Quantum Computing Seminar 7

YongHyun “Aeren” An

Samsung Research

January 6, 2025

Computational Cost

Computational Cost

- Q) Is runtime of a program a good way to define the computational cost?

Computational Cost

- Q) Is runtime of a program a good way to define the computational cost?
- Q) Is the number of gates a good way to define the computational cost?

Computational Cost

Elementary gates (quantum circuit)

Computational Cost

Elementary gates (quantum circuit)

- **Elementary gates** is a set of gates, each acting on some fixed number of qubits.

Computational Cost

Elementary gates (quantum circuit)

- **Elementary gates** is a set of gates, each acting on some fixed number of qubits.
- Our choice of gates consists of
  - Single qubit gates: X, Y, Z, H, S, S^H, T and T^H
  - Controlled-not gate: CX
  - Single qubit standard basis measurements

Computational Cost

Elementary gates (quantum circuit)

- **Elementary gates** is a set of gates, each acting on some fixed number of qubits.
- Our choice of gates consists of
  - Single qubit gates: X, Y, Z, H, S, S^H, T and T^H
  - Controlled-not gate: CX
  - Single qubit standard basis measurements
- This set of gates forms a **universal gate set**: we can approximate any unitary operation, on any number of qubits, and to any degree of accuracy we wish, with circuits composed of these gates alone.

Computational Cost

Elementary gates (quantum circuit)

- **Elementary gates** is a set of gates, each acting on some fixed number of qubits.
- Our choice of gates consists of
  - Single qubit gates: X, Y, Z, H, S, S^H, T and T^H
  - Controlled-not gate: CX
  - Single qubit standard basis measurements
- This set of gates forms a **universal gate set**: we can approximate any unitary operation, on any number of qubits, and to any degree of accuracy we wish, with circuits composed of these gates alone.

Elementary gates (boolean circuit)

We choose AND, OR, NOT, $FANOUT$ as elementary gates for boolean circuit, which forms a universal gate set for deterministic computation.

Computational Cost

Circuit size and depth

Circuit size is the number of gates in a circuit.
Circuit depth is the number of layers of gates in a circuit

Computational Cost

Circuit size and depth

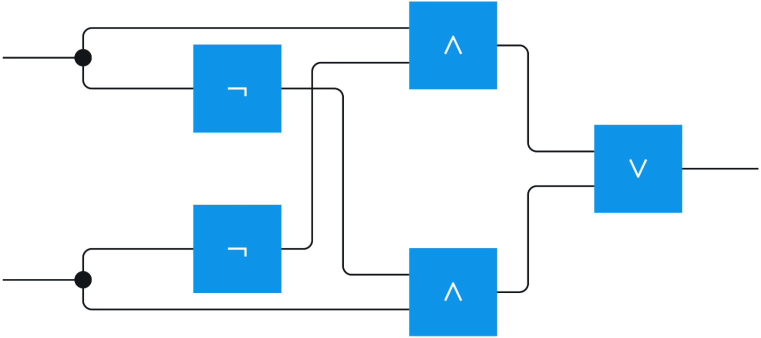
Circuit size is the number of gates in a circuit.
Circuit depth is the number of layers of gates in a circuit

- We'll mostly associate circuit size with computational cost, despite the fact that circuit depth is more realistic estimation of runtime, for simplicity.

Computational Cost

Exercise: XOR circuit

What is the size and depth of the following circuit which computes the XOR of two bits?



Computational Cost

Problem (Integer addition)

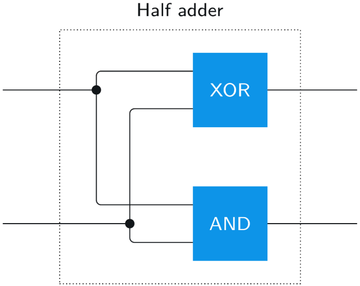
Input: N-bit integers A and B
Output: $(N+1)$-bit integer $A+B$

Computational Cost

Problem (Integer addition)

Input: N-bit integers A and B
Output: $(N+1)$-bit integer $A+B$

The LSBs can be added with the following half adder.

