

Computing Naturally in the Billiard Ball Model

Liang Zhang

Center of Unconventional Computing and Department of Computer Science
University of the West of England, Bristol, United Kingdom
`Liang3.Zhang@uwe.ac.uk`

Abstract. Fredkin’s Billiard Ball Model (BBM) is considered one of the fundamental models of collision-based computing, and it is essentially based on elastic collisions of mobile billiard balls. Moreover, fixed mirrors or reflectors are brought into the model to deflect balls to complete the computation. However, the use of fixed mirrors is “physically unrealistic” and makes the BBM not perfectly momentum conserving from a physical point of view, and it imposes an external architecture onto the computing substrate which is not consistent with the concept of “architectureless” in collision-based computing. In our initial attempt to reduce mirrors in the BBM, we present a class of gates: the m -counting gate, and show that certain circuits can be realized with few mirrors using this gate. We envisage that our findings can be useful in future research of collision-based computing in novel chemical and optical computing substrates.

1 Introduction

Collision-based computing [2] is an unconventional paradigm of computation, where quanta of information are represented by compact patterns or localizations traveling in spatially-extended architectureless medium, such as billiard balls in an idealized friction-free environment [9], gliders in cellular automata [5, 10, 1, 15], discrete solitons in optical waveguide lattices [6] and wavefragments in a sub-excitable Belousov–Zhabotinsky medium [12]. The information is encoded in binary as the presence or absence of localizations corresponds to logical 1 and 0. The computation is performed by mutual collisions of localizations. Trajectories of localizations approaching a collision site represent input values; and trajectories of localizations traveling away from a collision site represent output values. Pre-determined stationary wires and gates are not needed in collision-based computing. Anywhere within the medium space can be used as wires (trajectories of traveling localizations) and gates (collective collision sites of localizations).

Fredkin’s Billiard Ball Model [9] is one of the fundamental models of collision-based computing and a model of universal physical computation, based on idealized elastic collisions of mobile billiard balls. Additionally, fixed mirrors are used to route signals. As one may find that mirrors are almost unavoidable when realizing complicated circuits in the BBM. However, there are several reasons we want to remove mirrors from the model. First, as Margolus [11] stated, fixed mirrors are “physically unrealistic”, since in reality perfectly fixed mirrors “must be

infinitely massive”, otherwise a slightly shift in position of a single mirror could result in the model losing its “digital nature”. Second, from the collision-based computing point of view, mirrors stand for external architecture onto the computing medium. Although they are not stationary wires that signals propagate along and the virtual wires formed by traveling balls are still dynamical, mirrors are routing devices that control the moving directions of signals. Therefore the model is not entirely architectureless. More importantly, for anyone who intends to use the BBM as a blueprint to guide their research in collision-based computing using substrates in the real world, it may not be easy to look for an analogy to mirrors.

Margolus [11] once successfully replaced mirrors with constant particle streams together with the dual-rail logic to deflect signals in a lattice gas version of his Soft Sphere Model (SSM, a model very similar to the BBM, with the difference that balls are compressible during collisions), in an attempt to achieve momentum conservation. Also, a concept of rest particle was introduced into the model to fix the problem of signals crossing without mirrors. Obviously, the dual-rail logic does not apply to the BBM, and the rest particle has no room in the BBM simply because the BBM cannot be simulated by a lattice gas model. Thus methods to remove mirrors in the SSM cannot be directed applied to the BBM.

In this paper, we present a class of gates: the m -counting gate. By using the gate as building blocks, we demonstrate that certain circuits can be realized in the BBM with few mirrors in place, which is one step closer to a physical realistic, architectureless, and momentum conserving BBM. The structure of the paper is as follows, we first review the BBM in Sect. 2. Then we introduce the m -counting gate in Sect. 3, and demonstrate how we use such a gate to realize binary adders and parallel binary counters in Sect. 4 and Sect. 5. As a conclusion, we discuss the difference between the computational logic of the m -counting gate and primitive gates in the BBM, and how our findings may benefit the future research in Sect. 6.

