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-

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

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.

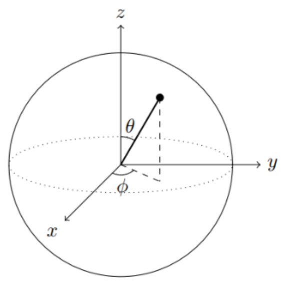


Figure 1: Bloch Sphere

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	$A \to \overline{A}$	X-Gate	A —$\boxed{X}$— $\overline{A}$
AND	$A, B \to AB$	Toffoli	$A, B, 0 \to A, B, AB$
OR	$A, B \to A+B$	anti-Toffoli	$A, B, 1 \to A, B, A+B$
XOR	$A, B \to A \oplus B$	CNOTs	$A, B, 0 \to A, B, A \oplus B$
NAND	$A, B \to \overline{AB}$	Toffoli	$A, B, 1 \to A, B, \overline{AB}$
NOR	$A, B \to \overline{A+B}$	anti-Toffoli	$A, B, 0 \to A, B, \overline{A+B}$

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 -

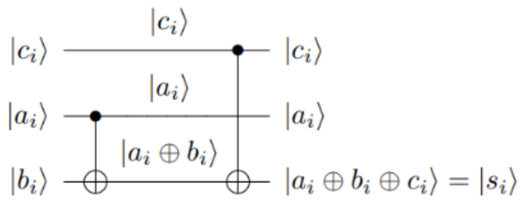


Figure 2: Circuit Diagram of Quantum Sum

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.

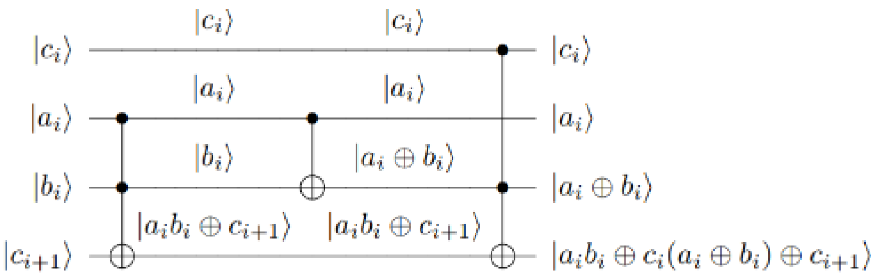


Figure 3: Circuit Diagram of Quantum Carry

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 $|0001\rangle$.

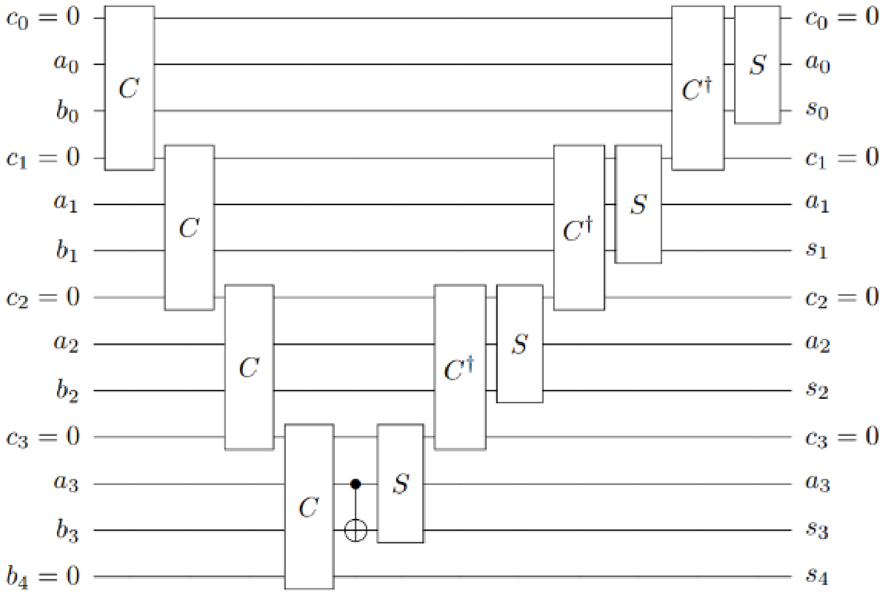
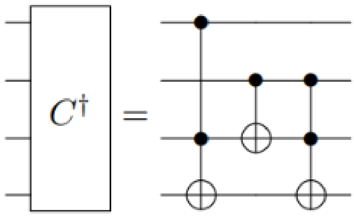


Figure 4: Circuit Diagram of Quantum Ripple-Carry Adder

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, $CNOT\,|a_3\rangle\,|b_3 \oplus a_3\rangle = |a_3\rangle\,|b_3 \oplus a_3 \oplus b_3\rangle = |a_3\rangle\,|b_3\rangle$.

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 References

1. Chapter 4 from "Introduction to Classical and Quantum Computing"
2. Lecture Notes by Professor D K Ghosh

3. Lecture Videos by Professor D K Ghosh

4. Quantum Ripple Carry Adder

5. Ripply Carry Adder