Evaluation Of Compact FPGA Implementations For All SHA-3 Finalists

Bernhard Jungk

University of Applied Sciences Wiesbaden, Germany
Easycore GmbH, Erlangen, Germany

3rd SHA-3 Conference
1 Overview

2 The Algorithms

3 Results
Design Goals

- Comparison of all SHA-3 finalists for FPGAs (Xilinx)
  - Overall goal: Low area usage, yet high throughput-area ratio
  - No usage of BlockRAM, DSP units, ...
  - Inclusion of the padding function
  - Identical hardware interface
Design Goals

- Comparison of all SHA-3 finalists for FPGAs (Xilinx)
- Overall goal: Low area usage, yet high throughput-area ratio
  - No usage of BlockRAM, DSP units, ...
  - Inclusion of the padding function
  - Identical hardware interface
Design Goals

- Comparison of all SHA-3 finalists for FPGAs (Xilinx)
- Overall goal: Low area usage, yet high throughput-area ratio
- No usage of BlockRAM, DSP units, ...
- Inclusion of the padding function
- Identical hardware interface
Design Goals

- Comparison of all SHA-3 finalists for FPGAs (Xilinx)
- Overall goal: Low area usage, yet high throughput-area ratio
- No usage of BlockRAM, DSP units, ...
- Inclusion of the padding function
- Identical hardware interface
Design Goals

- Comparison of all SHA-3 finalists for FPGAs (Xilinx)
- Overall goal: Low area usage, yet high throughput-area ratio
- No usage of BlockRAM, DSP units, ...
- Inclusion of the padding function
- Identical hardware interface
Main idea #1: Folding of algorithm

- Benefit #1: Usage of RAM instead of registers
- Benefit #2: Reusing redundant parts of the compression functions
- Drawbacks: Reduced throughput, additional multiplexers, larger control logic
## Design Strategy

- **Main idea #1:** Folding of algorithm
- **Benefit #1:** Usage of RAM instead of registers
- **Benefit #2:** Reusing redundant parts of the compression functions
- **Drawbacks:** Reduced throughput, additional multiplexers, larger control logic
Main idea #1: Folding of algorithm
- Benefit #1: Usage of RAM instead of registers
- Benefit #2: Reusing redundant parts of the compression functions
- Drawbacks: Reduced throughput, additional multiplexers, larger control logic
Design Strategy

- Main idea #1: Folding of algorithm

- Benefit #1: Usage of RAM instead of registers

- Benefit #2: Reusing redundant parts of the compression functions

- Drawbacks: Reduced throughput, additional multiplexers, larger control logic
Main idea #2: Pipelining of folded architectures

- Benefit #1: Improved clock frequency
- Benefit #2: No (or few) additional clock cycles, if the pipeline is designed carefully.
- Drawback: Often more area is required
- But: For FPGAs a register can often be placed in the same slice as a LUT
- Control logic sometimes more difficult, sometimes easier to implement
Design Strategy

- Main idea #2: Pipelining of folded architectures
- Benefit #1: Improved clock frequency
  - Benefit #2: No (or few) additional clock cycles, if the pipeline is designed carefully.
- Drawback: Often more area is required
- But: For FPGAs a register can often be placed in the same slice as a LUT
- Control logic sometimes more difficult, sometimes easier to implement
Design Strategy

- Main idea #2: Pipelining of folded architectures
- Benefit #1: Improved clock frequency
- Benefit #2: No (or few) additional clock cycles, if the pipeline is designed carefully.
- Drawback: Often more area is required
- But: For FPGAs a register can often be placed in the same slice as a LUT
- Control logic sometimes more difficult, sometimes easier to implement
Design Strategy

- Main idea #2: Pipelining of folded architectures
- Benefit #1: Improved clock frequency
- Benefit #2: No (or few) additional clock cycles, if the pipeline is designed carefully.
- Drawback: Often more area is required
- But: For FPGAs a register can often be placed in the same slice as a LUT
- Control logic sometimes more difficult, sometimes easier to implement

Bernhard Jungk  
University of Applied Sciences Wiesbaden, Germany  
Easycore GmbH, Erlangen, Germany  
Evaluation Of Compact FPGA Implementations For All SHA-3 Finalists
Design Strategy

- Main idea #2: Pipelining of folded architectures
- Benefit #1: Improved clock frequency
- Benefit #2: No (or few) additional clock cycles, if the pipeline is designed carefully.
- Drawback: Often more area is required
- But: For FPGAs a register can often be placed in the same slice as a LUT
- Control logic sometimes more difficult, sometimes easier to implement
Two implementations, 1 or 2 halves of a $G_i$ function

- Quasi-Pipeline:
  - Depth 4 for 1 $G_i$ half
  - Depth 2 for 2 $G_i$ halves
Overview

The Algorithms

Results

Questions?

