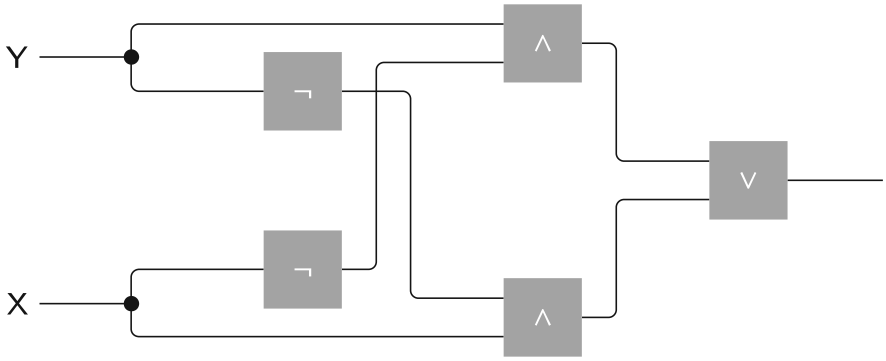
Circuits

In computer science, *circuits* are models of computation in which information is carried by wires through a network of *gates*, which represent operations on the information carried by the wires. *Quantum circuits* are a specific model of computation based on this more general concept.

Although the word "circuit" often refers to a circular path, circular paths aren't actually allowed in the circuit models of computation that are most commonly studied. That is to say, we usually consider *acyclic circuits* when we're thinking about circuits as computational models. Quantum circuits follow this pattern; a quantum circuit represents a finite sequence of operations that cannot contain feedback loops.

Boolean circuits

Here is an example of a (classical) Boolean circuit, where the wires carry binary values and the gates represent Boolean logic operations:



The flow of information along the wires goes from left to right: the wires on the left-hand side of the figure labeled X and Y are input bits, which can each be set to whatever binary value we choose, and the wire on the right-hand side is the output. The intermediate wires take whatever values are determined by the gates, which are evaluated from left to right.

The gates are AND gates (labeled $\wedge$), OR gates (labeled $\vee$), and NOT gates (labeled $\neg$). The functions computed by these gates will likely be familiar to many readers, but here they are represented by tables of values:

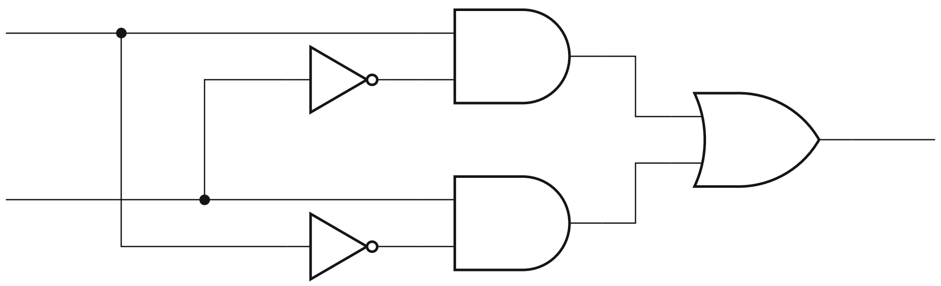
a	$\neg a$
0	1
1	0

ab	$a \wedge b$
00	0
01	0
10	0
11	1

ab	$a \vee b$
00	0
01	1
10	1
11	1

The two small, solid circles on the wires just to the right of the names X and Y represent *fan-out* operations, which simply create a copy of whatever value is carried on the wire on which they appear, allowing this value to be input into multiple gates. Fan-out operations are not always considered to be gates in the classical setting; sometimes they're treated as if they're "free" in some sense. When Boolean circuits are converted into equivalent quantum circuits, however, we do need to classify fan-out operations explicitly as gates to handle and account for them correctly.

Here's the same circuit illustrated in a style more common in electrical engineering, which uses conventional symbols for the AND, OR, and NOT gates:



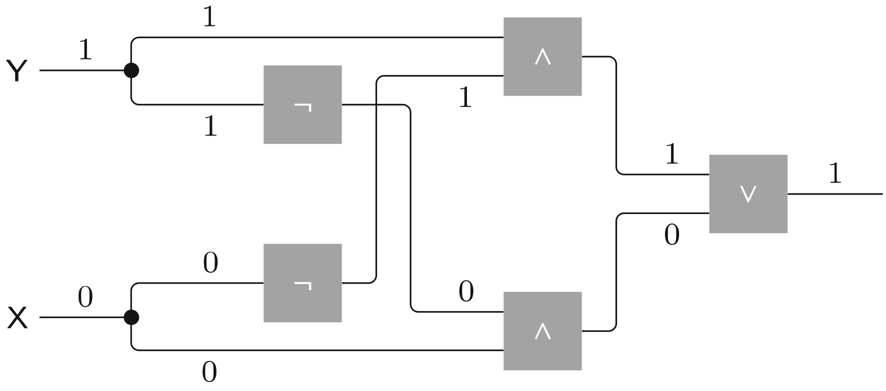
We won't use this style or these particular gate symbols further, but we will use different symbols to represent gates in quantum circuits, which we'll explain as we encounter them.

