Computing with Virtual Cellular Automata Collider

Genaro J. Martínez
Escuela Superior de Cómputo
National Polytechnic Institute
Mexico City, Mexico
Unconventional Computing Centre
University of the West of England
Bristol BS16 1QY, UK
Email: genaro.martinez@uwe.ac.uk

Andrew Adamatzky
Unconventional Computing Centre
University of the West of England
Bristol BS16 1QY, UK
Email: andrew.adamatzky@uwe.ac.uk

Harold V. McIntosh
Departamento de Microcomputadoras
Universidad Autónoma de Puebla
Puebla, Puebla, México
Email: mcintosh@unam.mx

***Abstract*—We present computer models of nano-scale computing circuits based on propagation of localised excitations or defects in complexes of polymer chain rings. A cyclotron automata are sets of rings of one-dimensional array of finite states (cellular automata) which exhibits a wide range of travelling localisations (gliders). When information (e.g. values of logical variables) is encoded in the initial positions and velocity vectors of the gliders the cyclotron automata are becoming power abstract machines which execute high-performance computing. The computing is based on collisions between the mobile localisations. We present collisions that emulate basic types of interactions between localisations typical for spatially-extended non-linear media: fusion, particles, elastic collision, and soliton-like collision, all they implement basic computing primitives. Mobile localisations in complex one-dimensional cellular automata are compact sets of non-quiescent patterns translating along evolution space. These non-trivial patterns can be coded as binary strings (regular expressions) or symbols travelling along a one-dimensional ring, interacting with each other and changing their states, or symbolic values, as a result of interactions and computation.**

***Keywords*—collider; collisions; beam routing; cellular automata; localisations; computability**

I. Introduction

Unconventional computing — search for novel principles of efficient information processing and computation in physical, chemical and biological systems [8] – approach its state of maturity and started to sprout into numerous application domains. Cytoskeleton computing on actin filaments is amongst most prospective approaches: data-signals represented by localisations (excitations, energy densities, ionic clouds, defects) travelling along actin filaments implement computation via collisions [7]. Thus actin filaments a role of nano-scale collision-based computers.

A collision-based computer employs mobile compact finite patterns and mobile self-localised excitations to represent quanta of information in active non-linear mediums. Information values, e.g. truth values of logical variables, are given by either absence or presence of the localisations or other parameters of the localisations. The localisations travel in space and when collisions occur the result can be interpreted as computation. Any part of the medium space can be used as a wire. localisations can collide anywhere within a space sample, there are no fixed positions at which specific operations occur, nor location specified gates with fixed operations. The localisations undergo transformations, form bound states, annihilate or fuse when they interact with other mobile patterns. Information values of localisations are transformed as a result of collision.

Cellular automata (CA) are best mathematical machines to represent nano-scale polymer chain computers because of their architectural properties: array of finite state machines matches array of polymer units [7]. CA models of collision-based computing were introduced and extensively studied in [2]. In this book Tommaso Toffoli has introduced the concepts of *symbol super collider* [34].

In theoretical computer science there are many models of unconventional computing, which are based on processes in spatially extended non-linear media with different physics and non-classical logics. Most know examples include the reversible computing [10], [30], conservative logic [14], reaction-diffusion computers [5], quantum computers [15], Physarum computers [6], cellular automata computers [1], [16], [21], [33], [37], [26], optical or molecular computing [3], solitons or competing patterns computing [17], [2], [23], or hot ice computers [4].

Our previous results on CA collider-computers were published in [26], [24] (Computer Scientists Build Cellular Automaton Supercollider, Technology Review, Published by MIT, May 25, 2011. http://www.technologyreview.com/view/424096/computer-scientists-build-cellular-automaton-supercollider/). In present paper, we consider a number of designs to code localisations with different velocities and contact points. These localisations are represented as a set of regular expressions on a torus. Each localisation is coded as a regular expression and initialised on specific initial conditions. The computation is implemented via cyclical collisions on such a cyclotrons.

This computable device presented in the paper is inspired by high-performance sub-micron digital technologies [13], [3], [16], [33] and complex CA able of computation [31]; its logical schemes are based on symbol super collider theory [34] with an extension of the beams routing as a set of localisations [26].

