Pierre-Antoine Graham

Homework 2

Eduardo Martín-Martínez, Bindiya Arora
Quantum Information

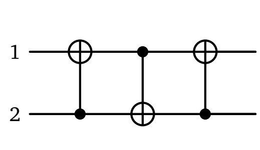
Perimeter Institute for Theoretical Physics
March 31, 2024

Contents

1 Back to basics: quantum circuits

In what follows, we evaluate the matrix expressions representing a quantum circuit unitary acting on a sequence of qubit input. We work in the computational basis $\{|0\rangle, |1\rangle\}$ and use the notation X, Y, Z for the Pauli gates in this basis.

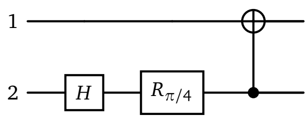
(a) First we consider the conjugation of a CNOT by two CNOT with control and target qubit reversed:



$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

which exchanges the qubits ($|00\rangle \to |00\rangle, |01\rangle \to |10\rangle, |10\rangle \to |01\rangle, |11\rangle \to |11\rangle$) and constitutes a SWAP gate. The matrix expression for the reversed CNOT was obtained by writing its action on the computational basis which reads $|00\rangle \to |00\rangle, |01\rangle \to |11\rangle, |10\rangle \to |10\rangle, |11\rangle \to |01\rangle$.

(b) Then we calculate the matrix expression of the entanglement-generating circuit



$$= (1_1 \otimes |0\rangle\langle 0|_2 + X_1 \otimes |1\rangle\langle 1|_2) R_{\pi/4,2} \left(1_1 \otimes \frac{1}{\sqrt{2}}(X_2 + Z_2) \right)$$

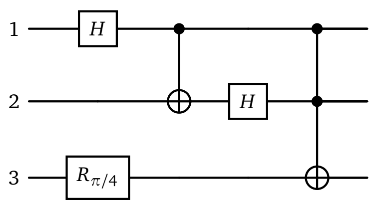
$$= (1_1 \otimes |0\rangle\langle 0|_2 + X_1 \otimes e^{i\pi/4} |1\rangle\langle 1|_2) \left(1_1 \otimes \frac{1}{\sqrt{2}}(X_2 + Z_2) \right)$$

$$= \frac{1}{\sqrt{2}} \left(1_1 \otimes |0\rangle_2 (\langle 0|_2 + \langle 1|_2) + X_1 \otimes e^{i\pi/4} |1\rangle_2 (\langle 0|_2 - \langle 1|_2) \right) = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ e^{i\pi/4} & -e^{i\pi/4} \end{pmatrix} \right)$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & e^{i\pi/4} & -e^{i\pi/4} \\ 0 & 0 & 0 & 0 \\ e^{i\pi/4} & -e^{i\pi/4} & 0 & 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & e^{i\pi/4} & -e^{i\pi/4} \\ 0 & 0 & 1 & 1 \\ e^{i\pi/4} & -e^{i\pi/4} & 0 & 0 \end{pmatrix}.$$

If we set the phases to 1, we recover the Bell state mapping $|00\rangle \to (|00\rangle + |11\rangle)/\sqrt{2}$, $|01\rangle \to (|00\rangle - |11\rangle)/\sqrt{2}$, $|10\rangle \to (|01\rangle + |10\rangle)/\sqrt{2}$ and $|11\rangle \to (-|01\rangle + |10\rangle)/\sqrt{2}$.

(c) Finally, we calculate the matrix expression associated with a three-qubit circuit as follows:



$$= \texttt{TOFFOLI}\ R_{\pi/4,3} H_2 (|0\rangle\langle 0|_1 \otimes 1_2 + |1\rangle\langle 1|_1 \otimes X_2) H_1$$

$$= \texttt{TOFFOLI}\ R_{\pi/4,3} \frac{1}{\sqrt{2}} (|0\rangle_1 (\langle 0|_1 + \langle 1|_1) \otimes H_2 + |1\rangle_1 (\langle 0|_1 - \langle 1|_1) \otimes H_2 X_2)$$

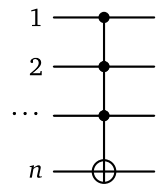
$$= \texttt{TOFFOLI}\ R_{\pi/4,3} \frac{1}{2} \left(\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right)$$

$$= \texttt{TOFFOLI}\ R_{\pi/4,3} \frac{1}{2} \left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix} \right)$$

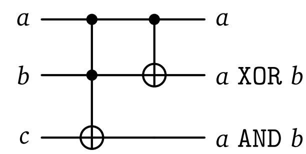
$$= \texttt{TOFFOLI}\ R_{\pi/4,3} \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1_3 & 0 & 0 & 0 \\ 0 & 1_3 & 0 & 0 \\ 0 & 0 & 1_3 & 0 \\ 0 & 0 & 0 & X_3 \end{pmatrix} R_{\pi/4,3} \begin{pmatrix} 1_3 & 1_3 & 1_3 & 1_3 \\ 1_3 & -1_3 & 1_3 & -1_3 \\ 1_3 & 1_3 & -1_3 & -1_3 \\ -1_3 & 1_3 & 1_3 & -1_3 \end{pmatrix}.$$

2 Quantum Adder

(a) The TOFFOLI gate can be generalized to n qubits by increasing the number of control qubit to $n-1$ conditioning a NOT operation on qubit n. The circuit corresponding to this generalization is presented in Fig. 1 (a).



