Qubit State Visual Debugger

Adrien Hadj-Chaib
ahadjchaib@ucla.edu
Computer Science Department
University of California, Los Angeles

Alejandro Zapata
azapataa@g.ucla.edu
Computer Science Department
University of California, Los Angeles

June 13, 2021

1 What is This Tool For?

This is a Python 3 (3.7+) visual debugging tool for Cirq circuits. Use it to visualize qubit superposition, check for entanglement, ensure that your circuit behavior matches your intuition for what you are trying to code, and understand the state of your qubits at specific moments.

2 Tool Dependencies

This tool makes use of Numpy, Seaborn, Matplotlib, and Cirq. Please make sure to install these dependencies to use the tool.

- https://numpy.org/install/
- https://seaborn.pydata.org/installing.html
- https://matplotlib.org/stable/users/installing.html
- https://quantumai.google/cirq/install

2.1 How to Use it?

The tool is a function with the following signature:

```
def debug(circuit, iterations=1000, measure_insert=None, kind="qubit_entanglement" )
```

Given a Cirq **circuit** object, this function will visualize the qubit relationships via repeated simulation.

iterations is the number of times simulation is run; a higher number yields more data, so the default is set to 1000.

measure_insert is default to **None**; supply a moment index (integer) to have the debugger insert measurement gates at that moment (other measurement gates will be removed, since more than one per qubit is not allowed).

kind specifies the type of visualization you would like to see; there are three options: **state_superposition**, **qubit_superposition**, and **qubit_entanglement** An explanation and examples for each of these modes follows below.

3 Debugging Operations

3.1 Qubit Entanglement

Qubit entanglement, one of the most prominent features of quantum computing, occurs when two or more qubits share a state which, when observed, results in the state of the qubits involved becoming known. This debugging tool allows for a simple visualization to help in the understanding of entangled *pairs* of qubits. When using this option, the visual generated will be a heatmap matrix of qubit relationships where each entry is a score of the relationship between two qubits (the diagonal of this matrix is not useful since it describes qubit i vs qubit i).

The relationship score is a number between 0 and 1, where 0 entails that the two qubits involved output equal measurements, and 1 entails that the two qubits involved output inverse measurements of each other; thus revealing an entanglement between the two. A score around 0.5 indicates that two qubits share no relationship with each other. The score used involves the Hamming distance between two qubit measurement sequences (1110001 and 0001110 have a hamming distance of 1 since they are opposite, so the two qubits involved in these sequences are possibly entangled).

Because two qubits can be made to output inverse measurements of each other via means other than entanglement, this visualization should be used to check that qubits have been entangled successfully, and not for identifying entangled qubits with complete certainty.

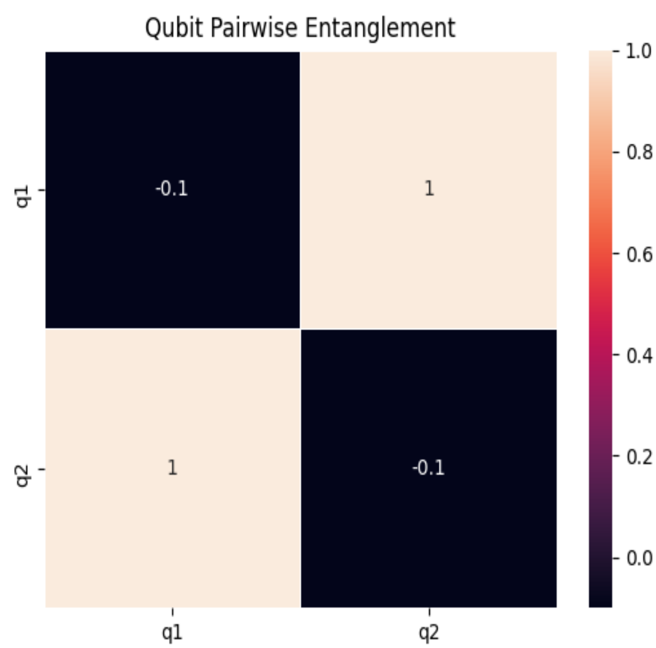


Figure 1: Two entangled qubits, q_1 and q_2, share a score of 1.

