Multiqubit gates and C-gates

Introduction to Quantum Computing

Kenneth MAUSSANG

Université de Montpellier

Ecole Centrale Casablanca

2020 – 2021

I.1. Definition

A binary quantum gate is a unitary operation on two qubits, *e.g.* a unitary map $\mathcal{H}_2 \otimes \mathcal{H}_2 \to \mathcal{H}_2 \otimes \mathcal{H}_2$, where $\mathcal{H}_2$ is a hilbert space of dimension 2×2. A basis of $\mathcal{H}_2 \otimes \mathcal{H}_2$ is

$$\{|00\rangle, |10\rangle, |01\rangle, |11\rangle\}.$$

C-gate: let A and B be two qubits. Let M be a unitary quantum gate acting on B. The controlled-M gate (or C-M gate) is the binary gate acting on $A \otimes B$ defined as follow

$$\text{C-}M = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \mathbb{I}_B + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes M,$$

where $\mathbb{I}_B$ is the identity operator on qubit B. A C-gate is the operation such that M is applied to B only if the qubit A is in the state $|1\rangle$.

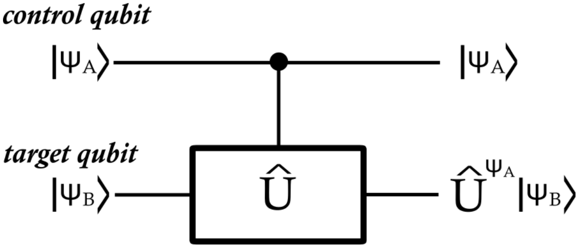
Important example: C-NOT gate.

$$\text{C-NOT} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\hat{U}_{\text{C-NOT}} |00\rangle = |00\rangle, \quad \hat{U}_{\text{C-NOT}} |01\rangle = |01\rangle,$$
$$\hat{U}_{\text{C-NOT}} |10\rangle = |11\rangle, \quad \hat{U}_{\text{C-NOT}} |11\rangle = |10\rangle.$$

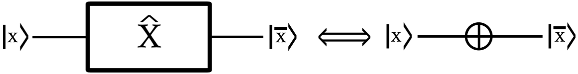
$$\hat{U}_{\text{C-NOT}} \left(\frac{1}{\sqrt{2}} \left(|00\rangle + |10\rangle \right) \right) = \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle \right).$$

C-U gate where the controlled qubit is A.

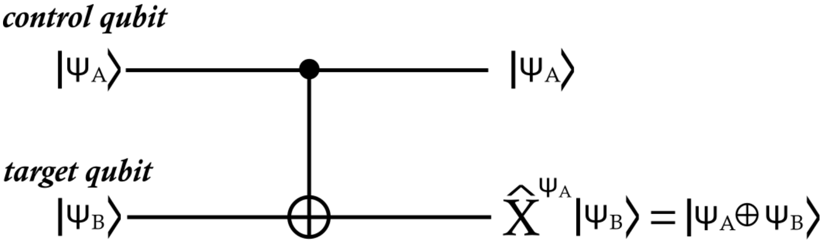


Example: C-NOT gate.

NOT quantum gate



C-NOT gate where the controlled qubit is A.



A C-NOT gate might be seen as a way to implement the XOR classical gate.

Theorem: *All quantum circuits can be constructed using only C-NOT gates and single-qubit gates.*

C-NOT gate is self inverse

$$(\text{C-NOT}) \cdot (\text{C-NOT}) = \mathbb{I} \otimes \mathbb{I}.$$

Ref: Elementary gates for quantum computation, Physical Review A, **52**, 5, 3457-3467 (1995).

Multiqubit gates and C-gates

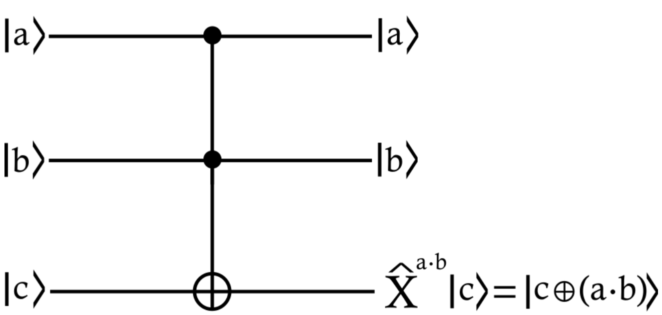
The **Toffoli gate**, originally devised as a universal, reversible classical logic gate by Toffoli, is especially interesting because depending on the input, the gate can perform logical AND, XOR and NOT operations, making it universal for classical computing.

Toffoli is often referred to a "controlled-controlled-NOT" gate (C^2-NOT).

Ref: Reversible Computation, T. Toffoli, Cambridge, MA, p.36 (1988).

II.1. The Toffoli gate

The circuit diagram of a Toffoli gate is the following



Toffoli is self inverse

$$\text{Toffoli} \cdot \text{Toffoli} = \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I}.$$

