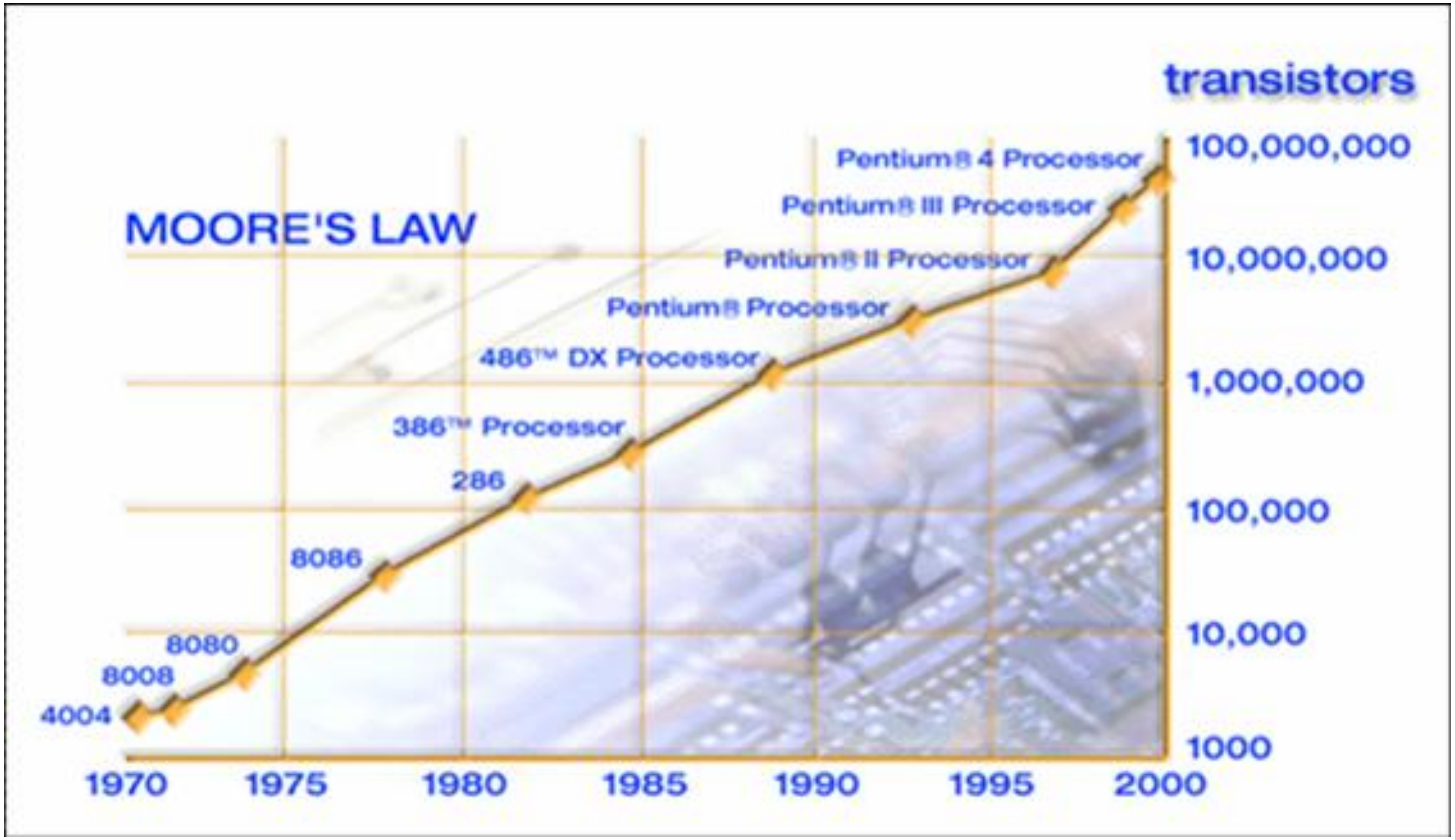
Simulation of a Quantum Computer

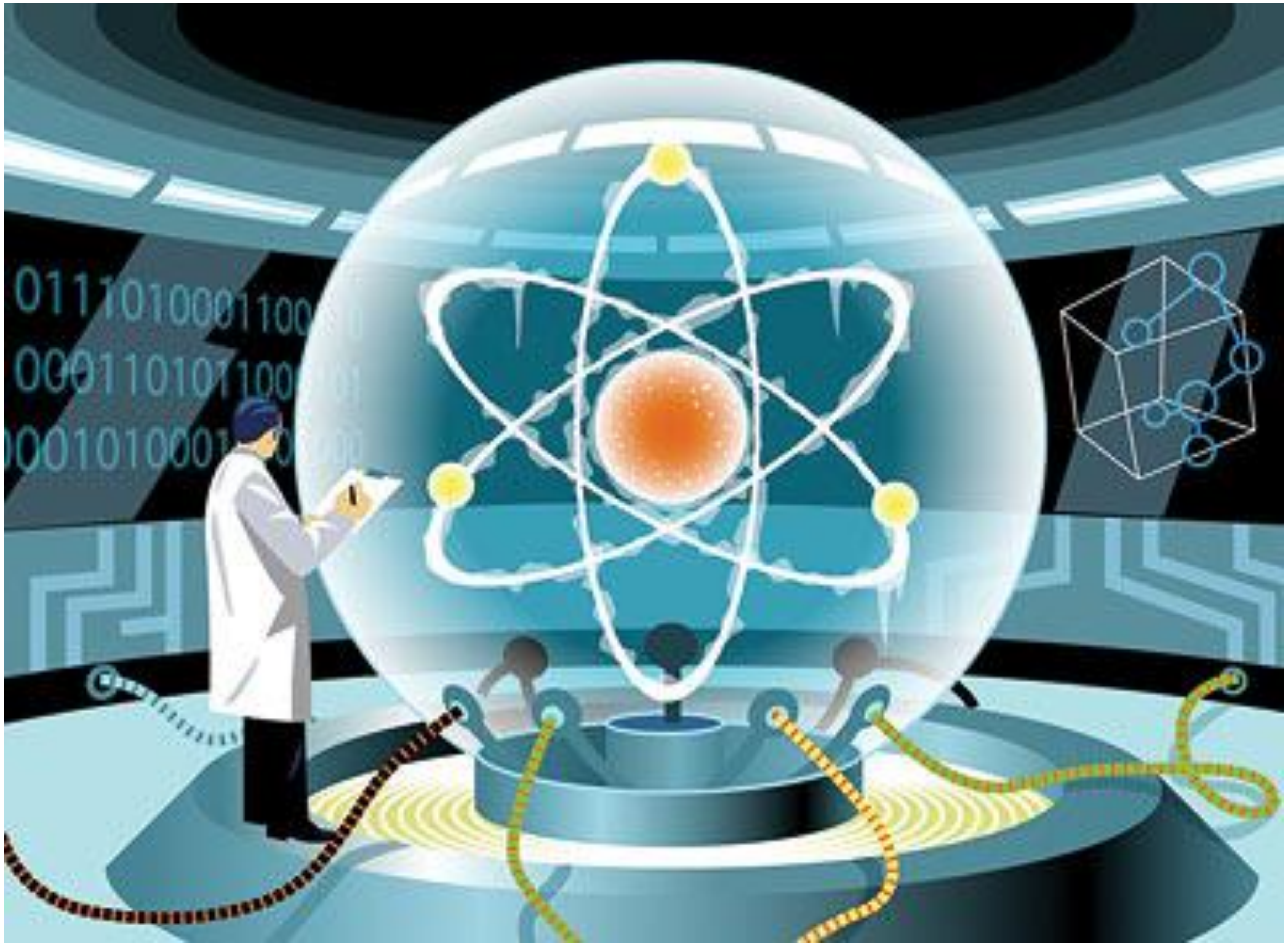
Roshenac Mitchell, Max Nolte, Martin Rüfenacht,
Mark Stringer, Justs Zariņš

