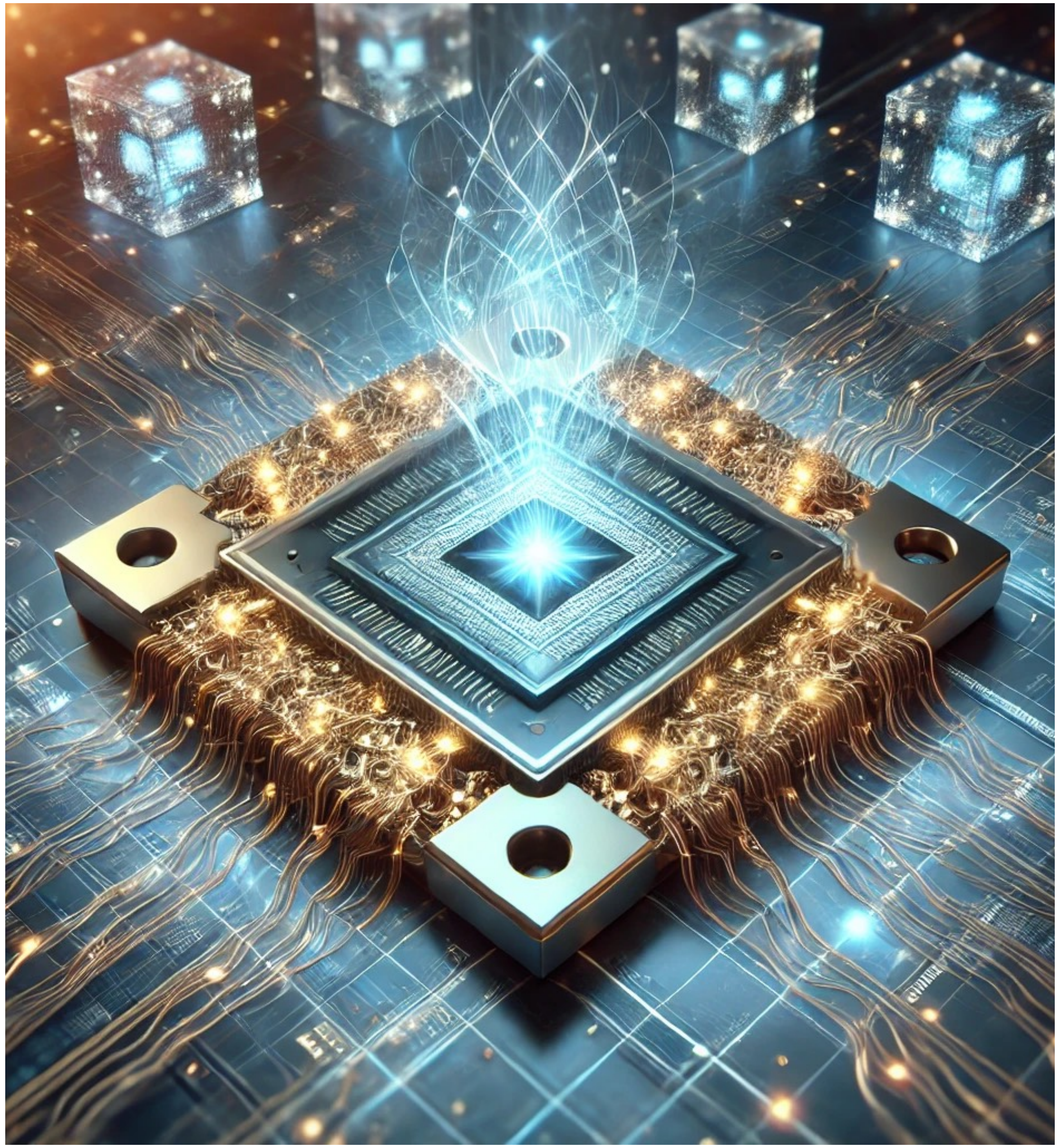
Prague Stringology Conference 2024

Caterina Viola
(Università degli Studi di Catania)

A Quantum Circuit for the Cyclic String Matching Problem

(joint work with Arianna Pavone - UniPA)



Finanziato dall'Unione europea
NextGenerationEU
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing
Italiadomani Piano Nazionale di Ripresa e Resilienza
MUR Ministero dell'Università e della Ricerca
Thanks to the National Centre for HPC, Big Data and Quantum Computing (Project CN00000013 - spoke 10)

The cyclic string matching problem

Input:

- a string $x = x_0 x_1 \cdots x_{m-1}$ of length m (pattern);
- a string $y = y_0 y_1 \cdots y_{n-1}$ of length n (text).

$x, y \in \Sigma^\star$ and $m \ll n$.

Objective: find any occurrence of any cyclic shifting of the pattern within the text,

The cyclic string matching problem

Input:

- a string $x = x_0x_1\cdots x_{m-1}$ of length m (pattern);
- a string $y = y_0y_1\cdots y_{n-1}$ of length n (text).

$x, y \in \Sigma^\star$ and $m \ll n$.

Objective: find any occurrence of any cyclic shifting of the pattern within the text,

i.e., find any $s \in \{0,\ldots,m-1\}$, $j \in \{0,\ldots,n-1\}$ such that

$$R_s(x) := x_sx_{s+1}\cdots x_{n-1}x_0\cdots x_{s-1} = y_jy_{j+1}\cdots y_{j+n-1}\,.$$

The cyclic string matching problem

Example:

- $x =$ AGGCA;
- $y =$ CGACGTGTCCAAGGAGTATCAACGT.

The cyclic string matching problem

Example:

- $x = \mathrm{AGGCA}$;
- $y = \mathrm{CGACGTGTCCAAGGAGTATCAACGT}$.

The cyclic shiftings of x are

$R_0(x) = \mathrm{AGGCA}$, $R_1(x) = \mathrm{GGCAA}$, $R_2(x) = \mathrm{GCAAG}$, $R_3(x) = \mathrm{CAAGG}$, $R_4(x) = \mathrm{AAGGC}$.

The cyclic string matching problem

Example:

- x = AGGCA;
- y = CGACGTGTCCAAGGAGTATCAACGT.

The cyclic shiftings of x are

$R_0(x)$ = AGGCA, $R_1(x)$ = GGCAA, $R_2(x)$ = GCAAG, $R_3(x)$ = CAAGG, $R_4(x)$ = AAGGC.

We find y = CGACGTGTCCAAGGAGTATCAACGT.

The cyclic string matching problem

Example:

- $x =$ AGGCA;
- $y =$ CGACGTGTCCAAGGAGTATCAACGT.

The cyclic shiftings of x are

$R_0(x) =$ AGGCA, $R_1(x) =$ GGCAA, $R_2(x) =$ GCAAG, $R_3(x) =$ CAAGG, $R_4(x) =$ AAGGC.

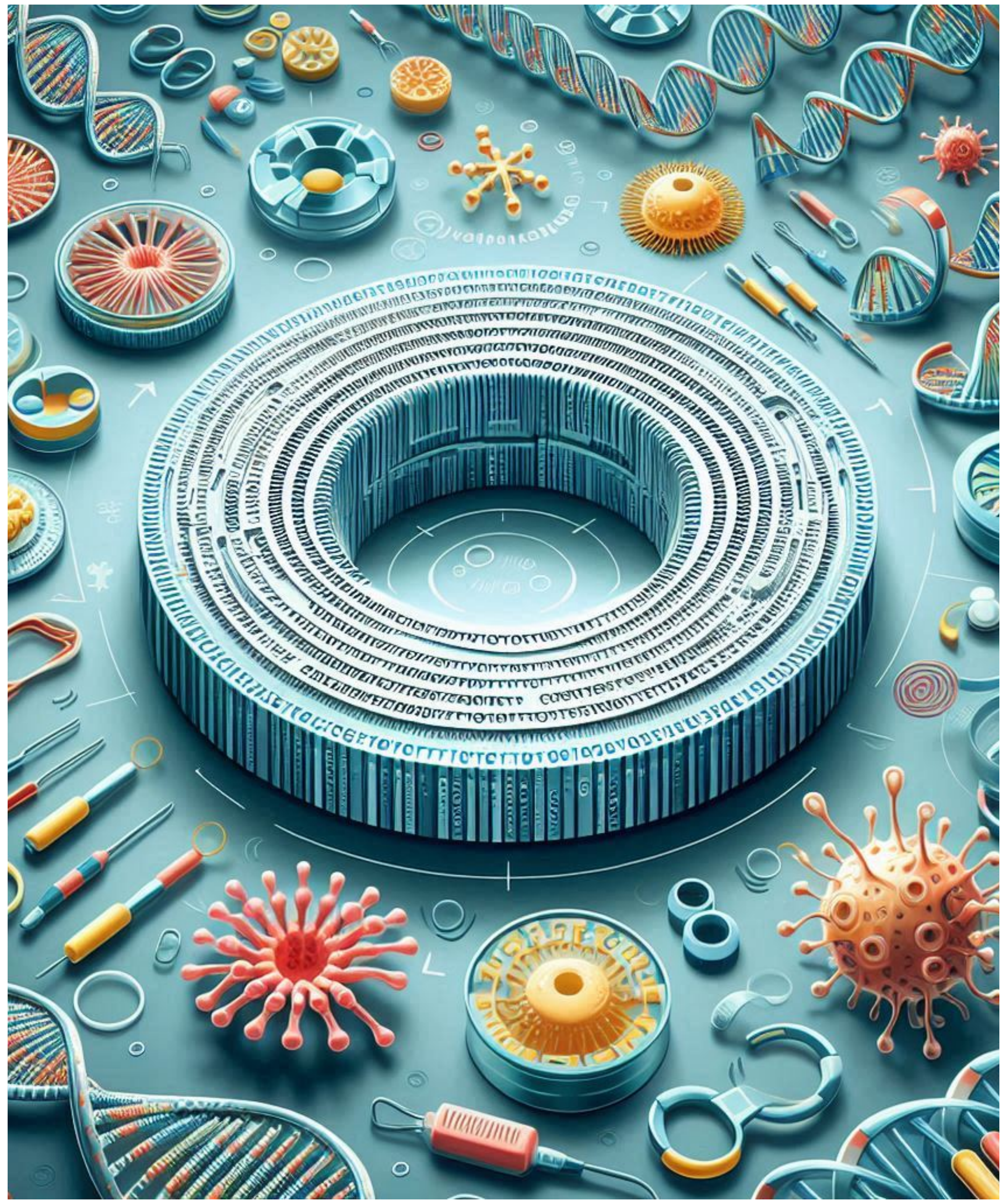
We find $y =$ CGACGTGTCCAAGGAGTATCAACGT.

A solving algorithm should return $s = 3$ and $j = 9$.

