JOURNAL OF COMPUTATIONAL PHYSICS 51, 250–260 (1983)

A Special-Purpose Processor for the Monte Carlo Simulation of Ising Spin Systems

A. HOOGLAND, J. SPAA, B. SELMAN, AND A. COMPAGNER

Laboratory of Applied Physics, University of Technology, Delft, The Netherlands

Received August 12, 1982

A description is given of a special-purpose computer for the study of static and dynamic properties of stochastic Ising-like systems of up to 4×10^6 spins. Elementary Monte Carlo steps are executed in the machine at a rate of 1.5 MHz. The first results are given, obtained as part of a testing procedure of the machine.

1. INTRODUCTION

In statistical mechanics, the calculation of static and dynamic properties of Ising systems by means of the Monte Carlo method is today a standard technique [1–4]. The accuracy of the method depends strongly on the range of system sizes that one can study and on the size of the statistical sample drawn through the Monte Carlo process from the relevant part of the configuration space for the system. This is dramatically true in the critical region, where correlation distances and correlation times are very long. Therefore, the large amount of computing time necessary to obtain accurate results on a general-purpose computer constitutes a practical limitation of the method.

The speed of the calculation can be greatly increased when the most time-consuming part is carried out in a special-purpose processor dedicated to that task. Ising systems, the configurations of which have a completely binary nature, are ideally suited for such a machine. This report describes the processor built by one of us (A.H.) for the generation, by means of the Monte Carlo method, of a long sequence of configurations of Ising spin systems; the first results obtained with this machine are included.

It should be mentioned here that recently a similar machine for the Ising model was constructed by Pearson *et al.* [5] in Santa Barbara (California). A rather different special-purpose processor, not for Ising systems but for the simulation of fluid systems by means of the molecular-dynamics method, has recently been constructed by Bakker *et al.* [6] of our laboratory. The appearance of these machines is of course due to the present availability of medium- and large-scale integrated circuits, in particular of high-density and high-speed memory chips at a low cost per bit. With these machines, a much greater calculational efficiency can be obtained at a lower price than with general-purpose computers.

0021-9991/83 $3.00

In Section 2 of this paper the class of Ising systems will be described for which our processor was designed. The general hardware structure of the machine is the subject of Section 3; some technical details are given in Section 4. The random number generator employed and its hardware realization are described in Section 5. Finally, Section 6 contains the first results produced by the machine as part of a general testing procedure.

2. Scope of the Processor

The design of a special-purpose processor is a compromise between flexibility with respect to the problem studied on one hand, and the simplicity and cost of the machine on the other hand. We wanted to study the stochastic Ising model for different lattices in two and three dimensions with the inclusion of next-nearest neighbour interactions, and with a number of spins considerably larger than is usually possible with software Monte Carlo programs. The design of the machine was developed with these very general requirements in mind.

The class of Ising systems that the completed processor can deal with can be described (not exhaustively, though) by the following Hamiltonian:

$$H = B \sum_{i=1}^{N} s_i + K_1 \sum_{\text{nn}} s_i s_j + K_2 \sum_{\text{nnn}} s_i s_j + K_3 \sum_{\text{tr}} s_i s_j s_k + K_4 \sum_{\text{sq}} s_i s_j s_k s_l . \tag{1}$$

Here, $s_i = \pm 1$ is the Ising spin variable, N the total number of spins, and B the external magnetic field. The second term takes the interactions between nearest neighbour (nn) spins on the lattice into account, K_1 being the coupling constant. The third term, with coupling constant K_2, contains the next-nearest neighbour (nnn) interactions. These first three terms can be present for all four lattice types that can be studied by our processor: 2-D triangular, 2-D square, 3-D simple cubic, and 3-D face-centred cubic. The fourth term, with coupling constant K_3, can be present only when the 2-D triangular lattice is studied; it contains the three-spin interactions of each triangular "plaquette," i.e., of each elementary triangle of that lattice. The last term, with coupling constant K_4, can only be included when the 2-D square lattice is studied, and contains the four-spin interactions of each square "plaquette" on that lattice, i.e., the products of the spin values of spins i, j, k, and l located at the four corners of each elementary square of that lattice.

