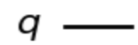
Quantum Challenge 2023 Week 1 Report

Seoul National University
Electric & Computer Engineering
Sangyeon Lee
이 상 연

1. Fair Quantum Coin Operator & Four-Sided Fair Quantum Coin Operator

1) Quantum Coin State

Quantum Coin state $|s\rangle \in H_c = \{a_0|0\rangle + a_1|1\rangle : a_0, a_1 \in \mathbb{C}\}$



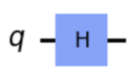
The quantum coin state is represented using the up and down states of a 1 qubit register.

2) Fair Quantum Coin Operator

Fair Coin Operator

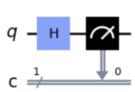
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\langle 0| + \frac{|0\rangle - |1\rangle}{\sqrt{2}}\langle 1|$$

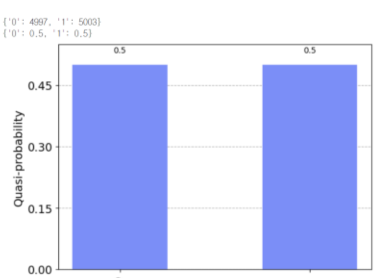
Fair Quantum Coin Operator



This operator uses the Hadamard gate to create superposition states from the 0 and 1 states. Each superposition state has an equal probability (0.5) of measuring either 0 or 1.

Verifying Fair Quantum Coin Operator

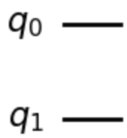




Verification: The fair quantum coin operator was applied once, and the quasi-probability of the 0 and 1 states was measured, with each being 0.5.

3) Four-Sided Quantum Coin State

Four sided Quantum Coin state $|s\rangle \in H_c' = \{a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle : a_0, a_1, a_2, a_3 \in \mathbb{C}\}$



0, 1, 2, 3 are used to represent the four-sided quantum coin state and are encoded in a 2 qubit register.

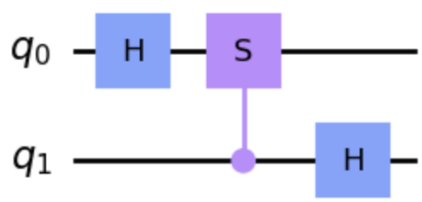
4) Four-Sided Fair Quantum Coin Operator

Four-Sided Fair Quantum Coin Operator

Four Sided Fair Coin Operator

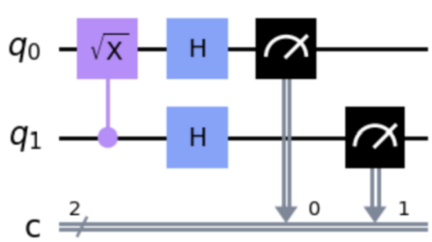
$$H' = \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & i & -1 & -i\\ 1 & -1 & 1 & -1\\ 1 & -i & -1 & i\end{bmatrix} = \frac{|0\rangle+|1\rangle+|2\rangle+|3\rangle}{\sqrt{4}}\langle 0| + \frac{|0\rangle+i|1\rangle-|2\rangle-i|3\rangle}{\sqrt{4}}\langle 1| + \frac{|0\rangle-|1\rangle+|2\rangle-|3\rangle}{\sqrt{4}}\langle 2| + \frac{|0\rangle-i|1\rangle-|2\rangle+i|3\rangle}{\sqrt{4}}\langle 3|$$

(Fourier Coin)

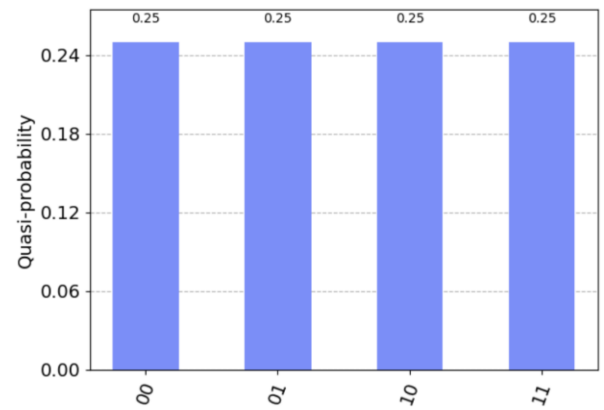


This operator employs the Quantum Fourier Transform (QFT). When N=4, the QFT matrix is used to create superposition states from the 0, 1, 2, 3 states. Each superposition state has an equal probability (0.25) of measuring each state.

Verifying Four-Sided Fair Quantum Coin Operator



{'00': 2486, '01': 2461, '10': 2558, '11': 2495}
{'00': 0.25, '01': 0.25, '10': 0.25, '11': 0.25}



Verification: The fair quantum coin operator was applied once, and the quasi-probability of the 0, 1, 2, 3 states was measured, with each being 0.25.

5) Fairness Check

$$|H_{ij}| = |\langle i|H|j\rangle| = \frac{1}{\sqrt{2}},\quad \text{all matrix elements of } H \text{ has same absolute value } \frac{1}{\sqrt{N}},$$

$$|H'_{ij}| = |\langle i|H'|j\rangle| = \frac{1}{\sqrt{4}} \quad \text{thus unbiased coin.}$$

The size of each coin operator's entry is 1/√N, which suggests it is unbiased. This was confirmed in the verifying process of each coin operator.

