International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 6 Issue 2|| February 2017 || PP. 01-15

Relations of geometry and space -time to general-purpose quantum parallel computing and artificial intelligence

C. H. Wu
Department of Electrical and Computer Engineering, Missouri University of Science and Technology
301 W 16th St., Rolla, Missouri 65409 USA

Abstract: *Parallel computing execute the manipulations of spatial relations or the corresponding pattern transformations among the processors. A general-purpose quantum parallel computing scheme of this nature has been presented earlier. It is shown that the quantum computing can exist only through the employment of huge bundles addition-rule based one-dimension cellular automaton (CA) with the type of quantum processor such as the one that is presented. It is symbolic substitution rules that must be employed rather than the use of quantum logic gates and qubit concept is fundamentally flawed for general-purpose computing. The processor (US Patent 8,525,544) possesses four instructions on a two- bit cell and stores two one-bit data. The consequences of this scheme are that geometry, physics and quantum computing are intertwined and many different forms of computing existed are geometry-controlled through the spatial relations among the processors. Because the addition-rules in the CA impose a Euclidean Plate (EP) concept, the necessary wiring connections of the quantum processors in the CA must have three distinctive intrinsic lengths through a Euclidean geometry requirement. In this work, we discuss first the space-time relation for quantum parallel computing and its relation to the artificial intelligence and the irreversibility in computing. Then we show when the intrinsic wiring lengths are interchanged, it will transform the EP structure of CA into that of Non-Euclidean-Plate (NEP) structures. Addition rules are reduction processes from the CA rules and an addition operation generates a proper space-time block at CA. If an equal probability is imposed on those CA transition rules, then birth-and-death of space-time blocks performed by the processors can be achieved by a seed, which is Non-Euclidean computing in nature. The relation between geometry and physics is well-known as well as the relation between physics and computing. Here we show directly that geometry and quantum parallel computing are related and the relevant intrinsic wiring lengths of the processors are the critical concept. Thus "a new kind of science" from CA is originated from NEP. The spatial relations other than the EP concept are to be incorporated into the general-purpose computing for artificial intelligence.*

Keywords : *Cellular Automata*
Quantum parallel Computing
Non-Euclidean Computing
Symbolic Substitutions
Artificial Intelligence

I. Introduction

Physics, geometry and computing are three distinctive subjects as they may appear at first. But it is well-known that geometry and physics are intertwined. All our physical interpretations are based on a Euclidean geometry concept. A much deeper discussion of this relation with respect to the modern physics has been discussed by Yang recently [1]. At the same time, construction of general-purpose quantum parallel computers must utilize spatial relations, which can be both Euclidean and Non-Euclidean. This is the center part of discussion in this work. Similarly quantum parallel computing employs a few particular aspects of physics that are usable or suitable for the computing purpose. A physical process can be considered as a natural parallel computing process, as proposed by Fredkin and Zuse [2], and a standardized computing procedure is then selected, using many quantum processors, to measure up or to quantify the results. Similarly, Feynman proposed to use quantum computer more conveniently to simulate other quantum phenomena directly [3]. But a direct connection between general-purpose quantum parallel computing and geometry alone has yet to be clearly demonstrated. Indeed, physics, geometry and computing are in triangular relations (see Fig.1a) in such a way that changes between any pair can affect the other one or two pairs. In general, all forms of parallel computing are originated from the manipulations of spatial relations among processors resided in a cellular automata (CA) architecture [4-6]. The local behavior in the cell of CA is governed by the quantum processor used, while the long range behavior is governed by the interconnections among the processors. Those two behaviors resulted in a set of spatial relations, or the corresponding CA transition rules. For example, when addition rules of the computing are activated, the local and long range behaviors are imposed in a particular set of spatial relations

that must be Euclidean in nature. Thus all the wiring connecting lengths in between quantum computing processors and to the inside of the processor itself are determined by this computing equilibrium condition, which is no different than the condition for those chemical bonding lengths in forming a cluster of molecules. Between physics and geometry, the chemical bonding lengths are determined by imposing a minimum free energy concept of the system (see Fig. 1b). Between physics and geometry for quantum parallel computing, however, the interconnecting lengths between the processors are determined by imposing a Euclidean geometry concept. So the role of Euclidean geometry for parallel computing is the same as the role of the minimum free energy in physics for determining the system configurations. Since those large number of processors are space-filling, a Euclidean geometry must be used to trace the locations of all the processors where the computing results are resided. Otherwise there will be more energy (more efforts) used in the computing due to the extra tracings needed for the geometrical wiring relations. It is clear that our counting of numbers must rely on a Euclidean concept in order to place a sequence of successive numbers assigned. For example, an infinite straight line in Euclidean space with points equally spaced provides a minimum effort or energy to trace and read (or obtain) the number of a point that is marked on the line as the reference (see FIG.1a). Geometry and parallel computing, which employs many quantum processors, are intertwined at the elementary computing level of the addition operation for any two bit-strings, the first step of general-purpose computing. Since addition operation is shown to be performed on one-dimensional CA [4-5], this is similar to the Euclidean Plates (EP) concept recently discussed in the deformation resulting in the multi-scale structures or surfaces in materials. In materials, Non-Euclidean plates (NEP) are stacks of identical surfaces whose quasi-two dimensional intrinsic geometry is Non-Euclidean or those cannot be simply realized in a flat configuration [7]. The distinctions between EP and NEP are illustrated in Fig.2a and Fig.2b. In both cases, the middle plane preserves the length and thickness. To translate the above material system into the corresponding EP and NEP electrical systems, the electrical wiring lengths need to be introduced. Those are shown by conveniently separating the cells (or the size of the computing processors) alternatively and replacing with two connecting wires between cells. Those are shown in Fig.2c and Fig. 2d. For EP electrical system of Fig.2c, the two wires between the cells are equal while for NEP of Fig. 2d, one wire is longer, while the other is shorter than the length in Fig.2c.

It is clear that such EP system can exist only in a very singular situation and any deviations can be made by the combination of displacements and rotations with respect to the reference frame, such as in Fig.2e. Note that the scaling factor in Fig.2e with respect to the reference EP cannot be known *a priori*. Thus any alterations of the wiring lengths will result in having NEP systems, which are the most common occurrences. However, our interpretation of any physical phenomena observed, which relying on our corresponding computation processes, must be consistent with the rules of additions. Yet the rules of addition must be EP-based in CA. When NEP- based computations are performed, a 'new kind of science' [8-9] will emerge. Those non-physical results obtained are simply non-physical because they are viewed from the Euclidean concept of computations. In this work, it is shown that there are three intrinsic wiring lengths associated with the quantum processor in CA when geometry and parallel additions are considered alone. Before that argument, the space-time relations for parallel computing need to be discussed. Irreversibility of Euclidean computing and the conditions for artificial intelligence are presented in Section II. In Section III, we summarized the general-purpose quantum parallel computing scheme from a proper processor for the CA architecture. The fundamental flaws of qubit concept are addressed again [6]. In Section IV, classifications of Euclidean and Non-Euclidean computing through the addition-rule compatibility are presented. An addition operation generates a proper space-time block in CA. In Section V, the birth-and-death of space-time blocks in a Non-Euclidean computing is presented by an example of a structured Sierpinski Triangle created by a seed. A space-time block performed by some processors can be reproduced or eliminated and are the foundation for a true artificial intelligence. The conclusions are presented in Section VI.

II. On the relationship between space and time in parallel computing and the consequences to artificial intelligence and computing irreversibility

To understand parallel computing, which is distinctly different from the sequential computing of our current computer that is so familiar to us, the addition operation needs to be examined first by the interconnection scheme involved with all the processors. Anything that is computable in the Euclidean concept can be decomposed into algebraic operations [10], starting from addition operation, which is purely sequential in computing nature, even if a massive quantum parallel computer is used. That means the same number of "ripple-carry" steps for an addition operation is to be used for any computers and there is no "step-wise advantage" through a parallel computing scheme. But it is understood that each step does consume a different time to complete by different computers. Addition operation in the quantum parallel computing scheme that have been shown [4-5] is performed in a one-dimensional cellular automaton (CA) as shown Fig3a. Many bundles of CA chains connected may or may not form two-dimensional automata (also CA) for higher-order computations,

such as for multiple additions or multiplications and so on. When more CA chains are used simultaneously for parallel computing, more space are filled, since each processor occupies certain volume.