The largest value of N that the spin configuration memory of the machine can cope with is $N = 2^{22}$. Due to the particular organization of the spin memory, necessary for an efficient determination of the local configuration surrounding the spin that is subjected to the Monte Carlo process (the "central spin"), not all system sizes below $N = 2^{22}$ can be simulated (the "local configuration" contains all spins that interact,

through K_1, K_2, K_3, and/or K_4, with the central spin). The system sizes that can be studied are

$$N = 2^k \times 2^l, \qquad 3 \leqslant k, l \leqslant 11 \quad \text{(in 2 dimensions)}, \tag{2}$$

$$N = 2^k \times 2^l \times 2^m, \qquad 2 \leqslant k, l, m \leqslant 7 \text{ (in 3 dimensions)}. \tag{3}$$

The boundary conditions adopted in the machine are periodic, of the toroidal type (no helical shifts).

The above description of the scope of the machine is not completely exhaustive. Instead of the terms with K_2, K_3, and K_4 in Eq. (1), other types of interactions could also be taken into account, such as those leading to the so-called ANNNI (Anisotropic Next-Nearest Neighbour Ising) models [7]. Also, instead of B a staggered external magnetic field B_{st} could be introduced. The main restriction of the machine is that the number of spins in the local configuration with which a given spin is interacting does not exceed 30.

3. General Hardware Structure

The processor is a bus-oriented system, the functional organization of which is depicted in Fig. 1. This diagram is a simplified representation of the actual hardware and it just shows what information is processed in which part of the machine. In terms of this diagram we will now describe how the Monte Carlo simulation takes place in the machine; the technical details will be discussed later.

The heart of the machine is the spin memory, with a capacity of 2^{22} bits, in which the momentary spin configuration of the Ising system is stored. At the start of the

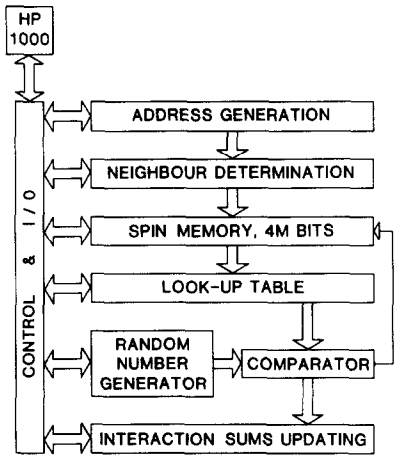

FIG. 1. The functional organisation of the Delft Monte Carlo processor for Ising systems.

Monte Carlo calculation the memory is loaded with an initial configuration (at free choice) by means of the HP-1000 minicomputer. Then the address generator selects a certain spin site. The selection of this "central" spin takes place either in a fixed sequence or in a random fashion, using the same random number generator as employed in the MC process itself. The address of this site is passed on to the neighbour determination section, which (before the calculation was started) was programmed to generate the addresses of all spins of the "local configuration" that interact through one of the desired couplings with the "central" spin. The collection of these addresses is then forwarded to the spin memory, which as a result produces the collection of the spin values at these addresses. The simultaneous production of these spin values was made possible by a special organization of the spin memory; this organization constituted a major design problem and will be discussed later. The collection of the spin values of the local configuration surrounding the central spin defines an address of the look-up table; at this address the value $p_+/(p_+ + p_-)$ has been stored beforehand, depending on the lattice structure and the interactions that one wants to study. Here, the quantity p_+ is the Boltzmann factor associated with the local configuration when the central spin is up, and p_- the Boltzmann factor of the same local configuration but with the central spin down. (There is a certain degeneracy here: not all different local configurations have different energies. This is used to reduce the number of entries in the look-up table, by preprocessing the collection of spin values of the local configuration to some extent, i.e., by making use of the local combinatorics).

The value $p_+/(p_+ + p_-)$ produced by the look-up table is now compared, in the comparator, with the last random number R produced by the random number generator (RNG); the constructional details and the properties of the RNG are given in Section 5. When $R < p_+/(p_+ + p_-)$ holds, the new value of the central spin is set equal to $+1$ ("up"), otherwise it is set equal to -1 ("down"). Whenever the new value of the central spin differs from the old one, the necessary change in the spin memory is made, and the interaction sums (the relevant sums of Eq. (1)) are updated. This particular Monte Carlo procedure is completely standard, and was used originally by Yang [8]; his recipe obeys detailed balance and guarantees that the system relaxes to thermodynamic equilibrium. As an alternative, Fosdick's MC procedure [9] could be realised by changing the contents of the look-up table.

