Kilo instruction processors

Mateo Valero,
UPC Barcelona

A promising approach for dealing with very long latency memory accesses (cache misses to main memory) is to dramatically increase the number of in-flight instructions in an out-of-order processor. Current processors commit instructions in program order. Consequently, a huge quantity of resources are needed to maintain thousands of instructions in flight. We need to do research in new techniques oriented to better use of resources. We observe that many inefficiencies can be eliminated if we change the model of in order commit of instructions. We need to design processors that support some form of out-of-order commit of instructions. But, of course, we also need to maintain precise exceptions.

To implement out-of-order instruction commit, we propose checkpointing a few very specific instructions with the objective of reducing and managing all the critical resources in the architecture such as ROB, Register File and Instruction Queues. We apply checkpointing, for example, to long-latency load instructions or hard-to-predict branch instructions. This mechanism of checkpointing:

- makes the existence of the classical ROB unnecessary.
- allows release the resources in an aggressive way. For example, strategic checkpointing allows an efficient implementation of early release and late allocation of registers.
- allows more intelligent management of the instruction queues.

In this talk, we will comment some of our papers describing the previous mechanisms and we will open new topics for research.