

# Analysis of Quantum Computers and Randomness

Piyush Kumar<sup>1</sup>, Dr. Deepak Chahal<sup>2</sup>

<sup>1</sup>Student, Department of IT, Jagan Institute of Management Studies, Sector -05, Rohini, New Delhi. India

<sup>2</sup>Professor, Department of IT, Jagan Institute of Management Studies, Sector -05, Rohini, New Delhi. India

**Abstract-** Generating a True random number is impossible via discrete devices which we use in our day to life, to account this quantum computers come into picture which uses principle like superposition, qubits and quantum gates to generate true random numbers.

**Index terms-** Quantum computers, superposition, entanglement, qubits, quantum gates

## I. INTRODUCTION

Well whenever we hear about Quantum computers we think about jargon related to science, we think it is beyond our domain of knowledge and it is something which is literally of no use for us, but the reality is totally opposite.

In today's world things like security is a major issue, even though we have technologies like block chain and various encryption or decryption technologies, but with the fast moving time and advancement of technology and knowledge we will ultimately end up exploiting them, as we have seen in past earlier we always used HTTP but with time they were tend to become unsafe, and we finally moved on to what known as HTTPS in today's world. Similarly, encryption is one thing which makes our data safe from third parties like hackers any other user that comes between or during the transfer of data. Integrity of data refers to protecting information from falsely being modified by an unauthorized party. Information is valuable only if it is correct, tampered information could prove costly to both the sender and the receiver party [1].

So whenever we talk about encryption one of the things which comes in our mind is randomness, you can think like if you have a lock and you have 1000 keys and I give you a chance to pick one them to open the lock then probability of selecting one out of all is always 0.001 which is actually not that bad, but devices or processors (to be specific) are so powerful

nowadays, they can easily hit and try such small number in just microseconds. Any average processor nowadays is capable of doing  $10^8$  operations in a second, which means it can try  $10^8$  combinations within a second. So picking a key to decrypt (open a lock) out of 1000 possible keys is pretty easy.

## II. PROBLEMS WITH CURRENT RANDOM NUMBER GENERATORS (PRNG)

We have implemented many algorithms to generate random numbers and various programming languages like python, java, JavaScript that uses them to generate the same, but in reality they are pseudorandom numbers. There is always a way to reverse engineer and find out how the random number is generated which totally destroys the true meaning of random number, since they are supposed to be random therefor there should be no way to know how they are generated.

As we know the computers, which we use in our day-to-day life are discrete in nature, that is they work on discrete values like and they work by following an instruction model. So expecting them to generate true random numbers is wrong in the very first place.

Let's take an example of language like Python3 whose code is given below:

```
Import random  
Print (random.randint (1, 10000))
```

Now this code is pretty straight forward, it generates a random number between [1, 10000]. For many people this will do the task but again it's a pseudorandom number, which uses one of the pseudorandom number generator (PRNG) famously known as Mersenne Twister (which uses Mersenne primes). Many other programming languages use time and date as seed for random number generation but again, we can just guess the next random number by using that future timestamp.

Another famous approach is to use chaos theory, which is a branch of mathematics that deals with the chaotic nature of equations, but the branch itself tells the detection of chaos. Now if the chaos can be calculated then how it is supposed to generate random numbers?. Even though it is one of the hardest way to guess or to calculate the chaotic point or random number to be specific, but still we can calculate. So it is also not a true approach to find or generate a true random number.

The real question arises, is there a way to truly generate a random number which is unpredictable in any way?

The answer is yes, via quantum computers. Before getting into quantum computer lets discuss about quantum superposition, and to demonstrate this we have a very well-known experiment known as: Schrödinger's cat experiment.

### III. SUPERPOSITION AND SCHRÖDINGER'S CAT EXPERIMENT

This famous experiment was done by none other than the Nobel prize-winning physicist Schrödinger's in 1935, though the idea was initially coined by Albert Einstein. It reveals various details like quantum entanglement, quantum superposition, parallel world and many more, which we are going to discuss in brief. Because it applies to quantum theory this was also done to point out a problem/ flaw in already existing Copenhagen visualisation regarding the concept of superposition.



Now we are not focusing on the detail of these experiments, as there are many things which can be concluded from this experiment which totally changed the perspective of quantum theory.

The basic notion behind this experiment was quantum superposition, that is before the box was opened to check the cat lives state it was in superposition state that is, It was both dead and alive,

as soon as we try to observe the cat, superposition collapses, and we found the cat either dead or alive (in real experiment the cat was alive). This is also known as the observer effect. There are other things like quantum entanglement but those will be too much for this paper to cover.

One topic that needs to be known for advance topics is quantum entanglement that is the various elements in the system that affects the overall quantum behaviour. For example here the radioactive element is entangled with the state of the cat. This is a whole new topic to discuss about, as it tells about the concepts of parallel universe and all the outcomes happening in each of it. It's an open field of research till this date as no solid evidence or theories are proven regarding this.

There are many other theories related to quantum world and the particles that exhibit such behaviors are famously known as quantum particles, there state is always superposition between wave and particle behavior, and under the observer effect they tend to maintain one particular state. For example light (photon) can exhibit both particle as well as wave nature which can be easily observed by double slit experiment (wave nature) and photoelectric effect (particle nature).

Quantum computers rely on the ability for quantum particles to exist in a superposition of multiple states at once to perform calculations. Because quantum computer harness the ability of quantum particles they use the property of superposition to build a true random number generator.

