Fair and Comprehensive Methodology for Comparing Hardware Performance of Fourteen Round Two SHA-3 Candidates using FPGAs

Kris Gaj,
Ekawat Homsirikamol, and
Marcin Rogawski
George Mason University
U.S.A.

Co-Authors

Ekawat Homsirikamol
a.k.a "Ice"

Marcin Rogawski

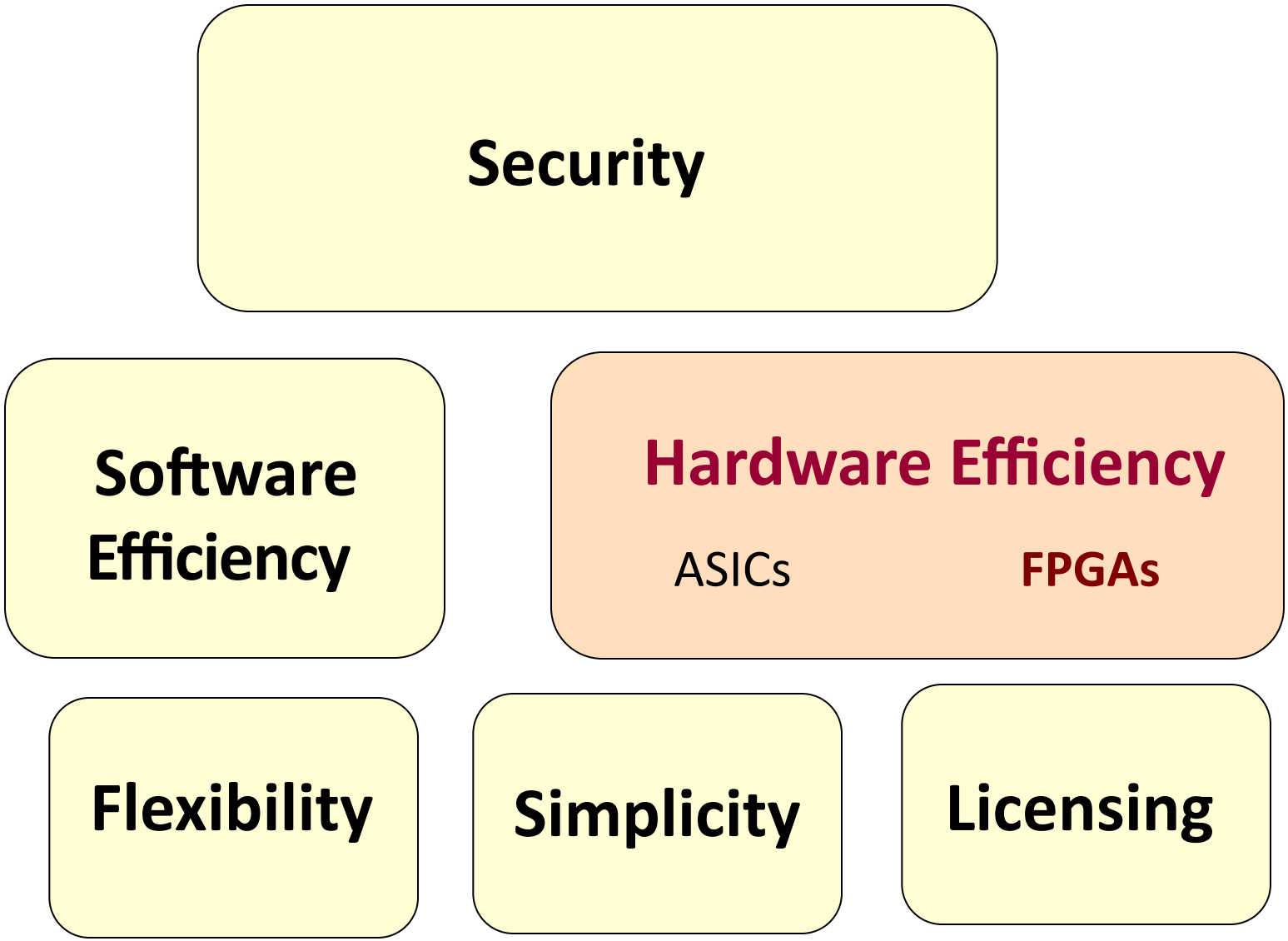
Developed optimized VHDL implementations of 14 Round 2 SHA-3 candidates + SHA-2 in two variants each (256 & 512-bit output), for some functions using several alternative architectures

Outline

- Motivation & Goals
- Methodology
- Results
- Comparison with Other Groups
- Future Work & Conclusions


Motivation
&
Goals

SHA-3 Contest - NIST Evaluation Criteria



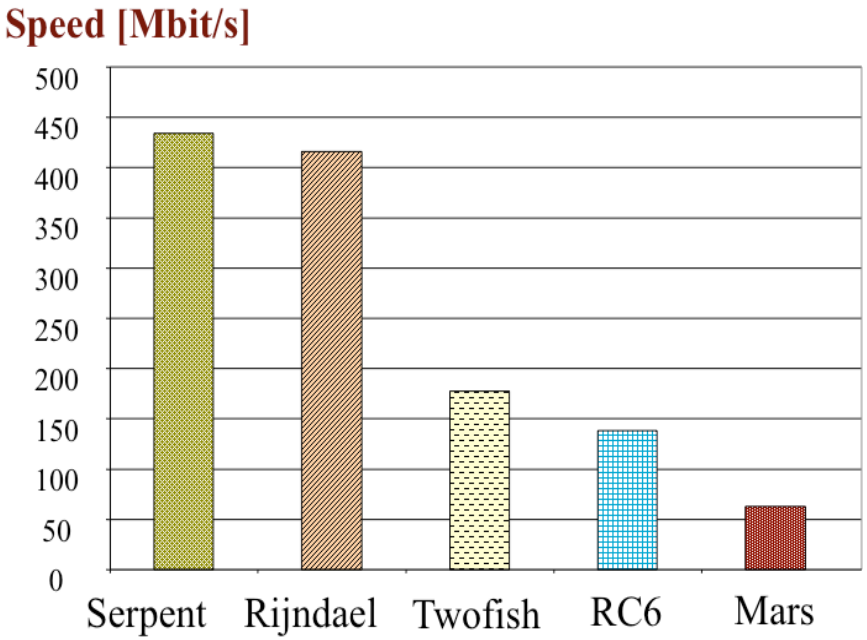
Results of Security Evaluation So Far
SHA-3 Zoo Page

Hash Name	Principal Submitter	Best Attack on Main NIST Requirements	Best Attack on other Hash Requirements
BLAKE	Jean-Philippe Aumasson		
Blue Midnight Wish	Svein Johan Knapskog		
CubeHash	Daniel J. Bernstein	preimage	
ECHO	Henri Gilbert		
Fugue	Charanjit S. Jutla		
Grøstl	Lars R. Knudsen		
Hamsi	Özgül Küçük		
JH	Hongjun Wu	preimage	
Keccak	The Keccak Team		
Luffa	Dai Watanabe		
Shabal	Jean-François Misarsky		
SHAvite-3	Orr Dunkelman		
SIMD	Gaëtan Leurent		
Skein	Bruce Schneier		

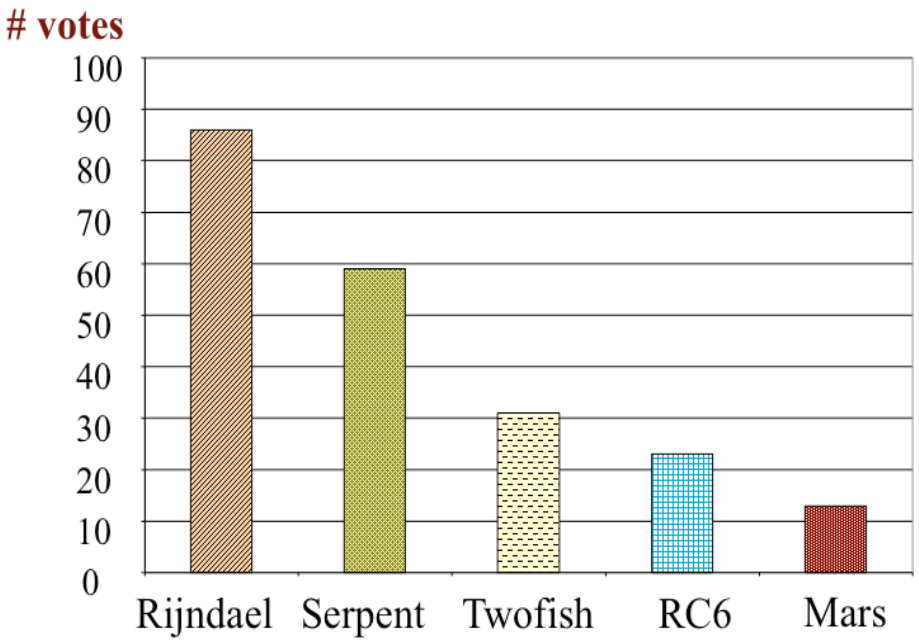
Lessons from the Past - AES Contest – 1997-2000

Round 2 of AES Contest, 2000

Speed in FPGAs



Votes at the AES 3 conference



Our Goals

- **Fair and comprehensive methodology** for evaluation of hardware performance in FPGAs

- **High-speed** fully autonomous implementations of all **14 SHA-3 candidates** & SHA-2 **256-bit variants** optimized for the **maximum throughput to area ratio**