The particular circuit in this example computes the *exclusive-OR* (or XOR for short), which is denoted by the symbol $\oplus$:

ab	$a \oplus b$
00	0
01	1
10	1
11	0

X OR

In the next diagram we consider just one choice for the inputs: $\mathsf{X} = 0$ and $\mathsf{Y} = 1$. Each wire is labeled by value it carries so you can follow the operations. The output value is 1 in this case, which is the correct value for the XOR: $0 \oplus 1 = 1$.

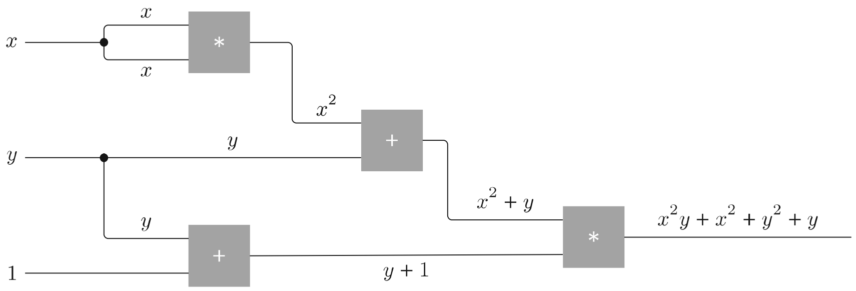


The other three possible input settings can be checked in a similar way.

Other types of circuits

As was suggested above, the notion of a circuit in computer science is very general. For example, circuits whose wires carry values other than 0 and 1 are sometimes analyzed, as are gates representing different choices of operations.

In *arithmetic circuits*, for instance, the wires may carry integer values while the gates represent arithmetic operations, such as addition and multiplication. The following figure depicts an arithmetic circuit that takes two variable input values (x and y) as well as a third input set to the value 1. The values carried by the wires, as functions of the values x and y, are shown in the figure.

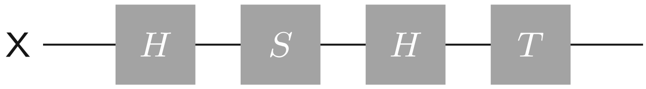


We can also consider circuits that incorporate randomness, such as ones where gates represent probabilistic operations.

Quantum circuits

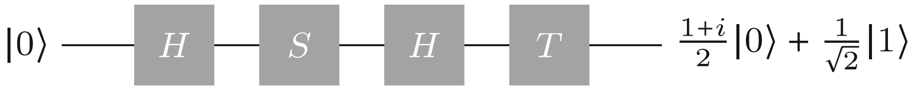
In the quantum circuit model, wires represent qubits and gates represent operations on these qubits. We'll focus for now on operations we've encountered so far, namely *unitary operations* and *standard basis measurements*. As we learn about other sorts of quantum operations and measurements, we can enhance our model accordingly.

Here's a simple example of a quantum circuit:

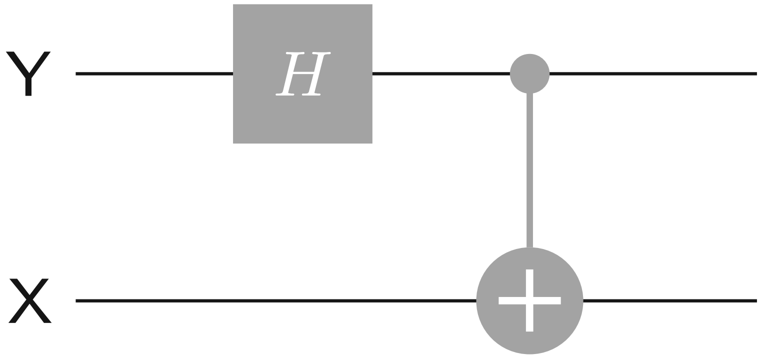


In this circuit, we have a single qubit named X, which is represented by the horizontal line, and a sequence of gates representing unitary operations on this qubit. Just like in the examples above, the flow of information goes from left to right — so the first operation performed is a Hadamard operation, the second is an S operation, the third is another Hadamard operation, and the final operation is a T operation. Applying the entire circuit therefore applies the composition of these operations, $THSH$, to the qubit X.