3.2 Qubit Superposition

The state of a qubit is oftentimes nondeterministic, especially when the circuit is composed of superposition operations such as a Hadamard gate, and/or entanglement operations such as a CNOT gate. These operations compound as the circuit grows, making tracking the superposition state of a single qubit at a given moment difficult. Having access to such information can provide useful insight while debugging or verifying a circuit's correctness.

We run the circuit **iterations** times gathering all the measurements. For each qubit, we compute the probability α of measuring 1 from frequency, reported on the graph. We can then deduce the state of the qubit q:

q = (1.0 - α) $|0>$ + α $|1>$

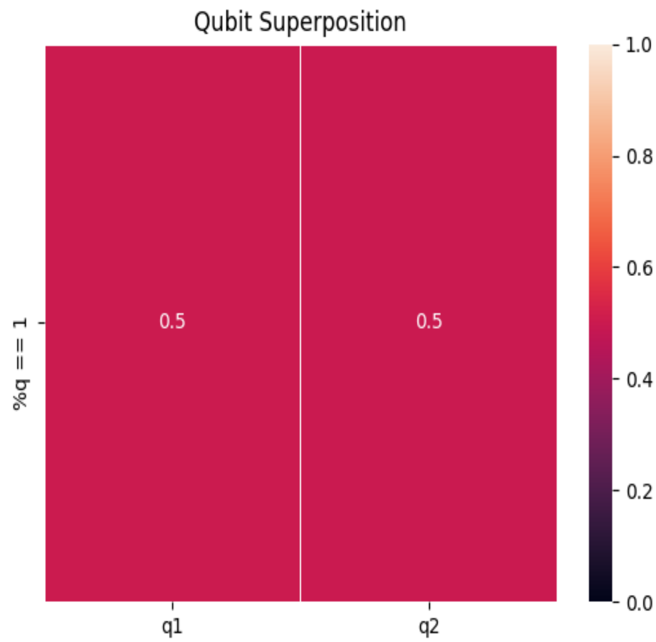


Figure 2: Two qubits that have the same probability of being $|0>$ or $|1>$

3.3 State Superposition

Similar to the qubit case, the whole state of the quantum circuit at a given moment is nondeterministic. Being able to visualize the state distribution can also prove useful in inspecting a circuit. We run the circuit **iterations** times gathering all the measurements. Measurements are stored in in the following format:

$\mathrm{m}_{state} = \mathrm{m}_{n-1}$ ... $\mathrm{m}_1\mathrm{m}_0$, such that

$measurement(q_i) = m_i \in \{0, 1\}$, making of m_{state} a n long binary bit-string representing the whole state of the circuit.

We then compute the frequency of every measurement, giving us the probability of each possible circuit state for the measured moment. The frequencies are then placed in a 2D matrix, of dimension *row* x *col* such that:

$number\_states \leq row \times col$

The matrix entries are numbered from left to right, and top to bottom, with the top left being index 0, bottom right being index $(row \times col) - 1$. Therefore each matrix entry has a probability value, and an integer index. From there we extract the following information:

m_{state} = index as binary value

$P(\mathrm{m}_{state})$ = matrix[index]

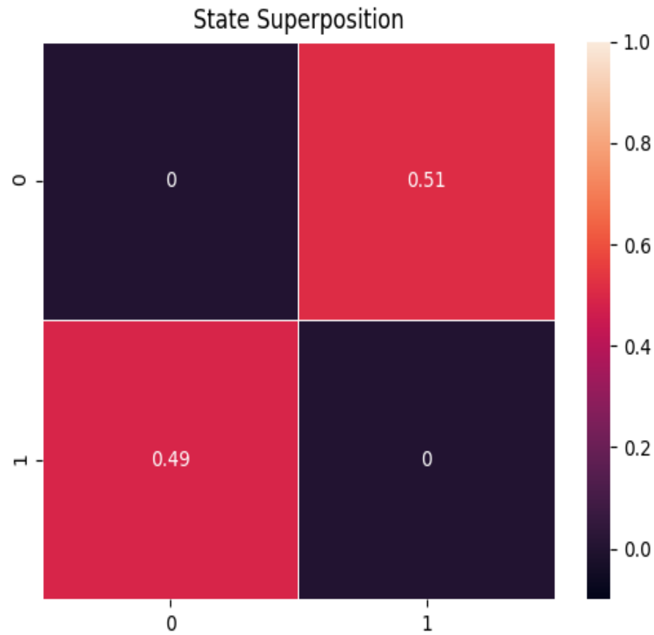


Figure 3: State = $0.51 * |01> +0.49 * |10>$.