2 The Billiard Ball Model

In the BBM, identical balls with finite diameter travel diagonally on a two-dimensional grid with the same magnitude of velocity. Balls are all initially placed on one of the dots of the grid, and the magnitude of their velocity is specifically chosen so that balls can appear at diagonally adjacent dots at every integer time (Fig. 1a). On two occasions balls can change their moving directions: colliding against each other (Fig. 1b) and/or being deflected by a fixed mirror (Fig. 1c). The diameter of balls is chosen to be the distance between two vertical (or horizontal) adjacent dots of the grid and the fixed mirrors should be carefully placed to ensure that collisions only happen when balls on one of the dots of the grid. Binary information, 1 and 0, is encoded by the presence or the absence of a ball at a given site of the grid at a given time. Therefore, trajectories of balls can be seen as routes of traveling signals, or wires. The routing of signals and the gates can be realized by collisions of balls (with the help of mirrors, sometimes).

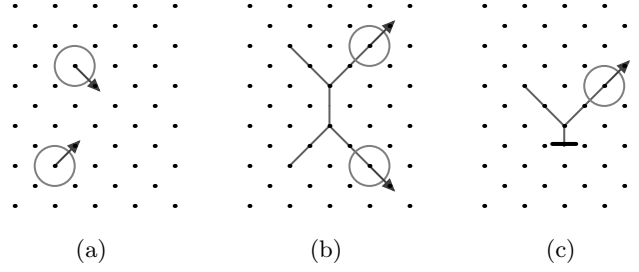


Fig. 1. The Billiard Ball Model. (a) balls move diagonally on a two-dimensional grid; (b) two balls collide and change their moving directions; (c) a ball moving South-East is deflected by a fixed mirror and heads North-East instead. In these graphs, arrows show moving directions of the balls, diagonal lines represent trajectories that balls traveled and vertical lines indicate collisions, either between two balls or between a ball and a mirror.

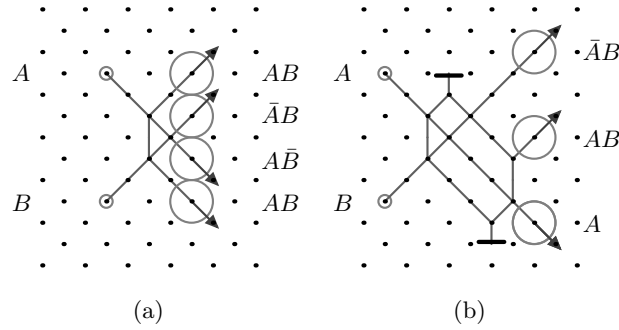


Fig. 2. Primitive gates in the Billiard Ball Model: (a) the interaction gate, and (b) the switch gate.

Two primitive gates were introduced in [9], namely, the interaction gate and the switch gate (Fig. 2). The interaction gate is merely realized by a possible collision of two balls. Fig. 2a shows a superposition of all possible trajectories concerning the gate: when two balls start traveling from positions denoted by small circles on the left, they will collide with each other; and if only one ball is present at the starting positions, it will travel straight through. The switch gate (Fig. 2b) is constructed on the basis of the interaction gate with two additional mirrors. The output signals AB of the interaction gate are routed by mirrors so that one of them falls into the same route as the ball $A\bar{B}$, thus realizing the switch gate. It has been demonstrated that both gates are capable of universal computation.

Later works related to the BBM include a cellular automaton (BBMCA) [10] that simulates the behavior described in the BBM and the Soft Sphere Model [11], both developed by Margolus. Both models dealt with mirrors in a different approach to the BBM. In the BBMCA, mirrors are formed by the computing substrate itself, which is amazing because no external medium is involved. However, when experimenting on real media, it is still hard to find an analogy to the mirrors. In the SSM, especially in a lattice gas version of the model, Margolus managed to remove all of the mirrors. In the BBM, maybe we cannot remove them all, at least not until we find a way to overcome the problem of signals crossing. But we can try to reduce mirrors to a minimum level, and that is where the m -counting gate can help us in certain situations.

3 The m -counting Gate

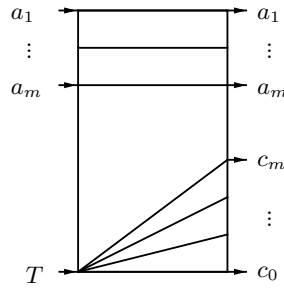


Fig. 3. The graphical realization of the m -counting gate in the Billiard Ball Model, which can be used to count how many logical TRUTHS there are in all m input variables ($a_1...a_m$).

The m -counting gate has m input variables ($a_1...a_m$) and its function is to count the number of logical TRUTHS n out of all m input variables. A graphical

realization of the gate in the BBM is shown in Fig. 3, and realizations of three simplest instances of the gate in the BBM are shown in Fig. 4.

