Amsterdam University of Applied Sciences

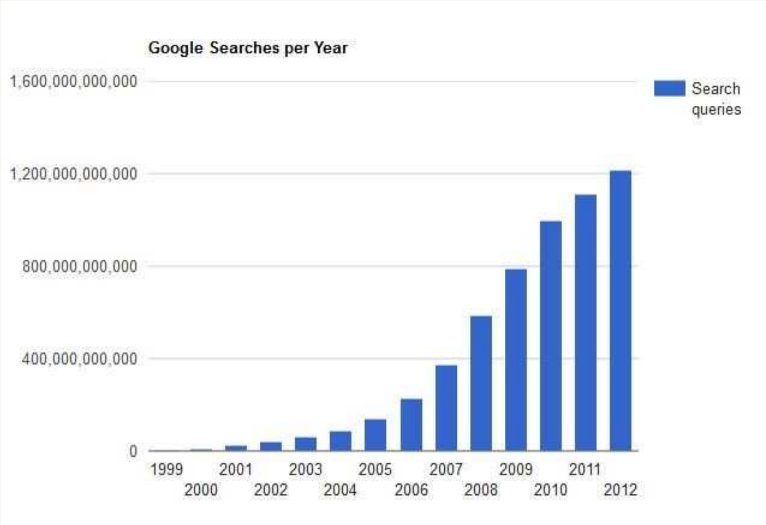
Quantum Search: Grover's algorithm

Quantum Capita Selecta

Bernardo Villalba Frías, PhD

b.r.villalba.frias@hva.nl

Searching



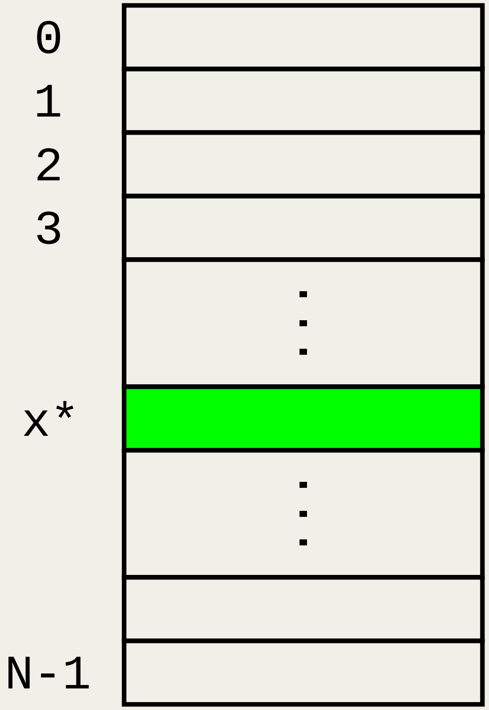
- We search a lot:
  - 82166 Google searches every second
  - 3.5 billion searches per day
- Structured search:
  - Based on the PageRank algorithm
  - Search data structures to improve performance

Searching

- Unstructured search:
  - Table with N entries
  - Special item: x^*
  - Find x^* (assuming 1 solution)
- Consider the n–to–1 bit function:

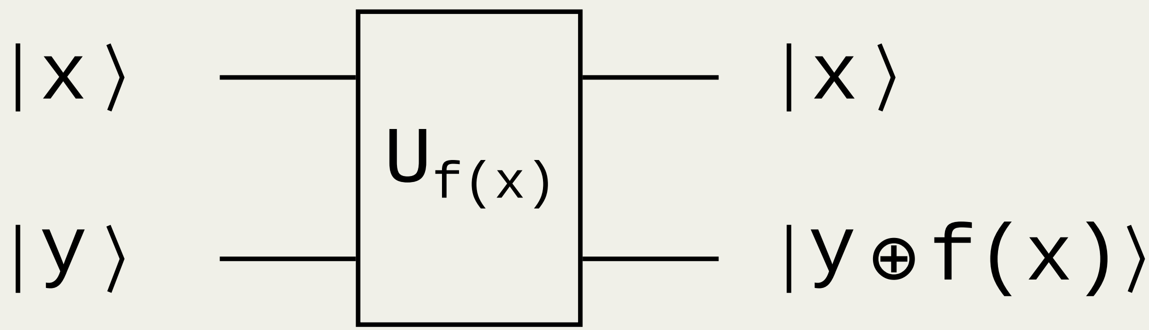
$$f(x) = \begin{cases} 1 & \text{if } x = x^* \\ 0 & \text{if } x \neq x^* \end{cases}$$

- In case $n = 2$:
  - $N = 2^n = 4$
  - Expected function calls: 2.25



Oracle

- Model for the interaction with the database
- Implements a function $f(x)$
  - $f(x)$ encodes information about the item x^*
- Query the oracle to fetch results from the function
- First possible implementation:
  - Add an extra (ancillary) qubit

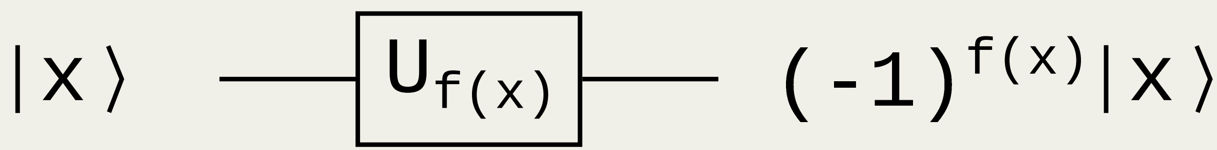


Oracle

Amsterdam University of Applied Sciences

- Second possible implementation:
  - Flip the input phase if and only if $f(x) = 1$

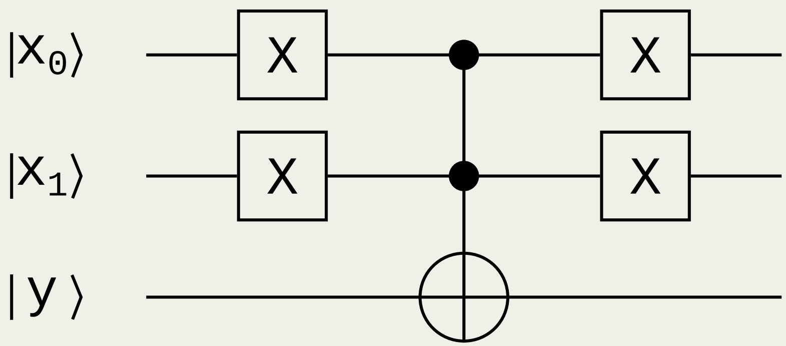
$$(-1)^{f(x)} |x\rangle = \begin{cases} |x\rangle & \text{if } f(x) = 0 \\ -|x\rangle & \text{if } f(x) = 1 \end{cases}$$

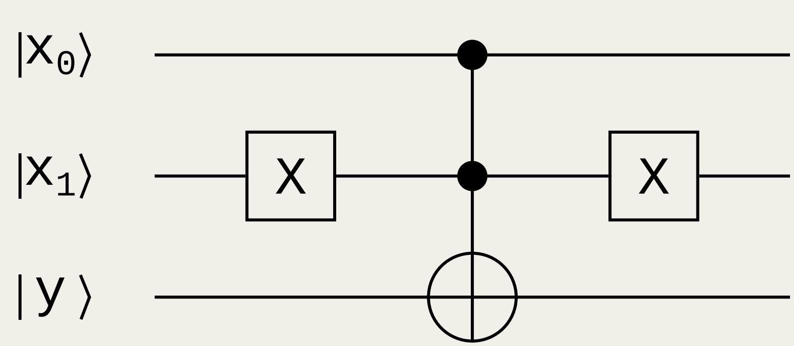


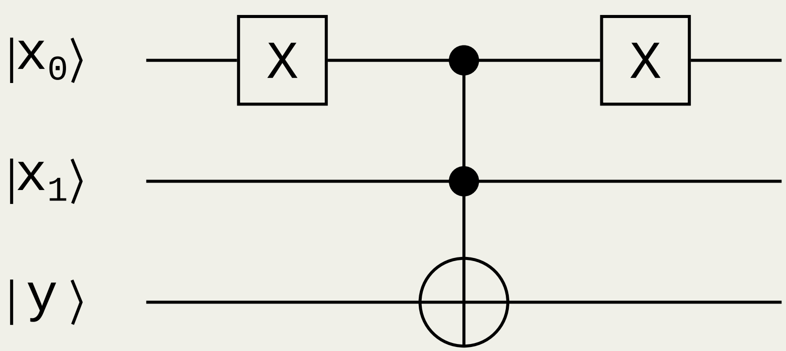
- Each implementation has its own particularities
- Solution is marked with a phase of -1

