JOURNAL OF COMPUTATIONAL PHYSISCS 64, 380–388 (1986)

A Special Purpose Computer for the Electrical Conductivity of Disordered Media

F. HAYOT, H. J. HERRMANN, AND J.-M. NORMAND

Service de Physique Théorique, CEN-Saclay, 91191 Gif-sur-Yvette Cedex, France

AND

P. FARTHOUAT AND M. MUR

Service d'Electronique et d'Informatique des Particules Elémentaires, CEN-Saclay, 91191 Gif-sur-Yvette Cedex, France

Received April 17, 1985; revised July 25, 1985

We describe the architecture of a special purpose computer designed to calculate with high precision the critical exponents characterizing the threshold behavior of electrical conductivity in random mixtures of conductors–insulators, or superconductors–conductors. The algorithm that is used, which determines the architecture, is that of the strip method. Computations are being done in 64-bit floating point arithmetic, at a rate of 8 MFLOPS. © 1986 Academic Press, Inc.

I. INTRODUCTION

In many laboratories physicists have begun to set up powerful computing systems, using commercially available products, to calculate numbers relevant to physics problems recognized to be of particular importance. The main reason is the lack of time available, for practical of financial reasons, on present supercomputers. It has, for instance, been estimated by Weingarten [1] that a hundred years of running on a Cray 1S are required to obtain meaningful results for the hadronic spectrum in lattice gauge theories. Another reason is simply the desire to acquire a knowhow at a time when the use and necessity of powerful computers is spreading to many fields.

Many of the special purpose computers built or planned get their computing power from the use of identical processors running in parallel, as for example the Caltech hypercube [2]. The project we are concerned with is not of this type. It is a very fast processor doing 64-bit floating point arithmetic, at a rate comparable to that of the Cray 1S for the physics problem being studied. This is a typical percolation problem of the conductivity of random resistor networks. It is only the very recent commercial availability of very fast 64-bit floating point components that makes the realization of the project possible, within a reasonable span of time.

0021-9991/86 $3.00

Copyright © 1986 by Academic Press, Inc.
All rights of reproduction in any form reserved.

This paper is organized as follows. In Section II, we describe the physics problem and the algorithm we use to solve it. We mention the results obtained previously on the Cray 1S. In Section III, we explain with reference to the previous section the aim of our project, and the constraints that are implied. We describe the architecture of the special purpose computer, with emphasis on the very fast processor. Other components such as the memory and a random number generator are discussed. In Section IV we present the status of our project and the outlook.

II. The Physics Problem: Conductivity of Random Resistor Networks

The numbers this special purpose computer is supposed to calculate with high precision are the critical exponents of the threshold behavior of electrical conductivity in infinite random resistor networks. There are two types of behaviour:

(i) The first one concerns a random mixture of insulators (resistance $r = \infty$) and conductors (resistance $r = 1$).

Suppose each link in the network is either a conductor with probability p or an insulator with probability $1 - p$. For p above a critical value p_c, the percolation threshold, a current will traverse the network. The conductivity behaves at the threshold as $(p - p_c)^t$ (see Fig. 1a). The exponent t characterizes this behavior and is believed to be universal (it only depends on the dimension of space). It is moreover related to random diffusion and to the spectrum of vibrations on the percolation cluster at p_c [3].

(ii) The second one concerns a random mixture of superconductors (resistance $r = 0$) and conductors ($r = 1$).

Here one supposes that a network link is superconducting with probability p and conducting with probability $1 - p$. For p approaching the percolation threshold $p_c (p \leqslant p_c)$ the conductivity will become infinite, with the characteristic behavior (see Fig. 1b) $(p_c - p)^{-s}$, where again s is universal.