The rest of the paper is organised as follows. Section II gives a brief introduction to CA notations. Section III introduces the concept of Toffoli colliders. Section IV shows how to recognise mobile localisation in a cyclotron with CA. Section V shows how basic computation can be done in the CA super-colliders. Outlines of further work are given in Section VI.

mds

June 3, 2015

II. Elementary cellular automata

One-dimensional CA is an array of *cells* x_i where $i \in Z$ (integer set) and each cell x takes a value from a finite alphabet Σ. A sequence of cells $\{x_i\}$ of a finite length n describes a string or *global configuration* c on Σ. A set of finite configurations is expressed as Σ^n. An evolution is represented by a sequence of configurations $\{c_i\}$ produced by the mapping $\Phi : \Sigma^n \rightarrow \Sigma^n$; thus the global relation is symbolised as:

$$\Phi(c^t) \rightarrow c^{t+1} \tag{1}$$

where t represents time and every global state of c is defined by a sequence of cell states. The global relation is determined over the cell states in configuration c^t updated at the next configuration c^{t+1} simultaneously by a local function φ as follows:

$$\varphi(x^t_{i-r}, \dots, x^t_i, \dots, x^t_{i+r}) \rightarrow x^{t+1}_i. \tag{2}$$

One-dimensional CA can be synthesised with two parameters (k, r) [36], where $k = |\Sigma|$ is the number of states, and r is the neighbourhood radius. This way, the class of elementary CA (ECA) is defined by parameters $(2, 1)$. There are Σ^n different neighbourhoods (where $n = 2r + 1$) and k^{k^n} different evolution rules.

We consider CA with periodic boundary conditions. A projection of these global states (rings) shows a better view for localisation interactions.

III. Toffoli's symbology

In the late 1970s Fredkin and Toffoli developed a concept of a general-purpose computation based on ballistic interactions between quanta of information that are represented by abstract particles [34]. The Boolean states of logical variables are represented by balls or atoms, which preserve their identity when they collide with each other. They came up with the idea of a billiard-ball model of computation, with underpinning mechanics of elastically colliding balls and mirrors reflecting the balls' trajectories [21].

With similar principles Toffoli presents in [34] the concept of *symbol super collider*. In this section, we will show its basic operations.

The following basic functions with two input arguments u and v can be expressed via collision between two localisations:

1) $f(u, v) = c$, fusion
2) $f(u, v) = u + v$, interaction and subsequent change of state
3) $f_i(u, v) \mapsto (u, v)$ identity, solitonic collision
4) $f_r(u, v) \mapsto (v, u)$ reflection, elastic collision

To map Toffoli's supercollider [34] onto a one-dimensional CA we use the notion of an idealised localisation $p \in \Sigma^+$ (without energy and potential energy). For our study, the localisation p is represented by a binary string (regular expression) of cell states for two complex ECA particularly: ECA rule 110 [29] and rule 54 [25]. ECA rule 110 is proved to be a computationally universal because it simulates a cyclic tag system [11], [37]. The ECA rule 54 is universal because it simulates functionally complete sets of logical gates [22]. Both rules compute and simulate via interacting locations.

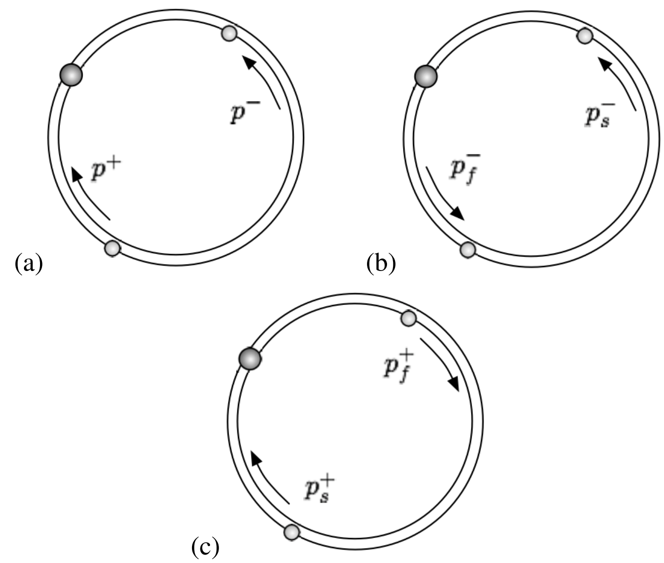

Fig. 1. Representation of abstract localisations in a one-dimensional CA ring beam routing.

Figure 1 shows two typical scenarios where localisations p_f and p_s travel in a CA cyclotron. The first scenario (Fig. 1a) shows two localisations travelling in opposite directions which then collide. Their collision site is shown by a dark circle as a contact point. The second scenario demonstrates a typical beam routing where a fast localisation (p_f) eventually catches up with a slow localisation (p_s) at a collision site (Fig. 1b). If the localisations collide like solitons [17], then the faster particle p_f simply overtakes the slower localisation p_s and continues its motion (Fig. 1c).

In essence, we will use these scenarios of the interactions between localisations to synchronise more than two or three localisations, connecting phase transitions between beam routings as a meta state of a finite machine [26].

IV. Description of cyclotrons

Typically, we can find all types of localisations manifested in CA gliders, including positive p^+, negative p^-, and patterns with neutral p^0 displacements. Neutral displacement of patterns in one dimension is related directly to *still life* configurations in two-dimensional CA Conway's Game of Life. Also we can compose compound and large localisations assembled from elementary localisations.

