

Quantum Uncomputation as Garbage Collection

SPLS 6th March 2024

Kengo Hirata (University of Edinburgh)

This talk

“Uncomputation” in quantum programming



Heavy Garbage Collection

Need verification

Type system,
Verification,
Compiler Optimization

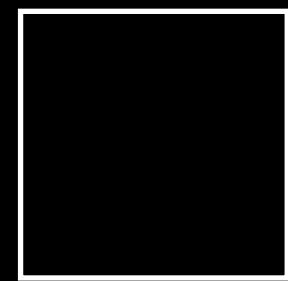
Table of contents

1. Quantum Uncomputation
2. Computer Science Aspects
3. Framework for Uncomputation

Table of contents

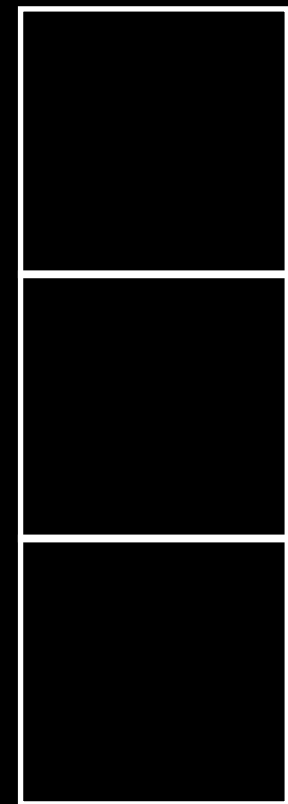
1. Quantum Uncomputation
2. Computer Science Aspects
3. Framework for Uncomputation

