Session 1: How to Program a Quantum Computer

Dominic Moylett
Quantum Engineering Centre for Doctoral Training,
University of Bristol
dominic.moylett@bristol.ac.uk

July 18, 2016

Worksheet Details

This worksheet is for the second session of the Quantum Computing day in the 2016 "Quantum in the Summer" school. This session is focused on introducing the key concepts in quantum computing: The quantum bits that we use to store data and the quantum gates that we use to manipulate that data.

1 What is Quantum Computing?

Quantum physics is a collection of bizarre behaviours that happen to really small particles. The nature of quantum physics makes simulations of these particles really hard, even on today's fastest computers.

Quantum computing came about after a number of proposals by physicists such as Richard Feynman. They suggested that in order to make computers able to simulate quantum systems, we should build computers that take advantage of the same effects that makes quantum physics hard to simulate. These computers run on a very different model of computation to our current laptops and smart phones, and can lead to more powerful machines in turn. In the decades that have followed since then, academics at many institutions – including the University of Bristol – have studied how quantum physics can benefit our technology, and how we can realise these benefits in practice.

In this workshop, we aim to highlight the building blocks of quantum computers: Quantum bits and quantum gates. We will see how these building blocks can give way to the bizarre physical effects of superposition and entanglement, and how we can take advantage of these concepts to perform challenges such as teleportation. These concepts will be used even further in Session 2, when we look at how quantum computers can speed up our computation.

To understand quantum computing, we will use QCEngine, a quantum computing simulator developed by the University of Bristol Centre for Quantum Photonics' own Eric Johnston. All you need to get started is a laptop with a web browser installed. Open up your browser of choice and go to the website `http://machinelevel.com/qc/doc/qcengine_workbench.html`. This will load the engine with a default script. Simply delete this script and you will then be ready to explore the world of quantum computing!

2 Quantum Bits

A classical computer is made up of bits: 1s and 0s that represent data. Quantum computers store data in quantum bits, or *qubits*. We write qubits as $|1\rangle$ and $|0\rangle$. These symbols are called *kets*[1], and are used to describe quantum states.

In the Quantum Playground, we define qubits using the command `VectorSize`, which we write at the top of our code. This specifies the number of qubits we will be using, and the playground initialises these qubits all to the value $|0\rangle$ Try running the code below by typing it into the text box, pressing "Compile" and then pressing "Run":

```
VectorSize 6
```

This line initialises 6 qubits, all in the $|0\rangle$ state. We can see this on the left, where we see a black image with one white square in the bottom left hand corner. This image describes our state. Each square of this image represents a quantum state, and hovering over a square with the mouse will show us the quantum state, and numbers labelled Re and Im. The number Re is 0 for all quantum states except for the state $|0\rangle$, where Re is 1. We will see what these numbers mean in the next sections.

3 Measurement

While images like this showing off our complete quantum state are nice, we cannot look at quantum states this way in practice. Instead, we have to *measure* our qubits. We can do this using the `MeasureBit` command:

```
VectorSize 6
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
```

The second line measures qubit 0 and puts the result in the `measured_value` variable. The third line displays the result.

[1] This notation was invented by Paul Dirac, one of the best-known quantum physicists and born and raised in Bristol.

From this we get the output "Qubit 0 is $|0\rangle$". So, if we set all our qubits to $|0\rangle$ and measure one of them, we find that the measured qubit is in the $|0\rangle$ state.

4 Quantum Gates

But data is only one half of computation. We also need be able to perform operations. Classical computers do this with logic gates, such as the NOT gate, which flips a bit: $NOT(0) = 1$ and $NOT(1) = 0$.

In quantum computing, we also have logic gates, which are called quantum gates. One example is the X gate, which is the quantum equivalent of a classical NOT gate: $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. Let's apply an X gate to qubit 0 in the playground:

```
VectorSize 6
SigmaX 0
```

Now our image is black with one white square denoting an Re number of 1 for the $|1\rangle$ state. So qubit 0 has flipped from $|0\rangle$ to $|1\rangle$. We can also check by measuring:

```
VectorSize 6
SigmaX 0
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
```

Now when we measure our qubit, we find that the result is $|1\rangle$. We can also check the other direction by applying X to our qubit twice before measuring it, and find that $XX|0\rangle = X|1\rangle = |0\rangle$.