4 Use Cases

4.1 5 Qubit Hadamard Circuit

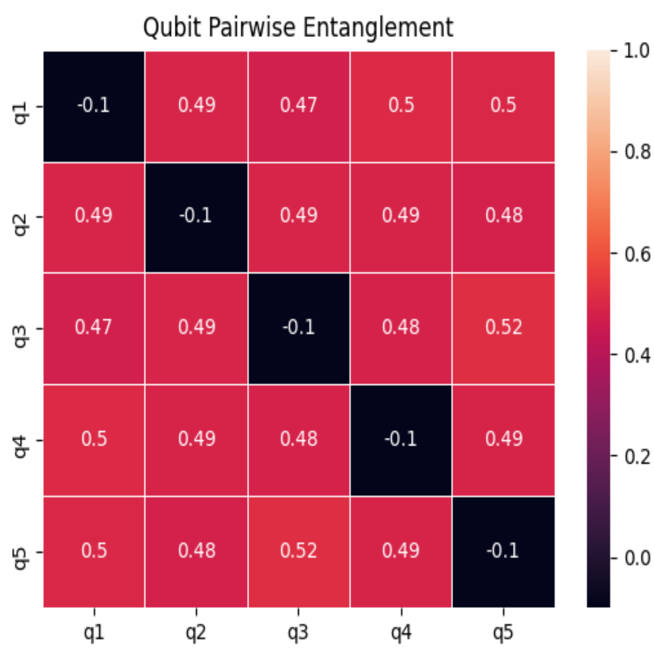


Figure 4: Each pair of qubits is disentangled from each other.

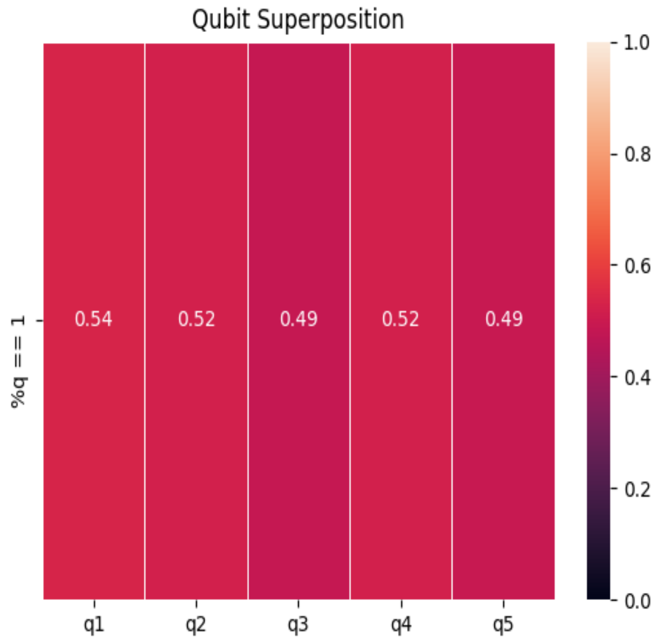


Figure 5: Each qubit is almost as likely to be $|0>$ or $|1>$.

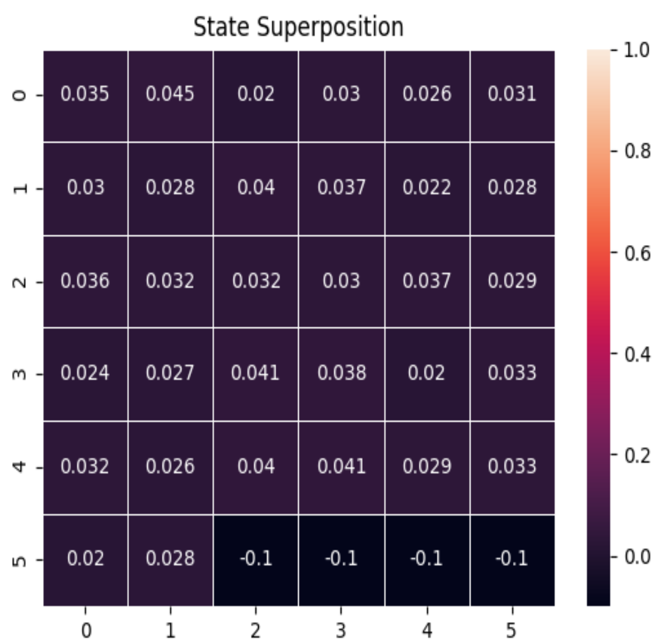


Figure 6: All 5 qubit states are close to being equally distributed.