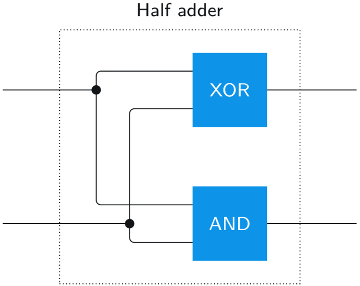


Computational Cost

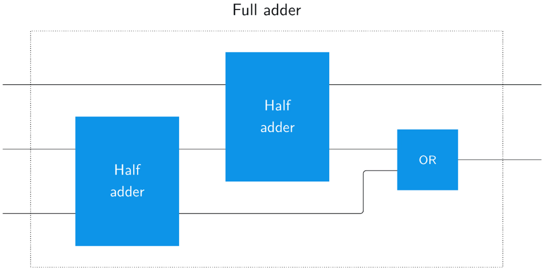
Problem (Integer addition)

Input: N-bit integers A and B
Output: $(N+1)$-bit integer $A+B$

The LSBs can be added with the following half adder.



Other bits can be added with the following full adder.

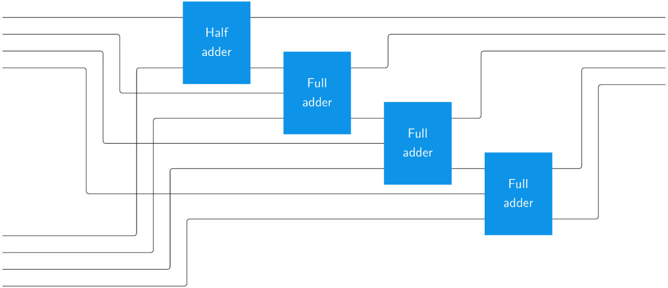


Computational Cost

Problem (Integer addition)

Input: N-bit integers A and B
Output: $(N+1)$-bit integer $A+B$

For example, following is the full circuit for $N = 4$.



Computational Cost

Problem (Integer multiplication)

Input: N-bit integers A and B
Output: $(2N-1)$-bit integer $A \cdot B$

Computational Cost

Problem (Integer multiplication)

Input: N-bit integers A and B
Output: $(2N-1)$-bit integer $A \cdot B$

Cost: $O(N^2)$ with naive multiplication, $O(N \cdot \log(N))$ with Van Der Hoven multiplication.

Computational Cost

Problem (Integer multiplication)

Input: N-bit integers A and B
Output: $(2N-1)$-bit integer $A \cdot B$

Cost: $O(N^2)$ with naive multiplication, $O(N \cdot \log(N))$ with Van Der Hoven multiplication.

Problem (Integer division)

Input: N-bit integers A and $B > 0$
Output: Integer q and r satisfying $0 \leq r < B$ and $A = q \cdot B + r$

Computational Cost

Problem (Integer multiplication)

Input: N-bit integers A and B
Output: $(2N-1)$-bit integer $A \cdot B$

Cost: $O(N^2)$ with naive multiplication, $O(N \cdot \log(N))$ with Van Der Hoven multiplication.

Problem (Integer division)

Input: N-bit integers A and $B > 0$
Output: Integer q and r satisfying $0 \leq r < B$ and $A = q \cdot B + r$

Cost: Same as integer multiplication

Computational Cost

Problem (Integer GCD)

Input: N-bit integers A and B
Output: GCD of A and B

Computational Cost

Problem (Integer GCD)

Input: N-bit integers A and B
Output: GCD of A and B

Cost: $O(N)$ application of integer divisions. (Note: there are faster algorithm such as half-gcd algorithm)

Computational Cost

Problem (Integer GCD)

Input: N-bit integers A and B
Output: GCD of A and B

Cost: $O(N)$ application of integer divisions. (Note: there are faster algorithm such as half-gcd algorithm)

Problem (Integer modular exponentiation)

Input: N-bit integers B, $E \geq 0$, and $M > 0$
Output: $B^E \mod M$

Computational Cost

Problem (Integer GCD)

Input: N-bit integers A and B
Output: GCD of A and B

Cost: $O(N)$ application of integer divisions. (Note: there are faster algorithm such as half-gcd algorithm)

Problem (Integer modular exponentiation)

Input: N-bit integers B, $E \geq 0$, and $M > 0$
Output: $B^E \mod M$

Cost: $O(N)$ application of integer multiplications and divisions.

