**Quantum Algorithms**

Personal notes based on lecture material and assigned reading from Princeton's [ELE 396: Quantum Computing](https://registrar.princeton.edu/course-offerings/course_details.xml?courseid=010006&term=1154), taught by Stephen Lyon.

**Important Identities**

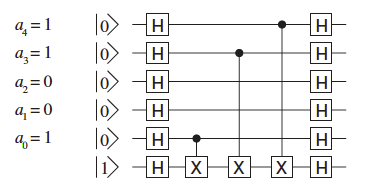
N-bit Hadamard

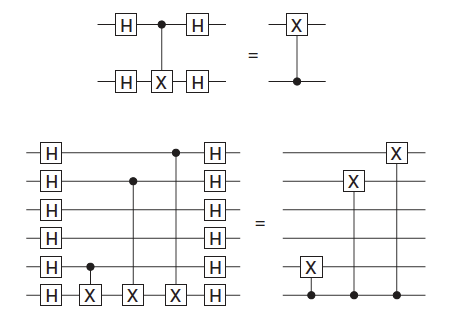
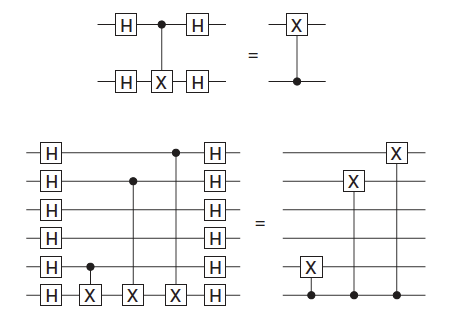
**Deutsch-Jozsa**

* Setup
  + Input: a black-box for computing unknown function
  + Details: is either a constant or a balanced function
    - If it is constant, all inputs map to either 0 or 1
    - If it is balanced, exactly half the inputs map to 0 and the other half to 1
  + Problem: determine whether is constant or balanced by making queries
* Input
* Apply (n+1)-bit Hadamard to resulting in
* Apply
* Apply n-bit Hadamard again
* Measure input registers in computational basis ()
  + Coefficient of is given by
    - If is balanced, this sum is 0.
    - If is constant, this sum is .
  + If measurement yields , must be constant. If measurement yields any other value, must be balanced
* Classical v. quantum algorithm analysis
  + Deterministic: classical requires queries, while quantum requires 1
  + Probabilistic: classical can solve Deutch-Jozsa with probability of error at most using 2 queries, and less than with queries
  + *Linear gap* in the case of exponentially small error (not that impressive!)

**Bernstein-Vazirani**

* Setup
  + Input: a black-box for computing unknown function
  + Details:
  + Problem: want to determine n-bit “secret” hard-coded value
* Same procedure
  + Know that
  + Measuring input register *always yields*, and so we are done!
    - Coefficient of is given by
    - Coefficient of is given by
* A second way of looking at this problem exists
  + Circuit diagram
    - Key idea: can be represented as series of CNOTs on the output register, controlled by those input bits that correspond to nonzero bits of
    - Applying n-bit Hadamards before and after uncovers the hardcoded value of in the blackbox





* Classical v. quantum algorithm analysis
  + Deterministic: classical computer must call subroutine times to determine ( bits of) while a quantum computer need only call the subroutine once

**Simon’s Problem**

* Setup
  + Input: a black-box for computing unknown function
  + Details: iff
  + Problem: want to determine period of
* Input
  + is the input register and is the output register
* Apply n-bit Hadamard to input
* Apply
* Measure output register
  + Some value of corresponding to random and
  + Resulting state
* Apply n-bit Hadamard to input

Note that is a modulo-2 bitwise inner product. Clearly, if , the second summation equals 0.

* Measure the input register
  + Yields random satisfying
  + Gives a linear equation in the bits of
    - E.x.
  + invocations should yield linearly independent equations (from which can unambiguously be determined) with probability at least
* Classical v. quantum algorithm analysis
  + Classically: would have to feed subroutine different values of for appreciable chance of finding a pair that XOR to **a** (birthday problem)
    - Exponential in number of bits
  + Quantum: need only a linear number of invocations () to have a very good chance of accurately determining
    - Note Simon’s algorithm is a zero-error algorithm – though it possible that invocations may not be sufficient to determine , there is no chance of getting an *incorrect answer*

**Quantum Fourier Transform**

**Shor’s Algorithm**

* Input
  + is often
* Apply n-bit Hadamard
* Apply
  + is randomly selected from [2, N-1]
* Measure output register, leaving input register in superposition of values which give that particular output
* Now apply the Quantum Fourier Transform () to the input register
* Measuring the QFTed input register yields with probability
* Note that is strongly peaked where is an integer
  + Assuming is large enough,

averages out to 0, *except* when

* + Alternatively, is likely near where is an integer
  + If we use bits, probability that measured is within of
* Determine partial sums of continued fractions expansions of
  + Denominators of continued fractions are candidates for (order of )
    - Test to verify
  + Given
    - There is a 50% chance that is even
    - If so, is a nontrivial factor of N
      * This follows from:
    - If not, select new and repeat

**Grover’s Algorithm**

* Setup
  + Input: a blackbox for computing unknown function
  + Details: if and if
  + Problem: want to determine secret , where
* Classical approach
  + Guess possible values of
    - Assumption: can’t do better than random guessing
  + Need guesses on average
* Input
* Apply (n+1)-bit Hadamard
* Apply blackbox

where

Applying thus reflects the state across the axis to

We can express as the operator since

* Next, we reflect around with the operator
  + Note that to reflect around we subtracted twice the projection along , which is equivalent to reflecting around and then negating
  + To reflect around , we thus reflect around and negate

Note that

But

Crucially, has a circuit implementation.

Now our state makes an angle of with respect to where previously it made an angle of

* We repeat the previous two steps until is approximately equal to . Note that each iteration adds to the angle between and .

Then approximately

iterations are needed to yield . But , so comes out to be