ATHENa 2.0

&

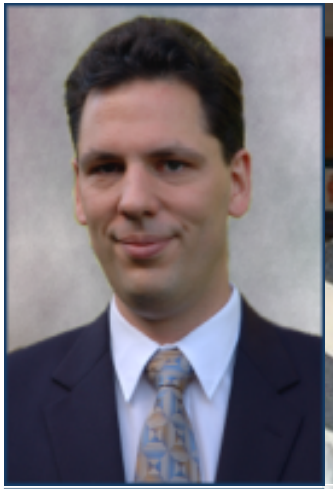
ATHENa Database of Results

Kris Gaj, Jens-Peter Kaps,
Benjamin Y. Brewster, John Pham,
Ekawat Homsirikamol,
and Rajesh Velegalati

Supported in part by the National Institute of Standards & Technology (NIST)

ATHENa Team - 2012

Jens-Peter Kaps
Associate Professor



Benjamin Brewster
MS CpE student

John Pham
MS CpE student

Ekawat "Ice" Homsirikamol
PhD ECE student

Rajesh Velegalati
PhD ECE student

ATHENa – **A**utomated **T**ool for **H**ardware **E**valuatio**N**

http://cryptography.gmu.edu/athena

Open-source benchmarking environment, written in Perl, aimed at

AUTOMATED generation of
OPTIMIZED results for
MULTIPLE hardware platforms.

The most recent version
0.6.3 released in May 2012.

Basic Dataflow of ATHENa

User

FPGA Synthesis and Implementation

6

5

Database query

Ranking of designs

2

3

HDL + scripts + configuration files

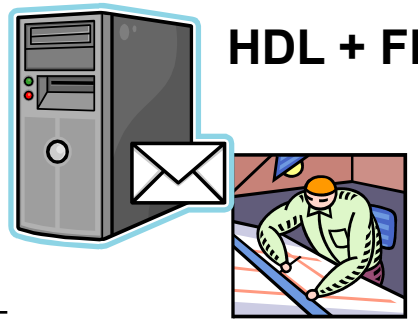
Result Summary + Database Entries

ATHENa Server

1

Download scripts and configuration files

HDL + FPGA Tools



4

Database Entries

Designer

0

Interfaces + Testbenches

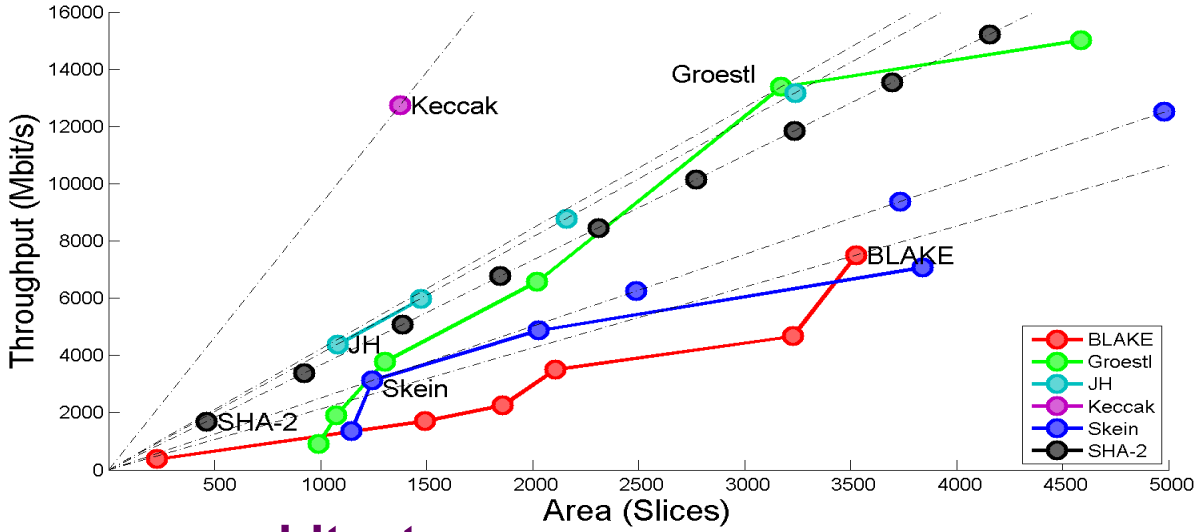
Benchmarking of the SHA-3 Finalists as a Test Case

- 6 algorithms (BLAKE, Groestl, JH, Keccak, Skein, SHA-2)
- 2 variants (with a 256-bit and a 512-bit output)
- 7 to 12 different architectures per algorithm
- 4 modern FPGA families (Virtex 5, Virtex 6, Stratix III, Stratix IV)

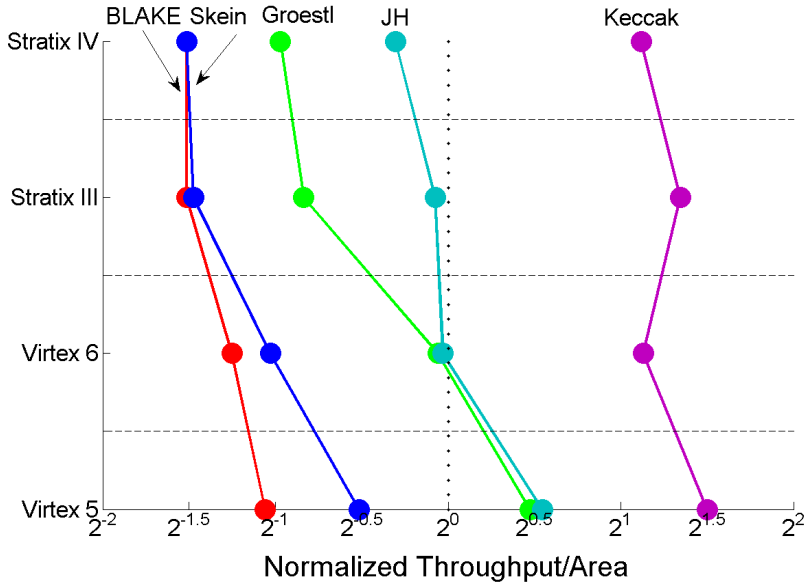
Total: ~ 120 designs
~ 600+ results

SHA-3 Performance Graphs (256-bit variants)

