Quantum Programming Foundations: Quantum Circuits

Jens Palsberg

Apr 10, 2019

Outline

Hook: The machine model of quantum computing is a quantum circuit. A quantum circuit can be drawn as a diagram, which shows which qubits we work with, which matrices we apply, and when. All this is just as simple as it sounds, but the things we can do with quantum circuits are amazing.

Purpose: Persuade you that quantum circuits are both simple and expressive.

Preview:

1. Diagram notation for quantum circuits.
2. Superdense coding, teleportation, and no-cloning.
3. Universal sets of operations.

Transition to Body: Let us first look at how to draw a quantum circuit.

Main Point 1: Diagram notation for quantum circuits.
[Input qubits]
[Unitary matrices]
[Measurement]

Transition to MP2: Now let us look at what we can and cannot do with quantum circuits.

Main Point 2: Superdense coding, teleportation, and no-cloning.
[Superdense coding]
[Teleportation]
[No-cloning]

Transition to MP3: What is the machine language of quantum circuits?

Main Point 3: Universal sets of operations.
[Universal sets exists]
[Universal sets can be quite different]
[Different quantum computers support different sets of matrices]

Transition to Close: So there you have it.

Review: We can draw circuit diagrams and do amazing little tasks, and we can talk about universal sets of operations.

Strong finish: Now we are ready to study quantum algorithms, which we will express as quantum circuits.

Call to action: Play around with the quantum circuit simulator that we linked online.

Detailed presentation

Hook: The machine model of quantum computing is a quantum circuit. A quantum circuit can be drawn as a diagram, which shows which qubits we work with, which matrices we apply, and when. All this is just as simple as it sounds, but the things we can do with quantum circuits are amazing.

Purpose: Persuade you that quantum circuits are both simple and expressive.

Preview:

1. Diagram notation for quantum circuits.
2. Superdense coding, teleportation, and no-cloning.
3. Universal sets of operations.

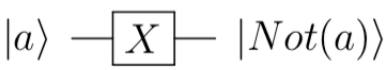
Transition to Body: Let us first look at how to draw a quantum circuit.

Main Point 1: Diagram notation for quantum circuits.

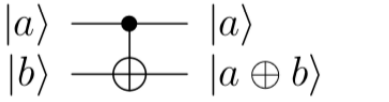
[Input qubits]
The input qubits appear on the left. Each qubit has its own line that shows what happens to it as time moves from left to right.

[Unitary matrices]
Since quantum gates needs to be reversible, quantum gates must have the same number of inputs and outputs, unlike classical gates. Examples:

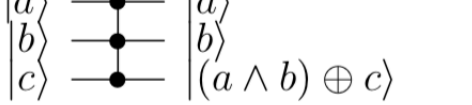
- NOT(X) gate



- CNOT(X) gate

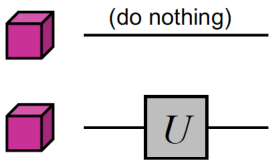


- Toffoli (CCNOT) gate



Note that the CNOT gate has the control bit on top (the filled circule) and the target bit on the bottom (the empty circle).

Let us apply a one-qubit matrix to two qubits.



$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix}$$

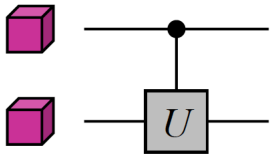
Maps basis states as:

$|0\rangle|0\rangle \rightarrow |0\rangle U|0\rangle$
$|0\rangle|1\rangle \rightarrow |0\rangle U|1\rangle$
$|1\rangle|0\rangle \rightarrow |1\rangle U|0\rangle$
$|1\rangle|1\rangle \rightarrow |1\rangle U|1\rangle$

The resulting 4x4 matrix is

$$I \otimes U = \begin{bmatrix} u_{00} & u_{01} & 0 & 0 \\ u_{10} & u_{11} & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

Now let us use one qubit to control the application of a matrix U to another qubit.



