COE 530
Quantum Computers and Architecture

Lecture 7

Classical Simulation for Quantum Computing

Reference:
Chapter 9, Quantum Computing Systems

Quantum Computing Simulation Methods

- Quantum computing simulations can be carried out through several classical simulation techniques and specialized quantum-inspired approaches:
  - State Vector Simulation
  - Unitary Simulation
  - Density Matrix
  - Stabilizer Simulations
  - Matrix Product States

- Each simulation method determines the internal representation of the quantum circuit and quantum operations

State Vector Simulation

- In quantum computing, the state of an n −qubit system is represented by a **state vector** in a 2^n-dimensional complex vector space (Hilbert space)

- For example, the state $|\psi\rangle$ of an n-qubit system is represented by

$$|\psi\rangle = \alpha_0|00\ldots0\rangle + \alpha_1|00\ldots1\rangle + \cdots + \alpha_{2^n-1}|11\ldots1\rangle$$

Where the amplitudes α_i are complex numbers

- State vector simulation tracks the quantum state $|\psi\rangle \in \mathbb{C}^n$

Mechanics of State Vector Simulation

- State vector simulation works as the following

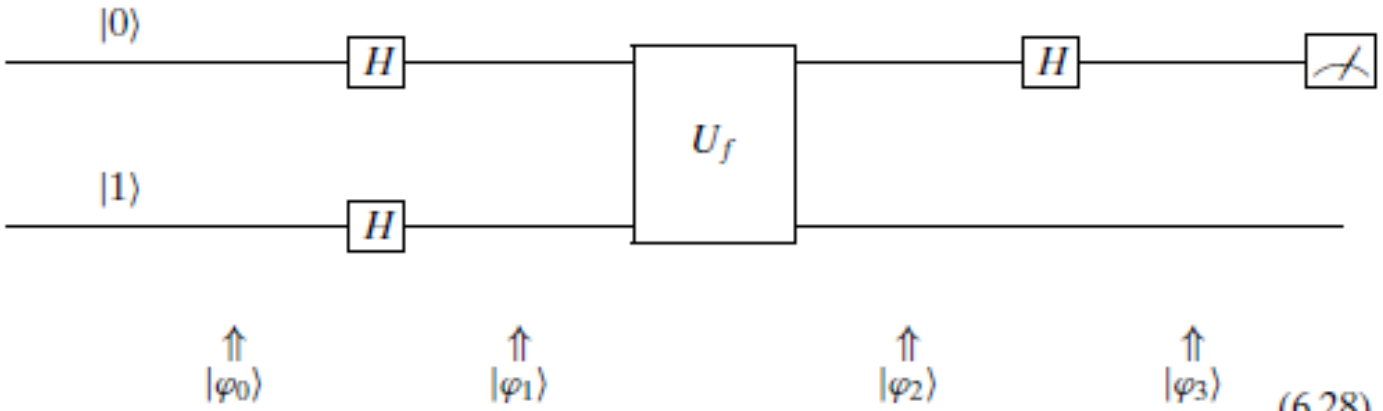
1. Initialization
   - Start from an initial state, usually $|0\rangle^{\otimes n}$, represented as a vector of size 2^n
2. Quantum Gates as Matrix Multiplication:
   - Quantum gates are represented by $2^n \times 2^n$ complex matrices.
   - Applying a gate means performing matrix-vector multiplication:

$$|\psi_{new}\rangle = U|\psi_{old}\rangle$$

   where U is the gate's unitary matrix.
3. Measurement:
   - Probabilities for measurement outcomes are computed by squaring amplitudes:

$$P(i) = |\ \alpha_i\ |^2$$

Example



(6.28)

Page 176 in Yanofski ”Quantum Computing for Computer Scientist”

Advantages vs. Disadvantages of State Vector Simulation

- Advantages:
  - Exact Representation
    - Provides precise, deterministic results with exact amplitudes and probabilities
  - Simple and Intuitive
    - Direct mapping of quantum mechanics mathematics to simulation code
  - Easily interpretable
    - Allows straightforward extraction of amplitudes, probabilities, and expectation values
- Limitations and Challenges:
  - Exponential Resource Requirements
    - Rapidly becomes infeasible beyond 40-50 qubits, even on supercomputers
    - Memory cost is $O(2^n)$ and time cost per gate is $O(2^n)$
  - Limited Scalability
    - Not suitable for simulating large-scale quantum systems or highly entangled states, due to exponential growth

