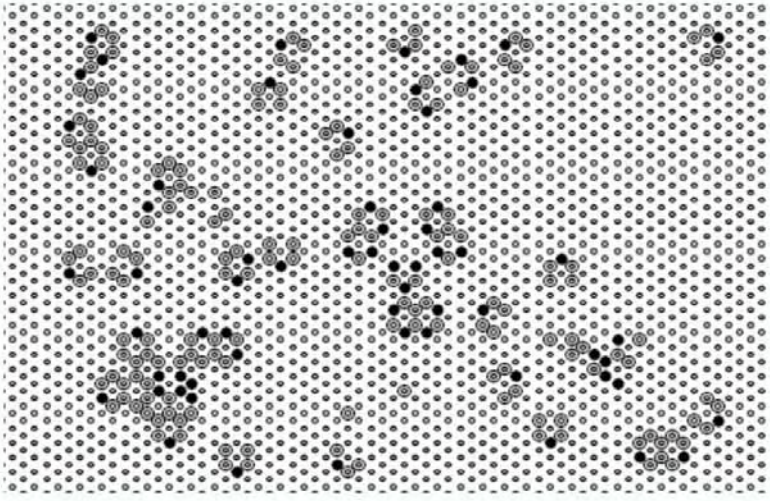
CHRISTOF TEUSCHER AND ANDREW ADAMATZKY
EDITORS

UNCONVENTIONAL COMPUTING 2005
FROM CELLULAR AUTOMATA TO WETWARE



Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 0955117038 ISBN-13: 978-0955117022

Luniver Press

Proceedings of the 2005 Workshop on Unconventional Computing

From Cellular Automata to Wetware

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Christof Teuscher and Andrew Adamatzky
Editors

Proceedings of the 2005 Workshop on Unconventional Computing

From Cellular Automata to Wetware

Luniver Press

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Published by Luniver Press
Frome BA11 6TT United Kingdom

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Teuscher C. and Adamatzky A. (Eds.)
Proceedings of the 2005 Workshop on Unconventional Computing: From Cellular Automata to Wetware (Luniver Press, Beckington, 2005)

ISBN 978-0-9551170-2-2

2nd edition

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Preface

For more than half a century, the von Neumann computer architecture (i.e., the stored program concept) and the abstract concept of the Turing machine have largely dominated computer science in many variants and refinements. One might certainly ask — "how the future of these two major paradigms will look"? Whereas it is unlikely that they will disappear, there seems to be a growing need for novel and unconventional computing paradigms to face specific needs and challenges in new fields and application domains. This quest is also motivated by the observation that fundamental progress in several fields of computer science sometimes seems to stagnate. For example, one of the keys to machine intelligence is computers that learn, and we are still just scratching the surface of this problem. Another example is our inability to create and program complex systems that is simply not keeping up with the desire to solve complex problems.

The goal of the workshop is to bring together a multidisciplinary core of scientists who are working in the field of unconventional computing, to provide a common ground for dialog and interaction, to highlight the latest advances, and to discuss the main directions for the future.

From over thirty papers submitted to the workshop, we selected eight contributions which got highest praises from referees and which, when considered together, give a good sampling of hot research points in the vibrant field of unconventional computing.

Thus, papers by Gorecki and Gorecka (Chemical Wave Based Programming in Reaction-Diffusion Systems) and by Harding and Miller (Evolution In Materio: Evolving Logic Gates in Liquid Crystal) discuss results of cutting edge research in designing logical systems in non-linear spatially extended systems, Belousov-Zhabotinsky excitable chemical medium, and liquid crystal.

Oya, Asai and Amemiya (A Single-Electron Reaction-Diffusion Device for Computation of a Voronoi Diagram) and Suzuki, Takayama, Motoike and Asai (Striped and Spotted Pattern Generation on Reaction-Diffusion Cellular Automata – Theory and LSI Implementation) demonstrate how to overcome the drawbacks (i.e., the speed of computation is somewhat limited by the slowly travelling diffusive and phase waves in the chemical medium) of reaction-diffusion computers by designing networks of single-electron oscillators and silicon large-scale integrated circuits.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

A novel approach to designing cellular automata with possibly useful computational properties is tackled by Turner and Stepney (Rule Migration: Exploring a Design Framework for Modelling Emergence in CALike Systems). More classical topics of non-classical computation (i.e., DNA computing and quantumcomputation) are studied by Ibrahim, Tsuboi, Ono and Khalid (Experimental Implementation of Direct-Proportional Length-Based DNA Computing for Numerical Optimization of the Shortest Path Problem) and by Lebar Bajec and Mraz (Towards Multi-State Based Computing Using Quantum-Dot Cellular Automata). A non-trivial solution to a classical computer science problem is given by Salzberg (A Reflexive Busy Beaver Problem).

Finally, to spice things even more up, we have also decided to invite two non-referred guest presentations: Rambidi, Ulyakhin, and Tsvetkov (Several Remarks on Practical Implementation of Image Processing by Chemical Reaction-Diffusion Media) will highlight their recent results in image-processing in Belousov-Zhabotinsky chemical systems, and there will be an excurse in artificial chemistries applied to computation by Matsumaru, Centler and Dittrich (Chemical Organization Theory as a Theoretical Base for Chemical Computing).

Last but not least, we would like to express our gratitude to the members of the workshops program committee, who vigorously reviewed every submission on time, and to the ECAL2005 organisers for giving us the opportunity to hold this workshop!

September 2005

Christof Teuscher and Andy Adamatzky,
Workshop Chairs,
Los Alamos and Bristol

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Program Commitee

Andy Adamatzky, University of the West of England, UK
Tetsuya Asai, Hokkaido University, Japan
Stefania Bandini, University of Milano-Bicocca, Italy
Bastien Chopard, University of Geneva, Switzerland
Peter Dittrich, Friedrich-Schiller-University Jena, Germany
Enrico Formenti, University of Nice-Sophia Antipolis, France
Jerzy Gorecki, Polish Academy of Science, Poland
Andrew Ilachinski, Center for Naval Analyses, USA
Martin Kutrib, University of Giessen, Germany
Norman Margolus, MIT Artificial Intelligence Laboratory, USA
Jacques Mazoyer, Ecole Normale Superieure de Lyon, France
Julian F. Miller, University of York, UK
Jonathan W. Mills, Indiana University, USA
Kenichi Morita, Hiroshima University, Japan
Nicholas G. Rambidi, Moscow State University, Russia
Chris Salzberg, University of Tokyo, Japan
Ken Steiglitz, Princeton University, USA
Susan Stepney, University of York, UK
Oliver Steinbock, Florida State University, USA
Christof Teuscher, Los Alamos National Laboratory, USA
Tommaso Toffoli, Boston University, USA
Hiroshi Umeo, Osaka Electro-Communication University, Japan
Burton Voorhees, Athabasca University, Canada
Joerg R. Weimar, Technical University Braunschweig, Germany
Thomas Worsch, Universitaet Karlsruhe, Germany
Andrew Wuensche, DDLab and UWE, UK
Klaus-Peter Zauner, University of Southampton, UK

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Table of Contents

Unconventional Computing 2005

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Chemical programming in reaction-diffusion systems

Jerzy Gorecki[1] and Joanna Gorecka[2]

[1] Institute of Physical Chemistry, Polish Academy of Science,
Kasprzaka 44/52, 01-224 Warsaw, Poland.
Faculty of Mathematics and Natural Sciences,
Cardinal Stefan Wyszynski University
Dewajtis 5, 01-815 Warsaw, Poland
and
ICM UW, ul. Pawinskiego 5A, 02-106 Warsaw, Poland
[2] Institute of Physics, Polish Academy of Science,
Al. Lotnikow 36/42, 02-668 Warsaw, Poland

Abstract. The paper is concerned with the chemical programming. We demonstrate that information processing properties of a reaction diffusion medium can be controlled by a chemical signal composed of excitation pulses. The simplest "programmable" chemical signal processing device is a memory that can be loaded or erased by a pulse arriving from appropriate input channels. We discuss how chemical memory can be incorporated into more complex information processing devices like signal switches or programmable signal channels.

1 Introduction

An extensive discussion on the applications of spatially distributed excitable chemical systems for information processing can be found in the literature [1–3]. One of possible translations of chemical dynamics into the language of informatics comes form the digital approach to computing and from the binary logic [4–6]. Two logical states can be easily identified in an excitable chemical system. The logical "true" state corresponds to concentrations of reagents that characterize excitation, whereas the logical "false" state is related to the non-excited state. A pulse of excitation traveling in a chemical medium may be regarded as a propagating bit of information. Within such interpretation no information is recorded or processed when the system remains in the stationary

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

state. This property is plausible for those who try to find links between the chemical information processing and the information processing in biological systems [7], because the medium acts in a pretty obvious way - it does nothing if it is not excited by some external factors. If external excitations act on the system the information about them spread out in the medium in the form of diffusion driven pulses of concentration. The sequence of pulses forms a chemical signal.

Information coded in pulses of concentration can be processed in areas of space, where signals interact. It has been demonstrated that many operations on chemical signals can be easily performed in a nonhomogeneous chemical medium. Such a medium is usually build of active areas, where the system is excitable and passive areas, where the system has a single, strongly attracting stationary state so the excitations are rapidly dumped. One can easily build channels in which signals propagate using stripes of active medium surrounded by the passive one. The excitations do not decay instantaneously in the passive areas because the reagents can diffuse. So if two active areas are separated by a stripe of passive medium then a pulse of excitation propagating in one active part can cross the stripe and excite the other active part if the stripe is sufficiently narrow. In the experimental practice a given pattern of active and passive areas can be created by introducing an immobilized catalyst in a nonhomogeneous way [8–10] or (for the case of photosensitive reaction) by a non-uniform illumination [14].

A few simple rules describe the behavior of pulses in a typical excitable medium. An excitable system is characterized by a refractory time. It means that the system has to relax for a period of time before it can be re-excited. The time after which a subsequent excitation is successful depends on the strength of excitation. The form of this dependence leads to many interesting effects related to the transformation of signal frequency after crossing a barrier [11–13]. The refractory stage of an excitable medium is responsible for the annihilation of colliding pulses. After head on collision of two pulses in a channel the excitation disappears because the medium behind the pulses is in a refractory state so the excitation introduced by one of colliding pulses does not excite the medium behind the other one.

As mentioned above an excitation may cross a barrier of passive medium if the barrier is narrow enough. The width of a passive barrier that can be crossed by an excitation depends on many factors including the the barrier geometry, the angle between the normal to the barrier and the wave vector of a pulse, and, the number of pulses arriving at the barrier. For the information processing it is important that the width of a stripe of passive medium can be selected such that a pulse propagating

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

perpendicularly to the stripe crosses it whereas the excitation introduced by a pulse propagating parallel to the barrier does not. If a junction is non-symmetrical geometrically, for example rectangular at one end and triangular at the other, it can transmit signals in one direction only because the excitation coming from the rectangular part is stronger than that from the triangular one. This property was used to construct the chemical information diode which was one of the first signal processing devices analyzed in numerical simulations and realized in experiments [6, 15, 8, 9, 16]. More complex signal processing reactors, like the logical gates [4–6], the coincidence detectors [6, 17, 18] and switches of the direction of a traveling pulse [17, 18] have been also studied and most of these devices have been already tested experimentally. Using them as building blocks one can construct reactors, that allow for even more complex signal processing, like for example counting the number of pulses arriving to a given point of space [19].

However all devices described above can be classified as the instant machines [20] that perform a specific operation completely defined by the geometrical arrangement of active and passive areas. In order to modify the functions they perform one has to "re-wire" the circuit, i.e. change the geometrical distribution of active and passive areas for the required propagation of pulses.

In experiments with Ru-catalized Belousov-Zhabotinsky(BZ) reaction such re-wiring can be introduced by time dependent illumination because the excitation pulses propagate toward the dark region [21, 22]. Such technique can be used to program a signal processing medium, but it uses an external factor (illumination) and requires a feedback between monitoring and external control. In this paper we discuss how the information processing performed by a reaction-diffusion medium can be controlled and programmed by a chemical signal composed of pulses.

2 Read-write memory cell

The simplest element of chemical information processing system that can be programmed by a chemical signal is a memory cell in which information may be written-in, storied, read-out and, if necessary, re-written. One of the realizations of a chemical memory is based on an observation that a pulse of excitation can rotate on a ring shaped excitable area indefinitely long if the reactants are continuously supplied and the products removed [23]. Therefore, a ring of excitable medium can be regarded as a memory cell capable of storing one bit of information. If there is a rotating pulse the state of memory corresponds to the logical 1, if there is no pulse - the state of memory corresponds to the logical 0. The idea of

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

a ring as a chemical memory was investigated by the Kyoto group [6, 24]. Here we present a further development of the memory ring studied by us both in numerical simulations based on Rovinsky-Zhabotinsky model of ferroine catalyzed BZ reaction [25, 26] and in the experiments with the Ru-catalized BZ reaction [27].

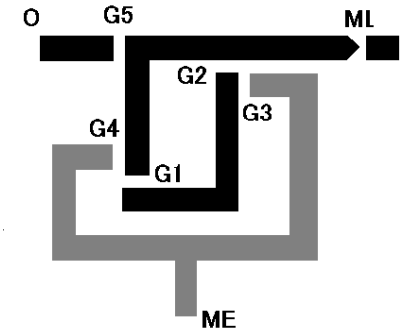


Fig. 1. The geometrical distribution of active channels in read-write memory cell. The memory ring and the loading channel (ML) are black, the erasing channels are gray (ME).

Fig. 1 shows the geometrical arrangements of passive and active areas in the considered read-write memory cell. Both black and gray channels are excitable and the white parts are passive. The black areas define the memory ring, the output channel O and the loading channel ML. The gray segments mark the erasing channels. The widths of passive gaps (G1-G5) separating excitable areas are selected such that a pulse of excitation propagating perpendicularly to the gap can excite the active area on the other site of the gap, but the gap is impenetrable for pulses propagating parallel to it. The memory cell can be loaded by a pulse of excitations coming from ML channel. In a loaded cell the pulse of excitation rotates anticlockwise and the information about the state of memory is periodically sent out as pulses of excitation via the output channel O. An erasing pulse can enter the memory ring through gaps G3 and G4. If the memory is loaded it annihilates with the loading pulse. The excitation of the memory ring introduced by an erasing pulse propagates clockwise. It does not excite the output channel and it is not able to cross the gaps G1 and G2. As the result an erasing pulse does not load the memory ring.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

The memory presented in Fig. 1 is more reliable than the one discussed before [24] because it uses a twin erasing channel. In the case of a single erasing channel the erasing pulse does not change the state of a memory if the memory ring in front of the erasing channel is in the refractory state. As shown in Fig. 2 the erasing channels can be also placed inside the memory ring. This memory design is more compact than the previous one but it needs the third dimension to operate.

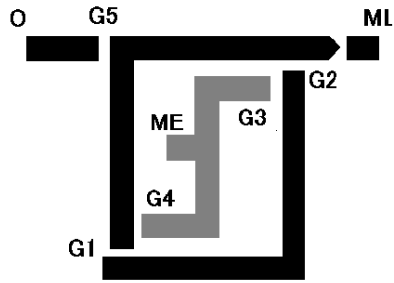


Fig. 2. An alternative construction of read-write memory cell with the erasing channels placed inside the memory ring.

3 The use of memory cell in chemical "programming"

By the chemical programming we understand the controlling of information processing performed by a reaction diffusion medium with the use of a chemical signal send from a number of control channels. As expected we use the read-write chemical memory is an important element of our programmable systems, because its state can be easily controlled by external pulses. We show that chemical programming can be realized by signals sent through the loading and erasing inputs of memory cells.

Figures 3 and 4 show the idea of switchable signal channel. Three segments of active medium A, B and C linked by the signal diodes D1 and D2 form a channel transparent to pulses traveling from the input I to the output O if the memory M is not loaded. The pulses propagating in the channel A-B-C do not interfere with the memory because they move parallel to the gap G. When the memory M is activated it periodically sends pulses to the segment B. They enter the segment B through the gap G and after entering they propagate parallel to the passive area in

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

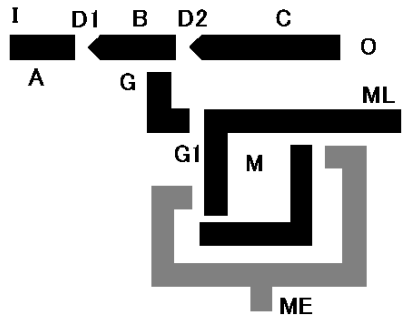


Fig. 3. The geometry of active areas in a switchable unidirectional signal channel. The channel is formed by A, B and C segments. The transmission through the segment B is controlled by the state of memory M.

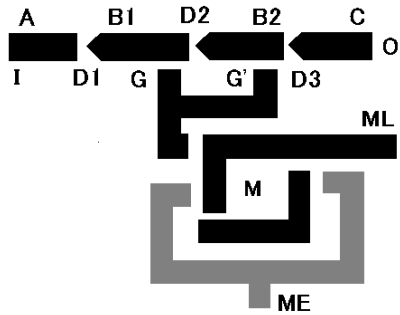


Fig. 4. The geometry of more reliable unidirectional signal channel. The number of channels where pulses from the memory interact with the signal is doubled (now the transmission through the segments B and B' is controlled by the state of memory M).

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

the diode D2. Therefore they do not excite the segment C. The pulses generated by the memory propagate toward the diode D1 and die at it. If there is a signal coming from the input I then it is annihilated within the section B by these pulses. As the result the system presented in Fig. 3 is a unidirectional signal channel that can be open or closed depending on the state of the memory M, thus it is programmable by pulses coming from the memory load (ML) and the memory erase (ME) channels.

However the switchable signal channel shown in Fig. 3 is not fully reliable. Let us imagine a pulse propagating along the channel A-B-C. If an erasing pulse from the memory appears at the gap G at the moment when the pulse is in the segment B in front of the gap then the erasing pulse will not cross the gap. To avoid such situations we can increase the number of segments where annihilation can occur. More reliable switchable signal channel with two segments controlled by the same memory cell is shown in Fig. 4.

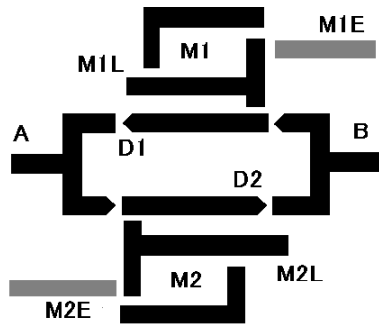


Fig. 5. The geometry of active areas in a switchable bidirectional signal channel. Depending on the states of memories M1 and M2 the signal may be transmitted in both directions, in a selected one or it may be stopped completely.

Linking two switchable channels oriented in opposite directions, as illustrated in Fig. 5, one can construct a switchable signal channel that can be completely closed, open in one of the directions or completely open depending on the state of the memory cells M1 and M2. The channel is programmed by the signals loading memories (M1L, M2L) and erasing them (M1E, M2E).

The programmable signal separator is shown in Fig. 6. The signal arrives from the output A, separates and enters two parallel signal channels. If both memories M1 and M2 are not loaded than the signal is transmitted through the diodes and arrives at both outputs B1 and B2.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

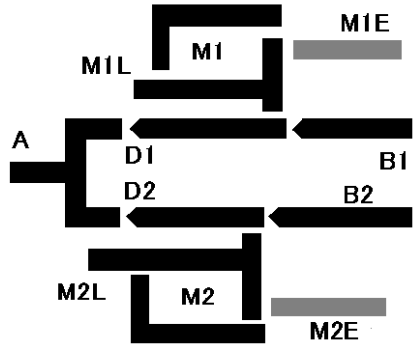


Fig. 6. The geometry of active areas in a signal separator. Depending on the states of memories M1 and M2 the signal coming from the segment A may be directed into both B1 and B2 channels, into a selected one or it can be completely stopped.

However, a loaded memory closes the corresponding channel. Thus, depending on the states of memories the signal coming from the segment A may be directed into both B1 and B2 channels, into the selected one or it can be completely stopped. If the outputs are attached to other signal processing devices we obtain a circuit that can perform required signal transformations in a programmable way.

In biological systems the knowledge and experience of the past are frequently forgotten. Such effects can be also realized in chemical signal processing devices. Fig. 7 shows a possible construction of a self-erasing memory cell. The self erasing channel (SEC) marked light gray is added to the memory ring shown in Fig. 1. SEC links the memory output with the erasing channel via the diode D. Without this diode the erasing signal would be sent to the memory output. A loaded memory may be externally erased by a signal from ME channel or it erases itself automatically a certain time after the first output signal appears. This time is of course regulated by the length and excitability of the SEC. The presented memory can be used to construct systems that are able to learn, but keep their knowledge for a period of time only.

Swichable signal channels may be also controlled by output signals. This gives us an opportunity to build self-learning chemical systems. A simple scheme of such system is drawn in Fig. 8. It consists of a single input channel A and two output channels B1 and B2. LFD and HFD denote low- and high- frequency detectors respectively. Such detectors can be easily constructed using the coincidence of the consecutive pulses [28,

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

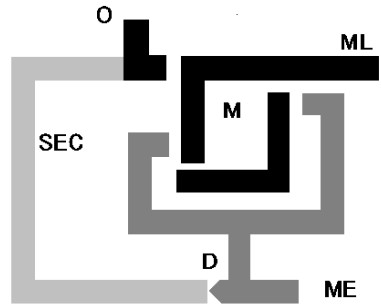


Fig. 7. The geometrical distribution of active channels in a self-erasing memory cell. The memory ring and the loading channel (ML) are black, the erasing channels are gray (ME) and the channel (SEC) that directs the output signal to the erasing channels is light gray.

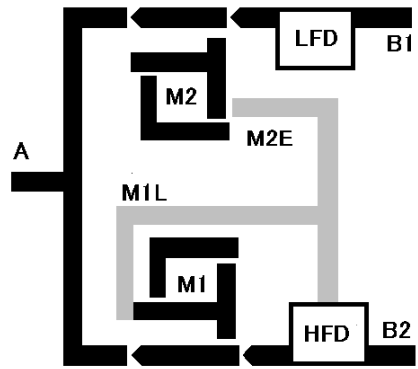


Fig. 8. The geometry of a switchable channel controlled by the frequency of input signal. If the frequency of signal is high the signal is directed to the channel B1, if the frequency is low the signal is directed to the channel B2.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

29]. In Fig. 8 we have schematically shown the connections between the memory cells M1 and M2 and the high frequency detector. The connections between memories and LFD are symmetrical. Let us assume that a high frequency signal arrives at the input. HFD detects it and sends a pulse that erases M2 and loads M1. As the result the channel B2 is closed and the channel B1 open. All signals are transmitted through B1 up to the moment when a low frequency signal appears. When LFD detects it will send appropriate loading and erasing signals that reverse the states of memories. Now the channel B2 is open and B1 closed up to the moment when a high frequency signal appears. The system shown in Fig. 8 exhibits a hysteresis of the transmitting channel with respect to the signal frequency.

In this paper we described how a read-write memory cell can be used to program information processing in a reaction-diffusion medium. The proposed geometrical arrangement of active and passive areas in switchable signal channels seems generic and it can be adopted for other excitable systems with the normal dispersion relation [30], if the widths of channels and distances in passive gaps are properly selected. There is no doubt that switchable signal channels can play an important role in neural networks build of excitable medium as they allow to establish or cut connections between nodes. As shown the connections may be also controlled by the output signals (Fig. 8) that allows to construct chemical devices characterized by self-learning.

4 Acknowlgement

The authors are grateful to professor Kenichi Yoshikawa from The Department of Physics of The University of Kyoto for fruitful discussions and help in experiments. The research was supported by the Polish State Committee for Scientific Research project 1 P03B 035 27.

References

1. A. Adamatzky and B. De Lacy Costello, Phys. Rev. E **66** (2002) 046112.
2. N. Rambidi, BioSystems **64** (2002) 169.
3. T. Ichino, Y. Igarashi, I.N. Motoike and K. Yoshikawa, J. Chem. Phys. **118** (2003) 8185.
4. A. Toth and K. Showalter, J. Chem. Phys. **103** (1995) 2058.
5. O. Steinbock, P. Kettunen and K. Showalter, J. Phys. Chem. **100** (1996) 18970.
6. I. N. Motoike and K. Yoshikawa, Phys. Rev. E **59**(1999) 5354.
7. H. Haken, *Brain Dynamics* Springer Series in Synergetics, Springer-Verlag Berlin and Heidelberg (2002).

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

8. K.Agladze, R. R. Aliev, T. Yamaguchi and K. Yoshikawa, J. Phys. Chem. **100** (1996) 13895.
9. T. Kusumi, T. Yamaguchi, R. R. Aliev, T. Amemiya, T. Ohmori, H. Hashimoto and K. Yoshikawa, Chem. Phys. Letters **271** (1997) 355.
10. K. Suzuki, T. Yoshinobu and H. Iwasaki, Jpn. J. Appl. Phys. **38** (1999) L345 ; K. Suzuki, T. Yoshinobu and H. Iwasaki, J. Phys. Chem. A **104** (2000) 6602.
11. J. Sielewiesiuk and J. Gorecki, Phys. Rev. **E 66** (2002) 016212.
12. J. Sielewiesiuk and J. Gorecki, J. Phys. Chem. **A 106**(2002) 4068.
13. K. Suzuki, T. Yoshinobu and H. Iwasaki, J. Phys. Chem. A **104** (2000) 5154.
14. K.Agladze, A. Toth, T. Ichino and K. Yoshikawa, J. Phys. Chem. A **104** (2000) 6677.
15. K. Yoshikawa, I. N. Motoike and K. Kajiya, ISICE Trans. Electron. **E80-C** (1997) 931.
16. T. Yamaguchi, T. Kusumi, , R. R. Aliev, T. Amemiya, T. Ohmori, M.Nakaiwa, K. Urabe, S. Kinugasa, H. Hashimoto and K. Yoshikawa, *Acta Chimica Hungarica - Models in Chemistry* **135**(1998) 401.
17. J. Sielewiesiuk and J. Gorecki, Acta Physica Polonica B **32** (2001) 1589.
18. J. Sielewiesiuk and J. Gorecki, J. Phys. Chem. A **105** (2001) 8189.
19. J. Gorecki, K. Yoshikawa and Y. Igarashi, J. Phys. Chem. A **107** (2003) 1664.
20. N.G. Rambidi, BioSystems **44** (1997) 1.
21. T. Sakurai, E. Mihaliuk, F. Chirila, and K. Showalter, Science **296** (2002) 2009.
22. Adamatzky A, Chaos, Solitons Fractals, **21** (2004) 1259.
23. A. Lazar, Z. Noszticzius, H-D. Forsterling and Z. Nagy-Ungvrai, Physica D **84** (1995) 112.
24. I. N. Motoike, K. Yoshikawa, Y. Iguchi and S. Nakata, Phys. Rev. E **63** (2001) 036220.
25. A. B. Rovinsky and A. M. Zhabotinsky, J. Phys. Chem. **88** (1984) 6081.
26. A. B. Rovinsky, J. Phys. Chem. **90** (1986) 217.
27. J. Gorecka and J. Gorecki, *Read-write memory as a step towards chemical programming*, prepared for publication.
28. J. Gorecka and J. Gorecki, Phys. Rev. E **67** (2003) 067203.
29. I. N. Motoike and K. Yoshikawa, Chaos, Solitons and Fractals **17** (2003) 455.
30. O. Steinbock, Phys. Rev. Lett. **88** (2002) 228302.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

A single-electron reaction-diffusion device for computation of a Voronoi diagram

Takahide Oya, Tetsuya Asai, and Yoshihito Amemiya

Graduate School of Information Science and Technology, Hokkaido University,
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814 Japan
oya@sapiens-ei.eng.hokudai.ac.jp

Abstract. In this paper, we propose a novel single-electron device for computation of a Voronoi diagram (VD). A cellular-automaton model of VD formation [18] was used to construct the device that consisted of three layers of a 2-D array of single-electron oscillators. Through extensive numerical simulations, we show pattern formation on the proposed device and demonstrate the VD computation.

1 Introduction

Computation of a Voronoi diagram (VD) is one of the typical problems in computer science, and VDs are used in graphics, statistics, geography and economics [1] and [2]. The key feature of VD construction is a partition of two- or three-dimensional space on a sphere of influences generated from a given set of objects, points, or arbitrary geometrical shapes. Therefore, VDs are widely applied in data analysis; for example they have been used for identifying star and galaxy clusters [3], market analyses [2], and for modeling gravitational influences [4], cell and tissue growth [5, 6], ecological competition [7], crystal growth [8], molecules [9], animal skin pigmentation [10, 11], and geographical surfaces [12]. Its many computer science applications include pattern recognition via computation of a skeleton [13], path planning in the presence of obstacles [14], and computer graphics or computer-generated images [15]. Voronoi tessellation is also extensively used in the field of computational geometry where, it is applied to solve variations on the nearest neighbor problem [16, 17].

The reaction-diffusion (RD) algorithm (a cellular automaton (CA) model) for computation of a VD was first constructed in [18, 19]. The

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

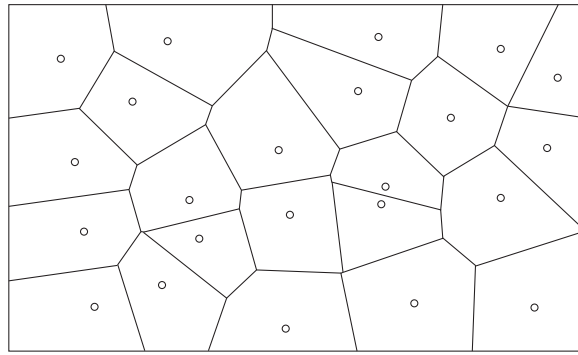


Fig. 1. Voronoi diagrams of planar points.

authors have proposed a single-electron device that is analogous to RD systems where electron-tunneling phenomena propagate on a 2-D array of single-electron oscillators [20]. This electrical RD device consists of a two-dimensional array of single-electron nonlinear oscillators that are combined with one another through diffusive coupling. In this paper, by employing the RD algorithm for the VD computation, we construct an architecture of low-power ultrafast VD computers by means of single-electron circuits.

2 Voronoi Diagram

Given a non empty finite set $\mathbf{P}$ of planar points, a planar VD of the set $\mathbf{P}$ is a partition of the plane into such regions, that for any element of $\mathbf{P}$, a region corresponding to a unique point p contains all those points of the plane that are closer to p than to any other node of $\mathbf{P}$. A unique region

$$vor(p) = \{z \in \mathbf{R}^2 : d(p,z) < d(p,m) \forall m \in \mathbf{R}^2 \, m \neq z\}$$

assigned to point p is called a Voronoi cell. The boundary of the Voronoi cell of a point p is built of segments of bisectors separating the point p and its geographically closest neighbors from $\mathbf{P}$. The union of all boundaries of the Voronoi cells comprises the VD (Fig. 1):

$$VD(P) = \cup_{p \in \mathbf{P}} \partial vor(p).$$

The lower bound of a VD computation (of a d-dimensional set of n points) is $\Theta(n)$ and the worst-case complexity is $\Theta(n^{\lceil d/2 \rceil})$ [17]. The

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

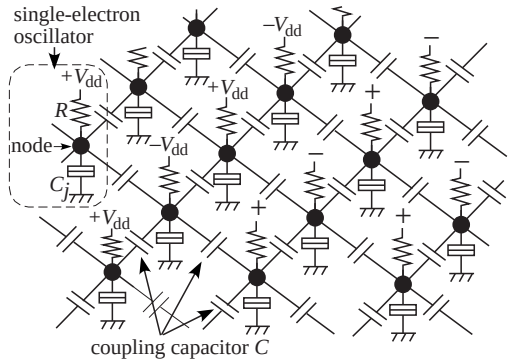


Fig. 2. Circuit configuration of single-electron reaction-diffusion device [20].

Fig. 3. Traveling nonlinear voltage wave that is generated by the original SERD device (simulated). The device has 100 × 100 oscillators. Simulated with parameters: tunneling junction capacitance $C_j = 1$ aF, tunneling junction conductance = 1 μS, high resistance $R = 137.5$ MΩ, coupling capacitance $C = 1$ aF, bias voltage $V_{dd} = 16.5$ mV, and zero temperature. In this figure, high voltage is bright and low voltage is dark [20].

CA and RD algorithms exploit the natural parallelism of the problem; namely the distance between neighboring points is represented by the time of a wave-front traveling in space, therefore the time complexity of VD computation in the RD medium is determined by the maximum distance between two geographically neighboring points of a given set, which in turn is limited by a diameter D of the given planar set. That is, the worst-case complexity of RD computation of VD is $\Theta(D)$, independent of dimension or number of points. Assuming that we can pack given set preserving topological relationships between its elements, the computational lower bound will be $\Theta(\sqrt[d]{n})$, in the d-dimensional case.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

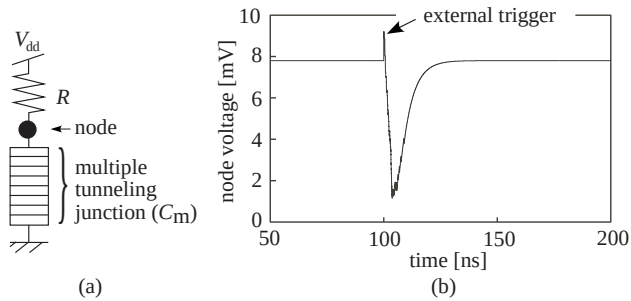


Fig. 4. Single-electron oscillator with multiple-tunneling junction. (a) circuit configuration and (b) its operation (simulated). Simulated with parameters: tunneling junction capacitance $C_{\rm m} = 10$ aF (500 aF/50 junctions), tunneling junction conductance $= 5$ μS, high resistance $R = 20$ GΩ, bias voltage $V_{\rm dd} =$ 7.8 mV, and zero temperature.

3 Single-Electron Circuits for computation of Voronoi Diagram

To compute a VD, we propose the use of single-electron reaction-diffusion (SE-RD) devices. The original SE-RD device consists of arrayed single-electron oscillators and can imitate the operation of chemical RD systems [20]. For this paper, we have improved the structure of the SE-RD device for computing a VD and have simulated its operations.

Figure 2 shows the original SE-RD device. The main component is a single-electron oscillator that consists of a tunneling junction C_j and a high resistance R connected in series at a node and biased by a positive voltage $V_{\rm dd}$ or a negative one $-V_{\rm dd}$. It has voltage $V_{\rm node}$ of the node, and $V_{\rm node}$ shows the excitatory oscillation that is indispensable for imitating RD systems [20].

To compute a VD with RD systems, nonlinear waves that travel at a constant speed are necessary [18, 19]; i.e., the wave-fronts must be smooth and their speed must be constant. The original SE-RD device can generate nonlinear voltage waves. However, the original device is not suitable for computing a VD because the wave-fronts are not smooth and their speeds are not constant, as shown in Fig. 3. The tunneling probability of each electron at the oscillators is the reason why the waves can not travel at a constant speed. To make the wave fronts smooth and the speed of the waves constant, new oscillators in which the probability is averaged are necessary. In this paper, we propose using new oscillators with multiple-tunneling junction (MTJ) as shown in Fig. 4. The MTJ

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

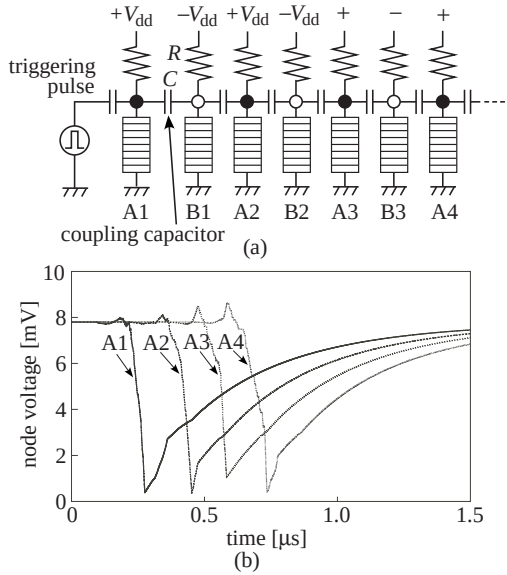


Fig. 5. One-dimensional chain of improved oscillators. (a) circuit configuration and (b) its operation (simulated). Simulated with parameters: tunneling junction capacitance $C_{\rm m} = 10$ aF (500 aF/50 junctions), tunneling junction conductance $= 5$ μS, high resistance $R = 20$ GΩ, coupling capacitance $C = 2.2$ aF, bias voltage $V_{\rm dd} = 7.8$ mV, and zero temperature.

oscillator consists of a multiple-tunneling junction $C_{\rm m}$ that has n tunneling junctions and a high resistance R connected in series at the node and biased by $V_{\rm dd}$. It has a voltage $V_{\rm node}$ that shows the excitatory oscillation like the original oscillator does. There are many tunneling junctions in the oscillator, so the tunneling probability is averaged. As a result, $V_{\rm node}$ changes smoothly as shown in Fig. 4 (b). The improved SE-RD device has such oscillators.

Adjacent oscillators have to be coupled with a capacitor for the voltage waves to travel on the improved device. We simulated a one-dimensional chain of improved oscillators and confirmed its operation as shown in Fig. 5. The oscillators that in the figure are denoted by A1, A2,..., with their nodes represented by closed circles are connected to their adjacent oscillators through intermediary oscillators that are biased by a negative voltage $-V_{\rm dd}$ (these are denoted by B1, B2, ..., with their nodes represented by open circles) and coupling capacitors C (Fig. 5 (a)). When electron tunneling occurs in an oscillator in this structure, the node voltage of the oscillator decreases gently, and this induces

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

electron tunneling in an adjacent intermediary oscillator. The induced tunneling changes the node voltage of the intermediary oscillator from low to high, and this induces electron tunneling in an adjacent oscillator. As a result, changes in node voltage that are caused by the electron tunneling are transmitted from one oscillator to another along the oscillator chain (Fig. 5 (b)). Note that the voltage waves travel at almost constant speed. This is because the tunneling probability is averaged over all oscillators.

An improved SE-RD device can be constructed by connecting oscillators into a network by means of intermediary oscillators and coupling capacitors as shown in Fig. 6. Each oscillator is connected to its four adjacent oscillators by means of four intermediary oscillators and coupling capacitors. Nonlinear voltage waves travel on the improved device at a constant speed as shown in Fig. 7. We can compute the VD by using information on collision points of the nonlinear waves for which we use CA model [18, 19] for finding collision points. According to Refs. [18, 19], in the CA model a cell that connects eight adjacent cells changes its state according to the states of the adjacent cells. The cell state transition rule is as follows:

$$x^{t+1} = \begin{cases} \beta, & \text{if } x^t = \bullet \text{ and } 1 \leq \sigma(x)^t \leq 4 \\ \alpha, & \text{if } x^t = \beta \text{ and } 1 \leq \sigma(x)^t \leq 4 \\ x^t, & \text{otherwise} \end{cases} \tag{1}$$

where x is the state of the middle cell, t is the time step, $\bullet$ is a resting cell, α is colored precipitate, β is reagent and $\sigma(x)^t$ is the number of β cells in the eight adjacent cells. In this model, the collision points are memorized as the precipitate of reagents.