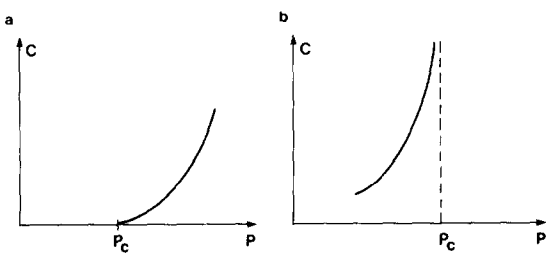

FIG. 1. (a) Threshold behavior (at p_c) of conductivity for a random mixture of conductors and insulators. (b) Threshold behavior (at p_c) of conductivity for a random mixture of conductors and superconductors.

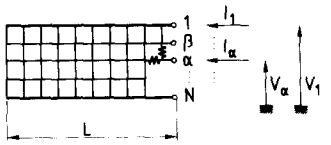

FIG. 2. Random mixture of conductors–insulators. Step by step building of the network by adding a new link either longitudinal on line α or transverse between α and β. The top and bottom are perfect conductors. There are N current lines and potentials, corresponding to the width of the strip (or the transverse section of a bar in three dimensions). L denotes the length of the strip (or bar).

Our main aim is to increase by an order of magnitude the precision with which the exponents t and s are presently known. The algorithm we use to do the calculation is based on the strip method [4], which also has lead, up to now, to the most precise determinations of t and s. To simulate the infinite medium, one dimension (the longitudinal one) is made as large as possible. Finite size effects in the transverse directions are handled by finite size scaling arguments, as shown in Ref. [4, 5].

The method is illustrated in the 2-dimensional case in Fig. 2 and Fig. 3 for the cases respectively of insulator–conductor and superconductor–conductor networks. The network is modelled by a retangular grid, of width N and length L. The network is built step by step (at $p = p_c$), where the comparison of a random number with p_c decides whether the new link is insulating (or conducting) or conducting (or superconducting). In the first case the top and bottom of the network are perfect conductors, in the second case the leftmost part is a perfect conductor, and there are periodic boundary conditions from top to bottom. The construction of the network consists in adding longitudinal links on line α, and transverse links between lines α and β (see Fig. 2), and the calculation consists in deriving after each addition of a link the new form of the conductance or resistance matrix, relating the currents to the potentials. For the insulator–conductor case potentials are fixed and currents calculated, and the new elements of the symmetric (N by N) conductance matrix after adding a longitudinal or transverse link are given in terms of the old ones by

$$A'_{ij} = A_{ij} - \frac{A_{i\alpha}A_{\alpha j}}{\frac{1}{r} + A_{\alpha\alpha}} \qquad \text{(longitudinal link)} \tag{1}$$

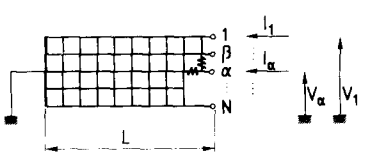

FIG. 3. Random mixture of conductors and superconductors. The description is the same as for Fig. 2, except that here there are periodic boundary conditions from top to bottom, and the left-hand side of the strip (or bar) is now a perfect conductor.

and

$$A_{ij} = A_{ij} + \frac{1}{r}(\delta_{i\alpha} - \delta_{i\beta})(\delta_{\alpha j} - \delta_{\beta j}) \qquad \text{(transverse link)} \tag{2}$$

where $1 \leqslant i, j \leqslant N$, and $r = \infty$ or $r = 1$ according to whether the link is insulating or conducting. Details about this method are given in Ref. [4].

For the case of a conductor–superconductor mixture, currents are fixed and potentials calculated and the new elements of the (symmetric) $N \times N$ resistance matrix are given in terms of the old ones by

$$R'_{ij} = R_{ij} + r\,\delta_{i\alpha}\,\delta_{\alpha j} \qquad \text{(longitudinal link)} \tag{3}$$

and

$$R'_{ij} = R_{ij} - \frac{(R_{i\alpha} - R_{i\beta})(R_{\alpha j} - R_{\beta j})}{r + R_{\alpha\alpha} + R_{\beta\beta} - R_{\alpha\beta} - R_{\beta\alpha}} \qquad \text{(transverse link)} \tag{4}$$

where $1 \leqslant i, j \leqslant N$, and r = 1 or $r = 0$ according to whether the link is conducting or superconducting. (For details see Ref. [5].)