4. Technical Details

The technical realisation of the 4M bits random-access spin memory (the S-RAM) and its interwoven relation with the neighbour-determination section will now be described. On one hand, the Ising spin lattice is thought to be subdivided into a square or cubic array of cells of 64 spins each; for 2-D lattices each cell is a square block of 8×8 spins, for 3-D lattices it is a block of $4 \times 4 \times 4$ spins. On the other hand, the S-RAM is in reality subdivided in 64 separate memory banks, each organized as a 64K $\times$ 1 RAM.

Equivalent positions within different cells of the lattice will never appear in the same local configuration (usually consisting of nearest and next-nearest neighbours only), therefore their addresses can be located on the same memory bank. The requirement that all spin values of the local configuration can be determined simultaneously can now be met if the information stored in different memory banks can be read in a parallel manner. The chosen cell size (i.e., the number of separate memory banks) is simply the smallest integer number that is both square and cubic; it is also larger, as it should be, than the largest number of spins in the local configurations that we wish to consider. For Ising systems smaller than the maximum size $N = 2^{22}$, not all the space available on any one memory bank will be used; the limitation to the system sizes of Eqs. (2) and (3) guarantees that all memory banks will always be used, which is a simplifying feature.

All 64 memory banks are connected with the 22-bit wide A-bus and the 31-bit wide S-bus (see Fig. 2). The A-bus carries the address of the central spin that has to be processed. The S-bus is used to route the spin values of the local configuration to the look-up table on lines that depend exclusively on the position of these spins with regard to the central spin.

Of the address, 16 bits are used to select the cell in which the central spin resides. The other 6 bits provide the address of the central spin within that cell. These 6 bits are connected to identification and decoding circuitry, identical on each of the 64 memory banks. This identification and decoding circuitry activates only those memory banks that contain a spin of the local configuration and routes the values of these spins to the selected lines of the S-bus. When the central spin has a position

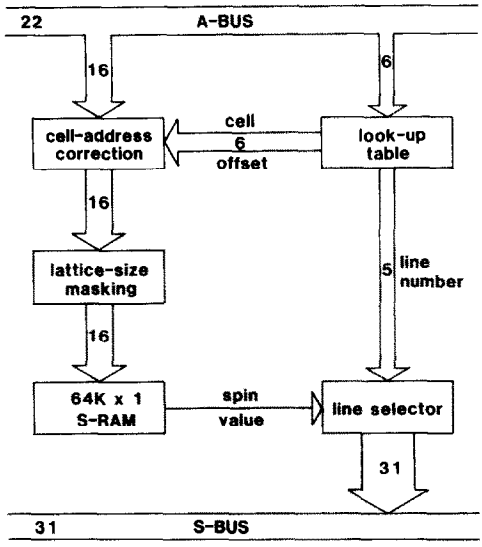

FIG. 2. One of the 64 identical parts of the spin memory, each with its own neighbour identification and decoding section.

close to a cell boundary, some of the spins of the local configuration may be part of neighbouring cells. On memory banks that contain such spins, the 16-bit cell address is corrected. The actual decoding is performed by means of look-up tables that are loaded in the initialisation phase of the MC calculations, according to the particular lattice structure and depending on the particular memory bank that it is attached to.

By means of masking certain higher order bits of the possibly corrected 16-bit cell address, periodic boundary conditions can be maintained for lattice sizes down to 8×8 or $4 \times 4 \times 4$.