To apply this rule to our device, we consider the use of single-electron threshold detectors, specifically the single-electron boxes (SEB) that we have proposed as logic gate devices [21, 22]. The SEB consists of a single-electron trap (two identical tunneling junctions C_j, connected in series, a capacitor C_{L} and a bias voltage V_{dd}) as shown in Fig. 8 (a). This circuit has a hysteretic sawtooth function for V_{dd} as shown in Fig. 8 (b). We make use of this characteristic for threshold operation. Here we consider the threshold operation for computing a VD based on the CA model. We assume the threshold value that is the number of β cells in the eight adjacent cells to be 4.5; i.e., no electron tunneling occurs in the SEB when the node voltages of four or fewer adjacent oscillators are changed by electron tunnelings in the oscillators. On the other hand, electron tunneling occurs in the SEB when the node voltages of five or more adjacent oscillators are changed. In addition, we can find the collision points by comparing the state of the center oscillator with the state of the SEB threshold detector. To compare the states, we use the

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

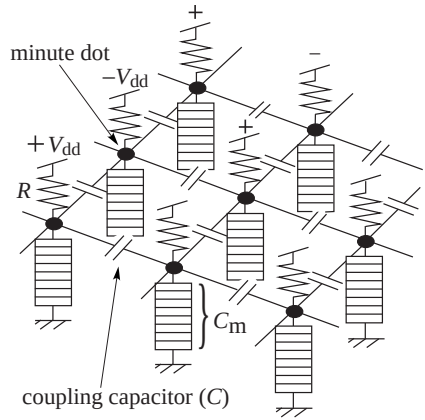


Fig. 6. Improved two-dimensional RD device consisting of network of improved single-electron oscillators. Each oscillator is connected with 4 neighboring oscillators by means of four intermediary oscillators and coupling capacitors.

Fig. 7. Traveling nonlinear wave that is generated by the improved SE-RD device (simulated). 50×50 oscillators are placed in the device. This simulation used the same parameters as in Fig. 5 (b). In this figure, high voltage is bright and low voltage is dark.

SEB with the threshold set to 1.5; i.e., no electron tunneling occurs in the SEB when electron tunnelings occur in both the above SEB and the center oscillator.

Figure 9 shows an improved device with three layers for computing a VD. The top layer ((a) in Fig. 9) is the improved SE-RD device shown in Fig. 6. The middle layer ((b) in Fig. 9) is the first logic layer of SEB threshold detectors. Here, the SEB that is biased by the negative voltage $-V_{b1}$ ((e) in Fig. 9) are placed directly under the oscillators biased by $+V_{dd}$ (oscillator 9 in Fig. 9) and connects to the eight adjacent oscillators of the top layer (oscillators 1–8 in Fig. 9 (d)); i.e., the SEB accepts eight signals from the eight oscillators as inputs. The bottom layer ((c) in Fig.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

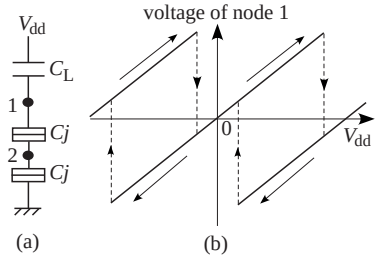


Fig. 8. Single-electron box. (a) Circuit configuration and (b) its operation [21, 22].

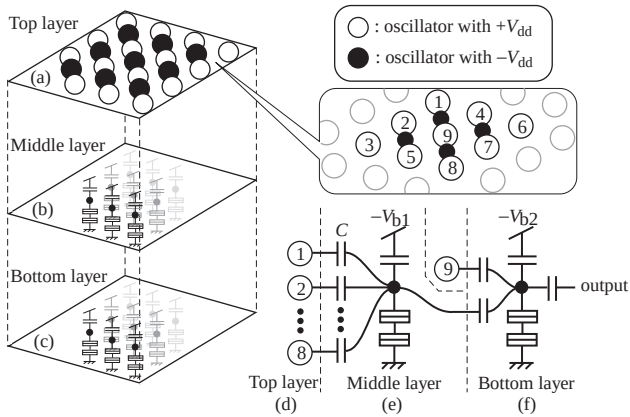


Fig. 9. Improved device that has three layers for computing a VD. The top layer is the device shown in Fig. 6. The middle layer is the first logic layer with SEB threshold detectors. The bottom layer is the second logic layer and it produces the VD.

9) is the second logic layer. Here, the SEB that is biased by the negative voltage $-V_{b2}$ ((f) in Fig. 9) connects to the oscillator 9 and the SEB in the second layer ((e) in Fig. 9); i.e., the second SEB accepts two signals from the oscillator 9 and the SEB in the second layer as inputs. The bottom layer produces the output; i.e., its output is used to draws the VD.

4 Simulation Results

We tested the device's operation by computer simulation. Figures 10, 11 and 12 show the simulation results. Figure 10 shows the density of

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

node voltages on the top layer. A bright color means the node voltage is high. A dark color means the node voltage is low. Figures 11 and 12 show the voltages on the middle layer and on the bottom layer. In Fig. 11, 'A' indicates the wave-front in the top layer, 'B' indicates the wave-front in the middle layer, and 'C' indicates collision points. In this simulation, we triggered three oscillators of the top layer as planar points for a VD. Nonlinear voltage waves traveled at a constant speed and gave the data to the middle and bottom layers. In the middle layer, the SEBs changed their node voltage when five or more oscillators of the upper eight oscillators changed their node voltage. Wave-fronts in the top layer had four or fewer oscillators that changed their voltages. As a result, traveling nonlinear waves in this layer (B in Fig. 11) followed the waves in the top layer (A). When wave 'A' collided with other waves in the top layer, the collision points had five or more oscillators that changed their voltages. Therefore, wave 'B' in this layer overtook 'A' and collided with other waves just like spanning a valley with a bridge (C). In the bottom layer, the SEBs changed their node voltages when both the voltage of the oscillators in the top layer and the SEBs in the middle layer are low. Namely, traveling waves that did not collide were memorized by the bottom layer as a high voltage. When the nonlinear waves of the top layer collided with each other, the voltages of the collision points in the top were low and the node voltages of the SEBs in the middle were high. As a result, the node voltages of the SEB in the bottom were kept low. Therefore, the bottom layer memorized the result of computing the VD. Figure 13 shows simulation results of computing a VD with five planar points by using proposed device. The results imply that we can compute a VD with any planar points on the device.

5 Summary

We proposed a single-electron reaction-diffusion devices for computing a Voronoi diagram. The novel SE-RD device consists of three layers. The top layer is an improved SE-RD device in which nonlinear voltage waves are generated and travel, and the middle and bottom layer are threshold detectors. The operations of the middle and bottom layer are based on the CA model [18, 19]. The bottom layer outputs the results of computing a VD by using data from the top and middle layers.

In some simulations, the constructed VDs had noise, as can be seen in Fig. 12 and the bottom snapshots of Fig. 13. The reason may be the tunneling probability of the bottom layer or the way to apply the CA model to the middle and bottom layers. However, we were able to compute a whole VD with our device even though there was some noise.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Other points that we must consider are the size of the proposed device and the number of coupling capacitors between each layer. There are $i \times j$ oscillators with $+V_{\rm dd}$ and $(i-1) \times (j-1)$ oscillators with $-V_{\rm dd}$ in the top layer, and $i \times j$ SEBs in the middle and bottom layers, respectively. If we fabricated an oscillator, for example, measuring 100 nm × 100 nm [23], the top layer would be 0.01 ij μm^2. A top layer with 100 × 100 oscillators would be only 100 μm^2. The device would not take up a large amount of space it would be at most 3 $\times 0.01 \times ij$ μm^2. In addition, if we could construct the layers on top of one another, the area would be just 0.01 ij μm^2. On the other hand, there are eight coupling capacitors per SEB between the top and middle layer. Therefore there are 8 ij capacitors between two layers. If the top layer had 100 × 100 oscillators, there would be 80000 capacitors. In this work, we have assumed that the size of such capacitors is zero, which is not realistic. Thus, we must consider a simpler structure as a future work.

Acknowledgments

The authors wish to thank Prof. Andrew Adamatzky of University of the West of England for his invaluable discussions and suggestions during our research. This study was partly supported by the Industrial Technology Research Grant Program in '04 from the New Energy and Industrial Technology Development Organization (NEDO) of Japan.

References

1. Klein, R.: *Concrete and Abstract Voronoi Diagrams*. Springer-Verlag, Berlin (1990).
2. Okabe, A., Boots, B., Sugihara, K., and Chiu S. N.: *Spatial Tesselations: Concepts and Applications of Voronoi diagrams*. John Wiley and Sons, Chichester (2000).
3. Pásztor L.: Partition based point pattern analysis methods for investigation of spatial structure of various stellar populations. Astronomical Data Analysis Software and Systems III, eds. Crabtree, D. R., Hanisch, R.J. and Barnes J. (ASP Conference Series) **61** (1994).
4. Drysdale, S.: Voronoi Diagrams: Applications from Archaology to Zoology. `http://www.ics.uci.edu/ eppstein/gina/scot.drysdale.html`.
5. Honda, H. and Eguchi, G. J.: How much does the cell boundary contract in a monolayered cell sheet. J. Theor. Biol. **84** (1980) 575–588.
6. Blackburn, C. G. and Dunckley, L.; The application of Voronoi Tessellations in the Development of 3D Stochastic Models to Represent Tumour Growth. Zeitschrift fur Angewandte Mathematik und Mechanik. **76** (1996) 335–338.
7. Barlow, G. W.: Hexagonal territories. Animal Behavior. **22** (1974) 876–878.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

8. Hargittai, I. (Ed): *Symmetry: Unifying Human Understanding*. Pergamon Press, New York (1986)
9. Pontius, J. Richelle, J., and Wodak, S. J.: Deviations from standard atomic volumes as a quality measure for protein crystal structures, J. Mol. Biol. **264** (1996) 121-136.
10. Walter, M., Fournier, A., and Reimers, M.: Clonal mosaic model for the synthesis of mammalian coat pattern *Proc. Graphics Interface*, `http://www.graphicsinterface.org/proceedings/1998/118/`.
11. Meinhardt, H.: *The alogorithmic beauty of Sea Shells*. Springer, New York (1995).
12. Bray, N. H., Anderson, J. B., Devine, J. D., and Kwasnik, J. M.: Topological Properties of Random Crack Networks. Mathematical Geology. **8** (1976) 617–626.
13. Adamatzky, A. and De Lacy Costello, B. P. J.: Collision-free path planning in the Belousov-Zhabotinsky medium assisted by a cellular automaton. Naturwissenschaften. **89** (2002) 474–478.
14. Earnshaw, R. A. (Ed): *Theoretical Foundations of Computer Graphics and CAD*. NATO ASI series, series F: Computer and Systems Sciences. **40** Springer -Verlag, Berlin (1998).
15. Deussen, O., Hiller, S. van Overveld, C., and Strothotte, T.: Floating Points: A Method for Computing Stipple Drawings. Computer Graphics Forum. **19** (2000) 41–50.
16. Graham, R. and Yao, F.: A whirlwind Tour of Computational Geometry. American Mathematical Monthly. **97** (1990) 687–701.
17. Preparata, F. P. and Shamos, M. I.: *Computational Geometry*. Springer Verlag, Berlin (1985).
18. Adamatzky, A.: Reaction–diffusion algorithm for constructing discrete generalized Voronoi diagram. Neural Netw. World. **6** (1994) 635–643.
19. Adamatzky, A.: Voronoi–like partition of lattice in cellular automata. Mathl. Comput. Modelling. **23** (1996) 51–66.
20. Oya, T., Asai, T., Fukui, T., and Amemiya, Y.: Reaction-diffusion systems consisting of single-electron circuits. Int. J.Unconventional Computing. **1** (2005) 177–194.
21. Oya, T., Asai, T., and Amemiya, Y.: Single-electrron logic device with simple structure. Elec. Lett. **39** (2003) 965–967.
22. Oya, T., Asai, T., Fukui, T., and Amemiya, Y.: A majority-logic device using an irreversible single-electron box. IEEE Trans. Nanotech. **2** (2003) 15–22.
23. Oya, T., Asai, T., Fukui, T., and Amemiya, Y.: A majority-logic nanodevice using a balanced pair of single-electron boxes. J. Nanosci. and Nanotech. **2** (2002) 333–342.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

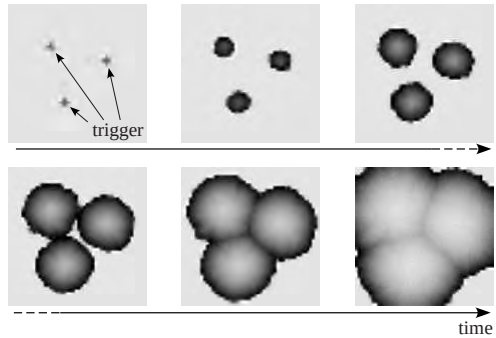


Fig. 10. Expanding circular pattern in the top layer of the device. Snapshots for six time steps. The simulation used the same parameters as in Fig. 5 (b) for the simulation.

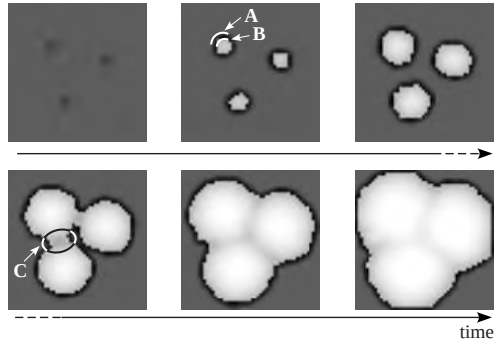


Fig. 11. Expanding circular pattern in the middle layer of the device. Snapshots for six time steps. Parameters: tunneling junction capacitance $C_{\rm j}$ = 20 aF, tunneling junction conductance = 5 μS, bias capacitance $C_{\rm L}$ = 10 aF, coupling capacitance C = 2.2 aF, bias voltage $-V_{\rm b1}$ = -26.5 mV, and zero temperature.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

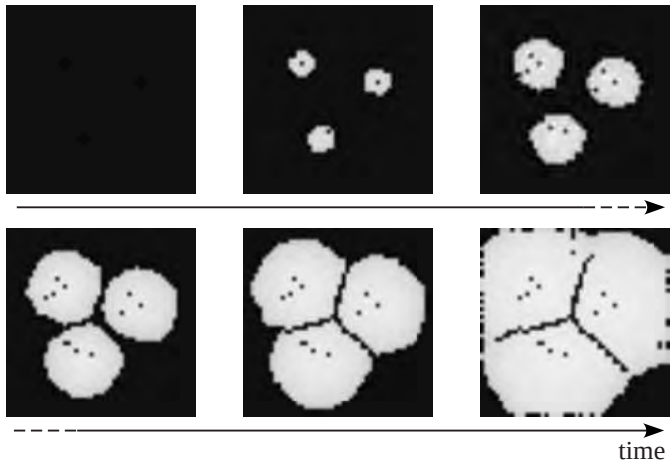


Fig. 12. Expanding circular pattern in the bottom layer of the device. Snapshots for six time steps. Traveling nonlinear waves in this layer construct a VD. Parameters are the same as in Fig. 11 without bias voltage $-V_{\mathrm{b2}}$ = -18.5 mV.

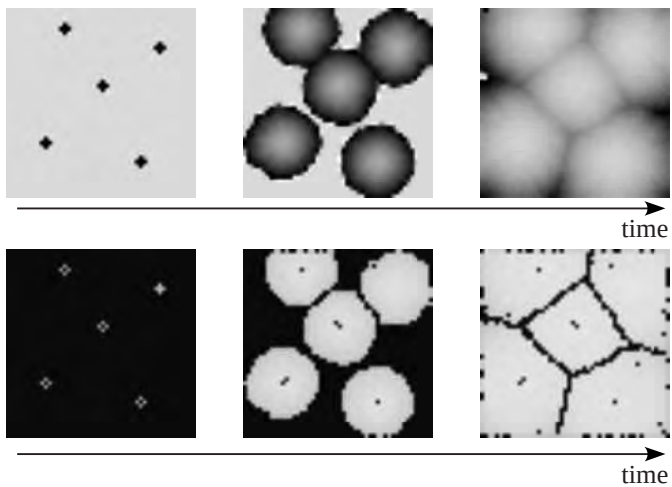


Fig. 13. Simulation results of computing a VD with five planer points by using proposed device. Upper three snapshots show the voltage density of the top layer, and the bottom three snapshots show the voltage density of the bottom layer.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Experimental implementation of direct-proportional length-based DNA computing for the shortest path problem

Zuwairie Ibrahim[1,2], Yusei Tsuboi[1], Osamu Ono[1], and Marzuki Khalid[3]

[1] Institute of Applied DNA Computing, Meiji University, 1-1-1 Higashi-mita, Tama-ku, Kawasaki-shi, Kanagawa-ken, 214-8571 Japan
zuwairie, tsuboi, ono@isc.meiji.ac.jp
http://www.isc.meiji.ac.jp/~i3erabc/IADC.html

[2] Department of Mechatronics and Robotics, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Darul Takzim, Malaysia
zuwairie@fke.utm.my

[3] Center for Artificial Intelligence and Robotics (CAIRO), Universiti Teknologi Malaysia, City Campus, Jalan Semarak, Kuala Lumpur, Malaysia
marzuki@utmkl.utm.my

Abstract. Bio-molecular or DNA computing has emerged as an interdisciplinary field that draws together chemistry, molecular biology, computer science, engineering, and mathematics. From DNA computing point of view, it has been proved that it is possible to solve weighted graph problems such as Traveling Salesman Problem (TSP) and the shortest path problem by exploiting some characteristics of DNA. Those characteristics are length, concentration, and melting temperature of DNA. In this paper, we present an alternative length-based DNA computing approach whereby the cost of each path is encoded by the length of the oligonucleotides in a proportional way. The advantage is such that, after an initial pool generation and amplification, polyacrylamide gel electrophoresis can be performed to separate the respective DNA duplex according to their length which directly decodes the results. For an efficient initial pool generation, parallel overlap assembly method is employed. After amplification is done by polymerase chain reaction (PCR), the result of the computation is visualized by polyacrylamide gel electrophoresis. The experimental results show the effectiveness of the proposed direct-proportional length-based computation

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

and prove that the shortest path problem has been successfully solved on a DNA computer.

1 Introduction

A new computing paradigm based on DNA molecules has appeared in 1994 when Leonard M. Adleman [1] launched a novel *in vitro* approach to solve the so-called Hamiltonian Path Problem (HPP) with seven vertices by DNA molecules. The goal of the HPP is to determine whether a path exists that commences at the 'start city' and finishes at the 'end city', and passes through each of the remaining cities exactly once. While in conventional silicon-based computers, information is stored as binary numbers in silicon-based memories, in this novel approach, the information of the vertices is encoded by random DNA sequences. The computation is performed in bio-molecular reaction fashion which involves hybridization, denaturation, ligation, magnetic bead separation, polymerase chain reaction (PCR), and so on. The output of the computation, also in the form of DNA molecules can be read and visualized by electrophoretic fluorescence operation.

Four years later, in 1998, a length-based DNA computing which is called constant-proportional length-based DNA computing specifically for Traveling Salesman Problem (TSP) is proposed by Narayanan and Zorbalas [2]. A constant increase of DNA strands is encoded according to the actual length of the distance. A drawback of this method is that, there is a possibility of an occurrence of concatenated DNA strands of two distances which could be longer than the DNA strand of the longest distance that has been encoded. This may lead to errors in computing the shortest path [3]. This scheme, however, has not been realized by any laboratory experiment.

On the other hand, Yamamoto *et al.* [4] presented concentration-controlled DNA computing for accomplishing a local search for the shortest path problem. Although DNA computing with concentration control method enables local search among all the candidate solutions, it cannot guarantee that the most intensive band is the DNA representing the shortest path in the given graph. In addition, it is technically difficult to extract a single optimal solution from the most intensive band [3].

Lee *et al.* [5] proposed a DNA computing approach based on DNA melting temperature for solving TSP problem. Denaturation Temperature Gradient Polymerase Chain Reaction (DTG-PCR) has been introduced where DNA duplex of correct solutions will be denatured and amplified by the PCR operation. As the denaturation temperature increases, other DNA strands will be also subsequently amplified. However,

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

the amount of correct solutions will also be exponentially increased which does affect the final solution.

Due to the unsolved disadvantages, the constant-proportional length-based DNA computing has not yet been implemented in any laboratory experiments. Hence, with the aim to solve the limitation of the constant-proportional length-based approach, an alternative approach called direct-proportional length-based DNA computing is proposed. The shortest path problem has been selected for consideration of using the proposed technique. In this approach, the cost of an edge is encoded as a direct-proportional length oligos. After an initial pool generation and amplification, since numerous numbers of solution candidates are generated, by using the standard bio-molecular laboratory operations, it is possible to extract the optimal combination which represents the solution to the shortest path problem.

2 The Shortest Path Problem

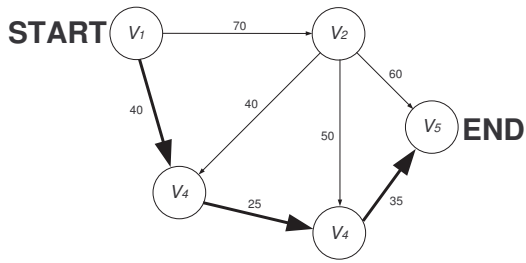


Fig. 1. Example showing a weighted directed graph $G = (V, E)$ with the shortest path shown as $V_1 - V_3 - V_4 - V_5$.

Even though the shortest path problem is belonging to the class P, i.e., it is not hard to solve this problem, it is worth to be solved by DNA computing because numerical evaluations are required during the computation. The input to the shortest path problem is a weighted directed graph G=(V, E, ω), a start node u and an end node v. The output of the shortest path problem is a (u,v) path with the smallest cost. In the case given in Fig. 1, if u is V_1 and v is V_5, the cost for the shortest path will be given as 100 and the optimal path is clearly shown as V_1–V_3–V_4–V_5.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

3 DNA Sequence Design and Synthesis

Consider a directed graph and the output of the shortest path computation as shown in Fig. 1. Let n be the total number of nodes in the graph. The DNA sequences correspond to all nodes and its complements are designed. Let $V_i(i = 1, 2, \ldots, n)$ and $\overline{V_i}(i = 1, 2, \ldots, n)$ be the 20-mer DNA sequences correspond to the ith node in the graph and its complement respectively. By using the available software for DNA sequence design, DNASequenceGenerator [6], the DNA sequences V_i is designed and listed in Tab. 1. Melting temperature, T_m is calculated based on Sugimoto nearest neighbor thermodynamic parameter [7]. The GC contents (GC%) and melting temperature (T_m) of each sequence are also shown. Table 2, on the other hand, shows the complement of the node sequences.

Node, V_i	20-mer Sequences (5'-3')	GC%	Melting Temperature, T_m (^{o}C)
V_1	AAAGCTCGTCGTTTAGGAGC	50	60.9
V_2	GCACTAGGGATTTGGAGGTT	50	60.3
V_3	GCTATGCCGTAGTAGAGCGA	55	60.5
V_4	CGATACCGAACTGATAAGCG	50	60.6
V_5	CGTGGGTGGCTCTGTAATAG	55	60.5

Table 1. DNA sequences for nodes.

Complement Node, $\overline{V_i}$	20-mer Complement Sequences (3'-5')
$\overline{V_1}$	TTTCGAGCAGCAAATCCTCG
$\overline{V_2}$	CGTGATCCCTAAACCTCCAA
$\overline{V_3}$	CGATACGGCATCATCTCGCT
$\overline{V_4}$	GCTATGGCTTGACTATTCGC
$\overline{V_5}$	GCACCCACCGAGACATTATC

Table 2. Complement node.

We introduce three rules to synthesize oligos for each edge in the graph as follows:

1. If there is a connection between V_1 to V_j, where $j \neq n$, synthesize the oligo for edge as $V_1(20)+W_{1j}(\omega\text{-}30)+V_j(20)$

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

2. If there is a connection between V_i to V_j, where $i \neq 1$, $j \neq n$, synthesize the oligo for edge as $V_i(20)+W_{ij}(\omega\text{-}20)+V_j(20)$
3. If there is a connection between V_i to V_n, where $i \neq 1$, synthesize the oligo for edge as $V_i(20)+W_{in}(\omega\text{-}30) + V_n(20)$

where V, W, and '+' denote the DNA sequences for nodes, DNA sequences for weight, and 'join' respectively. The synthesized oligos consist of three segments; two node segments and an edge segment. 'ω' denotes the weight value for corresponding DNA sequences for weight W_{ij}, where W_{ij} denotes the DNA sequences representing a cost between node V_i and V_j. The value in parenthesis indicates the number of DNA bases or nucleotides for each segment. The oligo is synthesized so that the number of DNA bases of that oligo and the cost at the corresponding edge are similar.

Edge	DNA Sequences
V_4–W_{45}–V_5	5'-CGATACCGAACTGATAAGCG ccaagCGTGGGTGGCTCTGTAATAG-3'
V_3–W_{34}–V_4	5'-GCTATGCCGTAGTAGAGCGA ccgtcCGATACCGAACTGATAAGCG-3'
V_1–W_{13}–V_3	5'-AAAGCTCGTCGTTTAGGAGCacgtcggttc GCTATGCCGTAGTAGAGCGA-3'
V_2–W_{23}–V_3	5'-GCACTAGGGATTTGGAGGTT ccgtcttttacccaagtaatGCTATGCCGTAGTAGAGCGA-3'
V_2–W_{24}–V_4	5'-GCACTAGGGATTTGGAGGTT acgtgttttaaggaagtacggtaagctgcg CGATACCGAACTGATAAGCG-3'
V_2–W_{25}–V_5	5'-GCACTAGGGATTTGGAGGTT gcgtcgcgtaaggcagtaccggactctgcc CGTGGGTGGCTCTGTAATAG-3'
V_1–W_{12}–V_2	5'-AAAGCTCGTCGTTTAGGAGC cggtggtttaacgaagtcctgtactatgggttatttgcag GCACTAGGGATTTGGAGGTT-3'

Table 3. DNA sequences for edges.

Table 3 lists all the synthesized oligos based on the proposed synthesis rules. Again, DNASequenceGenerator [6] is employed. The node segment and edge segment are distinguished by capital and small letters respectively. The complement sequences of each node are synthesized as well.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

4 Direct-Proportional Length-Based DNA Computing for the Shortest Path Problem

Currently, there are two kind of initial pool generation methods for solving weighted graph problem: hybridization/ligation and parallel overlap assembly (POA). The hybridization/ligation method has been firstly introduced by Adleman [1] to solve a HPP. For hybridization/ligation method, during the operation, the link oligos hybridize through the hydrogen bonds by enzymatic reaction. The hybridization/ligation reaction is well shown in Fig. 2 [8].

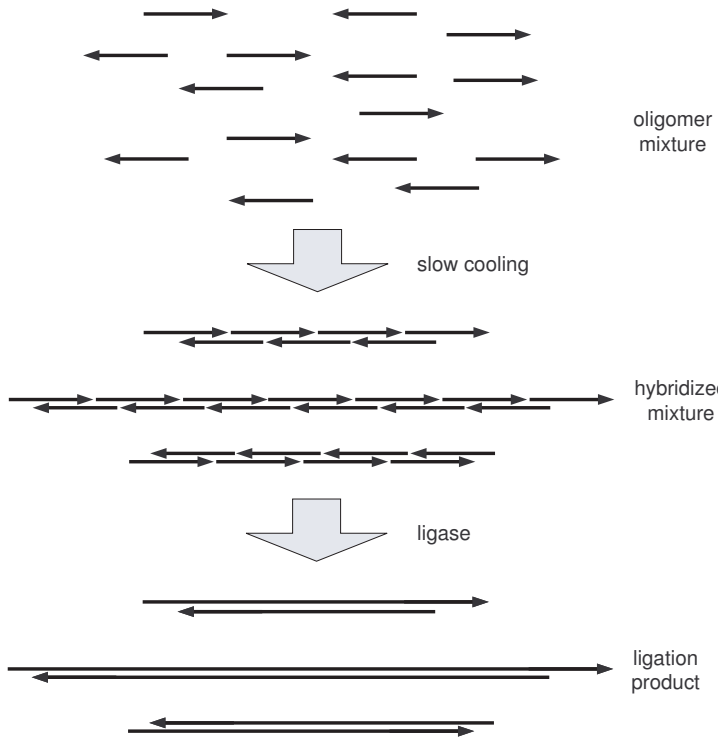


Fig. 2. Hybridization/ligation method for initial pool generation. The arrowhead indicates the 3' end.

POA has been used [9] and broadly applied in gene construction [10-12], gene reconstruction [13], and DNA shuffling [14]. POA involves thermal cycle and during the thermal cycle, the position strings in one oligo

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

anneals to the complementary strings of the next oligo. The 3' end side of the oligo is extended in the presence of polymerase enzyme to form a longer double stranded DNA (dsDNA). One cycle of parallel overlap assembly is depicted in Fig. 3 [8]. After a number of thermal cycles, a data pool with all combinations could be built.

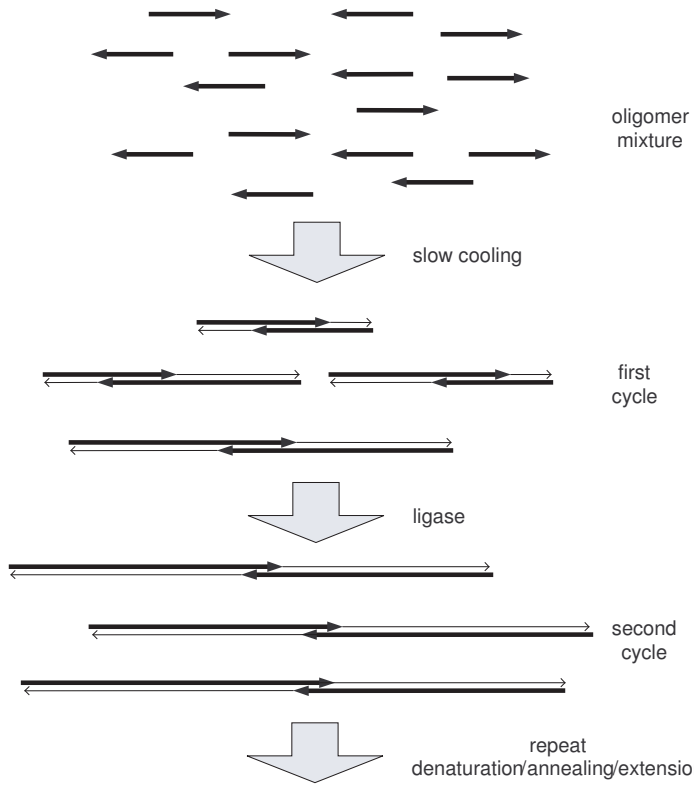


Fig. 3. Parallel overlap assembly for initial pool generation. The thick arrows represent the synthesized oligos which are the input to the computation. The thin arrows represent the elongated part during polymerization. The arrowhead indicates the 3' end.

Recently, Lee *et al.* [8] did a comparison between hybridization/ligation method and POA for initial pool generation of DNA computing. They came out with a conclusion that for the initial pool generation of weighted graph problems, POA method is more efficient than that of hybridiza-

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

tion/ligation method. According to [8], the advantages of POA over hybridization/ligation method for initial pool generation are as follows:

1. The initial pool size generated from the same amount of initial oligos is about twice larger than that of hybridization/ligation method. Though, if a larger problem is considered, the initial pool size is too small to contain the complete pool. POA, however, with more cycle and large experimental scale could include the practical pools.
2. Initially, two single-stranded DNA molecules partially hybridize in the annealing step and then they are extended by polymerase. The elongated DNA molecules are denatured to two single-stranded DNA in the next denaturation step, and they are subjected to the annealing reaction at the next cycle. Therefore, POA does maintain the population size and the population size can be decided by varying the initial number of oligos.
3. In hybridization/ligation method, the population size decreases as reaction progress. The population size decreased by a factor of the number of components composing it in hybridization/ligation method. As the problem size increases, the required initial pool size increases dramatically. Moreover, initial pool generation by POA requires fewer strands than hybridization/ligation method to obtain similar amount of initial pool DNA molecules because complementary strands are automatically extended by polymerase.
4. POA does not require phosphorylation of oligos which is prerequisite for the ligation of oligos.
5. POA demands less time than hybridization/ligation method. Hybridization requires one and half hour while ligation required more than 12 hours. Hence, POA for 34 cycles requires only two hours. Therefore, POA is much more efficient and economic method for initial pool generation.

As stated in [3],

> In addition, the fact that larger weights are encoded as longer sequences is contrary to the biological fact that; the longer the sequences are, the more likely they hybridize with other DNA strands, though we have to find the shortest DNA strands.

From the biological point of view, this argument is definitely true. In order to overcome the limitation of general length-based DNA computing, the authors discovered that by utilizing POA for initial pool generation, a phase where numerous combinations of random routes of the graph are generated in the solution, a shortcoming, which is the biological influence contributed by the length of the oligos could be eliminated.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

In order to generate the initial pool of the direct-proportional length-based DNA computing for the example problem by using POA method, the input to the computation are all the synthesized oligos as listed in Tab. 3 and the complement sequences for each nodes, which are listed in Tab. 2. These inputs are poured into a test tube and the cycles begin. In fact, the operation of POA is similar as polymerase chain reaction (PCR) but the difference is that POA operates without the use of primers. As PCR, one cycle consists of three steps: denaturation, hybridization, and extension.

At this stage, an initial pool of solution has been produced and it is time to filter out the optimal combinations among the vast alternative combinations of the problem. Unlike conventional filtering, this process is not merely throwing away the unwanted DNA duplex but rather copying the target DNA duplex exponentially by using the incredibly sensitive PCR process. This can be done by amplifying the DNA duplex that contain the start node V_1 and end node V_5 using primers. After the PCR operation is accomplished, there should be numerous number of DNA strands representing the start node V_1 and end node V_5 traveling through a possible number of nodes.

The output solution of the PCR operation then undergoes gel electrophoresis operation. During this operation, the DNA molecules can be separated in terms of its length and hence, the DNA duplex V_1–V_3–V_4–V_5 representing the shortest path starting from V_1 and ending at V_5 can be visualized. The overall procedure of direct-proportional length-based DNA computing is given in Fig. 4.

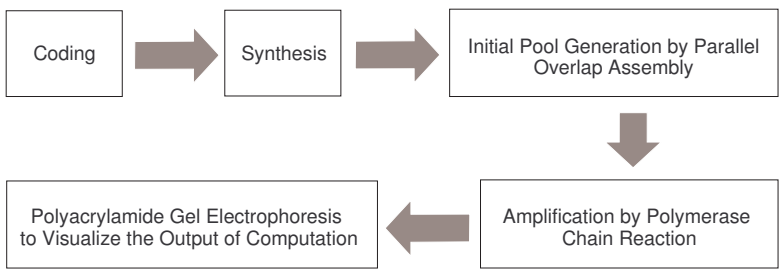


Fig. 4. The overall algorithm of direct-proportional length-based DNA computing.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

5 Experimental Protocols, Results, and Discussions

The initial pool generation by POA is performed in a 100 μl solution containing 12μl oligos (Proligo Primers & Probes, USA), 10 μl dNTP (TOYOBO, Japan), 10 μl 10x KOD dash buffer (TOYOBO, Japan), 0.5 μl KOD dash (TOYOBO, Japan), and 67.5 μl dd$H_2$0 (Maxim Biotech). The reaction consists of 25 cycles and for each cycles, the appropriate temperature are as follow:

- 94^{o}C for 30 s
- 55^{o}C for 30 s
- 74^{o}C for 10 s.

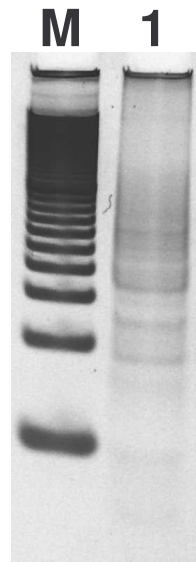


Fig. 5. Experimental results of gel electrophoresis on 10% polyacrylamide gel. Lane M denotes 20-bp ladder and lane 1 is the product of POA.

The product of parallel overlap assembly is shown in Fig. 5. According to the gel image, the band in lane 1 is blurs and thus, it is expected that all the candidate answers are successfully generated.

In order to select the paths that begin at V_1 and end at V_5, DNA amplification is done by employing PCR. The PCR is performed in a

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

25 μl solution consists of 0.5 $\mu l$for each primers, 1 μl template, 2.5 μl dNTP (TOYOBO, Japan), 2.5 μl 10x KOD dash buffer (TOYOBO, Japan), 0.125 μl KOD dash (TOYOBO, Japan), and 15.875 μl ddH_20 (Maxim Biotech). The reaction consists of 25 cycles and for each cycles, the appropriate temperature are as follow:

- 94^{o}C for 30 s
- 55^{o}C for 30 s
- 74^{o}C for 10 s

which is the same as POA. The sequences used as primers are AAAGCTCGTCGTTTAGGAGC (V_1) and GCACCCACCGAGACATTATC ($\overline{V_5}$).

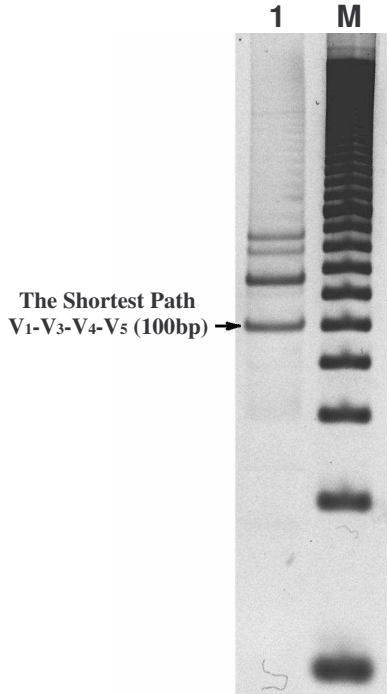


Fig. 6. Computation output on 10% polyacrylamide gel. Lane M denotes 20-bp ladder and lane 1 is the product of PCR.

