MP 472 Quantum Information and Computation

http://www.thphys.may.ie/staff/jvala/MP472.htm

Outline

Review
- computational complexity classes
- models of computation

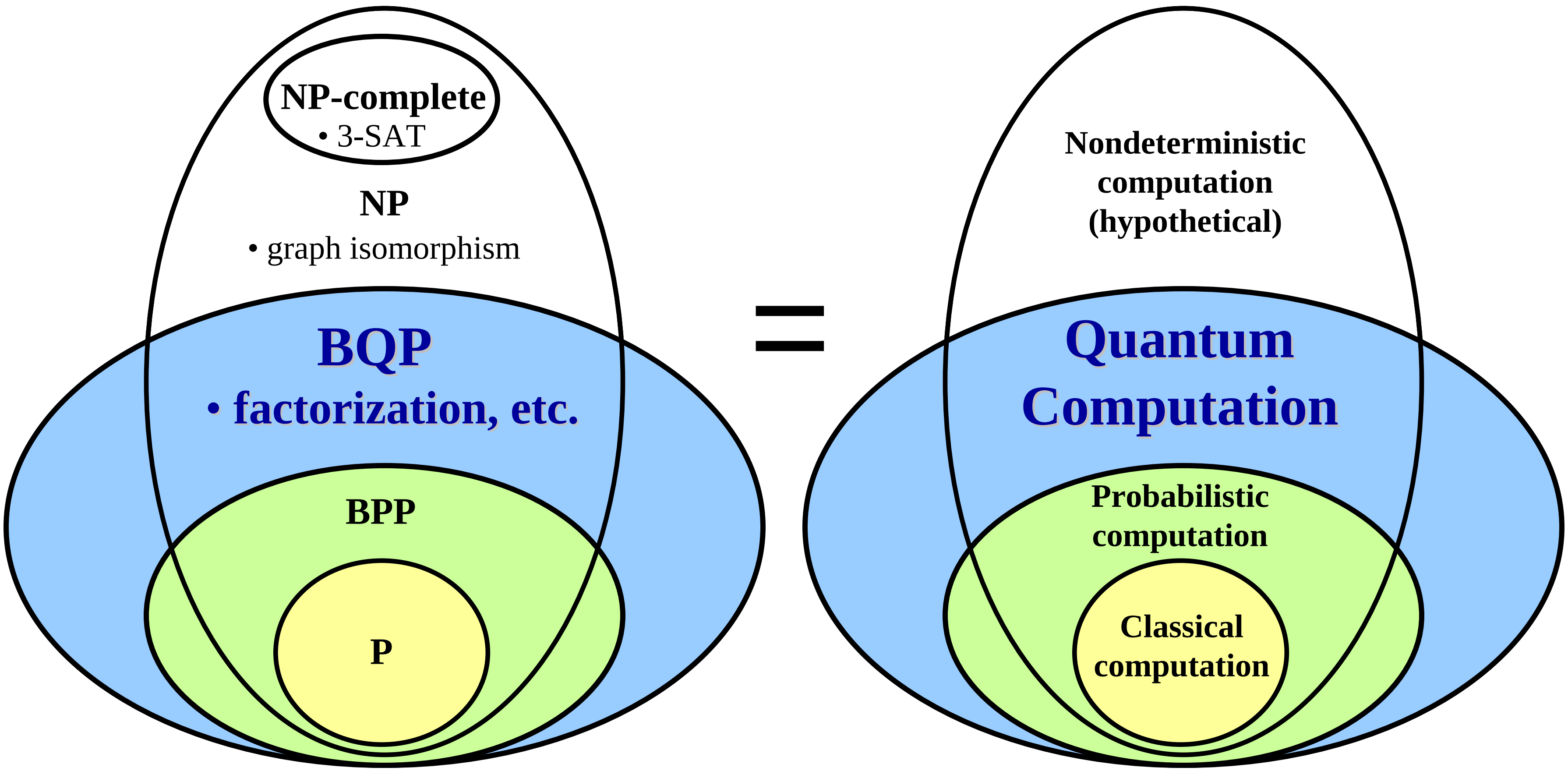
Deutsch-Jozsa algorithm
- task definition and classical complexity
- quantum algorithm
- two-qubit example
- Hadamard gate for a general n-qubit input
- Deutsch-Jozsa algorithm for n-bit functions

