ATHENa - Automated Tools for Hardware Evaluation

Motivation
Comparison of FPGA implementations of cryptographic algorithms that is
- fair
- comprehensive: based on multiple FPGA devices and CAD software tools
- automated: all tools run in a batch mode, without user supervision
- reliable: reproducible
- does not require revealing the code: practical, acceptable for majority of designers

Previous Work
Results of timing analysis measured directly calculated based on instructions of the circuit

Optimization of memory, execution time, power, balanced speed, area, throughput
Optimization target

Software
FPGAs
Intel, AMD, Xilinx, Altera
free software tools
GNU compilers
Xilinx WebPACK, Altera Quartus Web Edition
multiple options of tools
high-level optimizations possible but not portable assembly language
IP cores, manual placement & routing

Software
FPGAs
Intel, AMD, Xilinx, Altera
free software tools
GNU compilers
Xilinx WebPACK, Altera Quartus Web Edition
multiple options of tools
high-level optimizations possible but not portable assembly language
IP cores, manual placement & routing

Software
FPGAs
Intel, AMD, Xilinx, Altera
free software tools
GNU compilers
Xilinx WebPACK, Altera Quartus Web Edition
multiple options of tools
high-level optimizations possible but not portable assembly language
IP cores, manual placement & routing

Software
FPGAs
Intel, AMD, Xilinx, Altera
free software tools
GNU compilers
Xilinx WebPACK, Altera Quartus Web Edition
multiple options of tools
high-level optimizations possible but not portable assembly language
IP cores, manual placement & routing

Proposed Solution
ATHENa – Automated Tool for Hardware Evaluation
Set of scripts written in Perl aimed at an AUTOMATED generation of OPTIMIZED results for MULTIPLE hardware platforms, currently under development at George Mason University.

The first proof-of-concept version available at http://cryptanalysis.gmu.edu/athena

ATHENa Allows Comparing
- Algorithms, e.g. candidates in the SHA-3 contest
- Architectures and implementations, e.g., basic iterative vs. unrolled, GMU implementation vs. Bochum implementation
- Hardware platforms, e.g. Xilinx Virtex 6 vs. Altera Stratix IV
- Languages and tools, e.g., VHDL vs. Verilog vs. AHDL, Synplify Pro vs. Xilinx ISE

Basic Dataflow of ATHENa

Results
Results for Hash Functions SHA-1 and SHA-2
Xilinx FPGAs

Results for Hash Functions SHA-1 and SHA-2
Xilinx FPGAs

Applications & Extensions
Short-Term Application – SHA-3 Contest
- analysis of 14 hash functions qualified to the second round of the SHA-3 contest
BLAKE, Blue Midnight Wish, Cubist, ECHO, Fujisaki, G teenager, Hamel, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, Skein
GMU students implementing, optimizing, and benchmarking all 14 candidates in Fall 2009
Comparison vs. existing optimized implementations of SHA-1 and SHA-2 standards
VHDL codes and results of analysis published at the Athena web site by December 31, 2009

Possible extensions
- standard-cell ASICs
- actual experimental measurements in hardware (power and energy consumption, latency, throughput)
- taking into account resilience to side-channel attacks
- other fields (e.g. DSP)

Conclusions
- We propose a tool for a fair, comprehensive, reliable, and practical evaluation of cryptographic hardware
- Hope to discourage naive and/or dishonest comparisons, provide transparency, and overcome objective difficulties
- The proof-of-concept beta version 0.1 available at http://cryptanalysis.gmu.edu/athena
- Subsequent versions made available as the tool matures.
- All scripts and configuration file templates will be made available in public domain (GPL) through the project web site.