XOR gate and some one-bit gates.

Multiple bit gates: main result

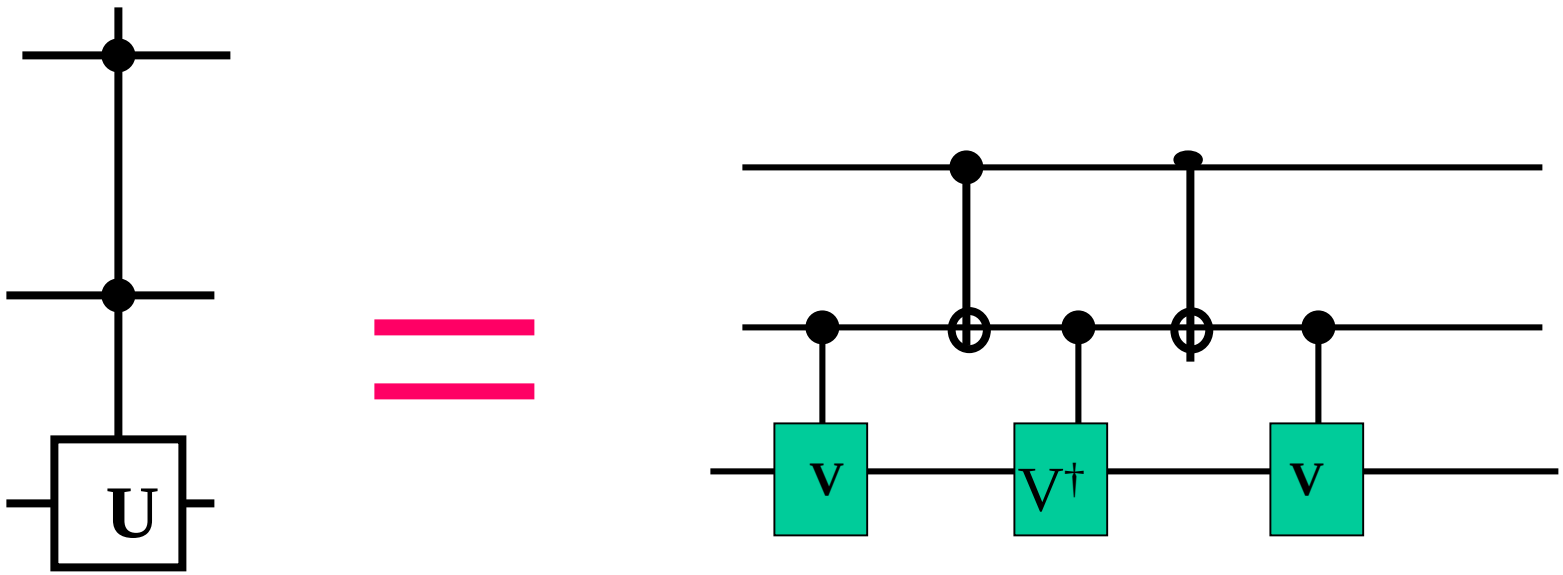
- The key observation here is the following:

Theorem: Any ***multiple qubit logic gate*** may be composed from **cnot** and **single qubit gates.**

This is one of the **most striking results** about quantum logic gates, since there exists no universal two-bit reversible classical logic gate.

