Quantum Computing Circuits

Practical Quantum Computing using Qiskit and IBMQ

Jothishwaran C.A.

Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee

September 26, 2020

Outline

Representing Quantum transformations

- Consider the figure shown below:

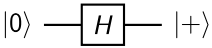

Figure 1: The Hadamard transformation

- This figure shows a qubit that is initially in the $|0\rangle$ state is transformed into the $|+\rangle$ by the H transformation.
- This picture is very similar to classical circuits where the transformation H is treated as the quantum equivalent of the classical logic gate.
- This picture is referred to as a quantum circuit. This is the model used to describe computation in the IBMQ experience. It should be noted that the above diagram is an open circuit.

Conventions for drawing circuits

- The following conventions are used while drawing quantum circuits:
  - The wires (horizontal single lines) represent the qubits and may be labelled.
  - Classical data/bits are represented by double lines
  - The operations (gates) on these qubits are drawn on the wire.
  - The operations are performed from the left to the right one at a time.
  - The stages of the operations maybe separated witn vertical lines, shown below in red.
  - Control qubits are represented by solid circles. Target qubits are represented by '$\oplus$'.

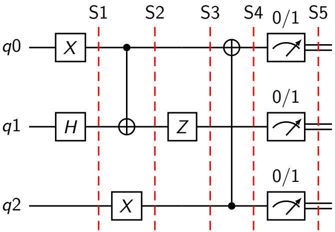

Figure 2: A labelled 3-qubit circuit with stages identified

The order of operations

- As can be seen from the previous circuits, a stage is an interval where an operation (gate) is performed on at least one qubit.
- The operations performed in the previous circuit maybe described as follows:
  - At S1: The X-gate is applied on $q0$; H-gate is applied to $q1$; No gate is applied on $q2$.
  - At S2: The $CNOT_1^0$ is applied, $q0$ controls the transformation on $q1$; X-gate is applied on $q2$.
  - At S3: Z-gate is applied to $q1$; No gates are applied on $q0$ and $q2$.
  - At S4: The $CNOT_0^2$ is applied, $q2$ controls the transformation on $q0$; No gate is applied on $q2$.
  - At S5: All the qubits are measured in the computational basis and the classicall result is transmitted.
- The circuit therefore depicts the order in which transformation are performed on the qubits.

A Classical Computer

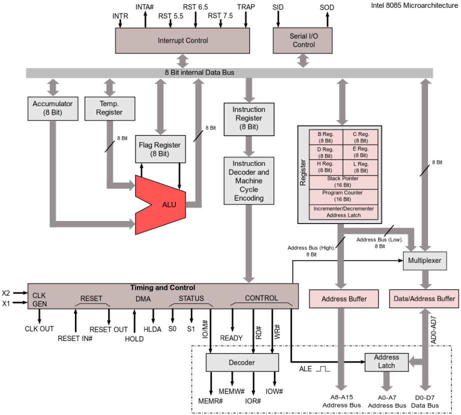

Figure 3: The Intel 8085 Microarchitecture[1]

[1] Appaloosa (https://commons.wikimedia.org/wiki/File:Intel_8085_arch.svg), "Intel 8085 arch", https://creativecommons.org/licenses/by-sa/3.0/legalcode

Classical Computing

- The information stored and processed in the CPU is in the form of classical bits.
- The data is stored in the registers of the memory. Similarly the state of the CPU is also stored in the various registers.
- The processing/computation is done by the Timing and Control unit, Arithmetic and Logic unit (ALU), etc.
- The information stored in the memory is transferred to the CPU through the data buses (the dark grey arrows).
- The data signal carried through the bus depends on the information stored in the register (0/1). The data can therefore be replicated from one register to another.

Replicating qubit states: is it possible?

