Galois Ultra Low Power
High Assurance
Asynchronous Crypto

Joe Kiniry
October 2016

Presented on behalf of Galois,
the University of Southern California (Prof. Peter Beerel), and Reduced Energy Microsystems (William Koven)
Executive Summary

- R&D performed with USC and REM has two main goals:
  - *synthesize high assurance HDL from formal specifications*, and
  - experiment with *asynchronous VLSI crypto*

- core strategy is to:
  - start with formally verified Cryptol specifications of three lightweight crypto algorithms (Simon, Speck, and AES),
  - automatically synthesize System Verilog-CSP (SVC) implementations and assurance artifacts,
  - fabricate test chips in a fairly old process (IBM 130 nm), and
  - characterize correctness, performance, and energy
Core Results

• all Simon and Speck cores operate correctly; multi-stage AES and the hardware error counter do not (there were design errors in hardware CAD; the HDL is correct)

• all but one chip operate properly (fair yield)

• measured frequency estimate (as if clocked) varies from ~300 MHz@~0.6 V to 2.1 GHz@1.35 V

• energy/bit at 140 nm ranges between 4.2 and 7.5 pJ/b at threshold

• static load varies between 0.0012 mA and 0.0029 mA

• energy varies between 0.05 mW and 10 mW and is in close alignment with (post place and route) simulation

• safe minimum operating voltage is the process threshold (~625mV)
Galois HACrypto Toolchain

**SPECIFICATIONS**
- Cipher and Protocol Specification via Standards or Research Papers
-Specification of Cipher (Cryptol) or Protocol (ProVerif, EasyCrypt, F*, FCF, etc.)
- Formal Specification of Cipher or Protocol Theorems, including Test Vectors
- Research Papers Characterizing Cipher or Protocol Properties

**EXECUTABLES**
- Cipher IMPLS
- Verilog

**SAW**
- SAWCore
- Translator
- Type Checker

**CRYPTOL**
- Interpreter

**HDL System**
- Translator
- Type Checker

**CORRECTNESS EVIDENCE**
- Runtime Verification of Theorems as Parameterized Tests
- Formal Verification of Theorems

**SAWScriptSymbolic**
- Generation of Symbolic Formulas

**SAT/SMT**
- SAT/SMT

**PERFORMANCE, ENERGY, SPACE EVALUATION**
- Runtime Verification on FPGA Test Bench
- Runtime Verification on ASIC Test Bench
- Traditional CAD Simulation

**CIPHER TEST BENCHES**
- HDL FLOW
- FPGA
- Other HDLs
- Altera/Xilinx CAD Tools
- Verilog

**Key for architecture specification**
- A
- B
- C
- dataflow from A to B
- A depends upon B

**CLUSTER OF SUBSYSTEMS**
- A
- B
- C

**Structured VC Gen**
Hardware Synthesis

• we have synthesized formally verified high performance System Verilog-CSP implementations of Simon, Speck, and AES

• the hardware synthesis pipeline is

  Cryptol → Abstract Circuit Representation → System Verilog-CSP

• Cryptol modules are mapped to CSP processes and function calls are mapped to CSP message sends

• our focus is on pipeline simplicity (for assurance) at this time; no pipeline optimizations were made to achieve high performance or low energy

• there are a variety of research opportunities with respect to secure asynchronous VLSI and platform targeting
Estimating Performance and Energy for Later Processes

• we fabricated using IBM’s 130 nm process

• our implementations, broadly speaking, use 4.2–7.5 pJ/bit at our lowest power (0.625 mA)

• a 40% reduction in area/power for each generation is generally a good estimate, but there are caveats

• other work discussed later focuses on 65 nm simulation

• consequently, we estimate that our energy use in 65 nm is 1.5 pJ/bit, and in the fJ/bit in the latest processes
Estimate Caveats

• 130 nm library we used had cells with only a single channel length and only a single Vt (threshold); by 65 nm and beyond, most libraries have cells with the same logic function (and size) but different channel lengths and different Vt's to allow for power/performance optimizations that didn't exist in our process.

