Quantum Computation- Quantum Algorithms-Hadamard Gates

- An important step in quantum algorithms is to use Hadamard gates to create superposition states:

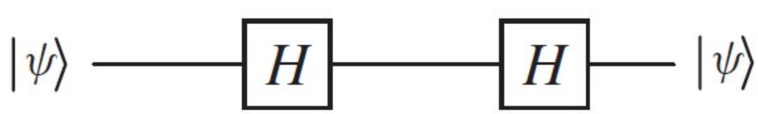
$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\begin{aligned} H|\psi\rangle = \alpha H|0\rangle + \beta H|1\rangle &= \alpha\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) + \beta\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\ &= \left(\frac{\alpha + \beta}{\sqrt{2}}\right)|0\rangle + \left(\frac{\alpha - \beta}{\sqrt{2}}\right)|1\rangle \end{aligned}$$

Quantum Computation- Quantum Algorithms-Hadamard Gates

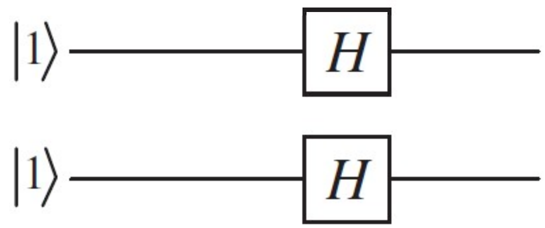
- If we apply a Hadamard gate twice, we get the original state back:

$$
\begin{aligned}
H\left[\left(\frac{\alpha+\beta}{\sqrt{2}}\right)|0\rangle+\left(\frac{\alpha-\beta}{\sqrt{2}}\right)|1\rangle\right] &= \left(\frac{\alpha+\beta}{\sqrt{2}}\right)H|0\rangle+\left(\frac{\alpha-\beta}{\sqrt{2}}\right)H|1\rangle \\
&= \left(\frac{\alpha+\beta}{\sqrt{2}}\right)\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)+\left(\frac{\alpha-\beta}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \\
&= \left(\frac{\alpha+\alpha+\beta-\beta}{2}\right)|0\rangle+\left(\frac{\alpha-\alpha+\beta+\beta}{2}\right)|1\rangle \\
&= \alpha|0\rangle+\beta|1\rangle=|\psi\rangle
\end{aligned}
$$



Quantum Computation- Quantum Algorithms-Hadamard Gates

- Now consider what happens when we apply two Hadamard gates in parallel. Let’s say that we do this for the state |1>|1>.



$|\psi\rangle = ?$

$$(H \otimes H)|1\rangle|1\rangle = (H|1\rangle)(H|1\rangle) = \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$= \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

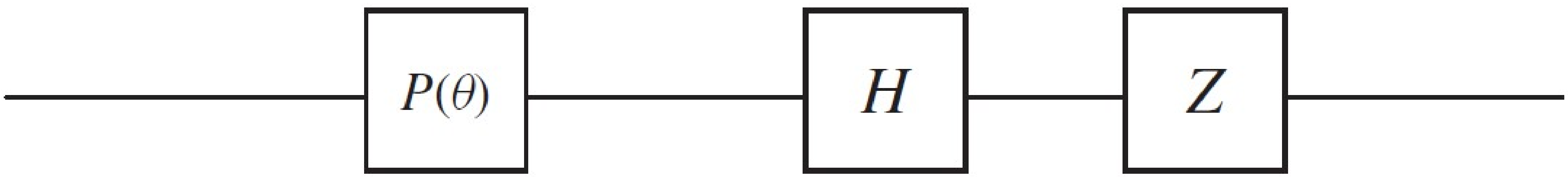
Quantum Computation- Quantum Algorithms-The Phase Gate

- Another useful gate that can be applied in the development of quantum algorithms is a variation of the phase gate we met previously, called the *discrete phase gate. We denote the discrete phase gate by R_k, where*

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{(2\pi i/2^k)} \end{pmatrix}$$

Quantum Computation- Quantum Algorithms-Serial and Parallel Operations

- The matrix representation of serial operations is written down by multiplying the matrices in *reverse order.*



$$ZHP(\theta)$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\theta} \\ -1 & e^{i\theta} \end{pmatrix}$$

Quantum Computation- Quantum Algorithms-Serial and Parallel Operations

- When quantum operations are performed in parallel ,we compute the tensor product

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

Quantum Computation- Quantum Algorithms-Quantum Interference

- The application of a Hadamard gate to an arbitrary qubit is an example of *quantum interference. Let's recall what happens when we calculate* H|ψ> for |ψ=α|0>+β|1>. *We get*

$$H|\psi\rangle = \left(\frac{\alpha+\beta}{\sqrt{2}}\right)|0\rangle + \left(\frac{\alpha-\beta}{\sqrt{2}}\right)|1\rangle$$

$$\alpha \rightarrow \frac{\alpha+\beta}{\sqrt{2}} \qquad \beta \rightarrow \frac{\alpha-\beta}{\sqrt{2}}$$