4.2 2 Qubit Grover Circuit

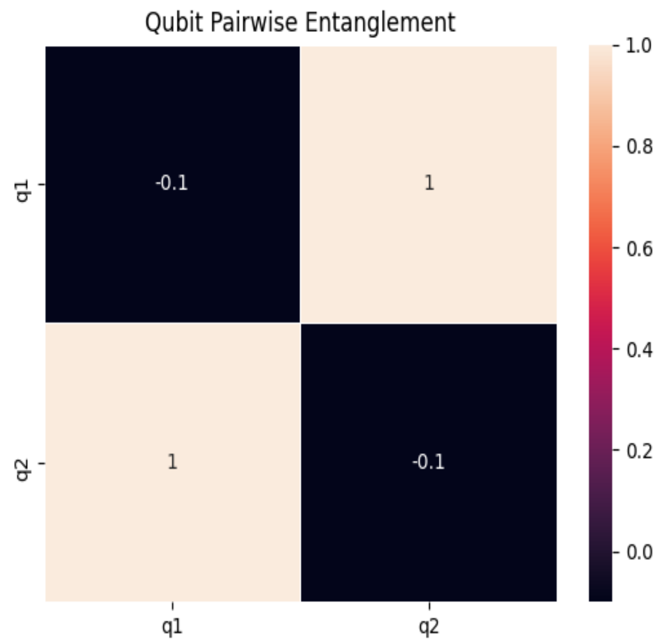


Figure 7: Both qubits have exactly opposite values.

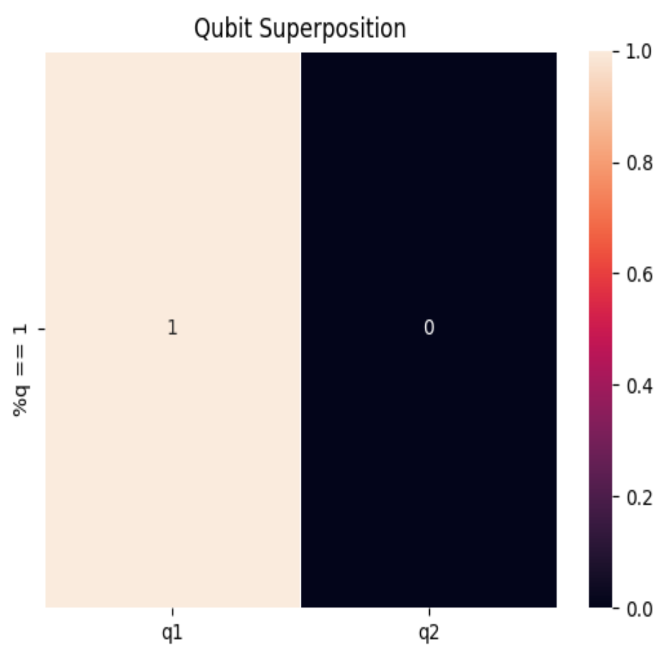


Figure 8: q_1 is guaranteed to be $|1>$, q_2 is guaranteed to be $|0>$.