For N fixed, as the length L tends to infinity, the conductivity σ_N is given in the two cases by

$$\sigma_N = \lim_{L \to \infty} \frac{A_{11}}{L} \qquad \text{or} \qquad \sigma_N^{-1} = \lim_{L \to \infty} \frac{R_{11}}{L},$$

the convergence towards these values being like $1/\sqrt{L}$. As mentioned above, the relevant exponent is extracted by using finite size scaling in the width N, which requires calculations to be performed at different values of N.

The results obtained on a Cray 1S by this method of strips of width N and length L, as well as the corresponding running time, are summarized in Table I. In three dimensions the strip is replaced by a bar of length L and the transverse cross sec-

TABLE I

Space dimension	$d = 2$	$d = 3$	
Bond-percolation threshold	0.5	0.2492	
t or s	$t = s = 1.30 \pm 0.01$	$t = 1.9 \pm 0.1$	$s = 0.75 \pm 0.04$
Strip size	$N \leqslant 20$, $L \leqslant 10^7$	$N \leqslant 441$, $L \leqslant 10^7$	$N \leqslant 100$, $L \leqslant 10^6$
Computing time on Cray 1S	5 hours	3 hours	25 hours

Note. This table gives results obtained up to now by the strip method on a Cray 1S, including the computing time. These results are the best available. For $d = 2$ it can be shown that one has $t = s$. Note that there exists no result at $d = 4$.

tion contains a number N of network sites. Let us point out that the precision of the results only depends on the length L of the network that fits into a given computer time.

III. Architecture of the Special Purpose Computer

Our aim is to increase 10-fold the precision of the numbers in Table I. Since this precision goes as the square root of L, lengths L, of the order of 10^9, a hundred times longer than those considered up to now, have to be introduced. This means a hundred times longer computation times on a Cray 1S than those given in Table I. The update of one element of the conductance or resistance matrix takes between 60 and 100 nsec on the Cray. Two requirements as to the performance of a very fast special purpose computer result:

(i) speed: the processor has to work at a clock rate giving a performance equivalent to that of a supercomputer;

(ii) precision: only 64-bit floating point arithmetic will do, because one must be able to distinguish from a number of the order of $L = 10^9 \simeq 2^{30}$ the resistance of unity that is added to the strip.

Both requirements are satisfied as far as components are concerned by using Weitek's 64-bit ALU (WTL 1065) and multiplier (WTL 1064), the first one turning out a result every 120 nsec (in the pipeline mode) and the second one every 480 nsec.

The architecture of the very fast processor, built from these arithmetic components, reflects closely the structure of the algorithm. The algorithm is based on formulae (1) to (4), and the most complicated of these (formula (4)) determines the minimum complexity required for the structure of the processor.

In fact it is modified formula (4) that is used. Experience with the Cray shows that in order to increase precision, calculations involving the differences of two large numbers, such as $R_{i\alpha} - R_{i\beta}$ in (4), ought to be avoided. (The R_{ij}'s increase proportionally to L.) At each update of the resistance matrix, it is therefore better to calculate not R'_{ij} but a modified matrix

$$\tilde{R}'_{ij} = \tilde{R}_{ij} - \tilde{R}_{11} - V_i V_j, \tag{4bis}$$

where

$$V_i = \frac{\tilde{R}_{i\alpha} - \tilde{R}_{i\beta}}{\sqrt{D}}, \qquad D = r + \tilde{R}_{\alpha\alpha} + \tilde{R}_{\beta\beta} - \tilde{R}_{\alpha\beta} - \tilde{R}_{\beta\alpha}.$$