2. Location on the Board

location on the board

$$|\psi\rangle \in \mathcal{H}_p = \left\{ \underbrace{\sum_{k=0}^{15} a_k |k\rangle}_{16\text{개 state}} : a_0, \cdots, a_{15} \in \mathbb{C} \right\}$$

encoding by $|j_3 j_2 j_1 j_0\rangle$

$|0000\rangle := |0\rangle$

$|0001\rangle := |1\rangle$

$\vdots$

$|1111\rangle := |15\rangle$

To encode the 16 board positions, a 4 qubit register is needed. The start and end points are encoded as 0000 and 1111, respectively.

3. Quantum Coin with Shift Operator

1) Shift Operator

Right Shift Operator

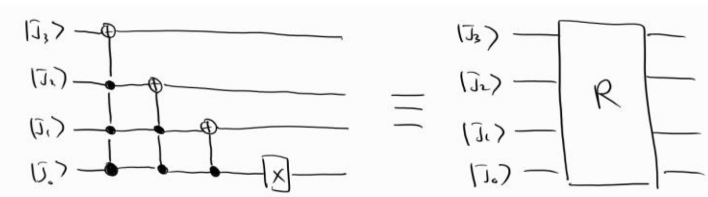
$$R=\begin{bmatrix} 0 & 0 & \cdots & & 1 \\ 1 & 0 & 0 & & 0 \\ \vdots & 1 & \ddots & & \vdots \\ & & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \qquad L=\begin{bmatrix} 0 & 1 & \cdots & & 0 & 0 \\ 0 & 0 & 1 & & & \vdots \\ \vdots & & 0 & \ddots & & 1 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & & & & & 1 \\ 1 & & \cdots & & & 0 \end{bmatrix}$$

$$L = |15\rangle\langle 0| + \sum_{k=0}^{14} |k\rangle\langle k+1|$$
$$= |1111\rangle\langle 0000| + |0000\rangle\langle 0001| + |0001\rangle\langle 0010| + \cdots + |1110\rangle\langle 1111|$$
$$R = |0\rangle\langle 15| + \sum_{k=0}^{14} |k+1\rangle\langle k|$$
$$= |0000\rangle\langle 1111| + |0001\rangle\langle 0000| + |0010\rangle\langle 0001| + \cdots + |1111\rangle\langle 1110|$$

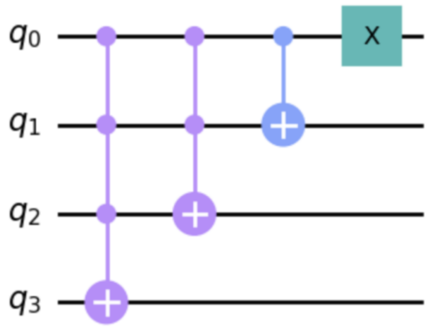
The document clarifies that the encoding method differs from a reference paper, leading to the representation interchange of left and right shift operators. I guarantee the correctness of the operator's matrix representation.

Encoding : $|0000\rangle \coloneqq \begin{matrix}1\\ \vdots\\ 0\end{matrix}$, …, $|1111\rangle \coloneqq \begin{matrix}0\\ \vdots\\ 1\end{matrix}$

Additionally, both left operator and right operator must be unitary operator. Therefore, right shift operator moves state 15 to 0, while the left shift operator moves state 0 to 15. These operators consist of C3X, CCX, CNOT, and X operators.

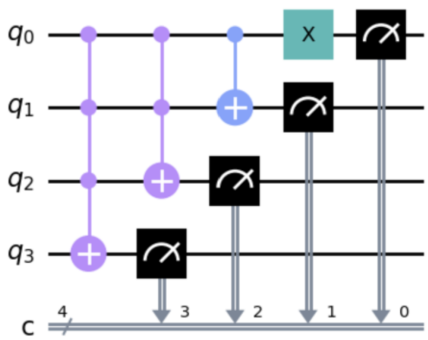


$$R = \left[I \otimes I \otimes I \otimes X\right]\left[I \otimes I \otimes \left(I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|\right)\right]$$
$$\left[I \otimes \left(I \otimes |00\rangle\langle 00| + I \otimes |01\rangle\langle 01| + I \otimes |10\rangle\langle 10| + X \otimes |11\rangle\langle 11|\right)\right]$$
$$\left[I \otimes |000\rangle\langle 000| + I \otimes |001\rangle\langle 001| + I \otimes |010\rangle\langle 010| + \cdots + X \otimes |111\rangle\langle 111|\right]$$
$$= \begin{bmatrix} 0 & 0 & & \cdots & & 1 \\ 1 & 0 & 0 & & & \vdots \\ & 1 & 0 & 0 & & \vdots \\ \vdots & & 1 & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & 0\ 0 \\ 0 & & \cdots & & & 1\ 0 \end{bmatrix} \quad \text{when } \underbrace{|0000\rangle}_{|0\rangle} = \begin{bmatrix}1\\0\\ \vdots\\ \vdots\\ 0\end{bmatrix}, \underbrace{|0001\rangle}_{|1\rangle} = \begin{bmatrix}0\\1\\ \vdots\\ \vdots\\ 0\end{bmatrix}, \cdots, \underbrace{|1111\rangle}_{|15\rangle} = \begin{bmatrix}0\\0\\ \vdots\\ \vdots\\ 0\\ 1\end{bmatrix}$$



quantum circuit diagram representing the shift operators.