Unitary Simulation

- A special case of state vector simulation that simulates the quantum state as the following

$$|\psi_{final}\rangle = U_{circuit}|\psi_{initial}\rangle$$

Where $U_{circuit} \in \mathbb{C}^{2^n \times 2^n}$ is a unitary matrix that represents entire unitary matrix of the circuit

- Simulating a quantum circuit entails performing the following sequence of unitary operators

$$|\psi_{final}\rangle = U_N U_{N-1} \dots U_1 |\psi_{initial}\rangle$$

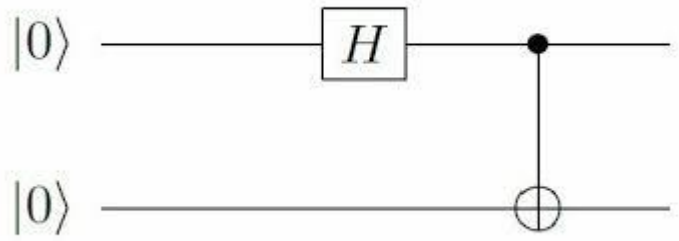
- Unitary simulation is usually used to simulate all possible input states of the circuit
- Statevector simulation asks:

  "What happens to this specific input?"

- Unitary simulation asks

  "What transformation does this circuit perform on any input?"

Example



- State vector

$$|\psi_0\rangle = |00\rangle$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

- Unitary

$$U_{circuit} = CNOT(H \otimes I)$$

$$= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}$$

Compute output of state inputs $|00\rangle, |01\rangle, |10\rangle, |11\rangle$

Advantages vs. Disadvantages of Unitary Simulation

- Advantages:
  - Simulate the output state of multiple input states
  - Efficient for simulation small circuit analysis and verification and studying operator properties
- Limitations and Challenges:
  - The limitation of state vector simulation
  - Memory cost is $O(4^n)$

Density Matrix Simulation

- Density matrix simulation is a method for simulating quantum systems that tracks the system using a density operator instead of a statevector

- Recall that a mixed state is a probabilistic mixture of states $|\psi_i\rangle$ with probabilities p_i

$$\rho = \sum p_i |\psi_i\rangle\langle\psi_i|$$

Where

- $p_i \geq 0$
- $\sum p_i = 1$
- $|\psi_i\rangle$ is a quantum state
- ρ is a positive semi-definite, Hermitian matrix with $Tr(\rho) = 1$

- Density matrix is used to describe most general quantum states including
  - Mixed states
  - Noise
  - Decoherence
  - Partial systems
  - Open quantum dynamics

Density Matrix Simulation

- Recall that state vector evolution is given by

$$|\psi_{new}\rangle = U|\psi_{old}\rangle$$

- In density matrix simulation,

$$\rho_{new} = U\rho_{old}U^{\dagger}$$

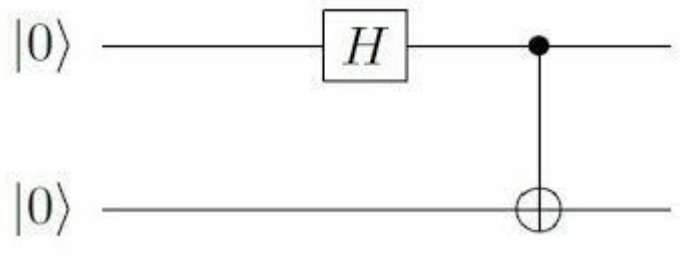
Or generally (with errors)

$$\rho_{new} = \sum_{k} E_k \, \rho_{old} E_k^{\dagger}$$

Example

- The density matrix for $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is

$$= \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$



- This represents pure state of EPR
- But if we add phase damping factor $\lambda = 0$

$$\frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

This is a mixed state of 50%$|00\rangle$ and 50% $|11\rangle$

Advantages vs. Disadvantages of Density Matrix Simulation

- Advantage
  - Modeling decoherence and noise
    - Mixed states can simulate quantum decoherence due to the ineraction with the envirnoment
- Disadvanteage
  - Computationally intensive
    - For n-qubit system, simulation needs 4^n parameters

Stabilizer Simulation

- Stabilizer simulations are classical methods for efficiently simulating quantum circuits composed only of **Clifford gates**

- Clifford gates are quantum gates U that transform any Pauli operator into another Pauli operator under conjugation, i.e.,

$$UPU^{\dagger} \rightarrow P'$$

Where P and P' are Pauli operators (i.e., $I, X, Y,$ and Z)

- Examples of Clifford gates
  - Hadamard
  - CNOT
  - Phase gate

What are Stabilizers?

