Computing Science

Evaluating simulations of Quantum computing

Robin Graf

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Signed:

Date: November 30, 2017

CS4040 Report

Department of Computing Science
University of Aberdeen
King's College
Aberdeen AB24 3UE

November 2017

Evaluating simulations of Quantum computing

Robin Graf

Department of Computing Science
University of Aberdeen

November 30, 2017

Abstract: Quantum computing is often mentioned as being a thing of the future or being just a theory; it already exists. IBM have released a 5 qubit computer available on their IBM Q website and are working on a 16 qubit computer which is in beta. This paper will discuss just what can be done with what quantum computers are already available and how quantum simulation measures up compared to it. Quantum Computing might be based on physics ,but its applications and implications are related to the field of computing, as such both fields are necessary when working with Quantum computing and understanding the advantages it may hold over classical computing

1 Introduction

Quantum computers use qubits instead of classical bits; while classical bits are represented with a 1 or a 0 (i.e. on or off), a qubit is better represented as a Bloch sphere, which can have a value pointing in any direction and when measured falls to either 1 or 0, depending on where it is pointing it is more likely to fall to 1 or 0. Qubits can also be entangled, meaning that their state is dependent on another qubit and when one of the qubits is measured(collapses into a classical state) then the other also acts as if it was measured. This offers some computational advantages over classical systems.

But just how much of an advantage does quantum computing offer? How close are quantum computers to being able to compute things that could not be computed on a classical computer in a

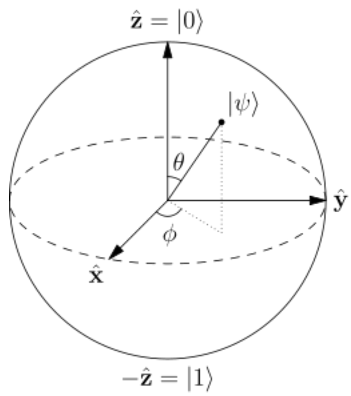

Figure 1: A Bloch sphere

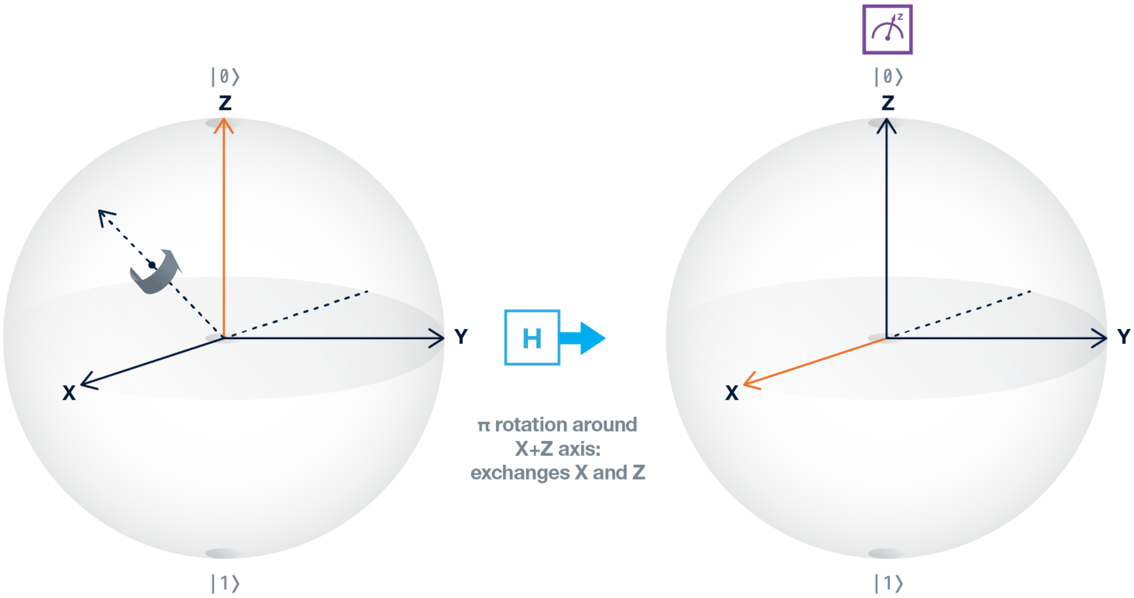

Figure 2: A applied Hadamard gate

reasonable amount of time? IBM has a 5 qubit processor available online and a 16 qubit processor in development also available online. This report will investigate the differences between IBM's actual quantum computer and a simulation of a quantum computer.

