The Limitations of Quantum Computers - Decoherence in IBM Qx systems

To what extent is the effectiveness of quantum computers affected by decoherence in the form of energy relaxation and dephasing

Computer Science

Word Count: 3988

Table of Contents

Introduction

Basics of Quantum Computing

Though the idea of a quantum computer has been around for a while, actually implementing it in real life proved to be difficult until recent times. From the first creation of a quantum annealer in 2011, in merely 6 years, IBM launched the first commercially usable quantum computer. One of the main advantages of quantum computing over classical computing is in cybersecurity. I became interested in this when I started to frequently use computers in school. I was subjected to a large scale data breach that included my email, causing me to lose important information and work, but I became interested in how cyberattacks are performed and how they succeed. Current hackers mostly attack vulnerable software and access the data from within the system instead of intercepting and viewing data transmissions. As RSA (Rivest–Shamir–Adleman) encryption, used for secure data transmission, relies on the difficulty of the proposed algorithm for large integers factorization, breaking it is very difficult. Integer factorization of a 232 digit integer is estimated to take 1,500 years using a single-core 2.2GHz AMD Opteron with 2GB RAM (Kleinjung et al.). However, research on quantum computing has revealed the potential to break RSA through efficient integer factorization. Peter Shor devised an algorithm in 1994 that has proven to be implementable in quantum computing by researchers later on. With classical computing, the best algorithm would take time that increases exponentially for the number of the digits while a quantum algorithm takes time in a polynomial rate (Chuang et al.). Other than being able to decrypt RSA encryption keys with high efficiency, quantum computing can be used to speed up database search functions, run molecule simulations for chemistry and nanotechnology, and attack block ciphers (Xie and Liang 2).

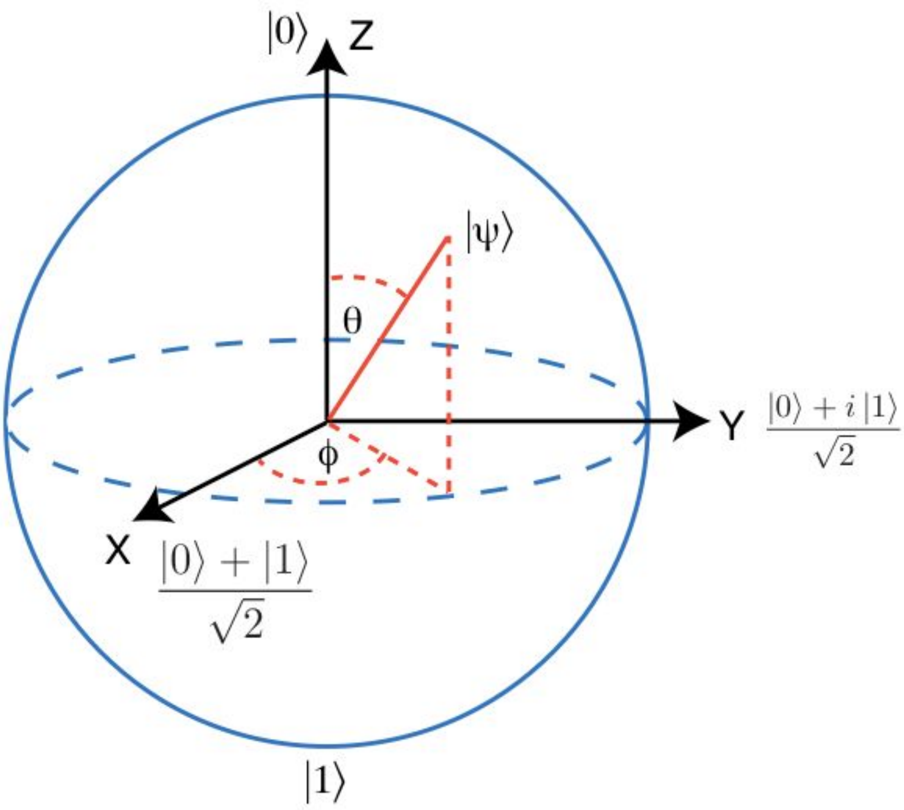
The main difference between quantum and classical computing is that the quantum computer uses qubits instead of bits. Instead of the bit being in a state of either a 0 or 1, each qubit has a probability of being either 0 or 1 (Bernhardt 121). This is then realized through physical systems such as internal states of trapped ions, electron position, or the spin of a particle (Rieffel and Polak 13). These qubits are then passed through quantum logic gates, gates very similar to classical logical gates, to build a circuit for an algorithm. The major difference of a quantum logic gate is that the process must be reversible. Thus, quantum logic gates are usually represented as unitary matrices, where a process is reversible when multiplied. Qubits can also be represented as matrices, where the number of qubits, n, can be represented as a 2^n dimensional linear vector. For simplicity, I will only write about the operations of one qubit. Any state of a qubit can be shown as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where α and β can be any number as long as $|\alpha|^2 + |\beta|^2 = 1$. One of the most common gate, the Hadamard gate (H), changes the qubit into a different quantum state. When the Hadamard gate is applied to one qubit of value |0> (Braket notation of a qubit that is equivalent to a bit with the value 0), the output becomes $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$, a superposition between |0> and |1> (Braket notation of a qubit that is equivalent to a bit with the value 1) where $\alpha = \frac{1}{\sqrt{2}}$ and $\beta = \frac{1}{\sqrt{2}}$ and probability of outputting 0 is 0.5. When the gate is applied to a |1> qubit the resultant superposition is $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ (Bernhardt 254). Through the use of various combinations of gates, the qubit can have more states and superpositions, so that a wider range of operations can be done to the qubit, allowing a variety of algorithms to be created. I will not focus too much on the maths behind quantum computing as this is a computer science extended essay, but it is important to understand that probabilities for expected results in an algorithm can be calculated for an algorithm. Another feature of the qubit is that it can perform quantum entanglement. Quantum entanglement relates two qubits on a higher scale than with classical bits. When qubits a and b are entangled, the measurement of qubit a would lead to the result of the measurement of qubit b to be the same as the result of qubit a (Rieffel and Polak 43). No matter the probability of a measuring to be a 0, if it is, b will output a 0. To achieve quantum entanglement, a pair of qubits must undergo controlled gates such as the controlled quantum NOT gate (Bernhardt 156). By using the properties of quantum entanglement, more complex algorithms with cryptography can be achieved. By understanding the basic functionality of quantum computing, we can now explore the largest limiting factor of its performance and my investigation on its effects.