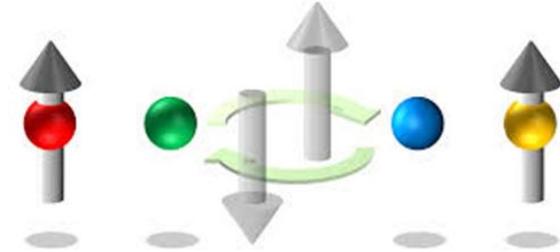
### IV. QUBITS AND QUANTUM GATES

The computers that we are using today uses binary values to work around which are basically [0,1], quantum computers on the other hand uses the same property of 1s and 0s but they physically implement these states by using properties of subatomic particles, for example spin of a particle, it can be either up or down, but again its in superposition that is both up and down therefor as soon as we observe it, it becomes up or down.

Therefore a bit in quantum computer is super positioned, that is both 1 and 0 until we observe it and this kind of bit is can be obtained via qubits.

A Qubit can have infinite value, it can be 1 or 0 or any combination of 1 and 0 with various probabilities.

This resolves our problem of predictive nature, because the value of a qubit is not known until it comes under the observer effect.



Now to manipulate binary data our computers have binary gates or logical gates but to work around qubits we need special kind of gates known as quantum gates. They are able to do whatever a logical gate can do including other awesome stuff. For example Hadamard gate pushes a qubit into a superposition state and amongst the fundamental gates in the domain regarding quantum gates.

### V. QUANTUM GATES

Like any other gates quantum gates are building blocks of the world of quantum circuits. They are reversible in nature unlike the normal gates in the discrete computers that we use in our day to day lives.

They are represented by unitary matrix, which simply means that the conjugate transpose of the matrix is also the inverse of the matrix:[1]

$$U^*U = UU^* = I,$$

Where  $U^*$  is the conjugate transpose.

The number of qubits in the input and output of the gate must be equal. The number of qubits in the input and output of the gate must be equal; a gate which acts on n qubits in the input and output of the gate must be equal.

The vector representation of a single qubit is: [2]

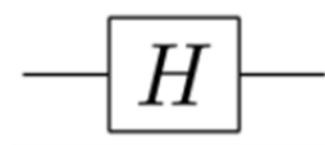
$$|a\rangle = v_0|0\rangle + v_1|1\rangle \rightarrow \begin{bmatrix} v_0 \\ v_1 \end{bmatrix},$$

Some of the quantum gates are given below for those who are interested are pauli gate, hadamard gate etc, the following table tells the same.

Operator	Gate(s)	Matrix
Pauli-X (X)	 $\oplus$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Since we are going to be utilising the quantum gates so lets look into a little into them including the famous Hadamard gate..

Hadamard (H) gate:



It basically works on a single qubit at a time and do the mapping in the following manner [2]

$$|0\rangle \text{ to } \frac{|0\rangle + |1\rangle}{\sqrt{2}} \text{ and } |1\rangle \text{ to } \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

It is simply conveying that the possibilities or the probability of the occurrence of 1 or 0 are 0.5 that is they are equally likely in nature.

The hadamard matrix has the following representation,

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

It is also on of the example of one-qubit, which comes under the quantum Fourier transform.[2]

Pauli-X gate



It also works on a single qubit at a time and is kind of similar to what we see in the not gate.[2]

in the sense that a measurement of the eigenvalue +1 corresponds to classical 1/true and -1 to 0/false).

The matrix has the following representation

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Pauli-Y gate

It equates to a rotation around the Y-axis of the Bloch sphere by pi radians.[2]

The matrix representation is following.

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

Pauli-Z gate

It equates to a rotation around the Z-axis of the Bloch sphere by pi radians.[2]

The matrix representation is following.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

There are other gates too, which are mentioned in the table already given, but to generate a random number they don't play as such a vital role as hadamard gate.

## VI. HOW TO GENERATE RANDOM NUMBERS USING QUBIT AND QUANTUM GATES IN A NUTSHELL

In order to generate a true random number from a quantum computer all we need to do is:

- Pass a qubit from one of the quantum gate to get a predefined state.
- Now your qubit either have 1 or 0 as value.
- Pass that qubit via a Hadamard gate.
- Now check the state of the qubit (that is observer effect).
- Whatever the final state is, that is 0 or 1 has the equal probability to occur.

The process of building applications has been a journey and it varies depending on one's application requirements and purpose [2]. Some companies even give us the opportunity to use their very quantum computers (with the help of cloud) for public uses. One of the famous one is IBM Q Experience, currently it only allow us to work on a 5 qubit processor, but since we are only concerned about random number it fulfills the drill.

## VI. CONCLUSION

As we know every number can be a combination of binary values [0,1] and we just generated a random bit so the resulting combination of bits will also be truly random. Until quantum computer becomes readily available we can surely go for cloud base services like IBM Q Experience by using its api to get our very own true random number. Generating a random number via quantum computer is just the tip of the iceberg, we can go for fancy sorting algorithms like BOGO sort which are really hard(impossible) to implement in our today's day to day devices, this sorts an array in just a worst complexity of O(n) and best case complexity of O(1).

Quantum computers are the future, but with the emergence of these strong computers our high tech world will travel to a new phase as discrete devices will be of no use, due to many reasons like poor security which a quantum pc can easily break into. So yeah, it won't be wrong to say that quantum computers are not only there to generate random numbers but also they will lead to a world which is truly random, which can't be predicted for the time being.

## REFERENCES

- [1] Varyani Y. et al. A Survey on Cryptography, Encryption and Compression Techniques, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11 | Nov 2019.
- [2] Kharb, L. (2018, January). A Perspective View on Commercialization of Cognitive Computing. In 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 829-832). IEEE.