A SURVEY ON QUANTUM COMPUTING THE GAME CHANGER IN GAMING INDUSTRY - ILLUMING USING QUBITS

V. Prasanna Sree Hari, D. Shanmuga Priyaa*

ABSTRACT

Illuming a virtual world of gaming as an illusion or mirage. Illuming is a prototype that acts as a device replicating the images and creating an object right opposite, which breaks the barrier between the Real and Virtual worlds. This leads to the pathway in transforming the industry from Classical Computing to Quantum Computing using Qubits in the processing unit. Using the power of quantum computing channelizes the processing power to the Quantum algorithms to process the inputs to have a variety of outputs rather than having one descried outcome which is part of classical computing.

Modifying the current Figure 1. CPU, GPU, FPGA, and ASIC processing to route it through quantum algorithms which helps us to get the realities outcome that is equivalent to the physical world. This helps us in deriving mixed reality using the current processor.

Keywords: qubit, GPU, Quantum computing, illusion.

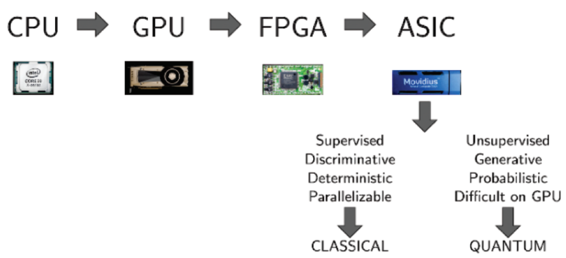

Figure 1. Variety of Processing Unit

I.INTRODUCTION

Computation using the current methodologies will not replace the modern devices, but it will leverage the power of quantum physics which can solve complex problems with a multi-dimension which takes more compute space and is difficult to achieve in current computers. Computing- this is complex computation using the quantum fundamentals that can be performed in today's latest computers which are called Quantum Computing.[1]

II. QUBITS CONTRA BITS

Classicistic computers are prioritized to compute calculations using 0's and 1's called bits as data. Figure 2. Qubits are the units that are used to compute Quantum computers, which can represent a combination of both 0's and 1's with an overlaid state called superposition.[2]

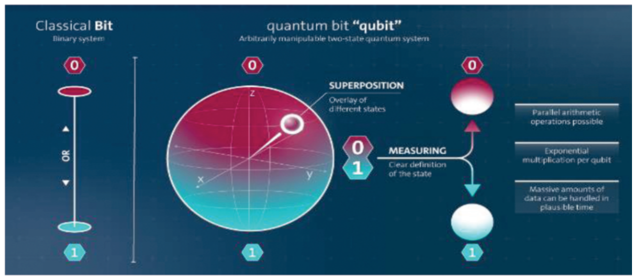

Figure 2. Qubit vs Bits

Superposition is an estimating unit that gives quantum PCs the possibility to be dramatically quicker than the present centralized computers and workers. Quantum PCs can do various computations with different sources of information with the inputs concurrently, which leads to the following

a. Parallel arithmetic operation

b. Exponential multiplication per qubit

c. It can handle massive amount of data at a time.[3]

Current PCs can deal with just one bunch of sources of info and each computation in one turn. They only process sequential processes. Computing with a specific number of quantum bits (qubits) suppose n (number of) for our model a quantum PC can direct computations on up to 2n (number of) input in a single relay which leads to parallel processing and having multiple inputs at a time.

That makes us clear when you deep dive deeper into the concepts and functionality of how a quantum PC works? Quantum computer's potential solves many challenging problems that are yet to be solved in the current world. [4].

Department of Computer Science,
Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India
*Corresponding Author

III. QUANTUM COMPUTING VERSUS CLASSICAL COMPUTING

Bits vs Qubits

Figure 3. Classical Computing computes the mathematical algorithm using Bits. It has two states at a point in time either 0 or 1. At a point, it can be only in one state.

Figure 4. Quantum Computing computes the mathematical algorithm using Qubits. Qubits have a combination of states it can hold 0 and 1 at the same time, which is called superposition. [7]

The pitfall of bits comes when we wanted to compute a problem with multiple variables in classical computers. In this case, the computer must conduct redundant calculations whenever the variable changes. Each calculation flows in an individual path as there is a change in the variable.