Decoherence

The largest obstacle to the effectiveness of quantum computing is quantum decoherence. Quantum decoherence is the loss of quantum coherence, the phase relation between quantum particles modeled as a wave function. A qubit in a superposition between |0> and |1> has the two states, in the form of waves, interacting with each other through superposition to form a wave that contains both the two states. When this goes away, only one state remains. For instance, when a qubit is measured by the quantum computer, the coherence disappears to result in one value of 0 or 1. In other words, if there is more quantum decoherence, there is less time for the qubit to stay in a state (decoherence time) where it is able to undergo logic gates and be measured to produce an output. It is impossible to remove quantum decoherence completely, as it would be required to be isolated from everything else, including quantum logic gates and environmental factors (Rieffel and Polak 239). Factors that increase quantum decoherence and decrease quantum coherence, also known as “noise”, include quantum gates and limitations of the implementation of qubits in the specific quantum computer such as lattice vibrations in an optical lattice implementation. For a superconducting qubit, lowering the temperature of the environment to around 20 millikelvins allows for a reduction of significant quantum decoherence. The IBM quantum experience reaches down to 15 millikelvins to minimize its decoherence (“Qubits as physical system”). The decoherence time in this type of environment usually ranges from nanoseconds to seconds. Other experiments have shown that a strong magnetic field may have an effect on some systems too. Because of this limitation, time-consuming algorithms may not be executable, as the qubit will lose all its coherence by the time the algorithm finishes.

Throughout the years, decoherence time of quantum computers has increased, allowing more procedures to be performed on qubits and, thus, algorithms to take in larger inputs. There are many types of decoherence depending on the physical implementation of the quantum computer. In the IBM quantum experience system, the two main decoherence is energy relaxation (T1) and dephasing (T2). To understand these, we must first understand the mathematical implication of a qubit. As previously stated, a qubit can be in a state between 0 and 1. This is often modeled by the Bloch sphere.

Fig. 1 Bloch Sphere ("Pauli matrices and the Bloch sphere")

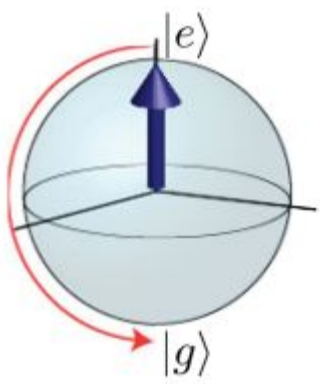


Any qubit can be represented by a positional vector on the surface of the Bloch sphere. For qubit $|\psi> = \alpha|0> + \beta|1>$, |0> and |1> is on the z-dimension, measured by the z measurement gate. If the qubit is at |0>, the measurement would output 0 and vice versa. When the qubit is in between, the measurement would either produce a 1 or a 0. The qubit at the end of the x-axis is |+>, otherwise known as $\frac{|0>+|1>}{\sqrt{2}}$. The x dimension can be measured by a Hadamard gate and a z dimension measurement. Using the z dimension measurement gate on |+> shows its probability to be 0.5 for 1 and 0.5 for 0. Shifting along the x-axis on the Bloch sphere changes the probability of being a 0 or 1. As seen in Fig. 1 the value at the x-axis is the result of a Hadamard gate applied to a |0> state. The y dimension is the complex number factor of a qubit, as α and β could be complex numbers. Any quantum gate applied onto a qubit can be shown through the Bloch sphere ("Writing down qubit states"). All gates perform a rotation on the vector so that it is in a new location. For example, a Pauli X gate acting on a single qubit rotates the vector around the x-axis 180º, mapping |0> to |1> and |1> to |0>. Quantum algorithms perform these rotatory manipulations and entanglement of qubits to produce a result. So how does decoherence affect a qubit from a point of view using a Bloch sphere?

Energy Relaxation and Dephasing

The first type of decoherence, energy relaxation (also knowns as depolarization or energy decay), is caused by the fact that superconducting qubits are energized when in |1> and not in |0>. Due to the fact that energy will dissipate, the qubit is unable to maintain at a perfect state of |1>. This causes it to shift on the Bloch sphere as shown in Fig. 2 towards |0>.

Fig. 2 - Bloch Sphere showing energy relaxation in superconducting qubits where |e> is |1> and |g> is |0> (ETH Zurich)

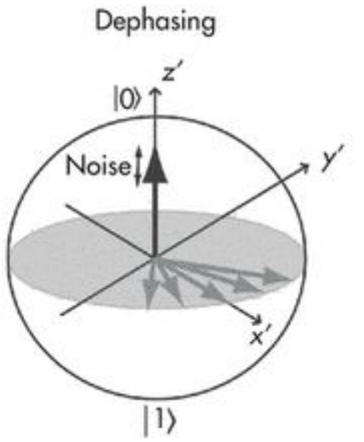


The major issue of this shift is that it distorts the qubit state as time goes by in a random manner. Although there is a trend with the time spent acting on the qubit and its energy relaxation, there is no way to return the qubit to its previous state without the collapse of the wave function. Furthermore, the decoherence of two entangled qubits will cause the qubit to not have the same states anymore, thereby changing the results of the qubits. Two ways scientists have tried to counteract decoherence, in general, is by reducing noise of surrounding (although impossible to be reduced to 0), and reducing the sensitivity of the qubit to noise around it (Yan et al.). Running an experiment many times is the easiest way to reduce the sensitivity of the data, but it is also affected by other sources of error such as gate error and T2. Another problem of T1 is that it can "fluctuate unpredictably up to an order of magnitude in time" (Klimov et al. 1). According to the IBM research website, IBM 5 Tenerife has a T1 of 46.30 microseconds, which is a short time to apply gates before it decays to |0>. Current technology allows T1 to be in ranges of 10-100 microseconds (Zhang et al.), a big difference that is hard to counteract or control.

The second type of decoherence in IBM Qx systems is T2, known as dephasing or decay time. T2 has a larger impact on the error of quantum computers as the qubit is more sensitive to the sources of dephasing. There are two ways of measuring T2: Ramsey experiment and Hahn echo method. The Ramsey method uses the Ramsey interferometry, developed by Norman Ramsey, using nuclear magnetic resonance to measure the frequency of atoms (Beaudoin). On the other hand, Hahn echo uses the spin-echo effect, sending the particle pulses of electromagnetic radiation in between varying times, and measuring the echo effect. Instead of the vector on the Bloch sphere rotating along the x-axis, dephasing causes the state shifts around the z-axis as shown in Fig. 3.

