# Molecular Dynamics Simulations on a Systolic Ring of Transputers

K. Boehncke, H. Heller, H. Grubmüller, and K. Schulten*
Beckman Institute and Department of Physics, University of Illinois,
405 N. Mathews Avenue, Urbana, IL 61801 U.S.A.

**Abstract**

For the purpose of molecular dynamics simulations, a parallel computer based on a systolic ring architecture, with Transputers as computational units, has been built and programmed in OCCAM II. The design is very compact and achieves high reliability even on continuous duty cycles of several months. A parity check logic ensures the correctness of the data stored in the external RAM. Our program for simulation of very large molecules (up to 32000 atoms) is compatible with the programs CHARMM and X-PLOR. It increases its computational throughput nearly linearly with the number of computational nodes. Exploiting the MIMD structure of our Transputer network, we have added to our program the capability to compute explicit hydrogen bonds. One node is now entirely devoted to this task. Benchmarks comparing simulations of several molecules –ranging from 66 to 12637 atoms– on the Transputer system with those on conventional machines are provided. The results demonstrate that the software and hardware developed provide an extremely cost-effective means for simulations of molecules. We have simulated the photosynthetic reaction center, a protein complex of 12637 atoms, as well as drug-DNA complexes, on our computer.

## 1. What is Molecular Dynamics ?

A principal aim of molecular biology is to understand the relationship between structure and function concerning the molecules of life. Static structures of several of these important molecules have been resolved using experimental techniques such as X-ray crystallography and nuclear magnetic resonance. For a long time, it was assumed that the static structure along with knowledge of the chemical bonds and the typical atomic charges involved would be sufficient for a full explanation of a molecule's function. In the recent decade, however, it became increasingly clear that the *motion* of the individual atoms plays a crucial role in determining many aspects of molecular properties and interactions. The details of these processes are often very difficult to measure experimentally, if at all. The information of interest can then only be obtained by computer simulations. Reviews of this exciting field, and examples of many applications, can be found in [1, 2].

Currently, many groups are developing approaches to allow an increasingly faithful representation of molecular dynamics in computer programs. In the near future, this will make it possible to leap beyond simple analysis of structure-function relationships and allow significant contributions to biology, biotechnology, and medicine by guiding the synthesis of new materials and drugs and predicting their respective effects. A longer-range

*To whom correspondence should be sent.

goal of computer simulations includes the prediction of 3-d protein structures from their respective amino acid sequences.

How are these simulations actually carried out? The exact time evolution of a system consisting of many atoms is described by the pertinent quantum-mechanical Schrödinger equations whose complexity –even with only small molecules– renders this description unfeasible for the task of dynamics simulations. Therefore, a classical approximation is used. In this approximation, atoms are represented as points of mass with a certain charge. The motion of each atom is then determined by summing up all forces acting on it and integrating Newton's equations of motion

$$\left.\begin{aligned} m_i \frac{d^2\mathbf{r}_i(t)}{dt^2} &= \mathbf{F}_i(t) \\ \mathbf{F}_i(t) &= -\nabla_i E(\mathbf{r}_1(t), \ldots, \mathbf{r}_N(t)) \quad . \end{aligned}\right\} \tag{1}$$

Here, $m_i$ and $\mathbf{r}_i$ are used to denote the mass and position of the $i$-th atom, respectively. $\nabla_i$ represents $(\frac{\partial}{\partial x_i}, \frac{\partial}{\partial y_i}, \frac{\partial}{\partial z_i})^T$, and $N$ is the total number of atoms in the system. $E$ is the empirical energy function, which is critically important for a realistic description of molecules. In the widely used molecular dynamics programs CHARMM [3] and X-PLOR [4], and in the program we have developed for our Transputer system, this energy function is of the form

$$E = \underbrace{E_{bond} + E_{angle} + E_{dihedral} + E_{improper}}_{E_{bonded}} + \underbrace{E_{el} + E_{vdW} + E_{hbond}}_{E_{nonbonded}} \quad , \tag{2}$$

where $E$ is defined as the total energy of the molecule. The individual contributions correspond to the different types of forces acting in the molecule. The first contribution $E_{bond}$ describes the high frequency vibrations along covalent bonds. The second contribution $E_{angle}$ represents the bending vibrations between two adjacent bonds, and the third contribution, $E_{dihedral}$, describes torsional motions around bonds. $E_{improper}$ relates to the motion of one atom relative to a plane defined by three other atoms. All of the previously mentioned energy contributions portray interactions due to a direct chemical bond between atoms. They are often grouped as so-called *bonded* interactions. The remaining three terms describe the *non-bonded* interactions between atoms that are not chemically bonded. $E_{el}$ accounts for the electrostatic (*Coulomb*) energy between the individual atomic charges, $E_{vdW}$ models the interaction of neighboring electron clouds, the *van der Waals*-interaction, and $E_{hbond}$ represents the hydrogen bonds.

