CSE439 Week 3: Qubits and Quantum Circuits (chapter 4 *plus* sections 5.1--5.3)

[The flow of Chapter 4 as written is to take the classical notion of computations by machines as given. When CSE396 was a required course at UB, everyone saw *Turing machines* (TMs); those may have been talked about briefly in CSE331, but otherwise the "random-access machine" concept of executing algorithms from that course is fine. (The one advantage of TMs is that you can say that their tape cells numbered 1,2,3,... represent "classical bits" that evolve over time, in analogy to the way we will speak of *qubits* evolving over time.) Now, however, we will take the *classical Boolean circuit* model as fundamental while contrasting it directly to quantum circuits. The strongest linkage is that the quantum **Toffoli gate** can simulate NAND and hence do all classical Boolean operations by itself. This is shown in section 5.3, though. Section 5.1 has the n-fold tensor product $\mathbf{H}^{\otimes n}$ of the basic 2×2 Hadamard matrix $\mathbf{H}$, which we have already seen, anyway. So please read all the above as one block.]

Unitary Versus Stochastic (section 3.6)

A (doubly) **stochastic** matrix has the property that its rows (and columns) are nonnegative real numbers that sum to 1. A simple example is

$$J = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

However, while J is Hermitian (like any symmetric real matrix), it is not unitary: $JJ^* = J^2 = J$, not the identity. There are doubly stochastic matrices that are not Hermitian either when we go up to 3×3, e.g.:

$$\begin{bmatrix} 1/2 & 1/3 & 1/6 \\ 1/2 & 1/6 & 1/3 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

However, every **permutation matrix** is both doubly stochastic (in the trivial manner of having a single 1 in each row and column) and unitary. A less trivial example of symmetric (Hermitian) doubly stochastic matrices arise from **undirected graphs** G that are **regular**---meaning every vertex in G has the same **degree** (meaning: number of edges connecting to it). The text in section 3.6 gives an example where negating some of the entries does create a unitary matrix. However, this is not a regular phenomenon as far as I know.

Operations: Joint and Entangled

Here is a statement that uses a lot of notational fuss to express the simplest of ideas:

Proposition: For any $m \times n$ matrix A, $p \times q$ matrix B, n-vector $\mathbf{x}$ and q-vector $\mathbf{y}$,

$$(A\mathbf{x}) \otimes (B\mathbf{y}) = (A \otimes B) \cdot (\mathbf{x} \otimes \mathbf{y}).$$

Proof. The dimensions are consistent: both sides give a column vector of mp entries. Showing equality is where our effort to interpret vectors $\mathbf{x}$ as functions $\mathbf{x}(u)$ of their indices in binary notation may help. Under this view, $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$ gives the function $\mathbf{z}(uv) = \mathbf{x}(u)\mathbf{y}(v)$, where uv means concatenation of binary strings, while the right-hand side is an ordinary numeric product. And a matrix A gives the two-argument function $A(u, w) = a_{u,w}$.

$[0.5, 0.5, -0.5, 0.5] \otimes [0.6, 0.8] = [0.3, 0.4, 0.3, 0.4, -0.3, -0.4, 0.3, 0.4]$
Indices: $[000, 001, 010, 011, 100, 101, 110, 111]$ $100 = 10 \cdot 0$: $-0.3 = (-0.5)(0.6)$.

Silly? style note: When we think of vector and matrix entries the way we usually do, we will use square brackets like in the text, e.g.: $\mathbf{x}[i]$, $\mathbf{A}[i, j]$. When the indices are regarded as binary strings rather than numbers, we will write things like $\mathbf{A}[u, w]$ and $\mathbf{C}[uv, wt]$ below, where $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$.

The vector $\mathbf{x}' = A\mathbf{x}$ becomes the function mapping a row-index u to $\mathbf{x}'(u) = \sum_w A(u, w)\mathbf{x}(w)$. Thus, putting $\mathbf{z}' = (A\mathbf{x}) \otimes (B\mathbf{y})$, the right-hand side is the function

$$\mathbf{z}'(uv) = \mathbf{x}'(u)\mathbf{y}'(v) = \left(\sum_w A(u, w)\mathbf{x}(w)\right) \cdot \left(\sum_t B(v, t)\mathbf{y}(t)\right)$$

Now by usual rules of re-ordering summations, the right-hand side of this can be rearranged as