The advantage of qubits is to compute the same problem with multiple variables using Quantum computers. In this case, the computer can process computation parallelly and need not go back on calculating the operation when there is a change in the variable. These need not go the complete path rather it provides the flexibility and parallelism using Quantum computers. They can investigate in a various number of ways at the same time, which is the thing that gives quantum PCs the possibility to work much rapidly [5][6]. They convey different outcomes in a tight reach, finding you nearer to the solution far quicker than traditional PCs can.

Combining both the Classical and Quantum computing leads to the pathway of Hybrid approach, which process multivariable problem to be solved by 2020. For instance, by using n-quantum PC's (nascent quantum pc) to narrow the range of possible solutions to a financial, supply chain or logistics problem, a company might reach the optimal solution up to max 10% faster.

This quantum computing expels enough to deliver massive barrier in areas such as gaming, cryptocurrency, cyber security, data science, drug development, cryptography, big data, scientific calculation, and so on.

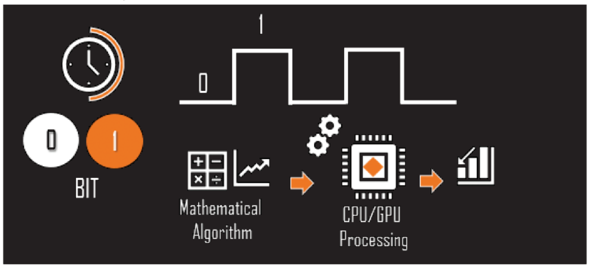

Figure 3. Bit Computation

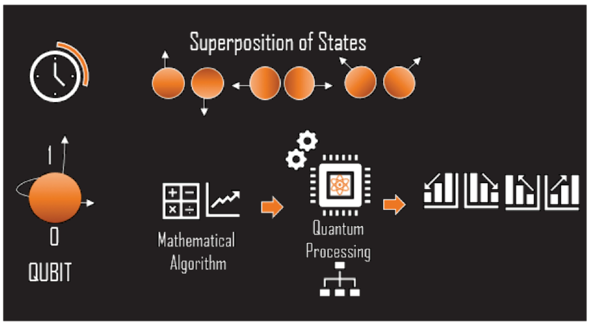

Figure 4. Qubit Computation

Quantum algorithms create a migration and a transformation on vector space. These algorithms transfer the compute logic using the axioms of quantum mechanics.

IV. THE QUBIT

The quantum bit is said to be a qubit, which has the major unit utilized in quantum PCs. Quantum mechanical speculation of a bit is utilized in classicistic PCs. Quantum bit (qubit) is a 2-Dimensional quantum framework. The condition of a quantum bit (qubit) can be communicated as,

$$\lambda = \zeta\, 0 + \eta\, 1. \qquad (1)$$

Here ζ and η are intricate number such that,

$\zeta^2 + \eta^2 = 1$. In the k-documentation(ket) or the D-documentation (Dirac), the shorthands for the vectors are encoded in two conditions of a 2-Dimensioal vector space. So as indicated by this documentation, Eq. (1) communicates the way that the condition of the quantum bit (qubit) is the 2-Dimensional complex vector. In contrast to a traditional bit, the 0 and 1 conditions of a quantum bit (qubit) can't be estimated without evolving it. Estimating a quantum bit (qubit), whose state given by Eq. (1), will yield the traditional worth of one or the other zero 0 with likelihood ζ2 or one 1 with likelihood η2. Quantum bit (qubit) executions and innovations are an extremely dynamic space of exploration that isn't the focal point of our audit. [15,16]

V. SYSTEM OF QUBITS

Quantum bit (qubit) generalizes to perform mathematical structure for multi-dimensional quantum systems. Norm one is a state of any quantum system in a complex vector space that is Normalized. The sum of all outcomes is the probability of all the outcomes which is measured as a sum of one when normalized [11].

