IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 11, November 2021

DOI: 10.17148/IARJSET.2021.81141

Quantum Walks via Quantum Cellular Automata for Simulating Quantum Computation

Tanish Singh[1]

Mansukhbhai Kothari National School, Pune

Abstract: The quantum walk is a subset of classical random walks. Considerable research has lately been conducted on quantum walks in relation to quantum computing. The walk may be thought of as a type of quantum cellular automata. In a broader sense, there is a one-to-one link between them. Furthermore, we look at another type of quantum cellular automata that may be thought of as a quantum version of stochastic cellular automata. The quantum walk formalism is a widely used and quite practical paradigm for modelling quantum systems such as Dirac equation simulations, discrete dynamics in both the low and high energy regimes, and the development of a multitude of quantum algorithms. The walker in a QW moves in a quantum superposition of pathways, and the ensuing interference serves as the foundation for a broad range of quantum algorithms, including quantum search, graph isomorphism issues, and rating nodes in a network.

To generate a block representation of n-dimensional quantum cellular automata, we apply the following general result to them. As a result, we have demonstrated that their generic, axiomatic formulation nevertheless produces a unifying, operational representation of them.

Some non-trivial ramifications are investigated, such as the fact that bijective non-reversible CA are not physical as closed systems, or that quantum information can move faster than classical information — within some specified dynamics. Finally, we present n-dimensional quantum cellular automata that can simulate anything else. This means that the initial configuration and local transition rule of any one-dimensional QCA may be stored within the universal QCA's initial configuration. The simulated QCA will then correspond to many phases of the universal QCA. In the sense that each cell in the simulated QCA is represented as a group of nearby cells in the universal QCA, the simulation preserves the topology. Because the encoding is linear, it bears no cost of the calculation.

Keywords: Physics, Quantum simulation, QCA, Quantum Walks

I.INTRODUCTION

Random walks, a sort of stochastic process, were first introduced at the turn of the twentieth century [1]. Since then, random walks have evolved into a powerful tool with applications in a wide range of fields. as in economics [2, 3], ecology [4, 5], and computer science [6] as well as physics [5, 6]. Quantum mechanics was created approximately at the same time. However, it took over 90 minutes. Years have passed since the quantum form of random walks, now known as quantum walks (QWs), was presented [7]. Despite its youth, the quantum equivalent of random walks is being used in a variety of applications, particularly in quantum simulations [8, 9] as well as in quantum algorithms [10]. The development of the cell at point x is described in two phases in traditional cellular automata. First, the surrounding cells of x's state are read. Second, based on the initial read measurement, the state of cell x is modified using a defined local rule. This is then performed for all of the cells in the automata in a translation invariant manner, that is, using the same updating procedure for all cells and without regard for updating ordering. The nearby scheme N ensures the localization of the automata by stating that only cells in the set Nx = N +x will impact the update of the cell at position x.

II. QUANTUM CELLULAR AUTOMATA: FORMAL DEFINITION

The development of the cell at point x is described in two phases in traditional cellular automata. First, the surrounding cells of x's state are read. Secondly, based on this initial read measurement, the state of cell x is modified in accordance with a predefined local rule. This is then performed for all of the cells in the automata in a translation invariant manner, that is, using the same updating procedure for all cells and without regard for updating ordering. The nearby scheme N ensures the localization of the automata by stating that only cells in the set Nx = N +x will impact the update of the cell at position x. Thirdly, the difficulties in correctly defining a quantum cellular automata stem from the difficulty of cloning quantum states [11]. Aside from that, classical automaton dynamics may be irreversible, in contrast to unitary quantum development. The Margolus block scheme [12] is one method for defining reversible automata. Partitioned Unitary Quantum Cellular Automata is defined here by merging the Margolus block scheme, the unitary features of the QCA

This work is licensed under a Creative Commons Attribution 4.0 International License

proposed by Pérez-Delgado and Donny Cheung [13], and certain dynamical qualities existing in the QCA launched by Watrous [14]. This combination demonstrates the qualities of locality and translation invariance.