Quantum algorithms overview
- quantum algorithms based on Fourier transform
- quantum simulations
- quantum search algorithms

P ⊆ BPP ⊆ BQP ⊆ PSPACE

- computational complexity classes provide classification of computational problems according to whether they are tractable (solvable in polynomial time) using certain (physical or conceptual) models of computation;

- **quantum computation** can solve problems that are not known to be tractable on a classical computer, i.e. it **is more powerful than classical computation**;



Deutsch-Jozsa algorithm

Task:
to compute whether a Boolean function *F* over n variables is constant or balanced.

A Boolean function *F* over n variables is said to be
• *constant* if it gives the same output for all the possible inputs;
• *balanced* if it outputs 0 for half of all the possible inputs and 1 for the other half.
(Remark: there are other functions but they will not be relevant to the DJ algorithm.)

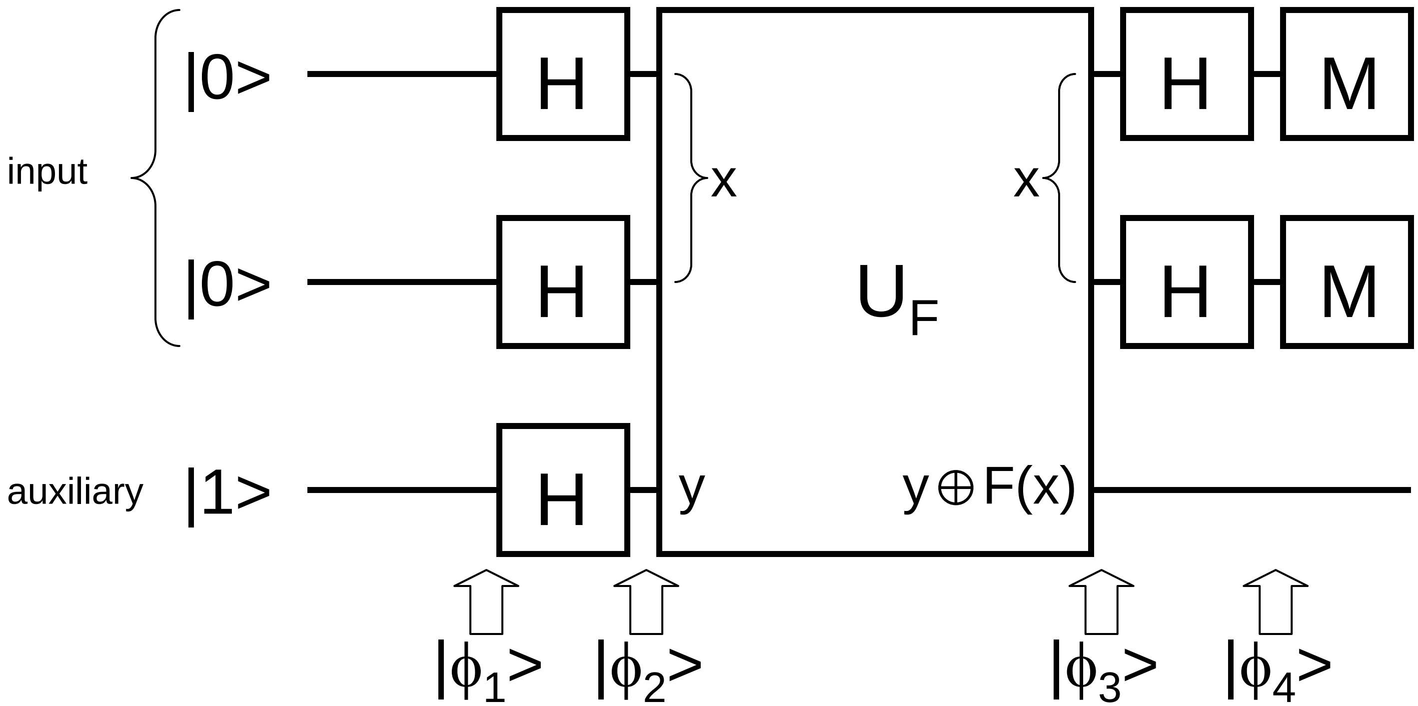
Examples:

	Constant:		Balanced:					
x_1 x_2	$F(x_1,x_2)$		$F(x_1,x_2)$					
0 0	1	0	1	0	1	0	0	1
0 1	1	0	1	0	0	1	1	0
1 0	1	0	0	1	1	0	1	0
1 1	1	0	0	1	0	1	0	1

Classical complexity is exponential:

in the worst case, the function (in a black box) needs to be called $2^{n-1}+1$ times to check more than a half of all inputs.

