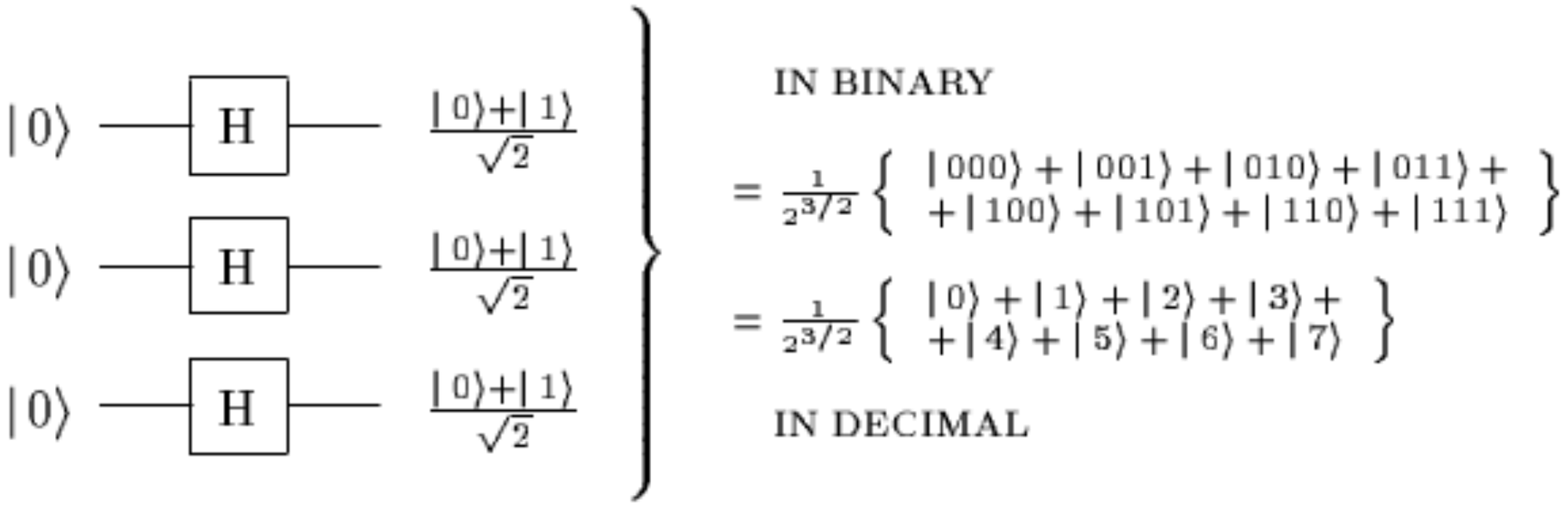
# Creating a uniform superpositions with Hadamard Gates