Let us consider the case where a quiescent state is substituted by cells synchronised together as a periodic background. This phenomenon is common in ECA rule 110. The rule's evolution space is dominated by a number of localisations emerging in various different orders. Consequently, the number of collisions between localisations increases quickly. Each localisation has a period, displacement, velocity, mass, volume, and phase. A full description of localisations in rule 110 is available at http://uncomp.uwe.ac.uk/genaro/rule110/glidersRule110.html, and for rule 54 visit http://uncomp.uwe.ac.uk/genaro/rule54/glidersRule54.html.

Number of collisions between localisations in rule 110 and 54 is determined by the localisation's' periods (phases) and a number of contact points. Of course, some localisations have an infinite number of extensions and thus the number of collisions is unlimited.

The number of collisions between localisation in rule 110 have a maximum level determined by the number of margins oms and ems. Thus, for an arbitrary localisation with oms contact points and other arbitrary localisation different from the first with ems contact points, we have the following number of collisions:

$$c \leq oms * ems \tag{3}$$

where c represents the maximum number of collisions between both localisations. Nevertheless, in some localisations the maximum level is not fulfilled. We have the exact number of collisions in the following equation:

$$c = |(oms_{p_i} * ems_{p_j}) - (oms_{p_j} * ems_{p_i})|. \tag{4}$$

where a pair p_i, p_j are localisations with different speed, for details please read [29]. Number of collisions between localisations in rule 54 follows a similar equation given in [25]. Fundamentally, these equations are determined by regular expression properties across its period, displacement, mass, and volume of these localisations.

A full description of regular expressions to code localisations in rule 110 and rule 54 is described in details in [29], [25].

Figure 2 displays a one-dimensional configuration where two localisations collide repeatedly and interact as solitons so that the identities of the localisations are preserved in the collisions. A localisation with negative direction p_F^- collides and overtakes a stationary localisation p_{C_1}. Figure 2a presents a whole set of cells in state 1 (dark points) where the periodic background makes it impossible to distinguish the localisations: p_F^- and p_{C_1}. However, we can apply a filter and thereby select localisations from their periodic background (Fig. 2b). Evolutions are simulated with *Discrete Dynamics Lab* (DDLab) [38] available in http://www.ddlab.org. Space-time configurations of a CA exhibiting a collision between localisations p_F^- and p_{C_1} are shown in Fig. 2c.

The number of collisions between p_F^- and p_{C_1} is reduced to four different collisions basically. But just one of the colllisions has the property as a soliton. So, the probability to get a solitonic reaction between these localisations is $\frac{1}{4}$. In this case, it is not difficult to synchronise different solitonic reactions with multiple localisations.

A number of solitonic reactions is proved to be useful in preserving information in the whole computable system [20].

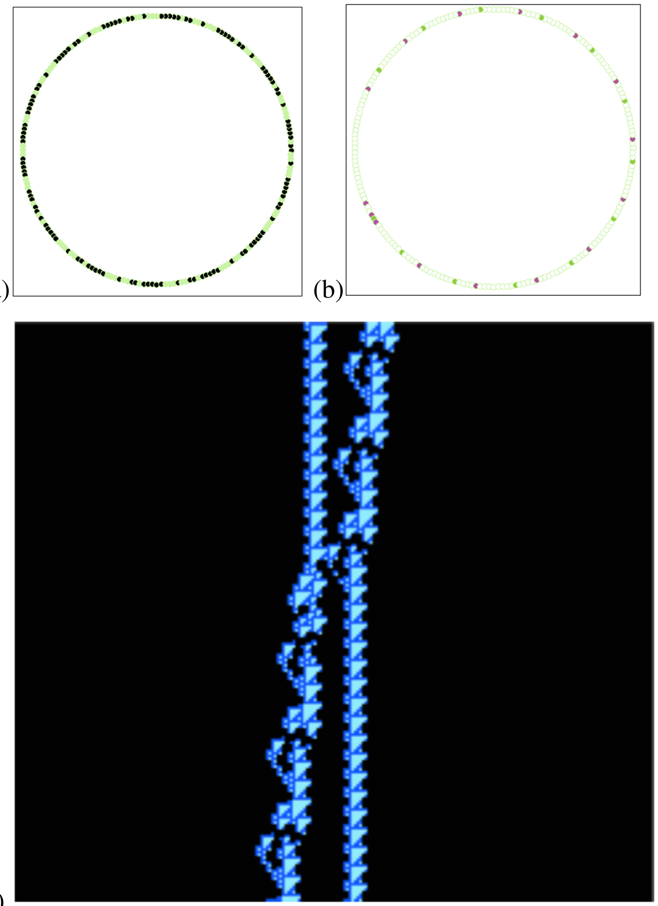
(a) (b) (c)

Fig. 2. Example of a soliton-type interaction between particles in ECA rule 110: (a) beam routing with all states, (b) same beam routing but with periodic background filtered out and showing exact state of localisations, (c) exact configuration at the time of collision.

