

Battles of Cryptographic Algorithms: From AES to CAESAR in Software & Hardware



Kris Gaj
George Mason University

Collaborators

Joint 3-year project (2010-2013) on benchmarking cryptographic algorithms in software and hardware sponsored by



software



**Daniel J. Bernstein,
University of Illinois
at Chicago**

FPGAs



**Jens-Peter Kaps
George Mason
University**

FPGAs/ASICs



**Patrick
Schaumont
Virginia Tech**

ASICs



**Leyla
Nazhand-Ali
Virginia Tech**



Outline

- Introduction & motivation
- Cryptographic standard contests
 - AES
 - eSTREAM
 - SHA-3
 - CAESAR
- Progress in evaluation methods
- Benchmarking tools
- Open problems



Cryptography is everywhere

We trust it because of standards



Buying a book on-line



Withdrawing cash from ATM



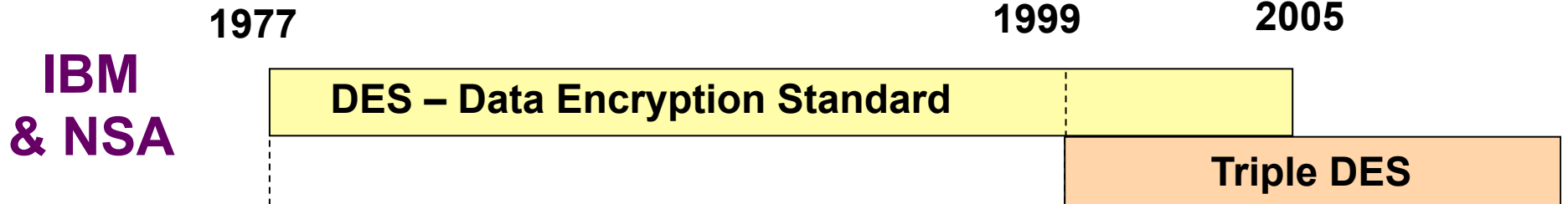
**Teleconferencing
over Intranets**



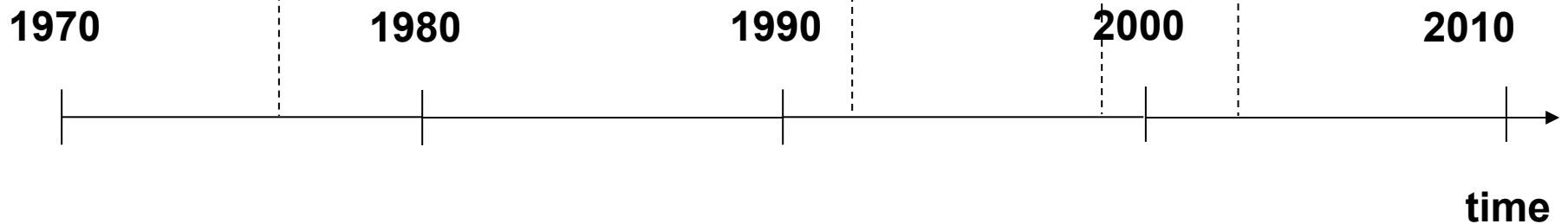
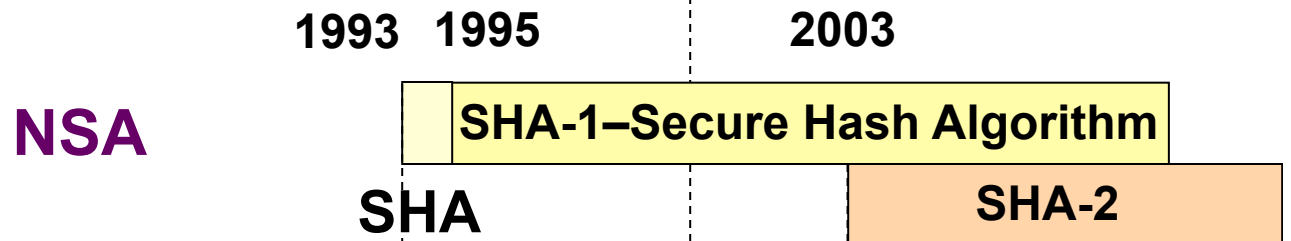
**Backing up files
on remote server**

Cryptographic Standards Before 1997

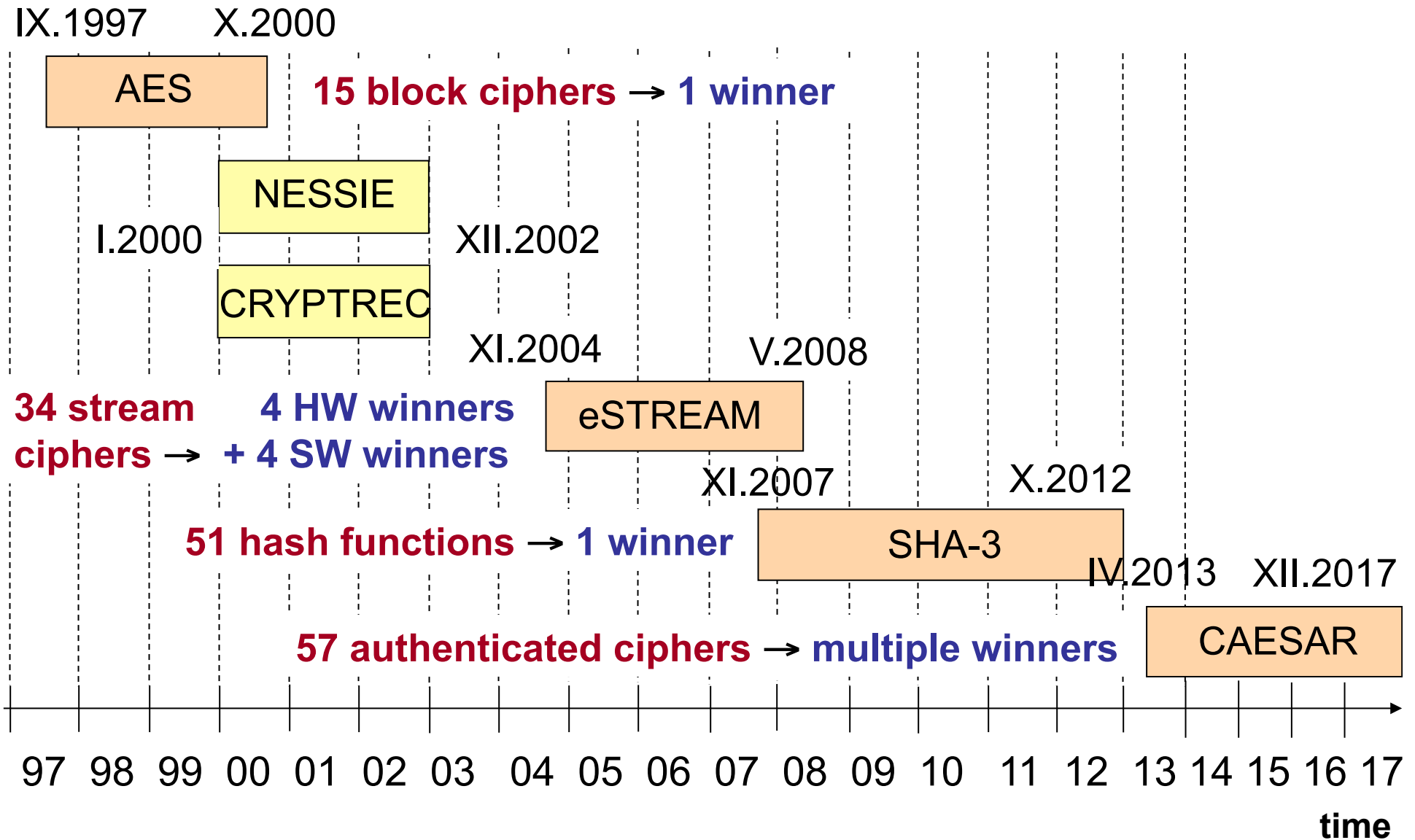
Secret-Key Block Ciphers



Hash Functions



Cryptographic Standard Contests



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



Cryptographic Contests - Evaluation Criteria

Security

Software Efficiency

μProcessors μControllers

Hardware Efficiency

FPGAs ASICs

Flexibility

Simplicity

Licensing

Specific Challenges of Evaluations in Cryptographic Contests

- Very wide range of possible applications, and as a result performance and cost targets

speed: tens of Mbits/s to hundreds Gbits/s

cost: single cents to thousands of dollars

- Winner in use for the next 20-30 years, implemented using technologies not in existence today
- Large number of candidates
- Limited time for evaluation
- The results are final



Mitigating Circumstances

- Performance of competing algorithms tend to vary significantly (sometimes as much as 500 times)
- Only relatively large differences in performance matter (typically at least 20%)
- Multiple groups independently implement the same algorithms (catching mistakes, comparing best results, etc.)
- Second best may be good enough



**AES
Contest
1997-2000**

Rules of the Contest

Each team submits

Detailed
cipher
specification

Justification
of design
decisions

Tentative
results
of cryptanalysis

Source
code
in C

Source
code
in Java

Test
vectors

AES: Candidate Algorithms



Canada:

CAST-256
Deal

USA:

Mars
RC6
Twofish
Safer+
HPC

Costa Rica:

Frog



Germany:

Magenta

Belgium:

Rijndael

France:

DFC

Israel, UK, Norway:

Serpent



Korea:

Crypton

Japan:

E2



Australia:

LOKI97

AES Contest Timeline

June 1998

15 Candidates

CAST-256, Crypton, Deal, DFC, E2,
Frog, HPC, LOKI97, Magenta, Mars,
RC6, Rijndael, Safer+, Serpent, Twofish,

Round 1

Security
Software efficiency

August 1999

5 final candidates

Mars, RC6, Twofish (USA)
Rijndael, Serpent (Europe)

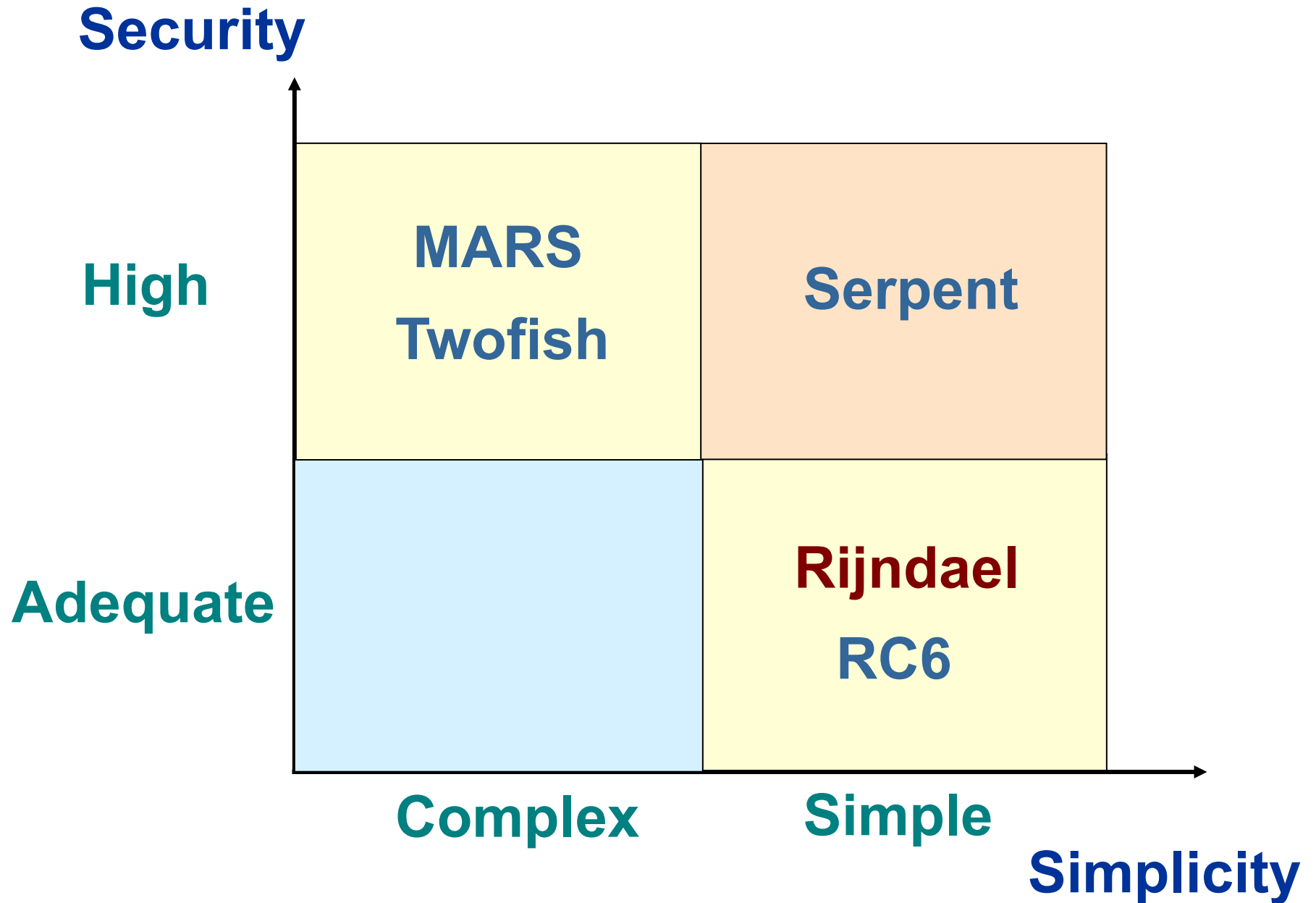
Round 2

Security
Software efficiency
Hardware efficiency

October 2000

1 winner: Rijndael
Belgium

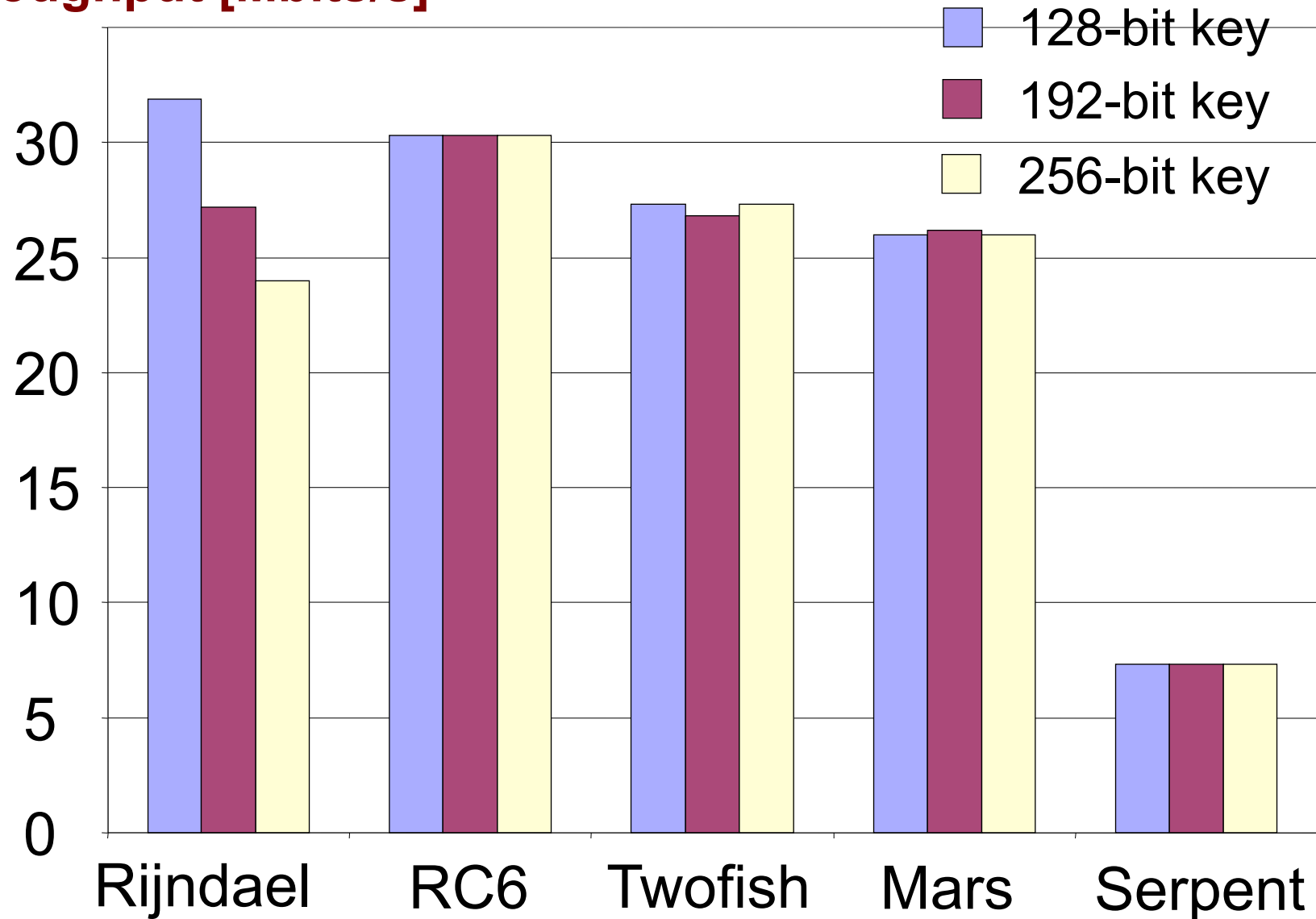
NIST Report: Security & Simplicity



Efficiency in software: NIST-specified platform

200 MHz Pentium Pro, Borland C++

Throughput [Mbits/s]



