Special Purpose Parallel Computer for Modelling Supramolecular Systems based on the Dynamic Lattice Liquid Model

Piotr Polanowski[1]*, Jarosław Jung[1], Rafał Kielbik[2]

[1]*Technical University of Łódź, Department of Molecular Physics, Żeromskiego 116, 90-924 Łódź, Poland*
[2]*Technical University of Łódź, Department of Microelectronics and Computer Science*
al. Politechniki 11, 90-924 Łódź, Poland
*e-mail: ppolanow@p.lodz.pl

Abstract: A predictive description of the synthetic processes leading to complex macromolecules requires consideration of the spatial models. Hierarchical, structural and dynamic complexities of such systems require new methodological developments. The presented work provides effective tools for the modelling of the complex synthetic processes in the 3D space under controlled conditions. We propose constructing a parallel computing system which realizes the 3D architecture and the cooperative dynamics based on the dynamic lattice liquid (DLL) model. This can create new standards in the spatial resolution of the models allowing simulations of the systems comparable in sizes and complexity with the biological cells in the future.
Key words: liquid dynamics, Monte Carlo simulations, FPGA, dedicated parallel computer

I. INTRODUCTION

Over the last years the computer simulations have become one of the fundamental investigation instruments, especially in the field of complex physical systems. However, such techniques meet serious difficulties. On the one hand, these are natural limitations connected with the time scale and the size; on the other hand, these are the difficulties that result from the conception of used algorithms which do not always represent the real physical situation. In this paper we present the conception of a parallel computer which is based on the DLL model. That computer consists of a large number of processors which are working together and executing the algorithm based on the DLL model.

Our way of proceeding arises from the full conviction, documented in the majority of cases, that such natural representation exists between the parallelism in the physical world and the parallelism which can be realized in the artificial way by using the special parallel machine [1, 2]. The algorithm based on the Dynamic Lattice Liquid model works as an agent which connects these two worlds. Moreover, utilizing parallelism in this kind of problems seems particularly advisable, because even if the size of the considered system increases significantly, it will not lead to critical increase of the computations time. This time is practically determined only by the time of algorithm executing by the single processor.

II. SIMULATION OF THE COMPLEX PHYSICAL SYSTEM

In the area of the complex physical system, such as macromolecular liquids, one can distinguish two fundamental kinds of the simulation methods: molecular dynamics and Monte Carlo. Molecular dynamics is based on the numerical solving of classical equations of the motion in the relation to a molecular object. This method is implemented by moving all the molecular objects with small steps according to the forces calculated from the modelled interatomic forces. Molecular dynamics is very well applicable for parallel treatment [3-5]. However, when one processor represents each molecular object, calculation of the forces requires a large number of the communications between the processors. Connections between the processors are usually the limiting feature of the parallel

computers. This disadvantage does not exist in the method based on the locality of movement, which are realized in the discrete lattice model like Boltzman lattice gas [6] or in Monte Carlo lattice schemes like DLL.

Usually dynamics realized by Monte Carlo methods corresponds to diffusive dynamics, which does not conserve a momentum. This means that the dynamics allows performing the local steps in the space of macromolecular conformation. Even if these local steps do not follow the same distribution as the real dynamics, the long-range motion is consistent with the real dynamics where diffusive motion dominates over the system behaviour. Such model is reasonable in the dissipative fluid environment of macromolecules when we are interested in describing large changes in the macromolecular system conformation, or the diffusion of the system as a whole. For the sake of that, diffusive dynamics does not include correlations in the motion of molecular objects arising from the motion of the medium one can observe that this corresponds to the Rouse relaxation [7]. Correlation between the molecular objects and the medium are included approximately in Zimm relaxation [7] and first Monte Carlo scheme for this type of dynamics is the scheme based on the DLL model.

