Implementing an Electronic Design Automation Tool for Cryptographic Hardware using Functional Languages

Kris Aerts, Davy Wolfs, Nele Mentens

KHLim – KU Leuven
3 Trends in 1 Project Paper

- Belgian Landscape of Higher Education
- Progress in our research topic
- FP pervading common practice
  - F#
  - Lambda expressions in C#
  - Java FX
  - Java: Scala & Closure
  - ...

2/26
Higher Education in Flanders (B)
Originally ternary (French system)

- Teachers, nursing elektromechanics, ...
- Industrial sciences, Interior architect, ...
- Alpha & beta sciences

University colleges
Education

- 3y Prof
BA
- 4y Aca
BA/MA

Learning a job
Academic level

Universities
Research & education

4y-5y Aca
BA/MA

4/26
Since Bologna: moving to binary

Consequences
- More research
  - From applied to more fundamental
- More cooperation

University colleges
Education
Learning a job Academic level

Universities
Research & education

3y Prof BA
4y Aca BA/MA

4y Aca BA/MA
4y-5y Aca BA/MA

5/26
Higher Education in Flanders (B)
Teaming up locally

Nele Mentens  
PhD in 2007  
“Secure and Efficient Coprocessor Design for Cryptographic Applications on FPGAs”

Me  
PhD in 2001  
“Visto: A Declarative Methodology for Graphical User Interfaces, based on Haskell”
Our research focus: crypto

Principles

- Data confidentiality
- User authentication
- Data integrity
- Non-repudiation

Applications

8/26
What’s the problem with crypto?

A developer’s view

Developing a cool new device

Adding crypto

Intrinsic motivation

Extrinsic motivation
What makes crypto difficult?

- Complex math
- Side channel attacks
  - Both destructive and non-destructive methods
  - A business on its own
- Limited resources (on FPGA)
  - Most surface is reserved for actual app
  - And/or only small time delay is allowed
The ("a") solution

- Electronic Design Automation Tool

**User**
- selects crypto-algorithm
- defines implementation platform
- imposes time/space/security constraints

**Machine outputs:**
- VHDL
- Verilog
- HW/SW co-design
Important considerations

Electronic Design **Automation tool**

- black box
- easy to use
- implemented in Lava / Haskell
- no programming skills needed

*Functional programming pervading*
Teaming up regionally

**COSIC**

Research concentrates on

- cryptographic algorithms and protocols,
- development of security architectures for information and communication systems
- the development of security mechanisms for embedded systems.

**DTAI**

Main themes of study are in the fields of

- declarative languages,
- machine learning,
- data mining,
- and knowledge representation.
Project plan

Bottom up approach  “Think locally, act globally”

1. Large number library
   - (Brute force) design exploration

2. Crypto generator for hardware
   - Entire core instead of functions
   - Algorithms are supplied: no programming needed
   - Countermeasures against side channels attacks

3. Extending tot HW/SW co-design
Step 1: preliminary tool

Design space exploration by combining

- different data path widths
- different architectures for +, *, …
- different finite state encodings
- optimisations towards space or speed

CREA-project of KU Leuven

- Creative, high risk projects
Lava: DSL in Haskell

halfadder (a, b) = (sum, carry)
  where
    sum = xor2(a, b)
    carry = and2(a, b)

halfadder’ (a, b) = (a <#> b, a <&> b)

rippleCarryAdder (carryIn, (as, bs)) = (sum, carryOut)
  where
    (sum, carryOut) = row fullAdder (carryIn, zipp (as, bs))
It works!

Montgomery algorithm for multiplying $2^m$-bit field numbers on a n-bit data path.

- **SOS**
  Separate operand scanning

- **CIOS**
  Coarsely integrated operand scanning

- **FIOS**
  Finely integrated operand scanning

- **FIPS**
  Finely integrated product scanning
It works!

If you have the time...
+20h on a somewhat decent PC

synthMMM [256]
[8, 16, 32, 64]
[CIOS, FIOS]
[FullProdCSAARCA, FullProdCSAACSE, FullProdCSAASklansky, FullProdCSAAVhdlAdd, VhdlMult]
[NetList, Vhdl]
[Xst.Area, Xst.Speed]
[EncOneHot, EncGray]
Making it faster

- Pruning the tree
- Intelligent search algorithms
More capable

Projected algorithmic and architectural expansion of the preliminary tool.
Making it more clever

- Machine learning
- (Monadic) Constraint Programming
Making crypto cores

- Current and future security algorithms
- Counter measures against side channel attacks incorporated in all (steps of all) designs
Expanding the scope

- HW/SW co-design
Teaming up globally

- Expanding framework to other domains

- *Later...*
Conclusion

- Combination of different research fields/groups with FP as the glue in between
  - Our scope broadens from FP to declarative/logic programming with applications in crypto and hardware
- FP is not visible to the end user, but is of vital importance
  - Builds on the success of applying FP in hardware design, but removes the stress of re-educating VHDL-designers
Questions?

The end.

Or actually the beginning.