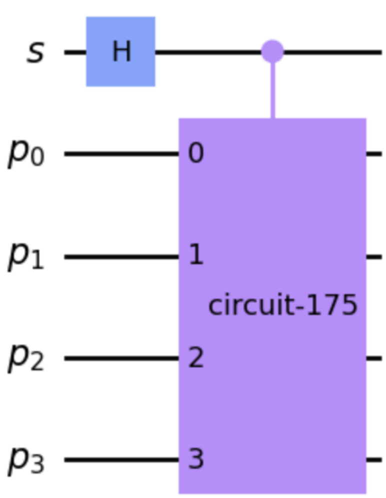
Verifying Right Shift Operator



Verification of Right Shift Operator: Several initial states were tested with the right shift operator, and the measurement results were found to be suitable.

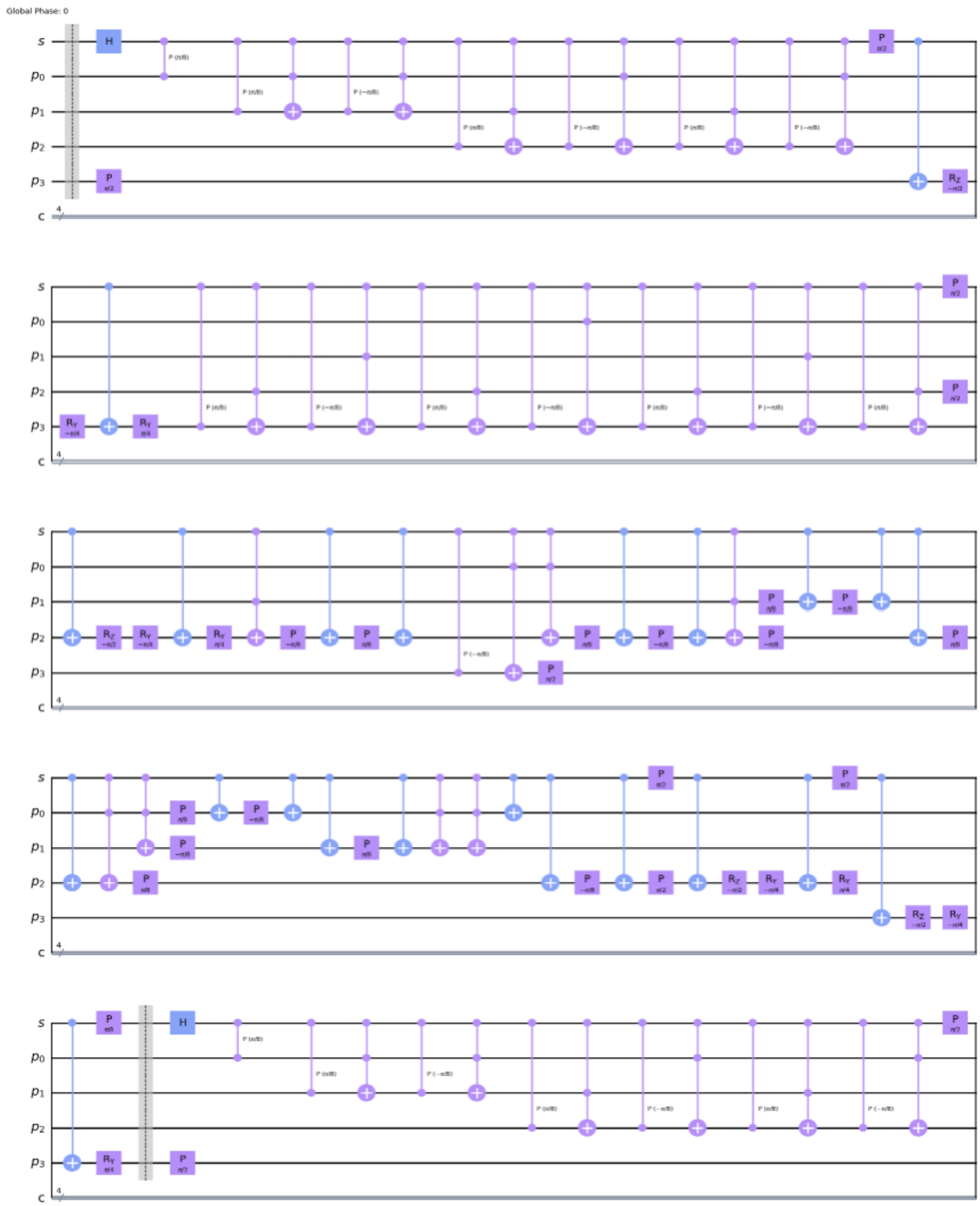
2) Shift Operator with Quantum Coin Operator

1 Step Board Evolution Operator



The quantum system of the board comprises a composite state of the Coin quantum register and position quantum register. Initially, the coin flip operator is applied, followed by identity matrix on the position register when the coin state is 1. When the coin state is 0, it's followed by shift operator on the position register. "circuit-175" in the figure is shift operator on position register.

N Step Board Evolution Operator

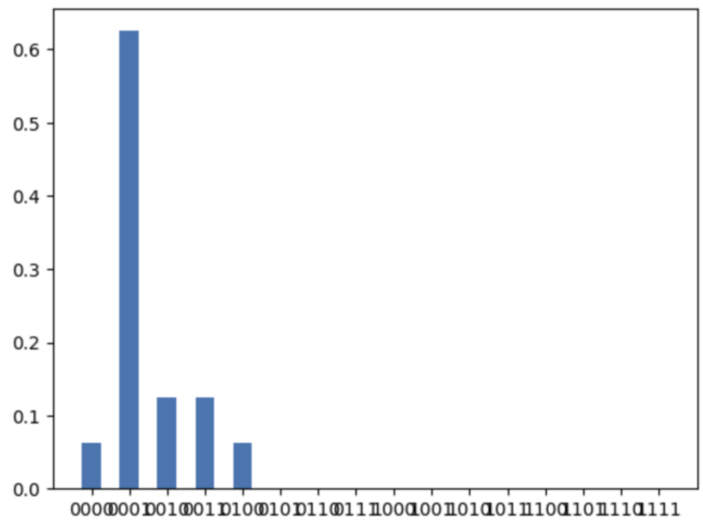


For implementation of 1 step board evolution operator on the IONQ Cloud, the operator needed to be decomposed into hardware-compatible components. The decomposition was done using Qiskit's transpiler function. Layer between barriers means 1 step of evolution. For implementing N step implementation, “for statement” in python need to be implemented.

