Quantum Computing: Working in the Blind

Max Festenstein

“It’s as if you were trying to do a complex jigsaw puzzle in the dark with your hands tied behind your back” [1].

Introduction

Since Turing wrote of a ‘Universal Computing Machine’ in his 1936 paper *On Computable Numbers*, technology has advanced at a rate unprecedented throughout human history [2]. This advancement can be marked as, since their conception, computers have become more powerful with every passing year. This increase itself was noted by Gordon Moore who postulated that the number of transistors you can fit into a given area will double with every other passing year, which effectively translates to a doubling of computing power over a 2 year period [3].

As transistors are physical objects, there is a lower limit to their size as, at present, there is no known way of decreasing their size smaller than atomic scales. As the size of the transistors within a computer are already only one order of magnitude larger in size than atoms, one could ask; “Where else is there to go?” [4]. If we want to find a way of improving computing technology beyond simply making computers bigger, we will need to find a new way of moving forward. One very good possibility is the quantum computer, which utilises the inherent weirdness of quantum mechanics in its operations and therefore may be able to solve problems that classical computers are simply incapable of solving. To try and understand how they work, one first needs to appreciate how even at the basic level of information, they are radically different from classical computers. So as classical computers have bits as their basis of information, quantum computers have quantum bits.

The Basic Element - Quantum Bits

The most fundamental element of a quantum computer is the quantum bit, or qubit. Qubits are objects with a wavefunction that only allows one of 2 possible observables to be measured, such as an electron that is either spin up or spin down [5]. One of the observables is assigned to correspond to a 1 and the other to a 0, much like a transistor in a classical computer being assigned to a 1 or 0 when it is on or off.

The qubit can be mathematically described as a 2-D unit vector, with the 1 and 0 states being considered orthogonal. Any linear combination of the 2 states for a single qubit is allowed as long as the wavefunction describing the qubit retains its unit magnitude. As with any other wavefunction, the square of the coefficients of the eigenstates will give the corresponding probability of observing either the 1 or 0 state when the qubit is measured. Therefore unlike a classical bit, which is either a 1 or a 0, the quantum bit can be, until measurement, both a 1 *and* a 0 [5]. A

visualisation of this is shown in figure 1 [6].

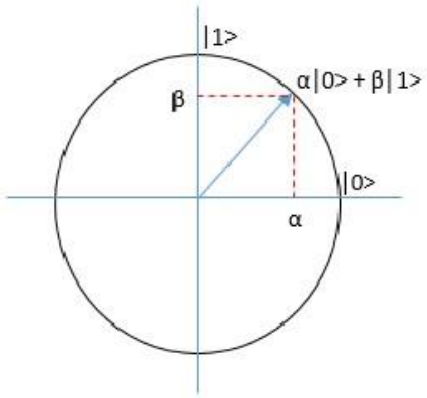


Figure 1: The wavefunction for a single qubit is displayed by the blue arrow. At any point it can be displayed as a sum of the two orthogonal 1 and 0 states, the weighting of each state given by the coefficient alpha or beta respectively.

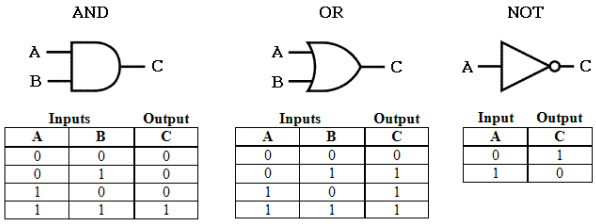
A wavefunction can be expressed as a sum of eigenstates, each of which correspond to a possible observable [7]. This means that in a system of many qubits, called a register of qubits, there will be a component with an associated probability for each possible binary string. For example, for a two qubit register, there will be eigenstates for the states 00, 10, 01 and 11.

This is where quantum computing gets its power. It relies on the fact that the qubit can exist in more than one state at once. As a qubit register is a quantum system, once a measurement of the system is made, the wavefunction will collapse into one of the eigenstates and all previous information about the wavefunction will be lost [7]. Therefore to preserve the superposition of the quantum state of the qubits, you can't make an observation of the system in the middle of a calculation. You have to let the system evolve over time to evaluate what you want it to. Only at the end of the computation can you 'take off the blindfold', make a measurement of the system and see what state the qubits are actually in. This, unsurprisingly, makes things difficult.

In a classical computer, bits interact through logic gates [8]. In a quantum computer, qubits are mathematically manipulated and interact with each other via imaginatively named quantum logic gates.

Making Things Happen - Quantum Logic Gates

Before trying to understand the quantum logic gate, it is worth briefly mentioning their classical counterpart. Logic gates perform the most basic computational processes within a computer in evaluating a Boolean function [8]. So for a given binary input or inputs, there will be a single binary output. This will depend on the type of logic gate and the inputs it is given. As there are usually not many different possible combinations of inputs, the possible inputs and their corresponding outputs are frequently given in a truth table for a logic gate. Three such cases, the AND, OR and NOT gates, are shown in figure 2 [9].