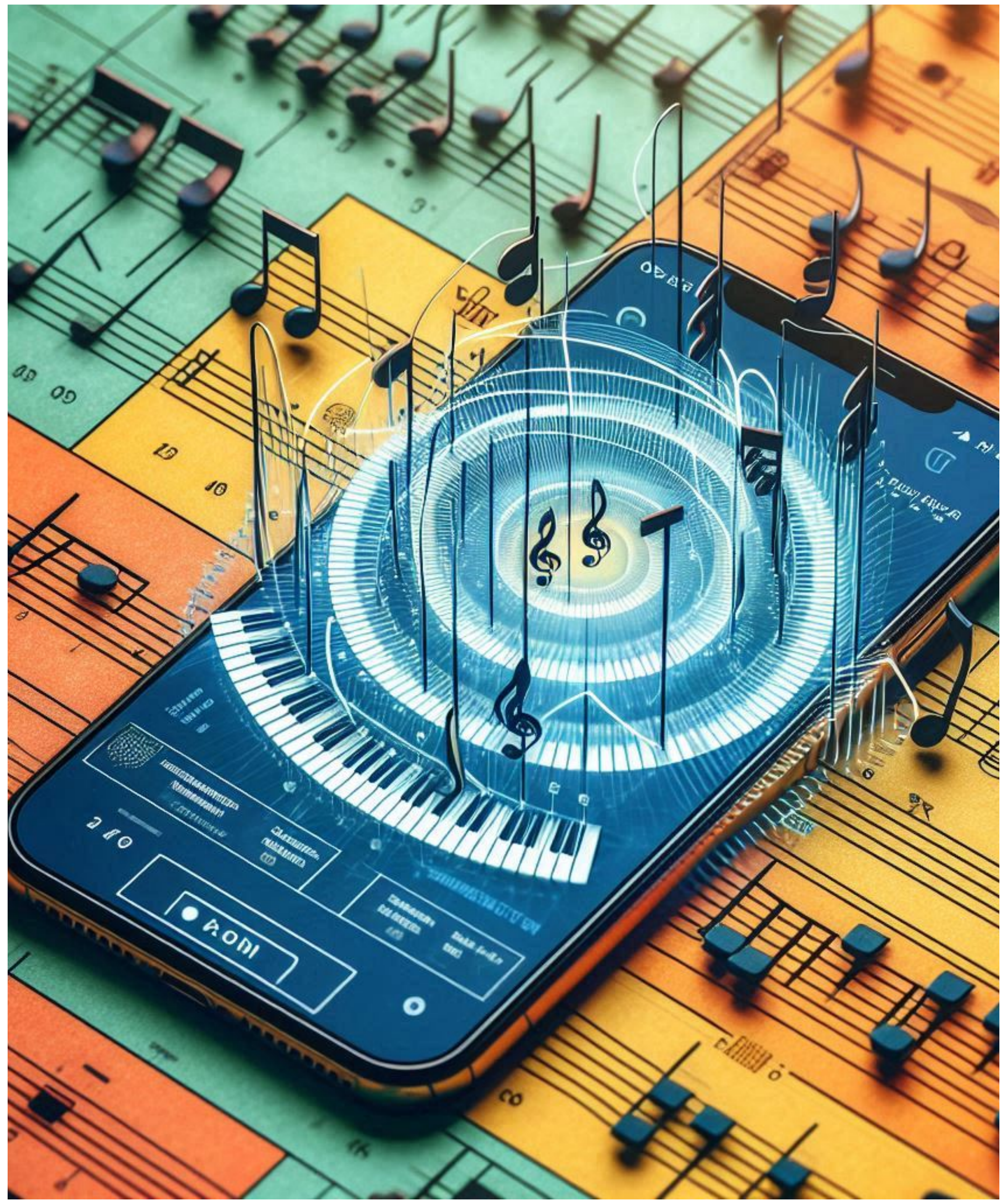
The cyclic string matching problem

A few examples of application

Pattern detection for frequency analysis of encrypted data.



Detecting circular DNA sequences like plasmids and viruses.



Identification of chord progressions or rhythm sequences for music and audio processing.

The most-efficient classical solution

The best-known classical solution is the $\mathcal{O}(n)$-Time algorithm by Lothaire (2005).

- Preprocess x by constructing a suffix automaton of the string xx;
- feed y into the automaton;
- the lengths of the longest factors of xx occurring in y can be found by the links followed in the automaton in time $\mathcal{O}(n)$.

Our result

We designed a quantum algorithm for the cyclic string matching problem.

Our result

We designed a quantum algorithm for the cyclic string matching problem.

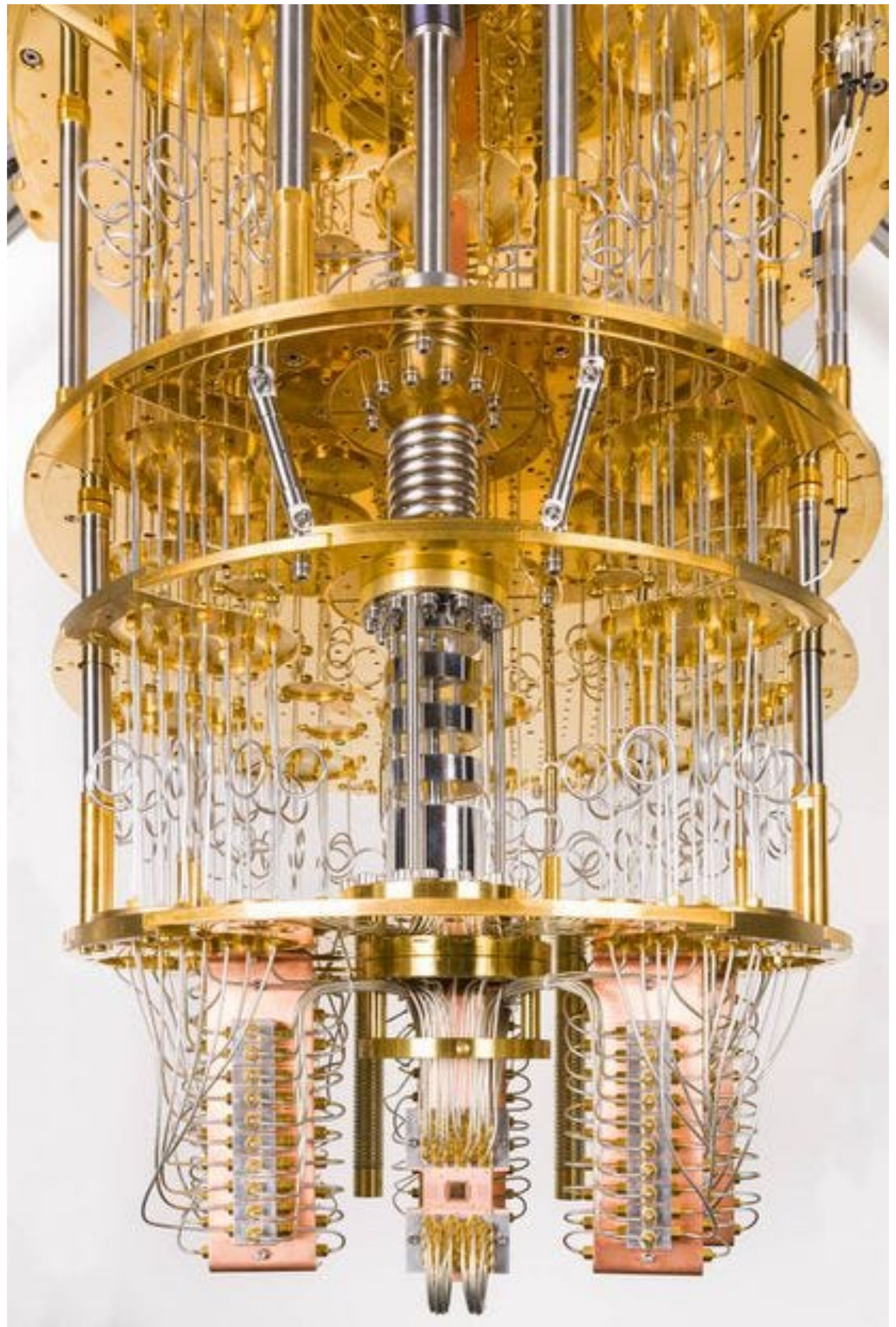
Our algorithm is formalised as a quantum circuit with

- $\tilde{\mathcal{O}}(\sqrt{n})$-Depth;
- $\mathcal{O}(n)$-Size.

It requires quadratically fewer time-steps than the most efficient counterpart algorithm running on a classical machine.

Quantum computing

Quantum algorithms are designed to take full advantage of the computational power of quantum machines.



A quantum computing machine works as predicted by the theory of quantum mechanics.

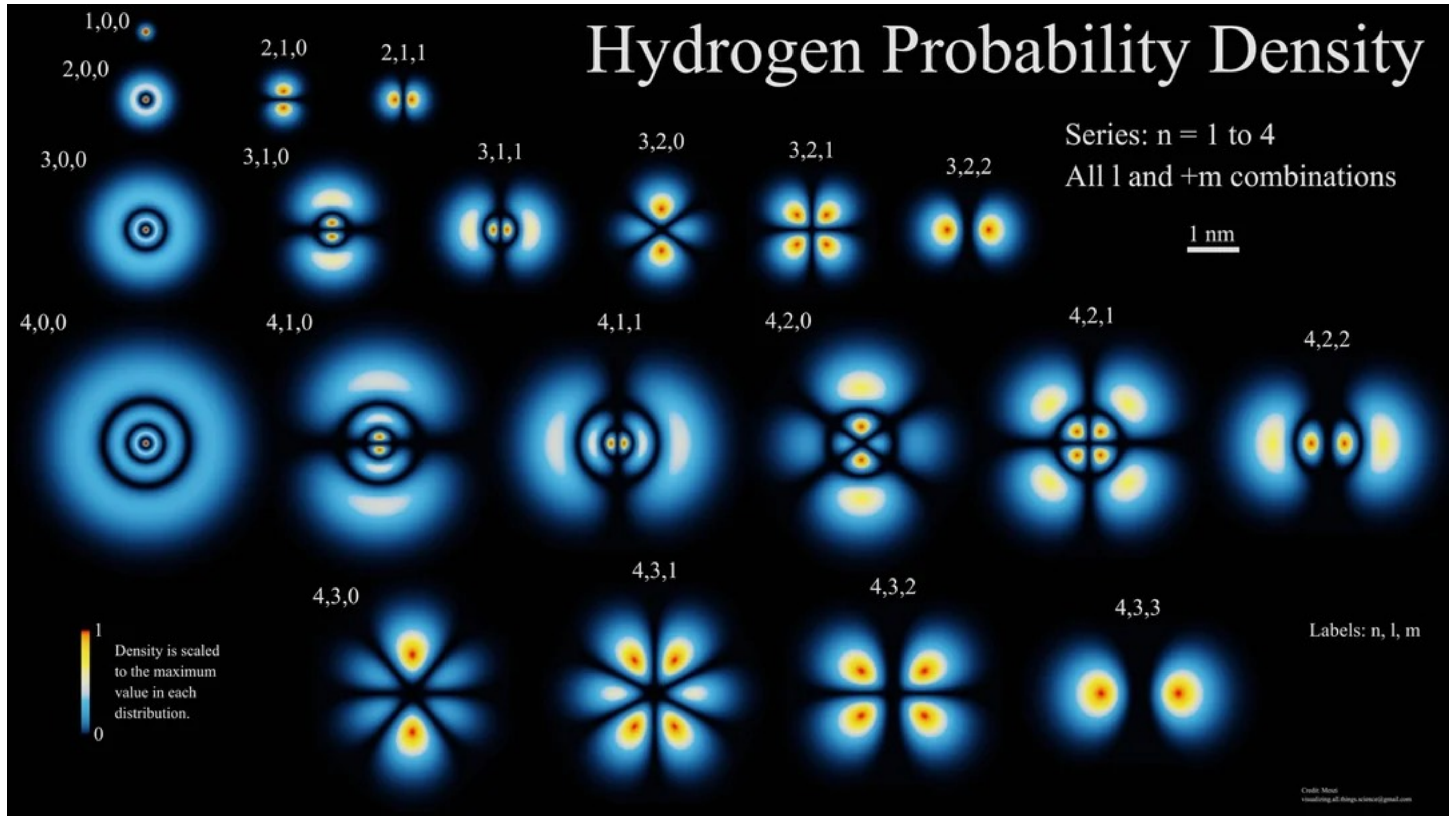
Quantum computing

In a quantum machine, the fundamental information units are simulated by particles whose size is at and below the scale of atoms.

Quantum computing

In a quantum machine, the fundamental information units are simulated by particles whose size is at and below the scale of atoms.

The state of each of such particles is described by a wave function, which can be thought of as a probability distribution of the classical states in which the particle can be observed.



