Monte Carlo Methods
Special Purpose Hardware and Machines for Monte Carlo

Dieter W. Heermann

Heidelberg University

December 14, 2020

Special Purpose Hardware and Machines for Monte Carlo

Application areas I

- Image Processing and Pattern Recognition
- Molecular Modelling
- VLSI Design (Optimization)
- Speech Processing
- Computational Physics / Biology / ...
- Financial Modelling
- Cellular Automata / Percolation (oil recovery)

Technologies I

- VLSI Systems
- Field Programmable Gate Arrays

On the hardware level machines were build to support concurrency. Examples are

- The Delft machine (5) (Delft Ising System Processor: DISP) that reflects in a direct way the structure of the Monte Carlo algorithm or
- The machine build by the Santa-Barbara group (6). The Santa Barbara architecture allows to exploit the inherent parallelism of Monte Carlo Ising simulations that result from the data structure and the condition of detailed balance. Instead of using just one processor, one can include many more so that one can update spins in parallel. The processor as such reflects, similar to the Delft computer, the structure of the Monte Carlo algorithm. The Santa Barbara machine optimizes the performance exploiting the data structure and the algorithmic structure.

Technologies II

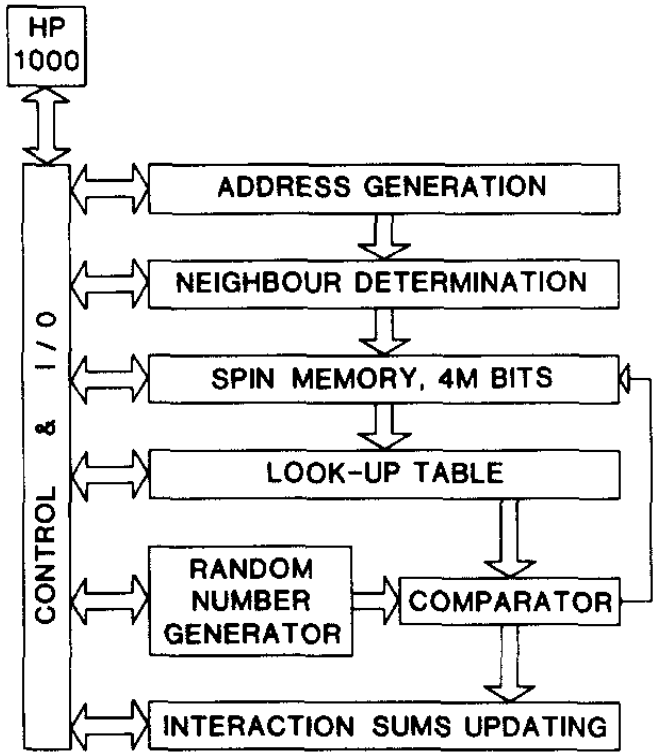

FIG. 1. The functional organisation of the Delft Monte Carlo processor for Ising systems.

Figure taken from: A special-purpose processor for the Monte Carlo simulation of ising spin systems, **A. Hoogland, J. Spaa, B. Selman and A. Compagner, Journal of Computational Physics Volume 51, Issue 2, August 1983, Pages 250-260**

Technologies III

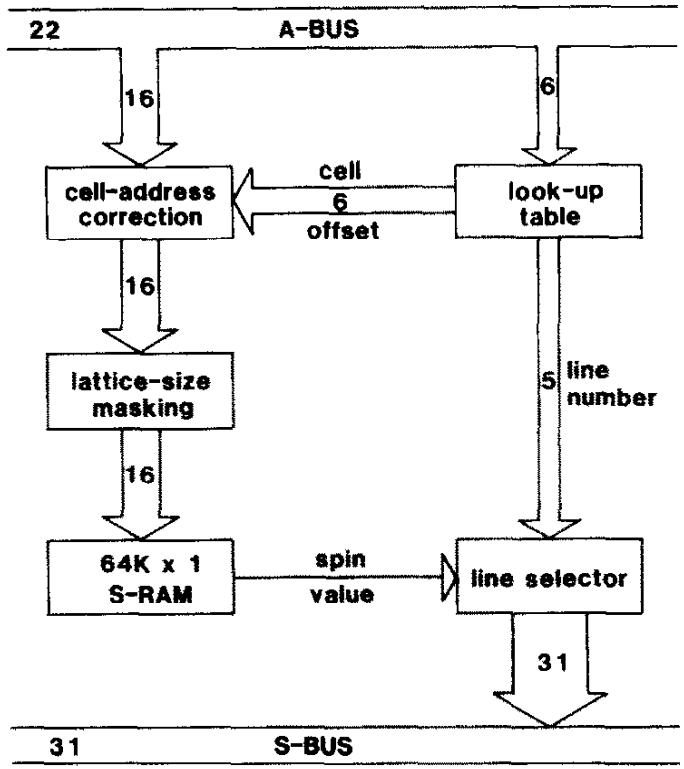

FIG. 2. One of the 64 identical parts of the spin memory, each with its own neighbour identification and decoding section.

Figure taken from: A special-purpose processor for the Monte Carlo simulation of ising spin systems, A. Hoogland, J. Spaa, B. Selman and A. Compagner, Journal of Computational Physics Volume 51, Issue 2, August 1983, Pages 250-260

Technologies IV

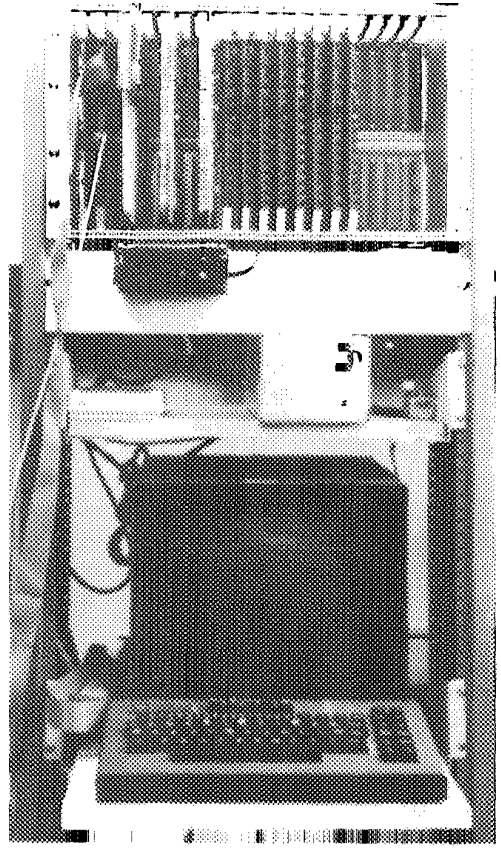

FIG. 2. Photograph of the Monte Carlo computer. Three larger boards on the left constitute the special processor described in Sec. III.

