

# Multiple Qubits

📅 12/11/2017 ⌚ 11 minutes to read Contributors 👤 👤 👤 👤 🛠 all

## In this article

[Representing Two Qubits](#)

[Measuring Two-Qubit States](#)

[Two-Qubit Operations](#)

[Many-Qubit Systems](#)

While single-qubit gates possess some counter-intuitive features, such as the ability to be in more than one state at a given time, if all we had in a quantum computer were single-qubit gates then we would have a device with computational power that would be dwarfed by even a calculator let alone a classical supercomputer. The true power of quantum computing only becomes visible as we increase the number of qubits. This power arises, in part, because the dimension of the vector space in which quantum state vectors grows exponentially with the number of qubits. This means that while a single qubit can be trivially modeled, simulating a fifty-qubit quantum computation would arguably push the limits of existing supercomputers. Increasing the size of the computation by only one additional qubit *doubles* the memory required to store the state and roughly *doubles* the computational time. This rapid doubling of computational power is why a quantum computer with a relatively small number of qubits can far surpass the most powerful supercomputers of today, tomorrow and beyond for some computational tasks.

Why do we have exponential growth for quantum state vectors? Our goal in this section is to review the rules used to build multi-qubit states out of single-qubit states as well as discuss the gate operations that we need to include in our gate set to form a universal many-qubit quantum computer. These tools are absolutely necessary to understand the gate sets that are commonly used in Q# code and also to gain intuition about why quantum effects such as entanglement or interference render quantum computing more powerful than classical computing.

## Representing Two Qubits

The main difference between one- and two-qubit states is that two-qubit states are four dimensional rather than two dimensional. This is because the computational basis for two-qubit states is formed by the tensor products of one-qubit states. For example, we have

$$\begin{aligned}
 00 &\equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & 01 &\equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \\
 10 &\equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, & 11 &\equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.
 \end{aligned}$$

It is easy to see that more generally the quantum state of  $n$  qubits is represented by a unit vector of dimension  $2^n$  using this construction. The vector

$$\begin{bmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{bmatrix}$$

represents a quantum state on two qubits if  $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$ . Just as with single qubits, the quantum state vector of multiple qubits holds all the information needed to describe the system's behavior.

If we are given two separate qubits, one in the state  $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$  and a second qubit in the state  $\begin{bmatrix} \gamma \\ \delta \end{bmatrix}$ , the corresponding two-qubit state is

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \otimes \begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \alpha \begin{bmatrix} \gamma \\ \delta \end{bmatrix} \\ \beta \begin{bmatrix} \gamma \\ \delta \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix},$$

where the operation  $\otimes$  is called the tensor product (or Kronecker product) of vectors. Note that while we can always take the tensor product of two single-qubit states to form a two-qubit state, not all two-qubit quantum states can be written as the tensor product of two single-qubit states. For example, there are no states  $\psi = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$  and  $\phi = \begin{bmatrix} \gamma \\ \delta \end{bmatrix}$  such that their tensor product is the state

$$\psi \otimes \phi = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{bmatrix}.$$

Such a two-qubit state, which cannot be written as the tensor product of single-qubit states, is called an "entangled state"; the two qubits are said to be [entangled](#). Loosely speaking, because the quantum state cannot be thought of as a tensor product of single qubit states, the information that the state holds is not confined to either of the qubits individually. Rather, the information is stored non-locally in the correlations between the two states. This non-locality of information is one of the major distinguishing features of quantum computing over classical computing and is essential for a number of quantum protocols including [quantum teleportation](#) and [quantum error correction](#).

## Measuring Two-Qubit States

Measuring two-qubit states is very similar to single-qubit measurements. Measuring the state

$$\begin{bmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{bmatrix}$$

yields 00 with probability  $|\alpha_{00}|^2$ , 01 with probability  $|\alpha_{01}|^2$ , 10 with probability  $|\alpha_{10}|^2$ , and 11 with probability  $|\alpha_{11}|^2$ . The variables  $\alpha_{00}$ ,  $\alpha_{01}$ ,  $\alpha_{10}$ , and  $\alpha_{11}$  were deliberately named to make this connection clear. After the measurement, if the outcome is 00 then the quantum state of the two-qubit system has collapsed and is now

$$00 \equiv \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

It is also possible to measure just one qubit of a two-qubit quantum state. In cases where you measure only one of the qubits, the impact of measurement is subtly different because the entire state is not collapsed to a computational basis state, rather it is collapsed to only one sub-system. In other words, in such cases measuring only one qubit only collapses one of the subsystems but not all of them.