Quantum bits

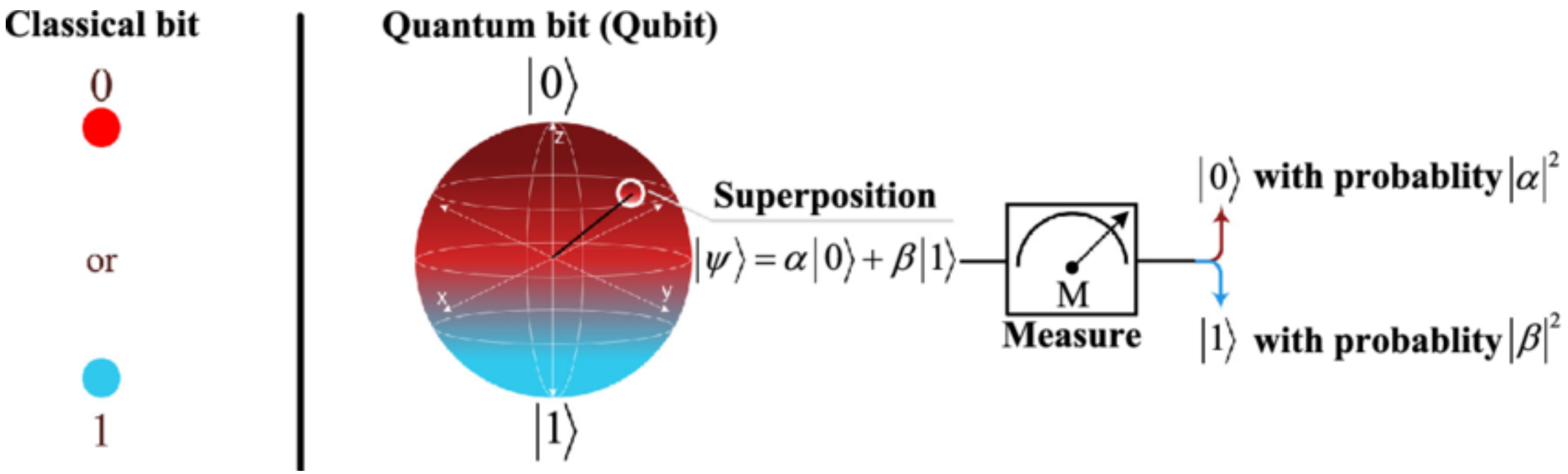
The fundamental unit in quantum information is the qubit.

Quantum bits

The fundamental unit in quantum information is the qubit.

$$|q\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}: \ |\alpha|^2 + |\beta|^2 = 1.$$

Mathematically, a qubit $|q\rangle$ is an element from the 2-dimensional Hilbert space, $\mathscr{H}$, over the complex field.



Multiple systems of qubits

A multiple system of qubits taken together is referred to as a quantum register.

$$|k\rangle = |q_0, q_1, \ldots, q_{n-1}\rangle = \sum_{j=0}^{2^n-1} \alpha_j |k_j\rangle,$$

Multiple systems of qubits

A multiple system of qubits taken together is referred to as a quantum register.

$$|k\rangle = |q_0, q_1, \ldots, q_{n-1}\rangle = \sum_{j=0}^{2^n-1} \alpha_j |k_j\rangle,$$

- where $\alpha_j \in \mathbb{C}$: $\sum_{j=0}^{2^n-1} |\alpha_j|^2 = 1$,
- k_j is the binary encoding of the jth smallest measurable value of k, and
- q_i is the ith least important qubit of k.

Multiple systems of qubits

A multiple system of qubits taken together is referred to as a quantum register.

$$|k\rangle = |q_0, q_1, \ldots, q_{n-1}\rangle = \sum_{j=0}^{2^n-1} \alpha_j |k_j\rangle,$$

- where $\alpha_j \in \mathbb{C}$: $\sum_{j=0}^{2^n-1} |\alpha_j|^2 = 1$,
- k_j is the binary encoding of the jth smallest measurable value of k, and
- q_i is the ith least important qubit of k.

Two qubits can be entangled, that is the wave functions describing their respective states can be correlated and, therefore, not independent. Mathematically,

$$|k\rangle \in \bigotimes_{i=0}^{n-1} \mathscr{H} \, .$$

Quantum operations

A quantum operation is a transformation of the state of a quantum register that is allowed by quantum mechanics.

Quantum operations

A quantum operation is a transformation of the state of a quantum register that is allowed by quantum mechanics.

Methematically, a n-ary quantum operation on a n-qubits register is a unitary transformation, i.e., a linear bounded operator $U\colon \mathscr{H}^n \to \mathscr{H}^n$ such that $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the complex conjugate of the transpose of U.

Quantum operations

A quantum operation is a transformation of the state of a quantum register that is allowed by quantum mechanics.

Methematically, a n-ary quantum operation on a n-qubits register is a unitary transformation, i.e., a linear bounded operator $U\colon \mathscr{H}^n \to \mathscr{H}^n$ such that $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the complex conjugate of the transpose of U.

There is a great variety of quantum operations, however each of them can be obtained as the composition of at most binary quantum operations.

Exploiting quantum mechanics to solve the cyclic string matching problem

Example:

Pattern: bac

Text: $abbac$

$m = 3, n = 5$

a	b	b	a	c
b	b	a	c	a
b	a	c	a	b
a	c	a	b	b
c	a	b	b	a

b	a	c
a	c	b
c	b	a

Exploiting quantum mechanics to solve the cyclic string matching problem

Example:

Pattern: bac

Text: $abbac$

$m = 3, n = 5$

a	b	b	a	c
b	b	a	c	a
b	a	c	a	b
a	c	a	b	b
c	a	b	b	a

is it a match ?

b	a	c
a	c	b
c	b	a

Exploiting quantum mechanics to solve the cyclic string matching problem

Example:

Pattern: *bac*

Text: *abbac*

$m = 3, n = 5$

a	b	b	a	c
b	b	a	c	a
b	a	c	a	b
a	c	a	b	b
c	a	b	b	a

is it a match ?

b	a	c
a	c	b
c	b	a

The idea is to harness quantum mechanics and check wether the first m qubits of the register containing the superposition of all possible cyclic shiftings of the text matches the register containing the superposition of all possible cyclic shiftings of the pattern.

Our contribution

Our contribution is a refinement of an algorithm from Niroula&Nam's for exact string matching.

Our algorithm uses a well known quantum algorithm that searches for a desired item within an unstructured database of items: Grover's search algorithm (1996).

There are several models of quantum computation. The model we adopt is that of quantum circuits.

Quantum circuits

A quantum circuit can be represented as a direct acyclic graph whose nodes are to be interpreted as the gates that operate on the information carried by the edges.

Quantum circuits

A quantum circuit can be represented as a direct acyclic graph whose nodes are to be interpreted as the gates that operate on the information carried by the edges.

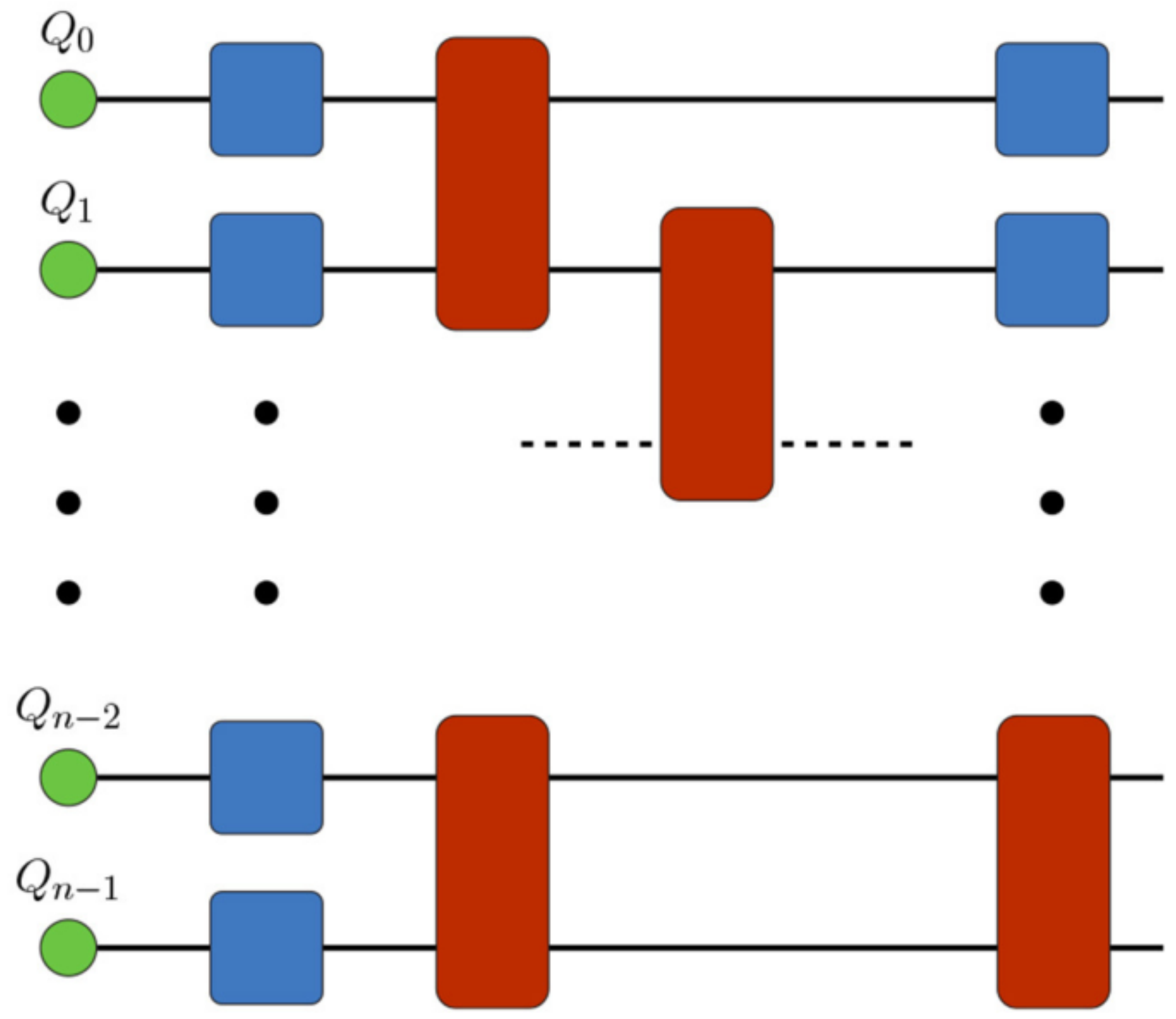
- Each gate operate an elementary operation on at most two qubits.
- A quantum circuit needs to be reversible, this means, in particular, that for each gate the number of input edges equals that of output edges.
- Because of the No-Cloning Theorem, we cannot either copy or fan-out the information carried by an edge. To overcome this problem is possible to use some ancilla qubits.
- Usually at the end of a circuit, one or more qubits are measured.

Quantum circuits

A quantum circuit can be represented as a direct acyclic graph whose nodes are to be interpreted as the gates that operate on the information carried by the edges.

- Each gate operate an elementary operation on at most two qubits.
- A quantum circuit needs to be reversible, this means, in particular, that for each gate the number of input edges equals that of output edges.
- Because of the No-Cloning Theorem, we cannot either copy or fan-out the information carried by an edge. To overcome this problem is possible to use some ancilla qubits.
- Usually at the end of a circuit, one or more qubits are measured.

A quantum circuit can be seen as a sequence of parallel wires (each corresponding to a qubit) passing through certain gates that operate on them.



Quantum circuits

We measure the complexity of a quantum circuit by its **depth** (corresponding to the number of time-steps needed).

Other important measures of the complexity of a quantum circuit are the size (number of elementary gates) and the space (number of qubits).

Quantum circuits

We measure the complexity of a quantum circuit by its **depth** (corresponding to the number of time-steps needed).

Other important measures of the complexity of a quantum circuit are the size (number of elementary gates) and the space (number of qubits).

There is a great variety of elementary quantum gates. We list a few of them.

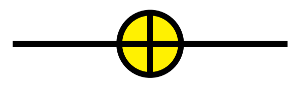
Quantum circuits

We measure the complexity of a quantum circuit by its **depth** (corresponding to the number of time-steps needed).

Other important measures of the complexity of a quantum circuit are the size (number of elementary gates) and the space (number of qubits).

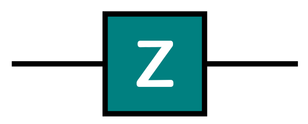
There is a great variety of elementary quantum gates. We list a few of them.

