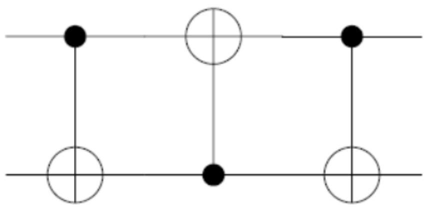
P238 Atomic Physics 2018 Cheng Chin

Home Work 5

(Due: 12/6/2018)

1. Two-qubit quantum logic operation

A. Show that the following combination of three C-NOT gates is equivalent to a SWAP gate:



Hint: C-NOT gate operation is given by $U_{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$

B. What is the inverse operation of U_{CNOT}?
Hint: The inverse operator U_{CNOT}^{-1} should satisfy $U_{CNOT}^{-1}U_{CNOT}=1$.

C. Is it possible to create a "two-way" control-NOT such that
If the input state of either qubit is |1> then flip the quantum state of the other qubit?

2. What a standard quantum computer can and cannot do.

A quantum computer can process *N* qubits in their initial state $|\varphi\rangle = |\varphi_1, \varphi_2 \dots, \varphi_N\rangle$ and yield the final state $|\phi\rangle = |\phi_1, \phi_{2\dots}, \phi_N\rangle$, where all φs and ϕs are either 0 or 1. The quantum operations U is a unitary mapping that links the input $\varphi = \{\varphi_1, \varphi_2 \dots, \varphi_N\}$ to the output $\phi = \{\phi_1, \phi_{2\dots}, \phi_N\}$ as $|\phi\rangle = U|\varphi\rangle$ with the specific function F such that $\phi = F(\varphi)$. Here we will discuss some of the functions that quantum computers can and cannot do.

A. Multiplication: Could you design a quantum circuit to do the multiplication $F(x)=2x$?
Hint: Here we can consider $\{\varphi_1, \varphi_2 \dots, \varphi_N\}$ as the binary code of the number φ, and the same for ϕ
For example, given 5 qubits $\{\varphi_1, \varphi_2 \dots, \varphi_5\}=\{1,1,0,0,0\}$, $\varphi=3$ and thus the output should be $\phi=2*3=6=\{0,1,1,0,0\}$. Can a quantum computer do such multiplication?

B. Addition: Is it possible to design a quantum circuit such that output $\phi=\varphi_1+\varphi_2+\varphi_{3\dots}+\varphi_N$?
Hint: For example, given {0,1,0,1,1} , $\phi=0+1+0+1+1=11$ and thus output = $\{1,1,0,0,0\}$

B. Sorting: Can we sort the input $\{\varphi_1, \varphi_2 \dots, \varphi_N\}$ and yield the output such that $\phi_N \geq \phi_{N-1\dots} \geq \phi_1$?
Hint: As an example {0,1,0,0,1} is sorted into {0,0,0,1,1}

C. Copy and paste: Can we copy the first half entries of the input and paste it to the second half, namely $\{\varphi_1, \varphi_2 ...., \varphi_N\}$ -> $\{\varphi_1, \varphi_2 ...., \varphi_{N/2}, \varphi_1, \varphi_2 ...., \varphi_{N/2}\}$, assuming N is even.

D. Memory: Can we store at least part of the input states $\{\varphi_1, \varphi_2 ...., \varphi_N\}$ in the computer such that all following computation can depend on it when the input state changes to $\{\varphi_1', \varphi_2' ...., \varphi_N'\}$?

E. Random number generator: Can we generate a random output ϕ independent of the input φ?

F. How would you design a quantum computer such that the above classical operations can be realized?

Hint: Many of the above functions demands non-unitary evolution of the system that increases or decreases its entropy. Argue that one can resolve the issue if you only get the output from a subset of the qubits after the computation, and ignore the rest qubits. In other words, the output is defined as $\phi = \{\phi_1, \phi_2 ..., \phi_M\}$ with $M<N$.