- **<u>Let us assume</u>** that there exists a two-qubit unitary transformation U that can replicate any qubit state.
- Considering a qubit in the state $|\psi\rangle$ and a second qubit in state $|0\rangle$ and U can be defined as follows:

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$

  the qubit state $|\psi\rangle$ is replicated in the second qubit.
- If there is another state $|\phi\rangle$ that can be replicated then,

$$U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$$

- Therefore for a normalised linear combination of $|\psi\rangle$ and $|\phi\rangle$, for the state, $|\chi\rangle = a|\psi\rangle + b|\phi\rangle$ U is expected to act as shown below:

$$\begin{aligned} U|\chi\rangle|0\rangle &= |\chi\rangle|\chi\rangle \\ &= (a|\psi\rangle + b|\phi\rangle) \otimes (a|\psi\rangle + b|\phi\rangle) \\ &= aa|\psi\rangle|\psi\rangle + ab|\psi\rangle|\phi\rangle + ba|\phi\rangle|\psi\rangle + bb|\phi\rangle|\phi\rangle \end{aligned}$$

Replicating qubit states: No Clones

- However, U is also a linear operator, therefore

$$\begin{aligned} U|\chi\rangle|0\rangle &= aU|\psi\rangle|0\rangle + bU|\phi\rangle|0\rangle \\ &= a|\psi\rangle|\psi\rangle + b|\phi\rangle|\phi\rangle \end{aligned}$$

- This is a contradiction and therefore the original assumption is wrong. This means U cannot act on $|\chi\rangle|0\rangle$ is the expected manner.
- It is impossible to replicate (or clone) an arbitrary, unknown quantum state.
- The above statement is referred to as the No cloning theorem.

Quantum Computing without clones

- The state of a quantum computer is given by the state of the qubits. Therefore, the qubits are the memory of a quantum computer.
- Since it is impossible to replicate (copy/clone) the state of a qubit. Quantum computation cannot be performed in the same sense as classical computation.
- This also prevents basic read/write operations in the classical sense on a quantum computer.
- All computations are therefore performed on the initial set of qubits that are also the memory.
- Quantum computation is therefore in-memory computation where the operations are performed directly on the memory of the computer.

Circuits as a physical representation

- Understanding that computation being represented here is in-memory, it is possible to reinterpret the circuit as follows.
  - Every stage denotes a different time interval, this is the reason why the operations (gates) are being performed in a definite order.
  - This also implies that a separate device needs to keep track of the time and decide when to apply the gates.
  - This device is usually a classical computer.
  - The circuit is therefore interpreted/complied on a classical computer which will in turn apply appropriate gate transformations on the quantum computer.

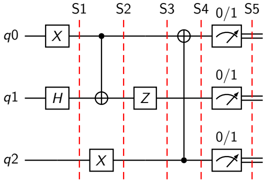

Figure 4: The example circuit

From circuits to matrices

- If a circuit contains n qubits, the overall transformation will correspond to an n-qubit operation.
- The operator corresponding to each stage can be evaluated by taking the tensor product of the independent operators, denoted by '$\otimes$'.
- If no operation is being performed on a qubit in a particular stage, one takes the tensor product with the 2×2 identity matrix
- Across multiple stages, the operator matrices are evaluated by matrix multiplication, denoted by '$\circ$'.
- These rules will be applied to the example circuit to get the resultant operation.

The “example” operation

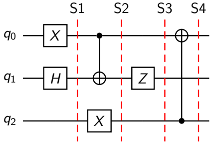

Figure 5: The example circuit with measurement removed

- It is assumed that the circuit state is represented in the *big-endian* notation: $|q_0\rangle |q_1\rangle |q_2\rangle$.
  - At S1: $X \otimes H \otimes I$.
  - At S2: $CNOT_1^0 \otimes X$.
  - At S3: $I \otimes Z \otimes I$.
  - At S4: $CNOT_0^2$, this operator is being represented as a 3-qubit operation.
- The resultant operation is therefore given by (note that the matrix order is reversed):

$$CNOT_0^2 \circ (I \otimes Z \otimes I) \circ \left(CNOT_1^0 \otimes X\right) \circ (X \otimes H \otimes I) |q_0\rangle |q_1\rangle |q_2\rangle$$