A cellular automaton, in its most basic form, is a lattice of cells, each of which is in one of a finite set of states at any one time. The state of each and every cell cell is updated according to some local transition function at each discrete time step. The current state of the associated cell, as well as the states of the cells in a finite sized neighbourhood around this cell, are the inputs to this function. Figures 1.1 and 1.2 depict a basic cellular automaton from [15]. This CA's lattice is the set of integers Z, implying that the CA is one-dimensional. Each cell's neighbourhood consists of the cell itself as well as its two nearest neighbours, one to each side. . The cells, represented by boxes, have two possible states, in this case represented by the box being either black or white. Figure 1.1 shows a pictorial representation of the CA transition function, as presented in [15]. Figure 1.2 gives a pictorial description of the automaton's evolution in time

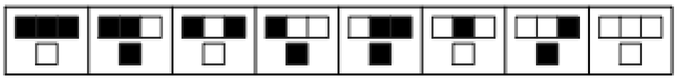

Fig 1.1: The first row depicts the cell's current condition as well as the states of its nearest neighbours on each side. This is the transition function's input. There are two conceivable states: white and black. The second row provides the transition function's output, or the state of the cell after the function has been applied. Reading from the left, if the cell is in the state black and both of its neighbours are black, the cell will be coloured white in the following time step. If, on the other hand, the right neighbour is white, the cell will remain black, and so on [15].

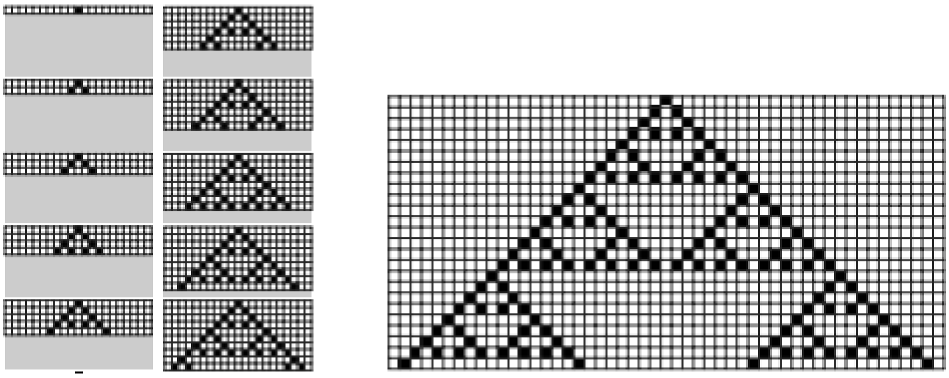

Fig 1.2: The diagrams depict the temporal development of the CA specified in Figure 1.1 from top to bottom, left to right. The original structure, depicted in the top left figure, consists of a single black cell surrounded by white cells. Time moves downhill. Each succeeding image illustrates the automaton's present condition as well as all phases since the first [15].

III. QUANTUM COMPUTERS--UNIVERSAL QUANTUM SIMULATORS

The difficulty of finding the associated time evolution operator U(t) = exp (iHt) given a Hamiltonian H that describes the physical system under consideration is comparable to the job of implementing a well specified unitary matrix. A quantum computer with a universal set of quantum gates may perform any unitary transformation, but not always in an efficient number of elementary operations [16]. Lloyd really showed that universal quantum computers can effectively calculate U(t) when H is a sum of local terms. When L = poly(N), as in the case of nearest neighbors interactions, this is polynomial in N. Cookbook for Quantum Simulations we will now suppose that we are working with a universal quantum computer, which is depicted in the usual circuit model as a digital device, i.e. qubit-based, obeying the algebra of Pauli matrices, and employing a universal set of quantum gates [16]. The quantum simulation issue may thus be defined and solved on such a computer by following a few simple steps, which we will detail below. To begin, create a model Hamiltonian of interest H. This should include all of the dynamical information required to describe and characterize the physical quantum system under study. In the mathematical structure of H, the best suited collection of variables and operators will appear. Second, translate the target Hamiltonian H onto its Pauli algebra form.