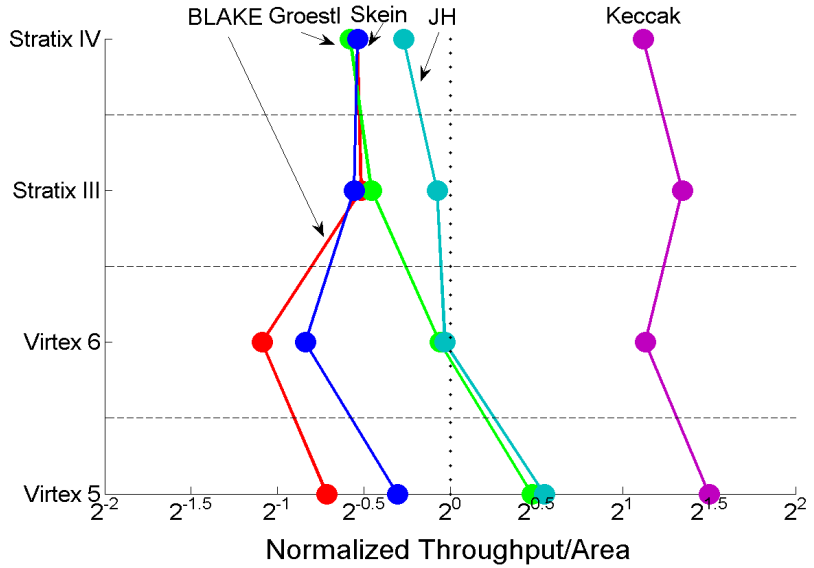
Virtex 5



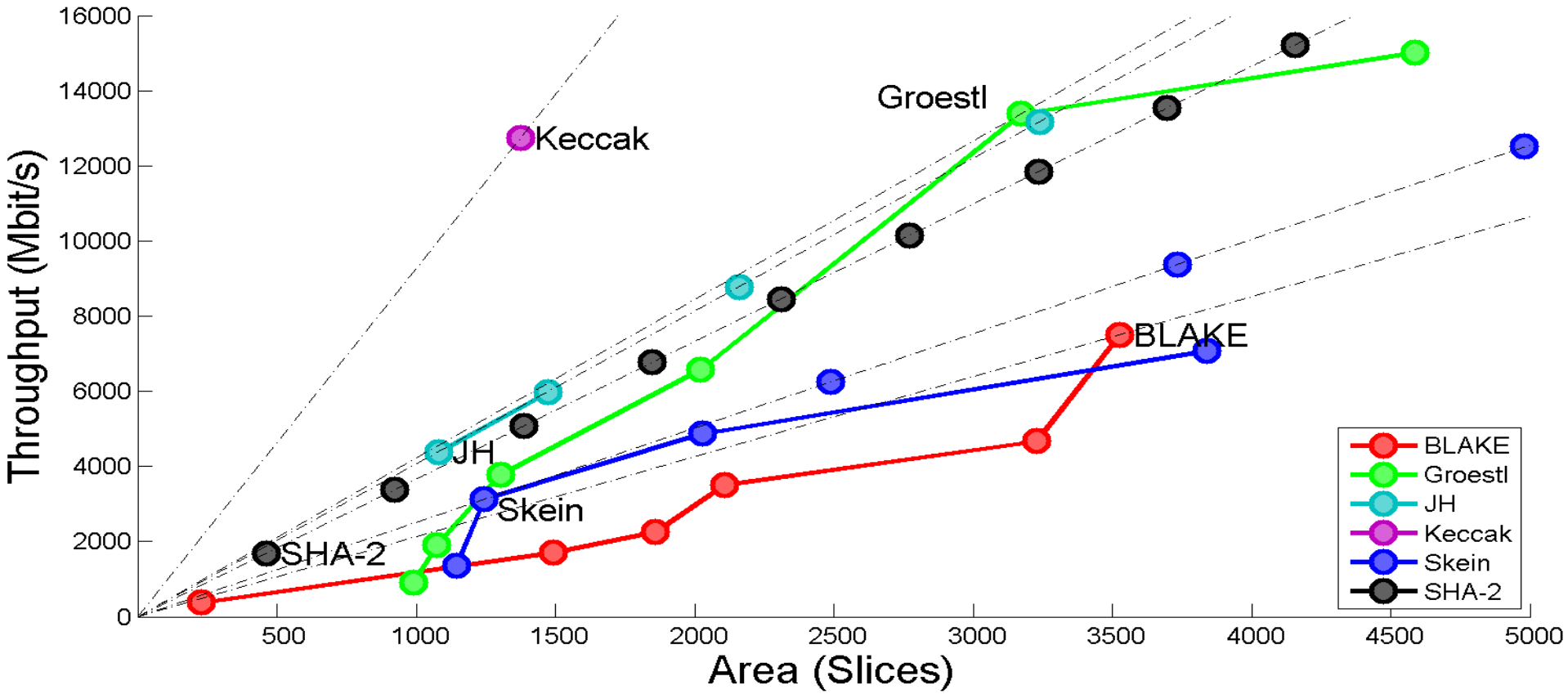
Best single-message architectures



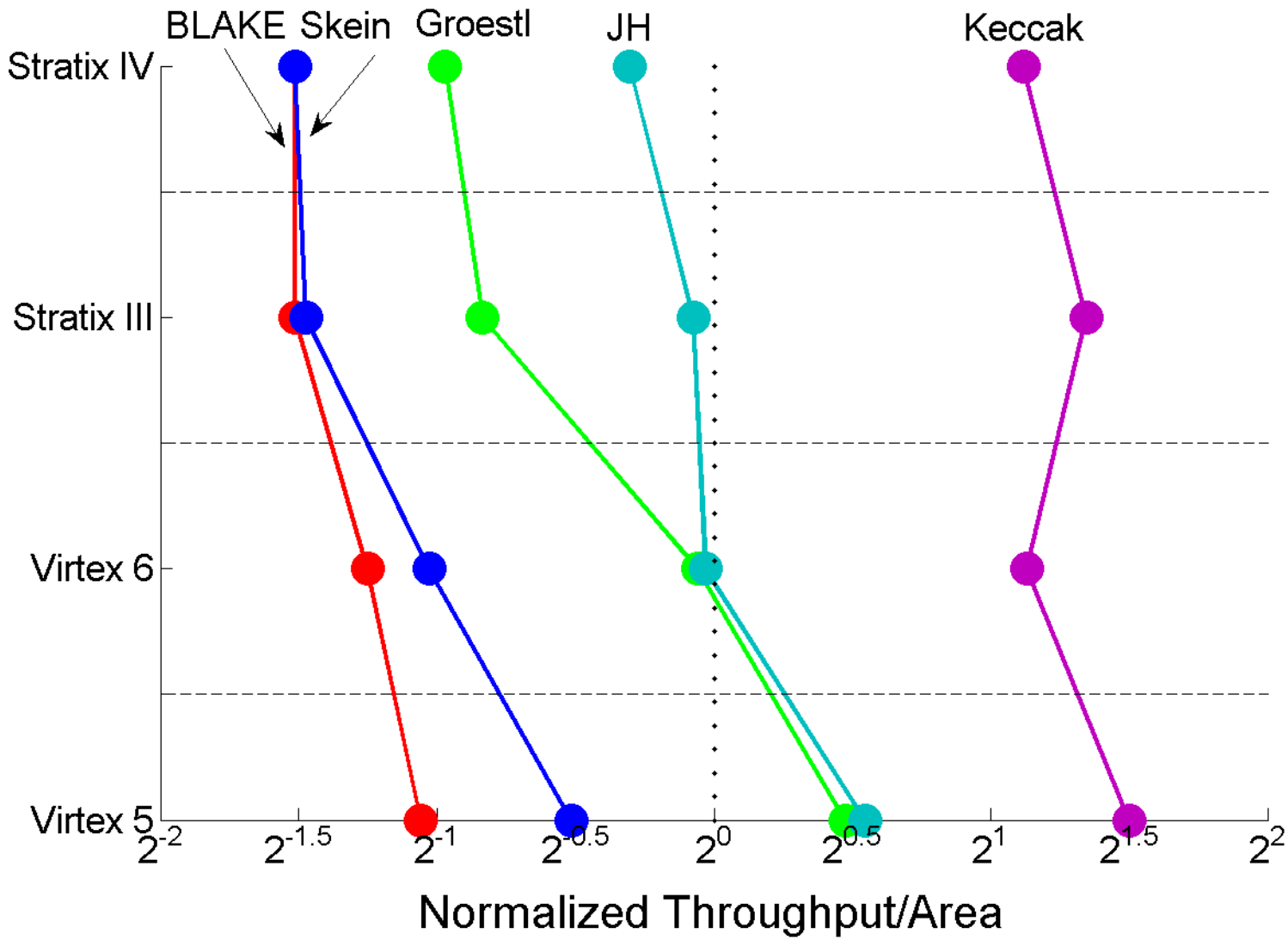
Best overall architectures



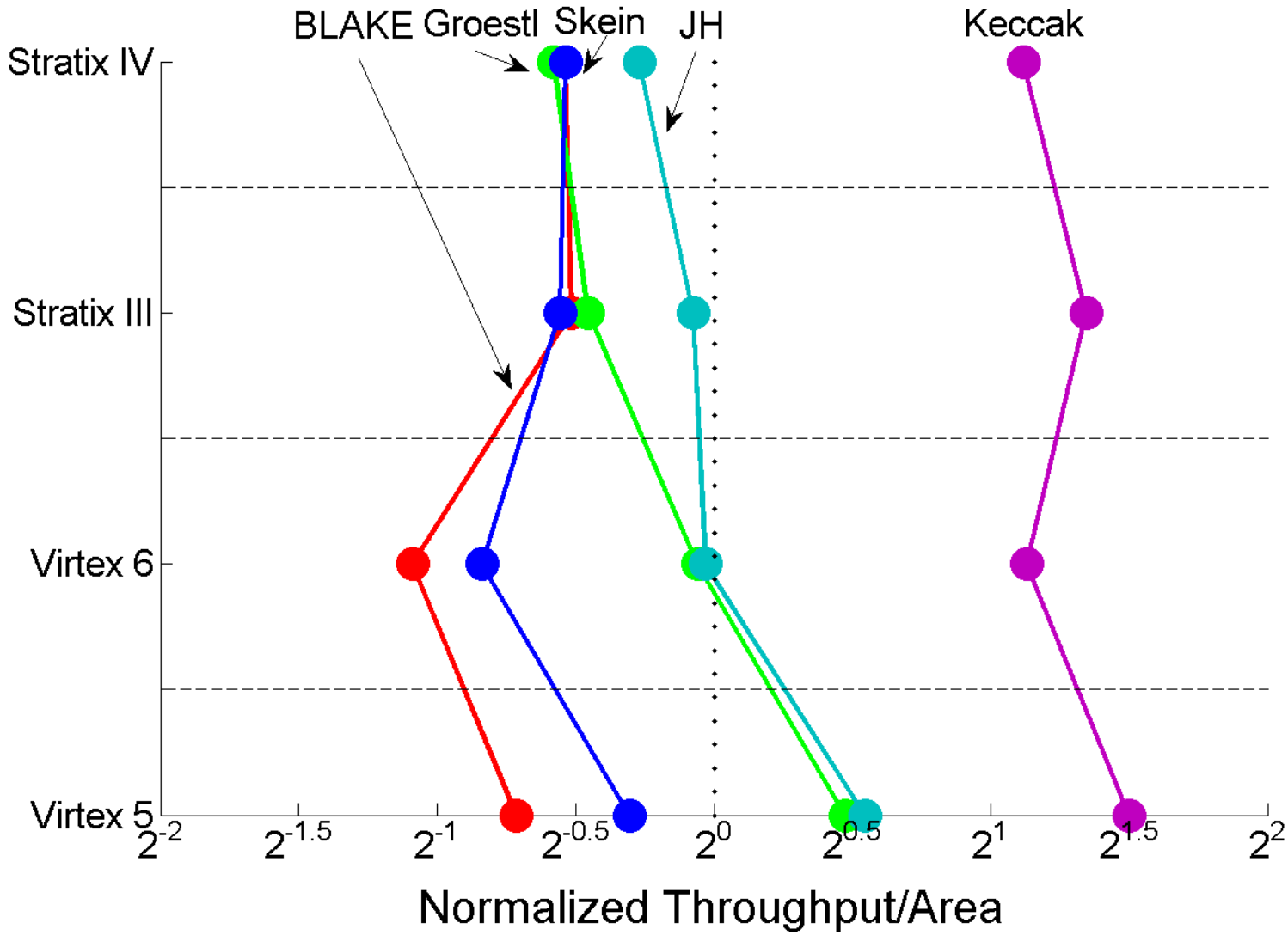
Throughput vs. Area Trade-offs in Virtex 5



Best Single-Message Architectures



Best Overall Architectures



Generation of Results Facilitated by ATHENa

- batch mode of FPGA tools
- automated choice of tool options
- ease of extraction and tabulation of results (Excel, CSV)
- close integration with the database of results

vs.