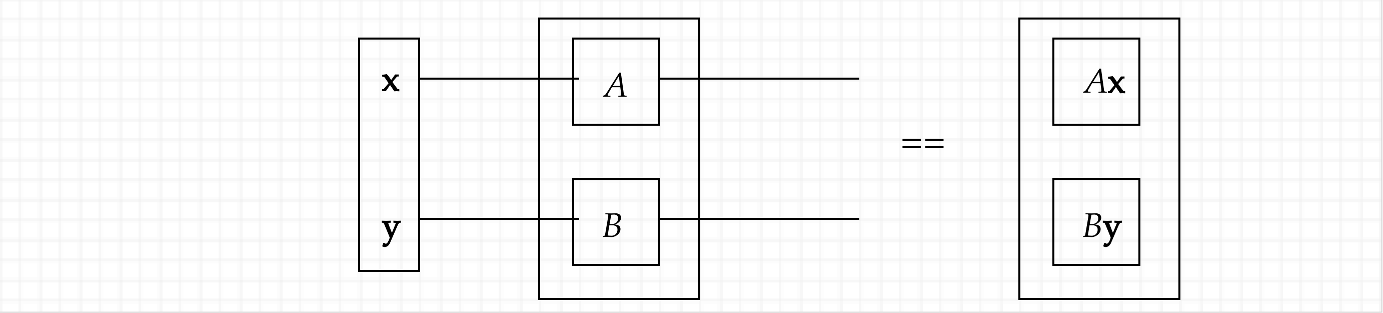
$$\sum_w \sum_t A(u, w)B(v, t)\mathbf{x}(w)\mathbf{y}(t)$$

With $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$, we can already recognize that the $\mathbf{x}(w)\mathbf{y}(t)$ part is the same as $\mathbf{z}(wt)$. And $A(u, w)B(v, t)$ is the same as $(A \otimes B)(uv, wt)$. So the whole thing becomes

$$\sum_{w,t} (A \otimes B)(uv, wt) \cdot (\mathbf{x} \otimes \mathbf{y})(wt),$$

which is exactly the meaning of $(A \otimes B) \cdot (\mathbf{x} \otimes \mathbf{y})$. So the two sides are equal. ☒

The simple idea is that $(A \otimes B) \cdot (\mathbf{x} \otimes \mathbf{y})$ does the A operation on x side-by-side with B doing its operation on y, but with no connection at all between them. We will soon have diagrams like this---



---note that we picture the inputs coming in from the left but when writing them as matrix arguments they will swing around to the right. As a tandem, this is formally the tensor product $\mathbf{x} \otimes \mathbf{y}$ coming in to $(A \otimes B)$. But really---and **locally**---it is just $A\mathbf{x}$ happening in one place and $B\mathbf{y}$ happening independently in another place. The upshot is this:

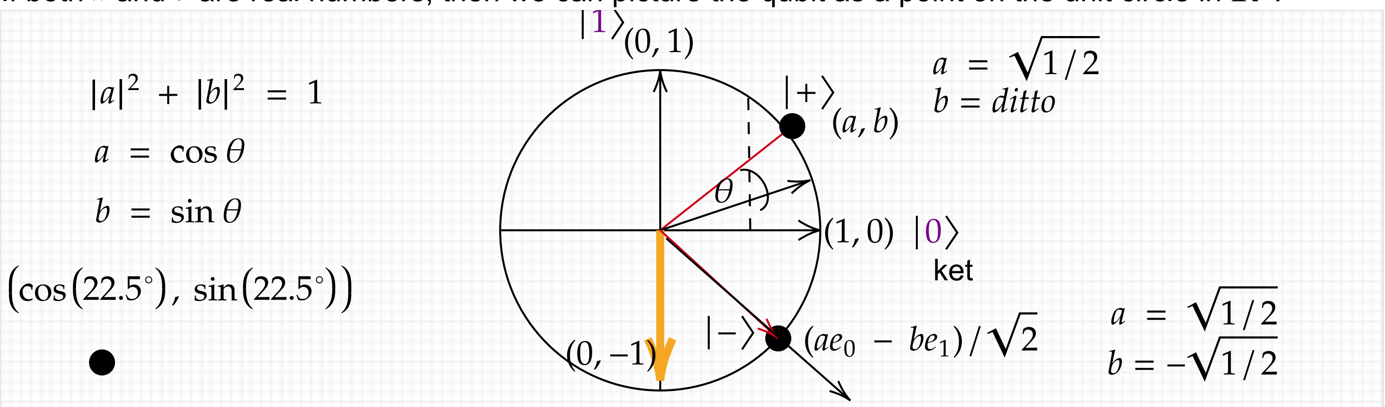
> **When we have entanglement, not independence, between the $\mathbf{x}$ part and the $\mathbf{y}$ part, then the notation will stay the same but the interpretation will change a whole lot.**

Qubits

A **qubit** is a physical system whose **state** ϕ is described by a pair (a, b) of complex numbers such that $|a|^2 + |b|^2 = 1$. (This is called the **Born Rule**, after Max Born.) The components of the pair *index* the *basic outcomes* $\mathbf{0}$ and $\mathbf{1}$. There are two ways we can gain knowledge about the values a and b:

- We can **prepare** the state from the known initial state $\mathbf{e}_0 = [1, 0]$ by known quantum operations, which here can be represented by 2×2 matrices.
- We can **measure** the state (with respect to these basic outcomes), in which case:
  - We either **observe** $\mathbf{0}$, whereupon the state becomes $\mathbf{e}_0$, or we observe $\mathbf{1}$, in which case the state becomes $\mathbf{e}_1 = [0, 1]$.
  - The probability of observing $\mathbf{0}$ is $|a|^2$, of getting $\mathbf{1}$ is $|b|^2$.

