Journal of Statistical Physics, Vol. 34, Nos. 5/6, 1984

Special Purpose Computers in Physics

H. J. Hilhorst,[1] A. F. Bakker,[1] C. Bruin,[1] A. Compagner,[1] and A. Hoogland[1]

Received September 30, 1983

This talk describes a new approach for large-scale computational problems which is particularly effective when a relatively simple algorithm is used. We demonstrate that it is possible to design and construct, at modest cost, special purpose computers for various classes of problems. By exploiting the principles of pipelining and parallel processing, and by adapting the hardware design to the specific structure of a particular algorithm, one can obtain a device which is as fast as or faster than general-purpose commercial supercomputers. The user of a such a processor has the double advantage of its speed and of its continuous availability for the particular problem for which it was constructed. In statistical mechanics special purpose computers have been built recently (i) for Monte Carlo simulation of the Ising model, and (ii) for the molecular dynamics of classical many-particle systems with short-range interactions. The design and performance of these machines are discussed and compared to those of commercial computers.

KEY WORDS: Special purpose computer; Monte Carlo processor for Ising model; molecular dynamics processor.

1. INTRODUCTION

Many branches of scientific research have become inseparably linked to the developments of computer technology.[1] This is particularly true in various areas of physics, as demonstrated in a number of excellent recent review articles.[2] In many cases numerical results have triggered analytic understanding and have led to the formation of new theoretical concepts. For several decades now, the increase of computing power has motivated physicists to study questions of increasing complexity. The fact remains

[1] Laboratorium voor Technische Natuurkunde, Postbus 5046, 2600 GA, Delft, The Netherlands.

0022-4715/84/0300-0987$03.50/0

that the computational power available today in universities and research centers is insufficient to attack a number of intriguing but difficult problems. It is therefore worthwhile to examine how, for problems requiring massive computation, one can exploit all the possibilities offered by modern technology.

Commercially available computing equipment varies considerably in architecture. For the purpose of this talk the following distinctions are useful. There are first of all the *general purpose* computers. These have been designed and constructed to deal with any problem for which a solution algorithm can be formulated. The architecture, and therefore the execution rate, of a general purpose computer is not optimized for any particular problem.

Secondly, there are what one could call the *operation oriented* computers. The prime example of these is the array processor, normally used in combination with a general purpose computer. Its architecture is such that all floating-point operations involving matrix algebra are executed at a very high speed. For a given problem one profits from this high execution rate only to the extent that one can formulate a solution algorithm in terms of matrix and vector operations on long vectors. A small part of an algorithm which is not vectorizable may become responsible for almost the entire execution time. The two types of architecture can coexist: modern supercomputers incorporate the key elements of both.

Computers of a third kind are the subject of this talk. These are the *special purpose computers*, which are *problem oriented*. Again, a special purpose computer will most often be used in combination with a general purpose computer. The idea is to take a particular problem as one's starting point, to choose the best algorithm for it, and to design and construct a hardware device—called a special purpose computer (SPC)—that will carry out that algorithm. There is no such thing as a program: it is replaced by the wiring of the hardware. If a more flexible use of the device is desired, one may include microprogrammable microprocessors in the design. In laying out the logical design one is free to arrange for parallel processing and pipelining wherever the algorithm allows for it, and to make use of any tricks which, in the given specific context, will increase the speed of execution.

In what follows we shall see how this basic idea can be fruitfully put into practice. We shall describe the design and performance of a number of laboratory-built SPCs each constructed to treat one single predefined problem of physics. Each of these SPCs is a one-of-a-kind computer: it is an example of the highest possible degree of specialization. While in our discussion we shall primarily have such highly specialized devices in mind,

it is of course possible to think of more "general purpose" SPCs, with a range of applications extending to certain classes of problems. One such machine is the commercially available ICL DAP, on whose features we shall also have the occasion to comment. This talk will not be addressed to the computer scientist but to the interested physicist with only a vague notion of how a computer works.