The method most often used to integrate the Newtonian equations of motion (1) is the Verlet algorithm [5]. The position $\mathbf{r}_i(t + \Delta t)$ of atom $i$ at the instant $t + \Delta t$ is then determined according to the formula

$$\mathbf{r}_i(t + \Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \mathbf{F}_i(t)\frac{(\Delta t)^2}{m_i} \quad , \tag{3}$$

where $\mathbf{F}_i(t)$ as defined in Eq. 1 represents the sum of all forces acting on the $i$-th atom at time $t$. The time step $\Delta t$ is determined by the fastest degrees of freedom in the molecule. These are typically bond vibrations of the light hydrogen atoms, and the time step must be no larger than $0.5 \cdot 10^{-15}s$ to insure correct integration of Eq. 1. If the bond lengths of



the hydrogen atoms are fixed during the simulation using the SHAKE algorithm [6], the time step can be increased to $1 \cdot 10^{-15}s$.

Because of the many interactions that must be considered, and the small size of this time step, the simulations one would like to carry out are severely hampered by the availability of suitable computer resources. In fact, dynamics calculations up to now have been limited to short simulation periods of a few nanoseconds and to biopolymers of at most a few thousand atoms. Furthermore, real biological systems are usually embedded in an environment of water or lipids, and a realistic description of a molecule should include many of the surrounding atoms. All of these factors combined demand the fastest computers available to carry out molecular dynamics simulations.

While integrating the Verlet equations (3), most computer time is spent on the evaluation of the non-bonded interactions. For each integration step, $N(N-1)/2$ pair contributions must be calculated due to the long range nature of the Coulomb interaction. In the programs CHARMM and X-PLOR the corresponding prohibitive computational effort is avoided by introducing a cut-off for these interactions; i.e., it is assumed that they do not contribute much to the dynamics for pairs of atoms seperated by more than a specified distance. Other approaches for the efficient calculation of non-bonded pair contributions are the *Fast Multipole Algorithm* developed by Greengard and Rokhlin [7] and a hierarchical distance class approach suggested in [8]. The latter method was recently implemented in our program and is fully described in [9]. The idea is based on a spherical subdivision of interatomic distances into several distance classes. The relative motion of atoms that are close to one another greatly influences their contribution to the total electrostatic energy gradient, so that these force contributions must be calculated at each integration step. Relative motion of atoms further separated does not change their force contributions a great deal, so that the contribution of these atoms is stored and only updated periodically. This computational method speeds up computation by a factor of six without a significant loss of accuracy.

## 2. Hardware

In November 1987, when we began our efforts developing molecular dynamics algorithms for Transputers, to our knowledge no commercial hardware meeting our specifications (large, parity checked DRAM on each node, status indicator, compact design) was available. Therefore, we decided to design and build our own six-layer boards.

The hardware consists of a systolic ring of Transputer nodes (see Fig. 1), and is fully described in [10, 11, 9]. The main parts of each computational node are an INMOS T800 Transputer running on 20 MHz, 4 MByte of parity checked DRAM (100 ns access time), a parity checking logic and a status indicator. The latter signals the internal state of each node to the user by means of three color-coded LED's (one each for a set error flag, a parity error, and access of the external DRAM). Deadlock situations and an uneven distribution of the workload can be easily detected by observing the DRAM access indicator.

Six of these nodes fit onto one double eurocard (160 mm by 233 mm) and ten of these cards fit into one 19" chassis, totalling 60 nodes which for molecular dynamics calculations match the computational power of a Cray X-MP processor.

All link connections, determining the topology of the network (here a systolic ring), are provided by the backplane of the chassis.

