Marcelo Pires Módica

"O Modelo de Arquitetura com Capacidade de Execução Condicional Condex-I e seu Compactador de Código"

The performance improvement of computers can be achieved for the exploration of Instruction Level Paralellism (ILP), which identify, in an application program, the instructions which can be executed in paralell. Processors with this characteristic have a lot of independent function units wich can execute different instructions of the same program. These machines can be classified as Superscalars or as Very Long Instruction Word (VLIW), according to the implementation level of algorithm that responds to the scheduling of the instructions.

In Superscalars architetures, the scheduling is directly done in hardware, acting during the execution of an application program (dynamic scheduling). In VLIW machines, the scheduling is done by the software supporting the architecture (static scheduling or compaction). So, the scheduling is done during the fase of code production.

A present problem in these architectures is related to the branch instructions. Nowadays, the solutions for performance loss originating in branch instructions are based in technics like: speculative execution, branch prediction, trace scheduling, and others.

A VLIW machine model developed to the ILP exploration is the CONDEX model, that provides the conditional execution of instructions, breaking the basic blocks through the treatment of control dependences. The CONDEX-I model, object of this work, aproaches the CONDEX architecture model of a real machine.

This one makes instruction compaction independent of the condition that can be satisfied to the execution. The CONDEX-I gets the instruction set of SPARC architecture, and the code wich is submited to CONDEX-I simulator, is made by a C language compiler.

The SPARC instruction set was changed and in this way the instrction "branch" gives values to the conditional registers of the model and the new instruction "combime" allowthe compaction of branch instructions in chains.

As a part of this project, it was made an automatic code compactor in C language to the CONDEX-I model. The process of code compaction developed, acts in branch instructions and detecting on the program to be realized, four kinds of language program structures. The process of code compaction is different to each instruction block because its execution presents own charecteristics.

It was selected 11 programs, 2 of then belongs to a SPEC benchmarks set and used 16 different configurations of machine to realize the practice. Through the analysis of results, we noted a reduction in number of long instructions of compact program to 9.5% of sequencial similar, and still 98.6% of ocupation of long instruction fields. Part of

compact programs was submited to a CONDEX-I machine simulator. In these tests we got speed up of 2.18.