NIST Report: Software Efficiency

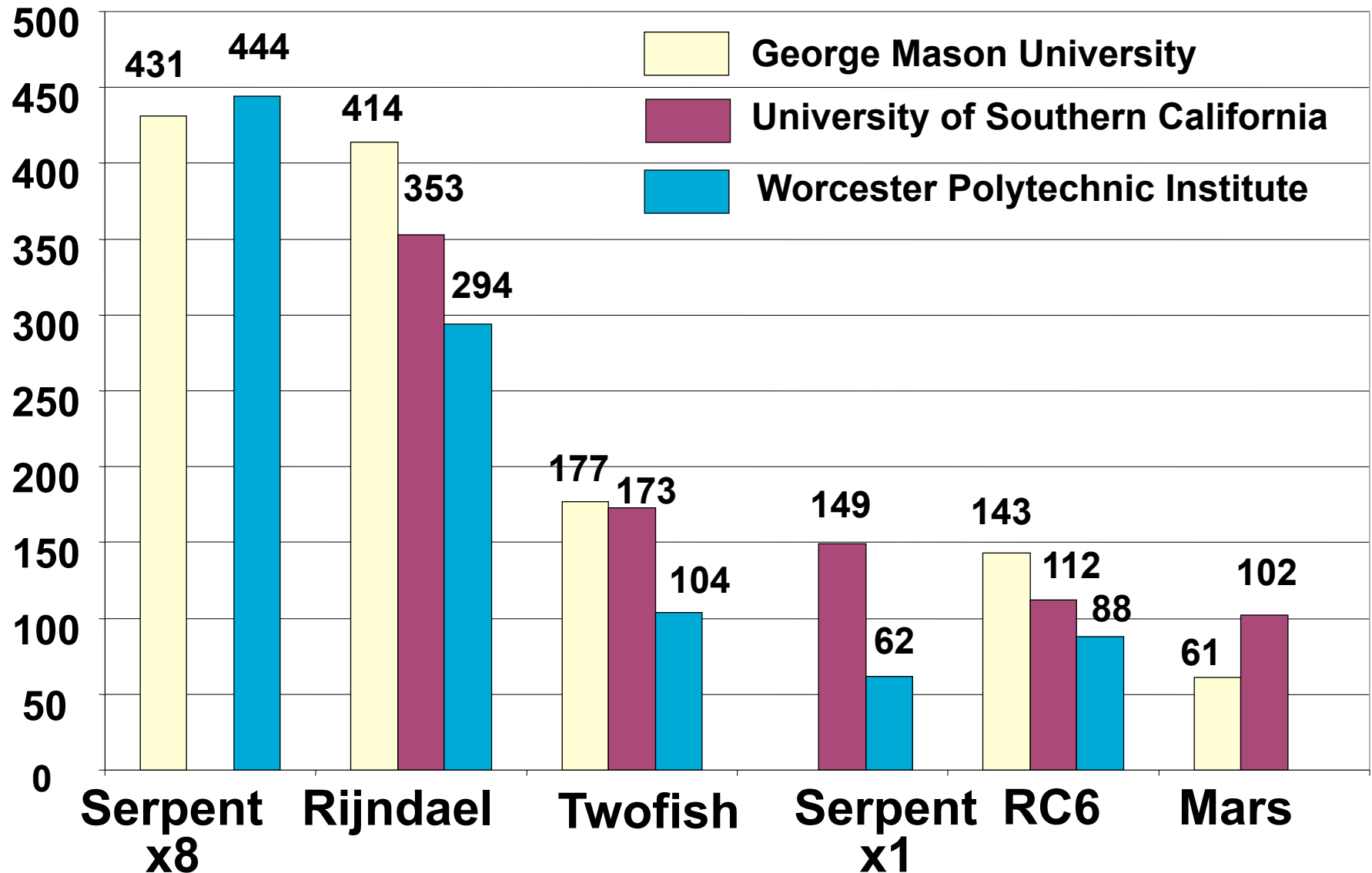
Encryption and Decryption Speed

	32-bit processors	64-bit processors	DSPs
high	RC6	Rijndael Twofish	Rijndael Twofish
medium	Rijndael Mars Twofish	Mars RC6	Mars RC6
low	Serpent	Serpent	Serpent

Efficiency in FPGAs: Speed

Xilinx Virtex XCV-1000

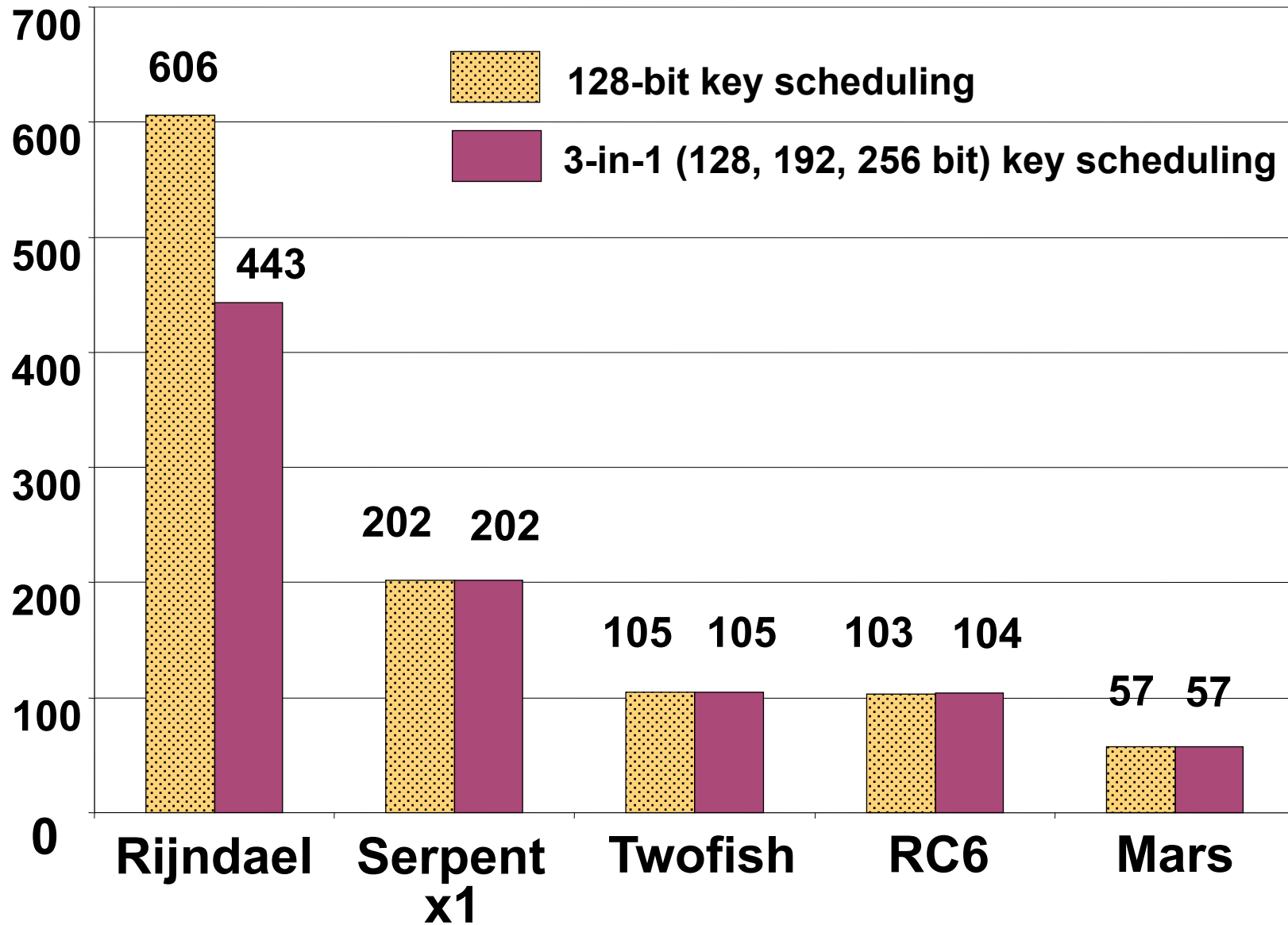
Throughput [Mbit/s]



Efficiency in ASICs: Speed

MOSIS 0.5 μ m, NSA Group

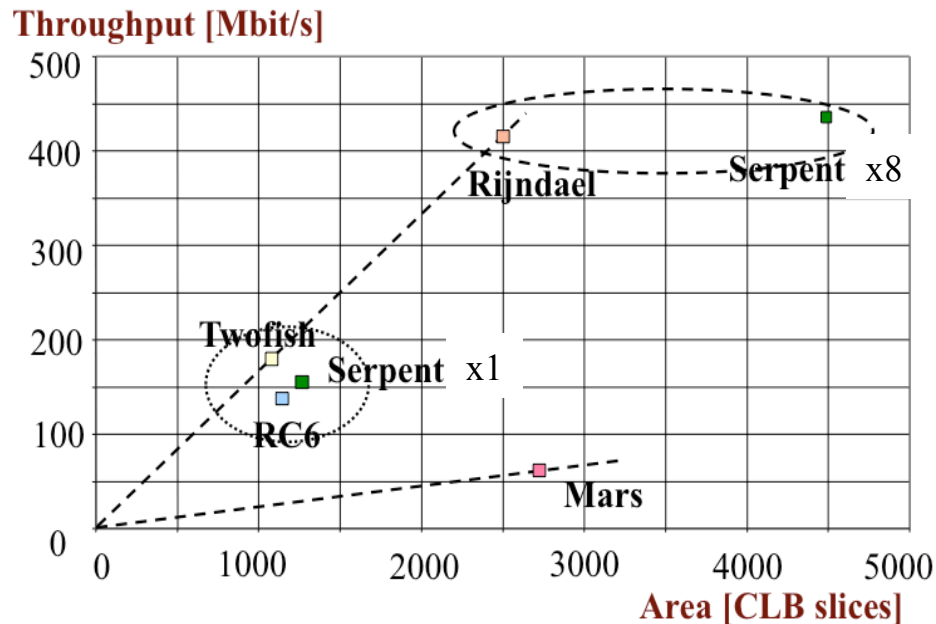
Throughput [Mbit/s]



Lessons Learned

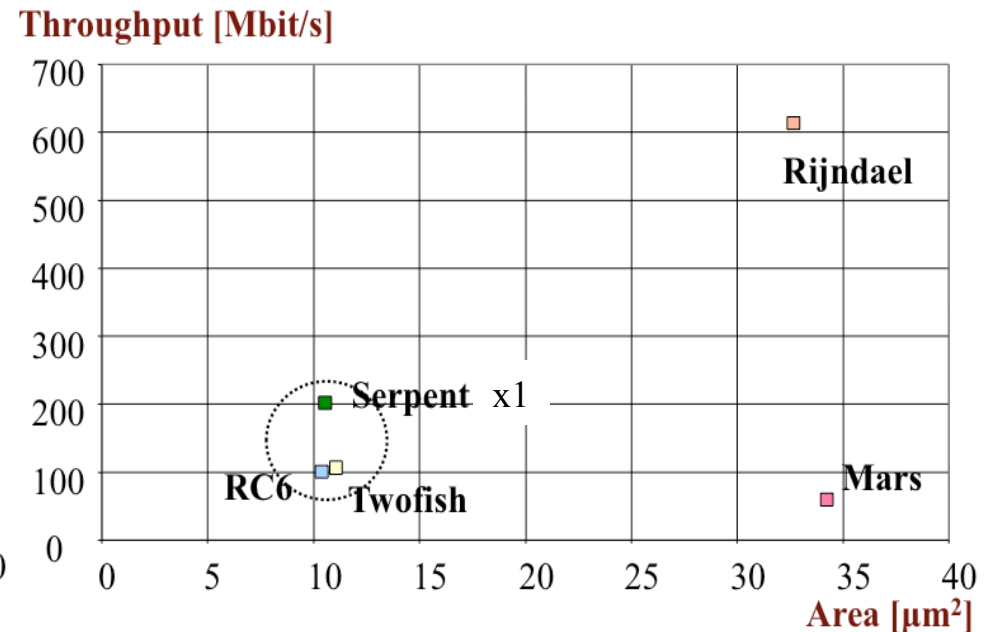
Results for ASICs matched very well results for FPGAs, and were both very different than software

FPGA



GMU+USC, Xilinx Virtex XCV-1000

ASIC



NSA Team, ASIC, 0.5 μm MOSIS

Serpent fastest in hardware, slowest in software

Lessons Learned

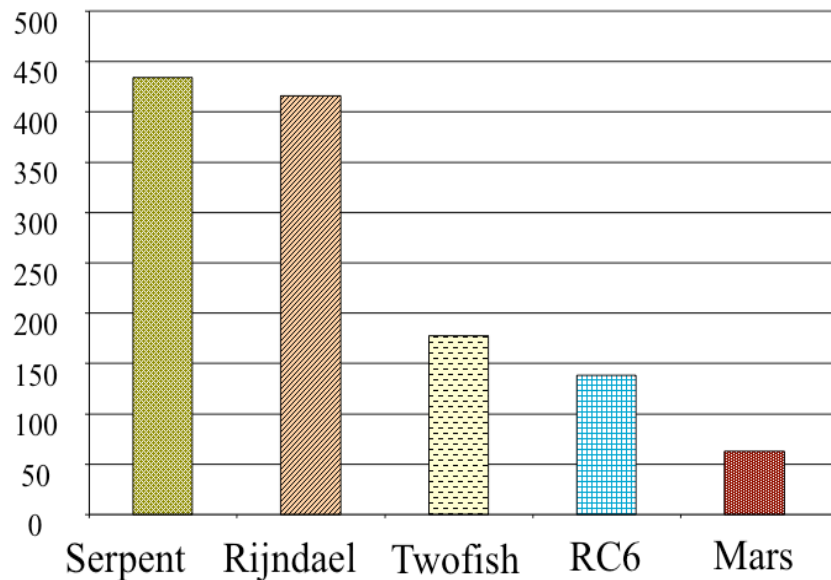
Hardware results matter!

Final round of the AES Contest, 2000

Speed in FPGAs

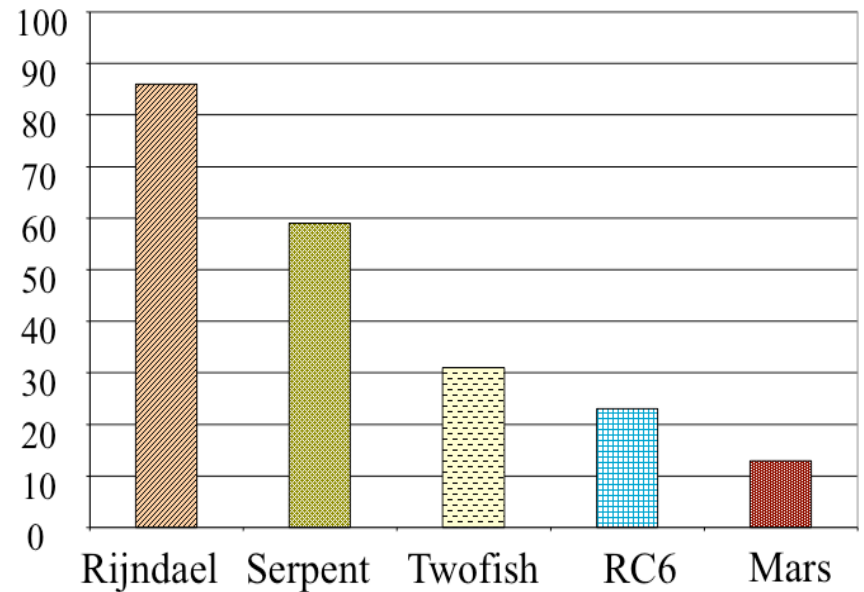
GMU results

Speed [Mbit/s]



Votes at the AES 3 conference

votes





**eSTREAM
Contest
2004-2008**



eSTREAM - Contest for a new stream cipher standard

PROFILE 1 (SW)

- Stream cipher suitable for **software implementations** optimized for **high speed**
- Key size - 128 bits
- Initialization vector – 64 bits or 128 bits

PROFILE 2 (HW)

- Stream cipher suitable for **hardware implementations** with **limited memory, number of gates, or power supply**
- Key size - 80 bits
- Initialization vector – 32 bits or 64 bits

eSTREAM Contest Timeline

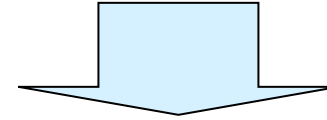
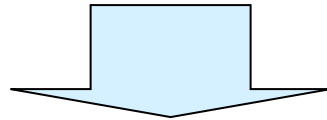
April 2005

PROFILE 1 (SW)

PROFILE 2 (HW)

23 Phase 1 Candidates

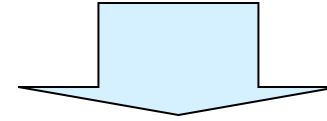
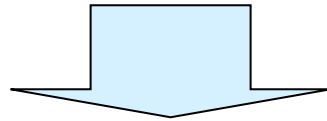
25 Phase 1 Candidates



July 2006

13 Phase 2 Candidates

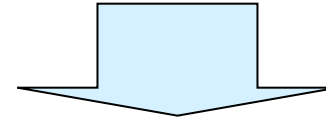
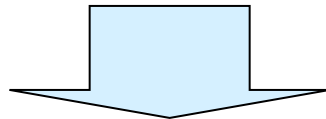
20 Phase 2 Candidates



April 2007

8 Phase 3 Candidates

8 Phase 3 Candidates



May 2008

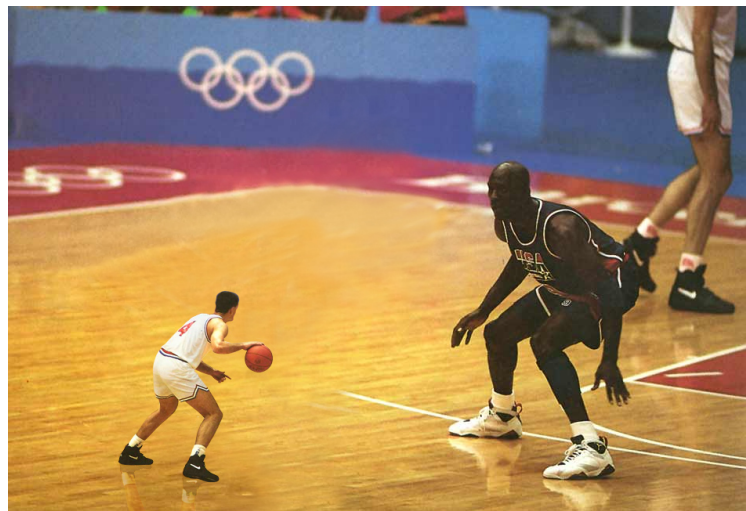
**4 winners:
HC-128, Rabbit,
Salsa20, SOSEMANUK**