Parity checking is important, since we want to expand our Transputer system of currently 18 nodes to more than 200 nodes. With such a system, a parity error is likely to occur every other month. Assuming a runtime of several months for the molecular dynamics code, this probability is unacceptably high. Together with an automatic restart mechanism in the software, the parity checking allows sufficiently long, unsupervised dynamics calculations without the risk of corrupting data due to memory faults.

# 3. The Program

As explained in the introduction, the computationally most expensive task is the evaluation of Coulomb and van der Waals forces. Due to the long range character of these forces, each atom interacts with every other atom, leading to a total of $N(N-1)/2$ force computations for each integration step. Since this large number of interactions clearly dominates the computation time, our parallelization strategy is mainly determined by these interactions (referred to only as "Coulomb interactions"). However, all the other contributions to the energy function (Eq. 2) have to be calculated as well, which turned out to be complicated for the explicit hydrogen bonding term, as explained in Section 3.4.

## 3.1 The Topology

In our computational scheme, the atoms of the molecule are assigned to different computational nodes irrespective of the atoms' positions in space. All information about these atoms, such as mass, charge, type, and coordinates, will be stored in those respective nodes. The evaluation of the forces acting on these atoms will also take place at the Transputer of the node that the atoms are assigned to. In the following discussion, all atoms allocated to a specific Transputer will be referred to as the 'own' atoms of that Transputer, all other atoms are the 'external' atoms.

The computation of the Coulomb forces is now separated into two components

$$\mathbf{F}_i = \sum_{\substack{\text{'own'}\\ \text{atoms } j}} \frac{q_i q_j \mathbf{r}_{ij}}{4\pi\varepsilon r_{ij}^3} + \sum_{\substack{\text{'external'}\\ \text{atoms } k}} \frac{q_i q_k \mathbf{r}_{ik}}{4\pi\varepsilon r_{ik}^3} \quad . \tag{4}$$

$\mathbf{F}_i$ is the force which acts on atom $i$, $q_i$ is the partial charge of this atom, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is the vector joining the atoms at positions $\mathbf{r}_i$ and $\mathbf{r}_j$, and $\varepsilon$ is the dielectric constant. The first sum can be done by all nodes in parallel, since it involves only information of 'own' atoms $i$ and $j$, which is always known locally at each node. The second sum, however, involves 'own' atoms $i$ as well as all 'external' atoms $k$ and, therefore, requires some communication.

The systolic ring topology [12], which is depicted in Fig. 1, provides the communication channels needed. The network Transputers (T800), one of which can be entirely devoted to the calculation of hydrogen bonds (T800, light gray), and a B004-board (T414) that plugs into an IBM compatible PC-386 as host, are joined by a bidirectional Transputer link (two antiparallel, unidirectional OCCAM II channels).

Using this link, each Transputer can send a copy of the information it has about its 'own' atoms to its neighbor in clockwise direction. Now each Transputer can calculate that part



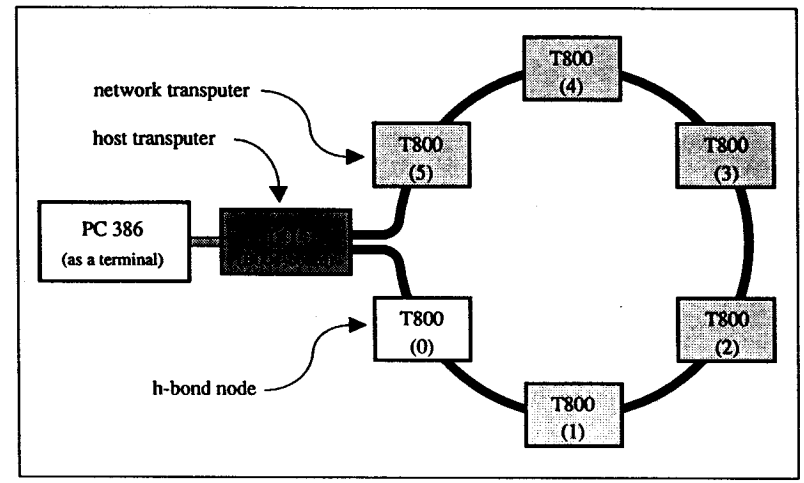


**Figure 1:** Schematic diagram of the systolic ring network used in our application.

of the sum in Eq. 4 concerning those 'external' atoms it just received from its neighbor. In the next step, the information about the 'external' atoms is again passed to the next neighbor in clockwise direction by each node. Knowing a new set of 'external' atoms, each node can calculate another part of the 'external' sum in Eq. 4. This process is repeated until all Coulomb interactions have been computed. By then, each node will have received its 'own' atoms back, which have circulated one full revolution on the ring. Using the total Coulomb force component, along with the bonded force contributions of its 'own' atoms, the entire force $\mathbf{F}_i$ acting on each atom is known, and all nodes can integrate one step according to Eq. 3.