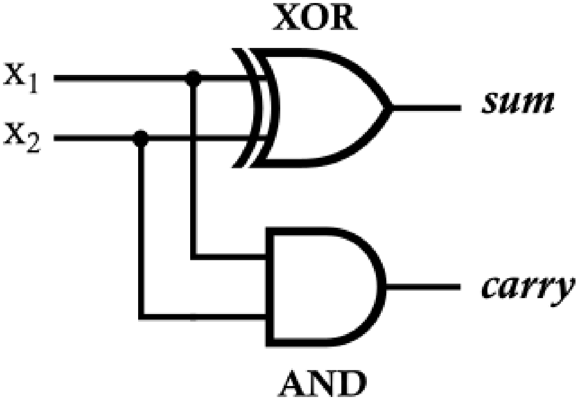
II.1. The Toffoli gate

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

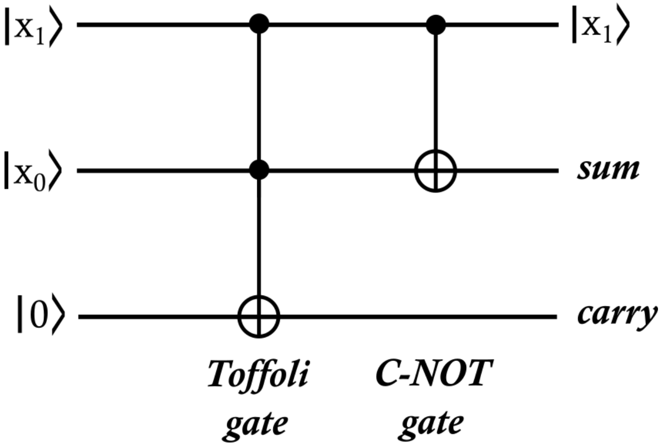
Theorem: *All quantum circuits can be constructed (in some approximated sense) using only Hadamard gates and Toffoli gates.*

Ref: Both Toffoli and controlled-NOT need little help to do universal quantum computation, Yaoyun Shi, Quantum Information and Computation, **3**, 1, 84-92 (2003).

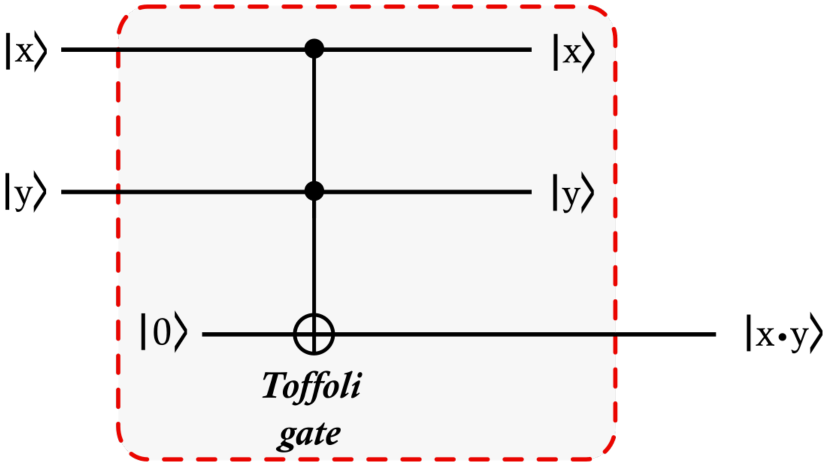
Example: A classical half-adder compute the sum and carry for two bits x_1 and x_0.



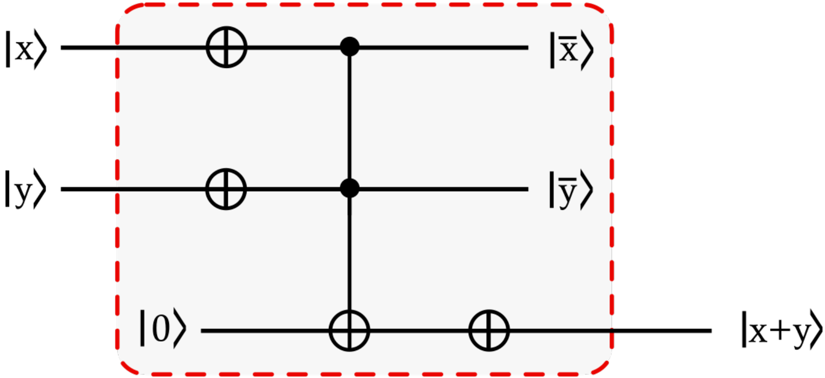
Quantum half-adder.



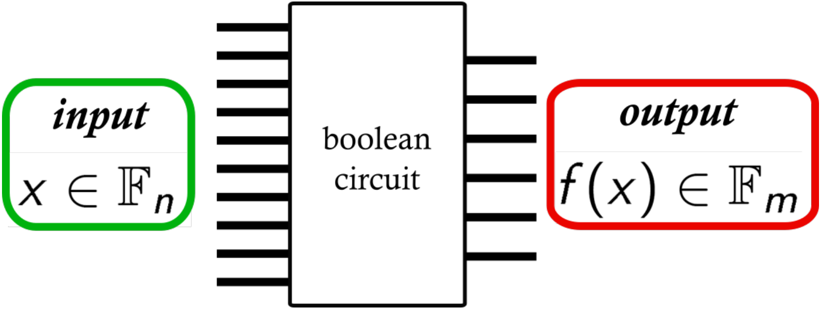
AND logical gate.



OR logical gate.



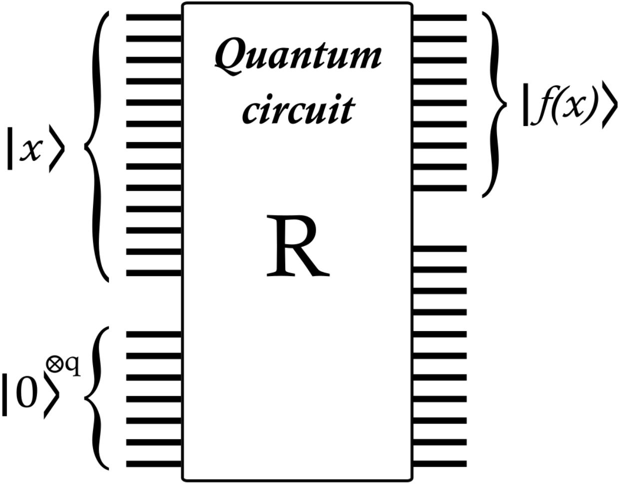
Let note $\mathbb{F}_n = \{0,1\}^n$. A boolean function $f : \mathbb{F}_n \to \mathbb{F}_m$ can't be a unitary operation. The number of inputs do not equal the number of outputs, so the map is not invertible.