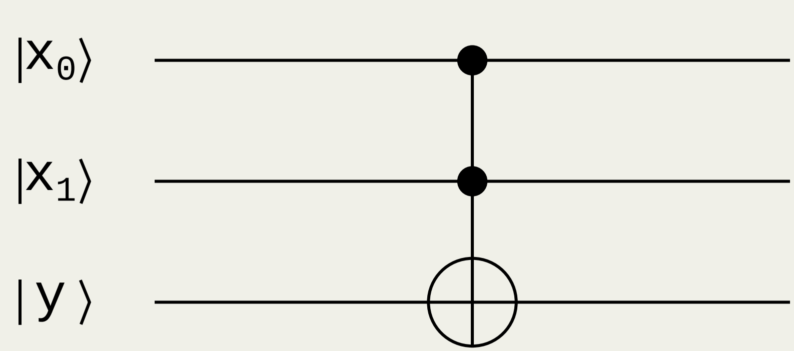
Oracle ($n = 2$)

- Using ancillary qubits





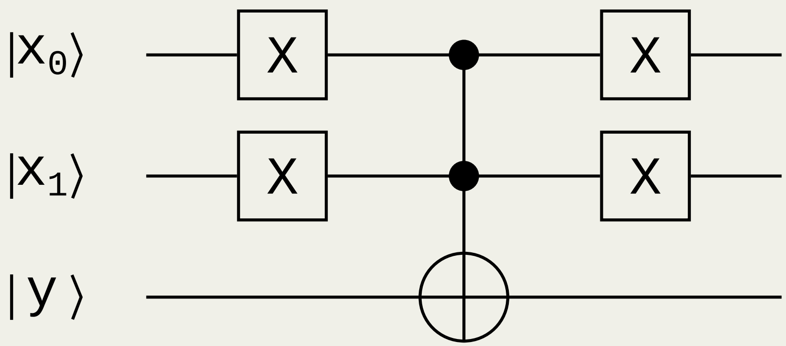




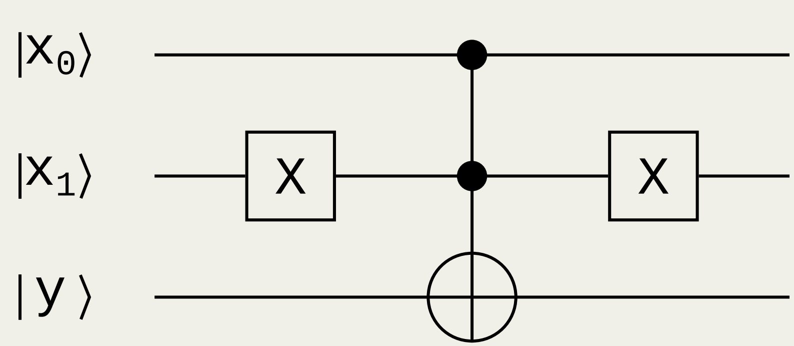
Oracle ($n = 2$)

- Using ancillary qubits

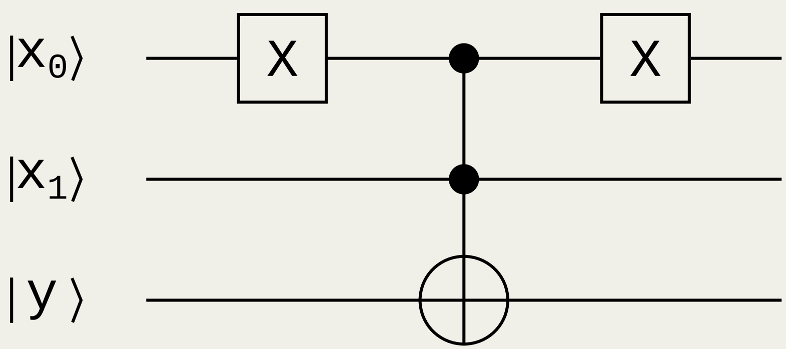
$x^* = 00$



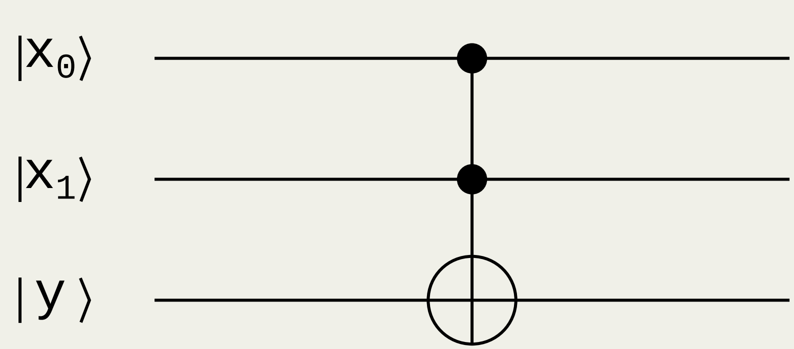
$x^* = 01$



$x^* = 10$



$x^* = 11$

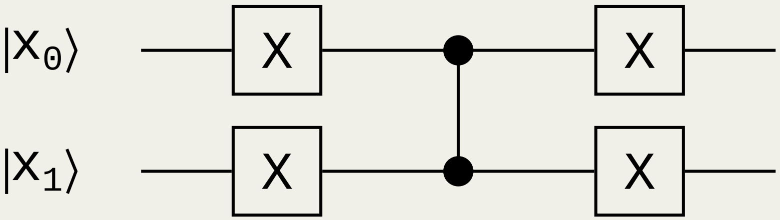


Oracle ($n = 2$)

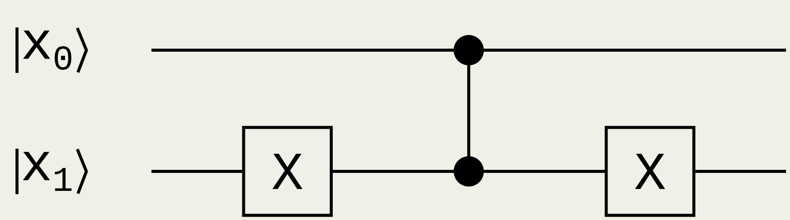
Amsterdam University of Applied Sciences

