Assignment 2 - SoC

To the Quantum Future

July 20, 2023

Here is the assignment for the next week of the SoC; as usual, you are expected to submit your attempts to these questions on your forked/personal git repo.

Some basic arithmetic operations for this week on quantum computers.

An n-bit number x can be encoded as the state $|x\rangle$ of a system of n qubits, called a register.

(a) Code up the circuit to swap the states of two 32-bit integers (swap the values of the two registers).

(b) Given a three bit register *abc*, code up a circuit to increment the number by 1 (mod 8). The result should be stored in-place, i.e., in the same register that are used for the inputs.

(c) (Optional) If you want to take this further, can you code up a circuit that takes two three bit registers and adds them up (mod 8), storing the result in the qubits for the second number?

Try doing this first using some *ancilla* qubits, i.e., extra qubits pre-initialised to some fixed state, which can serve as some place to store information. A challenge would be to do it without using any extra qubits.

Also, it's possible to do the previous two parts just with the Pauli-X, CNOT and Toffoli gates!

(d) The **Hamming Weight** of a binary number is the number of 1s in its binary representation.

For a binary number with 3 bits, construct a circuit that takes $|x\rangle \otimes |00\rangle$ to $|x\rangle \otimes |w(x)\rangle$ where $w(x)$ is the Hamming weight of x.

(Optional) The hamming distance h between two n-bit strings is the number of positions at which the corresponding bits are different. Can you use the above part to take $|x\rangle \otimes |y\rangle$ to $|x\rangle \otimes |h(x,y)\rangle$?