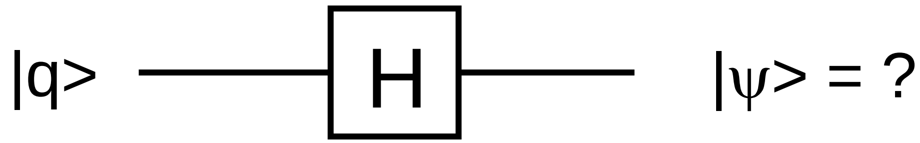
Deutsch-Jozsa algorithm (example for 2-qubit functions)



Initial state:

$$|\phi_1> \ = |0> \otimes |0> \otimes |1> = |001>$$

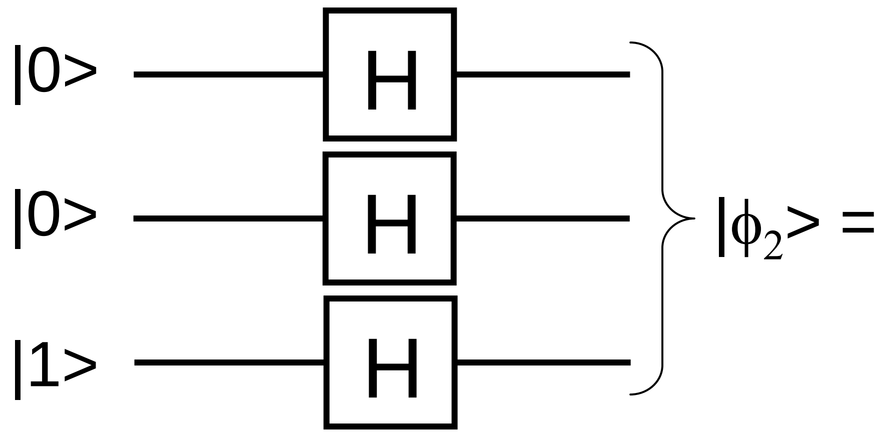
Hadamard gate



$|0\rangle \longrightarrow 2^{-1/2}(|0\rangle + |1\rangle)$

$|1\rangle \longrightarrow 2^{-1/2}(|0\rangle - |1\rangle)$

3 qubits in |001>:



$$= (2^{-1/2})^3 (|0\rangle+|1\rangle) \otimes (|0\rangle+|1\rangle) \otimes (|0\rangle-|1\rangle) =$$

$$= 2^{-1}(|00\rangle+|01\rangle+|10\rangle+|11\rangle) \otimes [2^{-1/2}(|0\rangle-|1\rangle)]$$

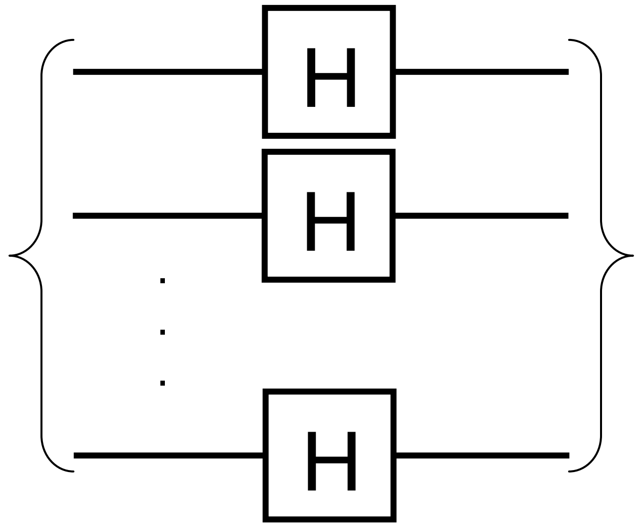
Hadamard gate: general n-qubit input

General n-qubit input state:

n-qubit state in computational basis (e.g. {|000>,|001>,…,|111>})

$$|\phi> = \sum_x c_x |x>$$

x is a bit string, i.e. $x=x_1x_2\ldots x_n$ where x_k are bits



$$|\psi> = 2^{-n/2}\sum_z\sum_x c_x (-1)^{x.z} |z>$$

bitwise inner product of the bit strings x and z modulo 2

Example:

$$|\phi> = |001>$$

-there is only one element in the sum over x which is labeled by the bit string $x=x_1x_2x_3 = 001$ (the coefficient $c_{001}=1$)

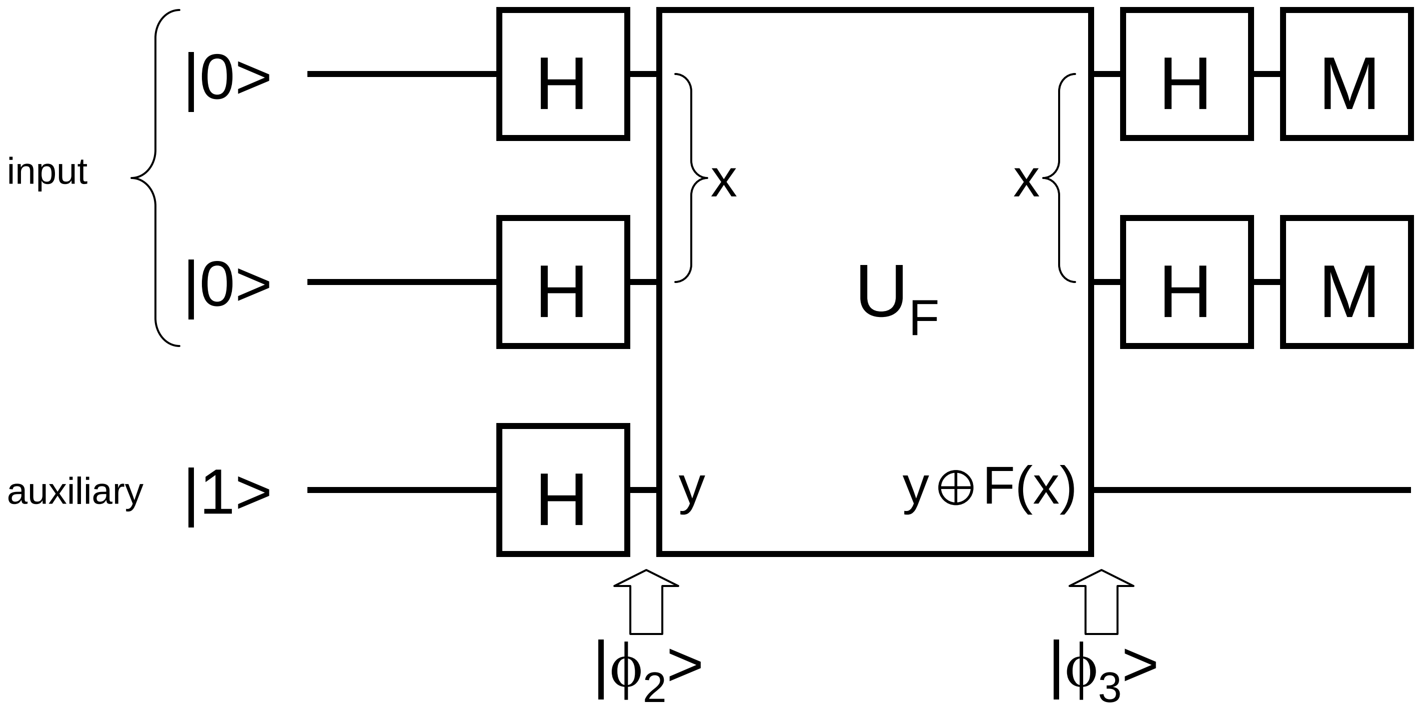
$$|\psi> = 2^{-3/2}\sum_z (-1)^{(0.z_1+0.z_2+1.z_3)(\mathrm{mod}\ 2)} |z = z_1z_2z_3> =$$