2. SPECIAL PURPOSE COMPUTERS: WHY AND WHEN?

The advantages of a special purpose computer are twofold. First, by a proper design one can obtain at modest cost an SPC which in speed equals or surpasses the fastest commercial supercomputers in executing the algorithm for which it was constructed. Secondly, once constructed, an SPC is available full time to its user and that at almost no additional cost. Since the design and construction of an SPC requires time and labor, it is natural to ask under what circumstances it pays to build an SPC.

It would seem that the conditions are the following. First, the problem must be such that any progress on it would require long or prohibitively costly calculations on a general purpose computer with or without an array processor attached to it. Secondly, the algorithm for solving the problem must be well established and well understood: once an SPC has been constructed, no major hardware modifications can be made. Ideal algorithms for SPCs are those in which identical sequences of operations are repeated time and again. Thirdly, the problem to be treated by the SPC must be of some lasting interest, as it may take typically from one to two years to design and build an SPC.

Special purpose computers are used in a wide range of situations. A field in which they have found a particularly important application is experimental high-energy physics: SPCs are essential to the analysis of the immense quantities of raw particle collision data. This application has been described in some detail by Nash.[3] Other examples include a special purpose device which multiplies SU(3) matrices,[4] an SPC for studying the SOS model of a crystal surface,[5] and the world's champion chess computer Belle.[6]

At this Statphys Conference we shall be concerned with applications in statistical mechanics. In Sections 3 and 4 two specific problems will be described for which SPCs are operational. The first one is the Ising model, and the second one is the molecular dynamics of a classical many-particle system. By its very nature each SPC is unique. We shall nevertheless attempt, in Section 5, to discuss certain global features that are common to the three laboratory-built SPCs which exist for the above problems. In

Sections 6 and 7 we shall discuss how in constructing each of these SPCs one has taken advantage of the specifics of the algorithm which it executes.

3. THE ISING MODEL

3.1. The Problem

In the Ising model of ferromagnetism (see, e.g., Refs. 7 and 8) a set of N "spin" variables s_i, each taking only the values $+1$ or -1, is located at the sites i of a regular spatial lattice. To each configuration $\{s_i\}$ a (reduced) energy $E(\{s_i\})$ is assigned. Typically $E(\{s_i\})$ is a sum on all single spin variables and on all products of two spins at pairs $\langle i, j\rangle$ of neighboring sites:

$$E(\{s_i\}) = -K\sum_{\langle i,j\rangle} s_i s_j - H\sum_i s_i \tag{1}$$

In certain cases three- or four-spin couplings, or couplings involving further-neighbor pairs, are of interest. A major problem is to calculate averages $\bar{f}$ of certain simple spin functions $f(\{s_i\})$ (for example, $f(\{s_i\}) = N^{-1}\sum_i s_i$) with respect to the Boltzmann weight $\exp[-E(\{s_i\})]$, i.e.,

$$\bar{f} = \frac{\sum_{\{s_i\}} f(\{s_i\})\exp\left[-E(\{s_i\})\right]}{\sum_{\{s_i\}}\exp\left[-E(\{s_i\})\right]} \tag{2}$$

One is particularly interested in the behavior of $\bar{f}(K, H)$ close to the critical point $(K, H) = (K_c, 0)$, where in an infinite system it is singular in K and H.

A second class of problems concerns the kinetic behavior of a time-dependent version of the Ising model, as originally introduced by Glauber (see, e.g., Ref. 9).

3.2. Algorithm

The standard Monte Carlo method (see, e.g., Ref. 10) for studying the Ising model is to generate a stochastic sequence of configurations $\{s_i^0\}$, $\{s_i^1\}, \{s_i^2\}, \ldots, \{s_i^t\}, \ldots$. The tth configuration can be obtained from the $(t-1)$th by the following simple updating algorithm:

i. Select a "central" site j on the lattice.

ii. Put $s_k^t = s_k^{t-1}$ for all $k \neq j$.

iii. Compute the change ΔE_j^t in the reduced energy that would result from replacing s_j^{t-1} by $-s_j^{t-1}$.

iv. Draw a random number r_t from a uniform distribution on the interval (0, 1); if $r_t < 1/[1 + \exp(-\Delta E_j^t)]$, then put $s_j^t = s_j^{t-1}$; otherwise put $s_j^t = -s_j^{t-1}$.