2 Background and related work

Quantum computers make use of qubits being able to be more than just 1 or 0 and to use this, additional gates are used(like OR, AND or XOR). These additional gates usually rotate the qubit's Bloch sphere; for example, the Hadamard gate rotates the qubit around the axis between z and x. This creates a state of superposition, and when measured, 50% of the time it will be 1 and 50% it will be 0 (measurement is always along the z-axis). Other gates that apply to 2 qubits rather than just one behave slightly differently. For example, the CNOT gate rotates one qubit if the control qubit is 1. This entangles the 2 qubits as one is dependent on the other. If one entangled qubit is measured, then all qubits that are entangled with it behave as if they were measured (a qubit is 1 or 0 after it is measured).

Boixo et al (2017) claim that "Quantum Supremacy" (the point at which a quantum computer could compute something a classical computer could not compute in a reasonable timeframe) would be achieved with around 50 qubits [7]. They simulate a 42 qubit quantum computer using a state of the art supercomputer. With a 7 x 6 circuit it took 1.72 seconds per gate (on average) and 989.0 seconds in total. IBM has a 5 qubit and a 16 qubit processor available online, [3] which have both been widely used to learn about quantum computing using a real quantum computer. Given it has some errors such as a result that should give a result of 50/50 percent 1 or 0 might only be 48/52; these kind of errors are due to states changing after interacting. A 50 qubit and a 20 qubit prototype have also been created, which is interesting as it would be more qubits that can be currently simulated [2].

There are also Quantum annealing machines, which do not have the same capacity as an actual quantum computer but are still very useful for many different problems such as optimisations among other things [1]. Mostly they are regarded as completely different from Quantum computers.

IBM has released an API to use their 5 qubit and 16 qubit quantum computers [4, 5], and the API also has the ability to simulate a quantum computer. I will be using the real 5 qubit ibmqx4 processor and comparing it to how well IBM's simulation of a 5 qubit processor works.

3 Research question

What is the difference between simulating a quantum computer and actually running a quantum computer? More specifically, what is the difference in the results and at what point is it faster to use a quantum computer than to simulate a quantum computer?

While the second question is more reliant on the hardware of both quantum computers and classical computers, I will focus on what exists now and speculate about what will be available soon. The first point also relies on the hardware that is available, but I will only compare the results of IBM's ibmqx4 (5 qubits) and ibmqx5 (16 qubits) and a simulation of the same hardware using an average computer [4]. Whatever the results in terms of performance, the fact that the real quantum computer has to be run on a cloud service will impact this.

Qubits have states other than 1 or 0, instead, they have; $|1\rangle$ and $|0\rangle$, but also $|+\rangle$ and $|-\rangle$, depending on where the Bloch sphere is pointing (called superposition). Another way to write this is $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ meaning the + state has 0.5 chance of being 1 and 0.5 chance of being 0, the same is true for the - state.

Qubits can also be entangled. If the first qubit is in the $|+\rangle$ state and a CNOT gate is applied to the second qubit in a 0 state, then the pair of qubits can be represented as $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. This means there is a 0.5 chance of the qubits being in state 00 and 0.5 chance of them being in state 11.

Given superposition and entanglement, I will determine whether either of these is affected differently in simulations and real quantum computing and whether that be due to random errors or some other reason. In industry, the performance of quantum computers is often measured in quantum volume, using the number of qubits and how many gates can be applied before error propagation becomes a large problem. The equation used to measure this is $QuantumVolume = Min(N, D)$ where N is the number of qubits and D is the number of gates that can be applied before decoherence is too big [6], D is determined using the rate of error.

4 Experimental Design

The API from IBM allows us to simulate up to 20 qubits, and 2^{20} is larger than 2^5 or 2^{16} instead of numbers of 32 we can work with numbers up to 2^{20}.