- Flipping phase

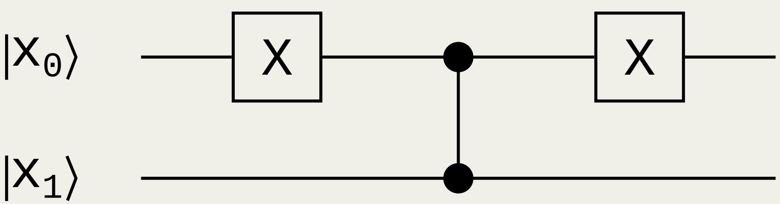
$x^* = 00$



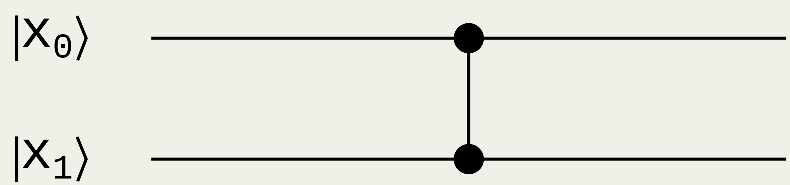
$x^* = 01$



$x^* = 10$

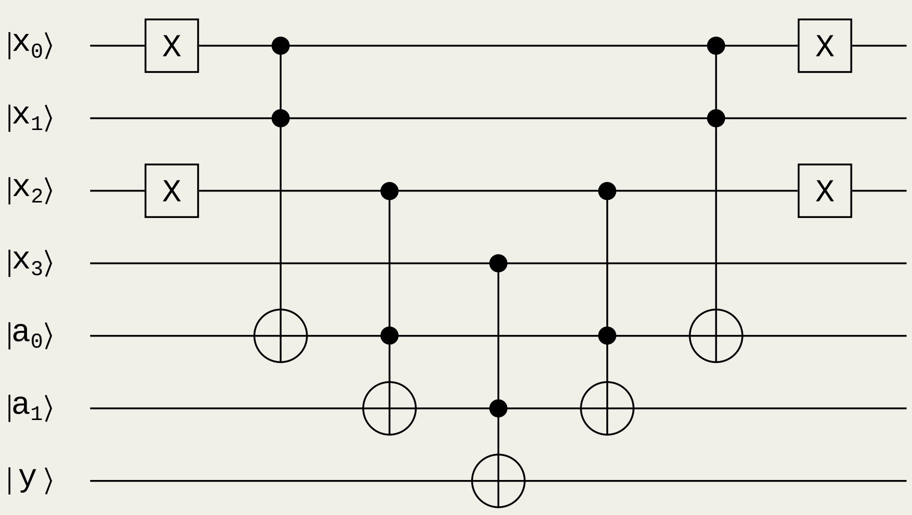


$x^* = 11$



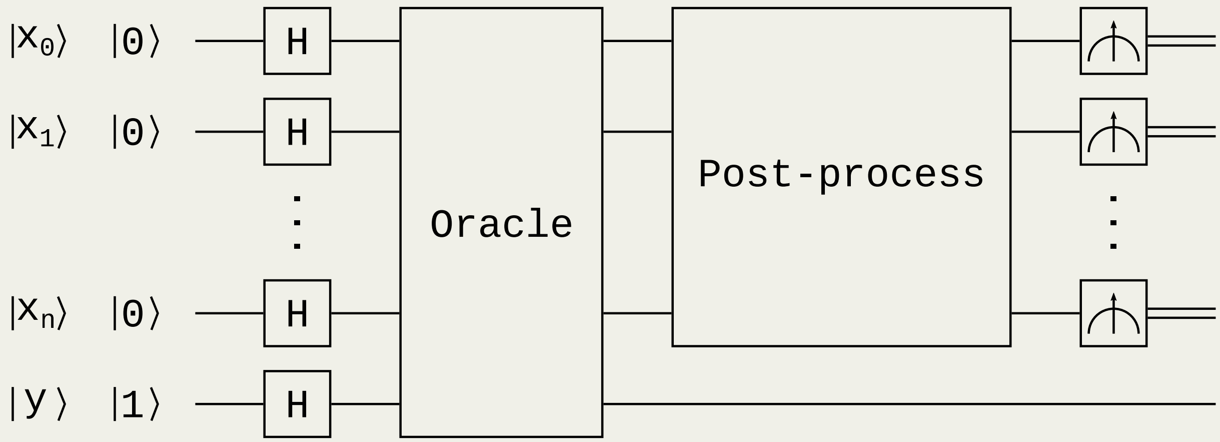
Oracle ($n > 2$)

- What about $n > 2$?
- Is there a n–qubit Toffoli gate?
  - Of course, because everything holds up on paper...
  - ... but, in Qiskit?
- Chaining some Toffoli gates
  - Using ancilla qubits to store intermediate results



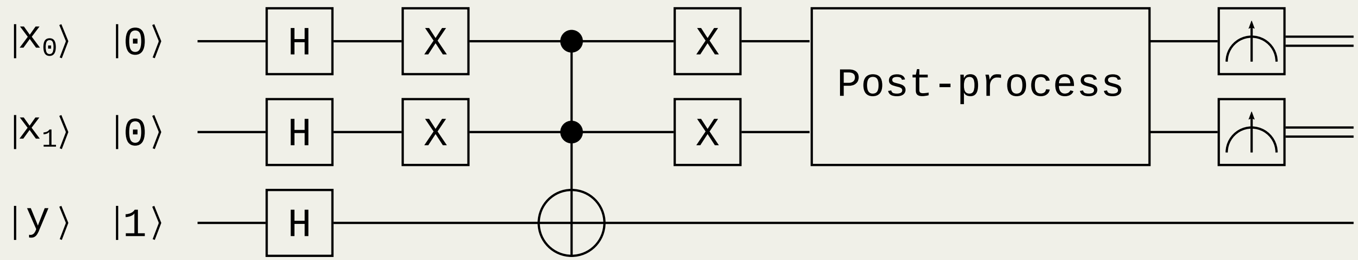
Grover's algorithm

Amsterdam University of Applied Sciences



1. Prepare input state (equal superposition state)
2. Apply oracle (phase–kickback)
3. Post–process (inversion around mean)
4. Measure

Grover's algorithm



$$\rightarrow |1\rangle \otimes |0\rangle \otimes |0\rangle$$

$$\rightarrow |1\rangle \otimes |00\rangle$$

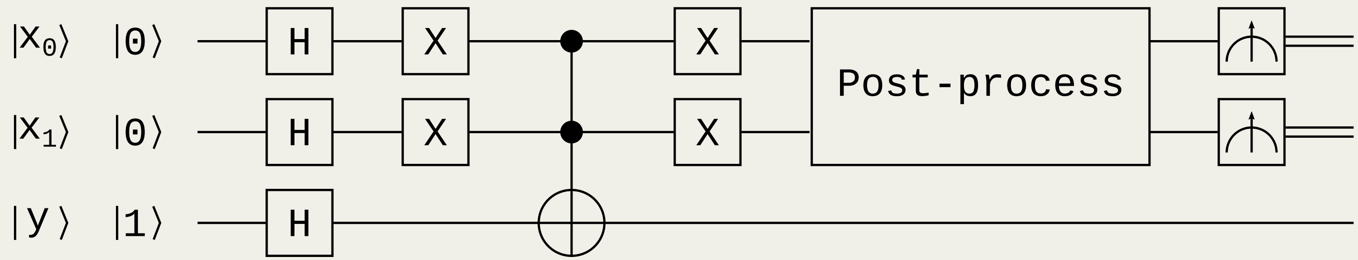
$$\rightarrow |100\rangle$$

$$H_{x_0} \rightarrow \frac{1}{\sqrt{2}} \left(|100\rangle + |101\rangle \right)$$

$$H_{x_1} \rightarrow \frac{1}{2} \left(|100\rangle + |110\rangle + |101\rangle + |111\rangle \right)$$

$$H_y \rightarrow \frac{1}{2\sqrt{2}} \left(|000\rangle - |100\rangle + |010\rangle - |110\rangle + |001\rangle - |101\rangle + |011\rangle - |111\rangle \right)$$

Grover's algorithm



$$\rightarrow \frac{1}{2\sqrt{2}}\left(|000\rangle - |100\rangle + |010\rangle - |110\rangle + |001\rangle - |101\rangle + |011\rangle - |111\rangle\right)$$

$$X_{x_0} \rightarrow \frac{1}{2\sqrt{2}}\left(|001\rangle - |101\rangle + |011\rangle - |111\rangle + |000\rangle - |100\rangle + |010\rangle - |110\rangle\right)$$

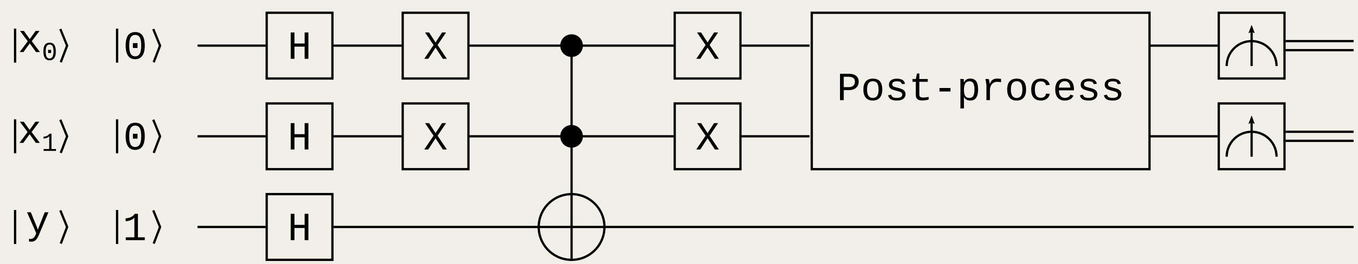
$$X_{x_1} \rightarrow \frac{1}{2\sqrt{2}}\left(|011\rangle - |111\rangle + |001\rangle - |101\rangle + |010\rangle - |110\rangle + |000\rangle - |100\rangle\right)$$

$$\mathit{CCNOT}_{x_0, x_1, y} \rightarrow \frac{1}{2\sqrt{2}}\left(|111\rangle - |011\rangle + |001\rangle - |101\rangle + |010\rangle - |110\rangle + |000\rangle - |100\rangle\right)$$

$$X_{x_0} \rightarrow \frac{1}{2\sqrt{2}}\left(|110\rangle - |010\rangle + |000\rangle - |100\rangle + |011\rangle - |111\rangle + |001\rangle - |101\rangle\right)$$

$$X_{x_1} \rightarrow \frac{1}{2\sqrt{2}}\left(|100\rangle - |000\rangle + |010\rangle - |110\rangle + |001\rangle - |101\rangle + |011\rangle - |111\rangle\right)$$

Grover's algorithm



$$\rightarrow \frac{1}{2\sqrt{2}}\left(|100\rangle - |000\rangle + |010\rangle - |110\rangle + |001\rangle - |101\rangle + |011\rangle - |111\rangle\right)$$

$$\rightarrow \frac{1}{2\sqrt{2}}\left(- |000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle - |101\rangle - |110\rangle - |111\rangle\right)$$