5 Superposition

Another example of a single qubit gate is the Hadamard gate, H. Let's see what this gate does to a qubit in the $|0\rangle$ state:

```
VectorSize 6
Hadamard 0
```

This now produces white squares for the states $|0\rangle$ and $|1\rangle$, both with Re numbers of $0.707107 \simeq \frac{1}{\sqrt{2}}$. Let's measure this state a few times to see what happens:

```
VectorSize 6
Hadamard 0
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
```

Running this multiple times, we see that sometimes we get $|0\rangle$ and other times we get $|1\rangle$. This is called a superposition, where measuring the qubit could give $|0\rangle$ or $|1\rangle$, and it is impossible to predict which result it will be. This is already a very useful reason for utilising quantum effects in technology, as classical computers are not good at generating numbers purely at random. And this technology is already being used for applications from security and simulations to gambling and gaming.

But how likely are we to measure $|0\rangle$ or $|1\rangle$? This is decided by the Re and Im[2] numbers we have seen. These numbers form the *amplitude*s of the state. When we measure a qubit, the probability of measuring a state with amplitude a is $|a|^2$, where $|a|$ is the absolute function[3]. In our case, the probability of measuring $|0\rangle$ and $|1\rangle$ are both $\left|\frac{1}{\sqrt{2}}\right|^2 = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$, so half the time we see a $|0\rangle$ and half the time we see a $|1\rangle$. Because the amplitude of each state is $\frac{1}{\sqrt{2}}$, we say that the state is:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

So we can generate a random bit based on the result of this state. Can we generate more random bits by measuring it multiple times? Sadly,this is not the case. This is because once we have measured a qubit in a superposition, that qubit has *collapsed* into either $|0\rangle$ or $|1\rangle$, and the randomness has now been lost. So we need to generate a new superposition for each random bit we want. Measuring a superpositon is like opening a mysterious box; we don't know what the box contains when it is closed, but once we open it we cannot unlearn what is inside.

But we have only been looking at the $|0\rangle$ state. What happens when we try applying H to a qubit in the $|1\rangle$ state?

```
VectorSize 6
SigmaX 0
Hadamard 0
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
```

This seems to produce the same result as before: The qubit is in a superposition, and when we measure it the particle collapses into $|0\rangle$ or $|1\rangle$. So does this produce the same result as $H|0\rangle$? Remove the last two lines of code and look at the image. Now the square for the $|0\rangle$ state is white with an amplitude of $\frac{1}{\sqrt{2}}$, but the $|1\rangle$ state is light blue, with an amplitude of $-\frac{1}{\sqrt{2}}$. Thus this state is:

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

This explains why the results when measuring look so similar, as the probability of getting $|0\rangle$ or $|1\rangle$ is still $\frac{1}{2}$[4].

Like the X gate, applying the Hadamard gate twice will produce the original qubit: $HH|0\rangle = H\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right) = |0\rangle$ and $HH|1\rangle = H\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) = |1\rangle$. You can check this yourself using the Quantum Playground. In fact, all quantum gates are reversible: After applying a quantum gate to a qubit, there is always another gate we can apply to get back to the original qubit. It is only after measuring that our operations cannot be reversed.

[2] For this workshop, we will only focus on the Re numbers and the Im numbers will be 0.

[3] The absolute function $|x|$ converts any negative numbers to positive numbers: $|-1| = |1| = 1$

[4] $\left|-\frac{1}{\sqrt{2}}\right|^2 = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$

You might wonder at this point what other quantum states we can possibly make with a single qubit. The answer is that we can make any qubit of the form $a|0\rangle + b|1\rangle$, where $|a|^2 + |b|^2 = 1$.

6 Multiple Qubits

Multiple qubits act similarly to multiple classical bits, where we can apply quantum gates to the individual qubits. We represent these qubits in the *tensor product* $\otimes$. For example, suppose the first qubit is $|0\rangle$ and the second is $|1\rangle$. We write this state as $|0\rangle \otimes |1\rangle$, and often shorten it to $|01\rangle$.

So for example, we can apply an X gate to the first and second qubits as follows:

```
VectorSize 6
SigmaX 0
SigmaX 1
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
MeasureBit 1
Print "Qubit 1 is |" + measured_value + ">"
```

And we find that they are in the $|11\rangle$ state, similarly to when we were acting only on individual qubits. Likewise, if we applied X to only qubit 0, then we would find that they were in the $|10\rangle$ state. So applying a quantum gate to one qubit doesn't affect the other.

If we apply a Hadamard gate to both qubits, we would find that they were in their own superpositions and would produce random measurements:

```
VectorSize 6
Hadamard 0
Hadamard 1
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
MeasureBit 1
Print "Qubit 1 is |" + measured_value + ">"
```

Like single qubits, we can produce any superposition over two qubits. Thus our state can be any form of $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ where $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. Some example states include $|01\rangle$, $\frac{|00\rangle + |10\rangle}{\sqrt{2}}$ and $\frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$.