Pauli X



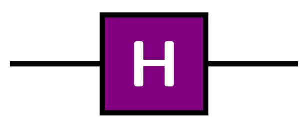
$$|0\rangle \mapsto |1\rangle, \quad |1\rangle \mapsto |0\rangle$$

Pauli Z



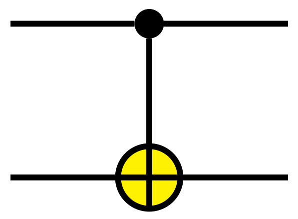
$$|0\rangle \mapsto |0\rangle, \quad |1\rangle \mapsto -|1\rangle$$

Hadamard



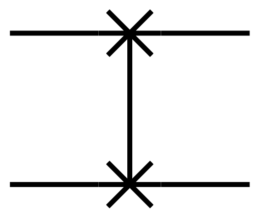
$$|0\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = |+\rangle, \quad |1\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = |-\rangle$$

CNOT



$$|q_0, q_1\rangle \mapsto |q_0, q_0 \oplus q_1\rangle$$

Swap



$$|q_0, q_1\rangle \mapsto |q_1, q_0\rangle$$

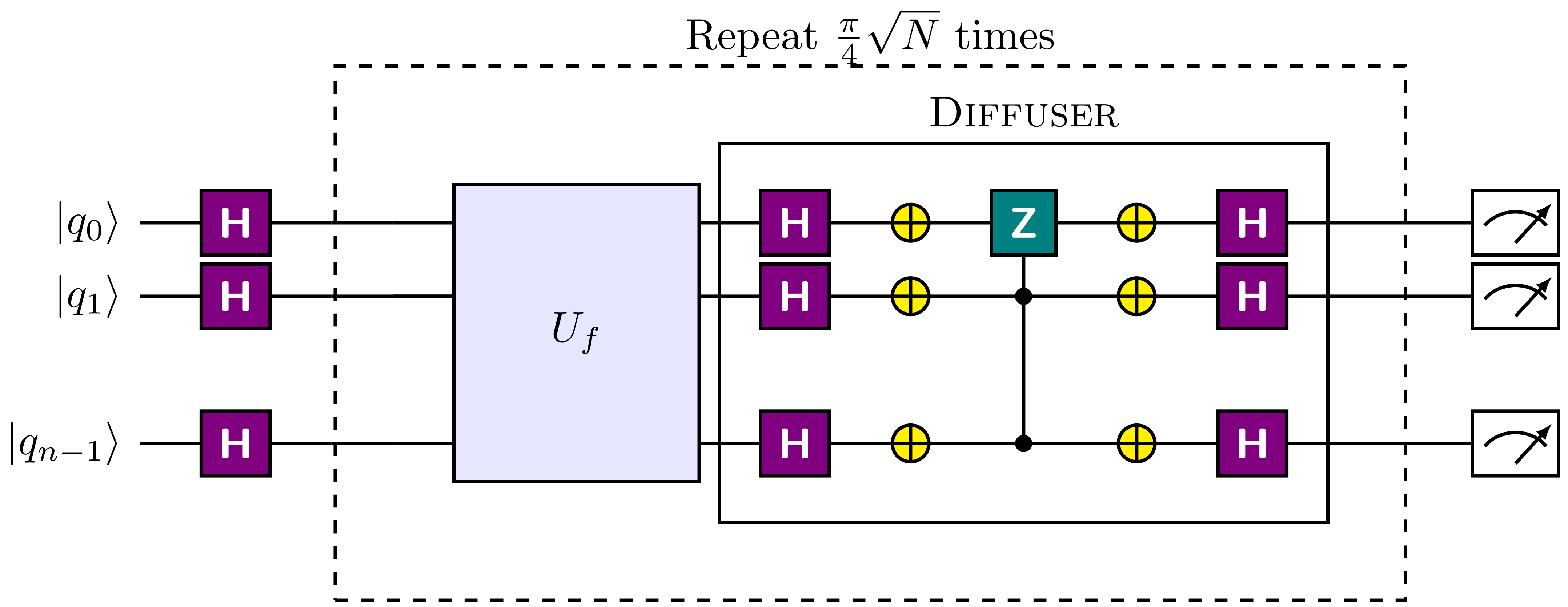
Grover's search algorithm

It searches a black-box list of N items in $\tilde{\mathcal{O}}(\sqrt{N})$-time, whereas classical algorithms need $\mathcal{O}(N)$-time.

Grover's search algorithm

It searches a black-box list of N items in $\tilde{\mathcal{O}}(\sqrt{N})$-time, whereas classical algorithms need $\mathcal{O}(N)$-time.

Let $f\colon \{0,1\}^n \to \{0,1\}$ such that $\exists!\ x^\star : f(x^\star) = 1$.
We have to search in a space of $N = 2^n$ items.

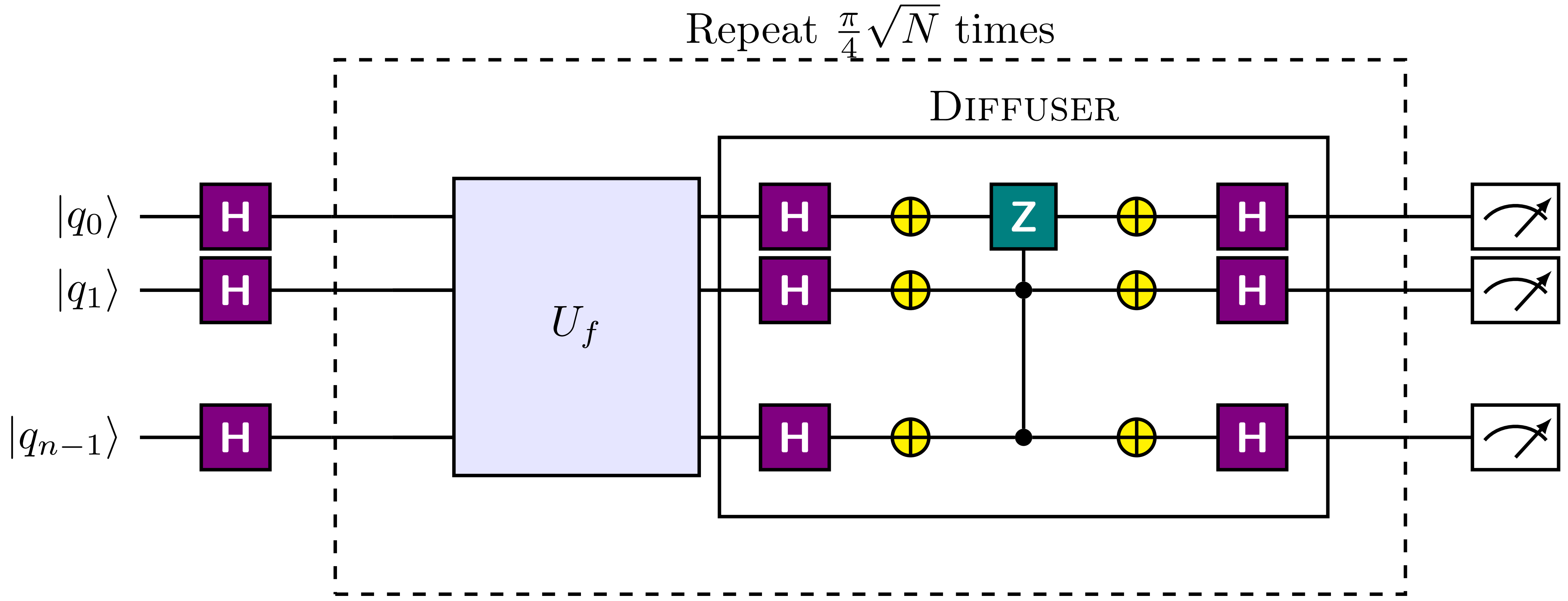


Grover's search algorithm

It searches a black-box list of N items in $\tilde{\mathcal{O}}(\sqrt{N})$-time, whereas classical algorithms need $\mathcal{O}(N)$-time.

Let $f\colon \{0,1\}^n \to \{0,1\}$ such that $\exists!\ x^\star : f(x^\star) = 1$.

We have to search in a space of $N = 2^n$ items.



- Initialisation: we start with the register $|0\rangle^{\otimes n}$, in which each qubit initialized to $|0\rangle$, and apply the Hadamard gate to each qubit, obtaining a superposition of all possible items $x_0, \ldots, x_{N-1}$, i.e. $\sum_{i=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x_i\rangle$.

- Iterative phase (performs a rotation of $\frac{2}{\sqrt{N}}$ rads):
  - Phase oracle: $U_f|x\rangle = (-1)^{f(x)}|x\rangle$. It flips the phase of $|x^\star\rangle$.
  - Diffuser: it simulates a reflection wrt $|+\rangle^{\otimes n}$.

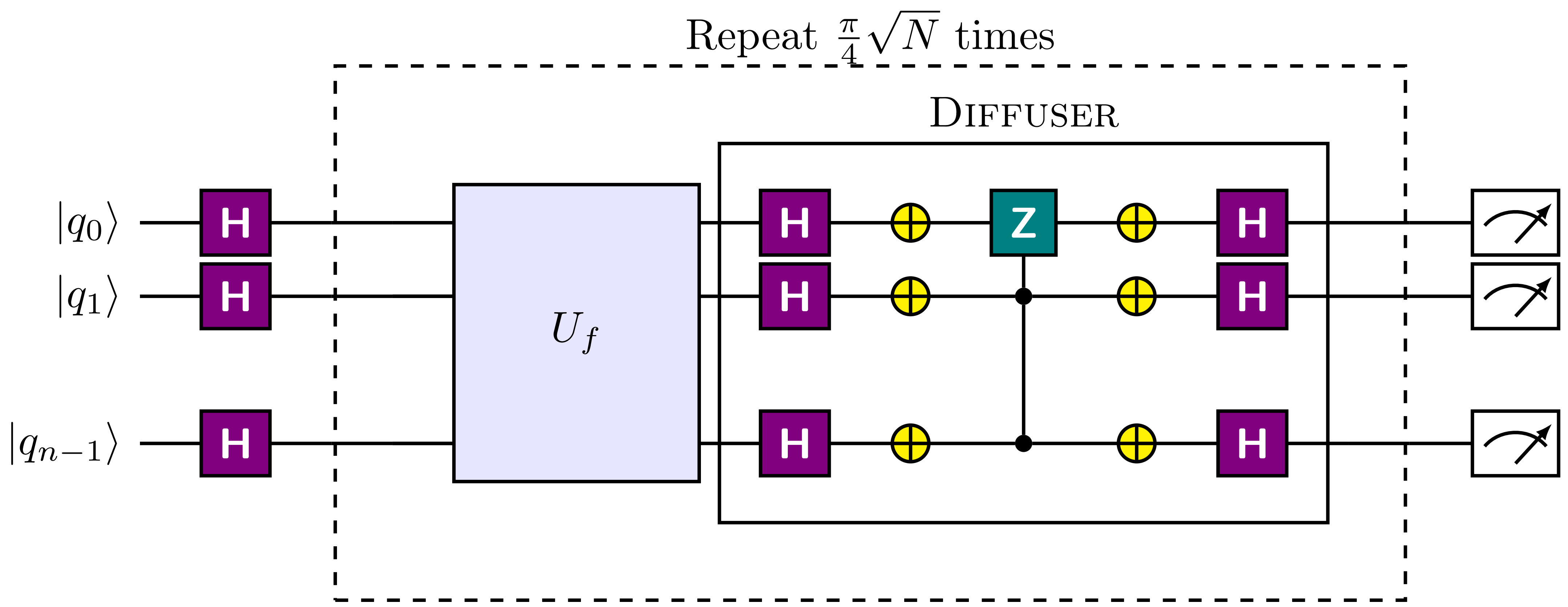
$\Big\}\ \frac{\pi}{4}\sqrt{N}$ times

Grover's search algorithm

It searches a black-box list of N items in $\tilde{\mathcal{O}}(\sqrt{N})$-time, whereas classical algorithms need $\mathcal{O}(N)$-time.

Let $f\colon \{0,1\}^n \to \{0,1\}$ such that $\exists!\, x^\star : f(x^\star) = 1$.

We have to search in a space of $N = 2^n$ items.



- Initialisation: we start with the register $|0\rangle^{\otimes n}$, in which each qubit initialized to $|0\rangle$, and apply the Hadamard gate to each qubit, obtaining a superposition of all possible items $x_0, \dots, x_{N-1}$, i.e. $\displaystyle\sum_{i=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x_i\rangle$.

- Iterative phase (performs a rotation of $\dfrac{2}{\sqrt{N}}$ rads):
  - Phase oracle: $U_f|x\rangle = (-1)^{f(x)}|x\rangle$. It flips the phase of $|x^\star\rangle$.
  - Diffuser: it simulates a reflection wrt $|+\rangle^{\otimes n}$.