In order to visualize the result of the computation, the product of PCR is subjected to polyacrylamide gel electrophoresis for 40 minutes at

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

200V. After that, gel electrophoresis, the gel is stained by SYBR Gold (Molecular Probes) and the gel image is captured. Figure 6 shows the product of PCR. Lane 1 consists of four bands showing that all the path that start at V_1 and end at V_5 have been successfully amplified. Those paths are V_1 – V_3– V_4– V_5 (100bp), V_1 – V_2– V_5(130bp), V_1 – V_2– V_4– V_5 (155bp), and V_1 – V_2– V_3– V_4– V_5(165bp). Clearly, the amplified paths have been sorted in term of length by gel electrophoresis and the output of the shortest path computation appears as the shortest band in lane 1.

6 Lower Bound

The direct-proportional length-based DNA computing is proposed essentially to overcome the shortcoming of constant proportional length-based DNA computing approach. However, by using this approach, the minimum weight of edges that can be encoded is limited and the weight falls in a very narrow range. This is because, the length of the solution is not only proportional to the length of the path it encodes but it also includes the number of vertices in the path. Hence, the lower bound, in term of minimum weight that can be encoded by direct-proportional length-based DNA computing is analyzed. Basically, according to the proposed DNA synthesis rules, the lower bound is achieved when:

$$\omega - \frac{3}{2}\beta = 0 \tag{1}$$

Hence, the minimum weight, $\omega_{\min}$, which can be encoded by oligos is attained as:

$$\omega_{\min} = \frac{3}{2}\beta \tag{2}$$

where β is the number of nucleotides used to encode each node of the input graph.

7 Conclusions

We have presented a new alternative approach called 'direct-proportional length-based approach' to solve weighted graph problems using DNA computing. Based on this approach, it is proposed that the directly proportional length of DNA could be used to encode the cost of each edge. After the initial pool generation and amplification, the DNA duplex is subjected to gel electrophoresis for the separation in term of length. It has been shown that the shortest path is represented by the shortest length of DNA duplex. For the sake of initial pool generation, two kinds

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

of methods are reviewed: hybridization/ligation and POA. For a successful demonstration of direct-proportional length-based DNA computing, we found that POA for initial pool generation is critically important. Finally, it is expected that the proposed approach, would extend the applicability of DNA computing for solving intractable weighted graph problems.

Acknowledgements

This research was supported partly by the IEEE Computational Intelligence Society (CIS) Walter J. Karplus Student Summer Research Grant 2004 for a research visit in September 2004 at the DNA Computing Laboratory, Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Hokkaido, Japan. The first author would like to thank Masahito Yamamoto for discussions that led to improvements in this work and also the permission to practice various kinds of biochemical experiments in the laboratory. Also, the first author is sincerely grateful to Atsushi Kameda, Satoshi Kashiwamura, and members of DNA Computing Laboratory of Hokkaido University for fruitful explanations and kind assistance during the practice of biochemical experiments, and anonymous reviewers for their important comments. Lastly, the first author is very thankful to Universiti Teknologi Malaysia (UTM) for a study leave.

References

1. Adleman, L.: Molecular Computation of Solutions to Combinatorial Problems, Science, Vol. 266 (1994) 1021-1024
2. Narayanan, A., Zorbalas, S.: DNA Algorithms for Computing Shortest Paths, in Proceedings of Genetic Programming (1998) pp. 718-723
3. Lee, J.Y., Shin, S.Y., Augh, S.J., Park, T.H., Zhang, B.T.: Temperature Gradient-Based DNA Computing for Graph Problems with Weighted Edges, Lecture Notes in Computer Science, Vol. 2568 (2003) 73-84
4. Yamamoto, M., Kameda, A., Matsuura, N., Shiba, T., Kawazoe, Y., Ahochi, A.: A Separation Method for DNA Computing Based on Concentration Control, New Generation Computing, Vol. 20 (2002) 251-262
5. Lee, J.Y., Shin, S.Y., Augh, S.J., Park, T.H., Zhang, B.T.: Temperature Gradient-Based DNA Computing for Graph Problems with Weighted Edges, in Preliminary Proceedings of the Eighth International Meeting on DNA Based Computers (2002) 41-50
6. Udo, F., Sam, S., Wolfgang, B., Hilmar, R.: DNA Sequence Generator: A Program for the Construction of DNA Sequences, In Proceedings of the Seventh International Workshop on DNA Based Computers (2001) 23-32

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

7. Sugimoto, N., Nakano, S., Yoneyama, M., Honda, K.: Improved Thermodynamic Parameters and Helix Initiation Factor to Predict Stability of DNA Duplexes, Nucleic Acid Research, Vol. 24 (1996) 4501-4505
8. Lee, J.Y., Lim, H.W., Yoo, S.I., Zhang, B.T., Park, T.H.: Efficient Initial Pool Generation for Weighted Graph Problems using Parallel Overlap Assembly, in Preliminary Proceedings of the Tenth International Meeting on DNA Based Computers (2004) 357-364
9. Kaplan, P.D., Ouyang, Q., Thaler, D.S., Libchaber, A.: Parallel Overlap Assembly for the Construction of Computational DNA Libraries, Journal of Theoretical Biology, Vol. 188, Issue 3 (1997) 333-341
10. Ho, S.N., Hunt, H.D., Horton, R.M., Pullen, J.K., Pease, L.R.: Site-Directed Mutagenesis by Overlap Extension using the Polymerase Chain Reaction, Gene, Vol. 77 (1989) 51-59
11. Jayaraman, K., Fingar, S.A., Fyles, J.: Polymerase Chain Reaction-Mediated Gene Synthesis: Synthesis of a Gene Coding for Isozymec of Horseradish Peroxidase, Proc. Natl. Acad. Sci. U.S.A., Vol. 88 (1991) 4084-4088
12. Stemmer, W.P., Crameri, A., Ha, K.D., Brennan, T.M., Heyneker, H.L.: Single-Step Assembly of a Gene and Entire Plasmid from Large Numbers of Oligodeoxyribonucleotides, Gene, Vol. 164 (1995) 49-53
13. DeSalle, R., Barcia, M., Wray, C.: PCR Jumping in Clones of 30-million-year-old DNA Fragments from Amber Preserved Termites, Experientia, Vol. 49 (1993) 906-909
14. Stemmer, W.P.: DNA Shuffling by Random Fragmentation and Reassembly: In Vitro Re-combination for Molecular Evolution, Proc. Natl. Acad. Sci. U.S.A., Vol. 91 (1994) 10747

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Striped and spotted pattern generation on reaction-diffusion cellular automata – Theory and LSI implementation –

Youhei Suzuki[1], Takahiro Takayama[2], Ikuko N. Motoike[2], and Tetsuya Asai[1]

[1] Graduate School of Information Science and Technology, Hokkaido University,
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814 Japan
[2] Future University - Hakodate, 116-2 Kamedanakano-cho Hakodate Hokkaido, 041-8655 Japan

Abstract. A novel reaction-diffusion cellular-automaton model that generates Turing-like spatial patterns is proposed. The model employs linear diffusion fields of activators and inhibitors and a discrete transition rule after diffusion. Theoretical analysis of the one dimensional model proved that i) spatial distribution given by a periodic square function is stable at the equilibrium and ii) the spatial frequency is inversely proportional to the square root of a diffusion coefficient of the inhibitors. We also designed LSI circuits that implement the RD model on a silicon chip that has a compact construction and low-power consumption. Circuit simulations revealed that the proposed LSIs could restore stripe and spot images in a short time regardless of the number of pixels in the image.

1 Introduction

In the process of ontogeny in a multicellular organism, the organism develops from a fertilized egg into matured differentiated cell groups, through repeated division/differentiates. Turing [1] suggested the concept of "Diffusion (driven) instability" for phenomena in systems where diffusion is able to enhance transition from a homogeneous state to a spatially in-homogeneous stable state. In his framework, time development in the system is described by the sum of reaction and diffusion. The former represents local production/extinction of the substance or state and

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

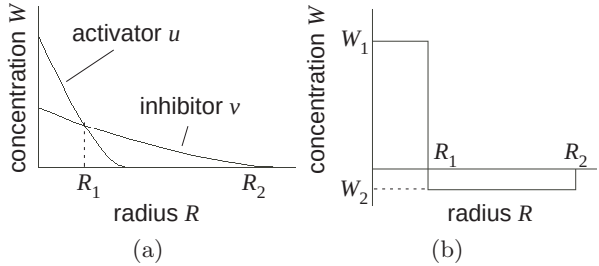


Fig. 1. Diffusion of activators and inhibitors on (a) continuous model and (b) discrete model.

the latter represents a transport process, which tends to dampen any inhomogeneity of the neighboring region, called the reaction-diffusion (RD) system. He gave an example where the spatial instability of a spatial homogeneous structure could take place through the addition of the diffusion effect. This Turing RD model is well known as one in which stable striped or spotted patterns are generated.

There are many ordered complex patterns in nature. For example, we can see patterns in animal skins where the patterns are formed spontaneously. Turing's and modified RD models have been studied because of their significance in explaining pattern formations on animal skins. Striped patterns can not only be seen in animal skin but also human fingerprints. Fingerprint patterns give us important cues for distinguishing individuals. Recent progress with digital microprocessors will certainly push advances in intelligent security systems that recognize fingerprints patterns. This paper deals with the basic mechanism and implementation of restoring striped/spotted patterns. We utilized spontaneous pattern formation with the RD Turing model to design the hardware RD system on a silicon chip that had compact and low-power construction.

2 Model and the Theoretical Analysis

We propose the use of RD systems as models for generating marking patterns on animals. The RD system is a complex system in which the reaction and diffusion of chemical species coexist under nonequilibrium conditions. It produces a variety of orders, rhythms, and self-organizing phenomena observed in nature and in life. Typical examples of such patterns are marking patterns on various animals, which are referred to as Turing patterns [2, 3].

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Turing patterns can usually be obtained by solving mathematical RD models described by a set of partial differential equations (PDE) that are represented by continuous spatiotemporal variables. Several attempts to reproduce Turing patterns with limited computational resources have been made over the years [4–9]. A typical example is the use of cellular automata (CA) where the space is separated by a set of discrete cells, and time and cell state are represented by discrete values. Gerhardt, Schuster and Tyson have discretized the RD model based on chemical system [5, 6]; Markus and colleagues have shown the way to avoid unisotropy of the pattern and described various shell patterns [7]; Weimar, Tyson, and Watson have generalized CA model based on RD model and evaluated CA in relation to PDE [8, 9]. In the way to construct CA model, simplifying the nonlinear dynamics in a continuous RD model is a difficult task because the differential equations are rewritten by conditional divergence rules in the CA.

Young [4] proposed a simplified discrete CA model for describing Turing patterns. He introduced a discrete model for diffusion effects between chemical substances and represented all the states (usually they had two variables; i.e., activators and inhibitors) with a single binary {1,0} variable. Then, he further simplified the diffusion of the two chemical substances.

One necessary condition for generating Turing patterns is that activators only influence their local neighbors (hard to diffuse), while inhibitors not only influence their neighbors but distant cells (easy to diffuse). Figure 1(a) illustrates the diffusion profile of activators and inhibitors in a continuous model, where R represent the distance from the center of diffusion, R_1 the position where activators and inhibitors have the same concentration, and R_2 the position where the concentration of inhibitors is asymptotically zero. When $R < R_1$, activators and inhibitors produce "active effects" on the field because the concentration of activators is higher than that of inhibitors. When $R_1 < R < R_2$, they produce "inhibitory effects" because the concentration of inhibitors is higher.

Young simplified the effects on distance R as we illustrated in Fig. 1(b). In his CA model, a cell whose state is "1" within $R < R_1$ has positive effects W_1, while a cell whose state is "1" within $R_1 < R < R_2$ has negative effects W_2. The transition of a cell in position $\mathbf{r}$ is determined by the weighted-sum of cells within $R < R_2$ whose states are "1" expressed as $\sum_{|\mathbf{r}-\mathbf{r_i}|\leq R_2} W$, where W represents the weight strength. If the summed value is zero, no transition occurs, while if the value is positive (or negative), the subsequent state of the cell is set to "1" (or "0"). This *step* transition rule corresponds to chemical reactions in continuous RD models. Young showed that stripe patterns and then spot patterns

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

appeared on the CA with fixed R_1, R_2 and W_1 by changing the value of W_2. Surprisingly, all the patterns became stable within 10 steps, even when random initial patterns were given to the CA.

2.1 A modified RD CA model

In Young's simplified CA, the diffusion terms in the continuous RD model are represented by the weighted summation of neighboring cells, while the reaction terms in the RD model are represented by the sign of the sum. Therefore, to describe a cell's transition, the cell has to refer to its neighboring cell's states. Since the number of neighboring cells is approximately calculated by $\pi \times R_2 \times R_2$, the number of physical connection wires (on CA hardware) to refer the neighboring cell's states increases significantly when R_2 increases. Moreover, the CA cannot generate spatially smooth patterns because step functions are used in the cell transition rule. A promising solution to these problems is using a discrete diffusion equation with a four-point spatial approximation method and an analog sigmoid function in the rule instead of the step function. Based on Young's simplification, we propose a novel RD CA model that is suitable for LSI implementation where a cell's transition is determined by the sigmoid function and weighted-summing computation is only restricted within the cell's nearest neighbors.

The weighted-summing computation described above is done by the diffusion fields. In other words, activators and inhibitors diffuse in individual diffusion fields and are then convoluted by a 2D array of cells. Each cell's state is determined by the difference between the concentration of activators, u, and inhibitors, v, at a given spatial point, (x, y). Diffusion equations for variables u and v are integrated for time δt. Then a cell's subsequent state is determined by the value of the sigmoid function for $u - v$. The dynamics can be formulated as

1. (Diffusion)
   $\partial u(\mathbf{r}, t)/\partial t = D_u \nabla^2 u(\mathbf{r}, t)$,
   $\partial v(\mathbf{r}, t)/\partial t = D_v \nabla^2 v(\mathbf{r}, t)$,
2. (Reaction)
   $u(\mathbf{r}, \delta t(n+1)) = v(\mathbf{r}, \delta t(n+1)) = f\big(u(\mathbf{r}, \delta t \cdot n) - v(\mathbf{r}, \delta t \cdot n) - c\big)$,
   $f(x) = \big(1 + \exp(-\beta x)\big)^{-1}$,

where n represents the time step, $\mathbf{r} = (x, y)$, c is the offset value of the sigmoid function, and β is the slope of the function. We defined this sequential operation as "one cycle". In the following, we show that the system produces spatiotemporal patterns by repeating this cycle.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

2.2 Theoretical Analysis

We analyze operations for the proposed model in 1D space and reveal the relation between the spatial frequency of equilibrium patterns and diffusion coefficients.

Since an impulse response of a diffusion equation is represented by the Gaussian, that of $u - v$ is given by a 'difference of Gaussian' (DoG) function:

$$\mathrm{DoG}(x,t) = \frac{1}{\sqrt{4\pi t}}\left[\frac{1}{\sqrt{D_u}}\exp\left(\frac{-x^2}{4D_u t}\right) - \frac{1}{\sqrt{D_v}}\exp\left(\frac{-x^2}{4D_v t}\right)\right], \quad (1)$$

where x represents the space. Differential distribution $u - v$, after activators and inhibitors are diffused for time δt, is thus given by

$$z_n(x) \equiv \int_{-\infty}^{\infty} r_n(x - X)\cdot \mathrm{DoG}(X, \delta t)\ dX, \quad (2)$$

where $r_n(x)$ represents an initial input to u and v at the n-th cycle. Therefore, the dynamics of the proposed model can be represented by

$$r_{n+1}(x) = f[z_n(x)], \quad (3)$$

where $r_{n+1}(x)$ represents the subsequent initial input. Assuming the equilibrium state, we obtain

$$r^*(x) = f[z^*(x)], \quad z^*(x) = \int_{-\infty}^{\infty} r^*(x - X)\cdot \mathrm{DoG}(X, \delta t)\ dX, \quad (4)$$

where $z^*(x)$ and $r^*(x)$ represent the equilibrium distribution of $u - v$ and the resulting sigmoid outputs. Assume the equilibrium distribution is

$$r^*(x) = \begin{cases} 1\ (-a < x < a) \\ 0\ (\text{else}), \end{cases} \quad (5)$$

where $a > 0$. For this input, we obtain

$$z^*(x) = \int_{x-a}^{x+a} \mathrm{DoG}(X, \delta t)\ dX, \quad (6)$$

$$= \frac{1}{2}\left[\mathrm{erf}\left(\frac{x+a}{p_u}\right) - \mathrm{erf}\left(\frac{x-a}{p_u}\right) - \mathrm{erf}\left(\frac{x+a}{p_v}\right) + \mathrm{erf}\left(\frac{x-a}{p_v}\right)\right], \quad (7)$$

where $p_{u,v} \equiv \sqrt{4D_{u,v}\delta t}$ and $\mathrm{erf}(\cdot)$ represents the error function. To ensure that input $r^*(x)$ is stable, $z^*(x)$ must be positive for $-a < x < a$, and be negative for other x. Figure 2 plots function $z^*(x)$ for given $r^*(x)$

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

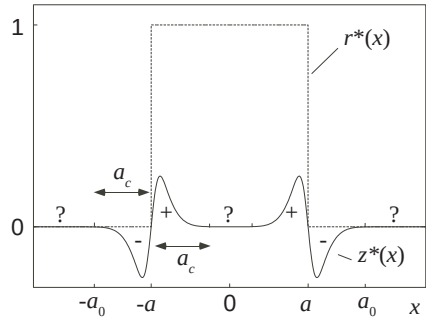


Fig. 2. DoG responses of proposed model for single square-pulse input.

when $D_u = 0.01$, $D_v = 0.1$, $\delta t = 0.01$, and $a = 0.2$. In this example, we see that the sign of $z^*(x)$ for the center ($x \approx 0$) and surrounds ($|x| > a_0$ in the figure) is indefinite because of $z^*(x) \approx 0$ in these regions. This results in unstable $r^*(x)$ at $x \approx 0$ and $|x| > a_0$.

An error function can be represented in the form of a normal (Gaussian) distribution

$$\frac{\mathrm{erf}(x)}{2} = \int_0^{\sqrt{2}\sigma x} \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{y^2}{2\sigma^2}\Big) dy, \tag{8}$$

where σ^2 represents the variance. Using the 3σ law of the Gaussian, we can approximately obtain the values of x where $z^*(x) \approx 0$ as

$$x = -a - \frac{3p_v}{\sqrt{2}},\ -a + \frac{3p_v}{\sqrt{2}},\ a - \frac{3p_v}{\sqrt{2}},\ \text{and}\ a + \frac{3p_v}{\sqrt{2}}, \tag{9}$$

which indicates that the region $-2a \le x \le 2a$ of $r^*(x)$ is stable as long as $a \le 3p_v/\sqrt{2}$ ($\equiv a_c$). Therefore, for a periodic square-wave input,

$$r^*(x) = \begin{cases} 1 \ \Big((4n-1)a < x < (4n+1)a\Big), & (n = 0, \pm 1, \pm 2, \dots) \\ 0 \ (\text{else}), \end{cases} \tag{10}$$

whose primary spatial frequency f_0 is given by $1/4a$, we conclude that a square wave of $f_0 \ge 1/4a_c = \sqrt{2}/12p_v$ is stable in the subsequent cycle.

For periodic waves of $f_0 < 1/4a_c$, regions of x where $z^*(x) \approx 0$ will exist, which results in $r^*(x) \approx 0.5$ (not 0 or 1). We estimate this region by employing piecewise linear function $f_{\mathrm{pwl}}(\cdot)$ instead of sigmoid function $f(\cdot)$. Because $df/dx|_{x=0} = \beta/4$, we obtain

$$f(x) \approx f_{\mathrm{pwl}}(x) \equiv \begin{cases} \beta x/4 + 0.5 \ (-2/\beta \le x \le 2/\beta) \\ 1 \ (x > 2/\beta) \\ 0 \ (x < -2/\beta), \end{cases} \tag{11}$$

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

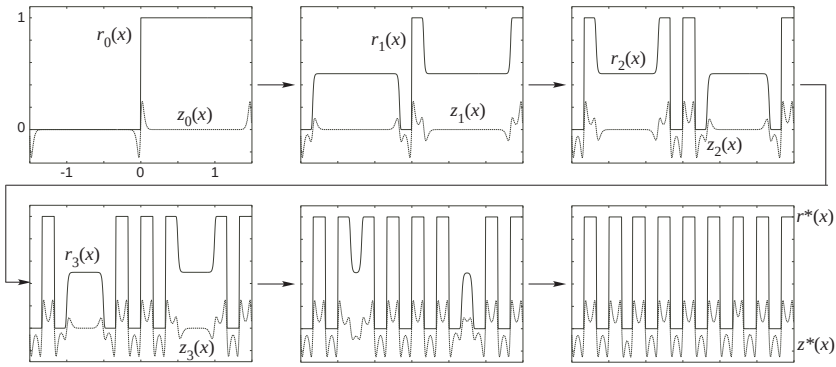


Fig. 3. Pattern formation on 1D model.

which means that $r^*(x)$ will not take 0 or 1 when $-2/\beta \leq z^*(x) \leq 2/\beta$. Therefore, the value of x where $z^*(x) = \pm 2/\beta$ determines stable wave frequency. To calculate this, we considered the following $z^*(x)$,

$$z^*(x) = \frac{1}{2}\left[\mathrm{erf}\left(\frac{x+a}{p_u}\right) - \mathrm{erf}\left(\frac{x+a}{p_v}\right)\right], \tag{12}$$

around $x = -a$ for simplicity. When the argument of the error function is large, the following asymptotic expansion can be used:

$$\mathrm{erf}(x) \approx 1 - \frac{1}{x\sqrt{\pi}} \exp(-x^2). \tag{13}$$

Therefore, when $p_u \ll p_v$ ($d_u \ll d_v$), one has

$$z^*(x) \approx \frac{1}{2\sqrt{\pi}}\left[\frac{p_v}{x+a} \exp\left(-\frac{(x+a)^2}{p_v^2}\right)\right]. \tag{14}$$

The value of x where $z^*(x_0) = 2/\beta$ is thus given by

$$x_0 = p_v \sqrt{\frac{F(2/k^2)}{2}} - a, \tag{15}$$

where $k \equiv 4\sqrt{\pi}/\beta$ and $F(\cdot)$ represents the inverse function of Lambert's W function. Therefore, we can conclude that i) $r^*(x)$ is not stable when the wavelength is larger than $2x_0$ and ii) the stable wavelength is proportional to p_v (square root of D_v).

Figure 3 has the simulation results for $\delta t = 0.01, D_u = 0.01, D_v = 0.1$ and $\beta = 10^4$ with a cyclic boundary condition. Step input was given at

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

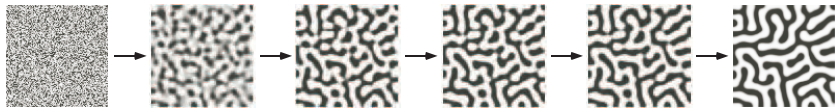


Fig. 4. Snapshots of stripe-pattern formation observed with our proposed model.

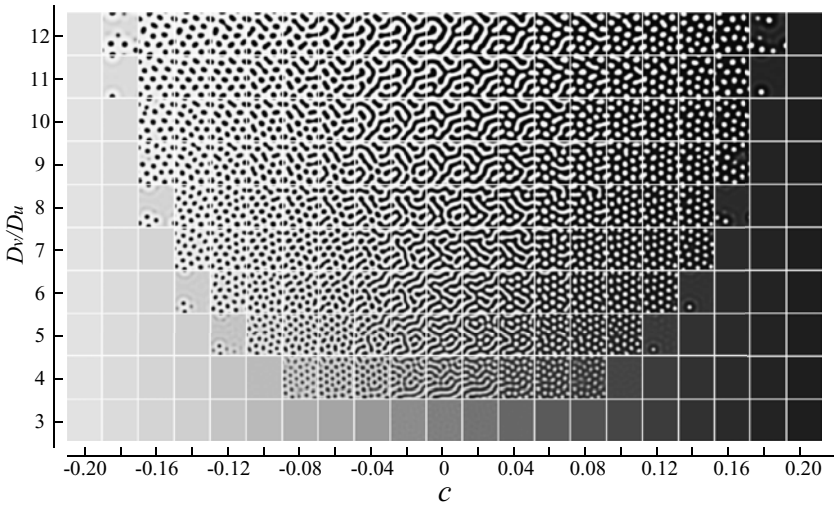


Fig. 5. Pattern diagram for proposed RD model.

the initial cycle. After a few iterations, a stable square wave appeared. The primary spatial-frequency agreed well with the theoretical prediction ($f_0 = 0.5x_0$). Furthermore, by changing the values of D_v, we numerically confirmed that the equilibrium wave frequency is inversely proportional to the square root of D_v.

Figure 4 is an example of striped pattern formation on a 2D model ($D_v/D_u = \beta = 10$, $c = 0$, $\delta t = 1$). The values of $f\big(u(\mathbf{r}, \delta t{\cdot}n) - v(\mathbf{r}, \delta t{\cdot}n) - c\big)$ are represented on a grayscale ($f(\cdot) = 0$: black, $f(\cdot) = 1$: white). The initial state was randomly set within the values of [0:1]. After approximately 10-cycle updates, a stable striped pattern was generated. The space was filled with striped patterns according to the initial spatial distribution. Therefore, if a striped pattern such as a fingerprint pattern is given to the CA, local patterns that do not fit the striped global patterns are replaced with striped patterns based on the global patterns.

Figure 5 shows a pattern diagram for two variable parameters (D_v/D_u and c). When the value of c was increased, the resulting patterns changed

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

from black spotted to white spotted via the stripe patterns. Also, the spatial frequency could be controlled by the value of D_v/D_u. That is, we can control the form of target patterns (spotted or striped) and the spatial resolution with these two parameters.

Physically, parameter c represents a total balance of activators u and inhibitors v in the model. When $c > 0$, v is predominant over u because the values of u must be larger than that of $v + c$ to ensure $f(u-v-c) > 0.5$, and vice versa when $c < 0$. This can easily be confirmed from Fig. 5 where the area occupied by inhibitors (black areas) is equal to the area occupied by activators (white) when $c = 0$, while inhibitors (black areas) become predominant as c increases.

3 RD Chip Architecture

We designed an RD chip for the RD model described in the preceding section. The basic concept was the use of a CA structure where i) each cell circuit had state memories and a diffusion circuit, and the differential amplifier had a sigmoidal response function, and ii) each cell was only connected to nearest-neighbor cell circuits to reduce the complexity of wiring.

Figure 6 illustrates the unit cell circuit we propose. It consists of a main memory circuit for the state variable (C_{m1} and C_{m2} with master-slave memory structure), a temporal memory (C_{ma}), a diffusion circuit with floating gates (FGs), a differential amplifier that has a sigmoidal response function (DIF), three buffering circuits (voltage followers labeled VF1-VF3), and five transfer gates.

Let us first describe the operations of the diffusion circuit. Let V_{c} and V_{fg} denote the voltage stored in a cell's main memory and that of the FG, respectively. If the initial charge, Q_0, of the FG is zero and the input gate capacitance of VF1 (C_{ox}) is much smaller than the capacitance of control gates C, then V_{fg} is given by

$$V_{\mathrm{fg}} = \frac{V_{\mathrm{n}} + V_{\mathrm{w}} + V_{\mathrm{s}} + V_{\mathrm{e}} + 4\ V_{\mathrm{c}}}{8}, \tag{16}$$

where V_{n}, V_{w}, V_{s}, and V_{e} represent the voltages stored in a neighboring cell's main memory. Assuming that each cell can store V_{fg} in a memory circuit, we obtain

$$\frac{V(t+\Delta t) - V(t)}{\Delta t} = \frac{V_{\mathrm{n}} + V_{\mathrm{w}} + V_{\mathrm{s}} + V_{\mathrm{e}} - 4\ V(t)}{8\Delta t}, \tag{17}$$

where $V_{\mathrm{fg}} = V(t+\Delta t)$ and $V_{\mathrm{c}} = V(t)$. This equation corresponds to the diffusion equation

$$\partial V(\mathbf{r}, t)/\partial t = D\nabla^2 V(\mathbf{r}, t), \tag{18}$$

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

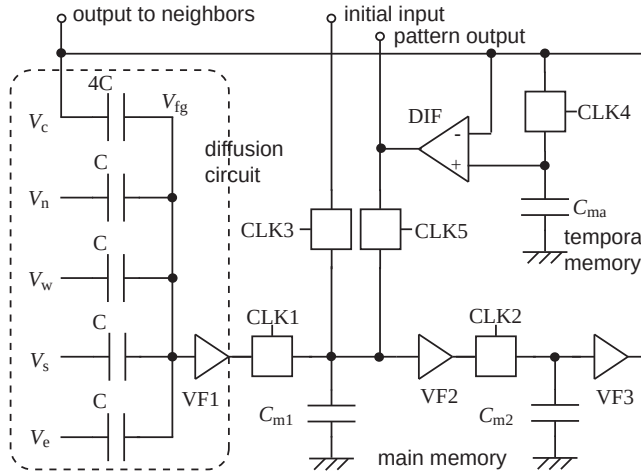


Fig. 6. Construction of unit cell circuit.

when the discrete space is represented on a square grid ($h \equiv \Delta x = \Delta y$ and $D = h^2/(8\Delta t)$). Therefore, by storing $V_{\rm fg}$ in memory and updating this repeatedly, the circuit can solve the discrete diffusion equation. The circuit employed a master-slave analog memory for safe memory updates where the master and slave operations were controlled by CLK1 and CLK2.

Our RD model, as it stands, requires two diffusion fields for variables u and v. Therefore, we propose a novel circuit architecture that has the same operation as our model using only *one* diffusion circuit. The diffusion coefficient and time are always represented in the solution to diffusion equations in the form of $D \cdot t$, which indicates that diffusion and time are interchangeable. For example, let us assume $D_v/D_u \equiv \Delta t_0/\Delta t_1$, and after diffusion for Δt_0 in one diffusion field, the diffused state is stored in a temporal memory, ($C_{\rm ma}$). The memorized voltage is further diffused for $\Delta t_1 - \Delta t_0$. At this time, the voltage in the main memory is equivalent to the voltage after diffusion with D_v for Δt_0. Since temporal memory $C_{\rm ma}$ stores the voltage after diffusion with D_u for Δt_0, one can obtain a diffused distribution for Δt_0 with different diffusion coefficients D_u and D_u, with one diffusion circuit and a temporal memory.

After one-cycle diffusion for Δt_0, the voltage stored in the main and temporal memories is supplied to the differential amplifier (DIF), which has a sigmoid transfer function. By employing FG MOSFETs in the differential pair, we can control offset value c for the sigmoid function. It should be noted that parameter β in the sigmoid function determines

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

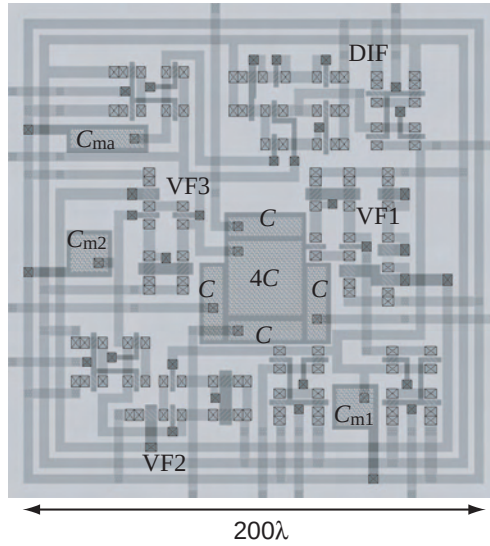


Fig. 7. Layout pattern of unit cell with fixed c (=0) for fingerprint restoration.

the smoothness of generated patterns. If $\beta \to \infty$, the function becomes a step function, which implies that the resulting patterns will not be smooth. The value of β is determined by the rate of FG capacitances for input and control gates, and cannot be controlled after the chip has been fabricated. The output of DIF is stored in C_{m1} by CLK5 evoked in the subsequent step. CLK3 is a control clock that captures an initial voltage, which is produced by the photodiode or electrostatic sensors mounted on each cell, in the main memory circuit.

The cell circuit was designed by assuming a standard n-well double-poly double-metal CMOS process for fabrication. Figure 7 shows the layout (λ represents the scaling parameter).

4 Simulation Results

We simulated the operations of a 2D array for the proposed cell circuit (180×180 cells). The circuit parameters were $C = 100$ fF, $C_{\mathrm{ox}} = 1$ fF, $C_{\mathrm{m1}} = C_{\mathrm{m2}} = C_{\mathrm{ma}} = 100$ fF, and $D_v/D_u = 10$. One cycle in the simulation was performed in 80 steps (8 steps for diffusion with D_u and 72 steps for D_v). The chip required 2 clocks for one-step operation because of master-slave memory operation.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

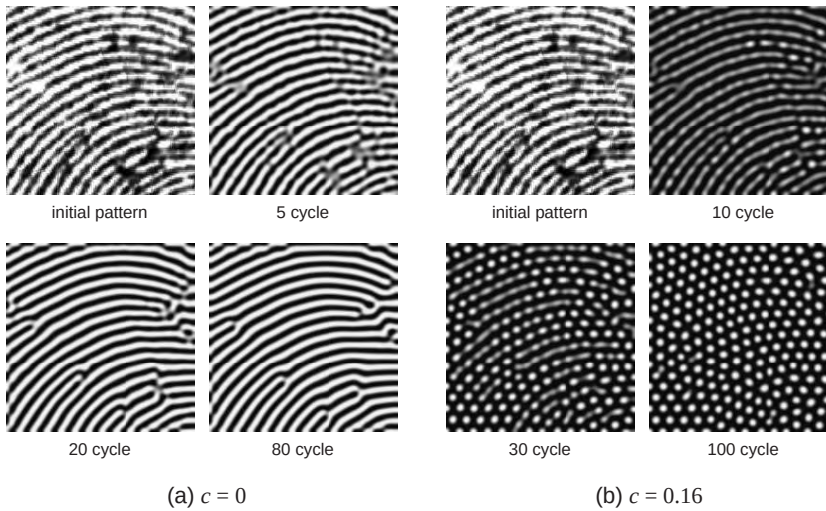


Fig. 8. Snapshots of pattern formation from initial fingerprint image.

Figure 8(a) has snapshots of pattern formation in the circuit. We could estimate that the system produces striped patterns when $c = 0$ from Fig. 5. Therefore, we used a fingerprint pattern as an initial input. We confirmed that noisy local patterns were repaired by their surrounding striped patterns, as time increased. The circuit required 50 cycles (8000 clocks) to reach equilibrium. Figure 8(b) shows the results for $c = 0.16$ (offset voltage of DIF was set at $0.16 \times V_{dd}$). The same initial input as in Fig. 8(a) was given to the circuit. As expected from Fig. 5, spotted patterns were obtained. The pattern formation process was the same as in Fig. 8(a) where noisy local spots were restored by surrounding global spotted patterns. Therefore, this circuit would be suitable for restoring regularly-arranged spotted patterns such as polka-dot patterns. The system took 100 cycles (16,000 clocks) until it reached equilibrium to restore the spotted patterns.

The number of cycles for generating equilibrium patterns only depends on the width of the stripes (radius of spots), and is independent of image size. This implies that the power consumed by repeating these calculations is significantly reduced when the number of pixels increases. Buffers and differential amplifiers can be turned off during static periods in the system clock to reduce static power consumption in a cell. These properties enabled us to develop a low-power LSI to restore striped and spotted patterns.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

5 Summary

We proposed a novel RD model that is suitable for LSI implementation and its basic LSI architecture. We developed image-processing LSI circuits based on pattern formation in reaction-diffusion (RD) systems. We introduced continuous diffusion fields and an analog state variable to Young's local activator-inhibitor model [4]. We produced a model pattern diagram on a 2D parameter space through extensive numerical simulations. We showed that the spatial frequency and form (striped or spotted) could be controlled with only two parameters. We then designed a basic circuit for the proposed model. We designed an RD LSI based on the analog computing method where the concentration of chemicals was represented by a 2D voltage distribution and the cell voltage was diffused step by step. By mimicking two diffusion fields with the proposed model in one diffusion circuit on the LSI, we reduced the area of the unit cell circuit. We confirmed the operations of the LSI were as expected through circuit simulations. Finally, we demonstrated fingerprint image restoration on the LSI and that it successfully restored smooth striped and spotted images from noisy input images.

Acknowledgment

This study was partly supported by Industrial Technology Research Grant Program in '04 from New Energy and Industrial Technology Development Organization (NEDO) of Japan, and a Grant-in-Aid for Young Scientists [(B)17760269] from the Ministry of Education, Culture Sports, Science and Technology (MEXT) of Japan.

References

1. Turing, A.M. The chemical basis of morphogenesis. Phil. Trans. R. Soc. Lond. **B 237** (1952) 37–72.
2. Nicolis, G. & Prigogine, I.: Self-organization in Nonequilibrium Systems — From Dissipative Structures to Order through Fluctuations. John Wiley & Sons, Inc., New York (1977).
3. Murray, J.D.: Mathematical Biology I & II (3rd Ed.). Springer, New York (2002).
4. Young, D.A. A local activator-inhibitor model of vertebrate skin patterns. Math. Biosci. **72** (1984) 51–58.
5. Gerhardt M. & Schuster H., A cellular automaton describing the formation of spatially ordered structures in chemical systems. Physica D. **36** (1989) 209–221.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

6. Gerhardt M., Schuster H. & Tyson J. J., A cellular automaton model of excitable media II. Curvature, dispersion, rotating waves and meandering waves. Physica D **46** (1990) 392–415.,
7. Markus M. & Hess B., Isotropic cellular automaton for modeling excitable media. Nature, **347** (1990) 56–58,
8. Weimar J. R., Tyson J. J., & Watson L. T., Diffusion and Wave Propagation in Cellular Automata Models for Excitable Media. Physica D **55** (1992) 309–327,
9. Weimar J. R. & Boon J.-P., Class of Cellular Automata for Reaction-Diffusion Systems. Phys. Rev. E **49** (1994) 1749–1752.
10. Schepers H. & Markus M., Two types of performance of aniIsotropic cellular automaton: stationary (Turing) patterns and spiral waves. Physica A **188** (1992) 337–343,
11. Kusch I. & Markus M., Mollusc shell pigmentation: cellular automaton simulations and evidence for undecidability. J. Theor. Biol. 178, 333-340 (1996).

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Several remarks on practical implementation of image processing by chemical reaction-diffusion media

N.G. Rambidi, S.G. Ulyakhin, and A.S. Tsvetkov

Physics Department, Moscow State University, Leninskie Gory, 117234 Moscow, Russia

Abstract. Belousov-Zhabotinsky chemical reaction-diffusion media represent information processing means fundamentally different from contemporary digital computers. Distributed character and complex nonlinear dynamics of chemical reactions inherent in the medium is the basis for large-scale parallelism and complex logical operations performed by the medium as primitives and equivalent to hundreds of binary fixed-point operations. Photo-sensitive catalysts controlling dynamics (modes of functioning) of the medium enable to easily perform input of initial data and output of computational results. It was shown that the spatio-temporal evolution of an image produced by Belousov-Zhabotinsky medium is adequate to image processing operations carried out using mathematical morphology technique. Conditions necessary to reliable use of Belousov-Zhabotinsky media for image processing are analyzed.