At each step the $\tilde{R}_{11}$ is subtracted. The successive $\tilde{R}_{11}$'s are accumulated in the memory in order to give the true R_{11}. There are two independent Random Access Memories (RAM) (Fig 4), the sizes of which are determined by the fact that we want to consider values of N up to 256- and 64-bit words:

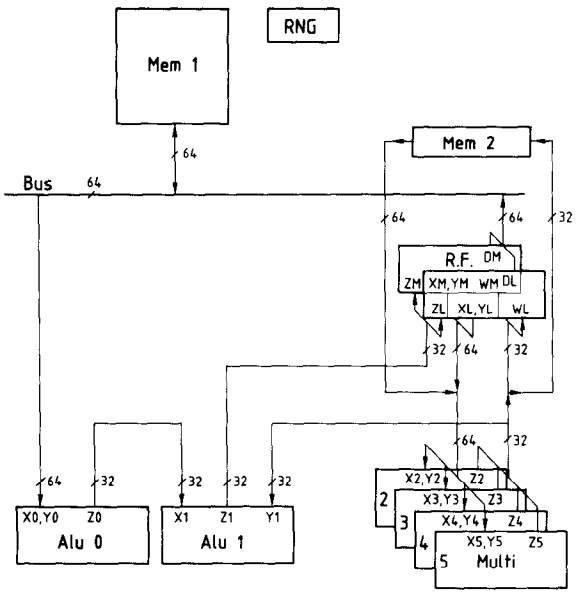

FIG. 4. Architecture of the processor, using Weitek components: two ALU's (WTL 1065), four multipliers (WTL 1064), and two register files (WTL 1066), which handle 64-bit floating point numbers. There are two independent memories, 1 and 2.

(a) Memory 2, of 2 Kbytes, which provides storage for the N-dimensional vector V_i.

(b) Memory 1, of 512 Kbytes, which is the main memory containing the matrix elements. This memory is a simplified and faster version (access time of 35 nsec) of the memory used in the construction of the IBM 3081/E emulator [6].

The aim of the calculation of formula (4bis) is to update a matrix element every 120 nsec. The main pipeline does simultaneously the $N(N+1)/2$ substractions $\tilde{R}_{ij} - \tilde{R}_{11}$ (the matrix is symmetric) and the $N(N+1)/2$ multiplications V_i by V_j, followed by a subtraction of $V_i V_j$ from $\tilde{R}_{ij} - \tilde{R}_{11}$ (cf. (4bis)). Within every 120 nsec an element R_{ij} is read from the main memory and a new one $\tilde{R}'_{ij}$ written into it. To implement the above pipeline there are two ALU's (WTL 1065), numbered 0 and 1 (cf. Fig 4), and a multiplier consisting of four WTL 1064 chips working sequentially in order to turn out a result every 120 nsec (in pipeline mode). A direct access from memory 2 to the inputs of the multiplier provides two 64-bit operands V_i and V_j every 120 nsec. Moreover a direct access of memory 1 to ALU 0 equally provides this arithmetic unit with two 64-bit operands every 120 nsec, used in the initial stages for calculating $\tilde{R}_{i\alpha} - \tilde{R}_{i\beta}$ which enters the expression of V_i.

There are two register files WTL 1066 in parallel, providing 32 registers 64-bits wide. They are used for transit between ALU 1 and main memory (for $\tilde{R}'_{ij}$), for storage of certain numbers, and especially to provide the first approximation for the

calculation of $1/\sqrt{D}$ (cf. (4bis)). The iterative procedure to reach 64-bit precision for $1/\sqrt{D}$ involves the multiplier, ALU 1, and register file.

