Introduction to High-Level Synthesis of Cryptographic Algorithms
Cryptographic Standard Contests

- IX.1997: AES
- X.2000: NESSIE
- XI.2000: CRYPTREC
- XII.2002: eSTREAM
- XI.2004: SHA-3
- IV.2008: CAESAR
- X.2007: 15 block ciphers → 1 winner
- XII.2002: 34 stream ciphers → 4 HW winners + 4 SW winners
- XI.2004: 51 hash functions → 1 winner
- X.2007: 57 authenticated ciphers → multiple winners
- X.2012: TBD
Why a Contest for a Cryptographic Standard?

• Avoid back-door theories
• Speed-up the acceptance of the standard
• Stimulate non-classified research on methods of designing a specific cryptographic transformation
• Focus the effort of a relatively small cryptographic community
Evaluation Criteria in Cryptographic Contests

- Security
- Software Efficiency
  - μProcessors
  - μControllers
- Hardware Efficiency
  - FPGAs
  - ASICs
- Flexibility
- Simplicity
- Licensing
AES Contest 1997-2000
Final Round

Speed in FPGAs

Votes at the AES 3 conference

Hardware results matter!
NIST SHA-3 Contest - Timeline

51 candidates

Round 1: 14 candidates
  Oct. 2008
  July 2009

Round 2: 5 candidates
  Dec. 2010

Round 3: 1 candidate
Throughput vs. Area Normalized to Results for SHA-256 and Averaged over 11 FPGA Families – 256-bit variants

Early Leader

Overall Normalized Throughput

Overall Normalized Area
SHA-3 finalists in high-performance FPGA families

![Diagram showing normalized throughput/area for different FPGA families: Stratix IV, Stratix III, Virtex 6, Virtex 5, BLAKE, Groestl, Skein, JH, Keccak.](image)
GMU/ETH Zurich ASIC

- standard-cell CMOS 65nm UMC ASIC process
- 256-bit variants of algorithms
Correlation Between ASIC Results and FPGA Results

ASIC

Stratix III FPGA

Normalized Throughput

Normalized Area
Correlation Between ASIC Results and FPGA Results

ASIC

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Normalized TP/Area Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keccak</td>
<td>2.00</td>
</tr>
<tr>
<td>SHA2</td>
<td>1.00</td>
</tr>
<tr>
<td>JH</td>
<td>0.78</td>
</tr>
<tr>
<td>BLAKE</td>
<td>0.61</td>
</tr>
<tr>
<td>Groestl</td>
<td>0.41</td>
</tr>
<tr>
<td>Skein</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Stratix III FPGA

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Normalized TP/Area Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keccak</td>
<td>2.03</td>
</tr>
<tr>
<td>SHA2</td>
<td>1.00</td>
</tr>
<tr>
<td>JH</td>
<td>0.92</td>
</tr>
<tr>
<td>Groestl</td>
<td>0.51</td>
</tr>
<tr>
<td>BLAKE</td>
<td>0.39</td>
</tr>
<tr>
<td>Skein</td>
<td>0.38</td>
</tr>
</tbody>
</table>
Hardware Benchmarking in Cryptographic Contests

- **Focus on ranking**, rather than absolute values
- Only relatively **large differences** (>20-30%) matter
- Winner in use for the next 20-30 years, implemented using **technologies not in existence today**
- Very **wide range of possible applications**, and as a result performance and cost targets
- **Large number** of candidates
- **Limited time** for evaluation
- Results are **final**
<table>
<thead>
<tr>
<th></th>
<th>Initial number of candidates</th>
<th>Implemented in hardware</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>15</td>
<td>5</td>
<td>33.3%</td>
</tr>
<tr>
<td>eSTREAM</td>
<td>34</td>
<td>8</td>
<td>23.5%</td>
</tr>
<tr>
<td>SHA-3</td>
<td>51</td>
<td>14</td>
<td>27.5%</td>
</tr>
</tbody>
</table>
CAESAR Competition

Goal: Portfolio of new-generation authenticated ciphers

First-round submissions: March 15, 2014

Announcement of final portfolio: 2018

Organizer: An informal committee of leading cryptographic experts

Number of candidate families:

Round 1: 57  
Round 2: 29  
Round 3: 15
Authenticated Ciphers

Bob

$K_{AB}$

$\text{Authenticated Cipher}$

IV  AD  Message

IV  AD  Ciphertext  Tag

Alice

$K_{AB}$

$\text{Authenticated Cipher}$

IV  AD  Ciphertext  Tag

IV  AD  Ciphertext  Tag

valid

$K_{AB}$ - Secret key of Alice and Bob

IV – Initialization Vector, AD – Associated Data
Potential Solution: High-Level Synthesis (HLS)

