A Parallel Architecture for a VLSI-Hardware-Realization of a Numerical Stable Variant of the Simplex-Method

Bernd Schütz, Reinhard Rauscher
University of Hamburg, Dept. of Computer Science
Troplowitzstr. 7, 2000 Hamburg 54, Germany
E-mail: schuetz@rz.informatik.uni-hamburg.dbp.de

Abstract. A hardware realization of the Simplex Method is represented. We customised the algorithm regarding numerical stability (arithmetics) and hardware proximity. The resulting hardware (VLSI custom chips, FPUs, RAM) of our accelerator is based on a parallel architecture with up to eight processing units.

1. Introduction

We present a hardware realization of the Simplex-Method, the most common and most efficient solution algorithm for linear programming [2]. Our objective was a fast and numerical stable implementation running in a PC/AT. This goal can only be achieved by using additional hardware. To overcome the limitation of commercially available software for such computers with our LP-accelerator hardware and to achieve a suitable numerical stability, we customized the implemented algorithm for our class of problems, also regarding alternative arithmetics. Next the algorithm has been parallized and transformed to a hardware-suited representation. By doing this a specialized tag-oriented parallel hardware architecture was developed [3]. Here the tag indicates zero-values. This was done due to the fact that typically 80 per cent of the operands in our problem-class are zero valued.

2. Architecture

The selected architecture as mentioned above consists of several PUs (Processing Units) connected to a shared memory for floating point values and controlled by the MCU (Master-Control-Unit). Each PU contains one FPU (Floating-Point-Unit), one LPU (Local-Processing-Unit), and a BSW (Bus-SWitch). The BSW realizes an interface between the LPU, the FPU and the floating point memory interface. The LCU consists of the program-ROM, register-file, integer-unit, status register and so on. The MCU only coordinates the PUs and contains the memory-management-unit. The MCU, LPU, and BSW are realized as custom chips.

In fig. 1 a typical configuration of the accelerator based on four PUs is sketched. The number of PUs can range from one to eight. Computer simulations of the architecture on the register-transfer-level showed that four floating-point-units give a suitable compromise of cost and performance. The realization of the architecture is currently under work at the University of Hamburg using the design systems from ES2 [1].

3. Results

Simulations of this architecture show that an IBM PC/AT equipped with one accelerator card gives a better performance than a SUN SPARCstation 2 running with our algorithm. Furthermore, the comparison with the Apollo demonstrates the speed-up that can be achieved by an application-specific architecture.

Problemsize (constraints x variables)	**CPU-time/Pivotstep [ms]** LPA-4	Apollo DN3000	SUN SS2
61 x 69	11.7	318	8.4
118 x 52	13.9	625	20.5
88 x 116	12.1	524	14.1
166 x 77	67.2	1899	59.3
292 x 251	201.7	4168	230.8
94 x 88	32.0	692	38.0

Table 1: Some performance analyses

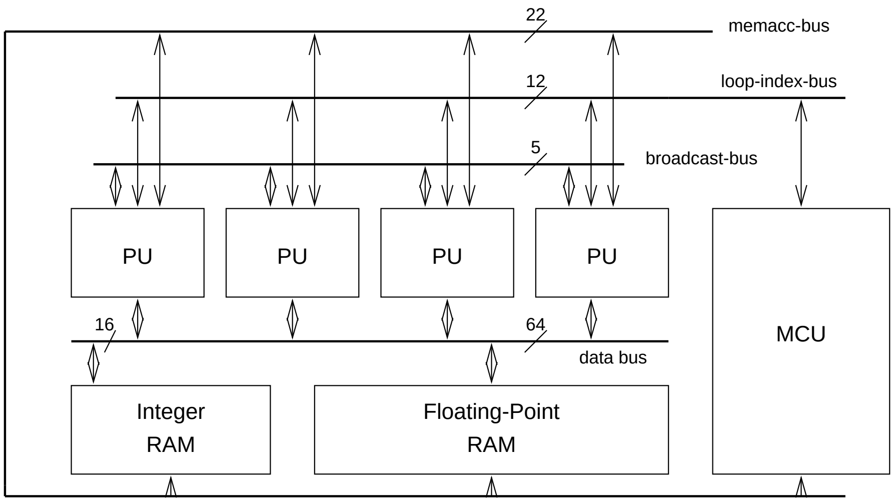

Figure 1: Overview of the accelerator architecture

References

[1] European Silicon Structures Ltd. *SOLO 2030 Release 3.1 Reference Manual*. Berkshire, 1991.

[2] Karl Heinz Borgward. *The Simplex Method - A Probabilistic Analysis*. Springer-Verlag, Berlin, 1987.

[3] A. Klindworth. *Entwurf, Simulation und Realisierung einer parallelen Rechnerarchitektur für ein revidiertes Simplex-Verfahren*. Diploma Thesis, Department of Computer Science, University of Hamburg, 1992