By applying Newton's law $\mathbf{F}_{ki} = -\mathbf{F}_{ik}$ it is possible to cut the computation time to about one half, avoiding redundant calculations, but instead sending partial forces in counter-clockwise direction. One drawback of this method is that we loose –in the case of an even number of nodes– one node, which now remains idle throughout the simulation [10, 11]. Making use of the MIMD structure of the Transputer network, however, this node can be devoted to the calculation of hydrogen bonds (see Section 3.4) without significantly slowing down the calculations.

## 3.2 Process Structure of a T800 Node

The computational nodes need to carry out the following processes: (i) send coordinates around the ring clockwise, (ii) send forces along the ring counter-clockwise, (iii) collect forces that relate to 'own' atoms, and (iv) synchronize as far as is necessary for the integration step. The processes and their relationship are shown schematically in Fig. 2.

Incoming atom information is processed by the process *calculation*, i.e. pair interactions are computed, and passed (through a *buffer*) to the neighbor in clockwise direction. Through the channel *force.to* and the *multiplexer*, forces for other Transputers are sent in counter-clockwise direction. A *routing process* decides whether the forces are related

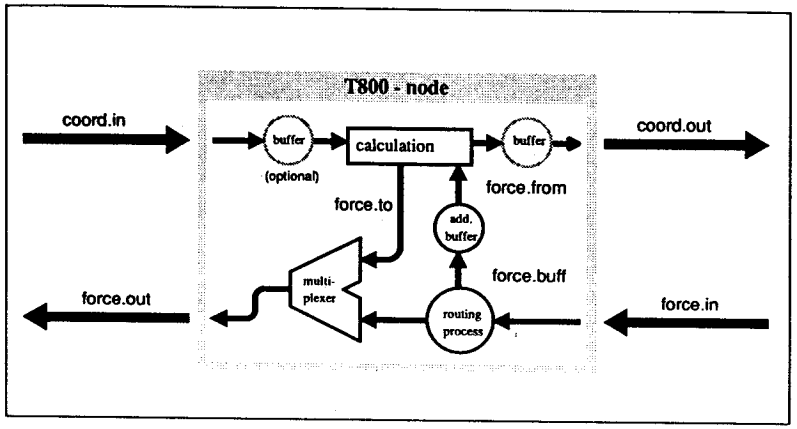


**Figure 2:** Process structure of the program running on the T800 nodes. Shown are the processes running in parallel. (From ref. [10]).

to 'own' atoms (they are then forwarded to the process *calculation*) or not (they are then passed to the *multiplexer* and sent to the neighbor in counter-clockwise direction).

## 3.3 Process Structure of the T414 Node

The T414 node does not take part in the computations, but it connects the network to the 'outside world', monitors possible error conditions (parity, error flag or user interrupt) and takes appropriate action, e.g. reboots the whole ring after a parity error detection. The internal process structure is shown in Fig. 3.

Since forces are of no concern here, a process *pass through* simply forwards them to the next Transputer. The process *master* transfers coordinate and energy information to the

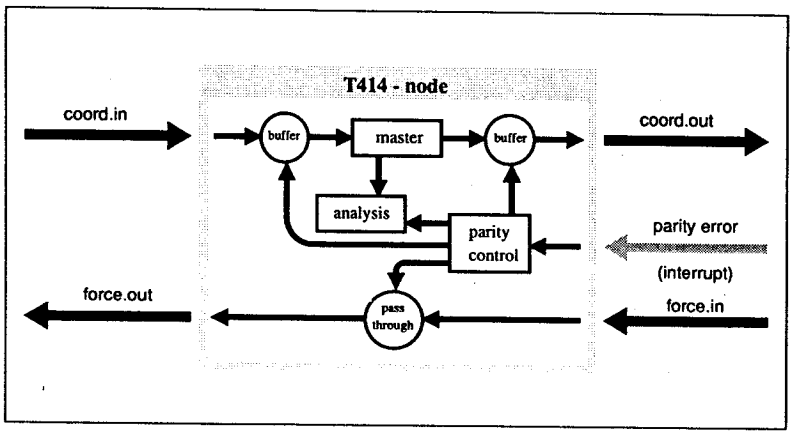


**Figure 3:** Process structure of the program running on the T414 host Transputer. Shown are the processes running in parallel. (From ref. [10]).



analysis process, if necessary. The process *analysis* writes that information on disk and prepares valid restart files. *Parity control* is connected to the event pin of the Transputer and responds to parity errors by immediately shutting down the calculation, discarding the erroneous data, and automatically initiating a reboot and a subsequent restart of the simulation using the latest restart file.

