Topic 16
Instruction-Level Parallelism and Uniprocessor Architecture

Pascal Sainrat and Mateo Valero

Co-chairmen

Research in Instruction-Level Parallelism (ILP) is concerned with architectural innovations in the processor to expose parallelism between the execution of instructions. Of course, the relationship with the research on the memory hierarchy and on compiler optimisation techniques is very strong. Another point is that such a research needs tools to simulate the mechanisms. Thus, researchers have to develop their tools. Such a tool is detailed in the paper on code cloning tracing by Lafage et al from IRISA.

Most of these topics are represented in this workshop although there are no papers on the lower levels of the memory hierarchy.

The memory hierarchy, and particularly, the first-level cache is highly related to ILP research since superscalar processors place higher demands on it for obtaining more instructions and more data per cycle. In addition to the requirement of higher bandwidths, latency is also an important issue. One way to reduce the latency is prefetching as proposed by Chi and Yuan. Another issue is the way the cache is managed. Software can afford hints for a better management, which might result in a good speedup as in the paper of Lebeck et al.

A big and old deal is what should be in the hardware and what should be left in the compiler. Returning to simpler processors while leaving part of the job to the compiler might arrive in the future. Thus, we should care of compiler studies. Moreover, compiler studies might have an impact on the architecture. The papers by Norris, Fenwick and Genius, Lelait concern compiler optimisations. The first one deals with register allocation that can have a great impact on the reordering of instructions while the second one apply techniques of register allocation to the data in memory in order to improve the use of the cache. The VLIW architecture highly depends on the quality of the compiler. The paper of Ebcioglu et al gives encouraging results for such an approach.

Increasing the ILP is, at last, limited by data dependencies. To overcome these dependences, we should predict the results of instructions. A sophisticated value predictor is proposed in the paper of Pinuel et al. However, predicting values need to recover efficiently a normal state when a misprediction occurs. Soti studies recovery in its paper. At a longer term, asynchronous processors might provide a solution to the problems of clock distribution on very large chips as expected in a dozen years. But, making processors asynchronous might need new architectural ideas as in Pessolano's paper.

Finally, the most important and most studied problem concerns control dependences which have a dramatic impact on performance. Predicated execution

P. Amestoy et al. (Eds.): Euro-Par'99, LNCS 1685, pp. 1241–1242, 1999.

reduces the problem by removing control dependences. Embedded processors are, particularly in Europe, a hot topic. Using predication in embedded processors might be useful when performance is an issue as explained in the paper by Connors et al. A more classical approach is the prediction of the outcome of branches and many schemes have been proposed. A new approach where predictors are cascaded is proposed by Driesen and Hözle. The last paper of this topic is also related to the branch problem. It shows that executing a mispredicted path does not always result in a useful prefetching as suggested previously.

It was very difficult for the scientific committee to choose among the good papers that have been received. We hope you will share our enthusiasm for the papers that are presented here.