Collision-Based Computing

Springer-Verlag London Ltd.

Andrew Adamatzky (Ed.)

Collision-Based Computing

Springer

Andrew Adamatzky
Faculty of Computing, Engineering and Mathematical Sciences,
University of the West of England, Bristol, BS16 1QY

British Library Cataloguing in Publication Data
Collision-based computing
1.Cellular automata
I. Adamatzky, Andrew
511.3
ISBN 978-1-85233-540-3

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress.

ISBN 978-1-85233-540-3 ISBN 978-1-4471-0129-1 (eBook)
DOI 10.1007/978-1-4471-0129-1

http://www.springer.co.uk

Originally published by Springer-Verlag London Berlin Heidelberg in 2002

Typesetting: Electronic text files prepared by editor
34/3830-543210 Printed on acid-free paper SPIN 10845232

Preface

This book is about how to do computation in a structureless medium populated with mobile objects. No wires, no valves, nothing is there: just compact patterns wandering in space, smashing one to another and calculating.

— What do I need to build a collision-based computer?
— A couple of balls will do! ... Do you enjoy snooker?
— You're kidding me?
— OK, solitons in optical media, breathers in polymers ...
— You know, I'm not a bench scientist ...
— Then, how about gliders in cellular spaces?
— Hmm, I'll think it over ...

A computing device may be either generally purposive, universal, or specialized. A universal processor can do almost anything; specialized — almost nothing. Personal computers are universal, microwave oven controllers are specialized. One may study two types of universality — logical, or computational, and simulational. Abstract machine as well as real physical, chemical or biological system, is called computationally universal if it implements a functionally complete, or universal, set of Boolean functions in its spatiotemporal dynamic. Most constructions studied in the book are computationally universal, because they realize universal systems of logical gates in hierarchical collisions of mobile objects. If a system simulates behavior of a universal machine, which universality has been already proved, it is called simulationally universal. Somewhere in this book you can find collision-based implementations of simulational universality, related usually to embedding of a Turing machine, a register machine, or a counter machine in a medium with colliding particles, balls or gliders.

A universal processing device can be either structured, heterogeneous, compartmentalized and stationary or structureless, homogeneous, architectureless and dynamic. Structured devices have wires to transmit information, valves to process it; structureless devices have nothing of it. Quanta of information are represented by mobile objects (either by their presence/absence or particular types, colors) that travel in the space. Trajectories of the objects can be seen as wires. The objects change their trajectories or states when smashed to other objects. Thus, information is transformed and computation is implemented.

There are several sources of collision-based computing. Studies dealing with collisions of signals, traveling along discrete chains, one-dimensional cellular automata, lie in the very beginning of the field. The ideas of colliding signals, existing from the nineteenth century in physics and physiology, have been put in automata framework just recently, around 1965, when papers by A. J. Atrubin [3], P. Fisher [13], and A. Waksman [28] were published. Namely, Atrubin studied multiplication in one-dimensional cellular automata; Waksman gave an eight-state solution for a firing squad synchronization problem; and, Fisher showed how to generate prime numbers in cellular automata. The Atrubin-Fisher-Waksman approach triggered development of various imaginable constructs aimed to boost computation-potential of homogeneous automata networks.

In 1971 E.R. Banks [5] showed how to build wires and simple gates in configurations of a two-dimensional binary-state cellular automaton. This was the architecture-based construct. Thus, a wire was represented by a particular stationary configuration of cell states; this was rather a simulation of a "conventional" electrical, or logical, circuit. Banks's design was relieved of its heterogeneity ten years later, when Game of Life made our world "wireless".

In 1982 Elwyn Berlekamp, John Conway and Richard Gay proved that Game of Life "can imitate computers" [6]. They mimicked electric wires by lines "along which gliders travel" and demonstrated how to do a logical gate by crashing gliders one to another. Chapter 25 of their "Winning Ways" [6] brought to us admirable computing designs that do not simply look fresh twenty years later but are still re-discovered again and again by Game of Life enthusiasts all over the Net. Berlekamp, Conway and Gay employed a vanishing reaction of gliders — two crashing gliders annihilate themselves — to build a NOT gate. They adopted Gosper's eater to collect garbage and to destroy glider streams. They used combinations of glider guns and eaters to implement AND and OR gates, and the shifting of a stationary pattern (block) by a mobile pattern (glider) when designing auxiliary storage of information.

"There is even the possibility that space-time itself is granular, composed of discrete units, and that the universe, as Edward Fredkin of M.I.T. and others have suggested, is a cellular automaton run by an enormous computer. If so, what we call motion may be only simulated motion. A moving particle in the ultimate microlevel may be essentially the same as one of our gliders, appearing to move on the macrolevel, whereas actually there is only an alteration of states of basic space-time cells in obedience to transition rules that have yet to be discovered." — finishes Berlekamp-Conway-Gay's book [6]. Their last words were about Fredkin.

Meantime, in 1978 Edward Fredkin and Tommaso Toffoli submitted a one-year project proposal to DARPA, which got funding, and thus started unfolding a chain of remarkable events. Originally, Fredkin and Toffoli aimed to "drastically reduce the fraction of" energy "that is dissipated at each computing step" (see Chapter 2). To design a non-dissipative computer they constructed a new type of a digital logic — conservative logic — that conserves both "the physical quantities in which the digital signals are encoded" and "the information present at any moment in a digital system" (see Chapter 2). They further developed these ideas in the seminal paper "Conservative Logic", published in 1982 ("a second edition" of the paper is included in book as Chapter 3). Thus, a concept of ballistic computers emerged. The Fredkin-Toffoli model of conservative computation — the billiard ball model — explores "elastic collisions involving balls and fixed reflectors". Generally, they proved that *given a container with balls we can do any kind of computation.*

The billiard ball model got yet further push when Norman Margolus invented a cellular-automaton implementation of the model. He published this result in 1984. "Margolus neighbourhood" and "billiard ball model cellular automata" are exploited widely nowadays. We have reprinted his "Physics-Like Models of Computation" as a chapter in our present book.