Classical Computer



State of 1 bit register

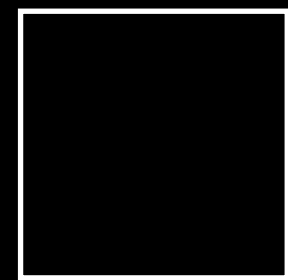
$\in \{0,1\}$



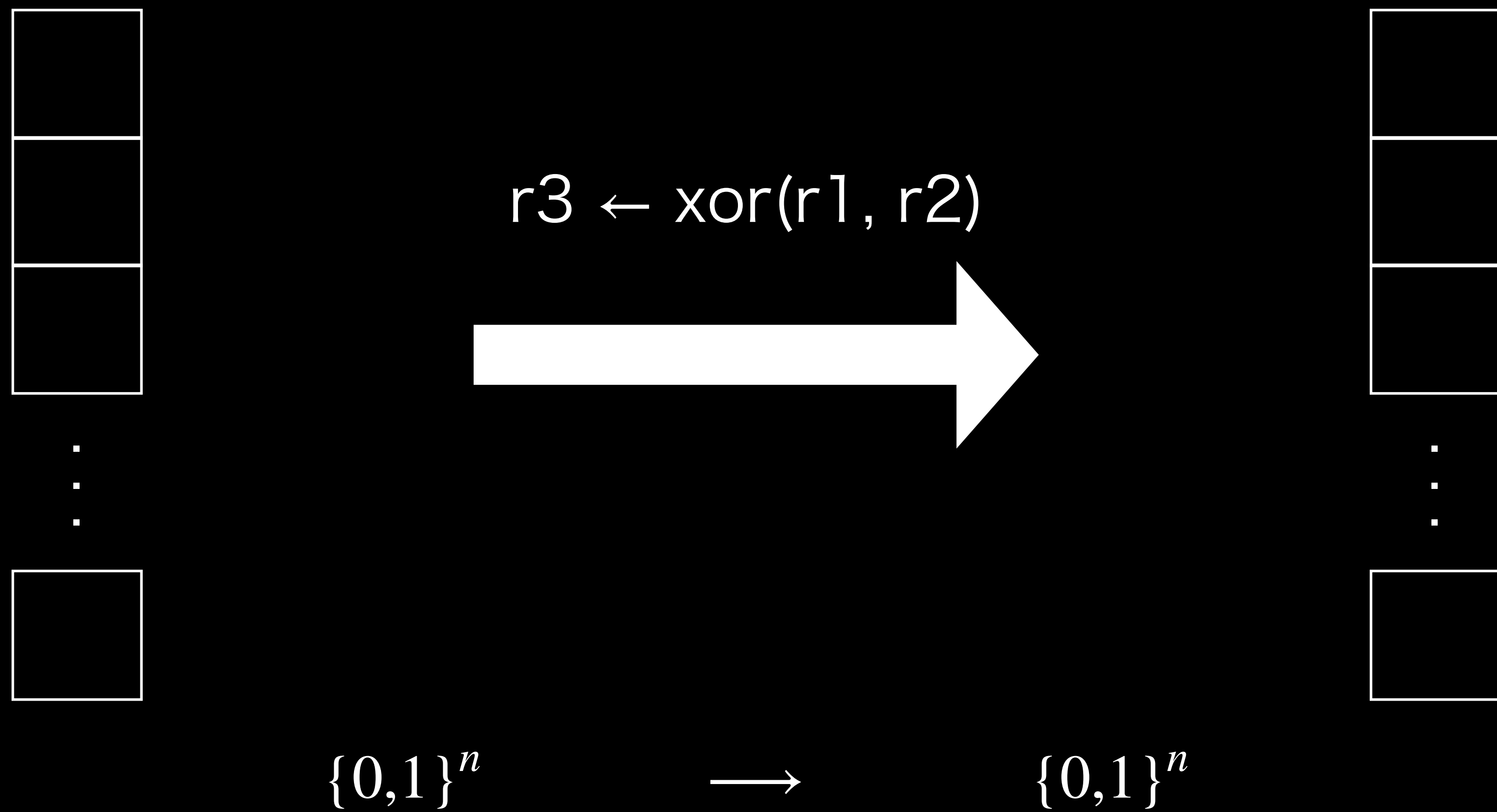
State of n bit register

$\in \{0,1\}^n$

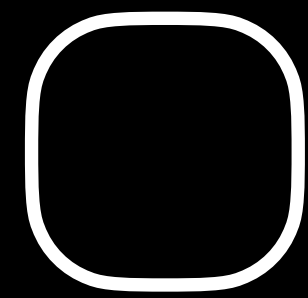
⋮



Classical Computer



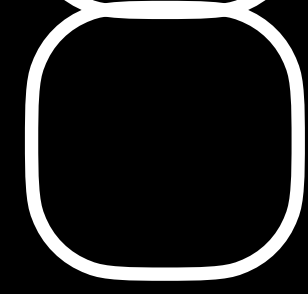
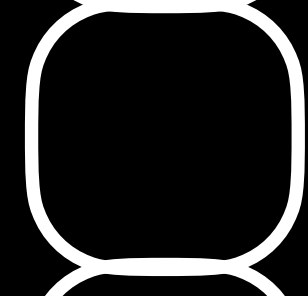
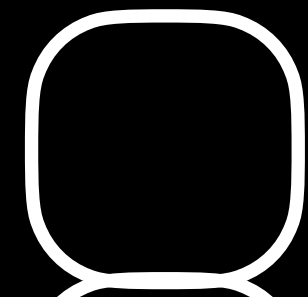
Quantum Computer



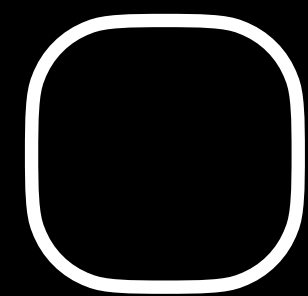