III. DLL MODEL

There are many algorithms for the simulation of the static and dynamic properties of the macromolecular systems on a lattice, but only a few of them properly reflect the dynamics over a broad time range and can work with the density factor $\rho = 1$, which means that each lattice site is occupied by one monomer unit. The DLL model can work with density factor $\rho = 1$ and become the basis for the first parallel algorithm, which takes into the account coincidences of elementary molecular movements resulting in the local cooperative structural transformations.

The macromolecular systems like polymer solution, melts, gels, glass or crystals exhibit a complex structure often extended over the various size scales, and a complex dynamic behaviour involving relaxation which extends over a broad time range. In addition, this kind of systems usually appears in a condensed state which makes their motion restricted and correlated with the motion of surrounding molecules. In spite of that, it is usually possible to well determine the distinguishable time scales. On the short time scale, as it is in the simple liquids, the monomeric fragments of the macromolecules oscillate around some quasi-fixed positions being temporarily "caged" by the neighbours. The more extensive translation

motions of the molecules in liquids or monomers in the polymer take place on a much longer time scale, due to the breaking of cages by the cooperative processes. Therefore, trajectories of the molecules consist of both the oscillatory components and of an occasional longer-range translational movement between subsequent quasi-fixed states, as schematically illustrated in Fig. 1.
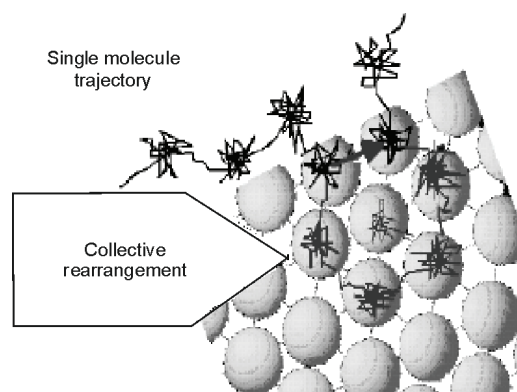

Fig. 1. The rearrangement of beads occupying all the sites in a lattice consists in the cooperative displacement of the system elements along the closed trajectory within which each element replaces one nearest neighbour

This scheme of local motion in a molecular liquid is commonly accepted and partially documented by the computer simulations of the dense Lennard-Jones systems [1, 2]. However, it is not clear under which condition the single diffusion steps can occur. In the DLL model a mechanism has been proposed, in which a single diffusion step is based on the cooperative movements which are believed to play an important role in the global relaxation process. This simple lattice model can be regarded as an alternative to the vacancy diffusion mechanism in liquids. The model considers the cooperativity of the molecular rearrangements related to their translational motions. The cooperativity in this model is precisely defined as resulting from the continuity condition of the mass transport, applied on the microscopic scale. It has been shown that with such model of molecular rearrangements various types of dynamic behaviour of the simple liquids and of polymers can be reproduced [8-12].

IV. DLL MODEL SIMULATION CONDITIONS AND IMPLEMENTATION DETAILS

The DLL model is based on a lattice structure. At first the model can be described for simple liquids consisting of beads representing small molecules. Positions of the mole-