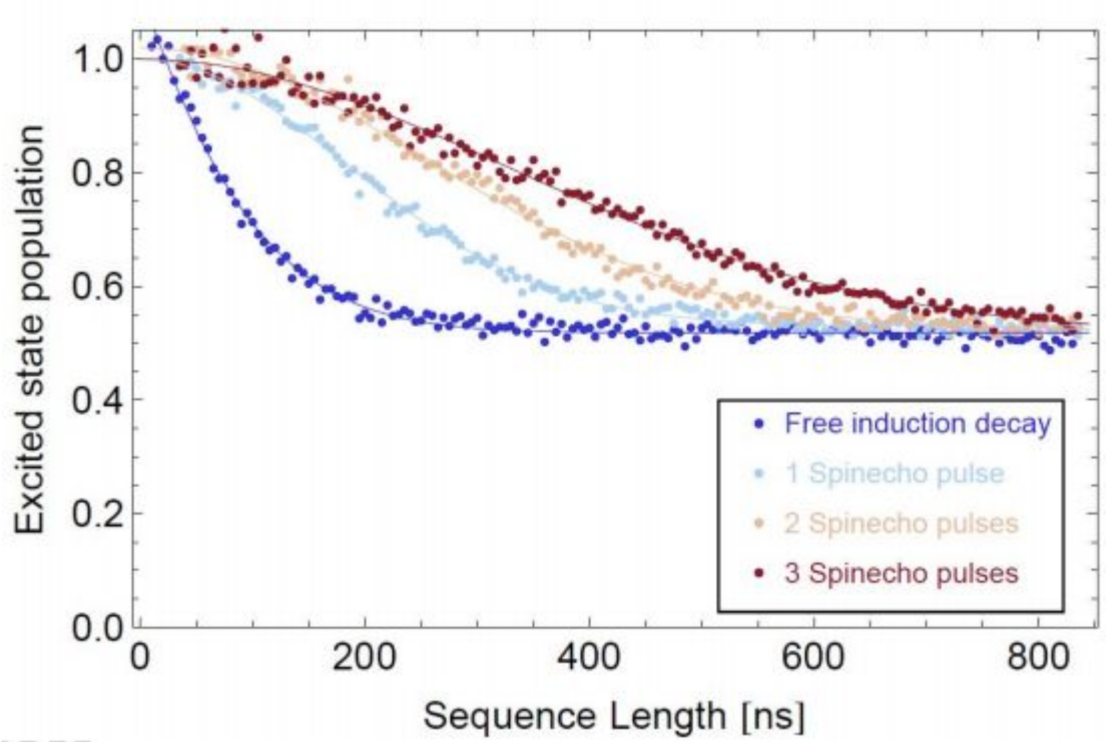
Fig. 3 - Bloch Sphere showing dephasing in superconducting qubits

where |e> is |1> and |g> is |0> (Narlikar 529)



Dephasing can be caused by magnetic flux noise, quantum coupling of systems, fluctuation in Josephson noises (in Josephson junctions in between superconducting material), and others depending on the type of quantum computer and its materials (Ithier 8). Spin echoes can, to an extent, reduce dephasing as it refocuses the quantum state with its pulses of electromagnetic radiation. As seen in Fig. 4, the spin echoes can increase the effective decoherence time of a qubit (Press 367). As the number of spin echoes increases, the rate of decrease of excited particles simulating a qubit decreases. Therefore, applying spin echoes to a qubit would reduce its rate of dephasing. According to the IBM research website, as of this moment, IBM Q 5 Tenerife has a T2 of ~15 microseconds, while IBM Q 5 Yorktown has T2 of ~44.2 microseconds ("Quantum Systems - IBM Q"). Using the IBM Q 5 Tenerife, the effects of decoherence can be investigated through quantum circuits and experimentation.

Fig.4 - Graph showing the effect of spin-echo pulses on excited state population (ETH Zurich)

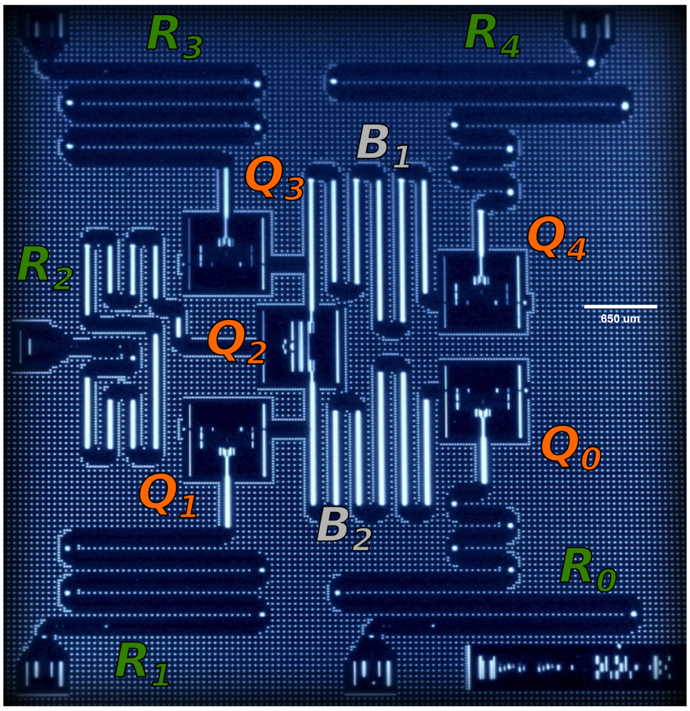


Investigation

System Details

The computer I will be using to run tests is the IBM quantum experience. This system of quantum computers, launched in 2016, is free for public use online. It uses processors made of superconducting transmon qubits. The superconducting transmon qubits are realized in superconducting electronic circuits stored in a dilution refrigerator, a refrigerator that can reach temperatures close to absolute zero. The superconducting materials in the system include niobium and aluminum. Transmon is a type of superconducting charge qubit designed to reduce noise. With a superconducting quantum computer, qubits states of |0> and |1> is manifested physically as the energy level in the circuit, where the highest energy is |1> and the grounded state is |0>. A charge qubit is a realization of qubits in charges of Cooper pairs of electrons (Gambetta et al.). Currently, there are 4 IBMQ systems: Tokyo, Melbourne, Tenerife, and Yorktown, containing 20, 14, 5, and 5 qubits respectively. For my research question, I will use the IBM Q 5 Tenerife, because I am not experimenting on a large number of qubits. In this specific quantum computer, the qubits implemented in a quantum chip, containing the qubits and their receivers (Fig. 5), are connected through "two coplanar waveguide resonators with resonances around 6.6 GHz (coupling Q2, Q3, and Q4) and 7.0 GHz (coupling Q0, Q1, and Q2)" ("IBM Q 5 Tenerife V1.x.x"). The frequency of the microwave sent to produce the qubit is 5.25 GHz. Using this machine, I will measure the effects of decoherence.