(a) n-qubit generalisation of the TOFFOLI gate.



(b) Circuit for the addition of two single-bit numbers a and b.

Figure 1: Circuits for 2 (a) and 2 (b).

(b) Given two single-digit binary numbers a and b, we can calculate $a+b$ with a quantum circuit by encoding them in input numbers in qubit states $|a\rangle$ and $|b\rangle$ where the digit forms the $0,1$ label of a computation basis element. In other words, the classical bit adder algorithm can be implemented with a quantum circuit. This algorithm requires qubits for inputs a and b and a qubit initialized to $|0\rangle$ that will eventually be updated to store the carry-on of $a+b$ (if we add $1+1$ we get 10 which is represented here by having 1 stored in the carry on qubit, and 0 stored in the output state for the b qubit Hilbert space. The state of a is unchanged to allow for the reversibility of calculation and its implementation as a sequence of unitary operations). The Quantum circuit implementing the single is presented in Fig. 1 (b). On one hand, the TOFFOLI gate flips the carry-on c to 1 iff both a and b are initialized to 1. On the other hand, the CNOT gate replaces the value of b by a XOR b storing the first digit of the addition output in b.

(c) If we consider adding 4 binary numbers a, b, c, d together, we need an additional carry-on qubit to represent the result since $1+1+1+1=100$ requires 3 qubits to describe its digits. In this case, we name the carry-on c_1 and c_2. The circuit performing the addition of 4 single-bit numbers together is presented in Fig. 2 The first two layers of this circuit add a and b in the way explained in (b). At the

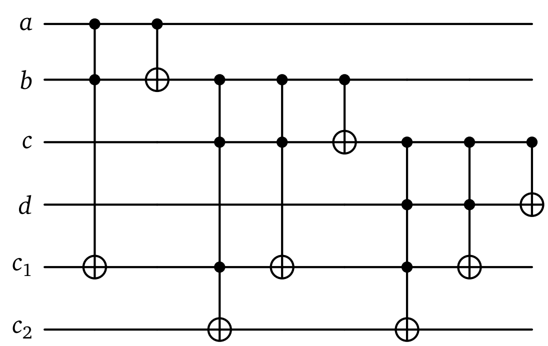


Figure 2: Circuit for the addition of four single bit numbers a, b, c, d. The output is stored in qubit d (first digit), c_1, c_2 (last digit)

second layer, the CNOT operation shifts the last digit output to the b qubit. The next group of operations uses the value stored in b, c, c_1 to add the updated b and c while taking the c_1 carry-on into account. The first gate of the third layer flips c_2 iff b, c, c_1 are all one (in our case this does not happen because $c_1 = 0$ implies $b = 0$). The fourth and fifth layers are copies of the operations described in (b) acting on b, c, c_1. At the fifth layer, the CNOT operation shifts the last digit output to the c qubit. At layer six, a generalized TOFFOLI is applied to possibly flip the c_2 carry-on if c, d, c_1 are all 1 (which is a real possibility in this case). The seventh layer updated the value of the c_1 carry on from the values of c, d (if it was 1 and gets a 1 contribution from c, d, it is flipped to 0. The required carry-on to c_2 associated with this flip was already done by the sixth layer). At the last layer, the CNOT operation shifts the last digit output to the d qubit. The final output of the addition is stored in qubits d, c_1, c_2.

3 Grover's algorithm on IBM composer

(1) The Grover algorithm is implemented in three parts. The first part is a layer of Hadamard gates preparing the qubits initialized to 0 in a uniform superposition of all binary strings in which we need to search. The second part consists in applying `NOT` gates to go to a basis where the target string is mapped to $1 \cdots 1$. Then, in that basis, a generalized `CZ` operation is applied to the last qubit and controlled by all other qubits. This Operation plays the role of the oracle applying a -1 factor to the desired state only. To go back to the initial computational basis, we then apply `NOT` gates to undo the previous basis transformation. In practice, a generalized `CZ` gate can be constructed by conjugating a generalized `CNOT` by two Hadamard gates on the target qubit space. Since $H^2 = 1$ (if controls do not activate, we apply 1) and $HXH = Z$ (if controls activate, we apply Z), this indeed realizes a generalized `CZ`. The third step consists of applying a reflection with respect to state $|s\rangle$ which is a uniform superposition of all states. To perform that reflection, we first move to a basis where this state is moved to $1 \cdots 1$. This is done by first applying Hadamard gates to all qubits (mapping $|s\rangle$ to the state with binary string $0 \cdots 0$) and then applying `NOT` to all of them (mapping $0 \cdots 0$ to $1 \cdots 1$). In this basis, a generalized `CZ` (with a target on the last qubit and controlled by all other qubits) is applied to add a -1 phase only to the $1 \cdots 1$ (In the geometric interpretation of Grover's algorithm, the reflection applies a -1 phase to all components except the $|s\rangle$ component. What is done here differs from this operation by a global -1 phase and produces the same results at the measurement step). We finally return to the computational basis by applying a layer of `NOT` gates followed by a layer of Hadamard gates. See the example circuits linked in the `Circuits.txt` file.

4 Acknowledgement

Thanks to Jonathan for a discussion about the quantum adder.