- A Stabilizer is a Pauli operator that leaves a given quantum state unchanged (except for a possible phase factor of +1)

- Any Pauli operator can be written as a product of X and Z operators with some phase $(\pm 1, \pm i)$

- Let $|\psi\rangle$ be a state of n-qubit system and S be an operator, we say that S stabilizes $|\psi\rangle$ if

$$S|\psi\rangle = |\psi\rangle$$

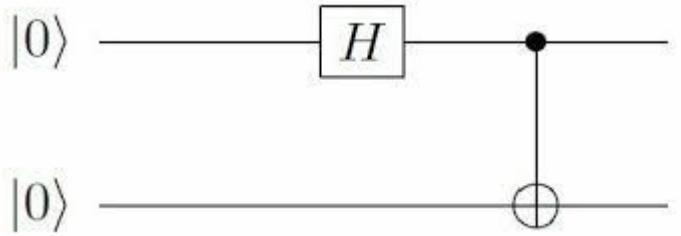
- In other words, we say that an operator S (from the set of Pauli operators) is a stabilizer operator if for all kets $|\psi\rangle$,

$$S|\psi\rangle = |\psi\rangle$$

$\forall S$, $i.e.$, $|\psi\rangle$ is an eiganket for all the stabilizers

Example

- Consider $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- Its Stabilizers are
  - $Z \otimes Z$
  - X$\otimes X$



- Instead of storing the complexity amplitudes, we store these operators
- Simulation steps
  - Initial state $|00\rangle$ -> $Z_1 Z_2$
  - Applying H on first qubits -> $X_1 Z_2$
  - Applying $CNOT$ -> $Z_1 Z_2, X_1 X_2$

To know how stabilizer simulation works, you need to understand the conjugation rule
If a state is stabilized by S, and we apply gate U, then $S \rightarrow USU^{\dagger}$
The updated operators define the new state

H rules
$HXH = Z$
$HZH = X$
$HYH = -Y$

CONT rules
$X_1 \rightarrow X_1 X_2$
$Z_1 \rightarrow Z_1$
$X_2 \rightarrow X_2$
$Z_2 \rightarrow Z_1 Z_2$

Stabilizer Simulation

- Gottesman-Knill Theorem:

  Any quantum circuit built solely from:
  - Stabilizer states as inputs (e.g., $|0 \dots 0\rangle$ or other states produced by Clifford gates),
  - Clifford gates (Hadamard, Phase S, and CNOT), and
  - Measurements in the computational (Z) basis

  can be simulated efficiently by a classical computer

- What does that mean?*
  - It shows that not all quantum circuits are intractable to simulate classically.
  - It also implies that circuits restricted to Clifford gates cannot give you the full "quantum speedup" that more general quantum circuits promise. You need at least one non-Clifford gate (commonly the T gate) to move beyond the stabilizer regime to universal quantum computation.
  - It underpins the idea of **quantum error correction**, where many of the basic operations are indeed Clifford and thus more straightforward to handle. – We will study this in detail in COE531

Advantages and Disadvantages of Stabilizer Simulation

- Advantages
  - Efficient Simulation of Certain Quantum Circuits
    - Stabilizer states can be simulated efficiently in polynomial time using classical computers for quantum circuits composed of Clifford gates and stabilizer states.
  - Deterministic and Efficient Simulation of Specific Algorithms
    - Certain quantum algorithms, like those based on the Quantum Fourier Transform (QFT) or Shor's algorithm when using Clifford operations, can be efficiently simulated using stabilizer circuits

- Limitations and Challenges:
  - Limited to Specific Quantum Circuits
    - The main disadvantage of stabilizer simulation is that it is only applicable to quantum circuits that can be described using stabilizer operations.
    - Circuits that involve non-stabilizer gates will require **exponential time** to simulate on a classical computer.
  - Approximation Rather than Exact Simulation
    - For quantum circuits with non-stabilizer gates, stabilizer simulation may still provide an approximate or probabilistic estimate of the quantum state, rather than an exact simulation.
  - Dependency on Circuit Structure
    - To use stabilizer simulation, the quantum circuit often needs to be decomposed into components that consist only of stabilizer operations.

