The HercuLeS HLS environment

Nikolaos Kavvadias
nkavvadias@ajaxcompilers.com

CEO, Ajax Compilers,
Athens, Greece
www.ajaxcompilers.com
Technology scales exponentially but humans do not.

Nikolaos Kavvadias
nkavvadias@ajaxcompilers.com

The HercuLeS HLS environment

Source: SEMATECH
The 2020 digital system: Reducing power by increasing abstraction

Near-Term System-Level Design Technology Requirements

ITRS predictions (2011-2012) per level of abstraction

% impact on total power reduction

- Physical
- Circuit
- RTL
- Architectural
- Behavioral

Nikolaos Kavvadias nkavvadias@ajaxcompilers.com

The HercuLeS HLS environment
The 2020 digital system: Highly-accurate estimators are needed

Accuracy of high-level synthesis estimates (performance, area, power, costs)

ITRS predictions (2011-2012)

% vs measurements

Nikolaos Kavvadias nkavvadias@ajaxcompilers.com

The HercuLeS HLS environment
The 2020 digital system: Tools for all platforms

Platforms fully supported by tools

ITRS predictions (2011-2012)

% expected adoption

Hardware development is hard (but why?)

Q  These problems have been solved, right?
A  No, they have not! They are still open and relevant

■ Technology partner?
  - Your time-to-market margin is shrinking for each new product
  - You don’t have zillions of employees. Your human resources are limited
  - Eventually you will run onto your monetary ceiling
  - If you don’t act now, you will be overridden by competition which uses novel approaches

■ Investor?
  - EDA (Electronic Design Automation) is a great market
  - Steady growth of 10-15% per year
  - EDA is core technology in rapidly growing markets such as mobile, medical, space, robotics
  - Greece lags far behind; high-tech mid-scale production is strategic
High-level synthesis (HLS) comes to the rescue

- Adoption of a high-level design and synthesis methodology imposing user entry from a raised level of abstraction
  1. Hide low-level, time-consuming, error-prone details
  2. Drastically reduce human effort
  3. End-to-end automation from concept to production

HLS An algorithmic description is automatically synthesized to a customized digital embedded system

- HercuLeS: An easy to use HLS environment
- Targets both hardware and software engineers/developers
  1. ASIC/SoC developers, FPGA-based/prototype/reference system engineers
  2. Algorithm developers (custom HW algorithm implementations)
  3. Application engineers (application acceleration)
Five things we already do better than competition

1. Competition tools cannot be extended  
   - In HercuLeS new frontends (e.g. for domain-specific languages), analyses and optimizations are easy to add

2. Insufficient and opaque representations  
   - HercuLeS uses open specifications and formats  
   - Its intermediate representation is NAC, a universal typed-assembly language  
   - Program graphs expressed in the open Graphviz format

3. Automatic character I/O, dynamic memory allocation and GMP arithmetic are not supported  
   - HercuLeS is the first HLS tool to support these features

4. Mandating the use of code templates  
   - HercuLeS does not rely on code templates since it uses a graph-based backend

5. Succumbing to vendor and technology dependence  
   - The generated HDL is human-readable and vendor- and technology-independent; portable across FPGA and ASIC

Nikolaos Kavvadias nkavvadias@ajaxcompilers.com  
The HercuLeS HLS environment
The HercuLeS flow
# Features

## HercuLeS GUI
- C frontend
- Graph-based synthesis engine

## C source code optimizer
- OpenMP 4.0 support

## Multi-precision integer arith.
- LLVM export from NAC IR

## Recursion for synthesis
- CDFGs in open format

## I/O synthesis
- Parallel scheduling
- Block RAM optimizations

## GHDL support
- IP integration
- IP packages

## Modelsim support
- VHDL code generation
- Testbench generation

## Xilinx ISim support
- VHDL-2008 floating-point
- VHDL-2008 fixed-point

## Register optimization
- Design for FT

## LLVM-based frontends*
- MATLAB frontend

## LLVM export from NAC IR

## HERCULES release versions

- FREE
- FULL
- BASIC
- ADDITIONS PLANNED

---

Nikolaos Kavvadias nkavvadias@ajaxcompilers.com

The HercuLeS HLS environment
The optimization space of HercuLeS

Arithmetic
- constant multiplication optimization
- constant division optimization
- optimization of polynomial expressions
- low-level superoptimizations

Loop-oriented
- partial loop unrolling
- full loop unrolling
- loop coalescing
- strip mining

Restructuring
- SSA construction
- if-conversion
- peephole transformations
- basic block partitioning
- black box function insertion
- GNU MP extensions

Source language
- N-Address Code
- Graphviz CDFGs
- VHDL

Level
- sequential scheduling
- ASAP scheduling
- dead node elimination

Hardware interpretation of SSA form
- synchronous embedded memories (block RAMs)
- operation chaining

Nikolaos Kavvadias nkavvadias@ajaxcompilers.com
The HercuLeS HLS environment
IPs generated by HercuLeS

Approximations

Path planning

Pattern matching

Graphics

Matrix processing

Transforms

Filters

Spectrum analysis

Linear systems

Nikolaos Kavvadias nkavvadias@ajaxcompilers.com

The HercuLeS HLS environment
More than 10x productivity increase!

The HercuLeS HLS environment
The HLS landscape

The HercuLeS HLS environment
HLS tools comparison (Abstraction; Types; Explore; Verify; Results; Documentation; Learning)
Preliminary results against Vivado HLS 2013.2 on Virtex-6 and/or Kintex-7 (* KC705 development board; XC7K70TFBG676-2 FPGA device)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Vivado HLS</th>
<th>HercuLeS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LUTs</td>
<td>Regs</td>
</tr>
<tr>
<td>*Array sum</td>
<td>83</td>
<td>172</td>
</tr>
<tr>
<td>Bit reversal</td>
<td>67</td>
<td>39</td>
</tr>
<tr>
<td>*Radix-2 division</td>
<td>218</td>
<td>226</td>
</tr>
<tr>
<td>Edge detection*</td>
<td>246</td>
<td>130</td>
</tr>
<tr>
<td>Fibonacci series</td>
<td>138</td>
<td>131</td>
</tr>
<tr>
<td>FIR filter</td>
<td>102</td>
<td>52</td>
</tr>
<tr>
<td>*FIR filter</td>
<td>89</td>
<td>114</td>
</tr>
<tr>
<td>Greatest common divisor</td>
<td>210</td>
<td>98</td>
</tr>
<tr>
<td>Cubic root approx.</td>
<td>239</td>
<td>207</td>
</tr>
<tr>
<td>Population count</td>
<td>45</td>
<td>65</td>
</tr>
<tr>
<td>Prime sieve*</td>
<td>525</td>
<td>595</td>
</tr>
<tr>
<td>*Sierpinski triangle</td>
<td>130</td>
<td>296</td>
</tr>
</tbody>
</table>
The HercuLeS GUI

- Bundled with HercuLeS v1.0.0 (2013a) released on June 30
- Specify code generation, simulation and synthesis options
- Includes embedded results viewer

Nikolaos Kavvadias
nkavvadias@ajaxcompilers.com
HercuLeS is an extensible HLS environment for hardware and software engineers

Straightforward to use from ANSI C, generic assembly (NAC) or custom DSLs to producing compact VHDL designs with competitive QoR

Product information

- HercuLeS GUI for specifying code generation, simulation and synthesis options
- Commercial distribution: http://www.ajaxcompilers.com
- Technical details: http://www.nkavvadias.com/hercules/
- FREE, BASIC, and FULL software licensing schemes (2013.a version)
Thank you for your time and interest!