Inputs		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Input	Output
A	C
0	1
1	0

Figure 2: The graphical representations for the AND OR and NOT gates are shown with their corresponding truth tables

Quantum logic gates are markedly different in many respects. One key difference is that they have the same number of outputs as inputs [5]. Another is that, as a quantum bit can exist in a superposition of the 1 and 0 states, the quantum logic gate evaluates its respective function on the superposition of the states of a qubit rather than on either one of the two states [5]. This second point means that quantum logic gates do not constitute a measurement, but in fact act as a quantum mechanical operators. The operators that are suitable to be quantum logic gates preserve the magnitude of their operand. This means that when a quantum logic gate operates on a qubit system, the wavefunction describing the qubit system remains normalised [10].

Unlike most quantum mechanical operators, the operations performed by quantum logic gates are fixed to exist in a relatively small number of dimensions. This is because the qubits that they operate on occupy only a small Hilbert space, the number of dimensions of which is equal to the number of possible observables [5]. E.g. for a 1 qubit system, the Hilbert space will only be 2 dimensional, corresponding to the observable 1 and 0 state of the single qubit. This means that quantum logic gates (or simply quantum gates) can and frequently are represented as matrix operators acting on unit vectors corresponding to the

wavefunction of the qubit or qubits they act on [5]. This is comparable to the operators for spin eigenstates on an electron, which are also represented in matrix form [11].

In both quantum and classical computing, the operations logic gates perform vary in complexity. For example the classical exclusive OR (or XOR) which for 2 inputs only returns a 1 output if one of the inputs is a 1, is more complicated than the NOT gate, which returns its output in the opposite state as the input [8]. This idea leads to the non-trivial result that logic gates with more complex operations can be represented as a combination of logic gates with more straightforward operations. This idea can be followed through until the logic gates reach their lowest level of complexity. The most basic set of logic gates with which can be used to build all others are called ‘Universal Gates’ [12].

For the quantum case, this makes sense theoretically as for any normalised wavefunction describing a quantum mechanical state, an operator exists to change the wavefunction to any other normalised linear sum of eigenstates [13]. This leads to the fact that, that in theory, a quantum gate can be constructed to perform any unitary operation on a register of qubits. One such set of universal quantum gates to perform any

operation are: the Hadamard, controlled NOT (or cNOT) and the phase control gates [14].

The cNOT gate, as shown in figure 3, is the closest of this set to behave as a classical logic gate [15]. Unlike the classical NOT gate, the cNOT gate takes a two qubit input. These are the control bit and the target bit. The target bit is the qubit on which the NOT operation is performed. This operation simply switches the value of the target qubit from either a 1 to a 0 or from a 0 to a 1. The control bit, as the name suggests, controls whether a NOT operation is performed on the target bit. This is achieved as if the control it is set to 0, the NOT operation is not performed, and if the control bit is set to 1, the operation is performed and the state of the target bit is inverted [5]. The unusual properties of quantum logic gates begin to present themselves here as unlike all single output classical logic gates, the cNOT gate has the same number of outputs as it does inputs.

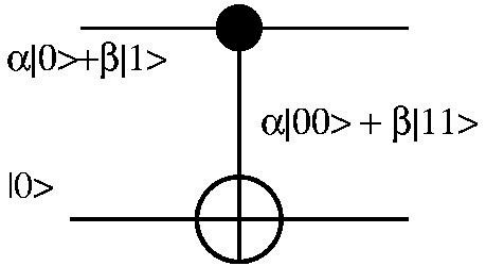


Figure 3: The control bit is shown here as a solid black circle and the target bit is the circle with a cross. As the control bit is in a superposition of the 1 and 0 state, the target bit is in a superposition of having and not having the NOT operation performed on it. α and β are the coefficients of the 0 and 1 eigenstates respectively.

These properties show themselves further in both the Hadamard and phase control gates, which rely on the quantum properties of the qubits so much that they have no classical comparator.

The Hadamard gate works to create a superposition of states for a single qubit. The superposition it creates depends on the state of the input bit. As the output of the Hadamard gate is a sum of both of the potential states of the qubit, the output is in a superposition of states [5]. The phase control gate serves to change the phase of the input qubit. This will have no effect on the overall probability of measuring a cubit in either state, however it can greatly affect the outcome of operations performed on a qubit [5]. Much like for phasor component of the simple harmonic oscillator eigenstate, the phase property of a qubit has no influence on the magnitude of the coefficient of the eigenstate, but does have an effect on some calculations performed on the wavefunction [5].