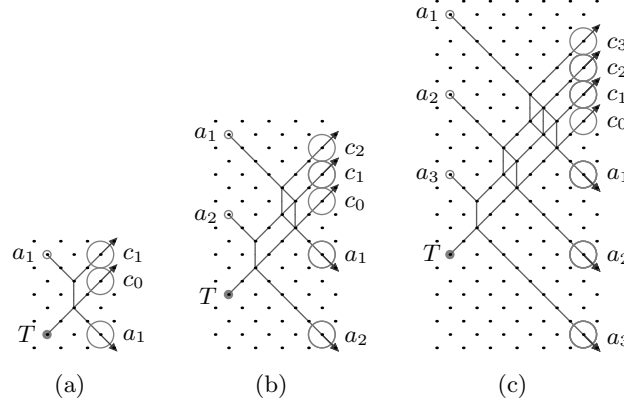


Fig. 4. Simplest examples of the m -counting gate in the Billiard Ball Model: (a) the 1-counting gate, where $c_0 = \bar{a}_1$ and $c_1 = a_1$ (b) the 2-counting gate, where $c_0 = \bar{a}_1\bar{a}_2$, $c_1 = a_1\bar{a}_2 + \bar{a}_1a_2$ and $c_2 = a_1a_2$, and (c) the 3-counting gate, where $c_0 = \bar{a}_1\bar{a}_2\bar{a}_3$, $c_1 = a_1\bar{a}_2\bar{a}_3 + \bar{a}_1a_2\bar{a}_3 + \bar{a}_1\bar{a}_2a_3$, $c_2 = a_1a_2\bar{a}_3 + a_1\bar{a}_2a_3 + \bar{a}_1a_2a_3$ and $c_3 = a_1a_2a_3$. Initial positions of input variables ($a_1...a_m$) are represented by circles and that of the constant TRUTH input T is represented by a dot. Output variables include an exact copy of all input variables ($a_1...a_m$) and a single ball positioned at c_n , denoting that there are n out of m input variables being constant TRUTH.

Generally, realization of the m -counting gate in the BBM consists of m balls ($a_1...a_m$) as input variables moving in one direction, together with a ball T representing a constant logical TRUTH moving in another direction. The gate has $2m + 1$ output variables, with m of which being an exact copy of all input variables ($a_1...a_m$) and other $m + 1$ output variables ($c_0...c_m$). The traveling routes of the output variables are solely determined by the input balls, with $a_1...a_m$ moving one line to the South of their original traveling routes. The route of c_0 is always the same as the constant TRUTH ball T , and routes of $c_1...c_m$ are 1 to m lines to the North of c_0 . Among these $m + 1$ output variables ($c_0...c_m$), only one of them, c_n , equals to logical TRUTH, showing the result of the m -counting gate. Depending on the number of input variable balls being logical TRUTH: n , exactly n collisions will happen during the computation, starting from the constant TRUTH ball T dynamically deflects the first ball it encounters. After each collision, the ball moving North-East changes its traveling route one line higher, as it would have done in a normal interaction gate 2a, so that after n collisions, the output variable $c_n = 1$. If there is no ball to deflect at all, the ball T remains its traveling route so that the output variable $c_0 = 1$.