The story we told you is just one of many possible explanations of how the field of collision-based computing arose.[1]

It is a painful experience to give the book a structure. All problems are equally interesting. All results are equally significant. All authors are equally prominent. Anyway, a linear order must be obeyed. Roughly third of the chapters deal with derivatives of the billiard ball model, other chapters study physical aspects of collision-based computing and the rest discuss particulars of traveling patterns in cellular automata.[2]

---

[1] To get to the Margolus neighborhood you've got to pass the Fredkin gate, walk across the Bunimovich stadium and get out of the Toffoli gate.

[2] While localization in nonlinear media are familiar to almost anyone, educated to a basic science level, a concept of cellular automata, which are tackled in almost every chapter of the book, still puzzles an average student or an academician. In a majority of "theoretical" works cellular automata play a role of a discrete substrate where all scenarios of collision-based computing unfold.

A cellular automaton is an all discrete universe, with discrete time, discrete space and discrete states. "Atoms", or cells, of the universe are arranged in regular structures, called lattices or arrays. Each cell takes discrete states and updates its states in a discrete time, depending on the states of its neighbors. All cells update their states in parallel.

Stanisław Ulam was the first who, in the late 1930s, suggested updating matrix elements in parallel and locally, depending on the states of each element's local neighborhood [26]. The idea was taken by John von Neuman and developed to a theory of self-reproducible machines [18,10]. Zuse's "structured cellular space" [33] is yet another good historical introduction to the field, particularly because it is written by a father of modern computing. Further readings may include Toffoli's and Margolus's "cellular machines" [25], which really turned thousands of amateurs to the field of automata structures, cellular automata simulations and a concept of discrete universe. Wolfram's collection of papers [31] could be a good starting point as well. The cellular-automaton approach to the physical world, outlined in [25], is developed further by Weimar [29]. A series of rigorous results

We start the volume with an arousable piece of text – "Symbol Super Colliders" — authored by Tommaso Toffoli. This is about physics and computation, importance of collision in physical and *other* worlds, and "spacetime tapestry" of an artificial computation. Chapter 1 will turn readers' minds in the right direction.

The following three chapters, Chapters 2, 3 and 4, are classical. They also show what happened over twenty years ago. They are "Design Principles for Achieving High-Performance Submicron Digital Technologies" (written in 1978 and never published before, Chapter 2) and "Conservative Logic" (originally published in *International Journal of Theoretical Physics* in 1982, reprinted here as Chapter 3) by Edward Fredkin and Tommaso Toffoli; and, "Physics of Computation" (originally published in *Physica D* in 1984, reprinted here as Chapter 4).[3]

Chapter 4 proposes a billiard ball model cellular automaton; this construct is employed in several chapters of the book. An impression of the transition from the past to the present is particularly strong when you are getting to Chapter 5 — "Universal Cellular Automata Based on the Collisions of Soft Spheres" — by Norman Margolus. Essentially, computation with soft spheres is more akin to computation in a lattice gas system. Norman Margolus derives perfect momentum conserving models of ballistic computation by removing mirrors out of the computation space. In this context, he considers reflections without mirrors, crossover and routing of signals, dual-rail logic, and updates his original billiard ball model cellular automaton to incorporate soft spheres. The chapter closes with an intriguing excursion in relativistic cellular automata and semi-classical models of collision dynamics. The next two chapters continue the study of ballistic computing models along the lines drawn by Norman Margolus.

Thus, in Chapter 6 — "Computing Inside the Billiard Ball Model" — Jérome Durand-Lose applies his expertise in reversible computing, automata models of transition phenomena and grain sorting in sand piles to derive intriguing results related to reversible cellular automata models of collision-

---

on cellular-automata simulation of nonlinear media can be found in Chopard and Droz's book [12].

Other recommended reading may include Aladyev's [4] and Voorhees's [27] "algebraic" treats of cellular automata, and Wuensche's and Lesser's [32] guide to global dynamics of one-dimensional cellular automata. Theoretical aspects of cellular-automaton theory of parallel computing are well presented in Garzon's book [14] and a collective monograph [16] edited by Mazoyer and Delorme. While talking about computing it is also worth looking at Chaudhuri's text [11] on cellular-automata based solutions of various codes and combinatorial logics, and Sipper's [22] and Adamatzky's [2] studies.

[3] These three texts are not simply reprinted, they are typeset from scratch, some figures are redrawn, and the chapters are carefully checked and corrected by the authors. So, Chapters 2, 3 and 4 look rather like second editions of the original papers.

based computing. Firstly, Jérome Durand-Lose constructs block cellular automata, aka partitioning cellular automata, as a generalization of billiard ball model cellular automata. Then he shows, via implementation of reversible logic gates, that a two-counter machine is simulated in block cellular automata. Several problems of intrinsic universality and uncomputability in billiard ball model cellular automata are tackled in the chapter as well.

