# Quantum Computing

- Lecture 4 (May 7, 2025)

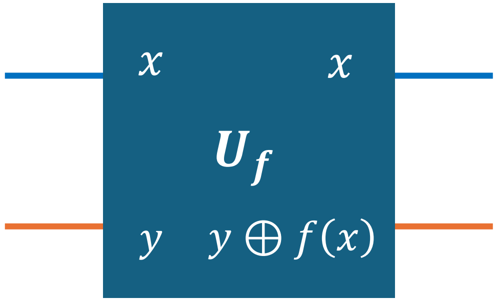
- Today:
  - Unitary operations on multi-qubit systems
  - Some examples (do it on the board)

# Unitary Operations

- A quantum gate is a unitary operator (⇔ A unitary represents some quantum gate)
  - A unitary operator has **linearity:** $\boldsymbol{U}(c_1\boldsymbol{v_1} + c_2\boldsymbol{v_2}) = c_1\boldsymbol{Uv_1} + c_2\boldsymbol{Uv_2}$

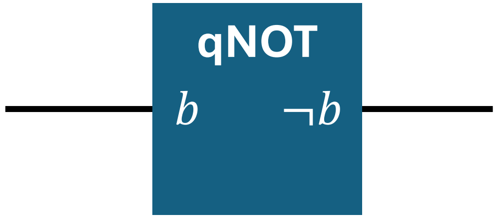
- Quantum gates operate on superposition: **Linearity**
  - View any quantum gate as a unitary linear operator (matrix)
  - Quantum gates act on superpositions according to linearity

- Make a classical computable function unitary $f \to \boldsymbol{U_f}$
  - Use **input qubits** and **ancilla qubits** to make it invertible
  - Any classical algorithm can be simulated by quantum computers



# Unitary Operations on Single-Qubit States

- Single-qubit unitary:
  - Examples: qNOT, Hadamard transform, …



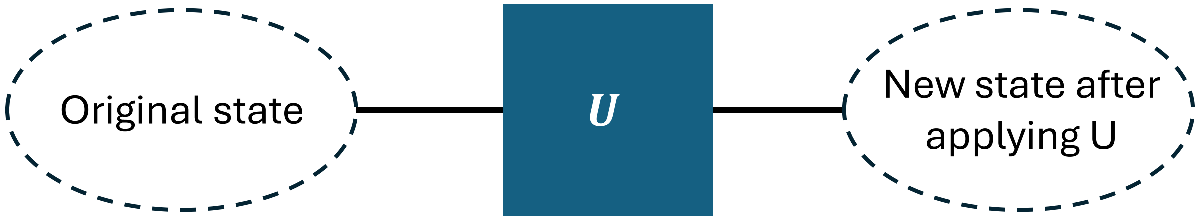

$$\mathbf{qNOT}(|b\rangle) \rightarrow |\neg b\rangle$$


$$\boldsymbol{H}(|b\rangle) \rightarrow \frac{|0\rangle + (-1)^b |1\rangle}{\sqrt{2}}$$

# Unitary Operations on Single-Qubit States

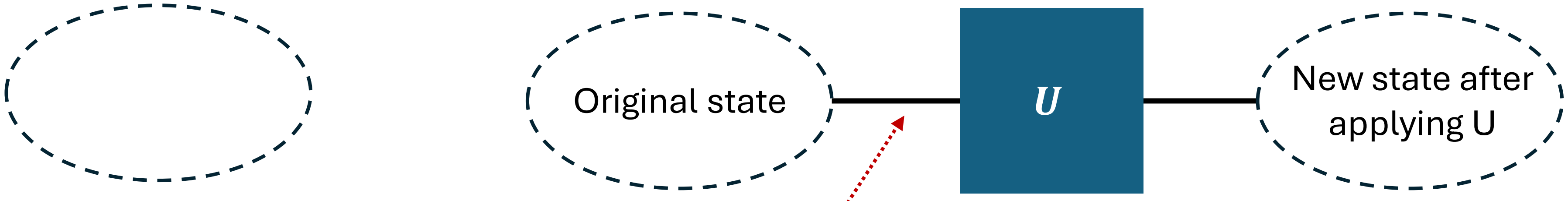
- Single-qubit unitary:
  - Examples: qNOT, Hadamard transform, …

Some **isolated** single-qubit system



# Unitary Operations on Single-Qubit States

- Single-qubit unitary:
  - Examples: qNOT, Hadamard transform, …



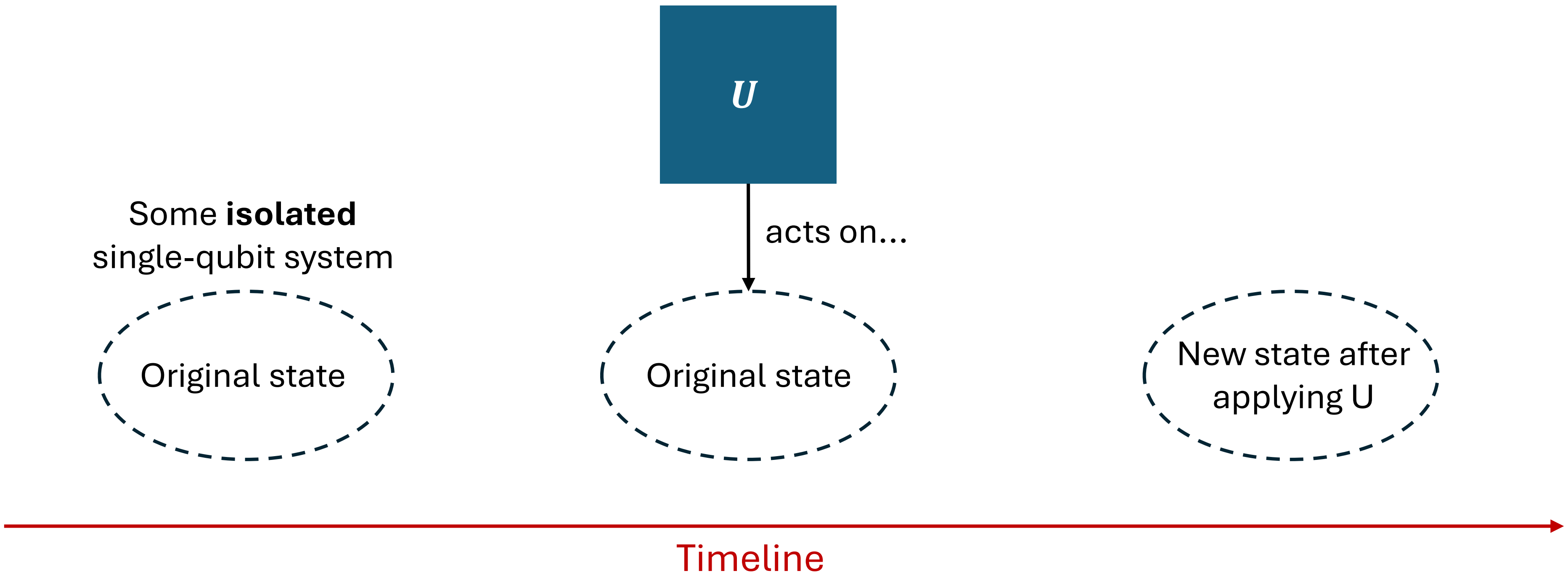

Some **isolated** single-qubit system

Misunderstand: The “wire” here **does not** represent a real wire!

Instead, it just visually **tracks the state of the system** through time.