Implementation of the Toffoli gate

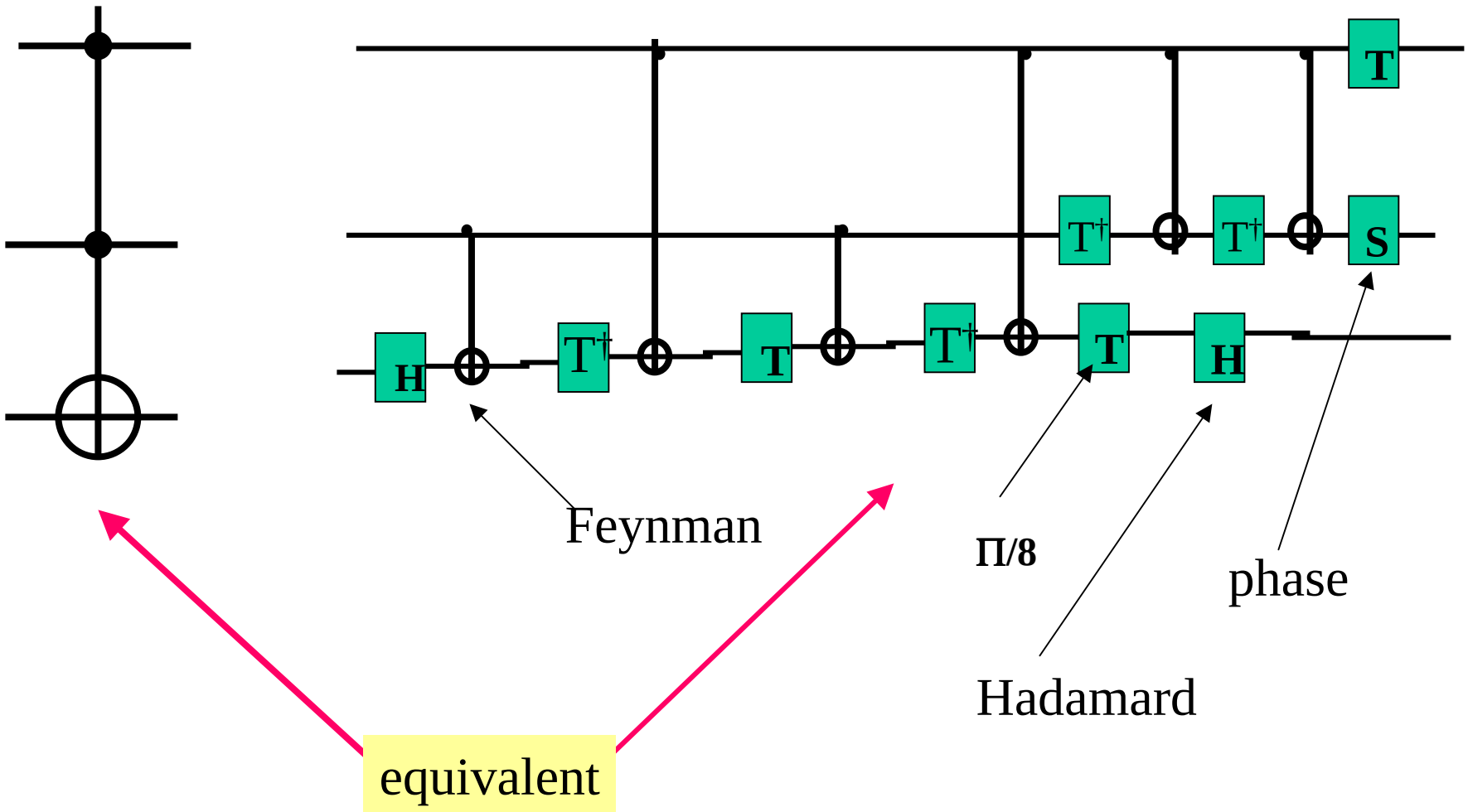
V is any unitary operator satisfying $V^2 = U$



The special case $V = (1- i)\,(I + iX)/2$ corresponds to the Toffoli gate

$$V^2 = X$$

Implementation of the Toffoli gate using Hadamard, phase, Feynman and Π/8 gates



Concluding on Quantum Logic Model

- The **inverter and Feynman gates** can be realized with Mach-Zender interferometer
- **Every Quantum** (unitary) function can be realized with Feynman gates and 1*1 gates.
- **Every 3*3 unitary gate** can be realized with 6 gates; 2 Feynman gates and 4 1*1 gates
- **Every 3*3 classical logic reversible gate** can be realized with a total of 5: 1*1 gates and 2*2 Feynman gates.

- Quantum XOR is sufficient for all logic operations on a quantum computer
- Quantum XOR can be used to construct arbitrary unitary transformations on **any finite set of bits**.
- Quantum gates have **the same number of inputs and outputs.**
  - they are not necessarily conservative.
  - they are reversible.

A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin and H. Weinfurter, ``Elementary gates for quantum computation,'' submitted to Phys. Rev. A 1995.

- **Is quantum logic realization practically possible?**
- **When?**

NSF seeks reliable quantum chip process

By R. Colin Johnson
EE Times
(07/04/01, 11:30 a.m. EST)

COLUMBUS, Ohio

— University researchers are aiming to craft a ***chip-manufacturing technology*** that can serve any of the diverse approaches to quantum computer architectures now being proposed.

The $1.6 million, four-year effort, undertaken for the National Science Foundation (NSF), hopes to come up with a **quantum-chip-making process** that is repeatable, reliable and attains good yields with room-temperature operation.