Figure taken from: Fast special purpose computer for Monte Carlo simulations in statistical physics, J. H. Condon and A. T. Ogielski, Rev. Sci. Instrum. 56, 1691 (1985); doi:10.1063/1.1138125 (6 pages)

Technologies V

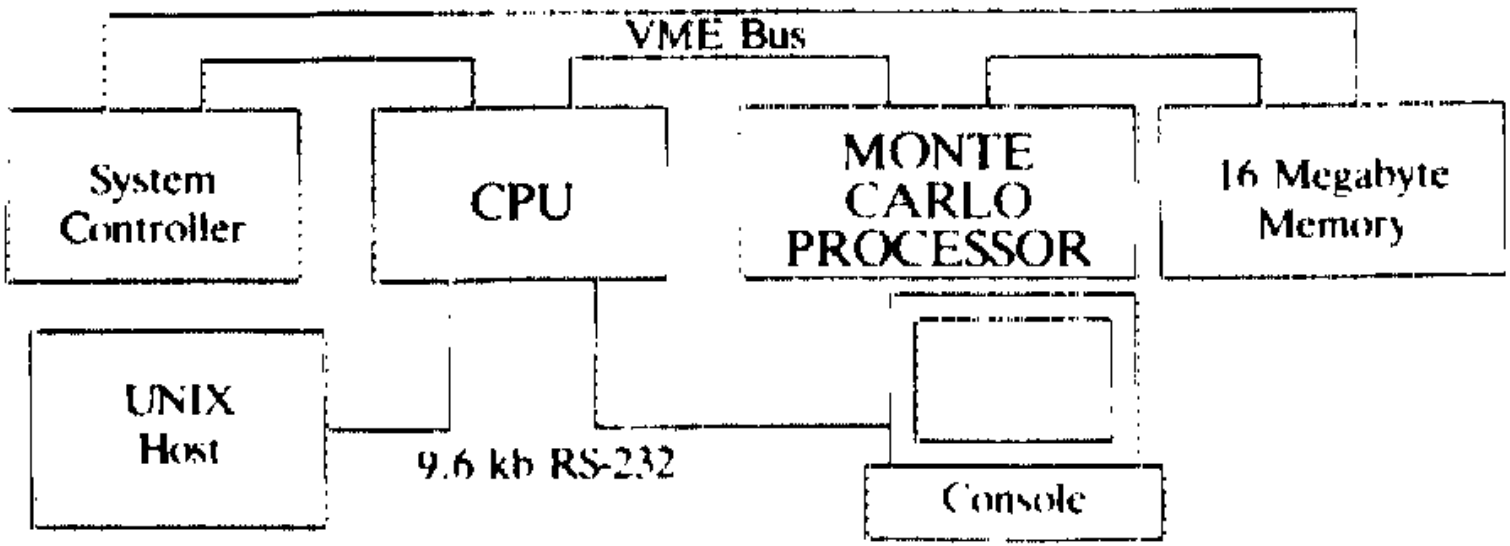

FIG. 1. Architecture of the special purpose computer for Monte Carlo simulations.

Figure taken from: Fast special purpose computer for Monte Carlo simulations in statistical physics, J. H. Condon and A. T. Ogielski, Rev. Sci. Instrum. 56, 1691 (1985); doi:10.1063/1.1138125 (6 pages)

Technologies VI

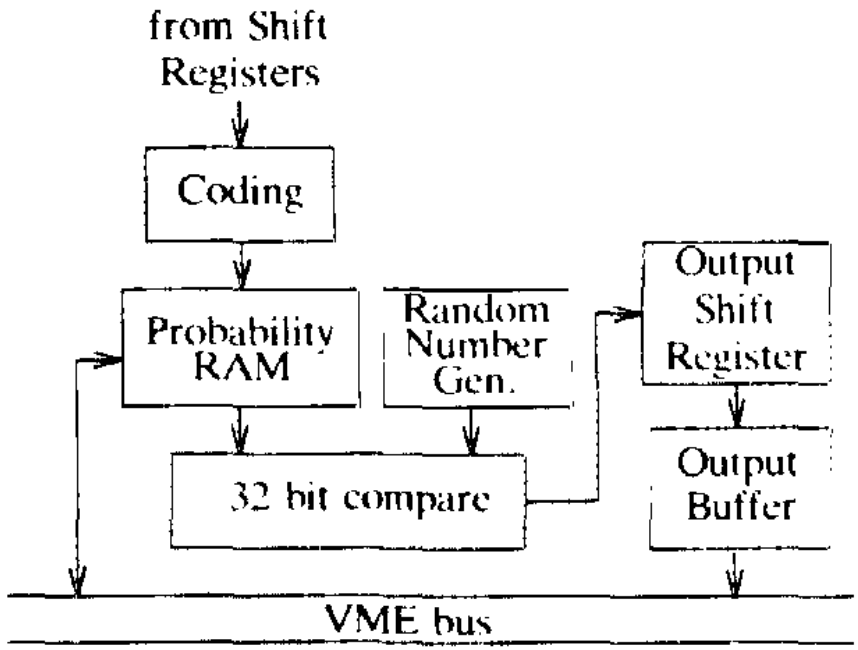

FIG. 6. Block diagram of the spin updating pipeline. The data from the serial input buffer registers are shifted into the decoder; the new updated spins accumulated in the parallel output buffer are written back to the memory.