In the first glance, it may appear that the strategy of parallel computing is to trade space with time. Namely, more CA chains used will save the number of computing steps time-wise when the computations are at a higher order than just one addition operation. This trading is not linear, however. For example, when a "reclusive-doubling" method is applied for adding a string of N numbers, a single adder (which is a CA chain) would add one number after the other in N operations. But with N/2 number of CA chains connected in parallel through a "power of two" interconnection scheme [10], the time steps are reduced to $\log_2 N$ operations, giving a speed-up factor of $N / (\log_2 N)$. So N adders used does not speed up by a factor of N. But within a CA chain itself, however, it may appear as if the time is being traded equally with the space. Namely, two processors operated at one step will have the same result as one processor operated at two steps. But for parallel computing, it is important to note that space and time are actually not traded equally inside each CA chain. Unlike that space and time are considered on the same equal footings strictly in our physics concept (such as in special relativity), the spatial-relation information provided by a larger space at a given time is always superior than that from a smaller space with several more time steps, as far as the computing quality is concerned. In other words, starting at the most elementary level, any two adjacent quantum processors in a CA chain can perform much more in one step than one processor alone can perform in two consecutive time steps, just like two hands can tie shoe laces while one hand cannot in two or many tries. The consequence of this argument is that now, in physics, we must also think space can be a superior quantity than time because a physical phenomenon, which can be considered as a natural parallel computing itself , has to be interpreted through the use of some standardized computations. Then each computation itself is derived through the use of some physical phenomena, such as the utilization of the phase of electron wave function through a gauge invariance consideration [11]. This is similar to pick a proper standardized ruler to measure the distance between two points on a given surface, Euclidean or Non-Euclidean. Utilizing a standard quantum process to quantify other more complicated quantum phenomena is in the same concept [3]. This is the case of utilizing a man-made ruler to quantify against a natural ruler. So physics and computing are intertwined. The net result is that if space can be a superior quantity than time from a parallel computing consideration, then it can also be true in any physical process. That means space and time are not always on the same equal footings in physics as well as in parallel computing.

The superiority of space quantity in computing is measured by the total number of instructions that a N-processor system possessed in one time-step, which is equal to 4^N , the size of the entire computing table from N processors and a physical size that is N times to that of a single processor. The corresponding total number of instructions one processor operated repeatedly in N steps, which is just $4N$, and is occupied by a physical size of a single processor. Here all the processors are the same and each possesses four instructions or four symbolic substitution rules (Fig.4a). This means the half-adder processor contains the entire computing table of adding two bits. Note that in a corresponding Qubit scheme by others [12-21], the 4 instructions here is replaced by the concept of having four superposition of states by two qubits, which is fundamentally flawed for general-purpose computing [4-6] , even though the total number of states in superposition in N qubits is the same as that of the instructions possessed by the N/2 processors. The principle of constructing the processor is described briefly again in Section III.

Thus for parallel computing, space quantity is advantageously utilized through a logarithmical scale consideration while time quantity is poorly utilized through a linear scale, in terms of the number of instructions possessed. For example, for N=2, one processor with 4 instructions will possess a total of 8 instructions of capacity in two steps, while the corresponding two processors in one step will possess a total 16 instructions in a superior capacity. Now the addition table for any two two-bit-strings is contained in the 16-instruction set. Thus the moment a two-bit pair is entered as the inputs, the computing can pick the correct outcome configuration out of the 16 configurations. If just one processor is used, there are only 4 instructions available in the table and one is forced to use the table two consecutive times to get the same result. So here two-processors occupies twice the space to prepare all possible results first and then "anticipate" any possible input pairs that may be entered for a result. The bigger the prepared addition table means larger spatial relations or more processors are used. The result is that there is a difference of the qualities as compared to the one processor situation. Those spatial correlations (in the 16 sets of the two-bit addition table) cannot be duplicated in two time-steps because more information is contained in the two-bit table than that in the one-bit table that is used twice. That means spatial relation is more "valuable" (more information or more bit content) or more "intelligent" exponentially than in the time relation.

Artificial intelligence (AI), the science and engineering of making intelligent machines, must be implemented generally through the combination of space (more processors) and time (delayed time through feedbacks to

previous routes) consideration. The space consideration means more quantities translate into higher quality and this is a necessary condition for a true AI, which is certainly cannot be derived from our current electronic computer using a sequential architecture. Thus massive parallel computing scheme, including the birth-and –death phenomena of the computing, is a necessary step and is the first condition for AI. The second and weaker condition of AI is through the use of some "feedback" loops in time domain, which is less valuable, another view that time is less valuable than space in computing. A parallel computing's learning, reasoning and problem solving capabilities will be in a much narrower scope if there is no growth mechanism implemented in the computer. Using feedback loops or hierarchical control mechanisms alone are not sufficient for a true AI.

Now the computing energy and computing irreversibility need to be addressed. All quantum computing are not thermodynamically reversible as pointed out by Landauer [22]. The erasure energy has a lower limit equal to $kT\ln(2)$. The question we address here is the reversibility in computing that also has been investigated earlier [6]. General-purpose quantum computing is not a sequence of linear operations on some initial states generally and is thus not a reversible computing as the qubit approach has suggested [12-15]. There are no merits, nor it is correct, in defining quantum computing solely through its reversibility in computing as shown by many qubit-based researchers [12-21]. In the quantum computing scheme [4-6], a CA chain is updated by an electron-wave (write) pulse and followed by two refresh pluses (to erase two previous data contained in the processor) into each cell in a charging-and-refreshing cycles. Thus for each energy in the pulse for writing (or storing) into the cell, twice of that amount of energy must be consumed to erase the previous data. This is a net loss from the erase part of the energy that is equivalent to the "write" energy, which also has the lower limit equal to $kT\ln(2)$.

Addition operation is a reduction process of spatial patterns through the CA rules shown in Fig. [3a]. There is unequal probability of distribution of the four states from the four pairs of operands. If two (three) processors are used, there are 16 (64) possible input patterns and the reduction process in one-step addition operation produces 9 (27) possible outcome patterns. The computing can be reversible only if one of the two input bits in each cell is not erased at each step so that a reversible CA can be implemented. Equivalently, that means both addition and subtraction must be performed simultaneously. But the space correlations between two processors (the sixteen 2 by 2 spatial patterns that constitute the entire addition table) can be duplicated by one processor only through 32 times in sequences. Those sequences can be achieved only if we slice the space and scan each sub-space into a two time sequence to establish the space-time mapping *a priori*. Thus the capability of 16 instructions contained from two processors in one step cannot be reproduced by one processor of 4 instructions in two steps. As the number of processors increases, the number of instruction capacity increases such that a single processor in many time steps cannot catch up with the difference and thus causes the irreversibility in computing.

If a physical system can be considered as a natural parallel computing system, then it is an irreversible process by Euclidean addition operation of consideration alone.

Once a physical process (or a natural parallel computing process) is concluded, those spatial content is not retained and only the result is observed or obtained by us. So in general, the distinction between "past" and "present" in time is that some information in the past space is always lost in the present space, an irreversible computing process through successive reduction processes. Therefore general-purpose quantum parallel computing is irreversible both thermodynamically and computing wise. The only reversible computing that can happen is when all the operations in the processors are independent of each other, such as in the situation of quantum Fourier Transform.

From the consideration of geometry, one processor occupies one cell space and then assume the rest of the space (the entire CA chain) is Euclidean. Two processors occupy two cells space and then assume the rest of the space is Euclidean and so on. The important concept is that when infinite processors are used, the infinite chain has to be Euclidean. This situation cannot be just assumed now but must be exactly the case or the addition operation will fail to achieve because the processors in the CA chain has to be actually filled spatially, not just assumed. But Euclidean is a very singular situation or a rare occurrence. Any deviations from the Euclidean condition can come from rotations and/or from translations with respect to the Euclidean reference frame established by the addition operation (FIG.2d and 2e). Thus those three basic wiring lengths connecting processors cannot be arbitrary and are defined by imposing the Euclidean condition first for the addition operation. This then establishes an equilibrium condition of parallel computing, similar to the establishment of the chemical bonding lengths for a cluster of molecules by imposing a minimum free energy condition at the equilibrium. Before that discussion is made, a review the quantum parallel computing scheme, namely, the processor and the correct architecture [4-6], must be presented.

III. Intrinsic wiring lengths possessed by a quantum processor

It is well-accepted that a general-purpose quantum parallel computer must compose of a large number interconnected processors and each processor must be capable of executing certain number of instructions as well as the storage of data, just like the classical counterparts. Beyond that, there has been no common consensus in the research communities on the processors and the computing architectures (and hence the interconnections) to be used until recently when a complete scheme on a suitable processor and the corresponding architecture are presented [4-6].