To see this consider measuring the first qubit of the following state, which is formed by applying the Hadamard transform  $H$  on two qubits initially set to the "0" state:

$$H^{\otimes 2} \left( \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \mapsto \begin{cases} \text{outcome} = 0 & \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\ \text{outcome} = 1 & \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \end{cases}.$$

Both outcomes have 50% probability of occurring. The outcome being 50% probability for both can be intuited from the fact that the initial quantum state vector is invariant under swapping 0 with 1 on the first qubit.

The mathematical rule for measuring the first or second qubit is simple. If we let  $e_k$  be the  $k^{\text{th}}$  computational basis vector and let  $S$  be the set of all  $e_k$  such that the qubit in question takes the value 1 for that value of  $k$ . For example, if we are interested in measuring the first qubit then  $S$  would consist of  $e_2 \equiv 10$  and  $e_3 \equiv 11$ . Similarly, if we are interested in the second qubit  $S$  would consist of  $e_1 \equiv 01$  and  $e_3 \equiv 11$ . Then the probability of measuring the chosen qubit to be 1 is for state vector  $\psi$

$$P(\text{outcome} = 1) = \sum_{e_k \text{ in the set } S} \psi^\dagger e_k e_k^\dagger \psi.$$

Since each qubit measurement can only yield 0 or 1, the probability of measuring 0 is simply  $1 - P(\text{outcome} = 1)$ . This is why we only explicitly give a formula for the probability of measuring 1.

The action that such a measurement has on the state can be expressed mathematically as

$$\psi \mapsto \frac{\sum_{e_k \text{ in the set } S} e_k e_k^\dagger \psi}{\sqrt{P(\text{outcome} = 1)}}.$$

The cautious reader may worry about what happens when the probability of the measurement is zero. While the resultant state is technically undefined in this case, we never need to worry about such eventualities because the probability is zero!

If we take  $\psi$  to be the uniform state vector given above and are interested in measuring the first qubit then

$$P(\text{measurement of first qubit} = 1) = (\psi^\dagger e_2)(e_2^\dagger \psi) + (\psi^\dagger e_3)(e_3^\dagger \psi) = |e_2^\dagger \psi|^2 + |e_3^\dagger \psi|^2.$$

Note that this is just the sum of the two probabilities that would be expected for measuring the results 10 and 11 were all the qubits to be measured. For our example, this evaluates to

$$\frac{1}{4} \left| \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right|^2 + \frac{1}{4} \left| \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right|^2 = \frac{1}{2}.$$

which perfectly matches what our intuition tells us the probability should be. Similarly, the state can be written as

$$\frac{\frac{e_2}{2} + \frac{e_3}{2}}{\sqrt{\frac{1}{2}}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

again in accordance with our intuition.

## Two-Qubit Operations

As in the single-qubit case, any unitary transformation is a valid operation on qubits. In general, a unitary transformation on  $n$  qubits is a matrix  $U$  of size  $2^n \times 2^n$  (so that it acts on vectors of size  $2^n$ ), such that  $U^{-1} = U^\dagger$ . For example, the CNOT (controlled-NOT) gate is a commonly used two-qubit gate and is represented by the following unitary matrix:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

We can also form two-qubit gates by applying single-qubit gates on both qubits. For example, if we apply the gates

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

and

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

to the first and second qubits, respectively, this is equivalent to applying the two-qubit unitary given by their tensor product:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{bmatrix}.$$

Thus we can form two-qubit gates by taking the tensor product of some known single-qubit gates. Some examples of two-qubit gates include  $H \otimes H$ ,  $X \otimes \mathbf{1}$ , and  $X \otimes Z$ .