If both a and b are real numbers, then we can picture the qubit as a point on the unit circle in $\mathbb{R}^2$:



The qubit (a, b) represents $ae_0 + be_1 = a(1,0) + b(0,1) = a|0\rangle + b|1\rangle$

What can be confusing in the diagram is that we also habitually use the unit circle in $\mathbb{R}^2$ to illustrate a single unit complex number c, that is, an element of $\mathbb{C}^1$ of magnitude 1. We would then write $c = a + bi$, and then $|c|^2 = 1$ is the same as $a^2 + b^2 = 1$. Our pair (a, b) of complex numbers, however, is an element of $\mathbb{C}^2$, which is 4-dimensional if we tried to view it in real space.

Multi-Qubit Matrices and Gates

The tensor product of two basic Hadamard gates is

$$\mathbf{H}^{\otimes 2} = \mathbf{H} \otimes \mathbf{H} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right].$$

This matrix carries the orthonormal two-qubit standard basis $e_{00}, e_{01}, e_{10}, e_{11}$ onto the four combinations of tensoring the $|+\rangle$ and $|-\rangle$ states, namely (transpose $\{\}^T$ omitted):

$$\begin{aligned}
|{++}\rangle &= |+\rangle \otimes |+\rangle = \tfrac{1}{2}(1,\ 1) \otimes (1,\ 1) = \tfrac{1}{2}(1,\ 1,\ 1,\ 1) = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \\
|+\,-\rangle &= |+\rangle \otimes |-\rangle = \tfrac{1}{2}(1,\ 1) \otimes (1,-1) = \tfrac{1}{2}(1,-1,\ 1,-1) = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2} \\
|-\,+\rangle &= |-\rangle \otimes |+\rangle = \tfrac{1}{2}(1,-1) \otimes (1,\ 1) = \tfrac{1}{2}(1,\ 1,-1,-1) = \frac{|00\rangle + |01\rangle - |10\rangle - |11\rangle}{2} \\
|-\,-\rangle &= |-\rangle \otimes |-\rangle = \tfrac{1}{2}(1,-1) \otimes (1,-1) = \tfrac{1}{2}(1,-1,-1,\ 1) = \frac{|00\rangle - |01\rangle - |10\rangle + |11\rangle}{2}
\end{aligned}$$

These four vectors are linearly independent and mutually orthogonal, so they form an orthonormal basis. We can see the mapping because forming the target vectors into a matrix (as column vectors) gives us exactly $\mathbf{H}^{\otimes 2}$.

Well, this is the case $m = 2$ of the Hadamard transform $\mathbf{H}^{\otimes m}$. Also note the following tensor products of 2×2 matrices:

$$\mathbf{H} \otimes \mathbf{I} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{array}\right],$$

$$\mathbf{I} \otimes \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right].$$

Some examples of states you can produce with these matrices are:

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

$$|0+\rangle = |0\rangle \otimes |+\rangle = \frac{1}{\sqrt{2}}(1,0) \otimes (1,1) = \frac{1}{\sqrt{2}}(1,1,0,0) = \frac{|00\rangle + |01\rangle}{\sqrt{2}}$$

Meanwhile,

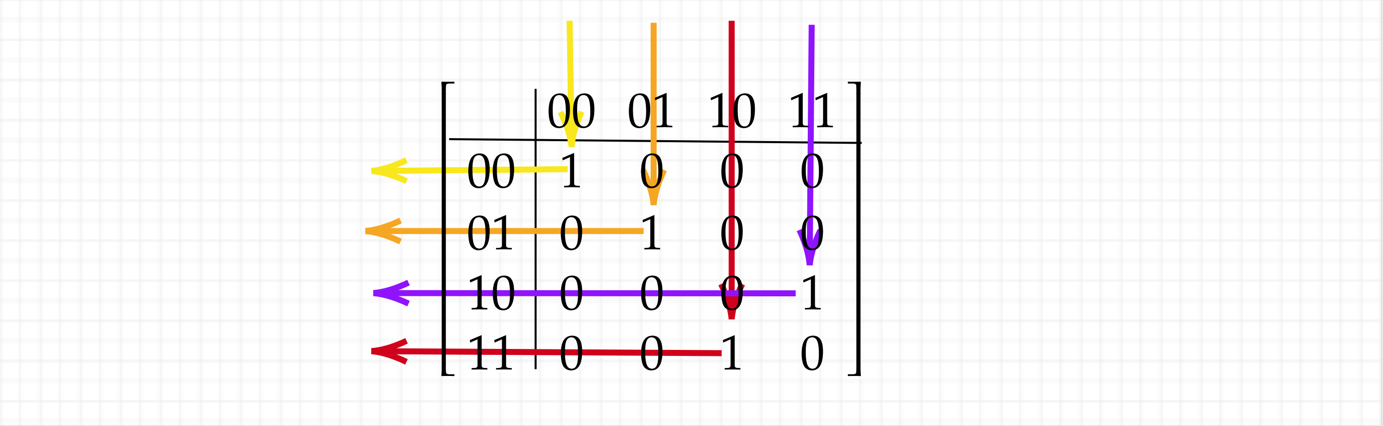
$$|{+1}\rangle = |{+}\rangle \otimes |1\rangle = \frac{1}{\sqrt{2}}(1,1) \otimes (0,1) = \frac{1}{\sqrt{2}}(0,1,0,1) = \frac{|01\rangle + |11\rangle}{\sqrt{2}}$$

