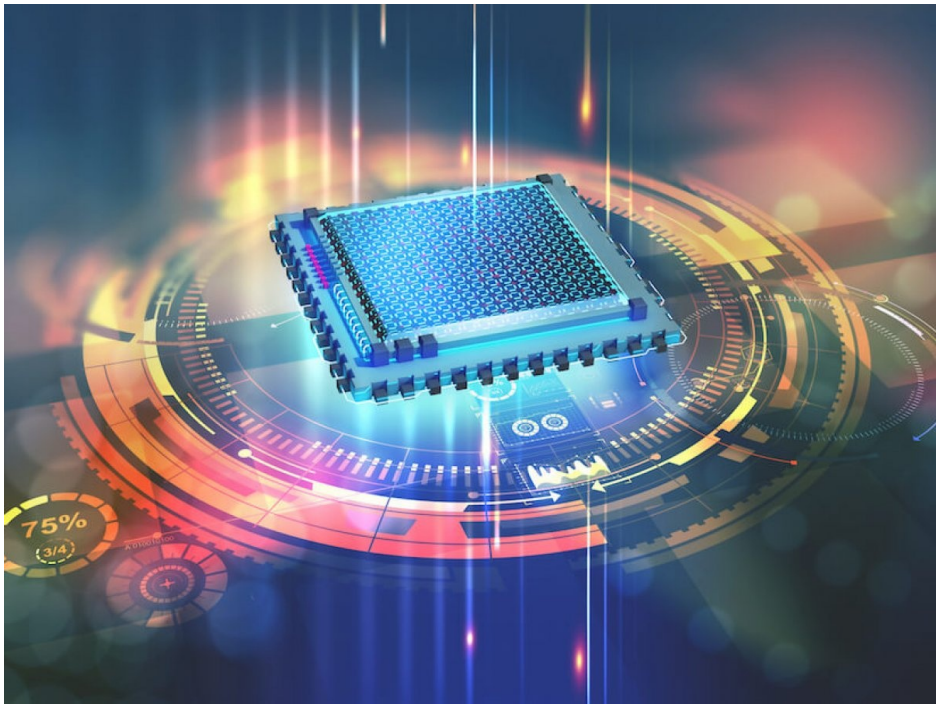
Deutsch’s Problem



Last class

- Unitary circuits, input and output registers
- Quantum Parallelism
- No cloning
- Uncertainty principle

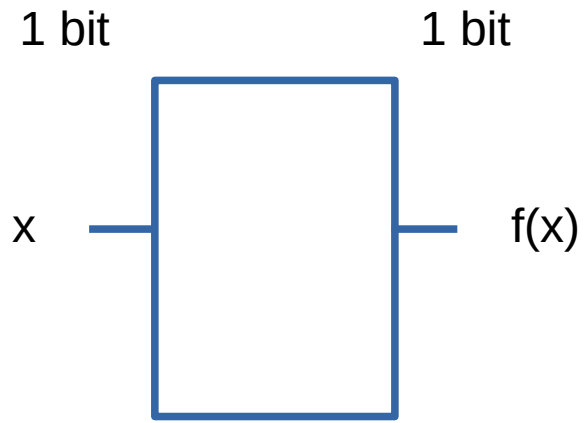
Today

- Deutsch’s problem
- Simplest example of quantum trade-off that sacrifices particular information to get relational information
- First “quantum supremacy” result

Is this it for quantum?

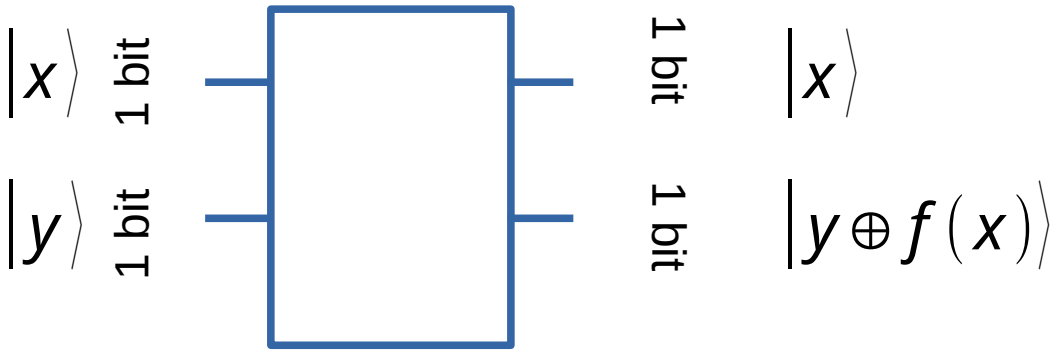
- Measuring destroys most of the information
- We can be more clever, apply more unitaries to the qubits before or after applying U_f
- We can learn something about the relations between different values of f(x)
- We lose the information of f(x)
- This tradeoff of information is typical of physics: Uncertainty principle