Sometimes we may wish to explicitly indicate the input or output states of circuits. For example, if we apply the operation $THSH$ to the state $|0\rangle$, we obtain the state $\frac{1+i}{2}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. This can be indicated as follows:



Quantum circuits often start out with all qubits initialized to $|0\rangle$, as we have in this case, but there are also situations where the input qubits are initially set to different states. Here's another example of a quantum circuit, this time with two qubits:



As always, the gate labeled H refers to a Hadamard operation, while the second gate is a *controlled-NOT* operation: the solid circle represents the *control qubit* and the circle resembling the symbol $\oplus$ denotes the *target qubit.*

Before examining this circuit in greater detail and explaining what it does, it is imperative that we first clarify how qubits are ordered in quantum circuits. This connects with the convention that Qiskit uses for naming and ordering systems that was mentioned briefly in the previous lesson.

Qiskit's qubit ordering convention for circuits

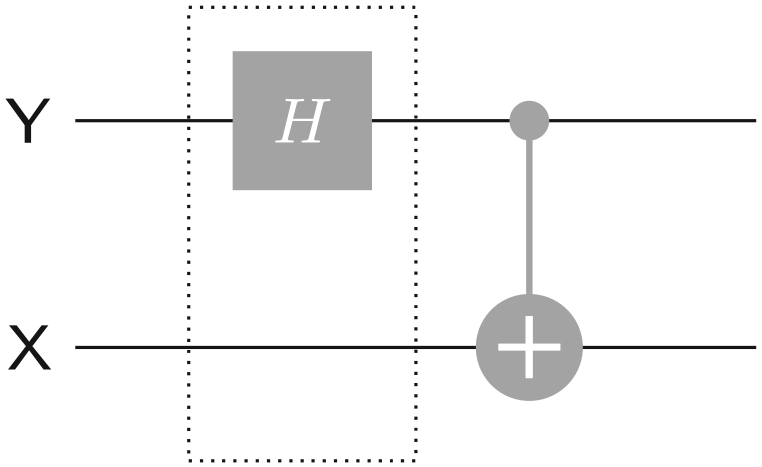
In Qiskit, the *topmost* qubit in a circuit diagram has index 0 and corresponds to the *rightmost* position in a tuple of qubits (or in a string, Cartesian product, or tensor product corresponding to this tuple), the *second-from-top* qubit has index 1, and corresponds to the position *second-from-right* in a tuple, and so on. The *bottommost* qubit, which has the highest index, therefore corresponds to the *leftmost* position in a tuple. In particular, Qiskit's default names for the qubits in an n-qubit circuit are represented by the n-tuple $(\mathsf{q}_{n-1},\ldots,\mathsf{q}_0)$, with q_0 being the qubit on the top and q_{n-1} on the bottom in quantum circuit diagrams.

Note that this is a reversal of a more common convention for ordering qubits in circuits, and is a frequent source of confusion. Further information on this ordering convention can be found on the Bit-ordering in Qiskit documentation page.

Although we sometimes deviate from the specific default names $\mathsf{q}_0,\ldots,\mathsf{q}_{n-1}$ used for qubits by Qiskit, we will always follow the ordering convention described above when interpreting circuit diagrams throughout this course. Thus, our interpretation of the circuit above is that it describes an operation on a pair of qubits (X,Y). If the input to the circuit is a quantum state $|\psi\rangle \otimes |\phi\rangle$, for instance, then this means that the lower qubit X starts in the state $|\psi\rangle$ and the upper qubit Y starts in the state $|\phi\rangle$.

To understand what the circuit does, we can go from left to right through its operations.