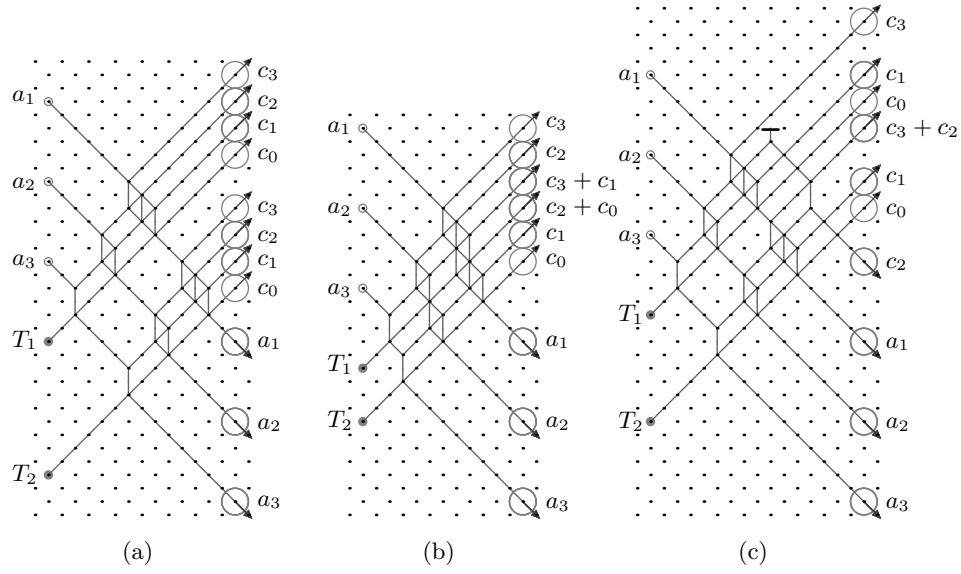


Fig. 5. Multiple m -counting gate in the Billiard Ball Model: (a) two 3-counting gates generate two sets of outputs ($c_0...c_3$); (b) move the constant TRUTH balls closer may generate outputs such as $c_3 + c_1, c_2 + c_0$; (c) however, to produce outputs in the form of $c_n + c_{n-1}$ needs the help of mirrors.

The feature of the m -counting gate that an exact copy of input variables is included in its output variables allows us to generate multiple instances of output variables ($c_0...c_m$) with only one set of input variables ($a_1...a_3$) and multiple instances of the constant TRUTH balls. As illustrated in Fig. 5a, the input variables ($a_1...a_3$) interact with the first constant TRUTH ball T_1 and generate a set of output variables ($c_0...c_3$), together with a copy of the input variables ($a_1...a_3$), which in turn interact with the second constant TRUTH ball T_2 , generating a second set of ($c_0...c_3$) as well as a copy of ($a_1...a_3$) again.

Relative positions of the multiple instances of output variables ($c_0...c_m$) are determined by relative positions of the constant TRUTH balls. Thus by adjusting relative positions of adjacent constant TRUTH balls, we can perform logical disjunction operation on certain output variables ($c_0...c_m$). As shown in Fig. 5b, while T_1 and T_2 move closer, traveling routes of the two sets of output variables ($c_0...c_3$) overlap, resulting in two outputs $c_3 + c_1$ and $c_2 + c_0$.

Since constant TRUTH balls cannot be too close to be next to each other, we cannot generate outputs in the form of $c_n + c_{n-1}$ directly. To do that, we need to use a mirror, as illustrated in Fig. 5c. Again, two constant TRUTH balls are used, producing two set of output variables ($c_0...c_3$). The mirror is placed to deflect one of the balls representing c_2 (the top one), which then collide with the other ball representing c_2 (the bottom one) and its traveling route end up the same as a ball representing c_3 , resulting in an output $c_3 + c_2$.

4 Binary Adders

The m -counting gate is very helpful in realizing binary adders with few mirrors or even no mirrors at all in the BBM. The simplest binary adder, the 1-bit half adder, resides in the 2-counting gate (Fig. 4b), with two input variables a_1 and a_2 being summands and the output variables c_1 and c_2 equal the SUM and CARRY value of the 1-bit half adder respectively. This circuit is constructed with no mirrors. Imagine if we were to build the same circuit using only interaction gates, there would have been no way to generate the SUM value without using fixed mirrors.

The binary 1-bit full adder, can be realized by four 3-counting gates and one extra mirror in the BBM, as illustrated in Fig. 6. It is actually a combination of the circuits in Fig. 5b and 5c, since if we consider input variables ($a_1...a_3$) as two summands and the CARRY-IN value of the adder, then the SUM value $S = c_3 + c_1$ and the CARRY-OUT value $C_{out} = c_3 + c_2$.

5 Parallel Binary (m,k) -counters

The concept of parallel binary (m,k) -counters was first introduced by Dadda [7] to construct parallel multipliers in 1965. An (m,k) -counter has m input variables, and it counts the number of input variables being logical TRUTHS: n . This concept is very similar to the above m -counting gates, with the difference that in (m,k) -counters the result n is recorded using k -bit binary numbers.