The setup



- Both input and output registers contain one bit
- Functions f that take one bit to one bit
- Two different ways to think about such f

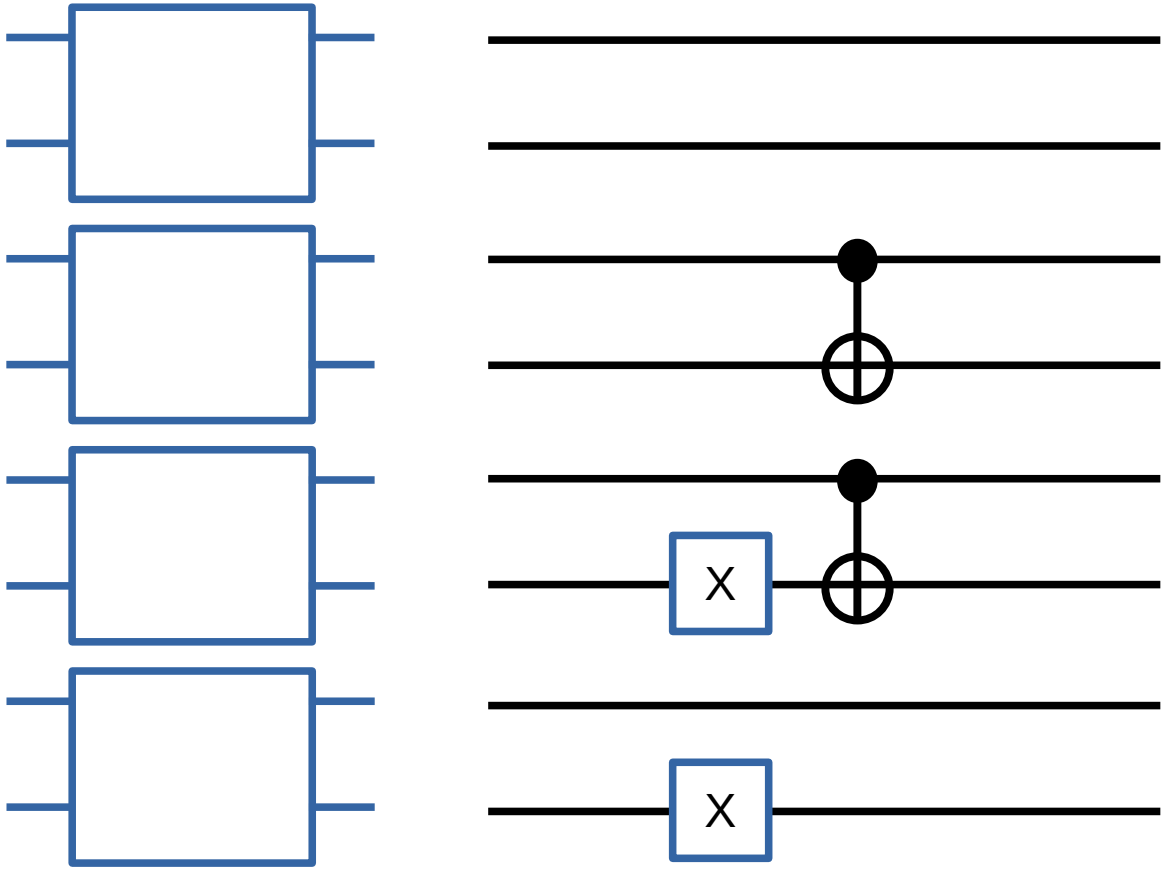
The setup, in quantum



How many functions?