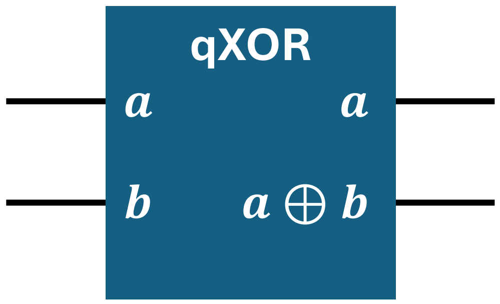
# Unitary Operations on Single-Qubit States

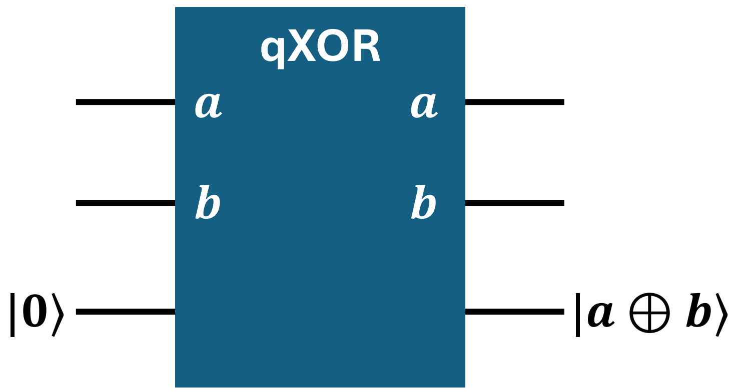
- Single-qubit unitary:
  - Examples: qNOT, Hadamard transform, …

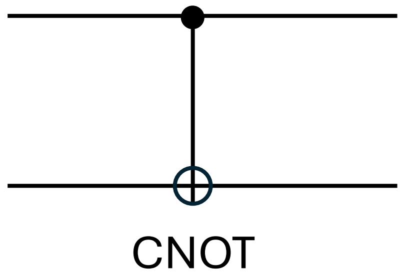


# Unitary Operations on Multi-Qubit States

- Multi-qubit unitary:
  - Examples: qXOR, CNOT, …

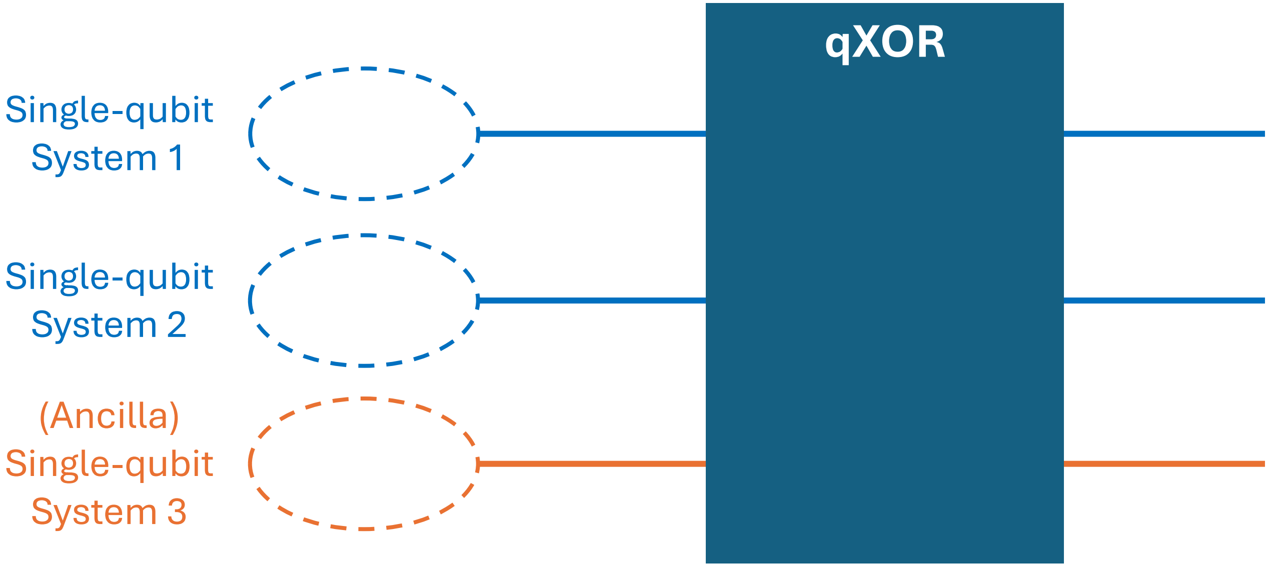


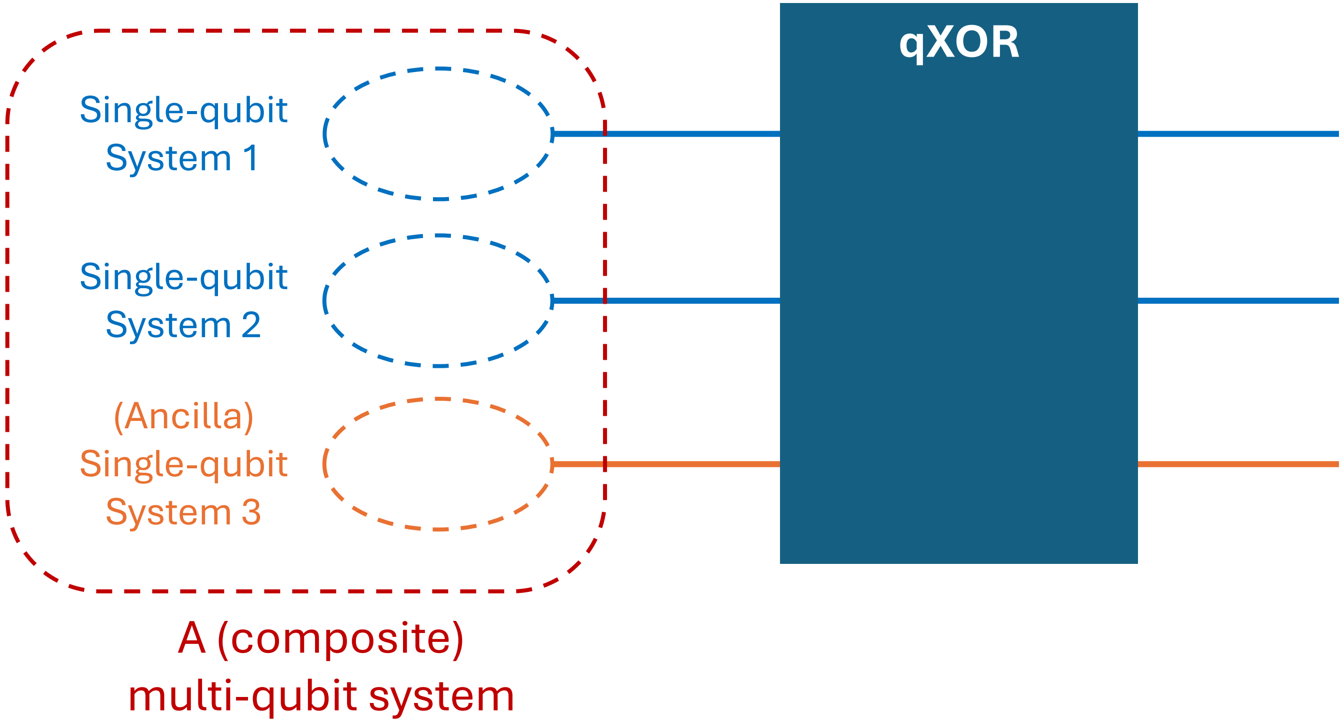
# Unitary Operations on Multi-Qubit States

- Multi-qubit unitary:
  - Examples: qXOR, CNOT, …



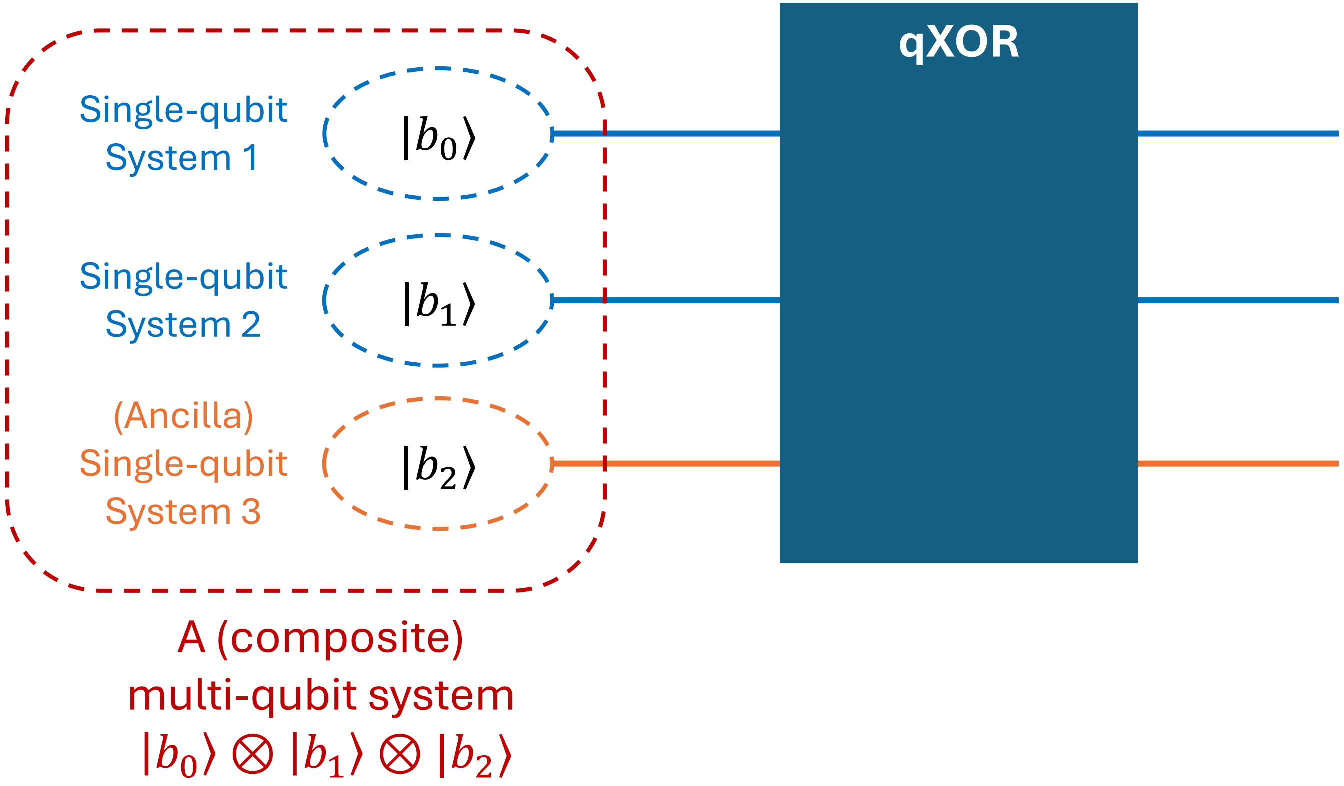
# Unitary Operations on Multi-Qubit States