An example of this is shown in the outcomes of a Hadamard gate operating separately on the 1 and 0 states in figure 4 [16].

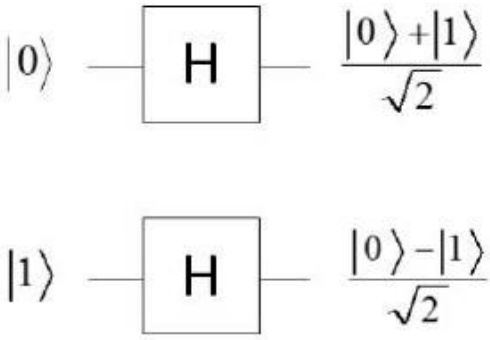


Figure 4: The superpositions of states after the hadamard gate is implemented on the eigenstates of a qubit is shown here. The sign difference between the 1 eigenstates for the two cases is due to the difference in phase between the two results.

Here, even though the probability of observing a 1 or a 0 is ½ after the Hadamard gate has operated for both cases, if the same operation was performed again, the outcomes would be drastically different [5]. This is due to the difference in sign of the coefficient of the 1 eigenstate between the 1 and 0 case.

Putting it All Together - Quantum Algorithms

Now that we've covered what qubits are and how they can be manipulated using quantum logic gates, the next thing to consider is how the quantum gates themselves may be combined to perform a useful function on a qubit input.

Again, it is the uncertain nature of the quantum state of the qubits that differentiates the quantum case from its counterpart. This results in the property of *parallelism* within the quantum computer.

Parallelism means that when a quantum algorithm acts upon a set of qubits, as long as no measurement takes place, the algorithm will be evaluated on all possible states of the qubits [5]. This means that a computation is performed on *all* potential states of the system. As the number of qubits that are operated on increases, this property allows for a vast reduction in the number of steps of computation compared to a classical algorithm.

This property arises as when a quantum mechanical operator acts on a wavefunction, it acts equally on all components of that wavefunction. Therefore, for a two qubit system, a system can be constructed with 4 potential observables: 00, 01, 10 and 11. When an operator acts on this system, the operation will be evaluated for all states.

As quantum logic gates behave as these operators and quantum algorithms are simply a combination quantum gates, a quantum algorithm will be evaluated for all four possible states. If this principle can in theory be scaled up to perform complex calculations simultaneously on a large register of qubits, quantum computers may become faster (by the number of steps needed for a calculation) at performing certain tasks than a classical computer [5]. One such case is outlined, where performing calculations on a large register of qubits simultaneously gives a large advantage over classical computing methods.

Quantum Searching - Grover's Algorithm

This algorithm would be very useful for those trying to analyse large data sets, such as personal online accounts. Say for example you were looking at a database that stored email addresses and phone numbers with entries stored in alphabetical order. This will mean there would be no order to the associated phone numbers. If you wanted to find an email address in that

list given only a phone number, using a classical algorithm you would have to look at each entry in the list and just keep going until you stumbled across the correct email address. This, on average, takes a number of steps equal to half the number of entries in the database.

The quantum search algorithm effectively works to look at every entry simultaneously. This is achieved by preparing a register of qubits such that there is one observable quantity assigned to each entry in the database, all in the 0 state. Each qubit is then passed through a Hadamard gate, putting the entire register into an equal superposition of the 1 and 0 state. A set of operations called amplitude amplifications are then performed on the entire system [17]. The properties of these operations are such to single out only the wavefunction corresponding to the desired phone number. This, on average, takes a number of iterations equal to the square root of the number of entries [17]. Therefore, for large databases, a quantum search algorithm is substantially faster than any classical search algorithm.

Conclusion

Whilst for certain applications, quantum computers can improve the efficiency for some processes and allow others to be simulated, they cannot be *complete* replacements for classical computers [5]. This is because the quantum computer is only advantageous to use when it is desired to perform the same calculation for many different cases. When it comes to performing a complicated calculation for a single or few variables, classical computers have already been developed to perform such calculations efficiently. Therefore doing the same operation with a quantum computer offers no real advantage.

The applications outlined are prevented by being actualised due to the issues associated with the practical construction of a quantum computer with a large qubit register. As of March 2016, the largest reprogrammable quantum computer consists only of 5 qubits [18]. So, despite the inherent advantages of quantum computing over classical computers for certain applications, quantum computers are prevented from eclipsing traditional computers simply due to the difficulty of scaling up the quantum computer.

In the topics covered, the basic theory behind quantum computation as well as one of their potential uses have been outlined. It is however, clear that there is still a very long way to go before they can be put to practical application.

Bibliography

[1] Pickover C.A., Quantum Computing. In: Clifford A. Pickover (ed.) *The Physics Book.* New York, New York: Sterling Publishing Co Inc; 2011. p. 474 – 475.