- Commonly acceptable **recommendations to NIST**

Methodology

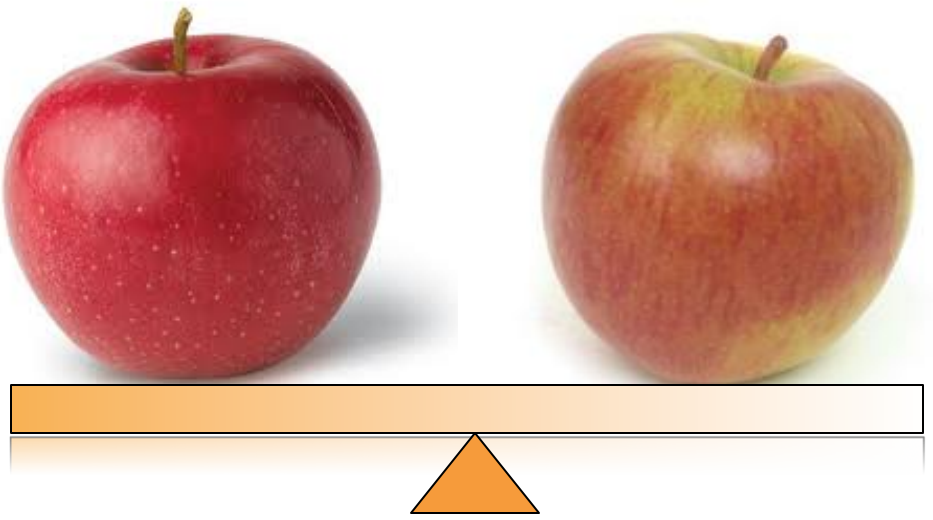
Comprehensive Evaluation

- two major vendors: Altera and Xilinx (~90% of the market)
- multiple high-performance and low-cost families

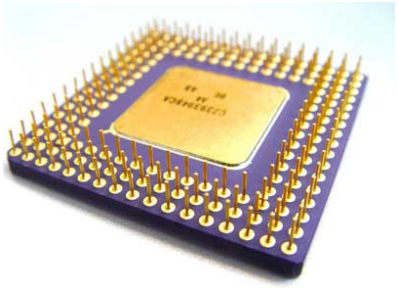
	Altera		Xilinx	
Technology	Low-cost	High-performance	Low-cost	High-performance
90 nm	Cyclone II	Stratix II	Spartan 3	Virtex 4
65 nm	Cyclone III	Stratix III		Virtex 5

Uniform Evaluation

- Language: VHDL
- Tools: FPGA vendor tools
- Interface
- Performance Metrics
- Design Methodology
- Benchmarking



Why Interface Matters?



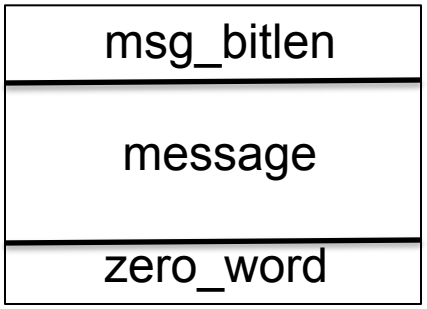
- Pin limit

Total number of i/o ports ≤ Total number of an FPGA i/o pins

- Support for the maximum throughput

Time to load the next message block ≤ Time to process previous block

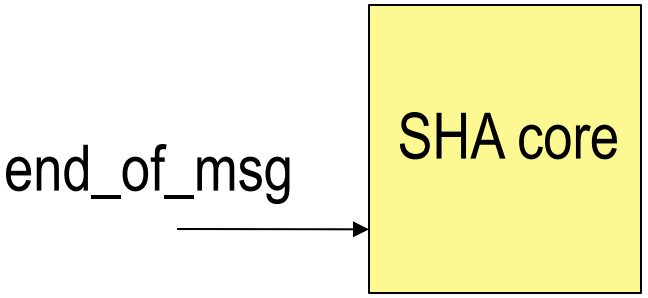
Interface: Two possible solutions



Length of the message communicated at the beginning

+ easy to implement passive source circuit

− area overhead for the counter of message bits

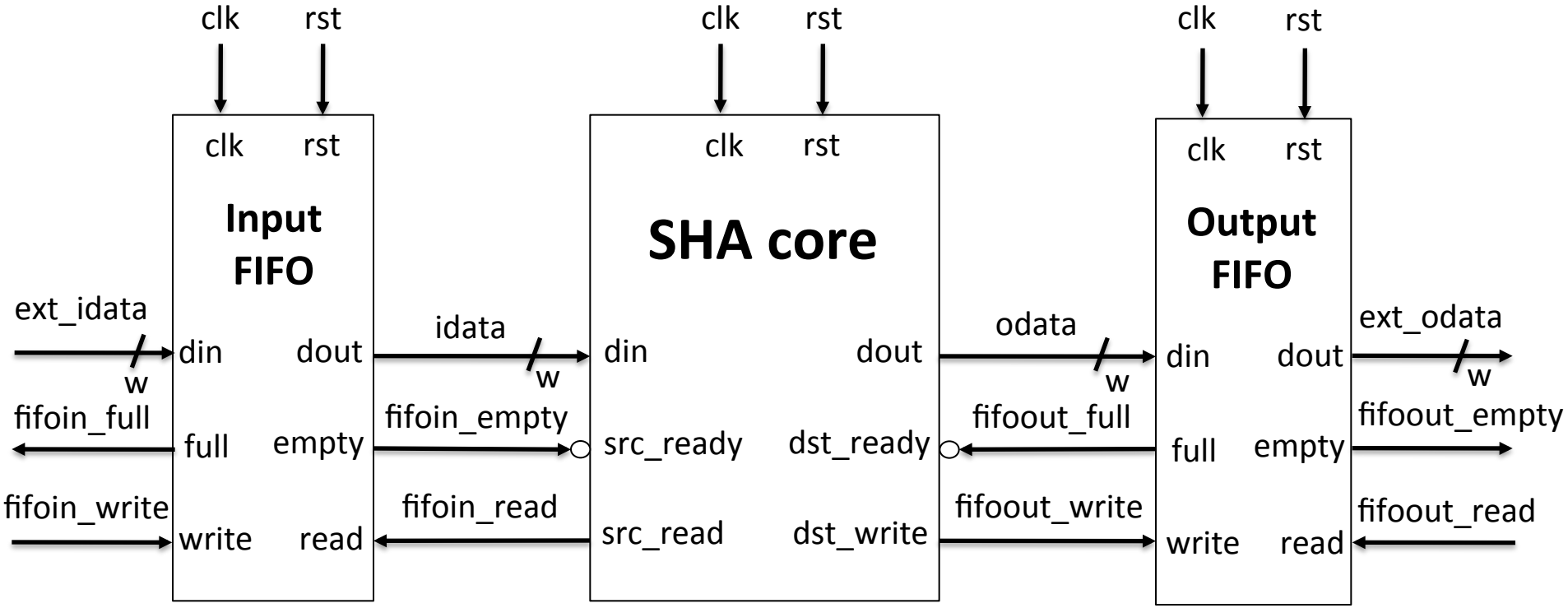


Dedicated end of message port

− more intelligent source circuit required

+ no need for internal message bit counter

SHA Core: Interface & Typical Configuration



- SHA core is an active component; surrounding FIFOs are passive and widely available
- Input interface is separate from an output interface
- Processing a current block, reading the next block, and storing a result for the previous message can be all done in parallel

Performance Metrics

Primary

1. Throughput
   (single long message)

3. Throughput / Area

Secondary

2. Area

3. Hash Time for
   Short Messages
   (up to 1000 bits)

Performance Metrics - Area

$$Resource\ Utilization_{\text{Spartan3}} = (\#CLB\ slices,\ \#BRAMs,\ \#MULs)$$

$$Resource\ Utilization_{\text{Cyclone III}} = (\#LE,\ \#memory\_bits,\ \#MULs).$$

We force these vectors to look as follows through the synthesis and implementation options:

$$Resource\ Utilization_{\text{Spartan3}} = (\#CLB\ slices,\ 0,\ 0)$$

$$Resource\ Utilization_{\text{Cyclone III}} = (\#LE,\ 0,\ 0).$$

Area

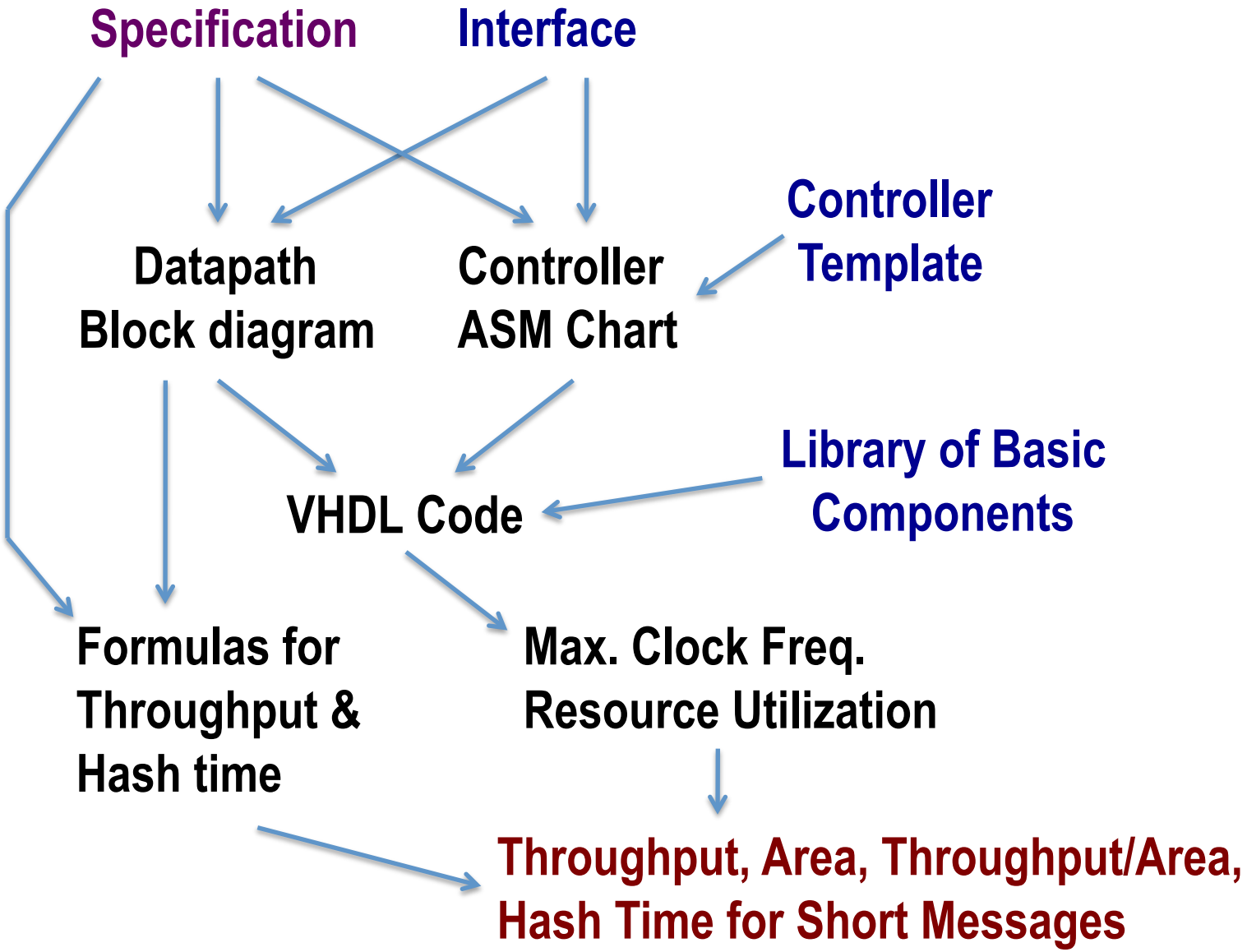
Choice of Optimization Target

Primary Optimization Target: **Throughput to Area Ratio**

Features:

- practical: good balance between speed and cost
- very reliable guide through the entire design process, facilitating the choice of
  - high-level architecture
  - implementation of basic components
  - choice of tool options
- leads to high-speed, close-to-maximum-throughput designs

Our Design Flow



Basic Operations of 14 SHA-3 Candidates

Function	NTT	Linear code	S-box	GF MUL	MUL	mADD	ADD /SUB	Boolean
BLAKE						mADD3	ADD	XOR
BMW						mADD17	ADD,SUB	XOR
CubeHash							ADD	XOR
ECHO			AES 8x8	x02, x03				XOR
Fugue			AES 8x8	x04..x07				XOR
Groestl			AES 8x8	x02..x05, 0x07				XOR
Hamsi		LC	Serpent 4x4					XOR
JH			4x4	x2, x5				XOR
Keccak								NOT,AND,XOR
Luffa			4x4	x02				XOR
Shabal					x3, x5		ADD,SUB	NOT,AND,XOR
SHAvite-3			AES 8x8	x02, x03				NOT,XOR
SIMD	NTT				x185, x233	mADD3	ADD	NOT,AND,OR
Skein							ADD	XOR
SHA-256						mADD5		NOT,AND,XOR

NTT – Number Theoretic Transform, GF MUL – Galois Field multiplication,
MUL – integer multiplication, mADDn – multioperand addition with n operands

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

http://cryptography.gmu.edu/athena

Benchmarking open-source tool, written in Perl, aimed at an

AUTOMATED generation of
OPTIMIZED results for
MULTIPLE FPGA platforms

Under development at
George Mason University.

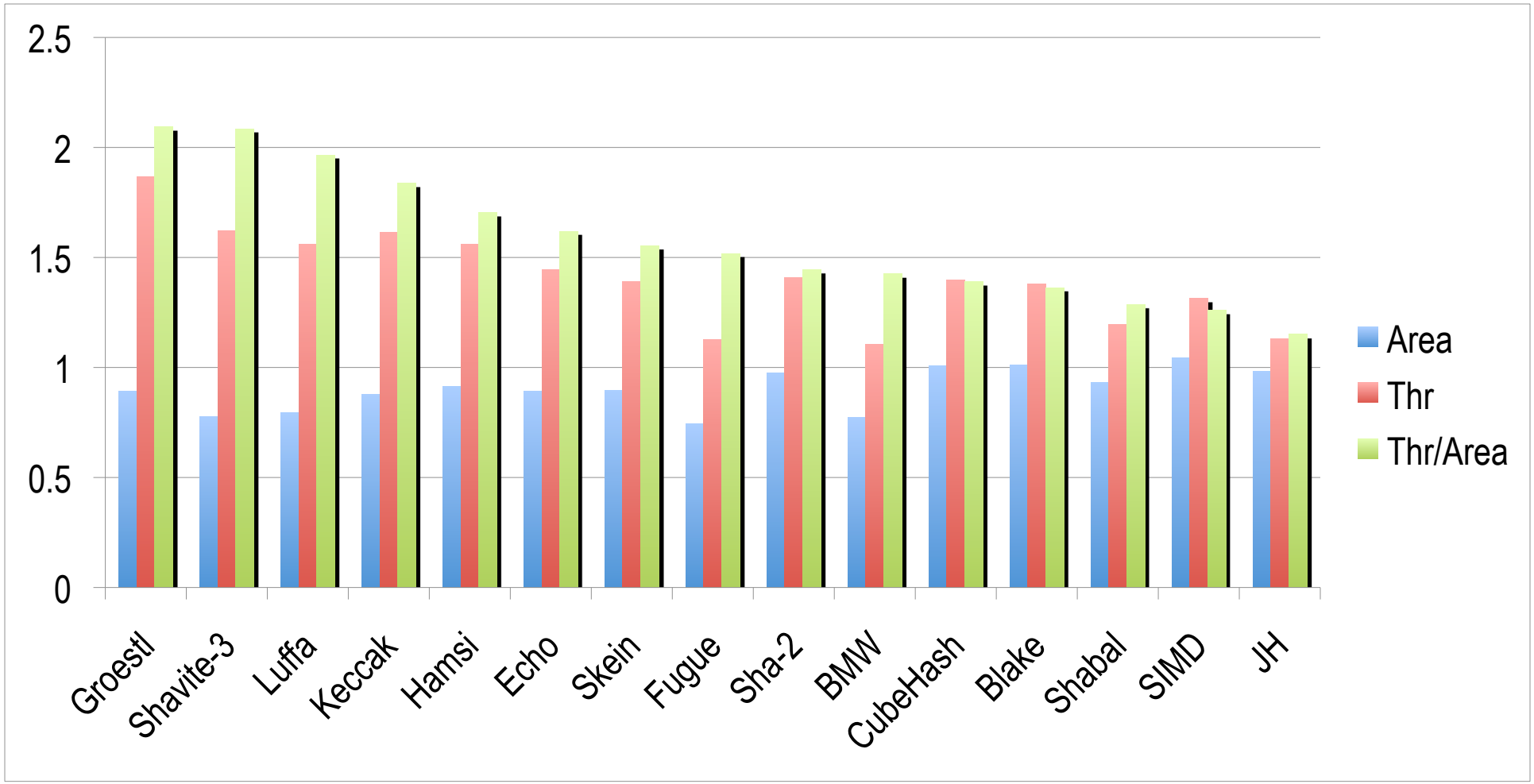
Generation of Results Facilitated by ATHENa

- batch mode of FPGA tools

vs.

- ease of extraction and tabulation of results
  - Excel, CSV (available), LaTeX (coming soon)
- optimized choice of tool options

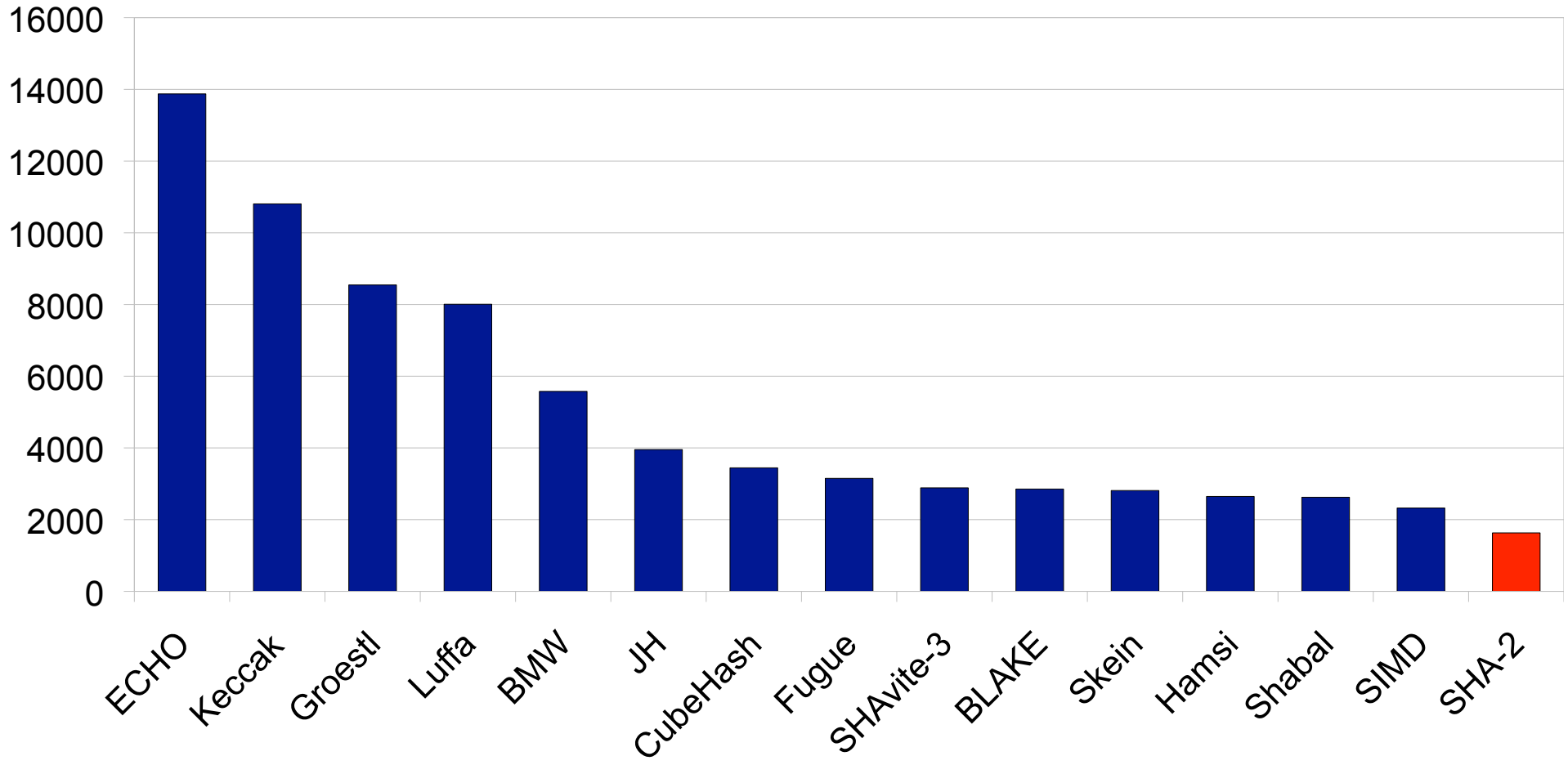
Relative Improvement of Results from Using ATHENa Virtex 5, 256-bit Variants of Hash Functions