Fig. 5 Image of the qubits implemented in the IBMQX Tenerife ("IBM Q 5 Tenerife V1.x.x")



Measuring T1

The IBM Quantum experience, being a free online quantum computer, has its limits. Queues and sending information results in large delays between sending the algorithm and receiving the algorithm. Without hands-on access to the device, I cannot use the python environment Qiskit to test the runtime. However, I can observe the effects of energy relaxation on a qubit. First I will energize the qubit with Pauli Y gate, changing the qubit from |0> to |1>, and physically energizing it. I will then place varying numbers of identity gates or "wait" gates, where the output of the procedure is identical to the input. At last, I will use a z measuring gate to find out if the qubit is a 0 or 1. On the IBM Qx, the number of shots, number of times run, can be varied from 1024 to 8192, costing different amounts of credit. I will run 5 trials of 1024 shots and number of ID (identity) gates ranging from 5 to 50 in increments of 5. The circuit shown in Fig. 6 will be implemented into the IBM quantum experience through their online server and sent to IBM. Once run, it goes into the queue, and when it finishes running, the results will look as shown in Fig. 7, where the percentage above 0 is the probability of the qubit being measured as 0 and the percentage above 1 shows the probability of the qubit being a 1. This is done by dividing the number of 0s from the 1024 shots by 1024. The barrier in the circuit prevents IBMQx to optimize the circuit by removing the identity gates. The expected result is that the probability of measuring a 0 will increase as the number of ID gates increase due to energy relaxation. As time passes, noise in the system would cause the qubit's energy to dissipate to its environment, moving from an energized state of |1> to |0>. As the probability of being 0 cannot fall below 0% and the unpredictable property of the energy relaxation, there would be likely a horizontal asymptote at 0% probability.

Fig. 6 - Example circuit on IBMQx with 20 identity gates (Appendix 2)

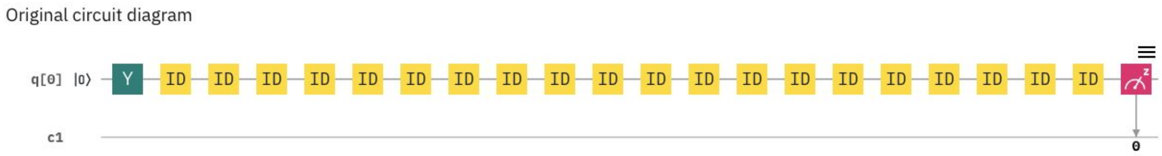
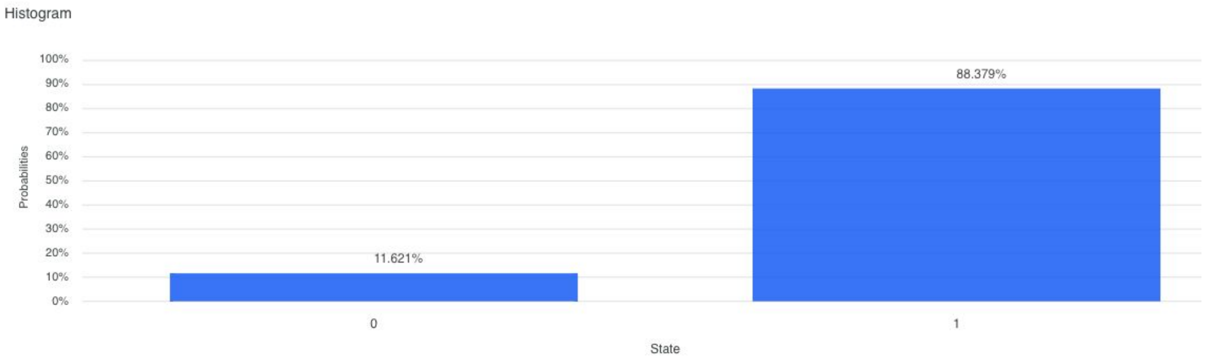


Fig. 7 - Circuit output on IBMQx of experiment with 20 identity gates (Appendix 2)



Measuring T2

The two methods of measuring T2, Ramsey and Hahn echo, can both be implemented as circuits for the IBMQx. The Ramsey method first applies a Hadamard gate to the qubit. Then, identity gates are applied to add wait time, and a x measurement (Hadamard gate + z measurement) is performed. The Hahn echo method first applies a Hadamard gate. Then half of the wait time is applied as identity gates, and a Pauli X gate is applied acting as a magnetic radiation pulse. The other half of the identity gates are added and a x measurement is performed. I have chosen to use the Hahn echo method as the Pauli X gate provides higher accuracy in measuring decoherence. I have found that the Qiskit python environment is more useful in measuring T2, as it can repeat circuits as well as plot it using Python and Matplotlib. The advantage of using Qiskit is that you can repeat the execution of circuits with different wait times in one execution. I will increase the number of ID gates from 1 to 50 in increments of 1 with 1024 shots each. The results can be read directly in Qiskit or viewed on the IBMQx website (Appendix 1). What is expected for P(1) (probability of outputting a 1) is a cosine curve shifted up ½, due to the varying shifts in complex number portion of the superposition around the vertical axis (Leek et al.). As the superposition of the qubit moves around the z-axis of the Bloch sphere, the complex component of α and β values of the qubit changes. This circular motion on the y-dimension of the Bloch sphere causes the output probability to vary between 0 and 1, starting at the excited state 1. As the probability cannot be negative, the curve will be entirely above the x-axis.

Analysis

T1 Results

The raw data for T1 is shown in Fig. 8.

Fig. 8 - Raw data for T1

Number of ID Gates	Probability of outputting 0 (%)				
	Trial 1	2	3	4	5
5	10.645	10.932	10.013	10.540	10.671
10	11.426	10.974	11.533	11.245	11.551
15	13.974	12.721	11.772	12.552	10.890
20	13.589	12.232	12.678	13.031	12.477
25	13.793	14.920	14.636	14.484	12.766
30	14.844	15.533	14.728	13.316	12.651
35	14.746	16.421	17.713	14.056	14.183
40	13.086	16.842	17.143	15.002	17.261
45	16.309	15.427	20.102	17.124	17.743
50	21.191	17.413	21.023	15.698	19.211