State of 1 qubit

$$\in \mathbb{C}^2$$

$$|0\rangle, |1\rangle, \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$



⋮



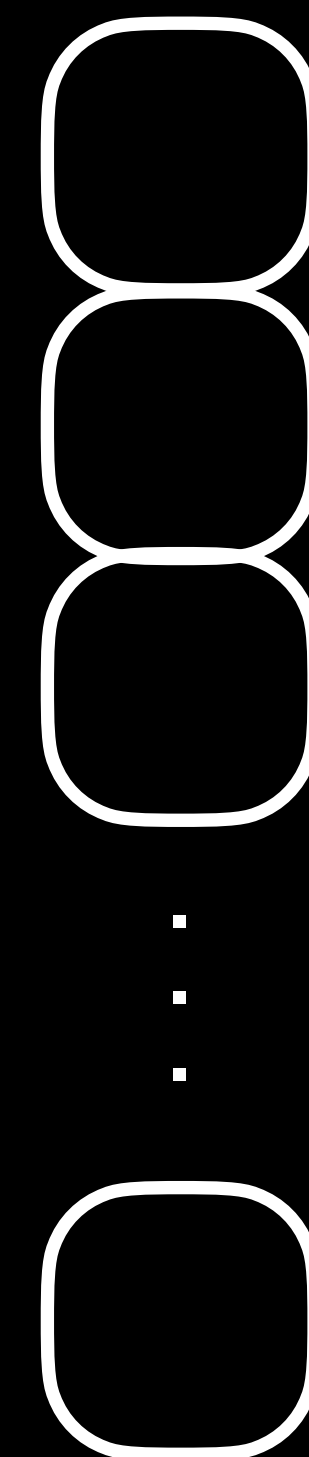
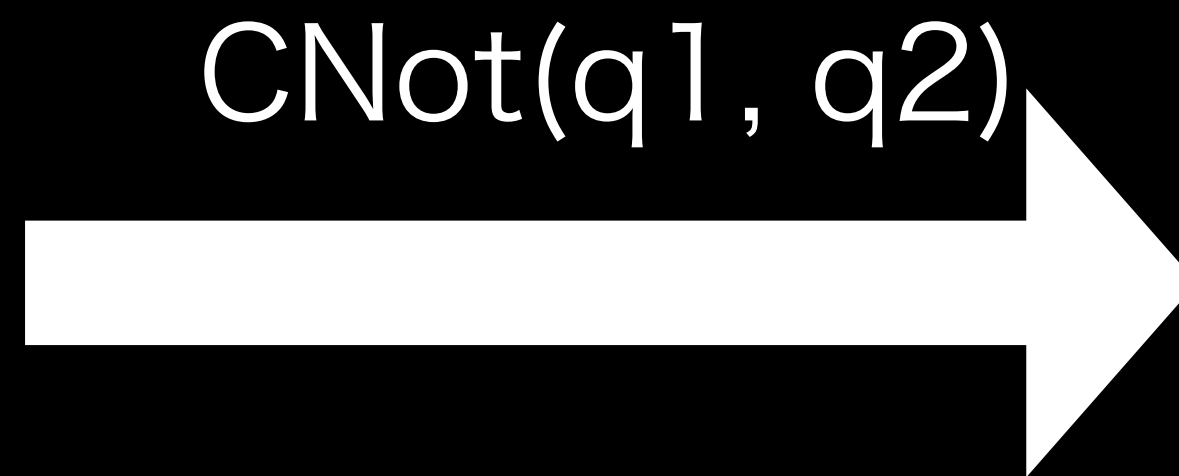
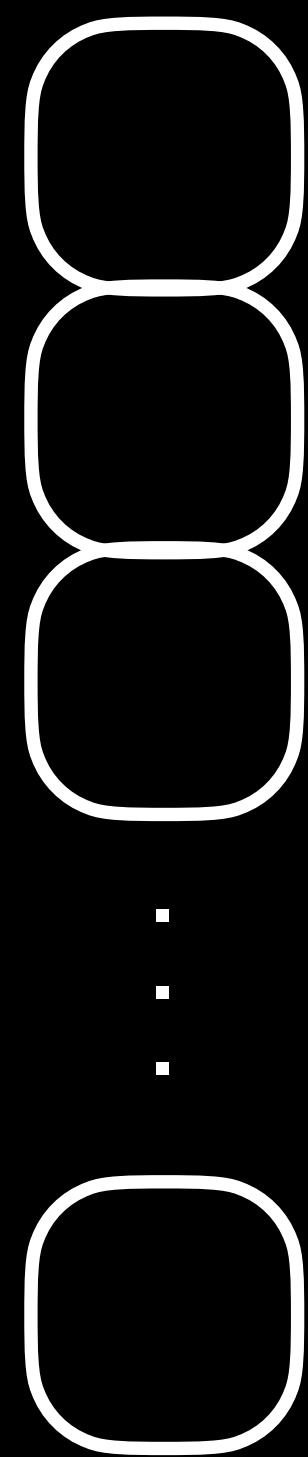
State of n qubit

$$\in (\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{(2^n)}$$

$$|00\rangle, \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Quantum Computer