The document includes a diagram illustrating the steps for implementing the N step evolution, with the probability of each state measured and visualized in bar graphs for N=4 and N=33. Measurement after N step evolution is implemented on my Jupiter Notebook, which uses IONQ Simulator. (From N=0 ~ N=49)

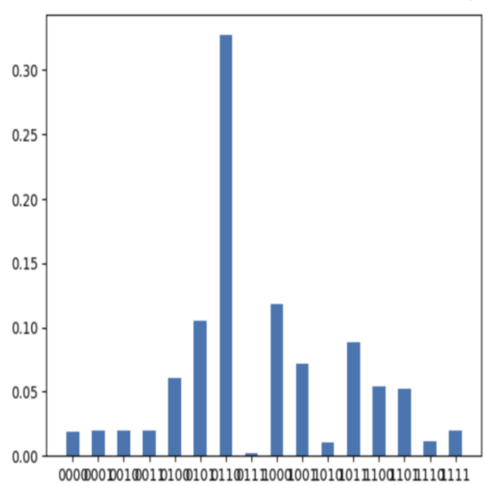
Verifying N Step Board Evolution Operator

```
4
{'0000': 48, '0001': 640, '0010': 130, '0011': 124, '0100': 58}
{'0000': 0.0625, '0001': 0.625, '0010': 0.125, '0011': 0.125, '0100': 0.0625}
```



N=4

```
33
{'0000': 26, '0001': 15, '0010': 14, '0011': 26, '0100': 64, '0101': 86, '0110': 358, '0111': 1, '1000': 106, '1001': 67, '1010': 6, '10
11': 89, '1100': 52, '1101': 49, '1110': 17, '1111': 24}
{'0000': 0.019192882, '0001': 0.019288786, '0010': 0.019597367, '0011': 0.019624859, '0100': 0.060641184, '0101': 0.10550928100000001,
'0110': 0.326655045, '0111': 0.002319723, '1000': 0.118175931, '1001': 0.071969516, '1010': 0.010329559, '1011': 0.088611752, '1100': 0.
054021015, '1101': 0.052439212, '1110': 0.011716500000000001, '1111': 0.019907385}
```



N=33

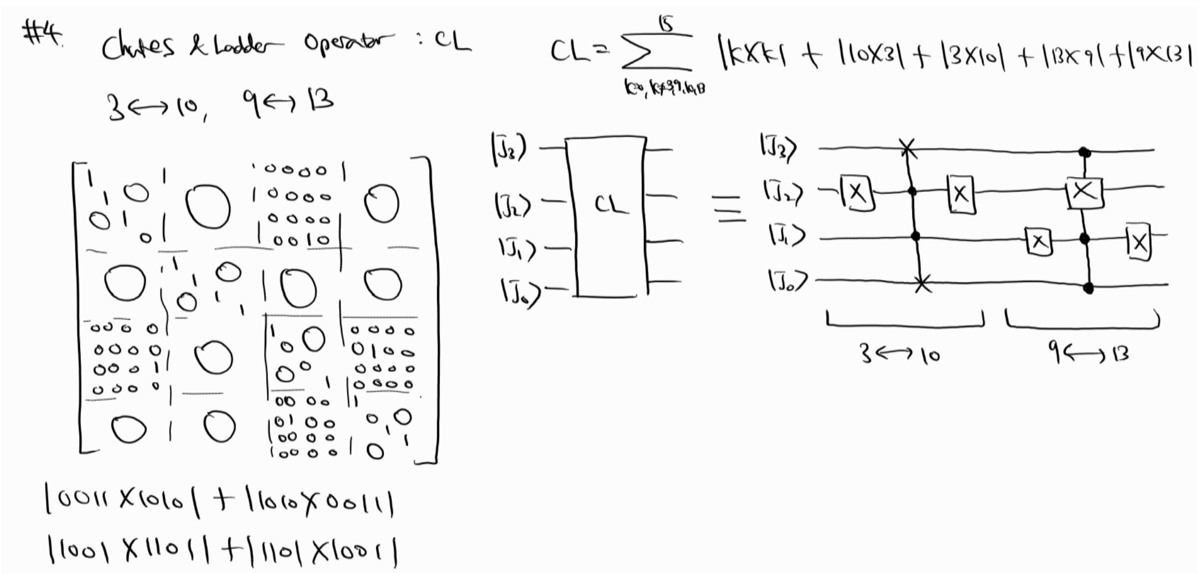
After measuring N step evolution, it can be observed that the superposition of the quantum state occurs. Unlike in Classical Walks where the distribution progressively approximates a Gaussian, in

Quantum Walks, there is no converging distribution even as N increases. It can also be observed that the distribution is not created symmetrically, which is due to the coin quantum operator being asymmetric.