Figure taken from: Fast special purpose computer for Monte Carlo simulations in statistical physics, J. H. Condon and A. T. Ogielski, Rev. Sci. Instrum. 56, 1691 (1985); doi:10.1063/1.1138125 (6 pages)

A. Cruz et al. / Computer Physics Communications 133 (2001) 165–176

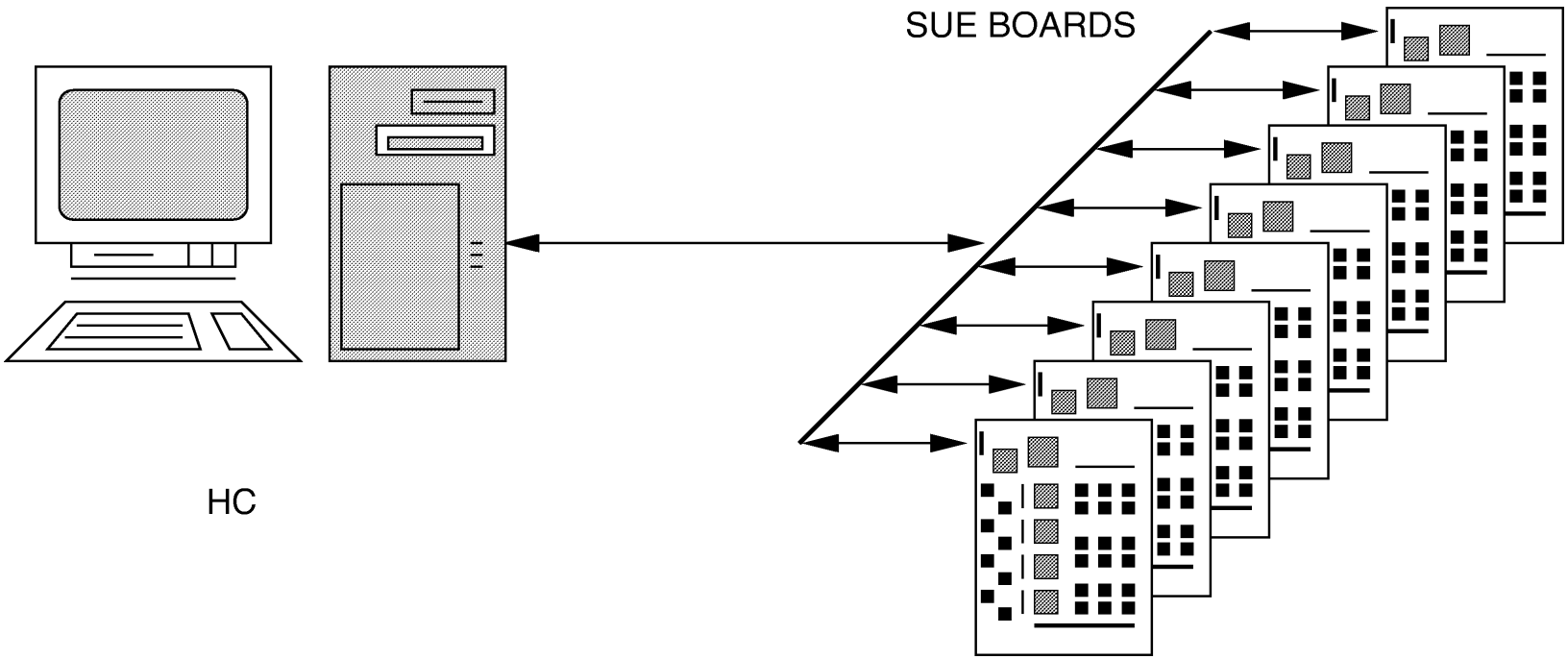

Fig. 1. Schematic view of the full $d = 3$ machine.

Figure taken from: SUE: A special purpose computer for spin glass models, A. Cruz, J. Pech, A. Tarancon, P. Tellezc, C.L. Ulloda, C. Ungil, Computer Physics Communications 133 (2001)

Technologies VIII

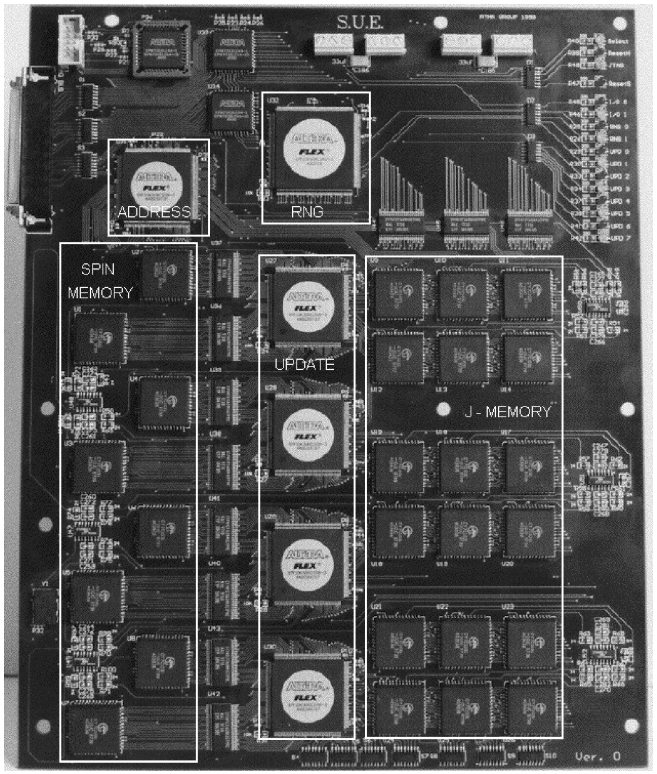

Fig. 2. SUE board.