The width of the data bus, 31 bits, is larger than is necessary as long as one restricts oneself to Ising systems defined by the Hamiltonian of Eq. (1). The largest number of data bus lines used for these systems is 19 (including a line for the central spin), which occurs for the s.c. lattice (with 6 n.n. and 12 n.n.n.) and for the f.c.c. lattice (with 12 n.n. and 6 n.n.n.). It is not feasible to use 19 bits directly as an address for the look-up table, which then would have far too many entries. However, as far as the energy stored in the local configuration depends only on the number of + spins on the n.n. and n.n.n. positions, the number of bits needed can be reduced considerably. By inspection of all possible cases, including the 2-D cases with multiple-spin interactions, one finds that 10 bits (corresponding to 1K look-up table entries) are sufficient. The actual counting procedure to reduce the original 19 bits to 10 is eliminated by the use of programmable read-only memories (PROMs).

The look-up table contains eight $1K \times 4$ static RAMs, thus offering a precision of 32 bits for the values of $p_+/(p_+ + p_-)$. This precision is more than sufficient.

The calculational speed of the machine is largely determined by the access time of the S-RAM elements, which is 250 nsec; faster low-power elements were not readily available at the time that the printed circuit boards of our processor were assembled. The resulting speed of the machine is about 650 nsec per elementary Monte Carlo step. With the faster elements available today a pipelined structure of the machine would have made sense and would then have increased the speed of the processor by a factor of 5 to 10.

5. The Random Number Generator

The random number generator (RNG) to be employed should produce numbers with a precision of 32 bits (identical to the precision of the look-up table), uniformly distributed on the interval $[0, 1)$, and without correlations. The RNG must be easily realizable in hardware and sufficiently fast. The RNG must be deterministic, such that the sequences of random numbers produced are reproducible, in order to facilitate testing procedures of the RNG and of the processor as a whole, and also to enable one, in a later stage, to repeat the same Monte Carlo experiment in a closer study. A deterministic finite state machine can only produce a sequence with a finite period; this period must not be exhausted in the Monte Carlo experiments that one wishes to perform.

Most of these requirements can be met by adopting as the RNG a two-bit feedback

shift register that produces a maximum-length sequence. The use of the general linear feedback shift register as an RNG was studied by Tausworthe [10], and the special case of a two-bit feedback register was studied by a number of authors, e.g., Zierler [11], Tootill *et al.* [12], and Kirkpatrick and Stoll [13]. It was shown by Tausworthe that these feedback shift registers, under rather general conditions, indeed produce random numbers that are uniformly distributed, and that show no pair-correlation whatsoever; the multiple-correlation behaviour and in particular the so-called run-performance is somewhat less satisfactory in general.

In Fig. 3 a general two-bit feedback shift register of p bits to generate a random number of L bits is depicted. From the initial state of the shift register, a sequence of p bits, the next state is produced by inserting the modulo-2 sum of the feedback bits in positions q and p into position 1; in the shift register, the original bit in position 1 is then shifted into position 2, and so on up to position p, the original contents of which is lost. If all bits are zero in the initial state, the register will remain in that state forever; if the register progresses through all other $2^p - 1$ states before repeating itself, it produces a maximum-length sequence. After L shifts, the contents of the L positions used to generate the random number is completely refreshed and the next random number can be read out.

Zierler [11] lists values for p and q that lead to maximum-length sequences; from that list we selected $p = 127$ and $q = 15$ (in practice, we take the complementary value $q = 112$, which however is equivalent: the only difference is that the sequence of states of the shift register is reversed). The value $p = 127$ was chosen since it is considerably larger (as it should be, according to Tausworthe [10]) than the number of bits of the random number that we want to use, $L = 32$, which is large enough for the precision desired and which is compatible with the HP-1000 computer system. Also, the period $2^{127} - 1$ is not by far exhausted in a Monte Carlo experiment of several days in which random numbers (as in our processor) are needed with a frequency of 1.5 MHz. The value $q = 15$ (or rather 112) was adopted since it seems to lead to an acceptable run-performance, according to arguments given by Tootill *et al.* [12]; further statistical tests are being performed by us for this particular RNG.

Because of the desired speed of the RNG the actual design differs from the simple configuration of Fig. 3. Instead, the configuration of Fig. 4 was used, which consists of 16 chips, each being an 8-bit shift register.

The result of a single shift in each of these registers, including the 16 feedbacks indicated in Fig. 4, is equivalent to 16 shifts of the original two-bit feedback shift register of 127 bits length. After two steps in the new configuration the 32 bits of the

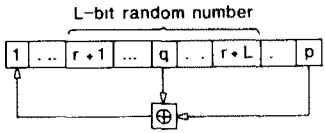

FIG. 3. A general two-bit feedback shift register.