[2] Turing A.M. *On Computable Numbers, With an Application to the Entscheidungsproblem.* 1936. Available from: https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf [Accessed 5th January 2017]

[3] Chen J, Jia L (ed.), Liu Z (ed.), Qin Y (ed.), Zhao M (ed.), Diao L (ed.). *Proceedings of the 2013 International Conference on Electrical and Information Technologies for Rail Transportation (EITRT2013)-Volume II.* Lecture Notes in Electrical Engineering – 288. Berlin: Springer; 2014. Available from: http://link.springer.com/chapter/10.1007%2F978-3-642-53751-6_42 [Accessed 5th January 2017]

[4] Intel. *Intel 14 nm Technology.* Available from: http://www.intel.com/content/www/us/en/silicon-innovations/intel-14nmtechnology.html [Accessed 5th January 2017].

[5] Kaye P, *An Introduction to Quantum Computing.* Oxford: Oxford University Press; 2007.

[6] Dubey R, Agarwal S, Singh R. A Survey: The Next Generation of High Quantum Performance of Quantum Computing Devices. *International Journal of Scientific & Engineering Research.* 2014; Volume 5 (2): 895. Available from: http://www.ijser.org/paper/A-Survey-The-Next-Generation-Of-High-QuantumPerformance.html [Accessed 5th January 2017]

[7] Phillips C. *The Postulates of Quantum Mechanics*. Imperial College London; 2015. Available from: https://bb.imperial.ac.uk/bbcswebdav/pid-805378-dt-contentrid-2965862_1/courses/DSS-PH1_QP-15_16/The%20Postulates%282%29.pdf [Accessed 5th January 2017].

[8] Carr C. *Electronics Course Notes.* Imperial College London; 2016. Available from: https://bb.imperial.ac.uk/bbcswebdav/pid-774810-dt-content-rid-2773258_1/courses/DSS-PH1_ELEC-15_16/Electronics_Notes_2016.pdf [Accessed 5th January 2017]

[9] *Logic Gates.* Department of Electrical & Computer Engineering – Brigham Young University; 2007. Available from:

http://newstudent.groups.et.byu.net/Labs/Logic%20Gates/LogicGates.html [Accessed 5th January 2017]

[10] Muthukrishnan A. *Classical and Quantum Logic Gates; An Introduction to Quantum Computing.* Rochester Centre for Quantum Information; 1999. Available from: http://www.optics.rochester.edu/~stroud/presentations/muthukrishnan991/LogicGates.pdf [Accessed 5th January 2017]

[11] Jaffe A. *Second Year Quantum Mechanics – Lecture 25 Spin Eigenvalues and Eigenstates.* Imperial College London; 2016. Available from: https://bb.imperial.ac.uk/webapps/blackboard/execute/content/file?cmd=view&conte nt_id=_954635_1&course_id=_10694_1 [Accessed 5th January 2017]

[12] Kerntopf P, Perkowski M, Khan M.H.A. On Universality of General Reversible Multiple-Valued Logic Gates. *Journal of Multiple-Valued Logic and Soft Computing.* 2006; Volume 12 (5-6). Pages 417-429. Available from: http://ieeexplore.ieee.org/document/1319922/ [Accessed 5th January 2017]

[13] DiVincenzo D.P. *Quantum Gates and Circuits*. IBM Research Division, Thomas J. Watson Research Centre, New York: Royal Society Publishing; 1998. Available from: http://rspa.royalsocietypublishing.org/content/454/1969/261.short [Accessed 5th January 2017]

[14] Schmassmann M. *Universality of Quantum Gates*. ETH Zurich; 2007. Available from: http://qudev.phys.ethz.ch/content/courses/QSIT07/presentations/Schmassmann.pdf [Accessed 5th January 2017]

[15] Bellissard J. *Quantum Gates.* Georgia Tech; 2015. Available from: http://people.math.gatech.edu/~jeanbel/4803/Lectures/Gates/qgates.html [Accessed 5th January 2017]

[16] Ciocirlan D. *On the Adiabatic Model of Quantum Computation.* BSc Thesis. Polytechnic University of Bucharest; 2014

[17] Strubell E. *An introduction to Quantum Algorthims.* College of Information and Computer Sciences – Umass Amherst; 2011. Available from: https://people.cs.umass.edu/~strubell/doc/quantum_tutorial.pdf [Accessed 5th January 2017]

[18] Debnath S, Linke N.M, Figgatt C, Landsman K.A, Wright K, Monroe C. Demonstration of a Small Programmable Quantum Computer with Atomic Qubits. *Nature.* 2016; Volume 536 (7614). Pages 63-66. Available from: http://www.nature.com/nature/journal/v536/n7614/full/nature18648.html [Accessed 5th January 2017]