Quantum computers must understand how to build qubits with state of a system combined to a given state of the individual quantum bit (qubits). Operation for a joint state of a system of quantum bits (qubits) is described as the tensor product, ⊗. Calculating the operation of two joint states is the same as taking the K-product (Kronecker product) of their corresponding vectors. Say we have two single-quantum bit states $\lambda = \begin{pmatrix} \zeta \\ \eta \end{pmatrix}$ and $\lambda' = \begin{pmatrix} \zeta' \\ \eta' \end{pmatrix}$.Complete system state is composed of two individual quantum bits (qubits) is given by,

$$\lambda \otimes \lambda' = \begin{pmatrix} \zeta \\ \eta \end{pmatrix} \otimes \begin{pmatrix} \zeta' \\ \eta' \end{pmatrix} = \begin{pmatrix} \zeta\zeta' \\ \zeta\eta' \\ \eta\zeta' \\ \eta\eta' \end{pmatrix} \qquad (2)$$

To reduce clutter in a tensor product sometimes the ⊗symbol is denoted altogether. States of a single ket. For example, λλ′ is expanded as λ ⊗ λ′, and 000 is expanded as 0⊗0⊗0. A brief way to calculate T-product (Tensor product) using the distributive property of the K-product (Kronecker product) for larger systems is called D-notation (Dirac notation). For a system of, say, three quantum bits (qubits) with each quantum bit (qubit) in the state.[9]

$\gamma i = \zeta i\ 0 + \eta i\ 1$, for i = 1, 2, 3, the joint state is,

$$\gamma 1\gamma 2\gamma 3 = \gamma 1 \otimes \gamma 2 \otimes \gamma 3 = \qquad (3)$$

$$\zeta 1\zeta 2\zeta 3\ 000 + \zeta 1\zeta 2\eta 3\ 001 + \zeta 1\eta 2\zeta 3\ 010 + \zeta 1\eta 2\eta 3\ 011 + \eta 1\zeta 2\zeta 3\ 100 + \eta 1\zeta 2\eta 3\ 101 + \eta 1\eta 2\zeta 3\ 110 + \eta 1\eta 2\eta 3\ 111 \qquad (4)$$

(2^3) possible bit associated with the 8 basis vectors which is a measurement of all three quantum bits(qubits). 2n is the exponential growth of quantum bits (qubits) in the dimension of the state space

VI. SUPERPOSITION AND ENTANGLEMENT

The linear combination of 2 quantum states when normalized to a valid quantum state is called Superposition [10][12]. A Linear combination of a few basis states is expressed as an upshot of any quantum state. For instance, i.e., in Eq. (1) that all condition of a quantum bit (qubit) can be communicated as a direct blend of 0 and 1. Additionally, the condition of any n qubit framework can be composed as a standardized straight mix of the 2n bit-string (states shaped by the tensor results of 0's and 1's). The liner independent (orthonormal) premise shaped by the 2n bit-string states is known as the calculated premise. The T (tensor) result of three diverse single-quantum bit states is depicted by Eq. (3) of the complete three quantum bits(qubits) state.[13]. Yet, it is workable for three quantum bits (qubits) to be in an express that can't be composed as the T (tensor) result of three single-quantum bit states. An illustration of such a state is,

$$\Psi = 1/\sqrt{2}\,(000) + (111) \qquad (5)$$

Ensnarement states are the framework that can't be communicated as a tensor result of conditions of its singular subsystems. For an arrangement of n qubits, this implies that a snared state can't be composed of a tensor result of n single quantum bit(qubit) states. In general, quantum information processing and quantum computing leads to the existence of entangled states perform an important consequence is a physical fact. This makes quantum PCs to be more remarkable than old-style PCs [17] . Trap makes it conceivable to make a total 2^{nD} complex vector space to do our calculations in, utilizing just n actual quantum bits (qubits). [1,8,10]

VII. QUANTUM CIRCUITS

Circuits are a diagrammatic representation of Quantum algorithms. To design and read quantum route horizontal Lines are the representation of qubits. Gates Circuits are then drawn on the quantum bit (qubit), [2]. left to right is the sequence of execution. Qubit lines are denoted as the initial state from the beginning of each qubit. A numerical articulation for the gate circuit, the entryways are recorded from right to left in the request for any application. These standards are good shown by a model. Given in Figure. 5 is a gate circuit to set up an ensnared 2-quantum bit (qubit) state called a B-state (bell State) from 00.