can be gotten as $\mathbf{H} \otimes \mathbf{I}$ applied to the column vector $(0,1,0,0)^T = |01\rangle$. However, the state $\frac{1}{\sqrt{2}}(1,0,0,1) = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$, which we saw in the last lecture is entangled, cannot be gotten this way. Instead, it needs the help of a 4×4 unitary matrix that is not a tensor product of two smaller matrices. The most omnipresent one of these is:

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

Any linear operator is uniquely defined by its values on a particular basis, and on the standard basis, the values are: $\mathbf{CNOT}e_{00} = \mathbf{CNOT}|00\rangle = |00\rangle$, $\mathbf{CNOT}e_{01} = \mathbf{CNOT}|01\rangle = |01\rangle$, $\mathbf{CNOT}e_{10} = \mathbf{CNOT}|10\rangle = |11\rangle$, and $\mathbf{CNOT}e_{11} = \mathbf{CNOT}|11\rangle = |10\rangle$. We can get these from the respective columns of the $\mathbf{CNOT}$ matrix, and we can label the quantum coordinates right on it:

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



$$\left[\begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 00 & 1 & 0 & 0 & 0 \\ 01 & 0 & 1 & 0 & 0 \\ 10 & 0 & 0 & 0 & 1 \\ 11 & 0 & 0 & 1 & 0 \end{array}\right]$$

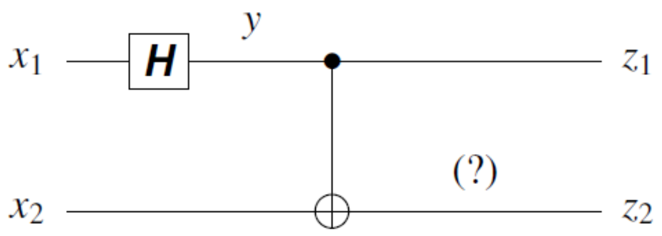
Because we multiply column vectors, the co-ordinates of the argument vector come in the top and go out to the left. If the first qubit is 0, then the whole gate acts as the identity. But if the first qubit is 1, then the basis value of the second qubit gets flipped---the same action as the **NOT** gate $\mathbf{X}$. Hence the name Controlled-NOT, abbreviated $\mathbf{CNOT}$: the **NOT** action is controlled by the first qubit. The action on a general 2-qubit quantum state $\phi = (a,b,c,d)$ is even easier to picture:

$$\mathrm{CNOT}\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \\ b \\ d \\ c \end{pmatrix}.$$

All it does is switch the third and fourth components---of any 4-dim. state vector. Hence, CNOT is a **permutation gate** and is entirely deterministic. Permuting these two indices is exactly what we need to transform the separable state $\frac{1}{\sqrt{2}}(1,0,1,0)$ into the entangled state $\frac{1}{\sqrt{2}}(1,0,0,1)$. Since we got the former state from $\mathbf{H} \otimes \mathbf{I}$ applied to $\mathbf{e}_{00}$, the matrix we want is

$$\mathrm{CNOT} \cdot (\mathrm{H} \otimes \mathrm{I})|00\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \left[\begin{array}{c|ccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{array}\right] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We can see the result coming from the first column. When we do a quantum circuit left-to-right, however, the $(\mathbf{H} \otimes \mathbf{I})$ part comes first on the left. The symbol for a **CNOT** gate is to use a black dot to represent the control on the *source qubit* and $\oplus$ (which I have used as a symbol for XOR) on the *target qubit*. This is more easily pictured by a **quantum circuit diagram**:



If $x_1 = |0\rangle$, then we can tell exactly what y is: it is the $|+\rangle$ state. And if $x_1 = |1\rangle$, then $y = |-\rangle$. If x_1 is any separate qubit state $(a,b) = a|0\rangle + b|1\rangle$, then by linearity we know that $y = a|+\rangle + b|-\rangle$. This expresses y over the transformed basis; in the standard basis it is

$$\frac{1}{\sqrt{2}}(a(1,1) + b(1,-1)) = \frac{1}{\sqrt{2}}(a+b, a-b)\ .$$

So we can say exactly what the input coming in to the first "wire" of the CNOT gate is. And the input to the second wire is just whatever x_2 is. But because that gate does entanglement, we cannot specify individual values for the wires coming *out*. The state is an inseparable 2-qubit state:

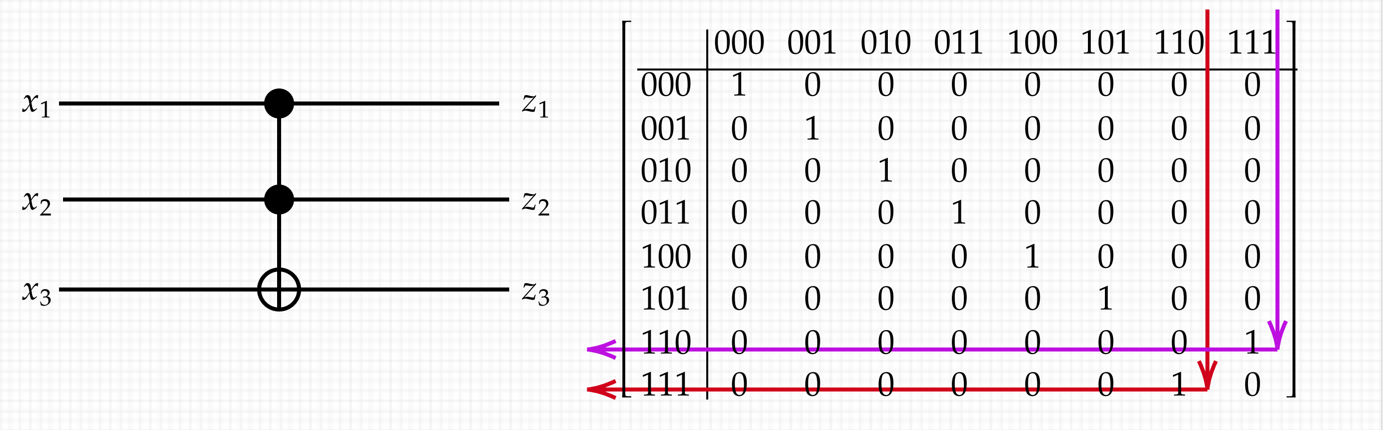
$$\frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big).$$

If you measure either qubit individually, you get 0 or 1 with equal probability. This is the same as if you measured the state $|{+}{+}\rangle$. But that state is outwardly as well as inwardly different. When *both* qubits to be measured, it allows 01 and 10 as possible outcomes, whereas measuring the entangled state does not. I've seen papers telling ways to visualize entangled states of 2 or 3 qubits, but none implemented by an applet so far---`quantum-circuit.com` just shows Bloch spheres with the black dot at the center for the "completely mixed state": | ¯_(ツ)_/¯ ⟩.

Three Qubits and More

The **CNOT** gate by itself has the logical description $z_1 = x_1$ and $z_2 = x_1 \oplus x_2$. This logical description is valid only for standard basis states. It means that if $x_1 = 0$ then $z_2 = x_2$, but if $x_1 = 1$ then $z_2 = \neg x_2$. Since this description is complete for all of the standard basis inputs $x = x_1 x_2 = 00, 01, 10, 11$, it extends by linearity to all quantum states. We can use this idea to specify the 3-qubit **Toffoli gate** (**Tof**). It has inputs x_1, x_2, x_3 (representing the components in each basis state) and symbolic outputs z_1, z_2, z_3 (which, however, might not have individual values in non-basis cases owing to entanglement). Its spec in the basis quantum coordinates is:

$$z_1 = x_1, \; z_2 = x_2, z_3 = x_3 \oplus (x_1 \wedge x_2).$$



$$\left[\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
\hline
000 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
001 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
010 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
011 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
100 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
101 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
110 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
111 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}\right]$$

Of particular note is that if x_3 is fixed to be a constant-1 input, then

$$z_3 = \neg(x_1 \wedge x_2) = NAND(x_1, x_2).$$

or rather

$$z_3 = x_3 \, \boldsymbol{XOR} \, (x_1 \wedge x_2) = x_3 \, XOR \, AND(x_1, x_2)$$

if $x_3 = 1$, then we get $1 \oplus (x_1 \wedge x_2) = \neg(x_1 \wedge x_2) = NAND(x_1 \wedge x_2)$.

Thus the Toffoli gate subsumes a classical NAND gate, except that you need an extra "helper wire" to put $x_3 = 1$ and you gate two extra output wires z_1, z_2 that only compute the identity on x_1, x_2 (in classical logic, that is---a non-basis quantum state can have knock-on effects even though all Toffoli does is switch the 7th and 8th components of the state vectors). If you have polynomially many Toffoli gates, then you get only polynomially much wastage of wires, and you can use the good ones to simulate any polynomial-size Boolean circuit of NAND gates. Because polynomial-time algorithms can be simulated by polynomial-sioze circuits, we have:

Theorem: P $\subseteq$ DQP $\subseteq$ BQP.