Quantum Computation- Quantum Algorithms-Quantum Interference

- Positive interference with regard to the basis state |0>. The two amplitudes add to increase the probability of finding 0 upon measurement. In fact in this case it goes to unity meaning we are certain to find 0.
- Negative interference where by the terms |1> and −|1> cancel. We go from a state where there was a 50% chance of finding 1 upon measurement to one where there is no chance of finding 1 upon measurement.
- Quantum interference allows us to gain information about a function *f (x) that* depends on evaluating the function at many values of *x. That is, interference* allows us to deduce certain *global properties of the function.*

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- Quantum parallelism can be described as the ability to evaluate the function *f (x) at many* values of *x simultaneously.*
- The algorithm we are going to describe is called Deutsch's algorithm.
- Let's see how to do this by considering a very simple function, one that accepts a single bit as input and produces a single bit as output.
- $$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x = 1 \end{cases}$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- Two more examples are the constant functions

$$f(x) = 0, \quad f(x) = 1$$

- The final example is the bit flip function

$$f(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x = 1 \end{cases}$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- The identity and bit flip functions are called *balanced because the outputs* are opposite for half the inputs.
- What we're going to see in the following development is that Deutsch's algorithm will let us put together a state that has all of the output values of the function associated with each input value in a superposition state.
- Then we will use quantum interference to find out if the given function is constant or balanced.

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- The first step in developing this algorithm is to imagine a unitary operation denoted by U_f *that acts on two qubits. It leaves the first qubit alone and produces* the exclusive or (denoted by $\oplus$) of the second qubit with the function f *evaluated* with the first qubit as argument. That is,

$$U_f|x, y\rangle = |x, y \otimes f(x)\rangle$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- (note that *x, y* $\in$*{0, 1}*). *Now, since |x> is a qubit, it can be in a superposition state.* Let's specifically start with an initial state |0> and apply a Hadamard gate

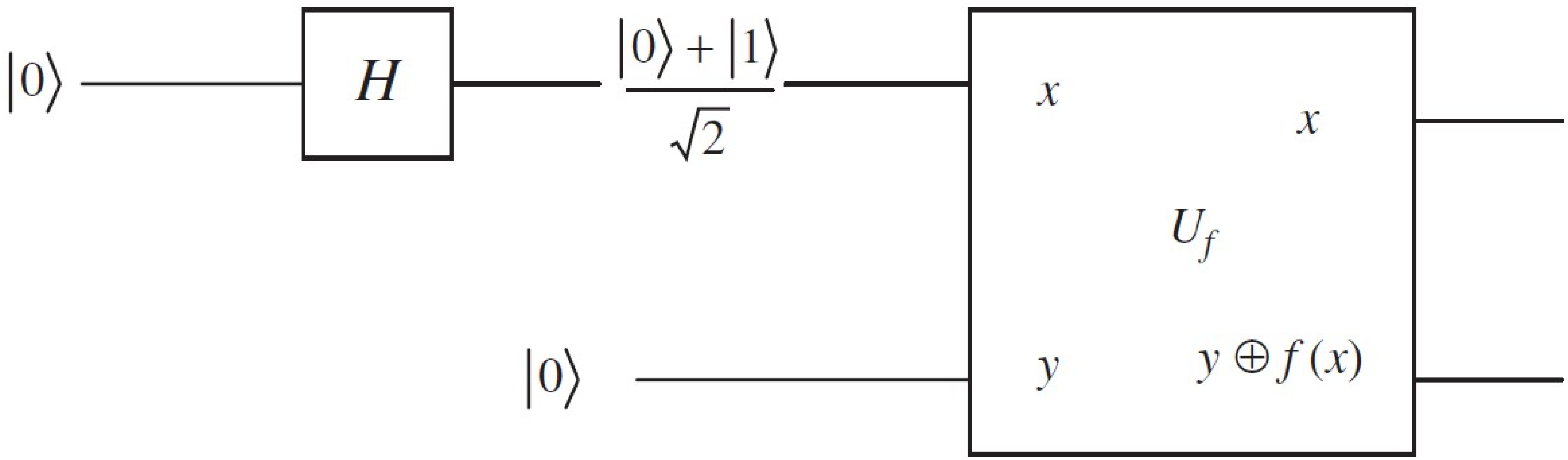
$$U_f\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|0\rangle=\frac{1}{\sqrt{2}}(U_f|00\rangle+U_f|10\rangle)=\frac{|0,0\oplus f(0)\rangle+|1,0\oplus f(1)\rangle}{\sqrt{2}}$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- This circuit has produced a superposition state that has information about every possible value of *f (x), in a single step. Suppose that f (x) is the identity function.* Then the final state is

$$U_f\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|0\rangle=\frac{|0,0\oplus f(0)\rangle+|1,0\oplus f(1)\rangle}{\sqrt{2}}=\frac{|00\rangle+|11\rangle}{\sqrt{2}}$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism



Circuit diagram for ***$U_f|x, y> = |x,y \otimes f(x)>$*** ***where we take the first qubit to be in a*** **superposition and the second qubit to be |0>.**

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- Recall that $0\oplus0=1\oplus1=0$, $0\oplus1=1\oplus0=1$, so more generally the output of this circuit is

$$U_f\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|0\rangle=\frac{|0,f(0)\rangle+|1,f(1)\rangle}{\sqrt{2}}$$

- We have a superposition state with all pairs of *x, f (x)* represented. But remember how quantum measurement works.

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- After measurement the system is in a state proportional to $|x>|f(x)>$ *for that single and specific value* of *x.*
- Moreover the value of *x that we obtain is completely random. So what we* have so far is the same as evaluating the function *f (x) at some randomly chosen x.* That doesn’t seem very useful.
- For a simple function on bits we can learn the value of *f (0) or f (1), but not both simultaneously*

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- While it seems like we haven't gotten anything any more useful than what we can do with a classical computer, it's in fact worse:
- we can't choose which value to reveal, *f (0)or f (1), that occurs randomly.*
- Deutsch's algorithm takes what we have done so far to exploit the fact that the system is in a superposition state *to obtain information about a global* property of the function—whether or not it is constant *f (0)=f (1) or balanced f (0) ≠ f (1). It does this by computing*
- $$|\psi_{out}\rangle = (H \otimes I)U_f(H \otimes H)|0\rangle|1\rangle$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- Deutsch's algorithm is implemented by the following steps:
  - Apply Hadamard gates to the input state |0>|1> to produce a product state of two superpositions.
  - Apply U_f *to that product state.*
  - Apply a Hadamard gate to the first qubit leaving the second qubit alone.

Quantum Computation- Quantum Algorithms-Quantum Parallelism

$$
\begin{aligned}
(H \otimes H)|0\rangle|1\rangle &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
&= \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)
\end{aligned}
$$

- By doing this, we provide a superposition state as input that will contain all possible values of the combination *(x, f (x)) once we apply* U_f.

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- Now let's apply U_f *to each term of* $(H\otimes H)|0>|1>$. For the first term we have

- $U_f|00> = |0, 0 \oplus f(0)> = (1 - f(0))|00> + f(0)|01>$

- This result takes into account the possibilities $0\oplus f(0)=0$ and $0\oplus f(0)=1$. To see how this works, note that if $f(0)=0$, then $0\oplus f(0)=0\oplus 0=0$ and

- $|0, 0 \oplus f(0)> = (1 - f(0))|00> + f(0)|01> = |00> + (0)|01> = |00>$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- On the other hand, if f (0)=1, then $0 \oplus f(0) = 0 \oplus 1 = 1$ and
- $|0, 0 \oplus f(0)\rangle = (1 - f(0))|00\rangle + f(0)|01\rangle = (0)|00\rangle + (1)|01\rangle = |01\rangle$
- Similar logic applied to the other terms gives

$$U_f|01\rangle = |0, 1 \oplus f(0)\rangle = f(0)|00\rangle + (1 - f(0))|01\rangle$$

$$U_f|10\rangle = |0, 1 \oplus f(0)\rangle = (1 - f(1))|00\rangle + f(1)|01\rangle$$

$$U_f|11\rangle = |0, 1 \oplus f(1)\rangle = f(1)|10\rangle + (1 - f(1))|11\rangle$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- Hence:

$$\begin{aligned}|\psi'\rangle &= U_f(H\otimes H)|0\rangle|1\rangle \\ &= (1-f(0))|00\rangle + f(0)|01\rangle + f(0)|00\rangle + (1-f(0))|01\rangle \\ &\quad +(1-f(1))|00\rangle + f(1)|01\rangle + f(1)|10\rangle + (1-f(1))|11\rangle\end{aligned}$$

- To get the final output state of Deutsch's algorithm, we apply H⨂I to |ψ'>. The Hadamard gate is applied to the first qubit, and the second qubit is left alone.