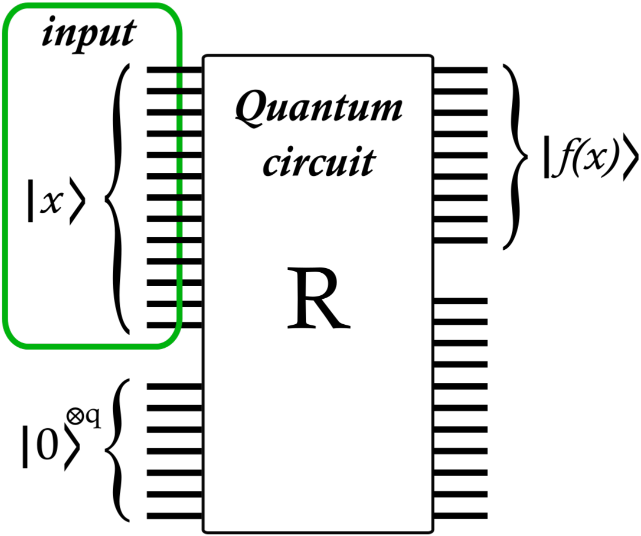


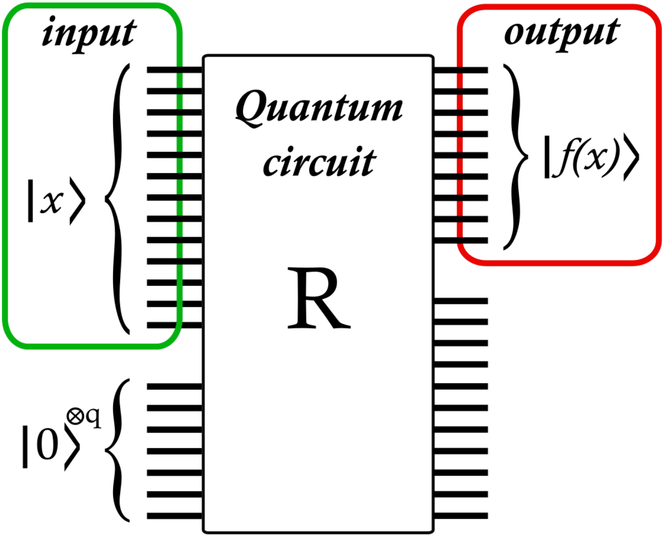
However, it is in fact possible to construct a quantum circuit that performs the same function than any classical boolean circuit.

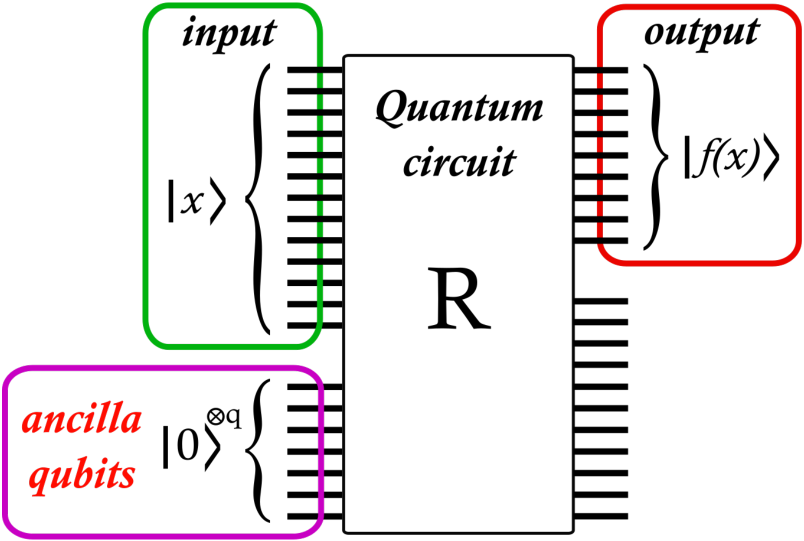
Let $f : \mathbb{F}_n \to \mathbb{F}_m$ a boolean function with k gates. It is possible to construct a quantum circuit, $\hat{R}$, that performs the same function.

This quantum circuit uses $\mathcal{O}(k)$ gates and requires $q = \mathcal{O}(k)$ additional qubits, so-called **ancilla qubits**, only used for the calculation. These ancilla qubits are not part of the quantum register, and are all initially in the pure state $|0\rangle$. Then, such a circuit outputs $n + q - m$ garbage qubits.

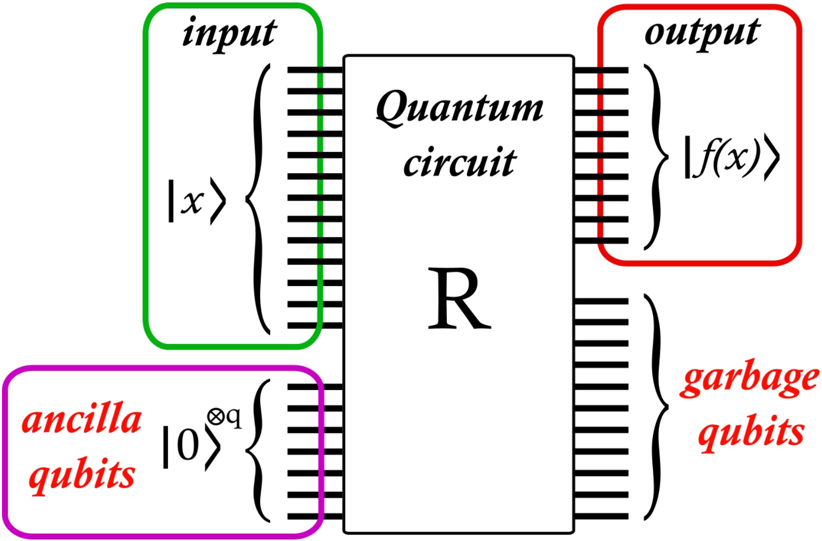

$|x\rangle$
Quantum circuit
R
$|f(x)\rangle$
$|0\rangle^{\otimes q}$


input
$|x\rangle$
Quantum
circuit
R
$|f(x)\rangle$
$|0\rangle^{\otimes q}$


input
$|x\rangle$
$|0\rangle^{\otimes q}$
Quantum circuit
R
output
$|f(x)\rangle$

