The University of Edinburgh
School of Physics and Astronomy

Quantum Computing Project: Report

by
Michael Chiang
Gennaro di Pietro
William McNichols
Christoph Messmer

Edinburgh, 23rd of February 2015

Contents

1 Introduction **2**
1.1 Aims . . . 2
1.2 Background . . . 2

2 Theory **2**
2.1 Qubits . . . 2
2.1.1 Generalised bits . . . 2
2.1.2 Measurement . . . 2
2.2 Quantum register . . . 3
2.3 Quantum gates . . . 3
2.3.1 Not gate . . . 4
2.3.2 Hadamard gate . . . 4
2.3.3 Phase gate . . . 4
2.3.4 Extension to bigger registers . . . 5
2.3.5 Gate representations . . . 5
2.4 Quantum algorithms . . . 6
2.4.1 Grover's Algorithm . . . 6
2.4.2 Shor's Algorithm . . . 7
2.5 Building a quantum computer . . . 7

3 Implementation **7**
3.1 Project organisation . . . 7
3.2 Development environment . . . 7
3.3 Programme structure . . . 7

4 Results **7**
4.1 Grover's Algorithm . . . 7
4.2 Shor's Algorithm . . . 7

5 Discussion **7**
5.1 Matrix or functional representation . . . 7
5.2 Improvements and further steps . . . 7

6 Conclusion **7**

7 Appendix **7**

References **8**

1 Introduction

1.1 Aims

1.2 Background

2 Theory

In this chapter we introduce the basic concepts of quantum computation, starting off with definitions of **qubits**, **quantum registers** and the presentation of several **quantum gates**. Afterwards we will talk about two **quantum algorithms** that we implemented in our virtual quantum computer. The last chapter will briefly talk about the challenges of **building a real quantum computer**.

2.1 Qubits

2.1.1 Generalised bits

A qubit (from *qu*antum *bit*) is the smallest unit in a quantum computer and therefore the quantum mechanical **generalisation of a classical bit**, as it is used in computers nowadays. A classical bit has one and only one of the two possible states

$$|0\rangle \quad \text{or} \quad |1\rangle \tag{1}$$

at the same time, whereas a qubit is able to be in a state $|\Psi\rangle$ which is a superposition of these two classical states:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \text{where } |\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

One can depict the states via matrices with basis $(|0\rangle\,, |1\rangle)$:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad |\Psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{3}$$

2.1.2 Measurement

The superposition of states leads to a new understanding of measurement. There are two things to consider:

1. **Probabilities** P
   Given a classical state ($|\Psi\rangle$ either $|0\rangle$ or $|1\rangle$) the result of a measurement is certain (and trivial). That's no longer true for the quantum state: Given the state $|\Psi\rangle$ in Eq. 2, it is solely possible to calculate the **probabilities** P_Ψ of the outcome:

$$P_\Psi(0) = |\langle 0|\Psi\rangle|^2 = \left|\alpha \underbrace{\langle 0|0\rangle}_{=1} + \beta \underbrace{\langle 0|1\rangle}_{=0}\right|^2 = |\alpha|^2 \tag{4}$$

$$P_\Psi(1) = |\langle 1|\Psi\rangle|^2 = \left|\alpha \underbrace{\langle 1|0\rangle}_{=0} + \beta \underbrace{\langle 1|1\rangle}_{=1}\right|^2 = |\beta|^2 \tag{5}$$

2. **Collapse of $|\Psi\rangle$**
   In classical measurements it is fair to say that the measurement itself has no (noticeable) influence on the result. This is no longer true in quantum mechanics: The wave function $|\Psi\rangle_i$ **collapses after a measurement** to a projection onto the measured eigenstate and therefore becomes a different state $|\Psi\rangle_f$:

$$|\Psi\rangle_i = \alpha\,|0\rangle + \beta\,|1\rangle \quad \xrightarrow{\text{Measurement: Value } m} \quad |\Psi\rangle_f = \begin{cases} |0\rangle\,, \text{ if } m = 0 \\ |1\rangle\,, \text{ if } m = 1 \end{cases} \tag{6}$$

2.2 Quantum register

A quantum register of size N is a **collection of N qubits**. Therefore we get 2^N basic states:

$$|b_{N-1}\rangle \otimes |b_{N-2}\rangle \otimes ... \otimes |b_1\rangle \otimes |b_0\rangle \tag{7}$$

where $b_i \in \{0, 1\}$. One can interpret this chain of zeros and ones as binary code, able to store numbers in the range of $[0, 1, ...2^N - 1]$. For example for a 3-qubit system we get:

$$\begin{aligned}
|0\rangle \otimes |0\rangle \otimes |0\rangle &\equiv |000\rangle \equiv |0\rangle & \qquad |1\rangle \otimes |0\rangle \otimes |0\rangle &\equiv |100\rangle \equiv |4\rangle \\
|0\rangle \otimes |0\rangle \otimes |1\rangle &\equiv |001\rangle \equiv |1\rangle & \qquad |1\rangle \otimes |0\rangle \otimes |1\rangle &\equiv |101\rangle \equiv |5\rangle \\
|0\rangle \otimes |1\rangle \otimes |0\rangle &\equiv |010\rangle \equiv |2\rangle & \qquad |1\rangle \otimes |1\rangle \otimes |0\rangle &\equiv |110\rangle \equiv |6\rangle \\
|0\rangle \otimes |1\rangle \otimes |1\rangle &\equiv |011\rangle \equiv |3\rangle & \qquad |1\rangle \otimes |1\rangle \otimes |1\rangle &\equiv |111\rangle \equiv |7\rangle
\end{aligned}$$

We call this collection the **computational basis** of our register.

However, in contrast to a classical system, a quantum register is able to be in a **state of superposition** which turns out to be the fundamental advantage for quantum computation. If for example the second qubit is set to a superposition $|\Psi_{b_1}\rangle = \frac{1}{\sqrt{2}}\binom{+1}{-1}$ the total state of the register will be:

$$\begin{aligned}
\left|\Psi^{\text{tot}}\right\rangle &= |\Psi_{b_2}\rangle \otimes |\Psi_{b_1}\rangle \otimes |\Psi_{b_0}\rangle && (8)\\
&= |0\rangle \otimes \left[\tfrac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)\right] \otimes |1\rangle && (9)\\
&= \tfrac{1}{\sqrt{2}}\left[|001\rangle - |011\rangle\right] && (10)\\
&\equiv \tfrac{1}{\sqrt{2}}\left[|1\rangle - |3\rangle\right] && (11)
\end{aligned}$$

Eq. 11 is always reducible to a (tensor) product of three single states, as Eq. 8 suggests. This is not always the case. Consider the 2-qubit system where

$$\left|\Psi^{\text{ent}}\right\rangle = \tfrac{1}{\sqrt{2}}\left[|00\rangle + |11\rangle\right]. \tag{12}$$

There is no way to separate this wave function into a (tensor) product of two states $\{|\Psi_{b_1}\rangle, |\Psi_{b_0}\rangle\}$. Thus, the state $|\Psi^{\text{ent}}\rangle$ is called **entangled**.

2.3 Quantum gates

After defining the quantum register, we now want to process it through a number of so-called quantum gates. These are (mathematically spoken) **unitary operations** applied to our register in order to change its total state $|\Psi^{\text{tot}}\rangle$. Since we work in the Hilbert space, all our operations are **linear**, therefore we can represent any gate working on a N-qubit register by a $2^N \times 2^N$ matrix.

In the following subsections we will first introduce the most important gates used in our project, and then speak about generalisations of these gates for bigger registers.

2.3.1 Not gate

The first example for a simple 1-qubit gate is the Not-gate. It simply maps the state $|0\rangle \to |1\rangle$ and vice versa and is therefore equivalent to a logic NOT. The representing matrix in the computational basis $\{|0\rangle, |1\rangle\}$ is:

$$G^{\text{not}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{13}$$

$$G^{\text{not}} |0\rangle = |1\rangle \tag{14}$$

$$G^{\text{not}} |1\rangle = |0\rangle \tag{15}$$

However, this gate exists in exactly the same form for classical computation.

2.3.2 Hadamard gate

A common gate in quantum computation is the Hadamard gate. It performs the Hadamard transformation on a single qubit system in the following way:

$$G^{\text{H}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{16}$$

$$G^{\text{H}} |0\rangle = \tfrac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{17}$$

$$G^{\text{H}} |1\rangle = \tfrac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{18}$$

The Hadamard gate is a 'real' quantum gate since it is able to set the state to a superposition of basic states.

For a N-qubit system it is necessary to define on which qubit a gate is acting. In the following we will use the subscript to depict this: The gate G_n is acting on the n-th qubit.

Note that a combination of Hadamard gates acting on every single qubit in a N-qubit register with initial state $|\Psi\rangle = |00...0\rangle$ will lead to an **uniform superposition** of all basic states:

$$\left(\prod_{n=0}^{N-1} G_n^{\text{H}} \right) |\Psi\rangle = G_{N-1}^{\text{H}} \underbrace{|b_{N-1}\rangle}_{=|0\rangle} \otimes ... \otimes G_0^{\text{H}} \underbrace{|b_0\rangle}_{=|0\rangle} \tag{19}$$

$$= \tfrac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes ... \otimes \tfrac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{20}$$

$$= 2^{-\frac{N}{2}} \sum_{n=0}^{2^N - 1} |n\rangle \tag{21}$$

2.3.3 Phase gate

The Hadamard gate already uses special properties of quantum computation, but all operations are part of the real subspace of the Hilbert space. In general the gates and quantum register can operate on a complex vector space. The phase gate is such a gate which is defined for a single qubit system as:

$$G^{\phi} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \tag{22}$$

$$G^{\phi} |0\rangle = |0\rangle \tag{23}$$

$$G^{\phi} |1\rangle = e^{i\phi} |1\rangle \tag{24}$$

2.3.4 Extension to bigger registers

As roughly mentioned before, any single qubit gate can by applied to the n-th qubit of a N-qubit register. Consider for example an arbitrary gate G_1 in a 2-qubit system. The resulting matrix G_{tot} is:

$$G_{\text{tot}} = G_1 \otimes \underbrace{G_0}_{=\mathbb{I}_2} \tag{25}$$

$$= \begin{pmatrix} g_{00} & g_{01} \\ g_{10} & g_{11} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{26}$$

$$= \begin{pmatrix} g_{00} \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) & g_{01} \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \\ g_{10} \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) & g_{11} \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \end{pmatrix} \tag{27}$$

$$= \begin{pmatrix} g_{00} & 0 & g_{01} & 0 \\ 0 & g_{00} & 0 & g_{01} \\ g_{10} & 0 & g_{11} & 0 \\ 0 & g_{10} & 0 & g_{11} \end{pmatrix} \tag{28}$$

The general expression for N-qubit systems is analogous:

$$G_{\text{tot}} = G_{N-1} \otimes ... \otimes G_n \otimes ... \otimes G_0 \tag{29}$$

$$= \mathbb{I}_2 \otimes ... \otimes \left(\begin{smallmatrix} g_{00} & g_{01} \\ g_{10} & g_{11} \end{smallmatrix}\right) \otimes ... \otimes \mathbb{I}_2 \tag{30}$$

2.3.5 Gate representations

The resulting matrix G_{tot} is in general a $2^N \times 2^N$-matrix with two and only two non-zero entries in very row and column. That means that only $2 \cdot 2^N$ of 2^{N^2} entries are non-zero. The ratio is 2^{1-N} which rapidly goes towards zero for big N. Thus, most entries of the gate matrices will be zero.

This leads to the consideration of the following three possible representations:

1. **Dense matrix representation**
   A dense matrix representation is the standard representation, storing every single entry of a matrix. This leads to two major disadvantages in our special case:

   (a) *Memory*
   A classical computer reserves a certain amount of memory for every entry of a conventional matrix, no matter if the value is zero or non-zero. Therefore using dense matrices for big quantum registers might cause a serious lack of working memory.

   (b) *Calculation*
   We get a lot of trivial (and unnecessary) calculations using the standard matrix multiplication rule.

   The quantum register, however, has mostly non-zero entries for most of the steps of the usual algorithms.

2. **Sparse matrix representation**

3. **Functional representation**

This leads to two major disadvantages:

2.4 Quantum algorithms

2.4.1 Grover's Algorithm

Grover's algorithm is a quantum search algorithm. To search through a non-ordered list with N entries on a classical computer, one would have to go through individual entries of the list, requiring $O(N)$ operations. This can be sped up to $O(\sqrt{N})$ using a quantum circuit implementing Grover's algorithm.

Suppose we are searching through a list of N elements, and an index has been assigned to each of the elements ranging . Assume there are M solutions The key object in Grover's algorithm is the quantum *oracle*, which, in the abstract sense, has the ability to *recognise* the so

$$|x\rangle \xrightarrow{O} (-1)^{f(x)} |x\rangle \tag{31}$$

The procedure of Grover's algorithm is as follows:

1. Apply the oracle O
2. Apply the Hadamard operation on all qubits $H^{\otimes n}$
3. Perform a phase shift of -1 on every state of the computational basis except $|0\rangle$:

$$|x\rangle \quad \longrightarrow -(-1)^{\delta_{x0}} |x\rangle \tag{32}$$

4. Apply the Hadamard operation on all qubits again $H^{\otimes n}$

Step 2 to step 4 is also known as the *inversion about mean* operation. Mathematically, this iteration can be written compactly as:

$$G = (2 |\psi\rangle \langle\psi| - \mathbb{I}) O. \tag{33}$$

The Grover iteration is to maximise the probability of observing the solution states. Geometrically, the iteration performs a rotation of the state vector $|\Psi\rangle$ towards the vector representing a uniform superposition of the solution states on the plane spanned by the

2.4.2 Shor's Algorithm

2.5 Building a quantum computer

3 Implementation

3.1 Project organisation

3.2 Development environment

3.3 Programme structure

4 Results

4.1 Grover's Algorithm

4.2 Shor's Algorithm

5 Discussion

5.1 Matrix or functional representation

5.2 Improvements and further steps

6 Conclusion

7 Appendix

References

[1] Ekert, Artur ; Hayden, Patrick ; Inamori, Hitoshi: *Basic concepts in quantum computation*, 2008

[2] Perry, Riley T.: *Quantum Computing from the Ground Up*. Singapur : World Scientific, 2012. – ISBN 978–9–814–41211–7