**4 winners:
Grain v1, Mickey v2,
Trivium, ~~F-FCSR-H v2~~**

Lessons Learned

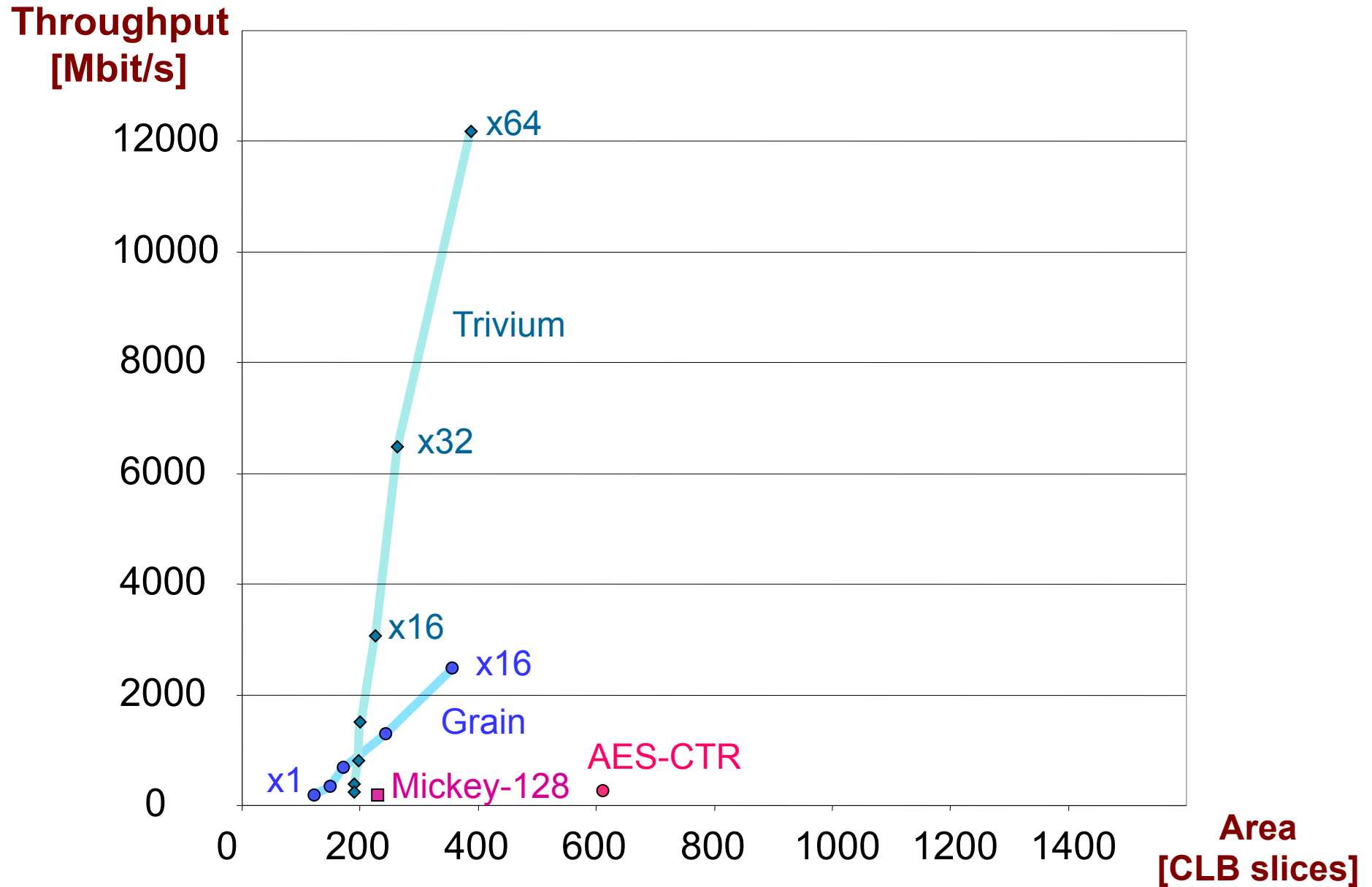
**Very large differences among
8 leading candidates**

- ~30 x** in terms of area (Grain v1 vs. Edon80)
- ~500 x** in terms of the throughput to area ratio (Trivium (x64) vs. Pomaranch)



Hardware Efficiency in FPGAs

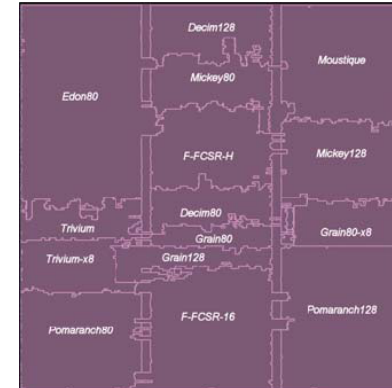
Xilinx Spartan 3, GMU SASC 2007



eSTREAM ASIC Evaluations

New compared to AES:

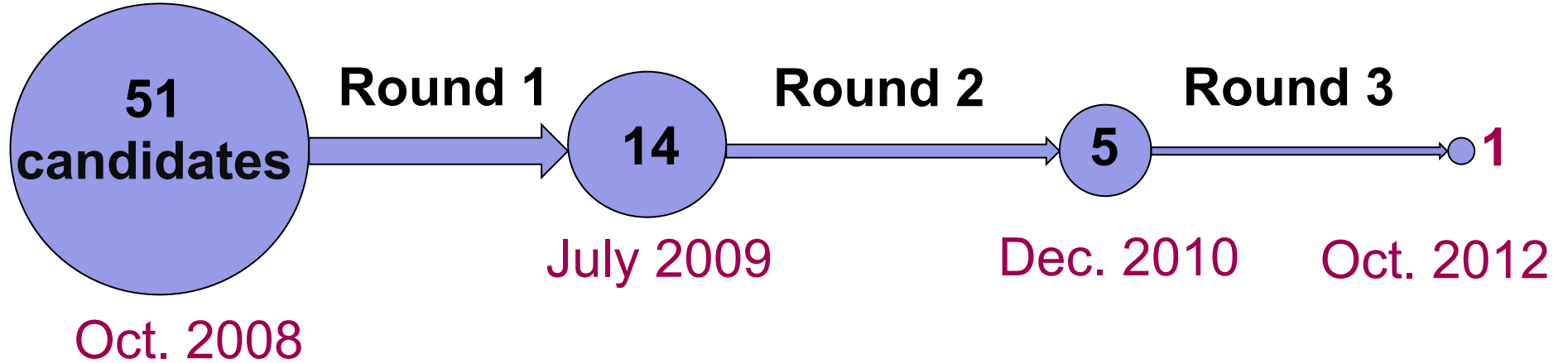
- **Post-layout** results, followed by
- Actually **fabricated ASIC chips** (0.18 μ m CMOS)
- **Two** representative **applications**
 - **WLAN @ 10 Mbits/s**
 - **RFID / WSN @ 100 kHz clock**
- More complex performance measures
 - **Power x Area x Time**





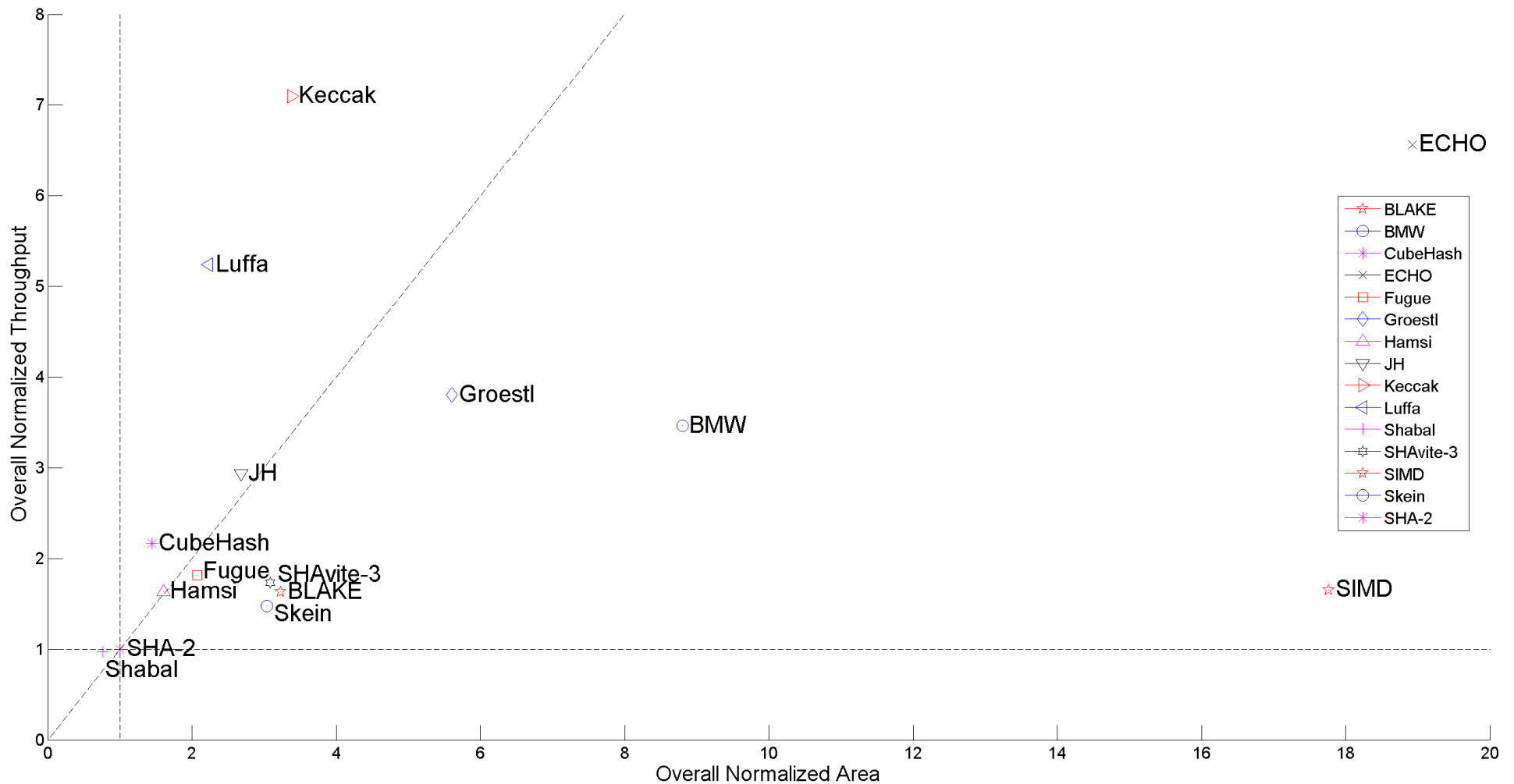
**SHA-3
Contest
2007-2012**

NIST SHA-3 Contest - Timeline

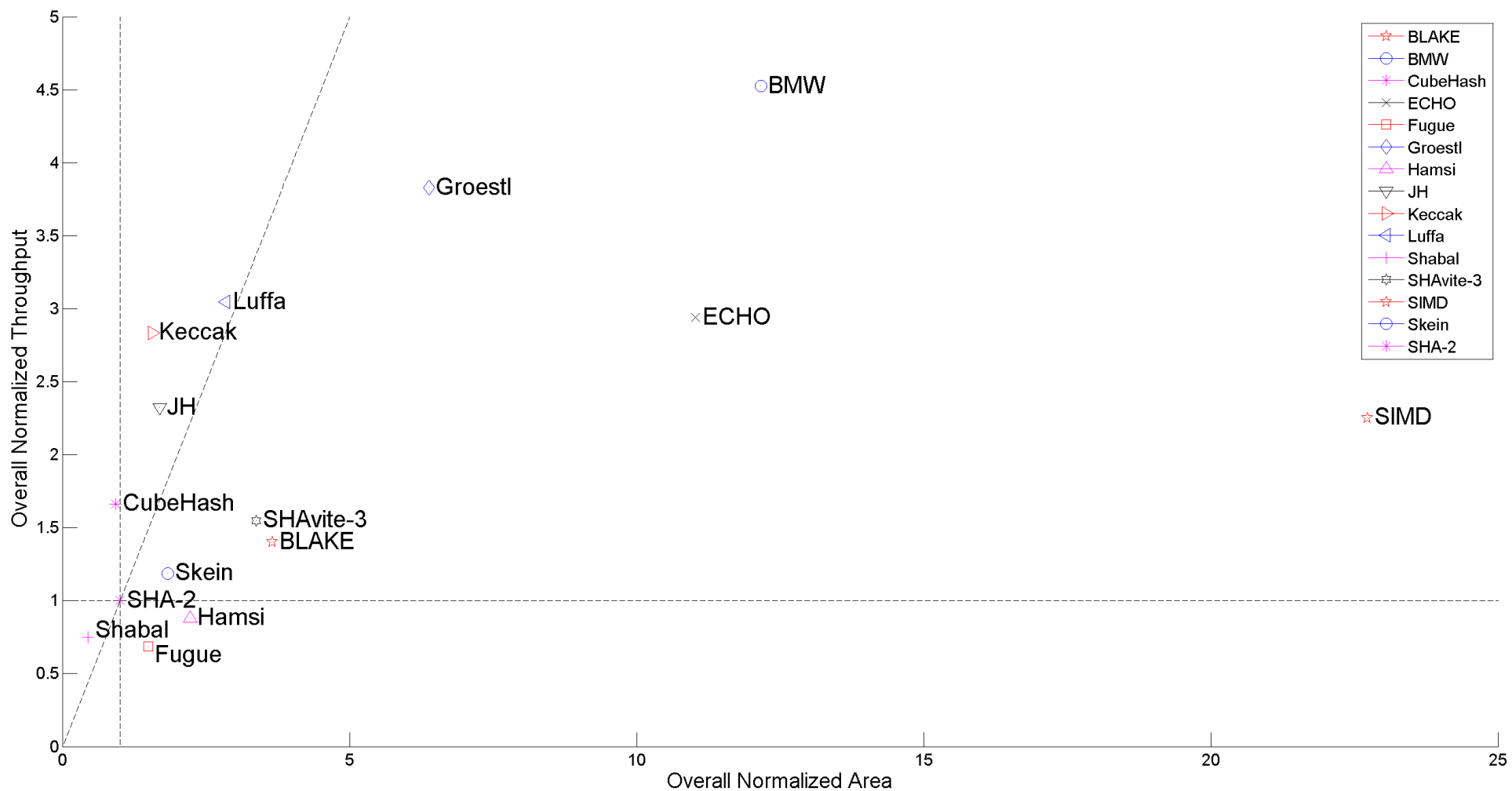


SHA-3 Round 2

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



Throughput vs. Area Normalized to Results for SHA-512 and Averaged over 11 FPGA Families – 512-bit variants



