THE HARVEST COMPILING ENVIRONMENT: AN ANSI-C COMPILER FOR HIGH LEVEL SYNTHESES IN THE FPGA SCENARIO

Alessandro Marongiu and Paolo Palazzari,
{a.marongiu, p.palazzari}@ylichron.it, tel. +39-06-3048 3167
Ylichron Srl - c/o C.R. ENEA Casaccia, Via Anguillarese, 301, 00123 Rome (Italy)

FPGA computing is always thought as a media to dramatically improve computational performances. The real obstacle to its widespread diffusion is primarily due to the lack of compiling tools which allow to use common specification languages (like the ANSI C); on the contrary, FPGAs have to be programmed either through very low level HDL languages (such as VHDL or Verilog) or through some non standard languages which are dialects derived from the C but which are very far from the standard C-language: in both the cases, the effort to develop an application cannot be kept and reused to port the same application onto different compiling environments. In order to overcome previous drawbacks, Ylichron developed a compiling chain, the Harwest Compiling Environment (HCE), which allows to specify algorithms to be mapped onto FPGAs through standard C programs: as a consequence, no special skills are required to access the power of FPGA computing and no special efforts have to be spent to learn proprietary languages.

HCE is a set of compilation tools which transform an ANSI C program into an equivalent, optimized VHDL ready to be compiled and run onto a prefixed target board (the DRC blades, hosted by the Cray XT5h systems, are among the supported targets). The HCE, which is embedded within the VisualStudio IDE, adopts the following design flow:

1- the starting C program is debugged and tested through the standard C tools;
2- once the code is debugged, it is transformed into a Control and Data Flow Graph (CDFG) (or into a System of Affine Recurrence Equations (SARE) in presence of regular iterative programs) which is analysed to individuate a parallel architecture which fits with the parallelism of the computation;
3- the CDFG/SARE is scheduled onto the parallel architecture in order to minimize its time completion;
4- the VHDL code is generated; such a code enforces the scheduling (3-) on the parallel architecture (2-) and instantiates all the necessary interfaces to activate the code from an hosting node.

HCE, currently freely distributed in alpha version to a selected list of alpha testers, is scheduled to be commercially released by the end of march 2008.

The presentation will be organized to firstly introduce the HCE philosophy and the underlying computational models (CDFG and SARE); then the structure of the SW nodes and of the corresponding HW modules will be presented, followed by the description of how computation hierarchy is managed and how the mapping from SW to HW is performed. The SW compile-time optimizations, as well as the post-mapping HW optimizations, will be shortly described.

During the talk, as explanatory example, the design of a FPU for vector of complex data will be considered: different coding styles (at the C level of abstraction) will be shown and their impact on actual performances (measured in terms of FSM complexity, synthesis time and area of the final design) will be discussed.

Finally, they will be examined the points that are left open in the present version of HCE (how to explore the space of architectures which can efficiently support the specified computation, how to merge CDFG and SARE computational models, how to limit the complexity of FSMs which are generated by HCE, ...) and which will constitute the focus of the optimization efforts within the next HCE releases.