Other algorithms are possible, but in practice one uses either this one or slightly modified versions of it.

The algorithm generates states $\{s_i\}$ with a probability distribution which satisfies a master equation. It is easily shown that for $t \to \infty$ each $\{s_i\}$ occurs with a frequency proportional to its Boltzmann weight. Hence averages like $\bar{f}$ can be approximated by averages on a finite sequence $\{s_i^t\}$, $t = 0, 1, \ldots, T$, provided T is sufficiently large. The convergence will in general be as $1/\sqrt{T}$. In the special case that the central sites are selected randomly, the sequence $\{s_i^t\}$ can be identified, furthermore, with the time evolution of a kinetic Ising model.[9]

In an average size Monte Carlo simulation N may be of the order of 10^3 and T/N of the order of 10^4. To get sufficiently accurate results near criticality, where large and slow fluctuations occur, both N and T/N may have to be as much as 100 times larger.

4. MOLECULAR DYNAMICS OF PARTICLES WITH SHORT-RANGE INTERACTIONS

4.1. Molecular Dynamics

An old problem in statistical mechanics is how to describe phase transitions and other complicated collective phenomena in a many-particle system (in continuous space, as opposed to the lattice model of the preceding section). Recent surveys relevant to the present discussion are, e.g., Refs. 11 and 12. In the simplest case one considers a system of N particles numbered $i = 1, 2, \ldots, N$, of mass m, with spherically symmetric pair interactions. Such a model may represent, for example, a collection of noble gas atoms. Let the particle positions and velocities be denoted by $\mathbf{r}_1, \ldots, \mathbf{r}_N$ and $\mathbf{v}_1, \ldots, \mathbf{v}_N$, respectively. The system's total potential energy is then

$$V(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \tfrac{1}{2} \sum_{i \neq j} \phi(\mathbf{r}_i - \mathbf{r}_j) \tag{3}$$

The particles obey Newton's equations of motion,

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i, \qquad m \frac{d\mathbf{v}_i}{dt} = -\frac{\partial V}{\partial \mathbf{r}_i} \tag{4}$$

By integrating these equations one can obtain, in principle, the phase space

trajectory $\{\mathbf{r}_i(t), \mathbf{v}_i(t)\}$ of the system for $-\infty < t < \infty$. Quantities of physical interest are averages of simple functions $f(\{\mathbf{r}_i(t), \mathbf{v}_i(t)\})$ along this trajectory.

4.2. Algorithm

One possible algorithm for integrating the equations of motion is the leapfrog scheme.[11] A time step Δt is chosen and one defines

$$\mathbf{r}_{i,n} = \mathbf{r}_i(n\Delta t), \qquad \mathbf{v}_{i,n-1/2} = \mathbf{v}_i((n - 1/2)\Delta t) \tag{5}$$

where n is an integer. One supplies an initial condition $\{\mathbf{r}_{i,1}, \mathbf{v}_{i,1/2}\}$ and at the nth step of the algorithm one calculates, in the order indicated,

$$\mathbf{v}_{i,n+1/2} = \mathbf{v}_{i,n-1/2} - \frac{\Delta t}{2m} \sum_{j \neq i} \frac{d\phi(|\mathbf{r}_{i,n} - \mathbf{r}_{j,n}|)}{d\mathbf{r}_{i,n}}, \qquad i = 1, \ldots, N \tag{6a}$$

$$\mathbf{r}_{i,n+1} = \mathbf{r}_{i,n} + \Delta t \mathbf{v}_{i,n+1/2}, \qquad i = 1, \ldots, N \tag{6b}$$

Repetition of this algorithm yields a phase space trajectory of the system (we ignore errors due to discretization and roundoff). The values of physical quantities calculated from a finite trajectory will converge to the asymptotic values for an infinite trajectory as the number of time steps $N_{\Delta t}$ increases. This convergence goes generally as $1/\sqrt{N_{\Delta t}}$, which is one reason why long simulations are needed.