Performance Metrics

Primary

1. Throughput

3. Throughput / Area

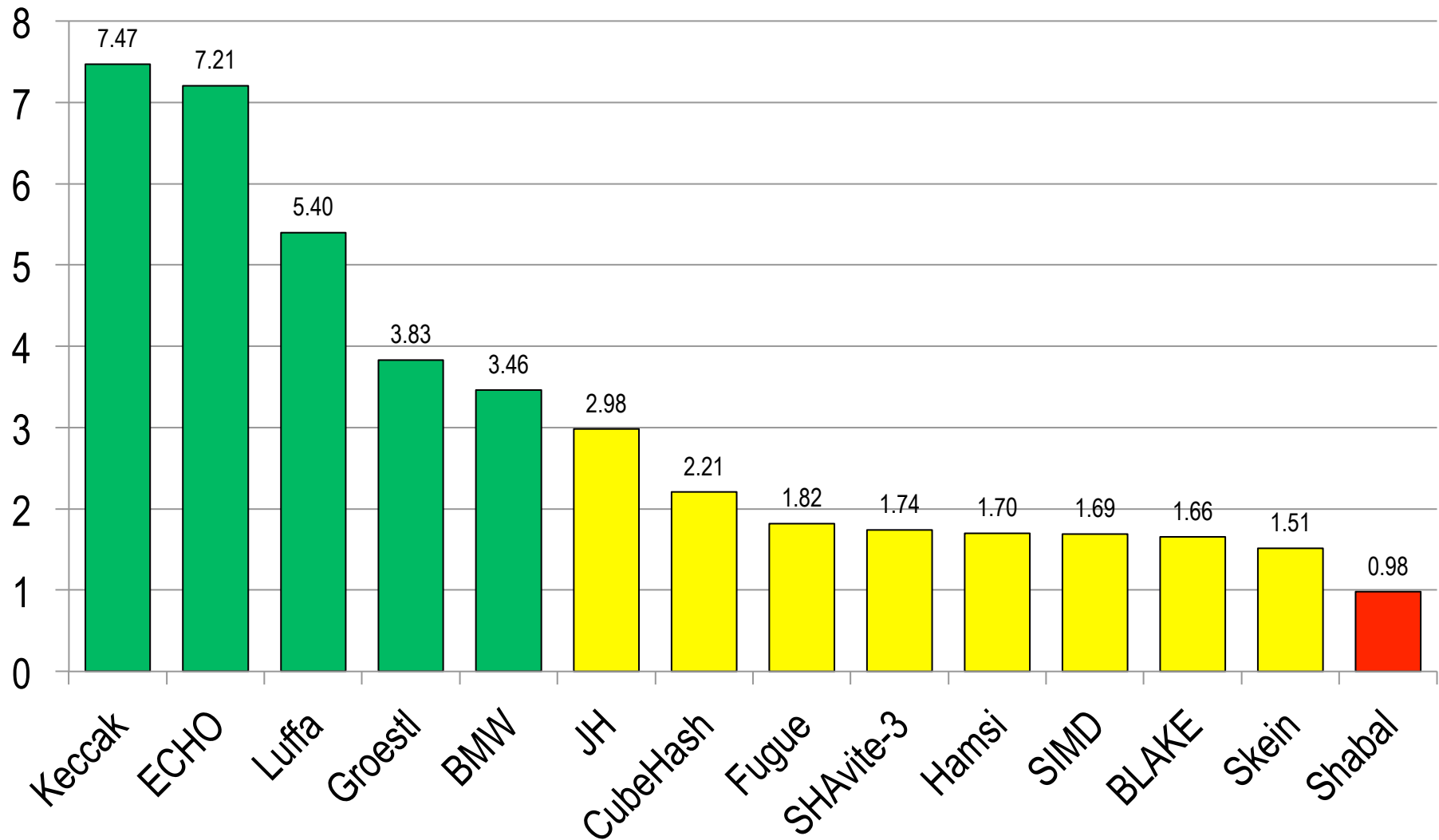
Secondary

2. Area

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








Overall Normalized Throughput: 256-bit variants of algorithms

Normalized to SHA-256, Averaged over 10 FPGA families



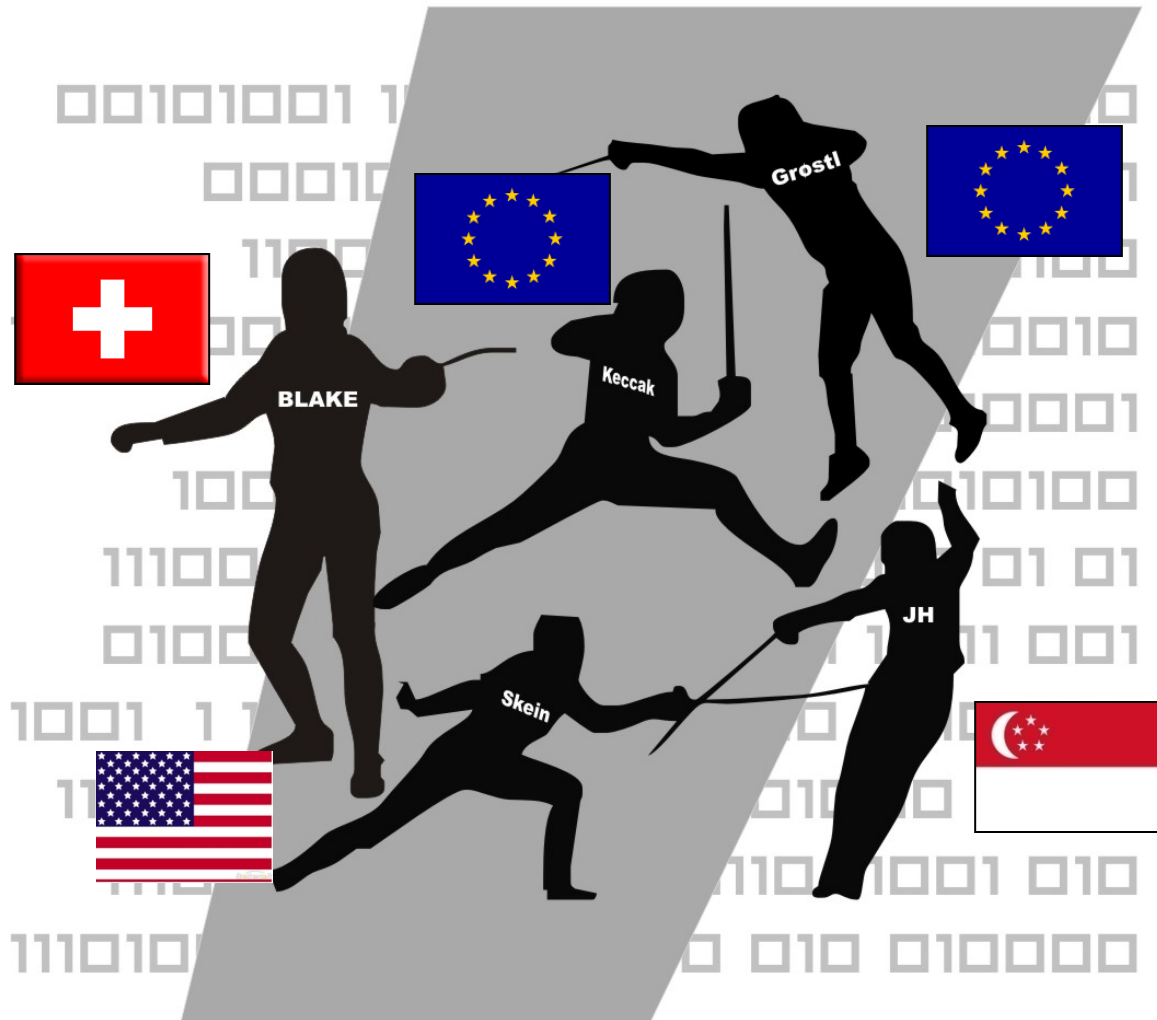
256-bit variants

512-bit variants

	Thr/Area	Thr	Area	Short msg.	Thr/Area	Thr	Area	Short msg.
BLAKE	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Red
 BMW	Yellow	Green	Yellow	Yellow	Yellow	Green	Red	Yellow
CubeHash	Green	Yellow	Green	Red	Green	Yellow	Green	Red
 ECHO	Yellow	Green	Red	Green	Yellow	Green	Red	Green
Fugue	Yellow	Yellow	Yellow	Yellow	Yellow	Red	Green	Yellow
 Groestl	Yellow	Green	Yellow	Yellow	Yellow	Green	Yellow	Green
Hamsi	Green	Yellow	Green	Yellow	Yellow	Red	Yellow	Yellow
 JH	Green	Yellow	Yellow	Yellow	Green	Yellow	Green	Green
 Keccak	Green	Green	Yellow	Green	Green	Green	Green	Green
 Luffa	Green	Green	Yellow	Green	Green	Green	Yellow	Green
Shabal	Green	Red	Green	Red	Green	Red	Green	Red
SHAvite-3	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
 SIMD	Red	Yellow	Red	Red	Red	Yellow	Red	Red
Skein	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow

SHA-3 Round 3

SHA-3 Contest Finalists



New in Round 3

- **Multiple Hardware Architectures**
- **Effect of the Use of Embedded Resources
(Block RAMs, DSP units)**
- **Low-Area Implementations**

Study of Multiple Architectures

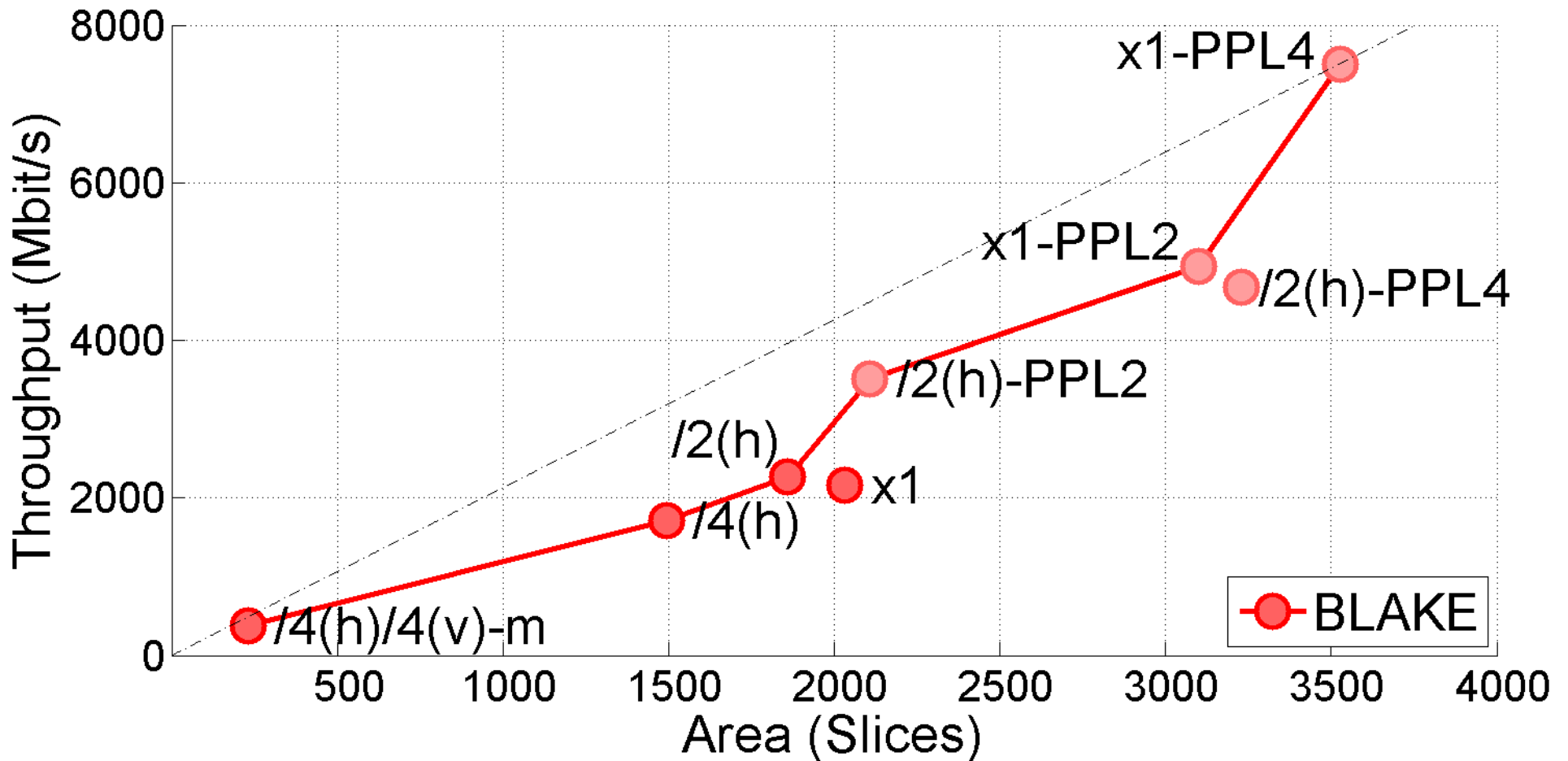
- Analysis of **multiple hardware architectures** per each finalist, based on the known design techniques, such as
 - **Folding**
 - **Unrolling**
 - **Pipelining**
- Identifying the **best architecture** in terms of the throughput to area ratio
- Analyzing the **flexibility** of all algorithms in terms of the speed vs. area trade-offs

Benchmarking of the SHA-3 Finalists by CERG GMU

- 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

BLAKE-256 in Virtex 5



x1 – basic iterative architecture

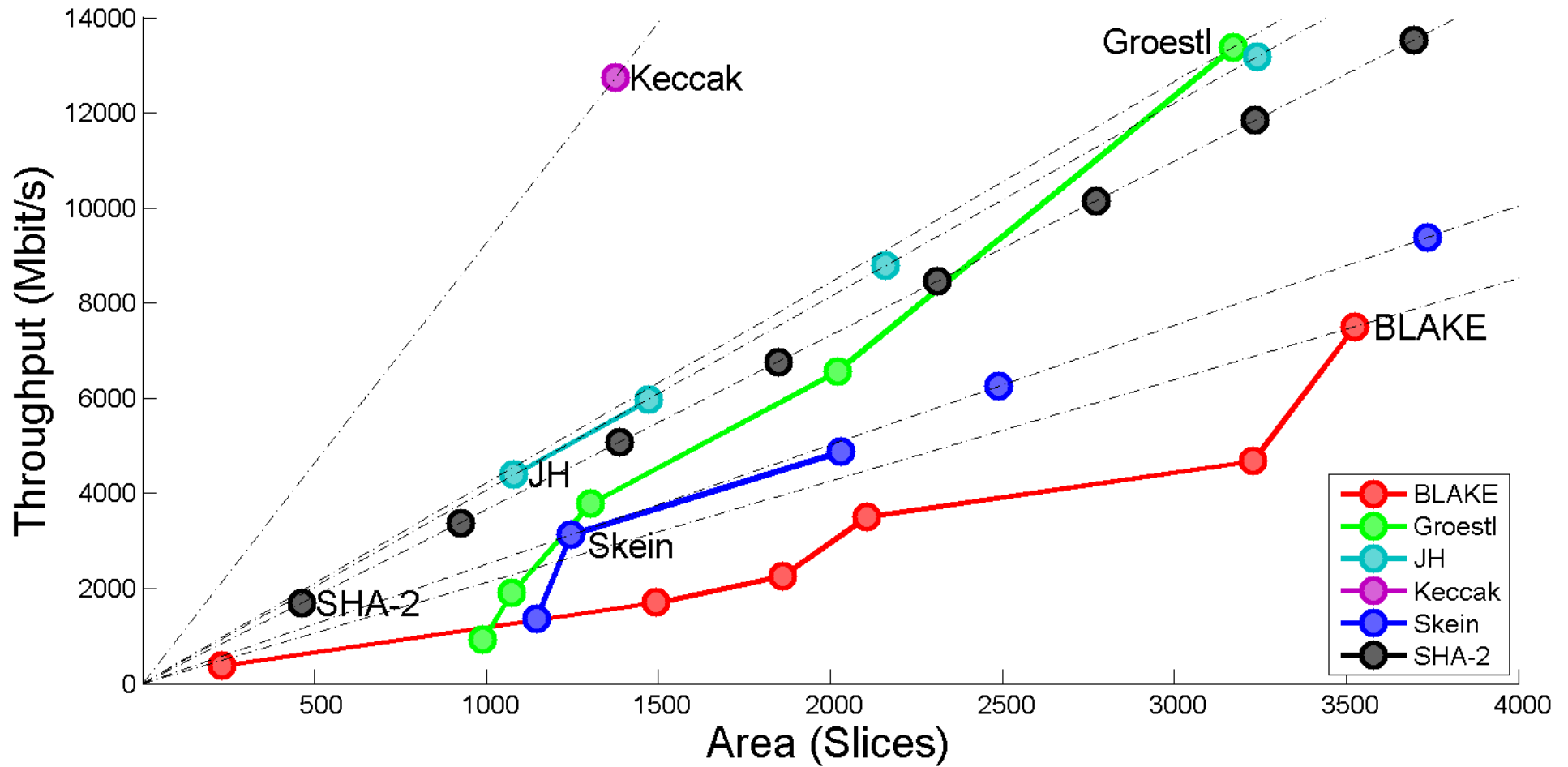
xk – unrolling by a factor of k

xk-PPLn – unrolling by a factor of k with n pipeline stages

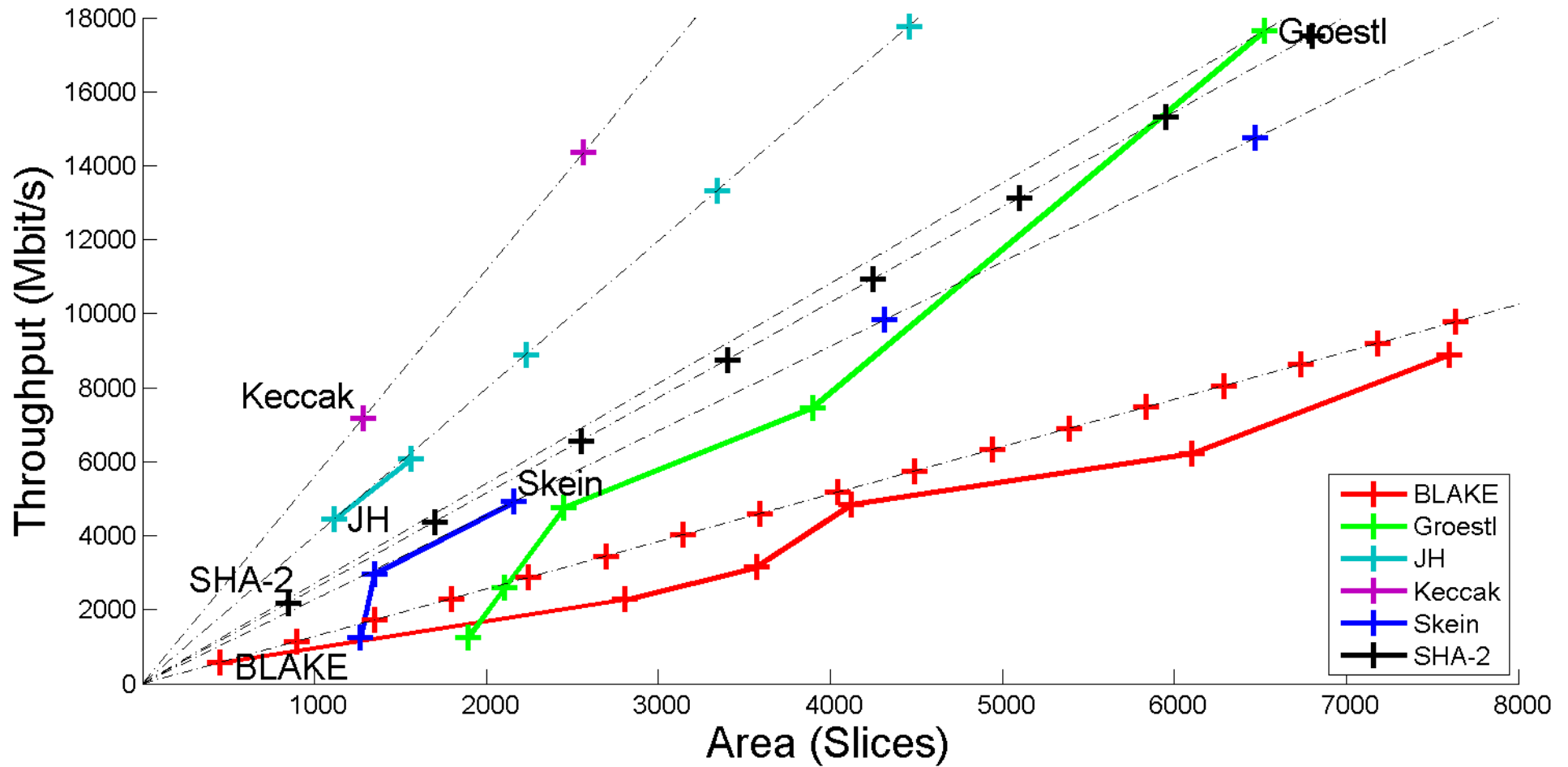
/k(h) – horizontal folding by a factor of k