$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix}$$

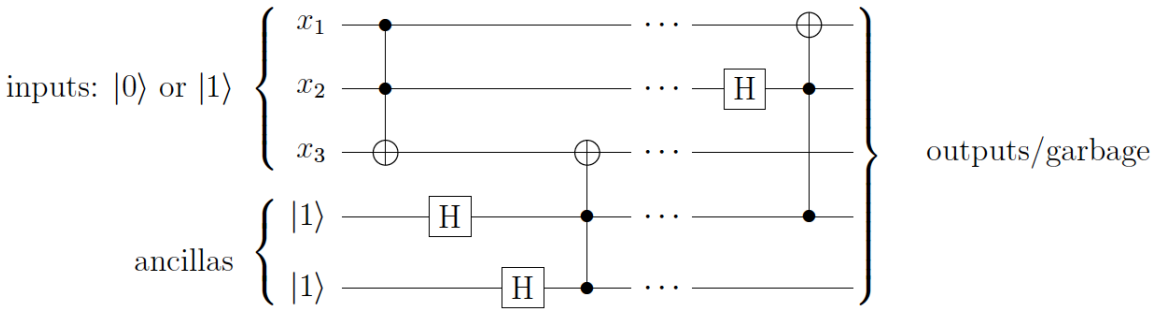
Maps basis states as:

$|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
$|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
$|1\rangle|0\rangle \rightarrow |1\rangle U|0\rangle$
$|1\rangle|1\rangle \rightarrow |1\rangle U|1\rangle$

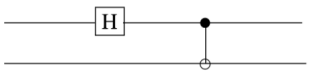
Resulting 4x4 matrix is controlled-U =

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

General idea of a circuit:



We can map $|0\rangle \otimes |0\rangle$ to the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ with the following circuit that uses an Hadamard and a *CNOT*.



Let us check the math. The first application of H to the first qubit of $|0\rangle \otimes |0\rangle$ gives us

$$|+\rangle \otimes |0\rangle \quad = \quad \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$$

and now we can apply $CNOT$ and get

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

The four Bell states are:

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\Phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\Psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned}$$

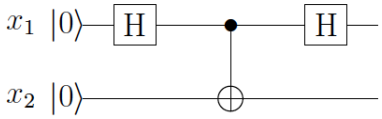
Exercise: show how to generate the other three Bell states by circuits.

Exercise: show that the four Bell states form an orthonormal basis.

For example, we could devise a quantum circuit that carries out Shor's algorithm: it takes as input an n-bit integer, uses roughly n^2 CCNOT and H gates, and has the property that when you measure the final output qubits, they give (with probability at least 99%) the binary encoding of the prime factorization of the input integer (plus some garbage bits).

[Measurement]

Example:



The initial state is $|00\rangle$.

After the first Hadamard gate, the state is

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$$

After the $CNOT$, the state is

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

After the final Hadamard gate, the state is:

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

So we get each of the four outcomes with probabilit $\frac{1}{4}$.

Transition to MP2: Now let us look at what we can and cannot do with quantum circuits.

Main Point 2: Superdense coding, teleportation, and no-cloning.

[Superdense coding]

Suppose we have n qubits. How many classical bits is that? Holevo's theorem (1973) gives the answer: n bits.

What are the consequences of Holevo's theorem? Example: suppose Alice is trying to encode two bits as a single qubit. Holevo's theorem says that this is impossible: if you have a single qubit, it boils down to a single bit.

However, Alice still wants to send two bits ab to Bob, by sending qubits. Does Alice have to send two qubits? Here is a different idea, called superdense coding, in which Alice sends a single qubit to Bob. The idea is that first Bob sends a qubit to Alice; perhaps way ahead of time.