An important class of potentials $\phi(r)$ are those that tend to zero so rapidly with increaseing interparticle distance r that they may effectively be set equal to zero beyond a cutoff radius r_c. We shall refer to these as short-range potentials. For these the sum in (6a) contains only a small number of terms and the execution time of this step in the algorithm goes down from an amount of order N^2 to an amount of order N. An average size simulation with a short-range potential may involve, say, 10^2 to 10^3 particles, and 10^4 or more time steps.

5. DESIGN OF SPECIAL PURPOSE COMPUTERS: GENERAL CONSIDERATIONS

In Sections 6 and 7 we shall consider in some detail the design of two SPCs for the Ising problem[13,14] and of an SPC for molecular dynamics.[15] Each of these three machines is unique and reflects the highly personal approach of its designer. We shall nevertheless try to describe a global structural framework within which the individual designs can be explained.

In all these hardware devices one can distinguish a number of sections according to their function.

(1) A main memory section contains all the data on which the algorithm acts (spin variables, or positions and momenta).

(2) A second memory section contains auxiliary data, stored in a lookup table. For the case of the Ising model, for instance, it may contain the transition probabilities, and for the case of molecular dynamics, the interaction potential and force.

(3) In a calculational section the actual arithmetic takes place. It may contain adders, multipliers, comparators, and so on. The calculational process consists of transferring a small amount of data from the memory to the calculational section, updating their values as prescribed by the algorithm, and writing them back into the memory. Data transport takes time and it is important that fast memory access be possible.

(4) If the algorithm is stochastic, as for Monte Carlo simulations, then the SPC has to contain a random number generator. This generator must be so designed that it produces (pseudo) random numbers of high quality at a very high rate (1 to 100 MHz). In both Ising SPCs discussed below use is made of linear feedback shift registers (see, e.g., Ref. 16).

(5) An SPC is driven by the system clock. The data flow is first of all controlled by the predefined and fixed hardware wiring. Nevertheless, certain specific control tasks may be handled by a special control section. This section may, for example, generate the addresses of memory locations and contain counters. If an SPC is to be relatively flexible, there will be many control tasks and a good solution is to use microprocessors in the control section.

(6) As the data flow through the SPC goes on, the memory content representing the physical system evolves in time. At certain points in this flow one can copy ("tap") data that one wants to save for later analysis, and store these in the memory of a measurement section of the SPC. (Alternatively, they may be sent directly to the host computer; see later.) The measurement section may also perform some of the data analysis itself. If one arranges for the operations in the measurement section to take place parallel with the execution of the main algorithm, the efficiency can be increased considerably.

(7) The hardware sections (1) to (6) constitute the SPC and may together be considered by the user as a black box. An input–output channel connects the SPC to a host computer. The input consists of the initial values for the memory, the values of all adjustable parameters and functions in the simulation, the programs for the microprocessors, and instructions to carry out specific tasks (such as one iteration of the algorithm). The output consists of the accumulated measurements, or of the entire memory contents. Usually an important part of data analysis re-

mains to be done outside the SPC. This may be done in the host computer, or in an array processor attached to it.

(8) The user communicates interactively with the SPC via a terminal of the host computer by means of a software package in a high-level programming language. The more flexible the SPC, the larger the software package will have to be. In any case it will serve to define the values of physical parameters and of the initial configurations, set the length of a particular simulation, and direct the output data towards their files.

Below we shall see how these general remarks apply to three different particular cases.

6. ISING PROCESSORS

Since for the Ising model a binary representation is natural, and since this model still presents a great challenge to statistical physicists, it is not surprising that Ising special purpose computers have been constructed. In 1982 two such devices became operational: one in Delft (the Netherlands), constructed by Hoogland,[13] and one in Santa Barbara (California), constructed by Pearson, Richardson, and Toussaint.[14] Furthermore, the architecture of the ICL DAP (see, e.g., Ref. 17) has recently proven itself particularly suited to the study of Ising-type problems.

