Chapter 2: Multiple Quantum Bits

Jessica John Britto
Indian Institute of Technology Kharagpur

June 23, 2022

Concepts covered in this chapter are - *States and Measurement, Quantum Gates, Quantum Adders, Universal Quantum Gates, Quantum Error Correction.*

1 Introduction

In the *little endian* convention, the qubits are labelled from right-to-left where the rightmost qubit is the zeroth qubit. The opposite of this convention is *big endian* and this convention is followed generally. A single qubit can be written in this parametrized form - $\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$ where $\theta\epsilon[0,\pi]$ and $\phi\epsilon[0,2\pi]$. Clearly, four complex terms are needed to represent qubit however, taking out a global phase from the qubit's representation equation, we have two imaginary coefficients and one real coefficient. Hence, a single qubit can be presented on a bloch sphere. For multi-qubit systems, bloch sphere cannot be used for their representation since there will be many parameters in the general equation of the multi-qubit systems.

Tensor/Kronecker product is valid for both the *kets* and *bras*. Tensor product is obtained by multiplying each term of the first matrix/vector by the entire second matrix/vector. For instance, in case of two qubits, its done the following way for *kets*-

$$|00\rangle = |0\rangle|0\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\begin{pmatrix}1\\0\end{pmatrix}\\0\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

And with the bras, it is done the following way-

$$\langle 00| = \langle 0| \otimes \langle 0| = (1\ 0) \otimes (1\ 0) = (1\,(1\ 0)\ 0\,(1\ 0)) = (1\ 0\ 0\ 0)$$

The inner product of $|01\rangle$ and $\langle 00|$ is done by matching up the qubits, for example,

$$\langle 00|01\rangle = \langle 0|0\rangle \cdot \langle 0|1\rangle = 1 \cdot 0 = 0$$

Single qubit gates are applied to a single qubit only and they are the hadamard, identity, X-, Y- and Z-, S- and T- gates and their properties in detail can be found here. Two qubit quantum states include SWAP gate, CNOT gate - is a quantum XOR, i.e, $CNOT\,|a\rangle\,|b\rangle = |a\rangle\,|a \oplus b\rangle$. Three-qubit gates such as Toffoli gate - $|a\rangle\,|b\rangle\,|c\rangle = |a\rangle\,|b\rangle\,|ab \oplus c\rangle$. Any one-qubit quantum state can be expressed in the following way -

$$U = e^{i\gamma}(\cos\frac{\theta}{2} - i\sin\frac{\theta}{2}(n_xX + n_yY + n_zZ))$$

Quantum gates are linear maps that keep the total probability equal to 1 and are unitary matrices. Classical reversible logic gates can be quantum gates. Single-qubit quantum gates are associated with rotations by some angle about some axis on the bloch sphere. Bloch sphere corresponds to spherical coordinate system.

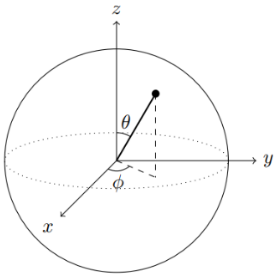
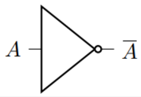
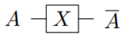
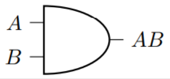
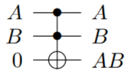
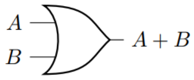
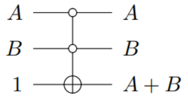
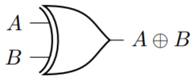
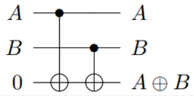
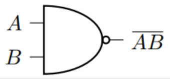
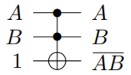
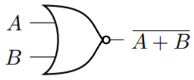
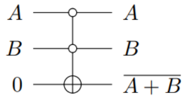


Figure 1: Bloch Sphere

Global phases are physically irrelevant since a qubit is a point on the bloch sphere and changing the global phases will point to the same qubit and does not affect the probability amplitude of each outcome. Relative phases are significant since these correspond to different points on the Bloch sphere.

Any classical gate can be made reversible by XORing the output with a third input, which is generally initialised to zero. Following table shows the list of classical gates with its corresponding quantum gates.