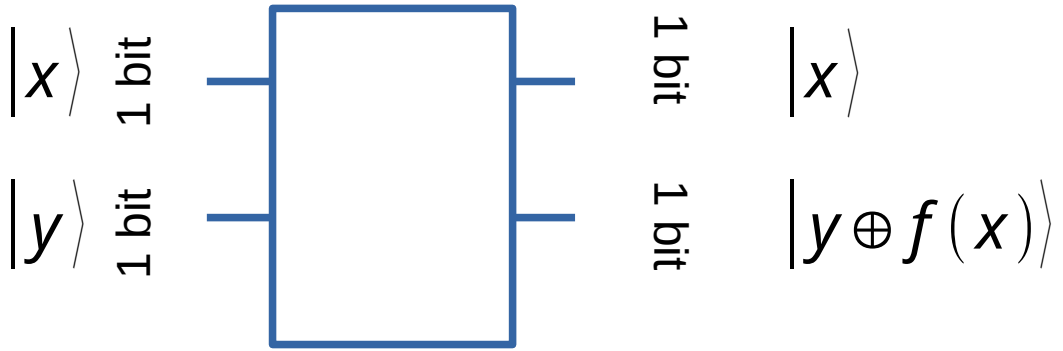
- How many different functions $f: \{0,1\}^n \rightarrow \{0,1\}$ are there that take as input n bits and output one bit?

The possibilities



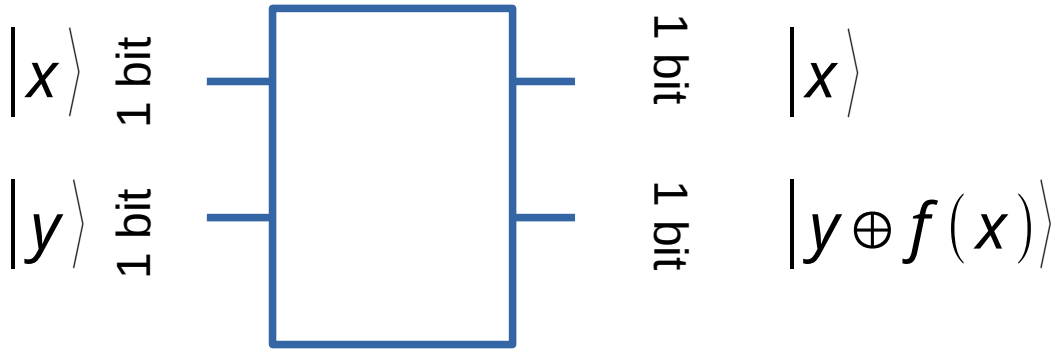
$f(0)$	$f(1)$
0	0
0	1
1	0
1	1

The setup, in quantum



- U_f is one of 4 possibly functions (table)
- We are given a black box that calculates one of the 4 f's by performing $U_f|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle$
- The black box performs one of the four computations, but we don't know which one
- We are allowed to use the box only once, what can we learn about f?

The setup, in quantum



Deutsch's problem

- We want to learn if f is constant (f(0)=f(1), satisfied by f_0 and f_3) or not, with one application of the black box

Classically

- We want to learn if f is constant (f(0)=f(1), satisfied by f_0 and f_3) or not, with one application of black box
- With a classical computer, we can either learn the value f(0) or f(1), so we can learn whether the function is one of (f_0, f_1) with f(0)=0 or (f_2, f_3) with f(0)=1
- A classical computer needs two queries to U_f to determine if it is constant or not!

In quantum

- With a quantum computer we can do better!
- Without learning any information about the values of f(0) or f(1), we can determine with one application of the black box if f is constant or not

More on Deutsch’s problem

- There is another way to look at Deutsch’s problem, which gives it nontrivial mathematical content
- One can think of x as specifying a choice of two different inputs to an elaborate subroutine that requires many additional Qubits, and one can think of f (x) as characterizing a two-valued property of the output of that subroutine
- For example f (x) might be the value of the millionth bit in the binary expansion of √(2 + x) so that f (0) is the millionth bit in the expansion of 2 while f (1) is the millionth bit of √3
- In this case the input register feeds data into the subroutine and the subroutine reports back to the output register.