Kenichi Morita and his colleagues have a solid track record in studies of cellular automata, reversible computing, grammar and grammar arrays, and logical systems for knowledge representations. Their first result is dated back to 1986 when Morita constructed a memory unit from Fredkin gate [17]. Other findings include computational universality of one- and two-dimensional cellular automata, self-reproduction and solution of firing squad synchronization problem in reversible cellular automata. The title "Universal Computing in Reversible and Number-Conserving Two-Dimensional Cellular Spaces" of Chapter 7, by Kenichi Morita, Yasuyuki Tojima, Katsunobu Imai and Tsuyoshi Ogiro, speaks for itself. There, Fredkin gate, a basic element of conservative logic, is embedded in a bit-conserving reversible partitioning cellular automaton. This embedding is demonstrated via generalization of Margolus's billiard ball model cellular automaton to more complicated grids and advanced state transition functions. Then (reversible) counter machines are compactly (with a use of rotating mirrors) implemented in a space-time dynamic of the reversible automata.

Novel ways of logical formalization of collision-based computing models are suggested in Chapter 8 — "Derivation Schemes in Twin Open Set Logic" — by Michael D. Westmoreland and Joan Krone. The authors offer several alternative derivation schemes and logical systems that may well describe ballistic models of computation in a more realistic way than it was done before. For example, with use of Westmoreland-Krone logical primitives a sensitivity to initial conditions and particulars of final results' measurement can be sensibly assessed.

The authors put the first touches on the picture of their theory in 1993 when Michael Westmoreland and Benjamin Schumacher wrote a paper on non-classical logics for classical systems [30]. In this chapter Michael Westmoreland and Joan Krone fuse their experience in proof rules, classical phase space logics, verification of functionality, quantum channels, inductive structures and three-valued logics to present a non-standard, alternative, logical description of collision-based computing models.

At this stage we've put aside reversibility of computation, the billiard ball model — we can do without it when entering the world of gliders, particles, automata signals, solitons and other mobile localizations in nonlinear media.

Chapter 9 — "Signals in Cellular Automata" — by Marianne Delorme and Jacques Mazoyer exposes a huge slice of modern theory of "geometrical computation" in one- and two-dimensional cellular automata. Actually, it was Jacques Mazoyer who strikingly improved the solution of the firing squad synchronization problem in 1987 [15]. Also, in 1998 Jacques Mazoyer and Marianne Delorme edited a collection of inquiries in computing potential of cellular automata [16]. That book is, possibly, the first ever consistent collection of contributions devoted entirely to the computation *not* simulation aspects of cellular automata.

In this book, Marianne Delorme and Jacques Mazoyer study various types of cellular automata, which support propagation of information quanta, or signals. Particularly, they show how a cellular automaton can transform one type of a signal to another. They demonstrate how to design a one-dimensional cellular automaton that supports infinite families of signals of different speeds. Feasibility of Delorme-Mazoyer constructions is demonstrated in problems of multiplication in one-dimensional cellular automata.

In 1974 Kenneth Steiglitz wrote a textbook on discrete systems [23]; then he authored other academic bestsellers — an introduction to discrete optimization [19] (printed in 1982 and reprinted in 1998) and a handbook on digital signal processing [24], published in 1996. This explains why Ken Steiglitz was one of those who discovered parity filter cellular automata [20]. The automata of this type are not simply analogs of infinite impulse response digital filters but they also exhibit soliton-like dynamics of localizations. This discovery led to the formation of a concept of particle machines — machines that perform computation by colliding particles in one-dimensional cellular automata. So far, in Chapter 10 — "Computing with Solitons: A Review and Prospectus" — Mariusz Jakubowski, Ken Steiglitz and Richard Squier invite us to take a brief tour into a theory of particle machines and its application to computing with one-dimensional solitons. Various designs of soliton gates are discussed in a context of massively-parallel processors.

Theory of particle machines and discrete, automata, analogs of solitons are studied in a context of iterated automaton maps, or iterons, by Paweł Siwak, Chapter 11 — "Iterons of automata". Paweł considers two classes of iterons. The first class includes mobile localizations, signals or particles, which emerge in classical cellular automata, cells of which update their states in parallel. The second class of iterons consists of traveling patterns arising in serially updated automata networks. The serial updating of an automaton chain is similar to a sort of filtering known as infinite impulse response or recursive filtering. The chapter gives us striking examples of phenomenology of particles in parallel and serial automaton chains. We are not told what types of logical functions can be implemented in collisions of the studied particles. However, for searching minds this is not a problem. Just from the pictures of particles collisions one can derive a great variety of possible collisions-based computational operations. Amongst other remarkable features, Siwak's chapter considers automata localisations in "classical" terms of mathematical machines, thoroughly classifies most types of discrete signals, builds a parallel between automata gliders, solitons and digital filters, and suggests ways to design automata rules that support solitons.

Steve Blair and Kelvin Wagner, authors of Chapter 12 — "Gated Logic with Optical Solitons", are renowned experts in theory and *practice* of optical computing. They are the people who brought soliton-based computing to the realms of the physical world. They are the guys who not simply think but act for the future. Their views on soliton logic, expressed in Chapter 12, result from decades of research in acousto-optic matrix multipliers, many-dimensional optical soliton dragging logic, adaptive optical networks, pulse propagation in non-resonant materials, interaction of solitons and nonlinearities, acousto-optic devices, cascadable spatial soliton circuits and tunable optic filters.

First, Steve Blair and Kelvin Wagner give an accessible introduction to digital logic and discuss a set of requirements to a logical gate. Second, they show why solitons are good for collision-based computing; temporal, spatial and spatio-temporal soliton logic gates are designed there. Third, a typical logical circuit requires composition of many gates, and they are studied in the rest of the chapter. To demonstrate that soliton logical gates may form self-consistent cascades with signal fanout, Steve Blair and Kelvin Wagner study gate transfer function, details of spatial soliton dragging and collision interactions. They prove feasibility of their approach by designing cascaded inverters and NOR gates.