I will create a GHZ state and measure the result. This involves creating a state of superposition and then entangling all of the qubits; the result should be that half the time all the qubits are 1 and

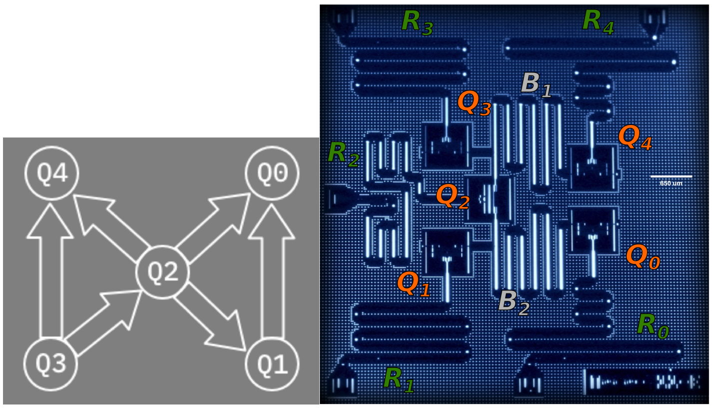

Figure 3: Topology of the ibmqx4 quantum processor

the other half 0 when measured. The layout of the real quantum computer [4] I am using is shown in figure 3. To create a GHZ state you need to put the most connected qubit into a state of superposition and then entangle all of the other qubits with the first qubit. A GHZ state is a good measure because it is common to use as the starting state for many quantum algorithms. To test the simulation I will see how long it takes to create a GHZ state on N qubits.

I will then also create a GHZ state on the ibmqx4 (5 qubits) and ibmqx5 (16 qubits) quantum computer to compare the results. However, due to the topology of the 16 qubit quantum computer (see Figure 4) I will only create a GHZ state of 8 qubits.

5 Results

5.1 Real quantum computing

IBMQX4 GHZ results:

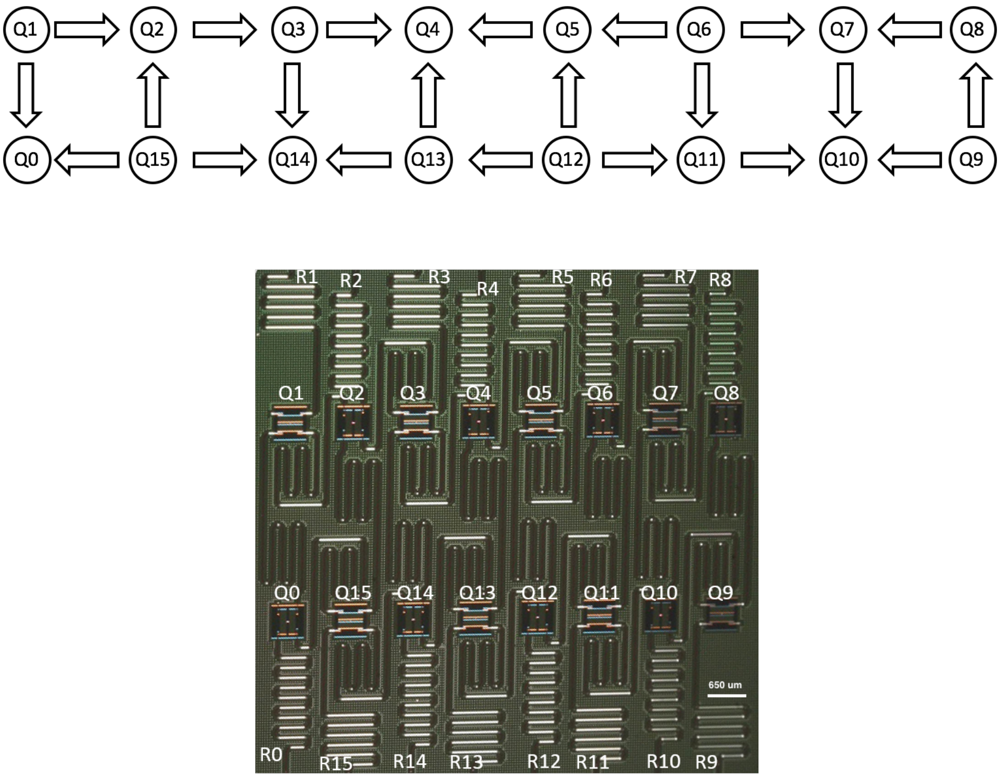

Figure 4: Topology of the ibmqx5 quantum processor

measured state	frequency
11110	36
10110	6
00110	3
11000	7
01010	2
10101	5
11010	8
10100	6
01101	5
11101	30
00000	408
00001	13
10001	2
01110	3
00111	26
00101	6
11011	68
01011	8
11100	17
01001	1
01100	1
01111	12
11001	8
00010	17
10000	9
00100	33
11111	234
10111	28
01000	6
00011	12
10011	4

As can be seen, when I ran the GHZ state on the ibmqx4 most of the results were either 00000 or 11111, which would be expected. But there were also other results; only 642/1024 were results that are correct. This means the accuracy is only 62.7%. However, because there is not 100 % accuracy that does not mean that it is not useful as error correction is used in other fields. And 62.7 % accuracy is very usable because we still know the two most frequent results and they are both the correct. But this means that if more gates are used (instead of just 2 per qubit, one for entanglement and one for measuring) then the accuracy goes down more, and so the limiting factor is not only how many qubits are available, but also how many gates can be applied until we can no longer distinguish between correct results and false ones using error correction.

