Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

204

EUROCAL '85

European Conference on Computer Algebra
Linz, Austria, April 1985
Proceedings Vol. 2: Research Contributions

Edited by Bob F. Caviness

Springer-Verlag
Berlin Heidelberg New York Tokyo

This series reports new developments in computer science research and teaching – quickly, informally and at a high level. The type of material considered for publication includes preliminary drafts of original papers and monographs, technical reports of high quality and broad interest, advanced level lectures, reports of meetings, provided they are of exceptional interest and focused on a single topic. The timeliness of a manuscript is more important than its form which may be unfinished or tentative. If possible, a subject index should be included. Publication of Lecture Notes is intended as a service to the international computer science community, in that a commercial publisher, Springer-Verlag, can offer a wide distribution of documents which would otherwise have a restricted readership. Once published and copyrighted, they can be documented in the scientific literature.

Manuscripts

Manuscripts should be no less than 100 and preferably no more than 500 pages in length.
They are reproduced by a photographic process and therefore must be typed with extreme care. Symbols not on the typewriter should be inserted by hand in indelible black ink. Corrections to the typescript should be made by pasting in the new text or painting out errors with white correction fluid. Authors receive 75 free copies and are free to use the material in other publications. The typescript is reduced slightly in size during reproduction; best results will not be obtained unless the text on any one page is kept within the overall limit of 18 x 26.5 cm (7 x 10½ inches). On request, the publisher will supply special paper with the typing area outlined.
Manuscripts should be sent to Prof. G. Goos, GMD Forschungsstelle Karlsruhe, Haid- und Neu-Str. 10–14, Postfach 6380, 7500 Karlsruhe 1, Germany, Prof. J. Hartmanis, Cornell University, Dept. of Computer-Science, Ithaca, NY/USA 14850, or directly to Springer-Verlag Heidelberg.

Springer-Verlag, Heidelberger Platz 3, D-1000 Berlin 33
Springer-Verlag, Tiergartenstraße 17, D-6900 Heidelberg 1
Springer-Verlag, 175 Fifth Avenue, New York, NY 10010/USA
Springer-Verlag, 37-3, Hongo 3-chome, Bunkyo-ku, Tokyo 113, Japan

ISBN 3-540-15984-3
ISBN 0-387-15984-3

The Parallel L-Machine for Symbolic Computation

Bruno Buchberger
Johannes Kepler University
A4040 Linz, Austria (Europe)

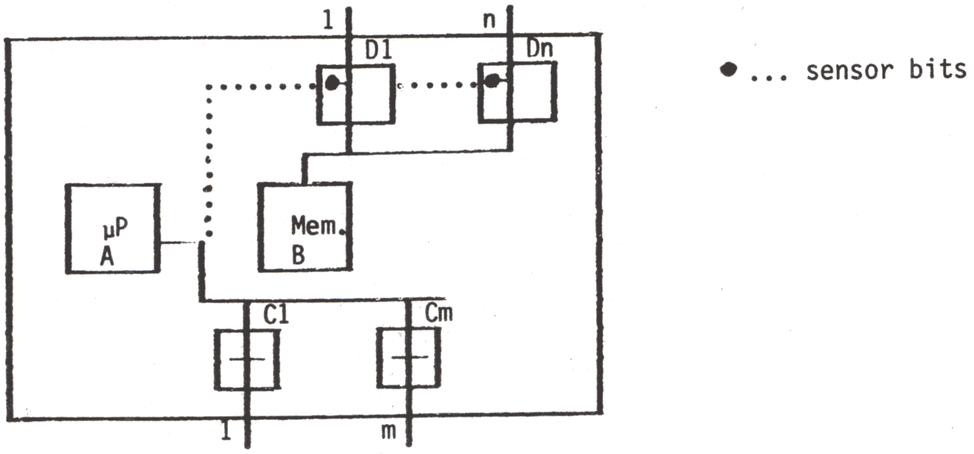
In this lecture we report on the present state of a parallel machine project pursued in the CAMP-LINZ working group since 1978 (see (Buchberger 1978)). Four basic components have been developed that can be combined flexibly for building large parallel machines of arbitrary interconnection topology (Buchberger 1983). On these machines the parallelism inherent in (symbolic) algorithms should be completely exploitable. For example, in (Bibel, Buchberger 1984) and (Aspetsberger 1985) some proposals are made for using the L-machine as a parallel inference machine. The design easily lends itself to LSI, VLSI and wafer implementation. Recent considerations are also directed towards optoelectronic implementation of parts of the concept. The L-modules defined below have been realized in traditional TTL logic already in 1978. A pilot implementation of a 8 processors by 8 memories L-machine (see below) in TTL logic has been successfully finished in October 1984, (Buchberger 1984).

