Week 3

Lecture 5
Reversible computation, One qubit adder, the Deutsch-Josza algorithm

Lecture 6
Two basic quantum algorithms: Bernstein-Vazirani and Simon's Algorithms

Lab 3
Logical statements, Reversible logic, Adder, Deutsch-Josza algorithm

Simple Quantum Algorithms: Simon and Bernstein-Vazirani

Physics 90045
Lecture 6

Overview

In this lecture we will discuss some of the early quantum algorithms,

1. Bernstein-Vazirani algorithm
2. Simon's algorithm

These algorithms can be taken as simple demonstrations of quantum computation, even if they are of limited practical use.

See:
Kaye, Chapter 6
Nielsen and Chuang, Chapters 1 & 4
Reiffel, 7.1-7.5

PHYC90045 Introduction to Quantum Computing

Bernstein-Vazirani Problem

Given a Boolean function, *f*:

$$f(x) = x \cdot s \mod 2$$

find s.

Recall, bitwise product: $x \cdot s = \sum_i x_i s_i$

PHYC90045 Introduction to Quantum Computing

Example: Linear Boolean function

Example:

$$f(x) = x \cdot 5 \mod 2$$

Remember, in binary, 5 = 101.

x	f(x)
000	0
001	1
010	0
011	1
100	1
101	0
110	1
111	0

Given a black-box which calculates this function, find s=5.

PHYC90045 Introduction to Quantum Computing

Solving BV Problem Classically

$$f(x) = x \cdot 5 \mod 2$$

x	f(x)
000	0
001	1
010	0
011	1
100	1
101	0
110	1
111	0

Input one single digit “1” at a time.

Can determine s using n queries.

Bernstein-Vazirani Problem

Given a Boolean function, *f*:

$$f(x) = x \cdot s \mod 2$$

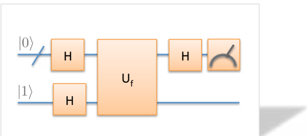
find s.

Recall: bitwise product: $x \cdot s = \sum_i x_i s_i$

- **Classical algorithm** *needs n queries*
- ***Quantum algorithm*** *needs just 1 query.*

Bernstein-Vazirani algorithm

The circuit is the same as for the Deutsch-Josza algorithm:



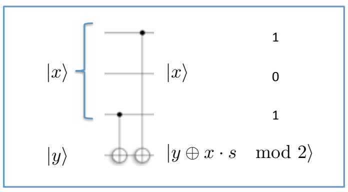
The guarantees on f are different:

$$f(x) = x \cdot s \mod 2$$

Recall: Deutsch-Josza algorithm required the function to either be constant or balanced.

Implementing a Linear Boolean Function

For s = 5 = 101_2 the function is evaluated using this circuit:

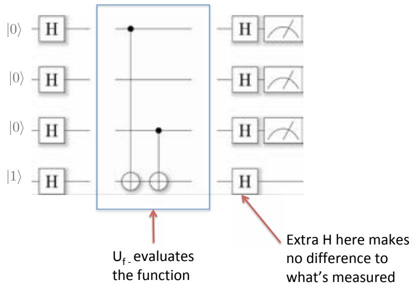


The bits of s determine the location of the CNOTs.

Every linear, Boolean function has a circuit of the same form.

PHYC90045 Introduction to Quantum Computing

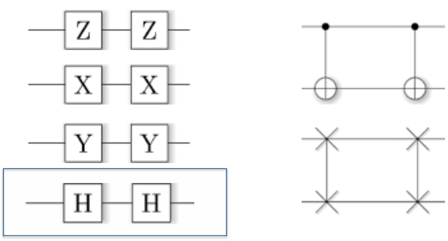
Full BV circuit for s=5



PHYC90045 Introduction to Quantum Computing

Many gates square to identity

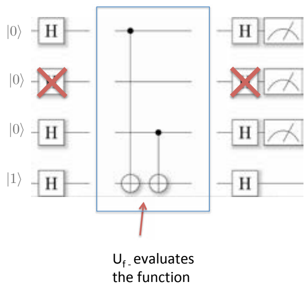
Compilation tips: Most physicists looking at quantum circuit diagrams aren't multiplying matrices in their head. They're identifying common patterns.



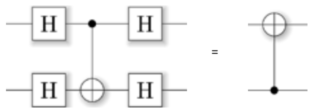
All of these combinations square to the identity (do nothing). You can check these on the QUI.