The design of the Delft Ising system processor[13] (DISP) combines speed with flexibility. The DISP has a memory of 2^{22} (about four million) spins. These variables, or a fraction of them, can be arranged in either a two- or a three-dimensional lattice ($d = 2$ or 3) with periodic boundary conditions. For $d = 2$, interactions can be defined between each spin s_i and any group of spins within the 7×7 array of which s_i is the center, and for $d = 3$ between each s_i and any group of spins within the $3 \times 3 \times 3$ array of which it is the center. This allows for many-spin interactions and in $d = 2$ also for further-neighbor interactions. Moreover, a direct magnetic field and certain staggered fields can be applied.

The memory is randomly addressable, i.e., individual spin variables can be independently stored into and retrieved from their location in memory. The calculational section performs, in particular, the operations (iii) and (iv) of the algorithm already described. Each pass through the algorithm involves a central spin and the spins with which it is interacting. Getting these variables from memory takes time. In the DISP the values of a central spin and its neighbors, say z in number, can be transported simultaneously to the calculational section due to a proper organization of the memory. This leads to a reduction factor of z with respect to an ordinary computer for this part of the execution time. The selection of the central sites can be performed either sequentially or at random, the latter

option allowing for the simulation of time-dependent phenomena in the kinetic Ising model, such as droplet formation and phase separation. At present a study of the two-dimensional anisotropic next-nearest-neighbor Ising (ANNNI) model[18–20] is being carried out.

The DISP, whose components have cost about $10000, has an updating rate of 1.5 million spins per second. It is interfaced to an HP 1000 host computer.

The Santa Barbara Ising SPC[14] was designed to carry out the Monte Carlo algorithm for the three-dimensional Ising model with nearest-neighbor interactions in the fastest possible way. It contains a memory of $(128)^3 = 2^{21}$ spins, and the lattice, which may contain this number of sites or a fraction of it, is subject to slightly skewed periodic boundary conditions. This SPC selects the central spins in the Monte Carlo algorithm in a fixed sequential order. The designers have cleverly taken advantage of the fact that due to this fixed order it is known in advance which data will be needed at what time in the calculational section. The memory data (the spin values) are arranged in one long shift register and move one step every clock cycle in such a way that the data needed by the algorithm present themselves automatically before a "window" of the calculational section at the right time. This idea circumvents the necessity of address generation and substantially reduces the amount of wiring needed. The calculational section has, in fact, 16 windows looking out on 16 different locations in the lattice, and does the calculation on each of them in a four-stage pipeline.

The Santa Barbara Ising processor can be used, for example, to study the precise location of the critical point, critical exponents, and the validity of hyperscaling[21] in the three-dimensional Ising model. The device, whose components cost only about $2000, can update as many as 25 million spins per second. It is attached to a VAX-11/780 host computer.

The commercially available ICL Distributed Array Processor has been described in detail elsewhere.[17] Basically, it contains a square array of 64×64 processing elements, each of which has 4096 bits of memory associated with it. The processing elements all execute the same operation at the same time. While each individual processing element is relatively slow, the machine derives its power from its enormous parallelism. Because of this architecture the DAP can deal particularly efficiently with problems defined on a lattice. For the Monte Carlo study of the Ising model Wallace and collaborators[22] attain an updating rate of 2.7 to 6 million spins per second (depending on the random number generator used). The stronger general purpose character of the DAP (as compared to the fully specialized processors discussed before) enables one to perform a complete Monte Carlo renormalization group[23,24] analysis with the DAP. This analysis yields results of a given precision in shorter running times.[22] Other lattice

problems (such as the Ising spin glass[25]), have also been programmed successfully on the DAP. The DAP is used in combination with an ICL host computer.

For comparison we finally quote the performance of a commercial general purpose computer for the Ising Monte Carlo algorithm. On a CDC 7600, and using a multispin coding technique, Kalle and Winkelmann[26] were able to obtain an updating rate of nearly 1 million spins per second. We conclude that for a sufficiently precisely defined problem the SPC approach can be a very cost-effective solution.