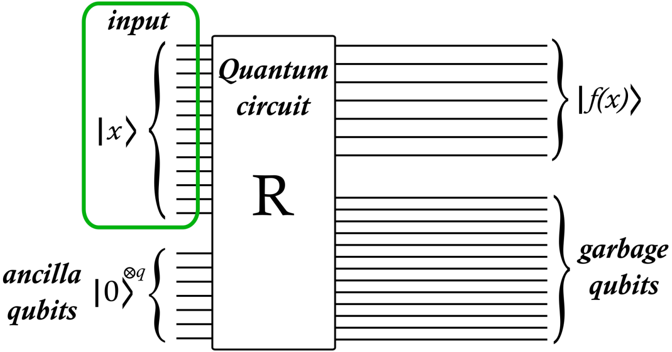

input
$|x\rangle$
Quantum
circuit
R
output
$|f(x)\rangle$
ancilla
qubits
$|0\rangle^{\otimes q}$

II.3. Boolean circuits

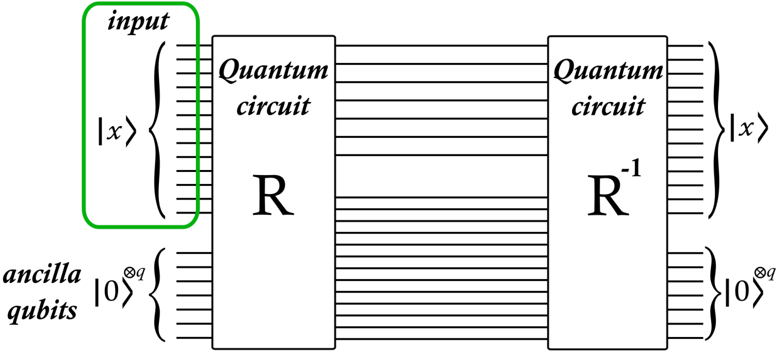


To obtain the invert $\hat{R}^{-1}$ of $\hat{R}$, one just has to take the mirror image of the circuit $\hat{R}$. This is used to "recycle" the ancilla qubits, so that they are reset to $|0\rangle$.

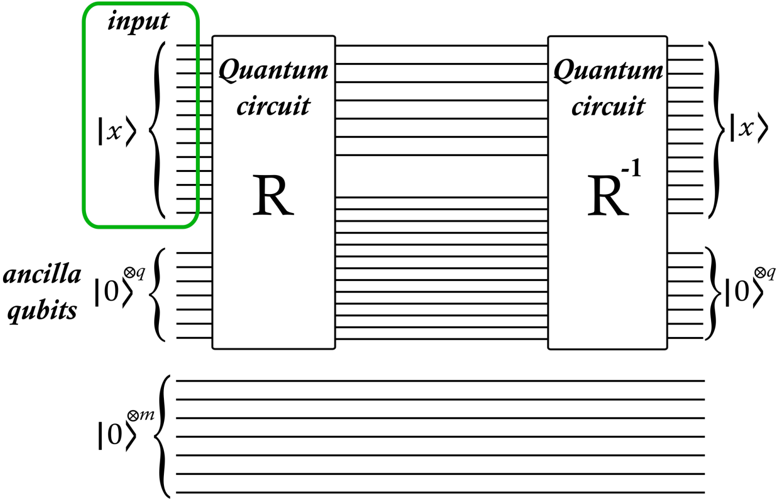
So a quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \to \mathbb{F}_m$ has the following structure



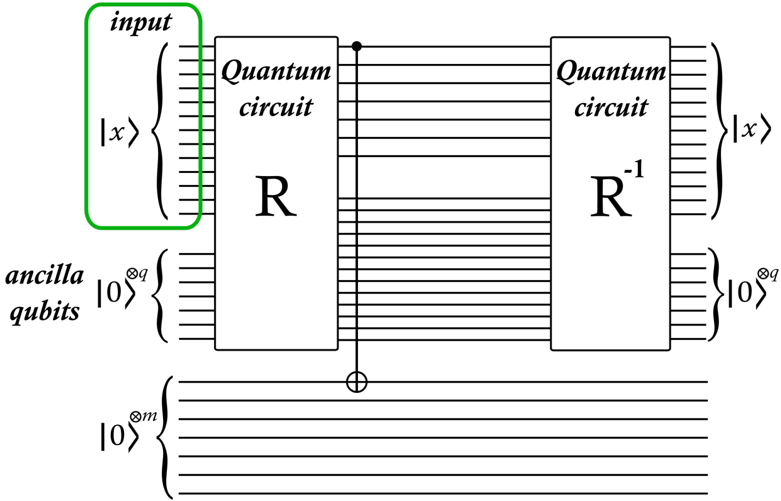
So a quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \to \mathbb{F}_m$ has the following structure



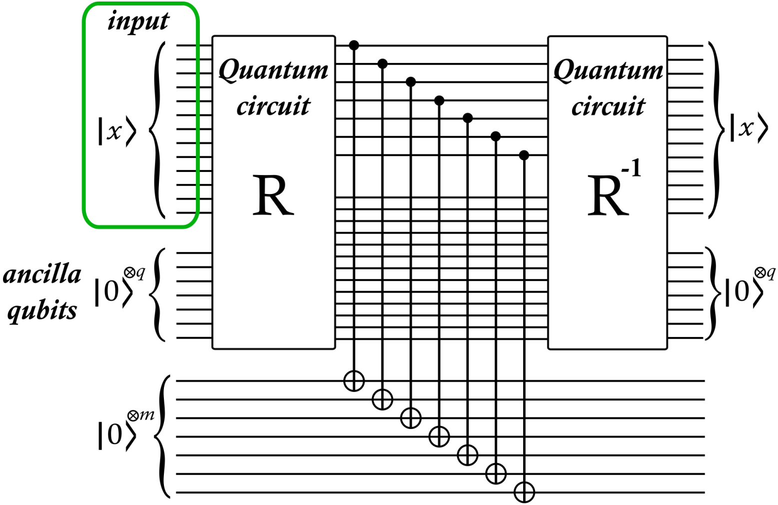
So a quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \to \mathbb{F}_m$ has the following structure