1 Introduction : von Neumann vs non von Neumann information processing devices and artificial intelligence problems

The latest decade was remarkable for the development of molecular information processing devices. Important approaches were elaborated that had made practical commercial applications of molecular devices visible. They were:

- engineering style design of hiropticene molecules [1],
- VLSI crossbar architecture based on rotaxane molecules [2]

and some other important suggestions.

However all these attempts used von Neumann principles inherent in contemporary digital computing. And the problem is that these principles do not seem to be efficient for solving artificial intelligence problems

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

which are of the great importance for the modern fields of human activity. According to estimates [3] digital computers will be in striking distance of being powerful enough to match human brainpower in 20th – 40th of our century. However they will be supercomputers composed of tens of thousands of fastest microprocessors and costing tens of millions of dollars. Therefore the elaboration of simple, easy in operation, and cheap information processing devices capable to solve even single artificial intelligence problems is one of the important needs of contemporary computing.

Neural network devices are more efficient in solving problems of artificial intelligence due to their biological background. However practical implementation of these devices faces troubles determined by planar semiconductor technology used. Therefore during the last decade attempts have been made to find another material implementation for neural nets consisting with their biological origin. One of them was based on chemical reaction-diffusion media that have showed real capabilities to solve problems of artificial intelligence, and first of all image processing and recognition [4].

Information processing by distributed reaction-diffusion media was under investigation during the last three decades. Biological origin of these systems was the starting point for different theoretical approaches aimed to solve effectively artificial intelligent problems [5-14]. Important experimental studies in this field were launched in the late eighties [15-19]. It enables to realize basic information capabilities of reaction-diffusion computing.

Primordially concentrations of medium components are spatially uniform. This uniform distribution could be changed by external physical or chemical stimuli. As a result spatio-temporal evolution of the medium composition begins. This situation could be considered as input of information and its following evolution. The character of the evolution is determined by medium specific properties. Media having nonlinear mechanisms of the component interaction are the most promising for information processing. Given high behavioral complexity of these media, modes of the input information evolution could be equivalent to tens or even hundreds binary operations of digital computer.

It has been shown during the last decades that chemical Belousov-Zhabotinsky media could be effectively used for information processing [15-19]. One of the most promising direction of using these media is image processing and recognition. However up to now at least two points are not clear.

Contemporary image processing is an original, practically oriented field of modern informatics. Image processing technique embraces a lot

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

of primitive operations that are based on several approaches, such as cellular logic, prairie fire conception, mathematical morphology [20-24].

So the first point is: are modes of input image evolution in the reaction-diffusion medium adequate to the actual image processing operations? Specific feature of light-sensitive reaction-diffusion media is high behavioral complexity. Modes of input image evolution in the reaction-diffusion medium are determined not only by the medium composition and temperature, but also by intensity of light illumination. And the second point is: might modes of image evolution be actual image processing operations or some of them are artifacts that do not reflect real features of the input image?

This work was designed to investigate these problems important for practical implementation based on Belousov-Zhabotinsky reaction-diffusion media and to look for the variety of image processing operations that could be effectively performed by these media.

2 Experimental

Experimental investigations discussed below were performed using the set up, that was sufficiently modified in comparison with its previous version [25]. Optical tract of the set up was partly changed to add some additional operational modes and sources of extraneous light were carefully removed. Different designs of reaction-diffusion reactors based on polymeric materials were elaborated.

Principal scheme of the set up is shown in Fig. 1. The computer-controlled Sanyo PLC-510M LCD video projector (VGA compatible, 270 ANSI lumens) was used for input of initial data. A Mintron OS-045D video camera was used to record the steps of the image processing by the medium (0.02 lx sensitivity, 600 TV lines resolution). Digitized records of images were saved in the memory of the Pentium III PC. It was possible to write single frames and video clips using VidCap software.

Numerical simulation of the image processing operations performed by the light-sensitive Belousov-Zhabotinsky media was based on modified FKN approximation [26,27]. Euler and Runge-Kutta methods were used to solve sets of differential equations. The software was designed as two independent modules. The first of them was used for the input of initial data and for the visualization of results of computations. The second performed calculations. Such software structure simplified the addition of the new data. Modules were written using C++ programming language and were compiled by Microsoft Visual C++ 6.0. The size of the software is about 1 Mb.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

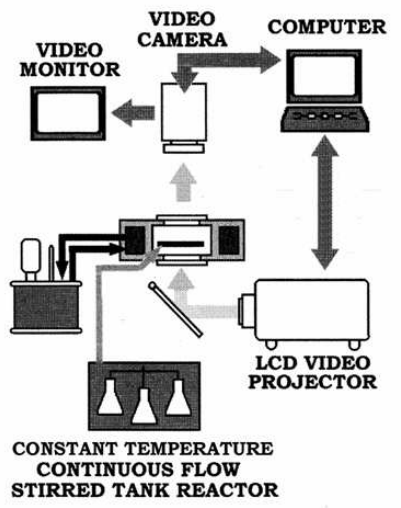


Fig. 1. Principal scheme of experimental setup.

Detailed description of the experimental technique was published elsewhere [25].

3 High behavior complexity of light-sensitive Belousov-Zhabotinsky media and some restrictions for the image processing performed by these media

Detailed description of principles and technique for using Belousov-Zhabotinsky media for image processing were published elsewhere [25]. Let us mention here some points only important for the following discussion.

Belousov-Zhabotinsky type media are the most known chemical nonlinear reaction-diffusion systems [26]. The Belousov-Zhabotinsky reaction is the oxidation of some organic substance (malonic acid) by the inorganic oxidizing agent (sodium or potassium bromate). Ions of transition metals (Fe or Ru mainly) are catalysts of the reaction. The principal scheme of the reaction corresponds to

$$\mathbf{C_3H_4O_4 + NaBrO_3 + H^+ \ ?Fe?\ C_3\ H_3BrO_4 + H_2O + CO_2}$$

The process of evolution of initial concentration distribution proceeds in the media due to chemical reactions and diffusion. The catalyst

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

changes its electronic state in the course of reaction when the medium goes from one stable state into another. As a consequence, the reagent changes its color (from red to blue and vice versa) and it is easy to visualize the process and to observe its spatio-temporal evolution. Therefore the catalyst works also as a "micro-macro" interface bringing into correspondence the evolution of the chemical concentration distribution and optical (macro} image of the process evolution.

The important information processing feature of Belousov-Zhabotinsky type media that determine medium processing capabilities is the neural network architecture (see [4]).

Belousov-Zhabotinsky type media based on a light-sensitive catalyst are convenient for the input of initial information. The light-sensitive catalyst initiates a sequence of photochemical reactions under light illumination [28]:

$$\mathbf{Ru}^{+2} + \mathbf{h}\nu \rightarrow \mathbf{*Ru}^{+2}$$
$$\mathbf{*Ru}^{+2} + \mathbf{C_3H_3BrO_4} \rightarrow \mathbf{Ru}^{+3} + \mathbf{Br}^{-} + \mathbf{organic\ products}$$
$$\mathbf{Ru}^{+3} + \mathbf{C_3H_3BrO_4} \rightarrow \mathbf{Ru}^{+2} + \mathbf{Br}^{-} + \mathbf{organic\ products}$$

that changes the composition of the medium in each its point according to light intensity in the point.

The mechanism of the Belousov-Zhabotinsky reaction represents a set of intermediate stages. The real set of these stages is not known exactly till now. The most widely accepted model - Field-Korosh-Noyes (FKN) approximation [26], contains 11 intermediate reactions and could be reduced to two kinetic equations that correspond to temporal evolutions of reaction inhibitor "u" ($HBrO_2$) and activator "v" (Fe^{3+} or Ru^{3+}):

$$\frac{\partial u}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\mu - u}{\mu + u}(qv + \phi) + u - u^2\right] + D_u \Delta u$$
$$\frac{\partial v}{\partial t} = \lambda u - u + D_v \Delta v$$

Here ε, q and λ are constants defined by initial concentrations of the medium components and kinetic of intermediate reactions, μ=0.001, ϕ describes a light illumination of the medium.

These equations are sufficiently simpler in the case of totally mixing media, when the diffusion could be neglected. Null-clines introduced in this case are the convenient basis to classify dynamic states of the medium. Null-clines corresponding to above equations (μ= 0.001, q=0.7) are shown in Fig. 2. S-type and linear nullclines described inhibitor ($HBrO_2$) and activator (Ru^{+3}) dynamics correspondingly. Here unstable point (λ=1, ϕ/q=0) corresponds to oscillatory mode, whereas stable point (λ=0.05, ϕ/q=0) corresponds to excitable state.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

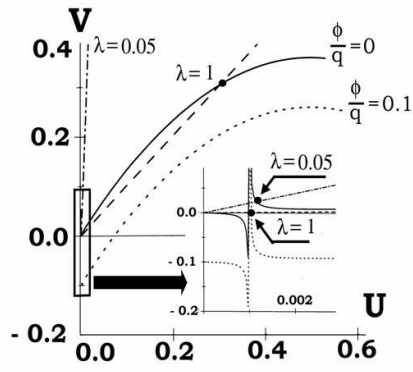


Fig. 2. Nullclines of Belousov-Zhabotinsky system.

The high behavioral complexity of light-sensitive Belousov-Zhabotinsky media is determined by the two factors responsible for the choice of dynamic state. They are the chemical composition of the medium and intensity of the light radiation. It is easy to see that there are, for instance, two possibilities to get excitable state. If light illumination is neglected (ϕ/q=0) two states , excitable and oscillatory could be obtained using different medium composition, that are described by different sets of constants (q=0.7, λ=1 for oscillatory state and q=0.7, λ=0.05 for excitable one). At the same time it is possible to move from the oscillatory state (q=0.7, λ=1, ϕ/q=0) to the excitable state using light illumination of the medium (q=0.7, λ=1, ϕ/q=0.1). This situation is definable for the understanding of the light-sensitive medium properties.

Considering image processing by light-sensitive media it is necessary to take into account three basic modes of the medium illumination.

The first one is using of the light illumination for the input of initial information into the medium. The exposure of the light illumination determines also the dynamical mode of the medium (see below).

There is another evident illumination mode that isn't usually discussed. It is necessary to illuminate the medium in the process of image evolution to have the possibility to record this process by video camera. It gives additional opportunities to control the medium dynamic mode.

And the third is the continuous illumination of the medium by a chosen non uniform light distribution in the process of the image evolution (see, for instance [29,30]).

Let us consider some features of the black and white image evolution in a light-sensitive medium. Suppose that exposures of the light illumination used for the input of initial data are low.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

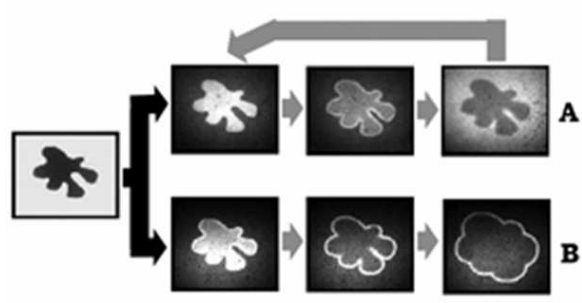


Fig. 3. Oscillating (A) and excitable (B) modes of Belousov-Zhabotinsky medium.

Suppose also that medium composition is chosen to initiate oscillatory dynamic mode. It is. possible to observe typical oscillations (Fig. 3A) if the intensity of the uniform illumination necessary for registration of the image evolution is low (brightness 5%, Photoshop HSB model). At the same time the typical excitable process could be observed (Fig. 3B) if the intensity of the uniform illumination in the process of the image evolution registration is high enough (brightness 30-40). Here and in all following figures the input of the initial image is shown (black arrows) and its evolution in the medium (grey arrows).

Thus the important feature of the light sensitive Belousov-Zhabotinsky media is that it is possible to observe both oscillatory and excitable modes of the medium dynamics having the same medium composition and different intensities of the light illumination in the process of the medium evolution registration.

The situation dramatically changes if the exposure of the light illumination used for the input of initial data in the medium is high. The train of the pulses reveals at the borders of images projected at the surface of the medium that spread along the medium (Fig. 4A). Mention here that this process was observed earlier [31] but under another experimental conditions. The revealing of the pulses stops if the input of the image is switched off, and the mode of the train following evolution can be both oscillatory and excitable depending on intensity of the uniform illumination in the process of the image evolution registration (Fig. 4B,C).

The remarkable feature of the light-sensitive Belousov-Zhabotinsky media is the fast switching of the evolution process (from oscillatory to excitable and back) controlled by the intensity of the light illumination (Fig. 5).

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

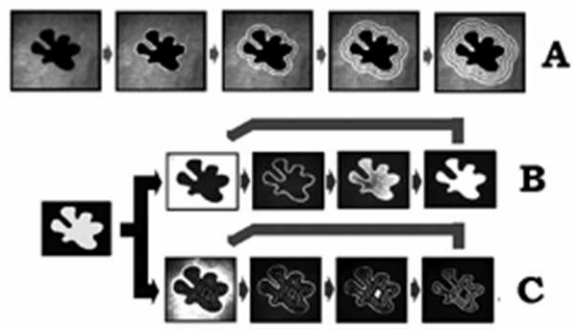


Fig. 4. Input of data (A) and oscillating models (B,C) of Belousov-Zhabotinsky medium.

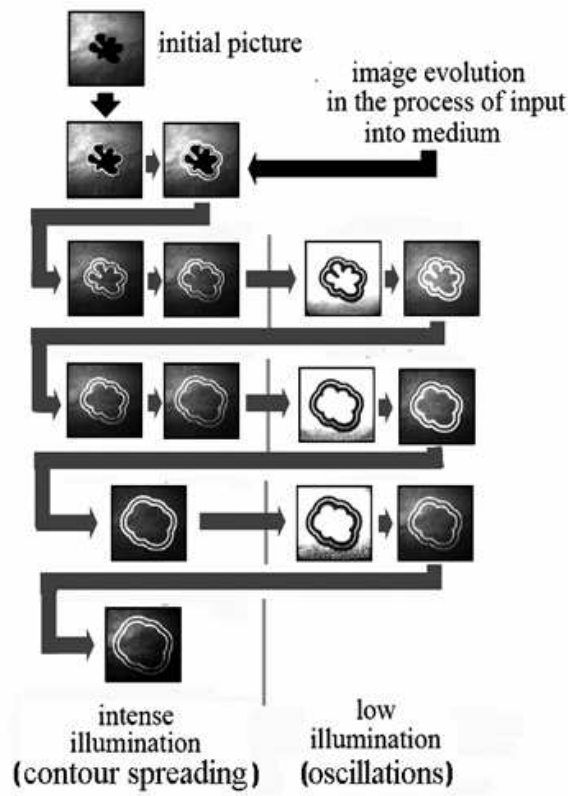


Fig. 5. Switching between oscillating and excitable modes in Belousov-Zhabotinsky medium.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Mentioned here that trains of pulses revealed under high illumination differs from typical excitable mode phenomenon.

The origin of this phenomenon seems to be made clearer taking into consideration that the increasing of the light illumination corresponds to the shift of the inhibitor null-cline and moving of the null-cline intersection point along inhibitor null-cline (from oscillatory to excitable state). Regretfully it was difficult to study this process in experiment because technical difficulties. Therefore an attempt was made to perform numerical simulation.

It was shown earlier [32] that numerical results obtained from an Oregonator model modified to include the photochemical production of Br^- and $HBrO_2$ are in semi quantitative agreement with the experimentally observed behavior. Therefore another possibility was chosen to simulate dynamic modes in the intermediate field between oscillatory and excitable modes.

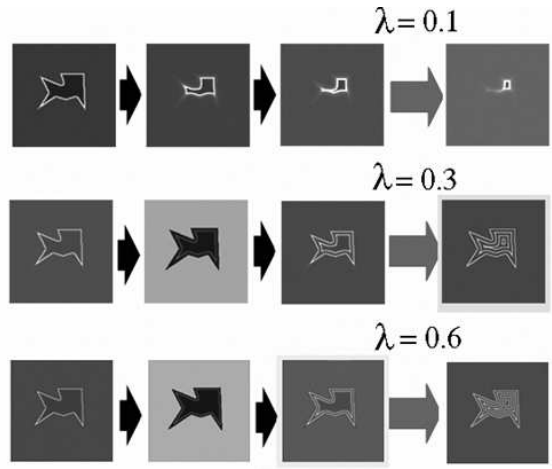


Fig. 6. Simulation of dynamic modes in Belousov-Zhabotinsky medium.

The inhibitor null-cline was used where ϕ/q =0. At the same time λ coefficient of the activator null-cline varied from λ=1 to λ=0.05 to describe moving intersection point from oscillatory to excitable state. Results of this simulation are shown in Fig. 6. It is possible to see the variety of intermediate dynamic modes including trains of pulses.

It should be concluded as a result of experimental study performed that processing of images by light-sensitive Belousov-Zhabotinsky media

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

could lead to revealing of trains of the image contours. These trains are determined by the intensity and duration of the light illumination, not structural features of image under investigation. It is urgent therefore to choose carefully exposures of the illumination to avoid artifacts in image processing.

4 Mathematical morphology vs image processing by Belousov-Zhabotinsky media

The methodological basis of image processing by reaction-diffusion media still remains vague enough till now. The process of the image evolution in the medium, not image processing, was considered virtually in the first investigation named "Image processing using light-sensitive chemical waves" [17] and in the following publications [18]. There were no attempts to understand how complete is the correspondence between modes of the image evolution in the medium and typical image processing operations used in practice. One of the most efficient ways to make clear this problem is to choose generally accepted and practically oriented information processing technique and to compare thoroughly basic principles, image processing operations and the variety of problems which could be solved.

The important approach to consider this problem – the theoretical basis and technique of image processing named "mathematical morphology", was elaborated during the last decades [23,24].

Binary mathematical morphology operates complex two-dimensional objects defined in discrete space. Object "A" is a set of pixel satisfied to the specific predetermined conditions:

In spite of numerical presentation of initial data the mathematical morphology operates images as a whole. Primitive operations of the mathematical morphology are dilation - $A \oplus B$ that increases the image, and erosion $A \Theta B$ that decreases it. Here some notion is introduced additionally: together with object A. It is structural element B which determine the character of the shape changes of the object A at its border.

Two basic operations of the mathematical morphology are defined based on these primitives: opening $A \circ B = (A \Theta B) \oplus B$, and closing $A \bullet B = (A \oplus B) \Theta B$.

Detailed consideration shows that combined use of opening and closing enables to perform practically all basic image processing operations: contour enhancement, segmentation and so on.

Information processing features of mathematical morphology approach and image processing by Belousov-Zhabotinsky media are shown in Fig 7.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Mathematical morphology	Chemical reaction-diffusion medium
Numerical method of image processing based on nonlinear transformations of images	Processing of images by a chemical medium based on nonlinear dynamic mechanisms
Operational data element is an image represented by a set of pixels	Operational data element is a single image
Consecutive pixel by pixel processing of the image by a numerical computer	Parallel processing of the image in all its points by a chemical reaction-diffusion medium
Multimode processing of images based on different types of structural elements	Only one circular structural element is virtually used
Dilation and erosion are initial image processing operations, two basic operations – opening and closing, could be reduced to initial ones	Contour$^{(+)}$ and contour$^{(-)}$ are initial image processing operations, opening and closing could be performed based on these operations
All practically actual black and white image processing operations could be performed	All practically actual black and white image processing operations could be performed
A variety of the half-tone image processing operations could be carried out	A variety of half-tone image processing operations equivalent to operations performed by mathematical morphology could be carried out

Fig. 7. Mathematical morphology and reaction-diffusion processing.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

There are evident differences between these two approaches. First of all one of them is a numerical method realized by modern digital computers, whereas another is a material realization of reaction-diffusion media capabilities. Nevertheless the information processing origin of these methods is adequate. It based on nonlinear mechanisms and wave character of information processing. Both of these approaches are biologically motivated. At the same time differences in information processing features of them aren't a matter of principle. Therefore all operations inherent in mathematical morphology could be carried out by Belousov-Zhabotinsky media.

4.1 Processing of black and white images

There are two operations performed by Belousov-Zhabotinsky media that could be considered as initial primitive ones. They are "contour$^{(+)}$" and "contour$^{(-)}$" operations and represent contour enhancement of the input image following by expanding (contour$^{(+)}$) or shrinking (contour$^{(-)}$) of the contour figure revealed (Fig. 8). The choice between these operations is determined by the choice of the negative or positive image form.

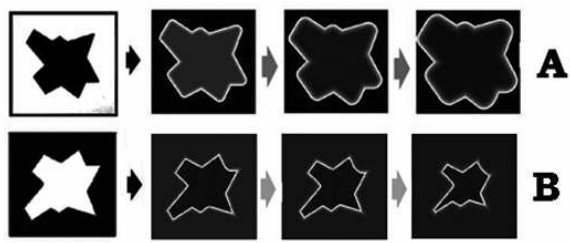


Fig. 8. Contour$^{(+)}$ (A) and contour$(-)$ (B) operations of Belousov-Zhabotinsky medium.

Mention here that considering image processing by Belousov-Zhabotinsky media positive and negative forms of the image should be distinguished.

To avoid vagueness in the following discussion let us define the positive image of a picture as image corresponding to typical picture inherent in human surroundings. If the notion of "typical picture" is uncertain (suppose in the case of geometrical figures, see, for instance, Fig. 2 and 6) let us define positive image as dark figure on the light background. Black and white and half-tone pictures will be used below. Images of

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

these pictures could be considered as a set of optical density values D_i corresponding to each point of the picture ($0 < D_i < D_\infty$, where D_∞ is a maximum value of the optical density). The negative image of the picture was defined as a set of inverted density values ($D_i^N = D_\infty - D_i$).

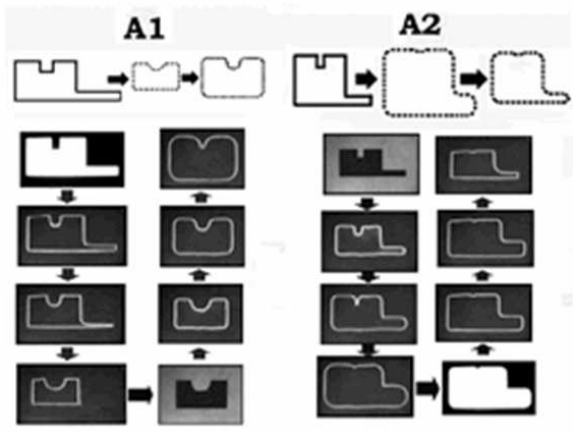


Fig. 9. Opening (A1) and closing (A2) operations carried out by Belousov-Zhabotinsky medium.

These primitive operations are adequate to dilation and erosion operations of the mathematical morphology. Moreover, it is possible to reproduce also opening and closing operations, using contour$^{(+)}$ and contour$^{(-)}$ primitives performed by Belousov-Zhabotinsky medium (Fig. 9). Therefore it is possible to conclude that reaction-diffusion medium can perform image processing operations typical for mathematical morphology.

All known image processing operation carried out by the mathematical morphology technique could be performed by by Belousov-Zhabotinsky media. Most of them could be carried out using contour (+) and contour(-) operations, equivalent to dilation and erosion. A lot of examples were published elsewhere (see, for instance, [25]). They embrace smoothing of immaterial features of the figure (enhancement its general shape) and segmentation of the figure, enhancement and removing of small features

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

of the image, thinning, skeleton and Voronoi diagram calculation and so on.

Fig. 10. Opening (A) and closing (B) operations performed by Belousov-Zhabotinsky medium.

Opening and closing operations. performed by Belousov-Zhabotinsky media enable to carry out more complex image processing operations. As example defect repair and removing noises from an image can be mentioned (Fig. 10). Here consecutive combined use of contour(+) and contour(-) gives the opportunity to reconstruct the initial image.

4.2 Half-tone images and images having several levels of brightness

Considerable and notable possibilities of the complex pattern analysis are opened in the case of half-tone images. In this case an image under consideration is first transformed into its negative form. This transformation is the continuous process when the negative image appears step by step beginning from the most dark (or the most bright, depending on positive or negative form of initial image) fields of the image. This capability of Belousov-Zhabotinsky media is one of the most remarkable medium feature.

Namely, Belousov-Zhabotinsky medium is the natural realization of the temporal sequence processor (see, for instance, [33]) that transforms complex spatial distribution of visual information into temporal sequence of its fragments. Because of this feature Belousov-Zhabotinsky media are capable to solve complex practically important problems. Let us give some examples.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Fig. 11. Enhancement of the hidden image by Belousov-Zhabotinsky medium.

The "hidden image" is a fragment of the picture having brightness very close to the brightness of the background. The example of this situation is shown in Fig. 11. The difference in brightness between the eagle image and the background is 10 units of Photoshop HSB model (the brightness of the picture was increased greatly to show the content of the picture, Fig. 11). The evolution of the picture in the Belousov-Zhabotinsky medium enables to enhance the hidden image in spite of a very big difference of its brightness in comparison with the background.

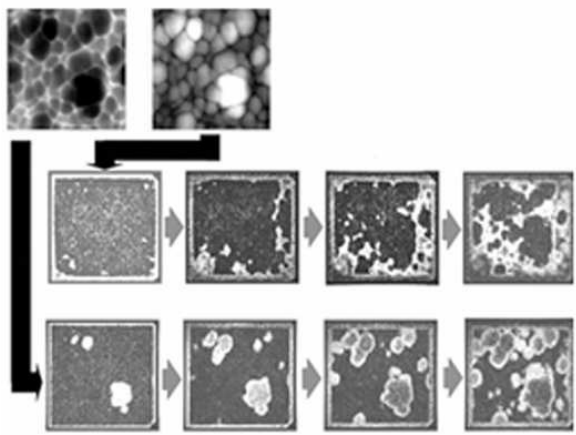


Fig. 12. Processing of half-tone images by Belousov-Zhabotinsky medium.

The important part of the image analysis in medicine, material science and some other fields is to reveal consecutively fields of the image having increasing (or decreasing) brightness. Examples of this process performed by Belousov-Zhabotinsky medium are shown in Fig. 12.

One of the useful tasks of the half-tone image processing is watershed operations. They give the opportunity to enhance the relief of the hill-like

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

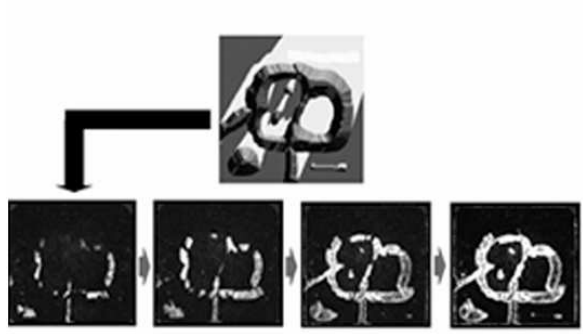


Fig. 13. Watershed operations carried out by Belousov-Zhabotinsky medium.

image. The example of the watershed operation performed by Belousov-Zhabotinsky medium is shown in Fig. 13.

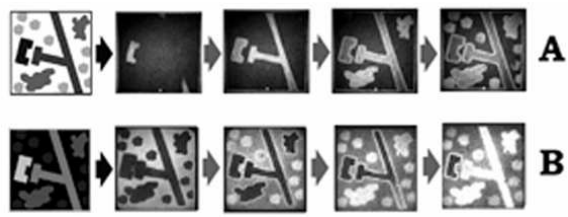


Fig. 14. Processing of multi-view aerial images by Belousov-Zhabotinsky medium.

Very important is processing patterns taken from aircrafts or satellites, and, particularly, urban roads detection [34]. Evolution of the test pattern similar to multi-view aerial imagery is shown in Fig. 14 Given both positive and negative forms of the pattern, fragments of the pattern having different brightness are revealed consecutively depending on the brightness of the fragments (Fig. 14).

An attempt to use mathematical morphology technique for urban road detection was performed lately [34]. The urban map and results of the preliminary detection of the road network by the mathematical morphology technique are shown in Fig. 15A,B1. The detection of the road network by Belousov-Zhabotinsky type medium using the same urban map is shown in Fig. 15B2. Easy to see, that these two road networks are in a good correspondence.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

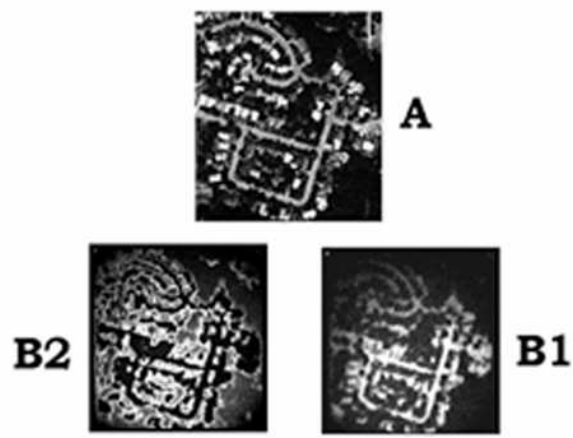


Fig. 15. Urban road detection by mathematical morphology technique (B1) and by Belousov-Zhabotinsky medium (B2) using the same urban map (A).

Some of the examples considered above were discussed in publications devoting to the mathematical morphology technique. The correspondence between these mathematical morphology operations and results obtained here using Belousov-Zhabotinsky media is satisfactory.

5 Some concluding remarks

In the nineties of the last century Lee A. Rubel, one of the fathers in the field of unconventional computing, said to J.W. Mills [35]:

> The future of analog computing is unlimited. As a visionary, I see it eventually displaying digital computing, especially in the beginning, in partial differential equations and as a model in neurobiology. It will take some decades for this to be done. In the meantime, it is my belief, it is a very rich and challenging field of investigations, although (or may be because) it is not in the current fashion.

Nowadays, a decade later, problems that could be solved by reaction-diffusion media are more and more complicated and practically purposeful.

Information processing by reaction-diffusion media is virtually the analog computation process which due to neural net medium architecture simulate some functions of the cerebral cortex activity. The basic

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

problem is that the mechanisms of problem solution by the cerebral cortex are yet far from detailed understanding. If they are known, it would enable to create powerful means solving single artificial intelligence problems based on reaction-diffusion systems. However lack of knowledge along with von Neumann principles that are obvious in our thinking leads nowadays to ineffective use of these unique means. At the same time, algorithms guessed rightly enable not only to elaborate powerful information devices but to make clearer to some extent mechanisms of the cerebral cortex activity [25].

References

1. CALMEC, http://www.calmec.com
2. HP-UCLA collaboration http://www.hp.com/hpinfo/news-room/press/23jan02b.htm
3. Moravec, H., When will computer hardware match the human brain? Journal of Evolution and Technology 1, 1998.
4. Molecular Computing, Sienko, Tanya, Adamatzky, Andrew, Rambidi, Nicholas. G., Conrad, M., Eds., The MIT Press, Cambridge, Massachusetts, London, England, 2003.
5. Conrad M., Molecular information processing in the central nervous system. Part1: Selection circuits in the brain, Lecture Notes in Biomathematics. Physics and Mathematics of the Nervous System, 4, Springer-Verlag, Berlin-Heidelberg-New York, 1974.
6. Conrad M., On design principles of a molecular computer, Communications of ACM 28, 464-480, 1985
7. Conrad M., The brain-machine disanalogy, BioSystems 22, 197-213, 1989
8. Rambidi N.G., Zamalin V.M., Sandler Yu.M., Molecular information processing devices and artificial intelligence problems, J.Molec. Electron. 4, S39, 1988
9. Rambidi N.G., Biomolecular computing: from the brain-machine disanalogy to the brain-machine analogy, BioSystems 33, 45-54, 1994
10. Seibert M., Waxman A.M., Spreading activation layers, visual saccades and invariant representation for neural pattern recognition systems, Neural Netw. 2, 9, 1989
11. Rubel Lee A., Digital simulation of analog computations, and Church's thesis, The Journal of Symbolic Logic 54, 1011-1017, 1989
12. Rubel Lee A., The extended analog computer, Advances in Applied Mathematics 14, 39-50, 1993
13. Natschlager T., Maass W., Markram H., The "liquid computer". A novel strategy for Real-time computing on time series, Special Issue on Foundations of Information Processing of TELEMATIK 8, 39-43, 200
14. Adamatzky A., Computing in reaction-diffusion and excitable media. Case studies0f unconventional processors, In Molecular computing, Sienko T., Adamatzky A., Rambidi N.G., Conrad M., Eds., The MIT Press, Cambridge, Massachusetts, London, England, 2003

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

15. Kuhnert L., A new optical photochemical memory device in a light-sensitive chemical active medium, Nature 319, 393-394, 1986
16. Kuhnert L, Photochemische manipulation von chemischen wellen, Naturwissenschaften 73, 96-97, 1986
17. Kuhnert L., Agladze K.I., Krinsky V.I., Image processing usinf light-sensitive chemical waves, Nature 337, 1989
18. Rambidi N.G., Chemical-based computing and problems of high computational complexity. The reaction-diffusion paradigm, In Molecular computing, Sienko T., Adamatzky A., Rambidi N.G., Conrad M., Eds., The MIT Press, Cambridge, Massachusetts, London, England, 2003
19. Adamatzky A., de Lacy Costello B., Experimental reaction-diffusion preprocessor for shape recognition, Phys. Lett. A
20. Unger, S.H., Cellular logic, Proc. IRE 40, ?10, 1959.
21. Blum. H., A transformation for extracting new descriptors of shape, in: Symposium: Models for perception of speech and visual forms, Whaten-Dunn, W., Ed., MIT Press, Cambridge, 1967.
22. Rambidi, N.G., Maximychev., A.V., Usatov, A.V., Molecular neural network devices based on non-linear dynamic media, BioSystems 33, 125-137, 1994.
23. Serra, J., Introduction to mathematical morphology, Computer vision, Graphics, and Image Processing 35, 283-305, 1986.
24. Young, N., Mathematical Morphology, `http://dmsun4bath.ac.uk/research/morphology/morphology.htm`
25. Rambidi, N.G., Ulyachin, S.G., Shishlov, D.E., Neganov, V.A., Tsvetkov, A.S., Information processing by chemical reaction-diffusion media: from computing to vision, Intern. J. Unconventional Computing, in press.
26. Field R.J., Burger, M., Eds., Oscillations and traveling waves in chemical systems, Wiley-Interscience, New York, 1985.
27. Rovinsky, A.B., Turing bifurcation and stationary patterns in the ferroin-catalyzed Belousov-Zhabotinsky reaction, J. Phys. Chem. 91, 4606-4513, 1987
28. Kadar, S., Amemia, T., Showalter, K., Reaction mechanism for light sensitivity of the Ru(bpy)32+ - catalyzed Belousov-Zhabotinsky reaction, J. Phys. Chem. A 101, 8200- 8206, 1997.
29. Sendina-Nadal, I., Roncaglia, D., Vives, D., Perez-Munuzuri, V., Gomez-Gesteira, M., Perez-Villar, V., Echave, E., Casademunt, J., Ramirez-Poiscina, L., Sagues, F., Percolation thresholds in chemical disordered excitable media, Phys. Rev. E 58, R1183-R1186, 1998.
30. Sendina-Nadal, I., Munuzuri, A.P., Vives, D., Perez-Munuzuri, V., Casademunt, J., Ramirez-Poiscina, L., Sancho, J.M., Sagues, F., Wave propagation in a medium with disordered excitability, Phys. Rev. Lett. 80, 5437-5440, 1998
31. Akagi, T., Okazaki, N., Yoshinobu, T., Matsumura-Inoue, T., Comparative studie of chemical waves and temporal oscillations 9n the -catalyzed photosensitive Belousov-Zhabotinsky reaction, Chem.Phys. Lett. 328, 214-220, 2000

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

32. Kadar, S., Amemiya, T., Showalter, K., Reaction mechanism for light sensitivity of the Ru-catalyzed Belousov-Zhabotinsky reaction, J. Phys. Chem. 101, 8200-8206, 1997
33. Ray, S.R., Kargupta, H., A temporal sequence processor based on the biological reaction-diffusion process `https://leitl.org/docs/rayPL.ps.gz`
34. Zhang, Ch., Murai, Sh., Baltsavias, E., Road network detection by mathematical morphology, Proc. of ISPRS Workshop "3D geospatial data production: meeting application requirements" ,7-9, April, Paris, France, 1999, pp.185-200
35. Modern Analog Field Computing, `http:/www.cs.indiana.edu/~jwmills/MAFC/mafc.html`

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Chemical organization theory as a theoretical base for chemical computing

Naoki Matsumaru, Florian Centler, and Peter Dittrich

Bio Systems Analysis Group
Jena Centre for Bioinformatics (JCB) and
Department of Mathematics and Computer Science
Friedrich-Schiller-University Jena
D-07743 Jena, Germany
http://www.minet.uni-jena.de/csb/

Abstract. In chemical computing, the solution appears as an emergent global behavior based on the local reaction rules. A theoretical analysis to cope with the emergent behavior is desired. In this paper, we demonstrated how chemical organization theory can help in designing and understanding chemical computing systems. As an example, a reaction network implementing an XOR logic gate and a flip-flop logic circuit are used. Reaction networks are decomposed into a hierarchy of self-maintaining sub-networks (called organizations) using stoichiometric information. The dynamical behavior of the reaction system is explained as a movement between organizations. The theoretical analysis provides an insight into the potential behavior of chemical reaction system. The encouraging results suggest that the theory of chemical organizations contributes to a theoretical framework for chemical computing.

1 Introduction

By employing a large number of simple components interacting with each other in an orchestrated way, biological systems invented a variety of information processing mechanisms, which are robust, self-organizing, adaptable, decentralized, asynchronous, fault-tolerant, and evolvable. This principle of biological information processing has been exploited to cope with the fast-growing complexity of technical information processing systems [1–3]. Since all known life forms process information using chemical processes [4], the chemical reaction metaphor has been proposed as a

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

source of inspiration [5, 6]. Using chemical reactions for formal computations has initially been suggested by Banâtre and Métayer [5]. In their GAMMA system [7], a chemical reaction is defined as a rewriting operation on a multiset, mimicking a well-stirred reaction vessel. In order to capture the spatial context of chemical systems, chemical rewriting systems have been extended to the chemical abstract machine (CHAM) [8], P-Systems [9, 10] stressing the importance of membranes, and MGS [11] allowing arbitrary topologies [12].

The difficulty of engaging the chemical reaction process for computing is that the solution appears as an emergent global behavior based on the manifold local interactions [13]. An emergent behavior of biological systems is simulated by combining simple biochemical signaling pathways in [14]. As recently demonstrated by Tsuda et al. with *Physarum* [15], the discrepancy between local and global behavior may also be problematic in a practical case. A satisfying theory of emergence is lacking [16].