Many main-stream researchers are pursuing qubit-based quantum computing [12-21]. But the superposition of states, the heart of this qubit concept, is actually irrelevant even for the addition of two bits when two qubits are entangled. It is then the four symbolic substitution rules, rather than the superposition of the four states, that can perform such an elementary computation. Thus qubit concept is fundamentally flawed [6] for general-purpose quantum computing. The reasons are: first, one must be able to make processor-to-processor interconnections correctly and, secondly, one must consider the internal coupling (the entanglement of qubits) and the external coupling (the readout) of the computing system as one complete system with no separation. For the first argument, when many qubits are entangled, they form a large extended electron wave packet. In it, if we consider one qubit as one processor, capable of performing a truth/false computation, then all processors must be shown interconnected specifically. Similarly, if we consider two qubits as one processor capable of computing the addition operation of just two bits, then all processors need to be shown how they are interconnected accordingly. Anything that is computable can vary from pure sequential nature of the computing, such as addition operation, to that of a pure parallel computing nature, such as Fourier transform or Shor algorithm [23] and anything in between the two extremes. Thus for general-purpose computing, one must show the addition operation first. If every two qubits are grouped together and considered as one processor, then the processor for the least significant bit-pair must be connected only to the processor for the second least significant bit pair. That is to its left side and hence there is a very specific spatial relation, that is the left-side processor interconnection needed for this computing. But the concept of the superposition of qubits makes no distinction of left and right or up and down. The entire wave packet cannot perform the spatial aspect of the computing, such as transferring the result of the right processor only to the left side of the processor, the dependency. Only when all processors perform independently can the qubit system be possible to perform sensible computation, such as Fourier Transform, a linear operation. But that is a special kind of parallel computing, not general-purpose one as demonstrated by the Turing Machine[10], which uses eight symbolic substitution rules on a one-bit processor [10]. Symbolic substitution rules impose spatial relations while logic gates cannot. Simply speaking, for parallel computing, the processor used must be rule-based and not logic-gate (or truth table) based. This is the great departure of the starting concept here as compared to the qubit concept pursued by others [12-21,23]. The second fundamental flaw is the integration of internal entanglements with the external perturbation, the readout process, as one complete system. One can consider one qubit as one harmonic-oscillator ring, as in a flux qubit in the form of an isolated Aharonov-Bohm (AB) ring or an artificial atom and with its angular momentum directions in superposition, that is maintained at a proper flux value. It is a closed system. But this closed harmonic oscillator AB ring can be coupled strongly to external perturbation made by a chain of harmonic oscillator of the same strength only. Any stronger or any weaker of a harmonic chain will not be able to couple to the ring better [24-25]. Now there are two issues: the first one is that the locations to connect the harmonic-oscillator chains to the ring to perform the scattering events for a readout are themselves part of the computing. They cannot be separated. The second issue is that if there are many quibits present in that closed system and they are all weakly coupled harmonic oscillator AB rings so as to maintain the superposition. Then any readable external perturbation will inherently destroy that superposition. In other words, it is not a computable system when one qubit has to depend on the one next to it. So instead of forcing the external perturbation to be weak to cooperate with the internal system, it is the other way around. Thus, the solution is to form a diatomic molecule by collapsing two qubits to bond more tightly and with three external terminals, instead of some weak probes attached to the two point-contacted AB rings (Fig.5a). This becomes the fundamental quantum processor, which transforms the four states in superposition by two qubits into four symbolic substitution rules on a two coupled AB rings and form a two-bit cell for CA [26]. This is to be compared to the eight symbolic substitution rules on one-bit cell as proposed by Turing [10], who has not shown how those eight rules can be implemented. The principles of constructing this quantum processor are based on: (1) the use of an extended quantum circulator concept [27] when the flux pair are in the opposite directions and (2) the utilization of Anderson Localization effect [28] due to an increase of the two scattering centers when the two AB rings are strongly coupled if the flux pair are in the same direction. In the operation, a test scattering pulse can be transported to any one of the two terminals (terminals C and D in FIG.5a) when the quantum circulator principle is employed. When a large reflection is needed back to the input terminal (terminal S in FIG.5a), Anderson Localization effect is realized. The four flux- pairs can then be mapped into the four symbolic substitution rules properly [26]. There are indeed many suitable flux conditions that those four

symbolic substitution rules can be reasonably realized based on the principles described above. It is not a unique situation that we have shown here in the reference [4, 26].

Once a processor is inserted into each cell of a CA chain, any addition operation of two long bit-strings can be performed only if the processor-to-processor connections are correctly shown in Fig. 6a. The CA architecture provides a minimum wiring scheme to connect all the processors. Thus those canonical connections are to be imposed for any addition operations. This is shown in Fig.6a and is described by the corresponding electric AB effect for convenience (see FIG.5b). Each quantum processor must have three terminals, labelled as S, C and D. When the input bit pair is in the state of S1=(0,0), a test pulse from S terminal will be scattered into the D terminal (a quantum circulator effect). When the bit pair is in S2=(0,1), the same pulse will be scattered back to S terminal and the charge arrives at V1 metal cylinder. Similarly when the bit pair is in S3=(1,0) , the result is the same as the case for the state of S2 (Anderson localization effect). But when the bit pair is in S4=(1,1), the pulse will be scattered C terminal, which is to be connected to the neighboring processor on the left side at its V2 metal cylinder (a quantum circulator effect again). This processor performs a pure quantum computing because manipulating the phase of electron wave function through the guage invariance requirement is a pure quantum effect. The quantum interference effect through the electron transport in multiply-connected space (the quantum networks) provides the results [29]. It is important to note that this can be achieved only after abandoning the concept of the superposition of states for the general-purpose quantum computing. Quantum coherence is maintained only within a cell because a readout needs to be made after each scattering event in general-purpose computing. For quantum Fourier transform, all qubits are independent of each other and are read as a whole. So the coherence length extends over the entire system.

It is now important to note that by bringing the quantum processor into each cell of CA, the EP concept for algebraic operations must be used.

The origin of the intrinsic wiring lengths possessed by a processor in quantum parallel computing can be understood by examining the relation between geometry and parallel computing. The wirings for addition operation is for C terminal be connected to V2 cylinder of the nearest left cell, for S-terminal be connected to V1 cylinder within the cell itself and for D terminal to be connected to a dump and be unused (Fig.5b). This is the canonical wiring scheme (Fig.6a). In principle, each processor can be allowed to locate arbitrarily even in a three-dimensional space with many random wiring lengths. If this is the case, the results to be read still need to be further traced and mapped into one-dimensional Euclidean reference plates. This tracing will cause additional computing efforts. To minimize the energy of those tracings, one has no choice but to use Euclidean concept so that all bit-locations are lined up in a proper order from least significant bit to the most significant one and those bits are resided in their respective processors of the CA chain. Therefore this quasi-one-dimensional EP is the minimum template for the wirings of a quantum adder chain. Associated with this EP template, there are three distinctive wiring lengths established. Those three lengths that make the EP construction possible are now the integral part of the processor. In others words, it is not the concept that those processors need to be wired together. Rather it is that each processor possesses with the three intrinsic and distinctive wiring lengths can establish the equilibrium condition of computing in EP. When the canonical connections are made, it will form a proper EP structure and perform the addition operation in a CA chain correctly, strictly from the point of view between geometry and parallel computing. Note that the processors used are all identical in size in this concept and there are scaling relations [24-27] when the size of the processor is scaled up and the three wiring lengths are then scaled accordingly. Once those lengths are established and fixed, other type of connections, or other type of computing are possible with those three intrinsic lengths unchanged. The deviations from the canonical connections will then change a EP structure to that of a NEP. In that case the computing is no longer addition-rule compatible. Two examples of Euclidean and Non-Euclidean computing are illustrated in Sections IV and V.

IV. Classifications of Euclidean and Non-Euclidean computing by addition rules

Note that once a half-adder processor is imbedded in each cell for any addition operation, the CA has only the right nearest neighbor interaction rule (and the right cell is actually wired into each cell as shown on Fig.6a.

The symbols of the upper row are the configurations for each parent cell and its right neighbor and the symbols for the lower row are for the child cell itself after the iteration. This particular set of the transition rules in Fig.3a is only one set out of a total of

4^{16} = 4,294,967,296 available sets. In Wolfram's version [8-9], this addition- rule based CA has the set number of 39278338 (which is the binary 00000010010101110101011100000010).

An example is illustrated again for this addition-rule based CA so as to be compared with the example in Section V. Let a 6-digit operand A=101011 and operand B=010101. The 7-digit result of this addition operation is 1000000. That is the decimal addition of 43 + 21 = 64 operation. According to the CA scheme shown in FIG.7, the two operands combined have the starting configuration of $S_2S_3S_2S_3S_2S_4$ and are located

at the cell locations, labeled as 6,5,4,3,2 and 1 on the top horizontal axis in Fig7 and is designated as the original parent configuration (labeled "0" on the left vertical axis). The rest of the 1-D space are then filled with S_1 's from cell 0, -1, -2 -3 and so on as well as from cell 7,8, 9 and so on as a uniform background. Using the 16 transition rules from Fig.[3a] for each cell with respect to its right neighbor, after the first iteration, the new configuration becomes $S_2S_2S_2S_2S_4S_1$ at the locations from cell 6 down to cell 1 (labeled as iteration "1" on the vertical axis). Repeatedly using the rules in FIG.3a for the iteration scheme, after the 7th iteration, the reduction process resulted in the configuration of $S_2S_1S_1S_1S_1S_1S_1$ and remains the same configuration after further iteration. The lower row is thus the result, which is the sequence of 1000000, which are located at the V_1 metal cylinders from cell 7 to cell 1. This example illustrates that an addition operation is a generation of a proper "space-time block" imbedded in a uniform background as shown in FIG.7 because addition operation utilized a specific set of spatial relations (FIG.3a or FIG.4a).