The four basic components are:
A ... an (arbitrary) microcomputer with some additional hardware instructions (for accessing "sensor bits" and for opening and closing bus switches)
B ... a memory (for shared access by arbitrarily many processors)
C ... a bus switch that can be opened and closed
D ... a bus switch containing a "sensor bit" for synchronization and for handling access conflicts.

The "sensor bit" is a crucial part of the design. It is a special memory of just one bit capacity. However, it has the potential to be accessed simultaneously by two processors. Typically, one of the processors has a read access while the other has a write access. The accesses are triggered independently by the individual clocks of the two processors. The sensor bits can be programmed by special instructions.

The following two structures are the most typical parallel structures that can be formed from the components A, B, C, D.

<u>First example</u>: "L-Modules"



Many L-modules can be connected in order to form homogeneous parallel structures of arbitrary size of arbitrary, but fixed topology (for example, trees, pipe lines, rectangular nets etc.)

Second example: "The parallel L-machine" (Very rough sketch)

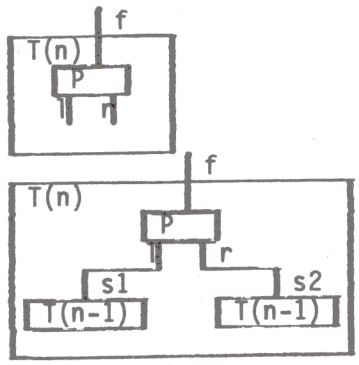
```
          B      B     B     ...
          |      |     |
A ——— C/D — C/D — C/D —...
          |      |     |
A ——— C/D — C/D — C/D —...
          |      |     |
A ——— C/D — C/D — C/D —...
...      ...    ...   ...
```

Logically and functionally this crossbar structure is totally equivalent to a full graph interconnection of m L-modules. This type of crossbar is fully extendible and does not need any central control.

On the software level, a compiler for the "L-language" (Buchberger 1984), (Hintenaus, Buchberger 1985), is presently developed. The high-level L-language provides a facility for a recursive definition of parallel cellular network structures. For example, a binary tree of L-modules P can be defined in the L-language as follows (we simultaneously present a linear and a graphical notation):

```
SUBNET Tree (f;n);
BEGIN
  IF n=1 THEN
  THEN   P(f,r,l)

  ELSE
         P(f,r,l);
         Tree(s1,n-1);
         Tree(s2,n-1);
         CONNECT l TO s1;
         CONNECT r TO s2
END;
```



References:

Apetsberger, K., 1985: Substitution Expressions: Extracting Solutions from Non-Horn-Clause Proofs. EUROCAL 85, these proceedings, 78-86.

Buchberger, B., 1978: Computer-Trees and Their Programming. Proc. 4th Coll. "Trees in Algebra and Programming", Univ. Lille, Feb. 16-18, 1978, pp. 1-18.

Buchberger, B., 1983: Components for Restructurable Multi-Micro-Processor Systems of Arbitrary Topology. Proc. of MIMI 83 (Lugano), Acta Press Anaheim, pp. 67-71.

Buchberger, B., 1984: The Present State of the L-Networks Project. Proc. of MIMI 84 (Bari), Acta Press Anaheim, pp. 178-181.

Bibel W., Buchberger B., 1984: Towards a Connection Machine for Logical Inference. (Invited talk, Int. Symp. on Fifth Generation and Super Computers, Rotterdam, December 84). Future Generations Computer Systems, 1/3, 177-188.

Hintenaus, P., Buchberger, B., 1985: The L-Language for the Parallel L-Machine (A Parallel Architecture for AI Applications). Proc. of the ÖGAI Conference, Vienna, September 1985, to appear.