Figure taken from: SUE: A special purpose computer for spin glass models, A. Cruz, J. Pech, A. Tarancon, P. Tellezc, C.L. Ulloda, C. Ungil, Computer Physics Communications 133 (2001)

Technologies IX

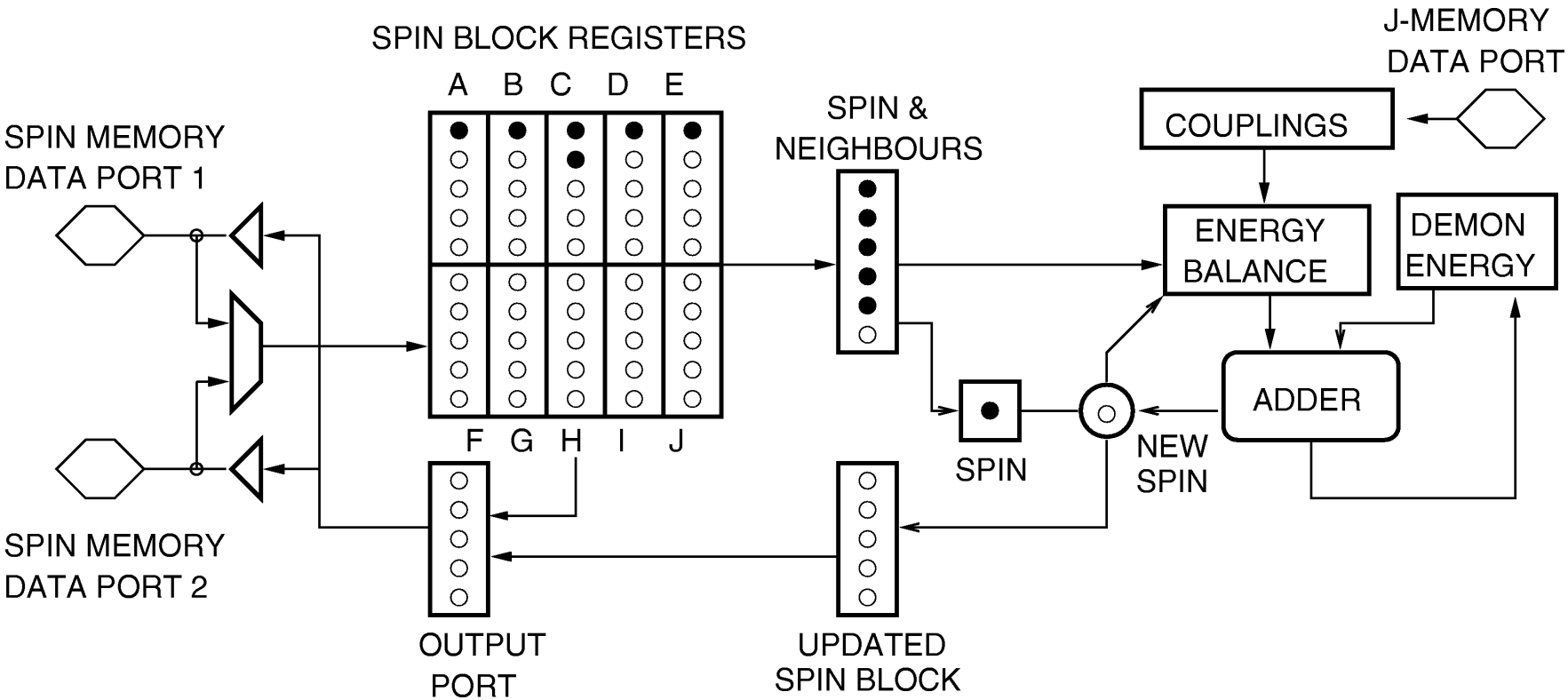

Fig. 3. Demon algorithm pipeline implemented in the UPDATE devices.

Figure taken from: SUE: A special purpose computer for spin glass models, A. Cruz, J. Pech, A. Tarancon, P. Tellezc, C.L. Ulloda, C. Ungil, Computer Physics Communications 133 (2001)

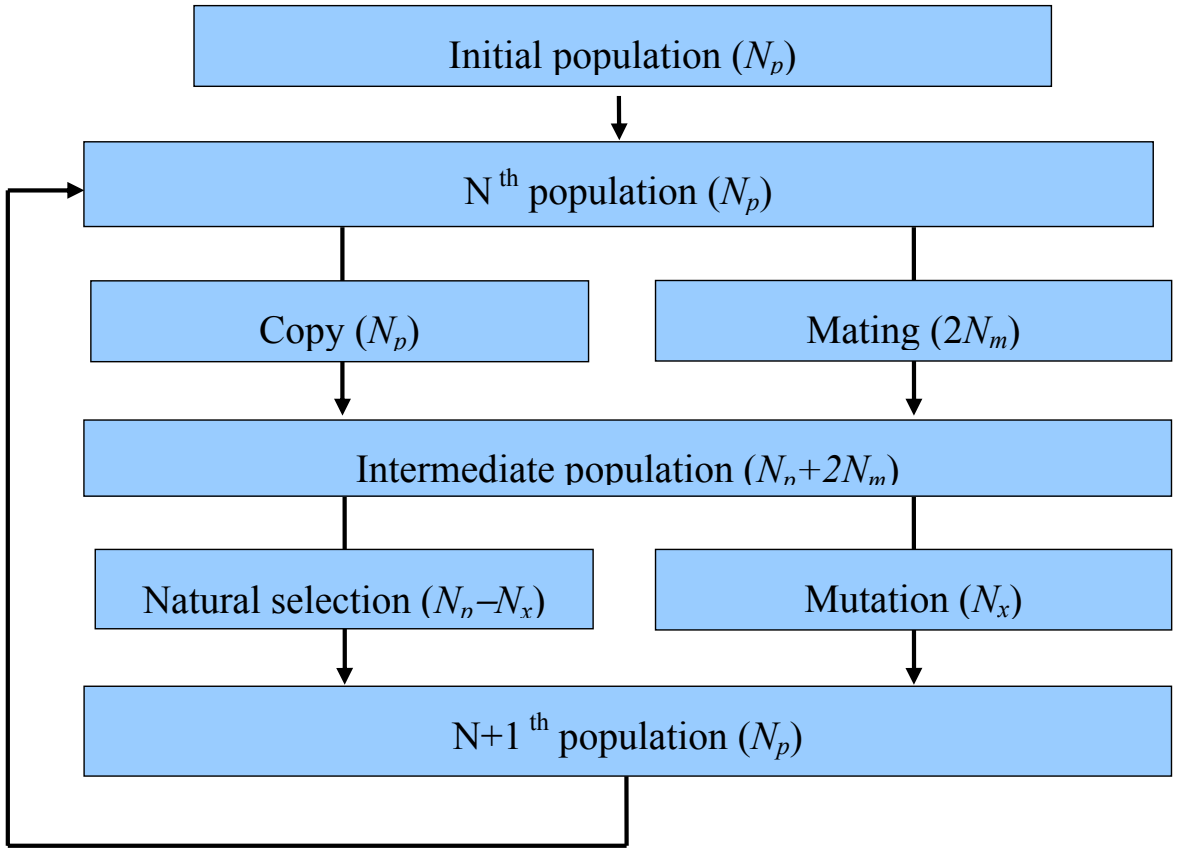

Figure 1: Procedure for updating the population in GA-DS method.

