

CHAPTER VII

CONCLUDING REMARKS

To sum things up, many current microprocessor architectures are approaching their limits as far as performance improvements are concerned. A major factor in improving performance, as mentioned in Chapter I, is some sort of parallelism. The problem stems from the fact that most parallelism techniques employed by current RISC and CISC architectures rely on hardware for parallel scheduling. Because of this, hardware is growing so complex it is presenting major challenges for further advancements in performance.

EPIC is an architecture which seeks to address the hardware complexity problem by placing the burden of parallel scheduling on the compiler. The major three principles which EPIC relies on are speculation, predication, and explicit parallelism. Speculation comes in a couple forms, data speculation and control speculation. Both of these aim to increase performance by allowing the compiler to speculate as to what is likely to happen soon, such as which path a branch might take. Predication is a technique used to eliminate some branches. This involves tagging each instruction with a five-bit pointer to a Boolean predicate bit, and only executing the instruction if the specific predicate is true. Explicit parallelism means that the compiler will explicitly tell the hardware how to execute instructions in parallel. This is in contrast to implicit

parallelism, meaning that it is not known how instructions will be executed in parallel, it will be implicitly decided by the hardware.

The Intel Itanium processor is an actual implementation of the EPIC architecture. The Itanium is a six issue, 64-bit processor that can fetch and execute six instructions per clock cycle at peak performance. There are well over 300 registers available on the processor, in contrast to usually around 32 seen in RISC architectures. Itanium also provides features such as register framing, static branch prediction, dynamic branch prediction, and an x86 emulation mode.

All in all, EPIC seems to be a very promising architecture for future microprocessors. Although EPIC seems to not be catching on too quickly at the moment, I believe this will change relatively soon. Remember, current architectures are only approaching their limits concerning performance improvements, they have not reached them yet. When these limits are finally reached, I have a feeling more and more EPIC implementations will appear in the marketplace. From now and into the short future, though, EPIC will probably only appear in environments where computing power is needed most, like scientific applications or database servers.