

# Classical Oracle

## Episode 26. Classical Oracle

Episode 27. Quantum Oracle: Definition

Episode 28. Quantum Oracle: Properties

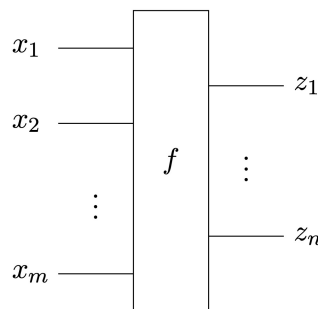
---

## Definition

An oracle (more specifically, *classical oracle*, to be distinguished from quantum oracle) is typically described by a binary function

$$f : \{0, 1\}^m \rightarrow \{0, 1\}^n,$$

that maps an  $m$ -bit input to an  $n$ -bit output. Bear in mind that the function  $f(x)$  may not be invertible in general.



**Figure 1.** A circuit diagram of classical oracle  $f : \{0, 1\}^m \rightarrow \{0, 1\}^n$ .  $x$  are an  $m$ -bit string,  $x \in \{0, 1\}^m$ , and  $z$  denotes the image of  $f$  at  $x$ ,  $z = f(x) \in \{0, 1\}^n$ .

## Example

Let the number of input and output bits.

```
In[*]:= $m = 3;  
          $n = 2;
```

Now, define the function  $f : \{0, 1\}^m \rightarrow \{0, 1\}^n$  properly. Q3 allows *two ways* for your convenience.

1. In the form of  $f[\{c_1, c_2, \dots, c_m\}] = \{t_1, t_2, \dots, t_n\}$

```
In[*]:= f[{0, 0, 1}] = f[{0, 1, 0}] = {1, 1};  
        f[{1, 1, 1}] = {1, 0};  
        f[{} ] = {0, 0};
```

```

In[*]:= in = Tuples[{0, 1}, $m];
out = f /@ in;

In[*]:= Thread[in → out] // TableForm
Out[*]//TableForm=
{0, 0, 0} → {0, 0}
{0, 0, 1} → {1, 1}
{0, 1, 0} → {1, 1}
{0, 1, 1} → {0, 0}
{1, 0, 0} → {0, 0}
{1, 0, 1} → {0, 0}
{1, 1, 0} → {0, 0}
{1, 1, 1} → {1, 0}

```

2. In the form  $f[c] = t$ , where  $c := (c_1 c_2 \dots c_m)_2$  and  $t := (t_1 t_2 \dots t_n)_2$ .

```

In[*]:= f[1] = f[2] = 3;
f[7] = 2;
f[_Integer] = 0;

In[*]:= in = Range[2^$m] - 1;
out = f /@ in;
Thread[in → out] // TableForm
Out[*]//TableForm=
0 → 0
1 → 3
2 → 3
3 → 0
4 → 0
5 → 0
6 → 0
7 → 2

```

## Unified Form

```

In[*]:= ff = Oracle[f, $m, $n]
Out[*]=
Oracle[f, 3, 2]

In[*]:= xx = Tuples[{0, 1}, $m]
Out[*]=
{{0, 0, 0}, {0, 0, 1}, {0, 1, 0}, {0, 1, 1}, {1, 0, 0}, {1, 0, 1}, {1, 1, 0}, {1, 1, 1}}

In[*]:= zz = ff /@ xx
Out[*]=
{{0, 0}, {1, 1}, {1, 1}, {0, 0}, {0, 0}, {0, 0}, {0, 0}, {1, 0}}

```

```

In[*]:= Thread[xx → zz] // TableForm
Out[*]//TableForm=
  {0, 0, 0} → {0, 0}
  {0, 0, 1} → {1, 1}
  {0, 1, 0} → {1, 1}
  {0, 1, 1} → {0, 0}
  {1, 0, 0} → {0, 0}
  {1, 0, 1} → {0, 0}
  {1, 1, 0} → {0, 0}
  {1, 1, 1} → {1, 0}

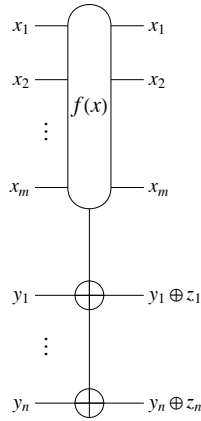
```

## Reversible Version

The reversible version of classical oracle  $f$  is an extended mapping,  $\{0, 1\}^{m+n} \rightarrow \{0, 1\}^{m+n}$ , defined by the association

$$(x, y) \mapsto (x, f(x) \oplus y),$$

where  $x \in \{0, 1\}^m$  and  $y \in \{0, 1\}^n$  are bit strings of the  $m$ -bit native register and the  $n$ -bit auxiliary register, respectively.



**Figure 2.** A reversible version of classical oracle  $f : \{0, 1\}^m \rightarrow \{0, 1\}^n$ .  $x \in \{0, 1\}^m$  and  $y \in \{0, 1\}^n$  are  $m$ -bit and  $n$ -bit strings, respectively, and  $z = f(x) \in \{0, 1\}^n$  denotes the image of  $f$  at  $x$ .

**Theorem:** Although the function  $f(x)$  itself may not be invertible, the extended mapping is always one-to-one regardless of the function  $f(x)$ .

## Why Oracle?

- An *oracle* in computer science is a “black box” entity that is able to produce a solution for any instance of a given problem without revealing how the solution was obtained.
- The details of its implementation and its complexity are disregarded.

- Oracles are commonly used in computer science to theoretically classify problems in computational complexity theory, to analyze the complexity of algorithms, or to provide a way to solve problems.
- The concept of an oracle provides a powerful tool in many other sciences as well, enabling us to concentrate on what a device does without worrying about how it is accomplished.
- In a *decision problem*, we are supposed to figure out the unknown property by making queries to the oracle.

---

## Summary

### Keywords

- Oracle
- Reversible computing
- Decision making

### Functions

- Oracle
- ControlledExp

### Related Links

- Section 4.2 of the Quantum Workbook (2022, 2023).
- Tutorial: Quantum Oracle
- Tutorial: Quantum Decision Algorithms