## 3.4 Adding the hydrogen bond node

The first versions of our molecular dynamics program [10, 11, 9] did not provide an explicit calculation of the hydrogen bond term in Eq. 2. Instead, hydrogen bond contributions to the energy function were implicitly included in the electrostatic energy by special partial charges on atoms involved in these bonds. In many molecules, especially those in which the structure depends heavily on hydrogen bonds (like the DNA double-helix), the force field parameters require an explicit calculation of the hydrogen bond energy.

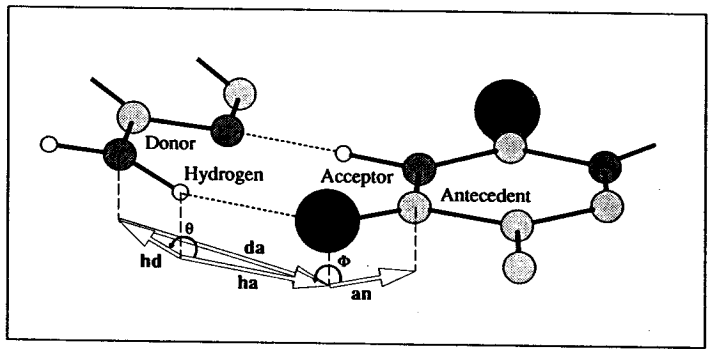


**Figure 4:** Schematic depiction of two typical hydrogen bonds found in a DNA molecule. The vectors and angles indicated are referenced in the text. $\theta$ is the angle between **hd** and **ha**, $\phi$ is the angle between **ha** and **an**.

The hydrogen bond interaction is the most complicated of all bonded energy terms in Eq. 2. This is due to the fact that such a bond usually consists of two pairs of atoms that are not bonded covalently. The participating atoms are called the donor, hydrogen, acceptor, and antecedent. A typical hydrogen bond is shown in Fig. 4, where the vectors and angles used in the energy calculation are also indicated. Note that, even though only one acceptor is shown for clarity, a hydrogen/donor pair may have several acceptors. The h-bond energy function that was implemented is based on and is completely compatible with the program X-PLOR. Basically, the hydrogen bond energy is a product of five terms

$$E_{hb}(\mathbf{d},\mathbf{h},\mathbf{a},\mathbf{n}) = E(r)\Theta(\theta)\Phi(\phi)\cdot\mathcal{S}_r\mathcal{S}_\theta \quad . \tag{5}$$

In detail, the first three terms are defined as follows (see Fig. 4 for a description of the vectors and angles mentioned):

$$E(r) = \frac{A}{r^6} - \frac{B}{r^4}; \quad \Theta(\theta) = \cos^4\theta; \quad \Phi(\phi) = \sin^2\phi \quad . \tag{6}$$

The parameters A and B are the control variables for the $r = |da|$ dependent potential. They are determined by the different atom types involved in each hydrogen bond. Since a cut-off was implemented in order to avoid calculating h-bonds between atoms that are farther apart than a specified distance and outside a certain angle range, it is important that switching functions be used. These switching functions, $\mathcal{S}_r$ and $\mathcal{S}_\theta$, insure that the energy contribution drops smoothly to zero at the cut-off distance or angle and, therefore, is continuous and differentiable. Force components acting on the participating atoms are calculated using the derivative of Eq. 5.

Two possibilities for implementing h-bond calculations were considered. First, it would have been possible to add the hydrogen bonding term to the energy calculations already carried out by each node. This alternative, however, was not implemented because the h-bond calculations could not easily be distributed over the network within the framework of our existing program, due to the long range, four-body character of this interaction. We therefore opted to exploit the MIMD structure of our Transputer network and to devote an entire T800 node exclusively to the task of calculating the h-bond contributions. This approach had the added benefit of using an otherwise idle node. Ideally, then, our algorithm would slow down the existing molecular dynamics program only minimally.

