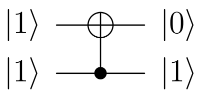
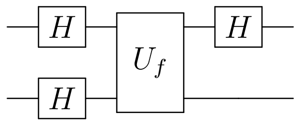
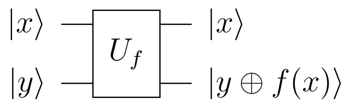


Deutsch-Josza Oracles

We're interested in the following setup:



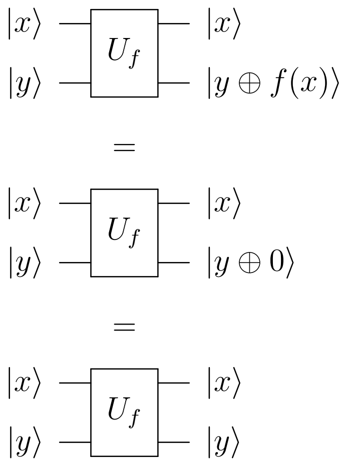
U_f is assumed to be a gate that acts as an $f(x)$-register targeted on the second qubit:



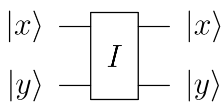
Assumption - f is either **constant** or **balanced**.

$\boxed{f \textbf{ constant}:}$

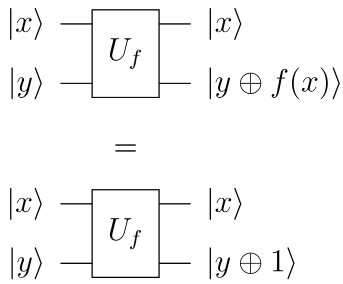
$f(x) = 0$



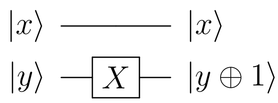
U_f is just the **identity** gate.



$f(x) = 1$



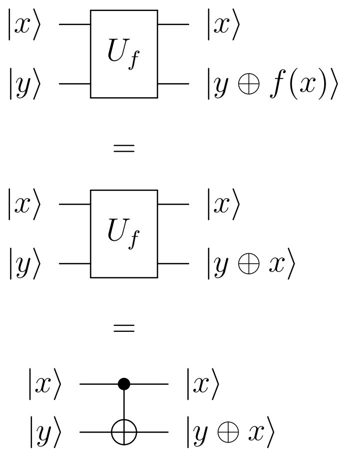
U_f is just the X gate applied to the second qubit.



f **balanced**

$$f(x) = x$$

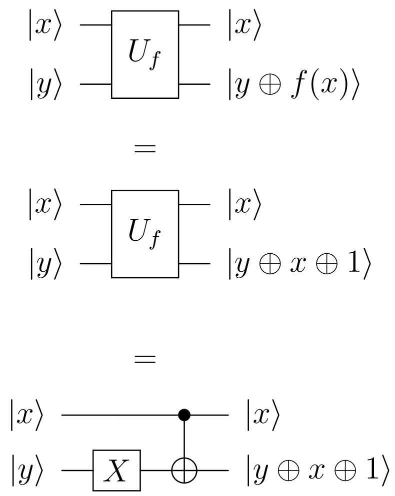
$$f(0) = 0 \qquad f(1) = 1$$



CNOT

$$f(x) = 1 \oplus x$$

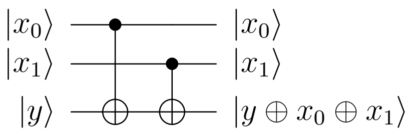
$$f(0) = 1 \qquad f(1) = 0$$



CNOT$\cdot X_2$

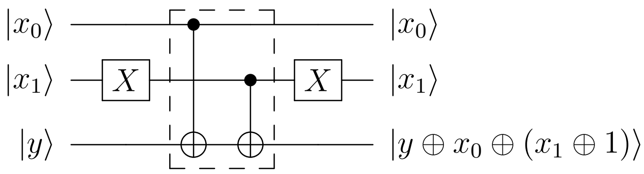
Let's create oracles corresponding to balanced functions.

1. Apply **CNOT** to each of the query qubits with target the register qubit.
   [Note: Qiskit basis when applicable.]



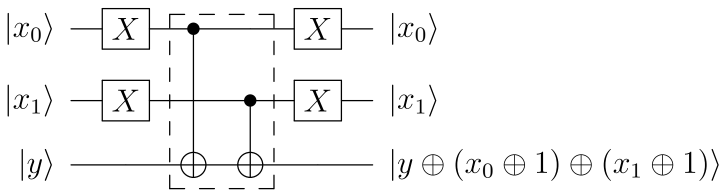
$f(x) = 0$	$f(x) = 1$
00	01
11	10

2. Sandwich the gate with **X** on a qubit to swap items of the table

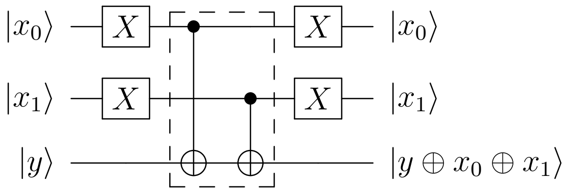


$f(x) = 0$	$f(x) = 1$
01	00
10	11

3. Sandwich with another pair of X's:



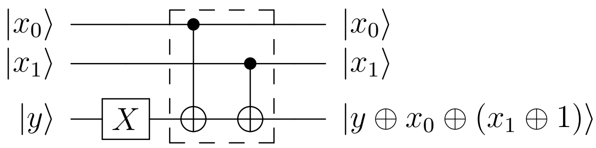
$=$



$f(x) = 0$	$f(x) = 1$
00	01
11	10

(Same function of step 1!)