1. The first operation is a Hadamard operation on Y:

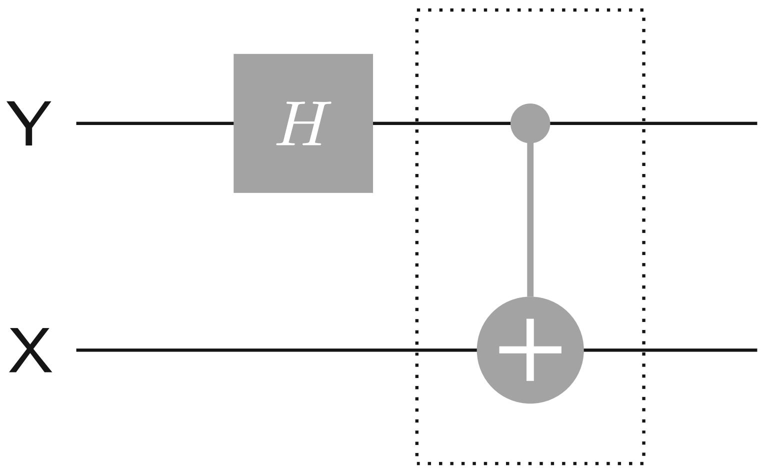


When applying a gate to a single qubit like this, nothing happens to the other qubits (which is just one other qubit in this case). Nothing happening is equivalent to the identity operation being performed. The dotted rectangle in the figure above therefore represents this operation:

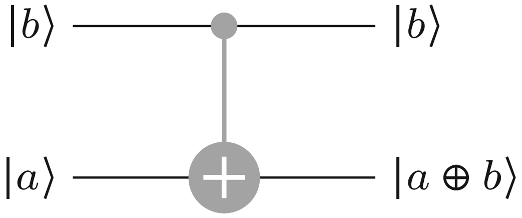
$$\mathbb{I} \otimes H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

Note that the identity matrix is on the left of the tensor product and H is on the right, which is consistent with Qiskit's ordering convention.

2. The second operation is the controlled-NOT operation, where Y is the control and X is the target:



The controlled-NOT gate's action on standard basis states is as follows:



Given that we order the qubits as (X, Y), with X being on the bottom and Y being on the top of our circuit, the matrix representation of the controlled-NOT gate is this:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The unitary operation implemented by the entire circuit, which we'll give the name U, is the composition of the operations:

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} = \left(\begin{matrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{matrix} \right.$$

In particular, recalling our notation for the Bell states,

$$\begin{aligned}
|\phi^+\rangle &= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\\
|\phi^-\rangle &= \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle\\
|\psi^+\rangle &= \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle\\
|\psi^-\rangle &= \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle,
\end{aligned}$$

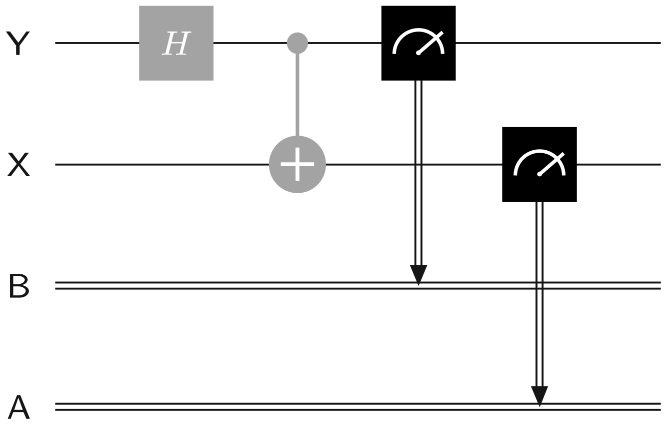
we find that

$$\begin{aligned}
U|00\rangle &= |\phi^+\rangle\\
U|01\rangle &= |\phi^-\rangle\\
U|10\rangle &= |\psi^+\rangle\\
U|11\rangle &= -|\psi^-\rangle.
\end{aligned}$$

This circuit therefore gives us a way to create the state $|\phi^+\rangle$ if we run it on two qubits initialized to $|00\rangle$. More generally, it provides us with a way to convert the standard basis to the Bell basis. (Note that, while it is not important for this example, the -1 phase factor on the last state, $-|\psi^-\rangle$, could be eliminated if we wanted by making a small addition to the circuit. For instance, we could add a controlled-Z gate at the beginning, which is similar to a controlled-NOT gate except that a Z operation is applied to the target qubit rather than a NOT operation when the control is set to 1. Alternatively, we could add a swap gate at the end. Either choice eliminates the minus sign without affecting the circuit's action on the other three standard basis states.)

try

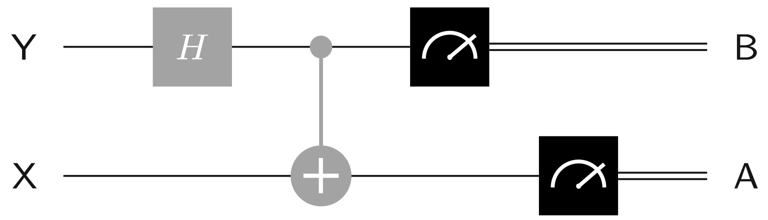
In general, quantum circuits can contain any number of qubit wires. We may also include *classical bit* wires, which are indicated by double lines, like in this example:



Here we have a Hadamard gate and a controlled-NOT gate on two qubits X and Y, just like in the previous example. We also have two *classical* bits, A and B, as well as two measurement gates. The measurement gates represent standard basis measurements: the qubits are changed into their post-measurement states, while the measurement outcomes are *overwritten* onto the classical bits to which the arrows point.