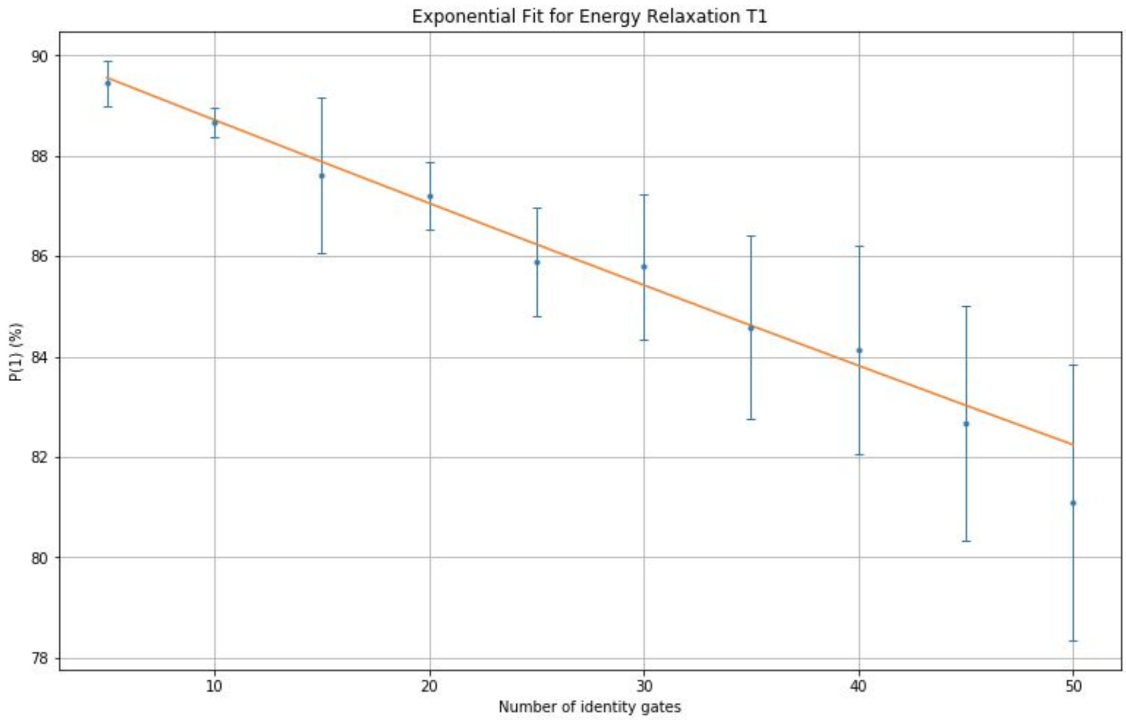
The processed data is shown in Fig. 9. The mean of the raw data was taken and calculated to be the probability of resulting in a 1 rather than a 0 by taking the difference between the value and 100 as expressed as a percentage. This is to aid visualization in comparing to other results by researchers.

Fig. 9 - Processed data for T1

Number of ID Gates	Mean of trials (%)	Uncertainty
5	89.4398	0.4595
10	88.6542	0.2885
15	87.6182	1.5420
20	87.1986	0.6785
25	85.8802	1.0770
30	85.7856	1.4410
35	84.5762	1.8285
40	84.1332	2.0875
45	82.6590	2.3375
50	81.0928	2.7465

The graph could be interpreted as a linear fit as a straight line can be drawn through all the error bars of each data point, but in reality, the probability of outputting a 1 cannot fall below 0%. Therefore, an asymptote must be formed at 0. Hence, an exponential fit decreasing would make sense. Nevertheless, it is very evident that there is an inverse correlation between waiting time and the accuracy of the z measurement of the qubit as the accuracy decreases as the wait time increases. Fig. 10 shows the exponential curve fitted to the data collected. The fitted line passes through all the error bars and points are above and below in a random pattern, indicating the line is well fit.

Fig. 10 - Exponential fit for energy relaxation



Exponential function of gates: $f(x) = 100 \times 0.9983^x - 9.5$

The y-intercept of the curve is 90.5, indicating a form of systematic error. In an isolated system, the probability of getting a 1 measurement with no wait time would be 100%, as there is no time for decoherence to take place. The reason the y-intercept is not at 100% may be because of T2. T2 exists in this experiment because it was run on a real quantum system, in which T2 affects the decoherence of the qubit. It is reasonable to think that more randomness would increase the probability of getting the "wrong" answer 0. It could also be that applying the Y-gate to the circuit causes some decoherence as it is an operation, but it would be hard to remove its effect as quantum energy relaxation is very unpredictable.

The exponential decay trend is also seen in other researchers' experiments. The calculations done by Matsuzaki and Nakano shows a similar trend in the blue line in Fig. 11. As their calculation only involves time after the qubit has been energized, the y-intercept is at

1.0. The same exponential decay trend is seen as the graph tends towards 0. The experiment done by Leek shows the same trend, plotted against the probability of outputting a 0 rather than 1 in Fig. 12. The experiment has more uncertainty and decays at a different rate than Matsuzaki's due to being a real system. Since the final trial of my experiment only resulted in ~80% of P(1), it is only the start of the graph, and more ID gates are needed to model the full curve and where its asymptote is. Unfortunately, due to constraints of the IBMQx, it is not possible for that large number of gates to be applied.

Fig 11. Population decay of a memory qubit modeled by a function of time. (Matsuzaki 5)

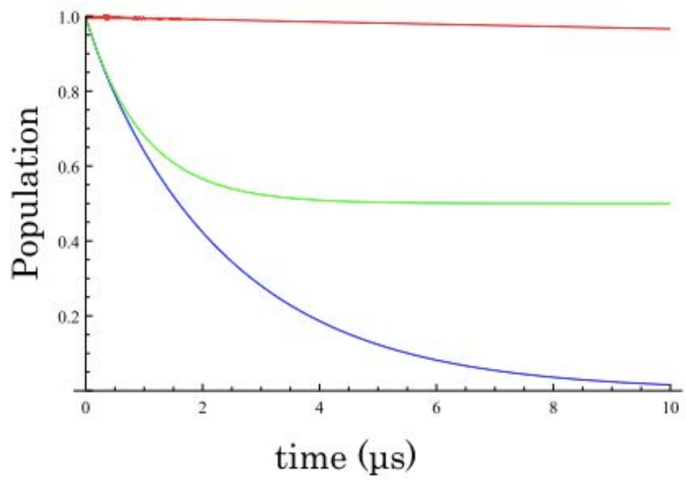
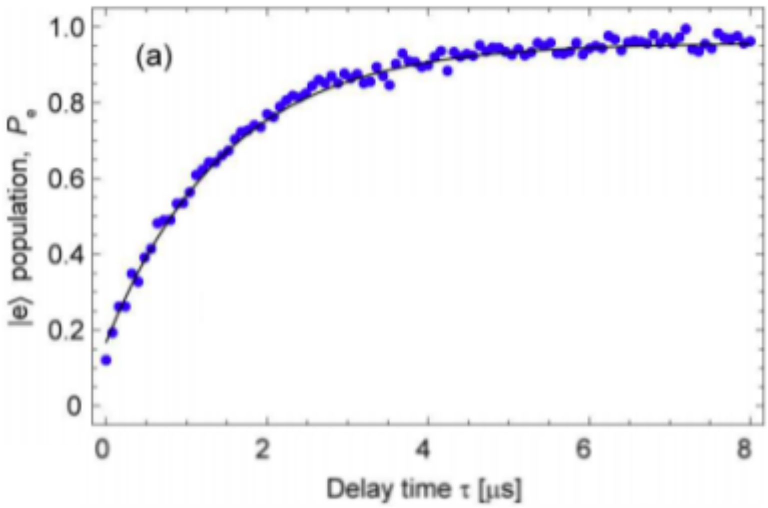


Fig. 12. Measurement of T_1 of a qubit for photon storage (Leek et al. 3)



T2 Results

Fig. 13 shows the data points and uncertainties of the data for T2. Uncertainty of probability is calculated by the Qiskit library by analyzing data of each trial.