- Multi-qubit unitary:
  - Examples: qXOR, CNOT, …



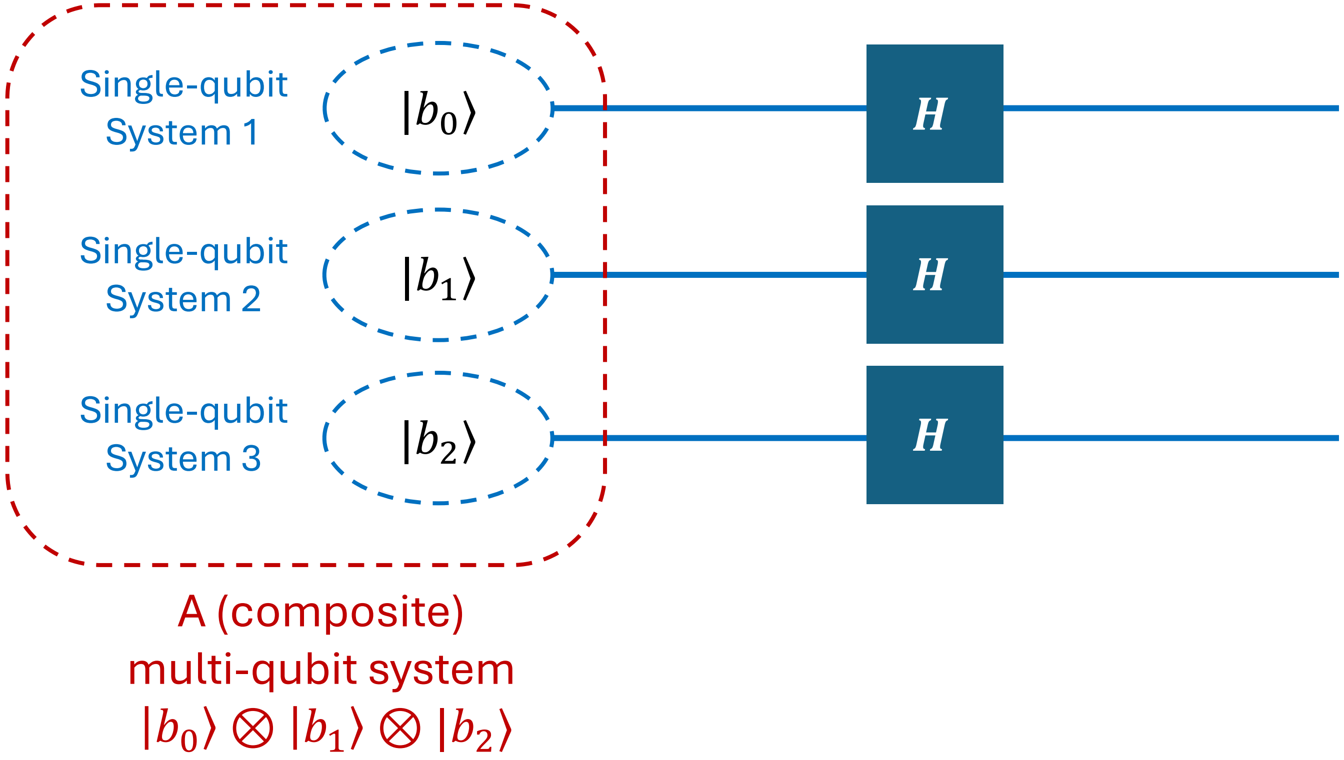
# Unitary Operations on Multi-Qubit States

- Multi-qubit unitary:
  - Examples: qXOR, CNOT, …



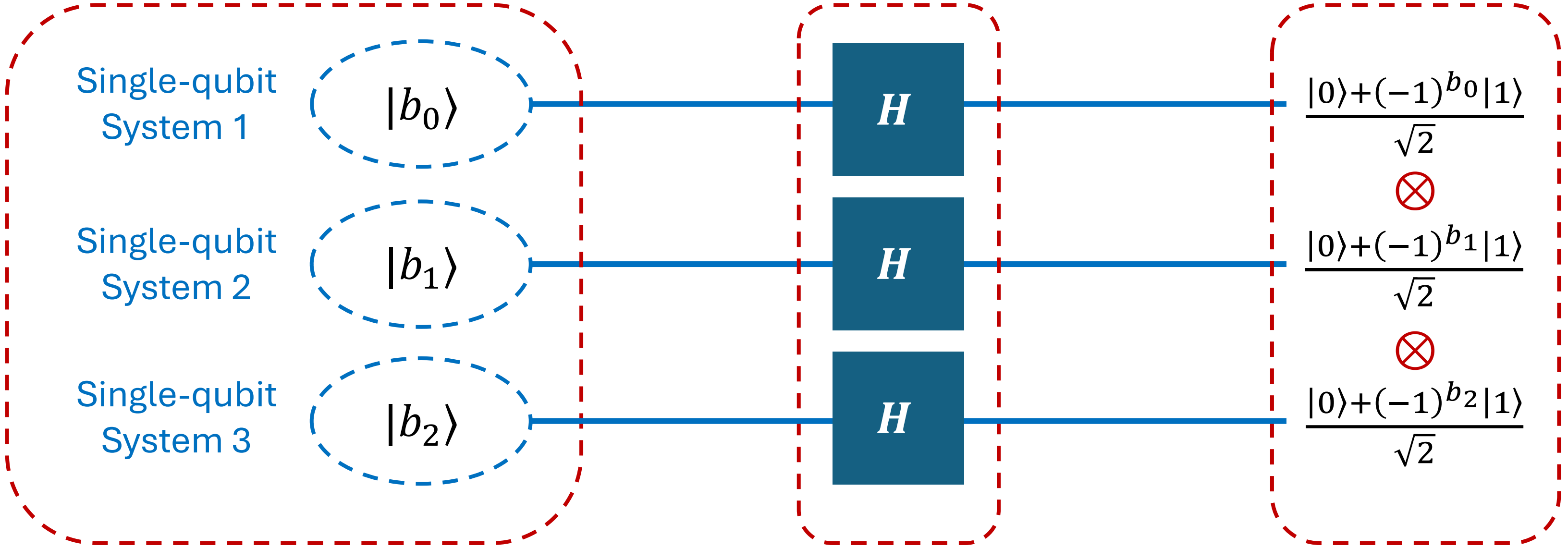
# Unitary Operations on Multi-Qubit States

- Multi-qubit unitary:
  - Examples: qXOR, CNOT, **Parallel action of Hadamard gates…**



# Unitary Operations on Multi-Qubit States

- Multi-qubit unitary:
  - Examples: qXOR, CNOT, **Parallel action of Hadamard gates…**




A (composite) multi-qubit system $|b_0\rangle \otimes |b_1\rangle \otimes |b_2\rangle$

$\boldsymbol{H}^{\otimes 3}$ (Parallel H acts on 3-qubit system)