/k(v) – vertical folding by a factor of k

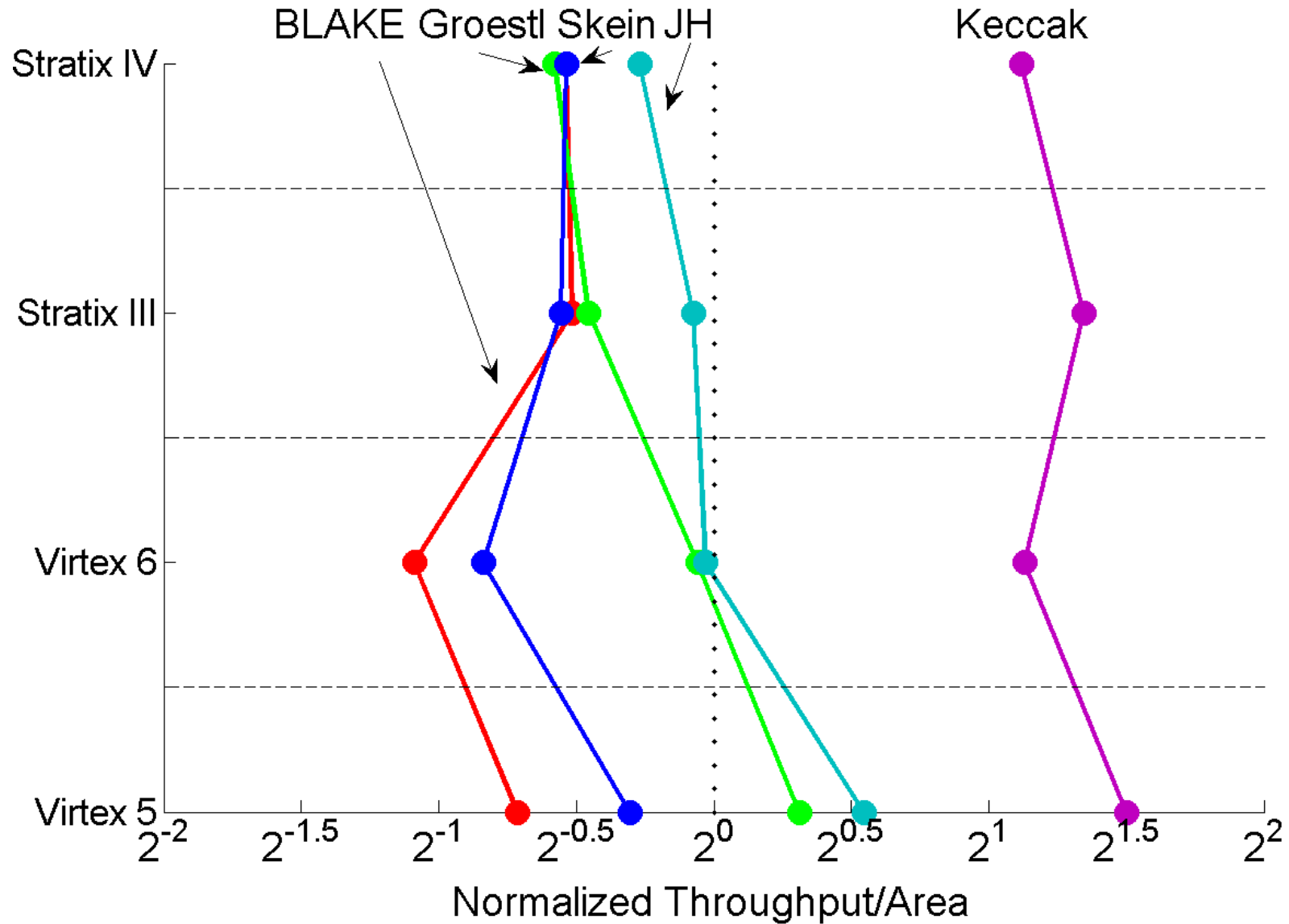
256-bit variants in Virtex 5



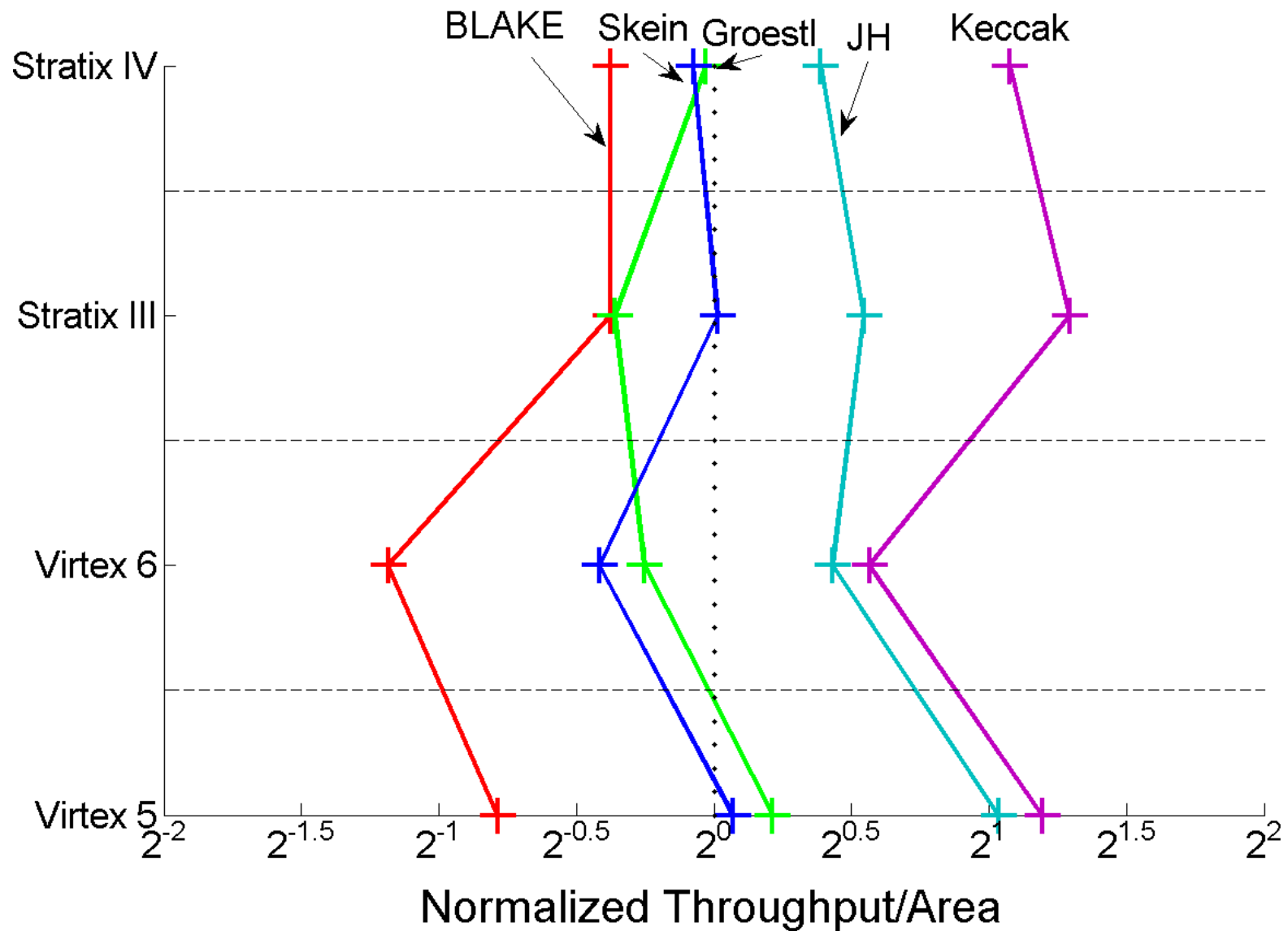
512-bit variants in Virtex 5



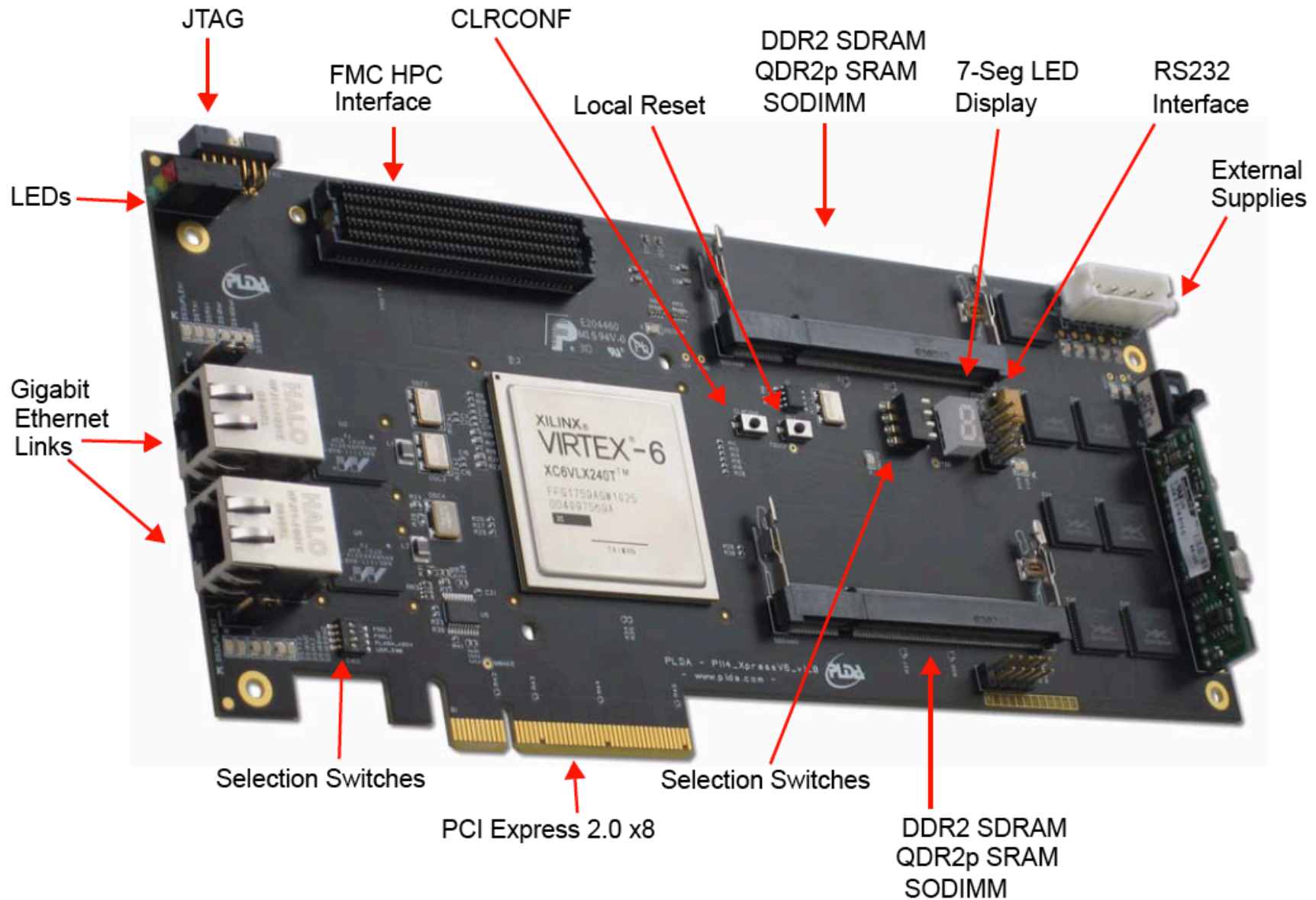
256-bit variants in 4 high-performance FPGA families



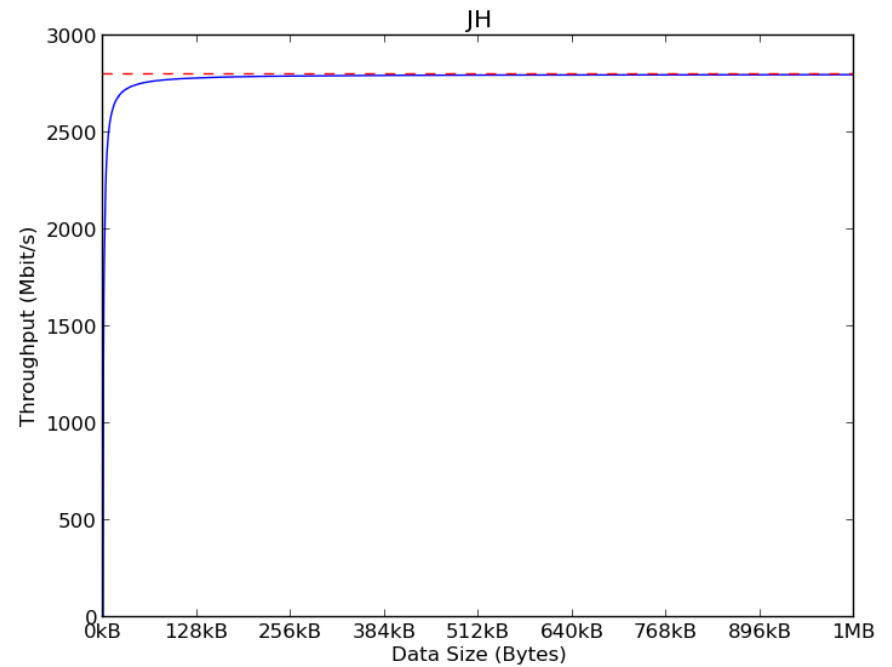
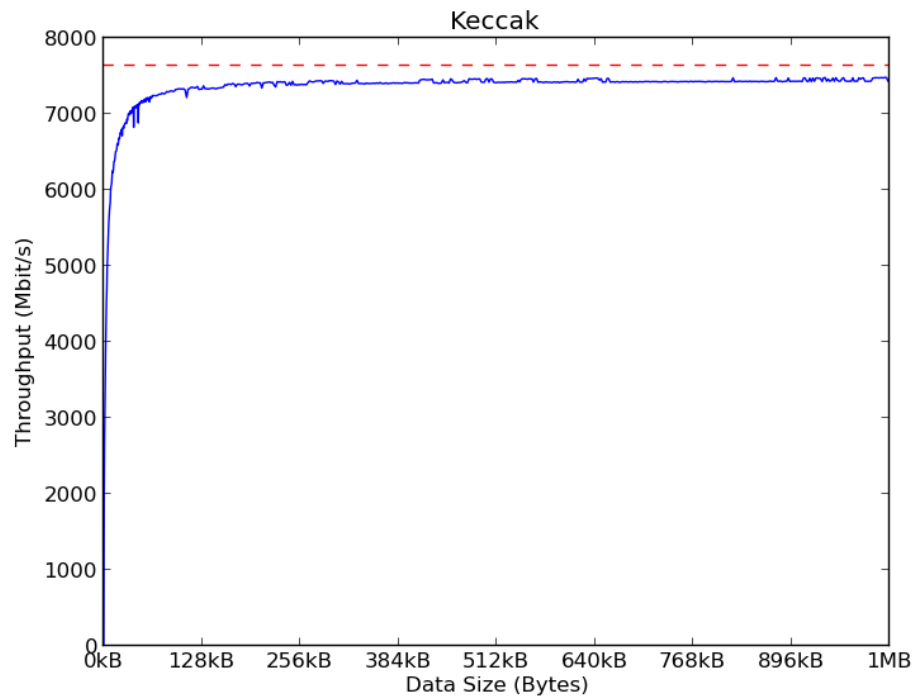
512-bit variants in 4 high-performance FPGA families



Experimental Testing using PCI Express Boards



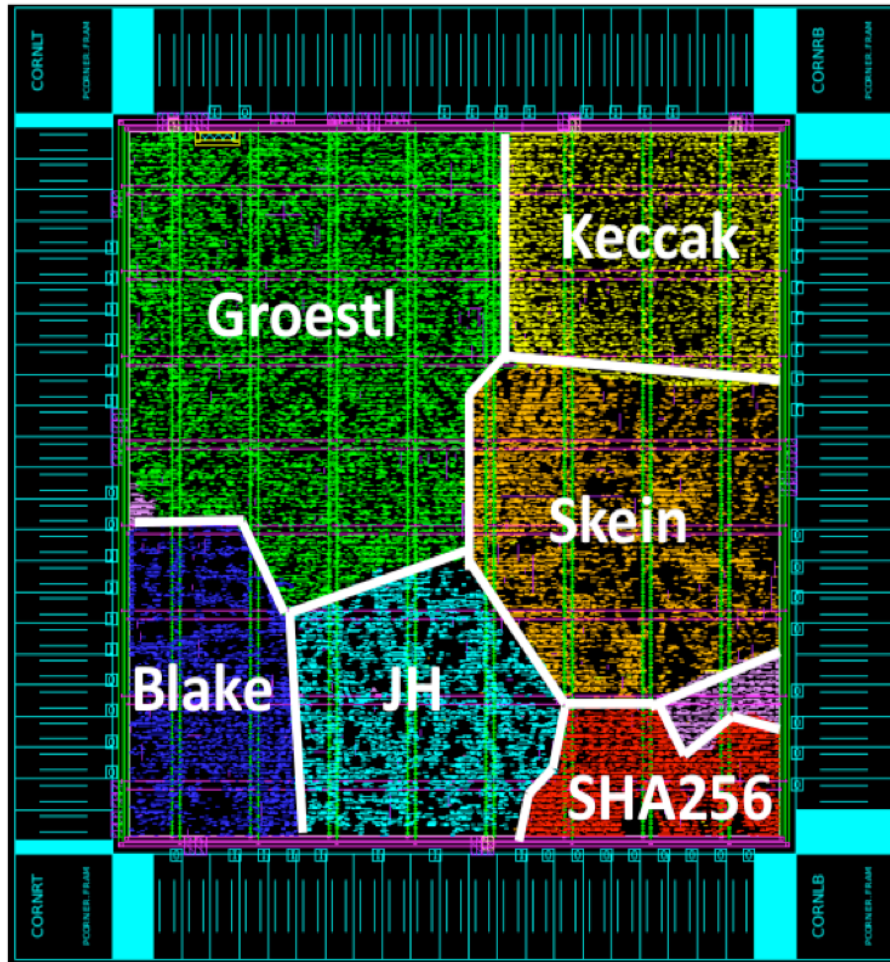
Experimental Throughput Measurements





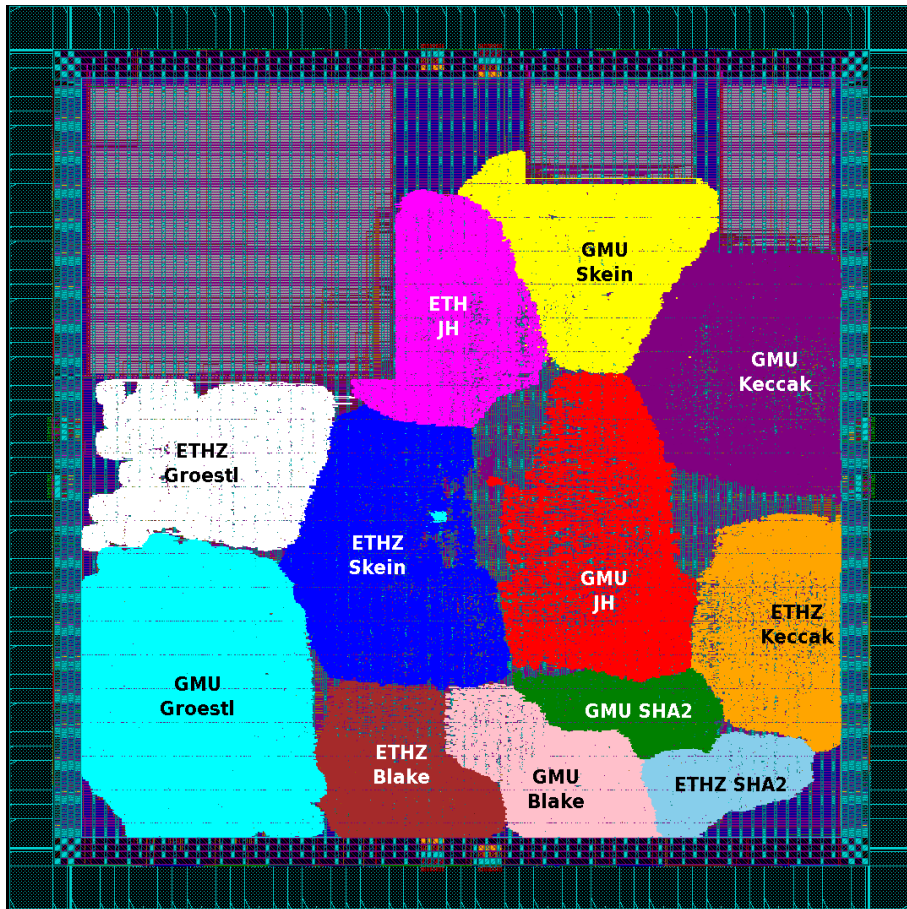
**SHA-3
in ASICs**

Virginia Tech ASIC



- IBM MOSIS 130nm process
- The first ASIC implementing 256-bit variants of 5 final SHA-3 candidates
- Taped-out in Feb. 2011, successfully tested in Summer 2011
- Multiple chips made available to other research labs