Fig. 13 - Processed data for T2 by Qiskit

Number of ID Gates	Probability of resulting in 1	Uncertainty of Probability
1	0.9550781250	0.006472892706
2	0.8828125000	0.010051361740
3	0.7373046875	0.013753070870
4	0.5732421875	0.015456452880
5	0.4531250000	0.015556183910
6	0.3085937500	0.014434782190
7	0.2109375000	0.012749190810
8	0.1484375000	0.011110410400
9	0.1767578125	0.011920737910
10	0.2148437500	0.012834809580
11	0.3593750000	0.014994289580
12	0.4609375000	0.015577243300
13	0.6396484375	0.015003199980
14	0.7451171875	0.013618603870
15	0.8681640625	0.010572259870
16	0.9482421875	0.006923052337

17	0.9619140625	0.005981367461
18	0.9277343750	0.008091475576
19	0.8134765625	0.012172765320
20	0.7226562500	0.013990237750
21	0.5742187500	0.015451902980
22	0.4140625000	0.015392480730
23	0.2578125000	0.013669695000
24	0.1904296875	0.012270002070
25	0.1699218750	0.011736379710
26	0.1914062500	0.012294001710
27	0.2968750000	0.014277530780
28	0.3603515625	0.015003199980
29	0.5517578125	0.015541059810
30	0.6699218750	0.014695029860
31	0.8066406250	0.012341634590
32	0.9023437500	0.009276540621
33	0.9580078125	0.006267852915
34	0.9541015625	0.006539526076
35	0.8906250000	0.00975341033
36	0.7626953125	0.01329470542
37	0.6250000000	0.01512884120
38	0.4921875000	0.01562309253
39	0.3466796875	0.01487227134
40	0.2207031250	0.01296002175
41	0.1884765625	0.01222163281
42	0.1640625000	0.01157288369
43	0.1953125000	0.01238878365
44	0.3330078125	0.01472779079
45	0.4462890625	0.01553458639
46	0.6035156250	0.01528647391
47	0.7275390625	0.01391330492
48	0.8544921875	0.01101912794
49	0.9384765625	0.00750899877
50	0.9736328125	0.00500702230

Fig. 14 - Matplotlib graph of results of T2 connected

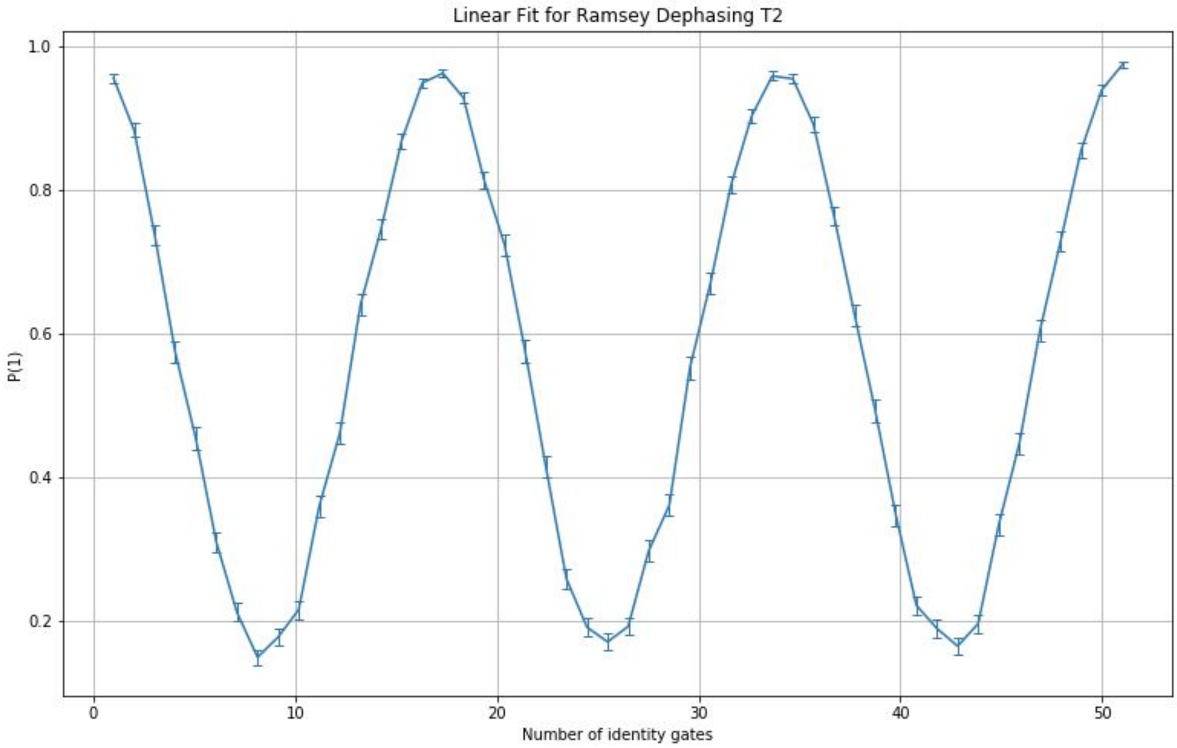
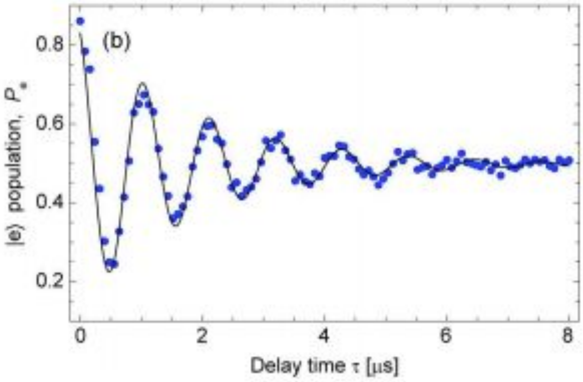


Fig. 14 uses matplotlib to plot the values along with its uncertainties. It is quite clear that the curve is a cosine curve shifted up 0.5 as expected. The expected wavelength of the cosine curve is based on what device is used to perform the calculation, so there is no expected value to compare to. However, as the error bars are small and a cosine curve can fit through it reasonably well, it can be said that the curve confirmed our expectations. There is a minimal effect of T1 on this data, since, as mentioned before, T2 contributes more to decoherence in superconducting qubits. Without the effect of decoherence, the expected value from a x measurement should be 50% of being 0 and 50% of being 1, but because of the decoherence in the complex numbers component of qubit states, it spreads around the 0.5 value rather than lean towards one option.