Classical		Reversible/Quantum	
NOT		X-Gate	
AND		Toffoli	
OR		anti-Toffoli	
XOR		CNOTs	
NAND		Toffoli	
NOR		anti-Toffoli	

2 Quantum Adders

For quantum sum, we need two CNOT gates to get the output as $|c_i\rangle |a_i\rangle |b_i\rangle = |c_i\rangle |a_i\rangle |a_i \oplus b_i \oplus c_i\rangle$, where c_i is the ancilla qubit, a_i and b_i are the input qubits. This is shown in the circuit diagram below -

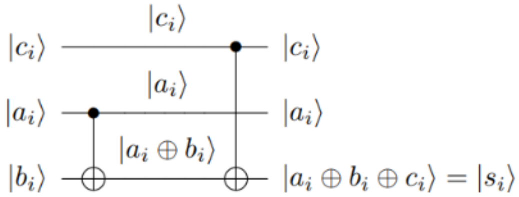


Figure 2: Circuit Diagram of Quantum Sum

For quantum carry, we need two Toffoli gates and one CNOT gate. Its circut diagram is shown below, where $|c_{i+1}\rangle$ is the carry-bit, c_i is the ancilla qubit, a_i and b_i are the input qubits.

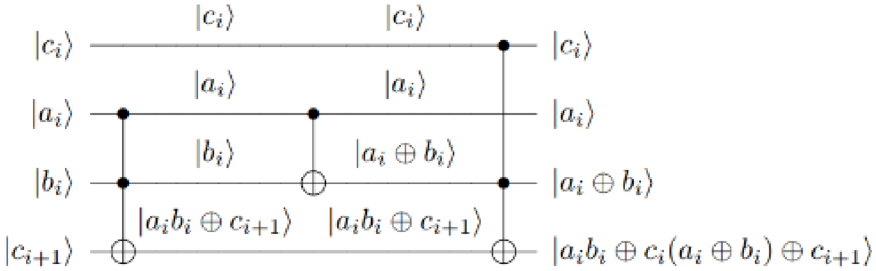


Figure 3: Circuit Diagram of Quantum Carry

2.1 Quantum Ripple-Carry Adders

Generally, a ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascaded, with the carry output from each full adder connected to the carry input of the next full adder in the chain.

Similarly, a quantum ripple-carry adder is used to find the sum of the values of the binary strings encoded in two quantum states. Let us take an example of two quantum states and their encoded values are 4-bit binary strings, and let the two states be $|0101\rangle$ and $|0001\rangle$.

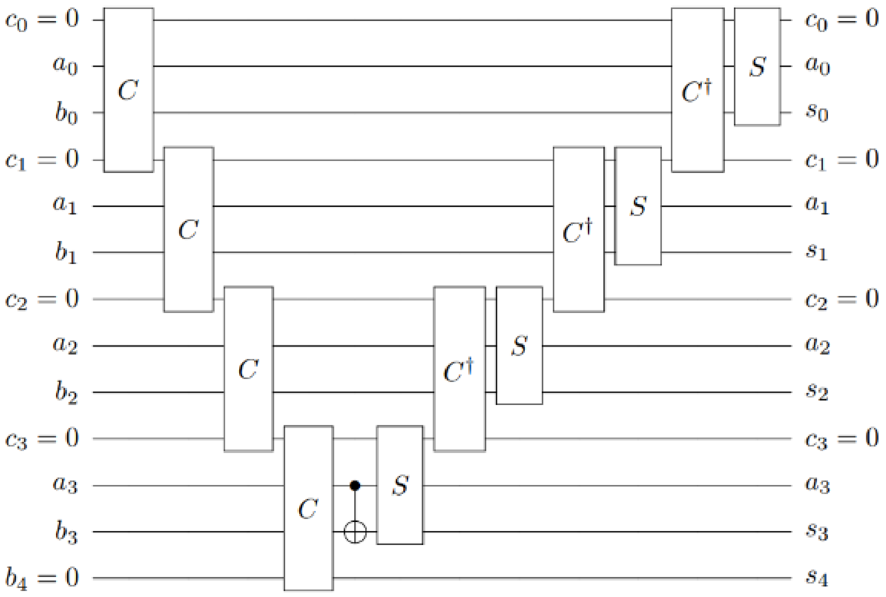
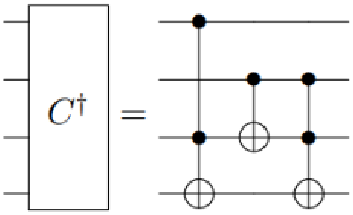


Figure 4: Circuit Diagram of Quantum Ripple-Carry Adder

As shown in the figure above, we need to find the carry-bit of each input bits. So, c_1 stores the carry bit as 1 since $1 + 1 = 10$ where 1 is the carry and 0 is the sum. This sum is stored in b_0. So, after the first case, the values of b_0 and c_1 have been changed. This procedure is followed until the last bit of the inputs. For the final case, we will have b_4 storing the value of the carry-bit, which will be 0 and b_3 has the sum of a_3 and b_3. Clearly from the diagram, $b_4 = s_4$.