cules are regarded as coinciding with the lattice sites. Due to the dense packing of the molecules, all the lattice sites are assumed to be occupied. It is also assumed that systems have some excess volume, therefore molecules have enough space to vibrate around their position defined by the lattice sites. However, they can not move easily over the larger distance because all the neighbouring lattice sites are occupied. The vibrations are assumed to take place with mean frequency $\nu_v = 1/\tau_v$, related to the short time scale (τ_v). Each of large enough displacement of a molecule from the mean position defined by the lattice is considered as an attempt to move to a neighbouring lattice site. To simplify, the attempts are assumed to take place only along the coordination lines, but are independent and randomly distributed among q direction, where q is the lattice co-ordination number. Most of the attempts remain unsuccessful, because we assume that all the time the system remains quasi continuous, i.e. no holes of molecular sizes are generated and multiple occupations of lattice sites are excluded, which is defined as excluded volume condition. The continuity condition of the system for the vector field of attempted displacement r can be written as follows: $\nabla J + \partial \rho / \partial t = \varepsilon$ with $\varepsilon \to 0$, where J is the current of displacing molecules and ρ is the density and should not deviate considerably from one if all lattice sites remain occupied, ε is introduced as a small number allowing density fluctuations but excluding generation of holes. Consequently, most of the attempted displacements have to be compensated by a return to the initial position within the period τ_v. Only those attempts can be successful which coincide in such a way that along a path including more than two molecules the sum of displacements is close to zero. In the system considered, only paths in the form of closed loops can satisfy this condition, as illustrated in Fig. 1. The probability to find an element taking part in such coincidences will determine the probability of longer range rearrangements and the slower time scale (τ_α). There is a possibility that many rearrangements appear simultaneously in the simulated system but intersections of loop contours are excluded because within a single time step an element can displace only in one direction and only to one neighbouring lattice site.

The model described above has been implemented as a dynamic simulation algorithm for polymers in a solvent on lattice in two and three dimensions. A system of beads (for simplicity) on the triangular lattice is considered. The beads are considered as representing solvent molecules or a unit of polymer chains and occupy all lattice sites. Linear polymers have as usual been represented as chains of beads connected by non-breakable bonds. It is assumed that the beads vibrating with certain frequency around the lattice

sites attempt periodically to change their position towards the nearest neighbouring sites. The attempts are represented by the field of randomly chosen unit vectors which are assigned to beads and point directions of attempted motions. An example of such assignment of attempted directions of motion is shown in Fig. 2, for a system representing polymer solution on the triangular lattice. All beads which do not contribute to correlated sequences (loops) satisfying the continuity condition are non-moveable.
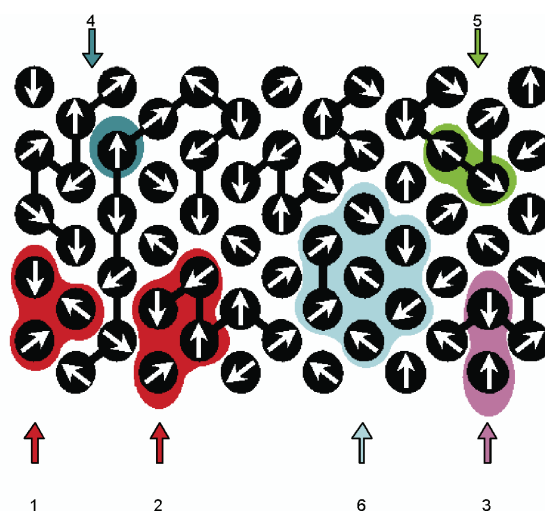

Fig. 2. The example of the vector field representing attempts of molecule movements in the direction of the neighbouring lattice point in the DLL model: cooperative motions (1) and (2); non correlated sequences: two beads try to move in opposite directions (3), bead leaves empty space (4), break of bonds in the polymer chain (5), particle would jump through a bond (6)

After setting all vectors of non-successful attempts to zero, only vectors contributing to a number of closed loops remain and constitute traces which are considered as paths of possible rearrangements (cases (1), (2) in Fig. 2). If the system is considered as athermal, all possible rearrangements found are performed by shifting beads along the closed loop traces, each bead to the neighbouring lattice site. Thus we can distinguish the following steps in this algorithm: (i) generation of the vector field representing attempts of moves, (ii) finding successful attempts and (iii) replacing beads within closed loop paths.

The procedure described above is considered as a single time step taking place within the time scale τ_v. The procedure is exactly repeated in subsequent time steps always with a new set of randomly chosen directions of attempts.

The system treated in this way can be regarded as provided with the dynamics consisting of local vibrations and occasional diffusion steps resulting from coincidence

of attempts of the neighbouring elements to displace beyond the occupied positions. Within longer time interval, this kind of dynamics leads to displacements of individual beads and chains along the random walk trajectories with steps distributed randomly in time.

