SPECIAL PURPOSE COMPUTERS IN STATISTICAL PHYSICS

Hans J. HERRMANN

SPhT, CEN Saclay, 91191 Gif-sur-Yvette Cedex, France

General interest paper

A new trend in physics, namely the building of special purpose computers (SPC), is reviewed. Special emphasis is given to the following questions: Why does one build SPC's? When is it worthwhile to build an SPC? How does one proceed if one wants to build an SPC? Finally the most important results that have been obtained for statistical physics through SPC's up to now will be sketched.

Imagine a theoretical physicist choosing a transceiver for an address bus. His Gradshteyn Ryzhik is deeply buried under electronic design sheets and data books from Texas Instruments. The phone rings and a salesman from Analog Devices gives him the price of a new sequencer.

This scenario is becoming a reality for the increasing number of physicists that are building special purpose computers (SPC). They have chosen a specific problem, like the molecular dynamics of a fluid, the Monte Carlo simulation of a spin glass or the transfer matrix algorithm for random conductivity, and now they build a computer specially tailored for this very purpose.

Why build a special purpose computer?

Isn't it much more comfortable to write a FORTRAN program and execute it on the next available general supercomputer (like a Cray XMP, an IBM 3090, a Cyber 205, etc.)? Several reasons have led people to do it the hard way:

First of all, general computers must usually be shared with many other users, so you will have access to a restricted number of hours, several hundred hours per year at best. If you build your own SPC it will calculate only for you every day of the year, 24 hours a day (except during blackouts).

There also is an important financial aspect: A grant proposal asking for $2 \times 10^7$ US dollars for a Cray 2 is likely to be rejected (unless perhaps you have

0378-4371/86/$03.50$ © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)
already won a Nobel prize). The chances, however, to get $5 \times 10^4$ US dollars for tools and components to build an SPC are considerably better.

Finally there is speed! Supercomputers are famous for being very fast (that is what makes them price-efficient compared to smaller general computers). Now, however, there exist on the market highly integrated ultrafast chips. If, in addition, one arranges these chips inside the SPC in such a way that many of them work simultaneously (doing the same thing (parallelism), or doing different things (vectorization)) the SPC can be as fast or even faster than a supercomputer. Technologically speaking, this is the reason why SPC's are mainly a development of only the last couple of years.

All these reasons in favour of SPC's have to be counterbalanced by the fact that the building of an SPC involves an enormous amount of work in a field that usually has to be learned from scratch.

When is it worthwhile to build an SPC?

Since the building of an SPC is a non-negligible enterprise one should first consider carefully if one has chosen the right problem. There are certain general criteria:

– The problem one wants to solve should be sufficiently fundamental and sufficiently difficult that after two years – which is about the average time one needs to get results from an SPC – the scientific community is still interested.

– The algorithm one uses must be relatively simple, so that the SPC won't get too complex and its construction time too long.

– The method to solve the problem should involve a huge repetition of identical steps (typically $10^8$ to $10^{10}$) so that one can run the SPC for many days without intervening.

– The algorithm and the evaluation of the data should have had extensive previous testing on general computers and one should be so sure about the procedure that one is willing to fix in a definitive way what calculations one will perform. It is usually very difficult to change an SPC once it is built.

How to build an SPC?

Once one has chosen the algorithm for which one wants to build the SPC there are certain necessary conditions required before construction can start: One needs the financial support to buy the hardware components and electronic equipment (like an oscilloscope or a logic analyzer). An enthusiastic team willing to work hard must be formed. At least one person with a solid experience in
electronics must belong to the team. Finally one has to find an electronics lab willing to put a reasonable infrastructure at your disposal.

For what follows, I will report some of the experience we have had in Saclay¹) building an SPC to calculate the electrical conductivity of percolation clusters via the strip method³).

The first, and most creative step is the design of the architecture of the SPC. One has to try to decompose the algorithm into the largest possible number of elementary calculations that can be performed simultaneously. Some calculations are completely independent from the rest, like the generation of random numbers, and can therefore be executed by a processor of its own (MIMD = multiple instruction multiple data arrangement), so they are implemented on a separate electronic board.

For the part of the calculation that will be executed the most one has to construct a “pipeline”. In the case of our Saclay machine the main calculation is the updating of a matrix \( R_{ij} \) through

\[
R'_{ij} = R_{ij} - R_{11} - V_i \ast V_j
\]

where \( V_i \) is a vector and all operations are in 64 bit floating point arithmetic. In fig. 1 we see a simplified diagram of the data flow. Like on a car assembly line the first subtraction of eq. (1) is performed for one matrix element while the second subtraction of eq. (1) is done for another matrix element (MISD = multiple instruction single data arrangement). In order that such a pipeline can work every unit must be finished at the same time. So one has to investigate which chips are available on the market. It happens that the fastest 64 bit multipliers need 480 nsec per multiplication (WTL 1064 of Weitek) while a subtraction can be done in 120 nsec (WTL 1065). So one has to put four multipliers in parallel as shown in fig. 1 (SIMD = single instruction multiple data arrangement) and one can get a result every 120 nsec.

The design of the architecture is, however, far from being finished! In fig. 1 \( R_{ij} \) and \( V_i \) are stored in memories. One needs a device, the address generator, that determines the location of a given matrix element within the memory. Then one

\[\text{Fig. 1. Schematic diagram for an architecture to calculate eq. (1).}\]
needs somewhere to store the "vector-instructions", i.e., the sets of bits that tell each of the chips what function it has to perform. The ensemble of all these vector-instructions is called the microcode. Which instruction is to be applied and how often is managed by a "sequencer". Roughly speaking the sequencer makes the loops and jumps that we are accustomed to use in FORTRAN programs. All these units must have the same speed as the central processing unit (CPU) of fig. 1, i.e., 120 nsec.