The performance of this architecture is summarized in Table II, where the time required for each step of the calculation is given. It is the pipelined calculation (a result every 120 nsec) of the $N(N+1)/2$ matrix elements R_{ij} which as soon as N is large enough ($N > 20$) is the main contribution to the time it takes to update an element. The time required for a complete calculation is thus proportional to $N^2 \times N = N^3$, there being N transverse links (in 2 dimensions for large N, $2N$ in 3 dimensions for large N) in the superconductivity case. The time is also proportional to L, the total length of the strip. In the conductivity case the N^3 dependence is an effective N^2 because the corresponding matrix has many zeroes [7].

Connected to the processor is a random number generator (RNG), which works with 32-bit integers. It is of the lagged Fibonacci type, extensively studied by Marsaglia [8]. It calculates a random number from an initial store of 17 numbers (not all even) by the recurrence

$$X_n = (X_{n-5} + X_{n-17}) \qquad \text{mod } 2^{32} \ (n \geqslant 18). \tag{5}$$

Its period [8] is $2^{31}(2^{17} - 1) \cong 2.8 \times 10^{14}$.

The RNG is built so it can readily accept bigger sequences than the (5, 17) doublet. A random number is calculated according to Eq. (5) in an ALU and compared to p_c. The result of this comparison is either 0 or 1 according to whether the random number is bigger or smaller than p_c. This signal is stored in an intermediate latch, which is unloaded whenever needed by the processor in the update of a matrix element. The RNG can produce a result every 120 nsec. It has two parallel RAM, which simultaneously provide the operands in Eq. (5), two sequential ALU's for addition and comparison, and a microcode in PROM controlling it.

A sequencer and an address generator will control our fast processor (Fig. 5) through a microcode in RAM loaded from the host computer. The microcode will provide much of the flexibility of the machine: change of space dimension, super-

TABLE II

	D	$1/\sqrt{D}$	Pipeline 1 "difficult" links	Pipeline 2 "difficult" links	"easy" links
Time in t_{cy}	53 (27)	168	$17 + 2N$	$29 + N(N+1)$	$29 + 2(6)$

Note. Time it takes to perform the successive steps in updating one element. The unit of time is the clock cycle ($t_{cy} \geqslant 60$ nsec). The numbers 17 and twice 29 correspond to start up times for pipelines. Pipeline 1 is the calculation of the N component vector V_i. Pipeline 2, the main pipeline, is the calculation of $N(N+1)/2$ matrix elements. "Difficult" links are the transverse ones for superconductor-conductor mixture, the longitudinal ones for conductor-insulator mixture. For "easy" links it is the reverse and the update takes $2\,t_{cy}$ in the conductivity case and $6\,t_{cy}$ in the superconductivity one. The calculation of D takes $53\,t_{cy}$ (respectively 27) for the superconductivity (respectively conductivity) case.

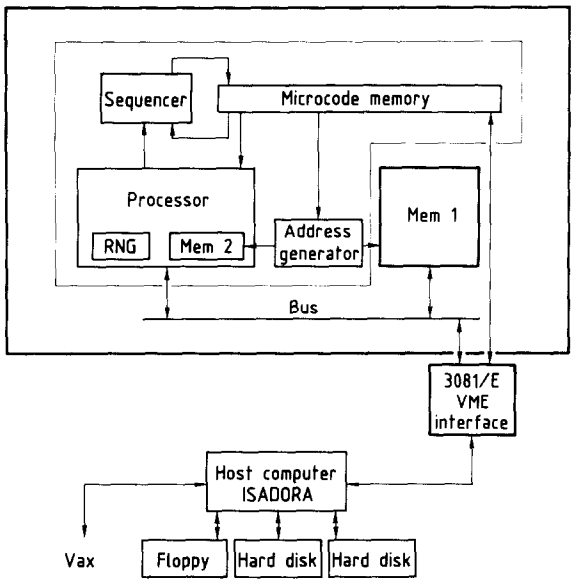

FIG. 5. Environment of the fast processor, with sequencer, address generator, and connection to a host through VME interface of the 3081/E emulator.

conductor–conductor versus conductor–insulator, periodic boundary conditions, and site instead of bond percolation.