ATHENa Database of Results

http://cryptography.gmu.edu/athenadb

FPGA Hash Function Results Table

Show Help

Compare Selected | SHA-3 Round 3 | SHA-3 Round 3 & SHA-2 | SHA-3 Round 2 | SHA-3 Round 2 & SHA-2

Show 25 entries

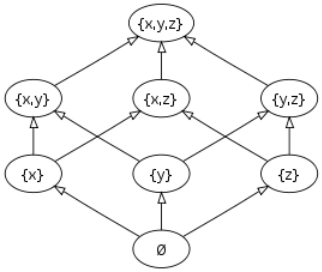
Copy | CSV | Excel

- About
- All ASIC Results
- All FPGA Results
- FPGA Rankings
- Login

	Algorithm		Design			
Result ID	Algorithm (Enable Unique)	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding
2469	JH	512	Throughput/Area	Pipelined x2-PPL2 (MEM)	2	Yes
2467	Skein	512	Throughput/Area	Pipelined x4-PPL5	5	Yes
2465	Skein	512	Throughput/Area	Pipelined x4-PPL5	5	Yes
2464	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
2462	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
2460	Groestl	512	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
2459	Groestl	512	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
2458	BLAKE	512	Throughput/Area	Pipelined x1-PPL2	2	No
2457	BLAKE	512	Throughput/Area	Pipelined x1-PPL2	2	No
2456	BLAKE	256	Throughput/Area	Pipelined x1-PPL2	2	No
2454	JH	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	Yes
2453	JH	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2452	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2451	BLAKE	512	Throughput/Area	Pipelined /2(v)-PPL4	4	No
2450	BLAKE	512	Throughput/Area	Pipelined /2(v)-PPL4	4	No
2449	Keccak	256	Throughput/Area	Pipelined x2-PPL2	2	No
2448	Keccak	512	Throughput/Area	Pipelined x2-PPL2	2	Yes

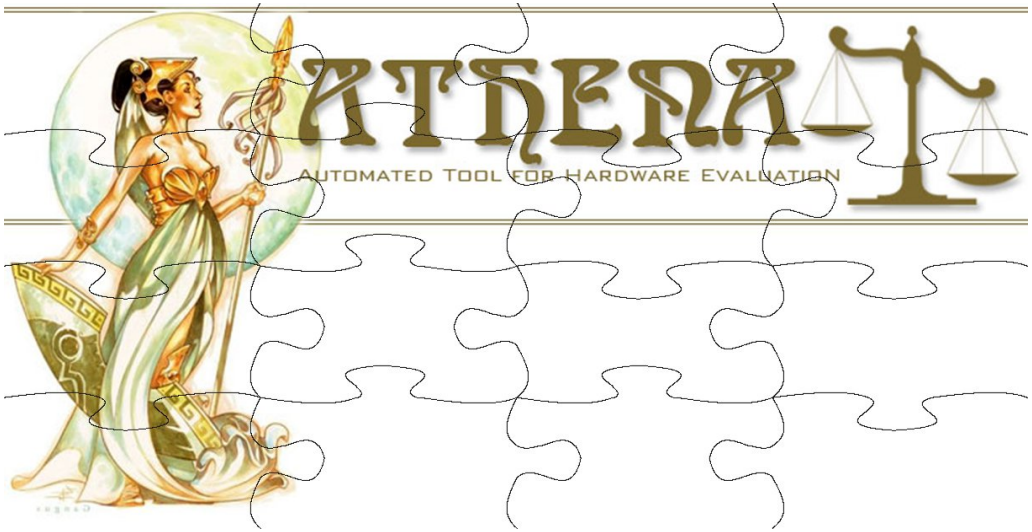
Major Improvements To Be Introduced in ATHENa 2.0

- **Parallel Execution on Multiple Computers**
  - Utilize idle resources
  - Increase throughput of benchmarking tasks
  - Decrease benchmarking time
- **Usability**
  - GUI
  - Monitoring and control
  - Benchmark configuration
- **Optimization Space Exploration**
  - Search more options
  - Decrease search time
  - Increase optimization end-performance



Other Improvements Introduced in ATHENa 2.0

- **Modular, Extensible, and Maintainable**
  - Python in place of Perl
  - Object-Oriented Design Principles
  - Client/Server Architecture
  - Ease of adding and removing functionality

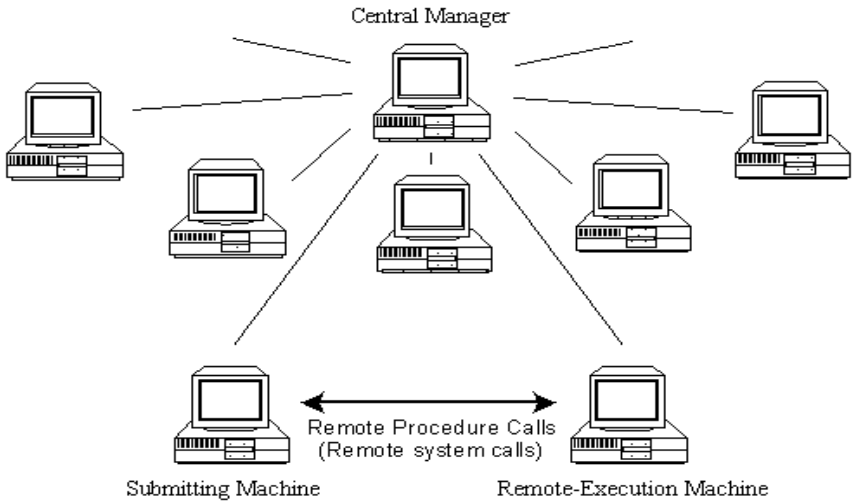


Parallel Execution on Multiple Computers: Choice of the Most Suitable Batch System

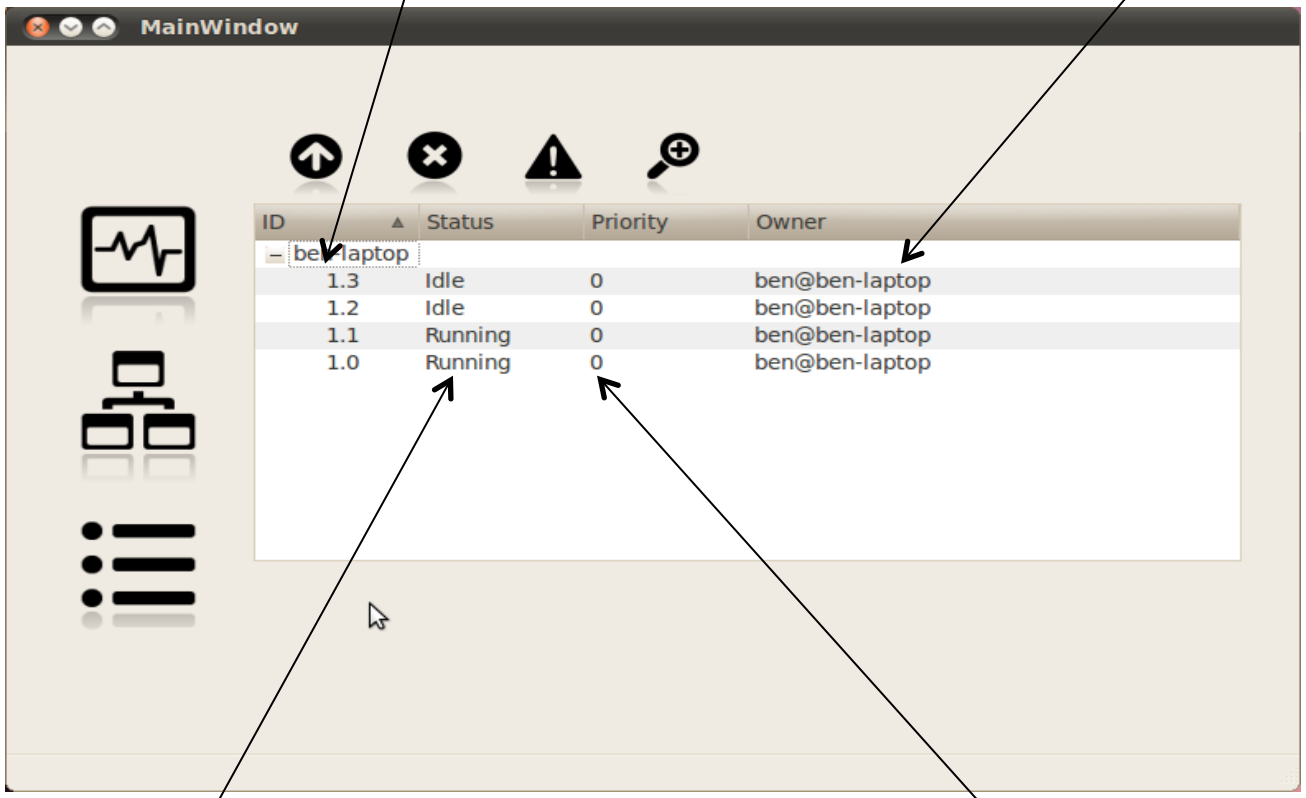
- **Candidates**
  - **Condor** – Workload Management System
  - **Globus** – Meta Scheduler for grid computing
  - **Torque** – Resource manager, based on PBS (Portable Batch System)
  - **MAUI** – Scheduler for use with Torque
  - **JPPF** – Java Parallel Processing Framework
  - **SLURM** – Simple Linux Utility for Resource Management
- **Evaluation Criteria**
  - **Flexibility**
  - **Cross platform**
  - **Easy to use and administer**
  - **Reliable**
  - **No language restrictions**
  - **Security**
  - **Fault Tolerance**