The DLL model implemented on sequential machines was used for simulation of dynamic behaviour of several systems: interdiffusion in polymer solution [9], reaction diffusion front problem [10, 11] and aggregation processes [12]. In all cases we obtained results which were in good agreement with theory and in some cases using the DLL algorithm allowed to obtain new results. Figure 3 shows results which were obtained in case of the 2D polymer solution with explicit presence of solvent. This figure shows mean square displacement of the chain centre of mass, monomer and solvent. Presented dynamic characteristics indicate the possibility of direct investigation of dynamic properties of solvent particle in polymer solution in short and long time what is difficult to obtain by using other methods.
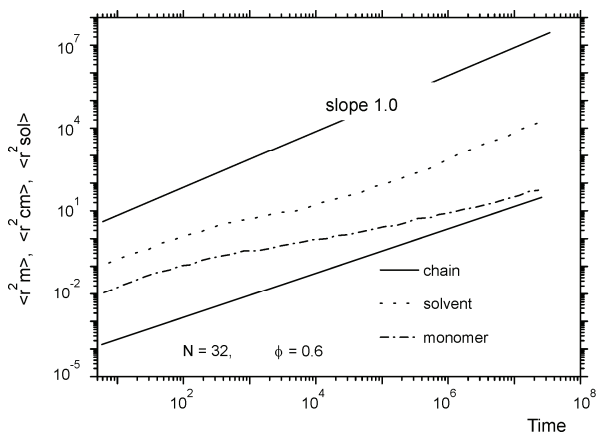

Fig. 3. Mean square displacements of the chain centre of mass (solid curve), monomer (dashed curve), and solvent (dotted curve) versus time for polymer concentration of 0.6. A slope of 1.0 indicated by the thin solid line

Implementation of the DLL model on sequential machines is very useful but if we want to solve real physical or chemical problems on a big scale we need special kind of hardware which will execute the DLL algorithm.

V. DLL MACHINE

The idea of the DLL machine described by us is based on the Dynamic Liquid Lattice algorithm. That is to say that in the athermal case the DLL machine can be considered as a kind of cellular automatic machine, whereas in the case when interactions are taken into consideration this automatic machine gets the features of the systolic computer.

The DLL algorithm description allows for a direct definition of the parallel machine. This machine consists of identical elementary cells which are placed in the lattice points of the face cubic centred (fcc) network with the coordination number of 12 (see Fig. 4).
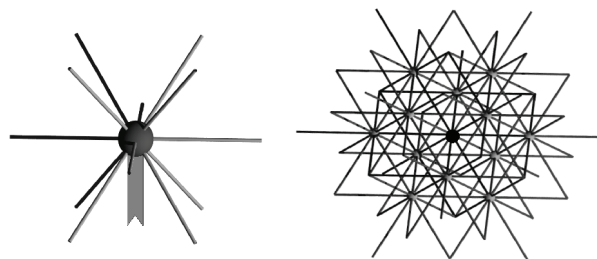

Fig. 4. Single DLL cell: equipped with signal bus and twelve channels for connection with neighbours (left side); with neighbours placed in face cubic centred lattice point (right side)

In order to provide correct communication, the cell is equipped with twelve bidirectional transmitting channels. Every cell contains a single element of the simulated system (e.g. a solvent molecule or monomeric unit of a polymer chain) which is represented by the internal combination of logical states. Apart from the kind of element, the time necessary for determination of its new position is the same for each cell. This time interval determines one work cycle for a cell, and at the same time for the whole DLL machine. Duration of the cycle is defined by the external synchronizer.

In order to control the simulation process and the data acquisition, the cells have to be equipped with the systems which allow communication with an external synchronizer. In addition to the twelve channels which are connecting the neighbouring cells, each cell has to be equipped with the control signal bus (cf. Fig. 4).