A complication arises because the h-bond node needs to know all nuclear coordinates of the possible h-bonding candidates at the beginning of an integration step. Calculation

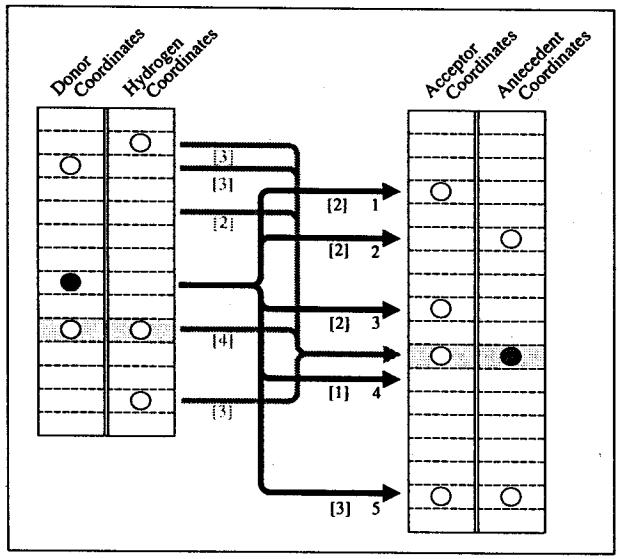


**Figure 5:** Schematic representation of the h-bond 'list'. This list is actually an array of pointers running from each donor to all its possible acceptors (black set of arrows). Once coordinates of a particular atom are known (depicted as a circle), the counters attached to the pointers (depicted as [...]) linking that atom's h-bonds are incremented. This is simple if a donor or hydrogen arrives at the list (black atom), as only those counters belonging to that donor-hydrogen pair must be updated (black arrows 1 through 5). Because there is no reverse mapping from acceptors to donors, an arriving acceptor or antecedent (grey atom) results in a scanning of the entire donor list, and all pertinent counters are incremented (grey arrows). As soon as coordinates of all atoms in a particular h-bond are known (light grey), they are sent to the calculation routine.



can therefore only begin when all coordinate packets have passed the node. By this time, however, all the other Transputers have already *finished* their integration step! It is clear that this approach would lead to an undesired asynchronous scenario with the other network nodes working and the h-bond node idle, followed by a working h-bond node and waiting other nodes.

To avoid such asynchronous calculations, we used the following method: The donor-acceptor distance of each individual h-bond is calculated. For bond lengths smaller than a previously specified cut-off distance, the hydrogen bond is added to a list comprising all h-bonds that are to be evaluated. The detailed implementation of this list is depicted in Fig. 5. The list is generated at the first integration step in the asynchronous manner described above, and is updated at later instances if necessary. In all ensuing steps, atoms passing the h-bond node are first compared to a list of all possible hydrogen bond participants. If the atom is found to participate, its type (donor, hydrogen, acceptor, or antecedent) is determined and, together with its coordinates, channeled to the h-bond calculation routine. There, the atom is read into those empty slots of the h-bond list pertaining to the bonds in which it participates. All bonds in the list have an assigned counter indicating how many atoms are already in their slots. When an atom is inserted into a particular slot, the counter for that h-bond is incremented. As soon as all atoms taking part in a hydrogen bond are known, their coordinates are passed to the calculation routine for the energy and forces. The h-bond forces acting on these four atoms are then immediately sent over the network to their respective Transputers. A schematic diagram of the functioning of the h-bond node is shown in Fig. 6.