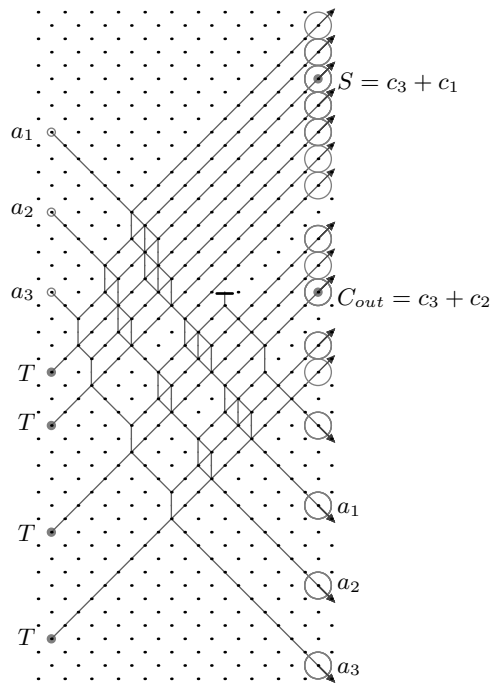


Fig. 6. The realization of a binary 1-bit full adder using four 3-counting gates and a fixed mirror in the Billiard Ball Model

Three of the simplest instances of (m,k) -counters: (1,1)-counter, (2,2)-counter and (3,2)-counter are already realized in previous sections, since they are equivalent to the 1-counting gate (Fig. 4a), the 2-counting gate (Fig. 4b) and the 1-bit full adder (Fig. 6).

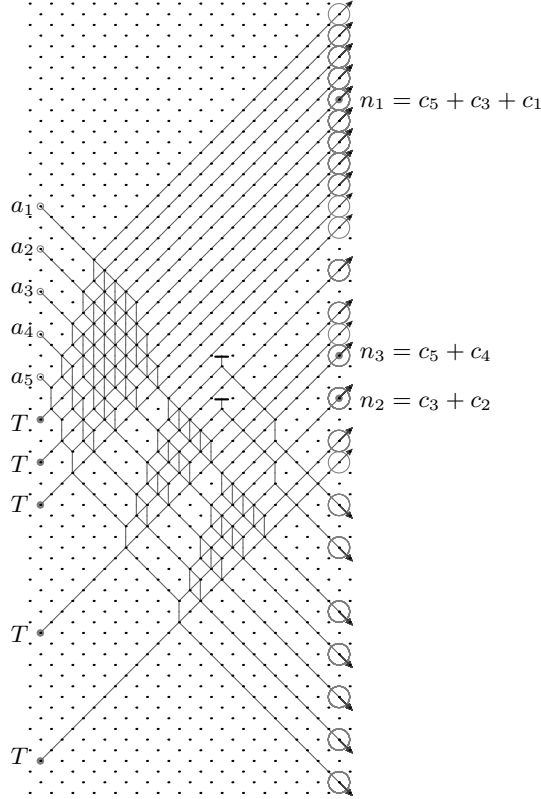


Fig. 7. A realization of the (5,3)-counter in the Billiard Ball Model, which uses five 5-counting gates and two fixed mirrors.

Here we demonstrate a realization of the (5,3)-counter (Fig. 7), where the counting result n is recorded in a 3-bit binary number: $n_3n_2n_1$. The realization uses the same techniques that are used in the 1-bit full adder. First, three 5-counting gates are used to producing an output of $n_1 = c_1 + c_3 + c_5$ by arranging three constant TRUTH balls in close distances. Then, two sets of output variables ($c_0...c_5$) are generated by another two 5-counting gates, followed by the pairs of c_4 s and c_2 s colliding between themselves, producing the outputs of $n_2 = c_2 + c_3$ and $n_3 = c_4 + c_5$.

Other (m,k) -counters ($m \geq 6$) can also use m -counting gates as building blocks, however, more mirrors may be placed to route signals in order to avoid unwanted collisions.

6 Discussions

The m -counting gate reveals one of the underlying logic naturally existing in the BBM. Thus we can build certain circuits without recourse to lots of fixed mirrors, which is not possible when using only primitive gates — the interaction gate and the switch gate. Although primitive gates are capable of universal computation, they do not naturally support disjunction operation or conjunction operation on more than two operands, which is why we often need multiple instances of such gates to realize a circuit and why we need a great many fixed mirrors to route signals. The switch gate itself is actually a good example of using mirrors to perform a disjunction operation.

On the contrary, the m -counting gate produces outputs with several conjunction and disjunction operations on the inputs, avoiding those mirrors that may have been needed if we use primitive gates. Take the realization of parallel binary counters, for example, as described by Swartzlander [13], conventionally an (m,k) -counter is implemented using a two-level gate network, which consists of 2^m-1 AND gates with m inputs as well as k OR gates with 2^m-1 inputs. The m -counting gate do not build circuits from the simplest AND gates and OR gates, rather, some of those two-level gate networks are outputs of the m -counting gate.