Classical



MOORE'S LAW
transistors
100,000,000
10,000,000
1,000,000
100,000
10,000
1000
Pentium® 4 Processor
Pentium® III Processor
Pentium® II Processor
Pentium® Processor
486™ DX Processor
386™ Processor
286
8086
8080
8008
4004
1970
1975
1980
1985
1990
1995
2000

Quantum



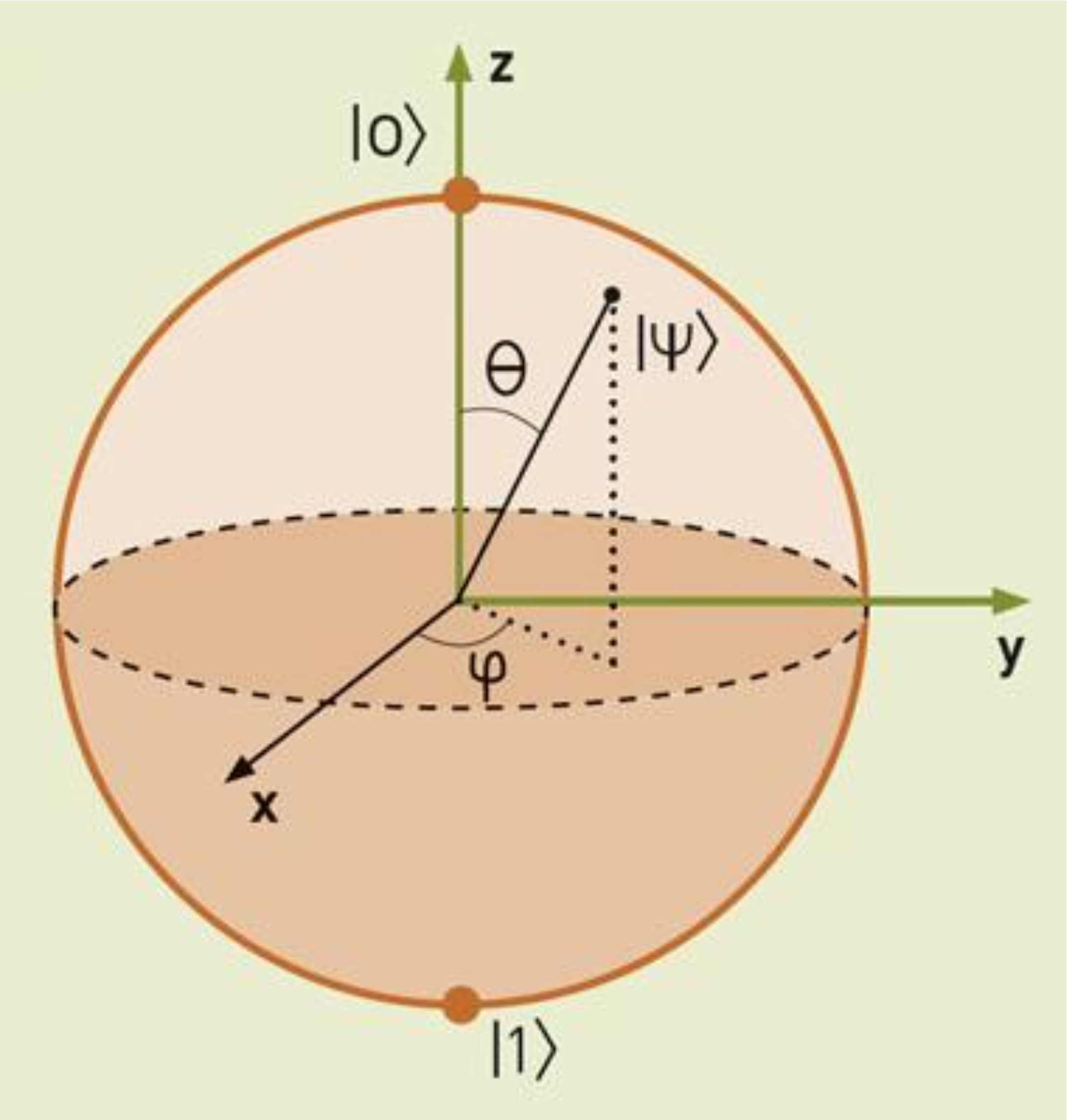
Quantum

Next

- Theory
- Design and Implementation
- Deutsch-Jozsa’s algorithm
- Grover’s algorithm
- Shor’s algorithm