Matrix Products States

- Matrix Product States are a way of expressing quantum states especially when dealing with large number of qubits

- Instead of writing down the entire wavefunction, MPS represent the quantum state in a compact form using matrices

- Let the state $|\psi\rangle = \sum_{\sigma_1,\sigma_2,\ldots,\sigma_N} c_{\sigma_1,\sigma_2,\ldots,\sigma_N} |\sigma_1, \sigma_2, \ldots, \sigma_N\rangle$ where $\sigma_n \in \{0,1\}$ denotes the n-th qubit and $c_{\sigma_1,\sigma_2,\ldots,\sigma_N}$ is the coefficient of the corresponding basis state

- It's clear that we need 2^N parameters to write the wave function of N qubits

https://pennylane.ai/qml/demos/tutorial_mps

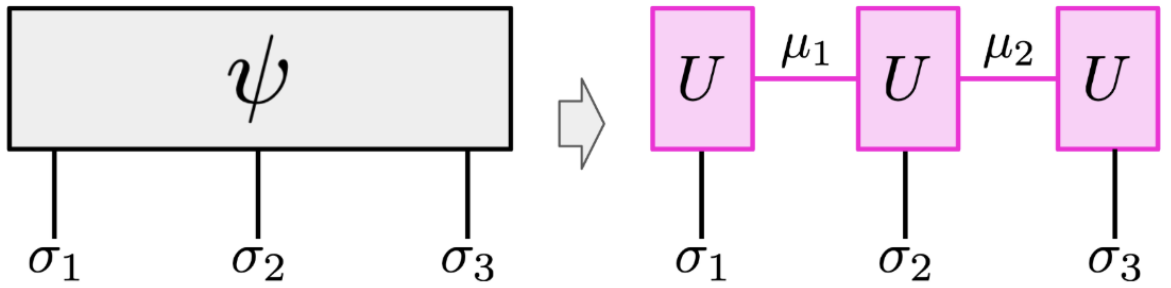
Matrix Products States

- Alternatively, we can write $|\psi\rangle$ by breaking it down into a series of smaller **matrices**, one for each particle, instead of storing the entire state vector.

- using less number of parameters

$$|\psi\rangle = \sum_{\sigma_1,\sigma_2,\ldots,\sigma_N} Tr\left(U_{\sigma_1}^{[1]} U_{\sigma_2}^{[2]} \ldots U_{\sigma_n}^{[N]}\right) |\sigma_1, \sigma_2, \ldots, \sigma_N\rangle$$

Where $U_{\sigma_n}^{[N]}$ is the matrix associated with the n-th qubit



Example of MPS

- Let $|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$
- Using MPS, it can be also written as

$$|\psi\rangle = \sum_{\sigma_1,\sigma_2,\sigma_3} U_{\sigma_1}^{[1]} U_{\sigma_2}^{[2]} U_{\sigma_3}^{[3]} |\sigma_1, \sigma_2, \sigma_3\rangle$$

Where

$$U_0^{[1]} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \end{bmatrix}, U_1^{[1]} = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$U_0^{[2]} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, U_1^{[2]} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$U_0^{[3]} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, U_1^{[3]} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

For $|000\rangle$

$$U_0^{[1]} U_0^{[2]} U_0^{[3]} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}$$

For $|111\rangle$

$$U_1^{[1]} U_1^{[2]} U_1^{[3]} = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}$$

- MPS works well when:
  - The system has **limited entanglement**
  - The circuit is mostly **1D (nearest-neighbor interactions)**
  - The depth is not too large
- In these cases, memory scales roughly as
  - $O(n\chi^2)$ where:
- n= number of qubits
- χ = bond dimension (entanglement size)

Basic Concept Beyond MPS *

- The fundamental idea of MPS is Singular Value Decomposition (SVD) in linear algebra

- SVD decomposes any rectangular matrix into three simpler matrices

- Given a mtrix A of size $m \times n$, SVD expresses A as a product of three matrices

$$A = U\Sigma V^T$$

Where

U is an $m \times m$ orthogonal matrix, where the columns of U are called left singular vectors

