Simulation and Quantum Computing

Houd Hassani

Advisor: Professor R ichard Sowers

REU Final Report

Spring 2022

Department of Industrial and Enterprise Systems Engineering

[ise.illinois.edu](https://ise.illinois.edu)

A picture containing bottle

Description automatically generated

This project focused on quantum computing, a developing technology which promises to disrupt the science of computation. Globally, over $24B has been invested in quantum technology[[1]](#footnote-1). Venture capital has also seen this as a growth area, investing over $1B[[2]](#footnote-2). The University of Illinois is positioning itself as a leader in the field, partnering with IBM to form the Discovery Accelerator Institute[[3]](#footnote-3).

Part of the explosive growth of interest in quantum technology comes from Schor’s algorithm[[4]](#footnote-4), which outlines a procedure for factoring integers in less computational time than is currently needed on classical computers. This has disruptive implications for cryptography. Abstractly, quantum computing seems to be “fast but noisy”. The promise of quantum computing depends on approaching problems in such a way that the speed of quantum computing outweighs the inherent noise in computation.

Our interest is in understanding some of the basic concepts of quantum computing. Qiskit[[5]](#footnote-5), an open-source python package for quantum computing, provides an entry-level framework for programming with quantum computing and thus learning some elementary ideas. This REU is focused on delving into some basic concepts, admittedly for the edification of the REU student and his faculty mentor. The result of this is a git repo, at <https://gitlab.engr.illinois.edu/r-sowers/quantum-computing>.

Our effort ended up clarifying a number of operational issues of quantum computing; each of these is developed in a separate notebook at the git repo.

* Quantum computing develops ‘quantum registers’ (collections of qubits) which are then converted into ‘classical registers’. We understood the framework for this[[6]](#footnote-6); part of the complexity of quantum computing is carefully thinking through the steps of converting quantum bits (qubits) into classical bits, i.e., measuring the result of a quantum computation. This measurement “collapses” quantum states, and should thus be thought through carefully.
* Quantum computing seems to still be somewhat in its infancy (most of IBM’s backends have less 5 or fewer qubits). A number of decisions about the order in which qubits are labelled and measured can still be set by the user. We understood[[7]](#footnote-7) some of these this provided useful background in what is going on.
* One of the simplest logic components of quantum computing is the CNOT (controlled NOT) gate, which implements an exclusive or as a combination of a “control” qubit and a “target” qubit. We understood the CNOT gate in several ways[[8]](#footnote-8).

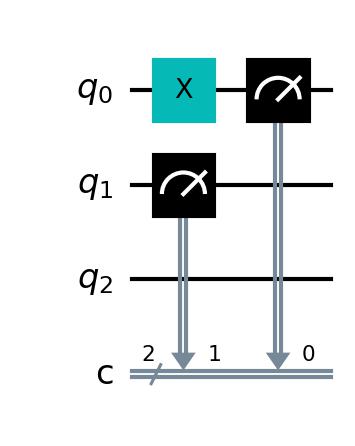


Figure : 3 quantum registers and 2 classical registers

Between the hardware implementation of quantum computing and qiskit implementation is a fair amount of mathematics. Quantum behavior is modelled by “wave functions” which quantify the probability that the quantum system will be in various states. The Bloch sphere is a useful visualization of this. The wave functions are typically written in terms of complex functions. Linear algebra and its framework for superposition gives a way to manipulate the likelihood that a system will be in a given state. Gates and operations can be described as *unitary* matrices. Kronecker products provide a consistent way of scaling ideas to many qubits. We understood a number of relevant ideas[[9]](#footnote-9).

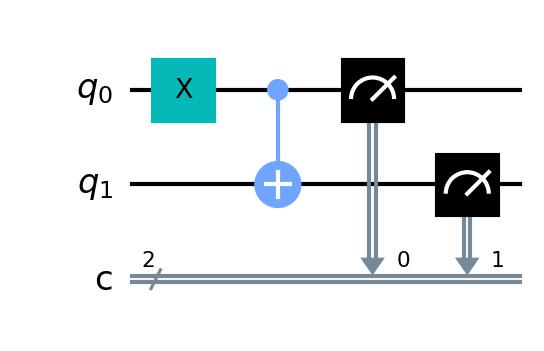


Figure : Controlled Not Gate

We did not make much progress in simulation. The semester’s efforts focused primarily on building a foundational understanding of things.

We did not touch upon

Bell states[[10]](#footnote-10); this is a canonical example of entanglement.

Direct conversion between operations and matrices. Basic logic gates (e.g., the CNOT gate) can be represented as unitary operations; in fact all unitary operations can be encoded into quantum computing[[11]](#footnote-11).

1. <https://qureca.com/overview-on-quantum-initiatives-worldwide-update-mid-2021/> [↑](#footnote-ref-1)
2. <https://pitchbook.com/news/articles/quantum-computing-venture-capital-funding> [↑](#footnote-ref-2)
3. <https://discoveryacceleratorinstitute.grainger.illinois.edu/> [↑](#footnote-ref-3)
4. <https://en.wikipedia.org/wiki/Shor%27s_algorithm> [↑](#footnote-ref-4)
5. <https://qiskit.org/> [↑](#footnote-ref-5)
6. <https://gitlab.engr.illinois.edu/r-sowers/quantum-computing/-/blob/main/QiskitExploration_Registers.ipynb> [↑](#footnote-ref-6)
7. <https://gitlab.engr.illinois.edu/r-sowers/quantum-computing/-/blob/main/QiskitExploration_Ordering.ipynb> and <https://gitlab.engr.illinois.edu/r-sowers/quantum-computing/-/blob/main/QiskitExploration_Measure.ipynb> [↑](#footnote-ref-7)
8. <https://gitlab.engr.illinois.edu/r-sowers/quantum-computing/-/blob/main/QiskitExploration_CNot.ipynb> [↑](#footnote-ref-8)
9. <https://gitlab.engr.illinois.edu/r-sowers/quantum-computing/-/blob/main/Matrices.ipynb> [↑](#footnote-ref-9)
10. <https://en.wikipedia.org/wiki/Bell_state> [↑](#footnote-ref-10)
11. <https://qiskit.org/documentation/tutorials/circuits_advanced/02_operators_overview.html> [↑](#footnote-ref-11)