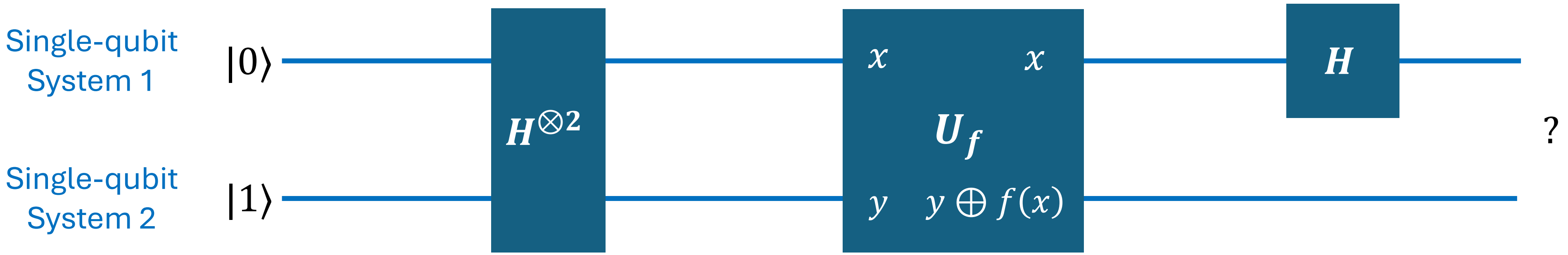
The state of the composite system after applying $\boldsymbol{H}^{\otimes 3}$

# Unitary Operations on Multi-Qubit States

- Multi-qubit unitary:
  - **Parallel action of Hadamard gates…**

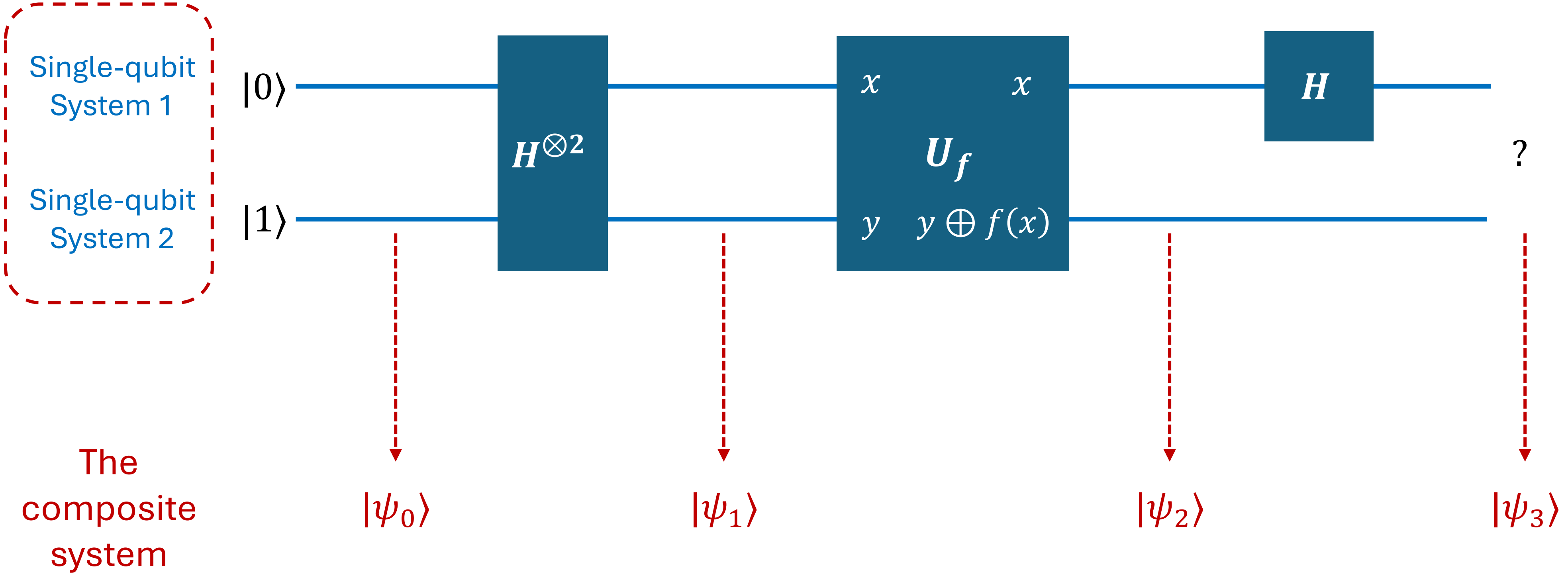
# Deutsch's Algorithm

- Let $f$ be a bit function...
- (Do it on the board)



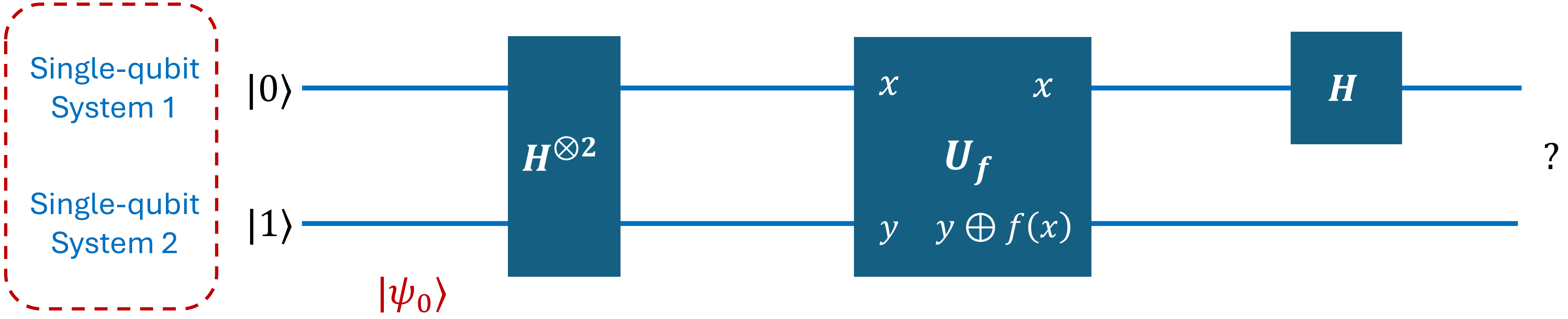
# Deutsch's Algorithm

- Let $f$ be a bit function…
- (Do it on the board)



# Deutsch's Algorithm

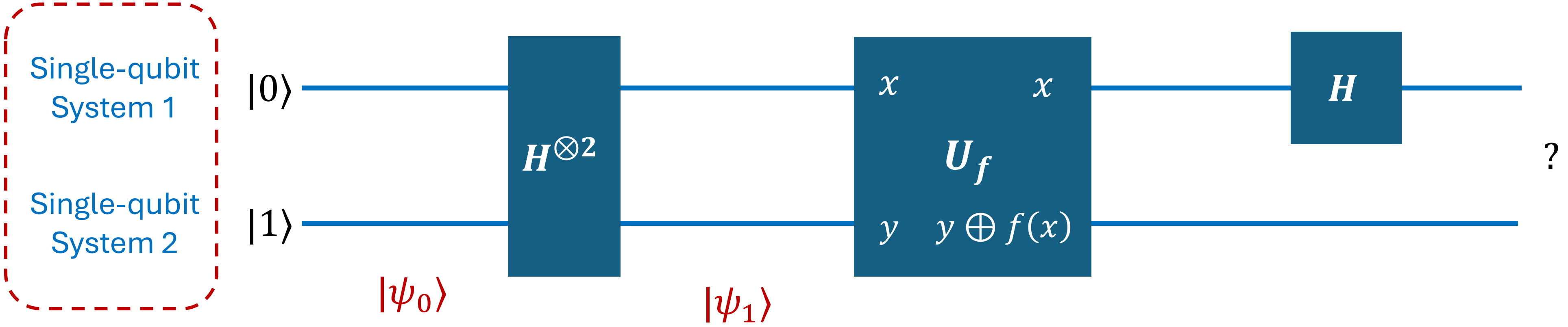
- Let $f$ be a bit function...
- (Do it on the board)




The composite system $|\psi_0\rangle = |01\rangle = |0\rangle \otimes |1\rangle$

# Deutsch’s Algorithm

- Let $f$ be a bit function...
- (Do it on the board)

