QRISE RNG Algorithm

Authored by team QUAFFLES :

Pranav Bhatt – pranavbha@berkeley.edu

Achyut Chebiyam - achebiyam@berkeley.edu

Ben Flitcroft - bflitcroft@berkeley.edu

+Started 04/01/2024

Microsoft Quantum research challenge at QRISE 2024: Resource estimation of quantum algorithm

<u>Challenge Intro / Brainstorming ?</u>

When deciding which project to work on we were particularly drawn to Microsoft's challenge as it gave us a pretty open ended choice for how we could go about the project including which algorithm we choose and how we evaluate our implementation. When brainstorming different algorithms we could explore we wanted to choose something that is unique but also something that we had a chance to interact with before. RNG's are used everywhere and thus we decided that using an RNG is possible in conjunction with python. We have invented a specific type of RNG that only outputs odd numbers.

<u>What is the RNG ? Our implementation</u>

RNG stands for random number generator and it is an important backbone of much of computer programming, allowing for randomization in a wide variety of applications. We used the random bit generator which takes in an input of an integer for the number of bits to generate. The way this implementation works is that it initialises that many qubits with value 0, then passes through Hadamard Gate, which randomly allocates each qubit with a value of of 1 or 0 and then returns those values in a list and then resets all the qubits back to 0. After implementing the sample code in python, we were able to take our random bit strings and convert them into integers by converting from binary to decimal. The improvement we made to the model is that we implemented the algorithm in such a way to randomly return odd numbers only. We did this by experimenting with different bit generation sequences and gates. Our Quaffle RNG generator includes our own implementation of the odd number generator.

<u>Resource Estimation results:</u>

QRISE RNG Algorithm

Azure Quantum Resource Estimator

▼ Results

Run name	T factory fraction	Physical qubits	Runtime	rQOPS
qubit_gate_ns_e3, surface_code	0.00 %	300	2 microsecs	3,000,000
qubit_gate_ns_e4, surface_code	0.00 %	108	1 microsecs	5,000,000
qubit_gate_us_e3, surface_code	0.00 %	300	3 millisecs	2,000
qubit_gate_us_e4, surface_code	0.00 %	108	2 millisecs	3,334
qubit_maj_ns_e4, surface_code	0.00 %	300	10 microsecs	600,000
qubit_maj_ns_e6, surface_code	0.00 %	12	2 microsecs	3,000,000
qubit_maj_ns_e4, floquet_code	0.00 %	312	901 nanosecs	6,666,667
qubit_maj_ns_e6, floquet_code	0.00 %	24	301 nanosecs	20,000,000

QRISE RNG Algorithm

```
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
15
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
7
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
1
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
7
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
5
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
3
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
23
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
1
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
11
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
3
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
3
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
5
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
3
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
3
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
25
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
27
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
21
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
7
PS C:\Users\prana\OneDrive\tess\QRISE> & C:/Users/prana/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/prana/OneDrive/tess/tro/src/RNG.py
13
```

<u>Conclusion / Evaluation :</u>

In this project we gained a greater and deeper understanding of Q# and the interface for quantum computing/programming. We explored the intricacies of RNG models to the extent that we were able to manipulate our system parameters to only output random odd numbers. Our model successfully returns random odd numbers every time and we are very glad that we could explore this research area to gain a more foundational understanding of the field of quantum computing.