Boolean Function Oracles

Introduction to Quantum Computing

Jothishwaran C.A.

Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee

May 20, 2023

Outline

Boolean Strings as Basis Vectors

- A single-qubit computational basis vector can be represented as $|x\rangle$, where $x \in \{0, 1\}$.
- If there n such qubits, each with computational basis state $|x_i\rangle$ where $x_i \in \{0, 1\}$ and $i \in [0, n-1]$.
- Computational basis vector for the n-qubit system can simply be defined as $|x_0\rangle \otimes |x_1\rangle \otimes \cdots \otimes |x_{n-2}\rangle \otimes |x_{n-1}\rangle$. There are 2^n such vectors.
- Each of these basis vectors can now be represented as $|x_0 x_1 \ldots x_{n-1}\rangle \equiv |x\rangle$, where $x \in \{0, 1\}^n$.
- This notation shall be adopted for defining multi-qubit basis vectors for all further discussions.

The X_n and H_n gates

- If the X gate is applied to each of the n qubits of the system. The combined operator can be represented as:

$$\underbrace{X \otimes X \otimes \cdots \otimes X}_{n\text{-times}} \equiv X^{\otimes n}$$

- When the H gate is applied to each of the n qubits of the system. The combined operator can be represented as:

$$\underbrace{H \otimes H \otimes \cdots \otimes H}_{n\text{-times}} \equiv H^{\otimes n}$$

- When these operators are applied to the n-qubit basis state $|0_n\rangle$ the results are as follows:

$$\begin{aligned} X^{\otimes n} |0_n\rangle &= |1_n\rangle \\ H^{\otimes n} |0_n\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \end{aligned}$$

Toffoli gate: a formal introduction

- The Toffoli gate (CCX) is a three qubit gate and has the following actions on the three-qubit computational basis:

$$
\begin{aligned}
CCX\,|000\rangle &= |000\rangle \;;\; CCX\,|100\rangle = |100\rangle \\
CCX\,|001\rangle &= |001\rangle \;;\; CCX\,|101\rangle = |101\rangle \\
CCX\,|010\rangle &= |010\rangle \;;\; CCX\,|110\rangle = |111\rangle \\
CCX\,|011\rangle &= |011\rangle \;;\; CCX\,|111\rangle = |110\rangle
\end{aligned}
$$

- The matrix form is given as:

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

The Toffoli Circuit

- Combining the circuit conventions defined previously and the definitions of Boolean functions The circuit corresponding and the actions to the Toffoli gate is:

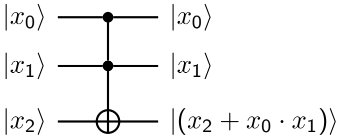

Figure 1: The Toffoli gate

- Here, each state $|x_i\rangle$ is a computational basis vector for the single-qubit state.
- Since there were no phase factors in the actions defined before, the circuit also defines the resultant states for the input state vectors.

A particular setup

- Consider the Toffoli gate with a condition the target qubit is initially fixed in the $|0\rangle$ state:

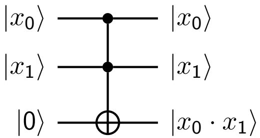

Figure 2: The Toffoli gate

- The effect of the Toffoli gate on these initial states can be summarized as:

$$CCX\,|x_0\rangle\,|x_1\rangle\,|0\rangle \;=\; |x_0\rangle\,|x_1\rangle\,|x_0 \cdot x_1\rangle$$

The AND Oracle

- The *CCX* gate with the initial state as shown before transforms the target qubit from $|0\rangle$ to $|x_0 \cdot x_1\rangle$
- Therefore for any initial control state represented by by $x_0 x_1$ the circuit transforms the input state into the output state $|x_0\rangle\,|x_1\rangle\,|x_0 \cdot x_1\rangle$
- This circuit is referred to as the Quantum AND Oracle: The circuit transforms an input state corresponding to the inputs of a classical AND gate, with a target qubit set to $|0\rangle$ into an output state where the AND operation is performed and stored in the target qubit.
- The two-bit classical AND gate now has an equivalent three-qubit quantum oracle.