This paper should contribute towards the establishment of a theoretical analysis of the emergent behavior in chemical computing. We suggest chemical organization theory [17, 18] as a tool helping to construct (program) and analyze (describe and understand) chemical computing systems. Comparable to different methods (e.g., [19]), chemical organization theory allows to decompose a reaction network into a hierarchy of self-maintaining sub-networks, called organizations, only using stoichiometric information. Although other approaches to divide biochemical reaction network into sub-networks in terms of functional modules have been investigated [20], a rigorously proven relation between sub-networks and potential emergent dynamics is mostly lacking. Since emergent properties appear in a dynamical situation, the theoretical method for analyzing emergence must take it into consideration.

Inspired by Fontana and Buss [21], we define a (chemical) organization as a set of molecular species that is (algebraically) closed and (stoichiometrically) self-maintaining [18]. It is important to note, that when we talk about organizations, we abstract details like concentration levels or the spatial distribution of a chemical species. On this relatively high level of abstraction, a system state is characterized only by the molecular species present and we can describe the dynamics of a system more qualitatively, namely, as a movement between sets of species instead of a movement in a more complex state space [17].

Borrowing the notion of chemical organizations defined as closed and self-maintaining sets of molecular species, we demonstrate in this paper, how the algebraic analysis of chemical reaction networks helps to understand the emergent dynamical behavior of (artificial) chemical computing. In Section 2, we describe the concepts from chemical organization

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

theory needed here. As an example of chemical computing, an XOR logic gate is implemented using an (artificial) chemical reaction network with the help of chemical organization theory in Section 3. A flip-flop logic circuit consisting of two NAND gates serves as another example in Section 4. Finally, in Section 5, we discuss the potential of the theory as a theoretical base for emergence analysis.

2 Chemical Organization Theory

The target of chemical organization theory are reaction networks. A formal definition of a reaction network is given by a tuple $\langle \mathcal{M}, \mathcal{R} \rangle$ called an algebraic chemistry:

Definition 1 (algebraic chemistry). *Given a set $\mathcal{M}$ of molecular species and a set of reaction rules given by the relation $\mathcal{R} : \mathcal{P}_M(\mathcal{M}) \times \mathcal{P}_M(\mathcal{M})$. We call the pair $\langle \mathcal{M}, \mathcal{R} \rangle$ an* algebraic chemistry*, where $\mathcal{P}_M(\mathcal{M})$ denotes the set of all multisets with elements from $\mathcal{M}$.*

Adopting a notion from chemistry, a reaction rule is written as follows: $\mathcal{R} = \{s_1 + s_2 + \cdots + s_n \rightarrow s'_1 + s'_2 + \cdots + s'_{n'} | \forall i : s_i, s'_{i'} \in \mathcal{M}\}$. We also rewrite $a + a \rightarrow b$ to $2a \rightarrow b$ for simplicity. Note that "+" is not an operator but a separator of elements. A reaction rule is similar to a rewriting operation [12] on a multiset where multiple occurrences of the same element are allowed.

A set of molecular species is qualified as an organizational sub-network if the following two properties are satisfied: closure and self-maintenance. A set of molecular species is closed when all reaction rules applicable to the set cannot produce a molecular species that is not in the set. This is similar to the algebraic closure of an operation in set theory.

Definition 2 (closure). *Given an algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$, a set of molecular species $C \subseteq \mathcal{M}$ is closed, if $B \in \mathcal{P}_M(C)$ for every reaction $(A \rightarrow B) \in \mathcal{R}$ with $A \in \mathcal{P}_M(C)$.*

Self-maintenance of the set is defined as follows:

Definition 3 (self-maintenance). *In an algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$, the i-th molecular species is $m_i \in \mathcal{M}$ and the j-th reaction rules is $(A_j \rightarrow B_j) \in \mathcal{R}$. Given a stoichiometric matrix $\mathbf{M} = (m_{i,j})$ where $m_{i,j}$ denotes the number of molecules of species i produced in reaction j, a set of molecular species $S \subseteq \mathcal{M}$ is self-maintaining, if there exists a flux vector $\mathbf{v} = (v_{A_1 \rightarrow B_1}, \ldots, v_{A_j \rightarrow B_j}, \ldots, v_{A_{|\mathcal{R}|} \rightarrow B_{|\mathcal{R}|}})^T$ satisfying the following three conditions:*

1. $v_{A_j \rightarrow B_j} > 0$ *if* $A_j \in \mathcal{P}_M(S)$

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

2. $v_{A_j \to B_j} = 0$ *if* $A_j \notin \mathcal{P}_M(S)$
3. $f_i \geq 0$ *if* $m_i \in S$ *where* $(f_1, \ldots, f_i, \ldots, f_{|\mathcal{M}|})^T = \mathbf{Mv}$.

These three conditions can be read as follows: When the j-th reaction is applicable to the set S, the flux $v_{A_j \to B_j}$ must be positive (Condition 1). All other fluxes are set to zero (Condition 2). Finally, the production rate f_i for all the molecular species $m_i \in S$ is nonnegative (Condition 3). Note that we have to find only one such flux vector in order to show that a set is self-maintaining.

Taking closure and self-maintenance together, we arrive at an organization:

Definition 4 (organization). *A set of molecular species $O \subseteq \mathcal{M}$ that is closed and self-maintaining is called an organization.*

We visualize the set of all organizations by a Hasse diagram, in which organizations are arranged vertically according to their size in terms of the number of their members (e.g., Figure 1). Two organizations are connected by a line if the lower organization is contained in the organization above and there is no other organization in between.

3 Case Study I: A Chemical XOR

To demonstrate how chemical organization theory can be used for chemical computing, an (artificial) chemical reaction network is designed to implement an XOR logic gate.

The two input and one output variables of the XOR gate are labeled as a, b, and c, respectively. Two molecular species are assigned to each variable to represent its boolean value. For example, the lowercase species a represents the state $\mathsf{a} = 0$. while the uppercase species A stands for $\mathsf{a} = 1$. Thus, the XOR reaction network consists of 6 molecular species $\{a, A, b, B, c, C\} = \mathcal{M}$. Note that with this kind of coding, the logic state of a variable, say a, can be undefined if the reaction vessel contains molecules of type a and A at the same time or if the reaction vessel contains neither a nor A.

The reaction rules representing an XOR gate are derived directly from the truth table: A boolean computation $1 \oplus 0 = 1$, for example, is converted to the reaction rule $A + b \to C$ so that we arrive at four reaction rules: $\mathcal{R}_1 = \{a + b \to c,\ a + B \to C,\ A + b \to C,\ A + B \to c\}$. The Hasse diagram in Figure 1 (A) shows the hierarchical organizational structure of this reaction network. Twenty eight sets of molecular species are found to be organizations, and the rest of 36 sets does not satisfy either of the closure or self-maintenance property.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

The set $\{a, b\}$, for example, is not an organization because it is not closed. The reaction $a+b \rightarrow c$ is applicable and produces a new molecular species c that is not a member of the set $\{a, b\}$. The set $\{a, b, c\}$ is closed but not an organization because it is not self-maintaining. A production rate vector $\boldsymbol{f}$ is calculated as follows:

$$\boldsymbol{f} = \boldsymbol{M}\boldsymbol{v} = \begin{pmatrix} -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -v_1 \\ 0 \\ -v_1 \\ 0 \\ v_1 \\ 0 \end{pmatrix} \tag{1}$$

where a stoichiometric matrix $\boldsymbol{M}$ is multiplied by a flux vector $\boldsymbol{v}$, $v_1 > 0$, satisfying the condition 1 and condition 2 from the definition of self-maintenance. The third condition cannot be satisfied because, for the molecular species $a = m_1$ and $b = m_3$, production rates f_1 and f_3 cannot be greater than 0.

In this particular case of the reaction network, all organizations consist of combinations of molecular species that do not react with each other. A set of molecular species that does not react[1] is obviously closed. Furthermore, it is self-maintaining since with no reactions among the set species, a zero flux vector $\boldsymbol{v} = \boldsymbol{0}$ fulfills all required conditions of Definition 3.

With the species set of an organization being closed and self-maintaining, it is more likely to observe the presence of molecular species of an organization than of any other species combination in the reaction vessel. If the dynamics of the reaction network is modelled using ordinary differential equations, there exists a related organization for every fixed point of the system [18].

When implementing XOR using only the four reactions mentioned above, we can see in Figure 1 (A) that half of the organizations represent contradictory situations, e.g., a and A being present in an organization simultaneously. If both the uppercase and lowercase species of a variable are present, the variable is computed to have a value of 0 and 1 at the same time. In order to avoid these cases, destruction reactions of the form $a+A \rightarrow \emptyset$ are added for each boolean variable a, b, and c. These reactions ensure that contradictory situations in the reaction vessel vanish quickly. The resulting set of reactions for the chemical XOR gate is: $\mathcal{R}_2 = \mathcal{R}_1 \cup \{a + A \rightarrow \emptyset,\ b + B \rightarrow \emptyset,\ c + C \rightarrow \emptyset\}$. Rule set $\mathcal{R}_2$ represents the XOR gate without any input specified. Its Hasse diagram of organizations is

[1] which means that there is also neither decay nor outflow

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

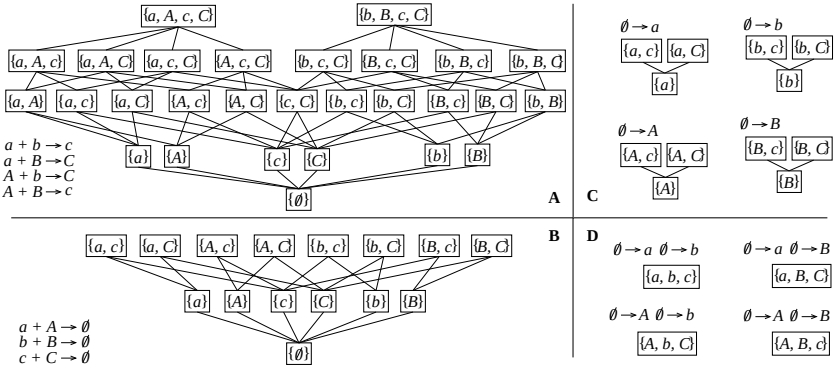


Fig. 1. Hierarchy of organizations for the chemical reaction network implementing an XOR logic gate. (A) The network consists only of the reactions directly derived from the truth table (see bottom left). (B) Destruction reactions are added to exclude contradictions. (C) Defining one input. (D) Adding the second input. The hierarchy of organizations collapses from (A) to (D), with the desired output as the only organization left in (D).

shown in Figure 1 (B). The number of organizations is reduced from 28 to 15.

Setting an input variable to a certain value is equivalent to adding an inflow reaction to the network. The inflow reaction creates a molecule representing the desired value of the variable from "nothing", e.g., $\emptyset \rightarrow a$ corresponds to setting $\mathsf{a} = 0$. Note that the network structure is changed due to the different input reactions so that the outcome of the algebraic analysis changes for each input. We start with providing one input only, leaving the other input variable undefined. Figure 1 (C) shows the results for the four resulting networks $\mathcal{R}_2 \cup \{\emptyset \rightarrow a\}$, $\mathcal{R}_2 \cup \{\emptyset \rightarrow A\}$, $\mathcal{R}_2 \cup \{\emptyset \rightarrow b\}$, and $\mathcal{R}_2 \cup \{\emptyset \rightarrow B\}$, respectively. We can see that providing one input signal has further reduced the behavioral freedom of the reaction system. Only three combinations of molecular species are left, which may be encountered in the reaction vessel as a stationary state. Furthermore we can see that – in this special case – the output is not determined from a stoichiometric point of view since, in all four Hasse diagrams, sets containing c and C are found to be closed and self-maintaining.

When we finally provide both inputs, the Hasse diagram of organizations collapses so that only one organization remains for every input condition (Figure 1 (D)). This implies that, no matter how we chose the dynamics, no other molecular species than those of the organization can sustain themselves in the reaction vessel regardless of the ini-

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

tial state. We can see that the remaining organization contains the desired output molecule c or C, respectively. The analyzed networks are $\mathcal{R}_2 \cup \{\emptyset \to a, \emptyset \to b\}$, $\mathcal{R}_2 \cup \{\emptyset \to A, \emptyset \to b\}$, $\mathcal{R}_2 \cup \{\emptyset \to a, \emptyset \to B\}$, and $\mathcal{R}_2 \cup \{\emptyset \to A, \emptyset \to B\}$.

3.1 Dynamical Simulation

To validate the results from applying organization theory to the XOR reaction network, stochastic simulations are performed using the simulator packages *Copasi* [22] and MGS [11]. Figure 2 shows a typical simulation run. The influx is defined as an irreversible constant flux with kinetic parameter set to 1. For all other reactions, we chose irreversible mass action kinetics. The parameters for the destruction reactions are set to $k = 0.1$, and those for the reactions in $\mathcal{R}_2$ are set to $k = 0.001$. At several simulation times, the input is changed in order to observe the switching of the XOR gate. Initially, there exist no molecular particles in the reactor, and two influxes of a and b are present. This corresponds to the case in which both the input variables a and b are set to 0. Since molecular species c is generated, the output is computed to c = 0.

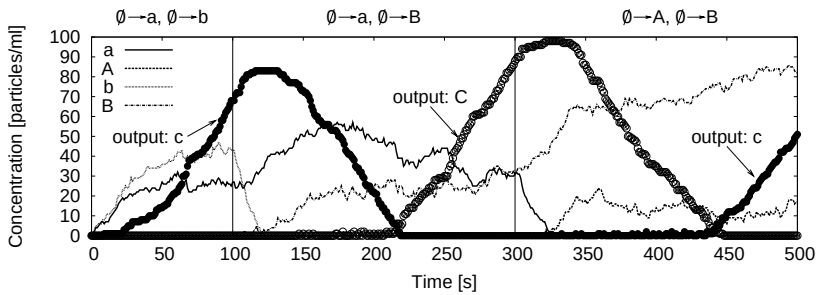


Fig. 2. Dynamic behavior of the chemical reaction network implementing an XOR logic gate. The time course of all 6 molecular species is shown. Irreversible mass action kinetics is assumed for all reactions. Reaction rates are set to $k = 0.001$ for reactions based on the truth table. Reaction rates of destruction reactions are set to $k = 0.1$. For all irreversible constant influxes like $\emptyset \to A$, the rates are set to $k = 1$. The reaction system is stochastically simulated with the biochemical network simulator *Copasi* [22] using a compartment size of 1 ml. See text for details.

At simulation time 100 s, the contents of input variable b is switched to 1 by replacing the influx of molecular species b with the influx $\emptyset \to B$.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

The molecular particles b and c with still high concentrations from the previous computation deteriorate and finally vanish. The desired output C does not appear until the time point of approximately 200 s. Then, instead of a, the molecular species A is applied as an input starting from simulation time 300 s. The remaining molecular species a and C from the previous computation decay first and the desired answer c emerges in the end.

As seen from the dynamical simulation, the computational result represented by the qualitative final state of the reaction vessel is independent of the initial state. The applied continuous input is the only factor deciding on the final state. The output molecules are generated continuously while undesired species are removed from the reaction vessel by collisions with their anti-particles. When applying two inputs, the analysis of the reaction network revealed that only one organization exists, predicting only one species composition (the species of that organization) to be closed and self-maintaining, and thus likely to be observed in the reactor. The stochastic simulation confirms the discussion.

4 Case Study II: A Chemical Flip-Flop

In this section, we apply our approach to a more complicated example: the flip-flop logic circuit. As opposed to the previous example, a flip-flop circuit is bistable, which is achieved by two feedback connections. When we analyze the organizations of our chemical instantiation of the flip-flop, the bistability of the circuit will also become apparent. This allows us to explain the dynamical behavior of the chemical flip-flop in terms of chemical organization theory on an abstract level, which does not need to refer to concentration levels.

The RS (Reset and Set) flip-flop circuit consists of two NAND gates connected in parallel as shown in Figure 3 (A). The behavior can be described by the truth table as shown in Figure 3 (B). The output of one logic gate is connected to one of the two inputs of the other gate, forming a feedback loop. The "set" operation $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (0, 1)$ changes the output Q to 1, and the "reset" operation $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (1, 0)$ sets Q to 0. When both inputs are set to 1, the output is kept as in the previous state. The one-bit information whether the output Q has been 0 or 1 is stored by the "hold" operation, i.e. $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (1, 1)$. Normally, the input $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (0, 0)$ is prohibited because the circuit will go into a state where $\mathsf{Q} = 0$ and $\bar{\mathsf{Q}} = 0$. Application examples for the flip-flop are memory and counter circuits.

Starting from the logic circuit diagram, a chemical reaction network is constructed in the same way as for the XOR gate. The input variables

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

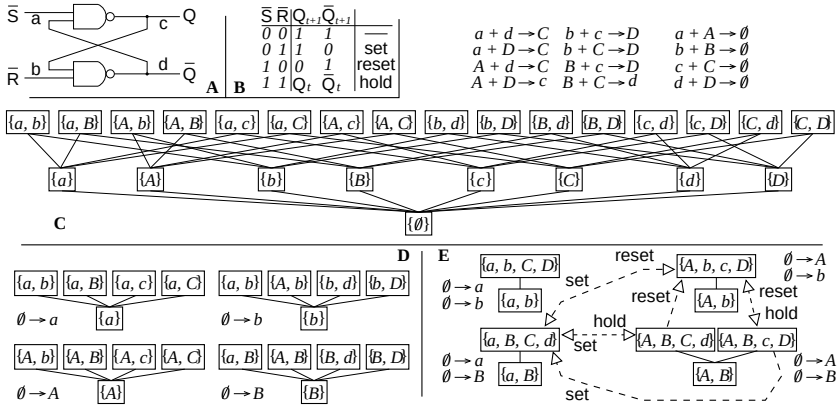


Fig. 3. A chemical reaction network is constructed to implement the RS flip-flop logic circuit with the help of chemical organization theory. The logic structure of the circuit is shown in (A), and the behavior is shown in (B) as a truth table. The chemical reaction network implementing the flip-flop with 8 molecular species and 12 reaction rules, and the hierarchical organizational structure of the network is shown in (C). By adding influx to the network, the number of organizations is reduced as shown in (D) and (E). When the molecular species related to both input variables a and b are supplied continuously as an influx, the organizational structure corresponds to the operation of the flip-flop logic circuit. See text for details.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

of the circuit are labeled a and b, and the output variables are c and d. For one NAND gate, the input variable is a and d, and the output is c. Four reaction rules are derived from the NAND truth table. Another four reaction rules are derived for the other NAND gate with b and c as input and d as output. Finally, destruction reactions for all variables are added so that contradictory states degenerate, as in the previous example. The flip-flop reaction network is constituted by 8 molecular species $\mathcal{M} = \{a, A, b, B, c, C, d, D\}$ and 12 reaction rules shown in Figure 3 (C).

When we applied our analysis to the flip-flop reaction network without any input flow, we found 25 organizations consisting of up to two molecular species, which do not react (Figure 3 (C)). When two input species are injected into the system, the number of organizations found in the network is reduced to two or three for each input case as seen in Figure 3 (E). When the output species c, C, d, and D are not present in the reactor, no reaction occurs due to the feedback loops. Thus, the smallest organization contains only the two inflow species. Above it, the designated output species are included in the organizations. This implies that the presence of the output species c, C, d, or D in the reaction vessel is necessary for the flip-flop operation. In other words, the input molecular species alone cannot generate the organization representing an operational mode of the flip-flop.

The operation of the flip-flop can be described by transitions between organizations containing output species: The set and reset operation move the reaction system to the states corresponding to organization $\{a, B, C, d\}$ (set) and $\{A, b, c, D\}$ (reset). Recall that for the set and reset operation we add $\{\emptyset \to a, \emptyset \to B\}$ and $\{\emptyset \to A, \emptyset \to b\}$ to the set of reaction rules, respectively.

For the hold operation ($\emptyset \to A, \emptyset \to B$), the flip-flop has two stable states represented by the organizations $\{A, B, C, d\}$ and $\{A, B, c, D\}$. If the reaction vessel had been in organization $\{a, B, C, d\}$ previously, it will move into organization $\{A, B, C, d\}$; and if it had been in organization $\{A, b, c, D\}$ before, it will move into organization $\{A, B, c, D\}$. Symbolically speaking, the lowercase input species is replaced by its uppercase due to the input change, but the output state remains unchanged.

For the sake of completeness, the cases in which only one influx is added to the network are shown in Figure 3 (D). A set of molecular species that no reaction rule (including decaying reaction) is applicable is an organization because no molecular species is produced (closed) or consumed (self-maintaining). The smallest organizations contain only the input species with the influx. Adding one species that does not interact with the input species forms another organization. Since adding

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

another species makes a reaction rule applicable and molecular species are used up with no reproduction, there exists no organization of size greater than two.

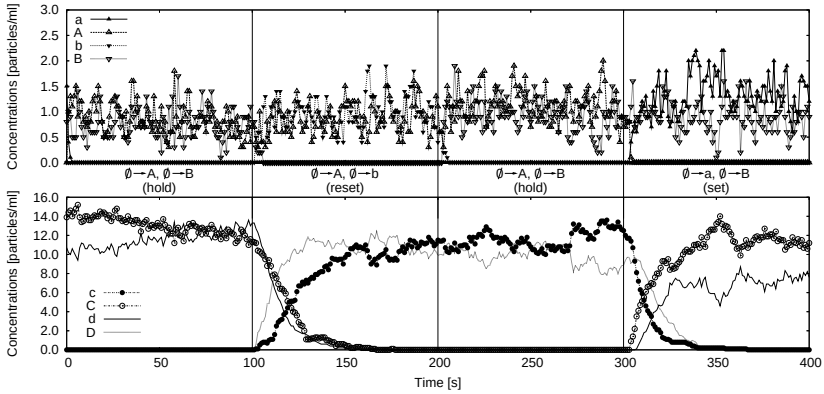


Fig. 4. Dynamic behavior of the chemical reaction network implementing a RS flip-flop logic circuit shown in Figure 3. The top figure shows the time course of the input species a, A, b, and B. The bottom figure shows the concentrations of the output species. Irreversible mass action kinetics is assumed for all reactions. The kinetic parameters are set to $k = 1$ for all zero order reactions like $\emptyset \rightarrow A$. The kinetic parameter is set to $k = 0.001$ for decaying reactions. For the other second order reaction rules to produce output species c, C, d, or D, the kinetic parameter is $k = 0.1$. The reaction system is stochastically simulated with the biochemical network simulator *Copasi* [22] using a compartment size of 10 ml.

4.1 Dynamical Simulation

In order to validate the discussion of the previous section we performed stochastic simulations (using MGS [11] and Copasi [22]) of reaction systems implementing the chemical flip-flop. Figure 4 shows a typical simulation run. The influx is defined as an irreversible constant flux with kinetic parameter set to 1. For all other reactions we chose irreversible mass action kinetics. The kinetic parameters are set to 0.1 for the second order reactions that produce output species c, C, d, or D. For decaying reactions, the kinetic parameters are set to 0.001. During the first "hold" phase (0 - 100 s), the concentration of C and d remain high. In the following "reset" phase (100 - 200 s), the input reactions $\emptyset \rightarrow A$ and $\emptyset \rightarrow b$ are added to "reset" the system so that the output variable c is set to 0.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

The concentration of C and d decreases gradually and species c and D accumulate in the reaction vessel. The system eventually reaches a state in which only members of the organization $\{A, b, c, D\}$ are present as expected from the algebraic analysis. In the next phase (200 - 300 s), the input flow of b is replaced by that of B, $\emptyset \rightarrow B$, to "hold" the output of the previous phase. Although the input species have changed, no qualitative change is detected in the bottom graph. Finally, in the last phase (300 - 400 s), the "set" operation is executed by changing the influx $\emptyset \rightarrow A$ to $\emptyset \rightarrow a$. The transition to the state represented by the organization $\{a, B, C, d\}$ is observed.

Although the same input species are injected in the two "hold" phases, the states of the reaction vessel in terms of molecular species present are different depending on the initial conditions. The bistable behavior of the flip-flop circuit is implemented dynamically by the chemical reaction system, which we have expected from our theoretical analysis in the previous section. The reaction network with the two influxes $\emptyset \rightarrow A$ and $\emptyset \rightarrow B$ has two organizations with four species: The system state in the first "hold" phase corresponds to the organization $\{A, B, C, d\}$, and the members of the organization $\{A, B, c, D\}$ are present during the second "hold" phase.

5 Concluding Remarks

In this paper, a theoretical analysis method is proposed that helps to discover and implement computing capabilities in (artificial) chemical reaction networks. Given a list of molecular species and a list of reaction rules, the reaction network is decomposed into a hierarchy of closed and self-maintaining sub-networks called organizations. We have shown that the hierarchy of organizations helps to assess the emergent dynamical behavior of the chemical reaction network under study. When the approach is applied to a reaction network implementing an XOR logic gate, the hierarchy of organizations helps to predict its emergent dynamical behavior. Defining different inputs leads to different organizations corresponding to the various states of the gate. As another example, a flip-flop logic circuit in which two NAND gates are connected to each other via feedback loops is implemented by a chemical reaction network. From the theoretical analysis, the bistable behavior is reflected by two organizations found in the network. Using chemical organization theory, we were able to explain the properties of the chemical flip-flop in a new, comprehensible way by referring to the Hasse diagram of organizations (Figure 3). Furthermore the "constructive" dynamics of the flip-flop could be described as a movement between organizations (Figure 3 (E)). This

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

description is more compact than a classical description referring to the 8-dimensional concentration state space, as demonstrated in Section 4.1.

When designing a system with numerous small, extensively interacting components, its global behavior cannot easily be predicted from the known local interactions. A general theory of emergence is desirable not only for analytical purposes but also for engineering such systems. If local interactions are restricted to processes that are expressible as chemical reaction rules, the theory of chemical organization helps to determine the system's repertoir of potential behavior patterns. Since only network structure is considered for the analysis, non-chemical reaction networks, e.g. social interaction networks, can be also investigated. The encouraging results presented in this paper suggest that the theory of chemical organizations is a promising candidate to contribute to a general theoretical framework to master self-organization in complex chemical-like information systems.

Acknowledgment

We acknowledge financial support by the *Federal Ministry of Education and Research* (BMBF) Grant 0312704A to *Friedrich-Schiller-University Jena* and by the German Research Foundation (DFG) Grant Di 852/4-1 to PD.

References

1. Müller-Schloer, C., von der Malsburg, C., Würtz, R.P.: Aktuelles Schlagwort: Organic Computing. Informatik Spektrum **27** (2004) 332–336
2. von der Malsburg, C.: The challenge of organic computing. Institut für Neuroinformatik, Ruhr-Universität Bochum (1999) Memorandum.
3. Würtz, R.P.: Organic computing for face and object recognition. In Dadam, P., Reichert, M., eds.: Informatik 2004. Volume 2., Gesellschaft für Informatik (2004) 636–640
4. Küppers, B.O.: Information and the Origin of Life. MIT Press, Cambridge, MA (1990)
5. Banâtre, J.P., Métayer, D.L.: A new computational model and its discipline of programming. technical report RR-0566, INRIA (1986)
6. Dittrich, P.: The bio-chemical information processing metaphor as a programming paradigm for organic computing. In: Proceeding of ARCS: Architecture of Computing Systems. LNCS, Berlin, Springer (2005) To appear.
7. Banâtre, J.P., Métayer, D.L.: The GAMMA model and its discipline of programming. Sci. Comput. Program. **15** (1990) 55–77
8. Berry, G., Boudol, G.: The chemical abstract machine. Theoretical Computer Science **96** (1992) 217–248

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

9. Păun, G.: Computing with membranes. Journal of Computer and System Sciences **61** (2000) 108–143
10. Păun, G.: Membrane Computing: An Introduction. Natural Computing Series. Springer, Berlin (2002)
11. Giavitto, J.L., Michel, O.: MGS: a rule-based programming language for complex objects and collections. In: Electronic Notes in Theoretical Computer Science. Volume 59., Elsevier (2001) 286–304 RULE 2001, Second International Workshop on Rule-Based Programming (Satellite Event of PLI 2001).
12. Banâtre, J.P., Fradet, P., Radenac, Y.: Principles of chemical programming. In Abdennadher, S., Ringeissen, C., eds.: RULE 2004 Fifth International Workshop on Rule-Based Programming, Technical Report AIB-2004-04, Department of Computer Science, RWTH Aachen, Germany (2004) 98–108
13. Banzhaf, W., Dittrich, P., Rauhe, H.: Emergent computation by catalytic reactions. Nanotechnology **7** (1996) 307–314
14. Bhalla, U.S., Iyengar, R.: Emergent properties of networks of biological signaling pathways. Science **283** (1999) 381–387
15. Tsuda, S., Aono, M., Gunji, Y.P.: Robust and emergent *physarum* logical-computing. Biosystems **73** (2004) 45–55
16. Müller-Schloer, C.: Organic computing: On the feasibility of controlled emergence. In: Proceedings of the 2nd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS2004, Stockholm, Sweden, September 8-10, 2004, ACM Press (2004) 2–5
17. Speroni di Fenizio, P., Dittrich, P.: Artificial chemistry's global dynamics. movement in the lattice of organisation. The Journal of Three Dimensional Images **16** (2002) 160–163
18. Dittrich, P., Speroni di Fenizio, P.: Chemical organization theory: towards a theory of constructive dynamical systems (2005) arXiv:q-bio.MN/0501016.
19. Kremling, A., Jahreis, K., Lengeler, J.W., Gilles, E.D.: The organization of metabolic reaction networks: A signal-oriented approach to cellular models. Metabolic Engineering **2** (2000) 190–200
20. Barabási, A.L., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. Nature Reviews Genetics **5** (2004) 101–113
21. Fontana, W., Buss, L.W.: 'The arrival of the fittest': Toward a theory of biological organization. Bulletin of Mathematical Biology **56** (1994) 1–64
22. Copasi (complex pathway simulator). (URL: http://www.copasi.org/, visited: 31 of May, 2005, last updated: 29 of April, 2005)

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

A reflexive busy beaver problem

Chris Salzberg

Department of General Systems Studies, Graduate School of Arts and Sciences
The University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan

chris@sacral.c.u-tokyo.ac.jp

Abstract. A search problem inspired by Rado's busy beaver game for Turing machines is proposed for a system of labeled directed graph structures upon which a simple read/write rule is applied locally. In the formulation employed, explicit separation of passive (information carrying) and active (information processing) structures, imposed by mandate of rules or physics in other computational formalisms, emerges instead as a result of a structure-function dynamic that is *reflexive*: objects may operate directly on their own structure. In contrast to the original busy beaver problem, in which topology of the tape is fixed and only machines are varied, in the reflexive problem it is the entire space of structures that is searched: solutions are those systems that maximize the number of steps taken in computation on their own structure. Solutions are presented for finite (hence computable) systems of between four and twelve "parts". Certain revealing results are highlighted and discussed.

1 Introduction

The busy beaver game, originally proposed by Rado in 1962 [6], is a problem in which the challenge is to construct a Turing machine on a given number of states and symbols that prints a maximal number of ones, or alternatively executes a maximal number of left/right shifts, and subsequently halts. Although the problem is simple to state and its solutions are finite and well-defined, determination of actual values are readily shown to be non-computable. There is hence no uniform procedure for finding such so-called "busy beavers", despite the fact that they can in principle be manually identified one by one using carefully-crafted heuristics and/or detailed analysis. The concreteness of this problem, when compared with equivalent yet less intuitive non-computable variations (e.g. the halting problem), has resulted in a great deal of interest in

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

the search for such machines. Despite significant obstacles, substantial efforts over recent decades have yielded important advances. For k-state Turing machines on a minimal set of two symbols, the maximum number of printed ones and left/right shifts, referred to respectively as $\Sigma(k)$ and $S(k)$, have been determined exactly up to $k = 3$ by Lin and Rado [2], for $k = 4$ by Brady [1], and lower bounds on $k = 5$ and $k = 6$ have been reported by Marxen and Buntrock [4, 5]. The functions $\Sigma(k)$ and $S(k)$ grow rapidly with increases in the number of states, lower bounds for $k = 6$ currently held by a machine that establishes $\Sigma(6) \geq 1.29 \times 10^{865}$ and $S(6) \geq 3 \times 10^{1730}$.

As might be expected, it has been found in these investigations that the dynamics of "holdout" machines, those that never halt and yet fail to evidence obvious signatures of non-halting behavior, become more diverse and complicated as k is increased. Indeed, the very question of "what" such machines are computing becomes increasingly difficult to ascertain, the busy beaver problem itself saying nothing about *how* such machines should, using a fixed number of states, achieve the requisite maximization of printed ones or left/right shifts. The complexity of these systems evidences that although Turing's formal definition of a computing machine is simple and intuitive [9], behaviour exhibited by concrete realizations thereof need not be so in general. Machlin and Stout emphasize that, "viewed as a collection of cooperating individual instructions, [the Turing machine] is an emergent structure" [3, p.87]. The emergence of computational function, uniquely framed in a simple and intuitive way in the busy beaver problem, is important as it reveals minimal formal structures that generate complex behaviour from only simple rules. Such emergence is of general interest throughout the various fields that converge in complex systems research.

Recently, a reformulation of the traditional computational machine or tape paradigm, as embodied by the Turing machine, has been proposed in a formalism referred to as *reflexive computing* [7]. Taking the form of a system of labeled directed graph structures upon which a single read/write rule is applied locally, the concept of reflexive computing effectively decouples the separation of passive (information carrying) and active (information processing) objects, imposed by mandate of rules or physics in existing models of computation and construction, from the specification of the formalism itself. Any boundaries between active and passive objects thus arise exclusively as an emergent property of nodes and their interconnections, the rules of the system themselves making no reference to higher-level object definitions. The term "reflexive" highlights the associated potential for objects in this system to operate directly on their own structure, a property that links the

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

formalism to biomolecular processes such as RNA/protein self-splicing. It has been argued that this reflexivity is a critical informational dynamic in biochemical systems, one that has been overlooked in previous computation-based studies.

Although the expressiveness and universality of the reflexive computing approach has been established via explicit construction of other computational models within the formalism, a broader search for emergent functionality of the kind evidenced by busy beaver machines has yet to be carried out. In this paper, we thus propose a class of problems, defined on the reflexive system outlined above and inspired by Rado's busy beaver game for Turing machines, aimed at initiating such a search. We broadly refer to this game as the *reflexive busy beaver problem*. In analogy to the non-computable function $S(k)$, the challenge of this problem is to find, for a given number of "parts" (as defined in the next section), a structure which executes the maximum number of steps in computation before halting. Whereas in the Turing machine problem operations are carried out on an infinitely long initially blank tape, in the reflexive version the computation occurs on the self-same (finite) structures that execute the operations. Although formalisms in both cases are computation universal and thus equivalent, this representational distinction results in significantly different problems, only the latter of which has the potential to express in a symmetric way the emergence of function from structure.

In the next section, the reflexive computing approach is outlined. For the purposes of this paper only a basic explanation of the rules is provided; a more thorough discussion is given in another publication [7]. In the Section 3, the reflexive problem is stated, and a method for searching reflexive busy beavers, derived from the tree normal form method used by Brady and others for searching the space of Turing machines, is discussed. Finally, results of the search for reflexive busy beavers are outlined in Section 4, with certain interesting discoveries highlighted in the conclusions of Section 5.

2 Formulation

The computational formalism on the basis of which the reflexive busy beaver problem is defined consists of three basic elements:

(i) A space of directed graph structures with input/output labels from a finite alphabet,
(ii) A local rule according to which digraph structures can read and be read by one another, producing as output symbols from the alphabet of (i), and

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

(iii) A self-modification mechanism by which this output is fed back into the system via overwriting of certain link output values.

For the purposes of this paper, the minimal alphabet $\mathscr{A} = \{0, 1\}$, augmented to an alphabet $\mathscr{A}^\dagger = \mathscr{A} \cup \xi$ by a special symbol ξ whose function is outlined below, are employed. The directed graph is composed of a set of nodes Q and described in terms of a transition function $\delta : \mathscr{A}^\dagger \times Q \to Q$ and output function $\varphi : \mathscr{A}^\dagger \times Q \to \mathscr{A}$ defined at each node $q \in Q$ and for each input $a \in \mathscr{A}^\dagger$ for which a transition is defined (transitions may also be undefined, see below). We will refer to a transition from a particular node using the form a/b, where in general $a \in \mathscr{A}^\dagger$ is the input and $b \in \mathscr{A}$ is the output.

To enable read/write processes to occur, the set of nodes Q is partitioned into two types of nodes referred to respectively as S-NODES ("sender" nodes) and R-NODES ("receiver" nodes). These are the "parts" from which both machine and tape structures (as well as many other non-hierarchical variants) may be constructed. They are defined in terms of the following structural constraints:

- S-NODES: Nodes with a maximum of one transition of the form ξ/a, for any symbol $a \in \mathscr{A}$, that may connect only to R-NODES.
- R-NODES: Nodes with multiple transitions of the form a'/a'', for symbols $a', a'' \in \mathscr{A}$, that may connect only to S-NODES.

The disjoint sets of S-NODES and R-NODES are referred to respectively as Q_S and Q_R. Note that, as in the case of state-transitions on a Turing machine, the number of links from R-NODES is restricted to at most one for each possible input symbol $a' \in \mathscr{A}$ (the system would otherwise be non-deterministic, see below).

Read/write processes on digraph structures occur via a local rule applied at a pair of dynamic locations on the graph; one can think of these moving locations as a kind of symmetric read/write head similar in certain respects to that of a Turing machine. This "symmetric head" serves as a contact point for interaction between two nodes of the graph, referenced by the pointers S and R that point exclusively to S-NODES and R-NODES, respectively. For brevity, in what follows transition and output functions from the node q_X pointed to by X, where $X \in \{S, R\}$, are referred to as $\delta_X(a) \equiv \delta(a, q_X)$ and $\varphi_X(a) \equiv \varphi(a, q_X)$, respectively. Update of S and R, as well as the graph itself, may then be described according to the following set of rules, applied simultaneously at each step:

1. Advance R to the node pointed to by the transition from q_S, $\delta_S(\xi)$,
2. Advance S to the node pointed to by the transition from q_R matching the output from S, $\delta_R(\varphi_S(\xi))$, and

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

3. Replace the value of the output function from q_S, $\varphi_S(\xi)$, by the value of the composite output function $\varphi_R(\varphi_S(\xi))$.

More concisely, we may express this operation as three assignments performed together at each step:

$$R := \delta_S(\xi) \tag{1}$$
$$S := \delta_R(\varphi_S(\xi)) \tag{2}$$
$$\varphi_S(\xi) := \varphi_R(\varphi_S(\xi)) \tag{3}$$

The rules (1) and (2) describe the dynamics of pointers on the graph, whereas (3) enables these dynamics to modify the graph itself via ξ-transition label overwrite. If there is no ξ-transition from q_S, or no matching transition from q_R, then the process is said to have reached a *stopping configuration*; this configuration may be thought of as analogous to the halt state of a normal Turing machine.

