Getting started coding quantum computers with QASM and Qiskit

Initial research report

Dylan Miracle

ICS 698-02

Spring 2021

Mar 23, 2021

Dr. Jigang Liu

Getting started coding quantum computers with QASM and Qiskit

Dylan Miracle
Department of Computer Science
Metropolitan State University
St. Paul, Minnesota, USA
dylan.miracle@my.metrostate.edu

May 1, 2021

Contents

1 Introduction

In the early days of digital computing a programmer needed a deep understanding of electrical engineering to program computers. The programming was low level involving digital logic gates and binary code. Abstractions were slowly built in to allow programmers to manipulate abstract data types so that now you can program a computer in a human readable language that uses high level concepts such as object inheritance and lambda calculus. Quantum computing is in its early phase and will likely progress to human human readable abstractions quickly. Developers who want to learn to program quantum computers will be served well to learn a programming framework to experiment with as opposed to taking a PhD in physics before learning the basics of quantum computing. It is not necessary to understand how a computer stores a string to build a web page, we should expect the same kinds of abstractions from quantum computing in the coming decade.

Frameworks such as qiskit and others already give a programmer access to many tools for interpreting and experimenting with quantum computers. This work aims to show an interested developer how to get started programming and interpreting quantum results. After reading this paper, for example, a developer would be able to create a webpage to display quantum results because they could understand how the result data is returned. Without needing to know the ins-and-outs of quantum computing they would be able to manipulate result data with a classical computer. This is a valuable tool because quantum computers will always be used with classical computers.

Quantum computing has a rich theoretical background but practical quantum computers have only recently become available to researchers and business users. These computers are still rare and operated by a small number of hardware vendors. The dominant paradigm for programing quantum computers is the gate model of quantum computing that defines a sequence of operations on a set of qubits. In this paper we introduce a fundamental quantum circuit and demonstrate its implementation using an intermediate representation, QASM, and a higher level quantum computing framework Qiskit.

In this paper we will explore how quantum computing will fit into a standard computing workflow.

This work gives an overview, from mathematical development of a simple quantum circuit to implementation with a high level quantum framework. Readers will get a taste of the complete quantum pipeline. This is valuable because much work focuses on one area of the ecosystem instead of walking those new to quantum mechanics through the complete process of how to implement some quantum code. For example, there are many papers detailing the use of a particular language, or that go into depth on specific quantum algorithms. Even though we focus on a specific framework the model of understanding quantum development proposed by this paper will be useful to anyone who wants to delve further into quantum computing.

2 Background

A basic discussion of quantum computing will involve developing two fundamental concepts in quantum computing, namely superposition and entanglement. Specifically we will use the example of the Bell states. These are the most basic states that exhibit the main features of quantum computing, namely superposition and entanglement of qubits. The development of this basic quantum circuit will start by exploring qubits and some basic quantum operations.

2.1 Qubits

Qubits are represented as two element column vectors. The elements are complex numbers. Since we measure qubits to be either 0 or 1 we will talk about them as being in the computational basis. This is not the only basis we can measure qubits. A single qubit can be visualized as a vector inside a unit sphere. This sphere is known as the Bloch sphere.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

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

$$\otimes$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

2.2 Bell state

We achieve a bell state by using a CNOT gate to combine a $|0\rangle$ with a superposition like $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. This creates what is known as an entangled state. This combination results in a state represented as

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

With the property that any individual measurement of on of the qubits is random, but the measurement of the second qubit is entangled with the measurement of the first. So a measurement of 1 for qubit 0, means we will get a 1 for qubit 1 as well. In the next section we will simulate such a state and visualize it with a histogram.