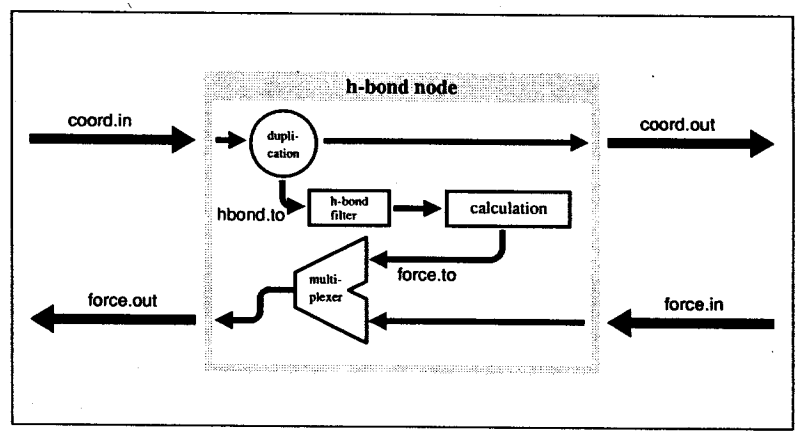


**Figure 6:** Process structure of the program running on the h-bond node. (The list update step is not shown.)

## 3.5 Compatibility with other programs

Compatibility with standard molecular dynamics packages was a key issue in the development of our program. The input files used (containing the nuclear coordinates, the structure information, and force field parameters) can easily be converted from the file format used by X-PLOR. The output files containing the trajectories of the individual

atoms can also be converted into X-PLOR format.[1] Recently, we have added the capability to read in X-PLOR restart information, in effect allowing the user the added freedom of transferring an ongoing computation between the two programs. This is often useful, because our Transputer program is designed as a fast dynamics number crunching application and does not yet include many of the advanced features found in X-PLOR such as stochastic boundaries, different heating methods, and analysis routines, to name just a few.

# 4. Benchmarks

Several benchmarks comparing the molecular dynamics performance of our Transputer system with other computers have been carried out. Different programs were used on the individual machines, the results are given in Table 1. It should be mentioned that many standard molecular dynamics packages like X-PLOR offer the possibility of using the SHAKE [6] algorithm (a feature not yet implemented in our Transputer dynamics code) to double the time step, in effect allowing the computers running this program to simulate at nearly twice the speed compared to the listings in Table 1.

| molecule | deca-alanine | segment of r.c. | reaction-center | DNA w. h-bonds |
|---|---|---|---|---|
| number of atoms | 66 | 3634 | 12637 | 538 |
| 1 T800 | 0.15 | 339.7 | — | — |
| 12 T800 | 0.06 | 31.6 | 386.9 | 1.55 |
| 24 T800 | 0.2 | 15.6 | 191.0 | 1.84 |
| SGI 220 GTX | n/a | n/a | — | 2.6 |
| Convex C-2 | n/a | n/a | — | 0.69 |
| Cray-2 | n/a | 4 | — | n/a |
| CM-2 | n/a | 2 | 23.0 | — |

**Table 1:** Benchmark results indicating the CPU time (in seconds) required for one integration step. No cut-off was used in any of the simulations. Different programs were used on the respective machines, all were properly optimized. The Transputer program (OCCAM II) ran on the T800 systolic ring, X-PLOR [4] (FORTRAN) was used on the Unix machines, and MD [8] (C-Source) was adapted for the Connection Machine CM-2. The symbol 'n/a' indicates that a particular benchmark was not carried out, and the symbol '—' indicates a benchmark that is not currently feasible because of technical reasons, such as memory requirements and/or program structure.

We can note, however, that the Transputer system does very well when large numbers of atoms are involved. In that case, the computational throughput increases nearly linearly with the number of nodes. Smaller molecules, like deca-alanine and the DNA segment used in the simulations, fare less well because of the increased communications overhead necessary when large numbers of Transputers are used.

[1] This feature was implemented by Markus Tesch.



# 5. Applications

A main subject of our interest is the photosynthetic reaction center of *Rhodopseudomonas viridis*. This important protein complex, encompassing 12637 atoms, acts like a biological "photodiode", in effect converting sunlight into an electrical potential. The quantum efficiency of this process is extremely high, and not yet fully understood. We hope to elucidate some of the underlying reasons by using molecular dynamics techniques. The X-ray structure of this molecule, provided by Deisenhofer and Michel [13], was used as a starting point. X-ray structures are normally minimized using the energy function Eq. 2 and a type of conjugate gradient minimization in order to remove artifacts that may exist in the crystal structure. These artifacts are due to the fact that resolution of X-ray structures is usually limited and the coordinates are localized at average positions that do not necessarily coincide with the preferred minima of Eq. 2. A new minimization technique adopted for our Transputer system was used [10], employing friction-controlled molecular dynamics with very small time steps. When the kinetic energy of the system corresponded to a temperature of $300K$, the procedure was modified, ending with a friction-free dynamics run. Simulations with the resulting structure are presently under way. All simulations carried out account for Coulomb forces exactly; i.e., no cut-off was used.