"We want to achieve a manufacturable process that will work with any one of the quantum-computing architectures being proposed today," said project leader Paul R. Berger, an associate professor of electrical engineering at Ohio State University.

The effort will be undertaken with the assistance of the University of Illinois at Urbana-Champaign, the University of Notre Dame, the University of California at Riverside, and the Naval and Air Force Research Laboratories.

NSF's Nanoscale Science & Engineering Program amasses nearly $500 million in research grants in various nanotechnology areas, including both nanoscale device and system architectures.

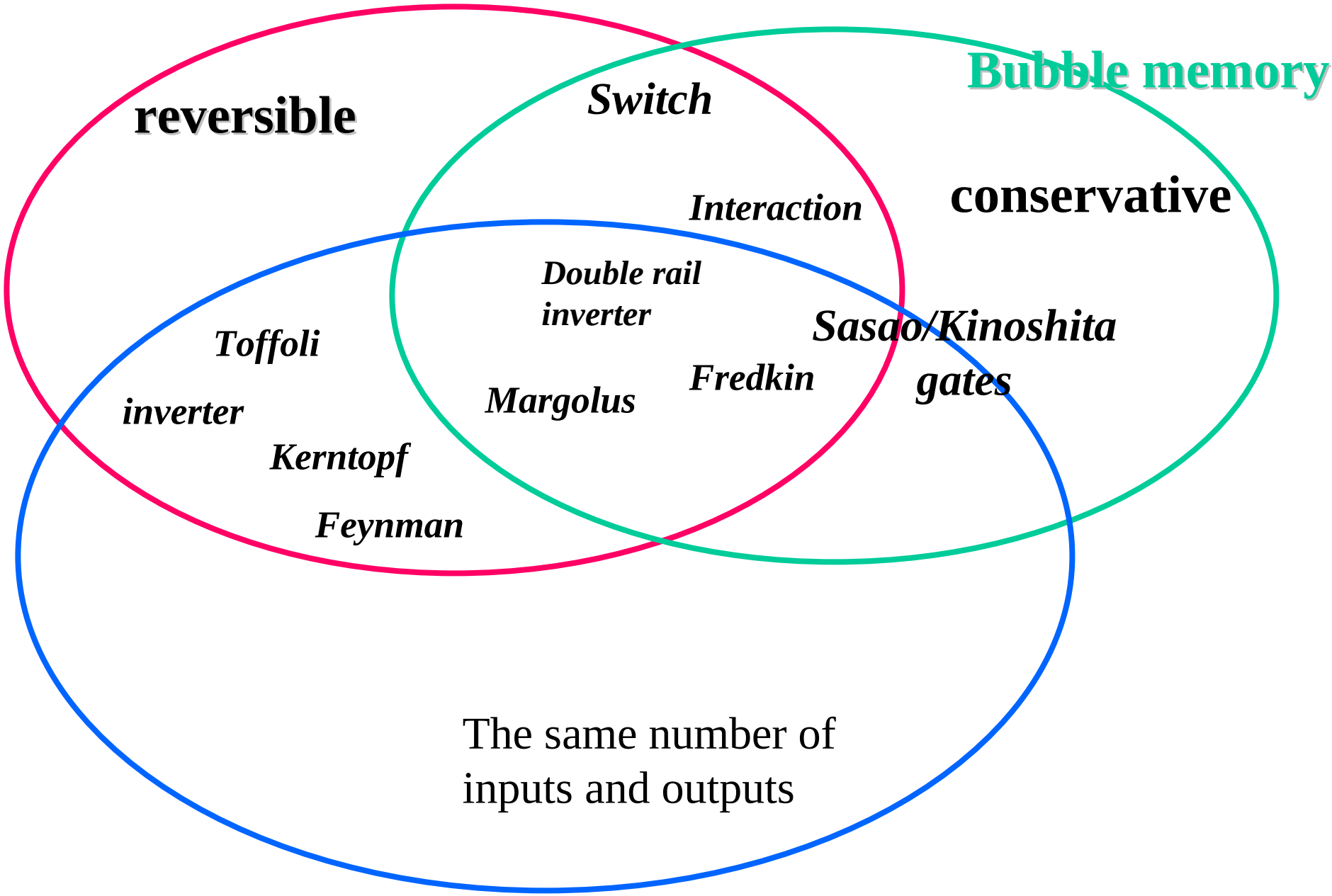
Quantum dots and single electron transistors are not yet quantum logic, but it is also coming