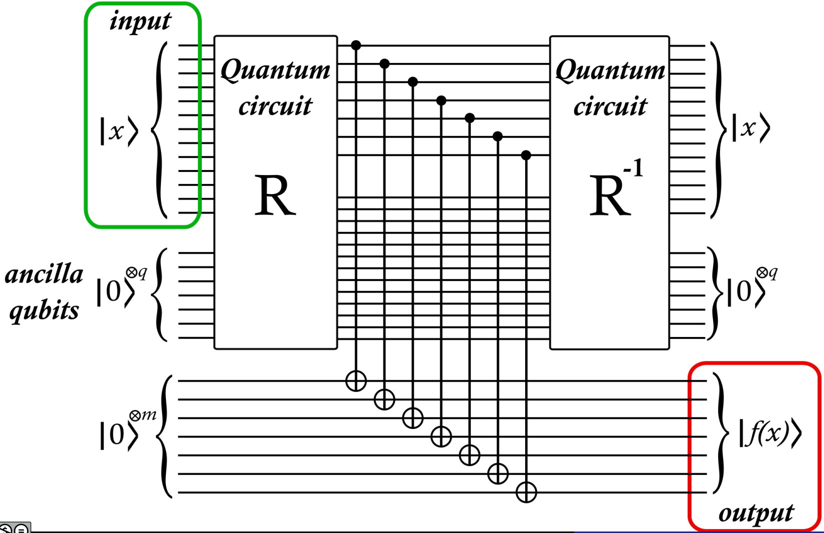
So a quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \to \mathbb{F}_m$ has the following structure



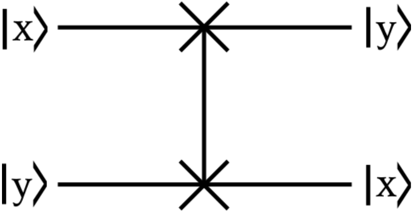
So a quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \rightarrow \mathbb{F}_m$ has the following structure



So a quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \to \mathbb{F}_m$ has the following structure

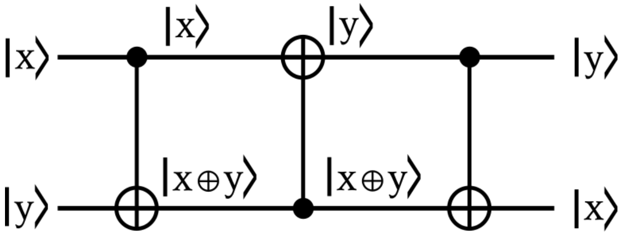


II.4. SWAP gate



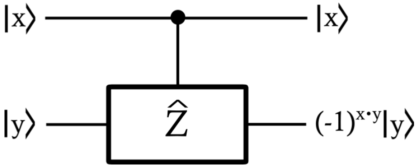
$$\mathrm{SWAP} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

A SWAP gate might be implemented with 3 C-NOT gates



$$
\begin{aligned}
|x\rangle \otimes |y\rangle &\xrightarrow{C_{12}} |x\rangle \otimes |y \oplus x\rangle \\
&\xrightarrow{C_{21}} |x \oplus (y \oplus x)\rangle \otimes |y \oplus x\rangle = |y\rangle \otimes |y \oplus x\rangle \\
&\xrightarrow{C_{12}} |y\rangle \otimes |(y \oplus x) \oplus y\rangle = |y\rangle \otimes |x\rangle
\end{aligned}
$$

II.4. C-Z gate

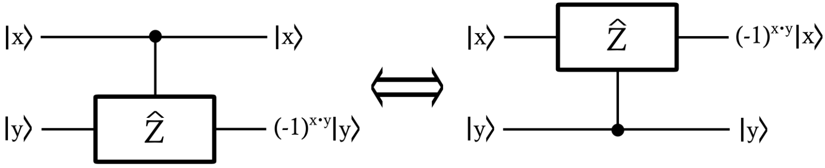


$$\text{C-}Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

$$|x\rangle \otimes |y\rangle \xrightarrow{C_Z} (-1)^{x \cdot y} |x\rangle \otimes |y\rangle.$$

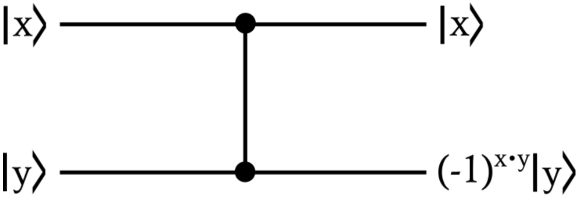
$|x\rangle$ and $|y\rangle$ play symmetric roles.

II.4. C-Z gate



$$(-1)^{x \cdot y}|x\rangle|y\rangle \qquad (-1)^{x \cdot y}|x\rangle|y\rangle$$

Since $|x\rangle$ and $|y\rangle$ play symmetric roles, a C-Z gate is then represented as follow



$$(-1)^{x \cdot y}|x\rangle|y\rangle$$

II.5. Orcale

Oracles are used in the context of query complexity. We assume to have access to an Oracle, for example a physical device that we cannot look inside, but to which we can pass queries and which returns answers.