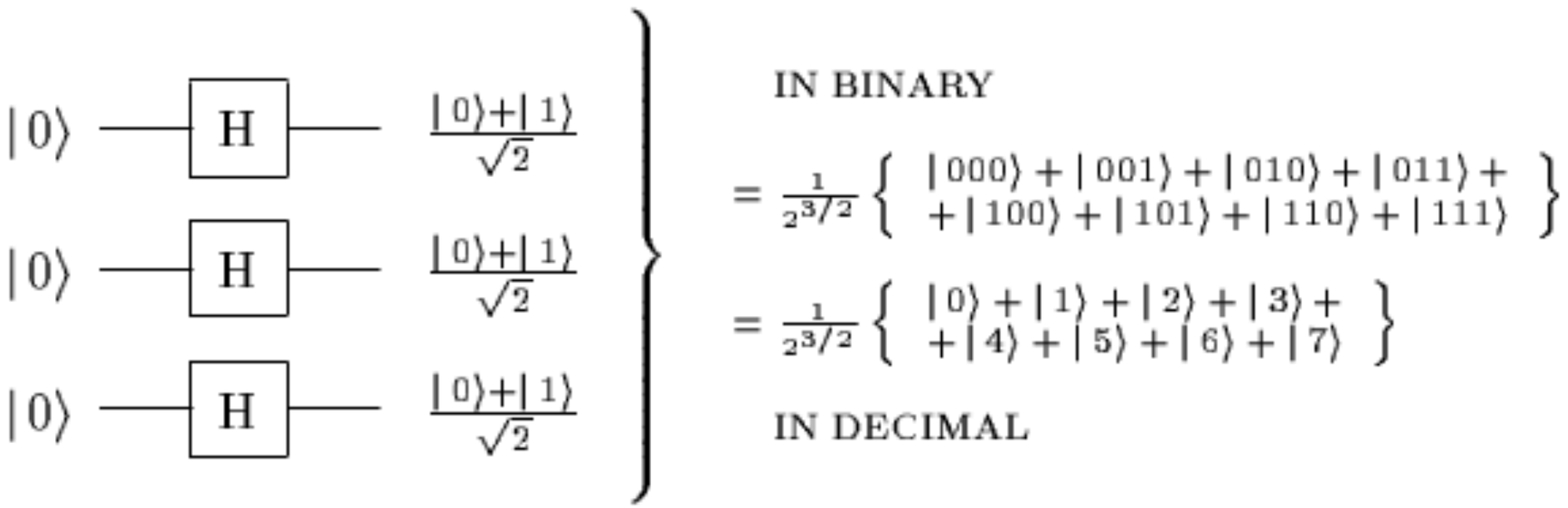

IN BINARY

$$= \frac{1}{2^{3/2}} \left\{ \begin{array}{l} |000\rangle + |001\rangle + |010\rangle + |011\rangle + \\ +|100\rangle + |101\rangle + |110\rangle + |111\rangle \end{array} \right\}$$

$$= \frac{1}{2^{3/2}} \left\{ \begin{array}{l} |0\rangle + |1\rangle + |2\rangle + |3\rangle + \\ +|4\rangle + |5\rangle + |6\rangle + |7\rangle \end{array} \right\}$$

IN DECIMAL

# Creating a uniform superpositions with Hadamard Gates




n qubit computational qubit basis:

$$|x_1\rangle \otimes |x_2\rangle \otimes |x_3\rangle \otimes \cdots \otimes |x_n\rangle$$

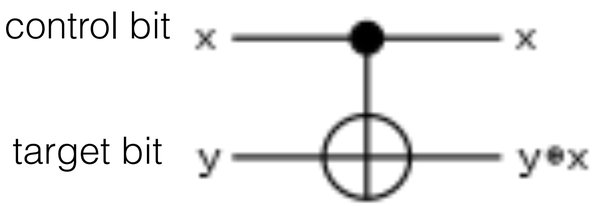
$|x\rangle \quad x \in \{0,1\}^n$ n bit string

$$|x\rangle \quad x \in \{0,1,2,\ldots,2^n-1\}$$

n qubit Hadamard:

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{y \cdot x} |y\rangle$$

# 2-qubit quantum gates




| input | | output | |
|---|---|---|---|
| x | y | x | y+x |
| \|0⟩ | \|0⟩ | \|0⟩ | \|0⟩ |
| \|0⟩ | \|1⟩ | \|0⟩ | \|1⟩ |
| \|1⟩ | \|0⟩ | \|1⟩ | \|1⟩ |
| \|1⟩ | \|1⟩ | \|1⟩ | \|0⟩ |

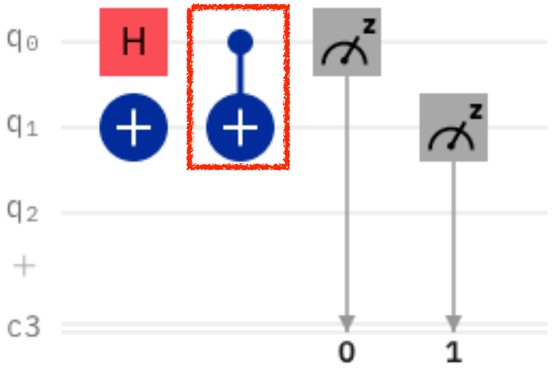
1. 2-qubit gate example: *controlled-NOT* or *CNOT*
   with the basis-state ordering $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, it is represented by

$$U_{\mathrm{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

   it could be represented by a 'classical' truth table

2. 1-qubit gates together with CNOT form a unversal q-gate set

q-circuit



# The problem

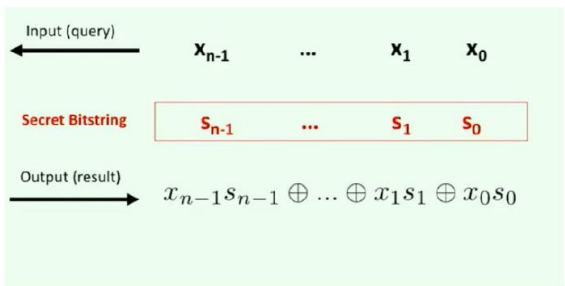
1. Secret bitstring $s$ e.g. $s = 1011$
2. We have an oracle that implements $f(x) = x.s$
   Dot product:

$$x.s = x_1 \cdot s_1 + x_2 \cdot s_2 + \ldots + x_n \cdot s_n \mod 2$$

$$\text{e.g. } x = 0101 \rightarrow f(x) = x.s = 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1$$