IBMQX5 GHZ results:

measured state	frequency
0000000000000000	307
0100000001111111	56
other	785

(the full result of running a GHZ state on IBMQX5 is in appendix A)

The error rate of creating a GHZ state on 8 qubits using IBMQX5 is significantly larger than when running a GHZ state on IBMQX4 with 5 qubits. And only 363/1024 are correct results, and in addition, one is seen far more often than the other, when both outcomes should be seen equally (0100000001111111 and 0000000000000000 should both appear around half of the time). This leaves us with an accuracy of 35.4%, and although this can still be dealt with, it further reduces the number of gates that can be applied to each qubit while still being coherent enough to tell what the results should be. While IBMQX has 16 qubits I only used 8 of them for the GHZ state since the connections between the qubits do not allow for all of them to be entangled together. This still means we can use all 16 qubits for computing, it simply requires some additional ingenuity as all qubits are connected to their adjacent qubits. For example, you could run Shor's algorithm to factor numbers up too 2^{16}. The accuracy of a quantum computer depends on the error rates of its qubits. The more qubits, the lower the accuracy and the higher the error rate the lower the accuracy. In general, 16 qubits have more errors than 5 qubits but are still coherent enough to do a significant amount of computing.

5.2 Simulated quantum computing

number of qubits	seconds taken to simulate
3	0.263
4	0.385
5	0.585
6	1.129
7	2.256
8	4.744
9	10.354
10	22.635
11	49.535
12	106.882
13	228.107
14	507.164
15	1067.694
16	2282.014
17	4867.742
18	10351.254
19	22155.968
20	46810.004

When simulating quantum computing, there are no errors as the causes of the errors are not simulated (and so half of the results are 1111... and the other half are all 0000...). This means that when calculating the quantum volume the limiting factor is the number of qubits that can be simulated. When not using a supercomputer to do this, it takes very long to simulate a reasonable amount of qubits; when simulating 16 qubits it took over half an hour (and that's just for a simple GHZ state), at this point, it would be quicker to use an online service to run 16 real qubits instead. Of course, the upside to simulating qubits is that because there are no errors you could have any number of gates and are not limited by decoherence. The simulated qubits take approximately twice as long to simulate for each additional qubit. This means that the time taken to simulate N qubits is $O(2^N)$ and so grows exponentially.

5.3 Comparison

Having run the GHZ states on real quantum computers and on simulated quantum computers, there are some clear differences. When working with 5 qubits it seems almost always better to just simulate, as the error rate on a real quantum computer limits how many gates you can apply to the qubits, but in a simulation this is not an issue. Additionally, simulating 5 qubits is very fast (it takes less than a second on a home computer). You are also not constrained by the topology (meaning you can entangle any 2 qubits if you wish).

However, if you wish to work with numbers above 32 a real quantum computer might be preferable. Simulating a quantum computer with 16 qubits starts to become less feasible when using a home computer (over 30 minutes) while using a real quantum computer online is done in less than a minute (depending on internet speed/availability). In summary it can be seen that there are some significant differences in simulating a quantum computer and actually running a real quantum computer.

6 Discussion

The field of quantum computing is fast-growing. Real quantum computers only having been available for a small amount of time, we are already seeing 16 qubit quantum computers available online and soon 50 qubit quantum computers. At that point, Quantum Supremacy would be achieved [7]. Quantum Computing is not yet at the point where it is breaking the current encryption, but it probably will be soon. Real quantum computers can already be used by people to learn about Quantum computing in a way that was not possible before. It would seem it is in a similar state to computing when it was focused on 16 bit computers and not as many people worked with computers. Quantum computing involves much from the field of physics, but in the future that may become less and the fundamentals of how a qubit behaves might be just as much a part of computing science as different common data-structures.