On a classical computer, the Oracle is given by a function f

$$f : \mathbb{F}_n \longrightarrow \mathbb{F}_m.$$

On a quantum computer, the oracle must be **unitary**.

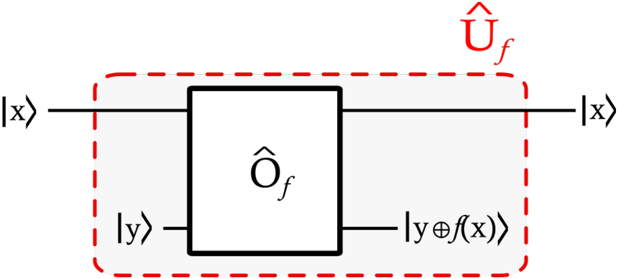
The operator $\hat{O}_f$ is a quantum oracle which can be seen as a unitary operator which performs the map

$$\hat{O}_f \left|x\right\rangle \otimes \left|y\right\rangle = \left|x\right\rangle \otimes \left|y \oplus f(x)\right\rangle,$$

with $\left|x\right\rangle \in \mathcal{H}_{2^n}$ and $\left|y\right\rangle \in \mathcal{H}_{2^m}$.

II.5. Orcale

Example: for $f : \{0,1\}^n \longrightarrow \{0,1\}$, we can construct $\hat{U}_f$ as follow.



$$\hat{O}_f |x\rangle|y\rangle = (-1)^{f(x)} |x\rangle|y\rangle$$

$|y\rangle$ is an ancilla qubit.

For $|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, then

$$\boxed{\hat{O}_f |x\rangle \otimes |y\rangle = (-1)^{f(x)} |x\rangle \otimes |y\rangle}.$$

Then, forgetting the ancilla qubit, $\hat{U}_f |x\rangle = (-1)^{f(x)} |x\rangle$. In such a case, $\hat{U}_f$ is a **phase oracle**.

Demonstration:

$$\begin{aligned}\hat{O}_f |x\rangle \otimes |y\rangle &= \frac{1}{\sqrt{2}}\Big(\hat{O}_f |x\rangle \otimes |0\rangle - \hat{O}_f |x\rangle \otimes |1\rangle\Big), \\ &= \frac{1}{\sqrt{2}}\left(|x\rangle \otimes |0 \oplus f(x)\rangle - |x\rangle \otimes |1 \oplus f(x)\rangle\right),\end{aligned}$$

$$\hat{O}_f |x\rangle \otimes |y\rangle = \frac{1}{\sqrt{2}} \begin{cases} |x\rangle \otimes (|0\rangle - |1\rangle) & \text{if} \quad f(x) = 0 \\ |x\rangle \otimes (|1\rangle - |0\rangle) & \text{if} \quad f(x) = 1 \end{cases},$$

$$\hat{O}_f |x\rangle \otimes |y\rangle = \frac{1}{\sqrt{2}} \begin{cases} |x\rangle \otimes |y\rangle & \text{if} \quad f(x) = 0 \\ -|x\rangle \otimes |y\rangle & \text{if} \quad f(x) = 1 \end{cases},$$

such that

$$\boxed{\hat{O}_f |x\rangle \otimes |y\rangle = (-1)^{f(x)} |x\rangle \otimes |y\rangle}.$$

Let consider a function $f: \{0,1\} \longrightarrow \{0,1\}$. There is four possibilities for f

$$\underbrace{\left\{\begin{array}{l} f(0)=0 \\ f(1)=1 \end{array}\right.,}_{\text{identity}} \underbrace{\left\{\begin{array}{l} f(0)=1 \\ f(1)=0 \end{array}\right.,}_{\text{swap}} \underbrace{\left\{\begin{array}{l} f(0)=0 \\ f(1)=0 \end{array}\right., \left\{\begin{array}{l} f(0)=1 \\ f(1)=1 \end{array}\right.}_{\text{constant function}}.$$

The function f is said to be **balanced** if $f(0) \neq f(1)$.
The function f is said to be **constant** if $f(0) = f(1)$.

With a classical computer, one need to evaluate the function f twice to determine whether it is balanced or constant.

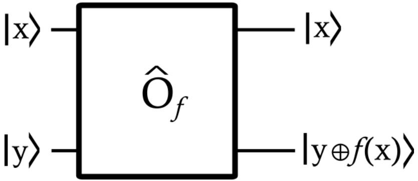
Example: f is constant : $f(0) = f(1) = 1$.

$$\hat{N}_f = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, \quad \hat{N}_f \left|0\right\rangle = \left|1\right\rangle, \quad \hat{N}_f \left|1\right\rangle = \left|1\right\rangle.$$

$\hat{N}_f$ is **not** unitary: $\hat{N}_f^\dagger \hat{N}_f \neq \mathbb{I}$. $\hat{N}_f$ does not preserve the norm and consequently the probability.

One constructs a quantum gate to realize f with a unitary operator $\hat{U}_f$

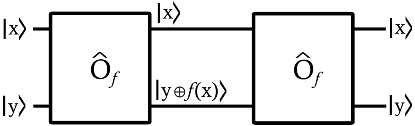
$$\hat{U}_f \left|x\right\rangle \otimes \left|y\right\rangle = \left|x\right\rangle \otimes \left|y \oplus f(x)\right\rangle.$$



$\left|x\right\rangle$ is the qubit on which one wants to evaluate the function f. $\left|y\right\rangle$ is a control qubit, allowing the operation to be unitary. $\oplus$ is a XOR operation.

$\hat{U}_f$ is reversible

$$\hat{U}_f{\dagger}\hat{U}_f = \mathbb{I}.$$



$$\hat{U}_f \hat{U}_f |x\rangle \otimes |y\rangle = \hat{U}_f |x\rangle \otimes |y \oplus f(x)\rangle = |x\rangle \otimes |(y \oplus f(x)) \oplus f(x)\rangle.$$

Or

$$\begin{aligned}(y \oplus f(x)) \oplus f(x) &= y \oplus (f(x) \oplus f(x)) \\ &= y \oplus 0 \\ &= y\end{aligned}$$

Remark:

$$\forall k \in \{0,1\},\ k \oplus k = 0.$$

$\hat{U}_f$ is unitary and might be used to evaluate f.