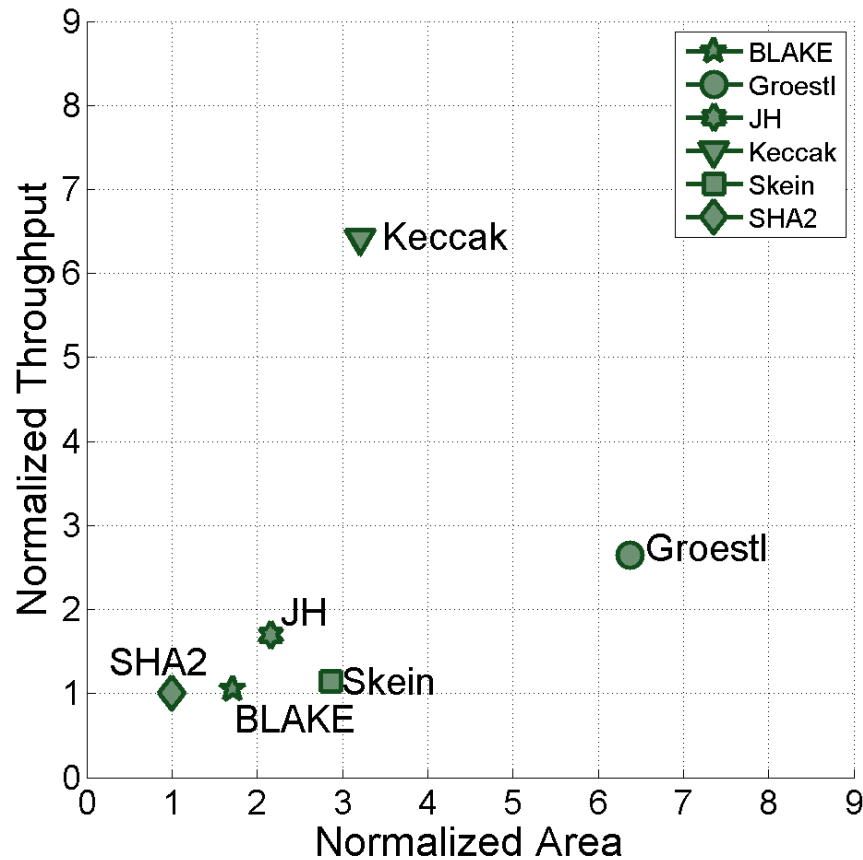
GMU/ETH Zurich ASIC



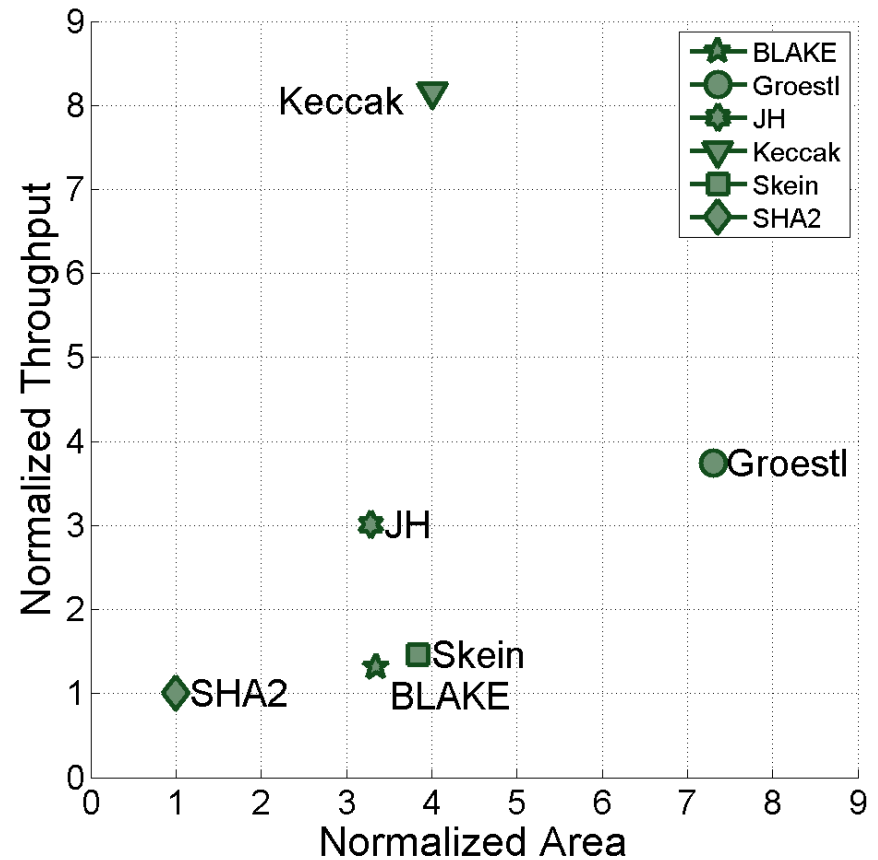
- standard-cell CMOS 65nm UMC ASIC process
- 256-bit variants of algorithms
- Taped-out in Oct. 2011, successfully tested in Feb. 2012

Correlation Between ASIC Results and FPGA Results

ASIC

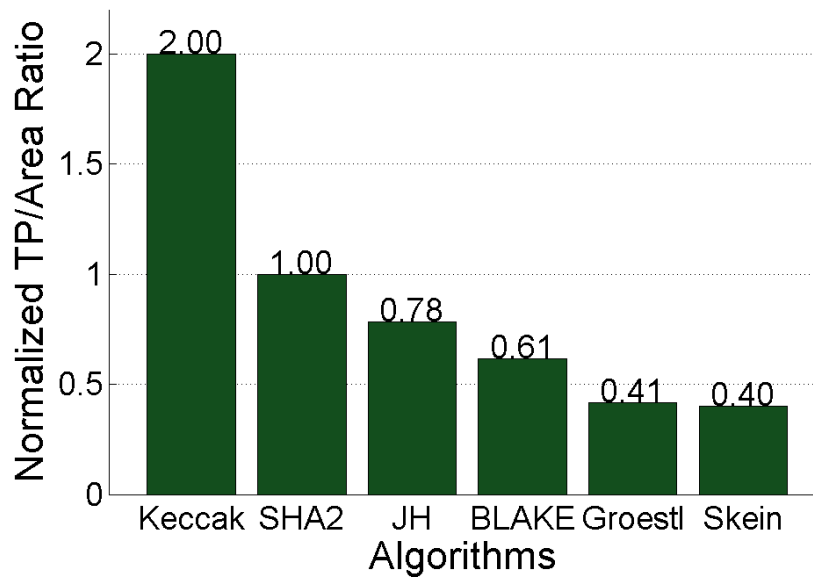


Stratix III FPGA

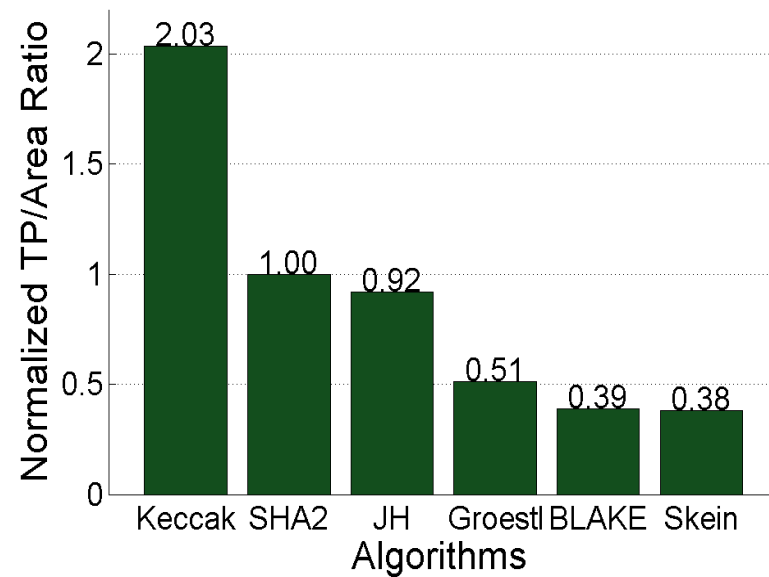


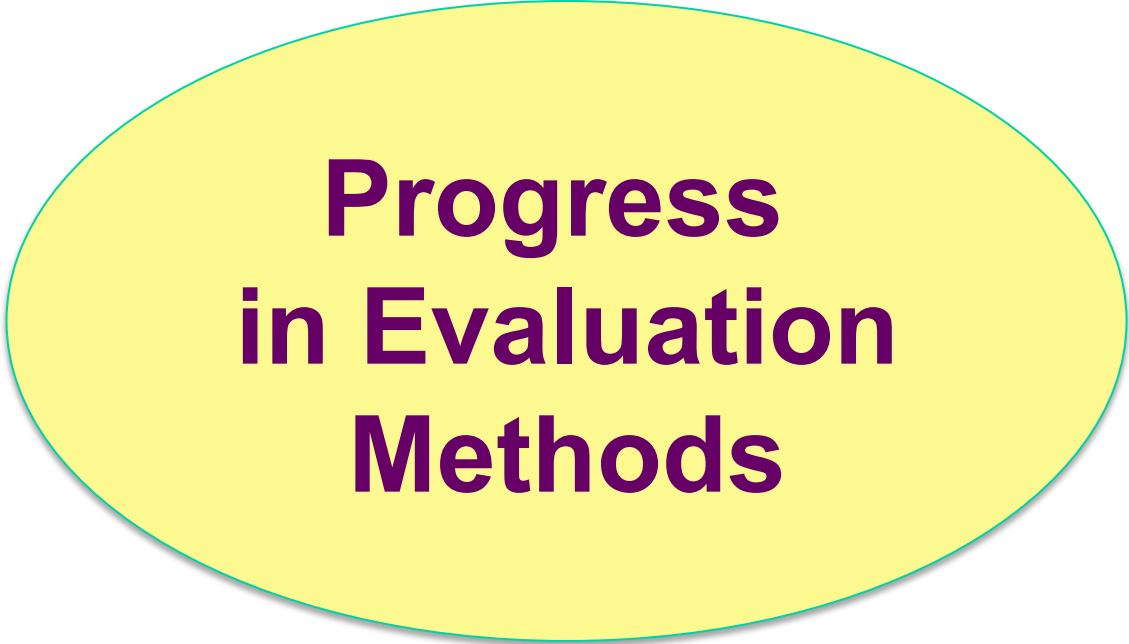
Correlation Between ASIC Results and FPGA Results

ASIC



Stratix III FPGA





**Progress
in Evaluation
Methods**

FPGA Evaluations - Summary

	AES	eSTREAM	SHA-3
Multiple FPGA families	No	No	Yes
Multiple architectures	No	Yes	Yes
Use of embedded resources	No	No	Yes
Primary optimization target	Throughput	Area Throughput/ Area	Throughput/ Area
Experimental results	No	No	Yes
Availability of source codes	No	No	Yes
Specialized tools	No	No	Yes

ASIC Evaluations - Summary

	AES	eSTREAM	SHA-3
Multiple processes/ libraries	No	No	Yes
Multiple architectures	No	Yes	Yes
Primary optimization target	Throughput	Power x Area x Time	Throughput /Area
Post-layout results	No	Yes	Yes
Experimental results	No	Yes	Yes
Availability of source codes	No	No	Yes
Specialized tools	No	No	No

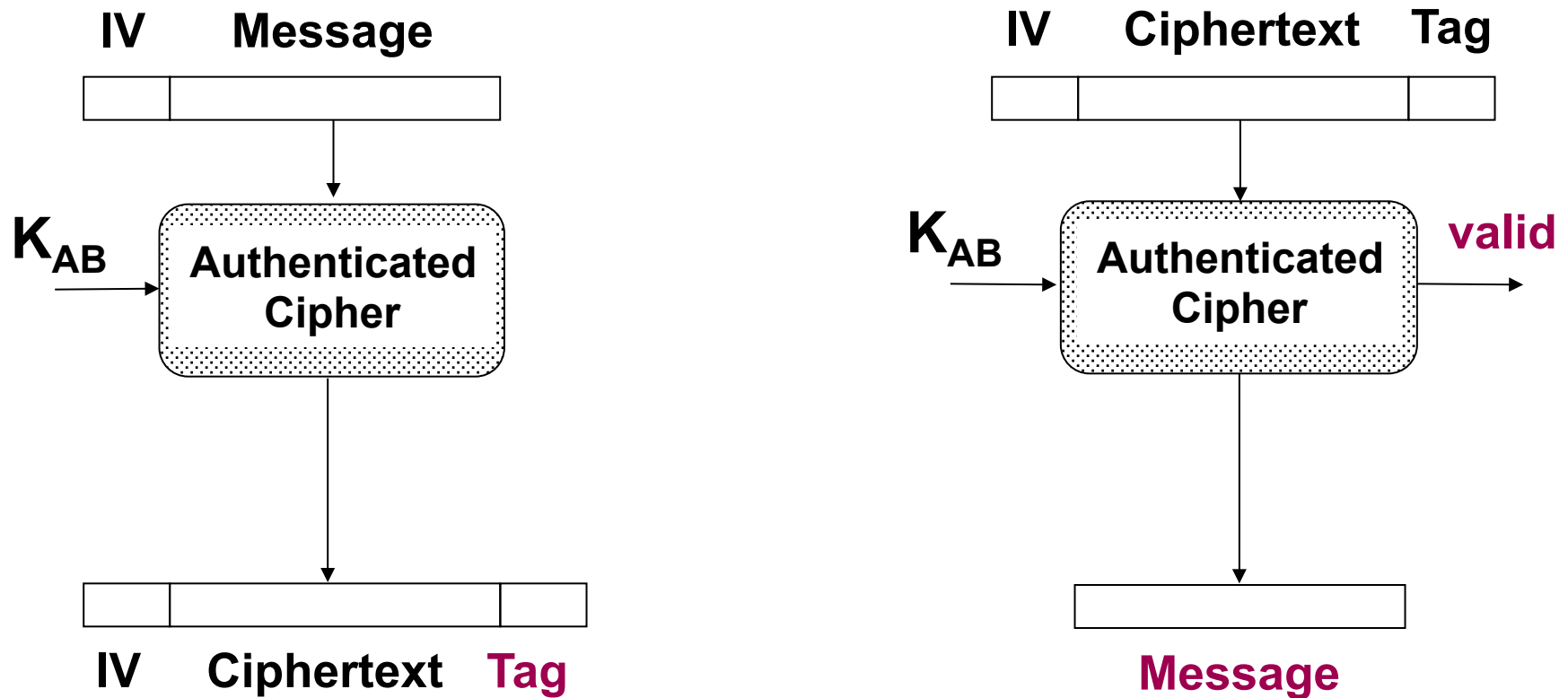


**CAESAR
Contest
2013-2017**

Authenticated Ciphers

Bob

Alice

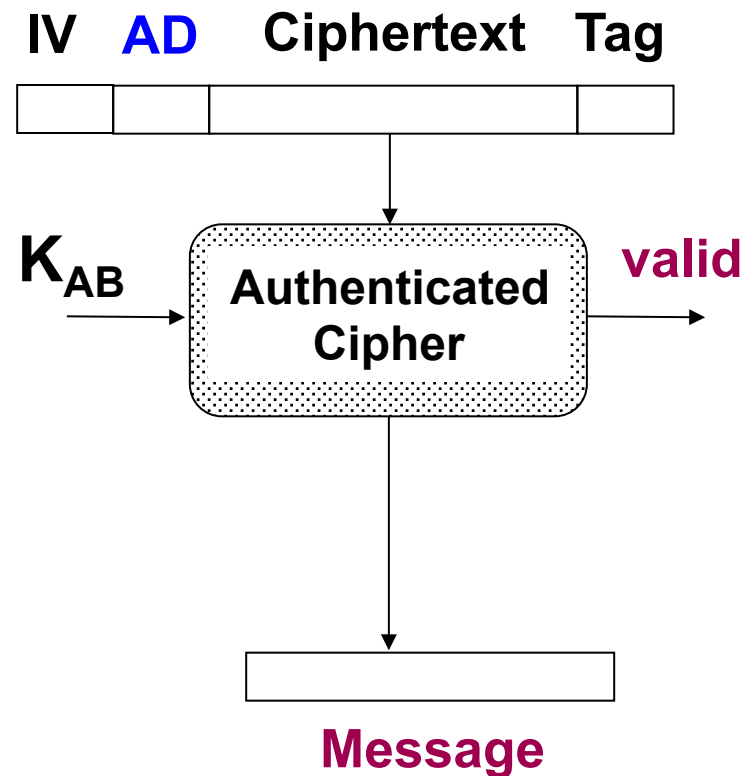
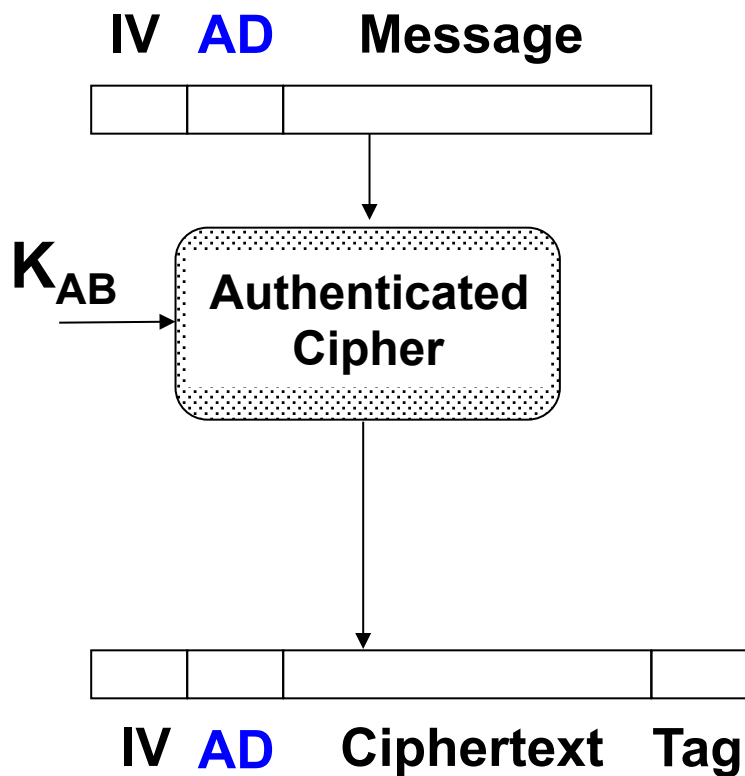