Unitary Gate



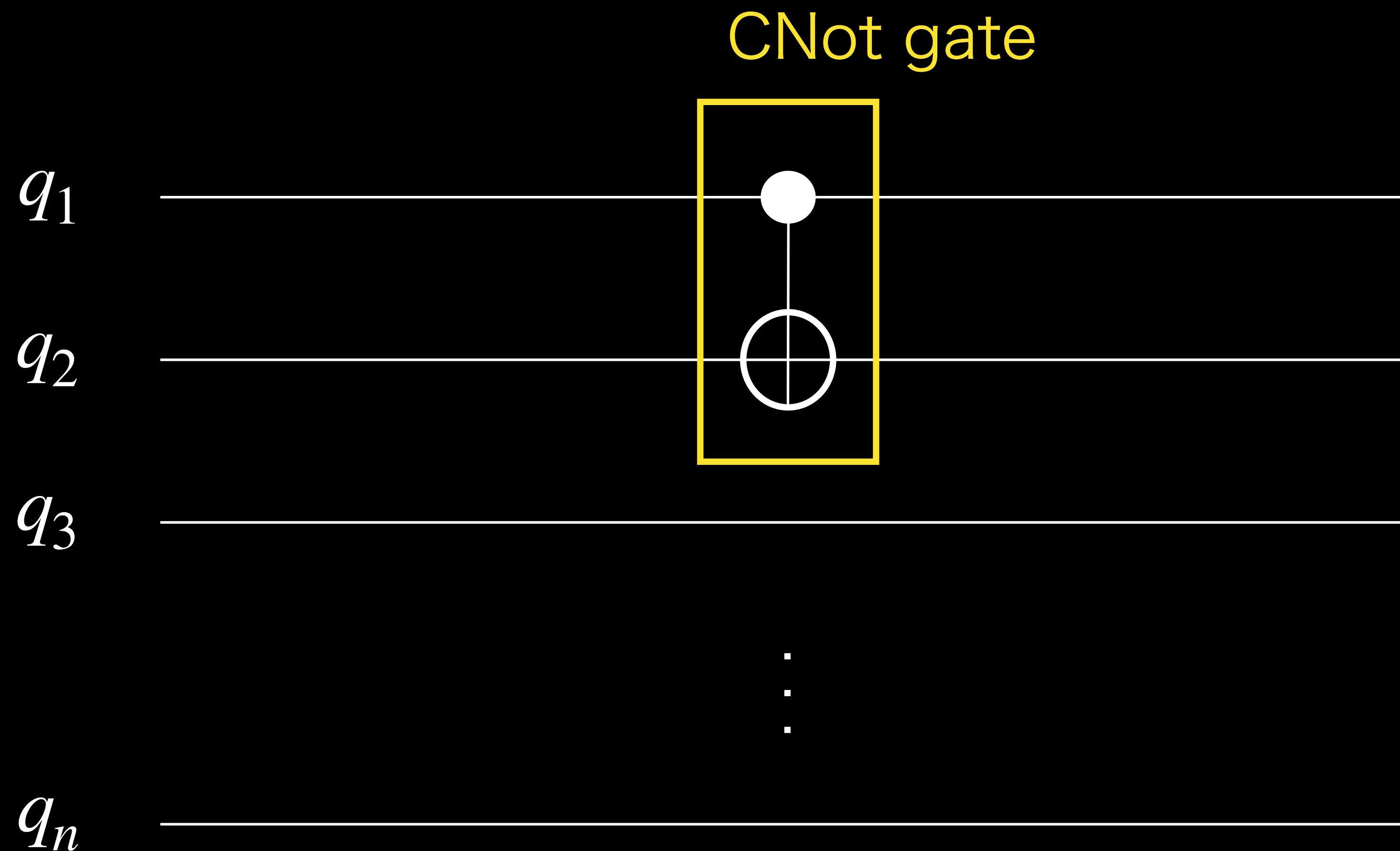
Unitary operation

$$(\mathbb{C}^2)^{\otimes n}$$



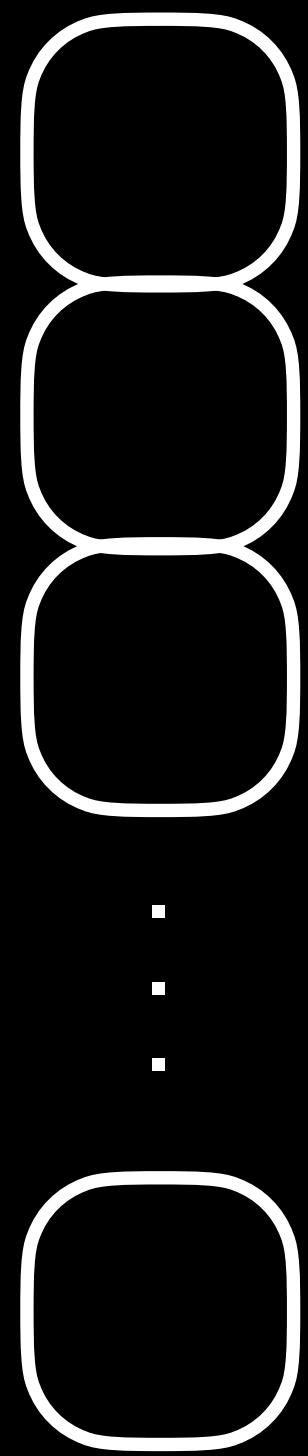
$$(\mathbb{C}^2)^{\otimes n}$$

Quantum Computer



Quantum Computer

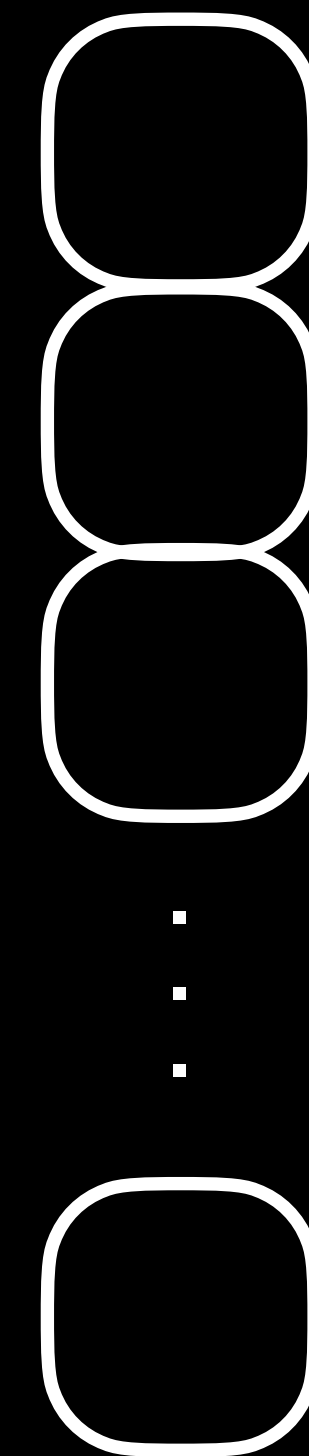
Measurement



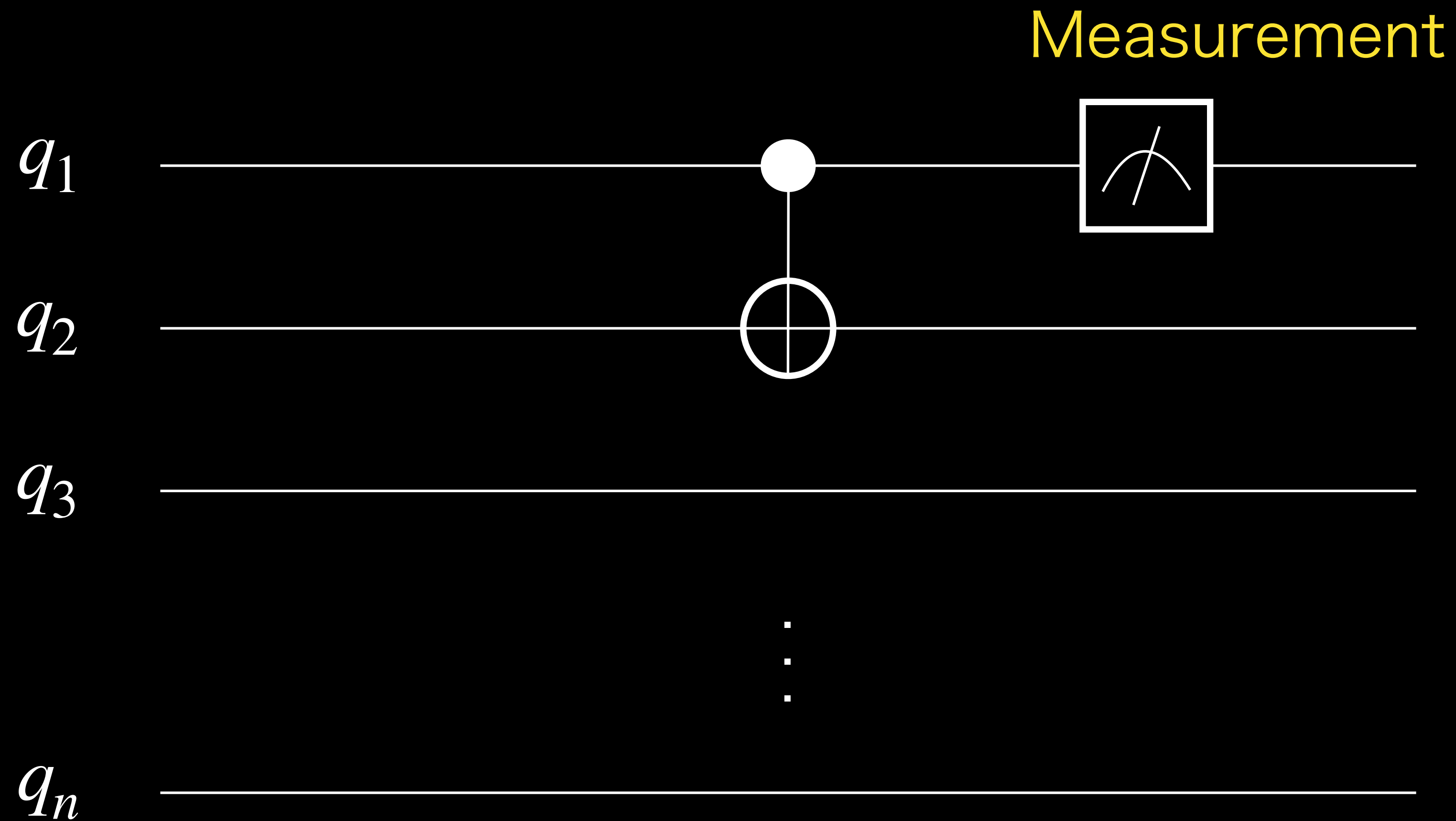
meas(q1)