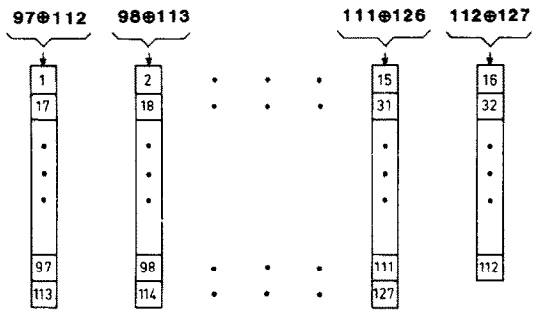

FIG. 4. Actual realization of the single two-bit feedback shift register with $p = 117$ and $q = 112$ in terms of 16 parallel shift registers of 8 bits each (the last bit of the last register is not used); the resulting more complicated feedback scheme is indicated.

random number (which in practice are read from positions 96 through 127) are completely refreshed, and the random number (produced now at a speed that is 16 times faster) is ready for use.

6. TESTING PROCEDURES AND RESULTS

During the design phase of the machine several parts were constructed in prototype in order to check their behaviour. This was followed by the construction of 32 identical printed circuit boards, each containing a combination of two memory banks with their own neighbour-determination circuitry. The other sections of the machine, each occupying a single board or part thereof, were wire wrapped.

The first main testing procedure was carried out when all boards were mounted and the machine was put together. All memories involved in the different sections are accessible both for loading and reading; this enables one to load them and subsequently to check whether the right information is stored at the right place (these test procedures are in fact incorporated in the initialisation phase for each MC calculation). For instance, when the look-up table and/or the RNG is loaded with special values (e.g., zeros only), the result of a single sweep through the lattice is particularly simple in terms of the initial configuration. In this way, one is able to check the addressing in the spin memory as well as the functioning of the updating section. For the largest systems possible and for the four different lattice structures, the correct functioning of the neighbour-determination section defining the local configuration was checked by comparing the actual spin configuration with the information transported in the 31-bit data bus to the look-up table. The dynamical behaviour of the machine was tested at normal speed by means of a logic analyzer. In these tests it was found that about 1% of the integrated circuits used had to be replaced.

The second main testing procedure consisted of a detailed comparison between

(relatively short) MC calculations of the completed processor and MC calculations with a software simulation of the processor. This software simulation copied the intended functioning of the machine in every detail, including the look-up table and the RNG in particular. The software simulation is in principle a completely independent MC programme (though due to its particular requirements a very inefficient one). Initializing both the machine and its software simulation with the same start configuration and the same state of the RNG, we found that after 100,000 MC steps (this number is determined by the slowness of the software simulation) exactly the same configuration and the same collection of interaction sums were produced by the machine and its simulation.

The third main testing procedure of the processor is a comparison between its results and exact values. Therefore, the 2-D square Ising lattice in zero field with nearest neighbour interactions was studied. For this case, exact formulas based on the Onsager solution for the energy and the specific heat have been given by Ferdinand and Fisher [14] for a number of different lattice sizes, using strictly toroidal periodic boundary conditions. The numerical values derived from these formulas are compared with the MC results of our processor in Table I, which lists for different system sizes and for different temperatures around the critical temperature the exact

TABLE I

The MC Results for the Internal Energy and the Specific Heat per Spin Compared with the Exact Values for Different Temperatures and for Different Sizes of the 2-D Sq. Ising System with n.n. Interactions