4. Preappend gate X to target qubit (to circuit of step 1):



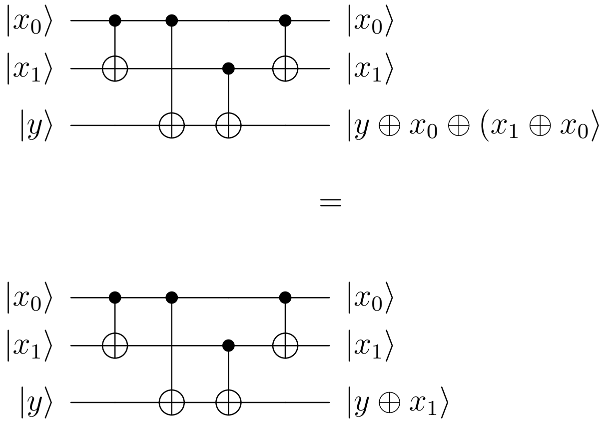
$f(x) = 0$	$f(x) = 1$
01	00
10	11

(Same function of Step 2!!!)

It's clear what's going on here: putting an X gate anywhere (then reapplying on the right to return query qubits to initial states) switches the columns of the table. This is because we're simply adding 1 to the parity of the register. If we want to add 1 to the register, from now on we only have to add a single **X** gate on the register qubit to acheive the same result.

The trick is to place other **CNOT** gate with target within the query qubits.

5. Starting with gate from Step 1, sandwich the two query qubits with **CNOT**s.

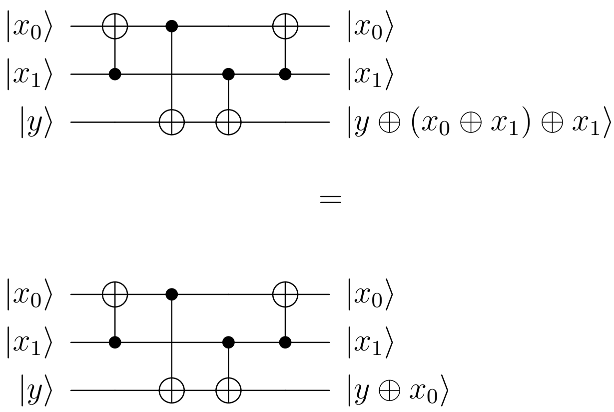


$f(x) = 0$	$f(x) = 1$
00	10
01	11

A new one!

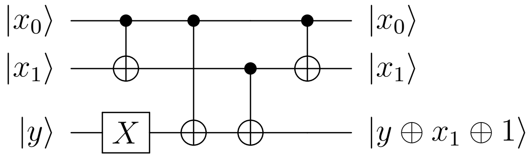
6. Let's try one of those again, with upside-down **CNOT**.

   We should expect to see $|y \oplus x_0\rangle$



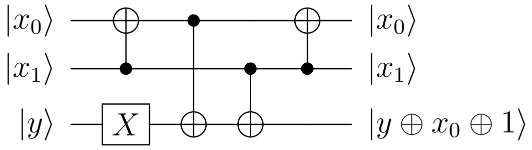
$f(x) = 0$	$f(x) = 1$
00	01
10	11

7. If we append **X** to each of these, we'll have counted all of the possible situations.



$f(x) = 0$	$f(x) = 1$
01	00
11	01

8. Now to the circuit with the upside-down **CNOT** from step 6:



$f(x) = 0$	$f(x) = 1$
01	00
11	10

You can "balance" any function in the following sense, essentially by adding a new linear term.

Proposition. *Let* $f : \{0,1\}^n \to \{0,1\}$ *be* ***any*** *function in the variables* $x_1, ..., x_n$*. Then the function*

$$g(x_1, ..., x_n, x_{n+1}) := f(x_1, ..., x_n) \oplus x_{n+1}$$

is balanced. (Adding by a new variable independent from those in the argument of f*)*

Proof. Write out the function values for f in terms of a truth-table in the variables $x_1, ..., x_n$, with a column for $f(x_1, ..., x_n)$. Introducing a new variable x_{n+1} has the effect of doubling the rows of the truth table, and in the $g(x)$ column you'll have a copy of $f(x)$ filling half the column, and the other half of the column displaying $f(x) \oplus 1$. $\square$