

SUE: A Special Purpose Computer for Spin Glass Models

A. Cruz¹, J. Pech^{1,2,3}, A. Tarancón¹, C. L. Ullod¹, C. Ungil¹

¹ Departamento de Física Teórica, Facultad de Ciencias, Universidad de Zaragoza, 50009 Zaragoza, Spain

² Institute of Physics, Academy of Sciences, 180 40 Prague, Czech Republic

³ CERN, 1211 Geneva 23, Switzerland

Abstract

The use of last generation Programmable Electronic Components makes it possible the construction of very powerful and competitive special purpose computers. We have designed, constructed and tested a three-dimensional Spin Glass model dedicated machine. Each single board can simulate 8 different systems, updating all the systems every clock cycle at 48 Mhz. The on-board reprogrammability permits us to change the lattice size, or even the update algorithm or the action. Here we present the first runs using two algorithms: Demon and Heat Bath. The machine includes a component dedicated to random number generation, which provides us with a 32 bit random number every cycle, as needed in canonical algorithms. The update speed of the whole machine, composed of 12 boards, is 217 ps/spin .

Keywords: Ising model, spin-glass, 3d, special purpose machine, programmable logic

1 Introduction

We describe a CPLD-based machine, dedicated to three-dimensional spin glass models with variables belonging to Z_2 and couplings to first neighbours, and report on the reliability tests which have been carried out. In a previous work [1], we presented the prototype for the two-dimensional model. The final version uses larger and faster devices and a multi layer (instead of double layer) PCB. On-board reprogrammability allows change of size, action or algorithm and smooths the debug process, and a 32-bit random number generator has been implemented, allowing the use of the Heat Bath algorithm. A device driver and user libraries have been developed for easy (transparent) use. At present we have built 12 boards and tested them using the Demon and Heat Bath algorithms in different lattice sizes. The update speed of each single board is 2.5 ns/spin .

2 The Physical Model

The action of the 3d Edwards-Andersons model with first neighbour couplings is $S = \sum_{i,j} \sigma_i \sigma_j J_{ij}$, where the value of the Ising spins σ can be $+1$ or -1 , and the couplings J_{ij} are random variables taking the values ± 1 with equal probability. We study the existence of phase transitions using as order parameter the overlap between two independent systems (*replicas*) with the same set of couplings J_{ij} . We should finally average over different realizations of the disorder (J_{ij}) to obtain physical results about the system. The simulation of the different systems can be easily parallelised.

Monte Carlo simulations have been used to study this model [2], but only sizes up to $L = 16$ have been simulated due to the strong slowing-down as the size grows. Only local algorithms achieve good efficiency, which, requiring very simple calculations, can be easily implemented in a dedicated machine. Two different updating algorithms have been implemented in our design so far, one microcanonical (Demon) and one canonical (Heat Bath).

The Demon algorithm [1] keeps the sum of the lattice energy and a demon energy constant. The conservation of the total energy (lattice plus demon), has been useful in the programming and test stages of development. In the Heat Bath algorithm, the new spin value for each site is decided according to the probability distribution of a single Ising spin in the effective magnetic field produced by the fixed neighbouring spins. That probability table is generated according to the value of the temperature and loaded onto the board before the simulation.

3 Operation and General Structure of the Machine

The machine (SUE, for Spin Updating Engine) consists of boards connected to a Host Computer (HC) running LINUX through a PCI Data Acquisition Card. Each board contains the hardware to store and update eight lattices in parallel. There are two degrees of parallelism: inside the processing module and between the modules. SUE performs the update of the configurations, but the measurements and analysis are made by the HC. After a certain number of iterations, SUE is stopped and the configuration is downloaded to the HC. In this sense, SUE and the HC work in parallel: while SUE updates the system the HC processes the previously read configurations. The time spent reading and writing the configurations is around 20% of the computation time in the smaller simulable lattice ($L = 20$), and decreases steeply with size (it is less than 5% of the total time for $L = 30$).