The computation time executed by the cell is insignificantly dependent on the relationship between the nearest neighbours. It is connected with the technical ability and the physical size of the machine. If the DLL machine and the sequential machine corresponding to it (synchronized with the same clock) had executed the DLL algorithm, the duration of one computation cycle for DLL machine would be approximately equal to the duration of one computation cycle of sequential machine divided by the number of lattice points (e.g. DLL machine equipped with 10^6 elements will be six orders of magnitude faster than its sequential equivalent).

Because the cycle of the single cell is practically equal to the whole DLL machine cycle, the internal construction of the single cell is of crucial importance for the whole machine performance. The cell work cycle corresponding to the DLL algorithm is shown in Fig. 5.

```
1.  Start
2.  Select randomly of movement direction (one of 12);
3.  Readout of the information from neighbouring cells about elements they contain;
4.  Execute of the MC step (energy test);
5.      If the creation of the new bond and/or break existing bond are possible than
6.      Create of new bond or break existing bond;
7.      While status of movements is not described do
8.          Searching for cooperative motions
9.              If Broken bonds than
10.                 Elimination of movement attempts which lead to break of bond;
11.             else
12.             Status of movements is described;
13.     end while
14.     If status of movements = success than
15.         Displacement of element in direction of sampled neighbouring cell and
            simultaneous adoption of element incorporated in neighbouring cell;
16.     else
17.     Do nothing;
18. end
```

Fig. 5. Algorithm of DLL cell work cycle

The cell has to execute the following actions during the single work cycle:
- checking if the change of element position is possible on the basis of the properties and the relationship between the element placed in the cell and the elements incorporated into the all twelve neighbours,
- realization of the potential exchange of the element in the DLL cell,
- sending the information to the control unit of the DLL machine about termination of the full cycle.

In order to provide the control of the simulation process and the data acquisition by the external synchronizer, the cells have to be able to do the following actions:
- data acquisition from the external synchronizer,
- data sending to the external synchronizer,
- readiness request for executing the next simulation cycle.

A building block of such DLL system will be a single chip that integrates multiple logical units, where each unit represents a single cell in the model, and the connectivity and the communication logic corresponding to the DLL architecture. This should allow for building a huge system by replicating such a chip. A prototype of such system should allow estimations concerning finite size limits of the DLL machine and its computational efficiency, as well as the technological problems related to energy supply, thermal stabilization and spatial requirements, etc.

Using the technique based on field programmable gate arrays (FPGA), a prototype machine was realized. This machine is shown in Fig. 6. It is hardware realization of the DLL algorithm. The prototype contains 216 elementary DLL cells. It is composed of seven PCBs (printed circuit boards). Six of them contain three XC3S4000 FPGAs on each board. In each FPGA an 2×6 array of DLL cells is implemented. Three FPGAs on the board constitute two dimensional arrays containing 36 (6×6) DLL cells. The seventh board, containing two XC2S150E FPGAs, controls the simulation being performed by six boards constituting three dimensional arrays containing 216 ($6 \times 6 \times 6$) DLL cells. Additionally, this seventh board is responsible for the communication of the whole system with a PC computer in order to initialize the simulation and to collect the simulation results. The execution of virtually unbounded simulations requires an implementation of cyclic boundary conditions. Therefore the construction of the machine is based on the cylinder shape in order to reduce the delay differences. The distance between each of the two subsequent boards is equal and therefore the communication latency for all elements of the whole system is more or less the same.
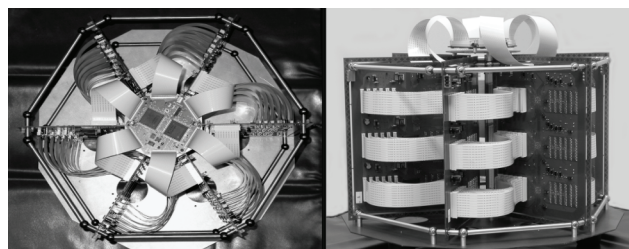

Fig. 6. Top (on the left hand side) and front (on the right hand side) view of the prototype

Obviously, the developed model is useless as far as the realistic physical or chemical simulations are concerned. In practical applications the DLL array should have about 10^6 ($100 \times 100 \times 100$) elements. Anyway, the existing model allows for verification of the ideas and proposed algorithmic solutions.