Compact Designs

BLAKE Compact Design Overview

- Two implementations, 1 or 2 halves of a $G_i$ function
- Quasi-Pipeline:
  - Depth 4 for 1 $G_i$ half
  - Depth 2 for 2 $G_i$ halves
Grøstl  Compact Design Overview

- **Hardware for P and Q shared**
- **Composite field S-box implementation**
Compact Design Overview

- Hardware for P and Q shared
- Composite field S-box implementation
JH 1st Compact Design Overview

- Very compact implementation (8 bit datapath)
- Shared core for constants computation and round function
JH 1st Compact Design Overview

- Very compact implementation (8 bit datapath)
- Shared core for constants computation and round function
JH 2nd Compact Design Overview

- Very broad datapath (320 bit), but still compact
- Makes output RAM unnecessary for degrouping
- Explicit LUT6_2 instances for S-boxes, linear transformation
Overview

compact designs

JH 2nd Compact Design Overview

- Very broad datapath (320 bit), but still compact
- Makes output RAM unnecessary for degrouping
- Explicit LUT6_2 instances for S-boxes, linear transformation

Bernhard Jungk
University of Applied Sciences Wiesbaden, Germany
Easycore GmbH, Erlangen, Germany

Evaluation Of Compact FPGA Implementations For All SHA-3 Finalists
Very broad datapath (320 bit), but still compact
- Makes output RAM unnecessary for degrouping
- Explicit LUT6×2 instances for S-boxes, linear transformation
Keccak Compact Design Overview

- $\frac{1}{8}$ round function
- Folding with Keccak-’slices’ instead of -’lanes’
- Rescheduled round function, permutation at the end
Keccak Compact Design Overview

- $\frac{1}{8}$ round function
- Folding with Keccak-'slices’ instead of -’lanes’
- Rescheduled round function, permutation at the end
Keccak Compact Design Overview

- $\frac{1}{8}$ round function
- Folding with Keccak-’slices’ instead of -’lanes’
- Rescheduled round function, permutation at the end
One pipelined MIX function, includes the key injection
## Analysis

### Clock Cycles

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Per round</th>
<th>Overhead</th>
<th>Complete compression function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-1</td>
<td>16</td>
<td>4</td>
<td>228</td>
</tr>
<tr>
<td>BLAKE-2</td>
<td>8</td>
<td>5</td>
<td>115</td>
</tr>
<tr>
<td>Grøstl</td>
<td>16</td>
<td>0</td>
<td>160</td>
</tr>
<tr>
<td>JH-1</td>
<td>160</td>
<td>0</td>
<td>6720</td>
</tr>
<tr>
<td>JH-2</td>
<td>4</td>
<td>0</td>
<td>168</td>
</tr>
<tr>
<td>Keccak</td>
<td>8</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>Skein</td>
<td>8</td>
<td>8</td>
<td>584</td>
</tr>
</tbody>
</table>