• there is also a bigger reduction in power/area from something like 90 nm to 55 LP (a optical low power shrink of 65 nm offered by both Global Foundries and TSMC) and similarly a bigger jump from 65 nm to 40 LP (again an optical low power shrink of 45 nm).

• so by the time you get all the way to something like 28 LPP, you almost get another generation's worth of improvement than would be implied by 130 -> 28 LPP via just 40% per generation.
Software Synthesis

- we have also *automatically synthesized formally verified high performance software implementations* of all three ciphers

- this synthesis pipeline transforms Cryptol programs into their SAW IR representations via symbolic evaluation, and then transforms that representation directly into LLVM

- our focus in this pipeline is on simplicity (for assurance) and where the opportunities for improvement are (for security)

- there are obvious R&D opportunities wrt formally verified side channels (e.g., Almeida et al.) and platform targeting (via automatic evaluation of functional and non-functional properties)
Assurance

- assurance means providing third-party verifiable evidence that claims we make are true (in all circumstances, given any input, etc.)

  - the strength of an assurance argument ranges from “we did some code review and ran a few unit tests” to “we formally specified and verified the following properties”

- our assurance case is based upon *Literate Cryptol* specifications

  - specs are literate (in the Knuth sense) versions of NIST and IETF standards
  - specs include models, reference implementation, and many theorems
  - spec compile to, e.g., NIST PDFs and ASCII RFCs
  - specs are also interpretable as mechanized models of algorithms and protocols, thus theorems are automatically proven (about models and implementations) and/or are used to automatically generate test benches
Formal Verification

• formal verification is about proving theorems, sometimes automatically and sometimes interactively

• our theorems focus on correctness; others are about security

• we automatically prove theorems about specifications and relationships among specifications and implementations

• some example theorems include

  • a decrypt of an encrypt is what we started with

  • this optimized code behaves exactly as that reference code

  • this LLVM compiled from that C behaves exactly as specified in that Cryptol specification for all possible inputs
Formal Validation

• formal verification is only possible with a mechanized semantics

• most hardware engineers do not understand proof, though some do understand Jasper-style equivalence checking

• we synthesize complete test benches from specifications by transmuting all theorems into SVC test code and assertions

• test benches are checked using a variety of techniques available in modern CAD tools (mainly simulation, finite explicit state model checking, and equivalence checking)

• with additional resources we could write a full mechanized semantics of System Verilog-CSP and provide even greater assurance
Related Work

• comparisons to the state of the art are difficult

• other implementations are clocked, report rough energy estimates from simulations rather than measurements, and often optimize for size

• in a clocked setting, size^2 ~ energy, but in an unclocked setting, there is little relationship
## Big Picture Results

<table>
<thead>
<tr>
<th></th>
<th>Block speed (KHz)</th>
<th>Bit speed (kbps)</th>
<th>Power (mW)</th>
<th>Energy/bit (pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chip</td>
<td>Simulation</td>
<td>Chip</td>
<td>Simulation</td>
</tr>
<tr>
<td>simon128</td>
<td>1,916.80</td>
<td>1,958.15</td>
<td>250,643</td>
<td>6.826</td>
</tr>
<tr>
<td>simon48</td>
<td>3,788.40</td>
<td>4,011.18</td>
<td>192,537</td>
<td>4.149</td>
</tr>
<tr>
<td>speck128</td>
<td>3,841.90</td>
<td>4,088.88</td>
<td>523,377</td>
<td>10.204</td>
</tr>
<tr>
<td>speck48</td>
<td>6,001.20</td>
<td>6,231.51</td>
<td>299,112</td>
<td>4.489</td>
</tr>
<tr>
<td>aes128</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results at 1.2 V, using TT library for simulation, all delay lines set to minimum viable setting.