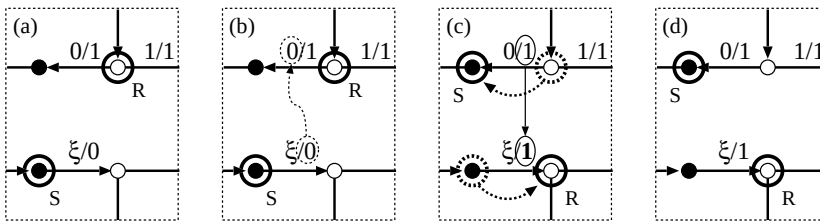


Fig. 1. One step of a reflexive read/write process on a subgraph of two S-NODES and two R-NODES. Black nodes are S-NODES, white nodes are R-NODES. Encircled nodes indicate current locations of S and R pointers. (a) S points to an S-NODE and R to a R-NODE. (b) The output $\varphi_S(\xi) = 0$ is matched to the 0-transition from q_R (since $\varphi_S(\xi)) = 0$). (c) S is advanced to $\delta_R(0)$, R is advanced to $\delta_S(\xi)$, and the value $\varphi_S(\xi)$ is overwritten by a new value $\varphi_R(\varphi_S(\xi)) = \varphi_R(0) = 1$. (d) The structures (upper and lower digraphs) originally referenced by S and R have been swapped. Read/write repeats from step (a) at the new locations.

The steps comprising one such read/write operation are illustrated in Fig. 1. The key property of the interaction defined by the rules above is that the structures referenced by the pointers S and R "flip" at each step (Fig. 1(c)). Thus although roles of the elementary S-NODE and R-NODE components of the reflexive system are clearly identified and distinguished by the rules of the system, aggregate structures built up from them are not assigned any explicit role. This general type of interaction has previously been termed *engagement* [7, 8]. Structures in an

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

engagement need neither act as information "carriers" nor as information "processors", although they may play either role in a particular topological context. This property of engagement grants the formalism the capacity to represent reflexive computational processes that machine/tape systems such as Turing machines simply cannot express in a natural way.

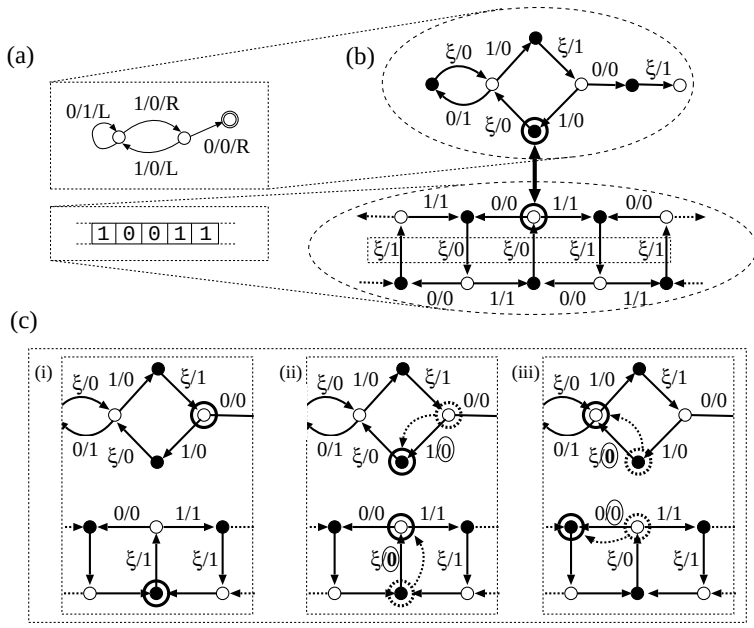


Fig. 2. Turing machine in the reflexive formulation. (a) Traditional representation of a two-state Turing machine. Rightmost node is the halt state. (b) Turing machine as an engagement system. Black nodes are S-NODES, white nodes are R-NODES. Encircled nodes mark the current positions of S/R pointers. (c) Steps from a read/write cycle of the TM. (i) The node q_S pointed to by S has a ξ-transition with output value '1' (tape value). The node q_R pointed to by R has a matching transition 1/0. (ii) '1' value is overwritten by '0', and S and R are advanced. (iii) Now structures referenced by S and R have flipped: S (on the machine structure) sends a '0', which triggers the R pointer on the tape to move to the "left".

To demonstrate more clearly the expressiveness of the reflexive formalism, and to establish computation universality, representation of a Turing machine from the elementary components introduced above is

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

constructed in Fig. 2. The method employed for this construction may be summarized in terms of the following constraints on the tape:

- S-NODES and R-NODES are paired to make writable "squares" of the TM tape. The data-containing ξ-transition from the S-NODE points to the R-NODE of the same square.
- These S-NODE/R-NODE pairs (squares) are arranged in a chain, with the 1-transition from the R-NODE pointing to the S-NODE of the "next" square (to the right), and the 0-transition pointing to the S-NODE of the "previous" square (to the left). Each S-NODE thus has two incoming links.
- Left/right branches on the tape have matching input/output values (0/0 and 1/1, respectively). This constraint corresponds to the fact that, in the traditional formulation, the tape never overwrites the machine.

In contrast to the constraints on the tape, the machine of the reformulated TM may have arbitrary topology within the limitations of S-NODE/R-NODE connectivity discussed above. Links from R-NODES on the machine structure carry the usual read/overwrite transition rules, and left/right motion commands appear as the output values of ξ-transitions from S-NODES (under a mapping from the commands $\{\mathtt{L}, \mathtt{R}\}$ to the alphabet $\{0, 1\}$). The complete system of machine and tape is shown in Fig. 2(b).

Operation of the Turing machine is shown in Fig. 2(c). Note that every read/write step breaks down into two "micro-steps", each corresponding to the pointer update operations (1) and (2) and to the overwrite (3). The first of the two micro-steps overwrites the data value on the tape; the second moves the "tape head" to the left or right, depending on the ξ-transition on the machine. Together, these steps reproduce the operation of a normal Turing machine under the constraints on the tape outlined above. The system halts when it reaches a stopping configuration as defined above.

Summarizing, the formulation outlined in this section has been shown to be reflexive, in the sense that aggregate objects (digraphs) may operate on their own structure, and also computation universal, since any Turing machine is readily representable in the format outlined above. In the next section, the reflexive busy beaver problem is presented using this formulation, and a straightforward method to search the space of such structures is described.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

3 Method

3.1 Problem statement

The reflexive busy beaver problem may be stated as: for a given number of S-NODES N_S and a given number of R-NODES N_R, what structure executes for the maximum number of steps before stopping, and how many steps does this structure spend executing the computation? If topology is subject to the constraints of a Turing machine (Fig. 2), with variation confined to the machine and all ξ-transitions on the tape initialized to zero, then this question reduces to Rado's busy beaver problem (assuming an infinitely long tape). From the flexibility of the formulation, however, it is evident that if such constraints are removed a variety of problems of a symmetric or reflexive nature may also be posed.

In this paper, we focus on the most straightforward problem, namely the finite, unconstrained version. We will assume an equal number of S-NODES and R-NODES, i.e. $N_S = N_R = N$. For a given finite number of nodes, this problem is evidently computable: after a fixed number of steps that depends on N, the system must either halt or enter a loop since the configuration space is finite (unlike the Turing machine with an infinite tape). Thus one can simply run the system up to this limit and decide whether it is a busy beaver or not.

The emphasis here, however, is not on computability, which in any case is a subject that has been amply explored in the past, but rather on *representation*, an issue that has arguably been given far less attention. It is not a priori clear, from the formalism itself, what form a reflexive busy beaver structure should take. One might wonder, for instance, whether a familiar structure such as the tape of the Turing machine (as shown in Fig. 2(b)) is itself a busy beaver, or whether different, novel topologies satisfy the requirement. It is to the determination of this form that we focus our attention for the remainder of this paper.

3.2 Tree normalization

The single most important tool employed in search procedures for solving the standard busy beaver problem is the method of *tree normalization*, an approach adopted by various authors and in various forms for enumerating and testing Turing machines in an efficient manner [1, 3, 4]. Tree normalization drastically reduces the search problem for Turing machines by combining the evaluation of each individual machine with the exploration of the space of all possible machines in a recursive way. A machine is initiated with a single state and transition, all other transitions initially marked as undefined. Simulation is carried out until

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

an undefined transition is required, at which point all possible values and target states are enumerated and recursively evaluated according to the same procedure. Target states to be considered are those that have already been explored plus one new one (if there still remain as of yet unexplored states). (For more details on tree normalization for Turing machines see e.g. [3].)

Tree normalization for the reflexive system outlined in the last section differs from the Turing machine search in that the structure *processing* and the structure *being processed* are the self-same object, and must thus be explored together. Thus whereas in the Turing machine problem, machine transitions are added as they are needed, in the reflexive problem the entire system is extended as needed. The tree is initiated with a root node representing a system with one S-NODE and one R-NODE, initially unconnected. To initiate a computational process, a ξ-transition from the S-NODE and a corresponding transition from the R-NODE are needed. As in the Turing machine problem, the output of this initial ξ-transition may be fixed since the naming of symbols is arbitrary (symmetry); we will assume this transition to be of the form $\xi/0$. In general, when a new transition is added, each possible symbol assignment, as well as each possible target node (from the set of already explored nodes plus one new node, if the maximum has not yet been reached), are explored recursively.

An example of first few branches of a search tree is shown in Fig. 3. Advantage of the tree normalization approach is exemplified in Fig. 3(a), a system that loops after a single iteration. A full search of the space of all possible graph structures would not only include this particular system, but various extensions, each of which exhibit the same behaviour. By combining the search process with the execution itself, such systems are all grouped together and tested only once; the search space is thus drastically reduced. Remark also that each non-looping branch in the search tree is an eligible busy beaver (the undefined transition that has been encountered can simply be left undefined so that the system halts). The system with one S-NODE and one R-NODE shown in Fig. 3(b), which halts after one step, is the busy beaver for $N = 1$. In general, the busy beaver is simply the deepest non-looping branch in the search tree.

4 Results

The space of reflexively computing systems were explored using the tree normal method outlined in the last section for N between 2 and 6. Results for N between 2 and 5 are shown in Fig. 4, from which certain immediate observations follow. Firstly, it is evident that the structural

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

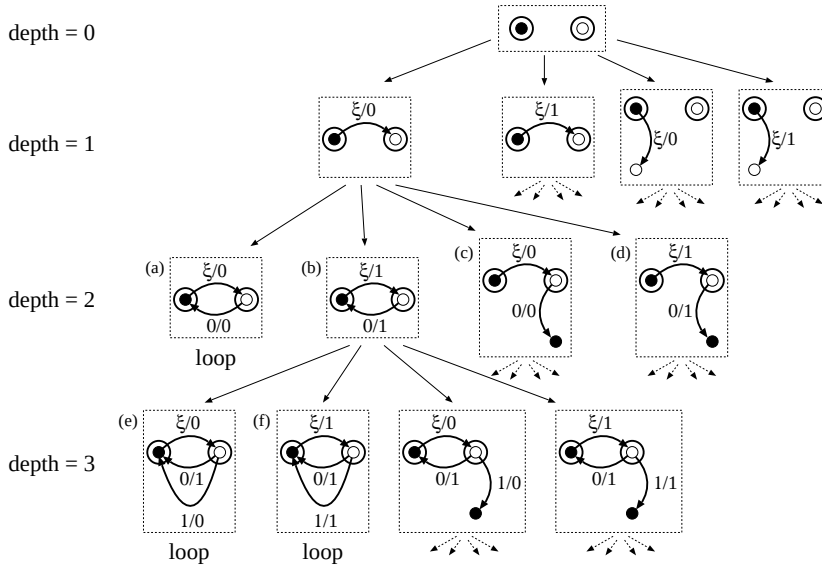


Fig. 3. Tree normalization for the reflexive busy beaver problem. Each branch corresponds to a different choice of link label and/or target node. Note that all branches have been included for clarity; in the real search routine the first transition is assumed to have output '0', thus the second and fourth branches at depth 1 are not explored. Branches at depth two are shown only for the leftmost depth one branch; depth three branches are shown only for branch (b). Leaves correspond to digraphs that have encountered a loop (as in (a), (e), and (f)). Note that graphs are shown as they are updated at each step, thus ξ-labels may be overwritten by the choice of output values on links from R-NODES (as in (b) and (d)). (b) is the busy beaver for $N = 1$, executing for one step before halting (both $N_S = 1$ and $N_R = 1$).

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

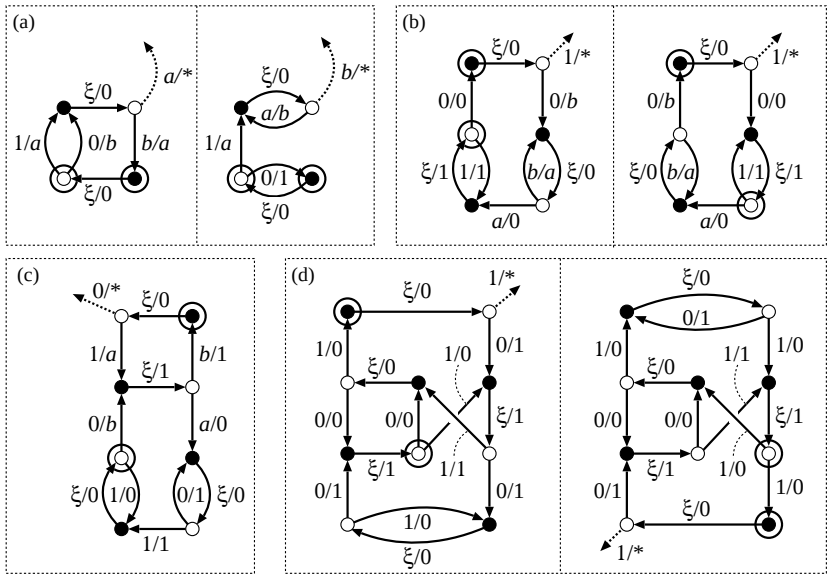


Fig. 4. Reflexive busy beavers for N between 2 and 5. Initial positions of S and R are indicated by encircled nodes. In (a), (b) and (d), a and b are any symbols in the alphabet $\mathscr{A} = \{0, 1\}$ for which $a \neq b$ (i.e. $\{a = 0, b = 1\}$ or $\{a = 1, b = 0\}$). For clarity, transition to a halting state is included for each digraph; in the actual solution this final transition is simply undefined ('*' indicates arbitrary output, since the transition is never actually used). (a) $N = 2$, runs for 6 steps before halting. (b) $N = 3$, runs for 33 steps. (c) $N = 4$, runs for 103 steps. (d) $N = 5$, runs for 304 steps.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

topology of the reflexive busy beaver machines is non-trivial in the sense that there is no obvious pattern from which the topology of a busy beaver at $N + 1$ can be predicted from that of the busy beaver at N. Note that in principle, since the problem is computable, such a result is possible and should not a priori be ruled out. The fact that a simple ring topology, the structure of the busy beaver for $N = 2$ and $N = 3$, does not outperform more complex topologies for higher values of N, however, indicates that full connectivity (any node reachable from any other node) is not the only important factor determining performance. We might thus infer that, as in the Turing machine problem, solutions to the reflexive problem have a certain emergent character that is not reducible to a simpler representation.

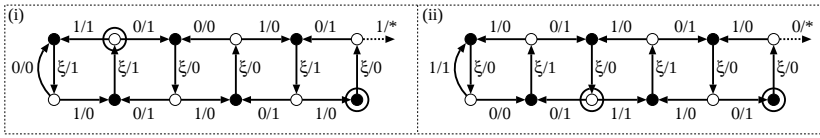


Fig. 5. Reflexive busy beavers for $N = 6$, a tape structure with topology nearly identical that of the Turing machine of Fig. 2(b) except for directionality and output values of links from certain R-NODES. Both structures execute for 832 steps before halting. Starting positions of S and R are indicated by encircled nodes.

The topology of the reflexive busy beaver for $N = 6$, shown in Fig. 5, was quite a surprise to the author of this paper, particularly in light of the lack of any obvious pattern between solutions for smaller values of N. The topology this machine, largely identical to the tape structure of the Turing machine shown in Fig. 2(b), takes the form of a linear chain of paired S-NODES and R-NODES, with the transition from the R-NODE at one end of the chain pointing to the S-NODE to which it is paired. Differences with Turing tape lie in the output values of transitions from R-NODES, which must be different in order for the structure to change the output values of ξ-transitions, and in the directionality of these links, which is not completely uniform as it is in the case of the Turing machine. Yet despite these differences, the likeness of the two structures, when considered in the space of all such possible structures, is statistically highly significant.

This significance is made clearer when the structure is compared to runners up in the $N = 6$ problem, the first two of which are shown in Fig. 6. Both second and third place solutions carry part of the same

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

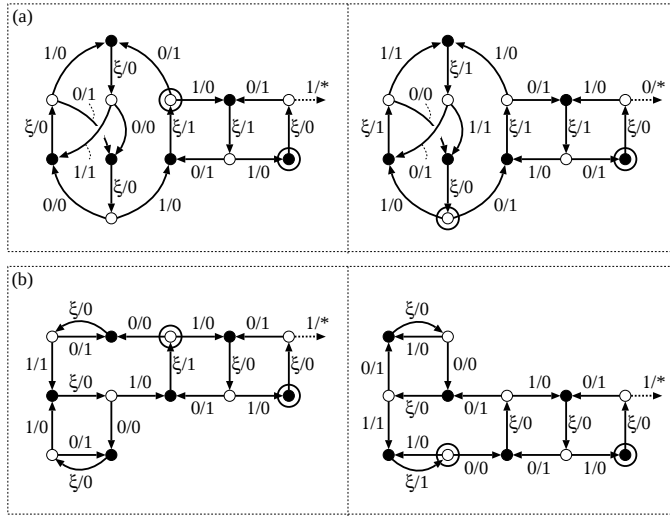


Fig. 6. Runners up in the $N = 6$ problem. Starting positions of S and R are indicated by encircled nodes. (a) Second place busy beavers that runs 831 steps before halting. (b) Third place L-shaped busy beavers that runs 792 steps before halting.

linear structure that makes up the busy beaver of Fig. 5, namely a tail structure of three S-NODES and three R-NODES that contains the "halting transition" (i.e. R-NODE with one undefined transition). The other half of the structure differs topologically in each case. The non-halting end of the second place busy beaver forms a kind of loop, whereas the third place machine assumes an L-shaped configuration. It is of interest to note that the second place busy beaver, with a score of 831 steps, comes within a single step of the busy beaver, which scores 832 steps. The loop tail structure of the second-place machine thus nearly reproduces the same performance as does the linear structure of Fig. 2. This result emphasizes the need to further explore busy beavers for larger N, to more conclusively assess whether the tape structure is generally optimal for the problem or just a particular result for $N = 6$.

Analysis of the behavior of busy beavers in execution reveals certain interesting patterns, shown in Fig. 7 for the $N = 6$ problem. Although functionality of the kind exhibited by counters and "christmas trees" in the Turing machine problem [1] have not been observed, there are clear patterns (highlighted in Fig. 7) that re-occur with only slight variations. It seems likely that the entire busy beaver execution consists of multiply

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Fig. 7. Temporal dynamics of the $N = 6$ busy beavers of Fig. 5(i) and Fig. 5(ii). Each horizontal bar is the temporal evolution of the output values of ξ-transitions, the first three over a 250 step interval and the last over a 82 step interval (for a total of 832 steps). Each black square represents a '0' value, each white square a '1' value. Ordering follows the topology of the tape, with the halting end (right end of the tape structures in Fig. 5) at the top. Some of the many repeating patterns are highlighted with boxes.

repeated behavior with just enough variation to avoid a closed loop in phase-space. An example of this type of dynamic is shown in Fig. 7(ii), in which an entire evolutionary pattern of over one hundred steps is repeated in its entirety with the exception of one changed output value on the halting end of the tape. On a shorter time scale, the highlighted pattern in Fig. 7(i), which repeats five times during execution, is each time varied by the output values of all three ξ-transitions at the halting end of the tape. Although not exactly a counter, there is evidently a resemblance. Further analysis of behavioral dynamics will be required to clarify this connection and to establish the origin of the observed patterns.

5 Conclusions

From the observations of the last section, a number of conclusions follow:

1. Reflexive busy beavers are topologically non-trivial in the sense that their structures do not follow an obvious trend as N is increased.
2. Behavior of reflexive busy beavers, as exhibited in temporal evolution of ξ-transition output values, is also non-trivial and exhibits complex recurring patterns that repeat at non-regular intervals.
3. Linear information topologies exhibit certain optimal properties for the reflexive busy beaver problem that may generalize to larger values of N.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

The results above may be taken as an encouraging sign that the system proposed in Section 2 (and further elaborated in [7]) has promising potential for engendering a deeper understanding of computational processes beyond those that fit easily into the machine-tape mould. More work has to be done to clarify the workings of these busy beaver architectures and the wider implications that they entail for the study and implementation of unconventional computing systems.

References

1. A. H. Brady. The determination of the value of rado's noncomputable function $\Sigma(k)$ for four-state turing machines. *Mathematics of Computation*, 40(162):647–665, 1983.
2. Shen Lin and Tibor Rado. Computer studies of Turing machine problems. *Journal of the ACM*, 12(2):196–212, 1965.
3. R. Machlin and Q. F. Stout. The complex behavior of simple machines. *Physica D*, 42:85–98, 1990.
4. H. Marxen and J. Buntrock. Attacking the busy beaver 5. *Bulletin of the EATCS*, 40:247–251, 1990.
5. H. Marxen and J. Buntrock. New list of record turing machines, March 2001. Personal website: `http://www.drb.insel.de/ heiner/BB/bb-6list`.
6. T. Rado. On non-computable functions. *Bell System Tech. J.*, 41:887–884, 1962.
7. Chris Salzberg. From machine and tape to structure and function: A reflexive formulation for computation and construction. *Artificial Life*, 2005. Submitted.
8. Chris Salzberg, Hiroki Sayama, and Takashi Ikegami. A tangled hierarchy of graph-constructing graphs. In *Artificial Life IX: Proceedings of the Ninth International Conference on Artificial Life*. MIT Press, 2004. In press.
9. Alan M. Turing. On computable numbers with an application to the entscheidungsproblem. *Proc. London Math. Soc. Ser. 2*, 42:230–265, 1936-1937.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Towards multi-state based computing using quantum-dot cellular automata

Iztok Lebar Bajec and Miha Mraz

University of Ljubljana, Faculty of Computer and Information Science,
1000 Ljubljana, Slovenia
iztok.bajec@fri.uni-lj.si,
WWW home page: http://lrss.fri.uni-lj.si/people/ilbajec/

Abstract. In this article we present an extended quantum-dot cellular automaton (QCA) cell. The extension is mainly focused on the enlargement of the range of possible states of a single QCA cell. In a QCA cell the electrons, owing to electrostatic repulsion, align along one of the two diagonal configurations that correspond to their maximal spatial separation. This gives the QCA cell the ability to encode two states or two logic values (0 and 1). By extending the QCA cell with four additional quantum dots we introduce the extended QCA (EQCA) cell and analyse its behaviour. Our approach is based on the semi-classical modelling approach. Using a special interpretation of electron configurations in the EQCA, the range of possible states is increased from two to three, which gives the EQCA cell the ability to encode the logic values (0, $\frac{1}{2}$ and 1). In our opinion the main benefit of this extension is the possibility for introducing "richer" processing and data storage capabilities without an increase in space requirements.

1 Introduction

With the seemingly endless miniaturisation of transistors and thus logic gates in integrated circuits the not so distant future reserves the challenge of integration in the nanometer scale. According to the well-known law, which originates from a prediction made by Gordon Moore forty years ago, the speed and complexity (i.e. the increase in the number of transistors per square inch) of integrated circuits doubles every 18 months. If this pace of miniaturisation increases or perhaps even if it does not subside, integration in the nanometer scale is to be expected in the next

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

five to ten years [9]. The nanometer scale production of logic gates actually means that their assemblage will be on the scale of molecules and atoms. With this in mind it comes as no surprise that discussions about possible approaches to nanocomputing evoke terms as 'DNA computing', 'biomolecular computing', 'quantum computing', 'quantum-dot cellular automata', etc. In this article we focus on the latter.

The quantum-dot cellular automata (QCA) became interesting in the early 1990s, when Lent et al. [3] published the first results suggesting a possible interpretation of logic values 0 and 1 based on a configuration of a pair of tunnelled electrons contained in a QCA cell. The following research on the behaviour of spatial arrangements of QCA cells resulted in the implementation of the functionally complete set of logic functions [2] (i.e. the logic functions AND, OR and NOT) [10], as well as more complex structures and tools for their design [12]. The primary research focus was, in fact it still is, dedicated to the implementation of the classical two-valued logic and the corresponding computer structures associated with it. Indeed, the main building block (the QCA cell) is still capable of representing only 1 bit of data (i.e. either the logic value 0 or 1).

In this paper, we employ the semi-classical modelling approach [5] for the study of an eight-dot QCA or extended QCA (EQCA) cell. The semi-classical approach considers the electrons, which are contained in the cell, as classical particles that can arrange in the quantum dots in such a way as to minimize the total electrostatic energy of the system. The only added non-classical property is the ability of electrons to tunnel between adjacent dots. The model is developed with a numerical simulation that includes all possible configurations of electrons in each cell, with the only exception being the configurations in which more than one electron is confined in a single dot.

With the introduction of four additional quantum dots and a special interpretation of the corresponding electron configurations, we extend the (two-state) QCA towards a hypothetical three-state EQCA. With a possible realization in the nanoscale this approach would represent a giant leap forward in computer structures. Nevertheless, in our opinion, the main benefit is the shift towards novel research directions – from two-state to multi-state cells and the corresponding processing and data storage structures. With this the primary focus would be moved from the general miniaturisation towards the research on how to obtain "richer" processing and data storage capabilities without an increase in space requirements.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

2 The semi-classical modelling approach

As already said, the semi-classical model for QCA cells is based on treating the electrons as classical particles, which can however tunnel between adjacent quantum dots [5]. The standard QCA cell has the structure presented in Fig. 1a[1] with four quantum dots separated by tunnelling barri-

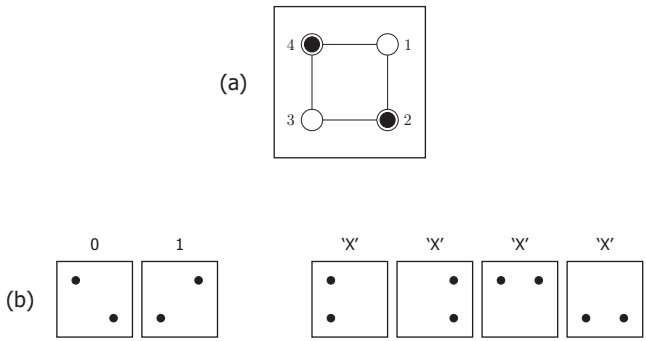


Fig. 1. Layout of a four-dot QCA cell with two electrons (a) and the corresponding possible configurations for two electrons in a cell (b).

ers. Owing to electrostatic repulsion, the electrons will tend to align along one of the two distinct configurations, which correspond to their maximum spatial separation. In the absence of external electric fields, these configurations will have exactly the same energy, while in the presence, for example, of a nearby cell with a well-defined charge distribution, one of them will be energetically favoured. Apart from these two there are also four more possible configurations; due to the very large associated electrostatic energy the configurations in which more than one electron is confined in a single dot are not considered. This means that there is a total of six configurations, as presented in Fig. 1b. Traditionally the two diagonal configurations are associated with logic values 0 and 1 (the configuration with electrons in dots 2 and 4 with 0 and the one with electrons in dot 1 and 3 with 1). The rest of all possible configurations are not associated with any logic value; they are marked as 'X' states.

In all of our simulations a positive background charge (in the case of an m-dot QCA, containing two electrons, the charge $\frac{2e}{m}$, where e is

[1] For reasons of simplicity all diagrams, except Figs. 1a and 4a, display only the electron configurations and display neither the quantum dots nor the tunnelling barriers.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

the electron charge), is assumed in each dot, in order to keep the cell overall neutral. This is necessary to avoid undesired effects associated with the monopole electric field component that would be produced by a non-neutral cell. The latter would as a result tend to push all electrons contained in the nearby cell towards the further side of the cell (i.e. toward the right side of it if the nearby cell is on the right). This means that the total electrostatic energy of an array of cells is, in the semi-classical model, expressed as the energy of a system of point charges

$$E = \sum_{i \neq j} \frac{\rho_i \rho_j}{4\pi\varepsilon_0 \varepsilon_r r_{ij}}, \tag{1}$$

where ρ_i is the charge associated with the ith dot, r_{ij} is the distance between dot i and dot j, ε_0 is the vacuum permittivity, and ε_r is the relative permittivity of the medium (we consider the case of GaAs/AlGaAs, assuming a uniform relative permittivity of 12.9). Repeating the evaluation of the total energy for all possible configurations, the ground state and the distribution of energy values can be determined. In the case of an arrangement of n m-dot cells, each containing two electrons, this means an exhaustive exploration of $\binom{m}{2}^n$ possible configurations.

3 QCA structures and QCA based switching

A *QCA structure* is a spatial arrangement of QCA cells performing a dedicated function. It can be decomposed into three segments. The first segment represents the input cells or *drivers*. In a physical sense these are usually located at the edge of the structure; their states are enforced by means of external electric field sources. The second segment is represented by the *internal* cells. With respect to their physical arrangement within the structure they transmit data from the input cells towards the third segment, the output or *target* cells. Typically the data that is input into the QCA structure is transformed with respect to the spatial arrangement of the internal cells. A more detailed description of the temporal dynamics and synchronisation mechanisms can be found in [3, 4, 6, 11]. As no direct connections can be made to internal cells and data can enter or leave the structure only on its edges, this scheme corresponds to edge-driven computation. In other words: the above means that the problem of designing a desired input/output transformation translates into the problem of finding the correct spatial arrangement of cells.

If several cells are lined up to form a wire, and a given logic value is enforced for the first cell, it will propagate along the wire in a domino fashion [10], until all cells have reached the same configuration, as presented in Fig. 2. As this means that, effectively, the logic value 0 or 1

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

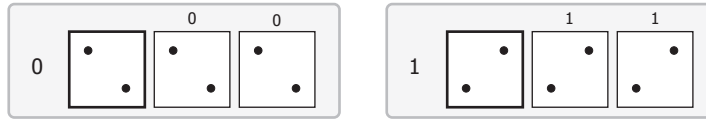


Fig. 2. The binary wire. Propagation of electron charge distribution along a line of QCA cells.

enforced (input) on one end of the wire is propagated to the other end of it (output), the wire is called a binary wire.

Nevertheless, the basics of designing QCA structures originate from the desire to find arrangements of QCA cells that implement the basic logic functions AND, OR and NOT [2]. This first succeeded to Lent et al. [3]. A QCA inverter, or NOT, is obtained by offsetting the target cell from the driver cell by 45deg as presented in Fig. 3. In this case a given

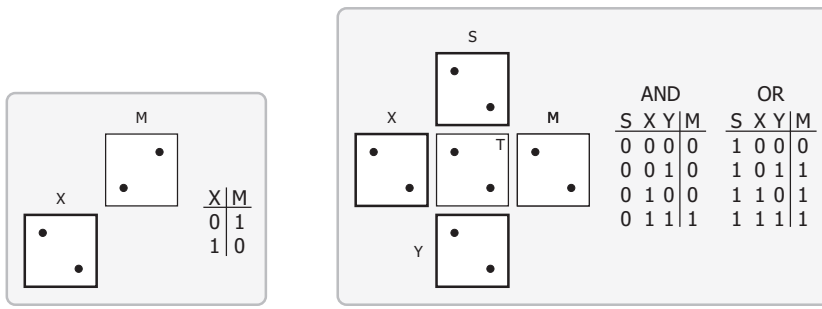


Fig. 3. The QCA inverter (left) and the QCA majority gate (right).

logic value enforced to the driver will be inverted in the target cell, the logic value 0 thus producing 1 and vice versa.

The AND and OR function, on the other hand, are constructed as an intersection of three binary wires [8]. This produces the topology presented in Fig. 3 denoted as the QCA *majority gate*. In this gate the three inputs (S, X and Y) 'vote' on the configuration of the internal cell (T), and the majority wins. The configuration of the internal cell is then propagated toward the output cell (M). One of the inputs (in our case S) can be used as a programming input to select the AND or OR function. If the programming input is the logic value 0, then the gate behaves as

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

the AND logic function, whereas if it is the logic value 1, then the gate behaves as the OR logic function.

The ability to transfer data and the functionally complete set of logic functions gives us the ability to construct any given switching structure and thus enables QCA computation.

4 Towards a multi-state QCA cell

4.1 The EQCA cell

The EQCA cell is an eight-dot QCA cell with the structure presented in Fig.4a. The dots are evenly distributed in a circular fashion, and again separated by tunnelling barriers. As in the QCA cell the electrons will, due to the electrostatic repulsion, tend to align along one of the distinct configurations that correspond to their maximal spatial separation. In the absence of external electric fields, these configurations will have exactly the same energy[2], while in the presence, for example, of a nearby cell with a well-defined charge distribution, one of them will be again energetically favoured. There are 4 distinct configurations with maximal spatial separation between electrons. Additionally there are also 24 more possible configurations. Again, the configurations in which more than one electron is confined in a single dot are not considered, due to the very large associated electrostatic energy. This means that there is a total of 28 configurations, as presented in Fig. 4b. We shall denote the diagonal configurations as configuration 'A' (i.e. electrons in dots 2 and 4) and configuration 'B' (i.e. electrons in dots 1 and 3). The vertical configuration (i.e. electrons in dots 5 and 7) and the horizontal configuration (i.e. electrons in dots 6 and 8) will, on the other hand, be denoted as configurations 'C' and 'D' respectively. The rest of all possible configurations will be marked as 'X' states.

If several cells are lined up to form a wire and either configuration 'A' or 'B' is enforced for the first cell, it will propagate along the wire, until all cells have reached the same configuration. Nevertheless if the enforced configuration is either 'C' or 'D' it will propagate along the wire in an alternating fashion, such as for example CDCDCDC... (see Fig. 5). This means that if we associate configuration 'A' with the logic value 0 and configuration 'B' with the logic value 1, in fact by retaining the associations of the classical QCA cell we retain also the binary wire

[2] Have in mind that our primary interest is extending the functionality of a cell for expanding its processing capabilities. This means that currently both the process of physical construction and the process of forcing/detecting a charge distribution are taken out of account.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

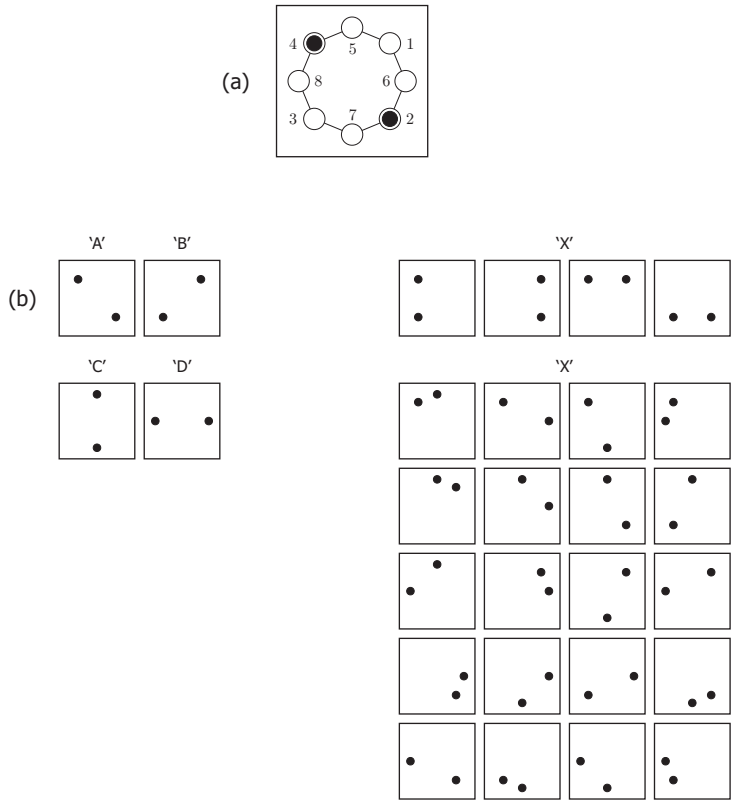


Fig. 4. Layout of an eight-dot EQCA cell with two electrons (a) and the corresponding possible configurations for two electrons in a cell (b).

capability. Additionally, if we interpret both configurations 'C' and 'D' as the logic value $\frac{1}{2}$ we end up with a three-state wire; we can effectively propagate the logic values 0, $\frac{1}{2}$ and 1. This ability opens up a whole new spectrum of possible processing capabilities. We are no longer talking about a binary universe, binary logic and binary processing but we are stepping into a three-state universe towards three-valued logic, and three-valued processing.

4.2 EQCA based switching

Multi-valued logic is a term used to describe all logics of three or more values. Three-valued logic, as a contrast to two-valued logic, which allows

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

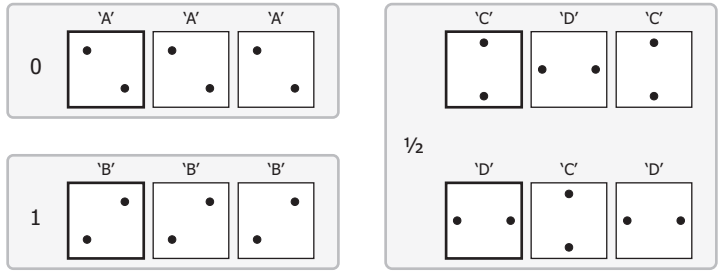


Fig. 5. The three-state wire. Propagation of electron charge distribution along a line of EQCA cells.

only logic values 0 (false) and 1 (true), allows also for the third option $\frac{1}{2}$ (possible). The truth tables for three-valued logic functions were set up by the Polish mathematician Jan Łukasiewicz [1] in the 1920s as a generalization of the corresponding truth tables found in two-valued logic (see Fig. 6). In multi-valued logic, however, the logic functions

NOT

X	M
0	1
½	½
1	0

AND

X	Y	M
0	0	0
0	½	0
0	1	0
½	0	0
½	½	½
½	1	½
1	0	0
1	½	½
1	1	1

OR

X	Y	M
0	0	0
0	½	½
0	1	1
½	0	½
½	½	½
½	1	1
1	0	1
1	½	1
1	1	1

Fig. 6. Łukasiewicz truth tables for tree-valued logic functions NOT, AND and OR.

