Thesis Overview:

Reducing Branch Misprediction Penalty through Confidence Estimation

Juan Luis Aragón
Universidad de Murcia (SPAIN),
Dept. Ingeniería y Tecnología de Computadores
Advisors: José González and Antonio González
February 25, 2003
jlaragon@ditec.um.es

Control dependences are one of the major limitations to increase the performance of current processors. A branch instruction supposes an interruption of the sequential flow of instructions traversing the pipeline because the next instruction address is unknown until the branch is executed. However, the fetch stage should introduce the successor instruction following the branch as soon as possible in order to maximize processor performance. Control speculation is employed in order to achieve this goal, predicting the outcome of the branch, guessing the successor address and speculatively starting the execution of those instructions from the predicted path. But despite the important benefits provided by branch prediction schemes, there are many mispredicted branches. This means that the pipeline is filled with many wrong path instructions, being necessary flushing the pipeline and restoring the correct state of the processor.

The delay between the time the branch misprediction is discovered and the processor starts fetching the correct path is known as *branch misprediction penalty*. This penalty results in performance degradation and also in an increase in energy consumption, since many useless instructions are processed and executed. Furthermore, many current processors have designs targeted at very high clock frequencies which leads to longer pipelines (e.g. more than 20 stages in the Intel Pentium 4). In such processors, the branch misprediction problem becomes even more crucial because branches take longer to be resolved and correct instructions take longer to reach execution after a misprediction.

The goal of this Thesis is reducing the global penalty associated to branch mispredictions, in terms of both performance degradation and energy consumption, through the use of confidence estimation. The reduction of this global penalty has been achieved, firstly, by increasing the accuracy of branch predictors, next, by reducing the time necessary to restore the processor from a mispredicted branch, and finally, by reducing the energy consumption due to the execution of incorrect instructions. All these proposals rely on the use of confidence estimation, a mechanism that assesses the quality of branch predictions by means of estimating the probability of a dynamic branch prediction to be correct or incorrect.

The first proposal of this Thesis is the *Branch Prediction Reversal Unit* (*BPRU*), a mechanism that selectively reverses those branch predictions likely to be mispredicted in order to improve branch prediction accuracy. This mechanism relies on the fact that many branch mispredictions can be avoided if they are selectively reversed based on some processor parameters. The novelty of this proposal is the inclusion of data values to assign confidence to branch predictions, attempting to use different information from that employed by the branch predictor (usually branch history and branch PC). For this reason, the first proposal considers the use of an underlying branch predictor based on data value prediction.

The second proposal generalizes the *BPRU*, improving its functionality with a twofold objective: first, that it could be used as an independent confidence estimator, and second, continuing with the application to branch inversion, that it could be used in conjunction with any correlating branch predictor in a hybrid scheme as an effective approach to increasing the original branch prediction accuracy.

The third proposal of this Thesis is *Dual Path Instruction Processing* (*DPIP*), a mechanism that alleviates the penalties introduced by branch mispredictions by attempting to maintain a high execution throughput immediately after a misprediction. The proposal fetches, decodes and renames instructions from the alternative path for low confidence predicted branches, at the same time as the predicted path is being executed. In addition, alternative path instructions can be *pre-scheduled* in an estimated execution order. Thus, after a misprediction, a high number of instructions from the alternate path can be immediately issued to execution, achieving an effect similar to very fast re-filling of the window. The key point this proposal is the balance between complexity, cost, and performance. In particular, complexity is significantly reduced with respect to multiple path execution proposals.

The last proposal of this Thesis is *Selective Throttling*, a mechanism that depending on the confidence degree assigned to branch predictions dynamically applies a different power-aware heuristic. We propose throttling at three different levels: fetch unit, decode unit and selection logic. Again, confidence estimation is used to assign the appropriate level of throttling. For those branches likely to be mispredicted, aggressive throttling will be applied. On the other hand, when the estimator is not quite sure about the confidence level of the prediction, less aggressive techniques, both in terms of power reduction and performance degradation, are used. The goal of this proposal is to obtain an optimal tradeoff between power and performance.