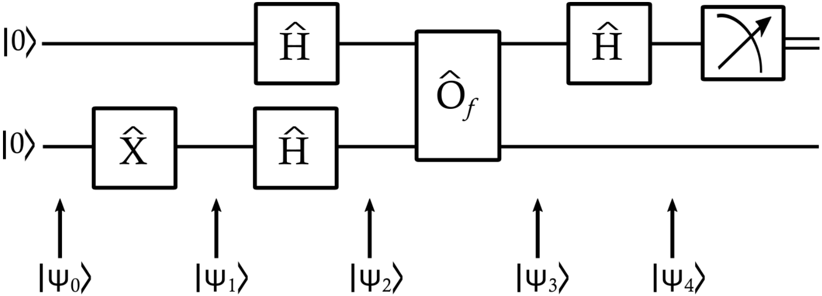
If the control qubit $|y\rangle = |0\rangle$, then

$$|x\rangle \otimes |y \oplus f(x)\rangle = |x\rangle \otimes |0 \oplus f(x)\rangle,$$

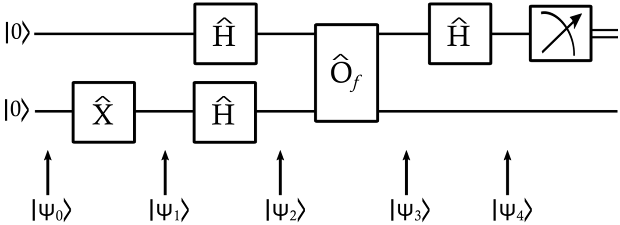
so that

$$|x\rangle \otimes |y \oplus f(x)\rangle = |x\rangle \otimes |f(x)\rangle.$$

The circuit diagram of the implementation of Deutsch-algorithm is the following



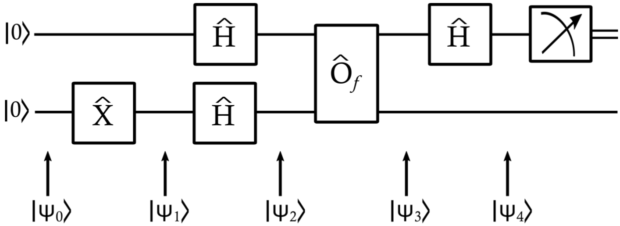
$$|\psi_4\rangle = \left(\hat{H}\otimes\mathbb{I}\right)\hat{U}_f\left(\hat{H}\otimes\hat{H}\right)\left(\mathbb{I}\otimes\hat{X}\right)|00\rangle.$$



$$|\psi_0\rangle = |00\rangle, \qquad |\psi_1\rangle = |01\rangle,$$

$$|\psi_2\rangle = \left(\hat{H}\otimes\hat{H}\right)|01\rangle = \frac{1}{2}\left(\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\otimes\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\right)\left(\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\right)$$

$$|\psi_2\rangle = \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) = \frac{|00\rangle-|01\rangle+|10\rangle-|11\rangle}{2}.$$



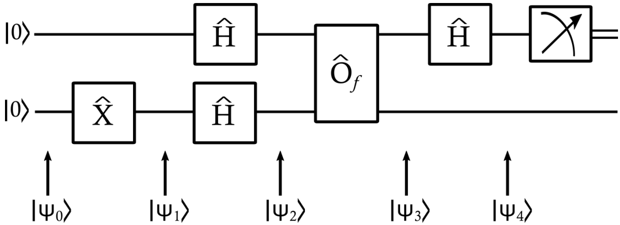
Oracle:

$$|\psi_3\rangle = \hat{U}_f\,|\psi_2\rangle = |x\rangle\left(\frac{|0\oplus f(x)\rangle - |1\oplus f(x)\rangle}{\sqrt{2}}\right).$$

For any state $|x\rangle$ in a pure state.

If $f(x) = 0,\ |\psi_3\rangle = |x\rangle\left(\frac{|0\oplus 0\rangle - |1\oplus 0\rangle}{\sqrt{2}}\right) = |x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right),$

If $f(x) = 1,\ |\psi_3\rangle = |x\rangle\left(\frac{|0\oplus 1\rangle - |1\oplus 1\rangle}{\sqrt{2}}\right) = |x\rangle\left(\frac{|1\rangle - |0\rangle}{\sqrt{2}}\right).$

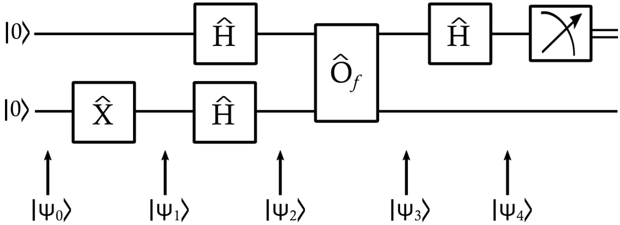


So finally

$$\boxed{|\psi_3\rangle = (-1)^{f(x)}\,|x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)}.$$

$$\text{If } |x\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \text{ then}$$

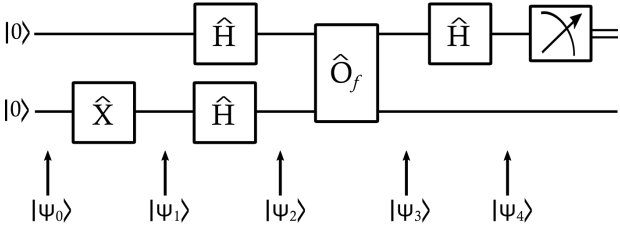
$$|\psi_3\rangle = \hat{U}_f \frac{|0\rangle}{\sqrt{2}}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) + \hat{U}_f \frac{|1\rangle}{\sqrt{2}}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right),$$



$$|\psi_3\rangle = \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

If f is constant: $|\psi_3\rangle = (\pm 1)\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$