In Chapter 13 —"Finding Gliders in Cellular Automata" — Andrew Wuensche describes how to classify cellular-automaton rules for a spectrum of ordered, complex — supporting gliders, and chaotic dynamics. Also methods of "automatic" filtering of gliders and parametrization of automata global dynamics are discussed there.

The chapter arose from Andrew Wuensche's inquires into space-time dynamics of discrete matter at the edge of phenomenology and complexity. In 1992 Andy presented us with his marvelous atlas of global dynamics of one-dimensional cellular automata [32]. His volume gives an accessible introduction to discrete dynamics, guides to parametrization of global dynamics of automata networks, and displays breathtaking pictures of cellular-automata global transition graphs. Wuensche's chapter in this book is as lavishly illustrated as any of his publications.

There are traveling localizations everywhere — solitons in optical media, breathers in polymers, excitons in mono-molecular arrays, worms in liquid crystals, groups of oscillons in vibrating granular materials and quasi-particles in reaction-diffusion systems. The phenomena are discussed in Andrew Adamatzky's Chapter 14 — "New Media for Collision-Based Computing". An illuminating comparison of logical-gate architectures realized in "real" systems and their automata models gives us a vision of what types of collision-based computer prototypes can be built in laboratories.

The findings, discussed in Chapter 14, result from research directions outlined in Adamatzky's previous books on reconstruction of cell state transition rules from global configurations of cellular automata [1] and on spatial computation in active nonlinear media [2].

Particle dynamics on two-dimensional lattices with fixed, rotating or flipping mirrors is amazingly interpreted in terms of Turing machines by Leonid Bunimovich and Milena Khlabystova in Chapter 15. In their lattice gas model of a Turing machine, a particle, hoping from one vertex to another, represents a reading or writing head of the machine. The lattice is populated with mirrors, which are analogs of symbols, written on the tape. When traveling on the lattice, particles rotate or flip mirrors thus updating contents of the Turing tape.

A knowledgeable reader would benefit from a look at the previous publications by Leonid Bunimovich, related to dynamics theory. We could refer to a wonderful collection of papers on dynamical systems, compiled and edited by Leonid in 1989. More recent results on dynamical systems are summarized in two more books — notes of Sinai's seminars on dynamical systems [8] and ergodic theory of dynamical systems [9]. The Bunimovich-Khlabystova results on dynamic of Lorentz gases may also help us to refresh and reconsider ideas related to classical models of collision-based universality, billiard ball model or number-conserving reversible cellular automata.

A book should close on a nice, accessible and, possibly, not too boring note. The last three chapters of the book do this well.

Chapter 16 combines arithmetic operations, implemented in a particle machine, with a self-replicating loop. All authors of the chapter — Enrico Petraglio, Gianluca Tempesti and Jean-Marc Henry — are from the famous Logics Systems Lab (EPFL, Switzerland), an incubator of embryonic electronics, bio-inspired machines and evolving reconfigurable hardware. The ultimate goal of the approach is to find efficient ways of "natural" growing of large-scale integrated circuits. In the chapter, a cellular automaton is developed that is capable of self-replication, based on a modified version of the Langton loop. Techniques of Jakubowski-Steiglitz-Squier particle-machine computation are adopted and modified to program the self-replicating automaton to implement such arithmetical tasks as binary addition and multiplication.

Game of Life cellular automaton is the first formal model which is proved to be collision-computationally universal [6]. It is the most famous and the most talked about cellular automaton. Surprisingly, the Game of Life did not get proper treatment in academic journals — significant results and miraculous constructions are still attributed rather to cyberspace. To fill the gap, and to attract more Game of Life fans to the field of collision-based computing, we include two chapters dealing with the Game of Life.

Chapter 17 — "Implementation of Logical Functions in the Game of Life" — by Jean-Philippe Rennard gives a brief introduction to the subject and then shows particulars of logical gate implementation via collision of glider streams. The chapter partly overlaps with a popular excursion to the field of Artificial Life, prepared by Jean-Philippe Rennard [21].

Sophisticated and detailed constructions of Game of Life implementation of a universal Turing machine are presented by Paul Rendell in his Chapter 18 — "Turing Universality of the Game of Life". The constructions include an adder, a sliding block memory, a memory cell and many more fascinating parts. Even a finite state device and a Turing tape are designed from stationary cellular patterns, glider and spaceship guns, and other curiosities.

Acknowledgements

All contributions were invited, some solicited, others begged for. The authors of the book are remarkable personalities and it is a miracle to have them all under one book cover. Thank you, colleagues and friends, for your genius way of thinking, for your hard working, for your deep vision and for your understanding of life.

We are in debt to many reviewers who scrutinized every small part of the text and assured us in originality, novelty and timeliness of the contributions.

Also, much thanks to Rights and Permissions Departments of Taylor & Francis Group, Elsevier Science, Plenum Press/Kluwer Academic Publishers and The Institute of Electrical and Electronics Engineers, Inc. for sincerely granting us their consents to publish some works.

Our publishers — Springer-Verlag London — were speedy, efficient, knowledgeable and supportive.