Note that for the addition operations, the half-adder processor in each cell has three output terminals and only two input terminals, V_1 and V_2 . Thus one of the output terminals (D terminal) is always discarded for Euclidean computing. Therefore other interconnection schemes or other spatial relations are available for computing. Those Non-Euclidean computing have been shown [5], including glider-like or oscillating structures, contrary to the assertion by Wolfram [8-9].

V. Non-Euclidean computing: An example of birth-and-death of space-time blocks from a structured Sierpinski triangle generated by a seed.

A small computing task is said to be accomplished when a block of processors can perform certain computing over a specific time interval as its mission. That mini-computing occupies a space-time block imbedded in a uniform background in a CA and can be reproduced (birth) later at several desirable locations simultaneously and then be eliminated (death) when the missions are accomplished. Thus the birth-and-death behavior in computing is the capability to duplicate from one space-time block of computing to several identical space -time blocks at the uniform background and then terminate those computing at the conclusion. That is an important feature needed for AI.

Generally speaking, parallel computing scheme are on how to replace one spatial pattern with another repeatedly, or implementing the symbolic substitution rules in CA. The spatial relations for addition rules shown in the framework of CA in Fig.3a illustrate a reduction process through the sixteen CA transition rules. Clearly S_1 and S_2 states have three times more the probability to occur than S_3 and S_4 states. So after some iterations, only S_1 and S_2 states remain and that is where the addition results are resided in a block of adjacent cells. If the symbolic substitutions are altered (Fig.4b) such that all four states have an equal probability to survive in each iteration (Fig.3b), then the birth-and-death behavior can be realized. A structured Sierpinski triangle starting from a seed is shown in Fig. 8a. Here the seed is deliberately chosen to be the same as the previous addition example in Section IV. Thus the initial configuration is a 6-cell-occupying structure of $S_2S_3S_2S_3S_2S_4$. The symbolic substitution rules are shown Fig.4b and the corresponding set of 16 CA transition rules are shown in Fig.3b, which clearly show that all four states have equal probability to survive, unlike the reduction process for the addition rules shown in Section IV. After two iterations, the 16 rules are reduced to only 8 rules used (marked with *). The structured Sierpinski triangle is in the form of many 8 x 8 space-time blocks stacked together as shown in Fig. 8a. Instead of the 8 CA transition rules used (marked * in Fig.3b), one can construct 8 large-scale and higher order space-time block transition rules. Those are shown in Fig.8c, where rule 6 illustrates the birth and rule 4 illustrates the death. Note that generating Sierpinski triangles without a starting seed (and hence without structure) have been shown by Wolfram earlier, where the entire 8x8 space-time block shown here is reduced to just a single bit [30-32]. The seed used here is at the upper row of FIG.8b.

In the triangle, any two time slices on the horizontal axis or any two space slices on the vertical axis are correlated. It becomes a deterministic system. At the iteration marked '3' of FIG.8a, there are now two solid identical space-time blocks, a birth process. Then at iteration marked '4' in the same figure, there is a death process and only the original solid block survives.

In AI, a useful space-time block, or a computing task, needs to be reproduced by a seed with certain copies made at some specified locations and then be eliminated at the conclusion of the mission. Those procedure-computing are Non-Euclidean in nature and need to be integrated (by interacting with a canonically connected CA chain through charge transfer at all V_1 and V_2 metal cylinders) into the general-purpose computer. For

example, an IF-THEN conditional branching of a computing task can be imposed if there exist, say, four copies of a space-time block simultaneously appear at specific desirable locations (such as at location '13' of FIG.8a) of the CA chain, then it triggers a new route of computing at another location (a new CA chain). In other words, the IF-THEN condition is set by a spatial threshold (or a spatial pattern) condition.

VI. Conclusions

From geometry point of view, parallel computers are composed by a huge amount of space-filling processors, which can be Euclidean or Non-Euclidean constructed. The spatial relations, or the interconnections provide the most important features or the signatures of the computing once a processor is chosen. The interconnections among the processors are at a minimum when cellular automata architecture is employed. Thus there exists a large varieties of computing scheme that can be constructed by varying the spatial relations of the same processor used in each cell. Addition operation is only one singular kind among them and must be Euclidean-based. But for a general-purpose parallel computer to be constructed, several important spatial relations besides the addition rules compatible computing, such as the spatial relations to generate birth-and-death of space-time blocks, must be utilized. For this understanding, the triangle relations among geometry, computing and physics need to be properly established first. The phrase of "quantum parallel computing" simply implies that some pure quantum phenomena are utilized inside the processor in each cell of the CA. Here, the principle of physics involved is the use of the guage invariance of electron wave function.

With two bits per cell in the one-dimension CA, there are 4^{16} , or roughly 4.3 billion, sets of spatial relations. Each set is itself a form of computing in its proper geometrical space. Clearly, parallel computing in the CA architecture defines a higher order form or a broader origin for a wide variety of computing defined by the spatial relations used. Our algebraic operations (and hence all the consequences) are consistent with only one of those huge available geometry-controlled computing sets. But the addition rules are reduction processes of four possible states, or four operand pairs, through the CA transition rules. That means the bits located at the V_2 cylinders (Fig.5) must be all '0's at the end of the computation. This provides the irreversibility in any addition operation, unless subtraction operation is also performed at the same time. Our interpretation of physics must be addition-rule compatible and thus all physical processes are irreversible from parallel computing point of view alone. However, if those four states possessed in each cell have an equal probability to survive in the CA transition rules so constructed in FIG.3b, then birth-and-death phenomena can occur (such as the example of generating a structured Sierpinski triangle from a seed block). But the birth-and-death phenomena, which are the necessary conditions for AI to occur, are addition-rule incompatible in CA and are Non-Euclidean computing in nature. But they need be incorporated into the non-algebraic-related parts (such as procedures) of the general-purpose parallel computing. All Non-Euclidean computing look new or odd as projected or viewed through a Euclidean frame.

Physical processes in nature can be considered as a sequence of continuous parallel computing on a huge CA. That means the whole world is a large CA with the giant space keep generating a huge chart of space-time block everywhere. Then one needs to utilize a standardized quantum processor to quantify the results or to interpret the huge chart by utilizing several appropriate sets of spatial relations in the form of interconnections. Each interconnection scheme produces certain type of CA chain capable of generating certain space-time block. Those small space-time blocks are then to be compiled together. That is to say that a computation processor implements physics laws if those local behaving quantum processors to be utilized are placed in some proper geometrical space, Euclidean and Non-Euclidean, for general-purpose computing with artificial intelligence capability. Physical laws utilized in the processor are for local behavior only and geometry means the interconnections imposed for a long-range development. When billions of those identical processors are used for parallel additions, the geometry used is Euclidean, that space has to be actually space-filling Euclidean because addition rules are based on an EP concept. This computing equilibrium condition forces each processor to possess three intrinsic wiring lengths when a particular size of the processor is utilized. The size is scalable. With billions of processors physically present, the Euclidean utilization of space allows a minimum effort to identify the location of each processor where the results are resided and are to be read. Thus the Euclidean requirement between the physics and geometry for parallel computing is similar to the minimum free-energy requirement between the physics and geometry in determining a cluster of molecules in thermodynamic equilibrium. EP concept leads to the computing equilibrium condition for a CA chain and the intrinsic wiring lengths of the quantum processor are the counterpart of the chemical bonding lengths.

With the wiring lengths as the integral part of the processor once addition operation can be established in a Euclidean concept, interchanging those wires to any different connections other than the canonical one will alter the spatial relations between the physics and geometry.

Pattern transformations may appear to be drawn correctly on two dimensional or on one dimensional EP in CA and give the impression that the pattern transformations are executed on a Euclidean space. This is because to analyze any spatial pattern transformation requires one to borrow a Euclidean frame as the reference or as a projection. Parallel processing must be viewed as having the capability of a massive spatial pattern transformations in one step. Furthermore if each processor cannot handle the half-adder capability, that is the four-instruction capability, then the iteration results from CA will include chaotic patterns. Thus the half-adder processor serves as a meaningful tool to classify those vast results from CA's.