Simpler Oracles

- If a two-bit classical gate has a three-qubit quantum oracle, then it maybe possible to define a single bit classical gate with a two-qubit oracle. Consider the following circuit:

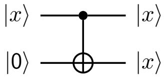

Figure 3: A single bit oracle

- This is the equivalent oracle for the single bit function $F(x) = x$. The other single bit oracle maybe defined as follows:

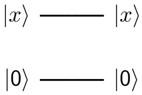

Figure 4: $F(x) = 0$

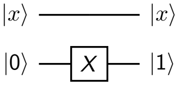

Figure 5: $F(x) = 1$

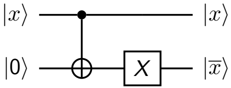

Figure 6: $F(x) = \overline{x}$

Oracles with *CNOT* gates

- As discussed before, the *CNOT* gate acts on a two-qubit basis state $|x_0\rangle |x_1\rangle$ as shown below

$$\mathit{CNOT} |x_0\rangle |x_1\rangle = |x_0\rangle |(x_0 + x_1)\rangle$$

- While this is equivalent to the XOR gate, it is not in the oracle form like the case with the *CCX* gate.
- However, consider the following circuit:

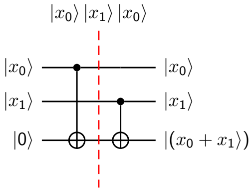

Figure 7: The XOR oracle

- This circuit is the three-qubit oracle equivalent to the classical XOR gate.

Oracles with *CNOT* gates

- A general n-bit classical Boolean function has an equivalent $(n+1)$-qubit quantum oracle representation U_F that acts as follows.

$$U_F \left|x\right\rangle \left|0\right\rangle = \left|x\right\rangle \left|F(x)\right\rangle$$

here, $x \in \{0,1\}^n$ and $\left|x\right\rangle$ is an element of the n-qubit computational basis.

- The circuit representation of the same is as shown below. It should be noted that the upper bundle of qubit wires represent the n-qubit state.

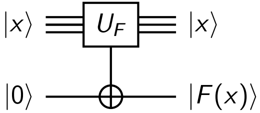

Figure 8: The general quantum Boolean oracle

- It is possible to build any Boolean oracle with the *CNOT*, *CCX* and *X* gates.

Varying the inputs of the Oracles

- Since the oracle is a unitary transformation, they are transformations since the inverse of U_F is simply the adjoint, $U_F^\dagger$.
- If the target qubit of an oracle is fixed to the $|1\rangle$ state, then applying the oracle U_F on such a state yields the following:

$$U_F \left|x\right\rangle \left|1\right\rangle \;=\; \left|x\right\rangle \left|1 + F(x)\right\rangle$$

- The oracle is also, by its definition a linear transformation and so, if we consider a state $|x_1\rangle + |x_2\rangle$ (normalization is disregarded) where $x_1, x_2 \in \{0,1\}^n$. Then, applying U_F on this state gives:

$$U_F(\left|x_1\right\rangle + \left|x_2\right\rangle) \left|0\right\rangle \;=\; \left|x_1\right\rangle \left|F(x_1)\right\rangle + \left|x_2\right\rangle \left|F(x_2)\right\rangle$$

  note that the separable input state may not be separable after U_F is applied.
- This is ability of a quantum oracle to evaluate multiple instances of the same function is a result of the linearity of the oracle. This feature is famously know as *Quantum Parallelism* and it has no classical equivalent.

Varying the inputs of the Oracles

- If the target state of the qubit is set to the $|-\rangle$ state, the action of U_F yields the following:

$$
\begin{aligned}
U_F \, |x\rangle \, |-\rangle \;&=\; U_F \, |x\rangle \, \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
&=\; |x\rangle \, \frac{1}{\sqrt{2}} \left(|0 + F(x)\rangle \;-\; |1 + F(x)\rangle\right)
\end{aligned}
$$

- The resultant target qubit is in the state $|-\rangle$ when $F(x) = 0$ and $-\,|-\rangle$ when $F(x) = 1$. The combined result can be written as,

$$
U_F \, |x\rangle \, |-\rangle \;=\; (-1)^{F(x)} \, |x\rangle \, |-\rangle
$$

- In this case, it can be seen that the state of the target qubit is the same but the output sate gains a phase that corresponds to the value of $F(x)$. This variation of the oracle is referred to as the Phase Oracle implementation of the Boolean function.