Simulation results using "SigCMin" SPEF with "MINIMUM" SDF values.
Performance

- reported maximum performance implementations in FPGA are in the 2.5 to 5.3 Mbps range (Gulcan, Aysu, Schaumont)

- reported maximum performance of other lightweight ciphers (not Simon & Speck) at 1.2 V & 65 nm process is 2 to 15 Gbps (Kerckhoff et al.)

- our (unoptimized) performance ranges from 170 Mbps (Simon 48) to 450 Mbps (Speck 128)
Simon 128/128 Delay

Delay vs Voltage (simon128)

- data collected from a single chip
- varied voltage from 0.538 V to 1.46 V
- shows internal cycle time of the core
Chip-to-Chip Variation

- tested 38 chips; 1 failed
- showing frequency of simon128 core @ 1.15 V (internal frequency is 66x faster)
Energy

- we run correctly at threshold voltage
- we are power invariant and performance scales nicely
- our energy measurements are for the encryption cells only; all the I/O pads, logic, and memory used for testing on a separate power domain
- precise numbers for pJ/bit for a few algorithms follow
- energy use in low-power scenarios is a composite of active and quiescent energy, thus our low quiescent energy (1 µA) is exciting
- we extrapolate for modern processes in the following slides
- averaged results from entire lot of chips (130 nm)

- fastest / average / slowest @ 1.255 V: 2.000 MHz / 1.964 MHz / 1.885 MHz

- highest / average / lowest @ 1.255 V: 7.67 mW / 7.43 mW / 7.11 mW
- averaged results from full lot of chips (130 nm)
- bars show min/max
- Estimated energy at 65 nm (40% reduction over two generations)
Energy Consumption

- metric: energy per encrypted bit
  - abstracts differences in key/block size as well as the internal cycles of each cipher
- original paper: “Simon has been optimized for hardware”
  - both Speck implementations consume less energy / bit
- ciphers are more efficient near their minimum voltage
  - ~3.6x decrease in energy consumption through voltage scaling (0.625 V vs 1.2 V)
Energy Consumption

- metric: energy per encrypted block
- favors smaller block sizes

![Energy per Encrypted Block Graph](image)
Power

- Speck 128/128 uses the most power during operation but is more efficient than Simon 128/128 due to fewer internal cycles.

- Similarly, Speck 48/72 uses more power than Simon 48/72.
Hardware Comparisons

- original paper:
  - area is shown in “gate equivalent”
  - 1 GE = 5.76 µm²
  - throughput measured at 100kHz
- our designs:
  - much larger, but much faster

### Table 6.2: Hardware performance for Simon and Speck.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Area (µm²)</th>
<th>Area (GE)</th>
<th>Throughput (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simon128</td>
<td>62,205</td>
<td>10,799</td>
<td>250,643</td>
</tr>
<tr>
<td>Simon48</td>
<td>41,683</td>
<td>7,237</td>
<td>192,537</td>
</tr>
<tr>
<td>Speck128</td>
<td>111,428</td>
<td>19,345</td>
<td>523,377</td>
</tr>
<tr>
<td>Speck48</td>
<td>43,547</td>
<td>7,560</td>
<td>299,112</td>
</tr>
<tr>
<td>Aes128</td>
<td>1,835,174</td>
<td>318,607</td>
<td>3,678,374</td>
</tr>
</tbody>
</table>

GULPHAAC Designs (at 1.2 V)
Lightweight on Embedded

- unclear if energy is per block or bit
- in either case, we are significantly lower (no surprise)
- ASIC vs µC implementation

Compact Implementation and Performance Evaluation of Block Ciphers in ATtiny Devices  
[Eisenbarth, 2012]
Towards Green Cryptography: A Comparison of Lightweight Ciphers from the Energy Viewpoint

[Kerckhoff et al. 2012]
Open Questions

• how much synthesis pipeline tuning is necessary for dramatically different backends (e.g., BSV or SV)?

• how much intelligence to build in for optimization (e.g., automatically measuring of, and learning from, LLVM backend behavior) and parallelization and pipelining?

• what other kinds of assurance artifacts matter to customers (most folks do not understand proof, but do understand testing—how do we accommodate?)