		U/NkT		C/Nk	
N	kT	MC	exact	MC	exact
16×16	1.80	−1.0330(1)	−1.0330	0.439(1)	0.439
	2.00	−0.8729(1)	−0.8728	0.723(2)	0.726
	2.20	−0.7049(1)	−0.7046	1.288(4)	1.291
	2.40	−0.5232(2)	−0.5229	1.406(4)	1.408
	2.60	−0.3999(1)	−0.3997	0.801(2)	0.799
32×32	1.80	−1.03302(3)	−1.03295	0.438(1)	0.439
	2.00	−0.8730(1)	−0.8728	0.723(2)	0.725
	2.20	−0.7034(1)	−0.7030	1.390(5)	1.401
	2.40	−0.5055(1)	−0.5053	1.286(4)	1.284
	2.60	−0.3958(1)	−0.3956	0.708(1)	0.708
64×64	1.80	−1.03297(2)	−1.03295	0.439(1)	0.439
	2.00	−0.87282(2)	−0.87278	0.721(1)	0.725
	2.20	−0.7031(1)	−0.7030	1.409(4)	1.415
	2.40	−0.50186(4)	−0.50176	1.136(3)	1.134
	2.60	−0.39553(1)	−0.39550	0.706(1)	0.705

Note. The figures between parentheses give the standard deviation, e.g., −0.7049(1) stands for -0.7049 ± 0.0001.

values of the internal energy per spin U/NkT, the specific heat per spin C/Nk, and the corresponding MC results of our processor.

The MC results (including their standard deviations) were obtained from 20 series of 25,000 observations of the number N_{+-} of opposite n.n. pairs on the lattice, taken at "time" intervals of 32 MCS/spin (elementary MC steps per spin). From these data the energy and the specific heat can be calculated. In the same series of observations the magnetization and the susceptibility as well as the distribution for the energy and magnetization are obtained. For a direct comparison with exact results the energy and the specific heat are most suitable.

All temperatures in Table I are around the critical temperature T_c, where the comparison is most sensitive. Outside the critical region the MC results agree with the exact values within the standard deviation. In the critical region, although the absolute accuracy of our results is satisfactory, as is apparent from the table, the difference between the calculated and the exact values tends to be larger than the calculated standard deviations. This cannot be attributed to the possibility that the observations are not truly independent. Although in the critical region a number of 32 MCS/spin may not be large enough, the standard deviatioans were calculated from the 20 partial results, each containing 25,000 observations. These partial results can be considered to be independent.

Therefore, the discrepancy found must probably be ascribed to the RNG employed, as indicated by tests of its run performance.[1] It should be noted that the very extensive MC calculations carried out in the processor at a rate of 1.5 MHz are a very severe test for any RNG.

Pending a solution of this problem, we consider our processor to be a very efficient means of performing MC experiments on Ising systems.

REFERENCES

1. K. BINDER, *Physica* **62** (1972), 508.
2. E. STOLL, K. BINDER, AND T. SCHNEIDER, *Phys. Rev. B* **8** (1973), 3266.
3. D. P. LANDAU, *Phys. Rev. B* **14** (1976), 255.
4. K. BINDER, *in* "Topics in Current Physics," Vol. 7, Springer-Verlag, Berlin, 1979.
5. R. B. PEARSON, J. L. RICHARDSON, AND D. TOUSSAINT, to be published.
6. A. F. BAKKER, C. BRUIN, F. VAN DIEREN, AND H. J. HILHORST, *Phys. Lett.* **93A** (1982), 67.
7. M. E. FISHER AND W. SELKE, *Phys. Rev. Lett.* **44** (1980), 1502.
8. C. P. YANG, *Proc. Symp. Appl. Math.* **15** (1963), 351.
9. L. D. FOSDICK, *Phys. Rev.* **116** (1959), 565.

[1] Since this was written, a much more powerful RNG was built. This RNG can be operated as a single two-bit feedback shift register with a length of up to 2^{15} bits. Alternatively it can be operated as a collection of 32 independent shift registers, each with a length of up to 2^{10} bits. In both cases it features programmable selection of shift register length and feedback position. In the first results obtained with this new RNG (which will be described in a forthcoming paper) the above discrepancy has disappeared altogether.

10. R. C. TAUSWORTHE, *Math. Comput.* **19** (1965), 201.
11. N. ZIERLER, *Inf. Control* **15** (1969), 67.
12. J. P. R. TOOTILL, W. D. ROBINSON, AND A. G. ADAMS, *J. Assoc. Comput. Mach.* **18** (1971), 381.
13. S. KIRKPATRICK AND E. P. STOLL, *J. Comput. Phys.* **40** (1981), 517.
14. A. E. FERDINAND AND M. E. FISHER, *Phys. Rev.* **185** (1969), 832.