K_{AB} - Secret key of Alice and Bob
IV - Initialization Vector

Authenticated Ciphers with Associated Data

Bob

Alice



K_{AB} - Secret key of Alice and Bob
IV – Initialization Vector, AD – Associated Data

Contest Timeline

- 2014.03.15: Deadline for **first-round submissions**
- 2014.04.15: Deadline for **first-round software**
- 2015.01.15: Announcement of **second-round candidates**
- 2015.04.15: Deadline for **second-round Verilog/VHDL**
- 2015.12.15: Announcement of **third-round candidates**
- 2016.12.15: Announcement of **finalists**
- 2017.12.15: Announcement of **final portfolio**

Preliminary Work @ CERG GMU

- Development of a **standard input/output interface**
- Implementation of the most popular authenticated ciphers:
 - **AES-GCM**
 - **AES-OCB3**
 - **AES-CCM**
- Enhancing capabilities of **benchmarking tools**
- Customizing **database of results**



**Benchmarking
Tools**

Tools for Benchmarking Implementations of Cryptography

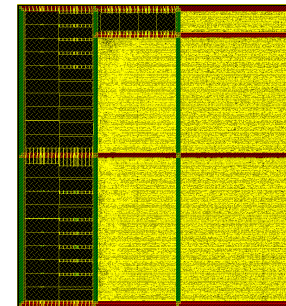
Software



FPGAs



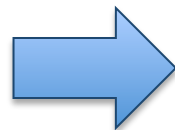
ASICs



eBACS

D. Bernstein (UIC)
T. Lange (TUE)

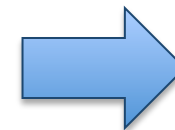
2006-present



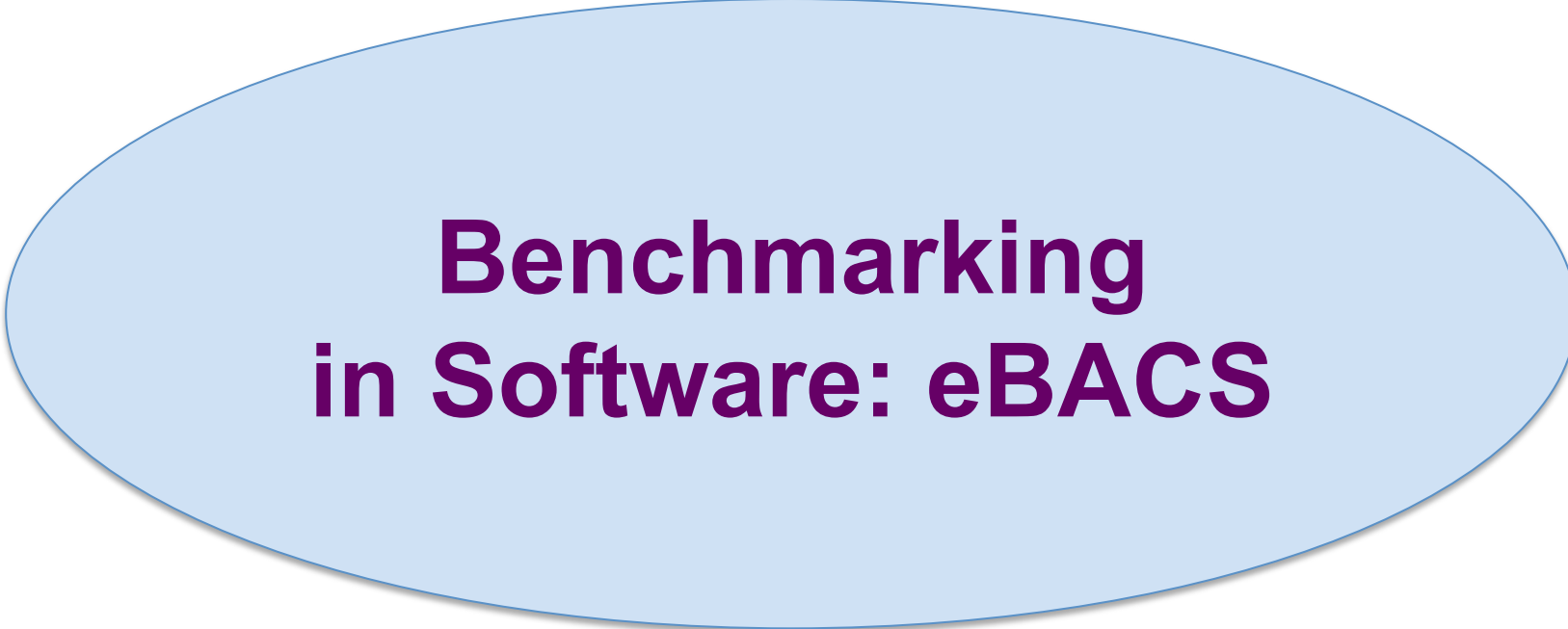
ATHENa

K. Gaj,
J. Kaps, et al.
(GMU)

2009-present



?



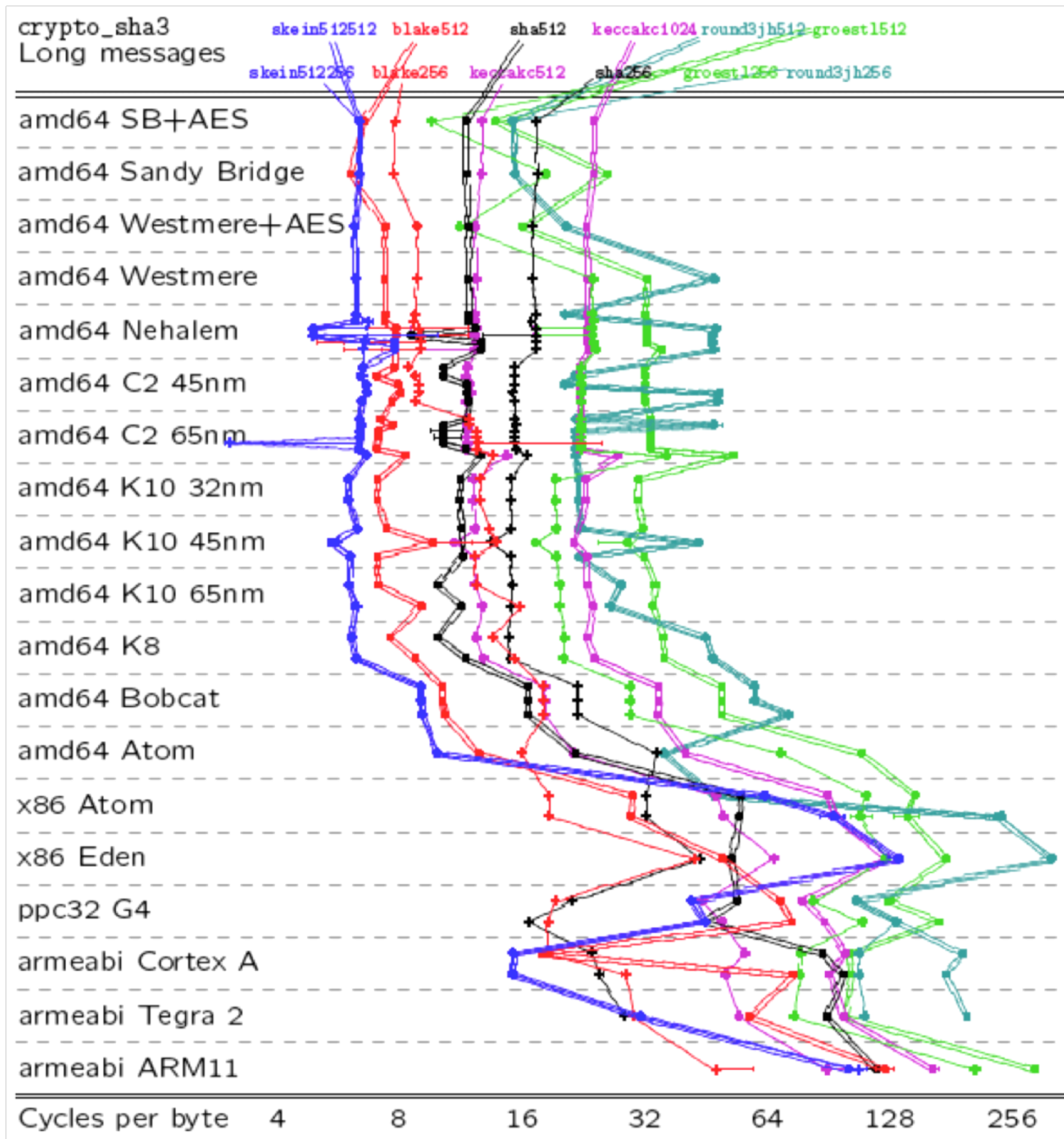
Benchmarking in Software: eBACS

eBACS: ECRYPT Benchmarking of Cryptographic Systems:

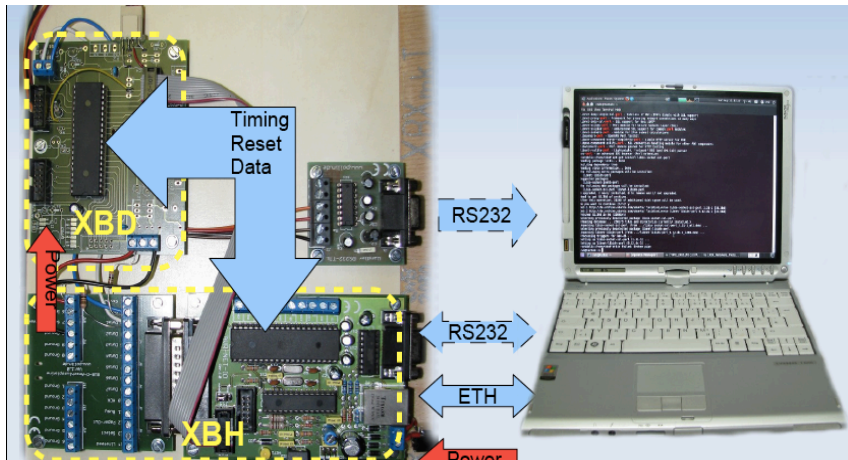
<http://bench.cr.yp.to/>

SUPERCOP - toolkit developed by D. Bernstein and T. Lange for measuring performance of cryptographic software

- measurements on multiple machines (currently over 90)
- each implementation is recompiled multiple times (currently over 1600 times) with various compiler options
- time measured in clock cycles/byte for multiple input/output sizes
- median, lower quartile (25th percentile), and upper quartile (75th percentile) reported
- standardized function arguments (common API)



SUPERCOP Extension for Microcontrollers – XBX: 2009-present



Allows on-board timing measurements

Supports at least the following microcontrollers:

8-bit:

Atmel ATmega1284P (AVR)

16-bit

TI MSP430

32-bit:

Atmel AT91RM9200 (ARM 920T)

TI AR7 (MIPS)

Intel XScale IXP420 (ARM v5TE)

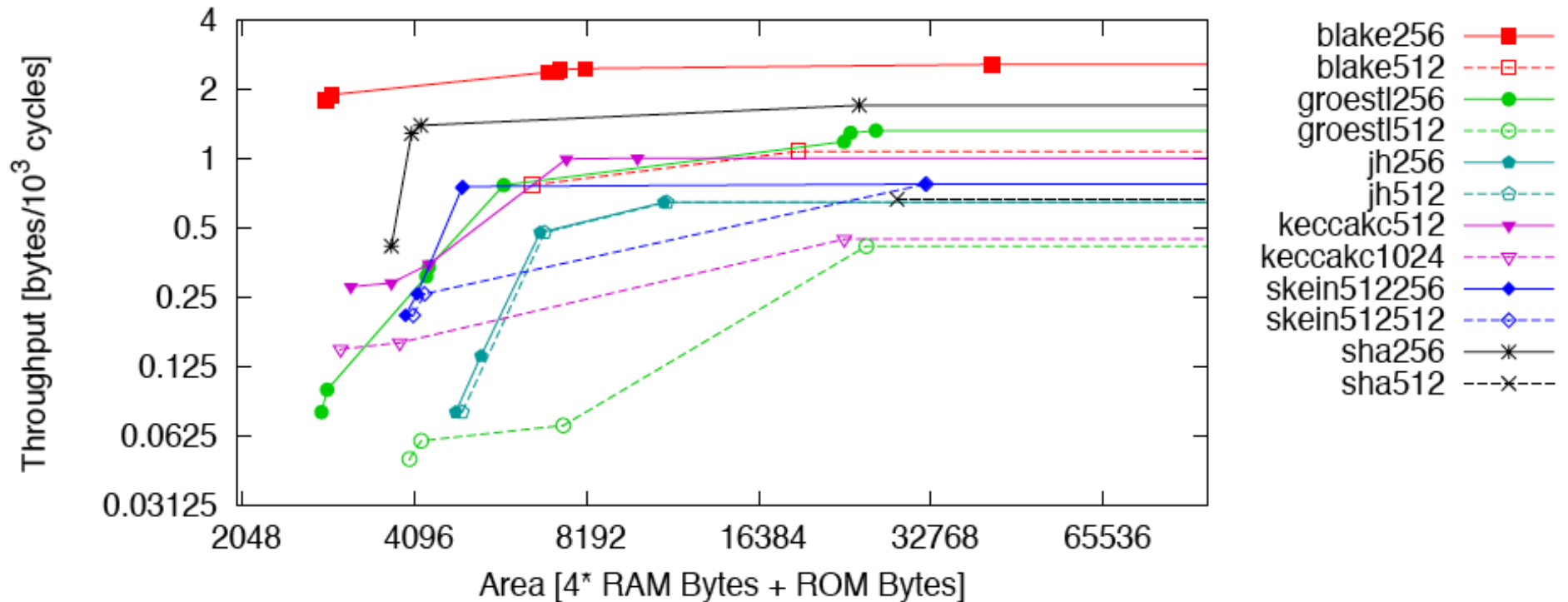
NXP LPC1114

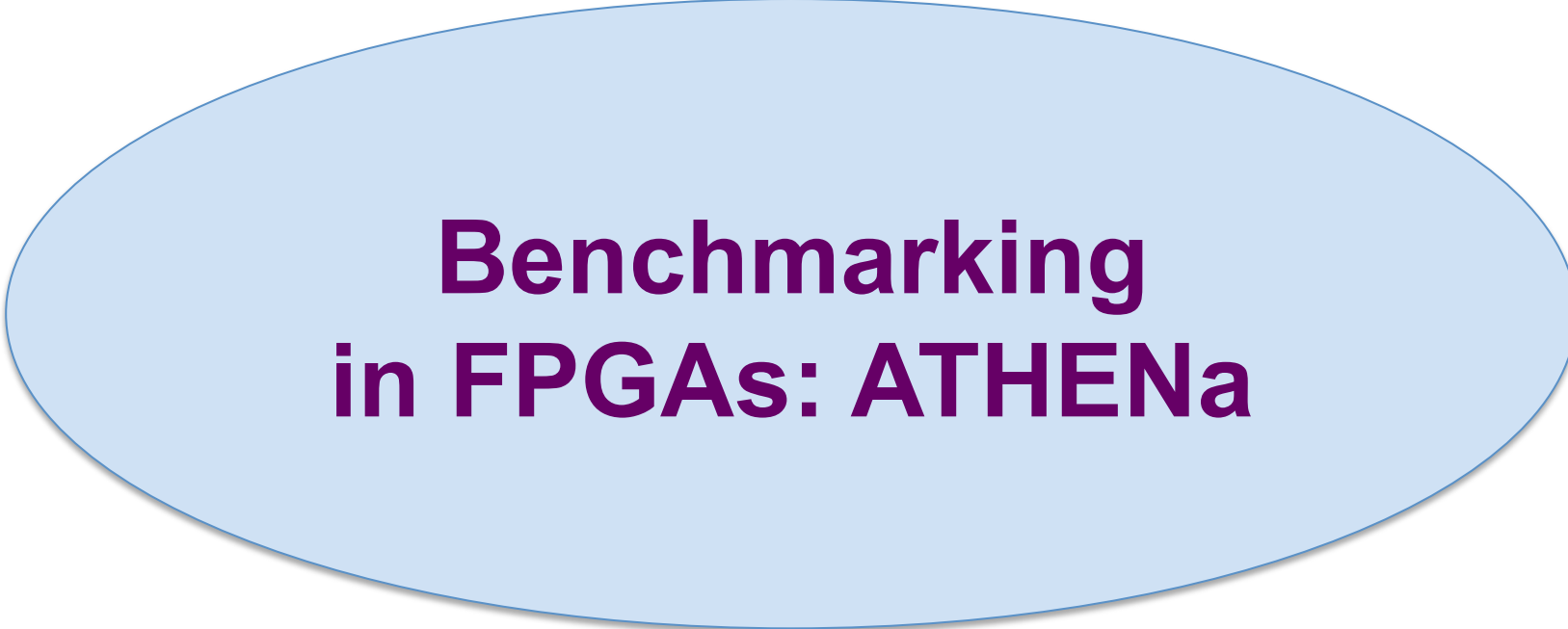
Cortex-M3 (ARM)

Developers:

- Christian Wenzel-Benner, ITK Engineering AG, Germany
- Jens Gräf, LiNetCo GmbH, Heiger, Germany

Microcontroller Performance for Texas Instruments MSP430





**Benchmarking
in FPGAs: ATHENa**

ATHENa – Automated Tool for Hardware Evaluation

<http://cryptography.gmu.edu/athena>



Open-source benchmarking environment,
written in Perl, aimed at
AUTOMATED generation of
OPTIMIZED results for
MULTIPLE hardware platforms.