7. MOLECULAR DYNAMICS PROCESSOR

Whereas the idea of constructing a special purpose computer is natural for the Ising model, which is entirely describable in terms of binary variables, there is much less *a priori* evidence that such an approach pays for the molecular dynamics algorithm. Working with real instead of with logical variables poses many extra problems. A feasibility study of a molecular dynamics SPC, carried out in 1978 by Bakker[27] in Delft, concluded nevertheless that it would be worthwhile to undertake such a project. In 1982 a molecular dynamics SPC, designed and constructed by Bakker, became operational. We shall describe the principal features of this machine. A detailed account will be given by Bakker in a future publication.[28]

Since hardware implementation of fixed-point operations is relatively straightforward and cheap compared to implementation of floating-point operations, the SPC was designed to work with fixed-point numbers. It was built to handle two- or three-dimensional problems. The particle memory, an important part of the processor, contains for each particle a word of 188 bits of information. Of these 188 bits, 24 are reserved for each of the position coordinates and 32 for each of the velocity coordinates; 16 are used for bookkeeping purposes. A maximum of $2^{14} - 1 = 16383$ particles can be treated. The SPC has been constructed so as to allow for a future extension of the memory to $2^{16} - 1 = 65535$ particles. The particles are confined to a square box $[-1, 1]^2$ (or to a cubic box $[-1, 1]^3$) with periodic boundary conditions in all directions. The fixed-point arithmetic imposes the condition that all particle velocities be scaled so as to keep each of their components within the interval $[-1, 1]$. A separate memory in the SPC contains the interparticle potential $\phi(r)$ and the force-over-distance $r^{-1}d\phi/dr$. Both are tabulated at 1024 points equidistant in the variable r^2 on an interval $(0, r_c^2)$. Two extra tables are used for linear interpolation. In the hardware design provisions have been made for an extension to systems

of up to four types of particles that can differ in mass and have different interparticle potentials.

The equations of motion are numerically integrated according to the leapfrog algorithm described previously. The steps (6a) and (6b) are carried out in separate parts of the calculational section. The "position step" (6b) is relatively simple as it involves only two (or three) multiplications and additions per particle. The "velocity step" (6a) is time consuming, since it requires that for each particle we consider all the other ones with which it interacts in order to calculate the total force on it. The implementation of this procedure in hardware is such that the required operations are done in parallel for the three spatial dimensions, as shown schematically in Fig. 1.

In this figure each triplet of parallel planes represents a triplet of simultaneously operating hardware units, performing the same operation for the x, the y, and the z component. Only the operations on the x components have been indicated explicitly. The two first triplets of planes, marked x_i and x_j, respectively, represent fast memories which are filled from time to time with the position coordinates $\mathbf{r}_i = (x_i, y_i, z_i)$ and $\mathbf{r}_j = (x_j, y_j, z_j)$, respectively, of all particles in two linked-list cells.[11] The next triplet of planes represents the simultaneous calculation of $x_{ij} \equiv x_i - x_j$, $y_{ij} \equiv y_i - y_j$, and $z_{ij} \equiv z_i - z_j$, and the following triplet the simultaneous calculation of x_{ij}^2, y_{ij}^2, and z_{ij}^2. These are summed to form r_{ij}^2, which is used to find the force-over-distance $F(r_{ij})/r_{ij}$ in the lookup table. The original values of x_{ij} etc. are then used (see the arrows in the figure) to calculate the Cartesian components $F_x(r_{ij}) = x_{ij}F(r_{ij})/r_{ij}$, etc., of the force. Finally, the changes $\pm\, p_{ij,x}$, etc., in the momentum components of the particles i and j are