Computational Cost

Problem (Integer factorization)

Input: N-bit integer A
Output: Factorization of A

Computational Cost

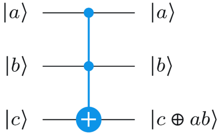
Problem (Integer factorization)

Input: N-bit integer A
Output: Factorization of A

Cost: The state-of-art factorization algorithm (general number field sieve) is conjectured to take $O(e^{1.9 \cdot N^{1/3} \log(N)^{2/3}})$

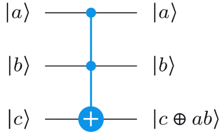
Simulation of Boolean Circuit

Boolean circuit can be simulated using the Toffoli (CCX) gate

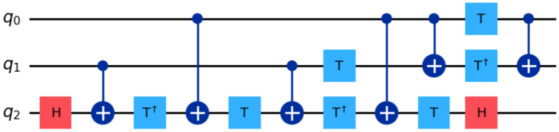


Simulation of Boolean Circuit

Boolean circuit can be simulated using the Toffoli (CCX) gate



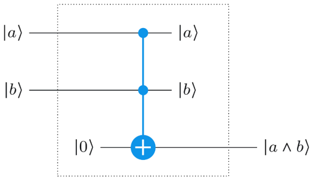
which can be implemented with the following circuit.



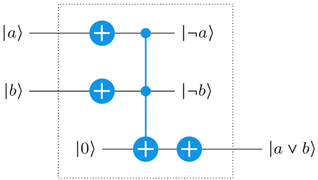
Simulation of Boolean Circuit

Given a boolean circuit C describing a function $f : \{0, 1\}^n \to \{0, 1\}^m$, we replace every NOT gate with an X gate, and the other elementary gates with the following quantum circuits.

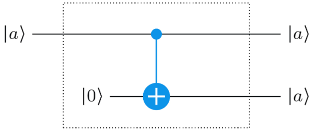
AND gate



OR gate

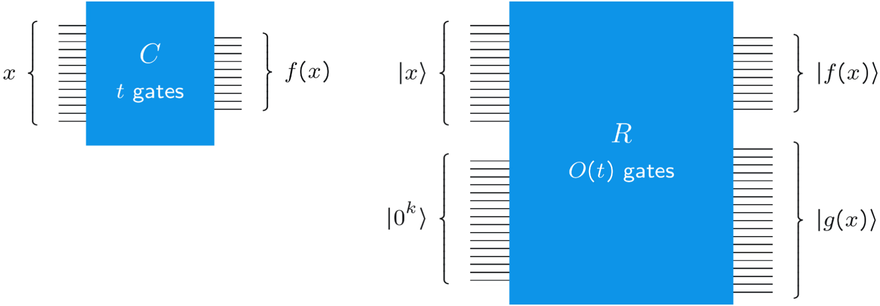


FANOUT gate



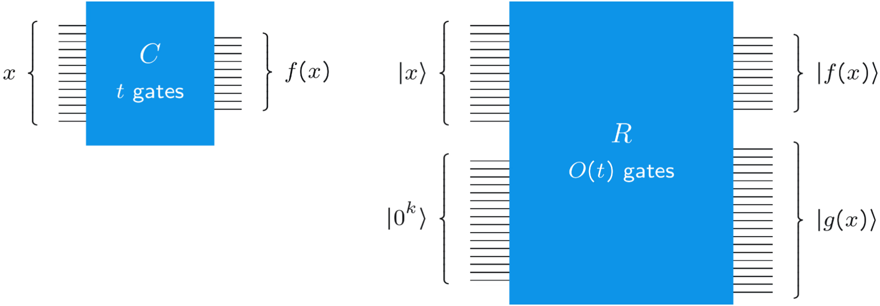
Simulation of Boolean Circuit

We obtain the quantum circuit R.