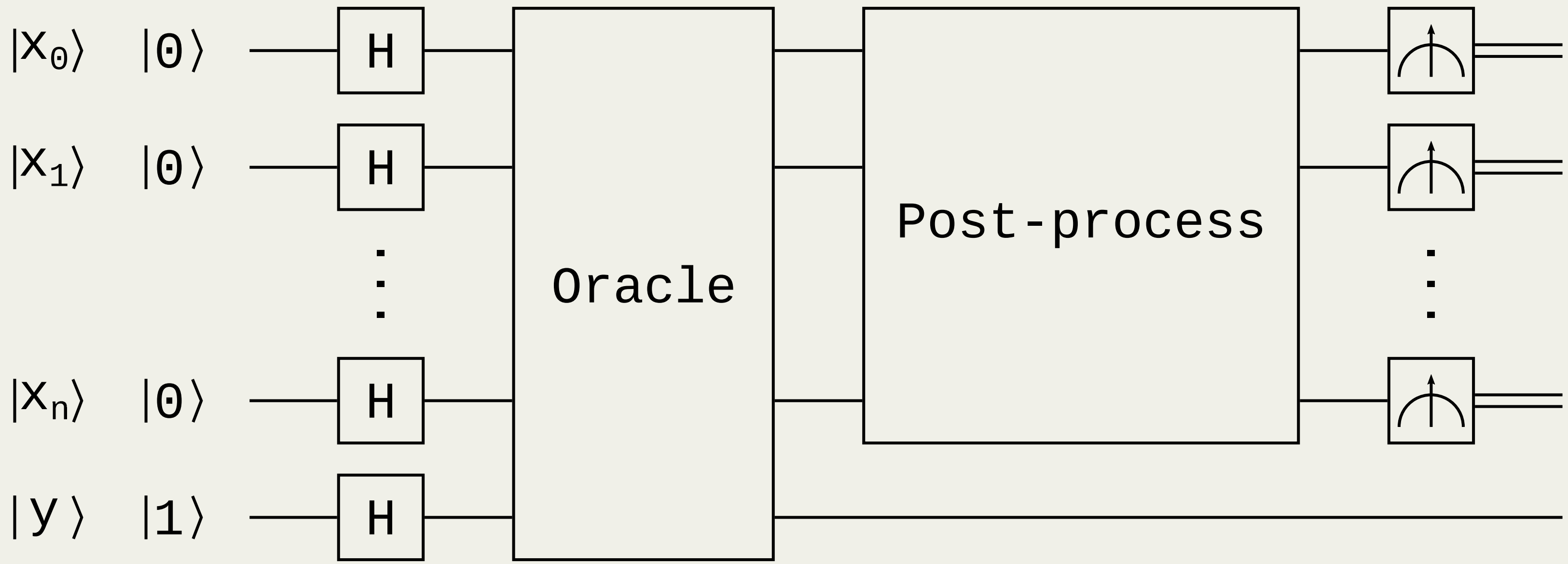
$$\rightarrow \frac{1}{2\sqrt{2}}\left(\left(|0\rangle \otimes \left(- |00\rangle + |01\rangle + |10\rangle + |11\rangle\right)\right) + \left(|1\rangle \otimes \left(|00\rangle - |01\rangle - |10\rangle - |11\rangle\right)\right)\right)$$

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

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

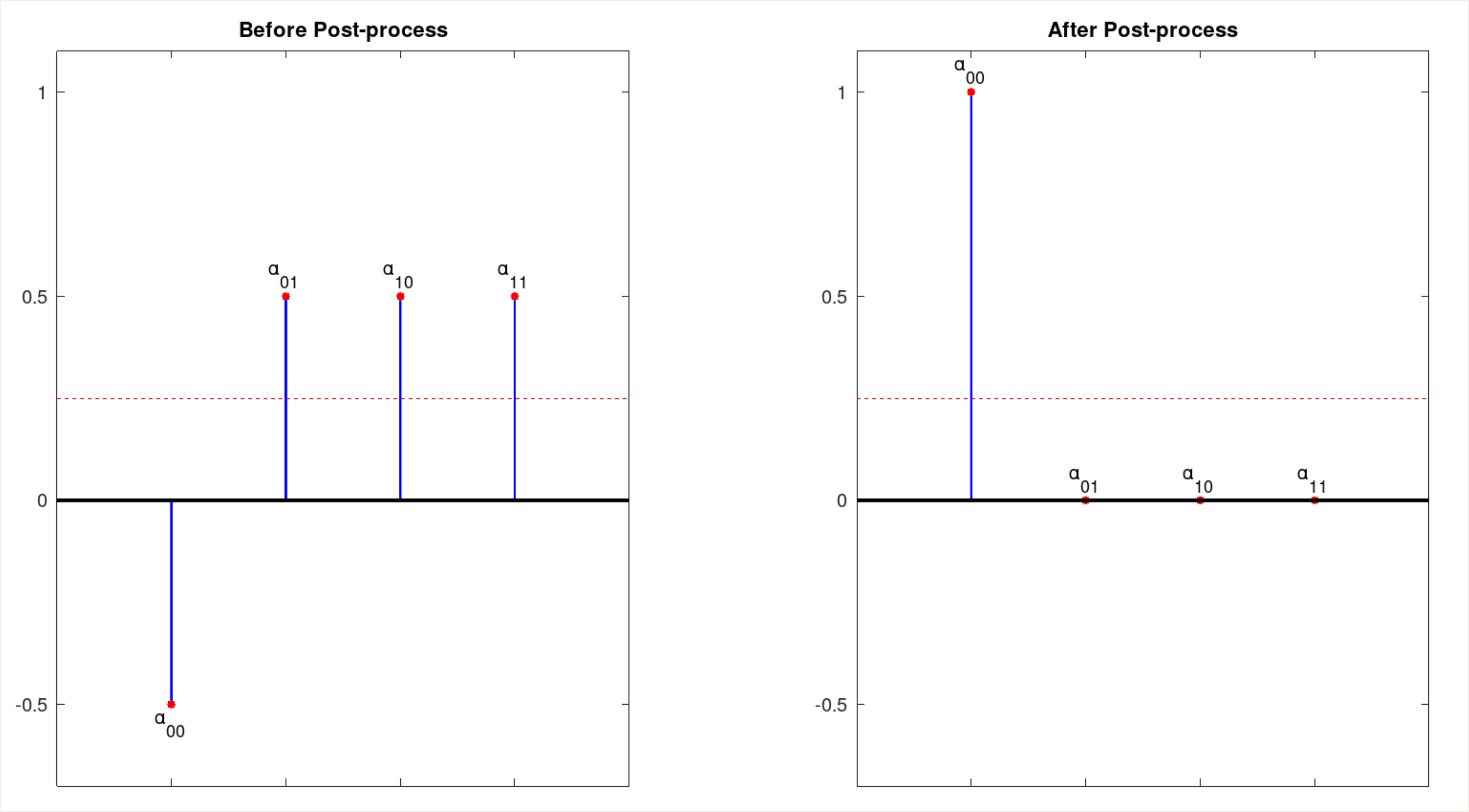
- Solution is marked with a phase of -1

Amsterdam University of Applied Sciences

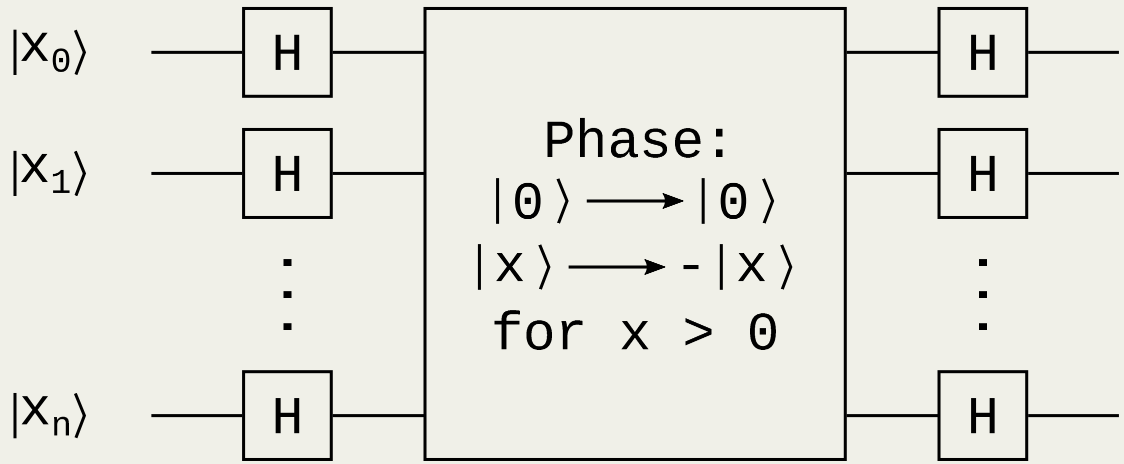


- Post–process
  - Map solutions to computational basis states
  - Extract solutions marked with a phase of -1 by **inversion around mean**