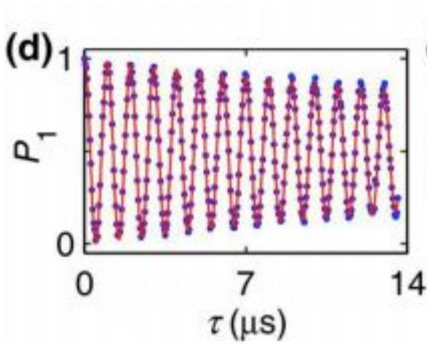
The experiment on T2 can also be compared to the research of P. J. Leek et al. In their experiment, a similar procedure was done to collect the variance of the probability of an energized quantum photon, |e>, using the Ramsey method for a certain amount of waiting time. |e> is essentially |1> in the IBM quantum systems. Fig. 15 shows their findings plotted.

Fig. 15 - Experiment on qubit calculating Ramsey T2 (Leek et al. 3)



Similar to my findings, Fig. 15 portrays a cosine graph shifted up 0.5, but unlike mine, the amplitude of the function decreases as the wait time increases. There could be many reasons for this. Perhaps the maximum identity gates tested (50) was not enough to produce a wait time that is long enough to exhibit a decrease in amplitude. Unfortunately, public access to the IBMQx is only up to 50 identity gates, but the results can show a decrease in amplitude the experiment is run with longer wait time. The most likely reason is that, because the system experimented on is different, the decoherence time is different. The result suggests that the IBMQx Tenerife has a longer decoherence time than the qubit Leek performed on, as there is less decrease in amplitude per wavelength. In the experiment done by Qiuxiang Guo (Guo et al.), there is a smaller decrease in different qubits due to the increase in decoherence time as shown in Fig. 16, indicating that the decoherence time can affect the shape of the graph.

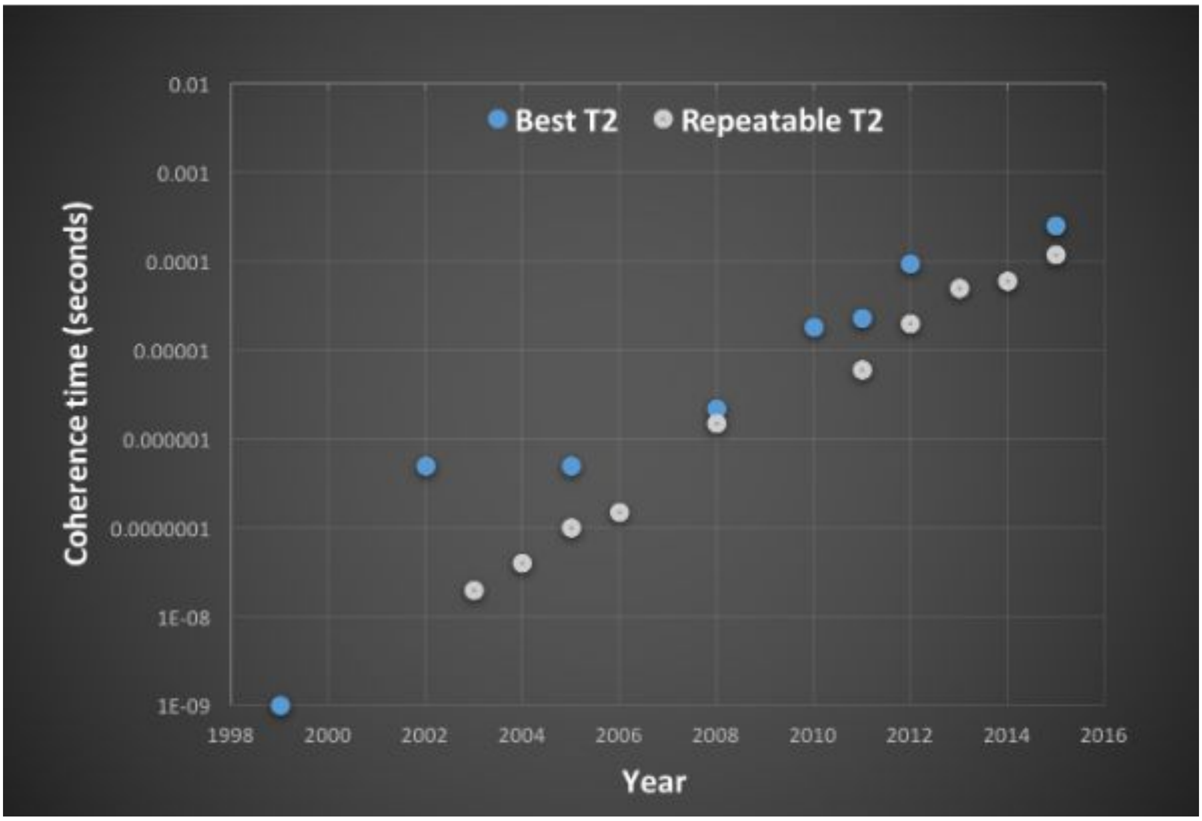
Fig. 16 Ramsey Interference Data of a qubit (Guo et al.)



From these two experiments, I have been able to confirm the effect of decoherence of a qubit in a superconducting transmon qubit system. Energy relaxation due to loss of energy to the environment is shown by the experiment on T1, where the qubit tends towards the grounded state |0> rather than being energized. Dephasing is shown by the experiment on T2, where the probability of 1 of the qubit creates a cosine curve through the Hahn echo method.

Conclusion

Fig. 17 - Improvement of Coherence times in IBMQx systems over the years (Qubits as physical systems)



Decoherence is one of the most limiting factors of quantum computers, unable to allow them to perform at their true potential. However, researchers have increased coherence time, or decreased decoherence, throughout the year as shown in Fig. 17. There have been various types of methods to increase decoherence type from using different materials, quantum error correction, and even changing the entire type of quantum computing system. Currently, the technology is not at the level yet where large integer factorization can take place to be able to break encryption systems. However, a full implementation of Shor's algorithm will allow an integer on the level of RSA-768 of a 232 digit number to require ~ 232^2 time as opposed to $e^{232^{1/3}}$ time with a classic computer (Chuang et al.) depending on the process speed. Researchers have also attempted to use dephasing to their advantage in fields such as thermometry (Razavian et al.). Though quantum computing has groundbreaking potential, major steps to reduce decoherence must be taken for its full effect to be felt.