$\frac{\pi}{4}\sqrt{N}$ times

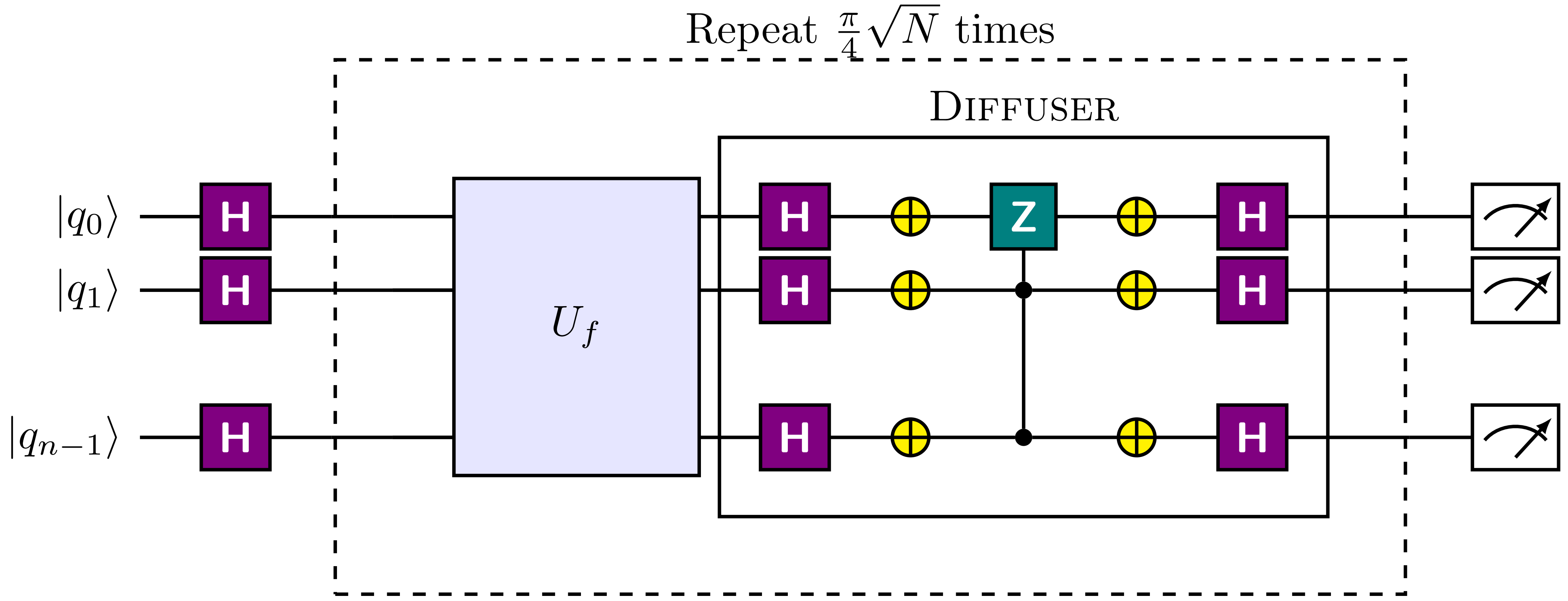
- Quantum measure. With high probability we measure the objective vector.

Grover's search algorithm

It searches a black-box list of N items in $\tilde{\mathcal{O}}(\sqrt{N})$-time, whereas classical algorithms need $\mathcal{O}(N)$-time.

Let $f\colon \{0,1\}^n \to \{0,1\}$ such that $\exists!\ x^\star : f(x^\star) = 1$.

We have to search in a space of $N = 2^n$ items.



- Initialisation: we start with the register $|0\rangle^{\otimes n}$, in which each qubit initialized to $|0\rangle$, and apply the Hadamard gate to each qubit, obtaining a superposition of all possible items $x_0, \ldots, x_{N-1}$, i.e. $\sum_{i=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x_i\rangle$.

- Iterative phase (performs a rotation of $\frac{2}{\sqrt{N}}$ rads):
  - Phase oracle: $U_f|x\rangle = (-1)^{f(x)}|x\rangle$. It flips the phase of $|x^\star\rangle$.
  - Diffuser: it simulates a reflection wrt $|+\rangle^{\otimes n}$.

$\frac{\pi}{4}\sqrt{N}$ times

- Quantum measure. With high probability we measure the objective vector.

Depth: $\mathcal{O}\left(\sqrt{N}\,\left(T(n) + \log(n)\right)\right)$, where $T(n)$ is the depth of the phase oracle, and $\log(n)$ is the depth of the multi-controlled Z gate.

A quantum circuit for cyclic string matching

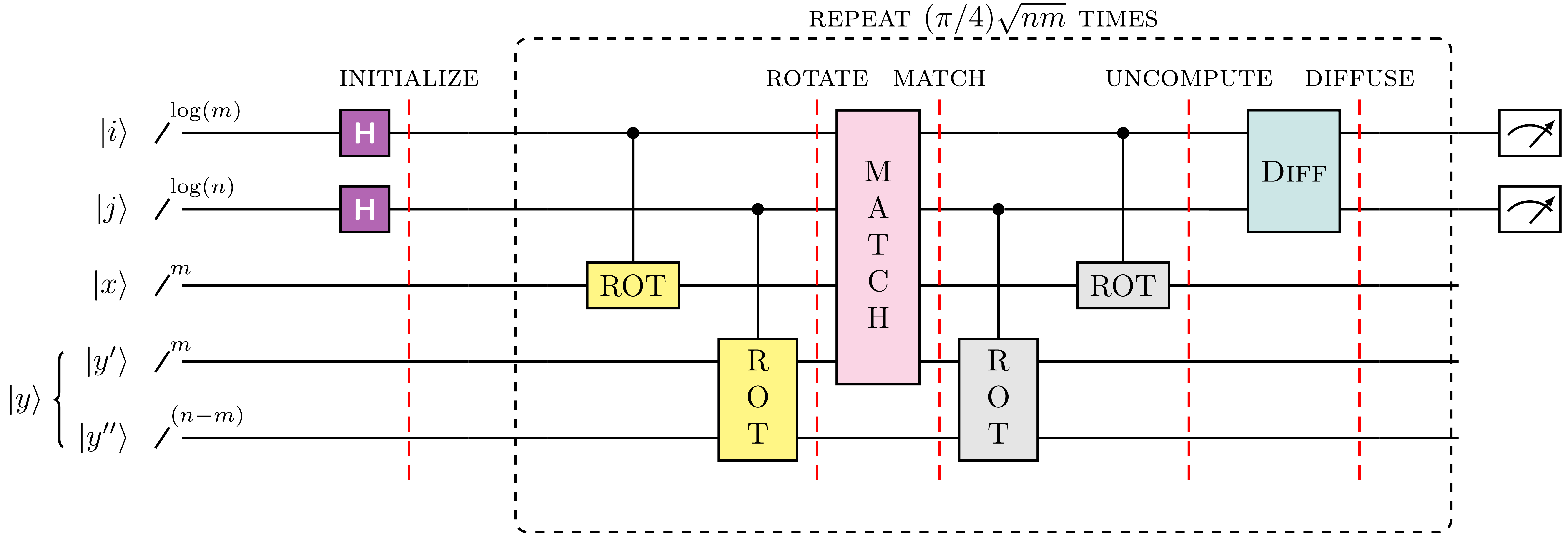
Input: $|\varphi\rangle = |i\rangle \otimes |j\rangle \otimes |x\rangle \otimes |y\rangle$.

- $|x\rangle$: m-qubit register containing the characters from the pattern.
- $|y\rangle$: n-qubit register containing the characters from the text.
- $|i\rangle = |0\rangle^{\otimes \log(m)}$.
- $|j\rangle = |0\rangle^{\otimes \log(n)}$.

A quantum circuit for cyclic string matching

Input: $|\varphi\rangle = |i\rangle \otimes |j\rangle \otimes |x\rangle \otimes |y\rangle$.

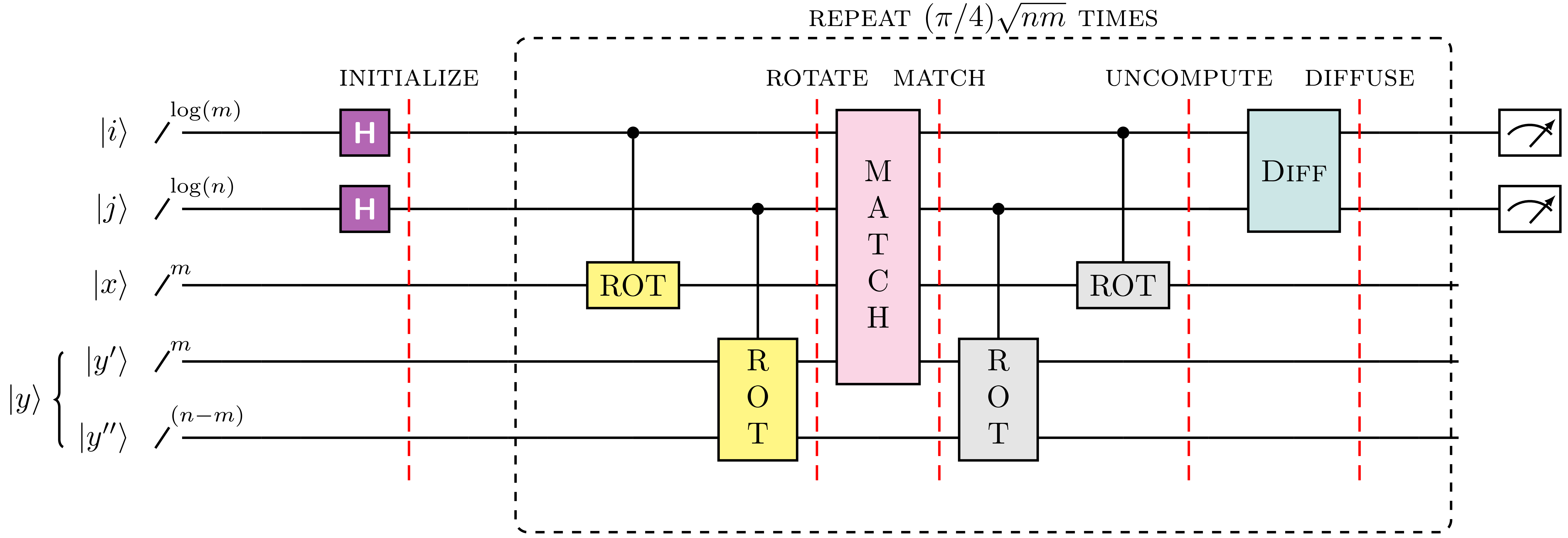
- $|x\rangle$: m-qubit register containing the characters from the pattern.
- $|y\rangle$: n-qubit register containing the characters from the text.
- $|i\rangle = |0\rangle^{\otimes \log(m)}$.
- $|j\rangle = |0\rangle^{\otimes \log(n)}$.