Well, we need to say more broadly what it means for quantum computations to be (polynomially) **feasible**. The community convention is simply to count up gates of 1, 2, or 3 qubits as constant cost. Gates involving more qubits are OK if they can be built up out of the small gates. We have already seen that $\mathbf{H}^{\otimes n}$ is just n binary Hadamard gates laid out in parallel. The n-qubit **quantum Fourier transform** can be built up out of $O(n^2)$ smaller gates---this actually has more "fine print" than sources usually say and is pursued in the chapter exercises of the textbook.

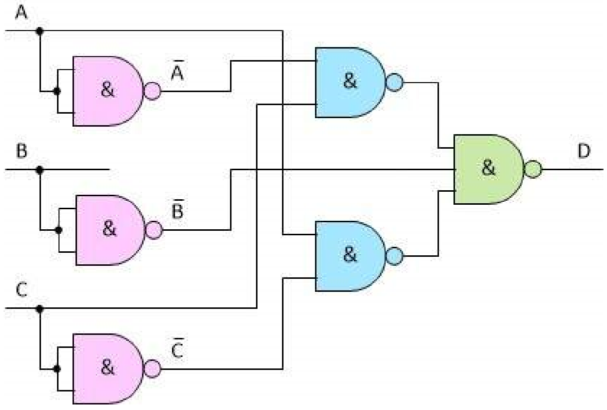
And BQP is to DQP as BPP is to P. We should describe measurements in more detail and see smaller-scale deterministic and randomized examples first.

Quantum Circuit Examples

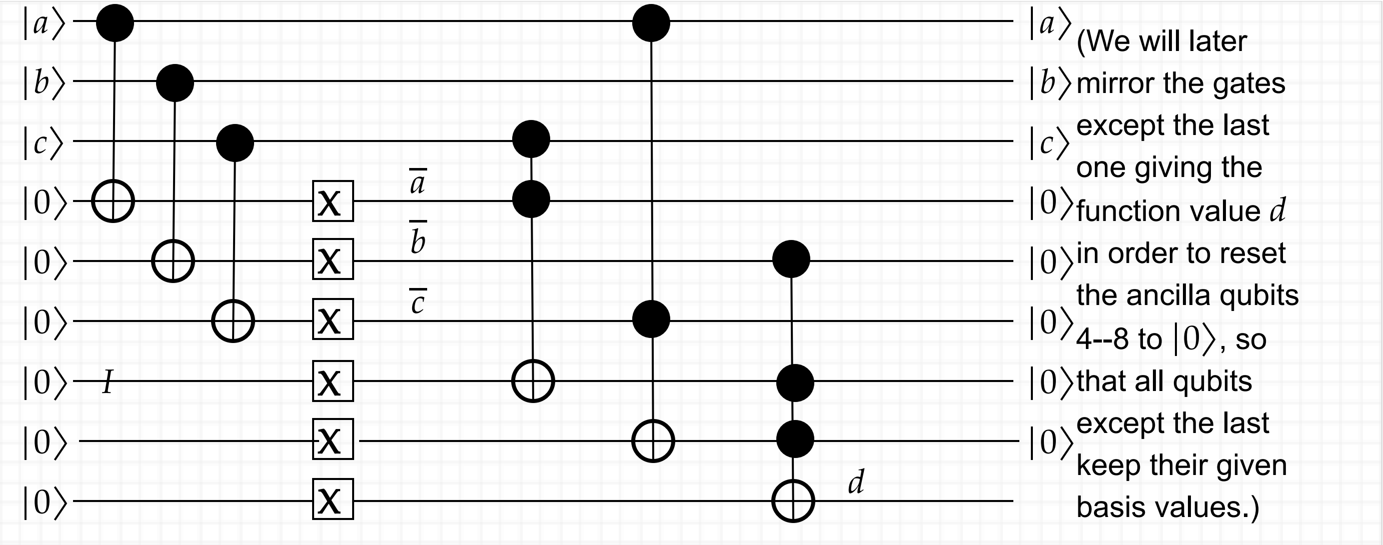
Theorem (cf. theorem 5.2 in section 5.3): Classical Boolean circuits can be efficiently simulated by quantum circuits that don't even do any superposition or entanglement.

