IndependentStudy

Matthew Robson

September 2025

1 Classical Information and Computation

This entire section was covered by my work in Digital Electronics. If you wish to view an example of my completed work, you can access it though my Git repository.

2 One Quantum Bit

2.1 Qubit Touchdown

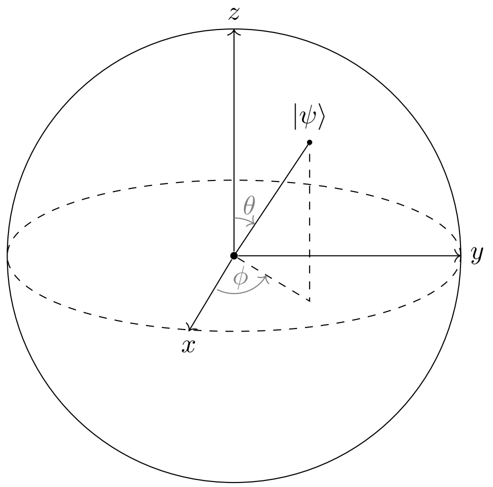
Qubit Touchdown is a game designed to introduce the player to the basics of a quantum bit.

2.2 Superposition

Qubits are represented as a super position of $|0\rangle$ and $|1\rangle$. $|0\rangle$ corresponds to the vector $(0, 0, 1)$ and $|1\rangle$ corresponds to the vector $(0, 0, -1)$. Here are the definitions of some commonly used kets:

- $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
- $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$
- $|-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle)$

Qubits can be defined by some vector $|\psi\rangle$ on the Bloch Sphere.



A common function used in quantum computing is the norm-square. The norm-square is defined as $|x|^2 = xx^*|x* = \overline{x}$.

2.3 Measurement

A qubit is most commonly measured in the z basis, as to give a $|1\rangle$ or $|0\rangle$. In the superposition $\frac{1}{\sqrt{2}}(|1\rangle + e^{\frac{i\pi}{6}}|0\rangle)$ the probability of measuring a 1 is equivalent to the norm-square of the coefficient of $|1\rangle$, as for the probability of measuring a 0. In this example, that would mean $p(|0\rangle) = |\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$, and $p(|1\rangle) = |\frac{e^{\frac{i\pi}{6}}}{\sqrt{2}}|^2 = \frac{1}{2}$. In the case where the probabilities of measuring in a basis would result in a sum of greater than one, there is a normalization constant placed in front to set the total probability to one. In our example, the $\frac{1}{\sqrt{2}}$ is the normalization constant. Measurement can be done in any basis, that is, between two positions that oppose each other on the Bloch Sphere.

2.4 Bloch Sphere Mapping

A global phase in the form of $e^{i\theta}$ may be placed in front of the superposition, but this phase will not impact the probability of measurement in any basis. Given some quantum state $|\psi\rangle$, this can be written as some $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle\, ||\alpha|^2 + |\beta|^2 = 1$. This means that α and β can be represented as cosine and sine, as in $\alpha = \cos\frac{\theta}{2}, \beta = e^{i\phi}\sin\frac{\theta}{2}$, explaining the above Bloch Sphere figure. In order to further understand this position, in some cases $|\phi\rangle$ will be represented using Cartesian coordinates (x,y,z). These coordinates are defined by the following set of equations.

- $x = \sin\theta\cos\phi$
- $y = \sin\theta\sin\phi$
- $z = \cos\theta$

As an additional note, by measuring a qubit, you collapse the state to one location, but if you were to measure the same qubit in alternating bases, you would be able to measure consecutive p(0.5) events. Ex. Alternate measuring in the $|0\rangle, |1\rangle$ basis, then in the $|+\rangle, |-\rangle$ basis.

2.5 Physical Qubits

There are many ways in which qubits are created in the real world. Some of these ways include:

- Photons
- Trapped ions
- Cold atoms
- Nuclear magnetic resonance
- Quantum dots
- Defect qubits
- Superconductors

2.6 Quantum Gates

Quantum gates are defined as linear, meaning that they will be distributed across superpositions. This can be shown as $U(\alpha|0\rangle + \beta|1\rangle) = \alpha U|0\rangle + \beta U|1\rangle$. Additionally, all quantum gates will be reversible, suggesting that all reversible classical gates can be represented as a set of quantum gates. In classical computing, there are two single bit gates, the identity gate, and the not gate, both of which can be regarded as quantum gates. The identity gate does nothing and therefore can be represented as doing nothing in a quantum computer, but the not gate is represented as the Pauli-X gate. The transformation from the Pauli-X-Gate is defined as $X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$ or more generally as a rotation 180°about the x-axis. There are also the Pauli-Y-Gate and the Pauli-Z-Gate, which transform in the same way, but as rotations in the y and z axes respectively. Additionally, there are a number of other defined gates, such as the phase gate (S), the t gate (T), and the Hadamard gate (H), all of which are rotations about different axes and for different angles. The Hadamard gate is particularly interesting, as it is a rotation about the x+z axis by 180°. We can define some general rotation gate U in terms of our previous rotation gates and some unit vector $\hat{n} = n_x\hat{x} + n_y\hat{y} + n_x\hat{z}$. This gives us the definition for U as $U = e^{i\gamma}[\cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}(n_xX + n_yY + n_zZ)]$. This means, if we are given some general rotation we can use a unit vector and its θ value to define it in our general rotation gate.