7 Entanglement

Not all classical gates are single bit. The classical AND gate for example, takes two classical bits as input and produces one classical bit as output.

We also have multiple qubit quantum gates. One of particular importance is the Controlled-NOT gate, or $CNOT$, which takes a control qubit as the first argument and a target qubit as the second. In the quantum playground, the first argument of the $CNOT$ command is the control qubit, and the second argument is the target qubit. Let's see what happens when the control qubit is $|0\rangle$:

```
VectorSize 6
CNot 0, 1
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
MeasureBit 1
Print "Qubit 1 is |" + measured_value + ">"
```

It seems that $CNOT|00\rangle = |00\rangle$. If we were to apply an X gate to qubit 1 before the $CNOT$ we would find that $CNOT|01\rangle = |01\rangle$. So when the control qubit is $|0\rangle$, the $CNOT$ gate does nothing. What if the control qubit is $|1\rangle$?

```
VectorSize 6
SigmaX 0
CNot 0, 1
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
MeasureBit 1
Print "Qubit 1 is |" + measured_value + ">"
```

Now our target qubit has been flipped, so we can see that $CNOT|10\rangle = |11\rangle$. And again, if we were to apply an X gate to our target qubit too, we would find that $CNOT|11\rangle = |10\rangle$. So if our control qubit is $|0\rangle$, then our $CNOT$ gate does nothing, but if our control qubit is $|1\rangle$, then $CNOT$ applies an X gate to the target qubit.

But these aren't the only possible states our control qubit can take. What if our control qubit is in a superposition?

```
VectorSize 6
Hadamard 0
CNot 0, 1
MeasureBit 0
Print "Qubit 0 is |" + measured_value + ">"
MeasureBit 1
Print "Qubit 1 is |" + measured_value + ">"
```

This produces a very different result. It now appears that our measurement of qubit 0 is random, but our measurement of qubit 1 is always the same state as qubit 0. If we measure qubit 1 first, then our measurement of qubit 1 will be random, but our measurement of qubit 0 will always be the same state as qubit 1. It seems that our qubits are in a superposition together, and if one collapses then the other collapses with it. We write this state as $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$.

This effect is called *entanglement*, and is one of the strangest results of quantum mechanics. Entangled states collapse at exactly the same time, regardless of distance between the two qubits. One qubit could be at the At-Bristol Science Centre and the other could be at the Digimakers Moon Base[5], and their collapse would still be at the same time. Classically, this simultaneous collapse should be impossible, as it would require one qubit to send a message describing its measurement result to the other faster than the speed of light, breaking the laws of relativity. This led Albert Einstein to describe quantum mechanics as "Spooky action at a distance."

While we haven't been able to entangle qubits between the At-Bristol Science Centre and the Digimakers Moon Base, physicists

[5] Construction pending.

at a number of institutions – most famously Delft University of Technology – have managed to successfully entangle qubits and witness these effects before light could have travelled from one qubit to the other. So there must be something more to quantum.

8 Further Reading

I first heard about quantum computing from a booklet written by the University of Bristol's Ashley Montanaro, which explains many of the concepts discussed here from a theoretical perspective. That booklet is available here: `http://www.cs.bris.ac.uk/~montanar/gameshow.pdf`

Ashley also recently gave a guest lecture on quantum computing in general, the slides of which are available here:`http://www.maths.bris.ac.uk/~csxam/teaching/history.pdf`

Juan Miguel Arrazola at the University of Waterloo has been writing a collection of blog posts explaining quantum mechanics to everyone:

- Part one is here: `https://uwaterloo.ca/institute-for-quantum-computing/blog/post/anyone-can-understand-quantum-mechanics-part-1`
- Part two is here: `https://uwaterloo.ca/institute-for-quantum-computing/blog/post/anyone-can-understand-quantum-mechanics-part-2`
- And part three will be coming soon, so make sure to keep an eye on their blog.

Note that while these are the easier aspects of quantum mechanics to understand, a lot of quantum mechanics in general relies on topics that are taught at GCSE or A Level Mathematics, so be careful diving straight in. But if you want to try and get ahead, some very useful mathematical concepts to learn include:

- Trigonometry ($\sin, \cos$, trigonometric identities...)
- Complex numbers ($e^{i\theta} = \cos\theta + i\sin\theta$, complex conjugates...)
- Linear algebra (Matrices, vectors, eigenvalues & eigenvectors...)

All of these concepts come up on a regular basis in quantum physics, so if you want more of a reason to study these concepts, now is your chance!