Note that while any two single-qubit gates define a two-qubit gate by taking their tensor product, the converse is not true. Not all two-qubit gates can be written as the tensor product of single-qubit gates. Such a gate is called an *entangling* gate. One example of an entangling gate is the CNOT gate.

The intuition behind a controlled-not gate can be generalized to arbitrary gates. A controlled gate in general is a gate that acts as identity (ie it has no action) unless a specific qubit is 1. We denote a controlled unitary, controlled in this case on the qubit labeled  $x$ , with a  $\Lambda_x(U)$ . As an example  $\Lambda_0(U)e_1 \otimes \psi = e_1 \otimes U\psi$  and  $\Lambda_0(U)e_0 \otimes \psi = e_0 \otimes \psi$ , where  $e_0$  and  $e_1$  are the computational basis vectors for a single qubit corresponding to the values 0 and 1. For example, consider the following controlled- $Z$  gate then we can express this as

$$\Lambda_0(Z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = (\mathbf{1} \otimes H) \text{CNOT}(\mathbf{1} \otimes H).$$

Building controlled unitaries in an efficient manner is a major challenge. The simplest way to implement this requires forming a database of controlled versions of fundamental gates and replacing every fundamental gate in the original unitary operation with its controlled counterpart. This is often quite wasteful and clever insight often can be used to just replace a few gates with controlled versions to achieve the same impact. For this reason, we provide in our framework the ability to perform either the naive method of controlling or allow the user to define a controlled version of the unitary if an optimized hand-tuned version is known.

Gates can also be controlled using classical information. A classically controlled not-gate, for example, is just an ordinary not-gate but it is only applied if a classical bit is 1 as opposed to a quantum bit. In this sense, a classically controlled gate can be thought of as an if statement in the quantum code wherein the gate is applied only in one branch of the code.

As in the single-qubit case, a two-qubit gate set is universal if any  $4 \times 4$  unitary matrix can be approximated by a product of gates from this set to arbitrary precision. One example of a universal gate set is the Hadamard gate, the T gate, and the CNOT gate. By taking products of these gates, we can approximate any unitary matrix on two qubits.

# Many-Qubit Systems

We follow the exact same patterns explored in the two-qubit case to build many-qubit quantum states from smaller systems. Such states are built by forming tensor products of smaller states. For example, consider encoding the bit string 1011001 in a quantum computer. We can encode this as

$$1011001 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Quantum gates work in exactly the same way. For example, if we wish to apply the  $X$  gate to the first qubit and then perform a CNOT between the second and third qubits we may express this transformation as

$$(X \otimes \text{CNOT}_{12} \otimes \mathbf{1} \otimes \mathbf{1} \otimes \mathbf{1}) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \equiv 0011001.$$

In many qubit systems, there is often a need to allocate and de-allocate qubits that serve as temporary memory for the quantum computer. Such qubit a qubit is called an ancilla. By default we assume the qubit state is initialized to  $e_0$  upon allocation. We further assume that it is returned again to  $e_0$  before de-allocation. This assumption is important because if an ancilla qubit becomes entangled with another qubit register when it becomes de-allocated then the process of de-allocation will damage the ancilla. For this reason, we always assume that such qubits are reverted to their initial state before being released.

Finally, although new gates needed to be added to our gate set to achieve universal quantum computing for two qubit quantum computers, no new gates need to be introduced in the multi-qubit case. The gates  $H$ ,  $T$  and CNOT form a universal gate set on many qubits because any general unitary transformation can be broken into a series of two qubit rotations. We then can leverage the theory developed for the two-qubit case and use it again here when we have many qubits.

While the linear algebraic notation that we have been using thus far can certainly be used to describe multi-qubit states, it becomes increasingly cumbersome as we increase the size of the states. The resulting column-vector for a length 7 bit string, for example, is 128 dimensional, which makes expressing it using the notation described previously very cumbersome. For this reason, we next present a common notation in quantum computing that allows us to concisely describe these high-dimensional vectors.

## Note

The feedback system for this content will be changing soon. Old comments will not be carried over.

If content within a comment thread is important to you, please save a copy. For more information on

the upcoming change, [we invite you to read our blog post](#).