General Multiqubit States

- Consider a system of n-qubits each of which are in the $|0\rangle$ state.
- The n-qubit state is defined as

$$|0_n\rangle \equiv |0\rangle_1 \otimes |0\rangle_2 \otimes \cdots \otimes |0\rangle_{n-1} \otimes |0\rangle_n$$

- Applying the X gate to all these qubits yields:

$$X_1 \otimes X_2 \otimes \cdots \otimes X_{n-1} \otimes X_n |0_n\rangle = |1\rangle_1 \otimes |1\rangle_2 \otimes \cdots \otimes |1\rangle_{n-1} \otimes |1\rangle_n$$

  this state is represented as $|1_n\rangle$
- Similarly applying H gate to all these qubits yields:

$$H_1 \otimes H_2 \otimes \cdots \otimes H_{n-1} \otimes H_n |0_n\rangle = |+\rangle_1 \otimes |+\rangle_2 \otimes \cdots \otimes |+\rangle_{n-1} \otimes |+\rangle_n$$

  this state is represented as an equal weight superposition of all states corresponding to n-bit classical strings x:

$$\sum_{x\in\{0,1\}^n} \frac{1}{2^{\frac{n}{2}}} |x\rangle$$

The Qubit register

- The quantum circuits are for the previous operation are given as follow:
- The n-qubit state is defined as

$$|0_n\rangle \;\overset{n}{-\!\!/\!\!-}\; \boxed{X^{\otimes n}} \;-\!\!-\; |1_n\rangle$$

Figure 6: X gates on each qubit of $|0_n\rangle$

and,

$$|0_n\rangle \;\overset{n}{-\!\!/\!\!-}\; \boxed{H^{\otimes n}} \;-\!\!-\; \sum_{x\in\{0,1\}^n} \frac{1}{2^{\frac{n}{2}}} |x\rangle$$

Figure 7: H gates on each qubit of $|0_n\rangle$

Boolean Function and Quantum gates

Practical Quantum Computing using Qiskit and IBMQ

Jothishwaran C.A.

Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee

September 26, 2020

Outline

Boolean Variables and Operations

- A Boolean variable is an element belongs to the set $\{0, 1\}$
- Some Boolean operations are defined as follows:

- The AND operation or conjunction '$\cdot$'

$$\begin{aligned} 0 \cdot 0 &= 0 \\ 0 \cdot 1 &= 0 \\ 1 \cdot 0 &= 0 \\ 1 \cdot 1 &= 1 \end{aligned}$$

- The OR operation or disjunction '$\vee$'

$$\begin{aligned} 0 \vee 0 &= 0 \\ 0 \vee 1 &= 1 \\ 1 \vee 0 &= 1 \\ 1 \vee 1 &= 1 \end{aligned}$$

- These operations have their corresponding physical versions known as logic gates.

Variables and Operations [contd.]

- The XOR operation or bit addition: '$\oplus$' or '$+$'

$$\begin{aligned} 0+0 &= 0 \\ 0+1 &= 1 \\ 1+0 &= 1 \\ 1+1 &= 0 \end{aligned}$$

- The NOT operation or negation: '$\neg$' or '$^-$'

$$\begin{aligned} \neg 0 &= \bar{0} = 1 \\ \neg 1 &= \bar{1} = 0 \end{aligned}$$

- In general we can write, $\bar{x} = 1 + x$.

- The XOR gate can also be seen as integer addition *modulo* 2, i.e. adding the numbers and taking the remainder after dividing by 2.
- Similarly, The conjunction (AND) operation can be seen as multiplication *modulo* 2.
- It is also to be noted that some of these logic gates form sets known as universal gates. This implies any Boolean operator can be expressed in terms of the universal operators.