Selected Batch System

- **Condor**
  - Workload Management System
  - Developed at the University of Wisconsin-Madison
  - Provides a job queuing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management
  - Supports heterogeneous computing resources
  - Actively supported and widely used
  - More details at http://research.cs.wisc.edu/condor/



Usability:
Graphical User Interface



Optimization Space Exploration: Our Approach

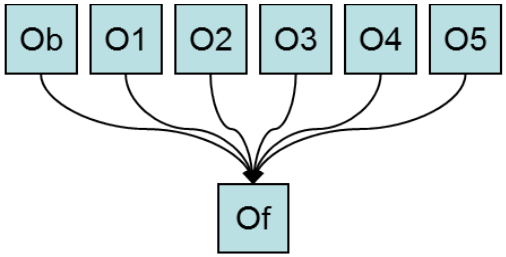
- Use algorithms inspired by previous research on the programming-language compilers
  - Least Effort – LE
  - Most Effort – ME
  - Batch Elimination – BE
  - Iterative Elimination – IE
  - Orthogonal Arrays – OA
- Optimize FPGA-specific algorithms introduced in current ATHENa
  - Frequency Search – FS
  - Placement Search – PS

Least Effort & Most Effort

- **Least Effort** – minimum execution time, worst results
  - Lazy or Naïve optimization
    - Set all options to the perceived high state
    - Only works well with binary options
    - Requires judgment on what is “high”
  - Used as a baseline
    - Minimum amount of work needed to optimize
    - Almost never optimal
- **Most Effort** – maximum execution time, best results
    - Also known as Exhaustive Search
    - Guarantee optimal result
    - Least time-efficient
    - Impractical for more than a handful of options
    - Number of runs needed: 2^n, where n is the number of options

Batch Elimination

Run	Option 1	Option 2	Option 3	Option 4	Option 5	Improvement Relative to Base Options
Ob	0	0	0	0	0	N/A
O1	1	0	0	0	0	+
O2	0	1	0	0	0	–
O3	0	0	1	0	0	–
O4	0	0	0	1	0	+
O5	0	0	0	0	1	+
Of	1	0	0	1	1	N/A



Ob – Base options (all options off)
Oi – Option i on, i=1..n
Of – Final options

Number of runs: n+2
Number of run levels: 2

Based on: Z. Pan and R. Eigenmann, "Fast and Effective Orchestration of Compiler Optimizations for Automatic Performance Tuning," Proc. International Symposium on Code Generation and Optimization, CGO 2006.

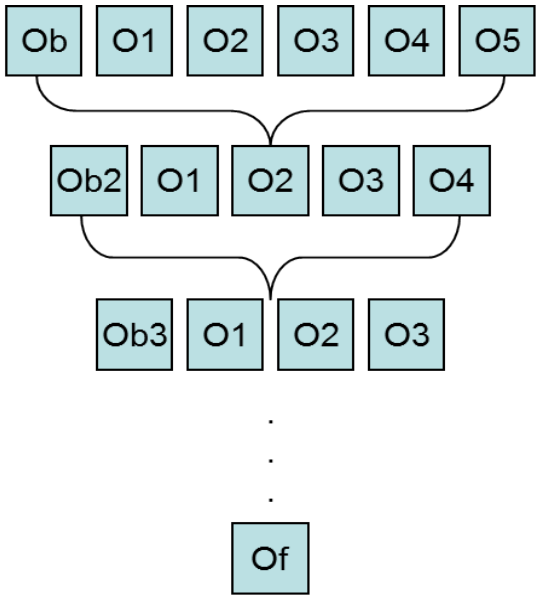
Potential disadvantage: The interaction between options is not tested

Iterative Elimination

Run	Option 1	Option 2	Option 3	Option 4	Option 5	Improvement Relative to Current Base
Ob	0	0	0	0	0	N/A
O1	1	0	0	0	0	+5
O2	0	1	0	0	0	–
O3	0	0	1	0	0	–
O4	0	0	0	1	0	+7
O5	0	0	0	0	1	+2
Ob2=O4	0	0	0	1	0	+7 from baseline 1
O1’	1	0	0	1	0	+3
O2’	0	1	0	1	0	–
O3’	0	0	1	1	0	–
O5’	0	0	0	1	1	+1
0b3=O1’	1	0	0	1	0	+3 from baseline 2
O2”	1	1	0	1	0	–
O3”	1	0	1	1	0	–
O5”	1	0	0	1	1	–
Of	1	0	0	1	0	+10 total

Iterative Elimination

Ob – Base options at the start (all options off)
Obj – Base options at level j
Oi – Option i on
Of – Final options



Maximum number of runs: (n+1)(n/2) (can be as few as n+1)
Maximum number of run levels: n (can be as few as 1)

Based on: Z. Pan and R. Eigenmann, "Fast and Effective Orchestration of Compiler Optimizations for Automatic Performance Tuning," Proc. International Symposium on Code Generation and Optimization, CGO 2006.

Orthogonal Arrays

- k x n matrix where the columns represent optimization options and the rows represent the settings used for each experiment
- The matrix is filled with 1's and 0's to represent whether or not a specified option is on or off
- Any two arbitrary columns contain the patterns: {00, 01, 10, 11} equally often

n Options

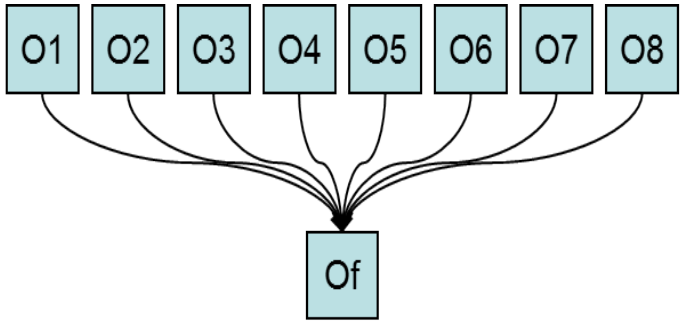
k Experiments

1	0	0	0	0
0	1	0	1	0
1	1	1	0	1
0	0	1	1	1
1	0	0	0	1
0	1	0	1	1
1	1	1	0	0
0	0	1	1	0

- The algorithm guarantees that half of the experiments will be conducted with an option Oi on and half with option Oi off
- For arbitrary two options Oi and Oj there are exactly k/4 experiments per each possible setting of these two options

Orthogonal Arrays

Run	Option 1	Option 2	Option 3	Option 4	Option 5
O1	1	0	0	0	0
O2	0	1	0	1	0
O3	1	1	1	0	1
O4	0	0	1	1	1
O5	1	0	0	0	1
O6	0	1	0	1	1
O7	1	1	1	0	0
O8	0	0	1	1	0
RIP(Oi)	+	+	-	-	+
Of	1	1	0	0	+



$$RIP(O_i) = \frac{\sum P(O_i = 1) - \sum P(O_i = 0)}{\sum P(O_i = 0)}$$

Number of runs: k+1

Number of run levels: 2

Based on: R.P.J. Pinkers, P.M.W Knijnenburg, M. Haneda, and H.A.G. Wijshoff, "Statistical Selection of Compiler Options," 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2004.

Experiments

- **Codes:** 4 SHA-3 candidate algorithms: BLAKE, JH, Keccak, and Skein
- **FPGA families:** Spartan 3 and Virtex 6
- **Version of tools:** Xilinx ISE v.13.1
- **Hosts:** Two eight core Linux workstations = total of 16 execute hosts
- **Optimization Target:** Throughput/Area Ratio

- **Experiment 1**
  - Determine ability of Batch Elimination, Iterative Elimination and Orthogonal Array to achieve optimal results
  - Limited search to **5 options**
  - Used Least Effort and Most Effort as basis for comparisons
- **Experiment 2**
  - Determine the ability of algorithms to fully optimize a design
  - Used expanded **9 option set** and **optimization algorithm chaining**
  - Used results generated with current ATHENa as basis for comparison

Results of Experiment 1

Spartan 3: Above Least Effort (%)

	BE	IE	OA
JH	5.3	16.0	15.5
BLAKE	7.9	33.0	-3.0
Skein	3.3	5.9	-1.9
Keccak	-1.3	10.8	8.5
Average %inc	3.8	16.4	4.7
Median %inc	4.3	13.4	3.2

Spartan 3: Below Most Effort (%)

	BE	IE	OA
JH	-9.8	-0.7	-1.1
BLAKE	-18.9	0	-27.1
Skein	-12.8	-10.6	-17.1
Keccak	-10.9	0	-2.1
Average %inc	-13.1	-2.8	**-11.9**
Median %inc	-11.8	-0.3	**-9.6**