4. Adding Chutes and Ladders in Board

1) Chutes & Ladder Operator (Notated as CL)

Chutes & Ladder Operator



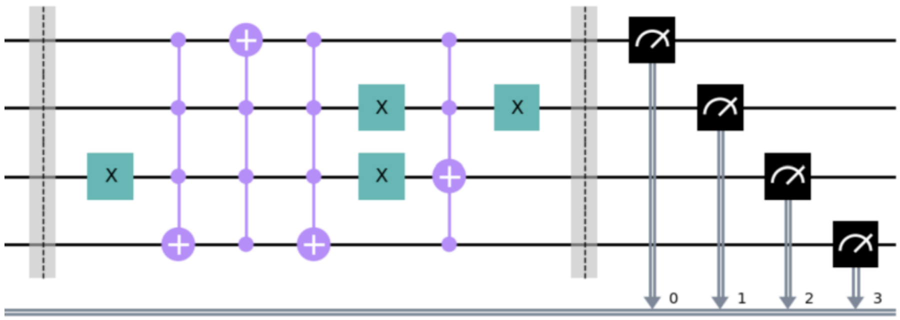
Explanation of figure:

3 <-> 10: When $|j_2 j_1\rangle = |01\rangle$, swap $|j_0\rangle$ and $|j_3\rangle$. Extra case of $|j_2 j_1\rangle$, operate Identical Matrix on $|j_0\rangle$ and $|j_3\rangle$. Therefore, controlled-swap gate is suitable (Control Bit is $|j_2 j_1\rangle$)

9 <-> 13: When $|j_3 j_1 j_0\rangle = |101\rangle$, X gate on $|j_2\rangle$. Extra case of $|j_3 j_1 j_0\rangle$, on $|j_2\rangle$. Therefore, C3X gate is suitable. Chutes & ladder operator is composed of serial connection of swapping 3 <-> 10 and 9 <-> 13.

Therefore, the entire operator applies the coin quantum flip operator to the coin state and then applies the controlled – shift operator to the position register. Lastly, taking the chutes & ladder operator to the position register is a step forward in evolution.

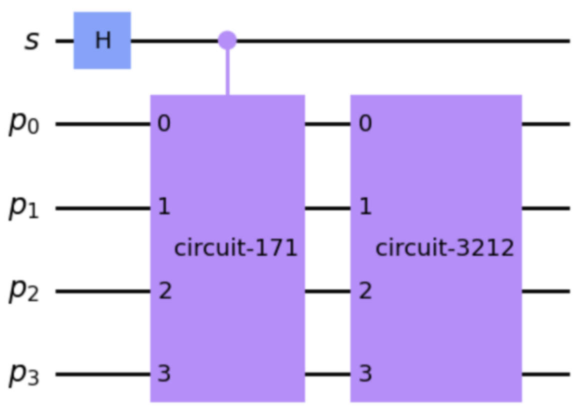
I went through a process to determine whether the Chutes & Ladder Operator operates appropriately for various initial states. Results 3 <-> 10, 9 <-> 13 It was confirmed that even instantaneous movement was working normally.



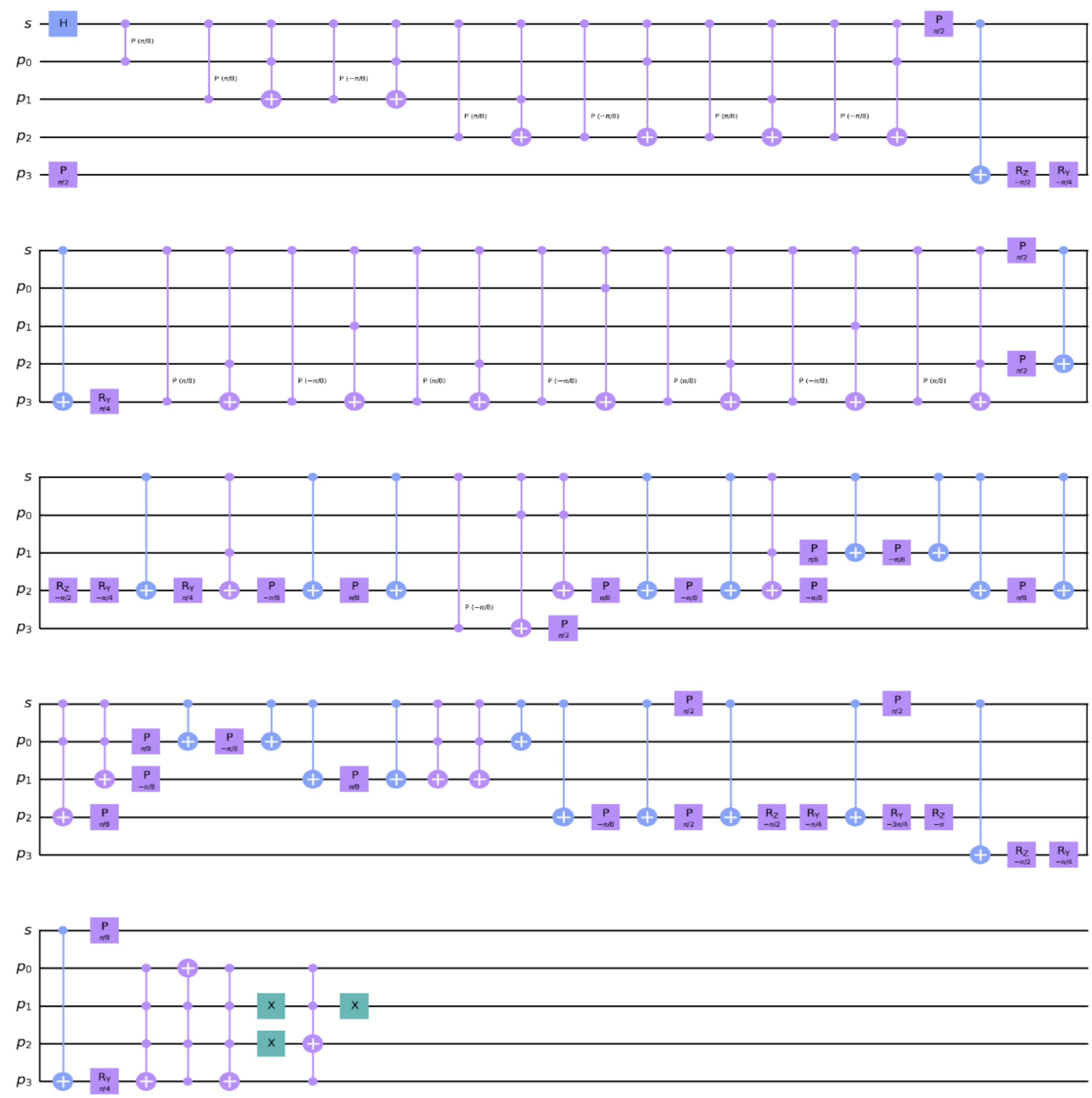
This is a circuit that decomposes the Chutes & Ladder Operator to suit the hardware. You can see that the Controlled-Swap gate has been decomposed into three CNOT gates.