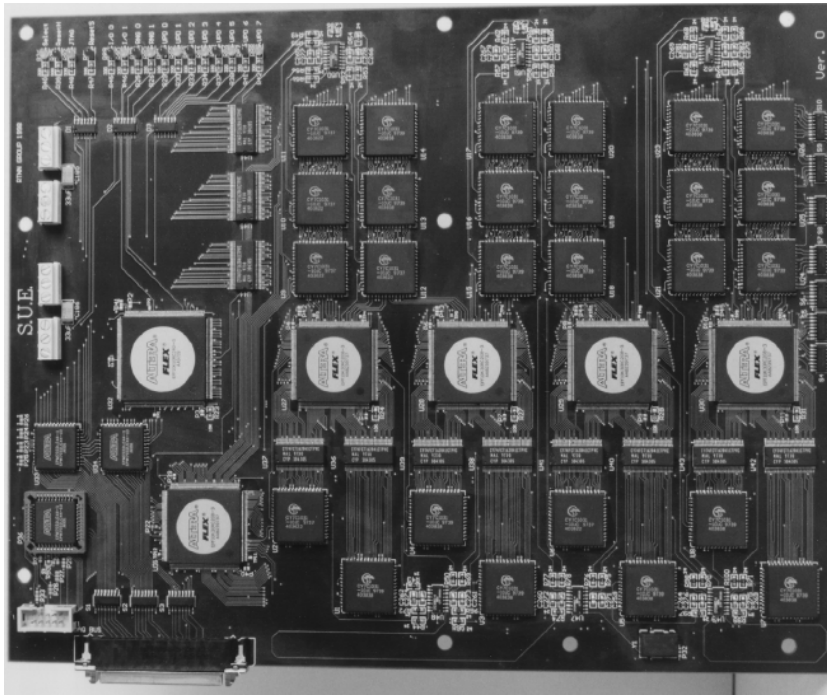


Figure 1: SUE board

The photograph of one of the boards can be seen in fig. 1. It contains 4 components FLEX 10K30 (*UPDATE*) responsible for the core of the Monte Carlo simulation. Each of them updates two lattices in parallel and has lines assigned to access the spin and coupling memories and the 32-bit bus where the random number is provided. That bus is used also to write and read the memories, the demon energy or the probabilities table from the HC. An array of static memory devices (SRAM) stores the couplings of the lattices and two SRAM devices near each *UPDATE* component store the spin variables. Latches are used as tri-state devices to control the data buses

of the spin memories at high frequency and to avoid fat-out problems addressing the couplings memories. Addressing of the memories and synchronization between the components are the main tasks of the fifth FLEX 10K30 (*ADDRESS*), and the *RNG* component is a FLEX 10K50 where a random number generator is programmed.

External communication is provided by three EPM7032 placed near the 68-pin SCSI-II connector. One of those responds to basic commands sent from the HC, allowing to select and program the board when the on-board programmable devices are not yet programmed. The programmed logic establishes 4 control lines in each direction allowing communication between the HC and the *ADDRESS* device, and a 32-bit data bus common to the HC and the *UPDATE* and *RNG* components. The two lower bits in this bus reach *ADDRESS* too, and act as extra control lines when needed. The clock signal is distributed across the circuit at 24 MHz, being doubled to 48 MHz near the final devices. A set of leds permits us to visualise the state of the machine.

3.1 Pipelined Updating

Each column along the x axis is divided in blocks to be stored consecutively in the memories, so to update one of the blocks the block itself and the preceding and following blocks along the x axis have to be supplied, together with the four neighbour blocks y_-, y_+, z_-, z_+ . For computing convenience, helicoidal boundary conditions are used. If an updated spin every clock cycle is wanted, the block size l has to be greater than five, as five blocks should be read for one to be updated. Simulable sizes range from $L = 20$ to $L = 72$.

The 18 coupling memory devices are arranged as a single bank of width $3 \cdot 16$ bits and depth $6 \cdot 64K$. Each one of the eight lattices uses 6 bits to store the couplings of each spin in the site with its six neighbours. The spin memory is duplicated and while one bank is read the other is written. Two $64K \cdot 16$ bits memory devices are connected to each *UPDATE* component, eight bits corresponding to each lattice simulated in that component. Each 8-bit word (9-bit using the parity bit) contains l consecutive spins, and the memories are read following a pattern that makes the block to be updated and its neighbouring blocks available to the *UPDATE* component.