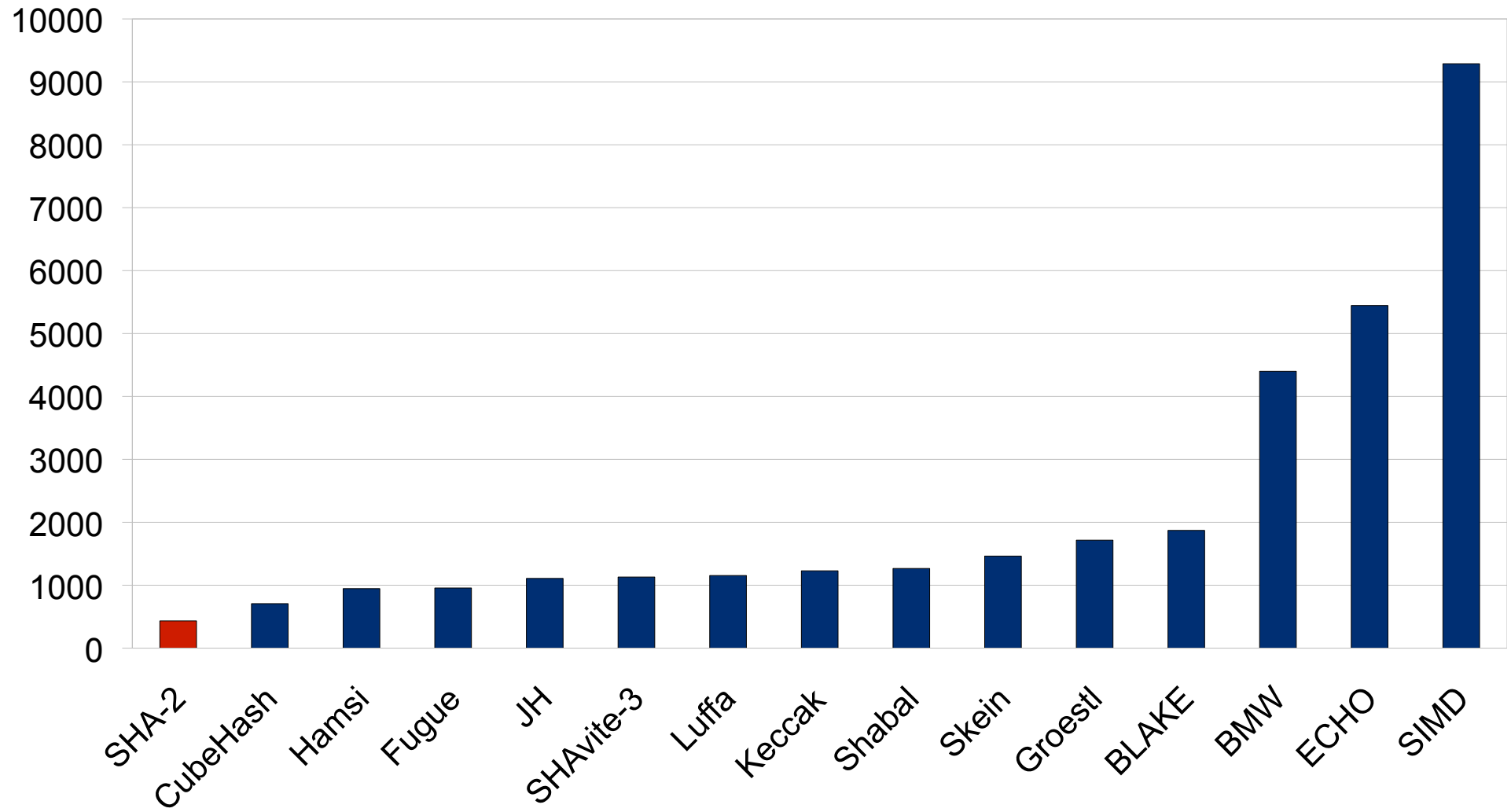
Ratios of results obtained using ATHENa suggested options vs. default options of FPGA tools


Results

Throughput [Mbit/s]
Virtex 5, 256-bit variants of algorithms



Area [CLB slices]
Virtex 5, 256-bit variants of algorithms



Normalization & Compression of Results

- **Absolute result**

  e.g., throughput in Mbits/s, area in CLB slices

- **Normalized result**

$$normalized\_result = \frac{result\_for\_SHA-3\_candidate}{result\_for\_SHA-2}$$

- **Overall normalized result**

  Geometric mean of normalized results for

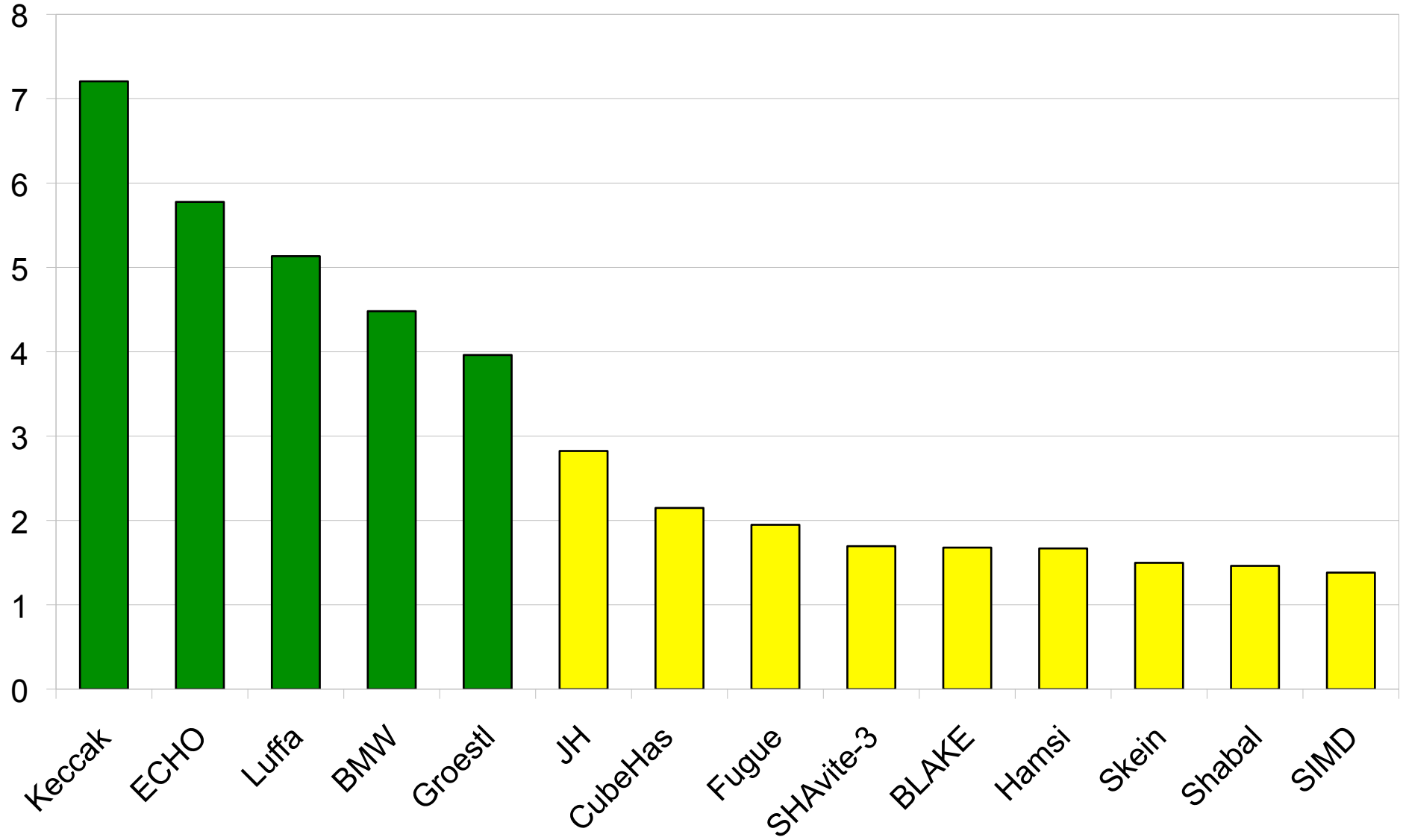
  all investigated FPGA families

Normalized Throughput & Overall Normalized Throughput

Candidate	Spartan 3	Virtex 4	Virtex 5	Cyclone II	Cyclone III	Stratix II	Stratix III	Overall
Keccak	6.10	6.37	6.63	8.56	7.91	7.23	8.01	7.21
ECHO	4.30	5.41	8.51	N/A	6.25	5.20	5.79	5.78
Luffa	5.16	5.14	4.91	5.58	4.94	5.02	5.21	5.13
BMW	3.00	4.39	3.42	4.50	4.31	5.53	5.02	4.48
Groestl	3.60	3.97	5.32	3.68	3.62	4.24	3.93	4.02
JH	2.58	2.68	2.43	2.84	3.17	3.06	3.10	2.82
CubeHash	2.03	2.10	2.11	2.12	2.13	2.30	2.25	2.15
Fugue	1.77	1.62	1.93	1.95	1.94	2.15	2.36	1.95
BLAKE	1.57	1.45	1.75	1.59	1.62	2.02	1.81	1.68
Hamsi	1.35	1.49	1.62	1.82	1.96	1.66	1.88	1.67
SHAvite-3	1.64	1.46	1.77	1.51	1.58	1.89	2.11	1.70
Skein	1.39	1.46	1.72	1.45	1.48	1.53	1.48	1.50
Shabal	0.89	1.62	1.61	1.63	1.41	1.73	1.55	1.46
SIMD	1.37	1.15	1.43	1.41	1.36	1.69	1.61	1.38