From the fact that addition operations by symbolic substitution rules are reduction processes by reducing the number of states on hand at every step, there is no Poincare recurrence time because it is an irreversible computing process, contrary to the assertion by Bennett [12,33]. Quantum dynamics behave, or can exist, only in small sections of a CA chain and preferably within a cell. In between any two sections or two regions of quantum dynamic systems, a readout or a collapse of wave function is needed. For example, quantum dynamics exist in pockets of pure crystalline regions until an inelastic scattering (a readout process) occurred at an impurity. On the other hand, a photon may travel to a large distance, only if no computation is performed. But the entire world cannot be a giant quantum dynamic system as described by Bennett [33]. In addition, quantum behavior cannot result in having a birth-and-death phenomena in a CA chain.

There are two conflicting views on quantum parallel computing by researchers. The qubit- approach considers a computing process as a set of unitary operations on some initial quantum states, which is a special form of evolution in a reversible CA operation and that is a closed system. Then the readout on the final quantum states is likely a guaranteed external perturbation. But such a close system is not computable when there is a slightest component of the sequential or dependency nature of computations involved. A two-qubit system is shown to be irrelevant for the addition for two bits and thus play no role for general-purpose computing. Qubit- approach is applicable only to Fourier transform, a pure parallel computing in nature. On the other hand, the approach presented here and in references [4-6] considers any computing process as generating a space-time block on a CA chain in Euclidean and Non-Euclidean manners, instead of a quantum dynamics in the Hilbert space. The utilization of quantum phenomena is limited to one cell space only. The coherence of the electron wave function does not extend to the entire CA chain. The reason is that at the most elementary level, the result of an addition operation must be able to be read at any bit-size level without making separate arrangements. That means electron wave function needs to be collapsed at each cell level.

Writing a bit stores a minimum energy of kT ln (2). But the need to erase or to clean out the previously stored two bits consumes twice as much of that energy. Thus as each processor undergoes an iteration step, there is a net loss of kT ln(2) minimum energy, regardless whether the computing involved is Euclidean or Non-Euclidean in nature.

Space and time are not on equal footing in parallel computing. Space contains exponentially more bits or computing information, in the form of the spatial relations, in one time step. On the other hand, time contains linear amount of bit informations in steps. The unmatched bit contents along the space and time axis results in irreversibility of information. The loss of spatial information from the past contributes to the computing irreversibility and hence to the irreversibility of any physical process as well.

References

[1]. C.N. Yang, Physics Today, 67, 45 (2014)
[2]. F. Fredkin, Physica D, 45, 254-270 (1990)
[3]. R.P. Feynman. Intl. J. of Theoretical Phys. 21, 6-7 (1982)
[4]. C.H. Wu and C.A. Cain, Physica E, 59, 243 (2014)
[5]. C.H. Wu, J. Cellular Automata 9, 271 (2014)
[6]. C.H. Wu, Intl. J. Engineering Science Invention, 3, 59 (2014)
[7]. E. Efrati, E. Sharon and R. Kupferman, J. Mechanics and Physics of Solids, 57, 762 (2009) .
[8]. S. Wolfram. A new Kind of Science, Champion, Il. Wolfram Media 2002.
[9]. S. Wolfram. Cellular Automata and Complexity: Collected Papers. Reading, MA. Addison-Wesley. 1994.
[10]. A.D. McAulay. Optical Computer Architectures. Wesley. New York (1991)
[11]. Y. Aharonov and D. Bohm, Phys. Rev. 115, 485 (1959)
[12]. C.H. Bennett. Intl. J. of Theoretical Physics, 21, 905-940 (1982).
[13]. P. Benioff. J. of Stat. Phys. 29 (3), 515–546 (1982).
[14]. D. Deutsch. Proceedings of the Royal Society of London; Series A, Mathematical and Physical Sciences 400 (1818), 97–117 (1985).
[15]. L. K. Grover. Proc. of the 28th Annual ACM Symposium on the Theory of Computing. 212 (1996).
[16]. D. P. DiVincenzo. Topics in Quantum Computers. Mesoscopic Electronic Transport.
[17]. Kluwer Academic Publishers. Dordrecht, Netherlands. 657 (1997).
[18]. D. Loss and D. P. DiVincenzo. Phys. Rev. A 57, 120 (1998).
[19]. D. D. Awschalom and M. E. Flatte. Nature Physics 3, 153 (2007).
[20]. R. Hanson and D. D. Awschalom. Nature 453, 1043 (2008).
[21]. L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. I. Schuster,
[22]. J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. Nature 460, 240 (2009).

[23]. L. DiCarlo, M. D. Reed, L. Sun, B. R. Johnson, J. M. Chow, J. M. Gambetta, L. Frunzio, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Nature 467, 574 (2010).
[24]. W. Shor. SIAM J. Comput. 26, 1484 (1997)
[25]. R. Landauer, IBM J. Research and Development 5,183 (1961)
[26]. C. A. Cain and C. H. Wu. J. Appl. Phys. 113, 154309 (2013).
[27]. C. H. Wu, L. Tran and C. A. Cain. J. Appl. Phys. 111, 094304 (2012).
[28]. C. A. Cain and C. H. Wu. J. Appl. Phys. 110, 054315 (2011). Also US Patent #8,525,544.
[29]. C. H. Wu and D. Ramamurthy. Phys. Rev. B 65,075313 (2002).
[30]. P.W. Anderson, Phys. Rev. 109,1492 (1958)
[31]. C. H. Wu and G. Mahler. Phys. Rev. B 43, 5012 (1991).
[32]. E.W. Weisstein, Sierpinski Sieve from MathWorld, a Wolfram Web Source, http://mathworld.wolfram.com/SierpinskiSieve.html
[33]. T. Toffoli and N. Margolus. Cellular Automata Machines: A New Environment for Modeling. Cambridge, MA. MIT Press. 1987.
[34]. M. Gardner. Mathematical Games: The fantastic combinations of John Conway's new solitaire game of life. Scientific American. 223, 120-123 (1970).
[35]. C.H. Bennett "Quantum Information, the ambiguity of the past, and the complexity of the present", Perimeter Institute lecture 12 May 2011, URL: http://pirsa.org/11050052

FIGURE LEGENDS

FIG.1: (a) The triangle relations among physics, geometry and quantum parallel computing. (b) The analogy of the triangle relations among the chemical bonding lengths, minimum free energy and the computing for a cluster of molecules in thermodynamic equilibrium.

FIG. 2: An EP of elastic material system is shown in (a) with individual plates. The location of each plate is marked to be consistent with a reference straight line, plate P1 is located at a one unit distance from the origin and plate P2 is located at two unit distance and so on. When a bending is made, each plate is relocated as shown in (b), such that the individual cannot be hammered into EP shape. The location of each plate now deviates from the previous straight line axis through rotation. An equivalent electrical EP system is shown in (c) , where there is one processor residing in each cell and is connected to its left neighbor with two equal wires. P1, P2, P3 are the processors and are located on the horizontal line of unit 1,2,3 unit distance from the origin respectively. This allows the computed data read from processor P1 to be associated with the least significant bit, P2 with second least significant bit and so on, for example. When one of the two wires is shortened as shown in (d), the location of each processor with respect the reference horizontal line is altered, similar to the elastic material system in (b) and cause additional problems in reading of the data from each processor to associate with the location. A second situation is shown in (e) when each processor is connected with only one wire to its left neighbor as an EP system. P1 is located at one unit of distance with the origin so that the data to be read from P1 is associated with the least significant bit, P2 at two units from the origin and associated with the second least significant bit and so on. The Euclidean system is established if all the processors are equally spaced in a straight line with a proper marking of its coordinate as described. When a processor with a proper size that is fixed, there is a proper EP , which form a one-dimensional CA. Each processor has it's a fixed external connecting length C , two internal lengths S and D. For the EP concept to function correctly, the lengths C,S and D must be considered as the intrinsic lengths of the processor. When internal length and external length are interchanged, for example, a shorter length S is connected as shown in (e), the location of each processor with respect the reference horizontal axis is altered and data so obtained are then not correct only with respect to the EP system. This is an example of a NEP computation for a parallel computing. Note that this translational shortening of the wire length exists a scaling factor with respect to the reference frame above. But the value of this scaling factor cannot be determined a priori.

FIG.3: Cellular automata parent-child transition rules with two bits per cell and the right neighbor interaction: (a) the 16 transition rules for any addition operation must be in EP concept. Note that in the 16 child states, S_1 and S_2 states have three times more probability to survive as compared to S_3 and S_4 states. Addition rules are reduction processes of the four states. The corresponding symbolic substitution rules are shown in FIG.4a, (b) the transition rules for generating the Sierpinski triangle of FIG.8. The 16 child states are equally distributed in the four states. The corresponding symbolic substitution rules are shown in FIG.4b.

