Experiences with Compiler Support for Processors with Exposed Pipelines

Field programmable gate arrays, FPGAs, have become an attractive implementation technology for a broad range of computing systems. We recently proposed a processor architecture, Tinuso, which achieves high performance by moving complexity from hardware to the compiler tool chain. This means that the compiler tool chain must handle the increased complexity. However, it is not clear if current production compilers can successfully meet the strict constraints on instruction order and generate efficient object code. In this paper, we present our experiences developing a compiler backend using the GNU Compiler Collection, GCC. For a set of C benchmarks, we show that a Tinuso implementation with our GCC backend reaches a relative speedup of up to 1.73 over a similar Xilinx Micro Blaze configuration while using 30% fewer hardware resources. While our experiences are generally positive, we expose some limitations in GCC that need to be addressed to achieve the full performance potential of Tinuso.