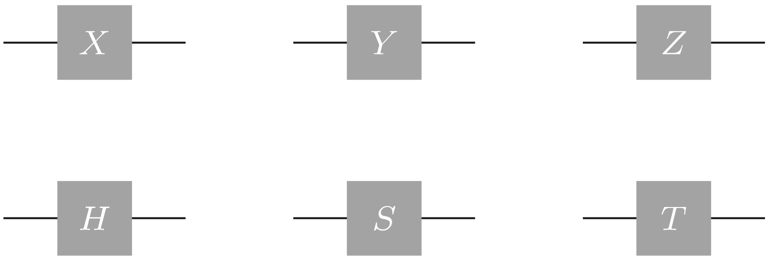
It's often convenient to depict a measurement as a gate that takes a qubit as input and outputs a classical bit (as opposed to outputting the qubit in its post-measurement state and writing the result to a separate classical bit). This means the measured qubit has been discarded and can safely be ignored thereafter, its state having changed into $\vert 0\rangle$ or $\vert 1\rangle$ depending upon the measurement outcome.

For example, the following circuit diagram represents the same process as the one in the previous diagram, but where we disregard X and Y after measuring them:

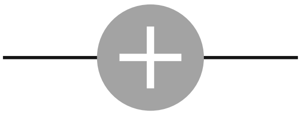


As the course continues, we'll see more examples of quantum circuits, which are usually more complicated than the simple examples above. Here are some examples of symbols used to denote gates that commonly appear in circuit diagrams:

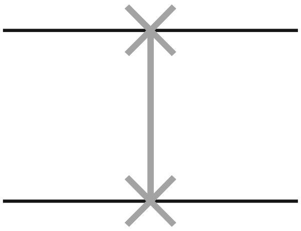
- Single-qubit gates are generally shown as squares with a letter indicating which operation it is, like this:



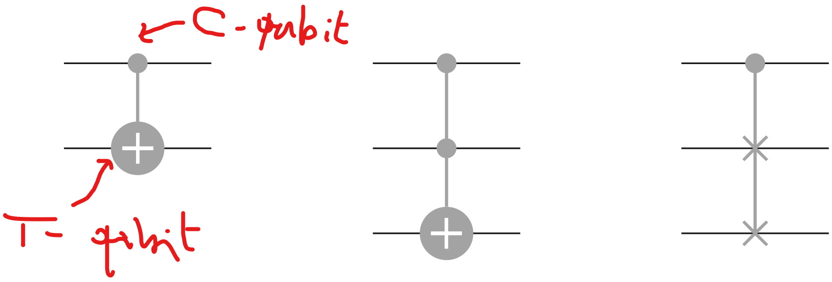
Not gates (or equivalently, X gates) are also sometimes denoted by a circle around a plus sign:



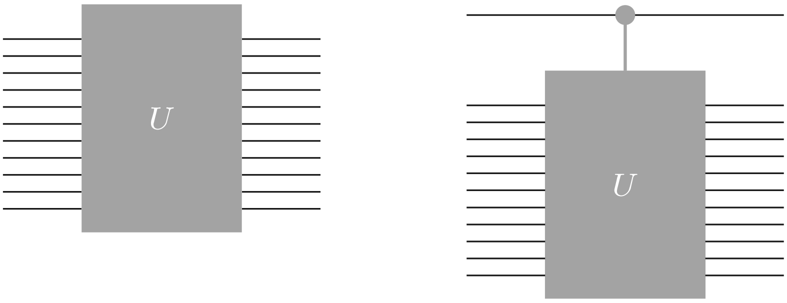
- Swap gates are denoted as follows:



- Controlled-gates, meaning gates that describe controlled-unitary operations, are denoted by a filled-in circle (indicating the control) connected by a vertical line to whatever operation is being controlled. For instance, controlled-NOT gates, controlled-controlled-NOT (or Toffoli) gates, and controlled-swap (Fredkin) gates are denoted like this:



- Arbitrary unitary operations on multiple qubits may be viewed as gates. They are depicted by rectangles labeled by the name of the unitary operation. For instance, here is a depiction of an (unspecified) unitary operation U as a gate, along with a controlled version of this gate:



Report a bug or request content on GitHub ↗.

Previous page

Next page