FIG.4: Symbolic substitution rules for search-and-replace of spatial patterns. The four rules and the four states are shown on the left two columns and dual-rail notations are used on the right two columns; (a) for addition operation. Note that addition operation of operand A and operand B is achieved by the four symbolic substitution rules here, rather than from the superposition of four states in two qubits; (b) for an equal survival probability of the four child states (FIG.3b) and used in generating the Sierpinski triangle example.

FIG.5: The structures of the fundamental quantum processor. (a) A quantum network constructed from two strongly coupled Aharonov-Bohm (AB) rings with three terminals on one of the rings. Two fluxes, Φ_1 and Φ_2 , are the bit inputs. A test pulse from the S terminal will be scattered into C, S or D terminals according to

the four flux pairings. This is the magnetic AB version. (b) The corresponding electric AB version: The charge on V_1 and V_2 metal cylinders replace the two fluxes in (a). The rings are drawn in square shape and rotated. QC is the quantum circulator such that a charging pulse will be directed to the AB rings, while the reflection will be routed to V_1 cylinder (See reference[4-6, 26-27]). Alternating charging and discharging cycles are used. The reset is for the discharging cycle before the new arrival of the test pulse charge to arrive at V_1 or V_2 cylinder. FIG.6: Processor-to-processor interconnections: (a) The canonical wiring diagram for additional operation in a CA chain, which is EP based. The interconnections are in consistent with Fig.3a and Fig.4a; (b) The NEP wiring diagram for FIG.3b and FIG.4b used in the birth-and –death example of FIG.8.

FIG.7: The space-time block generation of an addition operation in CA. The results of the successive iterations are shown when the original starting configuration is $S_2S_3S_2S_3S_2S_4$, which are located at cell locations from #6 through #1. The rest of the cells are filled with S_1 states as uniform background. Iteration sequences are labeled on the left columns and the cell locations are marked at the top rows. This is the case for the addition operation of operand A=101011 and operand B=010101.

The result of 43+21=64 is read from the binary bits at the 7th iteration configuration, $S_2S_1S_1S_1S_1S_1S_1$, through the polarities of V_1's located from cell 7 to cell 1. The polarity sequence at the V_1's is then - ++++++ or the result of 1000000.

FIG.8: The birth-and-death example from a Sierpinski triangle generated by a seed. (a) The horizontal axis is space and the vertical axis is the time sequence. The left vertical axis labelled as '0' contains two time sequences. A seed of $S_2S_3S_2S_3S_2S_4$, the same as the addition example shown in FIG.7, is used in the uniform S_1 background to start the iteration in a CA chain. Rules from FIG.3b are used. The seed and the first iteration are labelled as S at the top. The sub- sequent iterations are labelled in 8x 8 space-time blocks. Those blocks form a Sierpinski triangle in the uniform background of S_3.

The black (or red) 8x8 space-time block appears in two copies at vertical axis labelled '3' and the extra copy disappears at vertical axis '4' and reappears in 4 copies at vertical axis '7' and so on. (b) Representations of the four different 8x8 space time blocks. (c) Instead using the CA transition rules of FIG.3b to generate the triangle, an equivalent or the higher order block-block transition rules can be derived. The 8 block transition rules are shown for generating the same triangle. The seed is from the two operands of adding 21+43 in the earlier example. Similarly, if the result of the addition, which is 64, is used as a seed instead, then the same Sierpinski triangle is generated, except the four 8X8 space-time blocks in FIG.8b are reduced to four 4x4 blocks. So the use of FIG.3b is quite general for the triangle and FIG.8c remains valid. Note that out of the 16 rules of FIG.3b, only 8 rules (marked by * in FIB.3b and with the equal probability requirement) are actually used in the generation of the Sierpinski triangle.