Figure taken from: A 281 Tflops Calculation for X-ray Protein Structure Analysis with Special-Purpose Computers MDGRAPE-3, **SC07 November 10-16, 2007, Reno, Nevada, USA (c) 2007 ACM 978-1-59593-764-3/07/0011**

Technologies XI

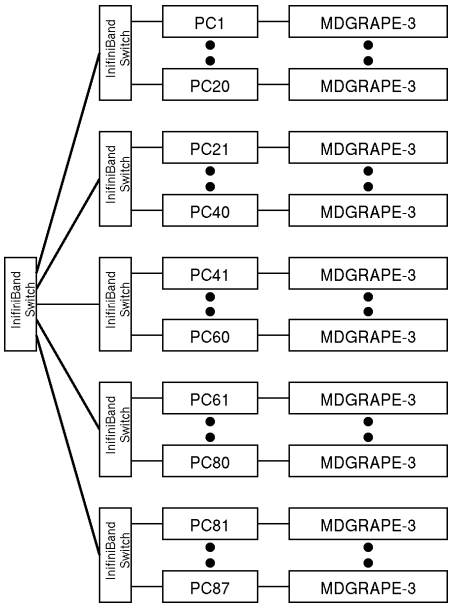

Figure 2: Block diagram (left) and photograph (right) of MDGRAPE-3 system used in the present work. It is composed of a host computer MDGRAPE-3 system. The host computer is a 174 dual-core CPU cluster of Intel Xeon processors. MDGRAPE-3 system consists of 348 boards with 12 MDGRAPE-3 chips and its peak performance is 302 Tflops.

Figure taken from: A 281 Tflops Calculation for X-ray Protein Structure Analysis with Special-Purpose Computers MDGRAPE-3, SC07 November 10-16, 2007, Reno, Nevada, USA (c) 2007 ACM 978-1-59593-764-3/07/0011

Technologies XII

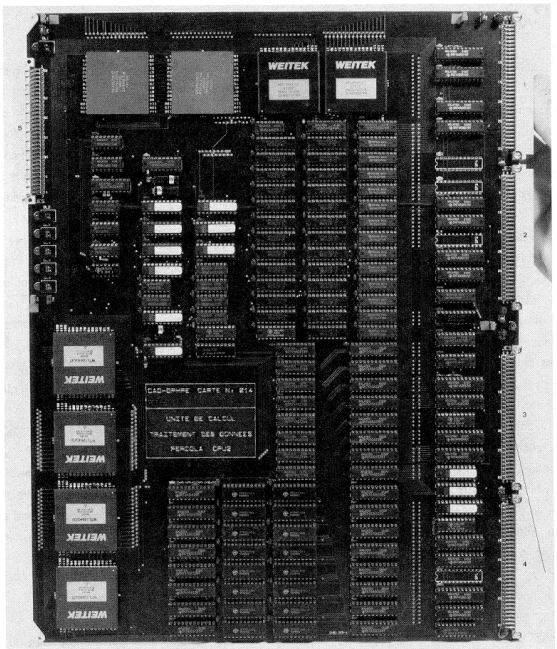

Fig. 6. Photography of the 64-bit data processing unit board (except M1), a ten layer printed circuit board of size 41 cm x 32 cm implementing the 8 Weitek's components (144-pin PGA).

Figure taken from: Jean-Marie Normand, PERCOLA : A Special Purpose Programmable 64-Bit Floating-Point Processor, Proceeding, ICS 88 Proceedings of the 2nd international conference on Supercomputing ACM New York, NY, USA 1988

GPU I

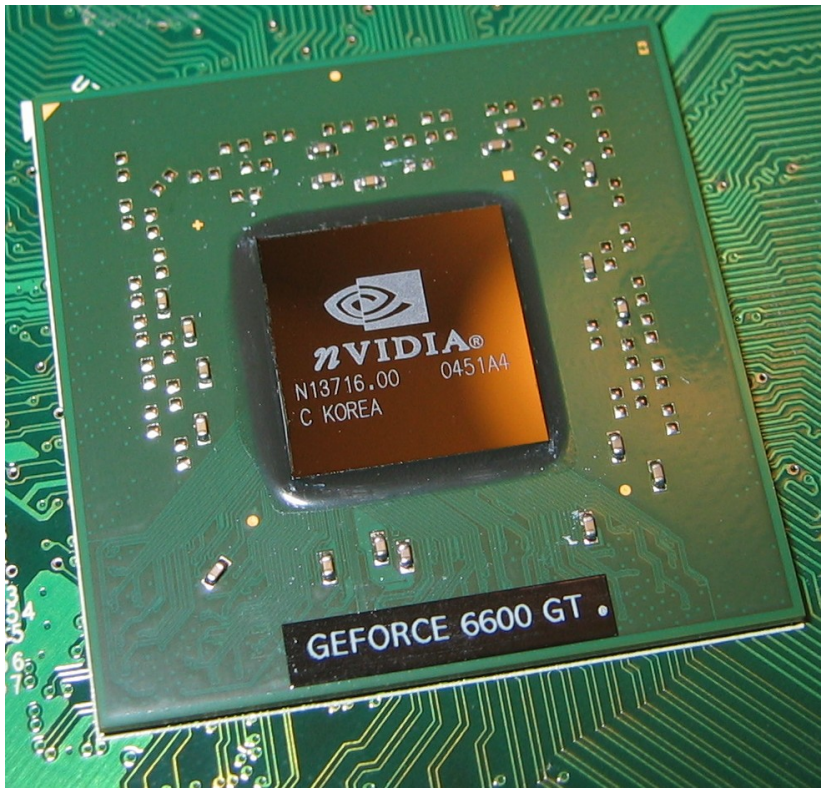

Figure taken from: http://en.wikipedia.org/wiki/Graphics_processing_unit

Graphics cards are optimized for floating point arithmetic

General Purpose Programming on a GPU

- Use CUDA/OpenCL

GPU II

- OpenCL is an open standard supported by all modern graphics card manufacturers that allows access to the graphics cards computing abilities.
- CUDA is a set of extensions on top of OpenCL specific to nVidia graphics cards