1. Bob creates the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and sends the first qubit A of that state to Alice but keeps the second qubit B.
2. Alice does the following (she creates one of the four states in the Bell basis):
   (a) If $a = 1$, then she applies Z to the qubit A.
   (b) If $b = 1$, then she applies X to the possibly transformed qubit A.
   (c) She sends the possibly transformed qubit A to Bob.
3. Bob does the following (he measures the qubits in the Bell basis):
   (a) He applies $CNOT(A, B)$.
   (b) He applies H to A.
   (c) He measures both A and B.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

ab	After 2a	After 2b	After 3a	After 3b	Bob sees
00	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle + \|10\rangle) = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)\|0\rangle$	$\|00\rangle$	00
01	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(\|11\rangle + \|01\rangle) = \frac{1}{\sqrt{2}}(\|1\rangle + \|0\rangle)\|1\rangle$	$\|01\rangle$	01
10	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle - \|10\rangle) = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)\|0\rangle$	$\|10\rangle$	10
11	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|10\rangle - \|01\rangle)$	$\frac{1}{\sqrt{2}}(\|11\rangle - \|01\rangle) = \frac{1}{\sqrt{2}}(\|1\rangle - \|0\rangle)\|1\rangle$	$-\|11\rangle$	11

[Teleportation]

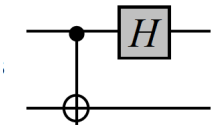
The task: Alice has a qubit $\alpha|0\rangle + \beta|1\rangle$ and wants to tell Bob what it is; she wants to do that by sending classical bits to Bob.

Like in superdense coding, Bob creates the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and sends the first qubit of that state to Alice.

Let us look at the state of all three qubits, of which the first two belong to Alice and the third belong to Bob:

$$(\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Now Alice measures her two qubits in the Bell basis and sends the result to Bob. Specifically, Alice applies the following to her two qubits:



after which Alice measures her two qubits and gets one of 00, 01, 10, 11, with probability $\frac{1}{4}$, and then sends those two bits to Bob.

Let us go through the steps of how this works. Let us rewrite the state of all three qubits:

$$(\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad = \quad \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$$

Now, after Alice has applied $CNOT$, the state of all three qubits is:

$$\frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle)$$

Then, after Alice has applied H to the first qubit, the state of all three qubits is:

$$\begin{aligned} & \frac{1}{2}(\alpha|000\rangle + \alpha|100\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|110\rangle + \beta|001\rangle - \beta|101\rangle) \\ = \quad & \frac{1}{2}(|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle) \end{aligned}$$

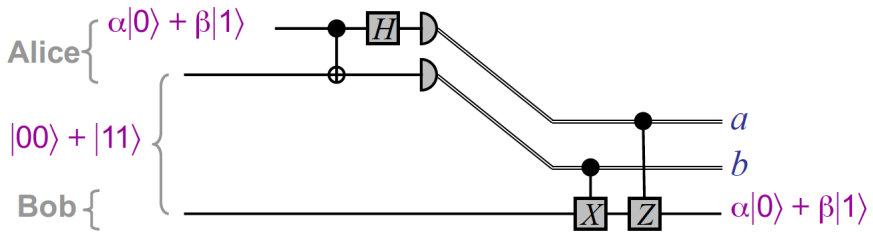
Now Alice measures her two qubits. What is the chance that she sees 00? We can see above that it is the square of $\frac{1}{2}$, which is $\frac{1}{4}$. Similar reasoning applies to the other three cases.

Bob receives the two classical bits ab from Alice, and then Bob does the following:

- If $b = 1$, then he applies X to his qubit.
- If $a = 1$, then he applies Z to his qubit.

The result is:

$$\begin{cases} 00 & : \quad \alpha|0\rangle + \beta|1\rangle \\ 01 & : \quad X(\alpha|1\rangle + \beta|0\rangle) \;=\; \alpha|0\rangle + \beta|1\rangle \\ 10 & : \quad Z(\alpha|0\rangle - \beta|1\rangle) \;=\; \alpha|0\rangle + \beta|1\rangle \\ 11 & : \quad ZX(\alpha|1\rangle - \beta|0\rangle) \;=\; \alpha|0\rangle + \beta|1\rangle \end{cases}$$



Note that in the process of teleporting a qubit from Alice to Bob, we destroyed Alice's qubit. This is the way it is supposed to be: we cannot clone an unknown qubit, as we will discuss next.