Classical bits and registers

- The single Boolean variable is the same as a classical bit, $x \in \{0, 1\}$ and is the quantity stored in a flip-flop.
- A register containing n-bits can be represented by the Boolean string $x \equiv x_0 x_1 \ldots x_{n-1}$; where , $x_i \in \{0, 1\} : \; i \in [0, n-1]$.
- The i^{th} element of the string will be denoted by $x^{(i)} = x_i$.
- There are 2^n possible values x, if treated as integers to the base 2 these strings will have values from 0 to $2^n - 1$.
- For instance, $n = 2$ the strings x are $\{00, 01, 10, 11\}$ corresponding to the integers 0 to $2^2 - 1$.
- These strings are also referred to as Boolean vectors and defined as, $x \in \{0, 1\}^n$.

Boolean functions and their forms

- A n variable Boolean function, F takes a Boolean string $x \in \{0,1\}^n$ as an input and gives a single Boolean variable $y \in \{0,1\}$:

$$F \ : \ \{0,1\}^n \to \{0,1\} \ : \ F(x) = y$$

- It is possible to represent any Boolean function using the universal gate sets. These representations are known as forms
- If the set $\{\neg, \cdot, \vee\}$ is used, there are two forms, the *Conjunctive Normal Form* (CNF) and the *Disjunctive Normal Form* (DNF).
- If the set $\{+, \cdot\}$ is used, the function is referred to be in the *Algebraic Normal Form* (ANF).

Single variable functions

- A single variable function can be represented as, $F(x) = y$. There are only four possible single variable Boolean functions.

$$
\begin{gathered}
F_0(x) = 0 \;\; ; \;\; F_1(x) = 1 \\
\text{and} \;\; F_2(x) = x \;\; ; \;\; F_3(x) = \bar{x} = 1 + x
\end{gathered}
$$

- These functions can also be represented as truth tables.
- The above functions are all represented in their Algebraic Normal Form.

Two variable functions

- A two variable function can be represented as, $F(x) = y \ : x \in \{0,1\}^2$.
- There are many two variable Boolean functions. Some examples are given below:

$$F(x) = 0 \ ; \ F(x) = x_1$$
$$\text{and} \ F(x) = x_0 \cdot x_1 \ ; \ F(x) = \overline{x_0 \cdot x_1} = \overline{x_0} \vee \overline{x_1}$$

- There are 16 different two variable Boolean functions, 256 three variable Boolean functions and 65536 four variable functions.

The CNOT gate

- Consider a system of two qubits where each qubit is in either the $|0\rangle$ or the $|1\rangle$ state.
- The possible states of the two-qubit system can now be labelled as $|x_0\rangle|x_1\rangle$, where $x_0, x_1 \in \{0, 1\}$.
- Applying the $CNOT_1^0$ gate two this system, the general result is as shown:

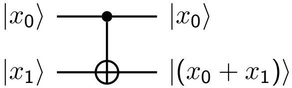

Figure 1: The *CNOT* gate

- Just like the the X-gate was the quantum analogue of the the classical NOT operation, The *CNOT* gate is the analogue of the classical XOR gate.

Two controllers

- Consider a system of three qubits where each qubit is in either the $|0\rangle$ or the $|1\rangle$ state.
- The possible states of the two-qubit system can now be labelled as $|x_0\rangle\,|x_1\rangle\,|x_2\rangle$, where $x_0, x_1, x_2 \in \{0, 1\}$.
- Consider a new gate where $|x_0\rangle$ and $|x_1\rangle$ both control the application of the X-gate on $|x_2\rangle$

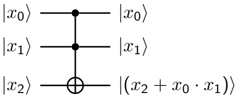

Figure 2: A new gate

- This new gate is called the CCX gate or the Toffoli gate and is crucial for realising Boolean functions on quantum computers.
- If the state $|x_2\rangle$ is set to $|0\rangle$ the this gate is a quantum implementation of the classical AND gate.