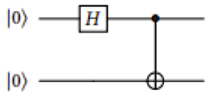

Figure 5. Quantum route for producing a Bell state

The route encodes to formulate,

C-NOT12 $(H \otimes I)$ 00 = 1 $\sqrt{2}$ (00 + 11).

Let us presently cautiously go over how the gate circuit creates the B-state (Bell State). The circuit are perused passed on to right. The quantum bits(qubits) are mathematically marked beginning from the top. First, the H door follows up on the topmost quantum bit (qubit) converting the condition of the framework to,

$H \otimes I\ 00 = (H\ 0) \otimes (I\ 0) = \left(\frac{0\rangle + 1\rangle}{\sqrt{2}}\right) \otimes 0\rangle$ $=\frac{1}{\sqrt{2}}(00 + 10)$. C-NOT12 acts on both quantum bits (qubits). The first quantum bit (qubit) implies that this quantum bit (qubit) is the control quantum bit (qubit) for the CNOT. The $\oplus$ symbol on the second quantum bit (qubit) implies that this quantum bit (qubit) is the target of the NOT gate (controlled by the state of the first quantum bit (qubit)). The action of the CNOT then gives,

$C\text{-}NOT_{12} \left(\frac{1}{\sqrt{2}}(00 + 10)\right) = \frac{1}{\sqrt{2}}$ $(C\text{-}NOT_{12}\ 00 + C\text{-}NOT_{12}\ 10) = \frac{1}{\sqrt{2}}(00\rangle + 11\rangle)$).

The special gate is an estimation unit of a quantum bit (qubit) with a meter image on it, given in Figure 6. The presence of this entryway on a quantum bit (qubit) implies that the quantum bit (qubit) should be estimated in the computational premise.

Figure 6. The Calculation of gate

VIII. QUANTUM ALGORITHMS

Every one of the essential components is required for the conversation of viable quantum calculations. A quantum calculation comprises of three fundamental stages:

• The state of a set of input qubits is encoded data for classical or quantum computing.

• This set of input quantum bits(qubits) are an arrangement of quantum gates.

• Classically interpretable results are obtained to measurements one or more qubits. [2,3,4]

IX. GPU COMPUTING

The high memory bandwidth and estimation power, that is adoptable for many applications as a multi cores (large number of processors) are available for the GPU to perform parallel architecture.[14] All resources of computers available such as memory space, grids, and threads with synchronization barriers are available and help the programmers to efficiently program it using the CUDA model. GPU's internal components run on C/C++ language. The mode of coding is done with the real code and the machine code. The host code, executing on a CPU, consists of ignorant and mostly subsequent estimation loads.[14]. GPU and basic pre-processing phase run on a certain structure. The allocated code runs on the GPU itself, representing the coordinating portion of the associated issues. Apart from supporting C/C++ programming language CUDA also supports other higher-end languages and their supported library functions. In this work, the augmentation for P-language (Python), named PythonCuda (PyCuda) [10], was picked over a lower-level, better-execution language because of the accompanying reasons:

• In python due to its fewer coding stipulations, modeling is done very easily.

• The VPE-qGM environment is developed in Python and therefore an extension of its execution library to support GPUs execution is easily obtained.

• The host-code is comprised of methods for the creation of the basic structures that later are copied to the GPU. Such creation is based on string formatting and manipulation of multidimensional structures, which can be easily prototyped with Python. On the other hand, the device-code perform a more restrictive and intensive computation, which is implemented in the C language as a regular CUDA kernel. By using the features of PyCuda, the technical challenges of the development process are reduced, and greater attention is given to the algorithmic problem. A basic PyCuda workflow is shown in Figure 7. Binary executables are obtained from a C-like CUDA source code (CUDA kernel) generated from PyCuda as a string, allowing run-time code generation. The kernel is also compiled during runtime and stored in a semi-permanent cache for future reuse if the source code is not modified. [5,6,7]

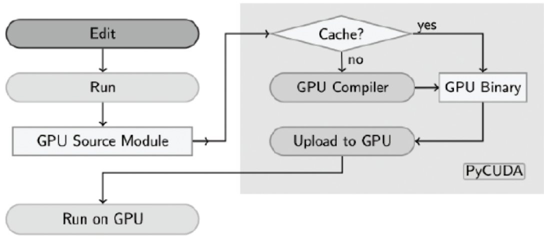

Figure 7. Basic PyCuda compilation workflow

X. TECHNICAL OBSTACLES

Hindrance is technical and Qubits are buoyant. One or Zero states are processing units for every bit in today's computers. During the computation, the one bit will not interfere with the other bit, but it ensures the process on a sequence. One and Zero are states of Qubits which is a combination and will interact with qubits to do multiple calculations at once.