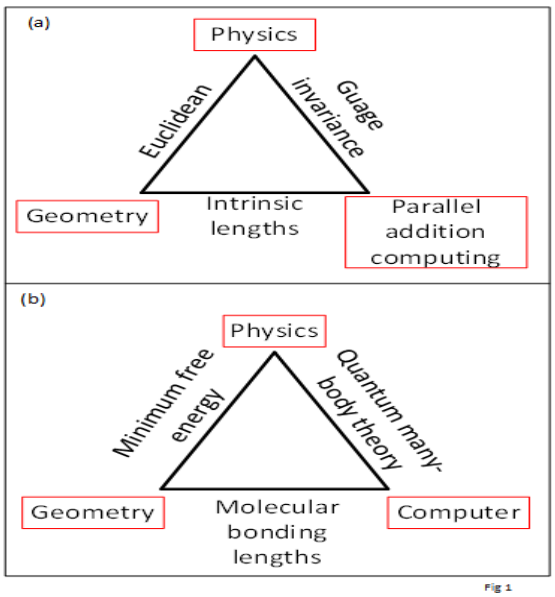

Fig 1

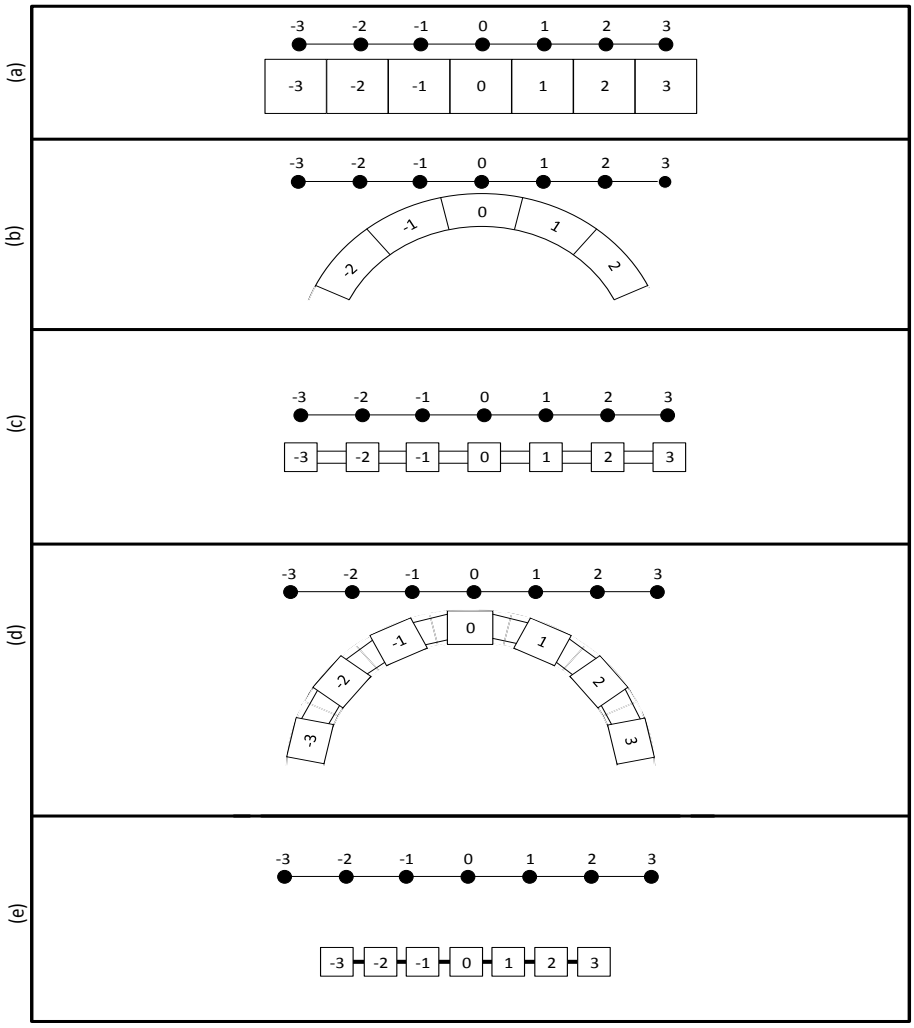
(a)
(b)
(c)
(d)
(e)
-3 -2 -1 0 1 2 3

Fig 2

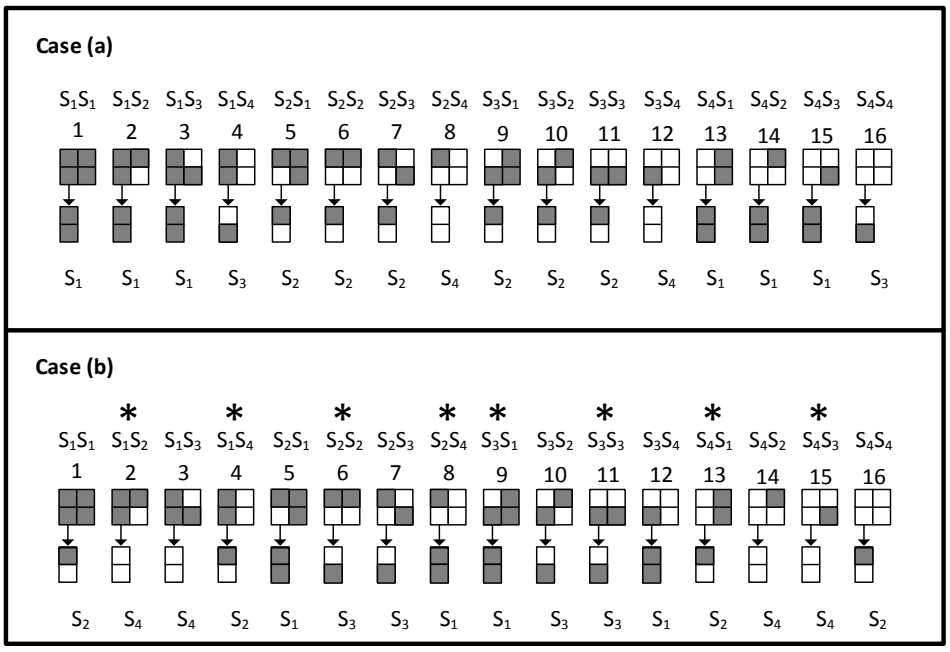
Case (a)
S1S1 S1S2 S1S3 S1S4 S2S1 S2S2 S2S3 S2S4 S3S1 S3S2 S3S3 S3S4 S4S1 S4S2 S4S3 S4S4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
S1 S1 S1 S3 S2 S2 S2 S4 S2 S2 S2 S4 S1 S1 S1 S3
Case (b)
S1S1 S1S2 S1S3 S1S4 S2S1 S2S2 S2S3 S2S4 S3S1 S3S2 S3S3 S3S4 S4S1 S4S2 S4S3 S4S4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
S2 S4 S4 S2 S1 S3 S3 S1 S1 S3 S3 S1 S2 S4 S4 S2

Fig 3

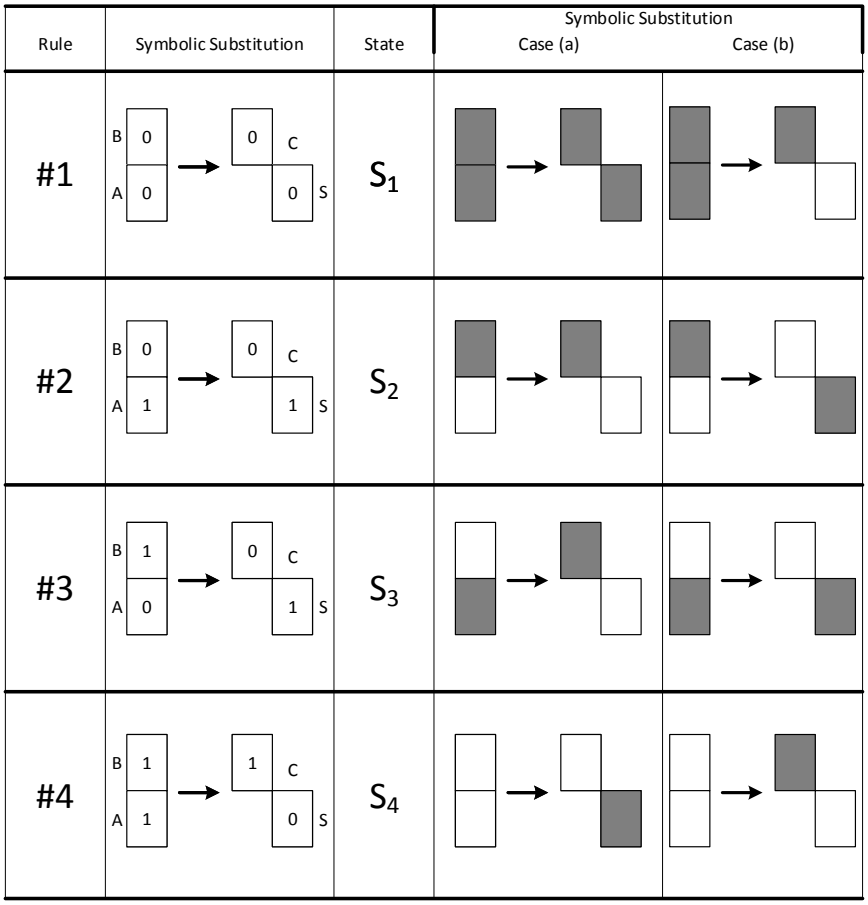
Rule
Symbolic Substitution
State
Symbolic Substitution
Case (a)
Case (b)
#1
B 0
A 0
0 C
0 S
S_1
#2
B 0
A 1
0 C
1 S
S_2
#3
B 1
A 0
0 C
1 S
S_3
#4
B 1
A 1
1 C
0 S
S_4

Fig 4

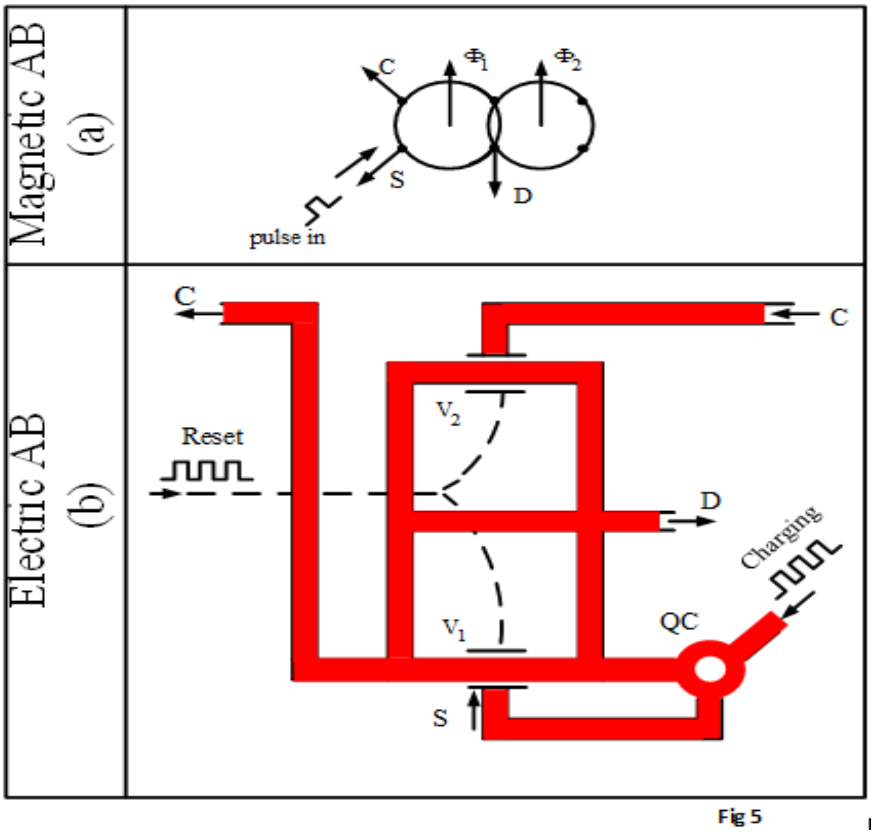
Magnetic AB
(a)
C
Φ_1
Φ_2
S
D
pulse in
Electric AB
(b)
C
C
Reset
V_2
D
Charging
V_1
QC
S

Fig 5

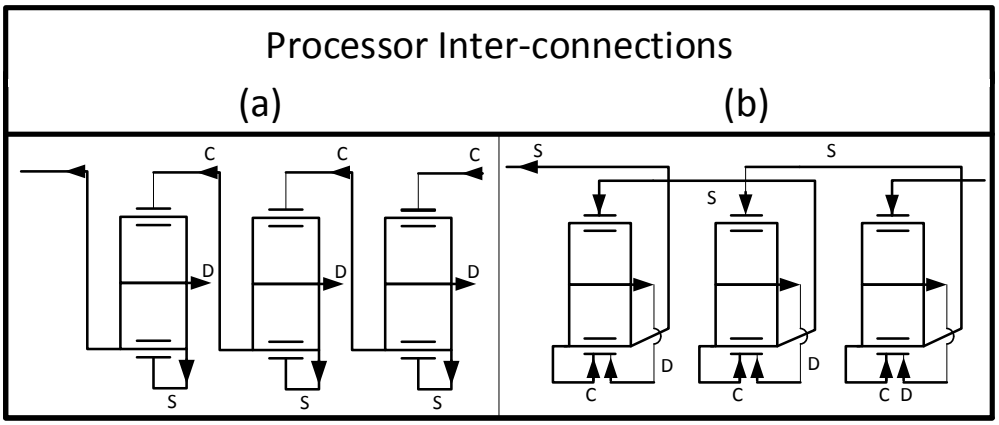

Figure 6

	8	7	6	5	4	3	2	1	0	-1	-2
0	S_1	S_1	S_2	S_3	S_2	S_3	S_2	S_4	S_1	S_1	·
1	S_1	S_1	S_2	S_2	S_2	S_2	S_4	S_1	S_1	·	·
2	S_1	S_1	S_2	S_2	S_2	S_4	S_1	S_1	·	·	·
3	S_1	S_1	S_2	S_2	S_4	S_1	S_1	S_1	·	·	·
4	S_1	S_1	S_2	S_4	S_1	S_1	S_1	S_1	·	·	·
5	S_1	S_1	S_4	S_1	S_1	S_1	S_1	S_1	·	·	·
6	S_1	S_3	S_1	S_1	S_1	S_1	S_1	S_1	·	·	·
7	S_1	S_2	S_1	S_1	S_1	S_1	S_1	S_1	·	·	·
8	S_1	S_2	S_1	S_1	S_1	S_1	S_1	S_1	·	·	·