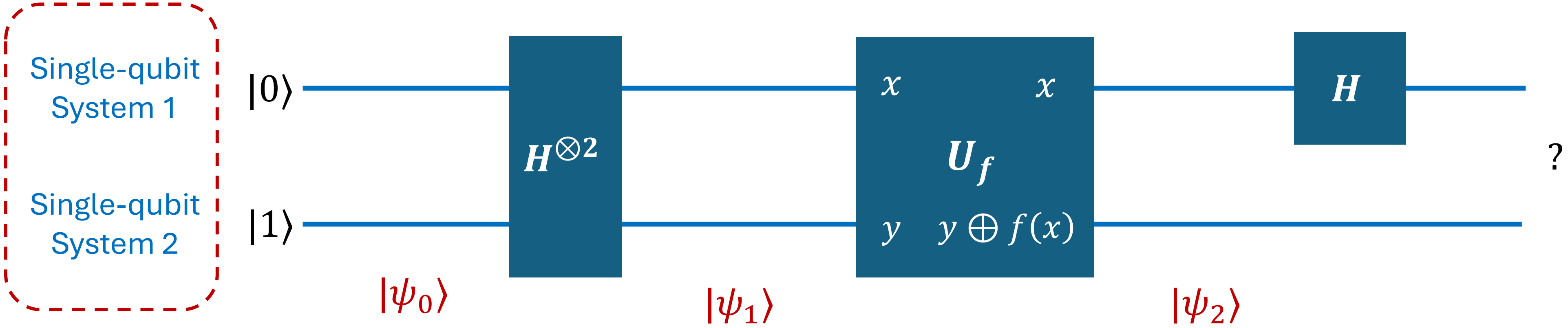



The composite system

$$|\psi_1\rangle = H^{\otimes 2}|\psi_0\rangle = \left(\frac{|0\rangle + |1\rangle}{2}\right) \otimes \left(\frac{|0\rangle - |1\rangle}{2}\right)$$

# Deutsch's Algorithm

- Let $f$ be a bit function...
- (Do it on the board)

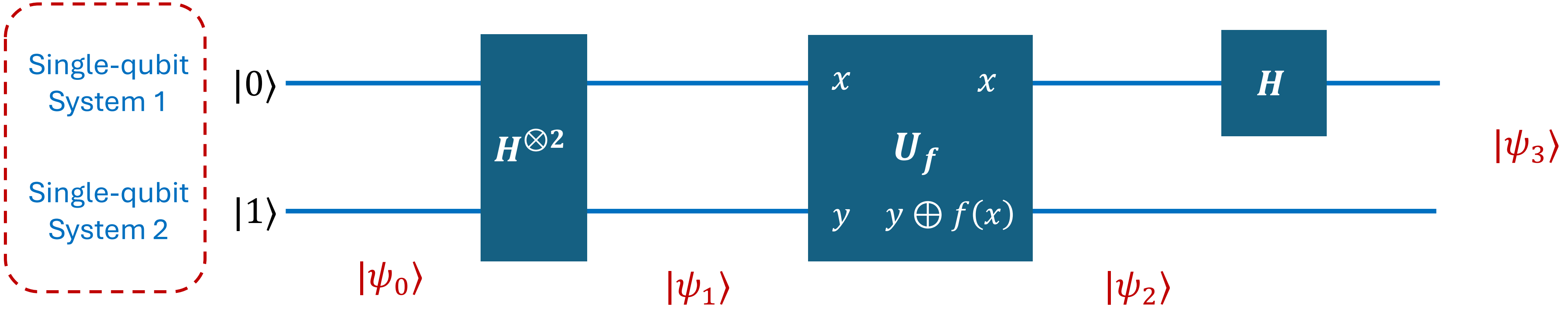



The composite system

$$|\psi_2\rangle = U_f|\psi_1\rangle = \left(\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{2}\right) \otimes \left((-1)^{f(0)}\left(\frac{|0\rangle - |1\rangle}{2}\right)\right)$$

# Deutsch's Algorithm

- Let $f$ be a bit function...
- (Do it on the board)

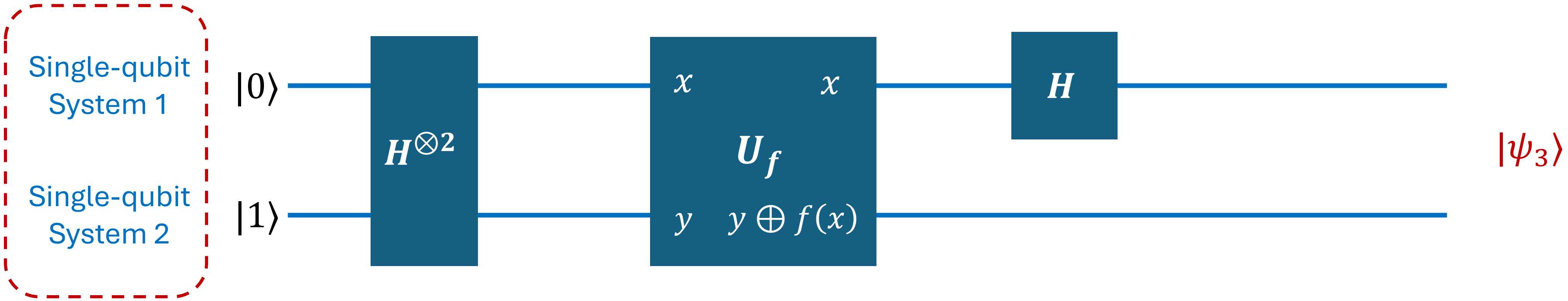



The composite system

$$|\psi_3\rangle = (\boldsymbol{H} \otimes \boldsymbol{I})|\psi_2\rangle = (|f(0) \oplus f(1)\rangle) \otimes \left((-1)^{f(0)} \left(\frac{|0\rangle - |1\rangle}{2}\right)\right)$$

# Deutsch's Algorithm

- Let $f$ be a bit function...
- (Do it on the board)



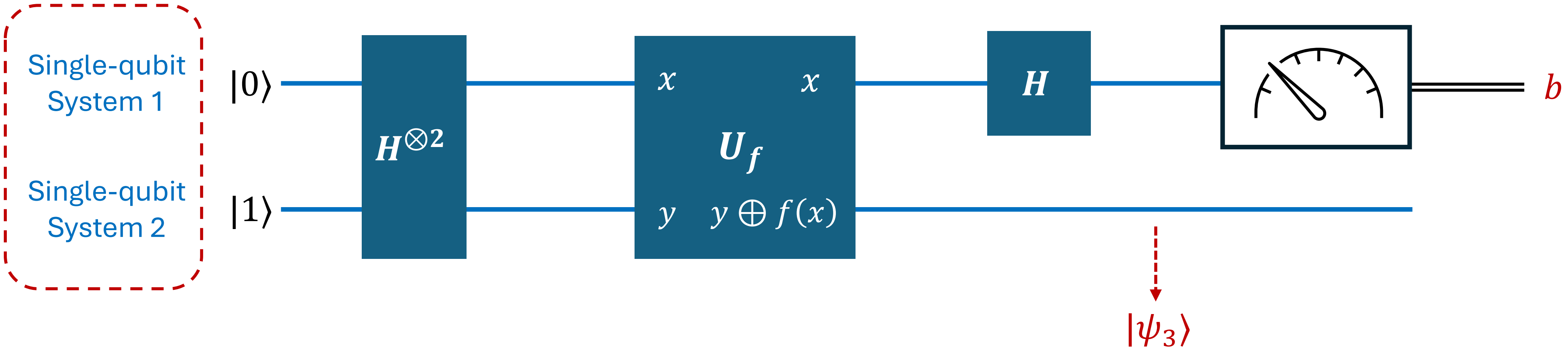

The composite system

$$|\psi_3\rangle = \pm|f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

(We just let $(-1)^{f(0)} = \pm$, which does not change the measurement outcome)

# Deutsch’s Algorithm

- Let $f$ be a bit function…
- (Do it on the board)