Controlling these communications, in any case, is exceptionally muddled. Information sources are lost or modified because of the unpredictable idea of quantum bits (qubits), which is mistake inclined. To build a multi-exchange, making a PC of significant scale would require many thousands or millions of qubits to be associated reasonably. The couple of quantum PCs that exist today can deal with not even close to that number.

Google, IBM, and Microsoft are trying to overcome these obstacles in their research labs to build both software and hardware. These organizations are building calculations to utilize the current equipment to perform quantum processing, which is far to go, equipment that might well end up appearing to be extremely unique from the present dim boxes, and programming to assist with making an interpretation of existing information into a qubit-prepared configuration. Even though quantum processing as an idea has been around since the mid-1980s, the principal genuine proof that quantum PCs can deal with issues excessively confounded for old-style PCs happened uniquely in late 2019, when Google declared that its quantum PC had addressed such a computation in only 200 seconds. They are a numerical exercise than whatever might be applied to business.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information. Cambridge University Press, 2000.

[2] L. Grover, "A fast quantum mechanical algorithm for database search," Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219, 1996.

[3] P. Shor, "Polynomial-time algoritms for prime factorization and discrete logarithms on a quantum computer," SIAM Journal on Computing, 1997.

[4] K. D. Raedt, K. Michielsen, H. D. Raedt, B. Trieu, G. Arnold, M. Richter, T. Lippert, H. Watanabe, and N. Ito, "Massive parallel quantum computer simulator," http://arxiv.org/abs/quant-ph/0608239, 2006.

[5] J. Niwa, K. Matsumoto, and H. Imai, "General-purpose parallel simulator for quantum computing," Lecture Notes in Computer Science, vol. 2509, pp. 230–249, 2008.

[6] A. Maron, A. Pinheiro, R. Reiser, and M. Pilla, "Distributed quantum simulation on the VPE-qGM," in WSCAD-SSC 2010. IEEE Computer Society - Conference Publishing Services, 2010, pp. 128–135.

[7] NVIDIA, NVIDIA CUDA C Programming Guide. NVIDIA Corp., 2012, version 4.2.

[8] R. Reiser and R. Amaral, "The quantum states space in the qGM model," in Proc. of the III Workshop-School on Quantum Computation and Quantum Information. Petr´opolis/RJ: LNCC Publisher, 2010, pp. 92–101.

[9] A. Maron, R. Reiser, M. Pilla, and A. Yamin, "Quantum processes: A new approach for interpretation of quantum gates in the VPE-qGM environment," in Proceedings of WECIQ 2012, 2012, pp. 80–89.

[10] ibmq-device-information. https://github.com/ Qiskit/ibmq-device-information/ tree/ master/ backends/ tenerife/V1. Accessed: 14-12-2019.

[11] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In Ryan O'Donnell, editor, 32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia, volume 79 of LIPIcs, pages 22:1–22:67. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[12] Héctor Abraham, Ismail Yunus Akhalwaya, Gadi Aleksandrowicz ...., and yotamvakninibm. Qiskit: An open-source framework for quantum computing, 2019.

[13] A. Ambainis, H. Buhrman, P. Høyer, M. Karpinski, and P. Kurur. Quantum matrix verification. 2002.

[14] Andris Ambainis. Quantum walk algorithm for element distinctness. SIAM Journal on Computing, 37(1):210–239, 2007.

[15] Andris Ambainis and R. Spalec. Quantum algorithms for matching and network flows. in Lecture Notes in Computer Science: STACS 2006, 3884, 2006.

[16] Itai Arad and Zeph Landau. Quantum computation and the evaluation of tensor networks. SIAM Journal on Computing, 39(7):3089–3121, 2010.

[17] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. SIAM journal on Computing, 26(5):1510–1523, 1997.