Quantum Computation- Quantum Algorithms-Quantum Parallelism

$$
\begin{aligned}
&(H \otimes I)[(1 - f(0))|00\rangle + f(0)|01\rangle] \\
&\quad = (1 - f(0))(H|0\rangle) \otimes |0\rangle + f(0)(H|0\rangle) \otimes |1\rangle \\
&\quad = (1 - f(0))\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes |0\rangle + f(0)\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes |1\rangle \\
&\quad = (1 - f(0))\left(\frac{|00\rangle + |10\rangle}{\sqrt{2}}\right) + f(0)\left(\frac{|01\rangle + |11\rangle}{\sqrt{2}}\right)
\end{aligned}
$$

Quantum Computation- Quantum Algorithms-Quantum Parallelism

- Applying H⊗I *to all the terms in and doing some algebra gives the final* output state of Deutsch's algorithm as

$$|\psi_{out}\rangle = (1 - f(0) - f(1))|0\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) + (f(1) - f(0))|1\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$|\psi_{out}\rangle = \pm|0\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \qquad (f(0) = f(1))$$

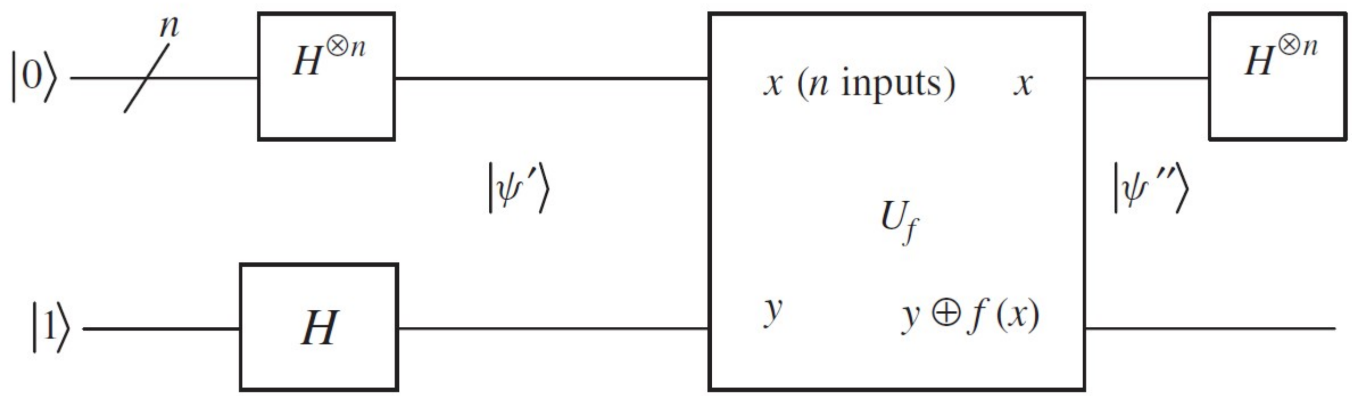
$$|\psi_{out}\rangle = \pm|1\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \qquad (f(0) \neq f(1))$$

Quantum Computation- Quantum Algorithms-Deutsch-Jozsa Algorithm

- The *Deutsch-Jozsa algorithm is a eneralization of Deutsch's algorithm.*
- Again, this algorithm allows us to determine whether a function *f (x) is constant or balanced,* but this time the function has multiple input values.
- If f(x) is constant, then the output is the same for all input values x. If the function is balanced, then f(x)=0 for half of the inputs and f(x)=1 for the other half of the inputs, and vice versa.

Quantum Computation- Quantum Algorithms-Deutsch-Jozsa Algorithm

- We start with an initial state that includes n qubits in the state |0> and a single qubit in the state |1>. Hadamard gates are applied to all qubits.



The Deutsch-Jozsa algorithm generalizes Deutsch's algorithm to handle a function with n input values and determine whether or not it is constant or balanced

Quantum Computation- Quantum Algorithms-Deutsch-Jozsa Algorithm

- We start off by calculating

$$|\psi'\rangle = (H^{\otimes n})(|0\rangle^{\otimes n}) \otimes (H|1\rangle)$$

$$|\psi'\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$|\psi''\rangle = \frac{1}{\sqrt{2^n}} \sum_{x} (-1)^{f(x)}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Quantum Computation- Quantum Algorithms-Deutsch-Jozsa Algorithm

- Applying a Hadamard gate to an n qubit state |x> gives

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_y (-1)^{x \cdot y} |y\rangle$$

- So the final output state is

$$|\psi_{out}\rangle = \frac{1}{2^n} \sum_y \sum_x (-1)^{x \cdot y + f(x)} |y\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

Quantum Computation- Quantum Algorithms-Deutsch-Jozsa Algorithm

- Now we measure the *n inputs.*
- There are two possible measurement results on |y> *(which is the state of n inputs at this stage) that are of interest. The possible results are as follows:*
  - Measurement of the first n input qubits in $|\psi_{out}>$ returns all 0’s. In this case f(x) is constant.
  - Otherwise, if at least one of the qubits in |y> is found to be a 1 on measurement, f (x) is balanced.