More on Deutsch’s problem

- During the calculation the input and output registers will in general become entangled with the additional Qubits used by the subroutine
- If the entanglement persists to the end of the calculation, the input and output registers will have no final states of their own, and it will be impossible to describe the computational process as the simple unitary transformation we saw earlier
- However, it is possible to set things up so that at the end of the computation the additional Qubits required for the subroutine are no longer entangled with the input and output registers, so that the additional Qubits can indeed be ignored

More on Deutsch's problem

- The simple linear transformation then correctly characterizes the net effect of the computation on those two registers.
- Under the first interpretation, Deutsch's problem answers the question of whether f is or is not constant, allowing one to learn something about the nature of the black box that executes U_f without actually opening it up and looking inside.
- Under the second interpretation, it becomes the nontrivial question of whether the millionth bits of $\sqrt{2}$ and $\sqrt{3}$ agree or disagree.
- Under either interpretation, to answer the question with a classical computer we can do no better than to run the black box twice, with both 0 and 1 as inputs, and compare the two outputs.

Attempt 1: Superposition

- To solve Deutsch's problem we could try preparing the input register in superposition of 0 and 1.

$$U_f(H\otimes 1)(|0\rangle\otimes|0\rangle)=\frac{1}{\sqrt{2}}|0\rangle\otimes|f(0)\rangle+\frac{1}{\sqrt{2}}|1\rangle\otimes|f(1)\rangle$$

- We can measure, and get either 0, f(0) or 1, f(1), but there is no improvement over classical computation

The trick

- We can pre and post process the state to yield what we want

$$U_f(H\otimes H)(X\otimes X)(|0\rangle\otimes|0\rangle)=U_f(H\otimes H)(|1\rangle\otimes|1\rangle)=$$

$$=U_f\frac{|0\rangle-|1\rangle}{\sqrt{2}}\frac{|0\rangle-|1\rangle}{\sqrt{2}}=U_f\frac{1}{2}(|0\rangle|0\rangle-|1\rangle|0\rangle-|0\rangle|1\rangle+|1\rangle|1\rangle)=$$

$$=\frac{1}{2}(|0\rangle|0\oplus f(0)\rangle-|1\rangle|0\oplus f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)=$$

$$=\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

The trick

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

- Case 1: f(0)=f(1), the output state is:

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(0)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(0)\rangle)=$$

$$=\frac{1}{2}((|0\rangle-|1\rangle)|f(0)\rangle-(|0\rangle-|1\rangle)|1\oplus f(0)\rangle)=$$

$$=\frac{1}{2}(|0\rangle-|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle)$$

The trick

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

- Case 2: $f(0)\neq f(1), f(1)=|1\oplus f(0)\rangle$ the output state is:

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|1\oplus f(0)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus 1\oplus f(0)\rangle)=$$

$$=\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|1\oplus f(0)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|f(0)\rangle)=$$

$$=\frac{1}{2}((|0\rangle+|1\rangle)|f(0)\rangle-(|0\rangle+|1\rangle)|1\oplus f(0)\rangle)=$$

$$=\frac{1}{2}((|0\rangle+|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))$$

The trick

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

- Case 1: $f(0)=f(1)$ the output state is:

  $$\frac{1}{2}((|0\rangle-|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))$$

- Case 2: $f(0)\neq f(1), f(1)=|1\oplus f(0)\rangle$ the output state is:

  $$\frac{1}{2}((|0\rangle+|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))$$

The trick

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

- Case 1: $f(0)=f(1)$ the output state is:

  $$\frac{1}{2}((|0\rangle-|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))$$

- Case 2: $f(0)\neq f(1), f(1)=|1\oplus f(0)\rangle$ the output state is:

  $$\frac{1}{2}((|0\rangle+|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))$$

- Apply H to the ***input register***

The trick

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

- Case 1: $f(0)=f(1)$ the output state is:

$$\frac{1}{2}((|0\rangle-|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))$$

- After applying H:

$$\frac{1}{2}(H\otimes 1)((|0\rangle-|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))=$$

$$=\frac{1}{2}(|1\rangle(|f(0)\rangle-|1\oplus f(0)\rangle))$$

The trick

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

- Case 1: $f(0)\neq f(1), f(1)=|1\oplus f(0)\rangle$ the output state is:

$$\frac{1}{2}((|0\rangle+|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))$$