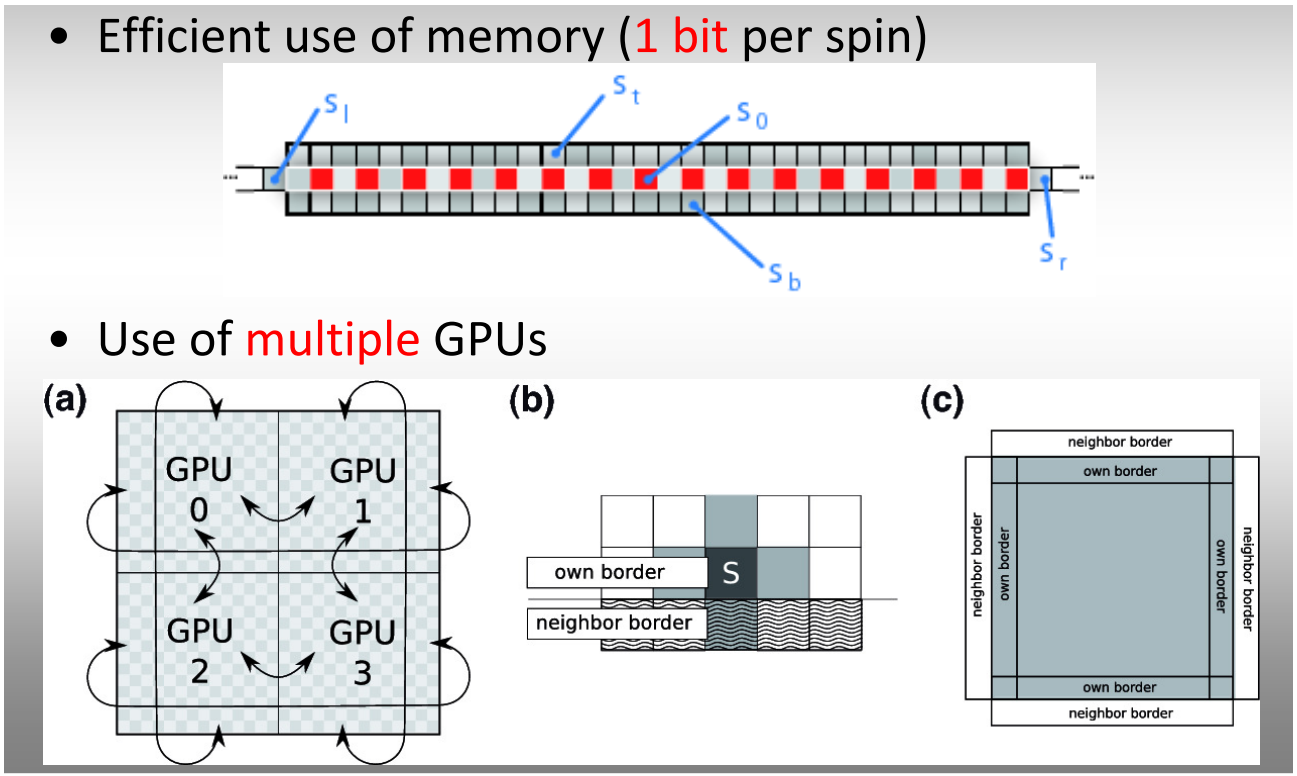

Figure taken from: The 2D Ising Model on GPU Clusters, Benjamin Block, University of Mainz, Institute for Physics