NOT, AND and OR are typically expressed as

$$\mathrm{NOT}(x) = 1 - x, \quad \mathrm{AND}(x, y) = \min(x, y), \quad \mathrm{OR}(x, y) = \max(x, y) \tag{2}$$

Needless to say, these equations hold also for Łukasiewicz's three-valued logic. The above means that the problem of processing with EQCA cells translates into searching for EQCA structures that implement these equations.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Our approach to this problem was the most obvious one. To test if the QCA structures, which are used to implement the two-valued logic functions would give the correct results even when constructed using EQCA cells.

Surprisingly enough, the inverter, or NOT, worked perfectly, as shown in Fig. 8, giving the correct output for any given input. In the case of input configuration 'A' the output configuration was 'B' and vice versa. Then again, if the enforced input configuration was either 'C' or 'D', the output configuration did not change.

The EQCA majority gate, on the other hand, proved to be a bit more troublesome. Indeed in the EQCA case the full range of possible input configurations increases from $2^3 = 8$, in the QCA case, to $4^3 = 64$.[3] From Fig. 7 it becomes evident that, when using the full range of

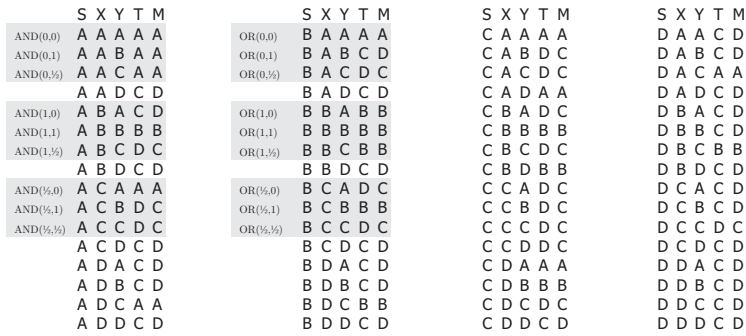


Fig. 7. The full range of possible input configurations for the EQCA majority gate.

all possible input configurations, the majority gate does not work as intended. However, if we suppose that configuration 'D' (see Fig. 4b) is merely a processing configuration (i.e. if we assume that it is not allowed for input cells, but is allowed only for internal cells) then we obtain the truth table presented in Fig. 8. This precondition is easily met. Indeed, if care is taken that, whenever a logic value is required to be transmitted over a EQCA wire, the wire is constructed from an odd number of EQCA cells, then the output cell will always assume the same configuration as the input cell, even in the case when the input configuration is 'C' or 'D'. The obtained truth table is remarkably similar to the one set up by

[3] With the given interpretation of configurations 'C' and 'D' (i.e. both as the logic value $\frac{1}{2}$), the number of distinct input combinations is in fact $3^3 = 27$.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

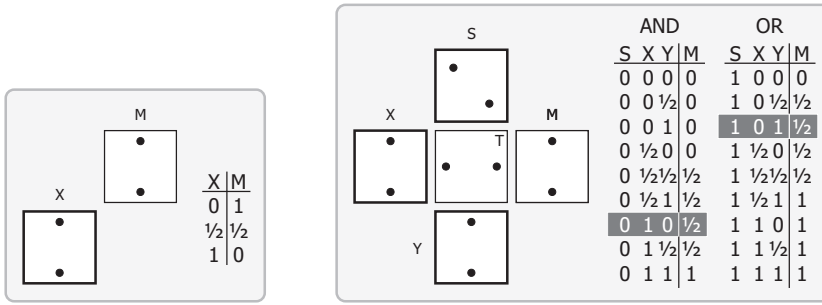


Fig. 8. The EQCA inverter (left) and the EQCA majority gate (right).

Łukasiewicz (compare Figs. 6 and 8). The only two problematic input combinations are AND$(1, 0)$ and OR$(0, 1)$ which both return $\frac{1}{2}$ instead of returning 0 and 1 respectively. Nevertheless, in both cases the output cell reaches configuration 'D', as shown in Fig. 9. Since configuration 'D'

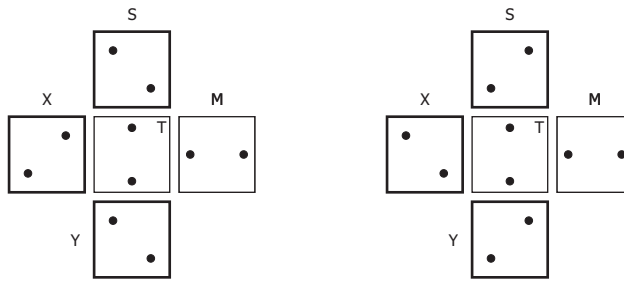


Fig. 9. The two erroneous input/output configurations for the EQCA majority gate when used for implementing the three-valued AND (left) and OR (right) logic functions.

is reached only from these two input combinations, a possible solution for obtaining the required truth table would be a special additional EQCA structure. Its sole objective would be to translate configuration 'D' into the correct output value and it would be employed only in the case when the output configuration is 'D'. Our current research is focused upon obtaining such a structure.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

5 Conclusion

In this article we present an extended quantum dot cellular automaton cell. The extension focuses on the enlargement of the range of possible states of a single cell. More specifically we present an eight-dot QCA, which together with a specific interpretation of electron configurations defines a three-state EQCA cell that is capable of transmitting logic values 0, $\frac{1}{2}$ and 1 over a three-state wire. Additionally we highlight the faced challenges and present some of the possible solutions for its application to the design of future processing structures based on multi-valued logic. The primary goal of this research is in the promotion of the idea to switch the focus from pure miniaturisation towards research for a better use of the existing level of miniaturisation. The era of analytical top-down approaches is nearing its end and we are entering the era of bottom-up synthesis approaches. Indeed it would be unsound to assume that in the years to come both the ethical as well as the technical aspects of miniaturisation will keep on flourishing with the same pace [7].

References

1. Borkowski L. (ed.): Jan Lukasiewicz: Selected Works. North-Holland Publishing Company, Amsterdam (1970)
2. Kohavi Z.: Switching and finite automata theory. McGraw-Hill Inc., New York (1978)
3. Lent C.S., Tougaw P.D., Porod W., Bernstein G.H.: Quantum cellular automata. Nanotechnology **4** (1993) 49–57
4. Lent C.S., Tougaw P.D.: Lines of interacting quantum-dot cells: A binary wire. J. Appl. Phys **74** (1993) 6227–6233
5. Macucci M., Iannaccone G., Francaviglia S., Pellegrini B.: Semiclassical simulation of quantum cellular automaton cells. Int. J. Circ. Theor. Appl. **29** (2001) 37–47
6. Niemier M.T., Kogge P.M.: Origins and motivations for design rules in QCA, in: Nano, Quantum and Molecular Computing, Ed. S.K. Shukla and R.I. Bahar, Kluwer Academic Publish. (2004) 267–293
7. Phoenix C., Drexler E.: Safe exponential manufacturing. Nanotechnology **15** (2004) 869–872
8. Snider G.L, Orlov A.O., Amlani I., Bernstein G.H.: Quantum-dot cellular automata: Line and majority logic gate. Jpn. J. Appl. Phys **38** (1999) 7227–7229
9. Steane S., Rieffel E.: Beyond bits: The future of quantum information processing. IEEE Computer **1** (2000) 38–45
10. Tougaw P.D., Lent C.S.: Logical Devices Implemented Using Quantum Cellular Automata. J. Appl. Phys. **75** (1994) 1818–1825
11. Tougaw P.D., Lent C.S.: Dynamic behaviour of quantum cellular automata. J. Appl. Phys. **80** (1996) 4722–4736

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

12. Walus K., Dysart T.J., Jullien G.A., Budiman R.A.: QCADesigner: A rapid design and simulation tool for quantum dots cellular automata. IEEE Transactions on Nanotechnology **3** (2004) 26–31

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Rule migration: Exploring a design framework for modelling emergence in CA-like systems

Heather R. Turner and Susan Stepney

Department of Computer Science, University of York,
Heslington, York, YO10 5DD, UK.
turner|susan@cs.york.ac.uk

Abstract. We propose a framework for *engineering* emergent behaviour that allows system specification and design at the level of the emergence. We discuss how rule migration can be used to translate this high-level multi-layer design (*mobile process* model) into a simple *cellular automaton* model with only local rules, from which the behaviour can be described as emergent. In this paper, we use a case study in 2D to illustrate the process involved in deriving a migration rule for a simple random walk. To conclude, we discuss how the mobile process architecture may be extended to model systems with multiple levels of emergence.

1 Introduction

The design of emergent systems is a contentious topic among many researchers. If the behaviour is *designed* into a system, then by some definitions of emergence, it is not really emergent. Those of us who hope to be able to *engineer* emergence at some point in the future tend to employ definitions that focus on the properties of the emergent system, rather than the means by which it was designed.

We are interested in developing an architecture for modelling emergence in cellular-automaton-like systems. In [6] we discuss, from a refinement perspective, the difficulties faced when trying to engineer an emergent system. By definition, there is no simple translation between macro-level desired emergent behaviour, and the micro-level rules. We conclude that it should be possible, by choosing an appropriate representation architecture for the micro-level, to find ways of bridging the gap.

In [7], we consider a case study of blood platelets moving and aggregating in a 1D system. We introduce an architecture in which mobile

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

processes (MPs) are used to *tag* emergent properties on a cellular grid, thus identifying them at an upper layer. Some of the operational logic can be transferred to the MPs, providing downward causation to the lower layer grid, thus making the lower layer simpler than one implemented as a pure cellular automaton (CA). We also introduce the idea of *rule migration*, as a means of translating the simpler MP models into pure CA models.

With such a framework, we can design a complicated model using the layers of the MP architecture. Emergent properties such as motion, and aggregation, can be described and observed at the higher level. The rules from the MP model are then translated by *rule migration* into low level CA rules, as illustrated in figure 1. These CA rules are unlikely to encode explicitly the higher level properties, and as such the properties can be considered *emergent properties* of the CA model.

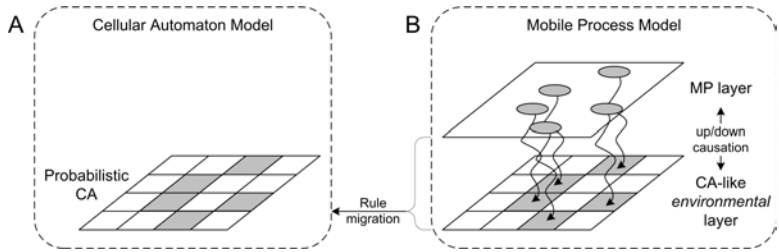


Fig. 1. An illustration of the relationship between (A) the CA model, and (B) the MP model.

In this paper, we extend the work on rule migration into 2D. We use a similar platelet case study system. It is important to ensure that our work generalizes into higher dimensions, particularly because most potential real world applications exist in 2D and 3D environments.

We constrain our model to ensure that objects within the model do not need to know their absolute position in their environment. Relative location is important, such as proximity to other objects, but no knowledge of absolute location, a global property, should be necessary. The layers of the MP architecture are particularly suited to the enforcement of such constraints. By placing restrictions on the communication protocols, we can ensure that no global position information is passed upwards from the lower layer. Intermediate layers can be used to abstract away many types of detail, and help us to build a structure where the infor-

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

mation at each layer is restricted to only that which is necessary, and local at its own scale.

For the purposes of this study, we take an abstract view of blood platelets as objects that can move through the blood vessel and form clots to repair damaged tissue. In the following section, we present our designs for two different platelet models of a simple case study system behaviour. One simulation is implemented as a low level CA, and the other is designed in terms of the high level processes. We explore the relationship between the two models, adjusting each of them, as necessary, until they are functionally equivalent. Once the two models are equivalent, we have a direct mapping (*migration rule*) between the implementations at the two levels, and thus a means of translating high level designs involving this behaviour into a low-level implementation. In Future Work, we discuss our plans to explore the mapping between other *general* behaviours implemented at the different levels. We hope eventually to build a dictionary of these migration rules for converting high level designs into low level architectures.

2 Case Study

There are two major considerations in the design of the platelet model: how to simulate movement through the channel, and how to generate platelet aggregation. Each of these can be reduced to simpler scenarios by capturing a more general case. As an example, in the more sophisticated model, platelets are expected to travel primarily in the direction of flow in the channel, by following a biased random walk. In the simpler case, we allow motion in any of the four compass directions on the square grid, each chosen with equal probability. The situation is slightly complicated by the presence of multiple platelets; these act as obstacles to movement, and can compete for environmental real-estate.

Breaking the model into smaller, simpler sub-properties has obvious positive implications when it comes to the design of a system. We can explore each one independently, choosing a level of abstraction that simplifies the design process. As the model develops, we can add more detail, until all aspects of each sub-property are adequately captured. The rules for the completed sub-properties are combined to provide the rules for the model as a whole.

In the subsections that follow, we discuss two different, but equivalent, implementations of the simple random walk by a platelet on a 2D square grid.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

2.1 Cellular Automaton Model

The probabilistic nature of the random walk cannot easily be implemented in a simple, deterministic CA. The CA model we present (see figure 1A) adheres to many of the properties that we consider to be the most important, defining characteristics of a CA. Space is modelled by a regular, square grid of sites that update in synchrony at discrete time intervals. Each site has an associated state, which in this case is composed of two variables that contribute, in total, nine possible unique states. The update rules, which define how the sites change state based on their own state, and those of their neighbours, are defined for the von Neumann neighbourhood of size 5. The automaton is homogeneous, that is, all the sites in the grid use the same rules. In contrast to the usual definition of a CA, the rules are probabilistic. They are also quite complicated. It is unnecessary to produce a look-up table for the 60,000 or so neighbourhood configurations; in the following sections, the rule is described as a compressed algorithm that could be used to generate the lookup table.

Returning to the design itself, we have a 2D grid with *occupied* and *empty* CA sites. The occupied sites, representing the platelets, have the ability to choose their direction of motion, based on the availability of empty sites in their neighbourhood. CA sites can update only their own state, and so have no direct means of indicating to the chosen destination site that it has been selected. We use a series of *passes* in the CA rule to allow the sites to indicate their preference, verify selection, and perform the necessary state updates.

We now describe in detail how this strategy can be employed with our platelet model. Figure 2 shows a possible initial configuration for the CA grid. Each site's state has two parts, a **type**, and a **direction**. The **type** can be *black*, *grey*, or *white*, to indicate new platelet, old platelet, or empty site. The **direction** can take the values N, S, E, W, and corresponds to neighbouring sites in each of the compass positions. The value of **direction** is relevant only when **type** is indicating the presence of a platelet. The different snapshots, or passes, identify different synchronisation phases of the CA rule.

At the beginning of each time step, a platelet is given the opportunity to try to move. It first evaluates the states of the neighbouring sites, to establish a *current* view of its surroundings. The platelet may move only to a cell that is empty during this evaluation. Having determined the availability of empty sites in the four neighbouring directions, the platelet selects one, with uniform probability, if any are available. A platelet indicates its chosen destination by setting its direction state appropriately. This is the situation shown in figure 2 as Pass 1.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

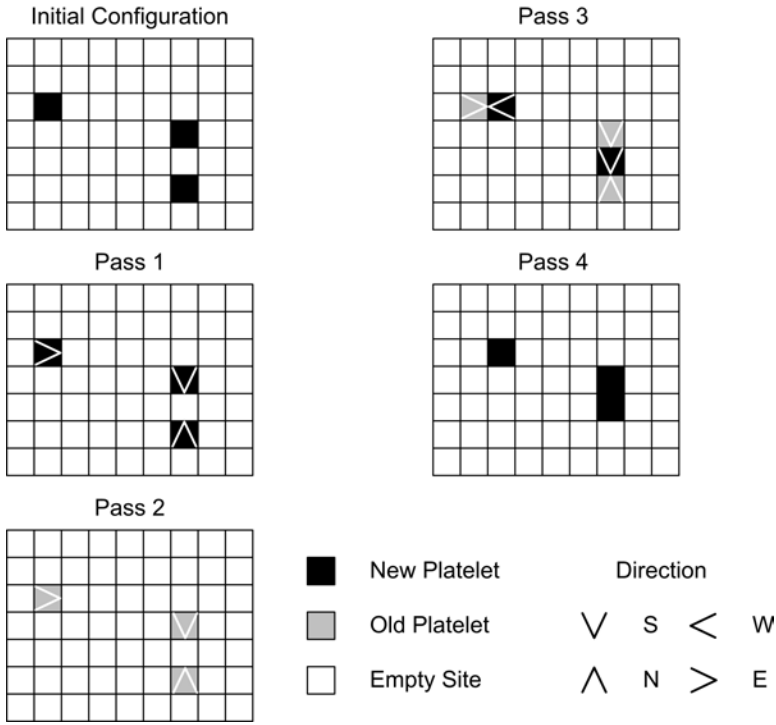


Fig. 2. This diagram shows one time step in the evolution of the CA. Four different passes update the states of the grid sites, comprising **type**, indicated by the colour of the grid square, and **direction**, identified by the arrows. By applying the simple local rules, we simulate observable random motion.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

The platelet cannot just *move* in the chosen direction, as that would involve updating the state of its neighbouring site. In Pass 2, the platelets change their **type** to *grey* to indicate that they are ready to move.

In Pass 3, the *white* (empty) sites evaluate their neighbourhood, and determine whether any platelets want to move into them. They evaluate both the **type**, and **direction** of their neighbouring sites, counting any prospective new occupants. They select with uniform probability any neighbouring platelet wanting to move, and set their **type** to *black*, and **direction** to point at the selected platelet. Otherwise the site is unchanged.

In Pass 4 the *grey* sites are updated to indicate whether the platelet successfully moved, or whether it failed at this particular time step. Each grey site checks that it has one *black* neighbour with its **direction** pointing back at the *grey* site. If so, the *grey* site changes its **type** to *white*, thus completing the move. If the neighbouring states indicate that the move is not accepted, the *grey* site changes back to *black*; it can attempt to move again in the next time step. The direction is no longer relevant.

The four pass structure of this algorithm is used for clarity; it is equivalent to one single rule that includes an extra state that identifies which pass the system is in.

The "movement" of the platelet is an emergent property of this CA. Nothing has actually moved: sites merely change state. The rules have to be carefully designed to correspond to movement and conservation of platelets.

2.2 Mobile Process Model

The mobile process (MP) model for this platelet example (see figure 1B) is a 2D extension of the 1D model presented in [7]. A cellular array forms the lower layer, and certain sites are "tagged" with mobile processes, residing at an upper layer, that represent the presence of a platelet. Some of the logic resides in the lower cellular layer, which we call the environmental layer after [2], and some is captured by the MPs, making the movement easier to model.

An important aspect of the two layer architecture is its ability to abstract away detail. For example, at the mobile upper layer, processes have no notion of *absolute* location. Communication channels act like sensors into the environment, providing a means by which the MP can sense and interact with its neighbourhood, without actually knowing *where* it is. The environment contains state information indicating whether connections to upper layer processes exist at each site. The topology of this architecture is illustrated in figure 3. In this model, each MP is

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

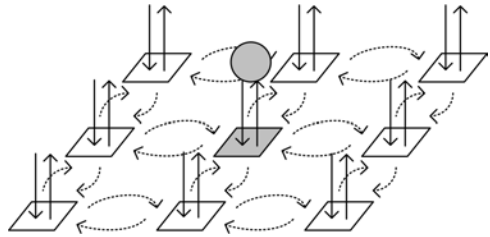


Fig. 3. A schematic illustration of the MP architecture used in the platelet model. The square grid represents the lower environmental layer, and arrows indicate the communication channels. The grey circle is a MP at the upper layer, and the grey square is a reflection of the state information that records its presence at the lower layer.

connected to precisely one site at the environmental layer, so there is a direct equivalence between sites representing platelets and processes representing platelets. At the environmental layer, there is an absolute frame of reference in the coordinates of the grid sites. Each site is directly connected to its four immediate neighbours by two communication channels, and it has two further channels by which it can connect to any MPs at the mobile layer. Upper layer MPs send information requests to their respective cellular site. The communication protocol permits the transmission of state information associated with neighbouring sites. In addition, the channels facilitate movement of the MP when it transfers from one site to a neighbouring site. For an example of how this movement might be implemented, see Appendix A.

In the platelet algorithm, platelets are represented by MPs at the upper layer. At each iteration of the model, the platelet tries to move. It submits a request to its environmental site to provide information regarding its neighbourhood. For this particular implementation, we choose to hide all location and direction information from the MP, but allow it to implement the probabilistic selection. To do this, the server counts the available neighbouring sites and reports back to the MP. The MP then chooses one with uniform probability, and sends the 'move' request to its environmental site. By a mechanism similar to that described in Appendix A, the MP attempts to move to the chosen neighbouring environmental site. If the movement is unsuccessful, because, for example, another process is chosen to move to the site, the MP's site re-evaluates the neighbourhood, and permits the MP to choose another direction. At this point, information about platelet locations in the two layers is no longer equivalent, the environmental layer must update. Sites that have

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

lost connections become *empty*, whilst those that find their channels newly connected become *occupied*.

3 Rule Migration

To achieve our goal of providing a general mechanism for translation between the more simply definable MP architecture, and a pure CA-like system, we first consider some specific examples. As we have shown in the previous sections, we have independently designed an MP model, and a CA model, for our simple platelet example.

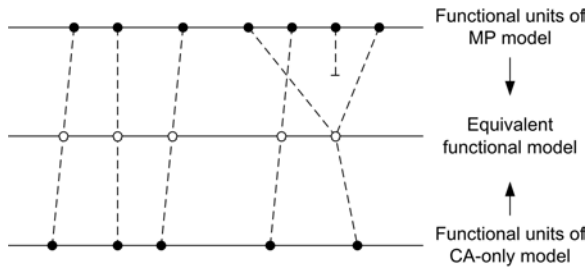


Fig. 4. This diagram is a schematic for the alignment process.

The next step is to make the two models functionally equivalent, that is produce the same result when given the same initial configuration. This process can be seen as an alignment operation, as we illustrate schematically in figure 4. We work from the top (MP model) downwards, and from the bottom (CA model) upwards, and make necessary modifications and adjustments until the two models can 'meet in the middle', and be equivalent.

In the MP model, the platelet appears to have more identity, and with this, it seems logical, and simpler, to offer it more freedom and choice. When two platelets attempt to move to the same location, one is prevented from moving. In the MP model, it is reasonable to allow it to "have a another go", and try another direction, at least until it runs out of options. The simplicity is achieved because the MP and its environmental site can freely exchange information without requiring the complicated state changes that would be required in the CA model. In the MP architecture, when the CA site fails to fulfil a move request, it can send a message to the MP, and ask it to pick a different direction. The MP can reply with a new choice, and another move is attempted.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

The pure CA model does not have this richer functionality, as it does not fit so naturally with the synchronous update model. In the process of aligning these two models, it is simpler to scale down the functionality of this more complicated MP model. The *second-chance* for the MP is removed from the model, and the alignment is complete. Figure 5 is the result of this process, and shows how the equivalent functionality is achieved by different means in each case. The details are discussed below. To further clarify what we consider to be equivalence, the configuration of the environmental layer of the MP model, and of the CA model, should be equivalent at the beginning and end of each iteration. In the intermediate processing, the CA model makes use of additional state information, and the MP model sends messages via its communication channels.

In figure 5, we show the finished product of the alignment process. The two models are functionally equivalent, and we show where the different sub-units are implemented in each case. In the diagram, numbered arrows link the parts of the algorithm that perform equivalent operations. These are described in more detail below:

1. Platelet environment is evaluated.
2. Platelet selects a direction in which to move.
3. In the CA model the platelets that are trying to move change state so that they can be distinguished in later passes from platelets that have successfully moved.
4. Platelet ascertains whether it was successful in its request to move.
5. In the CA, we evaluate all empty cells in the grid to determine whether any platelets want to move into them. Many empty cells will not be near platelets and so should not change state. In the MP model there is no equivalent as computation is carried out only in the locality of the MPs.
6. Much of the CA algorithm is covered in one step of the MP model. This step is the *actual* movement, and the communication channels are rearranged at this point. In the CA, we have the simple case where there is no competition for the destination site and the movement effectively occurs (although the source cell must be updated later, as identified by the second arrow marked with a 6). In the more complicated case where there is a conflict, the destination site can detect that more than one platelet is trying to move into it, and it has to choose one at random (this is not dissimilar from the random element to the process scheduling that determines which MP moves first). The successful platelet *moves* at this point.
7. The platelet did not get to its destination before it was occupied by another. The platelet does not move. In the CA case, the platelet location from the first pass is reactivated.
8. The environmental layer of the MP model is updated to reflect the new locations of the MPs.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

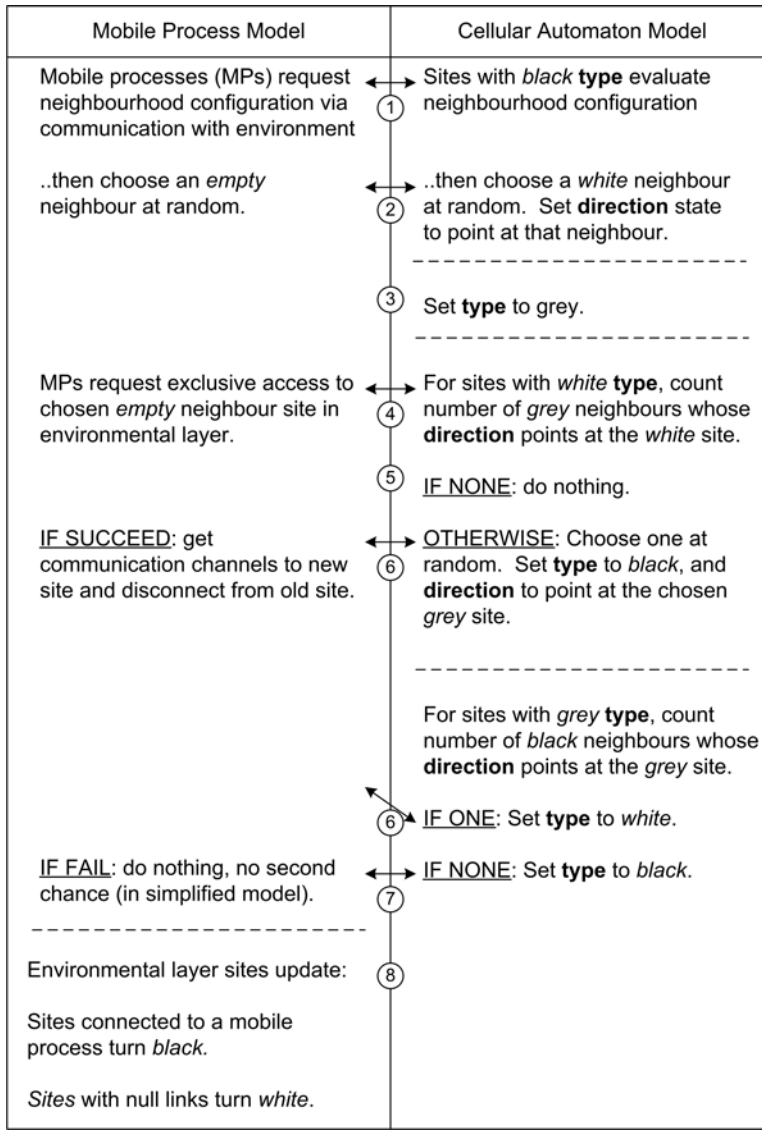


Fig. 5. The finished product of the alignment process. Numbered arrows show where the program sub-units are equivalent.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Although this correspondence needs to be formalised, we hypothesise that the functional equivalences described above can be generalised. By considering more examples, and through further experimentation, we propose to find patterns of operation in the MP model that correspond consistently to other patterns of behaviour in an equivalent CA model. If we can build a dictionary of such correspondences (migration rules), we will have accomplished a major step towards an architecture for exploring emergence in CA-like systems.

4 Future Work

If we can describe two levels in an emergence hierarchy in terms of our MP architecture, then by applying our migration rules, we can build a low level CA that displays emergent behaviour. The MP model can include moving, aggregating objects, and these can be programmed explicitly. After migration, the rules for the CA model will be unlikely to have such transparency. It will be difficult to discern the system behaviour from the CA rules, and the resulting behaviour will be difficult to explain in terms of the low level CA rules; it will require description in a higher-level language. We would argue that although there is much controversy surrounding the definition of *emergence*, the behaviour just described would be classified as emergent.

Extending this idea, we plan to develop migration rules between MP layers. Ultimately we want to be able to model systems with multiple levels of emergence. By describing these systems in terms of MPs at the highest level, we can migrate the rules to produce an MP model at a lower level. The rules from this model can then be migrated downwards, until eventually the rules are migrated to a CA model. The MP to CA rule migration is the special case, while the more general MP to MP migration rules will permit us to model, at least in principle, an arbitrary hierarchy of emergence.

Our next step is to explore more examples, with wider breadth. We will generalise the results here, and construct the migration rules. In addition, we will explore the combination of simple sub-properties such as movement and aggregation, into a more complete model.

We expect it to be possible during the design process to either combine the MP sub-properties and then migrate the rules to a CA-only model, or to perform the rule migration on the individual sub-properties, before combining them. We hope that properly designed rules will have a useful degree of commutativity (see figure 6).

Once we have satisfactory results with migration between the MP model and the CA model, we will consider the addition of upper layers. In

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

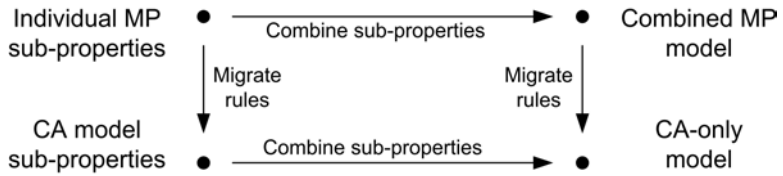


Fig. 6. Diagram illustrating the hypothesised commutative nature of the migration and combination procedures.

our case study system, this is likely to be the movement of aggregations of platelets. We will design rules at the level of the aggregation, and at the level of individual platelet MPs, and then formulate the appropriate migration rules.

5 Related Work

The hierarchical structure of emergence has been identified and discussed by many researchers. A recent literature survey by Fernando [3] provides a useful introduction to both the theoretical and experimental research in this area. He focusses on *hierarchical complexity*, and a study of the modelling constraints that are necessary to evolve this structure.

There appears to be a limited amount of material available concerning higher levels of emergence. Mayer and Rasmussen [5] also use a variation on a CA in their model. They call the architecture a *Lattice Molecular Automaton*. The state and dynamics of this model are highly complicated, and the rules are derived from the laws of physics. The authors claim that this detail is necessary to achieve the higher order structures. We intend to explore this similar work in more detail to discover the means by which the system is designed. It would be interesting to discover if we can derive a similar model from the top down, using our MP architecture, and rule migration.

Another researcher who uses a multi-layer architecture is Capcarrere [2]. His model, Phuon, is constructed from two layers, the active cellular layer, and a passive environmental layer. These layers seem to correspond quite closely to the layers in our MP architecture, and we have chosen to name our lower layer in the same way for consistency. That model particularly considers the simulation of growth and development, but in addition incorporates motion at the cellular layer. We will explore growth and development in the context of our own architecture and the migration rules.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

One of the initial problems we encountered in our case study was how to model movement using just CA rules. Other authors have considered this problem, although the explicit detail of how the movement is achieved in terms of the local rules is not always clear. Goel and Thompson [4] use movable finite automata in a specific biological case study. Wolfram [8] considers motion in a CA at the macro-level in terms of its ability to model fluid dynamics. That paper concentrates heavily on the mathematics of the macro-level behaviour, and the underlying rules for movement are not entirely clear.

6 Acknowledgements

This research is funded by EPSRC and BAE Systems as part of a CASE Studentship.

References

1. Barnes, F.R.M., Welch, P.H.: Communicating mobile processes. In East, I.R., Duce, D., Green, M., Martin, J.M.R., Welch, P.H. (eds), Communicating Process Architectures. Amsterdam, IOS Press, 2004.
2. Capcarrere, M.S.: An evolving ontogenetic cellular system for better adaptiveness. BioSystems **76** 177–189, 2004.
3. Fernando, C.: On the Evolution of Hierarchical Levels of Organisation. Unpublished DPhil literature survey, University of Sussex. Available from: http://www.informatics.susx.ac.uk/users/ctf20/end_of_year_report_sc.ps, accessed May 2005.
4. Goel, N.S., Thompson, R.L.: Movable Finite Automata. In Langton, C. (ed), Artificial Life, SFI Studies in the Sciences of Complexity. Addison-Wesley, 1988.
5. Mayer, B., Rasmussen, S.: Self-Reproduction of Dynamical Hierarchies in Chemical Systems. In Adami, C., Belew, R.K., Kitano, H., Taylor, C. (eds), Artificial Life VI: proceedings of the Sixth International Conference on Artificial Life. MIT Press, 1998.
6. Polack, F.A.C., Stepney, S.: Emergent properties do not refine. In REFINE 2005, BCS-FACS Refinement Workshop, ENTCS. April 2005.
7. Polack, F.A.C., Stepney, S., Turner, H.R., Welch, P.H., Barnes, F.R.M.: An Architecture for modelling emergence in CA-like systems. In ECAL 2005, LNCS. Springer, 2005.
8. Wolfram, S.: Cellular Automaton Fluids: Basic Theory. In Wolfram, S., Cellular Automata and Complexity: collected papers. Addison-Wesley, 1996.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

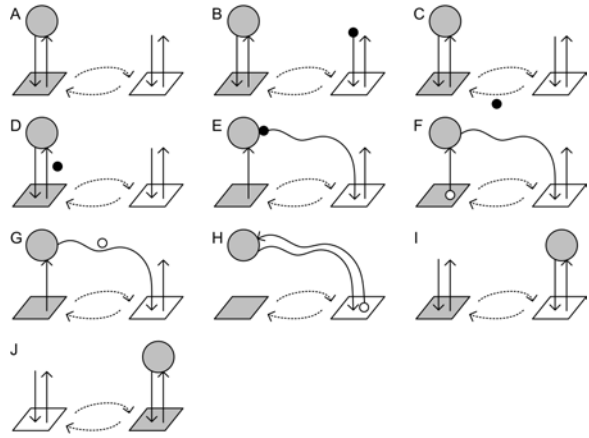


Fig. 7. An illustration of the series of steps (A-J) that implant the movement of a MP within the proposed architecture. Each step shows the two neighbouring environmental layer sites (squares) and the single MP (circle). Communication channels are represented by arrows, and small circles represent the mobile *channel ends*.

A Implementing Movement with Mobile Processes

Figure 7 illustrates how we might implement the movement of mobile processes around the grid whilst hiding explicit location information from the MP itself. Each channel pair consists of a *send* channel and a *receive* channel (A). When an MP decides to move, it evaluates the local neighbourhood to determine the available space; this proceeds via an interaction with the lower environmental layer. The MP itself might know only that it has, for example, three options, and can choose one at random, without actually knowing which direction it corresponds to. The movement is all handled by the lower layer. When an appropriate destination site is chosen, the MP requests the new communication channels, and relinquishes the old ones. This procedure can be achieved by the MP's environmental site requesting that its appropriate neighbouring site should send the *source* end of its *receive* channel to the MP (B,C). When the MP receives this, it connects to the channel, thus releasing one connection to the site below (D-F). This new channel is then used to send the *source* end of the MP's *receive* channel to the new site (G). The old site is now fully disconnected, while the MP now has a two-way communication channel to the new server site (H). At this point, the state information at the environmental layer no longer accurately represents

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

the configuration of MPs at the upper layer (I). The inter-layer connections for each site are examined. Where channels are not connected, the states are set to *empty*, but where active communication channels exist, the state is set to *occupied* (J).

The multi-layer model is designed to be implemented in occam-π, as this allows for an efficient parallel implementation, and incorporates mobile processes, and mobile channels [1]. As a result of the mobile processes executing in parallel, there is a possibility that multiple MPs will choose the same destination site. Only one MP succeeds in its movement, while other attempts to move to that site fail.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Evolution In Materio: Evolving logic gates in liquid crystal

Simon Harding and Julian F. Miller

Department of Electronics
University of York
York, UK
{slh,jfm}@evolutioninmaterio.com
http://www.evolutioninmaterio.com

Abstract. Intrinsic evolution has been shown to be capable of exploiting the physical properties of materials to solve problems, however most researchers have chosen to limit themselves to using standard electronic components. However, these components are human designed and intentionally have predictable responses, so they may not be the most suitable physical devices to use when using a stochastic search technique. Indeed allowing computer controlled evolution (CCE) to manipulate novel physical media might allow much greater scope for the discovery of unconventional solutions. Last year the authors demonstrated, for the first time, that CCE could manipulate liquid crystal to perform computational tasks (i.e frequency discrimination, robot control). In this paper, we demonstrate that it is also possible to evolve logic gates in liquid crystal.

A Introduction

Miller and Downing argued that evolution in hardware ought to be able to benefit from access to a richer physical environment [14]. This is because materials with rich, complex, non-linear dynamics would have many subtle physical internal interactions that could be utilized in potentially unexpected ways when external signals are applied (i.e. voltages). At present most evolvable hardware research is focused on conventional component based evolution. It is unlikely that conventional logic components that have been built to be as digital as possible would provide enough physical richness to allow evolution much scope for exploitation of the embedded physics. Evolving the configuration of actual physical variables applied to materials (we call this *in materio* evolution) may allow us to develop new computational systems that are based on

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

exploiting hitherto unknown physical properties of a complex system. Thompson in his work with FPGA circuits found that an evolutionary algorithm used some subtle physical properties of the system to solve problems [19]. To this day, it is not fully understood what properties of the FPGA were used. Of course, this lack of knowledge of how the system works prevents humans from designing systems that exploit these subtle and complex physical characteristics. However it does not prevent exploitation through artificial evolution. Thus there is a real chance that evolution in materio may allow the discovery of new physical effects that can be harnessed for computation.

In [7] Harding and Miller gave the first demonstration that an unknown complex physics could be exploited for computation by showing that liquid crystal could be used. They were able to evolve simple transistor like behaviour and the discrimination of pairs of dissimilar frequencies surprisingly rapidly (the latter task being inspired by Thompson's groundbreaking work) [6]. Very recently they have shown that a real time robot controller can be evolved in liquid crystal [8]. In this paper we present work demonstrating that it is even possible to evolve a liquid crystal display to perform digital logic operators.

B The Field Programmable Matter Array

In [14] a conceptual device was described that the authors called a Field Programmable Matter Array(FPMA). The idea behind the FPMA is that applied voltages may induce physical changes within a substance, and that these changes may interact in unexpected ways that may be exploitable under evolution.