Now, we have the carry-bits. However, $b_0, ..b_3$ values have been changed, which now store the values of sum - $a_0 + b_0, ..a_3 + b_3$ respectively. However, we know that the sum is $a_i \oplus b_i \oplus c_i$ where c_i is the carry-bit. Hence, the present values that are stored in the bs do not contain the expected output, therefore, we have to revert their values. For this, we add a $CNOT$ gate to a_3 and b_3 which changes $b_3 \oplus a_3$ to b_3, i.e, $CNOT\,|a_3\rangle\,|b_3 \oplus a_3\rangle = |a_3\rangle\,|b_3 \oplus a_3 \oplus a_3\rangle = |a_3\rangle\,|b_3\rangle$.

Following this, we apply the SUM gate to c_3, a_3 and b_3. Now, s_3 is $|c_3 \oplus a_3 \oplus b_3\rangle$ which is the expected sum. We have to revert the values of b_2 and c_3 back to their initial values. Initially, all c-s have been set to zero. We know that all quantum gates are reversible, hence for the reverting purpose, we apply the inverse-quantum-carry gate to c_2, a_2, c_3 and b_2. After applying this gate, we can apply the *SUM* gate and the same procedure is continued. Below is the circuit diagram of the inverse-quantum-carry gate.



3 Universal Quantum Gates

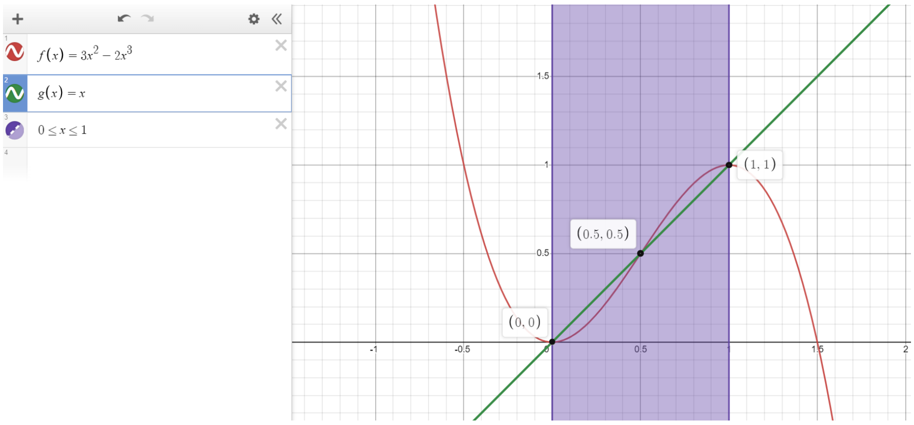
A set of quantum gates that allows us to approximate any quantum gate to any desired precision is called a universal gate set.

1. Hadamard gate can be used for putting qubit into superposition.
2. Z, S, and T only apply phases.
3. X and CNOT gates only flip $|0\rangle$ and $|1\rangle$.
4. Y applies both phases and flips.
5. CNOT and H gates together can be used for generating entanglement. They contain real numbers, so they do not produce states with complex amplitudes.

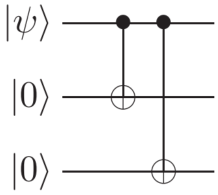
The Clifford group is the set of gates $\{CNOT, H, S\}$. Although this set satisfies all the requirements - entanglement, superposition, and complex amplitudes, however, the Gottesman-Knill theorem states - *A quantum circuit containing only these gates is efficiently simulated by a classical computer.* This means that the Clifford group is only as powerful as a classical computer, so a universal quantum gate set should contain more than this.

4 Quantum Error Correction

In classical communication, we make use of repetition codes to encode the bits before transmitting so as to protect them from the effects of noise in a binary symmetric channel and we can decode the bits by considering the majority voting. Majority voting fails when two or more bits are flipped in case of 3-repetition bit encodings. Let p be the probability of a bit getting flipped in the classical communication channel.Therefore, in this case, the transmission of information becomes reliable provided $p_e < p$ where $p_e = 3p^2 - 2p^3$, and this happens when $p < 0.5$ as we can see in the graph below.