A quantum circuit for cyclic string matching

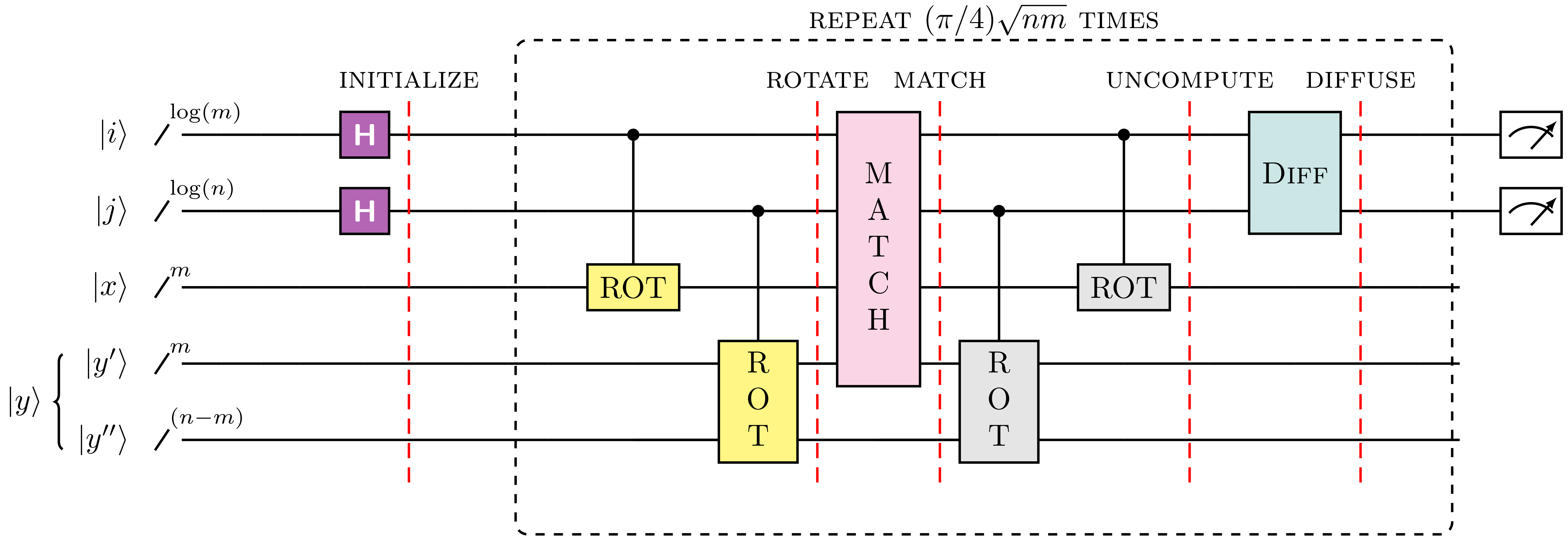
Input: $|\varphi\rangle = |i\rangle \otimes |j\rangle \otimes |x\rangle \otimes |y\rangle$.

- $|x\rangle$: m-qubit register containing the characters from the pattern.
- $|y\rangle$: n-qubit register containing the characters from the text.
- $|i\rangle = |0\rangle^{\otimes \log(m)}$.
- $|j\rangle = |0\rangle^{\otimes \log(n)}$.



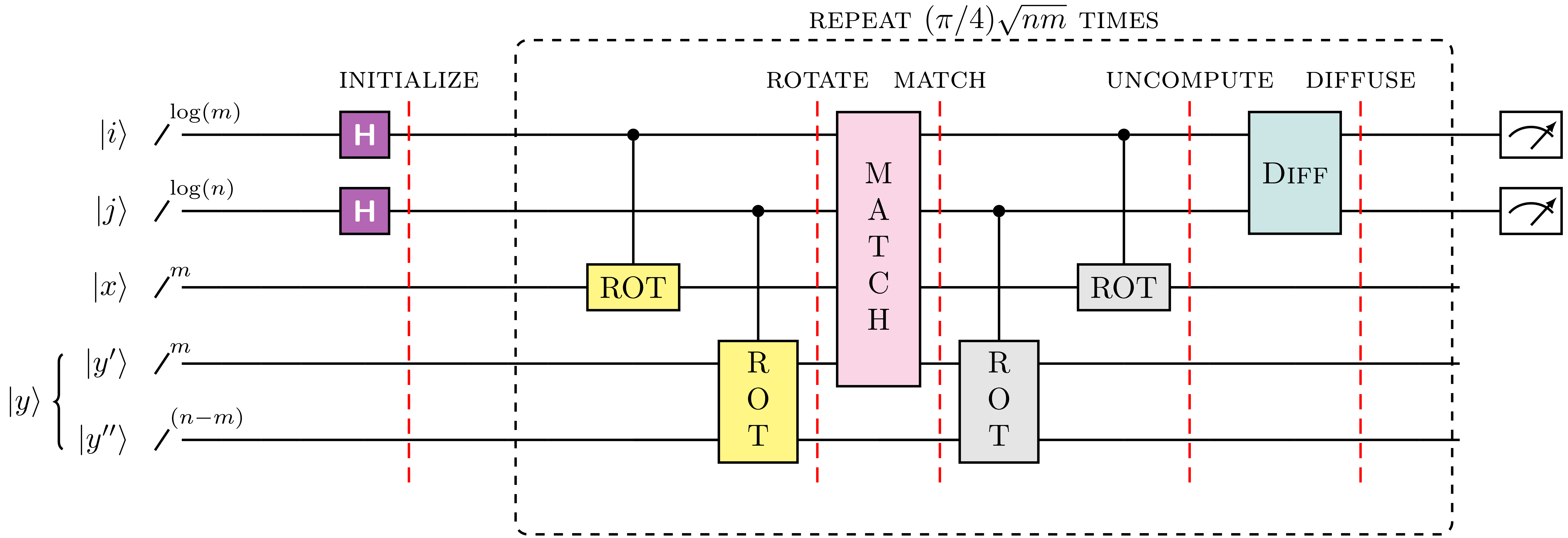
During the initialisation, the algorithm applies the Hadamard operator (H) to all the qubits in $|i\rangle$ and $|j\rangle$ so that their states are the superpositions of all possible shift values between 0 and $m-1$ and between 0 and $n-1$, respectively.

A quantum circuit for cyclic string matching



ROT: the controlled cyclic shift operator. After its application, the registers $|x\rangle$ and $|y\rangle$ are in the state of superposition of all their respective cyclic shiftings.

A quantum circuit for cyclic string matching

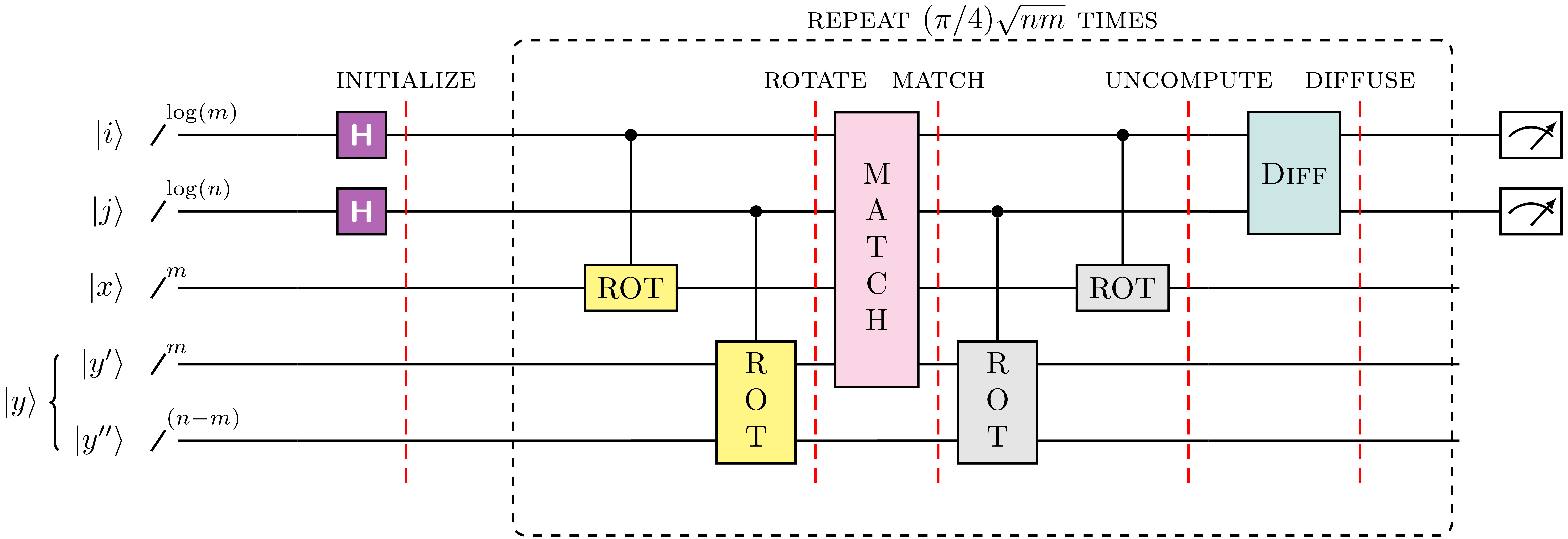


ROT: the controlled cyclic shift operator. After its application, the registers $|x\rangle$ and $|y\rangle$ are in the state of superposition of all their respective cyclic shiftings.

MATCH: a phase oracle for exact string matching, i.e., for the function $f\colon \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ such that

$$f(x, y[0..m-1]) = \begin{cases} 1 & \text{if } x_i = y_i \text{ for all } 0 \le i < m \\ 0 & \text{otherwise.} \end{cases}$$

A quantum circuit for cyclic string matching



ROT: the controlled cyclic shift operator. After its application, the registers $|x\rangle$ and $|y\rangle$ are in the state of superposition of all their respective cyclic shiftings.

MATCH: a phase oracle for exact string matching, i.e., for the function
$f\colon \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$
such that

$$f(x, y[0..m-1]) = \begin{cases} 1 & \text{if } x_i = y_i \text{ for all } 0 \leq i < m \\ 0 & \text{otherwise.} \end{cases}$$

DIFF: Grover's diffuser.

The cyclic shifting operator

Given an input n-qubit register $|q\rangle$, a **cyclic shift operator** R_s applies a rightward shift of s positions.

Formally,

$$|q_0, q_1, \ldots, q_{n-1}\rangle \mapsto |q_s, q_{s+1}, \ldots, q_{n-1}, q_0, q_1, \ldots, q_{s-1}\rangle\,.$$

The cyclic shifting operator

Given an input n-qubit register $|q\rangle$, a **cyclic shift operator** R_s applies a rightward shift of s positions.

Formally,

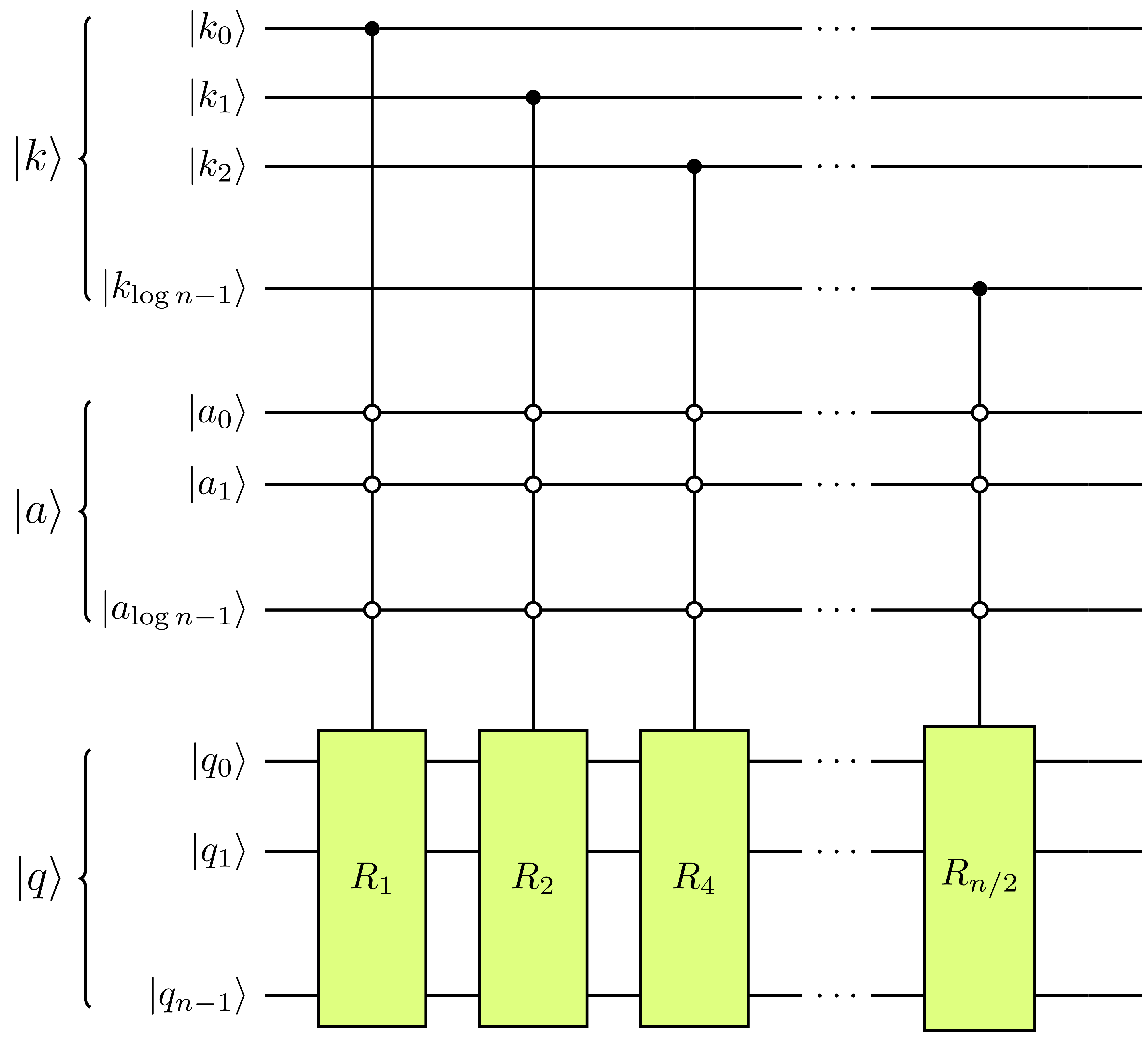
$$|q_0, q_1, \ldots, q_{n-1}\rangle \mapsto |q_s, q_{s+1}, \ldots, q_{n-1}, q_0, q_1, \ldots, q_{s-1}\rangle \, .$$

For every (n, s) , there exists a $\mathcal{O}(\log(n))$**-depth** quantum circuit that applies R_s to an input n-qubit register.