0 or 1

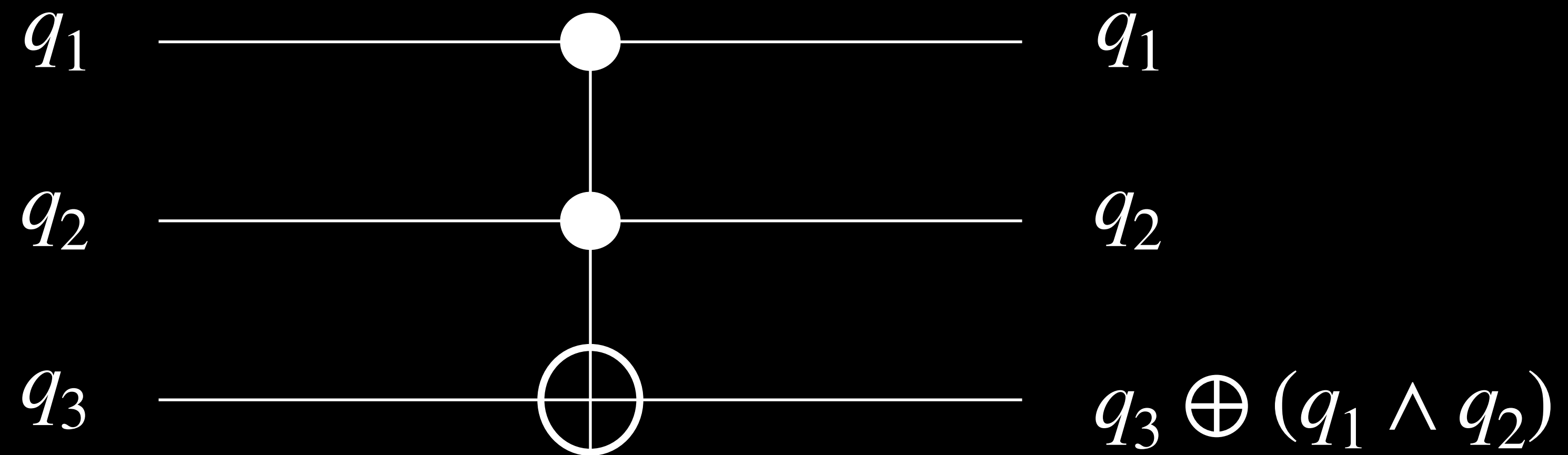


Quantum Computer



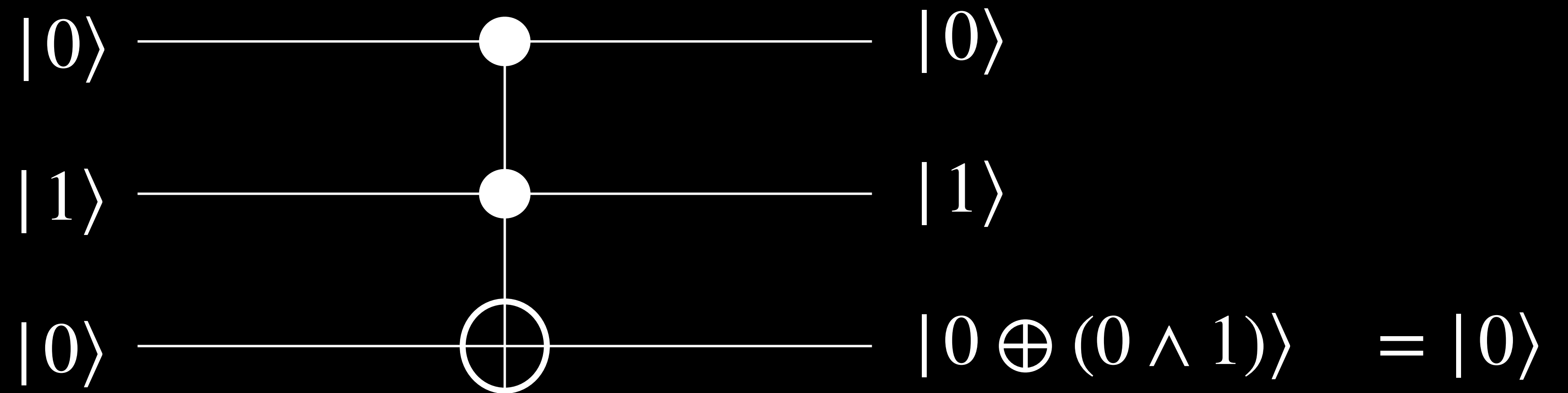
Example (Toffoli)

Toffoli Gate



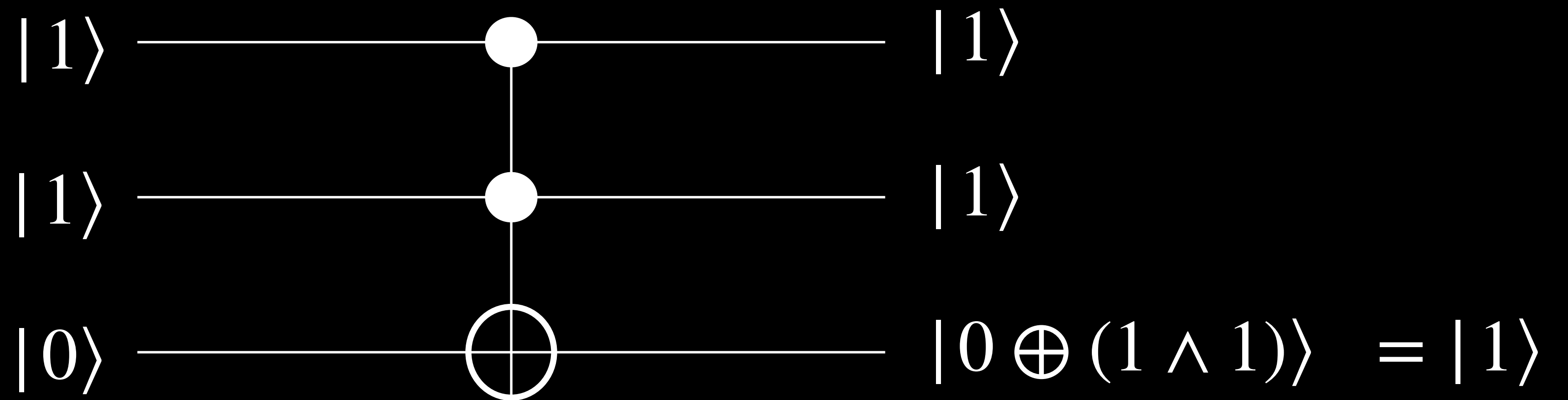
Example (Toffoli)