The fast processor, Memory 1, and RNG will be realized on three printed circuit boards plugged into a rack identical to that of the IBM 3081/E emulator. A fourth board will be dedicated to the interface between the 3081/E backplane and the host computer.

The host will be the ISADORA built at the high-energy electronics laboratory at Saclay from VME standard boards. It contains a Motorola 68000 microprocessor with 512-Kbyte memory, an additional double port VME-VMX 1-Mbyte memory (to rapidly spool memory 1), and one floppy and two hard Winchester disks to periodically save intermediate results. A connection to a VAX will provide additional possibility of storage. The tasks of the host are multiple: running checks, loading microcode and initial configuration, saving intermediate results, and analysing results.

IV. STATUS AND OUTLOOK

The project is well advanced. Most components, in particular the Weitek chips, have been delivered or are ordered. The main memory, being similar to that of the 3081/E emulator, is effectively tested. The time flow diagrams for the pipelines of the processor have been drawn. We expect the complete machine to be tested and

running by the end of the year. The total budget is about $ 70,000, half of which is used for high-performance testing equipment: logical analyser, and oscilloscope.

The precise measurements (one order of magnitude improvement) of the dynamic percolation exponents s and t in 2 and 3 dimensions and first estimates in 4 dimensions should help in clarifying their relationships to the geometrical exponents characterizing the percolation cluster. These relationships have been the subject of many conjectures such as those of Alexander and Orbach [3] and Kertész [9]. Problems similar to the one considered might be studied with the same processor. Let us mention the question of anisotropy in 2 dimensions [10], an extension of the present problem to the case where capacitors replace some resistors, and the equation of state, which is now a function of frequency and $p - p_c$, can be investigated. Let us also mention as a possible, different problem that of the elastic constants of a random network.

ACKNOWLEDGMENTS

We are grateful to R. Balian, P. Borgeaud, G. Dreyfus, and C. Moser, for their encouragement and advice.

REFERENCES

1. D. WEINGARTEN, "Algorithms for Monte Carlo Calculations with Fermions," IBM Yorktown Technical Report, 1984 (unpublished).
2. E. BROOKS, G. FOX, S. OTTO, M. RANDERIA, W. ATHAS, E. DE BENEDICTIS, M. NEWTON, AND C. SEITZ, *Nucl. Phys. B* **220** (1983), 383, and *Phys. Rev. Lett.* **52** (1984), 2324.
3. S. ALEXANDER AND R. ORBACH, *J. Phys. Lett.* **43** (1982), 8.
4. B. DERRIDA AND J. VANNIMENUS, *J. Phys. A* **15** (1982), L557; B. DERRIDA, J. G. ZABOLITZKY, J. VANNIMENUS, AND D. STAUFFER, *J. Stat. Phys.* **36** (1984), 31.
5. H. J. HERRMANN, B. DERRIDA, AND J. VANNIMENUS, *Phys. Rev. B* **30** (1984), 4080.
6. P. F. KUNZ, M. GRAVINA, G. OXOBY, P. RANKIN, Q. TRANG, P. M. FERRAN, A. FUCCI, R. HINTON, D. JACOBS, B. MARTIN, H. MASUCH, AND K. M. STORR, CERN preprint DD/84/4 and SLAC-PUB-3332, 1984 (unpublished).
7. B. DERRIDA, J. G. ZABOLITZKY, J. VANNIMENUS, AND D. STAUFFER, *J. Stat. Phys.* **36** (1984), 31.
8. G. MARSAGLIA, "Random Number Generation," in *Encyclopedia of Computer Science*, and "A Current View of Random Number Generation," Sixteenth Symposium of the Interface between Computer Science and Statistics, 1984.
9. J. KERTÉSZ, *J. Phys. A* **16** (1983), L471.
10. M. NAKANISHI, P. REYNOLDS, AND S. REDNER, *J. Phys. A: Math. Gen.* **14** (1981), 855.