References

1. Adamatzky A. *Identification of Cellular Automata* (Taylor and Francis, 1994).
2. Adamatzky A. *Computing in Nonlinear Media and Automata Collectives* (Institute of Physics Publishing, 2001).
3. Atrubin A.J. An iterative one-dimensional real-time multiplier. *IEEE Trans. Electron. Computers* **EC-14** (1965) 394–399.
4. Aladyev V. *Mathematical Theory of Homogeneous Structures and Their Applications* (Valgue Press, Tallinn, 1980).
5. Banks E.R. Information and transmission in cellular automata *Ph.D. Diss.* (MIT, 1971). Cited by [25].
6. Berlekamp E.R., Conway J.H. and Guy R.L. *Winning Ways for your Mathematical Plays* vol. 2: Games in Particular (Academic Press, 1982).
7. Bunimovich L.A. (Editor) *Dynamical Systems* (Springer-Verlag, 1989).
8. Bunimovich L.A., Gurevich B.M. and Pesin Ya.B. (Editors) *Sinai's Moscow Seminar on Dynamical Systems* (American Mathematical Society, 1995).
9. Bunimovich L.A., Dani S.G., Dobrushin R.L., Jakobson M.V. (Editors) *Dynamical Systems, Ergodic Theory and Applications* (Springer-Verlag, 2000).
10. Burks A.W. (Editor) *Essays on Cellular Automata* (University of Illinois Press, 1970).
11. Chaudhuri P.P., Chawdhury D.R., Nandi S. and Chattopadhyay S. *Additive Cellular Automata: Theory and Applications* (IEEE Computer Science Press, 1997).

12. Chopard B. and Droz M. *Cellular Automata Modeling of Physical Systems* (Cambridge University Press, 1999).
13. Fisher P. Generation of primes by a one dimensional real time iterative array *Journal of the ACM* **12** (1965) 388–394.
14. Garzon M. *Models of Massive Parallelism: Analysis of Cellular Automata and Neural Networks* (Springer-Verlag, 1995).
15. Mazoyer J. A six states minimal time solution to the Firing Squad Synchronization Problem *Theoretical Computer Science* **50** (1987) 183–238.
16. Delorme M. and Mazoyer J. (Editors) *Cellular Automata: A Parallel Model* (Kluwer Academic Publishers, 1998).
17. Morita K. Construction of a cellular structure memory unit out of Fredkin's reversible and conservative logic gate *Trans. IECE Japan* **J69-D** (1986) 860-867.
18. Neumann von J. *Theory of Self–Reproducing Automata* (University of Illinois Press, 1966).
19. Papadimitriou C.H. and Steiglitz K. *Combinatorial Optimization Algorithms* (Prentice-Hall, 1982; Dover, New York, 1998).
20. Park J.K., Steiglitz K., and Thurston W.P. Soliton-like behavior in automata. *Physica D*, **19** (1986) 423–432.
21. Rennard J.-P. *Vie Artificielle. Introduction et base d'implémentation Java* (Vuibert, Paris, 2002).
22. Sipper M. *Evolving of Parallel Cellular Machines* (Springer-Verlag, 1997).
23. Steiglitz K. *Introduction to Discrete Systems* (John Wiley, 1974).
24. Steiglitz K. *A Digital Signal Processing Primer* (Addison-Wesley, 1996).
25. Toffoli T. and Margolus N. Cellular Automata Machines (MIT Press, 1987).
26. Ulam S.M. *A Collection of Mathematical Problems* (Interscience Publishers, 1960), p. 30.
27. Voorhees B. *Computational Analysis of One–Dimensional Cellular Automata* (World Scientific, 1996).
28. Waksman A. An optimum solution to the firing squad synchronization problem *Information and Control* **9** (1966) 66–78.
29. Weimar J. *Simulation with Cellular Automata* (Logos–Verlag, 1998).
30. Westmoreland M.D. and Schumacher B.W. Non-Boolean derived logics for classical systems *Phys. Rev. A* **48** (1993) 977–985.
31. Wolfram S. *Cellular Automata and Complexity* (Addison–Wesley, 1996).
32. Wuensche A. and Lesser M.J. *The Global Dynamics of Cellular Automata* (Addison-Wesley, 1992).
33. Zuse K. *Rechnender Raum* (Friedrich Vieweg & Sohn, Braunschweig, 1969).

Bristol and Bath

Andrew Adamatzky
February 2002

Contents

1 Symbol Super Colliders
Tommaso Toffoli 1

1.1 Cellular Automata and Lattice Gases 3
1.2 Heat, Ice, and Waves 10
1.3 Colliding-Beams Particle Accelerators 14
1.4 Why Aristotle Didn't Discover Universal Gravitation 18
1.5 "On The Nature of the Universe" 20
1.6 Conclusions 22
References 22

Part I Twenty Years Ago

2 Design Principles for Achieving High-Performance Submicron Digital Technologies
Edward F. Fredkin and Tommaso Toffoli 27

2.1 Overview 29
  2.1.1 Objectives 29
  2.1.2 Conceptual Framework 30
  2.1.3 Organization 31
2.2 Principles of Conservative Logic 31
  2.2.1 Motivations 31
  2.2.2 Conservative Logic 33
  2.2.3 Implementation of Conservative Logic in Concrete Computing Devices 37
2.3 Prospects for Applications to Sub-Micron Digital Technologies 42
  2.3.1 Generalities 42
  2.3.2 Josephson-Effect Switching 42
  2.3.3 Integrated Optics 44
References 45

3 Conservative Logic
Edward Fredkin and Tommaso Toffoli 47

3.1 Introduction 47

3.1.1 Physical Principles Already Contained in the Axioms 48
3.1.2 Some Physical Principles that Haven't yet Found a Way into the Axioms 49
3.2 Conservative Logic: The Unit Wire and the Fredkin Gate 51
3.2.1 Essential Primitives for Computation 52
3.2.2 Fundamental Constraints of a Physical Nature 52
3.2.3 The Unit Wire 53
3.2.4 Conservative-Logic Gates; the Fredkin Gate 54
3.2.5 Conservative-Logic Circuits 55
3.3 Computation in Conservative-Logic Circuits; Constants and Garbage 56
3.4 Computation Universality of Conservative Logic 59
3.5 Nondissipative Computation 61
3.6 A "Billiard Ball" Model of Computation 64
3.6.1 Basic Elements of the Billiard Ball Model 65
3.6.2 The Interaction Gate 66
3.6.3 Interconnection; Timing and Crossover; the Mirror 67
3.6.4 The Switch Gate and the Fredkin Gate 68
3.7 Garbageless Conservative-Logic Circuits 70
3.7.1 Terminology: Inverse of a Conservative-Logic Network; Combinational Networks 71
3.7.2 Role of the Scratchpad Register. Trade-Offs Between Space, Time, and Available Primitives 76
3.7.3 Circuits that Convert Argument into Result. General-Purpose Conservative-Logic Computers 77
3.8 Energy Involved in a Computation 78
3.9 Other Physical Models of Reversible Computation 79
3.10 Conclusions 79
References 80