$H \rightarrow H(\{\sigma_a\})$

In layman's terms, this entails determining a proper encoding of the target system's degrees of freedom into a number N of qubits. The mapped Hamiltonian H that results will then be represented in terms of Pauli matrices. This mapping is simple for physical systems composed of collections of spin-1/2 objects since they also obey Pauli algebra, but it is theoretically achievable for a broad class of physical systems, as will be demonstrated in the following with a specific example. When such H is the sum of local terms, the quantum simulation will be efficient. In many practical applications, this is not a barrier because most physical processes are fundamentally local in nature. Local Hamiltonian models, on the other hand, are mapped into non-local ones. These models might be efficiently emulated on quantum hardware with multi-qubit gates, as will be discussed more below.

The overall quantum circuit encoding time evolution will be the n-times repeated conjunction of all the sequences corresponding to the components in the ST decomposition. Finally, include initial state preparation at the start of the circuit and a suitable set of measurements at the conclusion to retrieve expected values of the key observable quantities on the developed quantum state. The above items constitute a rather broad set of instructions for developing a quantum simulation method. In the following sections, we will provide clear examples to demonstrate how this is done in practice. Of fact, such approaches are not confined to simulations of real-world physical systems, but may be used to a broader range of computing tasks anytime the problem at hand can be captured in Hamiltonian quantum dynamics.

IV. CONSTRUCTION OF THE AUTOMATION

Examining a Torus, a two-dimensional lattice of size 2 s x 2 r with periodic border conditions with a qubit is stored at each lattice location. A Margolus partitioning is used to define the QCA transition rule [12]. The lattice is divided into cells of size 2 x 2 and there is a different partitioning for the time t being even or odd. A coordinate system on the torus with axes parallel to the boundaries before identification can be chosen. The qubits in the upper left corner of each cell have both coordinates even for t even, and both coordinates odd for t odd. As a result, a cell in step t overlaps with four cells in step t - 1. The QCA transitions from time t to time t+1 by simultaneously applying a unitary transition function τ to each cell. With the described QCA and a suitable 4-qubit unitary τ, one can perform universal quantum computation. A quantum logic network of local and next-neighbor unitary gates with width 2s and depth r, in particular, can be simulated.

$$\tau = S(1, 3)\ S(2, 4)\ H1 \exp\left(-i\pi/8\ [1- Z3]/2 * Z1\right)$$

$$\exp\ \left(i\pi\ (1 - Z4)/2\ (1 - Z1)/2\ (1 - Z2)/2\right.$$

S(a, b) represents a SWAP-gate between qubits a and b, H 1 represents a Hadamard transformation on qubit 1, and Z c represents a Pauli phase flip operator applied to qubit c. If |p34 is a computational basis state; it effectively stores two classical bits, p(3) and p(4). The transition function is then equivalent to a classically controlled unitary operation U (p) performed on |D12. The bit p(4) causes a (Z) gate to be applied to |D12, and the bit p(3) causes a (4-phase gate exp(i/8 Z_1) to be applied to qubit 1 of |D 12. In this way, |p encodes a basic programme step, which is carried out on the two-qubit "data" |D12. The SWAP gates allow quantum data and programme to pass through one another, allowing |D12 to interact with subsequent programme registers.

Columns are groups of qubits on the torus that have the same first coordinate x, 0 ≤ x ≤ 2 r − 1. All even columns contain data registers at time t = 0 and all odd columns are filled with programme registers |pj . The automaton's initial state is

$$|\psi(0)\rangle = \bigotimes_{i=0}^{r-1} |D_i(0)\rangle_{2i} |p_{i+1}\rangle_{2i+1}.$$

The labels inside the kets here and in the following specify the state, while the ones outside specify the location of the support within the lattice, i.e., the column. For example, |D1(0)2 represents data register no.1 at time t = 0, which is located on column 2. Only |D0 is used out of all data registers; the others are auxiliary. When the QCA begins to run, the data registers move right (counter-clockwise, as seen from the top) and the programme registers move left (clockwise), one column at a time. When the data registers are passed, the programme registers |p control unitary transformations U(p) that operate on the data registers |D. On the quantum data register |D0, a programme given by the data p1,..., pr, with p1 encoding the first and pr encoding the last step, is carried out in this manner. The programme that is executed corresponds to a quantum logic network comprising local and next-neighbor gates in a certain configuration; see Fig. 2. As will be explored more below, such networks are adequate for universal quantum computing. The identical programme steps that are used on the register |D0 are used on the auxiliary data registers |Di,, 1 ≤ i ≤ r − 1, but in reverse order. As a result, these registers are not utilised.