They suggested a number of materials that could potentially be used as the evolvable substrate in the FPMA. They all share several characteristics : the material should be configurable by an applied voltage/current, the material should affect an incident signal (e.g. optical and electronic) and should be able to be reset back to its original state. Examples of these include electroactive polymers, voltage controlled colloids, bacterial consortia, liquid crystal, and nanoparticle suspensions. In our previous work we have demonstrated that liquid crystal is indeed a suitable material to form the basis of the FPMA.

B.1 Liquid Crystal

Liquid crystal (LC) is commonly defined as a substance that can exist in a mesomorphic state [5][11]. Mesomorphic states have a degree of molecular order that lies between that of a solid crystal (long-range

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

positional and orientational) and a liquid, gas or amorphous solid (no long-range order). It is possible to control the orientation of liquid crystal molecules using electric fields. Normally the molecules in a liquid crystal align themselves along a common director, however this orientation is essentially random. By applying an electric field it is possible to change the angle of this director and force the molecules to rotate into a desired orientation. Changing the orientation of the liquid crystal changes its behaviour. The most well known of these effects is the change in its optical properties. Rotating the molecules changes the refractive index of the liquid, it is this effect that is used in liquid crystal displays(LCDs).

Changing the orientation of the molecules also alters the electrical properties of the liquid crystal. Figure 1 shows the equivalent electrical circuit for liquid crystal between two electrodes when an AC voltage is applied. The distributed resistors, R, are produced by the electrodes. The capacitance, C, and the conductance, G, are produced by the liquid crystal layer[15]. Changing the orientation alters these constants, and in this work we aim to exploit these electrical properties (and possibly other properties) by using applied fields to alter the molecular configuration.

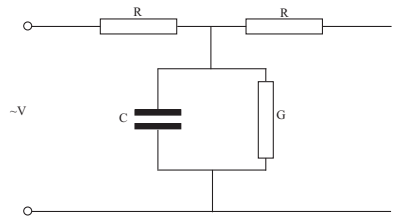


Fig. 1. Equivalent circuit for LC

C An Evolvable Motherboard with a FPMA

C.1 Previous Evolvable Motherboards

An evolvable motherboard(EM) was a term first coined by Layzell, [12] it is a circuit that can be used to investigate intrinsic evolution. The EM is a reconfigurable circuit that rewires a circuit under computer control. Previous EMs have been used to evolve circuits containing electronic components[12][3] - however they can also be used to evolve in materio by replacing the standard components with a candidate material.

An EM is connected to a PC that is used to control the evolutionary processes. The PC also has digital and analog I/O, and can be used

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

to provide test signals and record the response of the material under evolution, as shown in figure 4.

C.2 The Liquid Crystal EM

In the experiments presented here, a standard liquid crystal display with twisted nematic liquid crystals was used as the medium for evolution.

The display is a monochromatic matrix LCD with a resolution for 180 by 120 pixels. The displays are made up of several layers, as shown in figure 2. The liquid crystal layer(c) is sandwiched between the two sheets which are coated in electric connections(b,d). These layers are then positioned between two polarising filters, one in a horizontal orientation(a) the other vertically(e).

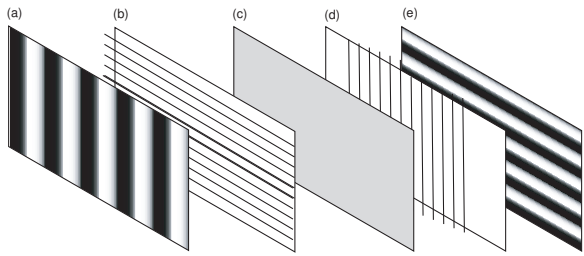


Fig. 2. Layers in a LCD

We have assumed that the electrodes are indium tin oxide. Typically such a display would be connected to a driver circuit. The driver circuit has a configuration bus on which commands can be given for writing text or individually addressing pixels so that images can be displayed. The driver circuit has a large number of outputs that connect to the wires on the matrix display. When displaying an image appropriate connections are held high, at a fixed voltage - the outputs are typically either fully on or fully off.

Such a driver circuit is unsuitable for our task of intrinsic evolution. We need to be able to apply both control signals and incident signals to the display, and also record the response from a particular connector. Evolution should be allowed to determine the correct voltages to apply, and may choose to apply several different values. The evolutionary algorithm should also be able to select suitable positions to apply and record values. A standard driver circuit would be unable to do this satisfactorily.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

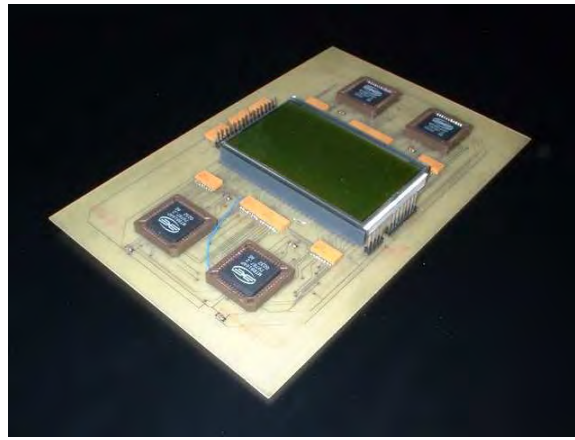


Fig. 3. The LCEM

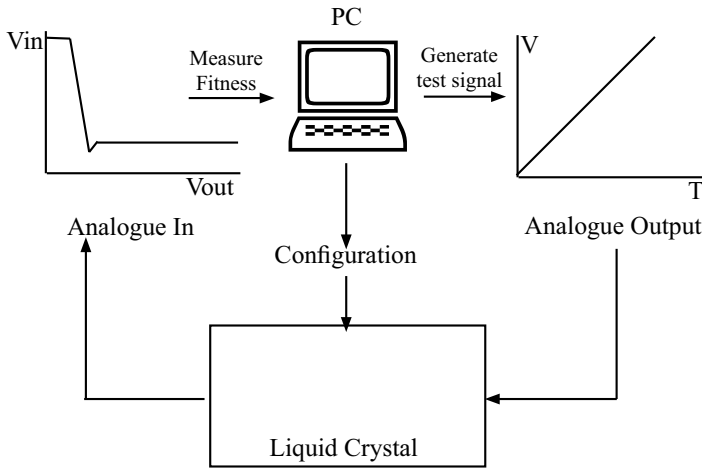


Fig. 4. Equipment configuration

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

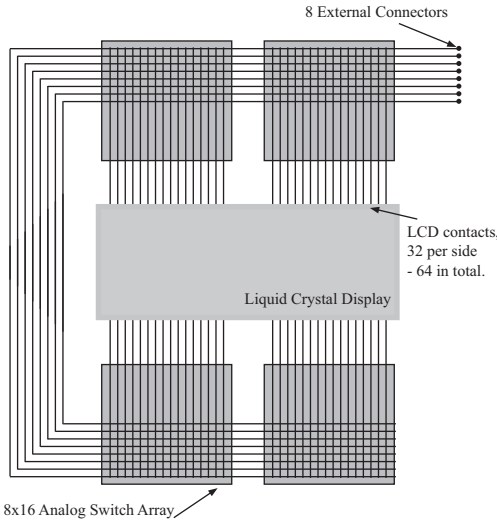


Fig. 5. Schematic of LCEM

Hence a variation of the evolvable motherboard was developed in order to meet these requirements.

The Liquid Crystal Evolvable Motherboard (LCEM) is circuit that uses four cross-switch matrix devices to dynamically configure circuits connecting to the liquid crystal. The switches are used to wire the 64 connections on the LCD to one of 8 external connections. The external connections are: input voltages, grounding, signals and connections to measurement devices. Each of the external connectors can be wired to any of the connections to the LCD.

The external connections of the LCEM are connected to the Evolvatron's analogue inputs and outputs. Connections can be assigned for the input signals, measurement, and for fixed voltages (plus a ground connection). The value of the fixed voltages is determined by a genetic algorithm[9], but is constant throughout each evaluation.

In these experiments the liquid crystal glass sandwich was removed from the display controller it was originally mounted on, and placed on the LCEM. The display has a large number of connections (in excess of 200), however because of PCB manufacturing constraints we are limited in the size of connection we can make, and hence the number of connections. The LCD is therefore roughly positioned over the pads on the

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

PCB, with many of the PCB pads touching more than 1 of the connectors on the LCD. This means that we are applying configuration voltages to several areas of LC at the same time.

Unfortunately neither the internal structure nor the electrical characteristics of the LCD are known. This raises the possibility that a configuration may be applied that would damage the device. The wires inside the LCD are made of an extremely thin material that could easily be burnt out if too much current flows through them. To guard against this, each connection to the LCD is made through a 4.7Kohm resistor in order to provide protection against short circuits and to help limit the current in the LCD. The current supplied to the LCD is limited to 100mA. The software controlling the evolution is also responsible for avoiding configurations that may endanger the device (such as short circuits).

It is important to note that other than the control circuitry for the switch arrays there are no other active components on the motherboard - only analog switches, smoothing capacitors, resistors and the LCD are present.

D Previous Systems Evolved In Liquid Crystal

D.1 Tone Discriminator

A tone discriminator is a device which when presented with one of two signals input signals returns a different response for the each signal. In [7] a device was evolved in liquid crystal to perform this task. In [19], on which this experiment is loosely based, the task is to differentiate two different frequency square waves, giving a low output for one and a high output for the other. We have evolved a system in liquid crystal that was also able to perform this task. It was found that it was easier to evolve solution in liquid crystal than it was to evolve a circuit in an FPGA as reported by Thompson. An example of the evolved response in shown in figure 6.

D.2 Real-time Robot Controller

This year we have recently demonstrated that it is also possible to evolve a robot controller in liquid crystal. A controller was evolved that allowed a simulated robot to move around an enclosed environment with obstacles without colliding with the walls, as shown in figure 7. The task is significantly harder than that of our previous work with liquid crystal (if only because the number of inputs and outputs to the display device has been doubled). Yet we found that it was relatively easy (in evolutionary

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

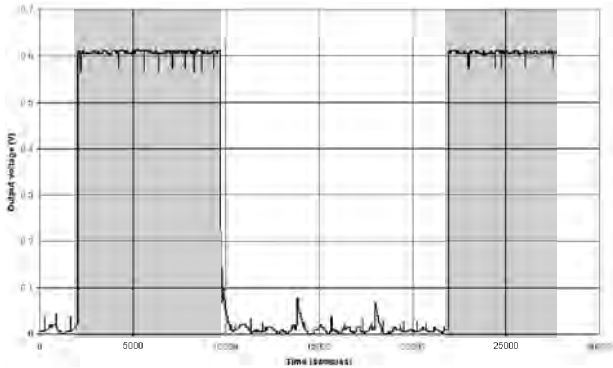


Fig. 6. Tone discriminator response. Dark areas indicate 5kHz input, light 100Hz

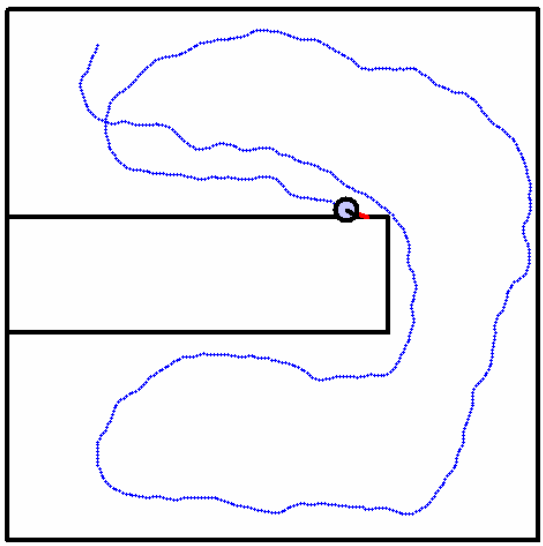


Fig. 7. Path of an robot controlled using liquid crystal

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

terms) to evolve a sophisticated robot controller.
The quality of results when compared to previous work is also high. The environment is more complex than that of comparable work, and unlike much work on evolving GP robot controllers or neural network controllers, we solve a real-time control task. The results also indicated an evolutionary computational effort that is comparable to other examples of evolved controller (with simpler tasks).

E Logic Gates

Our previous experiments reported on evolution in liquid crystal have evolved solutions that are analogue in nature. In this paper the potential of evolving digital circuits is explored. A digital circuit is one in which there are are only two possible states. In circuits these states are called HIGH and LOW, and represent the states of TRUE and FALSE in Boolean logic[10]. HIGH and LOW relate to the voltage present. In [6] non-linear functions were evolved - that had an interesting non-linear, step behaviour and suggested that liquid crystal could be capable of being evolved to act in a digital fashion.

E.1 Gates : The Building Blocks of Digital Circuits

Each of the logical operators in Boolean logic has an equivalent digital circuit - known as a gate. The main types of gates are inverters (NOT), AND, inverted AND (NAND), OR, inverted OR (NOR) and exclusive OR (XOR).

By building combinations of NOR or NAND gates it is possible to construct any logical function. This is an important observation, as it relates to Turing completeness. If a system is logically sufficient then it is possible to use it as a component of a Turing complete system, and hence use it to perform any arbitrary computation.

Hence, if it can be shown that combinatorial circuits can be evolved in liquid crystal, it can be shown that liquid crystal can be used to perform an arbitrary computation. This is in addition to any non-Von Neumann computation that it capable of performing.

But why, when transistor based technology is so good at implementing logic gates, would we want to try to produce them in a substance like liquid crystal? In [17], the authors conclude that:

> What is the purpose of demonstrating the feasibility of chemical-based computational devices? In comparison with electronic-based

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

> computations, chemical computations occur on time scales that are orders of magnitude longer and require space scales that are orders of magnitude larger. There can be only one reason for constructing computational devices based on excitable chemical media: to determine what such chemical systems can and cannot do. In so doing, we explore the possibilities for information processing by all excitable media, chemical and biological, with the chance of garnering insights into the workings of information processing in living systems.

de Silva argues that building molecular level gates out of chemical systems will have many practical uses, especially when constructing small devices - for example operating within a cell. He states that non-silicon will be "be natural for such logic devices to find ready application in physiology, medicine and biotechnology as sensors and diagnostic systems." [4]. Silicon devices are power and space hungry when compared to molecular level digital circuits, and de Silva suggests that "wet chemistry" may provide a superior alternative in some situations.

E.2 Gates in unusual materials

Logic gates in computers are made of transistors, however it is relatively easy to construct gates from other materials. There are many ways to use chemical systems to perform logical operations. For example, Steinbock [17] uses the propagation of chemicals through designed channels to produce several gates. There are several example of producing logic gates using DNA [16]. These DNA gates are designed, which is unfortunate given the previously demonstrated evolutionary properties of DNA. Moving up some levels Knight and Sussman described a technique that exploits the mechanisms in a biological cell[18]. In [2], Adamatzky describes gates made from chemical reactions, in particular utilising the properties of reaction diffusion wave fronts passing through specially designed channels.

In [4], de Silva describes the current technologies for creating molecular scale logic gates. He compares the many techniques including "chemically-controlled fluorescent and transmittance- based switches concerned with small molecules, DNA oligo nucleotides with fluorescence readout, oligonucleotide reactions with DNA-based catalysts, chemically-gated photochromics, reversibly denaturable proteins, molecular machines with optical and electronic signals, two-photon fluorophores and multichromophoric transient optical switches".

There are also many example of building logic gates out of mechanical systems, many systems are described in Merkle [13]. There is even a set of logic gates made of Lego [1].

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

F Evolving Gates in Liquid Crystal

A number of experiments were performed, each one to evolve a different gate. The target functions were NOT, OR, AND, NAND, NOR and XOR. Each experiment was repeated 10 times, and allowed to run for 100 epochs.

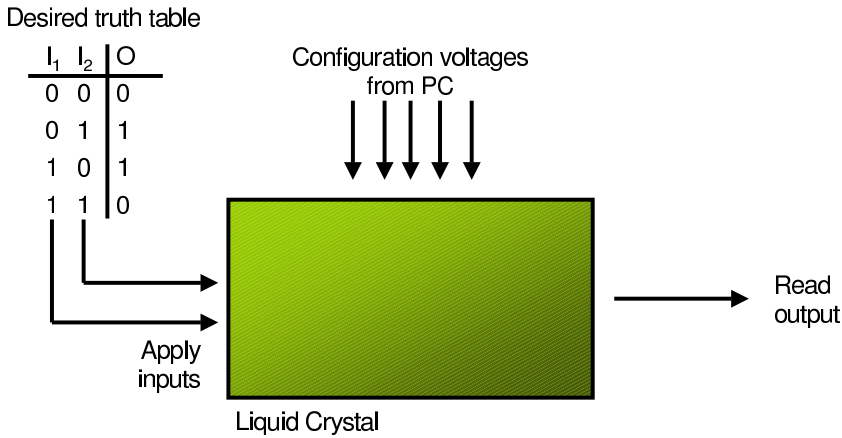


Fig. 8. Diagram showing evolution of logic gates

F.1 The Genotype and Genetic Operators

The genetic representation for each individual is made of two parts. The first part specifies the connectivity; the second part determines the configuration voltages applied to the the LCD.

Each of the 64 connectors on the LCD can be connected to one of the eight external connectors or left to float, figure 5. Each of the connectors is represented by a number from 0 to 7 and no connection is represented by 8. Hence the genotype for connectivity is a string of 64 integers in the range 0 to 8.

The remainder of the genotype specifies the voltages, in the range -10V to +10V that are supplied to the liquid crystal. In this configuration, four of the eight connections to the LCD are already predefined (ground, two incident signals and an output), leaving four voltages to be specified by evolution for configuration.

To clarify this further, the evolutionary algorithm determines three possible voltages and where they may be applied to any of the 64 connectors on the LCD. The algorithm also determines to which of the

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

connectors on the LCD the incident signals will be applied, the connector used to read the output signals from and which connectors should be grounded.

In all the following experiments, a population of 40 individuals was used. The mutation rate was set to 5 mutations per individual. A mutation is defined as randomly taking an element in one part of the genotype and setting it to a randomly selected new value. Elitism was used, with 5 individuals selected from the population going through to the next generation. Selection was performed using tournament selection based on a sample of 5 individuals. Evolutionary runs were limited to 200 generations.

F.2 The Fitness Function

Fitness evaluation was performed by applying each row in the truth table three times, and measuring the response of the liquid crystal. The order in which the input patterns was presented was randomized. The fitness was then calculated as the number of correct output responses given. A perfectly correct solution would receive a maximum fitness score of twelve. A solution where the output was always either HIGH or LOW or where the output was random would receive a fitness of 6. The responses for each evaluation are recorded as a truth table so that it is possible to determine if the evolved solution produced an intermittently working gate. For example, a fitness score of 10 would indicate that for at least one of the applications of the truth table all responses were correct.

We presented the digital inputs to the liquid crystal as static voltages. With +1V representing TRUE and 0V representing FALSE. The threshold voltage on the output was set at 0.1V, i.e. a response of more than +0.1V was accepted as a TRUE output and any other value was FALSE. After presenting an input combination to the liquid crystal, a pause of 20ms was given before reading the response. The sampling of the response was averaged over a short period. This was to ensure that any physical changes in the liquid crystal had time to take effect.

G Results

The results show that it is possible to evolve many gate types. The effort required to evolve each gate type is shown in table 1. The results show that most of the gate types are extremely fast to evolve, with most of the gates sometimes being produced in the initial population. However, it often takes much longer for the gates to be found. It is possible that the liquid crystal has some sort of memory of previous configurations

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Gate	Min. Evals.	Max. Evals.	Avg. Eval.s	Std. dev
AND	2	1788	910	527.45
OR	1	1779	769	576.01
XOR	44	1255	649	605.50
NOT	3	1750	536	749.58
NAND	13	1763	880	623.18
NOR	1	1788	907	526.97

Table 1. Results showing the minimum, maximum and average number of evaluations required to find a good solution, i.e. where fitness $\geq$ 10 out of 12

which may solve the problem, and as the system may not be fully reset it would allow rapid evolution in some circumstances. As expected, XOR proves hardest to evolve. This is probably due to the fitness landscape of the XOR problem - there is very little information in the four test cases for it to learn from. In addition, it is not linearly separable.

The results shown in table 2 show the detailed results from each run. It can be seen that it is rare for a fully functioning logic gate to be produced, and that the system normally only produces intermittently functioning devices. If a configuration has a fitness of 10 or more, then it can be assumed that in at least one of the applications of the truth table all the outputs were correct.

H Conclusions

We have shown that it is possible to evolve logic gates using liquid crystal, however the evolved gates are intermittent in their behaviour. In this instance, it is unclear if producing logic gates in liquid crystal is viable. However, previous work has shown that it is a suitable medium for less precise devices - such as robot controllers. It is our belief that the benefits of utilising unusual materials for computation will become more evident when performing non-Von Neumann computation. Here the inherent non-linear and chaotic behaviour may allow for more interesting computation. We have also demonstrated that computer controlled evolution is a suitable programming methodology for computation in materials. It is currently unclear if there is any mechanism by which such systems could be directly programmed by hand.

We have only explored a tiny fraction of the potential of computational matter. We have demonstrated that it is possible to program material systems, in this case using evolution, to provide computation in both classical and non-classical senses. At present our experimental

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Gate	Run	Max. Fitness	Evaluation	Gate	Run	Max. Fitness	Evaluation
AND	0	10	48	NOT	0	10	3
	1	11	77		1	10	48
	2	10	6		2	9	23
	3	10	18		3	9	82
	4	10	18		4	11	101
	5	11	484		5	10	43
	6	10	16		6	11	120
OR	0	11	17	NAND	0	10	53
	1	10	184		1	10	18
	2	11	452		2	10	382
	3	11	24		3	10	13
	4	10	10		4	10	35
	5	11	6		5	10	14
	6	12	1500		6	10	149
XOR	0	9	1221	NOR	0	11	13
	1	9	31		1	11	349
	2	9	329		2	11	1405
	3	10	1255		3	11	12
	4	9	953		4	11	1342
	5	9	734		5	11	77
	6	9	319		6	11	957

Table 2. Results from gate evolution. Maximum fitness is 12. A good solution should get at least a fitness of 10 to be considered as intermittently functional.

set up is rather crude. In the future we will construct field programmable matter arrays that will allow us to control the material in more sophisticated ways. We believe that this will enable the development of computational devices that offer advantages over conventional devices. It may be possible to build small devices operating at a molecular level, that require low power and may be more resistant to environmental factors.

References

1. *LEGO Logic Gates, http://goldfish.ikaruga.co.uk/logic.html.*
2. Andrew Adamatzky. *Computing in Nonlinear Media and Automata Collectives.* Institute of Physics Publishing, 2001.
3. Jamie Crooks. Evolvable analogue hardware. Meng project report, The University Of York, 2002.
4. A. P. de Silva and N. D. McClenaghan. Molecular scale logic gates. 10:574–586, 2004.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

5. D. Demus, J Goodby, G W Gray, H W Spiess, and V Vill, editors. *Handbook of Liquid Crystals*, volume 1,2A,2B,3. July 1998.
6. Simon Harding and Julian F. Miller. Evolution in materio: A tone discriminator in liquid crystal. In *In Proceedings of the Congress on Evolutionary Computation 2004 (CEC'2004)*, volume 2, pages 1800–1807, 2004.
7. Simon Harding and Julian F. Miller. Evolution in materio: Initial experiments with liquid crystal. In *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware (EH'04)*, pages 298–305, 2004.
8. Simon Harding and Julian F. Miller. Evolution in materio : A real time robot controller in liquid crystal. In *Proceedings of NASA/DoD Conference on Evolvable Hardware*, 2005.
9. John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.
10. Horowitz and Hill. *Art of Electronics, Second Edition*. Cambridge University Press, 1989.
11. I. C. Khoo. *Liquid Crystals: physical properties and nonlinear optical phenomena*. Wiley, 1995.
12. Paul Layzell. A new research tool for intrinsic hardware evolution. *Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware, LNCS*, 1478:47–56, 1998.
13. Ralph C. Merkle. Two types of mechanical reversible logic. *Nanotechnology*, 4:114–131, 1993.
14. J. F. Miller and K. Downing. Evolution in materio: Looking beyond the silicon box. In *Proceedings of NASA/DoD Evolvable Hardware Workshop*, pages 167–176, 2002.
15. A. F. Naumov, M. Yu. Loktev, I. R. Guralnik, and G. Vdovin. Liquid-crystal adaptive lenses with modal control. In *Optics Letters*, volume 23, pages 992–994, 1998.
16. Akimitsu Okamoto, Kazuo Tanaka, and Isao Saito. Dna logic gates. 126(30):9458–9463, Aug 2004.
17. O. Steinbock, P. Kettunen, and K. Showalter. Chemical wave logic gates. 100:18970–18975, 1996.
18. Jr. Thomas F. Knight and Gerald Jay Sussman. Cellular gate technology. 1997.
19. Adrian Thompson. An evolved circuit, intrinsic in silicon, entwined with physics. In *ICES*, pages 390–405, 1996.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Contibutors

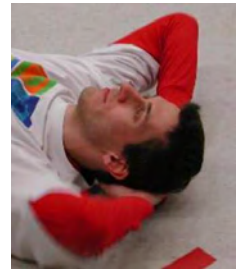
Andy Adamatzky is a Professor at the Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol, UK. His secret interests include novel and emerging computing paradigms and architectures, tuning his car, cellular-automata theory and applications, and collective artificial intelligence. He authored "Identification of Cellular Automata" (Taylor & Francis, 1994), "Computing in Nonlinear Media and Automata Collectives" (IoP Publishing, 2001), "Dynamics of Crowd-Minds" (World Scientific, 2005), and co-authored "Reaction-Diffusion Computers" (Elsevier, 2005).

Yoshihito Amemiya received the B.E., M.E., and Dr. Engineering degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 1970, 1972, and 1975. He joined NTT Musashino Laboratories in 1975, where he worked on the development of high-speed silicon LSIs. Since 1993, he has been a Professor with the Department of Electrical Engineering, Hokkaido University, Sapporo. His research interests are in the fields of signal processing circuits based on nonlinear analog computation, logic systems consisting of single-electron circuits, and information-processing devices making use of quantum nanostructures.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Tetsuya Asai is an Associate Professor in the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. His research interests concentrate around developing nature-inspired integrated circuits and their computational applications. Current topics that he is involved with include; intelligent image sensors that incorporate biological visual systems or cellular automata in the chip, neuro-chips that implement neural elements (neurons, synapses, etc.) and neuro-morphic networks, and reaction-diffusion chips that imitate vital chemical systems. He mostly prefers to play piano and guitar with his favourite Scotch.



Florian Centler, Bio Systems Analysis Group, Jena Centre for Bioinformatics (JCB), Institute of Computer Science Friedrich-Schiller-University Jena, Germany

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

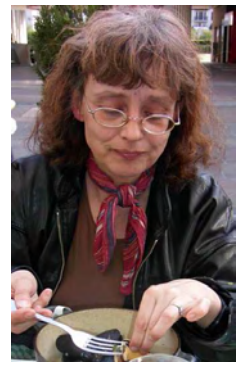
ISBN-10: 095511702X ISBN-13: 978-0955117022

Peter Dittrich is junior group leader of the Bio Systems Analysis Group at the Friedrich Schiller University Jena, Department of Mathematics and Computer Science; and member of the Jena Centre for Bioinformatics. He is interested in modeling, simulation, and theory of complex biological and social systems, and especially in their reflexive information processing capabilities.

Jerzy Gorecki, PhD in Physics (1984), DSc (Habilitation) in Physics (1994) is a Professor of Chemistry, Institute of Physical Chemistry of the Polish Academy of Sciences, and Professor of Mathematics and Natural Sciences, Cardinal Stefan Wyszynski University, Warsaw, Poland. His current research interests include microscopic simulations of chemical systems far from equilibrium and information processing in excitable chemical systems.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022



Joanna Gorecka, PhD in Physics (1993), is a Research Associate, Institute of Physics, Polish Academy of Science, Poland. Her current research interests include Information processing with nonlinear, spatially distributed systems.

Simon Harding joined the department of electronics at York after studying artificial intelligence and computer science at Birmingham University. He is currently studying towards a PhD in Electronics. His research is primarily focused on using computer controlled evolution to exploit the physical properties of materials, in particular liquid crystal and radiation damaged silicon. Recent work with Julian Miller has shown that liquid crystal is suitable for intrinsic evolution. The research included new designs of evolvable hardware and development of system for intrinsic material evolution.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Zuwairie Ibrahim received his B.Eng (Mechatronics, 2000) and M.Eng. (Image Processing, 2002) from Universiti Teknologi Malaysia. Since 2002, he is engaged with Department of Mechatronics and Robotics, Universiti Teknologi Malaysia as a lecturer. He is currently pursuing his PhD at the Institute of Applied DNA Computing, Meiji University, Kanagawa, Japan. He is a student member of IEEE, Int. Computational Intelligence Soc. (ICIS), Int. Soc. for Nanoscale Science, Computation and Engineering (ISNSCE), and Int. Signal Processing Soc. (ISPC). He is a recipient of IEEE Computational Intelligence Soc. (CIS) Walter J. Karplus Summer Research Grant 2004. His research interests include signal and image processing, automated visual inspection, evolutionary and unconventional computing such as molecular or DNA computing.

Marzuki Khalid is currently a Professor in Intelligent Control and Director at the Centre of Artificial Intelligence and Robotics (CAIRO) of University of Technology Malaysia. His current research interest is in the field of artificial intelligence with applications to control. He is a member of Editorial Board of the Int. J. of Engineering Applications of AI (Elsevier Science) and an Associate Editor of the J. of Systems and Control Engineering (Institute of Mechanical Engineers, United Kingdom). He is currently the IEEE Student Activities Chair for Region 10 (Asia Pacific) and also the Founding Member of the Asian Control Professors Association.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Iztok Lebar Bajec is an Assistant at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. He has MSc (2002) and BSc (2000) degrees in Computer and Information Science from the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. Since 1996 he has been involved with virtual reality. His current research interests are computer graphics, virtual reality, fuzzy logic, simulation, artificial life, and nanocomputing structures; integration of virtual reality in medical education, the integration of the Bluetooth technology in post intensive care units, modelling and simulation of organized groups of moving animals, construction of artificial life using simple linguistic descriptions and extension of quantum-dot cellular automata processing towards muti-valued logic.

Naoki Matsumaru, since October 2002: PhD student in Biosystems Analysis group, a member of Jena Centre of Bioinformatics, in Friedrich-Schiller-University Jena, Jena, Germany, led by Dr. Peter Dittrich; May 2002: Master of science in computer science from Wayne State University, Detroit, USA, supervised by Dr. Klaus-Peter Zauner; March 1998: Bachelor of computer science and computer engineering from University of Aizu, Fukushima, Japan, supervised by Dr. Kshirasager Knaik

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

J. F. Miller, BSc (Lond), PhD (City), PGCLTHE (Bham). Dr. Miller's BSc is Physics and his PhD was on the mathematics of soliton interaction. He joined the Intelligent Systems group in the Department of Electronics, Univ. of York in Oct 2003. He was in the Natural Computation group in the School of Computer Science, Univ. of Birmingham from Sept 1999-Sept 2003. He has chaired eight international conferences in Evolvable Hardware, Genetic Programming and Artificial Life. He is an associate editor of the journals: Genetic Programming and Evolvable Machines, IEEE Trans. on Evolutionary Computation. He is on the editorial board of journals: Evolutionary Computation, Unconventional Computing. His main interests are in the construction of complex and emergent systems that autonomously adapt and self-repair and the exploitation of emergence in materials and otherwise using evolution, and genetic programming. He published over 100 works in related areas.

Ikuko N. Motoike is a Research Associate in the Department of Complex Systems, School of Systems Information Science, Future University – Hakodate, Japan. Her interests are in basic mechanisms of informational processing in living creature, especially without nervous system. Her recent interests are in the effect of the geometry of reaction-diffusion system, on which signal wave propagates. She is attracted to self-sufficient life, and has longing for having green fingers.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Miha Mraz received his BSc (1992), MSc (1995), and PhD (2000) in Computer and Information Science from the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. His PhD thesis was on theory and applications of fuzzy cellular automata. His research is in fuzzy logic, intelligent processing, parallel processing and nanocomputing structures. He is the author or co-author of over 60 papers. He had also collaborated in many projects for industrial partners, such as Ericsson, Iskratel d.o.o., Gorenje d.o.o.. Currently he holds the position of an Assistant Professor at the Faculty of Computer and Information science, University of Ljubljana, Slovenia, where he lectures the courses on "Basics of Modelling and Simulation", "Computer Reliability and Diagnostics" and "Nanotechnologies in Computing Structures".

Osamu Ono received the Bachelor, Master and Doctor Degree in Engineering all from Waseda University, Tokyo, in 1974, 1976, and 1979 respectively. He is a Professor of the Department of Electrical and Electronic Engineering, Meiji University, Japan. He is the Director of Tokyo Branch of Japan Institute Electrical Engineering and a committee member of Japan Society for Simulation Technology (JSST). His research interest includes large scale industrial process, mechatronics, advanced mobile robotics and image processing. Currently, he is interested on the application of DNA computing in engineering field.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Takahide Oya is a 2nd year's PhD student in the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. His current research interests are intelligent single-electron devices that imitate the dynamics of chemical reaction-diffusion systems or neural networks. He has published seven papers, attended 16 international conferences as a speaker, and got two student paper awards of international conferences.

Nicholas G. Rambidi is professor of Physics Department at the Moscow State University, Moscow. Russia. His principal fields of interest are molecular structure and information processing at the molecular level. For the past decade, he has conducted research in molecular and biomolecular computing, designing reaction-diffusion information processing devices. Rambidi received his BS, PhD, and DrS in physical chemistry in 1954, 1959, and 1970, respectively. He is a member of the Russian Academy of Natural Sciences.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Chris Salzberg received his BSc in physics and mathematics from McGill University (2001) and MSc in computational science from the University of Amsterdam (2003). For his M.Sc. thesis, Chris studied a cellular automata-based model of evolving self-reproducing loops originally designed by Dr. H. Sayama. This work was extended during a year-long research visit to Tokyo to work with Dr. Sayama at his lab at the University of Electro-Communications. During this time, Chris became interested in the broader question of whether the separation between information-processing entity (machine) and information carrying entity (tape), normally assumed in computational formalisms, is a necessary one. Working with Dr. Sayama, Chris developed a number of perspectives within which this separation is essentially dissolved, resulting in a novel way of interpreting the concept of "information".

Susan Stepney is Professor of Computer Science at the University of York, UK, and leads the Non-Standard Computation research group there. She is a member of the ACM, Fellow of the British Computer Society, and moderator of the UKCRC Grand Challenge in Non-Classical Computation. Main research interests include novel applications of nature-inspired computation, modelling self-organising complex systems, and designing and reasoning about emergent properties.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Yohei Suzuki is a 1st year's master student in the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. He has experience in electrical and computer engineering and his recent results include LSI implementation of reaction-diffusion computers. He has recently published a japanese book of advanced computer technology including software emulators, file sharing, internet self-defense, and reverse engineering of various computer games (ISBN4-87593-320-7 C9455).

Takahiro Takayama is a 2nd year's master student in School of Systems Information Science, Future University – Hakodate, Japan. His current research interests include artificial intelligence, especially multi-agent framework in network system.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Christof Teuscher obtained his MSc (2000) and his PhD (2004) in CS from the Swiss Fed. Inst. of Techn., Lausanne. He is currently a researcher with the Los Alamos National Laboratory, where he investigates novel, reconfigurable, adaptive, defect- and fault- tolerant architectures for nano-scale devices. His interests range from hardware design to higher-level cognition. He organized the Turing Day in 2002, was program chair of the 5th Int. Workshop on Information Processing in Cells and Tissues, published two books on Alan Turing, more than 50 scientific papers, and was distinguished with several awards.

Yusei Tsuboi received B.Eng. (2001) and M.Eng. (2003) from Meiji University, Japan. He is currently pursuing his PhD at the Institute of Applied DNA Computing, Meiji University, Japan. Since 2004, he is research associate in Department of EEE, Meiji University. He is a student member of IEE: Systems, Man and Cybernetics Soc. (SMC) and Computational Intelligent Soc. (CIS). In 2005, he received Sasakawa Scientific Research Grant from Japan Science Society and also Assistance for Participation in Scientific Meetings Abroad from Inoue Foundation for Science. His current research interests are DNA computing, soft computing, and image processing.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Alexander Tsvetkov is a postgraduate student at the Physics Department of the Moscow State University. He received his BSc in 2003. His recent research work has focused on numerical simulation of chemical nonlinear media.

Heather Turner is currently in the second year of a PhD researching emergent behaviour in cellular automata and related models. She is particularly interested in exploring the suitability of these types of model for the support of multiple higher levels of emergence. Heather is a member of the Non-Standard Computation group in the Department of Computer Science at the University of York, and is supervised by Prof. Susan Stepney. Prior to commencing her PhD, she completed an MRes in Mathematics in the Living Environment, in the Department of Mathematics at the University of York. As her first degree, Heather studied for a BEng in Computer Science, also at the University of York. This course included 14 months of industrial experience, working in software development at two hi-tech firms in Ottawa, Canada.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

Sergey Ulyakhin is a postgraduate student at the Physics Department of the Moscow State University. He received his BS in 2003. His recent research work has focused on experimental investigations of information processing by chemical Belousov-Zhabotinsky media.

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022

THE BOOK BRINGS TOGETHER WORK FROM A MULTIDISCIPLINARY CORE OF SCIENTISTS WHO ARE WORKING IN THE FIELD OF UNCONVENTIONAL COMPUTING. THE GOAL WAS TO PROVIDE A COMMON GROUND FOR DIALOG AND INTERACTION, TO HIGHLIGHT THE LATEST ADVANCES, AND TO DISCUSS THE MAIN DIRECTIONS FOR THE FUTURE. TOPICS INCLUDE PROGRAMMING OF CHEMICAL SYSTEMS, EVOLVING LOGICAL GATES IN LIQUID CRYSTAL, IMAGE PROCESSING IN CHEMICAL MEDIA, REACTION–DIFFUSION ELECTRONIC CIRCUITS FOR COMPUTATION AND PATTERN GENERATION, RULE MIGRATION IN CELLULAR AUTOMATA, MULTI-STATE QUANTUM AUTOMATA, DNA COMPUTING OF SHORTEST PATH PROBLEMS, AND ARTIFICIAL CHEMISTRIES. THE PAPERS COLLECTED IN THIS BOOK PROVIDE A GOOD OVERVIEW OF HOT RESEARCH TOPICS IN THE VIBRANT FIELD OF UNCONVENTIONAL COMPUTING.

ISBN 978-0-9551170-2-2

Teuscher C. and Adamatzky A. (Eds.) Unconventional Computing 2005: From Cellular Automata to Wetware (Luniver Press, 2005).

ISBN-10: 095511702X ISBN-13: 978-0955117022