THE HERCULES HIGH-LEVEL SYNTHESIS ENVIRONMENT

Nikolaos Kavvadias

Ajax Compilers
Athens, Greece
email: nkavvadias@ajaxcompilers.com

Kostas Masselos

University of Peloponnese
Tripoli, Greece
email: kmas@uop.gr

ABSTRACT

HercuLeS by Ajax Compilers[1] is an extensible HLS environment that allows pluggable analyses and optimizations. It can be used for push-button synthesis from ANSI C and other source languages to custom hardware.

1. INTRODUCTION

The annual increase of chip complexity is 58%, while human designers' productivity increase is limited to 21%[2]. This technology-productivity gap can narrow through the adoption of methodologies that raise the specification abstraction level, ingeniously hiding low-level, time-consuming, error-prone details. HLS aims at eliminating human errors and shortening time-to-market by generating high-performance digital designs from high-level descriptions.

HercuLeS [1] confronts shortcomings and omissions of current HLS flows such as the lack of extensibility, the use of opaque intermediate representations (IRs), and vendor- and technology-dependent HDL code generation. In contrast to Xilinx Vivado HLS, HercuLeS uses open specifications throughout the HLS process. It exposes both its bit-accurate, typed-assembly IR named NAC (N-Address Code) and a low-level Graphviz[3]-based IR for third-party interfacing of new frontends, analyses and optimizations.

2. OVERVIEW

The HercuLeS flow is summarized in Fig. 1. Optimized C code is passed to GCC for GIMPLE dump generation. Textual GIMPLE is then processed by *gimple2nac*; alternatively the user can provide a domain-specific language (DSL) frontend for NAC generation. Core HercuLeS comprises of a frontend (*nac2cdfg*) and a purely graph-based backend (*cdfg2hdl*). *nac2cdfg* is used for SSA construction and CDFG extraction from NAC programs. *cdfg2hdl* is the actual synthesis kernel for automatic FSMD (Finite State Machine with Datapath) hardware and self-checking testbench generation.

[1] `http://www.ajaxcompilers.com`
[2] `http://www.itrs.net/reports.html`
[3] `http://www.graphviz.org`

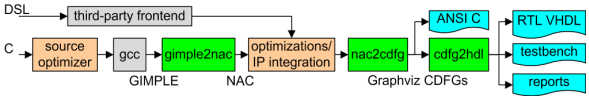

Fig. 1. The HercuLeS flow.

Various transformations can be applied at the NAC level including function call insertion to enable IP integration. *cdfg2hdl* performs operation scheduling by combining ASAP with chaining for cycle reduction. The generated VHDL code can be simulated with GHDL/Modelsim and synthesized in Xilinx XST using automatically generated scripts.

The HercuLeS v1.0.0 (2013a) distribution includes a GUI to make code generation, simulation and synthesis options accessible via an intuitive scheme. Then, a shell script is generated which steers these tasks transparently. The GUI also includes an embedded results browser.

3. BENCHMARKS

Fully-automatic synthesis of a video game to custom, FPGA-based, hardware realization will be presented. In this context, ANSI C I/O and graphic primitives are automatically mapped to efficient hardware entities. In addition, the HLS of GNU multi-precision integer[4] programs will be showcased such as the implementation of a spigot algorithm for π digits calculation[5]. A user-defined GMP API frontend generating NAC is used in order to efficiently accelerate number-theoretical GMP programs.

4. CONCLUSION

HercuLeS delivers a contemporary HLS environment that can be comfortably used for algorithm acceleration by predominantly software-oriented engineers. For the more experienced designers, it allows for developing value-adding domain-specific extensions.

5. REFERENCES

[1] N. Kavvadias and K. Masselos, "Automated synthesis of FSMD-based accelerators for hardware compilation," in *Proc. IEEE 23rd Int. Conf. on Application-Specific Sys., Arch. and Processors*, Delft, The Netherlands, Jul. 2012, pp. 157–160.

[4] `http://gmplib.org`
[5] `http://benchmarksgame.alioth.debian.org`