During calculation, both data and programme traverse half of the torus. After the automaton has operated for r time steps, the calculation is complete, and the register |D0(r)ir may be read from column r using local measurements.

DOI: 10.17148/IARJSET.2021.81141

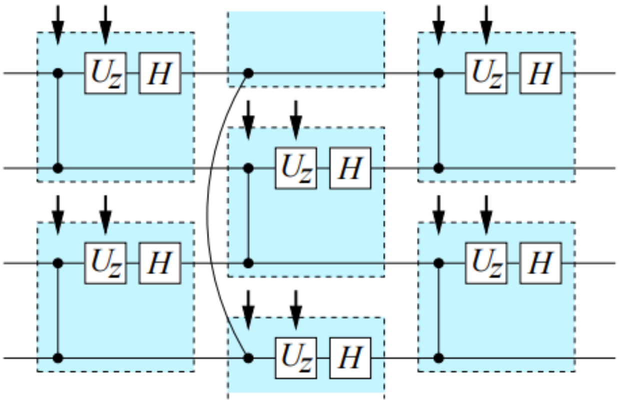

Fig 2: The QCA's simulation of the quantum logic network. The gates in each coloured box represent the result of one application of the elementary transition function. The arrows indicate classically controlled gates actuated by programme registers.

Let τ i,j indicate the cell's elementary transition function (1). (i, j). In this case, I is the column coordinate of the cell's top left qubit, and j is the appropriate point inside the column. The unitary transition function T_i is then applied to two consecutive columns i, i + 1.

$$T_i = \bigotimes_{j=0}^{s-1} \tau_{i,[2j+i]_{2s}}.$$

In this case, $[2j + i]_{2x}$ is a shorthand for 2j + I mod 2s, which will be utilised throughout the article.
If |pi is a computational basis state.
For I = 0 and t = r, in particular, one finds that is the output quantum register, with the unitaries U(p1)....U(pr) applied to the quantum register in its input state, |D0(0)i, in the right order. Before demonstrating the state |(t) of the QCA at time t, it allows us to retrieve certain previously stated QCA characteristics. The programme registers, other than being transferred, stay unaltered during the development, and there is no entanglement between columns. It is demonstrated that the quantum logic network simulated by the described QCA is, as claimed, universal. A universal gate set is formed from the CNOT-, Hadamard-, and the pi /4-phase gate exp (i/8 Z) [17]. By selecting the following p (3)-bit sequences, one may separately apply the Hadamard-, the/4-phase gate, and the identity to the simulated logical qubits for the given QCA.

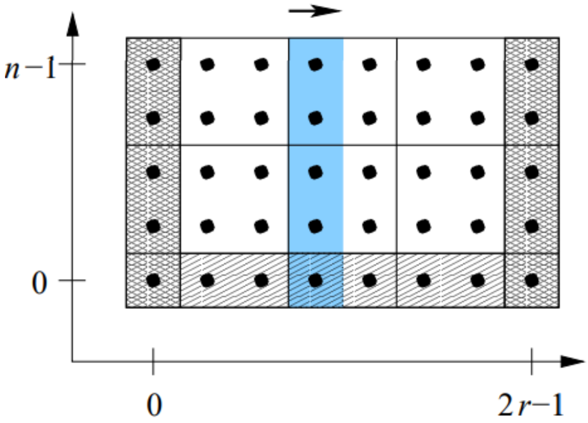

Fig 3: On a planar sheet, boundary requirements for the described QCA. As usual, the 2 x 2-cells depicted in white are operated upon by. A SWAP-gate is applied to the hatching 1 x 2-cells, and the identity operation is applied to the cross-hatched cells. The grey underlay column depicts a data register travelling to the right.

The disclosed QCA may be dubbed a deterministic programmable quantum gate array with some justification, although this name is already in use for a structure that has been demonstrated not to exist [18]. This finding is supported by our QCA. The programme information is traditional and applies throughout the whole curriculum[18]. Furthermore, in terms of temporal complexity, the presented QCA is as efficient as a quantum logic network with local and next-neighbor gates (within a constant).