V. IMPLEMENTING SOME BASIC FUNCTIONS

Toffoli considers a localisation travelling to collide with other localisation [34]. Here we will consider packages of localisations colliding simultaneously or sequentially. Thus a transition shall be mapped to a new meta state, where a set of localisations determines the new state on this machine. This way, we can develop finite machines where vertices are not just simple states, they are sets of strings that represent localisations turning inside cyclotrons. Consequently, transitions on these meta-vertices mean a change to other cyclotrons given for the sum of their collisions [26].

Let see a sample with multiple collisions. We can design a simple flip-flop pattern that is able of oscillate inside a cyclotron. In this case, we have the next following relation between five mobile localisations in ECA rule 110.

1) $p_F \leftarrow p_B = p_{D_1} + p_{A^2}$
2) $p_{A^2} \leftarrow p_{D_1} = p_B + p_F$

In this case, we have a diagram with two stages (cyclotrons). Specifically, we have five localisations, a slow localisation p_F that is reached for a fast localisation p_B, both with a positive orientation. Collision results yield three localisations travelling in opposite direction, two p_A localisations concatenated and represented as p_{A^2} and one p_{D_1} localisations. Next cyclotron stage is precisely a collisions between fast localisations p_{A^2} versus a slow localisation p_{D_1}, both traveling with a negative orientation. This way, we have a cycle of reactions (oscillator) that we could synchronize to simulate a very simple flip-flop pattern.

The cycle includes two cyclotrons, which, when connected one with another, represent a simple state machine. This machine has two meta-vertices (states) and the transition is determined by a contact point where gliders collide, as we can see in Fig. 3. Of course, this cycle can be seen as an attractor but made of collisions for this initial condition as well. The probability to get these reactions between $p_F \leftarrow p_B$ is $\frac{1}{4}$ and for $p_{A^2} \leftarrow p_{D_1}$ there is just one collision. Interaction between localisations $p_F \leftarrow p_B$ is illustrated in Fig. 4a. Interaction between localisations $p_{A^2} \leftarrow p_{D_1}$ is illustrated in Fig. 4b.

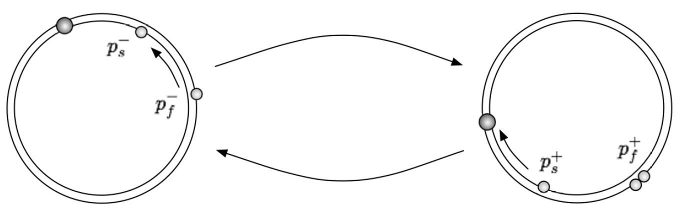

Fig. 3. Beam routing as cyclotrons working as a finite machine. In both cases fast particles reach to slow particles to change their state.

Flip-flop pattern is reached cycling these collisions. We can use a small cyclotron to simulate a basic flip-flop alone or, in this case, we can synchronise multiple reactions of the same kind inside a more large cyclotron, as show Fig 4c.

The regular expression to implement this operation is coded as follows. Here the symbol '$-$' means the concatenation operation, e is a string for the periodic background and localisations F and B are represented for a given phase.

$$e - F(\text{H,f}_1\text{\_1}) - e - B(\text{f}_1\text{\_1}) - e$$

Evolution has periodic boundary conditions and one exact distance preserves periodic collisions between such localisations. The multiple collisions are synchronised to construct a meta structure (Fig 4c). The regular expression to reproduce such a global behaviour is as follows:

$$(F(\text{H,f}_1\text{\_1}) - e - B(\text{f}_1\text{\_1}) - e)^*$$

the evolution coded in this initial condition needs 255 cells evolving in 480 generations. Of course, a change of codes, phases or distances will yield other collisions and therefore the whole system will change.

Collider implementation for this flip-flop pattern is displayed in Fig. 5. Delay between collisions' oscillation increases MOD 4 in factors of MOD 14 restricted by its periodic background. Initial condition employs 7,464 cells evolving over 25,000 generations, and 16 mobile localisations $p_F \leftarrow p_B$. Mobile localisations are situated just in the front of the view and they are moving circularly. History of trajectories and collisions are projected in three dimensions on the z-edge for

(a) (b) (c)