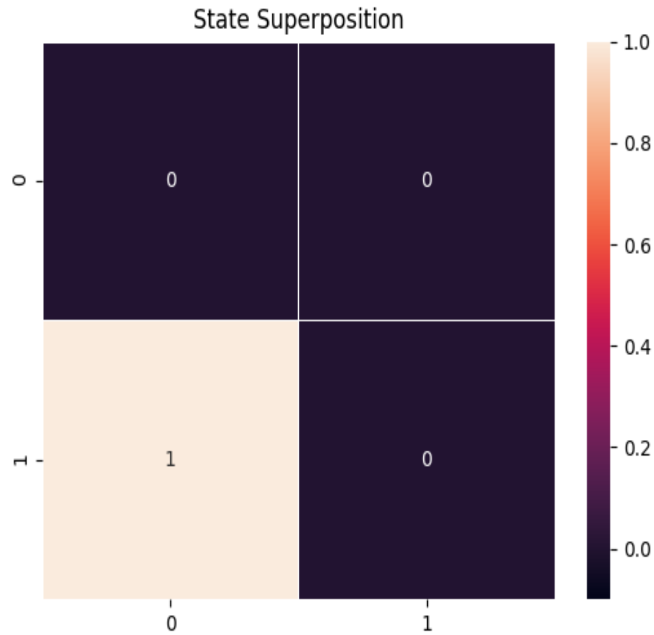
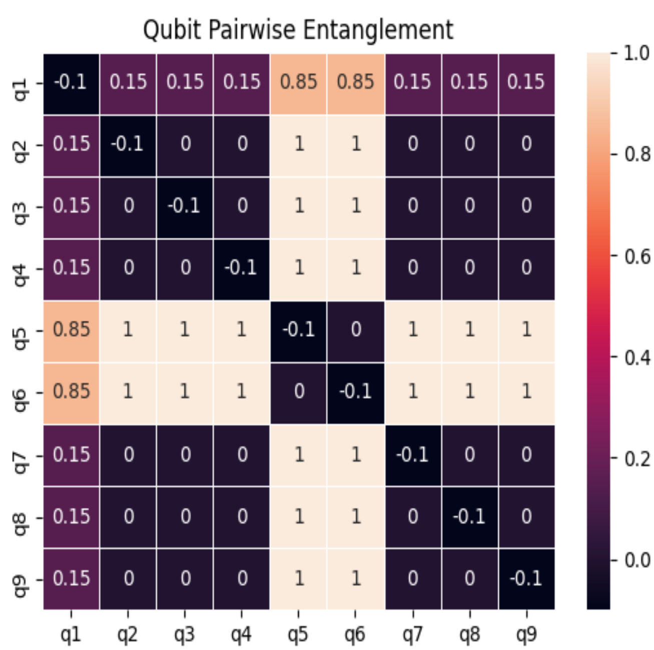


Figure 9: State $= |10>$.

4.3 9 Qubit Shor Code Circuit



	q1	q2	q3	q4	q5	q6	q7	q8	q9
q1	-0.1	0.15	0.15	0.15	0.85	0.85	0.15	0.15	0.15
q2	0.15	-0.1	0	0	1	1	0	0	0
q3	0.15	0	-0.1	0	1	1	0	0	0
q4	0.15	0	0	-0.1	1	1	0	0	0
q5	0.85	1	1	1	-0.1	0	1	1	1
q6	0.85	1	1	1	0	-0.1	1	1	1
q7	0.15	0	0	0	1	1	-0.1	0	0
q8	0.15	0	0	0	1	1	0	-0.1	0
q9	0.15	0	0	0	1	1	0	0	-0.1

Figure 10: q5 and q6 are possibly entangled with q2, q3, q4, q7, q8, and q9.

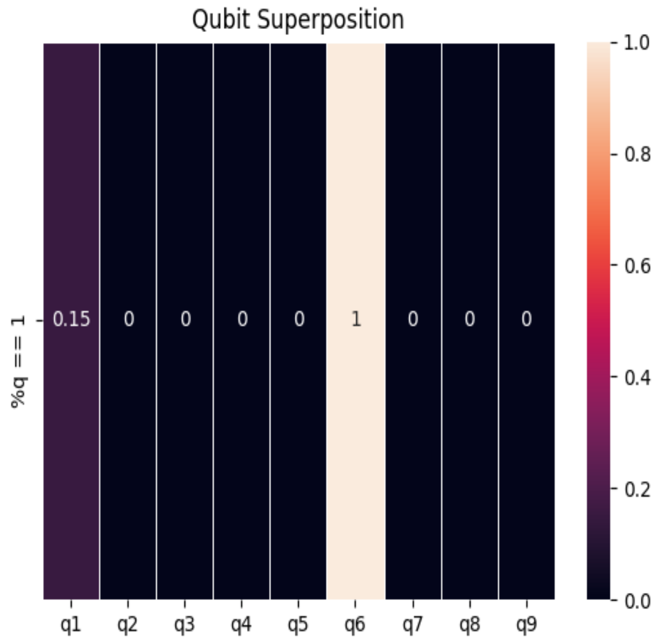


Figure 11: q_6 is guaranteed to be $|1>$; q_2, q_3 ,q_4, q_5, q_7, q_8, and q_9 are guaranteed to be $|0>$; $q_1 = 0.85 * |0> +0.15 * |1>$.