2.7 Quantum Circuits

A popular tool for drawing quantum circuits is Quirk, which can be found at https://algassert.com/quirk.

3 Linear Algebra

3.1 Quantum States

As all quantum gates are linear transformations, our entire quantum circuit can be written using the laws of Linear Algebra. For example:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

suggesting that

$$|\psi\rangle = \alpha \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \beta \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

In addition to column vectors to represent these states, we can also transpose these column vectors to row vectors by applying a *transpose*. This is shown as:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}^T = \begin{pmatrix} \alpha & \beta \end{pmatrix}$$

More commonly in quantum computing, the conjugate transpose is used, defined as:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}^\dagger = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix}$$

3.2 Inner Products

This notation gives us the tools to define $\langle\psi| = |\psi\rangle^\dagger$, or more simply, the conjugate transpose of the column vector of psi is written in row vector notation. By using our bras and kets we are now able to define inner products.

$$let\,|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, |\phi\rangle = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}, \langle\phi|\psi\rangle = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \alpha^*\gamma + \beta^*\delta$$

From this we can prove that $\langle\psi|\psi\rangle = 1$. Additionally, we can define two states to be orthonormal if when multiplied, result in zero. For example, the states $\langle 0|$ and $|1\rangle$ when multiplied will result in zero, and $\langle +|-\rangle = 0$. This property of our states is very useful, as if we want to calculate the amplitude of $|0\rangle$ in some $|\psi\rangle$ we can just multiply $|\phi\rangle$ by $|0\rangle$. This returns us just the amplitude for $|0\rangle$ because as previously defined, $\langle 0|0\rangle = 1$ and $\langle 0|1\rangle = 0$.

3.3 Quantum Gates

While states can be defined as vectors, a gate may be defined as a matrix. Simply, a two by two matrix is used to define a single qubit gate.

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

This is particularly useful because we can see how our states ($|0\rangle$ and $|1\rangle$) are just basis vectors that when multiplied by our gate will return us a single column of that gate. We can now define many of our gates as matrices.

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}$$

3.4 Outer Products

In addition to the inner products defined in the previous section, there are also outer products that can be defined as $|\phi\rangle \langle\psi|$. This results in the following:

$$|\phi\rangle \langle\psi| = \begin{pmatrix} \alpha\gamma^* & \alpha\delta^* \\ \beta\gamma^* & \beta\delta^* \end{pmatrix}$$

Additionally, we can show that based on our assertion about outer products, it must follow that for some $|\psi\rangle = \alpha |a\rangle + \beta |b\rangle$, $|a\rangle \langle a| + |b\rangle \langle b| = I$.

4 Multiple Quantum Bits

4.1 Entanglion

It so seems that there are multiple quantum computing based board games.

4.2 States and Measurement

We must now define our final form of product, the tensor product. We let $|0\rangle \otimes |0\rangle = |0\rangle |0\rangle = |00\rangle$. With two qubits, we are now given four options for our tensors. $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. We can now define a superposition of two qubits in the z-basis to be $c_0 |00\rangle + c_1 |01\rangle + c_2 |10\rangle + c_3 |11\rangle$ where the probability of measuring any one of these states is equal to the norm-square of its respective c coefficient. For a more explicit definition of the tensor product, we can view it as:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha \begin{pmatrix} \gamma \\ \delta \end{pmatrix} \\ \beta \begin{pmatrix} \gamma \\ \delta \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix}$$

This means, for our 2 qubit super position, it may be written as:

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

It is also important to note that a tensor product does not require the vectors to be in the same space as each other and so products such as $|1\rangle \otimes |1\rangle \otimes |0\rangle$ are perfectly valid. Additionally, you can have some set of c values such as

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

where you are creating what would be a contradiction in classical computing, as you are stating that $a_1 b_1 = \frac{1}{\sqrt{2}}, a_1 b_2 = 0, a_2 b_1 = 0, a_2 b_2 = \frac{1}{\sqrt{2}}$.