Inversion around the mean



Inversion around the mean



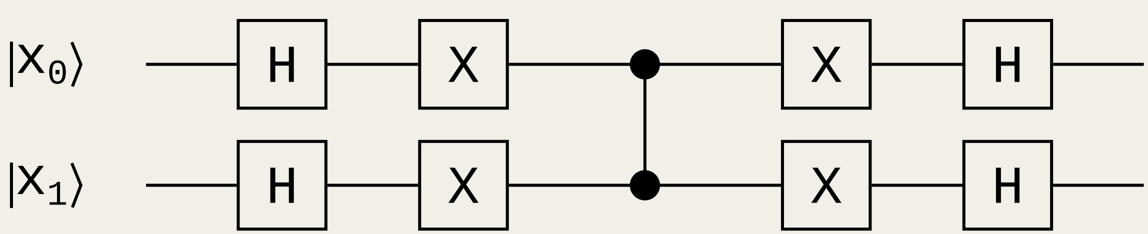
- Can be decomposed in:
    - H gates
    - A phase operation
    - H gates

Inversion around the mean

- For our two qubits example
  - The matrix is:

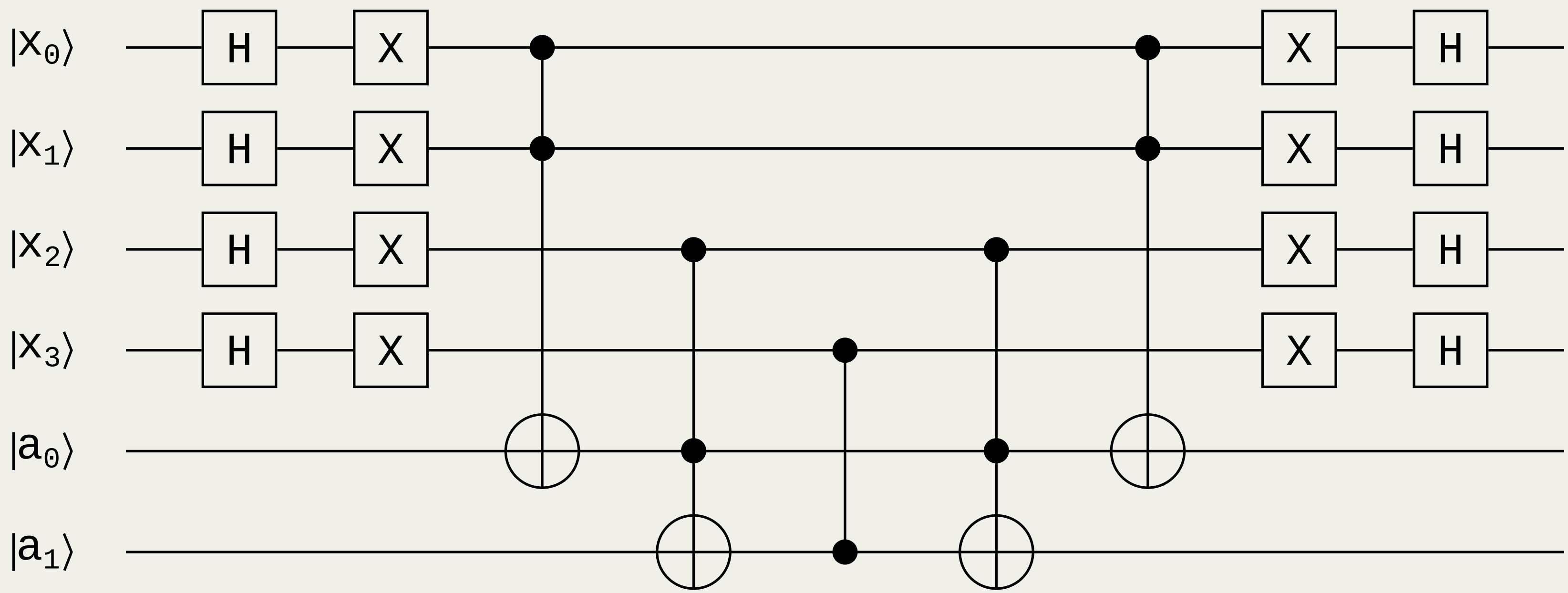
$$U_{IAM} = \frac{1}{2}\begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

  - The circuit is:



Inversion around the mean

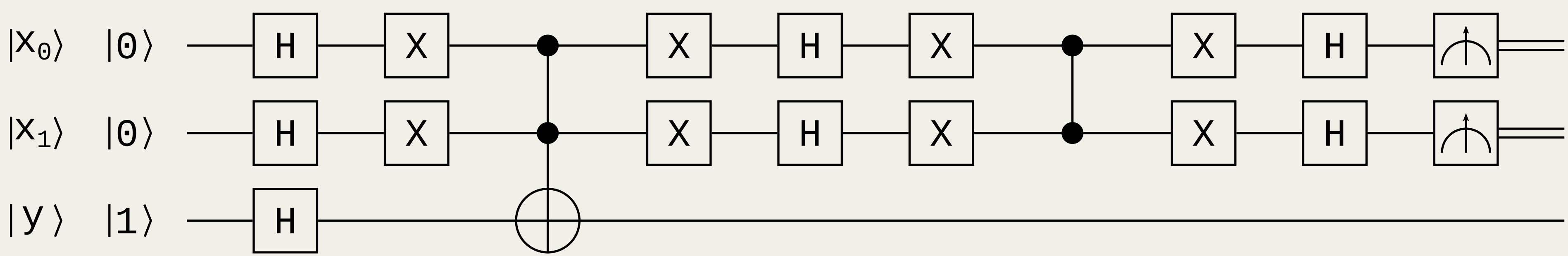
- What about a 4–qubits circuit?



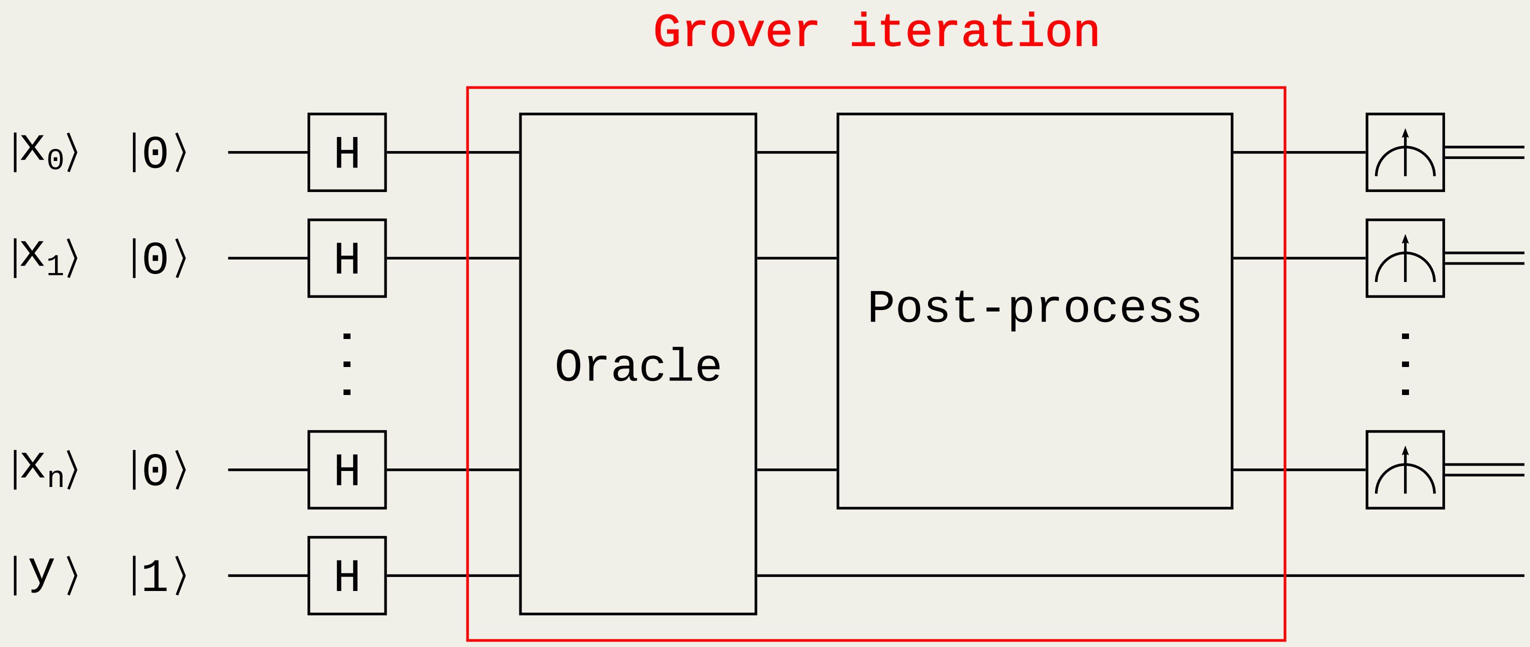
- For a generic n–qubits circuit:
    - Same approach: identify the pattern and continue it!