- High Level Language (preferably C or C++)
- High-Level Synthesis
- Hardware Description Language (VHDL or Verilog)
Case for High-Level Synthesis & Crypto Contests

- Each submission includes reference implementation in C
- Development time potentially decreased several times
- All candidates can be implemented by the same group, and even the same designer
- Results from High-Level Synthesis could have a large impact in early stages of the competitions and help narrow down the search
- RTL code and results from previous contests form excellent benchmarks for High-Level Synthesis tools, which can generate fast progress targeting cryptographic applications
Our Hypotheses

- **Ranking** of candidates in cryptographic contests in terms of their performance in modern FPGAs will remain **the same** independently whether the HDL implementations are **developed manually** or **generated automatically** using High-Level Synthesis tools.

- The **development time** will be **reduced** by a factor of **3 to 10**.

- This hypothesis **should apply to** at least:
  - AES Contest, SHA-3 Contest, CAESAR Contest
  - possibly Post-quantum Cryptography?
In-Use Tools supporting C, C++, Extended C

Commercial:

- **Vivado HLS**: Xilinx
- **CHC**: Altium; **CoDeveloper**: Impulse Accelerated; **Synthesizer**: FORTE; **eXCite**: Y Explorations; **ROCCC**: Jacquard Comp.
- **Catapult-C**: Calypto Design Systems; **CtoS**: Cadence; **DK Design Suite**: Mentor Graphics; **Synphony C**: Synopsys

Academic:

- **Bambu**: Politecnico di Milano, Italy
- **DWARV**: Delft University of Technology, The Netherlands
- **GAUT**: Universite de Bretagne-Sud, France
- **LegUp**: University of Toronto, Canada
HLS Tool Considered So Far: Vivado HLS

- **Integrated** into the primary Xilinx toolset, Vivado, and released in 2012
- **Free** (or almost free) licenses for academic institutions
- Good **documentation and user support**
- The largest number of **performance optimizations**
- On average the **highest clock frequency** of the generated code
Sources of Productivity Gains

- Higher-level of abstraction
- Focus on datapath rather than control logic
- Debugging in software (C/C++)
  - Faster run time
  - No timing waveforms
GMU Case Studies

• 5 Final SHA_3 Candidates

• 15 Round 3 CAESAR Candidates + AES-GCM
  (all Round 3 families except Keyak and AEZ)
SHA 3: Manual RTL vs. HLS-based Results
Altera Stratix IV
SHA 3: Manual RTL vs. HLS-based Results

Xilinx Virtex 6

[Graphs showing comparison between RTL and HLS in terms of throughput and area for various algorithms.]
CAESAR: RTL vs. HLS Throughput/Area

Consistently better than AES-GCM

Suboptimal HLS

Suboptimal RTL
CAESAR: RTL vs. HLS Ratios for Throughput/Area
Virtex 6

> 1.30
Suboptimal HLS

[0.90, 1.30]
RTL and HLS acceptable

(0.70, 0.90]
RTL may be improved

< 0.70
Suboptimal RTL

Suboptimal HLS

RTL may be improved

[0.90, 1.30]
RTL and HLS acceptable

(0.70, 0.90]
RTL may be improved

< 0.70
Suboptimal RTL
Conclusions for HLS Benchmarking

Accuracy:
• Good (but not perfect) correlation between algorithm rankings using RTL and HLS approaches

Efficiency:
• Changes in the reference code (code refactoring) needed to infer the desired architecture, and thus #cycles_per_block
• 3-10 improvement in the development time
• Designer can focus on functionality: control logic inferred
• Much easier verification: C/C++ testbenches
• A single designer can produce implementations of multiple (and even all) candidates

Bottom Line:
• Manual (RTL) design approach still predominant
• HLS design approach at the experimental stage – more research needed
Not Tried Yet: Academic HLS Tools

- **Bambu:** Politecnico di Milano, Italy
- **DWARV:** Delft University of Technology, The Netherlands
- **GAUT:** Universite de Bretagne-Sud, France
- **LegUp:** University of Toronto, Canada
Overall Conclusions

– Contests for cryptographic standards are important
  • Stimulate progress in design and analysis of cryptographic algorithms
  • Determine future of cryptography for the next decades
  • Promote cryptology: Are easy to understand by general audience
  • Provide immediate recognition and visibility worldwide

– Computer Scientists, Digital System Designers, System Developers can play an important role in these contests
  • Co-designers of new cryptographic algorithms
  • Evaluators
  • Tool developers
  • Early adopters of new standards

– Get involved! It is fun!