4 Physics-Like Models of Computation
Norman Margolus 83

4.1 Introduction 83
4.2 Cellular Automata 83
4.3 Reversible Cellular Automata 84
4.4 Entropy in RCA 85
4.5 Conservation Laws in Second-Order RCA 86
4.6 First-Order RCA 89
4.7 The Billiard Ball Model 91
4.8 The BBM Cellular Automaton 93
4.9 Relationship of BBMCA to Conservative Logic 97
4.10 Energy in the BBMCA 98
4.11 Conclusion 99
4.12 Appendix: A Second-Order, Reversible, Universal Automaton 99

References 103

Part II The Present and the Future

5 Universal Cellular Automata Based on the Collisions of Soft Spheres
Norman Margolus 107

5.1 Fredkin's Billiard Ball Model 109
5.2 A Soft Sphere Model 111
5.3 Other Soft Sphere Models 114
5.4 Momentum Conserving Models 115
5.4.1 Reflections Without Mirrors 116
5.4.2 Signal Crossover 116
5.4.3 Spatially-Efficient Computation 117
5.4.4 Signal Routing 118
5.4.5 Dual-Rail Logic 119
5.4.6 A Fredkin Gate 120
5.4.7 Implementing the BBMCA 121
5.4.8 Signal Routing Revisited 123
5.4.9 A Simpler Extension 125
5.4.10 Other Lattices 128
5.5 Relativistic Cellular Automata 129
5.6 Semi-Classical Models of Dynamics 132
5.7 Conclusion 132
References 133

6 Computing Inside the Billiard Ball Model
Jérôme Durand-Lose 135

6.1 Definitions 137
6.1.1 Block Cellular Automata 138
6.1.2 Reversibility 138
6.1.3 Simulation 139
6.1.4 Cellular Automata 139
6.1.5 Relations with Classical Cellular Automata 140
6.2 Universality of One-Dimensional Block Cellular Automata 141
6.3 Billiard Ball Model 143
6.3.1 Basic Encoding 144
6.3.2 Conservative Logic 146
6.3.3 Dual Encoding 147
6.3.4 Reversible Logic 148
6.4 Turing Universality of the BBM 149
6.4.1 Automaton 149
6.4.2 Counters 150

6.5 Intrinsic Universality of the BBM . . . 152
6.5.1 Partitioned Cellular Automata . . . 153
6.5.2 Intrinsic Universality of the BBM among R-CA . . . 154
6.5.3 Space-time Simulation . . . 156
6.5.4 Intrinsic Space-Time Universality of the BBM among CA . . . 157
6.6 Uncomputable Properties . . . 157
6.6.1 Reaching a Stable or Periodic Configuration . . . 158
6.6.2 Reaching a (Sub-)Configuration . . . 158
References . . . 159

7 Universal Computing in Reversible and Number-Conserving Two-Dimensional Cellular Spaces
Kenichi Morita, Yasuyuki Tojima, Katsunobu Imai and Tsuyoshi Ogiro 161

7.1 Number-Conserving Reversible Cellular Automaton . . . 163
7.2 Embedding Fredkin Gate in Simple Universal Two-Dimensional Bit-Conserving Reversible Partitioning Cellular Automata . . . 165
7.2.1 16-State Model with Rotation and Reflection Symmetric Rules . . . 168
7.2.2 16-State Model with Rotation Symmetric Rules . . . 170
7.2.3 8-State Triangular Model . . . 171
7.3 Compact Embedding of Reversible Counter Machine in Universal Number-Conserving Reversible Partitioning Cellular Automata . . . 171
7.3.1 Reversible Counter Machine . . . 172
7.3.2 4^4-State Model . . . 177
7.3.3 3^4-State Model . . . 193
7.4 Conclusion . . . 194
References . . . 198

8 Derivation Schemes in Twin Open Set Logic
Michael D. Westmoreland and Joan Krone . . . 201

8.1 Derived Logical Systems . . . 201
8.2 Twin Open Set Logic . . . 203
8.3 Twin Open Set Logic and Classical Logic . . . 209
8.4 Derivation Schemes in Twin Open Set Logic . . . 213
8.4.1 Nonstandard Derivation Methods (or what to do when you can't do *modus ponens*) . . . 215
8.4.2 Derivation Schemes under Nonstandard Entailment . . . 217
8.4.3 Nonstandard Implication . . . 220
8.5 Tautologies in Twin Open Set Logics . . . 221
8.6 Derivation Schemes in Collision Models . . . 223
8.7 Conclusion . . . 227
References . . . 229

9 Signals on Cellular Automata
Marianne Delorme and Jacques Mazoyer 231

9.1 Some Initial Definitions and Comments 232
9.1.1 Signals 233
9.1.2 Signals and Grids 240
9.2 Transformations of Signals 243
9.2.1 Transforming Marks on a Cell into Some Right Signal 244
9.2.2 Some Right Signals from Some Other Ones 250
9.3 Infinite Families of Signals and Grids 256
9.3.1 Infinite Families of Signals. 257
9.3.2 Computations and Grids 263
9.3.3 Infinite Families of Signals (or Waves) on Two-Dimensional Cellular Automaton 270
9.4 Conclusion 274
References 274