Very good news for Reed-Muller People

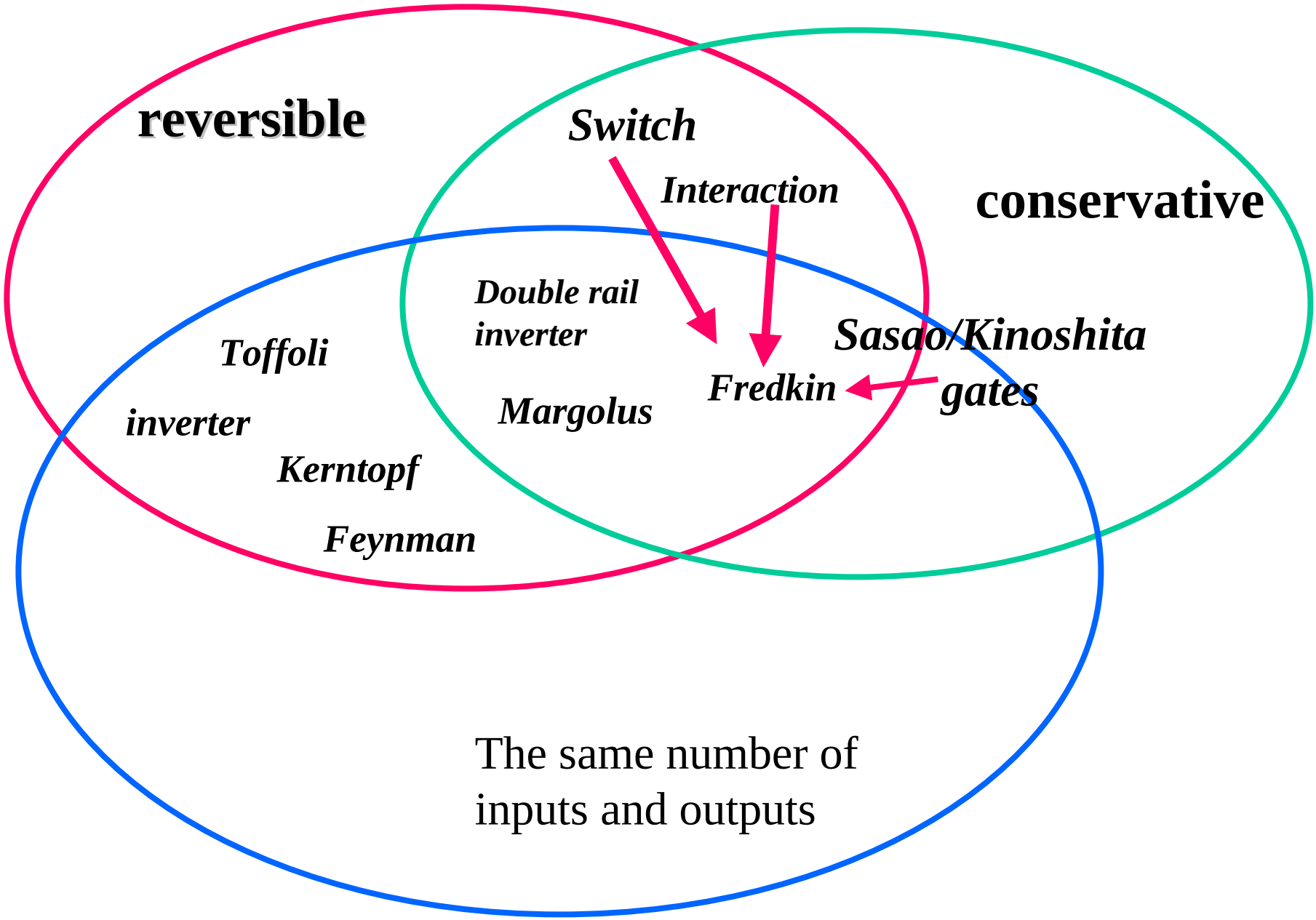
- **Quantum XOR** is the most important gate in Quantum Logic
- Synthesis of Quantum Circuits will require methods that are close to spectral and RM-based.
- **New Logic Synthesis is needed**