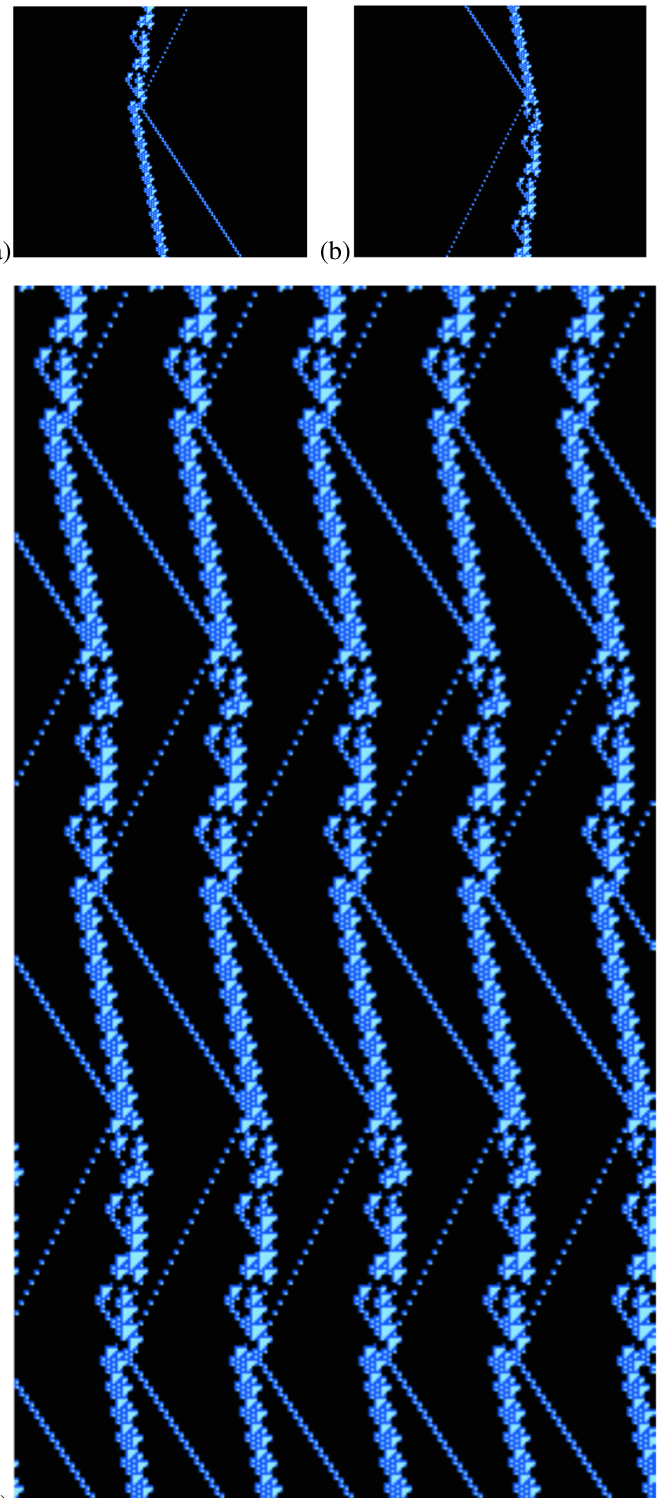

Fig. 4. Collisions between localisations: (a) $p_F \leftarrow p_B$ evolving with 260 cells in 220 generations, (b) $p_{A^2} \leftarrow p_{D_1}$ evolving with 260 cells in 220 generations, and (c) internal structure of collisions with periodic boundaries evolving with 255 cells in 480 generations.