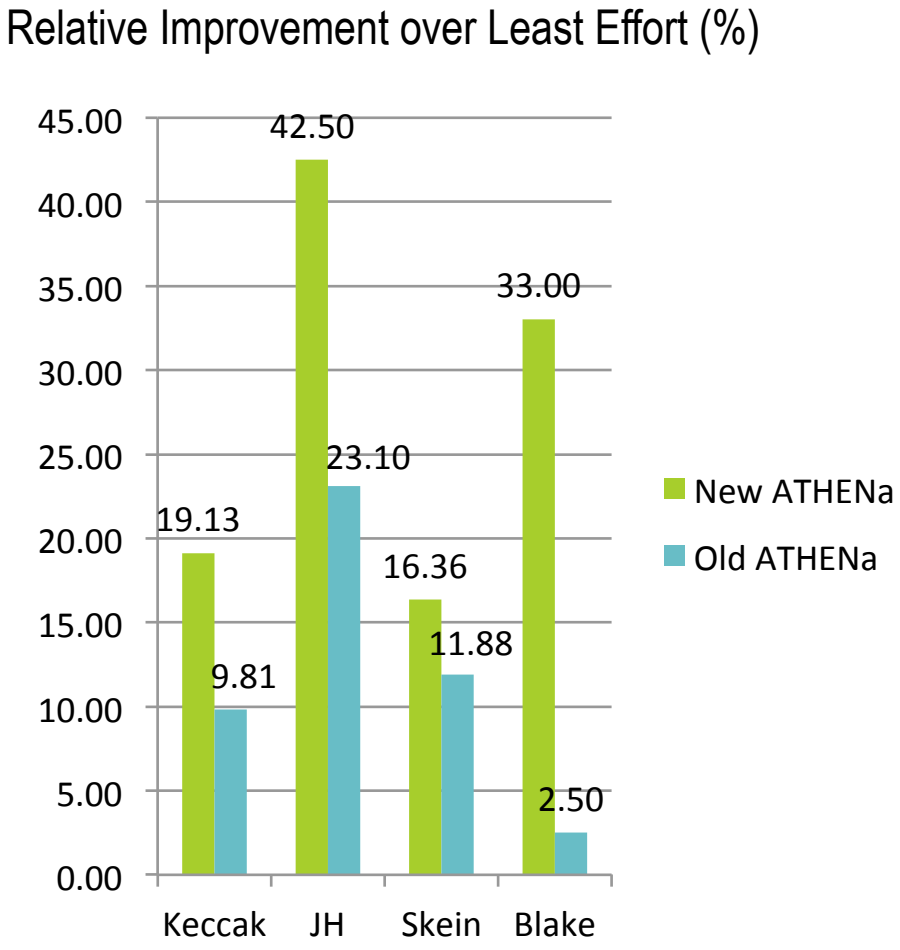
Virtex 6: Above Least Effort (%)

	BE	IE	OA
JH	8.6	13.5	13.5
BLAKE	26.4	36.4	26.5
Skein	-2.6	9.4	7.2
Keccak	-2.6	1.1	-3.7
Average %inc	7.5	15.1	10.9
Median %inc	3.0	11.4	10.3

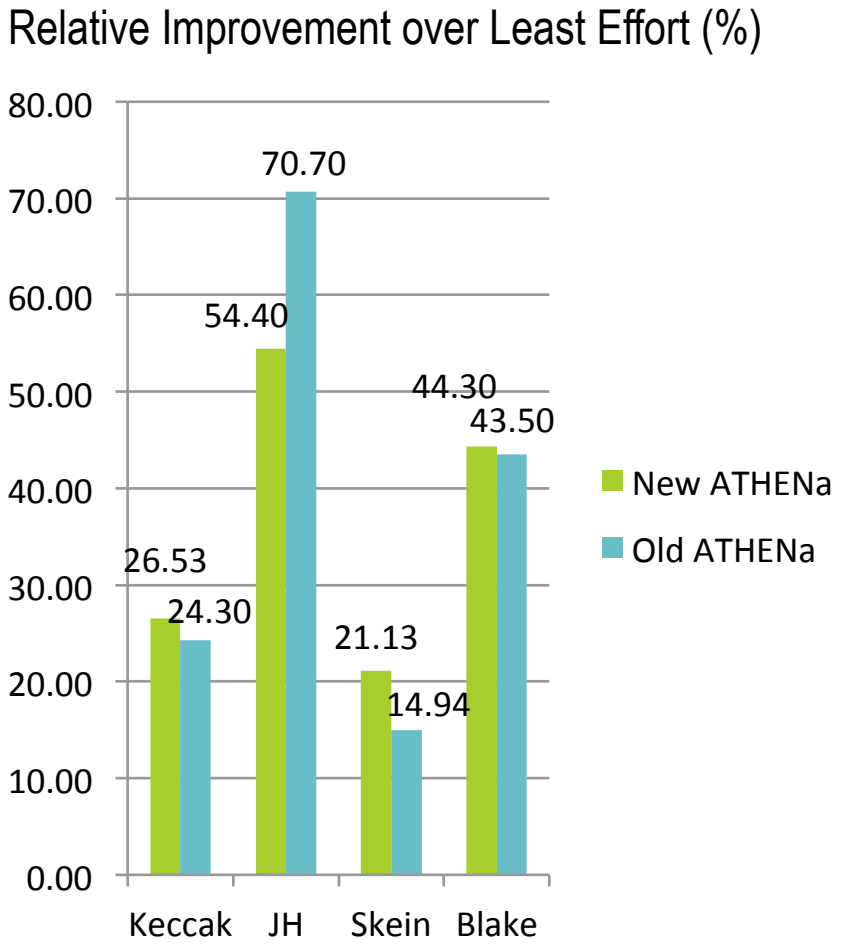
Virtex 6: Below Most Effort (%)

	BE	IE	OA
JH	-4.3	0	0
BLAKE	-7.3	0	-7.3
Skein	-11.0	0	-2.0
Keccak	-8.5	-5.1	-9.6
Average %inc	-7.8	-1.3	**-4.7**
Median %inc	-7.9	0	**-4.6**

Results of Experiment 2



Spartan 3



Virtex 6

Conclusions

- **Distributed architecture and parallelization increases throughput of benchmarking tasks**
  - Parallelization extended beyond core count of a single machine
  - Better more efficient use of resources
  - Greater flexibility
- **Optimization Space Exploration**
  - Increases number of options searched effectively
  - Iterative Elimination is a viable alternative to Most effort optimization with larger options sets
  - Optimization chaining yields results that outperform the current ATHENa in most cases

ATHENa Database of Results

http://cryptography.gmu.edu/athenadb

ATHENA

AUTOMATED TOOL FOR HARDWARE EVALUATION

FPGA Hash Function Results Table

Show Help

Compare Selected | SHA-3 Round 3 | SHA-3 Round 3 & SHA-2 | SHA-3 Round 2 | SHA-3 Round 2 & SHA-2

Show 25 entries

Copy | CSV | Excel

- About
- All ASIC Results
- All FPGA Results
- FPGA Rankings
- Login

	Algorithm		Design			
Result ID	Algorithm (Enable Unique)	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding
2469	JH	512	Throughput/Area	Pipelined x2-PPL2 (MEM)	2	Yes
2467	Skein	512	Throughput/Area	Pipelined x4-PPL5	5	Yes
2465	Skein	512	Throughput/Area	Pipelined x4-PPL5	5	Yes
2464	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
2462	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
2460	Groestl	512	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
2459	Groestl	512	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
2458	BLAKE	512	Throughput/Area	Pipelined x1-PPL2	2	No
2457	BLAKE	512	Throughput/Area	Pipelined x1-PPL2	2	No
2456	BLAKE	256	Throughput/Area	Pipelined x1-PPL2	2	No
2454	JH	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	Yes
2453	JH	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2452	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2451	BLAKE	512	Throughput/Area	Pipelined /2(v)-PPL4	4	No
2450	BLAKE	512	Throughput/Area	Pipelined /2(v)-PPL4	4	No
2449	Keccak	256	Throughput/Area	Pipelined x2-PPL2	2	No
2448	Keccak	512	Throughput/Area	Pipelined x2-PPL2	2	Yes

User Help

FPGA Hash Function Results Table

Hide Help

All results are categorized in groups, e.g., Algorithm, Design , Platform, etc. Clicking on the group name reveals additional columns for the respective group. Clicking on the group name again collapses the group.

Each column has a column title e.g. Result ID, Primary Opt. Target, TP [Mbits/s] (meaning Throughput) etc. Clicking on the column title sorts the column in ascending order, clicking on it again sorts it in descending order. In addition, the little arrow next to the title changes color to indicate the sort order and the column is shaded. A second order of sorting can be achieved by pressing the "Shift" key on the keyboard and then clicking on another column title.

- **About**
- **All ASIC Results**
- **All FPGA Results**
- **FPGA Rankings**
- **Login**

At the bottom of each column is a search box. Any entry in these boxes is only applied to the respective column. The search is not case sensitive, and is regex based. An OR operation can be accomplished by surrounding search terms w/ parenthesis and seperating them with a pipe, like so:

```
(term1|term2)
```

The number of results matching the current search criteria is shown at the end of the table, below the paging buttons.

Clicking on a Result ID opens a page that shows all details about that particular result.

To compare two results, click one result, then another to compare against. Then click the 'Compare Selected Results' action

In order to preserve the state of the table when examining a result, please right-click and select "open in new window" or "open in new tab". Alternatively, use middle click on a result id.