From Vector to Multi-Processor Machines I

Figure taken from: http://en.wikipedia.org/wiki/Cray-1

Transputer

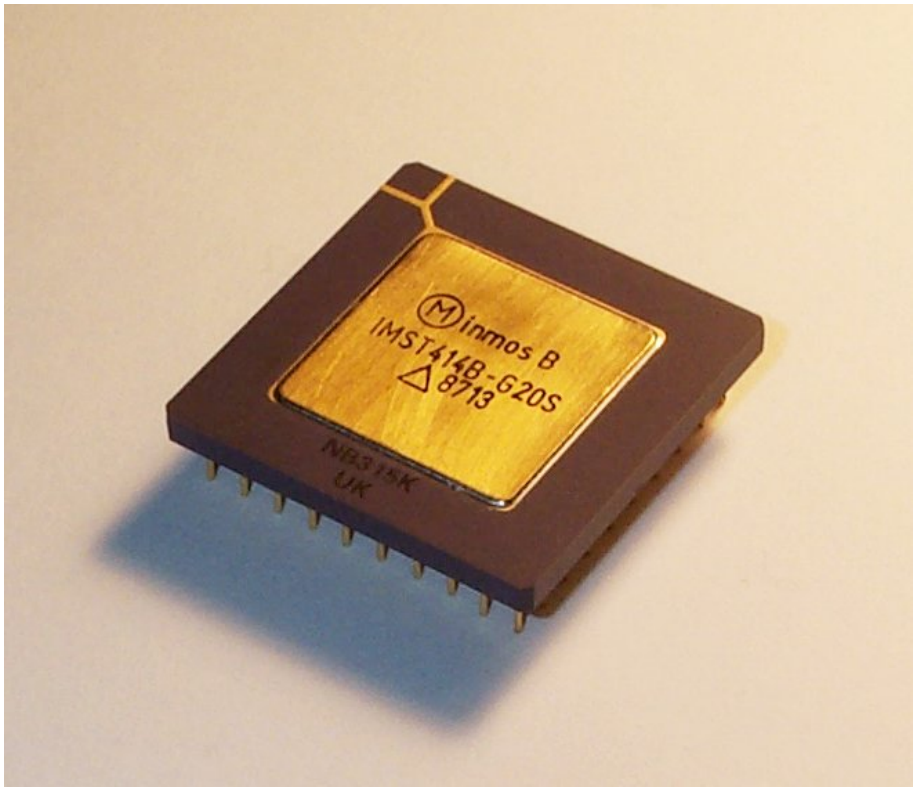

Figure taken from: http://en.wikipedia.org/wiki/Transputer

Transputer


```
SEQ  i = 0  FOR  nop
  SEQ
    pass.x[i] := x[i]
    pass.y[i] := y[i]
    pass.z[i] := z[i]

--
-- calculate forces on the particles within the processor
--
SEQ  packet = 0  FOR  MaxPackets
  SEQ
    -- send and receive the next packet
    SEQ  i = 0  FOR  nop
      PAR
        ToRight  ! pass.x[i];pass.y[i];pass.z[i]
        FromLeft ? got.x[i] ; got.y[i]; got.z[i]

    SEQ  i = 0  FOR nop
      SEQ j = 0  FOR nop
        SEQ
          xd := x[i] - got.x[j]
          yd := y[i] - got.y[j]
          zd := z[i] - got.z[j]
```

Tianhe-1A

Figure taken from: http://supercom.org/tag/supercomputer-tianhe-1a/

The APE machine

Table 1
The family of APE processors. The year in parenthesis is the time when the project was concluded. Physics runs in general have started quite earlier on prototypes or small scale machines.

	APE(1988)[1]	APE100(1993)[2]	APEmille(1999)[3]	apeNEXT(2004)[4]
Architecture	SISAMD	SISAMD	SIMAMD	SPMD
Number of nodes	16	2048	2048	4096
Topology	flexible 1D	rigid 3D	flexible 3D	flexible 3D
Memory	256 MB	8 GB	64 GB	1 TB
Registers (Word Size)	64(32)	128(32)	512(32)	512(64)
Clock speed	8 MHz	25 MHz	66 MHz	200 MHz
Peak speed	1 GFlops	100 GFlops	1 TFlops	7 TFlops

Figure taken from: The apeNEXT project, **Nuclear Physics B - Proceedings Supplements Volume 140, March 2005, Pages 176-182 LATTICE 2004 - Proceedings of the XXIInd International Symposium on Lattice Field Theory**

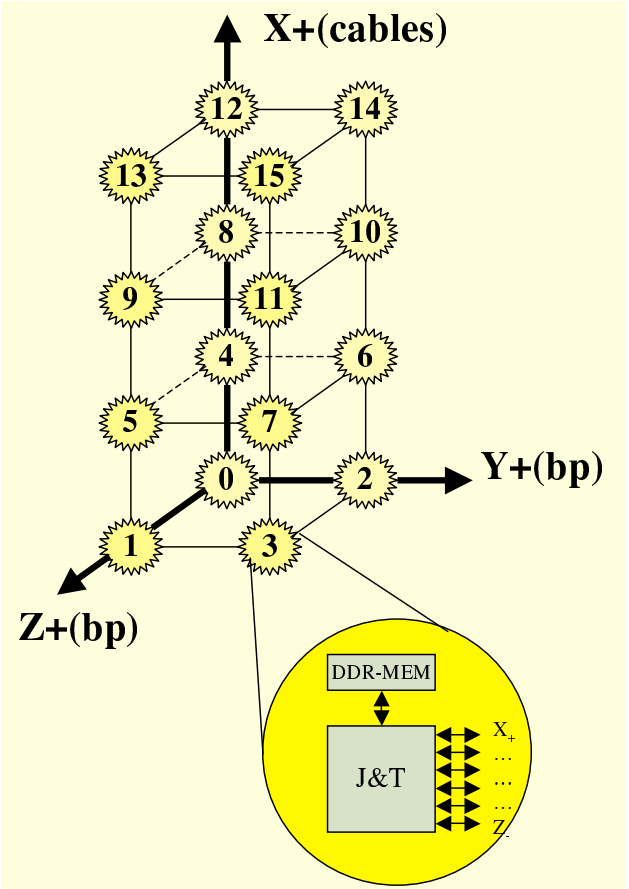

Figure 4. The apeNEXT architecture I.

Figure taken from: The apeNEXT project, **Nuclear Physics B - Proceedings Supplements Volume 140, March 2005, Pages 176-182 LATTICE 2004 - Proceedings of the XXIInd International Symposium on Lattice Field Theory**

