

Aaron Kobayashi
9/10/2002
One Billion Transistors

Assumptions:

There are a few basic assumptions that directly affect the plausibility of the paper. This paper is based on the eventual existence of one billion transistors on a single chip. While the concepts are applicable to chips that have fewer transistors, this paper was written in the context of a one billion transistor chip. A second assumption has to do with the greatly increased complexity of the described chip. We have to assume that the CAD tools and compilers of today can be extended and enhanced enough to be able to utilize the much more complex system described in this paper.

Main Concepts:

A chip with one billion transistors has a great deal of potential. This potential can only be unlocked with conscience consideration of the optimal utilization of this space. Due to the complex nature of processor design, decomposition (partitioning) of the processor into functional units is an absolute necessity. This paper proposes a partitioning scheme focused on maximizing the rate of instruction delivery to a single execution core. This scheme includes multiple types of caches, an aggressive hybrid branch predictor, an out of order fetch unit, and an extremely high performance execution core.

In any high performance processor, cache is essential to maintaining a high rate of instruction delivery. The proposed processor contains at least two different types of caches, the trace cache and the more traditional instruction cache. The trace cache is essential to the efficiency of execution. In contrast with the instruction cache which simply stores blocks of physically contiguous instructions for fast access by the processor, the trace cache stores logically contiguous instructions. The advantage of this approach is that the trace cache can prepare and order the next branch of instructions that the processor will execute before the processor branches and needs to execute that section of code. A trace cache will maximize the fetch bandwidth by reducing the number of incomplete fetches while simultaneously providing a system that will scale as the available cache size is increased.

To further maximize the instruction delivery bandwidth, it is necessary to have an advanced branch predictor. The penalties associated with taking incorrect branches are severe and must be avoided. The paper suggests using a branch predictor that is comprised of multiple types of predictors. This predictor would be based off of a current design known as a Multi-Hybrid predictor. These predictors are more accurate because they have the ability to utilize a predictor that works better in the short term while a more accurate predictor is "warming up".

Even with multiple caches and an advanced branch predictor, trace cache misses are unavoidable. To best deal with this unavoidable occurrence, an out of order fetch unit will be necessary. A traditional fetch unit will be forced to wait if the cache misses while an out of order fetch unit can simply fetch, decode and issue segments that follow the segment that has encountered a trace cache miss. The out of order fetch unit can also increase the accuracy of branch prediction by delaying the fetch of hard-to-predict blocks until an accurate prediction can be made. Such a system will be essential to the performance of the one billion transistor processor as it will make fewer mistakes while simultaneously increasing the overall instruction throughput.

The previously described mechanisms are all created to support what is probably the most important part of any processor, the execution core. These advanced support systems will not be effectively utilized if the execution core cannot perform a large amount of work per cycle. This paper points out that for the design to be efficient, the execution core must be able to process at least as many instructions per cycle as the fetch unit is providing. Creating a processor that can handle this amount of data is not easy. As the size of the core increases, the propagation delay of communication between functional units decreases. In order to minimize these delays, the paper suggests clustering the functional units and maintaining an individual register file for each cluster. Care will have to be taken so that the majority of values produced in one cluster are also consumed in the same cluster to limit the amount of inter-cluster communication.

Experimental Results:

The single biggest result that comes from this paper is the resolution that when one billion transistors are available on a single chip, the best implementation be a processor chip with only one execution core and an array of very advanced supporting structures.

Trace caches were shown to on average continually increase the instructions per cycle as the cache size was increased. As more transistors become available on a chip, more can be devoted to increasing the available cache size.

The hybrid branch predictor was also shown to continually increase in accuracy as the history size available increased. The effects of misprediction are costly and consequently as more transistors become available, the predictor can continue to become more accurate.

Such a design has been shown to be scalable even beyond one billion transistors given the scalability of trace caches, and hybrid predictor sizes.

References:

- Y.N. Patt, S.J. Patel, M. Evers, D.H. Friendly, and J. Stark, "One Billion Transistors, One Uniprocessor, One Chip," IEEE Computer, Vol. 30, No. 9 (September 1997), pp. 51-57.