Numeric and date columns may be searched using by specifying a range in ISO interval notation, or a simple one sided bound using >. For example, to search for all results with values between 1000 and 2000, inclusive, use:

```
[1000,2000]
```

To search for values greater than 2000 exclusive, use:

```
> 2000
Date ranges may also be entered using a date picker, activated by clicking "Enter Date Range" below the appropriate field.
```

An Expandable Group of Fields

Search boxes

Show 25 entries

Copy CSV Excel

	Algorithm		Design			
Result ID	Algorithm Enable Unique	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding
2469	JH	512	Throughput/Area	Pipelined x2-PPL2 (MEM)	2	Yes
2467	Skein	512	Throughput/Area	Pipelined x4-PPL5	5	Yes
2465	Skein	512	Throughput/Area	Pipelined x4-PPL5	5	Yes
2464	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
2462	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
2460	Groestl	512	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
2459	Groestl	512	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
2458	BLAKE	512	Throughput/Area	Pipelined x1-PPL2	2	No
2457	BLAKE	512	Throughput/Area	Pipelined x1-PPL2	2	No
2456	BLAKE	256	Throughput/Area	Pipelined x1-PPL2	2	No
2454	JH	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	Yes
2453	JH	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2452	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2451	BLAKE	512	Throughput/Area	Pipelined /2(v)-PPL4	4	No
2450	BLAKE	512	Throughput/Area	Pipelined /2(v)-PPL4	4	No
2449	Keccak	256	Throughput/Area	Pipelined x2-PPL2	2	No
2448	Keccak	512	Throughput/Area	Pipelined x2-PPL2	2	Yes
2447	Skein	512	Throughput/Area	Pipelined x8-PPL10	10	No
2444	Skein	256	Throughput/Area	Pipelined x8-PPL10	10	No
2442	Skein	512	Throughput/Area	Unrolled x8	1	Yes
2441	Skein	512	Throughput/Area	Unrolled x8	1	Yes
2439	Skein	512	Throughput/Area	Unrolled x8	1	No
2437	Skein	512	Throughput/Area	Unrolled x8	1	No
2435	Skein	256	Throughput/Area	Unrolled x8	1	Yes
2432	Skein	256	Throughput/Area	Unrolled x8	1	Yes
Result ID	Algorithm	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding

First Previous 1 2 3 4 5 Next Last

Showing 1 to 25 of 1,584 entries

A Group of Fields after Expansion

Show 25 entries

Copy CSV Excel

	Algorithm				Design		
Result ID	Group	Algorithm	Hash Size [bits]	Msg Blk Size [bits]	Primary Opt Target	Arch Type	Max #Streams
2469	SHA-3 Round 3	JH	512	512	Throughput/Area	Pipelined x2-PPL2 (MEM)	2
2467	SHA-3 Round 3	Skein	512	512	Throughput/Area	Pipelined x4-PPL5	5
2465	SHA-3 Round 3	Skein	512	512	Throughput/Area	Pipelined x4-PPL5	5
2464	SHA-3 Round 3	Skein	256	512	Throughput/Area	Pipelined x4-PPL5	5
2462	SHA-3 Round 3	Skein	256	512	Throughput/Area	Pipelined x4-PPL5	5
2460	SHA-3 Round 3	Groestl	512	1,024	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2
2459	SHA-3 Round 3	Groestl	512	1,024	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2
2458	SHA-3 Round 3	BLAKE	512	1,024	Throughput/Area	Pipelined x1-PPL2	2
2457	SHA-3 Round 3	BLAKE	512	1,024	Throughput/Area	Pipelined x1-PPL2	2
2456	SHA-3 Round 3	BLAKE	256	512	Throughput/Area	Pipelined x1-PPL2	2
2454	SHA-3 Round 3	JH	512	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4
2453	SHA-3 Round 3	JH	512	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4
2452	SHA-3 Round 3	JH	256	512	Throughput/Area	Pipelined x2-PPL4 (MEM)	4
2451	SHA-3 Round 3	BLAKE	512	1,024	Throughput/Area	Pipelined /2(v)-PPL4	4
2450	SHA-3 Round 3	BLAKE	512	1,024	Throughput/Area	Pipelined /2(v)-PPL4	4
2449	SHA-3 Round 3	Keccak	256	1,088	Throughput/Area	Pipelined x2-PPL2	2
2448	SHA-3 Round 3	Keccak	512	576	Throughput/Area	Pipelined x2-PPL2	2
2447	SHA-3 Round 3	Skein	512	512	Throughput/Area	Pipelined x8-PPL10	10
2444	SHA-3 Round 3	Skein	256	512	Throughput/Area	Pipelined x8-PPL10	10
2442	SHA-3 Round 3	Skein	512	512	Throughput/Area	Unrolled x8	1
2441	SHA-3 Round 3	Skein	512	512	Throughput/Area	Unrolled x8	1
2439	SHA-3 Round 3	Skein	512	512	Throughput/Area	Unrolled x8	1
2437	SHA-3 Round 3	Skein	512	512	Throughput/Area	Unrolled x8	1
2435	SHA-3 Round 3	Skein	256	512	Throughput/Area	Unrolled x8	1
2432	SHA-3 Round 3	Skein	256	512	Throughput/Area	Unrolled x8	1

Result ID | Group | Algorithm | Hash Size [bits] | Msg Blk Size [bi | Primary Opt Target | Arch Type | Max #Streams

Algorithm		Design				Platform
Algorithm (Enable Unique)	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding	Family
Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes	Virtex 5
BLAKE	256	Throughput/Area	Pipelined x1-PPL2	2	No	Virtex 5
Keccak	256	Throughput/Area	Pipelined x2-PPL2	2	No	Virtex 5
JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	Yes	Virtex 5
Keccak	256	Throughput/Area	Pipelined x2-PPL4	4	No	Virtex 5
Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	No	Virtex 5
BLAKE	256	Throughput/Area	Pipelined x1-PPL4	4	Yes	Virtex 5
Skein	256	Throughput/Area	Pipelined x4-PPL2	2	Yes	Virtex 5
Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	Yes	Virtex 5
JH	256	Throughput/Area	Pipelined x2-PPL2	2	Yes	Virtex 5
Groestl	256	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes	Virtex 5
BLAKE	256	Throughput/Area	Pipelined x1-PPL4	4	Yes	Virtex 5
BLAKE	256	Throughput/Area	Pipelined x1-PPL2	2	Yes	Virtex 5
BLAKE	256	Throughput/Area	Pipelined /2(v)-PPL4	4	Yes	Virtex 5
BLAKE	256	Throughput/Area	Pipelined /2(v)-PPL2	2	Yes	Virtex 5
BLAKE	256	Throughput/Area	Pipelined x1-PPL4	4	No	Virtex 5
BLAKE	256	Throughput/Area	Pipelined /2(v)-PPL4	4	No	Virtex 5
BLAKE	256	Throughput/Area	Pipelined /2(v)-PPL2	2	No	Virtex 5
Skein	256	Throughput/Area	Pipelined x4-PPL5	5	No	Virtex 5
Skein	256	Throughput/Area	Pipelined x4-PPL2	2	Yes	Virtex 5
Skein	256	Throughput/Area	Pipelined x4-PPL2	2	No	Virtex 5
Keccak	256	Throughput/Area	Pipelined x2-PPL4	4	Yes	Virtex 5
Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	Yes	Virtex 5
JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No	Virtex 5
JH	256	Throughput/Area	Pipelined x2-PPL2 (MEM)	2	No	Virtex 5
Algorithm	256	Primary Opt Target	Arch Type	>1	Padding	Virtex 5

Filtering the Results:
Hash Size=256, Max #Streams > 1, Family = Virtex 5

Sorting Results According to the Number of CLB Slices in the Ascending Order

Timing		Resource Utilization						
TP [Mbits/s]	Impl Freq [MHz]	CLB Slices ▲	LEs	ALUTs	LUTs	Flip Flops	MULTs	DSPs
5,972	256.608	1,473	-	-	5,052	3,011	-	0
4,711	391.083	1,842	-	-	5,138	6,206	-	0
5,338	198.098	1,858	-	-	4,755	5,744	-	0
2,482	92.090	1,934	-	-	3,987	5,160	-	48
16,121	355.619	1,950	-	-	5,330	6,254	-	0
4,873	180.832	2,030	-	-	5,267	5,672	-	0
11,562	255.037	2,035	-	-	5,446	6,315	-	0
7,041	295.683	2,099	-	-	6,461	6,228	-	0
3,510	195.389	2,107	-	-	6,867	5,344	-	0
12,523	276.243	2,123	-	-	5,433	6,258	-	0
3,506	195.160	2,136	-	-	6,794	5,324	-	0
3,838	318.573	2,147	-	-	5,640	6,512	-	0
8,289	348.068	2,312	-	-	6,330	7,480	-	0
5,143	143.143	2,353	-	-	6,942	5,553	-	0
17,677	194.970	2,390	-	-	6,921	6,252	-	0
9,073	177.211	2,680	-	-	5,135	4,041	-	0
12,479	243.724	2,971	-	-	11,153	4,933	-	0
4,761	134.825	2,976	-	-	8,012	5,769	-	0
8,526	353.857	3,085	-	-	7,926	11,312	-	0

Sorting Results According to Throughput (in Mbits/s) in the Descending Order

Timing		Resource Utilization						
TP [Mbits/s]	Impl Freq [MHz]	CLB Slices	LEs	ALUTs	LUTs	Flip Flops	MULTs	DSPs
26,690	294.377	3,714	-	-	9,557	12,429	-	0
21,717	239.521	3,764	-	-	9,765	12,437	-	0
17,677	194.970	2,390	-	-	6,921	6,252	-	0
16,353	319.387	4,177	-	-	12,591	9,788	-	0
16,121	355.619	1,950	-	-	5,330	6,254	-	0
15,015	293.255	4,587	-	-	13,225	10,116	-	0
13,382	261.370	3,172	-	-	11,567	5,097	-	0
12,523	276.243	2,123	-	-	5,433	6,258	-	0
12,479	243.724	2,971	-	-	11,153	4,933	-	0
11,562	255.037	2,035	-	-	5,446	6,315	-	0
9,073	177.211	2,680	-	-	5,135	4,041	-	0
8,526	353.857	3,085	-	-	7,926	11,312	-	0
8,526	353.857	3,085	-	-	7,926	11,312	-	0
8,289	348.068	2,312	-	-	6,330	7,480	-	0
7,547	210.040	3,495	-	-	9,231	10,298	-	0
7,510	209.030	3,526	-	-	9,476	10,287	-	0
7,077	262.605	3,840	-	-	8,646	11,981	-	0
7,041	295.683	2,099	-	-	6,461	6,228	-	0

