Performance-Constrained Pipelining of Software Loops onto Reconfigurable Hardware

Greg Snider

Mapping high-level code onto parallel computers goes back to the early days of computing. Mapping such code onto reconfigurable hardware is a more recent affair. This is one of the first papers in the field, targeting compilation techniques for the creation of accelerated loop compute pipelines, with the ability to trade-off the degree of acceleration according to power, energy and/or area constraints.

The paper considers the creation of compute pipelines by addressing two subproblems: allocating of necessary resources to implement the computation, and scheduling/orchestrating resources in time and space to efficiently implement the overlapped iterations. The proposed "Retimed Modulo Scheduling" algorithm adapts the classic Modulo Scheduling το handle pipelined hardware particularities, handling both the allocation and scheduling subproblems. Moreover, it can utilize both pipelined and unpipelined components, it performs optimizations such as schedule compaction, retiming, and flip-flop reduction, and allows tradeoffs between execution time, power, energy and area.

The tool was used to map a Viterbi decoder written in C onto Xilinx FPGAs with different optimization goals, showcasing the potential trade-offs between resource usage for performance in the form of coarser or finer grain pipeline stages. While the total energy differences were modest, the difference in resource usage between the high- and the low-performance implementations was almost 3x, highlighting the effects of the design-space exploration made possible by the use of this compiler.

The paper was well received and has attracted over 160 citations. Optimizations such as retiming are already available in the current FPGA cad tools and their use has been propelled by high-level synthesis tools, but this paper was well ahead of this wave.

Endorsement by: Dionisios Pnevmatikatos, Professor, National Technical University of Athens