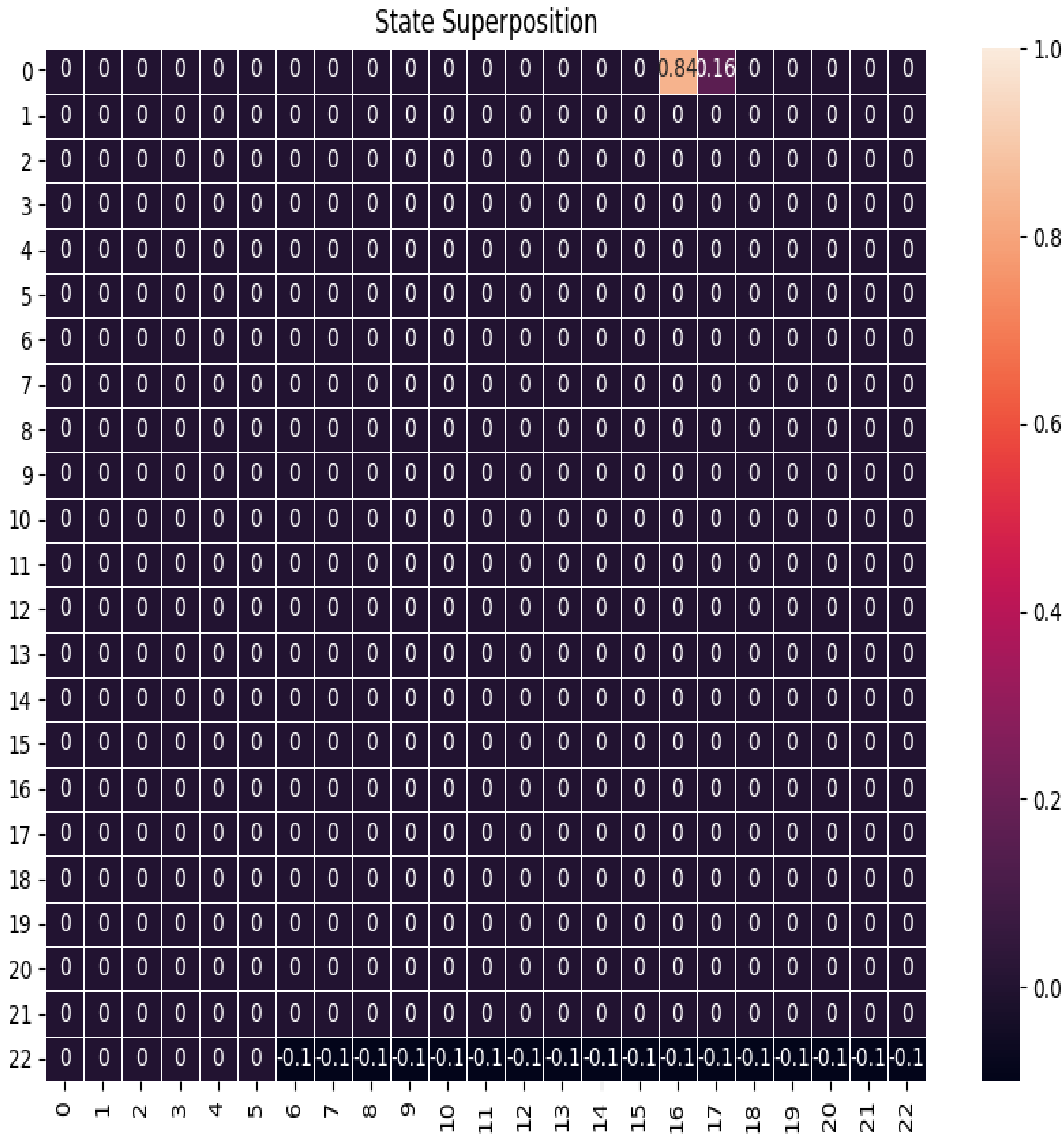


Figure 12: State $= 0.84 * |000010000> +0.16 * |000010001>$.