Ordered Listing with Multiple Results per Each Algorithm

Show 25 entries

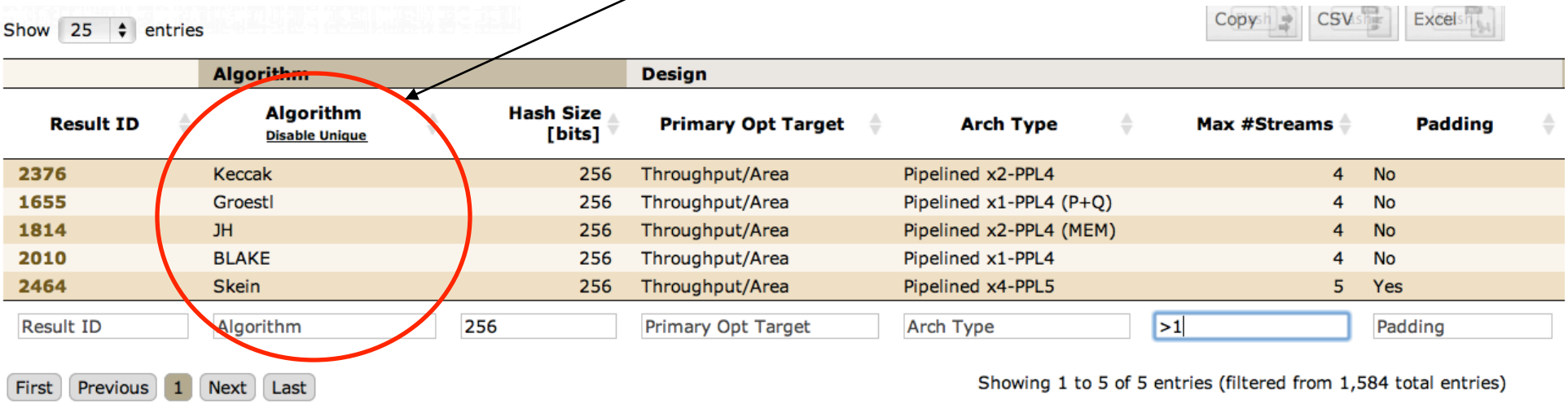
Copy CSV Excel

	Algorithm		Design			
Result ID	Algorithm Enable Unique	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding
2376	Keccak	256	Throughput/Area	Pipelined x2-PPL4	4	No
1863	Keccak	256	Throughput/Area	Pipelined x2-PPL4	4	Yes
2449	Keccak	256	Throughput/Area	Pipelined x2-PPL2	2	No
1655	Groestl	256	Throughput/Area	Pipelined x1-PPL4 (P+Q)	4	No
2372	Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	No
1657	Groestl	256	Throughput/Area	Pipelined x1-PPL4 (P+Q)	4	Yes
1645	Groestl	256	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
1843	Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	Yes
1643	Groestl	256	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	No
2122	Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	Yes
2120	Groestl	256	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes
1814	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2409	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	Yes
1800	JH	256	Throughput/Area	Pipelined x2-PPL2 (MEM)	2	No
2010	BLAKE	256	Throughput/Area	Pipelined x1-PPL4	4	No
2041	BLAKE	256	Throughput/Area	Pipelined x1-PPL4	4	Yes
2464	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
2121	JH	256	Throughput/Area	Pipelined x2-PPL2	2	Yes
1939	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	No
1770	JH	256	Throughput/Area	Unrolled x2 (MEM)	2	Yes
1920	Skein	256	Throughput/Area	Pipelined x4-PPL2	2	No
2133	BLAKE	256	Throughput/Area	Pipelined x1-PPL4	4	Yes
2037	BLAKE	256	Throughput/Area	Pipelined x1-PPL2	2	Yes
1922	Skein	256	Throughput/Area	Pipelined x4-PPL2	2	Yes
2456	BLAKE	256	Throughput/Area	Pipelined x1-PPL2	2	No
Result ID	Algorithm	256	Primary Opt Target	Arch Type	>1	Padding

First Previous 1 2 Next Last

Showing 1 to 25 of 32 entries (filtered from 1,584 total entries)

Ordered Listing with a Single-Best (Unique) Result per Each Algorithm



Show 25 entries

Copy CSV Excel

	Algorithm		Design			
Result ID	Algorithm Disable Unique	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding
2376	Keccak	256	Throughput/Area	Pipelined x2-PPL4	4	No
1655	Groestl	256	Throughput/Area	Pipelined x1-PPL4 (P+Q)	4	No
1814	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No
2010	BLAKE	256	Throughput/Area	Pipelined x1-PPL4	4	No
2464	Skein	256	Throughput/Area	Pipelined x4-PPL5	5	Yes
Result ID	Algorithm	256	Primary Opt Target	Arch Type	>1	Padding

First Previous 1 Next Last

Showing 1 to 5 of 5 entries (filtered from 1,584 total entries)

Details of Result ID 2469

Algorithm

Transformation Category:	Cryptographic
Transformation:	Hash
Group:	SHA-3 Round 3
Algorithm:	JH
Hash Size [bits]:	512
Message Block Size [bits]:	512
Other Parameters:	-
Specification:	**JH_FinalRnd.zip**
Formula for Message Size After Padding:	-

Design

Design ID:	**247**
Primary Optimization Target:	Throughput/Area
Secondary Optimization Target:	Throughput
Architecture Type:	Pipelined x2-PPL2 (MEM)
Description Language:	VHDL
Use of Megafunctions or Primitives:	No
List of Megafunctions or Primitives:	-
Maximum Number of Streams Processed in Parallel:	2
Number of Clock Cycles per Message Block in a Long Message:	43
Datapath Width [bits]:	512
Padding:	Yes
Minimum Message Unit:	1 byte
Input Bus Width [bits]:	128
Output Bus Width [bits]:	64
Implementation URL:	**index.php?id=source_codes**
Shared I/O Bus:	No
Throughput Formula:	1024/(43*T)
Execution Time Formula:	3+43*N+8
Source Available:	Yes

Measured Power [mW]:	-
Measured Dynamic Power [mW]:	-
Measured Static Power [mW]:	-
Measured Energy/Bit [mJ/Gbit]:	-
Operating Conditions used for Measurement (V, Temp, Etc):	-

Tool Information

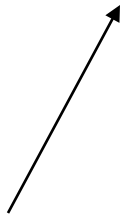
Synthesis Tool:	Xilinx XST
Synthesis Tool Version:	13.1
Synthesis Tool Options:	-generics { UF=2 PPL=2 HS=512 } -dsp_utilization_ratio 0 -opt_level 1 -bram_utilization_ratio 0
Implementation Tool:	Xilinx ISE
Implementation Tool Version:	13.1
Map Options:	-c 100 -cm area -t 21
Implementation Tool Options:	-ol high

Credits

Primary Designer Name(s):	Ekawat Homsirikamol
Primary Designer Email(s):	ehomsiri@gmu.edu
Co-designer Name(s):	Marcin Rogawski, Kris Gaj
Co-designer Email(s):	mrogawsk@gmu.edu, kgaj@gmu.edu
Primary Designer Affiliation:	CERG @ GMU
Co-Designer Affiliation:	CERG @ GMU

Other

Result Replication Files:	**link**
Result Entry Date:	2012-06-20
Result Modify Date:	2012-06-20
Design Entered By:	ice
Hidden:	No



Link to a Script that Allows Replicating Results with a Single-Run of Standard FPGA Tools

GMU Web Page with VHDL Source Codes and Block Diagrams of the SHA-3 Candidates and SHA-2

GMU Source Codes

- **Introduction**

The VHDL source codes provided below have been generated by members of the Cryptographic Engineering Research Group (CERG) at George Mason University in the period from January 2010 to present.

We reuse the same VHDL code to implement both 256 and 512 variants of all hash functions with the help of generics. A user needs only to select an appropriate value, HASH_SIZE_256 or HASH_SIZE_512, for the generic 'h' or 'hs' present in the top level entity. The VHDL file containing the top level entity of the given hash function is indicated in the source_list.txt file present in the 'sources' folder obtained from the zip file.

- **Source Codes for the SHA-3 Round 3 Candidates & SHA-2 - The Third SHA-3 Candidate Conference Release, March 2012**

Assumptions:

A. All architectures defined in
K. Gaj, E. Homsirikamol, M. Rogawski, R. Shahid, and M.U. Sharif, "Comprehensive Evaluation of High-Speed and Medium-Speed Implementations of Five SHA-3 Finalists Using Xilinx and Altera FPGAs," The 3rd SHA-3 Candidate Conference, Washington, D.C., March 22-23, 2012: **paper, slides**
B. **The GMU Interface and Communication Protocol Used in the Implementations of the SHA-3 Round 3 Candidates (non-pipelined architectures).**
C. Primary Optimization Target: Throughput/Area; Secondary Optimization Target: Throughput.
D. No salt, No special modes of operation.
E. No use of embedded resources, such as multipliers, DSP units and Block Memories.