The fact that the stated QCA exists on a torus simplifies the explanation but is not required. For modelling of the mentioned networks with r time steps and n qubits, a flat sheet of size n 2r suffices. The operations on the border that differ from are shown in Fig. 3. When this updated QCA is performed, the data registers continue to travel right and the programme registers continue to move left. However, the registers are mirrored on the left and right boundaries. As a result, mirrored programme registers on the left side of the lattice are treated as data by left-moving programme registers. Worse, on the right side of the cluster, reflected data registers function as programme on right moving data registers. As a result, the QCA state is no longer a tensor product of the column states, but rather an entangled state supported by multiple columns that increases from the right, one column at a time.

V. CONCLUSION

For universal quantum computation, I described a quantum cellular automaton. A four qubit unitary transformation working on Margolus cells of size 2 x 2 generates the transition function from one time step to the next. The programme is encoded in the system's initial state, and the automaton is let to evolve autonomously from inception until readout. Returning to our original query, it has been discovered that QCAs capable of executing difficult jobs in quantum information processing can be built for compact cells and adjacent schemes.I also introduced a partitioned quantum cellular automation, which is a well-formed quantum cellular automata as well as a theoretically simple and adaptable structure.

REFERENCES

[1] Frank Spitzer.-Principles of Randow Walk. Springer Science+Business Media-, LLC, second edition, 1976.
[2] Vijay Singal. Beyond the Randow Walk. Oxford, 2004.
[3] Robert-J.Shiller-and PierrePerron. Testing the random walk hypothesis: Power-versus frequency of observation. Economics Letters, 18:381–386, 1985.
[4] Francois Fouss, Alain Pirotte,-Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between-nodes of a graph with application to collaborative recommendation. IEE Transactions on Knowledge and Data Engineering,-19(3), March 2007.
[5] Fugao Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. 9 Physical Review Letters, 86:2050–2053, March-2001.
[6] Wiersma and-Diederik S.The physics and applications of random lasers. Nature Physics, 4:359–367, May 2008.
[7] Y. Aharonov, L. Davidovich,-and N. Zagury. Quantum random walks. Physical Review A, pages 1687–1690, August 1993.
[8] G Di Molfetta and A Pérez. Quantum walks-as simulators of neutrino oscillations in a vacuum and matter. New Journal of Physics, 18, October 2016.
[9] Pablo Arrighi, Stefano Facchini, and Marcelo Forets. Quantum walking in curved spacetime. Quantum-Information-Processing, 15:3467–3486, May 2016.
[10] Renato Portugal- Quantum Walks and Search Algorithms. Springer, 2013.
[11]Göran Lindblad. A general no-cloning-theorem. Letters in Mathematical Physics, 47:189–196, 1999.
[12] T. Toffoli-and N. Margolus. Cellular-Automata Machines. MIT Press Series in Scientific Computation, 1987.
[13] Carlos A., Perez Delgado, and Donny-Cheung. Local unitary quantum cellular automata. Physical Review A, 76(032320), 2007.
[14] J.Watrous.-On one-dimensional quantum cellular automata. In Proceedings of the 36th Symposium on Foundations of Computer Science, pages 528–537, 1995.
[15] S. Wolfram, A New Kind of Science.-Wolfram Media, 2002.
[16] M. A. Nielsen, I. L. Chuang, Quantum computation and quantum information, Cambridge University Press, Cambridge, UK 2000.
[17] P.O. Boykin et al., Inf. Proc. Lett. 75, 101 (2000).
[18] M.A. Nielsen and I.L. Chuang, Phys. Rev. Lett. 79, 321 (1997).

BIOGRAPHY

My Name is Tanish Singh, a Junior in High School at Mansukhbhai Kothari National School Pune and I'm currently planning to attend college and choose computer science as my major in undergraduate. I am a STEM enthusiast and entrepreneurial propensity. I am the owner and founder of a non-profit organisation called 'Forgo Plastic'. I am a recipient of a full scholarship for IBM's Quantum Computing Course 2021-2022 and also an ambassador at InspitiAI.