the summation goes over all computational basis elements (e.g. {|000>,|001>,…,|111>})

$$= 2^{-3/2}(|000> - |001> + |010> - |011> + |100> - |101> + |110> - |111>) =$$

$$= 2^{-1}(|00>+|01>+|10>+|11>) \otimes [2^{-1/2}(|0>-|1>)]$$

Deutsch-Jozsa algorithm (example for 2-qubit functions)



$$|\phi_2\rangle = \underbrace{2^{-1}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)}_{x} \otimes \underbrace{[2^{-1/2}(|0\rangle-|1\rangle)]}_{y}$$

$y \oplus F(x)$ means that y and F(x) are added modulo(2):

what does U_F do (to the quantum state $|\phi_2\rangle$)?

For this example let F be balanced:

x_1	x_2	$F(x_1,x_2)$
0	0	0
0	1	1
1	0	0
1	1	1

$F(x) \oplus y \pmod 2$

Let us use our example:

x_1	x_2	$F(x_1,x_2)$	y	$F \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	1	1	0

the function acts like it flips y whenever F=1

Let us look how this acts on

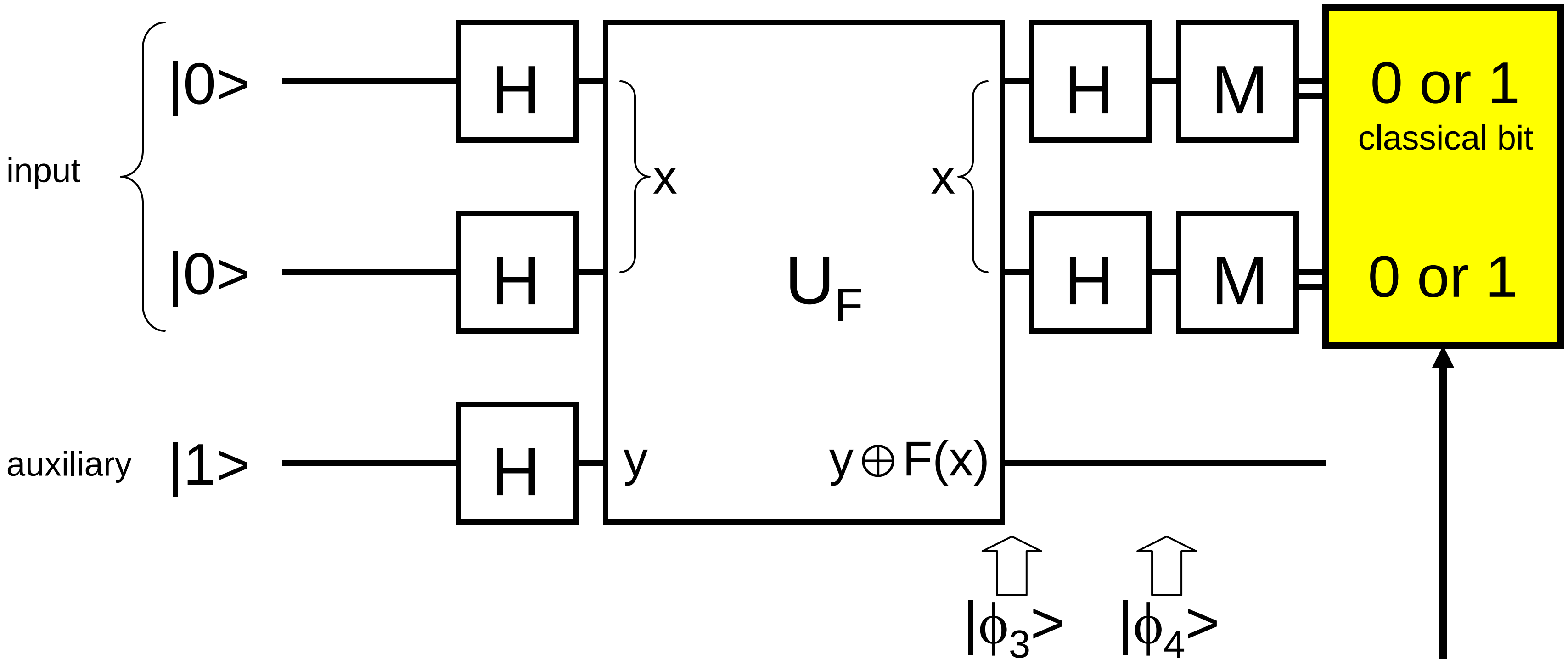
$$|\phi_2> = \underbrace{2^{-1}(|00>+|01>+|10>+|11>)}_{x} \otimes \underbrace{[2^{-1/2}(|0>-|1>)]}_{y} =$$