We can apply the same logic to QEC since the no-cloning theorem is not violated as we will be comparing the parities of qubits instead of measuring them, thereby not using the majority voting procedure. So, we will not be disturbing the state as we will not know the values of α and β. We can encode a single qubit state in three qubit states as shown in the diagram below and by doing we will get - $\alpha |0\rangle + \beta |1\rangle \longrightarrow \alpha |000\rangle + \beta |111\rangle$



Now detecting the error and recovering the state is done using the error-detection/ syndrome diagnosis. In this, we have four error syndromes, corresponding to the four projection operators,i.e,

$$\begin{aligned} P_0 &= |000\rangle \langle 000| + |111\rangle \langle 111| - (1) \\ P_1 &= |100\rangle \langle 100| + |011\rangle \langle 011| - (2) \\ P_2 &= |010\rangle \langle 010| + |101\rangle \langle 101| - (3) \\ P_2 &= |001\rangle \langle 001| + |110\rangle \langle 110| - (4) \end{aligned} \tag{1}$$

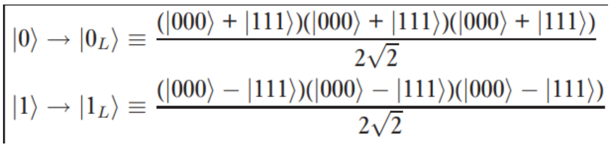
(1) means that no error has occurred on any qubit, (2) indicates error on

qubit-1, (3) indicates error on qubit-2 and (4) indicates error on qubit-3. If the error has occurred on qubit-1, then $\langle\chi| P_1 |\chi\rangle = 1$. This measurement is called error syndrome and since the measurement gives 1, it means error has occurred on qubit-1. The state χ is recovered applying the relevant pauli operators on the qubit which has been flipped. Quantum errors can be modelled as undesired, random applications of the X, Z operators to the qubit state. This method is not completely efficient since the quantum states are continuous in nature, so the error may occur partially on qubit. We can compare if two consecutive qubits are same, by applying this measurement on the qubits - $Z_j Z_k = (-1)^{b_j + b_k}$. If, for example, $Z_1 Z_2 = 1$ and $Z_2 Z_3 = 1$, then no error has occurred, else if $Z_1 Z_2 = -1$ and $Z_2 Z_3 = -1$, then on qubit-2, error has occurred, else if $Z_1 Z_2 = -1$ and $Z_2 Z_3 = 1$, then on qubit-1, error has occurred or else if $Z_1 Z_2 = 1$ and $Z_2 Z_3 = -1$, then on qubit-3, error has occurred. This related to classical bit parity.

Similar approach can be applied to the phase-flips of quantum states, however, this is done with respect to the X-basis instead of the Z-basis. This is obtained by following the same circuit diagram for the the bit-flips. Following this, we add hadamard gates to each of the three qubit encodings and we get - $|0\rangle \longrightarrow |+++\rangle$ and $|1\rangle \longrightarrow |---\rangle$. In this, error is detected by applying this measurement on the qubits - $X_j X_k = (-1)^{b_j + b_k}$ which is equivalent to $H^{\otimes 3} Z_j Z_k H^{\otimes 3} = X_j X_k$ and the state is recovered by applying Z gate on the qubit whose phase has been flipped.

4.1 The Shor code

This is obtained by encoding the single-qubit quantum state using the phase flip code and then encoding each of these qubits using the bit flip code. The Shor code is the concatenation of the bit flip and phase flip error correcting codes. This code can detect and recover the quantum state from any arbitrary errors on any one of the 9 qubits. We obtain the following upon doing so as shown in the figure 5. For phase flips on any one qubit, we make use of the syndrome measurement on the first two blocks of three qubits, and on the last two blocks. And, we then compare the signs to find which block of three qubits has a phase flip, and then revert to its original state.



$$|0\rangle \to |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \to |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