The m -counting gate has its limitation. First, though it is suitable as the building blocks to construct binary adders and parallel binary counters, it may not be that efficient when used to build other circuits. Second, the m -counting gate produce many outputs, and not all of them may be useful in constructing other circuits. Thus those signals that are no use can turn into obstacles and need to be clear out of the space, which may need more fixed mirrors. Nonetheless, we present a new way of exploit the Billiard Ball Model and the m -counting gate shows us some insights into how the BBM can compute in its natural fashion without mirrors.

We envisage that our findings can be useful in future research of collision-based computing since the BBM is used as a blueprint of computing schemes in novel computing substrates, for example, the light-sensitive sub-excitable Belousov–Zhabotinsky (BZ) medium. Adamatzky, de Lacy Costello and their colleagues have demonstrated both computationally and experimentally that mutual collisions of wave-fragments in the sub-excitable BZ medium can implement certain logical gates and basic operations [3, 8, 4, 14]. Further research in constructing complex circuits may require the implementation of “mirrors” by adjusting light intensity to control the traveling directions of wave-fragments, while our findings in the present paper show another approach to continue the research.

7 Acknowledgment

The author wishes to thank Andrew Adamatzky, Norman Margolus, Silvio Capobianco and Katsunobu Imai for their valuable comments and suggestions on this work.

References

- [1] Adamatzky, A.: Controllable transmission of information in the excitable media: the 2^+ medium. *Adv. Mater. Opt. Electron.* 5, 145–155 (1995)
- [2] Adamatzky, A. (ed.): *Collision-Based Computing*. Springer–Verlag, London, UK (2002)
- [3] Adamatzky, A.: Collision-based computing in Belousov–Zhabotinsky medium. *Chaos, Solitons & Fractals* 21(5), 1259–1264 (2004)
- [4] Adamatzky, A., de Lacy Costello, B.: Binary collisions between wave-fragments in a sub-excitable Belousov–Zhabotinsky medium *Chaos, Solitons & Fractals* 34(2), 307–315 (2007)
- [5] Berlekamp, E.R., Conway, J.H., Guy, R.K.: *Winning Ways for Your Mathematical Plays Volume 2: games in particular*. Academic Press (1982)
- [6] Christodoulides, D.N., Lederer, F., Silberberg, Y.: Discretizing light behaviour in linear and nonlinear waveguide lattices *Nature* 424, 817–823 (2003)
- [7] Dadda, L.: Some Schemes for Parallel Multipliers. *Alta Frequenza* 34, 349–356 (1965)
- [8] de Lacy Costello, B., Adamatzky, A.: Experimental implementation of collision-based gates in Belousov–Zhabotinsky medium *Chaos, Solitons & Fractals* 25(3), 535–544 (2005)
- [9] Fredkin, E., Toffoli, T.: Conservative logic. *Int. J. Theor. Phys.* 21(3–4), 219–253 (1982)
- [10] Margolus, N.: Physics-like models of computation. *Physica D* 10(1–2), 81–95 (1984)
- [11] Margolus, N.: Universal cellular automata based on the collisions of soft spheres. In: Adamatzky, A. (ed.) *Collision-Based Computing*. pp. 107–134. Springer–Verlag, London, UK (2002)
- [12] Sendiña-Nadal, I., Mihaliuk, E., Wang, J., Pérez-Muñuzuri, V., Showalter, K.: Wave Propagation in Subexcitable Media with Periodically Modulated Excitability *Phys. Rev. Lett.* 86, 1646–1649 (2001)
- [13] Swartzlander, Jr. E.E.: Parallel Counters. *IEEE Trans. Comput.* C-22(11), 1021–1024 (1973)
- [14] Toth, R., Stone, C., Adamatzky, A., de Lacy Costello, B., Bull, L.: Experimental validation of binary collisions between wave fragments in the photosensitive Belousov–Zhabotinsky reaction *Chaos, Solitons & Fractals* (In Press)
- [15] Wuensche, A., Adamatzky, A.: On Spiral Glider–Guns in Hexagonal Cellular Automata: Activator–Inhibitor Paradigm *Int. J. Modern Phys. C* 17(7), 1009–1025 (2006)