Such a circuit run in at most $\log(n)$ time-steps; the jth time-step consists of $\frac{n}{2^{j+1}}$ parallel swaps.

The controlled cyclic shifting operator

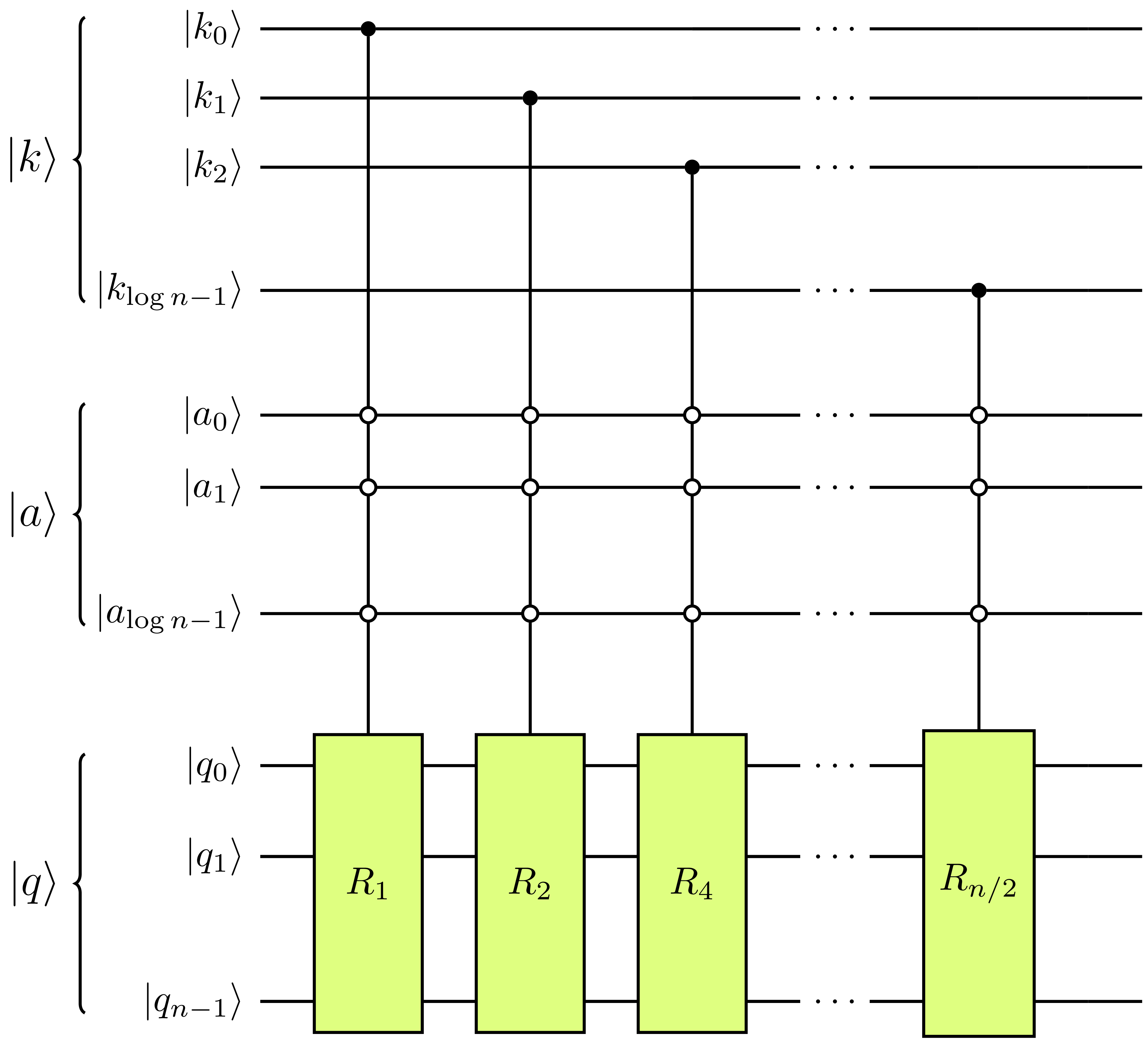
ROT creates a superposition of all circular rotations of the text y. It **applies to an input n-qubit register $|q\rangle$, a cyclic shifting of a number of positions that depends on a second input register $|k\rangle$ of length $\log(n)$.**



The controlled cyclic shifting operator

ROT creates a superposition of all circular rotations of the text y. It **applies to an input n-qubit register $|q\rangle$, a cyclic shifting of a number of positions that depends on a second input register $|k\rangle$ of length $\log(n)$.**

Since $|k\rangle = \bigotimes_{i=0}^{\log(n)-1} |k_i\rangle$, the operator ROT can be implemented by applying to $|q\rangle$ the operator R_{2^i} controlled by $|k_i\rangle$, for each $0 \leq i < \log(n)$. It requires $\log(n)$ ancilla qubits.

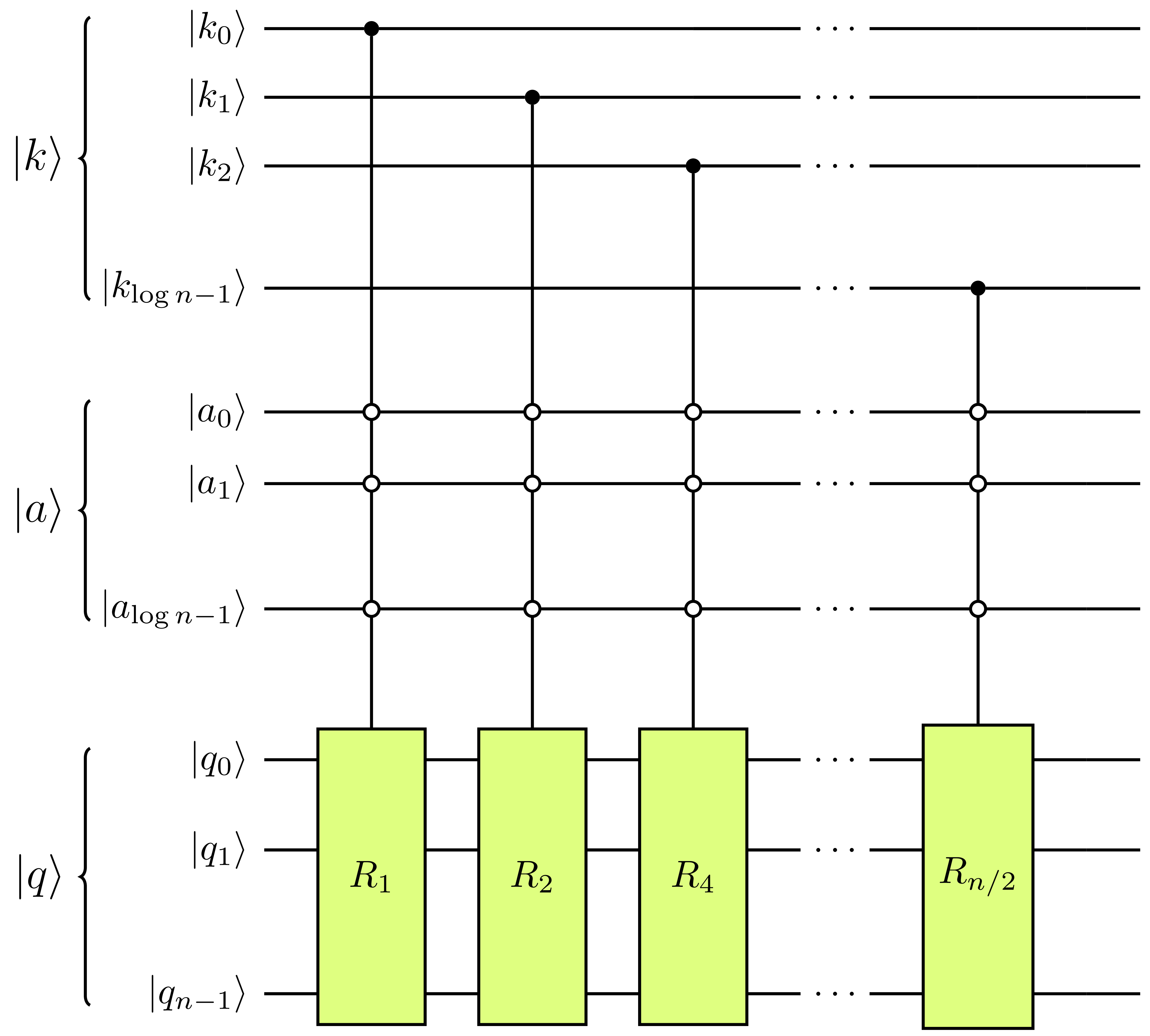


The controlled cyclic shifting operator

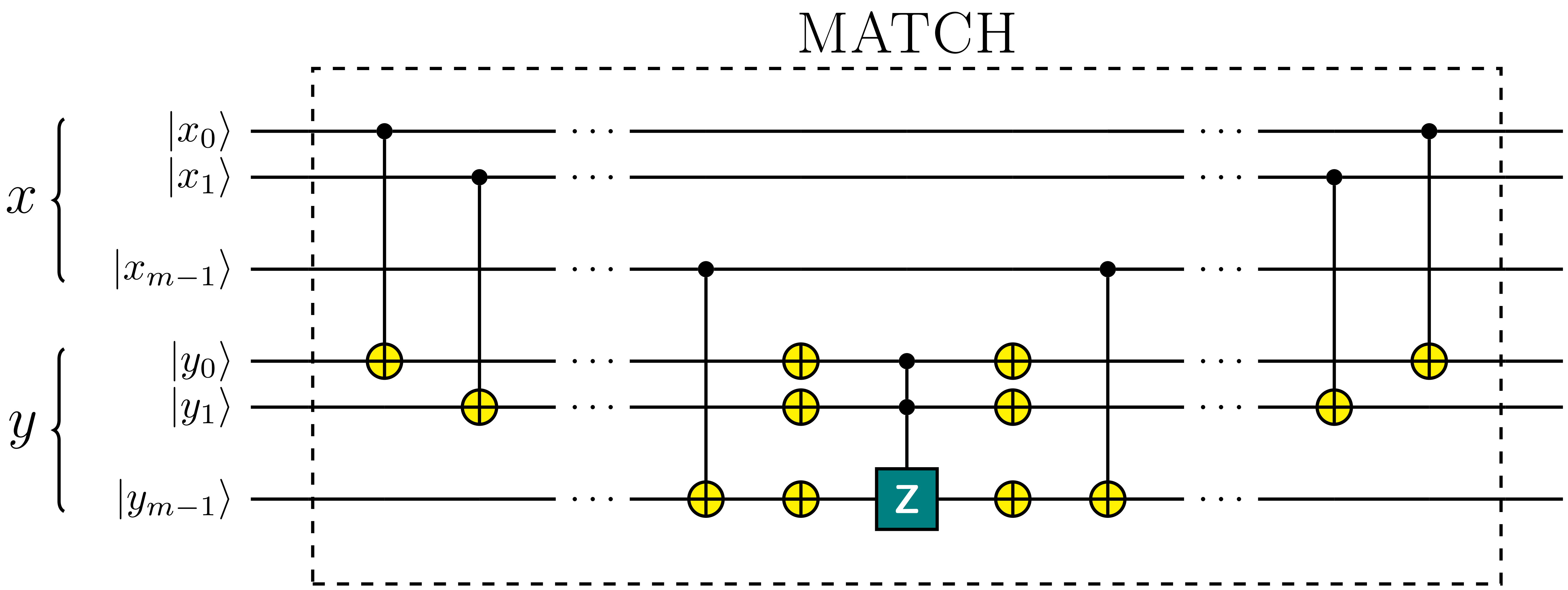
ROT creates a superposition of all circular rotations of the text y. It **applies to an input n-qubit register $|q\rangle$, a cyclic shifting of a number of positions that depends on a second input register $|k\rangle$ of length $\log(n)$.**

Since $|k\rangle = \bigotimes_{i=0}^{\log(n)-1} |k_i\rangle$, the operator ROT can be implemented by applying to $|q\rangle$ the operator R_{2^i} controlled by $|k_i\rangle$, for each $0 \leq i < \log(n)$. It requires $\log(n)$ ancilla qubits.