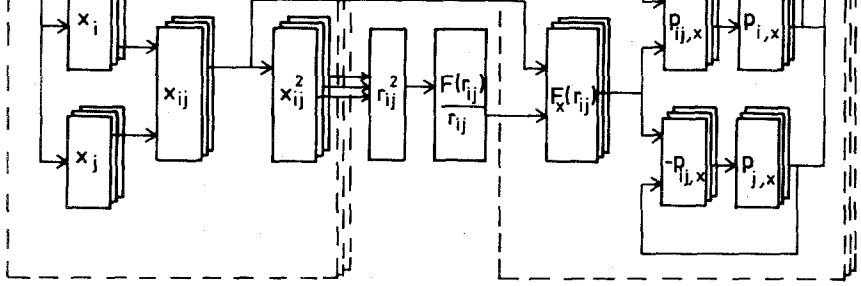

Fig. 1. Schematic representation of the momentum update section in the Delft molecular dynamics processor.

calculated and added to the original momenta $\mathbf{p}_i$ and $\mathbf{p}_j$. These, too, are stored in fast memories, from where they are transported back from time to time to the main memory.

The hardware section represented in Fig. 1 is divided up into ten pipe stages. At each clock cycle (which has a frequency of 4 MHz) a new pair of particles enters the pipeline. Both the pipelining and the parallel processing constitute important advantages over a sequential computer, and even over a vector processor, since the algorithm is not straightforwardly vectorizable.

For the control of the SPC use is made of four bit-sliced microprocessors. Three of these control, respectively, the communication with the host computer, the address generation for the memory, and the linked-list procedure.[11] The last two of these microprocessors are synchronized by a fourth one. All these microprocessors are microprogrammable. The microprograms can be modified if needed and are downloaded into the microprocessors by the host computer.

The SPC structure offers the important advantage that one can measure several physical quantities directly—without any extra cost in computing time—by tapping data from the pipe stages. These quantities are the kinetic and potential energy, and the pair correlation function $g(r)$ for $0 < r < r_c$. From these the pressure and the specific heat can be calculated. It will also be possible in the near future to calculate, at no extra cost in time, for every particle at each time step the vector distances to its six nearest neighbors. These determine the hexatic order parameter, which plays a key role in the theory of two-dimensional melting.[29] Certain other quantities, such as long-range correlations, have to be determined outside the SPC. To this end one can transfer, from time to time, a particle configuration to the HP 1000 host computer, to which an AP 120B array processor is attached.

For a system of 10000 particles the SPC attains a performance of about 1 time step every 2 sec. This is comparable to the speed of a modern supercomputer. The cost of the components of the Delft molecular dynamics SPC has been about \$30000. Presently it is being used for the investigation of the melting of monolayers[29–31,12,15] (two-dimensional systems) of Lennard-Jones particles. Furthermore, low-density properties of a two-dimensional Lennard-Jones system are being investigated by Bruin, Bakker, and Bishop.[32]

A comparison with the ICL DAP is again of interest. Molecular dynamics calculations on this machine have been carried out by Pawley and Thomas[33] for the condensed phase of SF_6, where the molecules are arranged in a regular lattice structure. The molecular dynamics algorithm for this problem is simpler than in Bakker's case because the nearest-neighbor relations between molecules remain fixed; on the other hand, the

intermolecular interactions (a Lennard-Jones potential between the F atoms) is more complicated. For a system of 4096 molecules—each of them treated by one processing element of the DAP—Pawley and Thomas obtain a rate of 1 time step every 3 sec.

8. CONCLUSION

The examples given here convincingly demonstrate the feasibility of the SPC approach to certain large-scale computational problems in physics. Numerous other problems, including many described by partial differential equations, share the characteristics of these examples: local interactions and identical operations at different locations in space. In view of the technological possibilities one can expect in the near future rapid new developments in the field of computational physics. At several institutions experimental SPC designs are in various stages of realization. It seems too early to indicate the precise direction of these developments. In particular, it is not clear at present where the compromise should lie between complete specialization (as in the case of the processors of Hoogland, Pearson *et al.*, and Bakker), and wider-range applicability (as exemplified by the ICL DAP). This question may well not have a unique answer. In any case, the investigation of problems in physics with the aid of special purpose computers is certain to grow in importance in the years to come.