Complete circuit

- As usual, put the pieces together
  - Prepare input state
  - Apply oracle
  - Post–process
  - Measure
- Complete circuit for item $x^* = 00$:



Does Grover's algorithm scale?

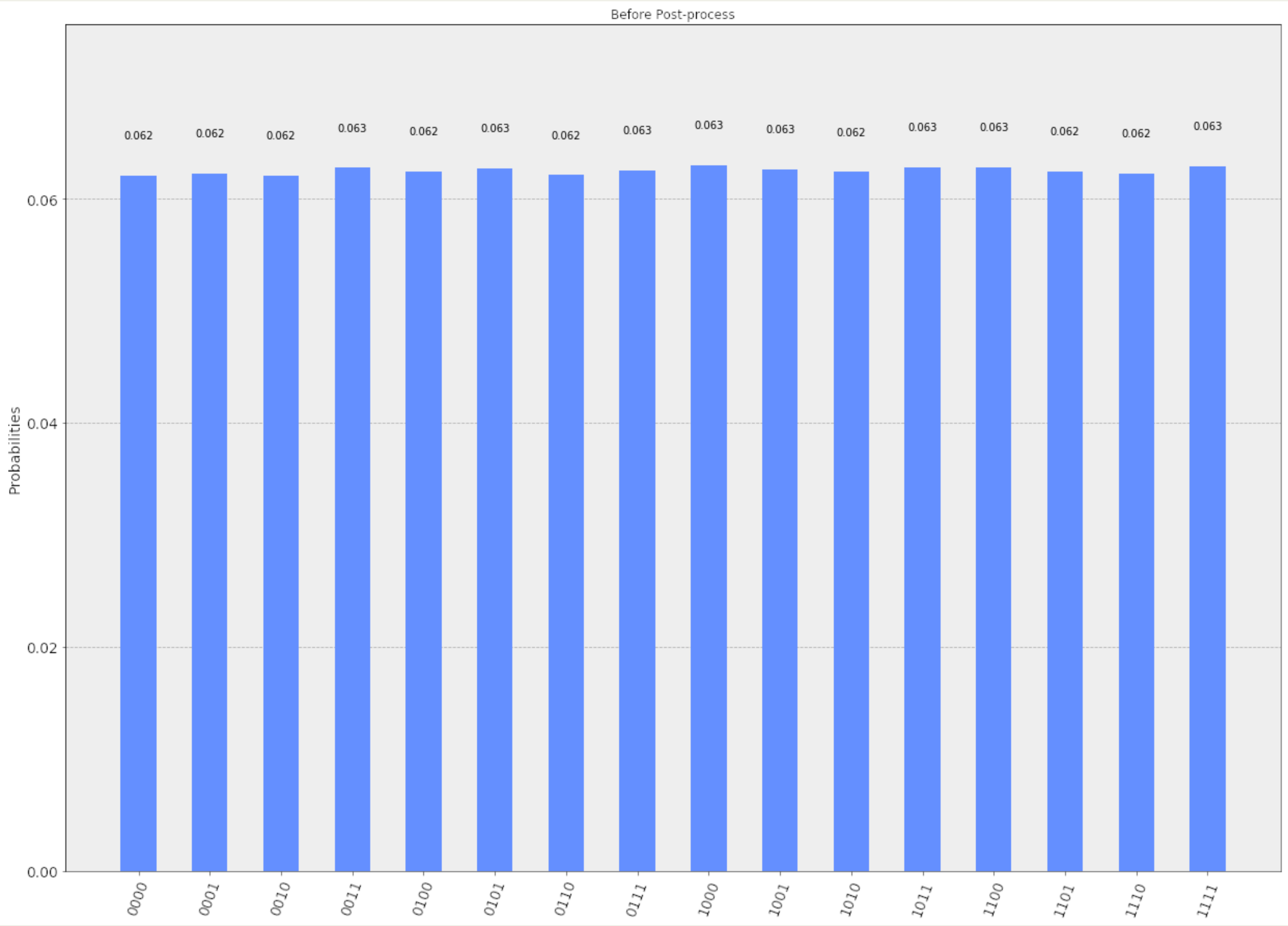


- Search space of N elements, exactly M solutions
- Repeat Grover iteration:

$$R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil$$

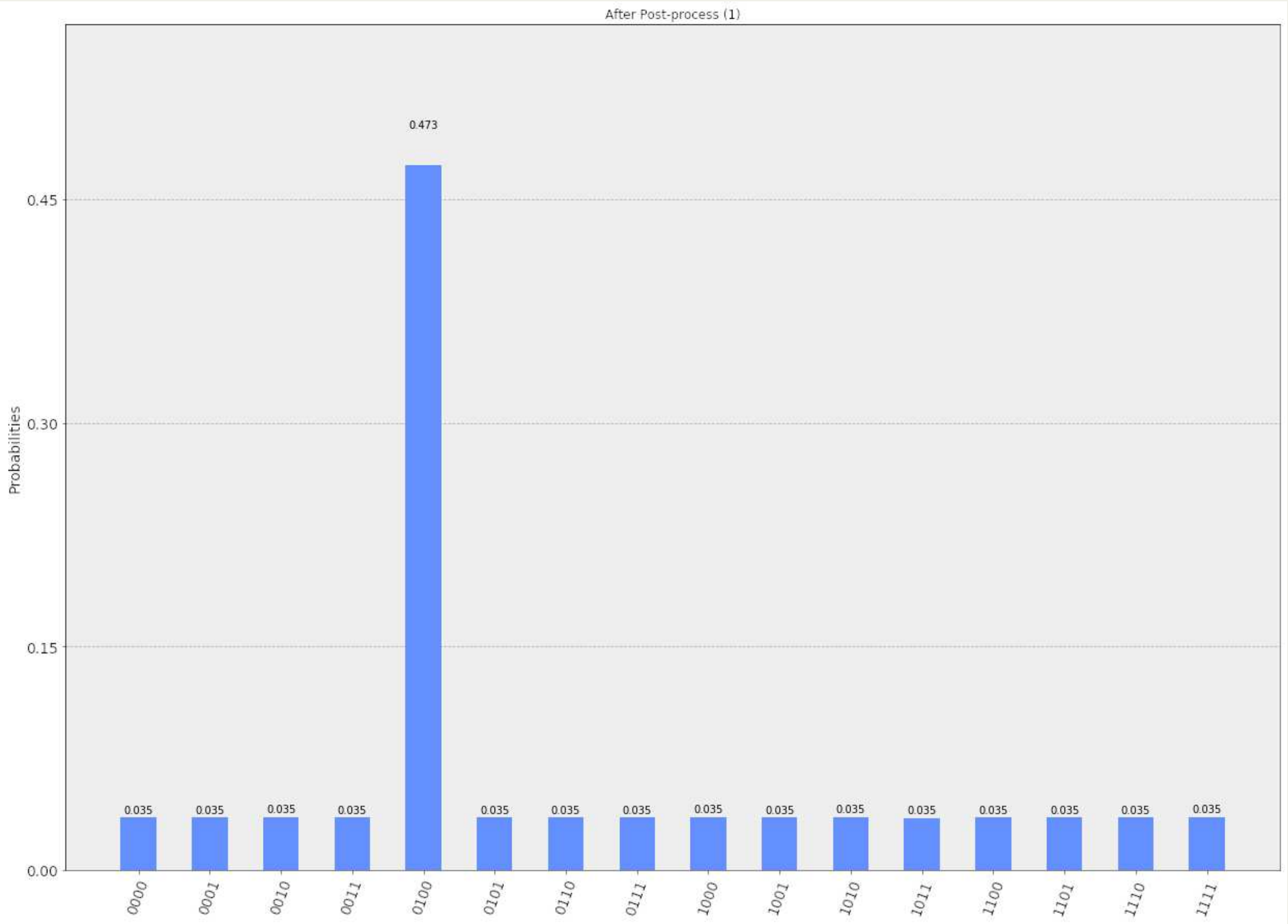
Example

- $N = 16$, $M = 1$, $x^* = |0100\rangle$ and $R \leq 4$



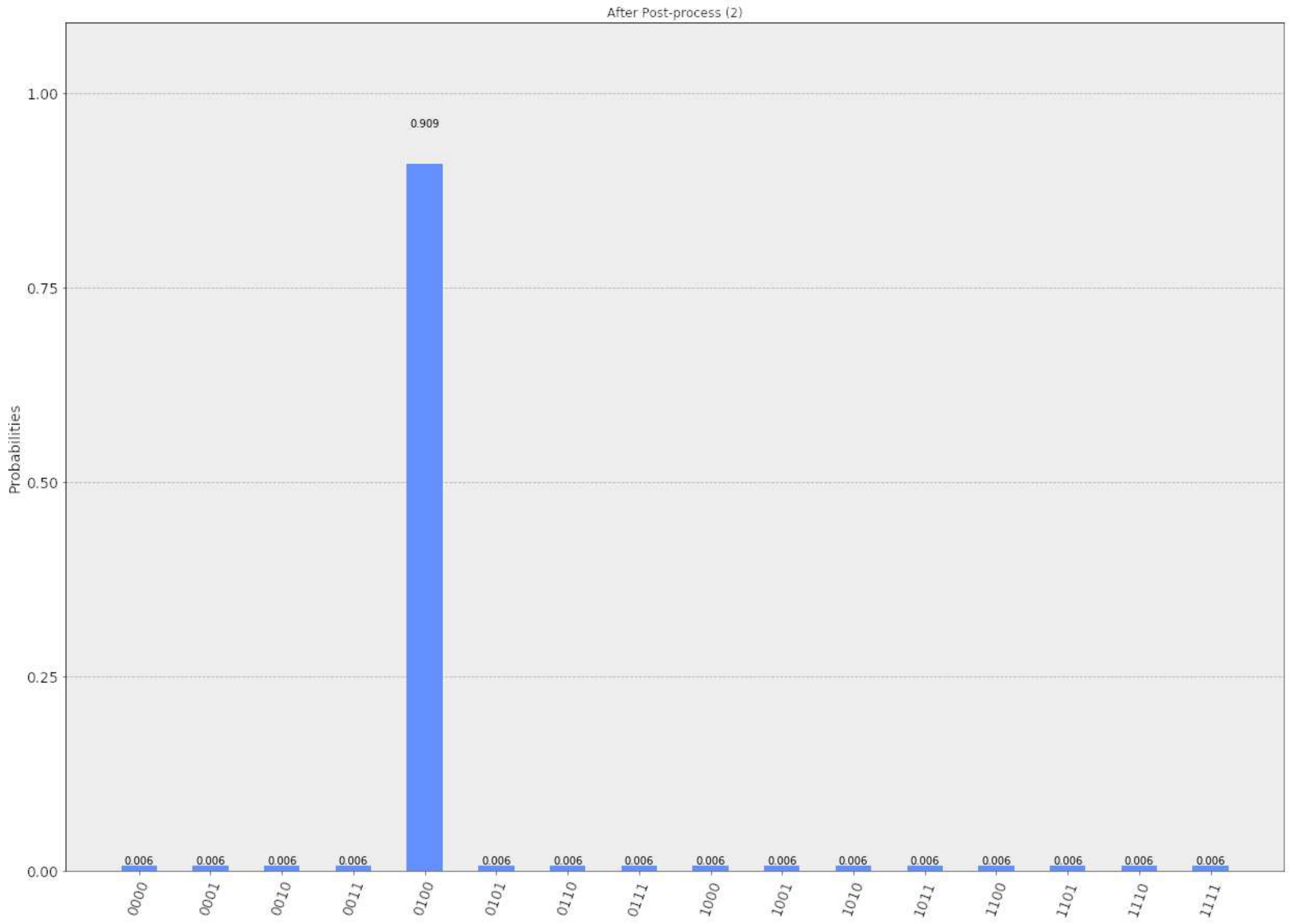
Example

- $N = 16$, $M = 1$, $x^* = |0100\rangle$ and $R \leq 4$



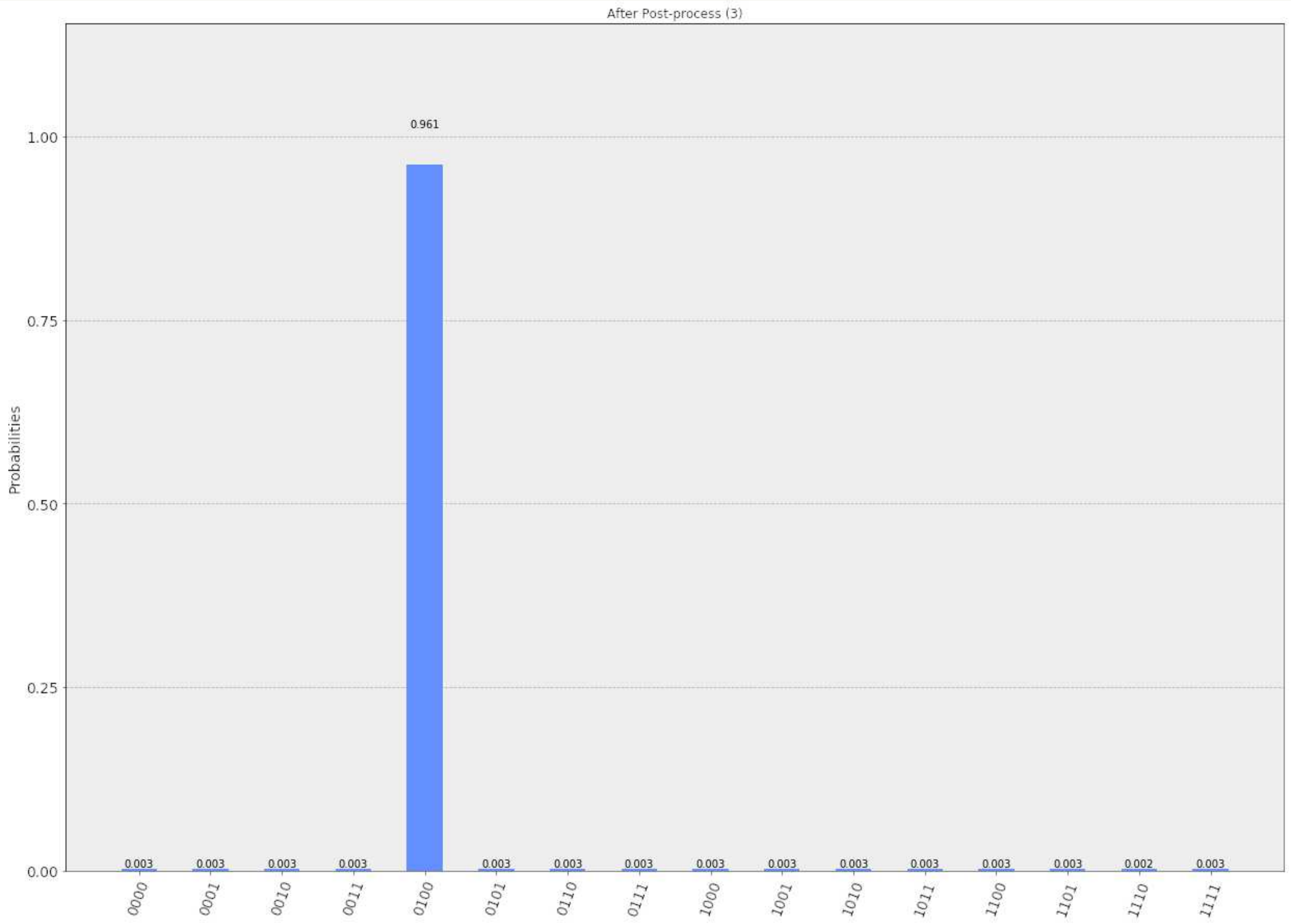
Example

- $N = 16$, $M = 1$, $x^* = |0100\rangle$ and $R \leq 4$



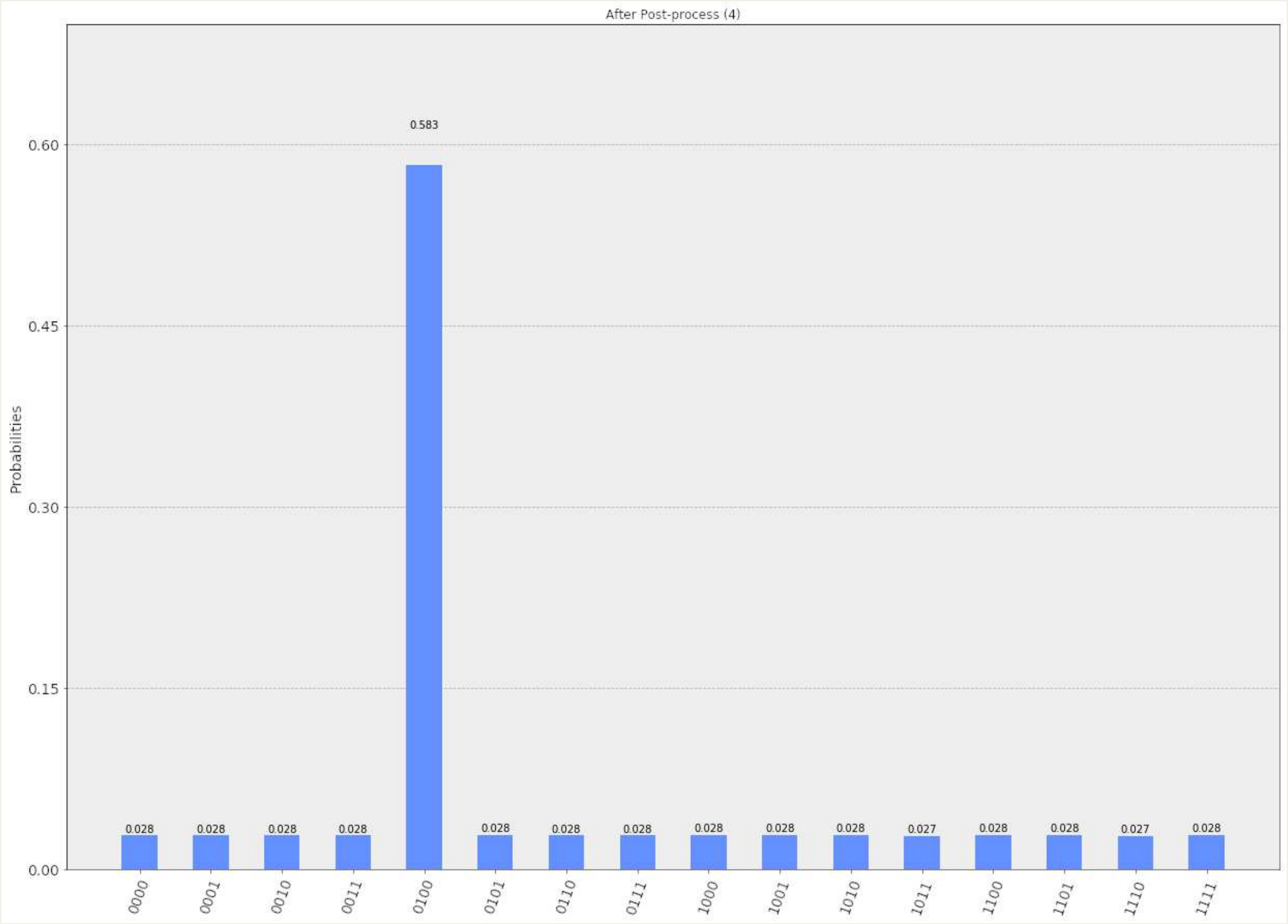
Example

- $N = 16$, $M = 1$, $x^* = |0100\rangle$ and $R \leq 4$