[No-cloning]
We can copy a classical bit; what about a qubit?

Theorem: No quantum operation maps $|\psi\rangle|0\rangle$ to $|\psi\rangle|\psi\rangle$.

Proof: Suppose we have such an operation U. Let us apply U to $|\psi_1\rangle$ and $|\psi_2\rangle$, for which $\langle\psi_1|\psi_2\rangle \neq 0$ and $\langle\psi_1|\psi_2\rangle \neq 1$. Now we use that U preserves inner products:

$$\begin{aligned}\langle\psi_1|\psi_2\rangle &= \langle(\psi_1 \otimes |0\rangle)|(\psi_2 \otimes |0\rangle)\rangle \\ &= \langle U(\psi_1 \otimes |0\rangle)|U(\psi_2 \otimes |0\rangle)\rangle \\ &= \langle(\psi_1 \otimes |\psi_1\rangle)|(\psi_2 \otimes |\psi_2\rangle)\rangle \\ &= \langle\psi_1|\psi_2\rangle^2\end{aligned}$$

which is a contradition. QED.

Transition to MP3: What is the machine language of quantum circuits?

Main Point 3: Universal sets of operations.

[Universal sets exists]
S is an exact universal set for a model if any gate from that model can be realized exactly using only combinations of gates from S.

For classical computing, $\{NAND\}$ is a universal set. Let us review why. We will get there in three steps. Our starting point is that we know that $\{AND, OR, NOT\}$ is universal. First, we implement OR in terms of AND and NOT gates using De Morgan's rule:

$$OR(x_1, x_2) = NOT(AND(NOT(x_1), NOT(x_2)))$$

We can also use standard notation:

$$x_1 \vee x_2 = \neg((\neg x_1) \wedge (\neg x_2))$$

Second, we implement AND in terms of $NAND$ gates:

$$AND(x_1, x_2) = NOT(NAND(x_1, x_2)$$

Third, we implement NOT in terms of $NAND$ gates:

$$NOT(x_1) = NAND(x_1, 1)$$

Notice that we used a helper bit, initialized to 1. So, the statement that, for classical computing, $\{NAND\}$ is a universal set means that $NAND$ gates plus helper bits is a universal set. We know that helper bits are important in reversible computing and we will use them again and again.

For reversible computing, we observe that $NAND$ is not reversible, so, we have to think harder. Fortunately, NOT is reversible: it is a bijection on $\{0, 1\}$, as the following table shows:

input	output
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$

Indeed, $NOT(NOT\ |x\rangle\ ) = |x\rangle$.

Now let us look at $CNOT$, also known as controlled-NOT:

$$CNOT(x_1, x_2) \quad = \quad (x_1, x_1 \oplus x_2)$$

Or, in table form:

input	output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

This mapping is a bijection so $CNOT$ is good for reversible computing. Indeed,

$$CNOT(CNOT\ |xy\rangle\ ) \quad = \quad |xy\rangle$$

Now let us take the $CNOT$ idea a step further and look at $CCNOT$, also known as controlled-controlled-NOT, or the Toffoli gate:

$$CCNOT(x_1, x_2, x_3) \quad = \quad (x_1, x_2, AND(x_1, x_2) \oplus x_3)$$

Or, in table form:

input	output
$\|000\rangle$	$\|000\rangle$
$\|001\rangle$	$\|001\rangle$
$\|010\rangle$	$\|010\rangle$
$\|011\rangle$	$\|011\rangle$
$\|100\rangle$	$\|100\rangle$
$\|101\rangle$	$\|101\rangle$
$\|110\rangle$	$\|111\rangle$
$\|111\rangle$	$\|110\rangle$

This mapping is a bijection so $CCNOT$ is good for reversible computing. Indeed,

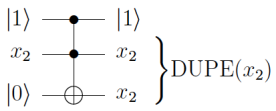
$$CCNOT(CCNOT\ |xyz\rangle\ ) \quad = \quad |xyz\rangle$$

Now for crux of this development about reversible computing: we can implement $NAND$ using $CCNOT$:

$$NAND(x_1, x_2) \quad = \quad CCNOT(x_1, x_2, 1)$$

So, for reversible computing, $CCNOT$ gates plus helper bits is a universal set.