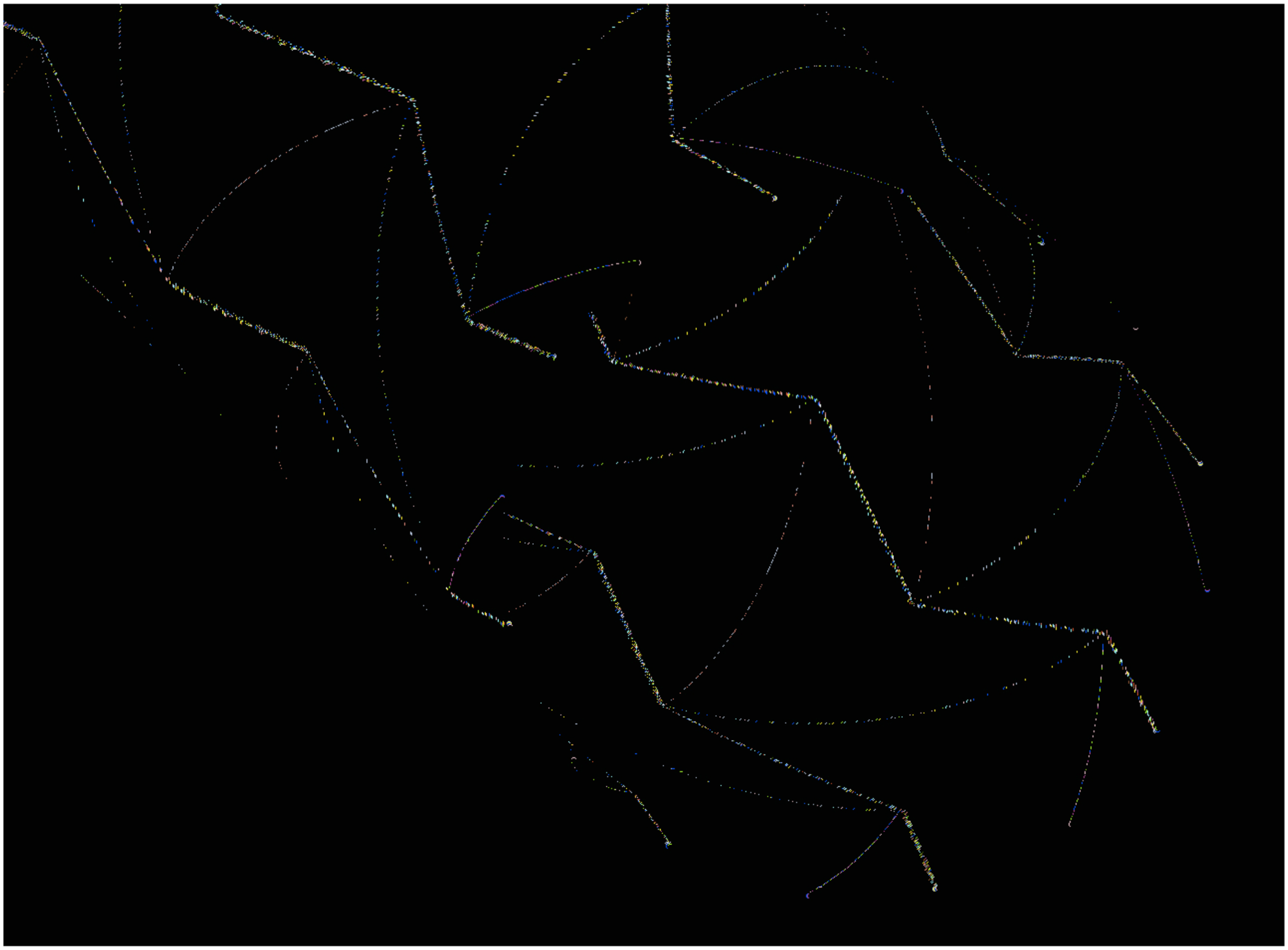

Fig. 5. Simple flip-flop oscillator implemented in a cyclotron with 7,464 cells in 25,000 generations. 16 mobile localisations $p_F \leftarrow p_B$ were coded.

a better view of dynamics. Of course, visibility of localisations is reduced to dots of colours as a consequence of the size of cyclotron.

Second sample is handled with mobile localisations in ECA rule 54. The problem is the design of a pattern that constructs sequentially the set of positive integers. Figure 6 simulates the construction of positive integers, starting with a triple collisions between two stationary localisations g_e and one mobile localisation $\overleftarrow{w}$, given the next sequence of collisions: $2g_e \leftarrow \overleftarrow{w}$. This initial collision yields a sequence of reactions that increases its intervals by space and structure. Every element of the positive integers is represented by the number of tiles in each stationary or mobile localisations. Of course, when these intervals expand the stationary localisation moves away gradually. Also, two mobile localisations emerge during each collision ($\overrightarrow{w}$ and $\overleftarrow{w}$), and they are controlled by an eater pattern. The eater pattern eliminate mobile localisations that are not necessary in the process.

VI. Final remarks

In the future work, we will implement a full equivalent Turing machine. In present design, we have considered acceleration of localisations, connection of two or three cyclotrons to synchronise main collisions in a central cyclotron. An interesting point here is how powerful is the concept of circular machines in unconventional computing devices, such as, circular Post machines [18], circular Turing machines [9], and cyclic tag systems [11]. Another field of future progress will be based around design and experimental laboratory implementation of real prototypes of collision-based computers made of polymer chains concatenated into rings and linked together. There is a bunch of problems to solve before practical implementation becomes possible: physical organisation of inputs and outputs, maintaining stable environmental conditions, life span of the polymer based computers. Actually, we are working to simulate the function of a full computable operation synchronazing multiple cyclotrons that will display an algorithm working completely.

References

[1] A. Adamatzky, *Computing in Nonlinear Media and Automata Collectives*, Institute of Physics Publishing, Bristol and Philadelphia, 2001.

[2] A. Adamatzky (ed.), *Collision-Based Computing*, Springer London, 2002.

