Survey of C-based Application Mapping Tools for Reconfigurable Computing

I. Troxel[1], V. Aggarwal, B. Holland, A. Jacobs, R. DeVille, and A. George
High-performance Computing and Simulation (HCS) Research Laboratory
Department of Electrical and Computer Engineering, University of Florida
Gainesville, FL 32611-6200

ABSTRACT

Researchers in areas such as cryptology, video compression, etc. have achieved unprecedented speedups using PLDs such as FPGAs by means of highly efficient bit-level manipulation, selectable precision, a flexible set of available resources, and a high degree of hardware parallelism. However, in order to realize such improvements, applications that were originally designed for fixed-width, pipelined, general-purpose processors must typically be redesigned with FPGAs in mind. Until recently, the task of mapping applications onto hardware has been painstakingly performed using Hardware Description Languages (HDLs) such as VHDL or Verilog and other artifacts from integrated-circuit design. However, porting complex applications of interrelated functions from high-level languages such as C or FORTRAN to an HDL while extracting inherent parallelism and other optimizations has proven extremely challenging, especially for application developers without extensive hardware design experience.

One of the fundamental issues limiting the acceptance of FPGA-based reconfigurable computing (RC) in the mainstream of high-performance computing, from embedded systems to supercomputers, is the availability of an effective programming model and set of tools that provide a reasonably straightforward and efficient transition from legacy code on traditional processors to hardware design for RC. Recently, vendors and researchers have developed a variety of application mapping tools to bridge the gap between the hardware- and algorithm-design realms and ease the transition from traditional processors to FPGA-accelerated systems, including several new tools in the past year alone. Such tools attempt to provide a high-level language (HLL) programming model or graphical design capture through which to produce synthesizable code with minimal redesign effort on the part of the application developer. This paper presents a survey of application mappers based on C and compares several of these HLL tools in terms of design philosophy, ease of use, etc. Additionally, a few common benchmark designs are created using several of the application mappers (e.g. Handel-C, Impulse-C, etc.) to begin to compare the tools on the basis of design size, performance, and development time. The paper also presents preliminary coding experiences and lessons learned.

[1] Corresponding author, email: troxel@hcs.ufl.edu, telephone: 352-392-9046.