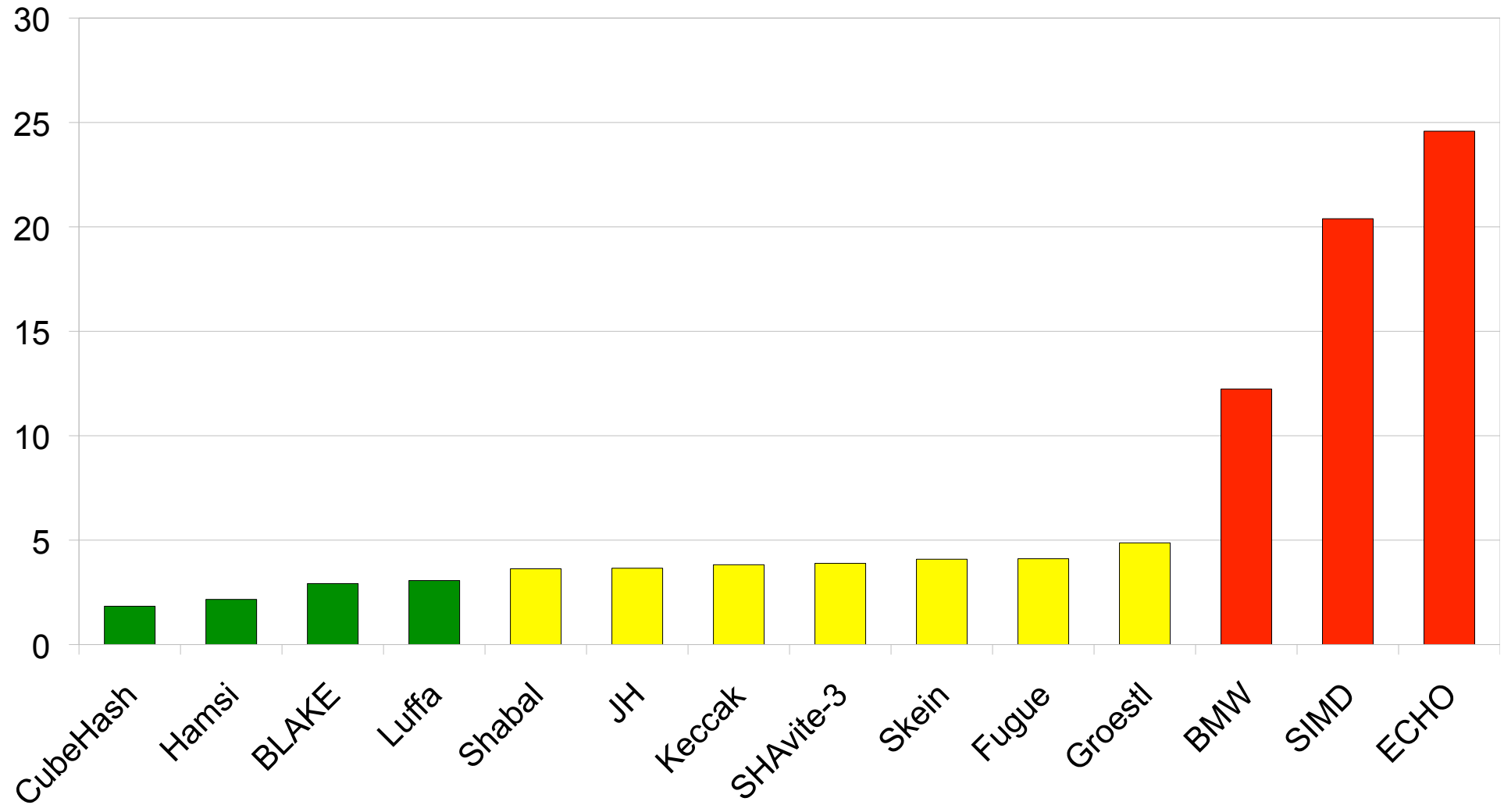
Overall Normalized Throughput: 256-bit variants of algorithms

Normalized to SHA-256, Averaged over 7 FPGA families



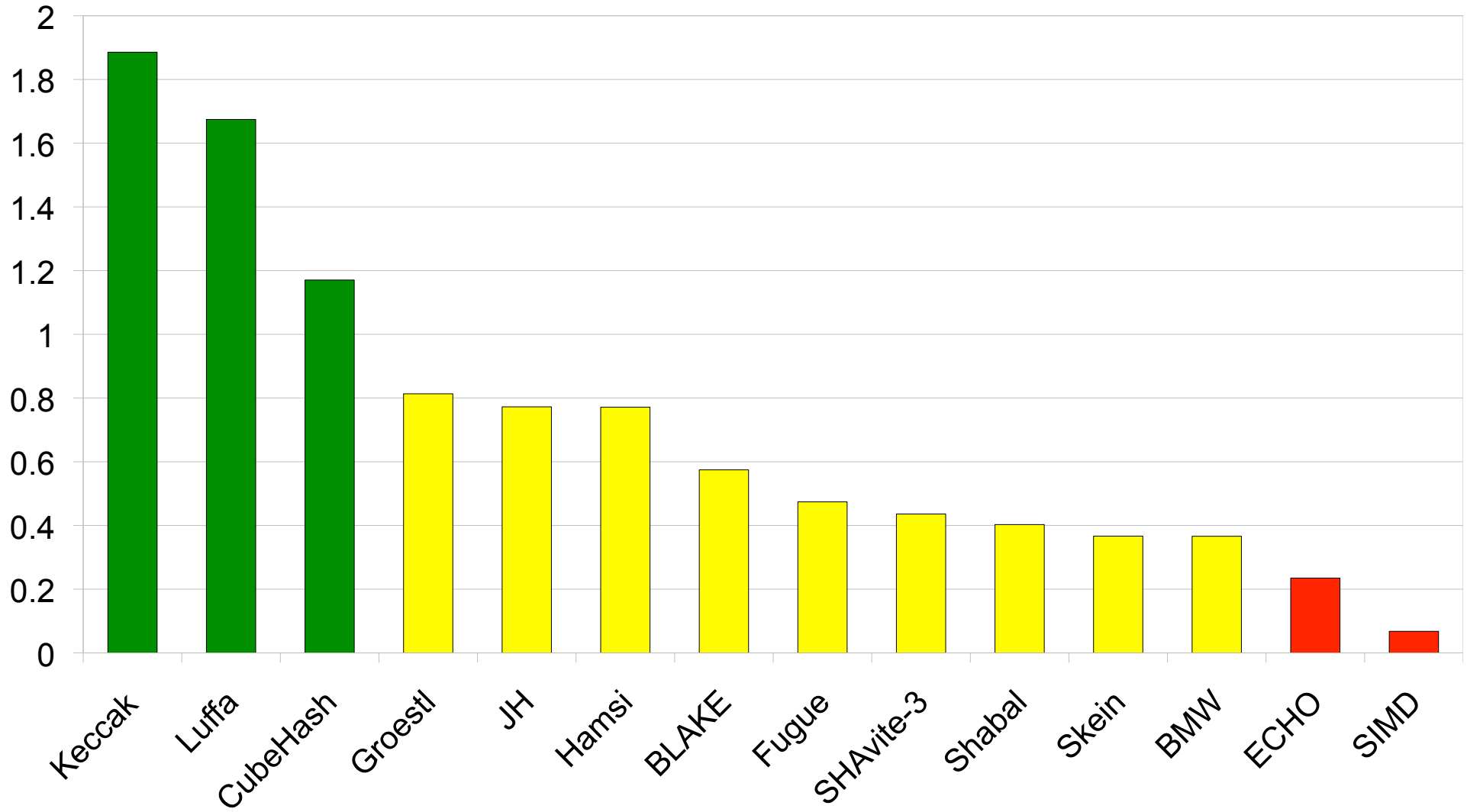
Overall Normalized Area: 256-bit variants of algorithms