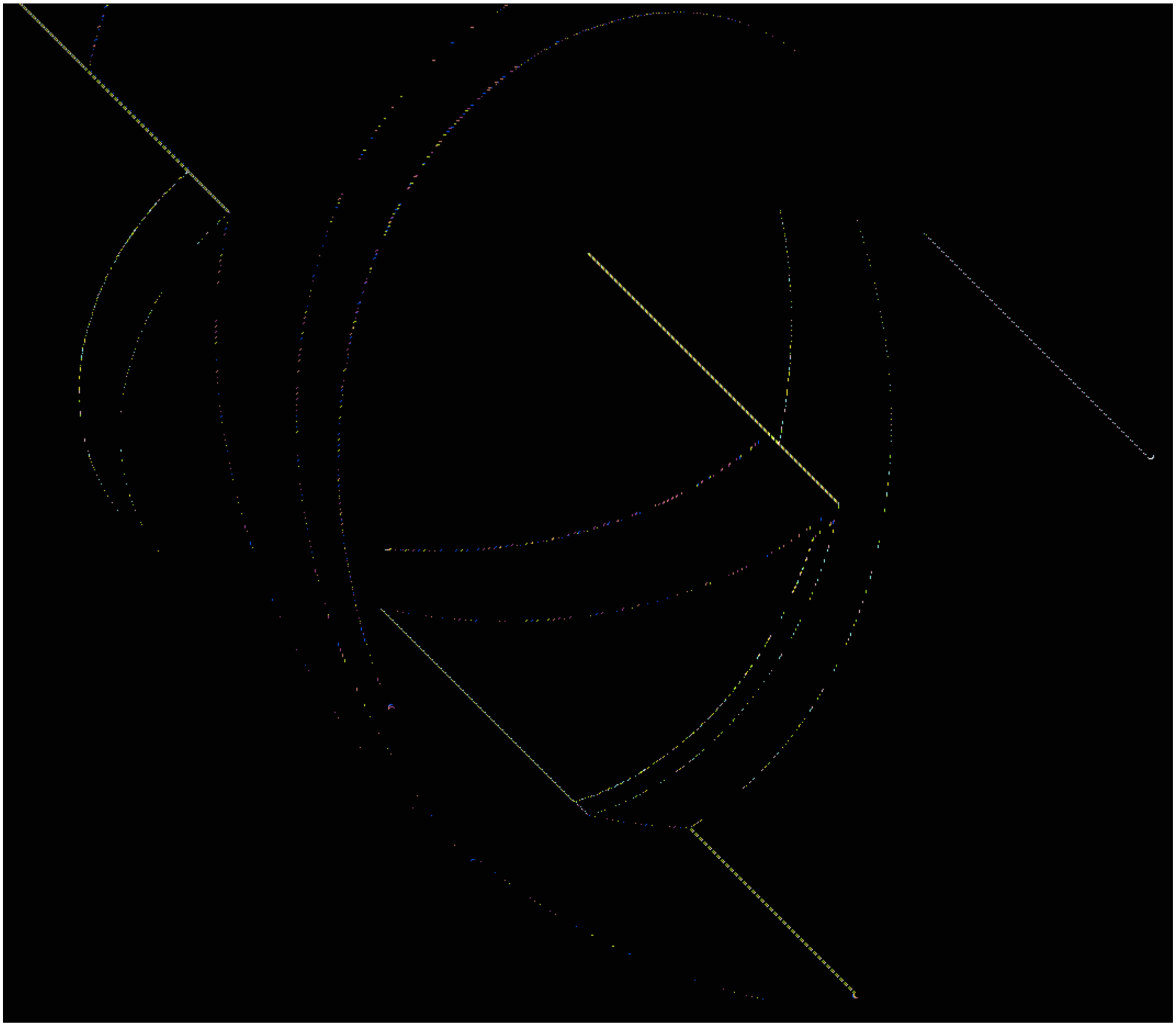

Fig. 6. This simulation displays mobile and stationary localisations colliding later of 20,000 generations. This evolution displays a simple function splitting a localisation in two new localisations w, and preserving a stationary localisation g_e. Performing a simple computable function which calculates the set of positive integers.

[3] A. Adamatzky, "New media for collision-based computing," In: A. Adamatzky (ed.), *Collision-Based Computing*, Springer London, chapter 14, pp. 411–442, 2002.

[4] A. Adamatzky, "Hot ice computer," *Physics Letters A* **374(2)** 264–271.

[5] A. Adamatzky, B. L. Costello, & T. Asai, *Reaction-Diffusion Computers*, Elsevier, 2005.

[6] A. Adamatzky, *Physarum Machines Computers from Slime Mould*, World Scientific Series on Nonlinear Science Series A, Volume 74, 2010.

[7] A. Adamatzky & R. Mayne, "Actin automata", *Int. J. Bifurcation and Chaos*, in press, 2015. Also arXiv:1408.3676 [cs.ET].

[8] A. Adamatzky, "Unconventional Computing", *Human Brain Projet magazine*, 2015.

[9] M. A. Arbib, *Theories of Abstract Automata*, Prentice-Hall Series in Automatic Computation, 1969.

[10] C. H. Bennett, "Logical reversibility of computation." *IBM journal of Research and Development* **17(6)** 525–532, 1973.

[11] M. Cook, "Universality in Elementary Cellular Automata," *Complex Systems* **15(1)** 1–40, 2004.