VI. CONCLUSION

The Dynamic Lattice Liquid model seems to be at the moment the only model in which the simultaneous elementary translations are related to the cooperative local structural transformations. The model provides a microscopic picture of cooperative molecular rearrangements (cooperative loops) observing the system continuity, the excluded volume conditions and dense packing of the molecules. This model is used as the basis of the architecture of the
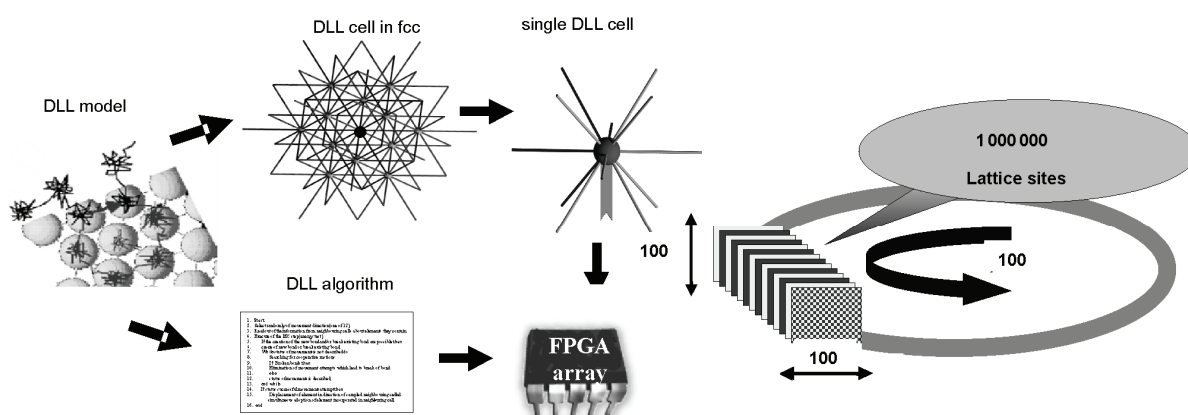
Fig. 7. High power, massive parallel DLL computer

dedicated cellular parallel computer. A schematic illustration of the pathway between the DLL model and the computing system is shown in Fig. 7.

The already tested and potential applications of the DLL machine concern all fields of the science and technology in which atomistic, molecular, macromolecular and supramolecular, structural and dynamic consideration are of importance and interest. This includes the following areas given here as examples:

- Chemistry – the modelling of synthetic processes leading to complex macromolecular systems (e.g. simulation of the polymerization processes in space allows the realistic consideration of physical effects on reaction kinetics and products),
- Materials technology – e.g.: (i) effects of various macromolecular architectures on self-organization, resulting supramolecular structures and dynamics, and consequently on material properties; (ii) behaviour of multi-component materials; (iii) modelling of the molecular behaviour under the various processing conditions,
- Surface and interfacial effects – modelling of the thin layers, macromolecules in confined space, pattern recognition, adsorption at surfaces, etc.,
- Nanoscale effects in all kinds of the soft matter – e.g. molecular effects in nanocomposites,
- Bio-macromolecules – behaviour in various environments and various aggregation states,
- Bio-relevant systems – membranes, functionalized polymers, molecular motors, intermolecular recognition,
- Magnetic properties – critical phenomena, ordering, vitrification,
- Molecular and macromolecular aggregation – clustering, heterogeneous media, microphase separation, pattern formation,
- Molecular machines – directed diffusion, directed transport of molecules, catalytic functions,
- Molecular modelling of devices – e.g. membranes, filters, batteries, etc.