4.3 Entanglement

Many states are able to be factored into many separate qubits such as $(\alpha_0 |0\rangle + \beta_0 |1\rangle) \otimes (\alpha_1 |0\rangle + \beta_1 |1\rangle)$. This allows for a space complexity of $\mathcal{O}(n)$ on a classical computer (which can be effectively simulated). The issue in simulation arises for states that can not be factored. One such example was our

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

which takes up a space complexity of $\mathcal{O}(n^2)$.

4.4 Quantum Gates

Importantly, we can write multiple gates applied in succession as the tensor product between those gates. Ex:

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \otimes \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} = \begin{pmatrix} \alpha \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} & \beta \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} \\ \gamma \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} & \delta \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} \end{pmatrix} =$$

$$\begin{pmatrix} \alpha A & \alpha B & \beta A & \beta B \\ \alpha \Gamma & \alpha \Delta & \beta \Gamma & \beta \Delta \\ \gamma A & \gamma B & \delta A & \delta B \\ \gamma \Gamma & \gamma \Delta & \delta \Gamma & \delta \Delta \end{pmatrix}$$

While single qubit gates are significant, in order to form a universal gate set we require two qubit gates. For example: $CNOT\,|a_0a_1\rangle$ which applies an inverse to a_1 if $a_1 = 1$. Thive give us the following matrix to represent $CNOT_{10}$:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

which when acting upon a superposition gives us the result resembling a Toffoli classical gate.

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} \rightarrow \begin{pmatrix} c_0 \\ c_1 \\ c_3 \\ c_2 \end{pmatrix}$$

Additionally, we can define $CNOT_{01}$ by:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

The reason that CNOT is so significant is that it can be used to create superpositions. For example, $CNOT(|+\rangle \otimes |0\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle$. The use of $|\Phi^+\rangle$ suggests that there are other commonly referred to super positions that should noted. They are as follows: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle$, $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |\Phi^-\rangle$, $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = |\Psi^+\rangle$, $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = |\Psi^-\rangle$. For a more generalized version of the CNOT gate, we can use the CU gate, a gate where U is applied to a qubit if the other qubit is a one. This can be written as:

$$CU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & c \\ 0 & 0 & b & d \end{pmatrix}$$

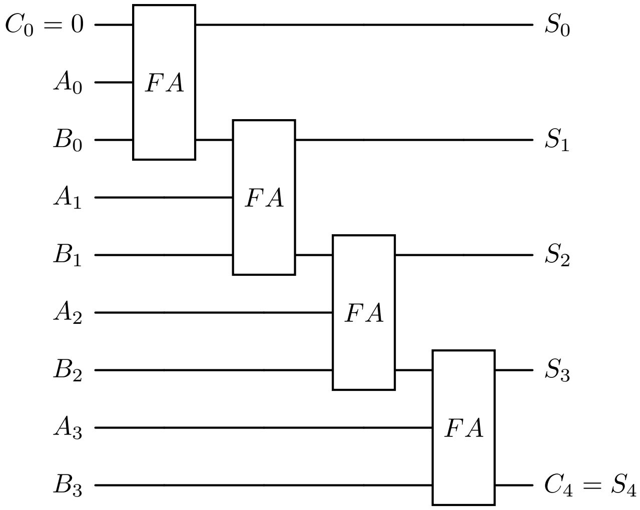
While this is useful, there are other useful two-qubit gates that this does not cover. For example, the swap gate, where two qubits are swapped. This can be given as:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Moving on from the examples of multi-qubit gates, the *No-Cloning Theorem* states that for some quantum state, the state is not able to be copied, because by copying it it would require the knowledge of the α and β values, which measurement would result in the collapsing of the qubit.

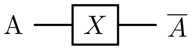
4.5 Quantum Adders

Adders in classical computing can be most simply given by:

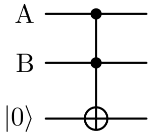


In order to construct this in a quantum circuit, we must first define how we will write classical gates with quantum gates.

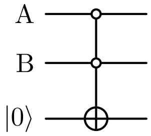
NOT:



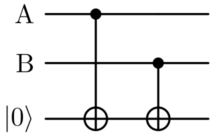
AND:



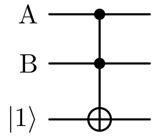
OR:



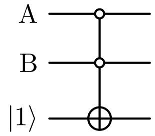
XOR:



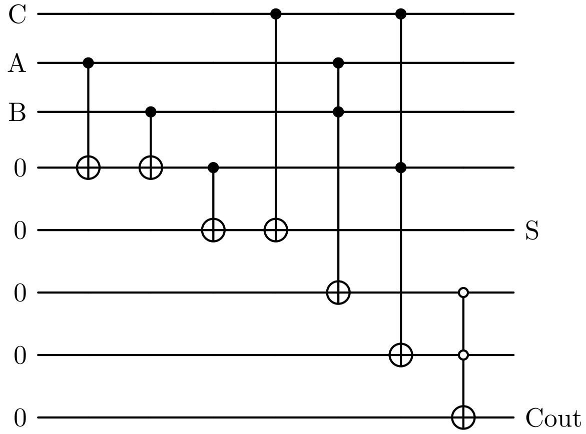
NAND:



NOR:



From these gates we are able to generate a full adder as given below.