Σ is an $\mathrm{m} \times n$ diagonal matrix, where the diagonal elements are called singular values of A

V^T is $n \times n$ orthogonal matrix, where the rows of V^T are called right singular vectors

Example of SVD

- The matrix $A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$ can be decomposed into

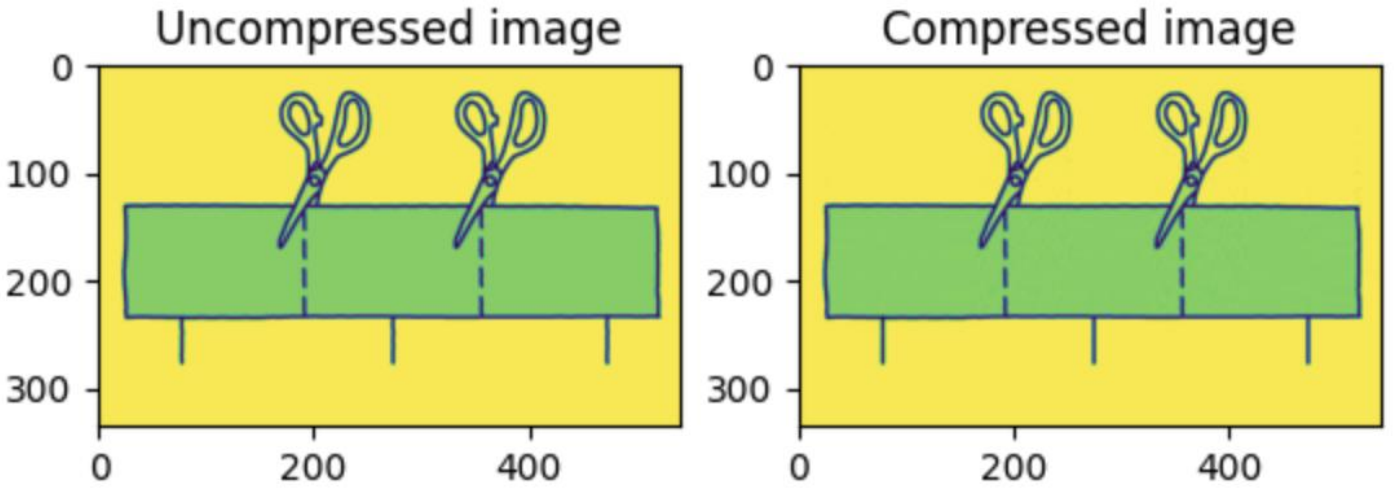
$$\Sigma = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}$$

$$\mathrm{U} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$$

$$\mathrm{V} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{pmatrix}$$

Example of SVD*

- Why SVD is important?
- It is used in computer vision to compress images without losing information



Advantages and Disadvantages of MPS

- Advantages
  - Efficient Representation
    - Offers a highly compact representation for states with low entanglement, (D^2N parameters instead of 2^N)
      - Example: 50 qubits need 18PB using state vectors, but only 6.6MB using MPS
  - Computational Methods
    - Used in numerical methods like the Density Matrix Renormalization Group (DMRG), used to approximate the ground state of many-body quantum systems
  - Capturing Local Entanglement
    - MPS can capture and represent local entanglement efficiently, making them ideal for systems where the entanglement is localized or weak
- Limitation and Challenges
  - Not Suitable for High-Entanglement States
    - MPS are best suited for low-entanglement states. For highly entangled states, such as those that appear in quantum circuits with non-local interactions or 2D systems, MPS may not provide an efficient description
  - Higher-Dimensional Systems
    - MPS are primarily designed for **1D systems**.

Comparison

Feature	State Vector, Unitary, Density matrix	Stabilizer	Matrix Product State
General applicability	General, works for any quantum circuit	Limited to stabilizer states and Clifford gates	Effective for low-entanglement, 1D systems
Efficiency	Exponential in system size	Polynomial for stabilizer circuits (Clifford gates)	Linear for 1D systems with limited entanglement
Memory Requirements	Exponential (2^N qubits)	Polynomial for stabilizer states	Polynomial (depends on bond dimension)
Exact Simulation	Yes	Yes (for stabilizer circuits)	Yes (for low-entanglement states)
Simulation Scope	Universal, works for all quantum circuits	Limited to Clifford gates and stabilizer circuits	Limited to 1D, low-entanglement states