Finally a connection to the external world must be established. One needs a host computer, for instance a PC, that will receive the data coming from the CPU, evaluate it and store it on a disk. The host computer need not be fast but must be reliable. From time to time it will make a complete backup of the state of the machine and make consistency checks: It is usually too complicated to use error correcting codes as those implemented on the big general computers, so if one finds an error during a calculation (due for example to a cosmic ray) it is restarted using the last backup as a starting point.

The host computer will not work with 64 bits and will, since it is slow, not be synchronous to the CPU, so an interface between the two is needed which is built on a separate board. The whole ensemble is sketched in fig. 2.

After the architecture is settled important electronic decisions must be taken: Which technology should be used to build the electronic boards: wirewrap or
printed circuits, or should one even make integrated circuits? Also one has to choose all the chips out of the data books of the different manufacturers and to compare them one has to carefully study their specifications. Since about 200 different chips are needed this is a major effort.

After all these decisions are made, a detailed layout has to be made for each board. Each pin of each chip must be properly connected and, of course, the chips must fit on the board.

In order that the connections are not too long and do not cross each other too much the placing of the chips on the board and the routing of the connection lines between them has to be done very carefully. CAD (= computer assisted design) programs are commercially available and can be extremely useful for this purpose. CAD programs can write all information concerning the locations of chips and wires on tape. These tapes can be sent to a company that will make the boards for you within several weeks.

In the meantime you have ordered the chips from the respective manufacturers and built the backplane into which the boards can be placed. You put everything together and the SPC is ready for testing. The debugging will take you some more months before actual calculations can start.

Special purpose computers that have been built

Most of the about twenty SPC's that have been built by physicists in the last five years are related to statistical physics. Their construction time ranges from four months to four years, their price from 200 US dollars to $2 \times 10^5$ US dollars. About half of the SPC's built are restricted to very specific applications; the others are more general. In addition there are at least 30 projects for SPC's about which I will not talk. A partial list of the more specific existing SPC's that are related to statistical physics is given in table I.

Cellular automata have been the cheapest$^3$) and the shortest times were needed to construct them$^4$). One example for a cellular automaton SPC has been

<table>
<thead>
<tr>
<th>Place</th>
<th>Type</th>
<th>Completed</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>M.I.T., USA</td>
<td>cellular automata</td>
<td>1982</td>
<td>3</td>
</tr>
<tr>
<td>E.N.S., France</td>
<td>cellular automata</td>
<td>1986</td>
<td>4</td>
</tr>
<tr>
<td>Santa Barbara, USA</td>
<td>Ising</td>
<td>1982</td>
<td>6</td>
</tr>
<tr>
<td>Delft, Netherlands</td>
<td>Ising</td>
<td>1982</td>
<td>10</td>
</tr>
<tr>
<td>Bell Labs, USA</td>
<td>Ising (spin glass)</td>
<td>1984</td>
<td>12</td>
</tr>
<tr>
<td>Delft, Netherlands</td>
<td>molecular dynamics</td>
<td>1983</td>
<td>16</td>
</tr>
<tr>
<td>Saclay, France</td>
<td>random conductivity</td>
<td>1986</td>
<td>1</td>
</tr>
</tbody>
</table>
presented by Vichniac during this conference). Since cellular automata SPC's can visualize their configurations very nicely on a screen, many interesting pictures have been made with them. Quantitative results have, however, been quite rare up to now. This might change in the future if recent ideas by Pomeau and others to model hydrodynamic flow through cellular automata are exploited.

SPC's of the Ising type have been more successful in yielding precision data. The Santa Barbara machine held the world record for a while for the speed of simulation of an Ising model with 30 million updates per second. (This record was later broken again by general computers.) It yielded precise values for the critical temperature and the dynamic critical exponent of the three dimensional Ising model making nearly $10^{12}$ updates. However, more or less at the same time about equally precise estimates were obtained by other methods. Also some incorrect results were produced for certain system sizes which were presumably due to correlations in the random number generator as was suggested for instance by calculations on the DISP. The DISP is an SPC, slower but more versatile than the Santa Barbara machine. Several other calculations have already been performed on it including some that were reported on at this conference.

Another successful Ising-type SPC is the spin glass machine built in Bell Labs. They found evidence for a finite transition temperature in the three-dimensional short range Ising spin glass. The same result was also found by other authors at the same time using conventional computers. The machine at Bell Labs was also used to study the random field Ising model.

Much more complex than the machines described above are molecular dynamics SPC's since they involve real numbers instead of integers and since they involve long range instead of short range interactions. The first molecular dynamics SPC was built by Bakker in a pioneering effort using fixed point arithmetic. This machine was used to give some evidence that in two-dimensional melting no hexatic phase exists in agreement with work done on conventional computers. Another SPC for molecular dynamics has just been completed in San Jose.

Finally a number crunching SPC of equal order of difficulty has just been finished in Saclay. It will be used to try to settle the question of the validity of the Alexander–Orbach conjecture for percolation clusters and to obtain accurate values for the dynamical critical exponents of percolation. Data are, however, not yet available.

Summarizing, we have described a new effort that has been made in the field of computational physics to overcome usual limitations of general supercomputers. It has to be left to the future to see if this development will survive in the long run or if a new generation of supercomputers will render it obsolete. Some more information about this field can be retrieved from older review articles.
Acknowledgement

Many individuals helped me to get acquainted with this field through interesting discussions. I am particularly indebted to M. Hajjar, F. Hayot, J.M. Normand and D. Stauffer.

References

4) D. d'Humieres, private communication.