**Table:** Clock Cycles for compression function

---

Bernhard Jungk

University of Applied Sciences Wiesbaden, Germany

Easycore GmbH, Erlangen, Germany

Evaluation Of Compact FPGA Implementations For All SHA-3 Finalists
## Results

### Virtex-5

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Slices</th>
<th>MHz</th>
<th>MBit/s</th>
<th>MBit/s/Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-1</td>
<td>251</td>
<td>211</td>
<td>477</td>
<td>1.90 (5)</td>
</tr>
<tr>
<td>BLAKE-2</td>
<td>374</td>
<td>163</td>
<td>725</td>
<td>1.94 (4)</td>
</tr>
<tr>
<td>Grøstl</td>
<td>368</td>
<td>305</td>
<td>975</td>
<td>2.64 (1)</td>
</tr>
<tr>
<td>JH-1</td>
<td>193</td>
<td>283</td>
<td>22</td>
<td>0.11 (7)</td>
</tr>
<tr>
<td>JH-2</td>
<td>377</td>
<td>278</td>
<td>847</td>
<td>2.24 (2)</td>
</tr>
<tr>
<td>Keccak</td>
<td>393</td>
<td>159</td>
<td>864</td>
<td>2.19 (3)</td>
</tr>
<tr>
<td>Skein</td>
<td>519</td>
<td>299</td>
<td>262</td>
<td>0.50 (6)</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Slices</th>
<th>MHz</th>
<th>MBit/s</th>
<th>MBit/sSlice</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-1</td>
<td>260</td>
<td>263</td>
<td>590</td>
<td>2.26 (4)</td>
</tr>
<tr>
<td>BLAKE-2</td>
<td>419</td>
<td>204</td>
<td>908</td>
<td>2.18 (5)</td>
</tr>
<tr>
<td>Grøstl</td>
<td>328</td>
<td>365</td>
<td>1168</td>
<td>3.56 (1)</td>
</tr>
<tr>
<td>JH-1</td>
<td>221</td>
<td>442</td>
<td>33</td>
<td>0.14 (7)</td>
</tr>
<tr>
<td>JH-2</td>
<td>352</td>
<td>344</td>
<td>1048</td>
<td>2.97 (2)</td>
</tr>
<tr>
<td>Keccak</td>
<td>397</td>
<td>197</td>
<td>1071</td>
<td>2.69 (3)</td>
</tr>
<tr>
<td>Skein</td>
<td>406</td>
<td>316</td>
<td>277</td>
<td>0.68 (6)</td>
</tr>
</tbody>
</table>

Bernhard Jungk  
University of Applied Sciences Wiesbaden, Germany  
Easycore GmbH, Erlangen, Germany  
Evaluation Of Compact FPGA Implementations For All SHA-3 Finalists
## Results

### Spartan-3

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Slices</th>
<th>MHz</th>
<th>MBit/s</th>
<th>MBit/s Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-1</td>
<td>948</td>
<td>88.6</td>
<td>198</td>
<td>0.20 (3)</td>
</tr>
<tr>
<td>BLAKE-2</td>
<td>1716</td>
<td>71.6</td>
<td>318</td>
<td>0.18 (4)</td>
</tr>
<tr>
<td>Grøstl</td>
<td>1220</td>
<td>148</td>
<td>473</td>
<td>0.38 (1)</td>
</tr>
<tr>
<td>JH-1</td>
<td>807</td>
<td>124</td>
<td>9.4</td>
<td>0.011 (7)</td>
</tr>
<tr>
<td>JH-2</td>
<td>2060</td>
<td>113</td>
<td>344</td>
<td>0.16 (5)</td>
</tr>
<tr>
<td>Keccak</td>
<td>1665</td>
<td>71.2</td>
<td>387</td>
<td>0.23 (2)</td>
</tr>
<tr>
<td>Skein</td>
<td>1347</td>
<td>128</td>
<td>112</td>
<td>0.083 (6)</td>
</tr>
</tbody>
</table>
# Results

**Spartan-6**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Slices</th>
<th>MHz</th>
<th>MBit/s</th>
<th>(\text{MBit/s} / \text{Slice})</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-1</td>
<td>257</td>
<td>155</td>
<td>477</td>
<td>1.85 (4)</td>
</tr>
<tr>
<td>BLAKE-2</td>
<td>413</td>
<td>113</td>
<td>725</td>
<td>1.75 (5)</td>
</tr>
<tr>
<td>Grøstl</td>
<td>344</td>
<td>236</td>
<td>975</td>
<td>2.83 (1)</td>
</tr>
<tr>
<td>JH-1</td>
<td>171</td>
<td>241</td>
<td>22</td>
<td>0.12 (7)</td>
</tr>
<tr>
<td>JH-2</td>
<td>372</td>
<td>185</td>
<td>847</td>
<td>2.27 (2)</td>
</tr>
<tr>
<td>Keccak</td>
<td>420</td>
<td>122</td>
<td>864</td>
<td>2.05 (3)</td>
</tr>
<tr>
<td>Skein</td>
<td>418</td>
<td>210</td>
<td>262</td>
<td>0.62 (6)</td>
</tr>
</tbody>
</table>
Thank you for your attention.