Acknowledgments

References

[1] B.J. Alder, T.E. Wainwright, *Studies of molecular dynamics I: general method*. J. Chem. Phys. 31, 459 (1959).

[2] R. Zangi, S. Rice, *Coopertive Dynamics it Two Dimension.* Phys. Rev. Letter 93, 35502 (2004).

[3] B.G. Fitch, A. Rayshubsky, M. Elftheriou, T.J.C. Ward, M.E. Giampapa, M.C. Pitman, J.W. Pitera, W.C. Swope, R.S. Germain. *Blue Matter: Scalling of N-body simulations to one atom per node*. IBM J. Res & Dev. 52, 145 (2008).

[4] J. Makino, *Grape project special-purpose computers for many-body simulation.* Comp. Phys. Communication 139, 45-56 (2001).

[5] R. Scrofano, M.B. Gokhale, F. Trouw, V.K. Prasanna, *Accelerating Molecular Dynamics Simulation with Reconfigurable Computers*. IEEE Transaction on Parallel and Distributed System 19, 765 (2008).

[6] B. Chopard, M. Droz, *Cellular Automata Modeling of Physical Systems*. Cambridge University Press (1998).

[7] I. Teraoka, *Polymer solution – an introduction to physical properties.* Wiley-Interscience, (2002).

[8] P. Polanowski, T. Pakula, *Studies of mobility, interdiffusion and self-diffusion in two-component mixtures using the dynamic lattice liquid model*. J. Chem. Phys. 118, 11139-11146 (2003)

[9] P. Polanowski, T. Pakula, *Simulation of polymer-polymer inter-diffusion using the dynamic lattice liquid model*. J. Chem. Phys. 120, 6306 (2004).

[10] P. Polanowski, Z. Koza, *Reaction – diffusion fronts in system with concentration dependent diffusivities*. Phys. Rev. E 74, 36103 (2006).

[11] P. Polanowski, M. Kozanecki, T. Pakuła, *Non-monono-tonical behaviour of reaction diffusion front width in frame*

of dynamic lattice liquid (DLL) model. Materials Science-Poland 24, 453 (2006).

[12] P. Polanowski, *Parallel simulation of random fractal growth using Dynamics Lattice Liquid (DLL) model*. J. Non. Cryst. Sol. 353, 4575 (2007).

PIOTR POLANOWSKI received his M.Sc degree in physics in 1987 from the University of Łódź. He earned the Ph.D. in Chemistry from the Technical University of Łódź in 2002. Currently he works in the Technical University of Łódz in the Department of Molecular Physics. His fields of interest covers simulations of complex molecular and macromolecular systems with saving proper dynamic behaviour, parallel computing (hardware and software) in application to complex molecular systems, simulation software development.

JAROSLAW JUNG received his M.Sc. degree in physics in 1987 from the University of Łódź and in electronics in 1990 from the Technical University of Łódź. He earned the Ph.D. in Chemistry from the Technical University of Łódź in 2001. Currently he works in the Technical University of Łódź in the Department of Molecular Physics. His fields of interest covers research of organic semiconductors, construction of equipments for measurement of photoconduction in organic semiconductors and construction of parallel computer designed to simulate the dynamics of macromolecular systems.

RAFAŁ KIEŁBIK has graduated his M.Sc. in electronics in 1998 at Technical University of Łódź. In 1999 he spent four months on scholarship in Barcelona, at Universitat Politècnica de Catalunya. In 2005 he received Ph.D. degree from Technical University of Łódź and in 2006 from Universitat Politècnica de Catalunya. He currently works in Department of Microelectronics and Computer Science at Technical University of Łódź. He is interested in programmable devices, hardware description languages and automatic synthesis of electronic devices.