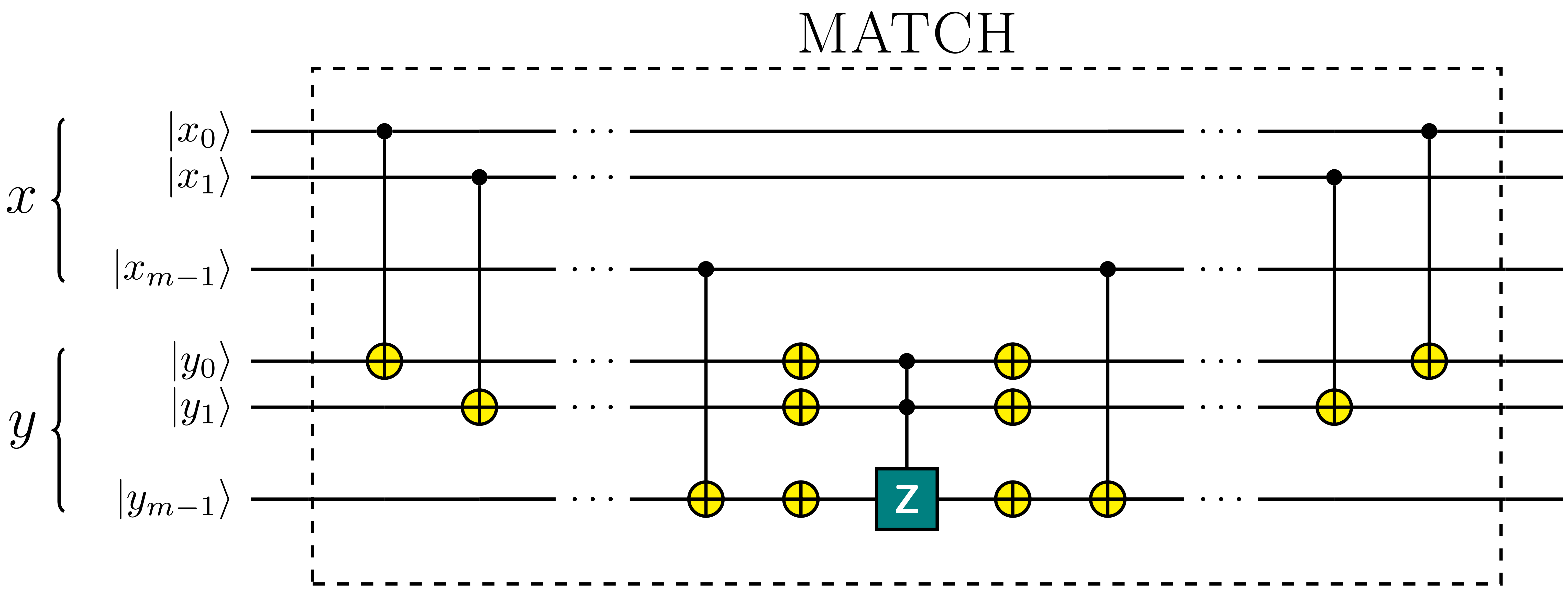
The **depth** of such a circuit implementing ROT is $\mathcal{O}(\log^2(n))$.



A phase oracle for exact string matching



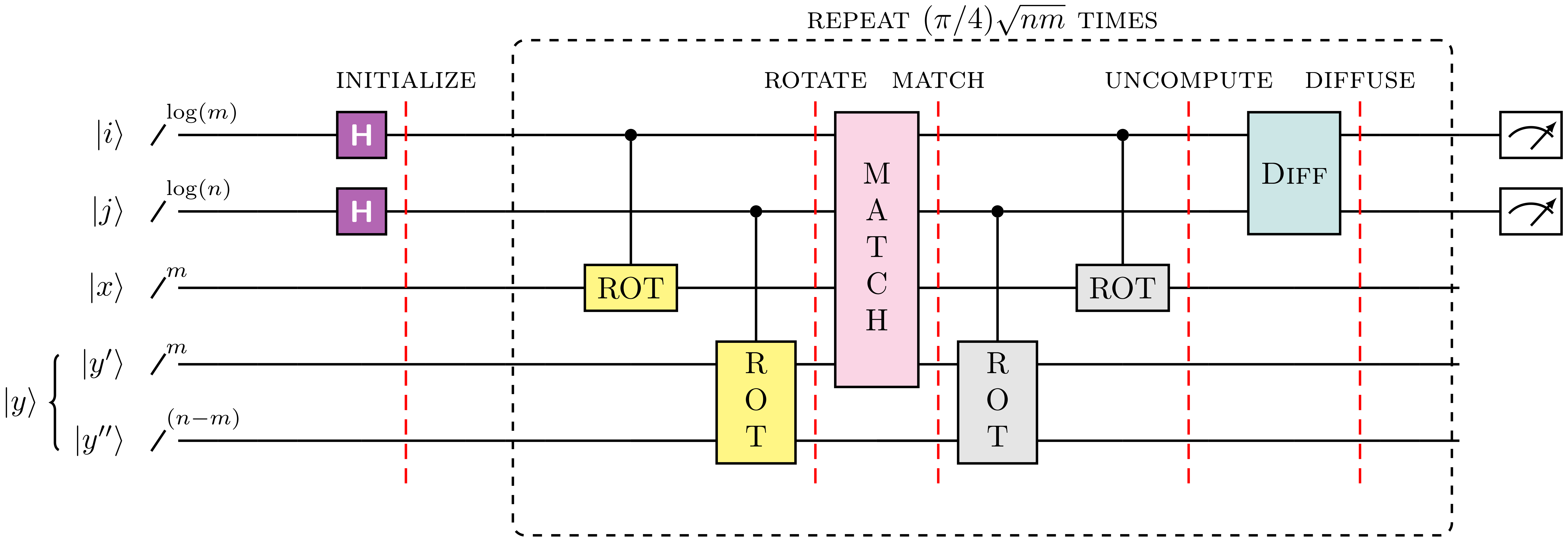
A phase oracle for exact string matching



- The n CNOTs can run in parallel as well as the n X gates.
- The multi-controlled flip-phase (Z) gate takes $\mathcal{O}(\log(m))$ time.

Depth: $\mathcal{O}\left(\log(m)\right)$.

A quantum circuit for cyclic string matching



ROT: the controlled cyclic shift operator. After its application, the registers $|x\rangle$ and $|y\rangle$ are in the state of superposition of all their respective cyclic shiftings.

Depth: $\mathcal{O}\left(\log^2(m)\right)$ and $\mathcal{O}\left(\log^2(n)\right)$, respectively.

MATCH: a phase oracle for exact string matching, i.e., for the function $f\colon \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ such that

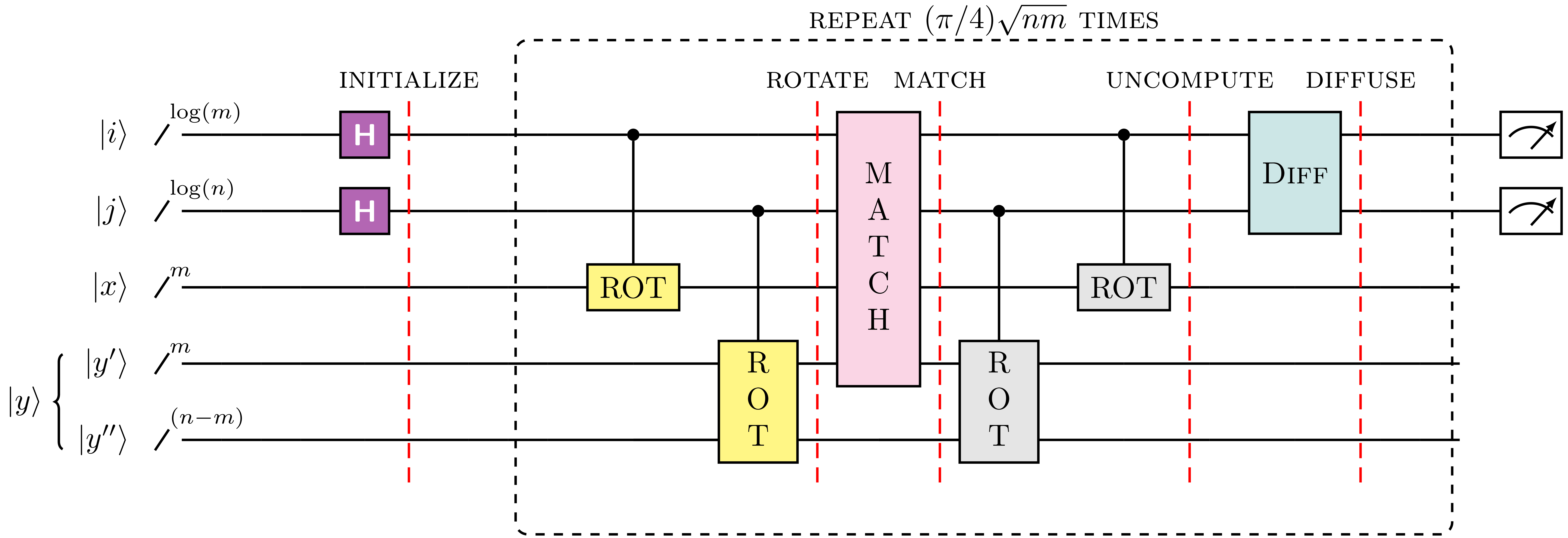
$$f(x, y[0..m-1]) = \begin{cases} 1 & \text{if } x_i = y_i \text{ for all } 0 \le i < m \\ 0 & \text{otherwise.} \end{cases}$$

Depth: $\mathcal{O}\left(\log(m)\right)$.

DIFF: Grover's diffuser.

Depth: $\mathcal{O}\left(\log(\log(n))\right)$.

A quantum circuit for cyclic string matching



<u>**Overall depth**</u>: $\mathcal{O}\left(\sqrt{n}\left(\log^2(m)+\log^2(n)+\log(m)+\log(\log(n))\right)\right)=\mathcal{O}\left(\sqrt{n}\left(\log^2(n)\right)\right)=\tilde{\mathcal{O}}\left(\sqrt{n}\right).$

<u>**Space**</u> : $\mathcal{O}\left(n+m\right)=\mathcal{O}\left(n\right)$. <u>**Size**</u>: $\mathcal{O}\left(n\log(n)\right)$.

Research perspective

- Can we improve this circuit to get a better performance (i.e., smaller depth or less qubits) so that it would be implementable on near-terms utility-scale quantum hardware?

- Quantum algorithms for approximate versions of cyclic shifting matching problem which find application in bioinformatics, etc.

- Quantum algorithms for Lyndon words and Lyndon factorisation.

Research perspective

- Can we improve this circuit to get a better performance (i.e., smaller depth or less qubits) so that it would be implementable on near-terms utility-scale quantum hardware?

- Quantum algorithms for approximate versions of cyclic shifting matching problem which find application in bioinformatics, etc.

- Quantum algorithms for Lyndon words and Lyndon factorisation.

Thanks

Ps. You can play with our codes written in qiskit at
https://colab.research.google.com/drive/1bbRjsYl7UCVT6P4gNwJulARfd64L1RCT?usp=sharing