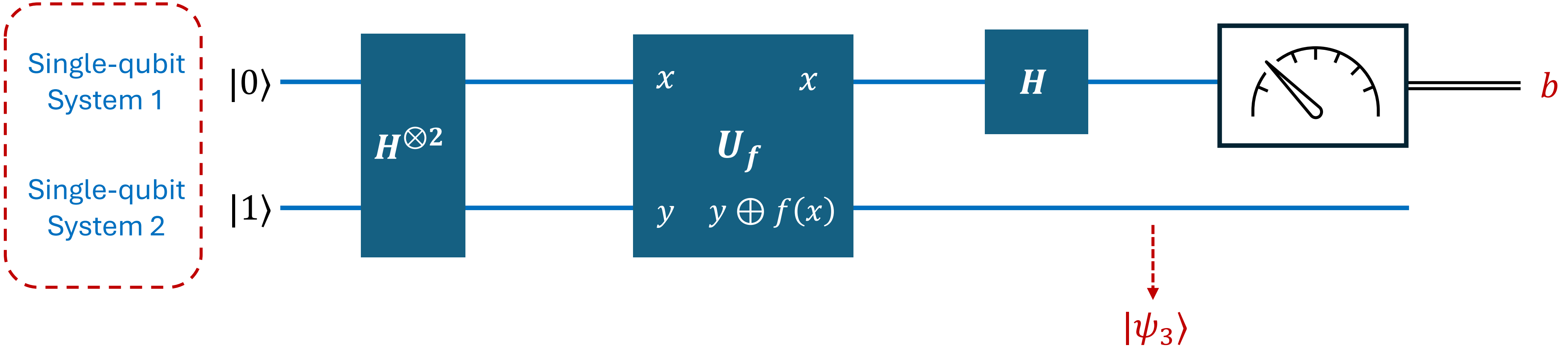



The composite system

$$|\psi_3\rangle = \pm|f(0) \oplus f(1)\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \xrightarrow{\text{Measure the first system}} b = f(0) \oplus f(1)$$

# Deutsch's Algorithm

- Let $f$ be a bit function...
- (Do it on the board)

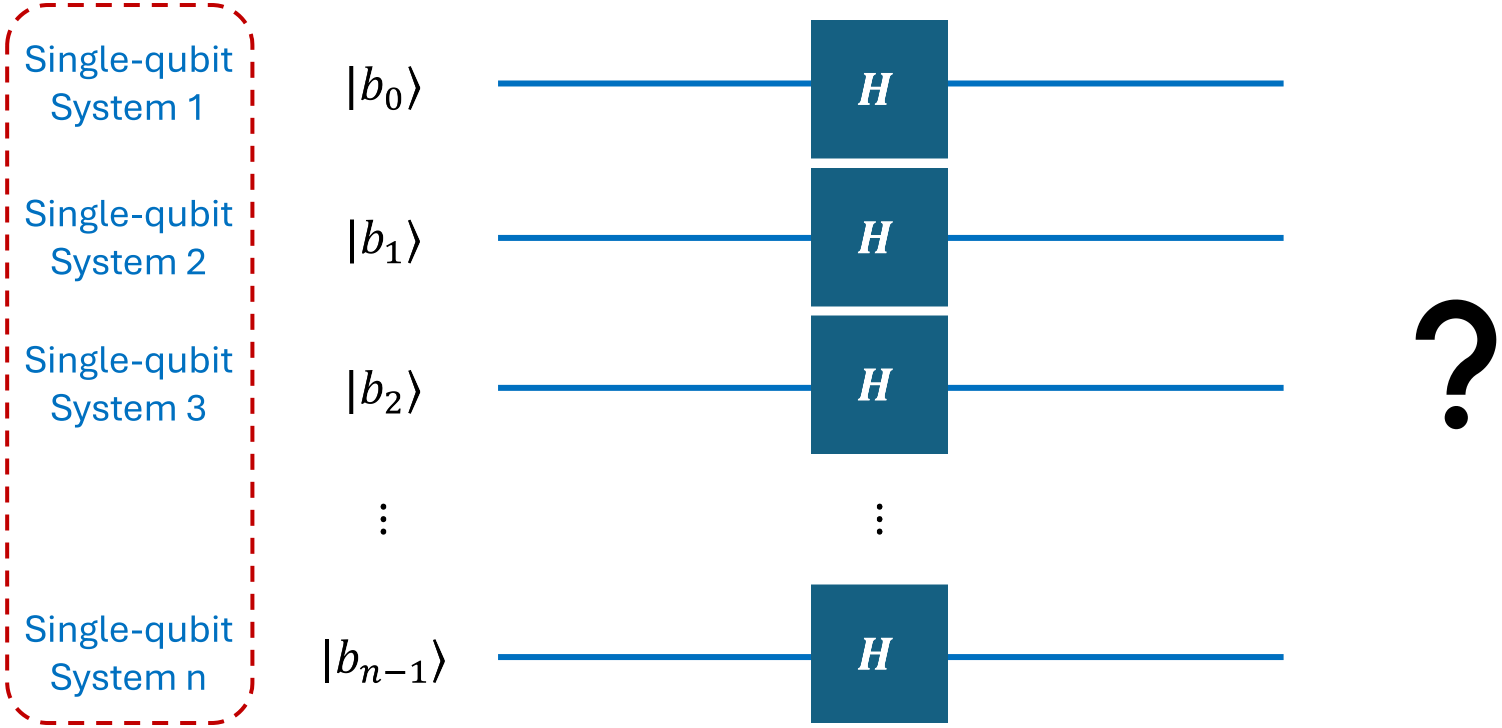



The composite system

$$|\psi_3\rangle \xRightarrow{\text{Measure the first system}} b = f(0) \oplus f(1)$$

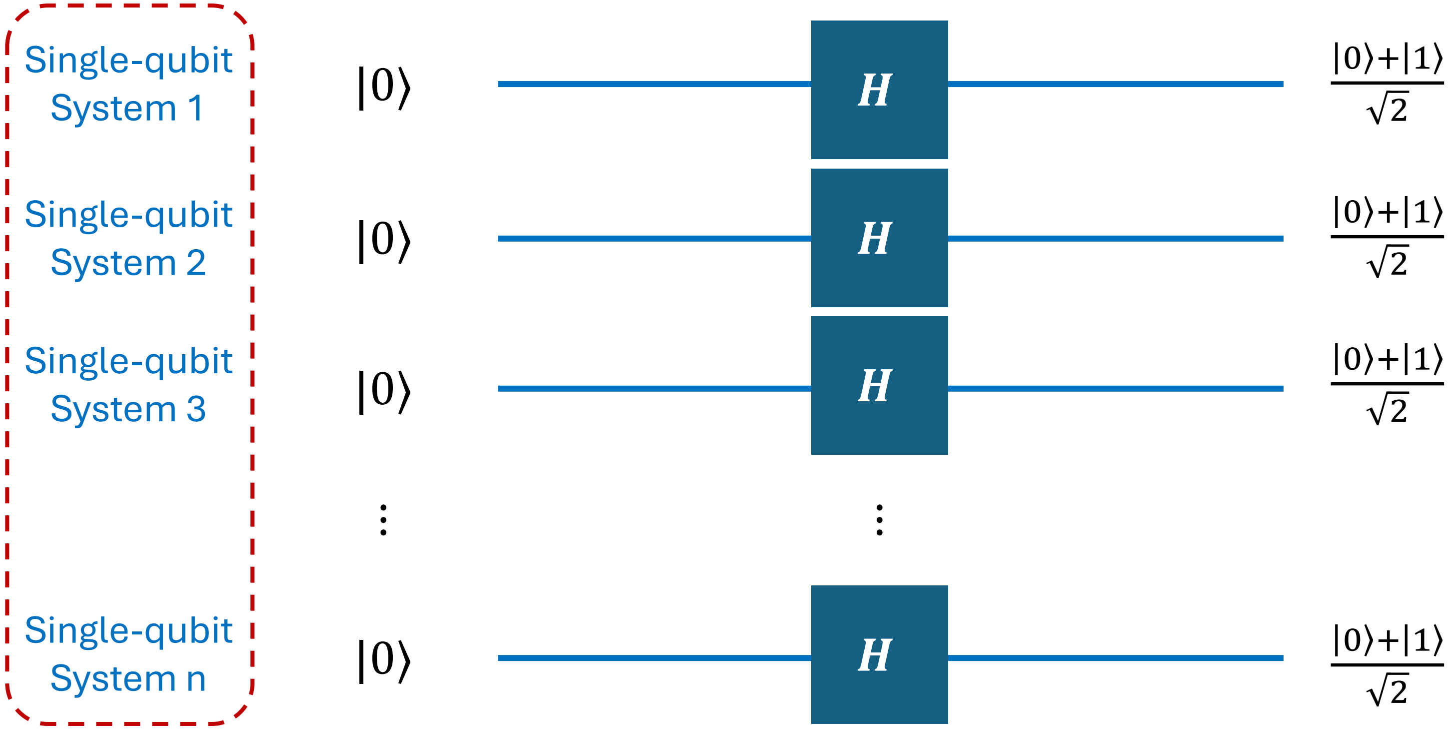
One (quantum) evaluation on $f$ and get $f(0) \oplus f(1)$