2) Chutes & Ladders Board Evolution Operator

1 Step Chutes & Ladders Board Evolution Operator



This is a circuit before decomposing the entire operator to suit the hardware. First, a Hadamard gate is taken for the coin state, and then a controlled-shift operator is taken using the coin state as the control bit. Next, the Chutes & Ladder operator verified in 1) is applied to the position state. Since each operator is a unitary operator, the 1 step operator is also a unitary operator. If you perform N steps of the operator corresponding to 1 step, you can obtain the superposition state after N steps.



This is a circuit that decomposes the entire 1 board evolution operator (with Chutes & Ladder) to suit IONQ hardware. The length of the circuit has become longer due to the controlled-shift operator.

5. Discussion

1) Role of Measurement in Quantum Chutes and Ladders Game

State measurement makes wave function collapse. Collapse of wave function arises randomness of Quantum Chutes and Ladders Game.

2) Measurement between Turns

Measurement between turns collapse wave function and transform quantum information to classical information, stored in classical register.

3) No Measurement between Turns

No measurement makes unitary evolution by each step. This procedure can obtain quantum superposition of states.

4) Quantum Analog of the "Memoryless" Nature of the Classical Game?

Quantum walks are quantum analogues of classical random walks.

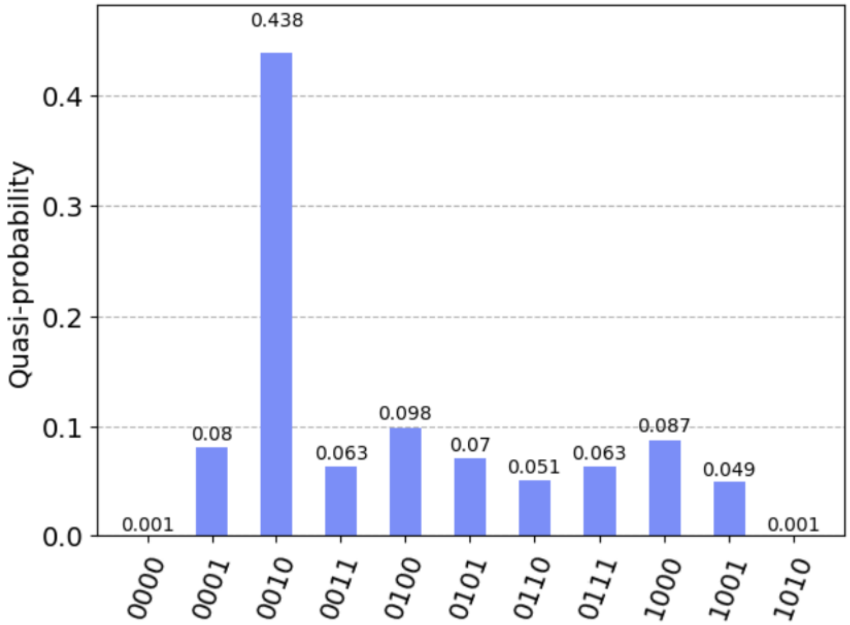
Classical Walk: Walker occupies definite states and randomness arises from stochastic transitions defined by Markov's Process.

Quantum Walk: Randomness arises through quantum superposition of states, non-random, reversible unitary evolution and collapse of the wave function due to state measurements.