10 Computing with Solitons: A Review and Prospectus
Mariusz H. Jakubowski, Ken Steiglitz, Richard Squier 277

10.1 Computation in Cellular Automata 278
10.2 Particle Machines (PMs) 278
10.2.1 Characteristics of PMs 279
10.2.2 The PM Model 280
10.2.3 Simple Computation with PMs 280
10.2.4 Algorithms 281
10.2.5 Comment on VLSI Implementation 283
10.2.6 Particles in Other Automata 283
10.3 Solitons and Computation 286
10.3.1 Scalar Envelope Solitons 286
10.3.2 Integrable and Nonintegrable Systems 287
10.3.3 The Cubic Nonlinear Schrödinger Equation 287
10.3.4 Oblivious and Transactive Collisions 287
10.3.5 The Saturable Nonlinear Schrödinger Equation 288
10.4 Computation in the Manakov System 288
10.4.1 The Manakov System and its Solutions 288
10.4.2 State in the Manakov System 289
10.4.3 Particle Design for Computation 290
10.5 Conclusion 292
References 294

11 Iterons of Automata
Paweł Siwak 299

11.1 Homogeneous Nets of Automata 303
11.2 Cellular Automata — Parallel Processing of Strings 309
11.3 Linear Automaton Media — Serial Processing of Strings 312

11.4 Particles of Cellular Automata ... 317
11.5 Filtrons of Serial Processing ... 323
11.6 The Automata Based on FCA Window ... 326
11.6.1 Jiang Model ... 327
11.6.2 AKT Model ... 329
11.6.3 FPS Filters ... 330
11.6.4 F Model ... 331
11.6.5 FM Filters ... 333
11.6.6 BRS Models ... 333
11.6.7 Soliton Automata ... 338
11.7 Automata of Box-Ball Systems and Crystal Systems ... 339
11.7.1 Ball Moving Systems ... 339
11.7.2 Crystal Systems ... 342
11.8 Conclusion ... 346
References ... 348

12 Gated Logic with Optical Solitons
Steve Blair and Kelvin Wagner ... 355

12.1 Solitons for Digital Logic ... 358
12.1.1 Temporal Soliton Logic Gates ... 359
12.1.2 Spatial Soliton Logic Gates ... 361
12.1.3 Spatio-Temporal Soliton Logic Gates ... 366
12.2 Cascadability of Spatial Soliton Logic Gates ... 367
12.2.1 Soliton Logic Gate Transfer Function ... 367
12.2.2 Interaction Details ... 370
12.2.3 Cascaded Inverters ... 371
12.2.4 Cascaded 2-NOR Gates ... 374
12.3 Conclusions ... 377
References ... 379

13 Finding Gliders in Cellular Automata
Andrew Wuensche ... 381

13.1 One-Dimensional Cellular Automaton ... 384
13.2 Trajectories and Space-Time Patterns ... 385
13.3 Basins of Attraction ... 386
13.4 Constructing and Portraying Attractor Basins ... 387
13.5 Computing Pre-Images ... 389
13.5.1 The Cellular Automata Reverse Algorithm ... 390
13.5.2 The Z Parameter ... 391
13.6 Gliders in One-Dimensional Cellular Automata ... 392
13.7 Input-Entropy ... 396
13.7.1 Ordered Dynamics ... 397
13.7.2 Complex Dynamics ... 397
13.7.3 Chaotic Dynamics ... 398

13.8 Filtering . . . 398
13.9 Entropy-Density Signatures . . . 401
13.10 Automatically Classifying Rule-Space . . . 402
13.11 Attractor Basin Measures . . . 405
13.12 Glider Interactions and Basins of Attraction . . . 407
13.13 Conclusion . . . 408
13.14 The DDLab Software . . . 409
References . . . 409

14 New Media for Collision-Based Computing
Andrew Adamatzky . . . 411

14.1 Molecular Chains . . . 412
14.2 Molecular Array Processors . . . 414
14.3 Bulk Media Processors . . . 416
14.4 Liquid-Crystal Processors . . . 421
14.5 Granular-Material Processors . . . 422
14.6 Reaction-Diffusion Processors . . . 423
14.7 Automata Models of Computing with Localizations . . . 426
14.7.1 Automata Solitons . . . 426
14.7.2 Models of Molecular Chains . . . 426
14.7.3 Models of Molecular Arrays . . . 432
14.7.4 Automata Worms . . . 435
14.7.5 Excitable Lattices and Reaction-Diffusion . . . 436
14.8 Conclusion . . . 438
References . . . 438

15 Lorentz Lattice Gases and Many-Dimensional Turing Machines
Leonid A. Bunimovich and Milena A. Khlabystova . . . 443

15.1 Dynamical Models of Turing Machines . . . 445
15.2 General Properties of Lorentz Lattice Gases (LLG) . . . 447
15.3 Description of Models . . . 449
15.3.1 Regular Lattices . . . 450
15.3.2 Delaunay Random Lattice . . . 450
15.4 Some Results on LLG with Fixed Scatterers . . . 450
15.5 LLG with Flipping Scatterers . . . 452
15.5.1 Flipping LLG with One Moving Particle . . . 453
15.5.2 Flipping LLG with Infinitely Many Moving Particles . . . 460
15.6 Conclusion . . . 464
References . . . 465