$$= 2^{-1}(|00> \otimes \underbrace{[2-1/2(|0>-|1>)]}_{y} + |01> \otimes \underbrace{[2-1/2(|0>-|1>)]}_{y} + |10> \otimes \underbrace{[2-1/2(|0>-|1>)]}_{y} + |11> \otimes \underbrace{[2^{-1/2}(|0>-|1>)]}_{y}\ )$$

$$\xrightarrow{U_F} 2^{-1}(|00> \otimes \underbrace{[2-1/2(|0>-|1>)]}_{F \oplus y} + |01> \otimes \underbrace{[2-1/2(|1>-|0>)]}_{F \oplus y} + |10> \otimes \underbrace{[2-1/2(|0>-|1>)]}_{F \oplus y} + |11> \otimes \underbrace{[2^{-1/2}(|1>-|0>)]}_{F \oplus y}\ ) =$$

$$= |\phi_2> = \underbrace{2^{-1}(|00> - |01>+|10> - |11>)}_{x} \otimes \underbrace{[2^{-1/2}(|0>-|1>)]}_{y}$$

Deutsch-Jozsa algorithm (example for 2-qubit functions)



$|\phi_3\rangle = 2^{-1}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) \otimes [2^{-1/2}(|0\rangle - |1\rangle)]$

$|\phi_4\rangle = |01\rangle \otimes [2^{-1/2}(|0\rangle - |1\rangle)]$

If both measurements give 0, the function is constant, otherwise it is balanced.

Homework: calculate what $|\phi_4\rangle$ corresponds to all other constant and balanced two-qubit functions.

Deutsch-Jozsa algorithm

Inputs:
A black box U_F which performs the transformation $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus F(x)\rangle$, for $x \in \{0,1,\ldots,2^{n-1}\}$ and $F(x) \in \{0,1\}$. It is promised that F(x) is either constant or balanced.

Outputs:
0 iff F is constant

Complexity/Runtime:
One evaluation of U_F. Always succeeds.

Exponential speed-up compared to classical algorithm!!!

Procedure:

1. $|\phi_1\rangle = |0\rangle^{\otimes n}|1\rangle$
2. $\rightarrow |\phi_2\rangle = 2^{-n/2}\sum_{x=0}^{2^n-1} |x\rangle[2^{-1/2}(|0\rangle - |1\rangle)]$
3. $\rightarrow |\phi_3\rangle = 2^{-n/2}\sum_{x} (-1)^{F(x)}|x\rangle[2^{-1/2}(|0\rangle - |1\rangle)]$
4. $\rightarrow |\phi_4\rangle = \sum_{z}\sum_{x} 2^{-n} (-1)^{x.z+F(x)}|z\rangle[2^{-1/2}(|0\rangle - |1\rangle)]$
5. $\rightarrow z$

bitwise inner product of x and z modulo 2

Other quantum algorithms

Algorithms based on Quantum Fourier Transform (Shor)

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

- classically FFT on $N=2^n$ numbers takes $N\log(N)$, i.e. $n2^n$ steps
- quantum algorithm takes $\log^2(N) = n^2$

Exponential speed-up
compared to classical algorithm!!!

- Applications: phase estimation (electronic structure of molecules …)
  order estimation and factoring (Shor's algorithm)
  discrete logarithm
  hidden subgroup problem …

Quantum simulations (Feynmann)

- simulating other quantum mechanical systems on a quantum computer

Quantum search algorithms (Grover)

- searching in a database (Grover's algorithm) – NP-hard problem

Quadratic speed-up
compared to classical algorithm!!!