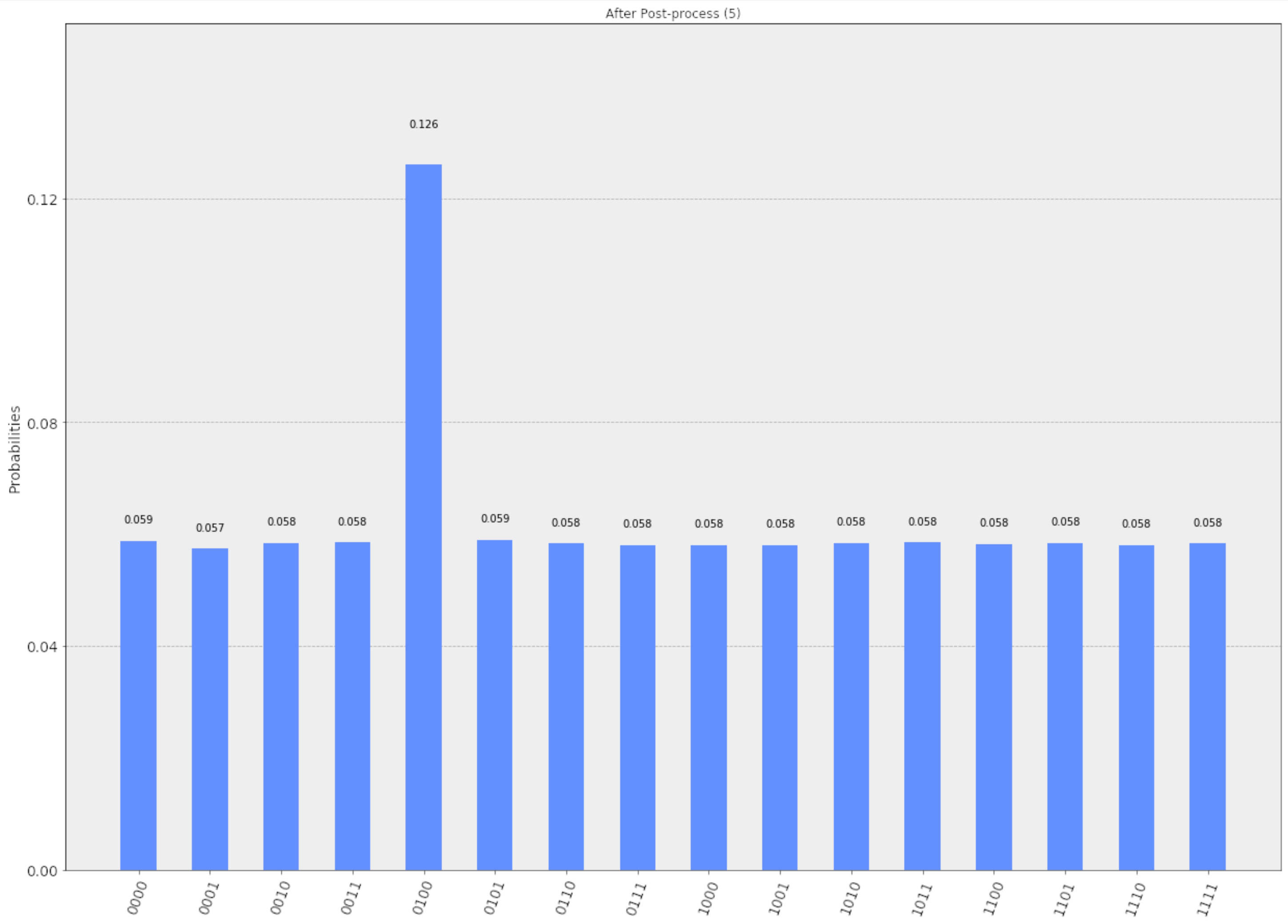
Example

- $N = 16$, $M = 1$, $x^* = |0100\rangle$ and $R \leq 4$



Example

- $N = 16$, $M = 1$, $x^* = |0100\rangle$ and $R \leq 4$



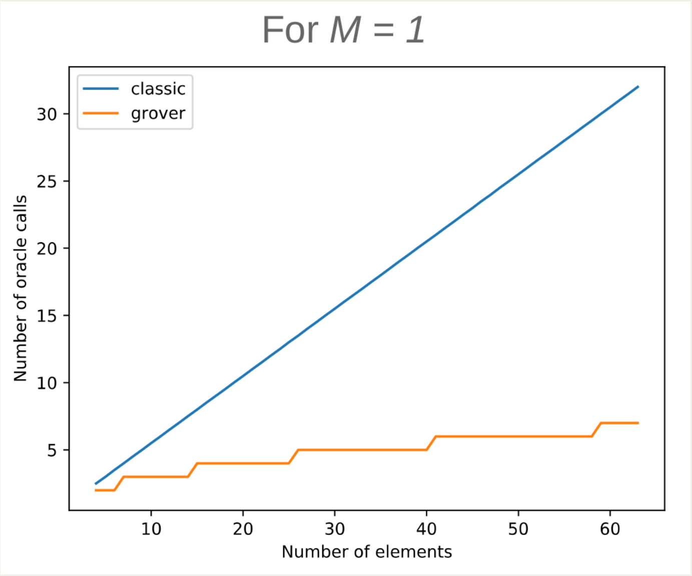
Complexity of Grover's algorithm

- Number of oracle calls required:

$$\mathcal{O}_{\text{c}}\left(\frac{N}{M}\right)$$

$$\mathcal{O}_{\text{g}}\left(\sqrt{\frac{N}{M}}\right)$$

- "Only" quadratic speedup
  - Still a huge improvement!




Questions?