Let us also note that we can duplicate bits using $CCNOT$:



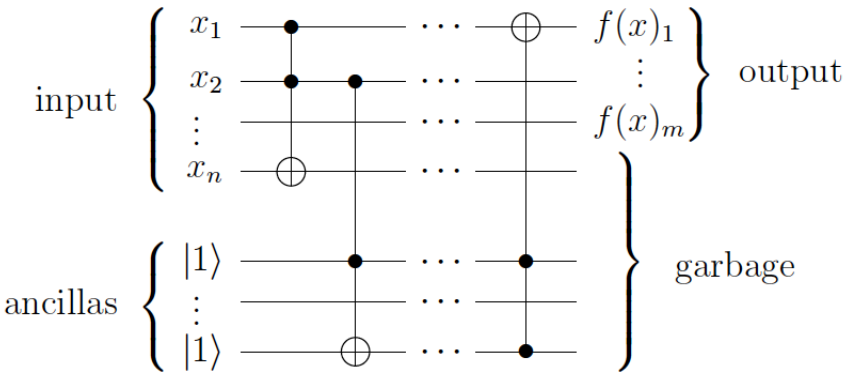
Here we see circuit notation for $CCNOT$: the two black dots are for the control bits, while the open circle is for the target bit.

Additionally, let us note that we can get $|0\rangle$ from $|1\rangle$ by using $CCNOT$:

$$CCNOT(1, 1, 1) \quad = \quad (1, 1, 0)$$

So, helper bits can always be 1.

Now we can picture the general case of reversible circuit that implements a function f from n bits to m bits.



For probabilistic computing, $\{NAND, CoinFlip_p\}$ is a universal set. Here, $CoinFlip_p$ is a coin flip operation on a single bit operating with bias p. In practice, $\{NAND, CoinFlip_{\frac{1}{2}}\}$ is a universal set; the fair coin can simulate any biased coin with arbitrary precision.

For quantum computing, $\{H, CCNOT\}$ is a universal set. This set can express all real unitary matrices. Once we add *Scale*, we can also express all complex unitary matrices.

$$Scale = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

[Universal sets can be quite different]
For quantum computing, $\{CNOT, H, R_{\frac{\pi}{4}}\}$ is a universal set. Here,

$$R_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

Another name: $T = R_{\frac{\pi}{4}}$. Note that $Z = R_\pi$ because $e^{i\pi} = -1$.

Here is another example of a universal set: $\{CNOT,\ ROT_{\frac{\pi}{8}},\ Scale\}$. Here,

$$ROT_{\frac{\pi}{8}} = \begin{pmatrix} \cos\frac{\pi}{8} & \sin\frac{\pi}{8} \\ -\sin\frac{\pi}{8} & \cos\frac{\pi}{8} \end{pmatrix}$$

Here, $CNOT$ and $ROT_{\frac{\pi}{8}}$ are sufficient to express all real unitary matrices.

[Different quantum computers support different sets of matrices]
The people designing quantum computers tend to build in support for a finite set of matrices. They may pick a specific set based on what is easiest to get to work with the kind of qubits that they have. This creates a mapping problem for anyone who wants to run an algorithm that uses one set of matrices on a computer that supports a different set of matrices.

Transition to Close: So there you have it.

Review: We can draw circuit diagrams and do amazing little tasks, and we can talk about universal sets of operations.

Strong finish: Now we are ready to study quantum algorithms, which we will express as quantum circuits.

Call to action: Play around with the quantum circuit simulator that we linked online.