# Parallel Hadamard Gates




$|b_0\rangle \otimes |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_{n-1}\rangle$

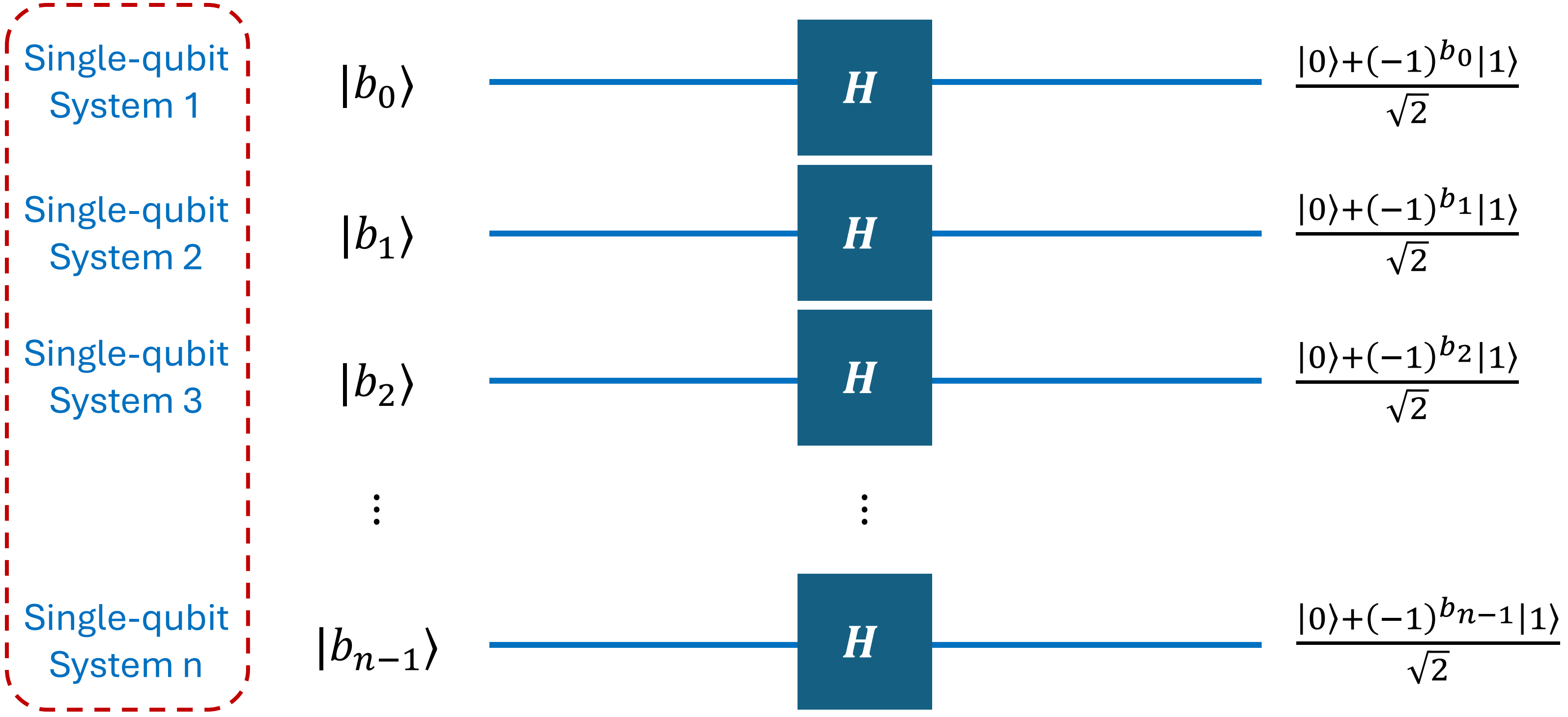
# Parallel Hadamard Gates




$$= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |\boldsymbol{x}\rangle$$

$$|b_0\rangle \otimes |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_{n-1}\rangle$$

# Parallel Hadamard Gates



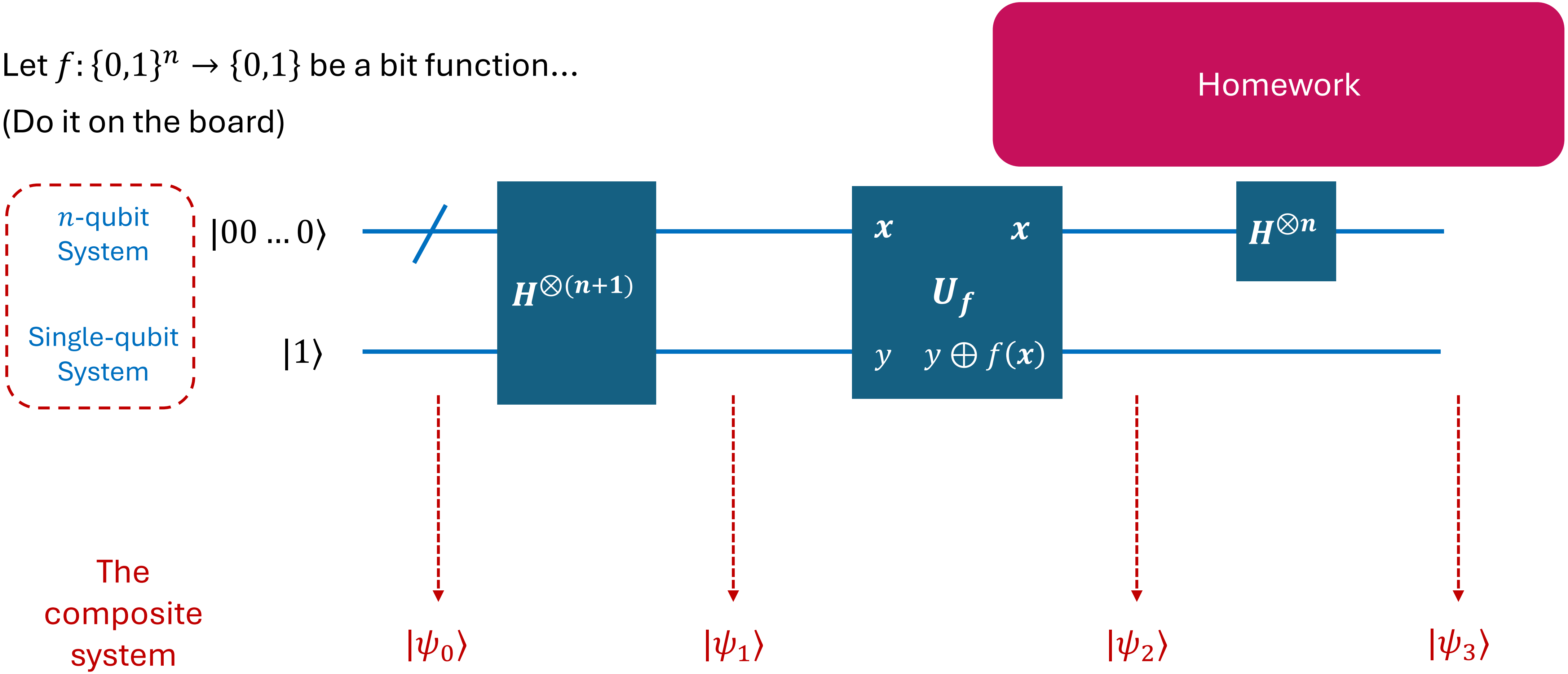

$|b_0\rangle \otimes |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_{n-1}\rangle$

Let $\boldsymbol{b} := b_{n-1}b_{n-2} \ldots b_0$ be the classical bit string

$$\boldsymbol{H}^{\otimes n}|\boldsymbol{b}\rangle = \sum_{x\in\{0,1\}^n} \frac{(-1)^{\boldsymbol{x}^T\boldsymbol{b}}}{\sqrt{2^n}}|\boldsymbol{x}\rangle$$