- **BLAKE**

BLAKE_readme.txt

Source Codes with and without Padding	Supported Architectures	Architecture Notation	Block Diagrams	Release Date
BLAKE_folded_pad.zip **BLAKE_folded.zip**	Basic iterative Folded horizontally	x1, /k(h), k=2,4	**BLAKE_x1_diagrams.zip** **BLAKE_fh2_diagrams.zip** **BLAKE_fh4_diagrams.zip**	03/23/2012
BLAKE_fh4v4_pad.zip **BLAKE_fh4v4.zip**	Folded horizontally and vertically with internal state stored in memory	/4(h)/4(v)-m	**BLAKE_fh4v4_diagrams.zip**	03/23/2012
BLAKE_PPL_pad.zip **BLAKE_PPL.zip**	Pipelined	x1-PPLn, n=2,4 /2(h)-PPLn, n=2,4		03/23/2012

Comparing Two Results with Each Other

Compare Selected | SHA-3 Round 3 | SHA-3 Round 3 & SHA-2 | SHA-3 Round 2 | SHA-3 Round 2 & SHA-2

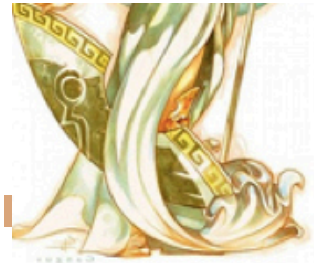
Show 25 entries

Copy | CSV | Excel

	Algorithm		Design				Platform
Result ID	Algorithm Enable Unique	Hash Size [bits]	Primary Opt Target	Arch Type	Max #Streams	Padding	Far
2376	Keccak	256	Throughput/Area	Pipelined x2-PPL4	4	No	Virtex 5
1863	Keccak	256	Throughput/Area	Pipelined x2-PPL4	4	Yes	Virtex 5
2449	Keccak	256	Throughput/Area	Pipelined x2-PPL2	2	No	Virtex 5
1655	Groestl	256	Throughput/Area	Pipelined x1-PPL4 (P+Q)	4	No	Virtex 5
2372	Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	No	Virtex 5
1657	Groestl	256	Throughput/Area	Pipelined x1-PPL4 (P+Q)	4	Yes	Virtex 5
1645	Groestl	256	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes	Virtex 5
1843	Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	Yes	Virtex 5
1643	Groestl	256	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	No	Virtex 5
2122	Keccak	256	Throughput/Area	Pipelined x1-PPL2	2	Yes	Virtex 5
2120	Groestl	256	Throughput/Area	Pipelined x1-PPL2 (P+Q)	2	Yes	Virtex 5
1814	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	No	Virtex 5
2409	JH	256	Throughput/Area	Pipelined x2-PPL4 (MEM)	4	Yes	Virtex 5
1800	JH	256	Throughput/Area	Pipelined x2-PPL2 (MEM)	2	No	Virtex 5

Comparing Two Results with Each Other: Outcome of the Comparison

Datapath Width [bits]:	1600	1600
Padding:	No	Yes
Minimum Message Unit:		1 byte
Input Bus Width [bits]:	128	128
Output Bus Width [bits]:	64	64
Implementation URL:	**index.php?id=source_codes**	**index.php?id=source_codes**
Shared I/O Bus:	No	No
Throughput Formula:	2176/(48*T)	2176/(48*T)
Execution Time Formula:	3+48*N+4	3+48*N+4
Source Available:	Yes	Yes
Source Code Files:	**link**	**link**
Design Entry Date:	2012-02-16 @ 18:54 EST	2012-02-16 @ 18:54 EST
Design Modify Date:	2012-04-10 @ 20:52 EST	2012-04-10 @ 20:53 EST
Design Name:	Keccak_x1_PPL2 (256) SHA3C3	Keccak_x1_PPL2 (256) Pad SHA3C3
Comments:		
Platform		
Device Vendor:	Xilinx	Xilinx
Family:	Virtex 5	Virtex 5
Device:	xc5vlx30ff676-3	xc5vlx30ff676-3
Timing		
Throughput [Mbits/s]:	16121	12523
Requested Synthesis Clock Frequency [MHz]:	377	283.9
Synthesis Clock Frequency [MHz]:	377.601	294.633
Requested Implementation Clock Frequency [MHz]:	377	283.9



Hash Function FPGA Ranking

Show Help

- About
- All ASIC Results
- All FPGA Results
- FPGA Rankings
- Login

Result Filtering

Algorithm Group
- (•) Round 3 SHA-3 Candidates & SHA-2
- ( ) Round 2 SHA-3 Candidates & SHA-2

Implementation Type:
- ( ) High Speed Implementations, Single Message Architectures
- (•) High Speed Implementations, All Architectures
- ( ) Low Area Implementations

Hash Size:
- (•) 256
- ( ) 512

Padding:
- (•) No
- ( ) Yes
- ( ) Any

Family: Virtex 6

Portability Resources:
- [x] Without Embedded Resources (Block Memories, DSP Units, etc.)
- [ ] Without Primitives or Megafunctions

Min Area: 0

Max Area: 1000000

Source:
- [ ] Source Available

Ranking:
- (•) Throughput/Area
- ( ) Throughput
- ( ) Area

The compared implementations may process different number of messages in parallel.
Please note that codes with primitives, megafunctions, or embedded resources are not fully portable.

Setting Criteria for Ranking of the Candidates (FPGA Implementations)

Results of Ranking according to the Given Criteria for FPGA Implementations

The compared implementations may process different number of messages in parallel.
Please note that codes with primitives, megafunctions, or embedded resources are not fully portable.

Update

Compare Selected

Show 25 entries

Copy CSV Excel

Result ID	Algorithm Disable Unique	Arch Type	TP/CLBs [(Mbits/s)/CLBs]	TP [Mbits/s]	CLB Slices	BRAMs	DSPs	Primary Designer Affiliation
2197	Keccak	Basic Iterative	14.940	13,670	915	0	0	NUST, Pakistan
1972	SHA-2	Quasi-pipelined	6.835	1,634	239	0	0	CERG @ GMU
1255	JH	Basic Iterative	6.729	5,700	847	0	0	CERG @ GMU
2189	Groestl	Basic Iterative	6.558	9,620	1,467	0	0	NUST, Pakistan
1919	Skein	Pipelined x4-PPL2	3.816	6,212	1,628	0	0	CERG @ GMU
2009	BLAKE	Pipelined x1-PPL4	3.184	8,056	2,530	0	0	CERG @ GMU
Result ID	Algorithm	Arch Type	TP/CLBs [(Mbits/s)/CLBs]	TP [Mbits/s]	[0,1000000]	NULLOK [0,0]	NULLOK [0,0]	Primary Designer Affiliation

First Previous 1 Next Last

Showing 1 to 6 of 6 entries (filtered from 1,584 total entries)

ATHENA

AUTOMATED TOOL FOR HARDWARE EVALUATION

Hash Function ASIC Ranking

Show Help

- About
- All ASIC Results
- All FPGA Results
- ASIC Rankings
- FPGA Rankings
- Login

Result Filtering

Algorithm Group:
- (•) Round 3 SHA-3 Candidates & SHA-2
- ( ) Round 2 SHA-3 Candidates & SHA-2

Implementation Type:
- ( ) 130nm Process, Virginia Tech, Optimization for Throughput/Area
- (•) 65nm Process, ETH Zurich and GMU, Optimization for Throughput/Area
- ( ) 65nm Process, ETH Zurich, Optimization for Minimum Area at Throughput = 2.488 Gbit/s

Ranking:
- (•) Throughput/Area
- ( ) Throughput

[Update]

Setting Criteria for Ranking of the Candidates (ASIC Implementations)

Results of Ranking according to the Given Criteria for ASIC Implementations

Show 25 entries

Copy CSV Excel

Result ID	Algorithm (Disable Unique)	Arch Type	Measured TP/Area [(Mbits/s)/kGEs]	Measured TP [Mbits/s]	Post-Layout Area [kGE]	Post-Layout Power [mW]	Measured Energy/Bit [mJ/Gbit]	Primary Designer Affiliation
1528	Keccak	Basic Iterative	512.540	23,737	46	8.160	-	ETH Zurich
1530	SHA-2	Quasi-pipelined	203.460	5,115	25	9.160	-	ETH Zurich
1533	JH	Basic Iterative	142.260	7,732	54	17.800	-	ETH Zurich
1531	BLAKE	Folded /2(h)	124.050	5,350	43	16.470	-	ETH Zurich
1535	Skein	Unrolled x4	90.530	6,509	72	26.190	-	ETH Zurich
1532	Groestl	Basic Iterative (P/Q)	70.100	11,235	160	46.010	-	ETH Zurich

Result ID | Algorithm | Arch Type | Measured TP/Area [(M | Measured TP | Post-Layc | Post-Layo | Measured Ene | Primary Designer Affiliation

First Previous 1 Next Last

Showing 1 to 6 of 6 entries (filtered from 18 total entries)

Future Work

ATHENa Database of Results

- Results including SCA countermeasures
- Databases of results regarding
  - Symmetric-key ciphers
  - Public-key cryptosystems (RSA, ECC, etc.)
  - Pairing-based cryptosystems
  - TRNGs, PUFs?

ATHENa Tool

- Additional FPGA vendors
- Power analysis
- Application to comparison and optimization of other cryptographic primitives (e.g., public key cryptosystems)
- Adapting ATHENa to other application domains (Digital Signal Processing, communications, etc.)

Thank you!

Questions?

Questions?

ATHENa: http:/cryptography.gmu.edu/athena