Toffoli Gate



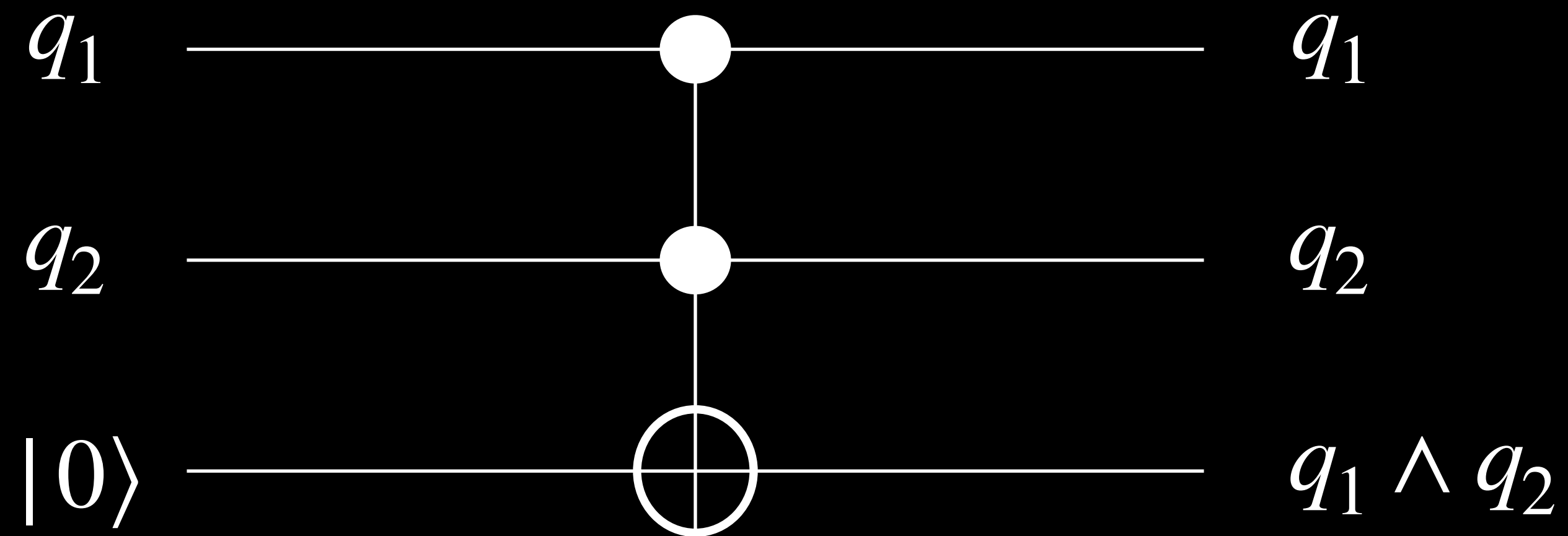
Example (Toffoli)

Toffoli Gate



Example (Toffoli)

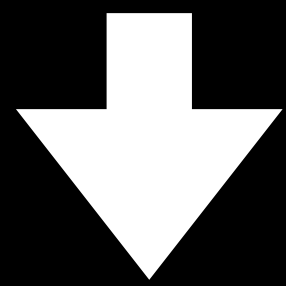
Toffoli Gate



$$\text{ret} \leftarrow q_1 \wedge q_2$$

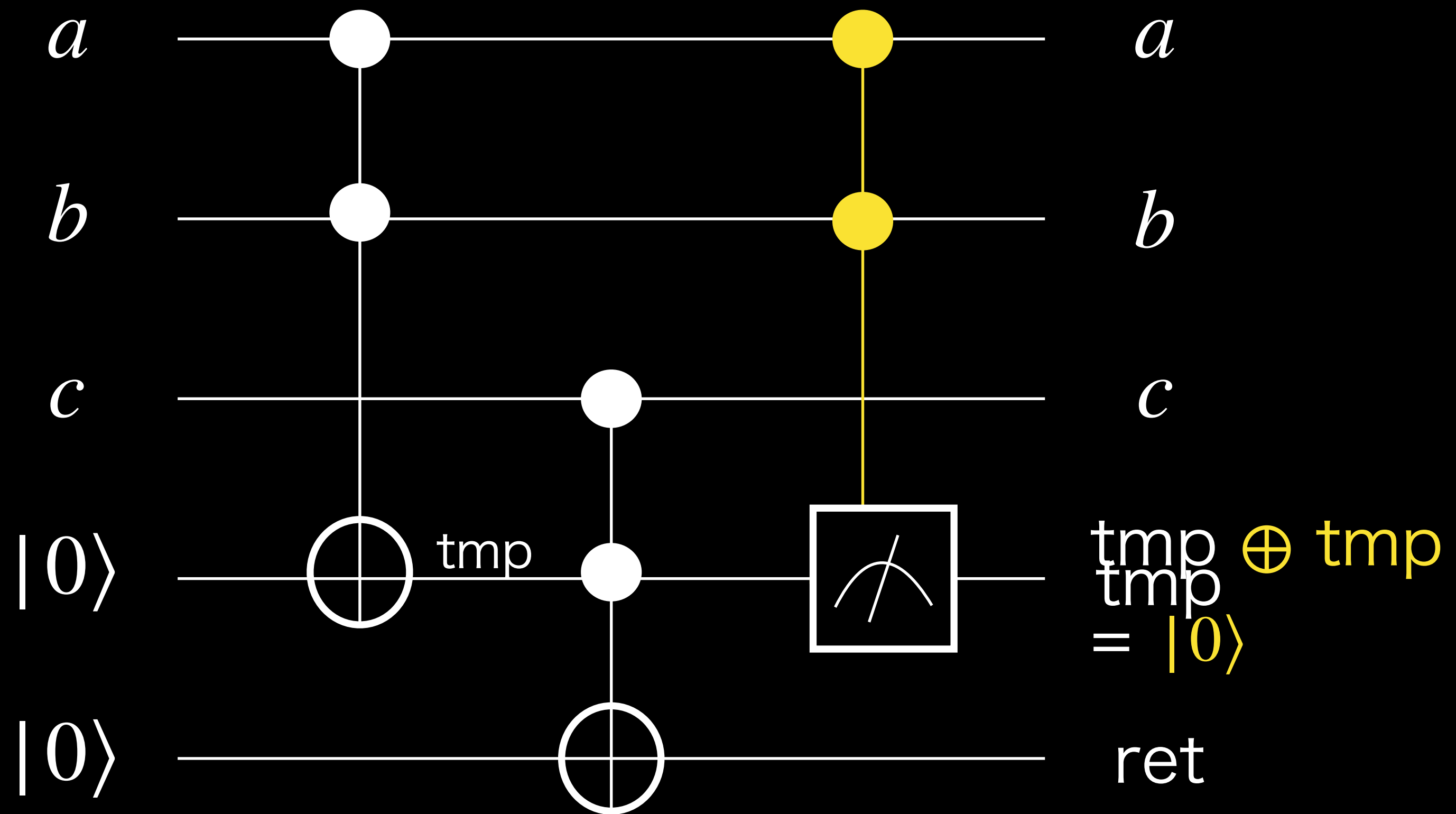
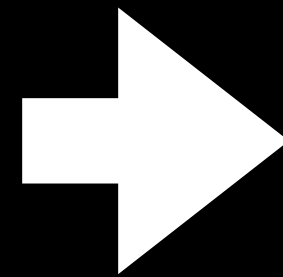
Example ($a \wedge b \wedge c$)

$$\text{ret} \leftarrow (a \wedge b) \wedge c$$



$$\text{tmp} \leftarrow a \wedge b$$

$$\text{ret} \leftarrow \text{tmp} \wedge c$$



Uncomputation

Given:

Two circuits

$$|x, 0\rangle \mapsto |x, f(x)\rangle$$

$$|x, 0\rangle \mapsto |x, g(x)\rangle$$

where

$$f, g: \{0,1\}^n \rightarrow \{0,1\}^n$$

Goal:

Create circuit

$$|x, 0\rangle \mapsto |x, g \circ f(x)\rangle$$

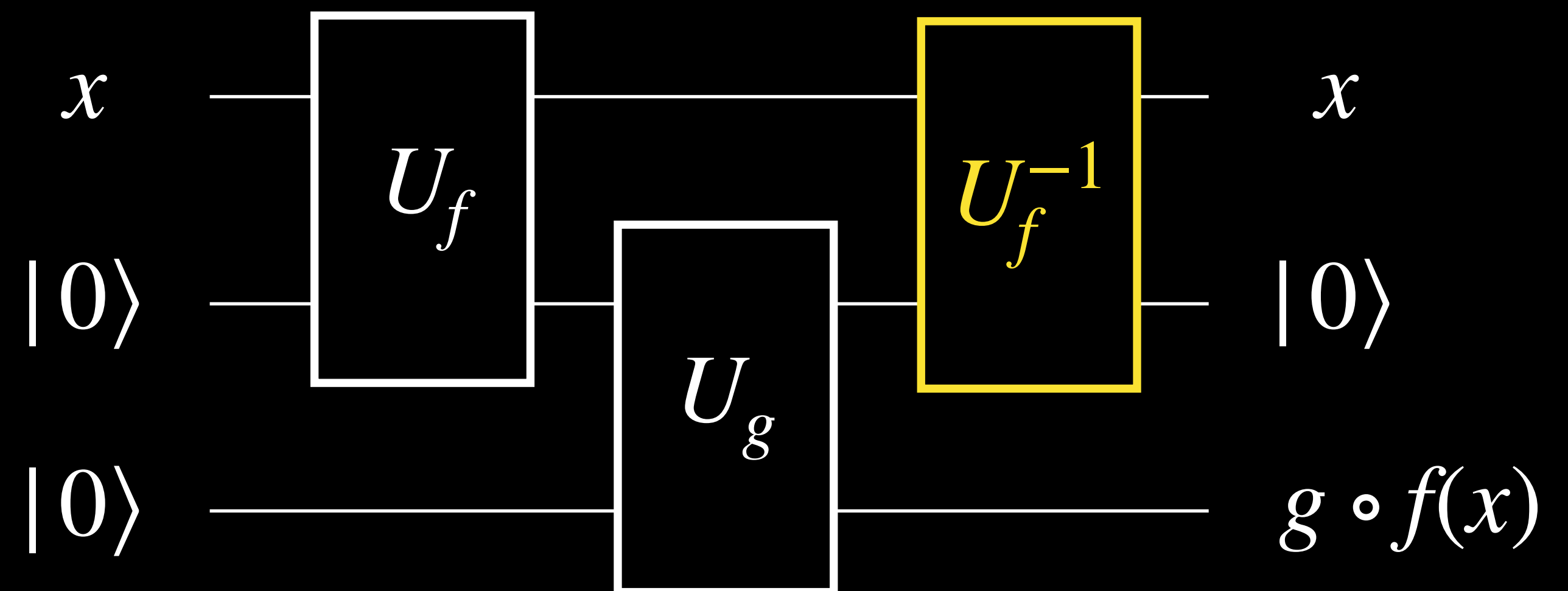


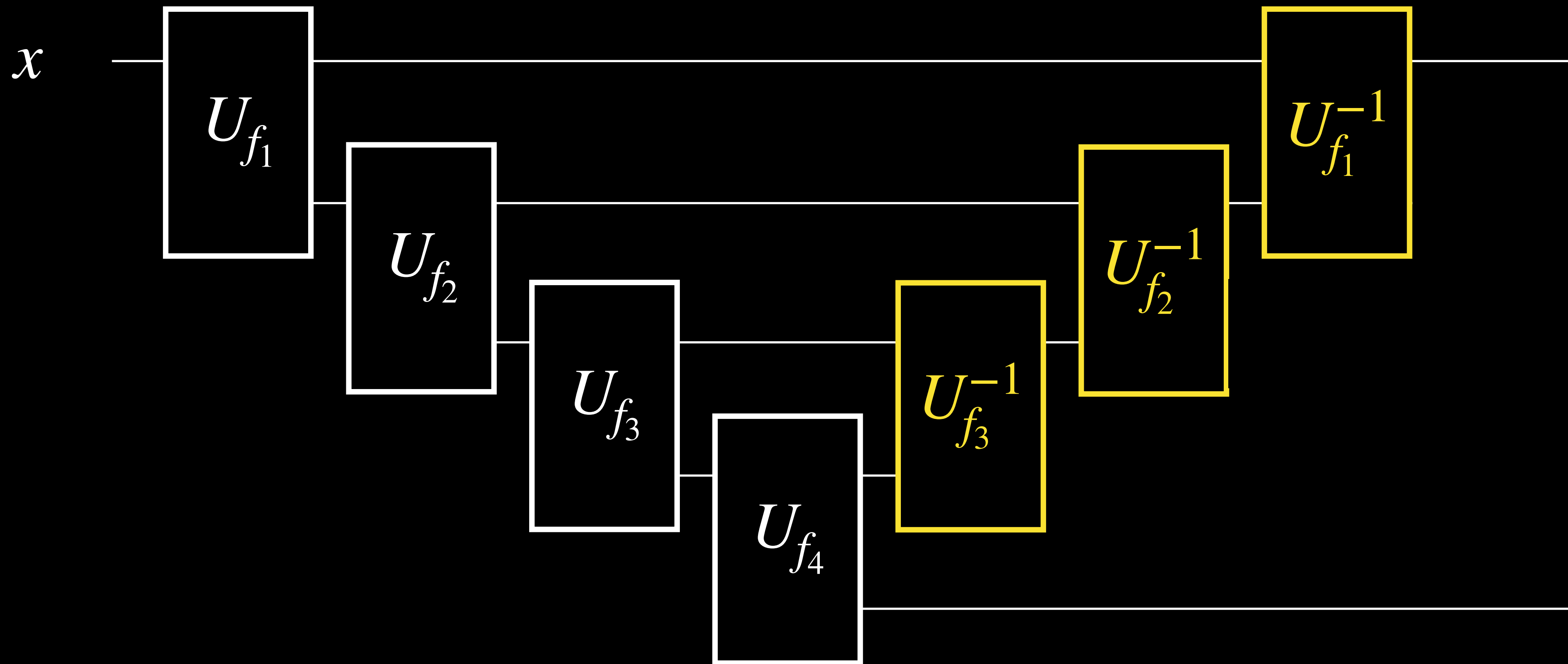
Table of contents

1. Quantum Uncomputation
2. Computer Science Aspects
3. Framework for Uncomputation

Computer Science Aspect

1. Space-Time Trade-offs

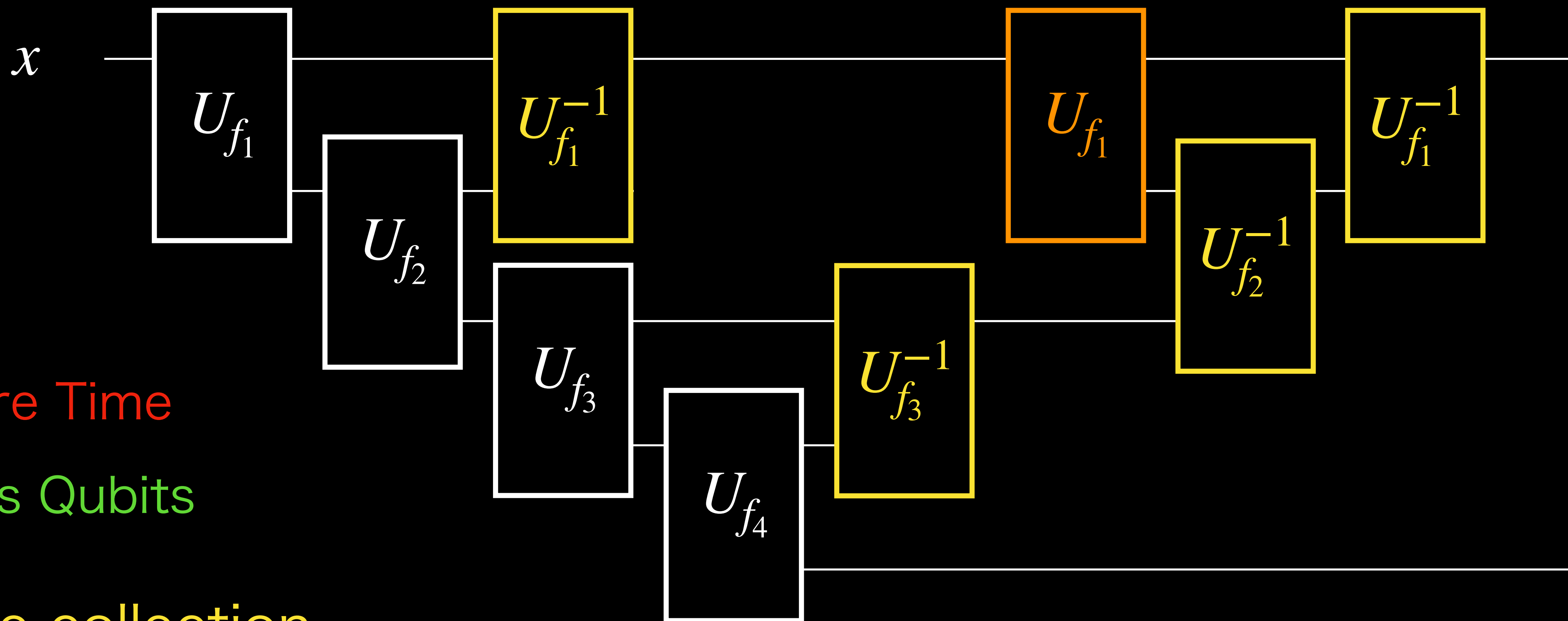
$$f_4 \circ f_3 \circ f_2 \circ f_1$$





Computer Science Aspect

1. Space-Time Trade-offs

$$f_4 \circ f_3 \circ f_2 \circ f_1$$



 More Time
 Less Qubits

Garbage collection

Computer Science Aspect

2. Verification

$$|00\rangle \mapsto |00\rangle - |01\rangle$$

Computer Science Aspect

2. Verification

$$|00\rangle \mapsto |0\rangle - |0\rangle$$

Computer Science Aspect

2. Verification

$$|00\rangle \mapsto |00\rangle - |00\rangle = 0$$

Abort

Cannot do uncomputation

Computer Science Aspect

2. Verification

Easy Uncomputation

Automatic uncomputation

Verified by type checker

Untrivial Uncomputation

Compile Error
or

Unsafe user-defined uncomputation

Cannot do Uncomputation

Compile Error

Table of contents

1. Quantum Uncomputation
2. Computer Science Aspects
3. Framework for Uncomputation

Language



Compile

Assembly

Type system

- Rust-like type system (lifetime, borrow)

Verification

- Type checking
- Library with unsafe uncomputation
- manual/automatic

Optimization

- Reversible pebble game
- Spooky pebble game

Silq [B.Bichsel et al. PLDI '20]



```
def and3(const a: B, const b: B, const c: B) qfree: B {  
  ret := (a && b) && c;  
  return ret  
}
```

`const` Related to immutable borrow in Rust

strong link with the garbage collection?

Rust-like language (rough idea)

lifetime α

variable definition

uncomputation



Immutable borrow $\&^\alpha$

Keep variables unchanged until the uncomputation occurs