Theory

Qubits



$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Quantum register

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

Quantum register

$$\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_x |\mathbf{x}\rangle$$

Gates

- Unitary operations
- Reversible
- Multiple gates form network

Hadmard gate

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

$$|x\rangle \;—\boxed{H}—\; (-1)^x|x\rangle + |1-x\rangle$$

Phase shift gate

$$\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

$|0\rangle$ left unchanged

$|1\rangle$ changed to $e^{i\phi}|1\rangle$

Controlled NOT

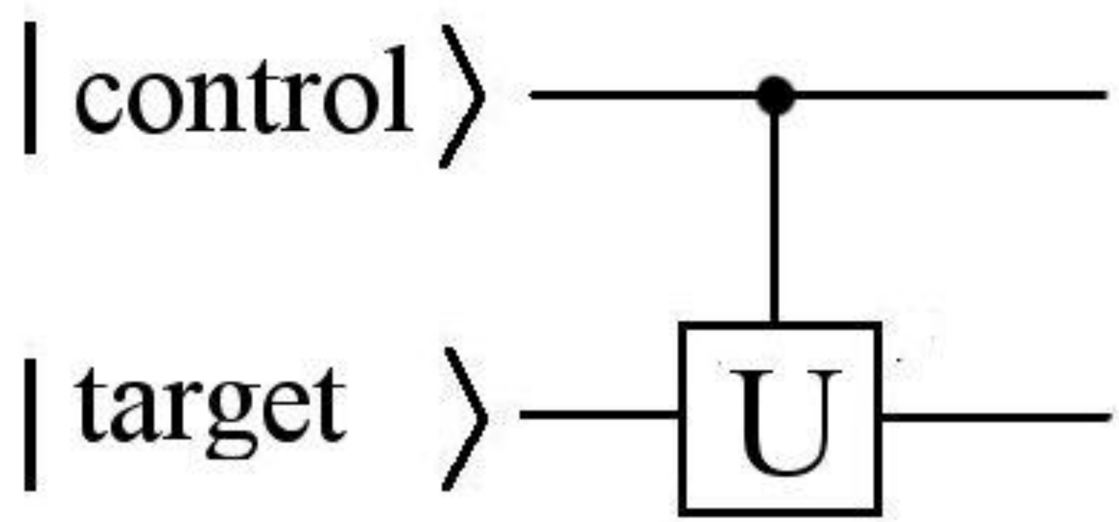
$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Control Before	Target Before	Control After	Target After
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Controlled V

