Homework 3 8.370/18.435/2.111 Due 9/28

1. Find a matrix whose square is σ_x.

   Hint: There are lots of ways of doing this. One way is to first find a matrix whose square is σ_z.

   Solution: Since σ_z is diagonal, finding a square root of it is easy: $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ is a square root of σ_z. Now, recall that $H\sigma_z H = \sigma_x$, so the square of HSH is

$$(HSH)(HSH) = HS^2H = H\sigma_z H = \sigma_x \, .$$

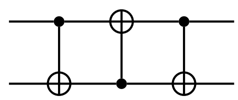
   And matrix multiplication gives

$$HSH = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$$

2. **Making a SWAP gate**

   Show how to create a SWAP gate using 3 CNOT gates.

   Solution: The following is a SWAP gate.



3. **Controlled Hadamard gate**

   Write down the 8×8 matrix that corresponds to a controlled Hadamard gete where the control is bit 3 and the target is bit 1.

   Solution:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

4. **The Fredkin gate**

   The Fredkin gate is a classical reversible three-bit gate. If the input is (x, y, z), then the output is $(0, y, z)$ if $x = 0$ and $(1, z, y)$ if $x = 1$. In other words, it is a controlled SWAP: if x is 0, you do nothing, and if x is 1, you swap y and z.

   (a) Show how you can obtain the gates AND, OR, NOT, and FANOUT by putting fixed values in one or two of the inputs of the Fredkin gate.

   Solutions:

i. AND: $F(x, 0, z) = F(x, x \wedge z, ?)$,
ii. OR: $F(x, 1, z) = F(x, ?, x \vee z)$,
iii. NOT: $F(x, 1, 0) = F(x, \neg x, x)$,
iv. FANOUT $F(x, 1, 0) = F(x, \neg x, x)$.

(b) Show that the Fredkin gate preserves the number of 1s in the system. This is useful, for example, if you want to show how to build a circuit that is implemented by ideal colliding billiard balls.

Solutions: You can look at the truth table, and see that the number of 1's is the same. An easier method is just observing that the only time a Fredkin changes the values of bits is when bits y and z are swapped, and swapping two bits does not change the number of 1's.

(c) A half-adder takes as input two bits x and y, and has two outputs $x \wedge y$ (x AND y) and $x \oplus y$ (x XOR y). Show how to build a half-adder using Fredkin gates (you will need more than two output bits).

Solution: With the AND gate above, it's easy to get the AND in the half-adder. But how do you get the XOR? It's not possible with just one input gate. One way is to use the formula

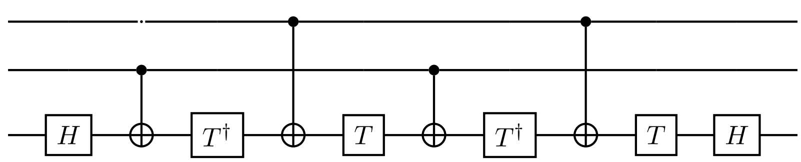
$$\mathrm{XOR}(a, b) = (a \wedge \neg b) \vee (\neg a \wedge b)$$

There's another way which uses fewer gates. Suppose you have four input bits, $(a, b, 0, 1)$. Now, use a Fredkin gate on bits $(1, 3, 4)$, and then use a Fredkin gate on bits $(2, 3, 4)$. If exactly one of a and b is 1, then bits 3 and 4 will be swapped once, and bit 3 will be a 1. If both or neither of a and b are 1, then they will either be swapped twice or not swapped at all, and bit 3 will be a 0. This gives an XOR.

Note that in this construction for XOR, we preserve the two input bits. We can then feed these into another Fredkin gate which takes their AND. So three Fredkin gates is sufficient to solve the problem.

5. **Constructing a quantum Toffoli gate**

(a) Consider the following quantum circuit, made from one-qubit gates and four CNOT gates:

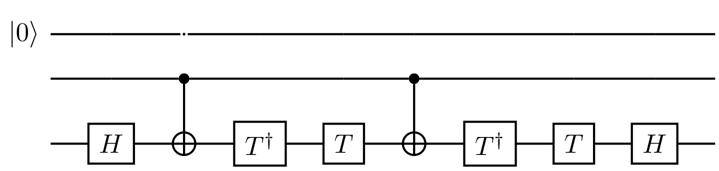


where the T gate is

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{2} \end{pmatrix}.$$

Show that this nearly yields a Toffoli gate. What unitary transformation does it implement, and how does it differ from a Toffoli gate?

Solution: If the top qubit is $|0\rangle$, the circuit reduces to



The $-\boxed{T^\dagger}-\boxed{T}-$ gates multiply to I, and then the two CNOT gates multiply to I, and finally the two H gates multiply to I, and we get the identity transformation.
If the second qubit is $|0\rangle$, a very similar cancellation occurs.
So the only thing we need to worry about is when the first two qubits are $|1\rangle$. In this case, the third bit undergoes the transformations.

$$-\boxed{H}-\boxed{X}-\boxed{T^\dagger}-\boxed{X}-\boxed{T}-\boxed{X}-\boxed{T^\dagger}-\boxed{X}-\boxed{T}-\boxed{H}- .$$

The middle stretch (between the two H gates) is $T\big(\sigma_x T^\dagger \sigma_x\big)T\big(\sigma_x T^\dagger \sigma_x\big)$. Here, $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$, so $\sigma_x T^\dagger \sigma_x = \begin{pmatrix} e^{-i\pi/4} & 0 \\ 0 & 1 \end{pmatrix}$, and

$$T\big(\sigma_x T^\dagger \sigma_x\big)T\big(\sigma_x T^\dagger \sigma_x\big) = \begin{pmatrix} e^{-i\pi/2} & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} = -i\sigma_z$$

But $H(-i\sigma_z)H = -i\sigma_x$, so the transformation is

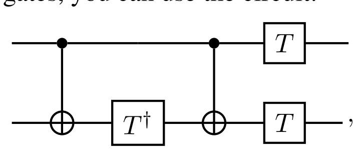
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i \\ 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \end{pmatrix}$$

(b) Explain how you can add one-qubit gates and CNOTs to the first two quantum wires to make this a true Toffoli gate.

Solution: If you can implement the gate

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

on the first two wires, you will correct the phase and have a true Toffoli gate. This is a controlled T^2 gate, and we explained how to do controlled gates in class. If you just want to use CNOT gates and T gates, you can use the circuit:



which can be analyzed very much like the circuit in part (a).