A pipeline structure programmed in the *UPDATE* components performs the algorithm step by step. In the case of block length $l = 5$, the algorithm runs over a state machine with ten states. In each of those states a block is read from one of the memories and stored in an internal register. When a block and its neighbours are all internally registered, spins in that block are sequentially put into the pipeline. After the whole block is updated, it is written back to the other memory. The writing sequence is equal to the reading one, and the neighbours blocks are overwritten (without changes). When a whole column has been updated, we change the role of the memories. The last updated column is wrong in the memory now being written, but it is automatically corrected as the y_- neighbour blocks are copied to that memory. Writing is obviously delayed with respect to reading (11 cycles), and a problem arises therefore when the column is to change. We solve it by storing the problematic blocks in internal registers (*cache*), and reading from there instead than from memory when needed.

3.2 Addressing Logic

The fifth 10K30 device controls and addresses the memories, establishes the functioning mode of the *UPDATE* and *RNG* components and takes charge of the communications with the HC. The board is accessed through a communications port with 32 bidirectional lines devoted to data transfer and 8 control lines (4 in each direction). Data lines are connected through tri-state circuits to the bus connecting the *UPDATE* and *RNG* devices. Control lines are connected to *ADDRESS*, which controls the board according to the commands sent from the HC, and to the 7032 component

allowing selection and programming of the board.

The implemented instruction set allows us to read and write couplings, spins and demon energies, load the RNG seeds and the probability tables used in the Heat Bath algorithm, set the number of iterations and start the simulation. Through five control lines connecting *ADDRESS* with the *UPDATE* devices, commands are encoded to control the data I/O operations involving those devices and to start the update process. Some other dedicated control lines are used during the simulation, to control the pipeline startup, the access to data lines and the cache mechanism.

3.3 Random Number Generator

The Altera 10K50 is a 32-bit pseudo-random number generator of the R250 kind. Those generators are known to suffer some problems in Monte Carlo simulations, but only with non-local algorithms. The *char* wheel used in the C implementation becomes a 32-bit wide shift register into the Altera, reproducing the effect of indexes incrementation. An adder sums two intermediate words and stores the result in the first position. This result also serves as input to a XOR function, together with the value in the last register, and the result of this function provides us with the pseudo-random number, every clock cycle. As only one random number is available in each board, the *replicas* (systems with the same couplings J_{ij}) must be simulated in different boards.

3.4 Software

The boards are connected to the HC through a data acquisition card PCI-DIO32HS from National Instruments. To access the DAQ, a Linux driver has been programmed, and also a user library to manage the boards in a easy way. A set functions is available to the user to command the boards and to program them with the logic corresponding to the algorithm and lattice size to be simulated. The functions writing and reading the memories get the data as arrays S, J_x, J_y, J_z of bytes, as is usual in multi-spin code, so the user needs not to worry about SUE internal details.

4 Performance

To assure the proper working of the machine beyond any doubt, an emulator has been written to run in a PC, so the machine configurations can be compared with those obtained in the PC emulation. Comparing the update speed achieved by SUE with that obtained running highly optimized multi-spin code in a CrayT3E supercomputer [3] we can see that the whole machine matches the computational power of one hundred processors in a CrayT3E.

System	Alpha EV5, 600 Mhz [3]	SUE (single board)	SUE (twelve boards)
Update speed (ns/spin)	22	2.5	0.22

5 Acknowledgements

We wish to thank L.A. Fernández, H.G. Ballesteros and J.J. Ruiz-Lorenzo for useful discussions. Partially supported by DGA (P46/97) and CICYT (AEN97-1768 and AEN99-0990).

References

- 1 J. Pech, A. Tarancón and C.L. Ullod, *Comp.Phys.Com.* **106** (1997) 10.
- 2 E. Marinari, G. Parisi and J.J. Ruiz-Lorenzo, in *Spin Glasses and Random Fields* (World Scientific, Singapore 1998).
- 3 M. Palassini and S. Caracciolo, *Phys. Rev. Lett.* **82** (1999) 5128.