- After applying H:

$$\frac{1}{2}(H\otimes 1)((|0\rangle+|1\rangle)(|f(0)\rangle-|1\oplus f(0)\rangle))=$$

$$=\frac{1}{2}(|0\rangle(|f(0)\rangle-|1\oplus f(0)\rangle))$$

The trick

$$\frac{1}{2}(|0\rangle|f(0)\rangle-|1\rangle|f(1)\rangle-|0\rangle|1\oplus f(0)\rangle+|1\rangle|1\oplus f(1)\rangle)$$

- Case 1: $f(0)=f(1)$ after H:

$$\frac{1}{2}(|1\rangle(|f(0)\rangle-|1\oplus f(0)\rangle))$$

- Case 2: $f(0)\neq f(1)$, $f(1)=|1\oplus f(0)\rangle$ after H:

$$\frac{1}{2}(|0\rangle(|f(0)\rangle-|1\oplus f(0)\rangle))$$

- Measure the ***input register*** to decide if f is constant (get 1), or not (get 0)

Uncertainty principle

$$\frac{1}{2}(|0\rangle|f(0)\rangle - |1\rangle|f(1)\rangle - |0\rangle|1 \oplus f(0)\rangle + |1\rangle|1 \oplus f(1)\rangle)$$

- Measure the ***input register*** to decide if f is constant (get 1), or not (get 0)
- We learn whether the function is constant or not

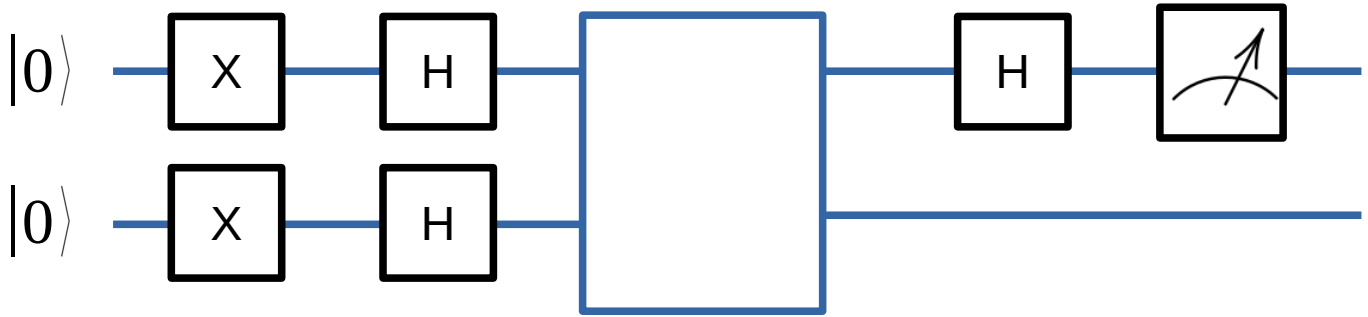
The whole operation

$$(H\otimes 1)U_f(H\otimes H)(X\otimes X)(|0\rangle\otimes|0\rangle)=$$

- $|1\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle-|1\oplus f(0)\rangle),\ if\ f(0)=f(1)$
- $|0\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle-|1\oplus f(0)\rangle),\ if\ f(0)\neq f(1)$
- Instead of querying the values of f(0) , f(1) the algorithm queries the XOR f(0)⊕f(1)
- However, we learn nothing about output, since it is in uniform superposition of 0 and 1 (remember that f(0) and 1⊕f(0) always have opposite values)

The circuit

$$(H \otimes 1) U_f (H \otimes H)(X \otimes X)(|0\rangle \otimes |0\rangle)$$



Deutsch’s problem, so what?