Normalized to SHA-256, Averaged over 7 FPGA families

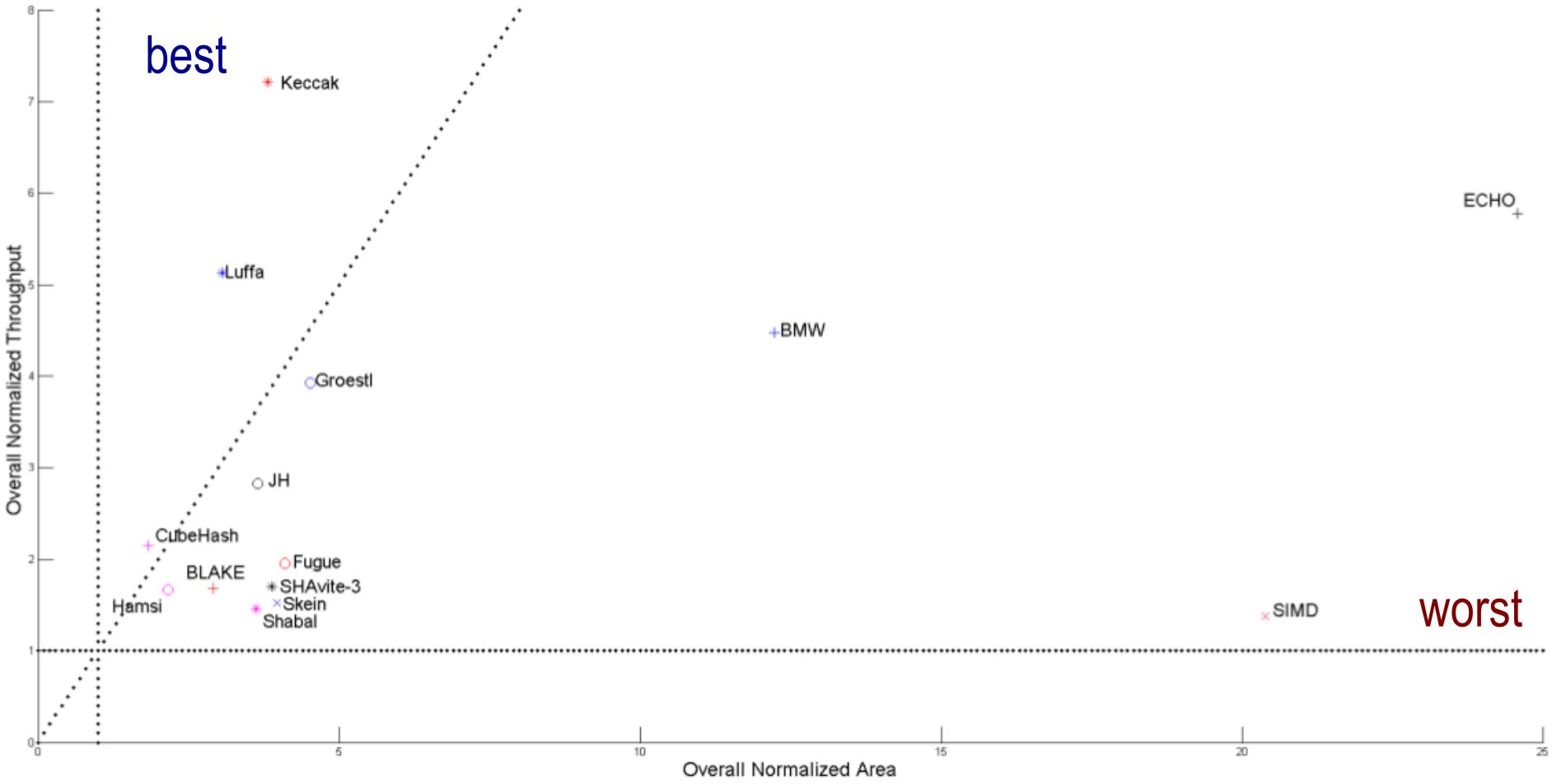


Overall Normalized Throughput/Area: 256-bit variants

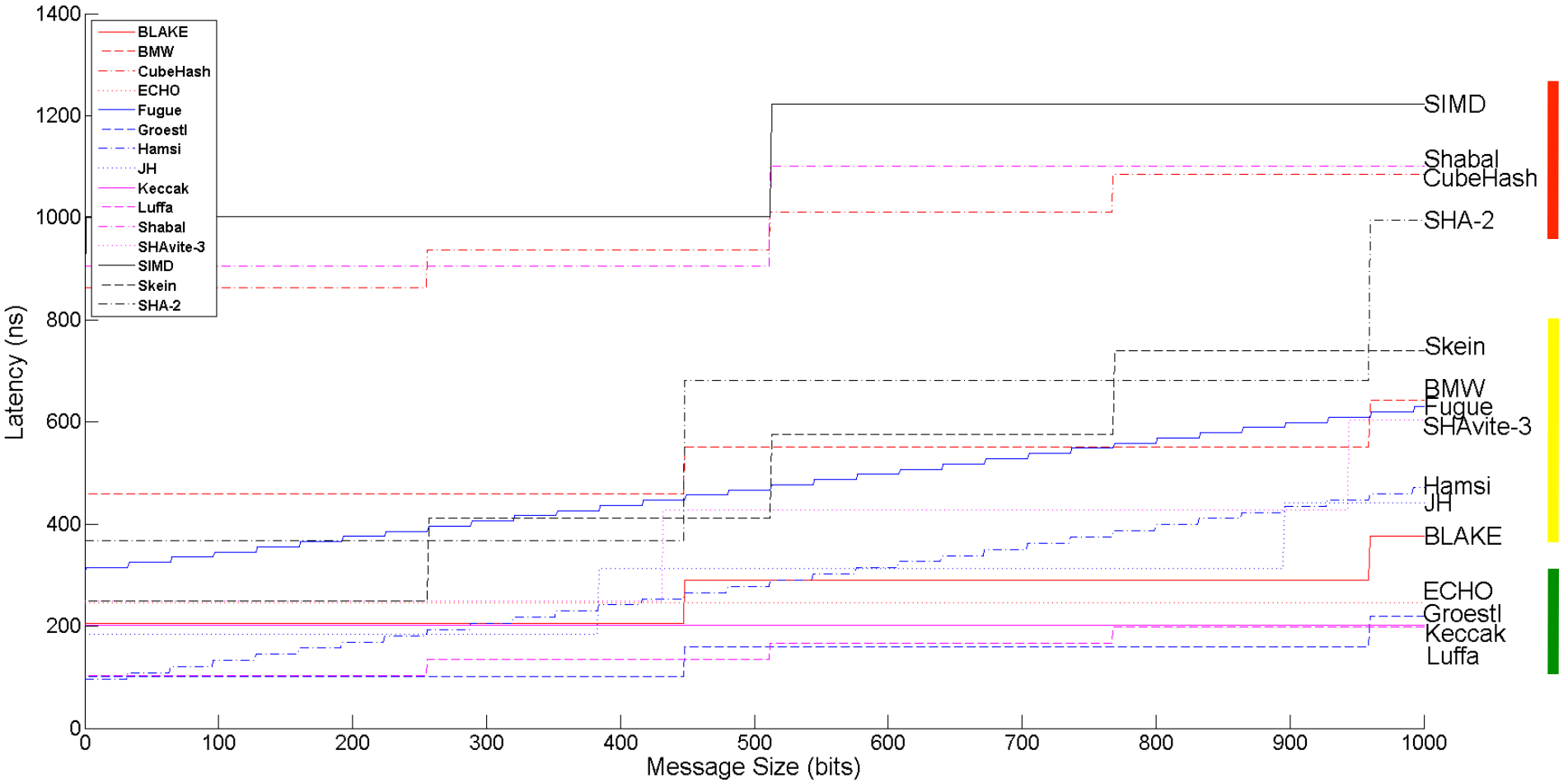
Normalized to SHA-256, Averaged over 7 FPGA families



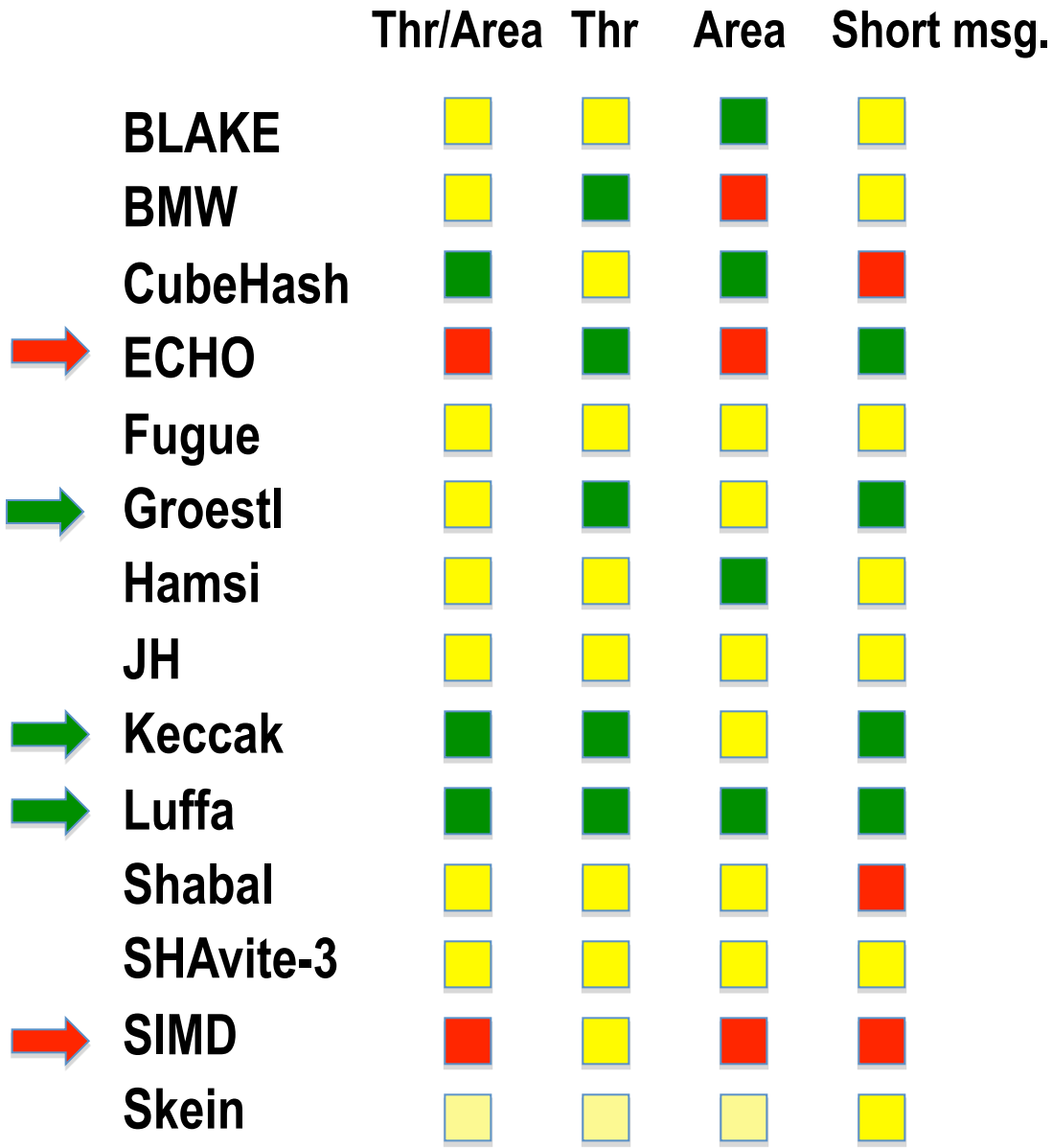
Throughput vs. Area Normalized to Results for SHA-256 and Averaged over 7 FPGA Families – 256-bit variants



Execution Time for Short Messages up to 1000 bits
Virtex 5, 256-bit variants of algorithms



Summary for SHA-3-256



Summary of Results

- Throughput/Area & Throughput most crucial for high-speed implementations
- Area cannot be easily traded for Throughput

Best performers so far

1-2. Keccak & Luffa

3. Groestl

Worst performers so far:

14. SIMD – throughput to area ratio

13. ECHO – area

More About our Designs & Tools

- SHA-3 papers
  - **Results for 512-bit variants**
  - ATHENa intro

FPL 2010 paper
- **ATHENa** features
- Case studies

- Cryptology e-Print Archive - **2010/445** (100+ pages)
  - **Detailed hierarchical block diagrams**
  - **Corresponding formulas for execution time and throughput**

- ATHENa web site
  - **Most recent results**
  - Comparisons with results from other groups
  - **Optimum options of tools**

Comparison with Other Groups

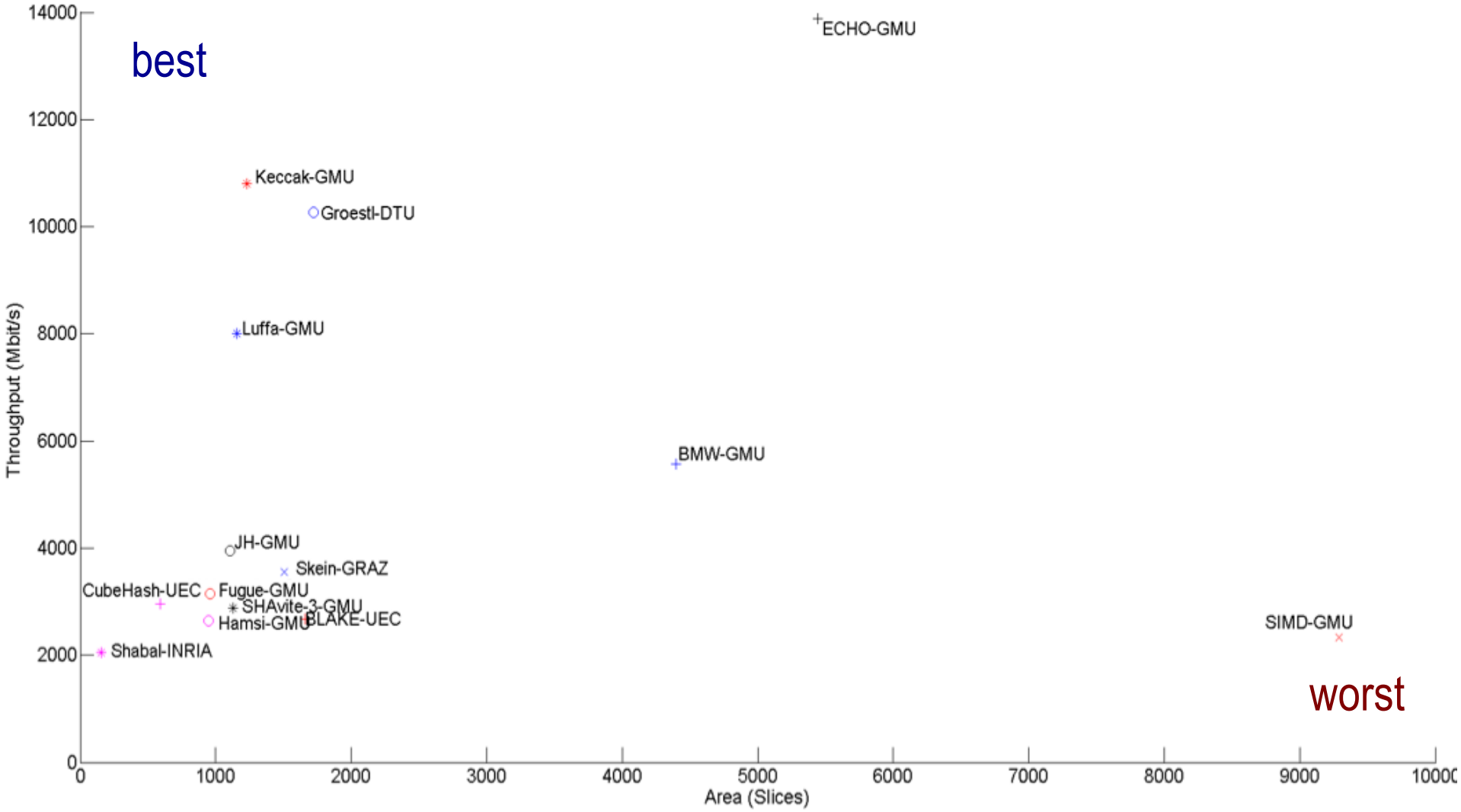
Comparison with Best Results Reported by Other Groups
Virtex 5, 256-bit variants of algorithms

	OTHER GROUPS				GMU		
	Area	**Thr**	**Thr/Area**	**Source**	**Area**	**Thr**	**Thr/Area**
BLAKE	1660	2676	**1.61**	Kobayashi et al.	1871	2854	1.53
CubeHash	590	2960	**5.02**	Kobayashi et al.	707	3445	4.87
ECHO	9333	14860	1.59	Lu et al.	5445	13875	**2.55**
Groestl	1722	10276	**5.97**	Gauvaram et al.	1884	8677	4.61
Hamsi	718	1680	2.34	Kobayashi et al.	946	2646	**2.80**
Keccak	1412	6900	4.89	Bertoni et al.	1229	10807	**8.79**
Luffa	1048	6343	6.05	Kobayashi et al.	1154	8008	**6.94**
Shabal	**153**	2051	**13.41**	Detrey et al.	1266	2624	2.07
Skein (estimated)	1632	3535	**2.17**	Tillich	1463	2812	1.92

Best Overall Reported Results as of Aug. 6, 2010
Virtex 5, 256-bit variants of algorithms

	BEST REPORTED RESULTS			
	Area	**Thr**	**Thr/Area**	**Source**
BLAKE	1660	2676	1.61	Kobayashi et al.
BMW	4400	5577	1.27	GMU
CubeHash	590	2960	5.02	Kobayashi et al.
ECHO	5445	13875	2.55	GMU
Fugue	956	3151	3.30	GMU
Groestl	1722	10276	5.97	Gauvaram et al.
Hamsi	946	2646	2.80	GMU
JH	1108	3955	3.57	GMU
Keccak	1229	10807	8.79	GMU
Luffa	1154	8008	6.94	GMU
Shabal	153	2051	13.41	Detrey et al.
SHAvite-3	1130	2887	2.55	GMU
SIMD	9288	2326	0.25	GMU
Skein	1632	3535	2.17	Tillich et al.

Throughput vs. Area: Best reported results
Virtex 5, 256-bit variants of algorithms