6. Compare between #3 & #4 (10 Step)

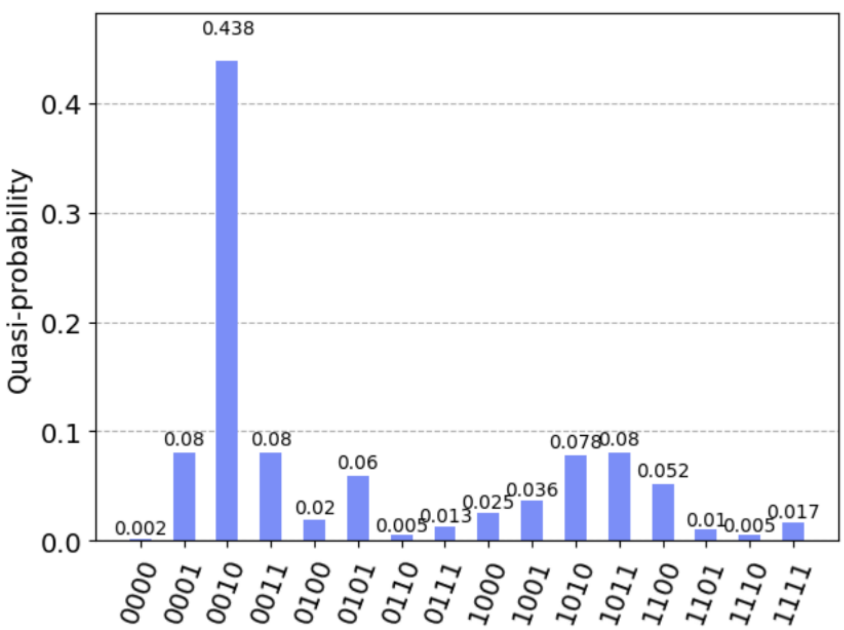
1) Task 3 (Not Containing Chutes & Ladders)

{'0000': 11, '0001': 787, '0010': 4341, '0011': 627, '0100': 1011, '0101': 726, '0110': 509, '0111': 621, '1000': 863, '1001': 495, '1010': 9}
{'0000': 0.000976562, '0001': 0.080078124, '0010': 0.438476562, '0011': 0.0625, '0100': 0.09765625, '0101': 0.0703125, '0110': 0.05078125, '0111': 0.0625, '1000': 0.086914062, '1001': 0.048828124, '1010': 0.000976563}



2) Task 4 (Not Containing Chutes & Ladders)

{'0000': 18, '0001': 816, '0010': 4368, '0011': 783, '0100': 184, '0101': 589, '0110': 49, '0111': 119, '1000': 260, '1001': 366, '1010': 759, '1011': 812, '1100': 566, '1101': 102, '1110': 44, '1111': 165}
{'0000': 0.001953125, '0001': 0.080078124, '0010': 0.438476562, '0011': 0.080078124, '0100': 0.01953125, '0101': 0.059570312, '0110': 0.004882812, '0111': 0.012695312, '1000': 0.025390624, '1001': 0.036132813, '1010': 0.078125, '1011': 0.080078124, '1100': 0.051757812, '1101': 0.009765624, '1110': 0.004882812, '1111': 0.016601562}



Firstly, when N=10, the version Not Containing [Chutes & Ladders] only has probabilities of observing states from 0000 to 1010, whereas the Containing version, due to the teleporting operator, has probabilities of observing all states from 0000 to 1111.

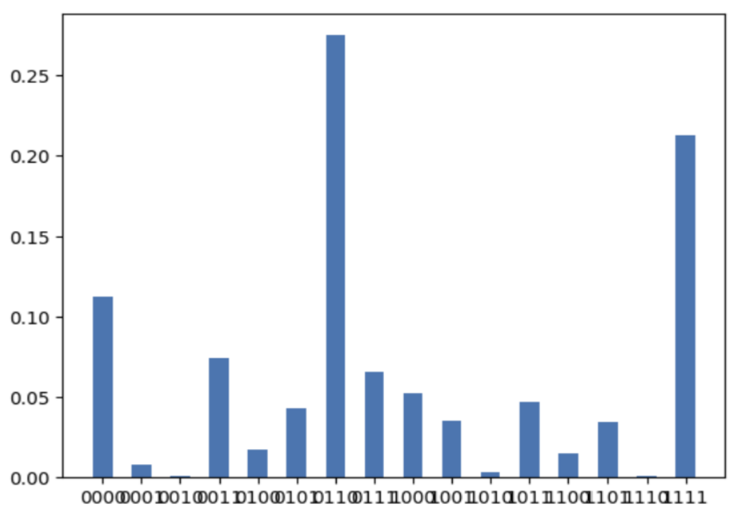
For the 1000 state, in the Containing version, the probability of the 0011 state has moved significantly towards around 1010 due to the ladder between 0011 and 1010. Therefore, it can be observed that the probability of the 1000 state in the Containing version is smaller than that in the Not Containing version.

In the case of the 1010 state, while the probability of observing the 1010 state in the Not Containing version is 0.001, in the Containing version, it is 0.078. This can be interpreted as the probability of the 0011 state continuously moving due to Chutes & Ladders. The existence of the 9 <-> 13 ladder can also be interpreted as creating a probability trap that grabs probability from 9~13. Hence, when N is small, it can be observed that the probability of observing states between 1001 and 1101 is relatively high.

3) N Step Board Operator Containing Chutes & Ladders

The version including Chutes & Ladders also underwent a process of measuring the probability after N steps for all natural numbers N from 0 to 49. Among these, the result graph for N=33 will be attached. When N is bigger than 25, Trap between 1001~1101 is disappeared.