Work Cited

Beaudoin, F., et al. "Ramsey Interferometry in Correlated Quantum Noise Environments." Physical Review A, vol. 98, no. 2, Aug. 2018, p. 020102. EBSCOhost, doi:DOI: 10.1103/PhysRevA.98.020102.

Chuang, I. L., et al. "Quantum Computers, Factoring, and Decoherence." Science, vol. 270, no. 5242, 1995, p. 1633. EBSCOhost, search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=edsjsr&AN=esjsr.2887909&site=eds-live.

Gambetta, Jay M., et al. Building Logical Qubits in a Superconducting Quantum Computing System. 2015. EBSCOhost, doi:10.1038/s41534-016-0004-0.

Guo, Qiujiang, et al. Dephasing-Insensitive Quantum Information Storage and Processing with Superconducting Qubits. 2018. EBSCOhost, doi:10.1103/PhysRevLett.121.130501.

"IBM Q 5 Tenerife V1.x.x". IBM Q 5 Tenerife backend specification V1.1.4, (2019). Retrieved from github.com/Qiskit/ibmq-device-information/tree/master/backends/tenerife/V1

Ithier, G., et al. Decoherence in a Superconducting Quantum Bit Circuit. 2005. EBSCOhost, doi:10.1103/PhysRevB.72.134519.

Klimov, P. V., et al. Fluctuations of Energy-Relaxation Times in Superconducting Qubits. 2018. EBSCOhost, doi:10.1103/PhysRevLett.121.090502.

Leek, P. J., et al. Cavity QED with Separate Photon Storage and Qubit Readout Modes. 2009. EBSCOhost, doi:10.1103/PhysRevLett.104.100504.

Matsuzaki, Yuichiro, and Hayato Nakano. Enhanced Energy Relaxation Process of Quantum Memory Coupled with a Superconducting Qubit. 2011. EBSCOhost, doi:10.1103/PhysRevB.86.184501.

Narlikar A. V. (2019). 18.2.2 Unitary Rotations and Bloch-Sphere Representation. In Oxford Handbook of Small Superconductors. Oxford University Press.

"Pauli matrices and the Bloch sphere". IBM Q Experience. quantum-computing.ibm.com/support/guides/introduction-to-quantum-circuits?page=5cae6f3935dafb4c01214bbc

Press, David, et al. "Ultrafast Optical Spin Echo in a Single Quantum Dot." Nature Photonics, vol. 4, no. 6, June 2010, pp. 367–370. EBSCOhost, doi:10.1038/nphoton.2010.83.

"QSIT11_reduced.pptx". ETH Zurich Quantum Device Lab, version 08, qudev.phys.ethz.ch/content/courses/QSIT11/QSIT11_V08_slides.pdf

"Quantum Systems - IBM Q". IBM Research. www.research.ibm.com/ibm-q/technology/devices. Accessed 17 July. 2019.

"Qubits as physical systems". IBM Q Experience. quantum-computing.ibm.com/support/guides/introduction-to-quantum-circuits?page=5cae6f9335dafb4c01214bbf

Razavian, Sholeh, et al. Quantum Thermometry by Single-Qubit Dephasing. 2018. EBSCOhost, search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=edsarx&AN=edsarx.1807.11810&site=eds-live.

Rieffel, Eleanor, and Wolfgang Polak. QUANTUM COMPUTING: A Gentle Introduction. The MIT Press, 2011.

"Writing down qubit states". IBM Q Experience. quantum-computing.ibm.com/support/guides/introduction-to-quantum-circuits?page=5cae6ee635dafb4c01214bbb#

Xie, Huiqin, and Li Yang. Using Bernstein-Vazirani Algorithm to Attack Block Ciphers. 2017. EBSCOhost, search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=edsarx&AN=edsarx.1711.00853&site=eds-live.

Yan, F., et al. The Flux Qubit Revisited to Enhance Coherence and Reproducibility. 2015. EBSCOhost, doi:10.1038/ncomms12964.

Zhang, Yu, et al. Optimizing Quantum Programs against Decoherence: Delaying Qubits into Quantum Superposition. 2019. EBSCOhost, search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=edsarx&AN=edsarx.1904.09041&site=eds-live.

Appendices:

Appendix 1 - Qiskit code to measure T2 (Taken and modified from

github.com/Qiskit/iqx-user-guide/blob/master/qiskit/python/t2_echo.py)

```
import numpy as np
import matplotlib.pyplot as plt

from qiskit import QuantumCircuit, QuantumRegister,
ClassicalRegister, execute

from qiskit import IBMQ
IBMQ.load_accounts()
print("Available backends:")
IBMQ.backends()
from qiskit.providers.ibmq import least_busy

backend = IBMQ.backends(name='ibmqx4')[0]
print("The best backend is " + backend.name())
# Define the Quantum and Classical Registers
q = QuantumRegister(1)
c = ClassicalRegister(1)

# Build the circuits
pre = QuantumCircuit(q, c)
pre.h(q)
pre.barrier()
meas_x = QuantumCircuit(q, c)
meas_x.barrier()
meas_x.h(q)
meas_x.measure(q, c)
circuits = []
exp_vector = range(1,51)
for exp_index in exp_vector:
    middle = QuantumCircuit(q, c)
    for i in range(15*exp_index):
        middle.iden(q)
    middle.x(q)
    for i in range(15*exp_index):
```

```
        middle.iden(q)
    circuits.append(pre + middle + meas_x)


# Execute the circuits
shots = 1024
job = execute(circuits, backend, shots=shots, max_credits=10)
result = job.result()

# Plot the result
exp_data = []
exp_error = []
for exp_index in exp_vector:
    data = result.get_counts(circuits[exp_index-1])
    try:
        p0 = data['0']/shots
    except KeyError:
        p0 = 0
    exp_data.append(p0)
    exp_error.append(np.sqrt(p0*(1-p0)/shots))

plt.errorbar(exp_vector, exp_data, exp_error)
plt.xlabel('time [31*gate time]')
plt.ylabel('Pr(+)')
plt.grid(True)
plt.show()
```

Appendix 2: IBM Quantum Experience Quantum Circuit code for the experiment on T1 of 20 identity gates. Code is run in on quantum-computing.ibm.com.

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[1];
creg c[1];

y q[0];
barrier q[0];
id q[0];
id q[0];
id q[0];
id q[0];
```

```
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
id q[0];
measure q[0] -> c[0];
```