PHYC90045 Introduction to Quantum Computing

BV circuit for s=5



Circuit identity: Inverted CNOT

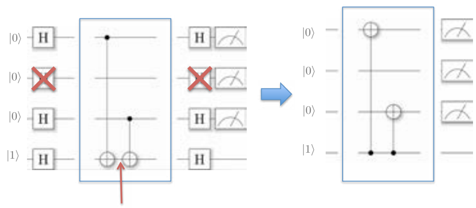


Exercise: You can verify this by writing out the matrices and multiplying!

PHYC90045 Introduction to Quantum Computing

Simple explanation of BV

Hadamard gates "conjugating" CNOT:

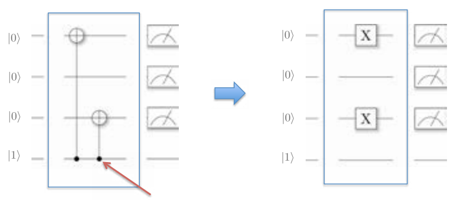


Insert H^2=I here

We can determine s with just one query, by making use of quantum superposition.

PHYC90045 Introduction to Quantum Computing

Simplifying circuit



Control is a 1, so these operations always happen

If in doubt, check using QUI

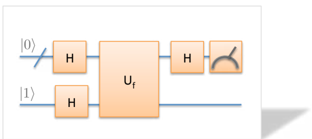
BV Solution

This circuit will measure:
101 = 5
which is correct (s=5)

A similar reduction would work for any s, but let us prove that formally.

Bernstein-Vazirani algorithm

The circuit is the same as for the Deutsch-Josza algorithm:

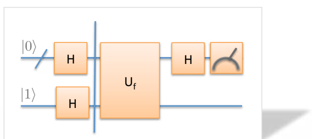


The guarantees on f are different:

$$f(x) = x \cdot s \quad \mathrm{mod}\ 2$$

Recall: Deutsch-Josza algorithm required the function to either be constant or balanced.

BV algorithm explained

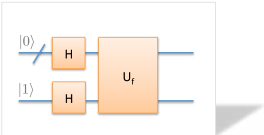


State after the initial Hadamard gates:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Sum of all computational basis states (again)

Recall: General Function Phase Kickback

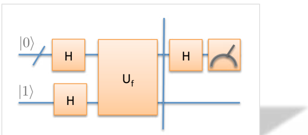


Using phase kickback, after the function has been applied:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x (-1)^{f(x)} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

If the function evaluates to "1" then the target qubit is flipped, and we pick up a phase. Otherwise, there is no phase applied. This is a simple way to write that.

BV algorithm explained

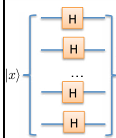


Using phase kickback, after the function has been applied:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \text{Phase kickback}$$

$$= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{x\cdot s} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \text{Since } f(x) = x \cdot s \mod 2$$

Recall: Hadamard applied to a general state



$$\mathrm{H}^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} a_z |z\rangle$$

Amplitude a_z -> how many times does the binary representation of z and x have 1's in the same location?

$$x_0 z_0 + x_1 z_1 + x_2 z_2 + \ldots + x_n z_n$$

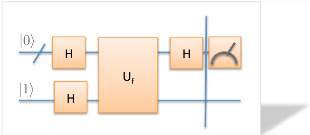
Shorthand for the bitwise dot product is: $x \cdot z = \sum_{j=0}^{n} x_j z_j$

When 1's in the same location, we get a sign change -> $(-1)^{x\cdot z}$

Hadamards applied to a general state (n qubits, N = 2^n):

$$\mathrm{H}^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} (-1)^{x\cdot z} |z\rangle$$

BV algorithm explained



Considering the upper register only:

$$|\psi\rangle = \mathrm{H}^{\otimes n}\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{x\cdot s}|x\rangle$$

$$= \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{x\cdot s}\frac{1}{\sqrt{N}}\sum_{z=0}^{N-1}(-1)^{x\cdot z}|z\rangle$$

$$= \frac{1}{N}\sum_{x=0}^{N-1}\sum_{z=0}^{N-1}(-1)^{x\cdot(s\oplus z)}|z\rangle = \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{x=0}^{N-1}(-1)^{x\cdot(s\oplus z)}\right)|z\rangle$$