3. Task: find the secret bitstring $s$ using as few queries as possible!

Oracle



# Classical solution

1. Best solution: evaluate the function for strings

$$x = 1000, 0100, 0010, 0001$$

$$f(1000) = 1 \cdot s_1 + 0 \cdot s_2 + 0 \cdot s_3 + 0 \cdot s_4 = s_1$$

$$f(0100) = 0 \cdot s_1 + 1 \cdot s_2 + 0 \cdot s_3 + 0 \cdot s_4 = s_2$$

   etc.

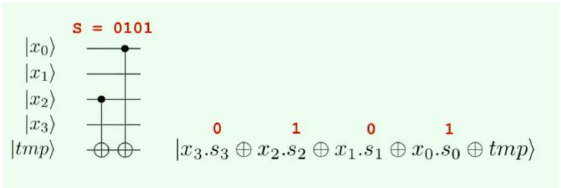
2. number of evaluations needed = the length of the secret bitstring

# Quantum Oracle

1. We need a *quantum* oracle that implements $f(x) = x.s$
2. input superposition $\rightarrow$ output superposition

**The Quantum Oracle**


**Implementing the Oracle with a Circuit**



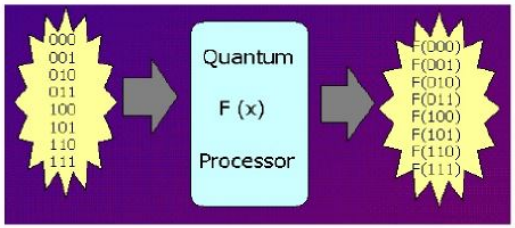

3. Quantum oracle can be implemented with CNOT gates
4. Reversibility requirement - ancilla (tmp) qubit needed

# We can do better with quantum circuits

classical: *n* runs needed
quantum: a single run is enough due to superpositions!

**The general hope of quantum computing**

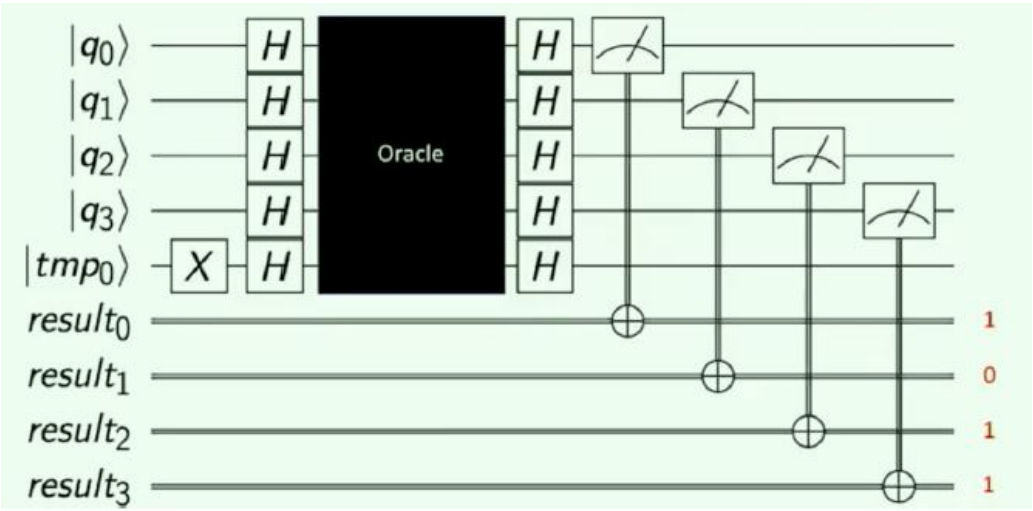

**the (naive) quantum parallelism**




$$\sum_{x=0}^{2^n-1} |x\rangle\,|y\rangle \longrightarrow \sum_{x=0}^{2^n-1} |x\rangle\,|f(x) \oplus y\rangle$$

# Quantum solution

1. First create an equal superposition of all states $x$ - Hadamard gates
2. Then apply the oracle to the superposition - all the $x$'s get evaluated in a single run!
3. Finally apply another set of Hadamards
4. Get the secret bitstring as output



# How does this work?