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

Controlled U



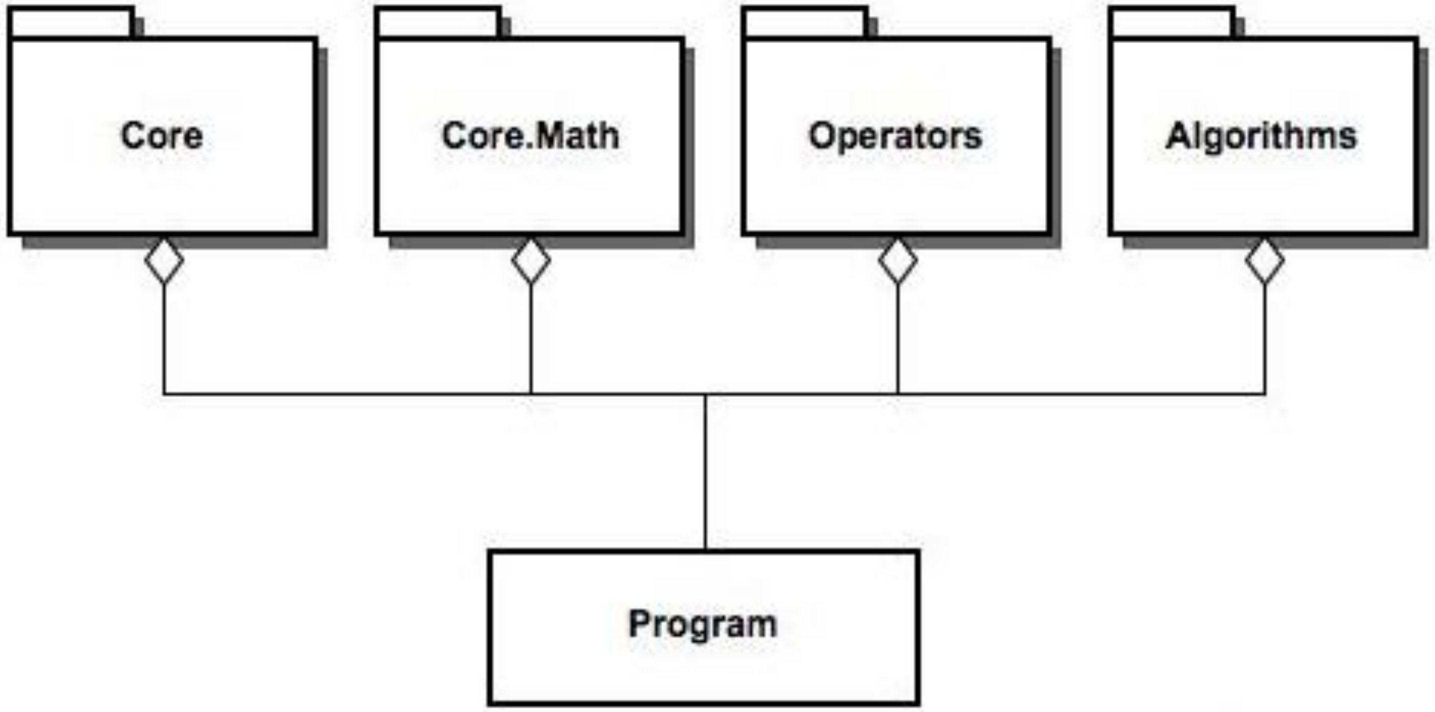
Swap

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

Design

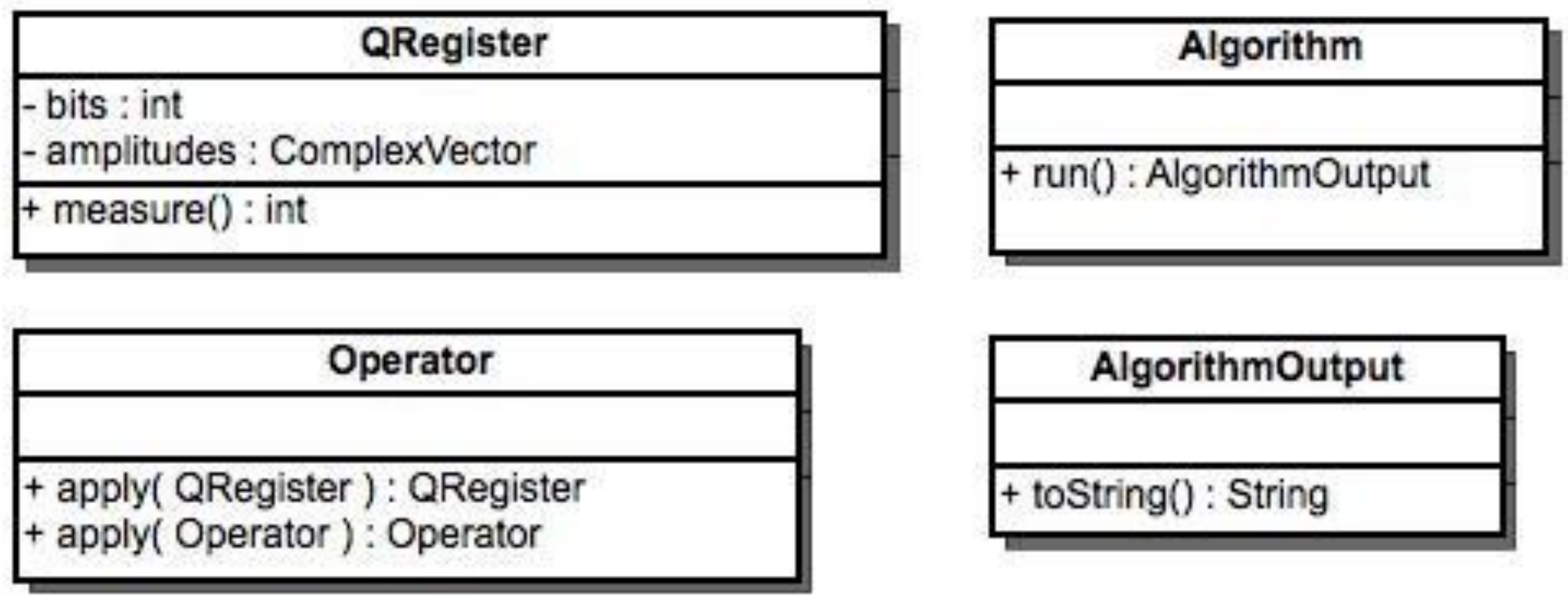
Design

- Library Centric
- Packages

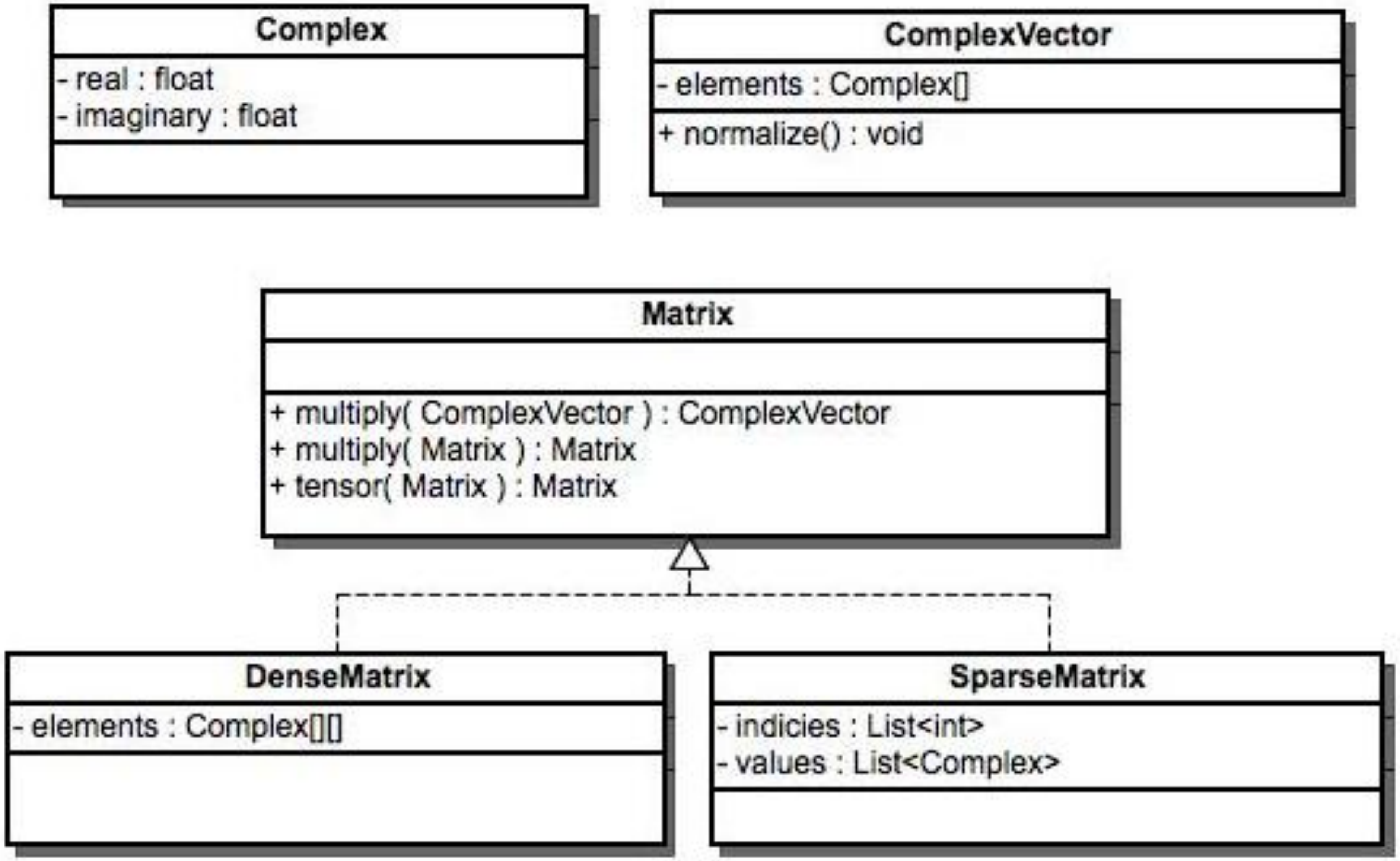


Core Package

- Data structure
- Interfaces
- Functional



Core.Math Package