If f is balanced: $|\psi_3\rangle = (\pm 1)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$



Applying finally an Hadamard gate on each qubit

$$|\psi_3\rangle = \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

If f is constant: $|\psi_4\rangle = (\pm 1)|0\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$

If f is balanced: $|\psi_4\rangle = (\pm 1)|1\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$

$$\text{If } f \text{ is constant: } \left|\psi_4\right\rangle = (\pm 1)\left|0\right\rangle \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right).$$

$$\text{If } f \text{ is balanced: } \left|\psi_4\right\rangle = (\pm 1)\left|1\right\rangle \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right).$$

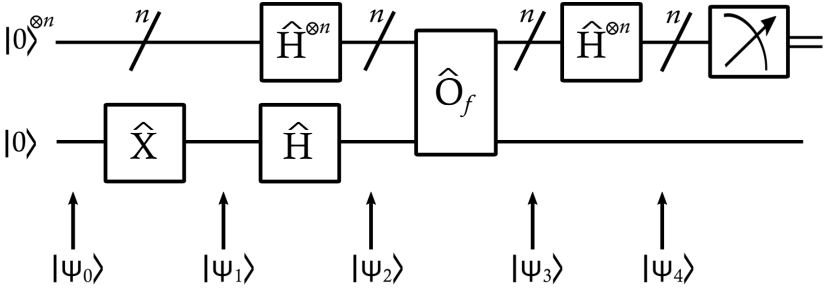
The measurement of the first qubit permits to determine unambiguously if the function f is balanced or constant. The Deutsch problem consists in determining whether a function $f : \{0, 1\} \longrightarrow \{0, 1\}$ is balanced or constant. A classical computer requires two solicitation of the Oracle, while a quantum algorithm requires only one solicitation of the oracle to get the result. That's a direct consequence of the quantum parallelism.

III.3. Deutsch-Josa algorithm

In the case of Deutsch-Josa algorithm, a more general case is considered with a function $f: \{0,1\}^n \longrightarrow \{0,1\}$.

- f is balanced if half of the inputs return 0 and the others return 1.
- f is constant if f only returns 0 or 1.

The Deutsch-Josa algorithm is based on the same principle than the Deutsch algorithm (case $n = 1$), with the following implementation



Classical solution: we need to ask the Oracle at least twice, but if we get twice the same value, we need to ask again... corresponding to at most $\frac{N}{2} + 1 = 2^{n-1} + 1$ queries of the Oracle, with n the number of input bits and $N = 2^n$ the number of realizable bit string.

The quantum solution with the Deutsch-Josa algorithm needs only one query !!!

Proof:

Initial state:

$$|\psi_0\rangle = |0\rangle^{\otimes n}|0\rangle = |000\cdots 00\rangle|0\rangle.$$

Preparation of the ancilla qubit with a $\hat{X}$ gate:

$$|\psi_1\rangle = |0\rangle^{\otimes n}|1\rangle.$$

Hadamard gate on the quantum register input:

$$|\psi_2\rangle = \left(\hat{H}^{\otimes n}|0\rangle^{\otimes n}\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

And

$$\hat{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{\langle x|0\rangle} \; |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle,$$

so that $|\psi_2\rangle$ is a superposition of all states as follow

$$\boxed{|\psi_2\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)}.$$

Oracle:

$$|\psi_3\rangle = \hat{U}_f\,|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes \left(\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} \right),$$

$$\boxed{|\psi_3\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}\,|x\rangle \right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)}.$$

Hadamard gate on the quantum register:

$$|\psi_4\rangle = \left(\hat{H}^{\otimes n}\left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle\right)\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

$$|\psi_4\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}\hat{H}^{\otimes n}|x\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

$$|\psi_4\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}\frac{1}{\sqrt{2^n}}\sum_{K\in\{0,1\}^n}(-1)^{\langle K|x\rangle}\;|K\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

$$|\psi_4\rangle = \left(\sum_{K\in\{0,1\}^n}\left(\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)+\langle K|x\rangle}\right)|K\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

Let define

$$C_K = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x) + \langle K | x \rangle} \ , \quad \text{and} \quad |\phi\rangle = \sum_{K \in \{0,1\}^n} C_K \, |K\rangle.$$

Then

$$|\psi_4\rangle = |\phi\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right).$$

The state $|\phi\rangle$ is measured at the end: probability to measure the string $|000 \cdots 000\rangle$

$$P(y = 00 \cdots 00) = |\langle 00 \cdots 00 | \phi \rangle|^2 .$$

$$
\begin{aligned}
P(y = 00\cdots00) &= \left| \sum_{K\in\{0,1\}^n} C_K \langle 00\cdots00|K\rangle \right|^2 = \left|C_{00\cdots00}\right|^2, \\
&= \left| \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \right|^2
\end{aligned}
$$

Or

$$
\sum_{x\in\{0,1\}^n} (-1)^{f(x)} = \begin{cases} +2^n & \text{if} \quad f(x) = 0 \\ -2^n & \text{if} \quad f(x) = 1 \\ 0 & \text{if} \quad f(x) \text{ is balanced} \end{cases},
$$

then

$$
P(y = 00\cdots00) = \begin{cases} 1 & \text{if} \quad f(x) \text{ is constant} \\ 0 & \text{if} \quad f(x) \text{ is balanced} \end{cases}
$$

So

$$P(y = 00\cdots00) = 1 \;\text{ for a balanced function,}$$
$$P(y = 00\cdots00) = 0 \;\text{ for a constant function.}$$

If the function is neither balanced nor constant, then

$$P(y = 00\cdots00) \in \,]\,0, 1\,[\;.$$