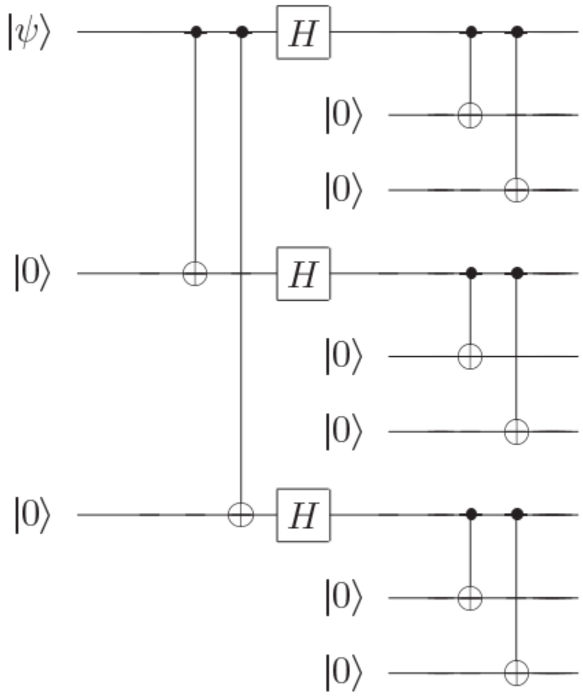


Figure 5: Shor Code Circuit Diagram

For example, the fourth qubit is flipped and a phase in the second block of the encoded state of $\chi = \alpha|0\rangle + \beta|1\rangle$. Then we apply the bit-flip stabilizers to each of the three blocks (or to all nine qubits, in general). On doing so, we obtain -1 for g_3 and 1 for the remaining first six stabilizers. For this bit-flip, we apply X_4 to the fourth qubit. For the phase-flip, we apply the last two stablilizers and we obtain -1 in both the cases, indicating that the phase has been changed in the second block. Therefore, for this, we apply Z gate to any one of the qubits in the second block or to all of them. Below is the table consisting of the eight stabilizers needed to detect and recovering the encoded state.

Name	Stabilizer
g_1	$Z_1Z_2I_3I_4I_5I_6I_7I_8I_9$
g_2	$I_1Z_2Z_3I_4I_5I_6I_7I_8I_9$
g_3	$I_1I_2I_3Z_4Z_5I_6I_7I_8I_9$
g_4	$I_1I_2I_3I_4Z_5Z_6I_7I_8I_9$
g_5	$I_1I_2I_3I_4I_5I_6Z_7Z_8I_9$
g_6	$I_1I_2I_3I_4I_5I_6I_7Z_8Z_9$
g_7	$X_1X_2X_3X_4X_5X_6I_7I_8I_9$
g_8	$I_1I_2I_3X_4X_5X_6X_7X_8X_9$

Using shor code, we can correct any arbitrary errors and this is done by expanding a trace-preserving quantum operation ϵ in an operator-sum representation with operation elements as E_i. To understand how this works, we can consider only a single term in this sum - $\epsilon(|\chi\rangle\langle\chi|) = \sum_i E_i |\chi\rangle\langle\chi| E_i^\dagger$. As an operator on a single qubit, E_i can be written as a linear combination I, X, Z and XZ, i.e, $E_i = e_{i0}I + e_{i1}X_1 + e_{i2}Z_1 + e_{i3}X_1Z_1$. $E_i|\chi\rangle$ is an un-normalized quantum state. Measuring the error syndrome, the state will collapse into any one of the four states. Depending on the result of the measurement, we can apply relevant operation on the result we obtained to get the original state χ.

Supposing the noise is affecting more than one qubit and we are assuming that the noise's effect on qubits is acting independently, then in this case, we can consider its effect as a sum of error on no qubits, one qubit and so on, with the terms of errors on the first two cases dominating the higher terms. Therefore, we can correct the zeroth and first order terms properly and leaving out the higher order errors, hence achieving a net suppression of error.

5 References

1. Chapter 4 from ”Introduction to Classical and Quantum Computing”
2. Lecture Notes by Professor D K Ghosh
3. Nielson and CHuang
4. Lecture Videos by Professor D K Ghosh
5. Quantum Ripple Carry Adder
6. Ripply Carry Adder
7. Spring School on Quantum Error Correction
8. Introduction to Quantum Error Correction and Fault Tolerance
9. QIP2021 Tutorial: Architectures for fault tolerant quantum computing
10. A bird's-eye view of quantum error correction and fault tolerance
11. Yale Quantum Error Correction Study Group

12. Peter Shor— Quantum error correcting codes and fault tolerance

13. Quantum Error Correction Codes by Boyu Zhou

14. Quantum Error Correction Aaron Somoroff, Ray Mencia, and Ron Belyansky Joint Quantum Institute, University of Maryland, College Park - Dec 2018