16 Arithmetic Operations with Self-Replicating Loops
Enrico Petraglio, Gianluca Tempesti and Jean-Marc Henry 469

16.1 Self-Replicating Cellular Automata 471
16.1.1 Von Neumann's Automaton 471
16.1.2 Langton's Loop 472
16.1.3 The New Automaton 472
16.2 Description of the Automaton 475
16.2.1 Cellular Space and Initial Configuration 475
16.2.2 Operation 476
16.2.3 Example 479
16.3 Collision-Based Computing: Theoretical Notions 480
16.3.1 Binary Addition 480
16.3.2 Binary Multiplication 482
16.4 Implementation on Self-Replicating Loops 483
16.4.1 Addition 484
16.4.2 Multiplication 485
16.4.3 Combinations of Multiplication and Addition 486
16.5 Conclusion 487
References 490

17 Implementation of Logical Functions in the Game of Life
Jean-Philippe Rennard 491

17.1 Basic Features of the Game of Life 492
17.2 Logical Gates 495
17.3 Collision Reactions 496
17.3.1 Glider Collisions 496
17.3.2 Eaters 498
17.4 Implementation of Logical Gates 498
17.4.1 Input 499
17.4.2 Output 500
17.5 Coupling the Components 500
17.5.1 The AND-Gate 501
17.5.2 The OR-Gate 502
17.5.3 The NOT-Gate 504
17.6 Implementation of Boolean Equations 505
17.6.1 Gates Associations 505
17.6.2 Management of the NOT-Gate 507
17.7 Binary adder 508
17.8 Conclusion 511
References 511

18 Turing Universality of the Game of Life
Paul Rendell ... 513

18.1 Some Game of Life Patterns ... 514
18.1.1 Adder ... 515
18.1.2 Sliding Block Memory ... 517
18.1.3 Memory Cell ... 518
18.2 Construction of the Turing Machine ... 521
18.3 The Finite State Machine ... 521
18.3.1 Selection of a Row ... 522
18.3.2 Selection of a Column ... 523
18.3.3 Set Reset Latch ... 523
18.3.4 Collecting Data from the Memory Cell ... 525
18.4 The Tape ... 526
18.5 Coupling the Finite State Machine with the Stacks ... 534
18.6 The Machine in the Pattern ... 536
18.7 Extending the Pattern to Make a Universal Turing Machine ... 537
18.8 Conclusion ... 537
References ... 538

Index ... 541

List of Contributors

Andrew Adamatzky
Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England
Bristol BS16 1QY
UK

Steve Blair
Department of Electrical Engineering
University of Utah
Salt Lake City
UT 84112-9206
USA

Leonid A. Bunimovich
Southeast Applied Analysis Center
Georgia Institute of Technology
Atlanta GA 30332-0160
USA

Marianne Delorme
Ecole Normale Supérieure de Lyon
Laboratoire de l'Informatique du Parallélisme
46 Allée d'Italie
F-69364 Lyon Cedex 07
France

Jérôme Durand-Lose
Laboratoire I3S, CNRS UMR 6070
930 Route des Colles, BP 145
F-06 903 Sophia Antipolis Cedex
France

Edward Fredkin
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213
USA

Jean-Marc Henry
Logic Systems Laboratory
Computer Science Department
Swiss Federal Institute of Technology
CH-1015 Lausanne
Switzerland

Mariusz H. Jakubowski
Computer Science Department
Princeton University
Princeton NJ 08544
USA

Katsunobu Imai
Faculty of Engineering
Hiroshima University
Higashi-Hiroshima 739-8527
Japan

Milena A. Khlabystova
School of Mathematics
Georgia Institute of Technology
Atlanta GA 30332-0160
USA

Joan Krone
Department of Mathematics and Computer Science
Denison University
Granville OH 43023
USA

Norman Margolus
MIT Artificial Intelligence Laboratory
200 Technology Square
Cambridge MA 02139
USA

Jacques Mazoyer
Ecole Normale Supérieure de Lyon
Laboratoire de l'Informatique du Parallélisme
46 Allée d'Italie
F-69364 Lyon Cedex 07
France

Kenichi Morita
Faculty of Engineering
Hiroshima University
Higashi-Hiroshima 739-8527
Japan

Tsuyoshi Ogiro
Faculty of Engineering
Hiroshima University
Higashi-Hiroshima 739-8527
Japan

Enrico Petraglio
Logic Systems Laboratory
Computer Science Department
Swiss Federal Institute of Technology
CH-1015 Lausanne
Switzerland

Jean-Philippe Rennard
1 Rue de la Providence
F-74000 Annecy
France

Paul Rendell
124 Manfield Avenue
Coventry
West Midlands CV2 2HN
UK

Paweł Siwak
Poznan University of Technology
60-965 Poznan
Poland

Richard Squier
Computer Science Department
Georgetown University
Washington DC 20057
USA

Ken Steiglitz
Computer Science Department
Princeton University
Princeton NJ 08544
USA

Gianluca Tempesti
Logic Systems Laboratory
Computer Science Department
Swiss Federal Institute of Technology
CH-1015 Lausanne
Switzerland

Tommaso Toffoli
ECE Department
Boston University
8 Saint Mary's St.
Boston MA 02215
USA

Yasuyuki Tojima
Faculty of Engineering
Hiroshima University
Higashi-Hiroshima 739-8527
Japan

Kelvin Wagner
Optoelectronic Computing
Systems Center
Department of
Electrical Engineering
University of Colorado
Boulder CO 80309-0425
USA

Michael D. Westmoreland
Department of Mathematics
and Computer Science
Denison University
Granville OH 43023
USA

Andrew Wuensche
Santa Fe Institute
1399 Hyde Park Road
Santa Fe NM 87501
and
Discrete Dynamics, Inc.
Rt.1, Box 92ac
Santa Fe NM 87501
USA