ACKNOWLEDGMENTS

The authors' work in computational physics has been greatly stimulated by Professor J. M. J. van Leeuwen and Professor B. P. Th. Veltman. For discussions on special purpose computers they are indebted to M. Bishop, J. W. Eastwood, H. J. Herrmann, R. W. Hockney, C. Moser, R. B. Pearson, and D. Toussaint.

REFERENCES

1. *Science* **215** No. 4534 (1982).
2. *Phys. Today*, **May** (1983).
3. T. Nash, *Phys. Today*, **May**:36 (1983).
4. N. Christ and A. Terrano, Columbia University preprint CU-TP-259 (1983).
5. H. J. Hilhorst, H. N. J. Vogelij, C. van Leeuwen, and B. P. Th. Veltman, in *Numerical Methods in the Study of Critical Phenomena*, J. Della Dora, J. Demongeot, and B. Lacolle, eds., Springer Series in Synergetics (Springer, Berlin, 1981), Vol. 9, p. 97.
6. J. H. Condon and K. Thompson, in *Advances Computer Chess 3*, M. R. B. Clarke, ed. (Pergamon, New York, 1982), p. 45.
7. C. Domb and M. S. Green (eds.), *Phase Transitions and Critical Phenomena*, (Academic, London), Vol. I (1972); Vol. 3 (1974).

8. B. M. McCoy and T. T. Wu, *The Two-Dimensional Ising Model* (Harvard U.P., Cambridge, Massachusetts, 1973).
9. K. Kawasaki, in *Phase Transitions and Critical Phenomena*, C. Domb and M. S. Green, eds. (Academic Press, London, 1972), Vol. 2, p. 443.
10. K. Binder (ed.), *Monte Carlo in Statistical Physics* (Springer, Berlin, 1979).
11. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw-Hill, New York, 1981).
12. F. F. Abraham, *Rep. Progr. Phys.* **45**:1113 (1982).
13. A. Hoogland, J. Spaa, B. Selman, and A. Compagner, *J. Comput. Phys.* **51**:250 (1983).
14. R. B. Pearson, J. L. Richardson, and D. Toussaint, *J. Comput. Phys.* **51**:241 (1983).
15. A. F. Bakker, C. Bruin, F. van Dieren, and H. J. Hilhorst, *Phys. Lett.* **93A**:67 (1982).
16. S. W. Golomb, *Shift Register Sequences* (Holden-Day, San Francisco, 1967).
17. R. W. Hockney and C. R. Jesshope, *Parallel Computers* (Adam Hilger Ltd., Bristol, 1981).
18. M. E. Fisher and W. Selke, *Phys. Rev. Lett.* **44**:1502 (1980).
19. J. Villain and P. Bak, *J. Phys. (Paris)* **42**:657 (1981).
20. W. Selke, *Z. Phys. B* **43**:335 (1981).
21. B. A. Freedman and G. A. Baker, *J. Phys. A* **15**:L715 (1982).
22. G. S. Pawley, R. H. Swendsen, D. J. Wallace, and K. G. Wilson, to be published.
23. R. H. Swendsen, *Phys. Rev. Lett.* **42**:859 (1979).
24. H. W. J. Blöte and R. H. Swendsen, *Phys. Rev. B* **20**:2077 (1979).
25. A. P. Young, *Phys. Rev. Lett.* **50**:917 (1983).
26. C. Kalle and V. Winkelmann, *J. Stat. Phys.* **28**:639 (1982).
27. A. F. Bakker, Internal Report SV 78-1, Department of Applied Physics, Delft University of Technology (1978).
28. A. F. Bakker, thesis, Delft University of Technology (1983).
29. D. R. Nelson and B. I. Halperin, *Phys. Rev. B* **19**:2457 (1979).
30. F. F. Abraham, *Phys. Rep.* **80**:339 (1981).
31. A. F. Bakker, C. Bruin, and H. J. Hilhorst, *Phys. Rev. Lett.* **52**:449 (1984).
32. C. Bruin, A. F. Bakker, and M. Bishop, to appear in *J. Chem. Phys.*
33. G. S. Pawley and G. W. Thomas, *Phys. Rev. Lett.* **48**:410 (1982).