Invitation to Use ATHENa

http://cryptography.gmu.edu/athena

User

FPGA Synthesis and Implementation

6

5

Database query

Ranking of designs

2

HDL + scripts + configuration files

3

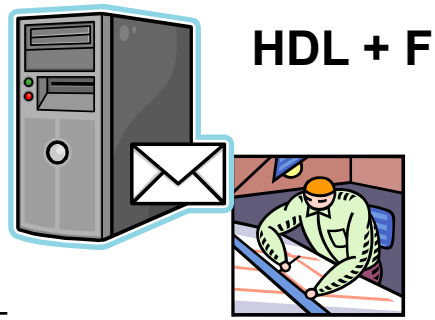
Result Summary + Database Entries

ATHENa Server

1

ATHENa scripts and configuration files

HDL + FPGA Tools



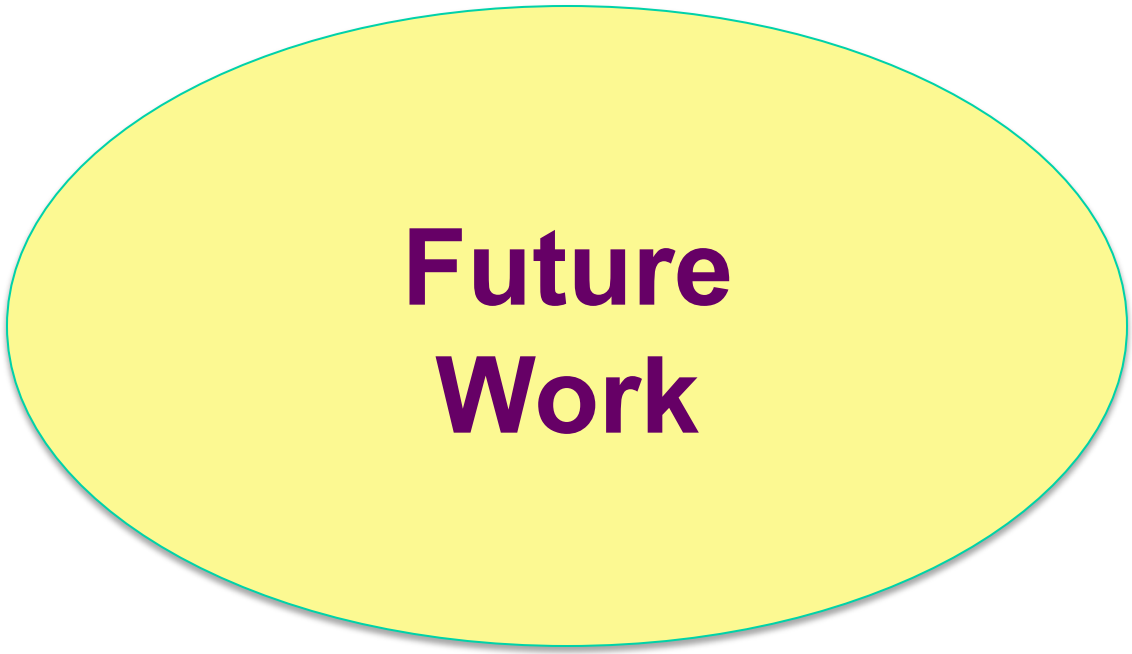
4

Database Entries

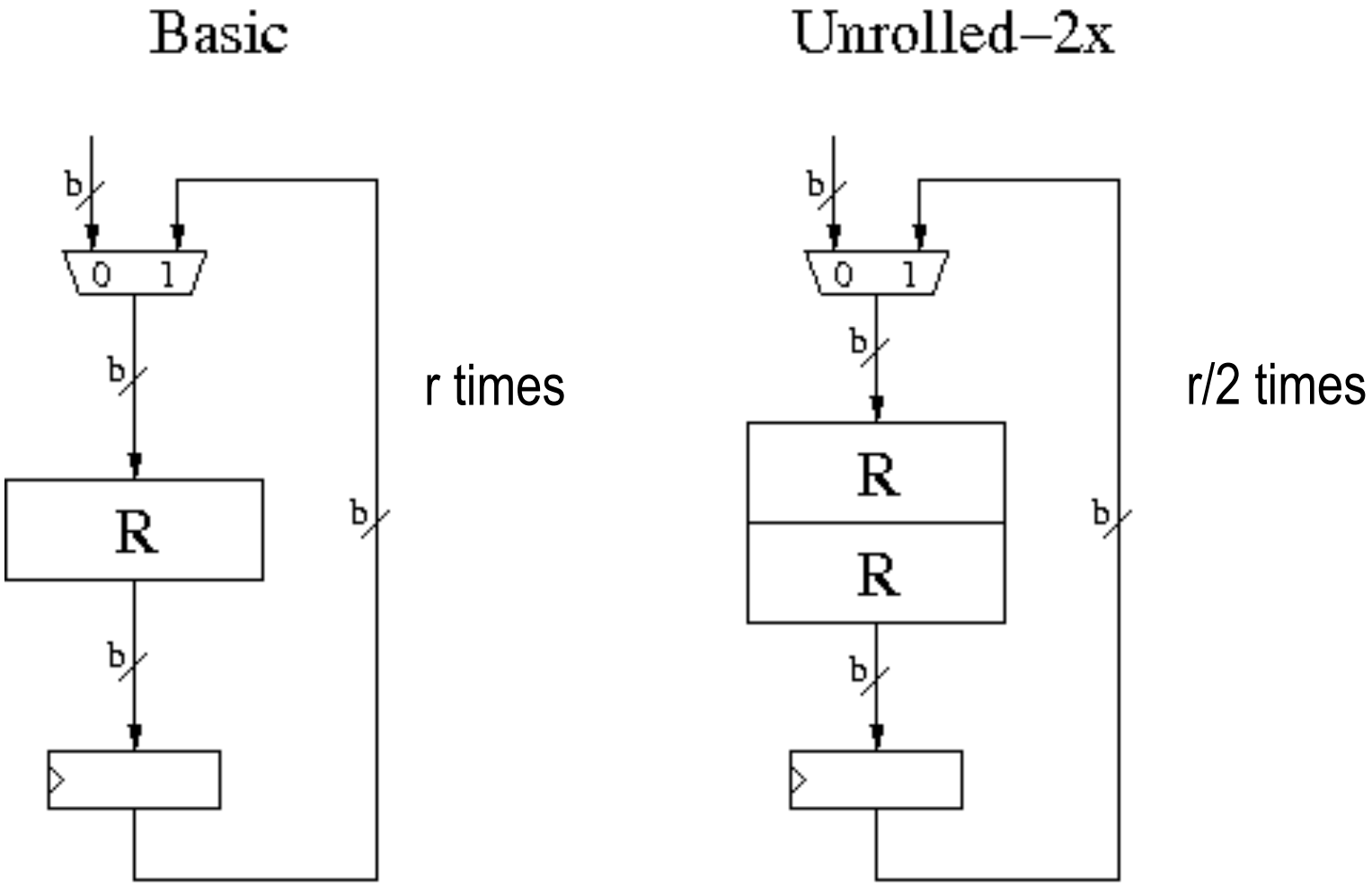
Designer

0

Interfaces + Testbenches

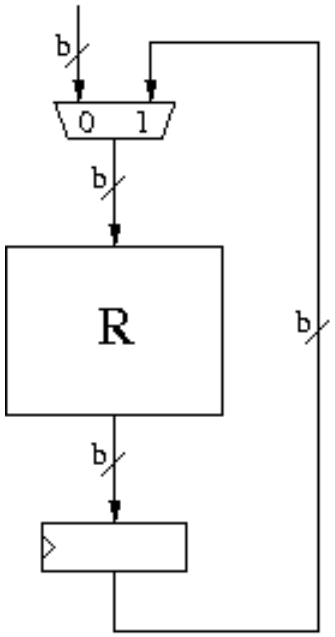

Future
Work

Analysis of Alternative Architectures - Unrolled



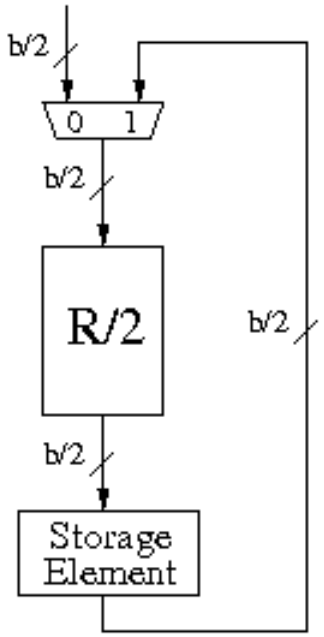
Analysis of Alternative Architectures - Folded

Basic



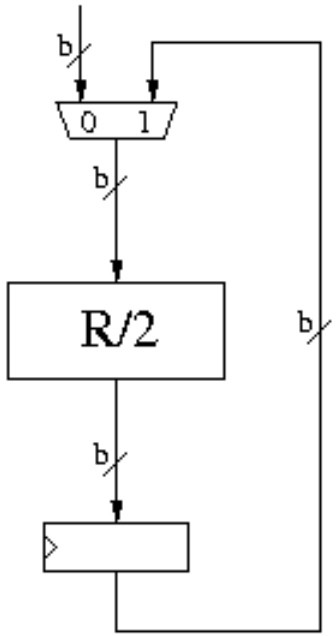
r times

Folded Horizontally–2x (fh2)



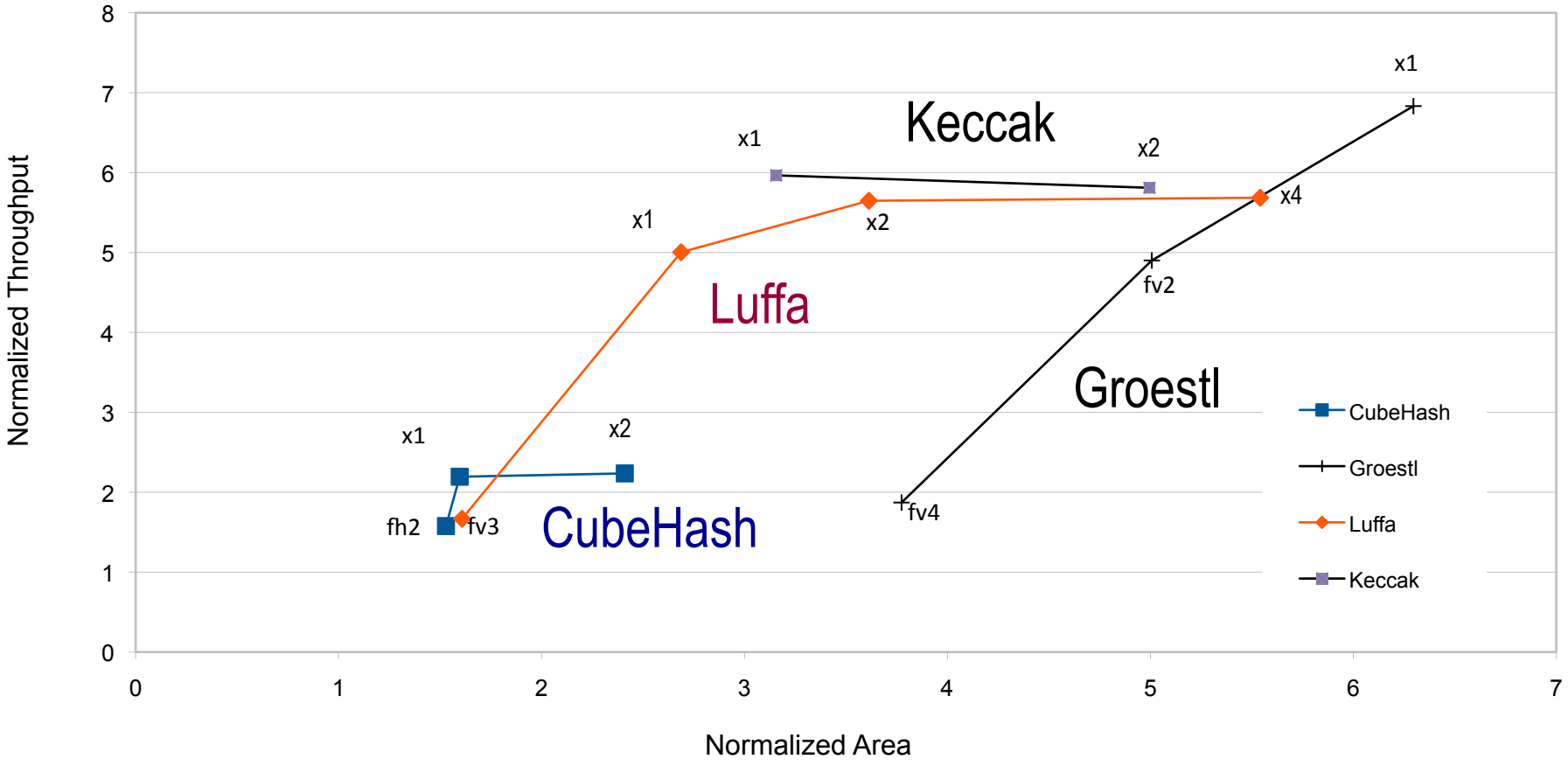
2·r times

Folded Vertically–2x (fv2)



2·r times

Analysis of Alternative Architectures
CubeHash, Groestl, Keccak, Luffa in Virtex 5



Conclusions

Conclusions

- **Fair and comprehensive methodology** for evaluation of hardware performance in FPGAs developed and applied to the evaluation of 14 Round 2 SHA-3 candidates

- **Large differences among competing algorithms** in terms of performance in FPGAs

- Three **front-runners**: **Keccak, Luffa, Groestl**
  Two candidates **trailing behind**: **SIMD, ECHO**

Thank you!

Questions?

Questions?

CERG: **http:/cryptography.gmu.edu**

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