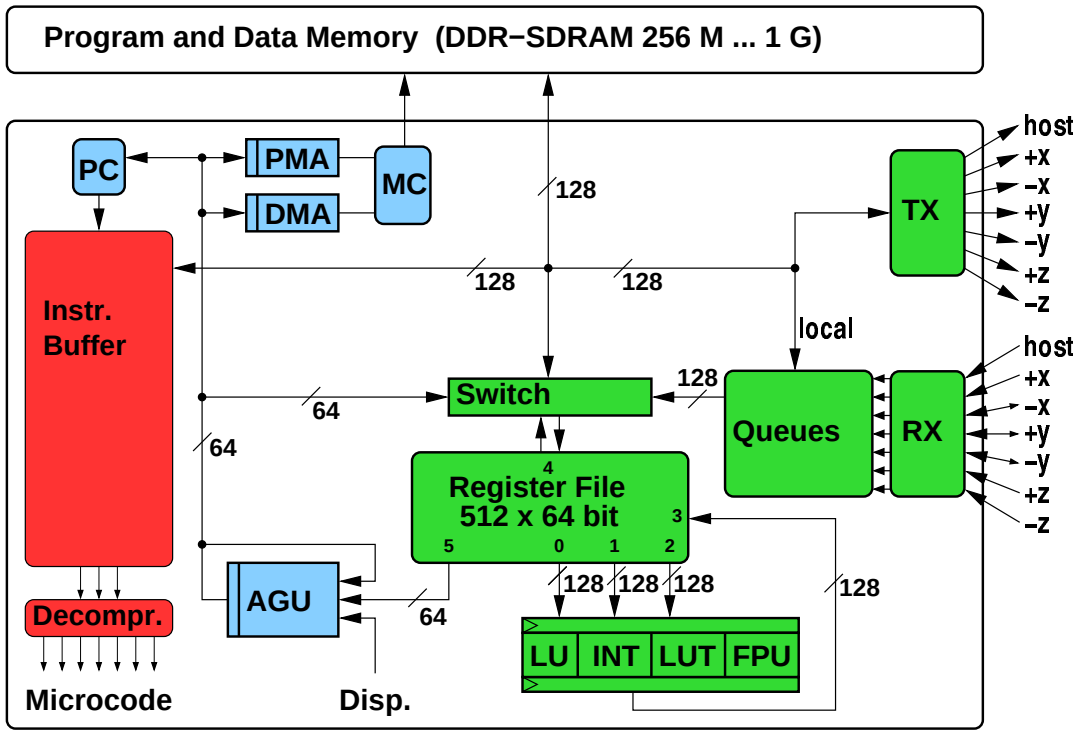

Figure 3. Block diagram of the apeNEXT processor.

Figure taken from: The apeNEXT project, **Nuclear Physics B - Proceedings Supplements Volume 140, March 2005, Pages 176-182 LATTICE 2004 - Proceedings of the XXIInd International Symposium on Lattice Field Theory**

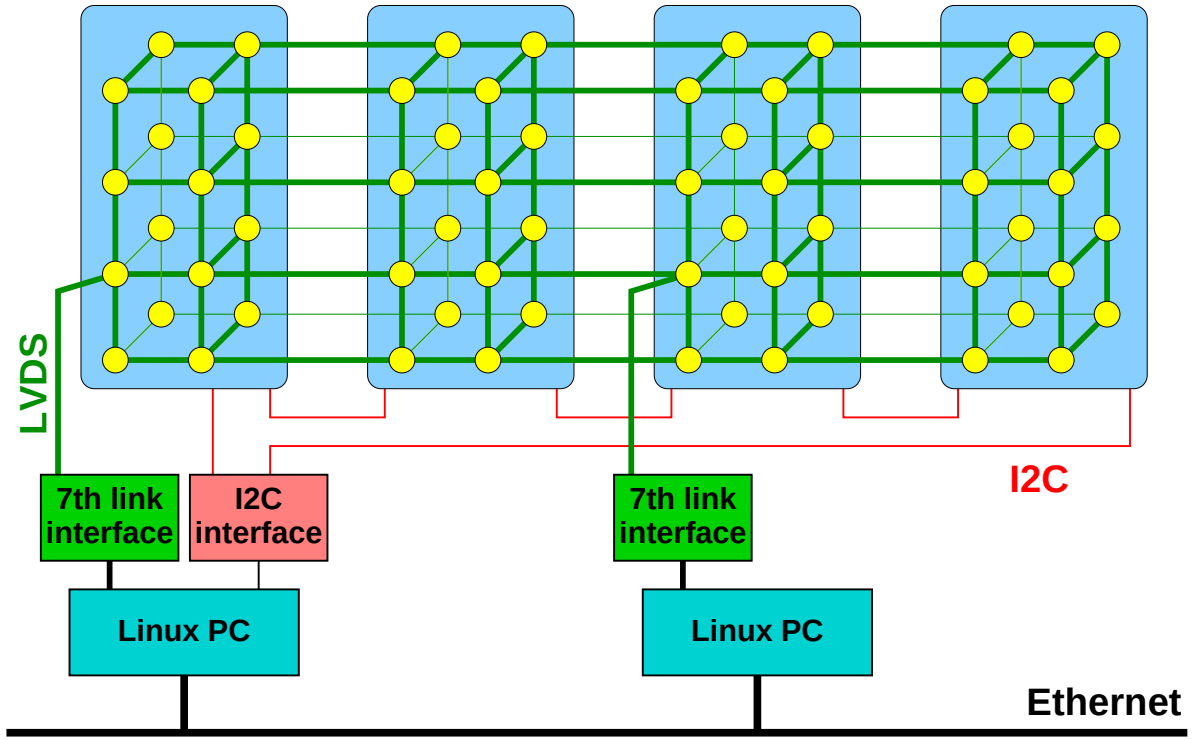

Figure 5. The apeNEXT architecture II.

Figure taken from: The apeNEXT project, **Nuclear Physics B - Proceedings Supplements Volume 140, March 2005, Pages 176-182 LATTICE 2004 - Proceedings of the XXIInd International Symposium on Lattice Field Theory**

Literature I

[1] H. J. Hilhorst, A. F. Bakker, C. Bruin, A. Compagner, and A. Hoogland, Special Purpose Computers in Physics, Journal of Statistical Physics, Vol. 34, Nos. 5/6, 1984

[2] D.W. Heermann and A.N. Burkitt, **Parallel Algorithms of Computational Science Problems** Springer Verlag, Heidelberg 1990

[3] K. Binder and D.W. Heermann, **The Monte Carlo Method in Statistical Physics: An Introduction** Springer Verlag, Heidelberg 1988

[4] D.W. Heermann, **Computer Simulation Methods in Theoretical Physics**, Springer Verlag, Heidelberg 1986

[5] A.Hoogland, J. Spaa, B. Selman, and A. Compagner, *J. Comput. Phys.* **51**, 250 (1983)

[6] R.Pearson, J.L. Richardson, and D. Toussaint, *J. Comput. Phys.* **51**, 241 (1983)

[7]
- D.J. Auerbach, A.F. Bakker, T.C. Chen, A.A. Munshi, W.J. Paul, *Mat. Res. Soc. Symp. Proc.* **63**, 219 (1985)
- D.J. Auerbach, W. Paul, A.F. Bakker, C. Lutz, W.E. Rudge, and F.F. Abraham, *J. Phys. Chem.* **91**, 4881-4890 (1987)

[8] Jean-Marie Normand, PERCOLA : A Special Purpose Programmable 64-Bit Floating-Point Processor, Proceeding, ICS '88 Proceedings of the 2nd international conference on Supercomputing ACM New York, NY, USA 1988