Research areas in reversible logic

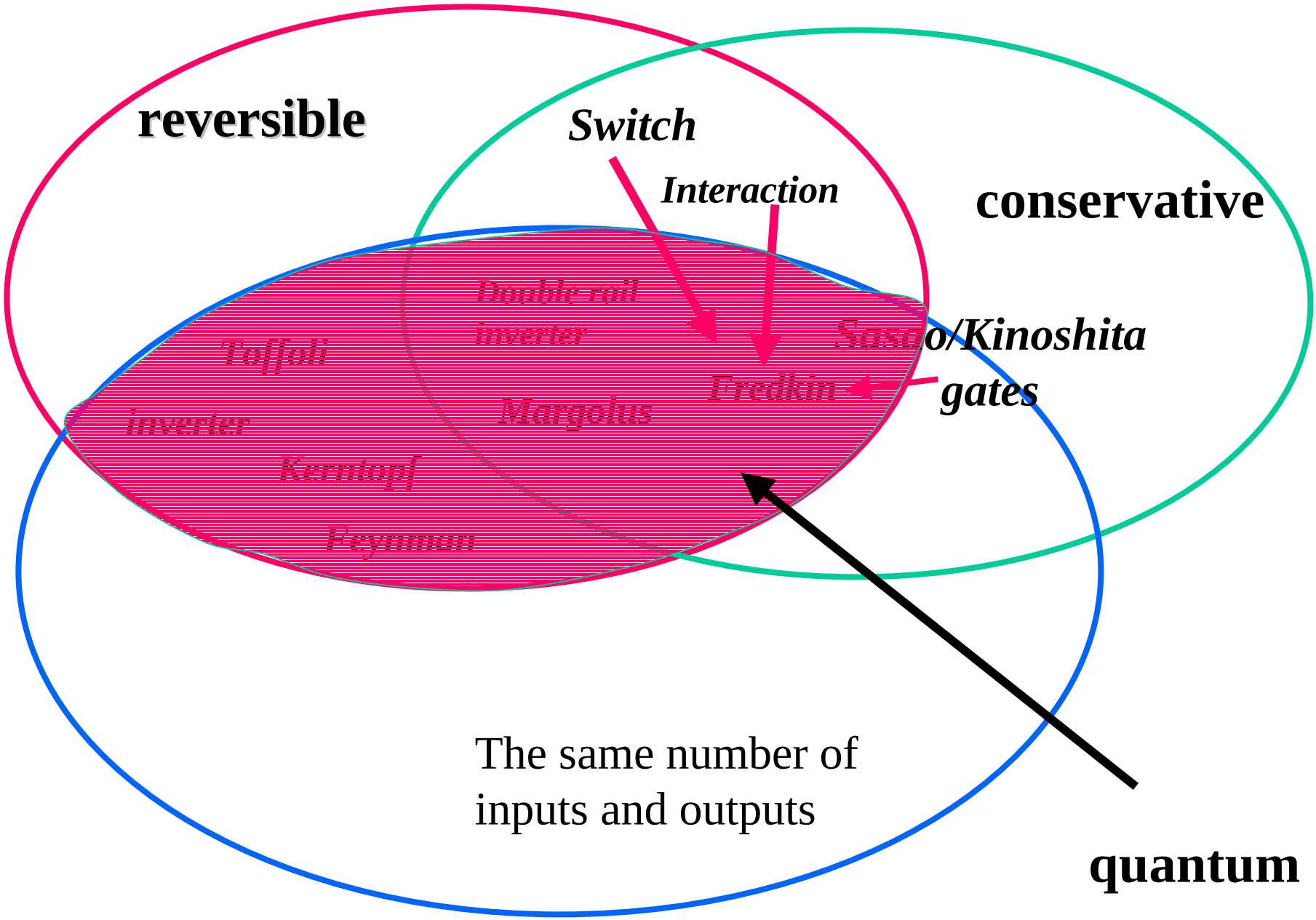
Types of reversible logic



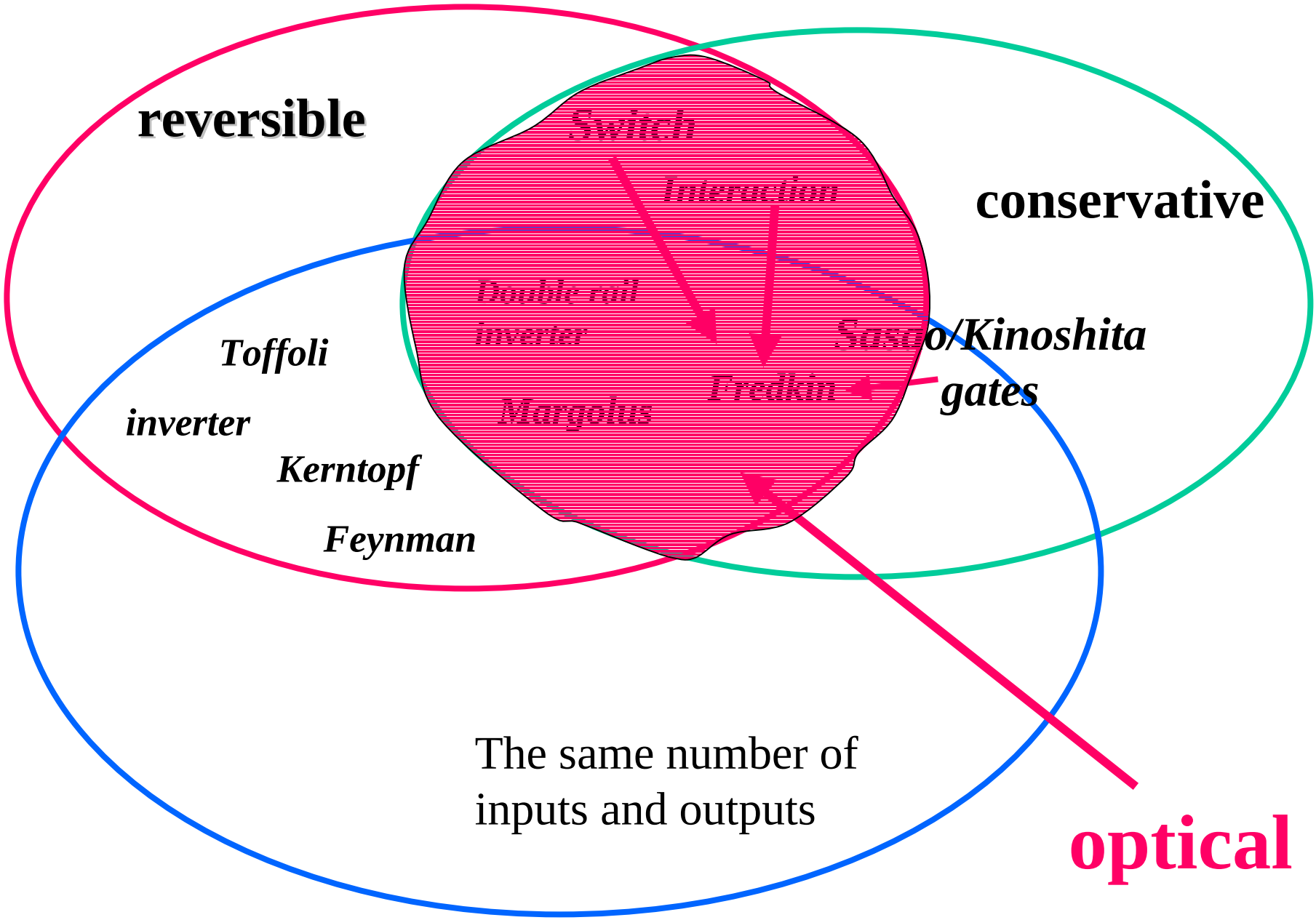
Types of reversible logic



Types of reversible logic



Types of reversible logic



Conclusions

- Use unitary matrices instead of truth tables.
- Compose and decompose unitary matrix to unitary matrices.
- Complex numbers instead of binary.
  - All our *"algebra"* was much simplified, but we explained the principles quite precisely, enough for programming compositions and decompositions.

Conclusions

- What we know as Kronecker Product is called Tensor Product in Quantum Logic.
- Many similarities with spectral transforms and especially Walsh can be found and used.
- Some spectral methods also use complex numbers.
- Hadamard gate and Hadamard transform.
- Quantum XNOR versus ESOP circuits and Kronecker Decision Diagrams.
- Generalizations of Shannon Expansion to Quantum Logic.

Conclusions

- In both classical reversible k*k logic and quantum logic, analysis of the circuit is based on composing unitary matrices.
- Synthesis of a circuit is based on **decomposing a unitary matrix to elementary quantum gates**.
- Good news is that it is enough to use quantum XOR as the only 2*2 gate and some 1*1 gates.
- Standard ways of decomposing 1*1 gates exist
- Quantum logic is linear, methods of Linearly Independent Logic can be used
- We will soon publish synthesis methods for quantum logic that are similar to the methods shown in our other paper yesterday.