Simulation of Boolean Circuit

We obtain the quantum circuit R.



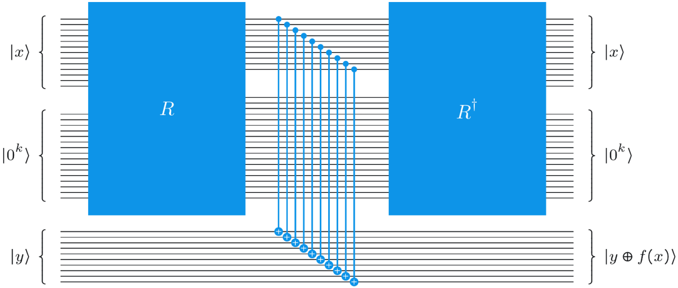
Here, k is the number of additional workspace qubits used, and $g : \{0, 1\}^n \to \{0, 1\}^{n+k-m}$ describes the garbage qubits at the end.

Simulation of Boolean Circuit

However, the garbage bits may be entangled with the output qubits, which is problematic if we want to do additional computation on output qubits.

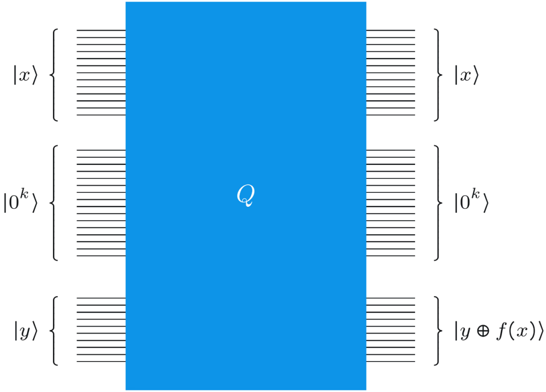
Simulation of Boolean Circuit

However, the garbage bits may be entangled with the output qubits, which is problematic if we want to do additional computation on output qubits.
To ensure that the garbage qubits are independent from the rest, we construct the circuit as the following.



Simulation of Boolean Circuit

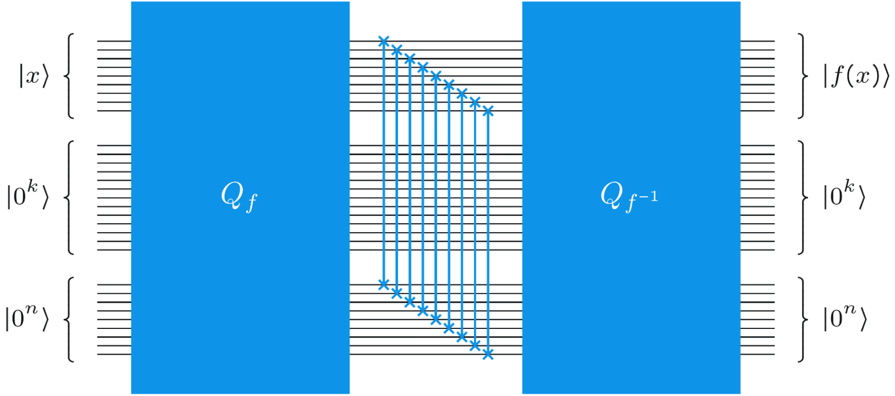
We now box everything and call it the query gate Q.



Note that if C has t gates, Q has $O(t)$ gates.

Simulation of Boolean Circuit

Extra If f is bijective, we can implement a circuit representing the gate U with $U|x\rangle = |f(x)\rangle$ as follows.



The End