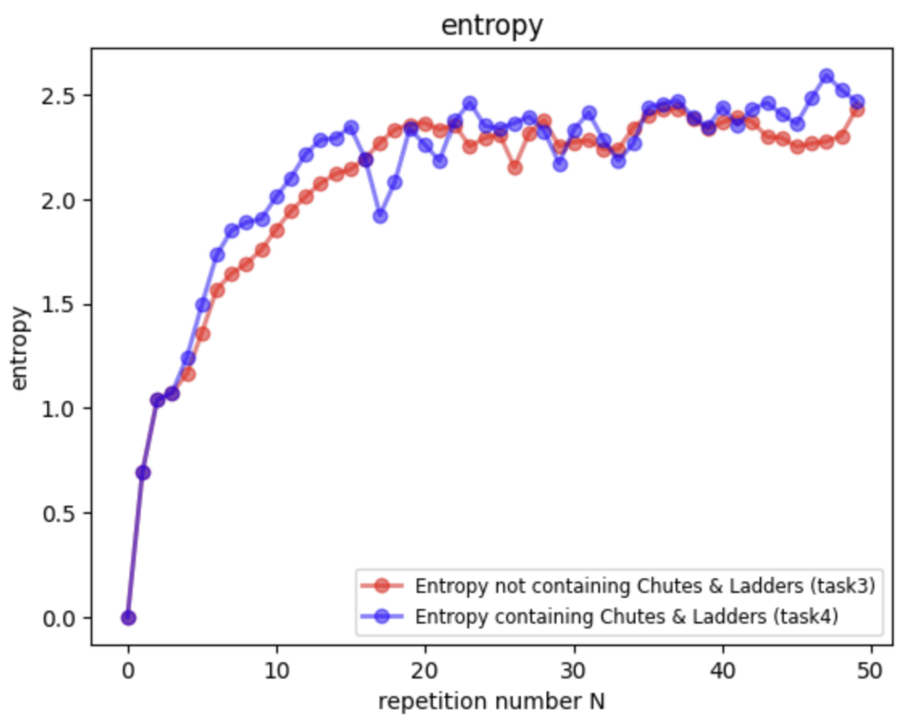
33
{'0000': 146, '0001': 7, '0010': 1, '0011': 60, '0100': 14, '0101': 40, '0110': 267, '0111': 63, '1000': 70, '1001': 35, '1010': 3, '1011': 51, '1100': 10, '1101': 41, '1110': 0, '1111': 192}
{'0000': 0.112706712, '0001': 0.007829257999999999, '0010': 0.0009880380000000001, '0011': 0.074423598, '0100': 0.017590672, '0101': 0.043315038, '0110': 0.275062879, '0111': 0.06596548299999999, '1000': 0.05275086499999994, '1001': 0.035058029, '1010': 0.0037321940000000003, '1011': 0.047303868, '1100': 0.015321038, '1101': 0.034309361, '1110': 0.001023436, '1111': 0.212619532}



N=33

Comparing **entropy** between task3 and task4

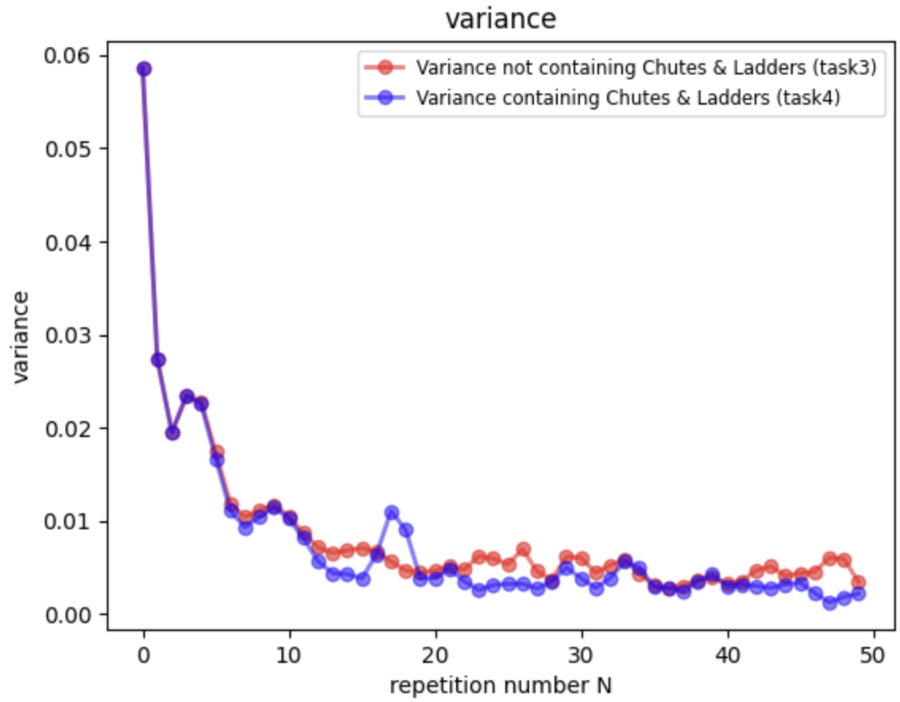
A graph of the entropy of the probability distribution depending on N was plotted. The red line in the graph below represents the entropy of the probability distribution in the case without Chutes & Ladders. The blue line represents the entropy of the probability distribution in the case including Chutes & Ladders.



In a classical walk, since the distribution converges to a specific distribution, as N increases, the entropy also converges to a specific value. However, in a quantum walk, regardless of whether Chutes & Ladders are included, the entropy does not converge to a specific value. It was observed that there is a consistent fluctuation in entropy even as N increases. This is attributed to the randomness of the quantum walk.

Additionally, it was observed that the entropy is higher when Chutes & Ladders are present. This is because Chutes & Ladders exchange probabilities in different areas(3<->10, 9<->13), leading to a more evenly spread probability.

However, when Chutes & Ladders are present, due to the randomness of the quantum walk, between N=16 and 19, the probability concentrates on the 1000 state. (This can be confirmed in the source code.) Therefore, between N=16 and 19, it can be observed that the entropy becomes smaller when Chutes & Ladders are included.



We can find similar result in Variance – N graph.