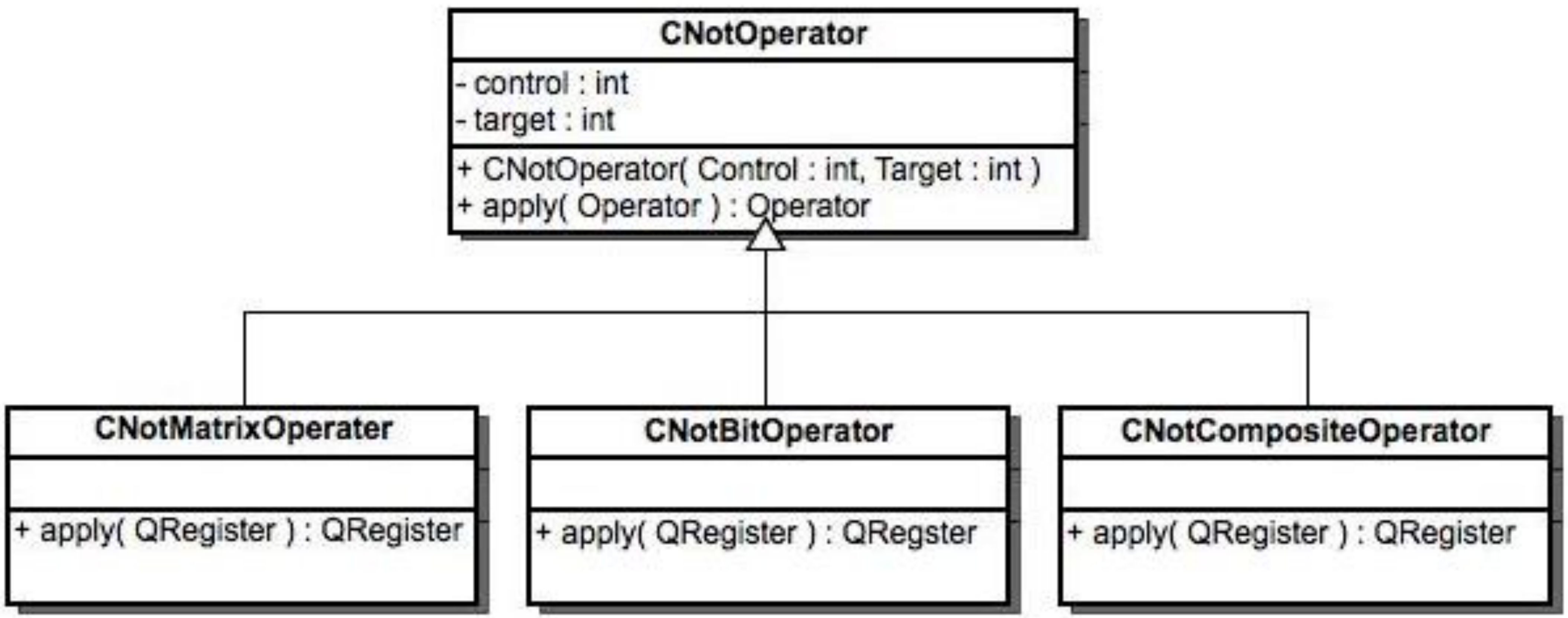


Operators Package

- Matrix
- Bit Manipulation
- Composite
- Bit assignment?

Operator	Matrix	Bit Manipulation	Composite
Hadamard	✓	✓	
Phase	✓	✓	
cV	✓	✓	
Swap		✓	
CNot	✓	✓	✓
CCNot			✓

CNot Analysis



CNot Analysis
Matrix Multiplication

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

CNot Analysis
Bit Manipulation

for every base:

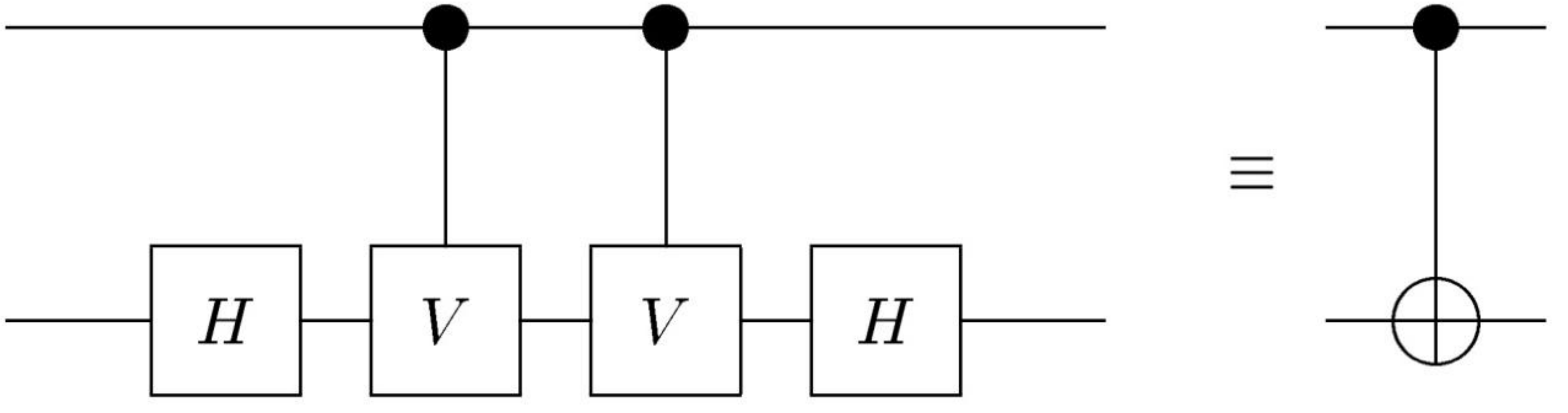
    test if control base

        swap target

    else

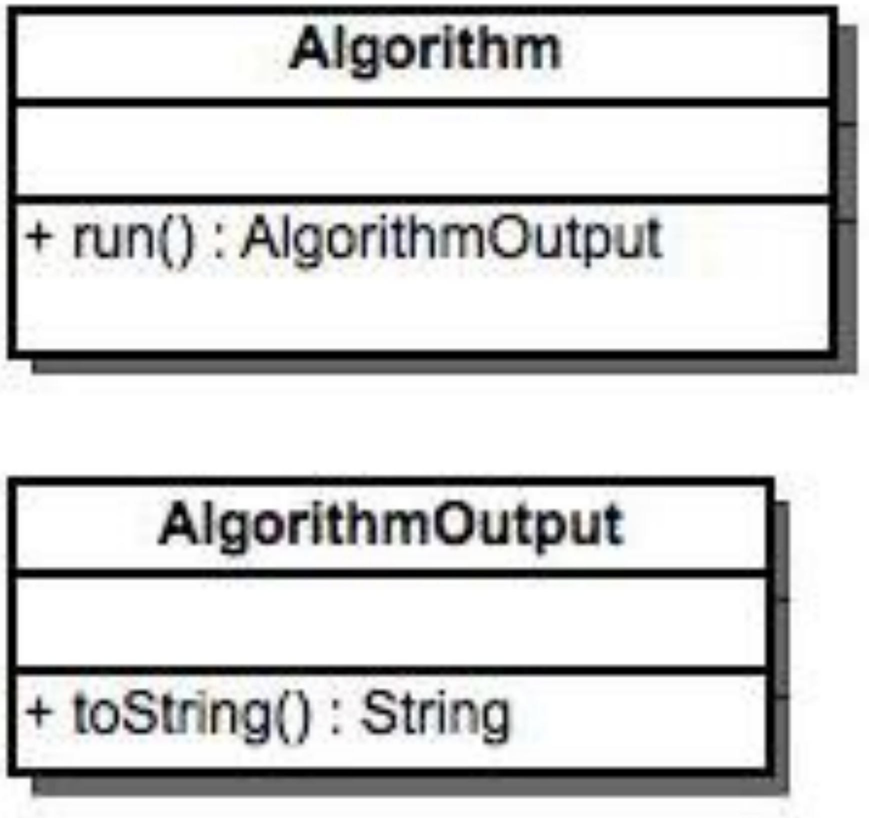
        do nothing

CNot Analysis
Composite Operator



Algorithms Package

- Deutsch-Jozsa
- Grover's Algorithm
- Shor's Algorithm



Deutsch-Jozsa Algorithm

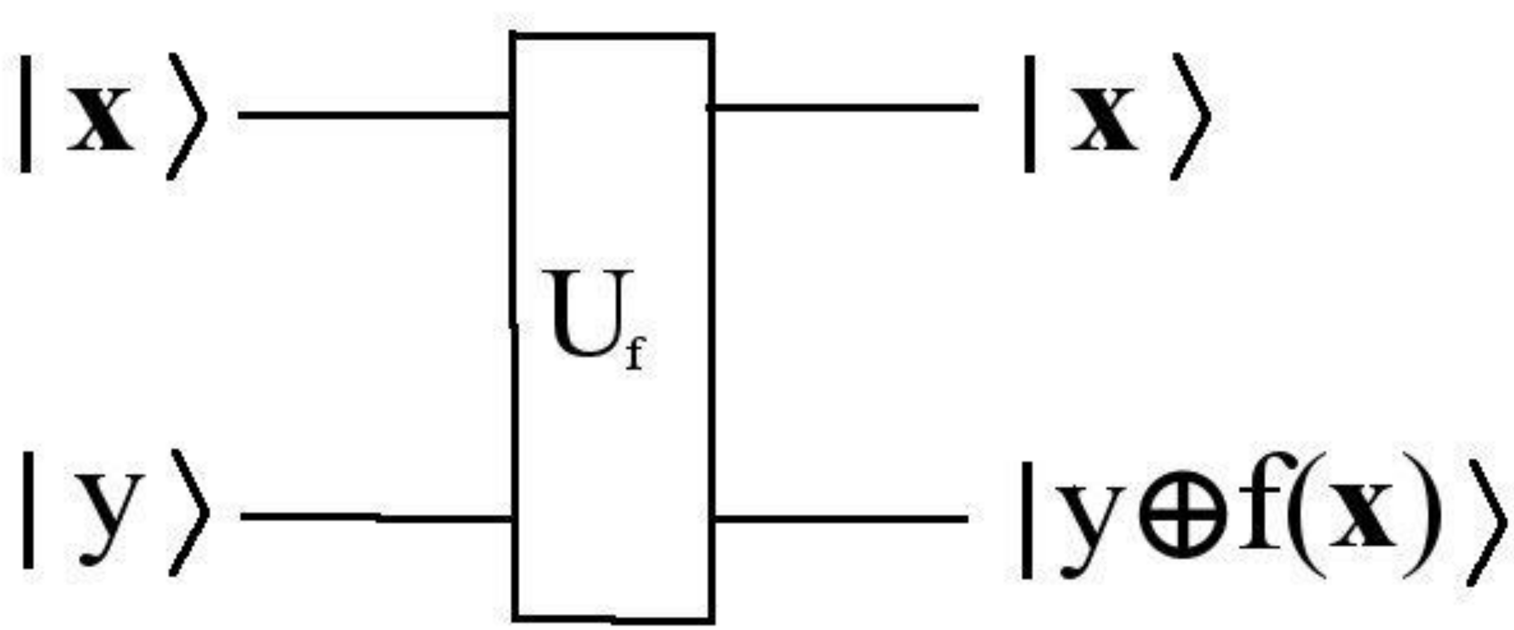
Deutsch-Jozsa Algorithm

- Determines whether function is constant or balanced

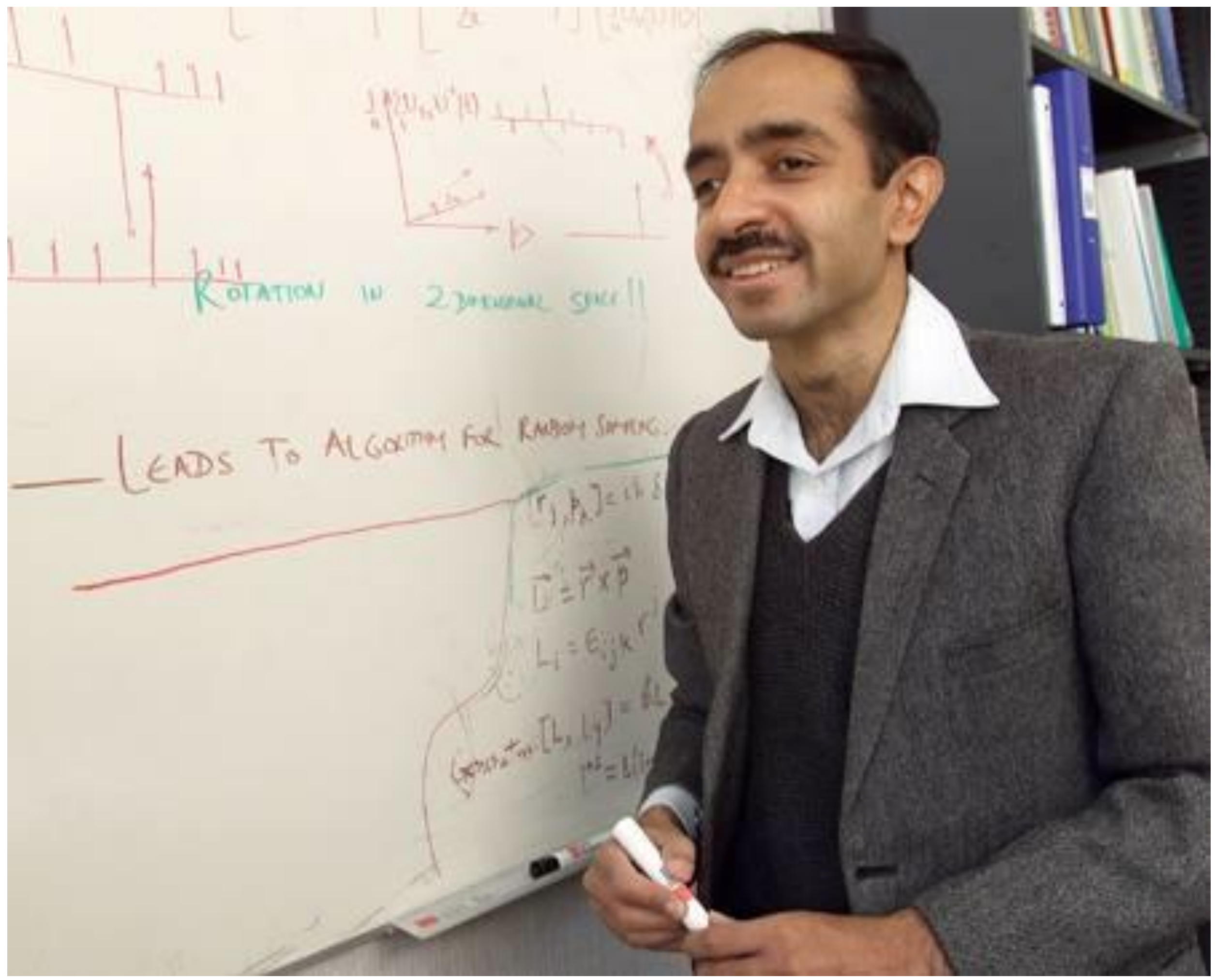
$$f : \{0,1\}^n \rightarrow \{0,1\}$$

Deutsch-Jozsa Algorithm

- Only needs one (quantum mechanical) evaluation



Grover’s algorithm



Grover's Algorithm

used for searching an unordered list for an element location

Classical computer = $O(N)$

- searches each entry sequentially

Grover algorithm = $O(\sqrt{N})$

- searches every entry simultaneously

Fastest possible order for searching in a quantum model

Grover’s Algorithm

- Initialisation

- Grover Iteration

- Measurement

Initialisation

creates a superposition of all basis states.

done by applying Hadamard operator to each qubit within the quantum register

$$|\psi\rangle = \frac{1}{\sqrt(N)} \sum_{i=0}^{N-1} |i\rangle$$

Grover Iteration

Iterated $\frac{\pi}{4}\sqrt{N}$ many times

Oracle - black box function represented by

$$\begin{aligned} U_\omega|\omega\rangle &= -|\omega\rangle \\ U_\omega|x\rangle &= |x\rangle \qquad \text{for all } x \neq \omega \end{aligned}$$

Diffusion Operator - inversion about the mean values of the amplitudes of each state

$$U_s = -HI_{|0\rangle}H$$

Measurement

causes quantum register to collapse

probability answer state is obtained is:

$$Prob(|x_0\rangle) \geq 1 - \frac{1}{N}$$

Implementation

Points of interest

- Grover Oracle
- Grover Diffusion Algorithm
- Grover Op Algortihm
- Grover Display

Oracle

Supplies information about the answer

-number of qubits required

-base count

semantically separates it from Grover Implementation

Diffusion Algorithm vs. Op Algorithm

Diffusion Algorithm

- analytic approach using a matrix as a Diffusion operator

Op Algorithm

- combination of base-wise operators to act on the QRegister

Op Algorithm WINS!