[12] L. J. Dixon, "Viewpoint: Particle Scattering Simplified." *Physics* **7** 107, 2014.

[13] E. Fredkin & T. Toffoli "Design Principles for Achieving High-Performance Submicron Digital Technologies," In: A. Adamatzky (ed.), *Collision-Based Computing*, Springer London, chapter 2, pp. 27–46, 2002.

[14] E. Fredkin, & T. Toffoli, "Conservative logic," *Int. J. Theoret. Phys.* **21** 219–253, 1982.

[15] N. A. Gershenfeld & I. L. Chuang, "Bulk spin-resonance quantum computation," *Science* **275(5298)** 350–356, 1997.

[16] A. J. G. Hey, *Feynman and computation: exploring the limits of computers*, Perseus Books, 1998.

[17] M. H. Jakubowski, K. Steiglitz, & R. Squier, "Computing with Solitons:

A Review and Prospectus," *Multiple-Valued Logic* **6(5-6)** 439–462, 2001.

[18] M. Kudlek & Y. Rogozhin, "Small Universal Circular Post Machine," *Computer Science Journal of Moldova* **9(25)** 34–52, 2001.

[19] Y. Lu, Y. Sato, & S. Amari, "Traveling Bumps and Their Collisions in a Two-Dimensional Neural Field," *Neural Computation* **23(5)** 1248–1260, 2011.

[20] G. J. Martínez, A. Adamatzky, F. Chen, & L. Chua, "On Soliton Collisions between localisations in Complex Elementary Cellular Automata: Rules 54 and 110 and Beyond," *Complex Systems* **21(2)** 117–142, 2012.

[21] N. Margolus, "Physics-like models of computation," *Physica D* **10(1-2)** 81–95, 1984.

[22] G. J. Martínez, A. Adamatzky, & H. V. McIntosh, "Phenomenology of glider collisions in cellular automaton Rule 54 and associated logical gates," *Chaos, Solitons and Fractals* **28** 100–111, 2006.

[23] G. J. Martínez, A. Adamatzky, K. Morita, & M. Margenstern, "Computation with competing patterns in Life-like automaton," In: A. Adamatzky (ed.) *Game of Life Automata*, Springer, chapter 27, pp. 547–572, 2010.

[24] G. J. Martínez, A. Adamatzky, & H. V. McIntosh, "Computing on Rings," In: H. Zenil (ed.), *A Computable Universe: Understanding and Exploring Nature as Computation*, World Scientific Press, chapter 14, pp. 283–302, 2012.

[25] G. J. Martínez, A. Adamatzky, & H. V. McIntosh, "Complete Characterization of Structure of Rule 54," *Complex Systems* **23(3)** 259–293, 2014.

[26] G. J. Martínez, A. Adamatzky, & C. R. Stephens, "Cellular Automaton Supercolliders," *International Journal of Modern Physics C* **22(4)** 419–439, 2011.

[27] H. V. McIntosh, *One Dimensional Cellular Automata*, Luniver Press, UK, 2009.

[28] M. Minsky, *Computation: Finite and Infinite Machines*, Prentice Hall, 1967.

[29] G. J. Martínez, H. V. McIntosh, J. C. S. T. Mora, & S. V. C. Vergara, "Determining a regular language by glider-based structures called phases fi_1 in Rule 110," *Journal of Cellular Automata* **3(3)** 231–270, 2008.

[30] K. Morita, "Reversible computing and cellular automata - A survey," *Theor. Comput. Sci.* **395(1)** 101–131, 2008.

[31] G. J. Martínez, J. C. Seck-Tuoh-Mora, & H. Zenil, "Computation and Universality: Class IV versus Class III Cellular Automata," *Journal of Cellular Automata* **7(5-6)** 393–430, 2013.

[32] N. Margolus, T. Toffoli, & G. Vichniac, "Cellular-Automata Supercomputers for Fluid Dynamics Modeling," *Physical Review Letters* **56(16)** 1694–1696, 1986.

[33] T. Toffoli, "Non-Conventional Computers", In: J. Webster (ed) *Encyclopedia of Electrical and Electronics Engineering* **14**, Wiley & Sons, 455–471, 1998.

[34] T. Toffoli, "Symbol Super Colliders," In: A. Adamatzky (ed) *Collision-Based Computing*, chapter 1, pp. 1–23, 2002.

[35] J. von Neumann, *Theory of Self-reproducing Automata* (edited and completed by A. W. Burks), University of Illinois Press, Urbana and London, 1966.

[36] S. Wolfram, *Cellular Automata and Complexity*, Addison-Wesley Publishing Company, 1994.

[37] S. Wolfram, *A New Kind of Science*, Wolfram Media, Inc., Champaign, Illinois, 2002.

[38] A. Wuensche, *Exploring Discrete Dynamics*, Luniver Press, UK, 2011.