The proof is basically that the Toffoli gate simulates NAND via $\mathrm{Tof}(x, y, 1) \;=\; (\overline{x} \;\vee\; \overline{y})$ and NAND is a universal gate. The extra lines for the constant 1 inputs also make the whole computation **reversible**. That is, $\mathrm{Tof}(x, y, z) \;=\; (x, y, z \oplus (\overline{x} \;\vee\; \overline{y}))$ is reversible. [RevNAND$(x, y) = (x, \overline{x} \;\vee\; \overline{y})$? (no, not reversible)]

Here is a sizable example of this theorem. Consider the following circuit of NAND gates from the blog article "Implementing Logic Functions Using Only NAND or NOR Gates" by Max Maxfield:



Here is the corresponding quantum circuit:



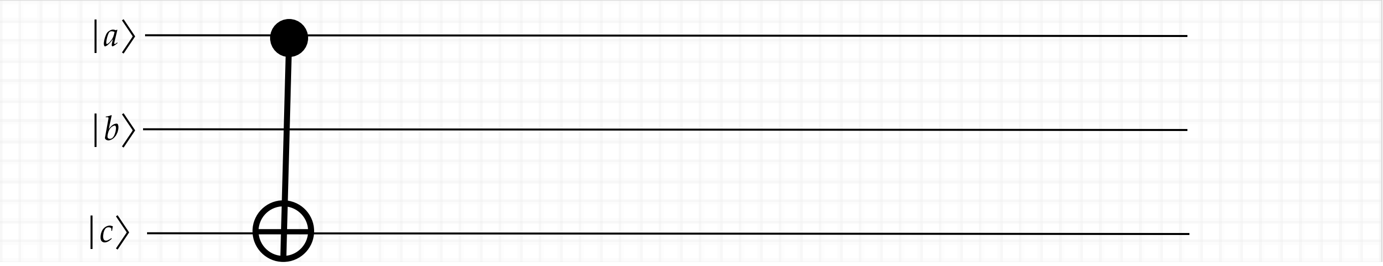
(We will later mirror the gates except the last one giving the function value d in order to reset the ancilla qubits 4--8 to $|0\rangle$, so that all qubits except the last keep their given basis values.)

Note also that the initial three $\mathbf{CNOT}$ gates effectively copy the Boolean values a, b, c so that they can be negated as $\bar{a}, \bar{b}, \bar{c}$ on the next three qubit lines. This is covered in section 6.2, and the last three qubit lines exemplify the trick in section 6.1 of using $\mathbf{NOT}$ gates to effectively initialize them to $|1\rangle$ rather than $|0\rangle$. *Caveat:* You can't copy an arbitrary quantum state using $\mathbf{CNOT}$---the **No-Cloning Theorem** mentioned in section 6.2 shows there is no way to do this in general. But particular states in a known basis can be copied this way.

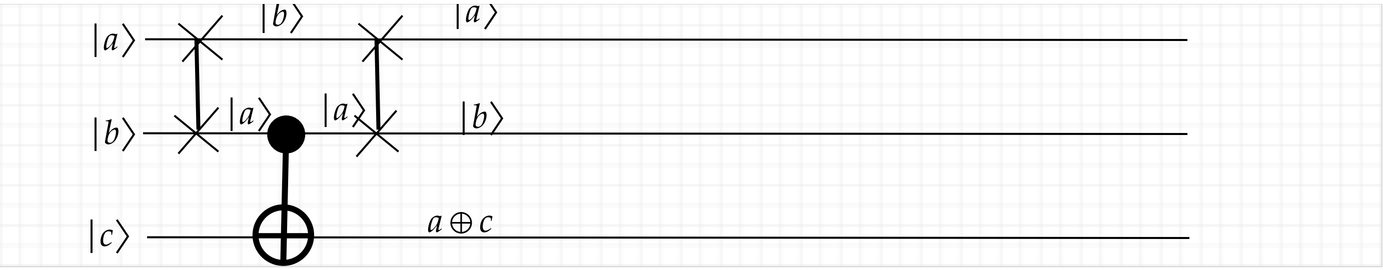
The "quantum extra", beginning with using the Hadamard gate to create superpositions, is what promises to take us beyond classical computing.

Circuits and Computations

Just like music can be divided into measures with a basic 'beat' unit, quantum gates going left to right are timesteps of a computation. If multiple gates are underneath each other, then they make a single tensor-producted operation---such as $\mathbf{X}^{\otimes 6}$ in the above diagram. If nothing happens on a certain qubit line at a given timestep, that is mathematically like tensoring with the identity matrix. A "squidgy" point has to do with the nearest-neighbor aspect of tensor products. Consider:



There is notation for $\mathbf{I} \otimes \mathbf{CNOT}$ and $\mathbf{CNOT} \otimes \mathbf{I}$, but not for "$\mathbf{I}$ in the middle." We can ignore this problem. Or---and often this has to be done with real tech---we can suppose the Swap gate is applied twice, e.g.



$$\mathbf{SWAP} \;=\; \left[\begin{array}{ccccc} & 00 & 01 & 10 & 11 \\ 00 & 1 & 0 & 0 & 0 \\ 01 & 0 & 0 & 1 & 0 \\ 10 & 0 & 1 & 0 & 0 \\ 11 & 0 & 0 & 0 & 1 \end{array}\right]$$

In such manner, we get the n-qubit circuit as a composition

$$C \;=\; U_t \circ U_{t-1} \circ \;\cdots\; \circ\; U_1$$

of $N \times N$ unitary matrices.