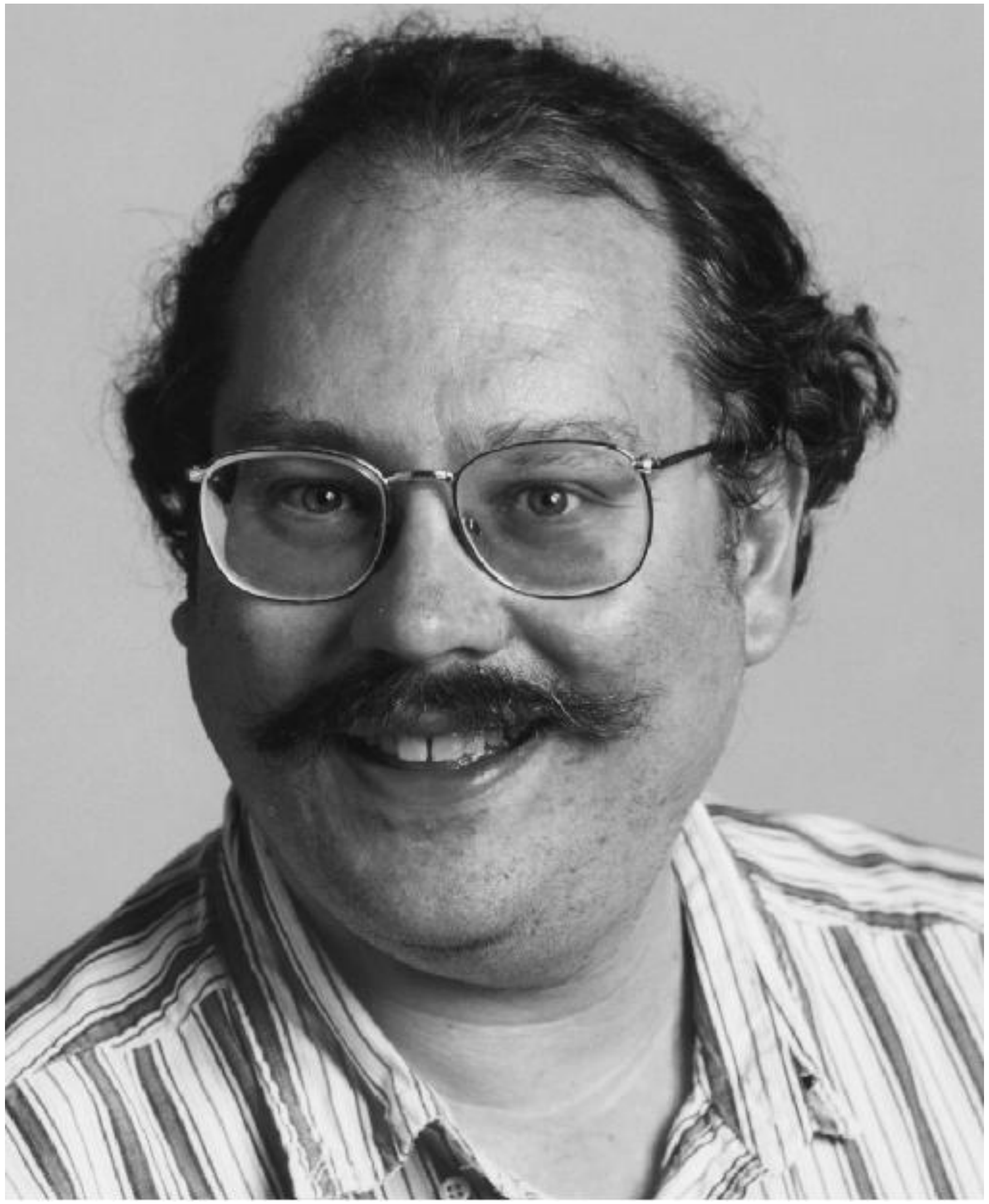
More efficient - less memory usage of the operators

More sequential - less complex to understand

Display

generated by Gram-Schmidt orthogonalisation of the answer base and initial zero base

Shor’s algorithm

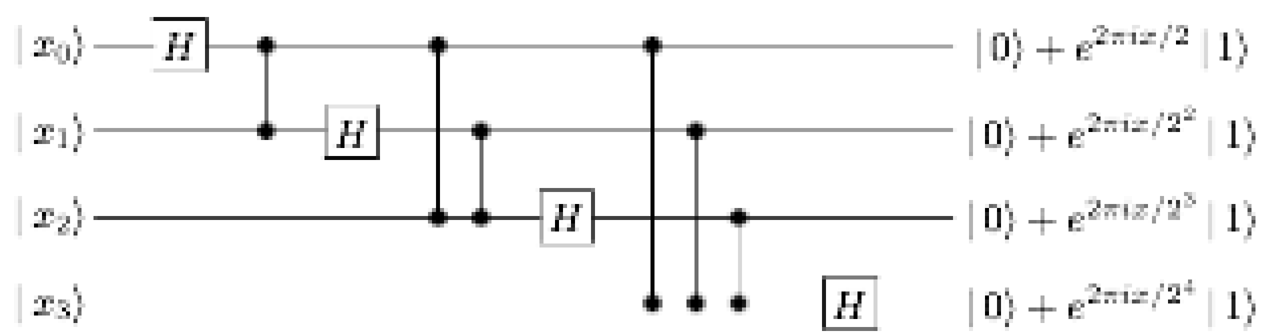


Shor’s Algorithm

- Factors a number into it’s prime factors
- Much faster than classical algorithms
- Does this by estimating the period

Shor's Algorithm

- Uses the quantum Fourier transform twice
- First time to put quantum register in superposition of states
- Second time to obtain an approximation to period



$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix}$$

Shor’s Algorithm

- Need 2 quantum registers
- One to Store arguments, other to store values of function
- First in state $|0\rangle$ second in state $|1\rangle$
- System in State $|0\rangle|1\rangle$

Shor's Algorithm

- Apply quantum Fourier transform to first register
- Puts it in superposition of all states

Shor's Algorithm

- Then apply gate shown to the right on the system
- This calculates all the values for the function simultaneously, faster than doing it classically

$$|x\rangle|1\rangle \rightarrow |x\rangle|m^x \bmod N\rangle$$

Shor's Algorithm

- Apply Quantum Fourier transform again to the first register
- The probability amplitudes for the periods add up to give high probability of correct answer upon measurement
- Other amplitudes cancel

Shor’s Algorithm

- Now measure the quantum register.
- This can then be used to find the period (in lowest terms) of the function by using a continued fraction expansion.

Shor’s Algorithm

- How does the period help us find the factors
- We know:

$$m^p = 1 \bmod N$$
$$m^p - 1 = 0 \bmod N$$
$$(m^{\frac{p}{2}} - 1)(m^{\frac{p}{2}} + 1) = 0 \bmod N$$

- So the two components above are factors of some multiple of N

Shor’s Algorithm

- Finding the greatest common devisor using Euclids algorithm for N and one of the components of the expression
- You obtain a factor
- The other can simply be found via division

Shor's Implementation

- Two gates used
- Quantum Fourier transform
- Unitary operator that applies the transform

$$|x\rangle|1\rangle \to |x\rangle|m^x \bmod N\rangle$$

Shor’s Implementation

- Quantum Fourier transform
- Implemented using a matrix representation
- Much faster to construct
- Same speed when applied to Register

Shor's Implementation

- Used bit manipulation for unitary transformation

- Faster than matrix/gates
- Conceptually easier to construct than a matrix/gates

Conclusion

Great success!

- Framework and Grover’s algorithm
- Two additional algorithms
- Good teamwork

The Future?

Live Demo