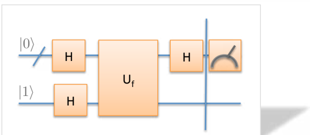
$$x\oplus z = x_0 + z_0 \mod 2, x_1 + z_1 \mod 2, \ldots$$

$$\mathrm{H}^{\otimes n}|x\rangle = \frac{1}{\sqrt{N}}\sum_{z=0}^{N-1}(-1)^{x\cdot z}|z\rangle$$

H's applied to basis state

PHYC90045 Introduction to Quantum Computing

BV algorithm explained



Simplifying the sum:

$$|\psi\rangle = \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{x=0}^{N-1}(-1)^{x\cdot(s\oplus z)}\right)|z\rangle$$

$$= \frac{1}{N}\sum_{z=0}^{N-1}(-1)^{0}|s\rangle$$

$$= |s\rangle$$

$$\sum_{x=0}^{N-1}(-1)^{x\cdot b} = \begin{cases} N, & b = 0 \\ 0, & b \neq 0 \end{cases}$$

This sum (over x) is zero unless

$$s\oplus z = 0$$

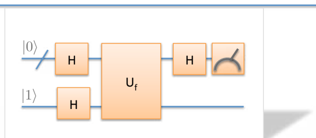
That is, z and s are bitwise identical, ie.

$$z = s$$

We will therefore measure s with certainty – the aim of the algorithm.

PHYC90045 Introduction to Quantum Computing

Bernstein-Vazirani Algorithm



Given a Boolean function, *f*:

$$f(x) = x\cdot s \mod 2$$

find s.

$$x\cdot s = \sum_i x_i s_i$$

- **Classical algorithm** *needs n queries*
- ***Quantum algorithm*** *needs just 1 query.*

Third Quantum Algorithm:
Simon's algorithm

Simon's Problem

Given a 2-to-1 function, f, such that

$$f(x) = f(x \oplus a)$$

Find a.

Unlike the previous two examples, here the range of f(x) is Z, integers.
Simon's algorithm is an example of a "Hidden (Abelian) subgroup problem" (HSP) and was the inspiration for Shor's factoring algorithm.

Example of a hidden a

x	f(x)
000	0
001	1
010	2
011	3
100	2
101	3
110	0
111	1

f(001) = f(111)

We would like to find the hidden 'a' s.t.
$$f(x) = f(x \oplus a)$$

In this case:
$a=110_2=6$

Solving Simon's problem classically

Just try different inputs until you see a collision:

f(000) = 0
f(011) = 3
f(111) = 1
f(010) = 2
f(001) = 1

Actually this is equivalent to the famous "birthday" problem, and takes fewer queries than you might expect. Probabilistically, if there are N different inputs we need

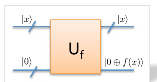
$$O(\sqrt{N})$$

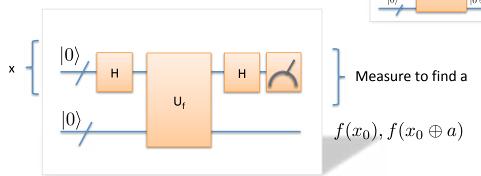
Evaluations of the function before we find a collision.

Simon's algorithm does the same with O(n) queries.

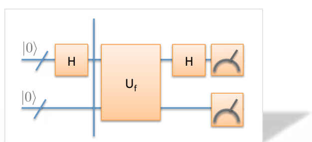
Simon's algorithm circuit

Randomly measure a result of the function. Collapse to a superposition of inputs which give that value. Send these through Hadamard gates, and measure:





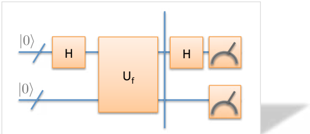
Simon's algorithm



After the initial Hadamard gates:

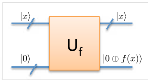
$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle |0\rangle$$

Simon's algorithm

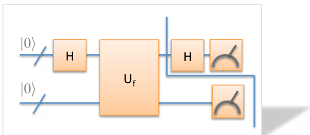


After evaluation of the function:

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x U_f|x\rangle|0\rangle$$
$$= \frac{1}{\sqrt{N}}\sum_x |x\rangle|f(x)\rangle$$



Simon's algorithm



It's easiest to consider that the bottom register is measured first. Before measurement the state is:

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x |x\rangle|f(x)\rangle$$

Some value, $f(x_0)$ will be measured at random, and the top register collapses to:

$$|\psi\rangle = \frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}}$$

Example: Measuring function

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x |x\rangle|f(x)\rangle$$
$$= \frac{1}{\sqrt{8}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle + |2\rangle|2\rangle + |3\rangle|3\rangle + |4\rangle|2\rangle + |5\rangle|3\rangle + |6\rangle|0\rangle + |7\rangle|1\rangle\right)$$

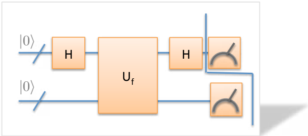
If we measure the second register, and measure obtain "3", the state collapses to only those states compatible with this measurement:

$$|\psi'\rangle = \frac{|3\rangle|3\rangle + |5\rangle|3\rangle}{\sqrt{2}}$$
$$= \frac{|3\rangle + |5\rangle}{\sqrt{2}} \otimes |3\rangle$$

First register: $|\psi\rangle = \dfrac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}}$

x	f(x)
000	0
001	1
010	2
011	3
100	2
101	3
110	0
111	1

Simon's algorithm

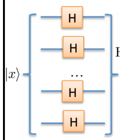


We now apply Hadamard to the top register:

$$|\psi\rangle = H^{\otimes n} \frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}}$$

PHYC90045 Introduction to Quantum Computing

Hadamard applied to a general state



$$\mathrm{H}^{\otimes n}|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} a_y |y\rangle$$

Amplitude a_y -> how many times does the binary representation of y and x have 1's in the same location?

$$x_0 y_0 + x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

Shorthand for the bitwise dot product is: $x \cdot y = \sum_{j=0}^{n} x_j y_j$

When 1's in the same location, we get a sign change -> $(-1)^{x \cdot y}$

Hadamards applied to a general state (n qubits, N = 2^n):

$$\mathrm{H}^{\otimes n}|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} (-1)^{x \cdot y} |y\rangle$$

(changed dummy index to y)

PHYC90045 Introduction to Quantum Computing

Simon's algorithm

$$|\psi\rangle = H^{\otimes n} \frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}}$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_y \left((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a) \cdot y} \right) |y\rangle$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_y (-1)^{x_0 \cdot y} \left(1 + (-1)^{a \cdot y}\right) |y\rangle$$

The amplitude of any state, y, is zero unless:

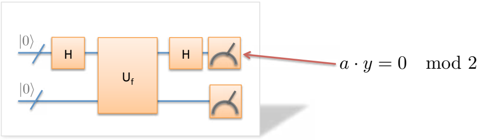
$$a \cdot y = 0 \mod 2$$

Therefore, the state therefore becomes:

$$|\psi\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{a \cdot y = 0} (-1)^{x_0 \cdot y} |y\rangle$$

Simon's algorithm

$$|\psi\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{a \cdot y = 0} (-1)^{x_0 \cdot y} |y\rangle$$



Each time we measure, we randomly measure a "y" which is orthogonal to "a":

Obtain n random y's this way and **perform Gauss/Jordan elimination** to obtain "a"

Example of Simon's algorithm

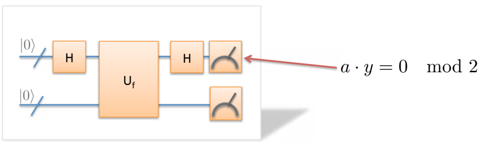
x	f(x)
000	0
001	1
010	2
011	3
100	2
101	3
110	0
111	1

We would like to find the hidden 'a' s.t.

$$f(x) = f(x \oplus a)$$

In this case, $a=110_2=6$

Running the circuit



We run the circuit, and at random, obtain measure the results:

001
110
111

We want to find,
$a=110_2=6$

In matrix form

We know that $a \cdot y = 0 \mod 2$

We have three values of 'y' for which this is true, so we can write a system of linear equations for the bits of 'a':

$$\mathbf{Y}\vec{a} = \vec{0}$$

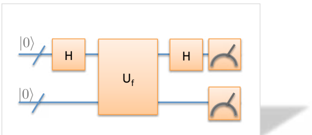
Measured values
001
110
111

$$\left[\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \quad \text{Solving for a}$$

Solution is degenerate. a=(0,0,0) or a_1=a_2=1 ie. **a=(1,1,0)**

Simon's Algorithm



Given a 2-to-1 function, f, such that

$$f(x) = f(x \oplus a)$$

Find a.

Classical algorithm: $O(\sqrt{N})$ Queries to the oracle (probabilistically)

Quantum algorithm: $O(n)$ Queries to the oracle