Principle of Linearity: For any quantum state $\boldsymbol{\Phi} \;=\; \sum_{i=1}^{N} a_i \mathbf{e}_i$,

$$C(\boldsymbol{\Phi}) \;=\; \sum_{i=1}^{N} a_i \, C\mathbf{e}_i \; .$$

In words, the action of a quantum circuit on any quantum state is determined by its actions on the (standard) basis states.

General Controlled Gates

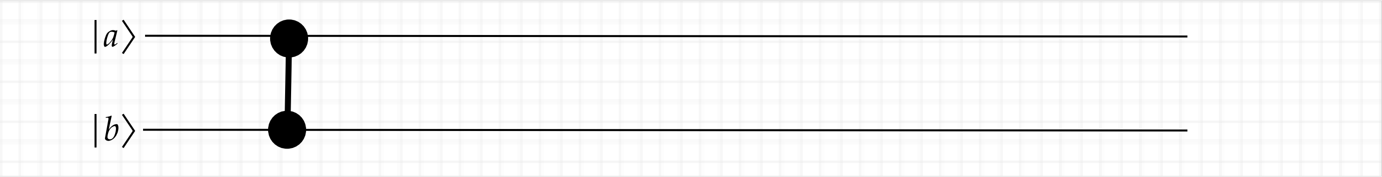
Related to the **CNOT** gate is the controlled version of the Z gate. Recall $\mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. The controlled version of any matrix A (in the standard basis) is the block matrix

$$\mathbf{C}A \;=\; \begin{array}{c|c|c} & 0u & 1u \\ \hline 0u & \mathbf{I} & 0 \\ \hline 1u & 0 & A \end{array},$$

where the hierarchical quantum indexing scheme is also shown. If the first qubit is 0 then the effect is the identity, while if it is 1, then the effect on the remainder $|u\rangle$ is to apply A. So

$$\mathbf{CZ} \;=\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Although the control is nominally on the first qubit, with $\mathbf{CZ}$ the effect on base states is to multiply the global state by -1 if and only if *both* qubits are 1. Hence it is really symmetric between qubits---the second qubit could equally be said to be controlling the first. The standard diagram for it is just two black dots connected by themselves:



Since a general vector $[u_1, u_2, u_3, u_4]^T$ becomes $[u_1, u_2, u_3, -u_4]^T$ after going through $\mathbf{CZ}$, it follows, upon writing $|a\rangle = [a_1, a_2]^T$ and $|b\rangle = [b_1, b_2]^T$, that

$$\mathbf{CZ}\cdot\big(|a\rangle \otimes |b\rangle\big) \;=\; \mathbf{CZ}\cdot[a_1b_1, a_1b_2, a_2b_1, a_2b_2]^T \;=\; [a_1b_1, a_1b_2, a_2b_1, -a_2b_2]^T.$$

Is this ever entangled, and if so, when? Note that if $|a\rangle$ and $|b\rangle$ are both $|1\rangle$, then $\mathbf{CZ}\cdot\big(|a\rangle \otimes |b\rangle\big) \;=\; \mathbf{CZ}|11\rangle \;=\; \mathbf{CZ}\cdot[0,0,0,1]^T \;=\; [0,0,0,-1] \;=\; -[0,0,0,1] \;=\; -|11\rangle$. To try to represent this as a tensor product $\begin{bmatrix} e \\ f \end{bmatrix} \otimes \begin{bmatrix} g \\ h \end{bmatrix} = [eg, eh, fg, fh]^T$, we need both e and g to be 0, so we are left with $fh = -1$. This is easy to solve with $f = 1$ and $h = -1$, or even $f = h = i$ since we can use complex numbers.

But now let $|a\rangle$ and $|b\rangle$ both be $|+\rangle$. Then we get

$$\mathbf{CZ}|{+}{+}\rangle \;=\; \mathbf{CZ}\cdot\tfrac{1}{2}[1,1,1,1]^T \;=\; \tfrac{1}{2}[1,1,1,-1]^T.$$

For determining entanglement we can ignore the $\frac{1}{2}$ factor. So the equations become $eg = 1$, $eh = 1$,

$fg = 1$, and $fh = -1$. The first three combine to give $g = \frac{1}{e} = h$, so $fg = fh = 1$, but that contradicts the fourth equation $fh = -1$. Thus $\mathbf{CZ}|{+}{+}\rangle$ is entangled. It follows that

> **It is possible for a quantum gate to leave one separable state separable while making another separable state become entangled**.

Example and a Circuit Diagram Pitfall

Note: It is tempting to think that CZ should be the transform of CNOT under the $H^{\otimes 2}$ change of basis. [I did this on the whiteboard. See the "Graph States" section of https://rjlipton.com/2022/01/05/quantum-graph-theory/ for the end-result matrix, there called "$\mathbf{E}$". I will type this up when I can.]