Another research area we are interested in involves drug-DNA interactions. This is a joint project with A.H.-J. Wang et al. Wang and coworkers have provided us with the X-ray coordinates of a short B-DNA segment to which an anti-tumor drug (distamycin) is attached [14]. The segment is depicted in Fig. 7, where it is shown with and without the drug, which binds to a specific base-pair sequence in the so-called *minor groove* of the DNA.

The binding mechanism is not fully understood. NMR experiments [15, 16] indicate a

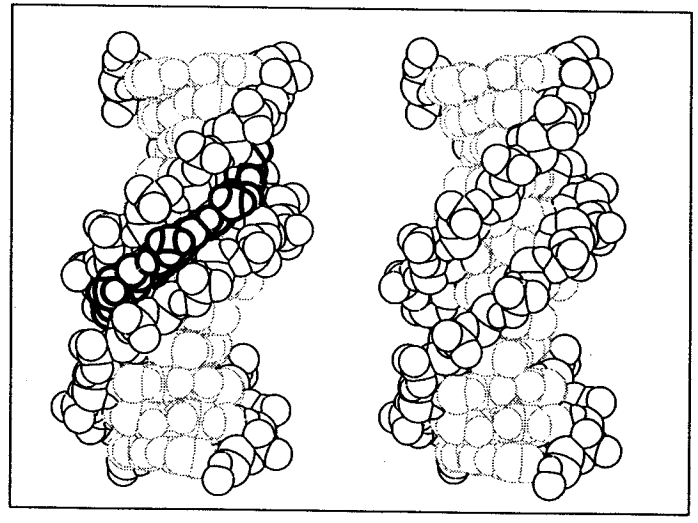

**Figure 7:** B-DNA segment consisting of 12 base-pairs with the drug distamycin bound to the minor groove (left) and without the drug (right).

dynamic flip-flopping of the drug in the minor groove. The time spans suggested by these experiments lie beyond the scope of current simulation capabilities using even the fastest computers. However, much can also be learned by shorter dynamics calculations. In a sense, one can 'experiment' with the system and move the drug to different locations, subsequently analysing the interaction energies between it and the DNA strand, possibly determining preferred binding sites. In the long run, such research could lead to an improved design of drugs.

## 6. Acknowledgements

We would like to thank Markus Tesch for many helpful discussions in the preparation of this paper. This work was supported in part by the National Center for Supercomputing Applications (NCSA) at the University of Illinois in Urbana-Champaign.

## References

[1] J. A. McCammon and S. C. Harvey. *Dynamics of proteins and nucleic acids.* Cambridge University Press, Cambridge, 1987.

[2] Ch. L. Brooks, M. Karplus, and B. M. Pettitt. *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics.* John Wiley, New York, 1988.

[3] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. *Journal of Computational Chemistry*, 4(2):187–217, 1983.

[4] A. T. Brünger. *X-PLOR.* Department of Molecular Biophysics and Biochemistry, Yale University, 260 Whitney Avenue, P.O. Box 6666, New Haven, CT 06511, 1988.

[5] L. Verlet. *Physical Review*, 159(1):98–103,1967.

[6] W. F. van Gunsteren and H. J. C. Berendsen. *Molec. Phys.*, 34:1311, 1977.

[7] L. Greengard and V. Rokhlin. Research Report 602, Yale Dept. of Computer Science, 1988.

[8] A. Windemuth. Master's thesis, Technische Universität München,1988.

[9] H. Grubmüller. Master's thesis, Technische Universität München, 1989.

[10] H. Heller, H. Grubmüller, and K. Schulten. *Molecular Simulation*, (in press). A major part of the present paper reviews material in this reference.

[11] H. Heller. Master's thesis, Technische Universität München, 1988.

[12] W. D. Hillis and Barnes. *Nature*, 326:27–30, 1987.

[13] J. Deisenhofer and H. Michel. *Science*, 245:1463 - 1473, 1989.

[14] M. Coll, C.A. Frederick, A. H.-J. Wang, and A. Rich. *PNAS, USA*, 84:8385 - 8389, 1987.

[15] D.J. Patel. *Eur. J. Biochem.*, 99:369 - 379, 1979.

[16] R.E. Klevit, D.E. Wemmer, and B.R. Reid. *Biochemistry*, 25:3296 - 3303, 1986.