With only simulation of Quantum computing, it is less plausible to learn about quantum computing as it does not allow for enough qubits to be simulated realistically on a home computer. However, a quantum computer is also not a home computer, but it is free as an online service. So the reliance is now on a connection to an online quantum computer. So considering that low numbers of qubits are easily simulated and if the goal is to work with low numbers of qubits, then the simulation is the way to progress, but if larger numbers of qubits are required then the simulation will not be enough and so a real quantum computer is needed. But if a real quantum computer is needed then at the moment only 16 qubits are available for most people.

Quantum computing is very much in its infancy. This means that it has much space to grow; 50 qubit computing is already in the works and error rates are being reduced also. Quantum computing is at an important crossroads: how will it be accepted by the computing community and how commonplace will it be? Cryptography will have to consider the capabilities of quantum computing, but will it make use of them? Quantum computing can offer lots to the field of optimisation, but how much impact will it have? There are still many questions about the field of Quantum computing (especially with how fast it is growing recently).

Further testing of real quantum computers and simulated quantum computers using algorithms other than creating a GHZ state would be useful. The limiting factor of current quantum computing is the number of gates that can be applied to a 16 qubit system while still maintaining coherence (the limits of 16 qubit computing). Another limiting factor is the number of qubits available in a system, which is currently 16 but will probably be 50 soon [2].

7 Conclusion

In this report we looked at the difference between simulated quantum computing and real quantum computing. The key differences are that simulations are useful for 0 error quantum computing with lower numbers of qubits or very custom topology. Real quantum computing is more useful if you require higher numbers of qubits (soon even the best computers will not be able to simulate the number of qubits that will be available).

References

[1] A.B. Finnila, M.A. Gomez, C. Sebenik, C. Stenson, and J.D. Doll. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics Letters*, 219(5):343 – 348, 1994.

[2] IBM. Ibm announces advances to ibm quantum systems & ecosystem. `http://www-03.ibm.com/press/us/en/pressrelease/53374.wss`, 2017.

[3] IBM. Ibm q. `https://www.research.ibm.com/ibm-q/`, 2017.

[4] IBM. Ibmqx4 quantum computer. `https://github.com/QISKit/ibmqx-backend-information/blob/master/backends/ibmqx4/README.md`, 2017.

[5] IBM. Ibmqx5 quantum computer. `https://github.com/QISKit/ibmqx-backend-information/blob/master/backends/ibmqx5/README.md`, 2017.

[6] Andrew Cross Jay M. Gambetta John Smolin Lev S. Bishop, Sergey Bravyi. `https://storageconsortium.de/content/sites/default/files/quantum-volumehp08co1vbo0cc8fr.pdf`, 2017.

[7] Vadim N. Smelyanskiy Ryan Babbush Nan Ding Zhang Jiang Michael J. Bremner John M. Martinis Hartmut Neven Sergio Boixo, Sergei V. Isakov. Characterizing quantum supremacy in near-term devices. 2016.

Appendix:

A Full results from IBMQX5

’0000000000000110’: 3, ’0000000001100011’: 2, ’0100000001011111’: 7, ’0100000001000101’: 1, ’0000000001101101’: 3, ’0000000001010110’: 1, ’0100000001001010’: 1, ’0100000000101101’: 3, ’0100000001110000’: 5, ’0100000001101001’: 3, ’0100000000001111’: 5, ’0100000001011100’: 1, ’0100000001010111’: 1, ’0100000000111101’: 2, ’0000000001101000’: 6, ’0100000000110100’: 1, ’0100000000001101’: 1, ’0000000000000001’: 24, ’0000000000000000’: 307, ’0100000000001000’: 3, ’0100000001010010’: 1, ’0100000000111000’: 1, ’0000000001101100’: 2, ’0000000001011101’: 1, ’0100000000000000’: 16, ’0100000001011000’: 2, ’0000000000000100’: 4, ’0000000000010111’: 8, ’0100000001111100’: 8, ’0000000000010100’: 1, ’0100000001110011’: 2, ’0000000001100010’: 1, ’0100000001110101’: 10, ’0100000001111010’: 7, ’0100000000111011’: 3, ’0100000001101011’:

2, ’0100000000010011’: 5, ’0000000001110100’: 1, ’0100000000010010’: 2, ’0000000000000011’: 8, ’0000000001111001’: 5, ’0000000000001011’: 1, ’0000000000010110’: 5, ’0100000000000111’: 1, ’0100000001101100’: 5, ’0100000001010101’: 2, ’0100000001101000’: 19, ’0100000000101001’: 1, ’0100000001111111’: 56, ’0100000001001001’: 1, ’0000000001111111’: 12, ’0000000000010011’: 4, ’0000000000010010’: 1, ’0100000001100100’: 1, ’0000000001101110’: 3, ’0000000001111110’: 8, ’0100000000001110’: 1, ’0100000001110111’: 14, ’0000000001011011’: 1, ’0000000000010101’: 2, ’0000000000000101’: 11, ’0000000001101111’: 1, ’0100000000011011’: 2, ’0000000001100001’: 1, ’0000000001110001’: 2, ’0100000000011100’: 1, ’0100000000011001’: 1, ’0000000000100110’: 1, ’0100000000011010’: 3, ’0100000001110110’: 7, ’0100000001110100’: 1, ’0100000000001010’: 2, ’0100000000010111’: 2, ’0100000000010110’: 3, ’0100000001111110’: 34, ’0000000001101010’: 4, ’0000000000111100’: 1, ’0100000001111101’: 16, ’0000000001000000’: 11, ’0100000000010101’: 4, ’0100000001111000’: 17, ’0100000000111111’: 11, ’0000000000011111’: 1, ’0100000001011110’: 5, ’0100000001011101’: 5, ’0100000001101111’: 14, ’0000000001101011’: 1, ’0000000001100101’: 4, ’0100000000000100’: 1, ’0100000000111010’: 1, ’0100000001010011’: 1, ’0000000000000010’: 10, ’0100000000011111’: 6, ’0000000001111101’: 7, ’0100000000001100’: 2, ’0100000000010000’: 1, ’0100000001100111’: 2, ’0100000001101101’: 2, ’0100000001100000’: 2, ’0100000001110010’: 1, ’0000000000011001’: 1, ’0000000001110011’: 1, ’0100000001101110’: 5, ’0100000000111100’: 7, ’0000000001111011’: 2, ’0000000001010000’: 2, ’0100000000101010’: 1, ’0100000001011001’: 3, ’0100000000000001’: 1, ’0100000001111001’: 14, ’0000000001100000’: 13, ’0000000001101001’: 3, ’0100000000100101’: 1, ’0000000001100111’: 2, ’0000000000100001’: 1, ’0100000001100101’: 1, ’0000000000101011’: 1, ’0000000000111111’: 2, ’0000000000000111’: 8, ’0000000000001111’: 2, ’0000000001001111’: 1, ’0100000001101010’: 4, ’0100000000101000’: 4, ’0100000000001011’: 1, ’0000000000001110’: 1, ’0000000000100000’: 13, ’0000000000011110’: 1, ’0100000001111011’: 19, ’0000000000001000’: 10, ’0000000001111010’: 2, ’0000000000111010’: 1, ’0000000001110111’: 7, ’0100000000011110’: 2, ’0000000001110110’: 2, ’0000000000110000’: 3, ’0000000001110000’: 10, ’0000000000001001’: 3, ’0000000001010111’: 1, ’0000000001011000’: 1, ’0100000000011101’: 2, ’0100000000101111’: 2, ’0100000001100010’: 1, ’0100000001011010’: 1, ’0000000001110101’: 1, ’0100000001110001’: 1, ’0000000001111000’: 6, ’0000000000010001’: 4, ’0000000000110011’: 1, ’0000000001111100’: 3, ’0100000001100110’: 2, ’0100000000111110’: 4, ’0100000000101110’: 1, ’0000000000010000’: 27, ’0100000001010100’: 1

B Full results from IBMQX4

’11110’: 36, ’10110’: 6, ’00110’: 3, ’11000’: 7, ’01010’: 2, ’10101’: 5, ’11010’: 8, ’10100’: 6, ’01101’: 5, ’11101’: 30, ’00000’: 408, ’00001’: 13, ’10001’: 2, ’01110’: 3, ’00111’: 26, ’00101’: 6, ’11011’: 68, ’01011’: 8, ’11100’: 17, ’01001’: 1, ’01100’: 1, ’01111’: 12, ’11001’: 8, ’00010’: 17, ’10000’: 9, ’00100’: 33, ’11111’: 234, ’10111’: 28, ’01000’: 6, ’00011’: 12, ’10011’: 4