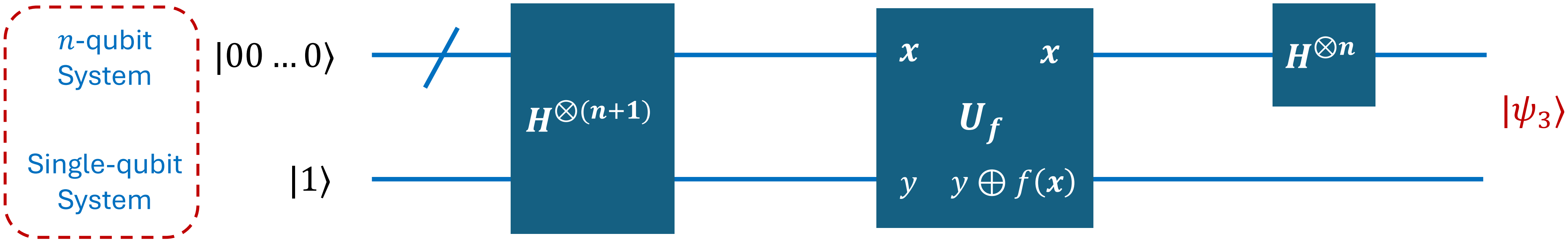
# The Deutsch-Jozsa Algorithm

- Let $f: \{0,1\}^n \rightarrow \{0,1\}$ be a bit function…
- (Do it on the board)



# The Deutsch-Jozsa Algorithm

- Let $f: \{0,1\}^n \rightarrow \{0,1\}$ be a bit function…
- (Do it on the board)

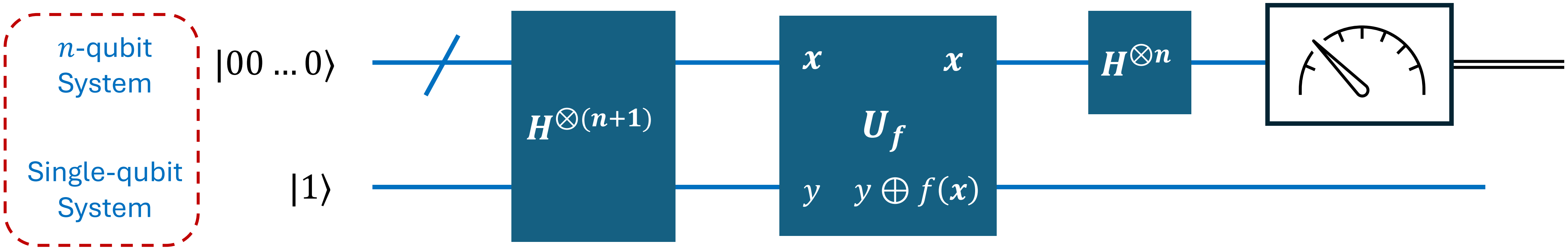



The composite system

$$|\psi_3\rangle = \sum_{z \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} \frac{(-1)^{x^T z + f(x)} |z\rangle}{\sqrt{2^n}} \left[ \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

# The Deutsch-Jozsa Algorithm

- Let $f: \{0,1\}^n \to \{0,1\}$ be a bit function…
- (Do it on the board)




The composite system

$$|\psi_3\rangle = \sum_{z \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} \frac{(-1)^{x^T z + f(x)}|z\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

Measure the first $n$ system

?

# The Deutsch-Jozsa Algorithm

$$|\psi_3\rangle = \sum_{\boldsymbol{z} \in \{0,1\}^n} \sum_{\boldsymbol{x} \in \{0,1\}^n} \frac{(-1)^{\boldsymbol{x}^T \boldsymbol{z} + f(\boldsymbol{x})} |\boldsymbol{z}\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \xrightarrow[\phantom{xxxxxxxxx}]{\text{Measure the first } n\text{-qubit system}} \boldsymbol{z}$$

- What's the probability of $\boldsymbol{z} = 00 \ldots 0$ ? What about $\boldsymbol{z} \neq 00 \ldots 0$ ?

# The Deutsch-Jozsa Algorithm

$$|\psi_3\rangle = \sum_{\boldsymbol{z} \in \{0,1\}^n} \sum_{\boldsymbol{x} \in \{0,1\}^n} \frac{(-1)^{\boldsymbol{x}^T \boldsymbol{z} + f(\boldsymbol{x})} |\boldsymbol{z}\rangle}{\sqrt{2^n}} \left[ \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \xrightarrow[]{\text{Measure the first } n\text{-qubit system}} \boldsymbol{z}$$

- What's the probability of $\boldsymbol{z} = 00 \ldots 0$ ? What about $\boldsymbol{z} \neq 00 \ldots 0$ ?

$$\sum_{\boldsymbol{x} \in \{0,1\}^n} \frac{(-1)^{\boldsymbol{x}^T \boldsymbol{z} + f(\boldsymbol{x})}}{\sqrt{2^n}}$$

# The Deutsch-Jozsa Algorithm

$$|\psi_3\rangle = \sum_{\boldsymbol{z} \in \{0,1\}^n} \sum_{\boldsymbol{x} \in \{0,1\}^n} \frac{(-1)^{\boldsymbol{x}^T \boldsymbol{z} + f(\boldsymbol{x})} |\boldsymbol{z}\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \xrightarrow[]{\substack{\text{Measure the first} \\ n\text{-qubit system}}} \boldsymbol{z}$$

- What's the probability of $\boldsymbol{z} = 00 \ldots 0$ ? What about $\boldsymbol{z} \neq 00 \ldots 0$ ?

$$\sum_{\boldsymbol{x} \in \{0,1\}^n} \frac{(-1)^{\boldsymbol{x}^T \boldsymbol{z} + f(\boldsymbol{x})}}{\sqrt{2^n}}$$

- What if $f$ is a *zero* function: $\forall \boldsymbol{x} \in \{0,1\}^n, f(\boldsymbol{x}) = 0$? Or $f(\boldsymbol{x}) = 1$?

- What if $f$ is a *non-zero balanced* function: $\sum_{\boldsymbol{x} \in \{0,1\}^n} f(\boldsymbol{x}) = 0$

# Deutsch-Jozsa Problem

- Constant-vs-balanced problem
- Let $f: \{0,1\}^n \rightarrow \{0,1\}$ be a bit function such that it is in either two cases:
  - $f$ is a *constant* function: $\forall \boldsymbol{x} \in \{0,1\}^n, f(\boldsymbol{x})$ is always a constant (0 or 1)
  - $f$ is a *balanced* function: $\sum_{\boldsymbol{x} \in \{0,1\}^n} f(\boldsymbol{x}) = 0$ (i.e., outputs 0 for half the inputs, and 1 for the other half)

- To decide whether $f$ is constant or balanced, **how many times** must we evaluate $f$?

# Deutsch-Jozsa Problem

- Constant-vs-balanced problem
- Let $f: \{0,1\}^n \rightarrow \{0,1\}$ be a bit function such that it is in either two cases:
  - $f$ is a *constant* function: $\forall \boldsymbol{x} \in \{0,1\}^n, f(\boldsymbol{x})$ is always a constant (0 or 1)
  - $f$ is a *balanced* function: $\sum_{x \in \{0,1\}^n} f(\boldsymbol{x}) = 0$ (i.e., outputs 0 for half the inputs, and 1 for the other half)

- To decide whether $f$ is constant or balanced, **how many times** must we evaluate $f$?

**Classical Computer**

Worst-case: $2^n$

*Probabilistic algorithm:*

$l \ll 2^n$ times,

with a failure rate of $\frac{1}{2^l}$

# Deutsch-Jozsa Problem

- Constant-vs-balanced problem
- Let $f: \{0,1\}^n \to \{0,1\}$ be a bit function such that it is in either two cases:
  - $f$ is a *constant* function: $\forall \boldsymbol{x} \in \{0,1\}^n, f(\boldsymbol{x})$ is always a constant (0 or 1)
  - $f$ is a *balanced* function: $\sum_{\boldsymbol{x} \in \{0,1\}^n} f(\boldsymbol{x}) = 0$ (i.e., outputs 0 for half the inputs, and 1 for the other half)

- To decide whether $f$ is constant or balanced, **how many times** must we evaluate $f$?

**Classical Computer**

Worst-case: $2^n$

Probabilistic algorithm:

$l \ll 2^n$ times,

with a failure rate of $\frac{1}{2^l}$

**Quantum Computers:**

Evaluate **once**,

with failure rate $0$

# References

- **[NC00]** *Quantum Computation and Quantum Information.*
  - Sections 1.4.3
- **[KLM07]** *An Introduction to Quantum Computing.*
  - Sections 6.2, 6.3, and 6.4
- **[RP11]** *Quantum Computing: A Gentle Introduction.*
  - Section 7.3.1

# Some Important Information

- **No lectures in the next week!**

- **First Homework Assignment** *(to be announced soon on Moodle)*
  - **Deadline: May 21th, 2025** (next lecture)

- **Final Exam:**
  - The final exam will be a **written exam (**you can bring any paper-based materials)
  - **Your final grade will be based solely on the written exam**
  - There will **be three to four homework assignments** during the course.
  - You **must complete all of them** in order to be eligible to take the final exam.