Figure 7

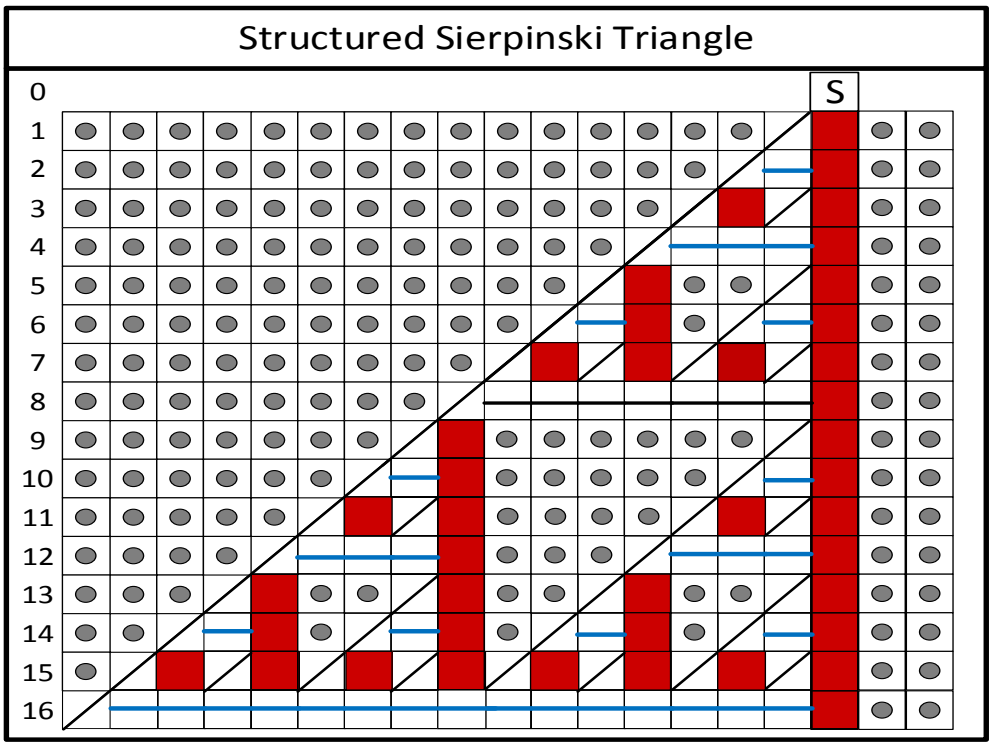

Fig 8 (a)

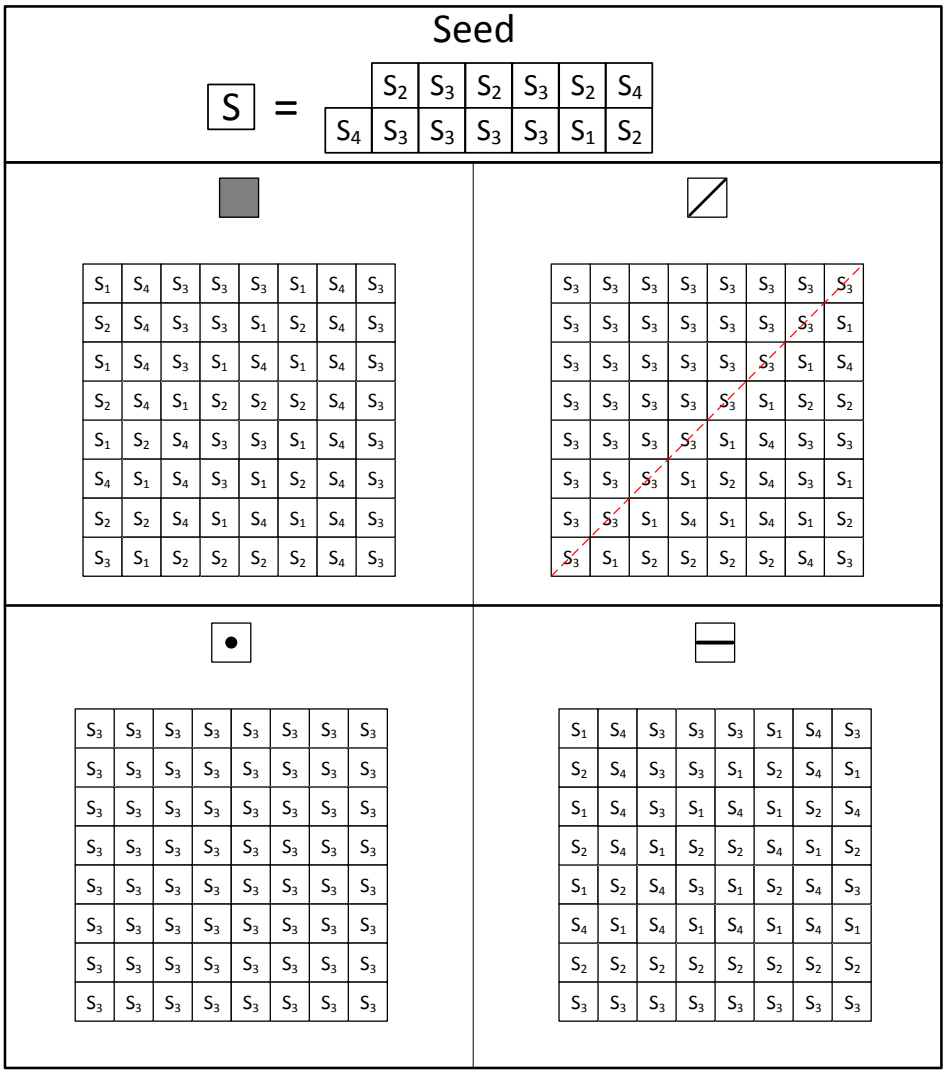
Seed
S =
S2 S3 S2 S3 S2 S4
S4 S3 S3 S3 S3 S1 S2

Fig 8(b)

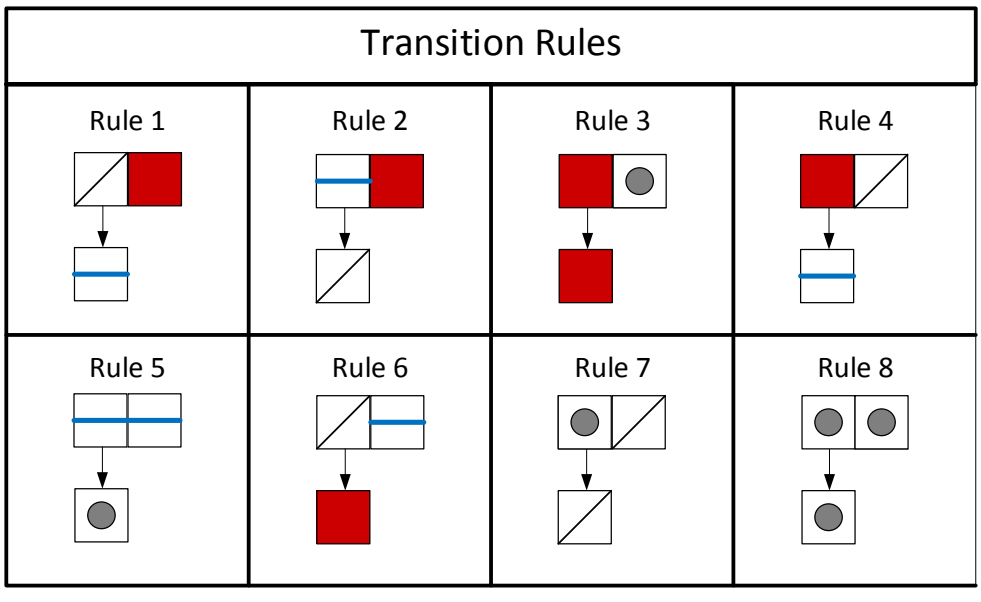
Transition Rules
Rule 1
Rule 2
Rule 3
Rule 4
Rule 5
Rule 6
Rule 7
Rule 8

Fig 8(c)