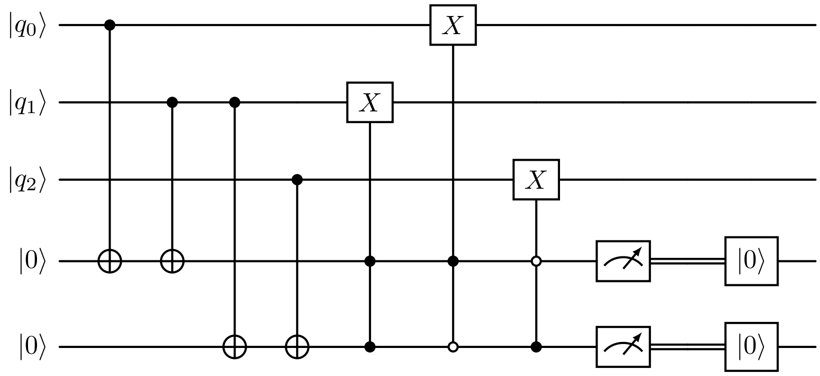
By using these full adders, we can follow essentially the same steps as classical computing but in order to regain use of our auxilirary qubits we must undo many of the steps with inverse gates.

4.6 Universal Quantum Gates

The most simple way to define the universal quantum gate set is as the CNOT gate and all sigle qubit gates. While this is simple, it may prove useful to recognize CNOT, H, and T as a set that is also universal because H and T may aproximate all other single qubit gates. In addition, the CH or controlled H gate is universal.

4.7 Quantum Error Correction

Dechoerence is the process in which a qubit is 'bumped', resulting in a change in phase (location on the bloch sphere). The ease in which a qubit can be bumped results in much more frequent bit flips in quantum computing when compared to clasical computing. In order to correct for this we will implement three qubits for every logical qubit. When a full qubit flip occurs, we can use some xors to correct for this. When a partial flip occurs (some phase shift) we can measure the xors, collapsing the error, then apply the correction xors depending on if we got a full bit flip from our measurement. This can be given by:



A similar opperation can be done to remove phase error (apply the same principle to $|-\rangle$ and $|+\rangle$). By using both of these forms of error correction we are given a new definition of our logical ψ. We can show this by: $|\psi\rangle = \frac{\alpha}{2^{3/2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) + \frac{\beta}{2^{3/2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$.

5 Quantum Programming

5.1 IBM Quantum

IBM has a quantum computing editor that gives you access to draw circuits and run them with a simulator or one of their quantum machines. Current pricing for these machines excedes $1.50 per second of use.

5.2 Quantum Assembly Language

Rather than drawing out your circuit with a mouse and clicking on gates to add them, you can be more efficient by using OpenQASM which is essentially an HDL for quantum computing.

5.3 Qiskit

Rather than using OpenQASM which appears C based in nature, you can use Qiskit which is more similar to Python's style of language. One of the benefits of using Qiskit is the ability to use Jupyter notebooks. In the use of Jupyter notebooks you can have better visuals and easier access to editing smaller sections of your entire code. Additionally, you can use any of the Python built-in packages in Qiskit.

5.4 Other Quantum Programming Languages

Ther are many other programming languages for quantum computing supported by other companies.

6 Entanglement and Quantum Protocols

6.1 Measurements

In section four we covered quantum entanglement at a high level, looking at situation where the probabilities of certain states we unable to be factored. We can now look at states and consider them to be maximally entangled if measurement of one qubit fully determines the second qubit, and partially entangled if the measurement of one qubit partially determines the second.

6.2 Bell Inequalities

Some text here

6.3 Monogamy of Entanglement

Some text here

6.4 Superdense Coding

Some text here

6.5 Quantum Teleportation

Some text here

6.6 Quantum Key Distribution

Some text here

7 Quantum Algorithms

7.1 Circuit vs Query Complexity

Some text here

7.2 Parity

Some text here

7.3 Constant vs Balanced Functions

Some text here

7.4 Secret Dot Product String

Some text here

7.5 Secret XOR Mask

Some text here

7.6 Brute-Force Searching

Some text here

7.7 Discrete Fourier Transform

Some text here

7.8 Phase / Eigenvalue Estimation

Some text here

7.9 Period of Modular Exponentiation

Some text here

7.10 Factoring

Some text here

8 Next Steps

8.1 Careers in Quantum Computing

Some text here

8.2 Technical Next Steps

Some text here

8.3 Questions

Some text here

8.4 Parting Words

Some text here