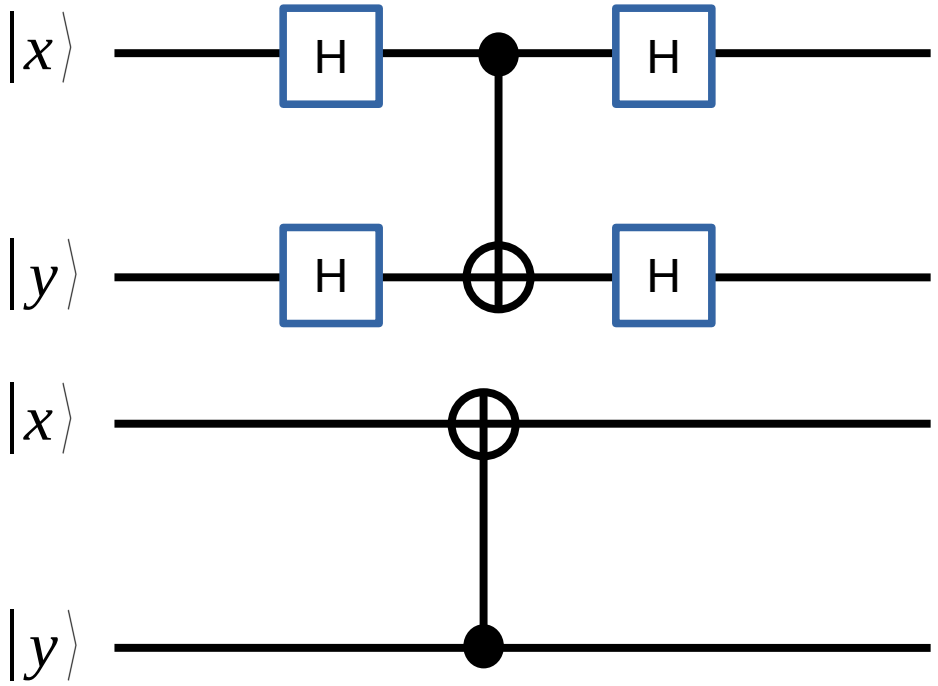
- Nice trick, but who really cares? Saving one query seems not so useful
- The idea - querying in superposition and putting the answer into the input register - is much broader and more powerful
- There is a straight line from Deutsch’s problem to Shor’s factoring algorithm
- You will get there in the second module

Our intuition fails

- Intuitively, X and H work on single qubits, and U_f is the identity on the first qubit
- Our intuition says that the first qubit can have no information on the output of f
- Yet, mathematics tells us that we find the result exactly on the first qubit
- This is due to entanglment: when entangled, one qubit can impact the other

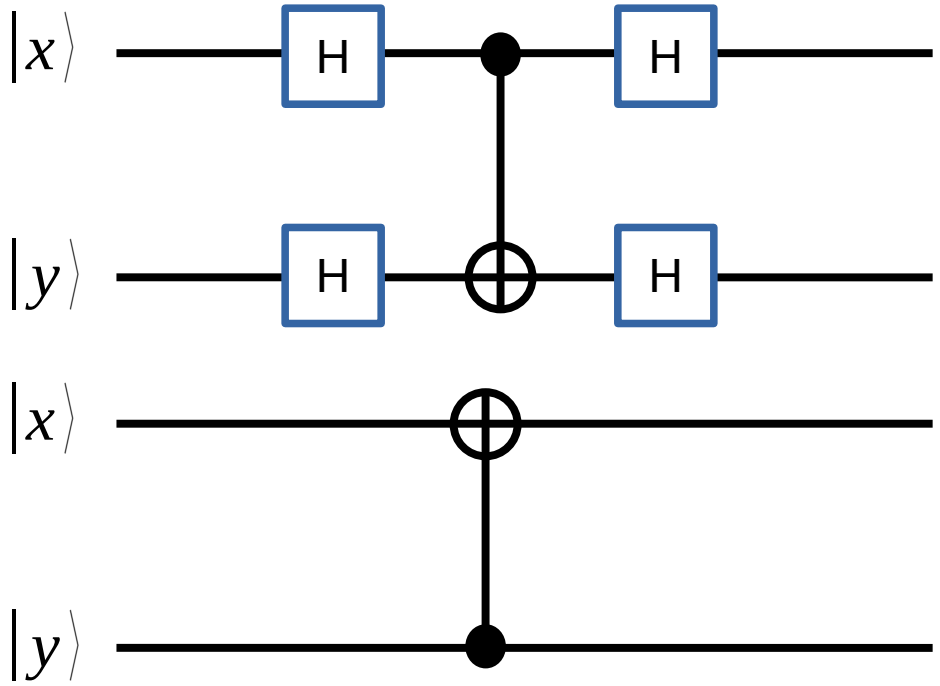
Our intuition fails

- Intuitively, which qubit should influence which qubit in the following two circuits?



Our intuition fails

- Intuitively, which qubit should influence which qubit in the following two circuits?



Actually, they define the same unitary!