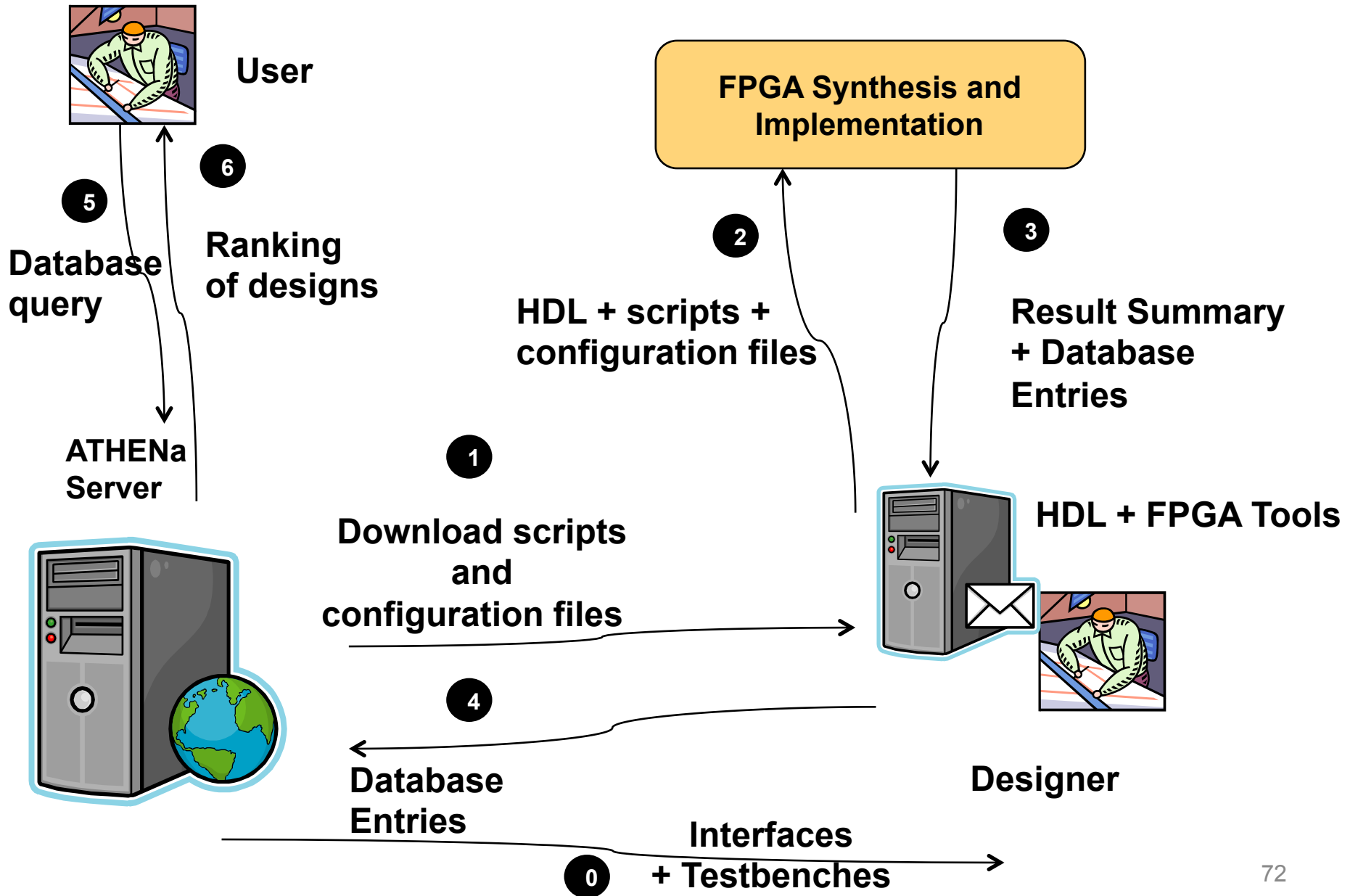
Why Athena?



"The Greek goddess Athena was frequently called upon to settle disputes between the gods or various mortals. Athena Goddess of Wisdom was known for her superb logic and intellect. Her decisions were usually well-considered, highly ethical, and seldom motivated by self-interest."

from "Athena, Greek Goddess of Wisdom and Craftsmanship"

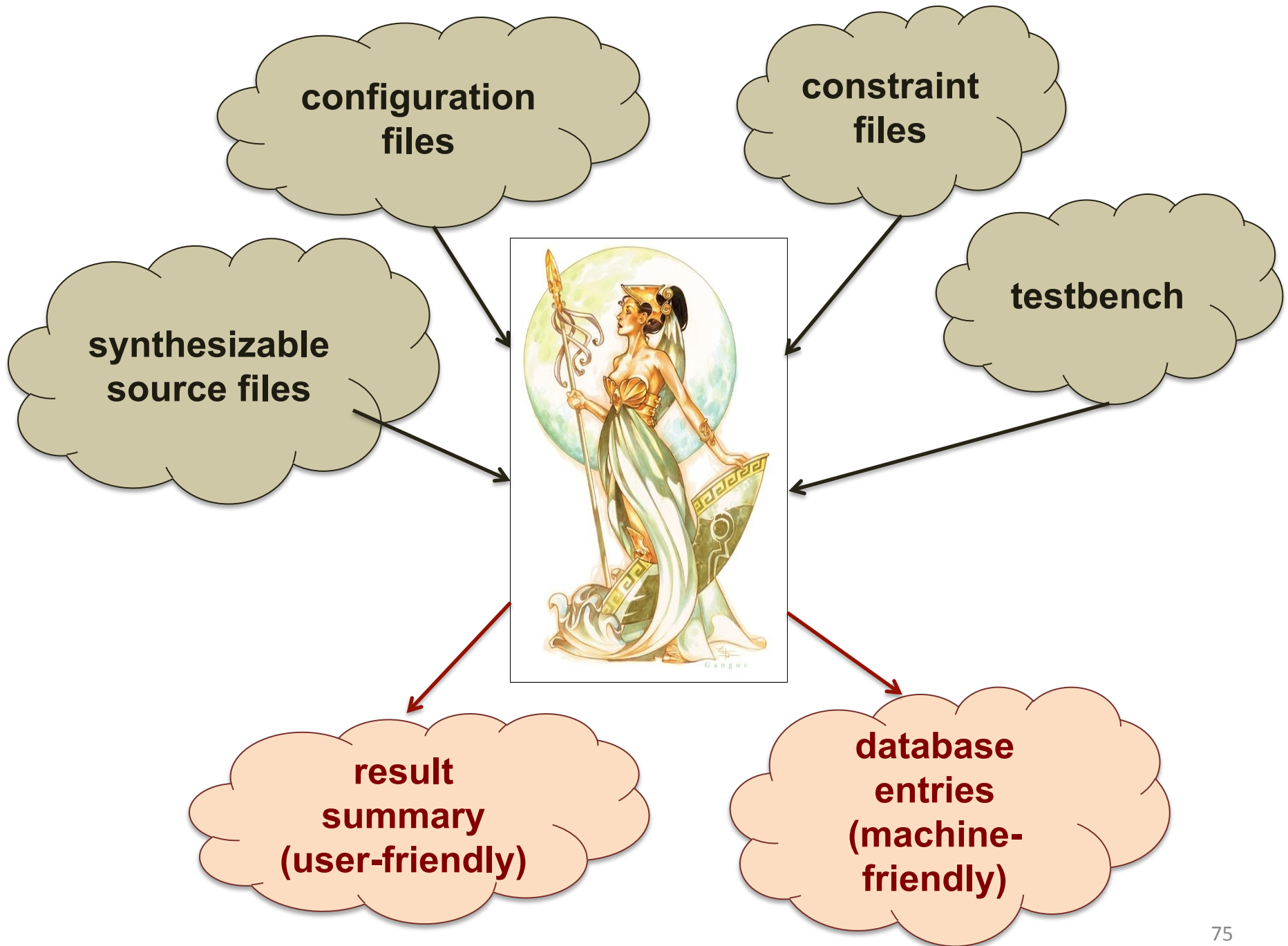
Basic Dataflow of ATHENa



Three Components of the ATHENa Environment

- **ATHENa Tool**
- **ATHENa Database of Results**
- **ATHENa Website**

ATHENa - Tool



ATHENa Major Features (1)

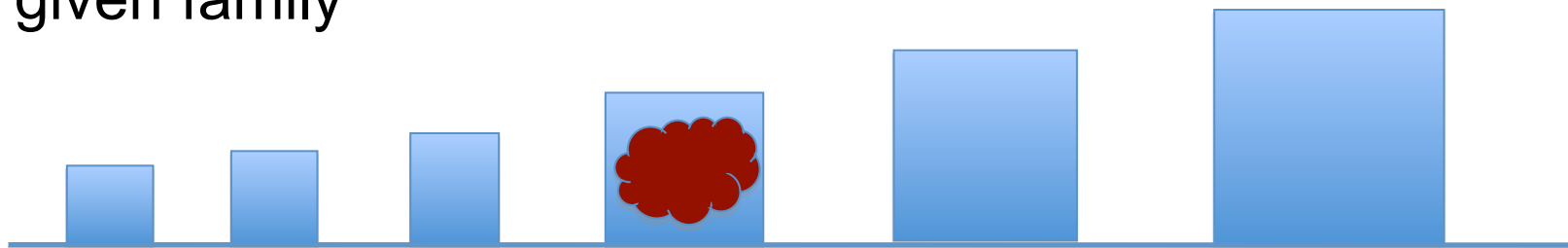
- synthesis, implementation, and timing analysis in **batch mode**
- support for devices and tools of **multiple FPGA vendors:**



- generation of results for **multiple families** of FPGAs of a given vendor

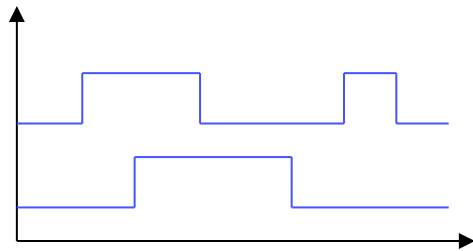


- automated choice of a **best-matching device** within a given family



ATHENa Major Features (2)

- **automated verification** of designs through simulation in batch mode



OR



- support for **multi-core processing**
- automated **extraction and tabulation of results**
- several **optimization strategies** aimed at finding
 - optimum options of tools
 - best target clock frequency
 - best starting point of placement

Generation of Results Facilitated by ATHENa



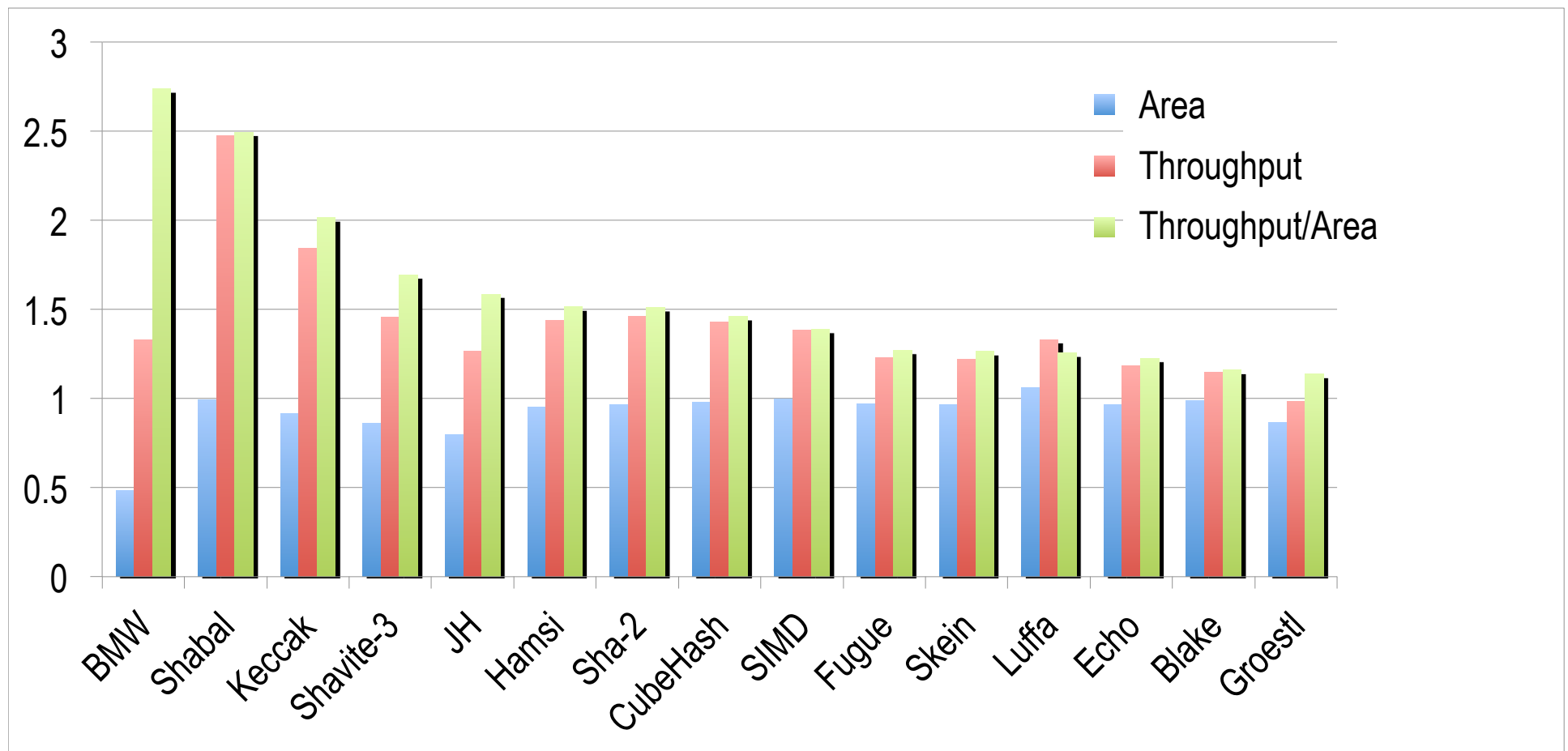
vs.

old days...

“working” with ATHENa...



Relative Improvement of Results from Using ATHENa Virtex 5, 512-bit Variants of Hash Functions



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



**ATHENa – Database
of Results**

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 Hash Size Primary Opt Target Arch Type Max #Streams Padding Family

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 a Single-Best (Unique) Result per Each Algorithm

Show entries

Copy CSV Excel

Algorithm		Design				
Result ID	Algorithm <small>Disable Unique</small>	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)

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

Matching fields in grey
Non-matching fields in red and blue

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

ATHENa Result Replication Files

- **Scripts and configuration files sufficient to easily reproduce all results (without repeating optimizations)**
 - **Automatically created by ATHENa for all results generated using ATHENa**
 - **Stored in the ATHENa Database**
-

In the same spirit of **Reproducible Research** as:

- J. Claerbout (Stanford University)
“Electronic documents give reproducible research a new meaning,”
in *Proc. 62nd Ann. Int. Meeting of the Soc. of Exploration Geophysics*, **1992**,
<http://sepwww.stanford.edu/doku.php?id=sep:research:reproducible:seg92>
.....
- Patrick Vandewalle¹, Jelena Kovacevic², and Martin Vetterli¹ (¹EPFL, ²CMU)
Reproducible research in signal processing - what, why, and how.
IEEE Signal Processing Magazine, May **2009**. <http://rr.epfl.ch/17/>

ATHENa - Website

ATHENa Website

<http://cryptography.gmu.edu/athena/>

- **Download** of ATHENa Tool
- Links to **related tools**

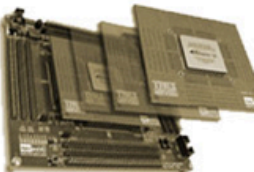
Cryptographic Competitions in FPGAs & ASICs

- **Specifications** of candidates
- **Interface** proposals
- RTL **source codes**
- **Testbenches**
- ATHENa database of **results**
- Related **papers & presentations**

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



0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100 0100100
 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101 1110101
 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100 1111100
 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000 0101000
 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001 1101001
 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100 1101100
 110101 110101 110101 110101 110101 110101 110101 110101 110101 110101 110101 110101 110101 110101 110101



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

Selected SHA-3 Source Codes Available in Public Domain

- AIST-RCIS: <http://www.rcis.aist.go.jp/special/SASEBO/SHA3-en.html>
- University College Cork, Queens University Belfast, RMIT University, Melbourne, Australia:
<http://www.ucc.ie/en/crypto/sha-3hardware/>
- Virginia Tech: <http://rijndael.ece.vt.edu/sha3/soucecodes.html>
- ETH Zurich: <http://www.iis.ee.ethz.ch/~sha3/>
- George Mason University: <http://cryptography.gmu.edu/athena>
- BLAKE Team: <http://www.131002.net/blake/>
- Keccak Team: <http://keccak.noekeon.org/>

Benchmarking Goals Facilitated by ATHENa

Comparing multiple:

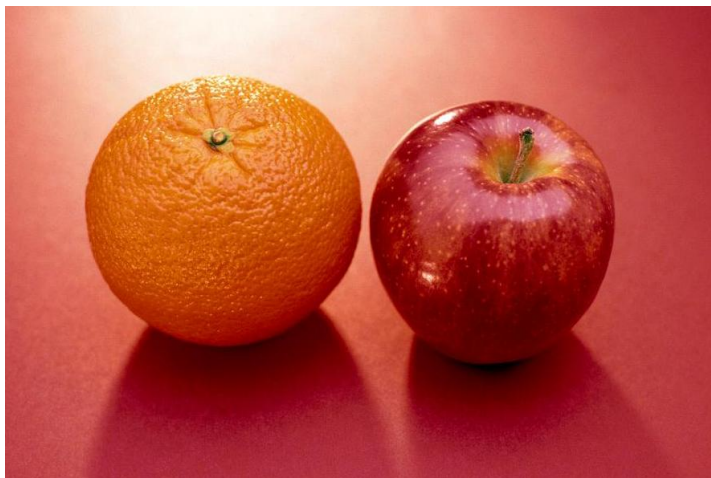
1. cryptographic **algorithms**
2. hardware **architectures or implementations** of the same cryptographic algorithm
3. hardware **platforms** from the point of view of their suitability for the implementation of a given algorithm, (e.g., choice of an FPGA device or FPGA board)
4. **tools and languages** in terms of quality of results they generate (e.g. Verilog vs. VHDL, Synplify Premier vs. Xilinx XST, ISE vs. Vivado)



**Open
Problems**

Objective Benchmarking Difficulties

- lack of standard one-fits-all interfaces
- stand-alone performance vs. performance as a part of a bigger system
- heuristic optimization strategies
- time & effort spent on optimization



or



Objective Benchmarking Difficulties

- lack of convenient cost metric in FPGAs
- accuracy of power estimators in ASICs & FPGAs
- human factor (skills of designers, order of implementations, etc.)
- verifiability of results

Resource Utilization Vector

(#Logic blocks, #Multipliers/DSP units, #RAM blocks)

Xilinx

Spartan 3: (#CLB_slices, #multipliers, #Block_RAMs)

Virtex 5: (#CLB_slices, #DSP units, #Block_RAMs)

Altera

Cyclone III: (#LEs, #multipliers, #RAM_bits)

Stratix III: (#ALUTs, #DSP units, #RAM_bits)

Potential Problems with Publishing Source Codes

- **Export control** regulations for cryptography

Check: Bert-Jaap Koops, Crypto Law Survey

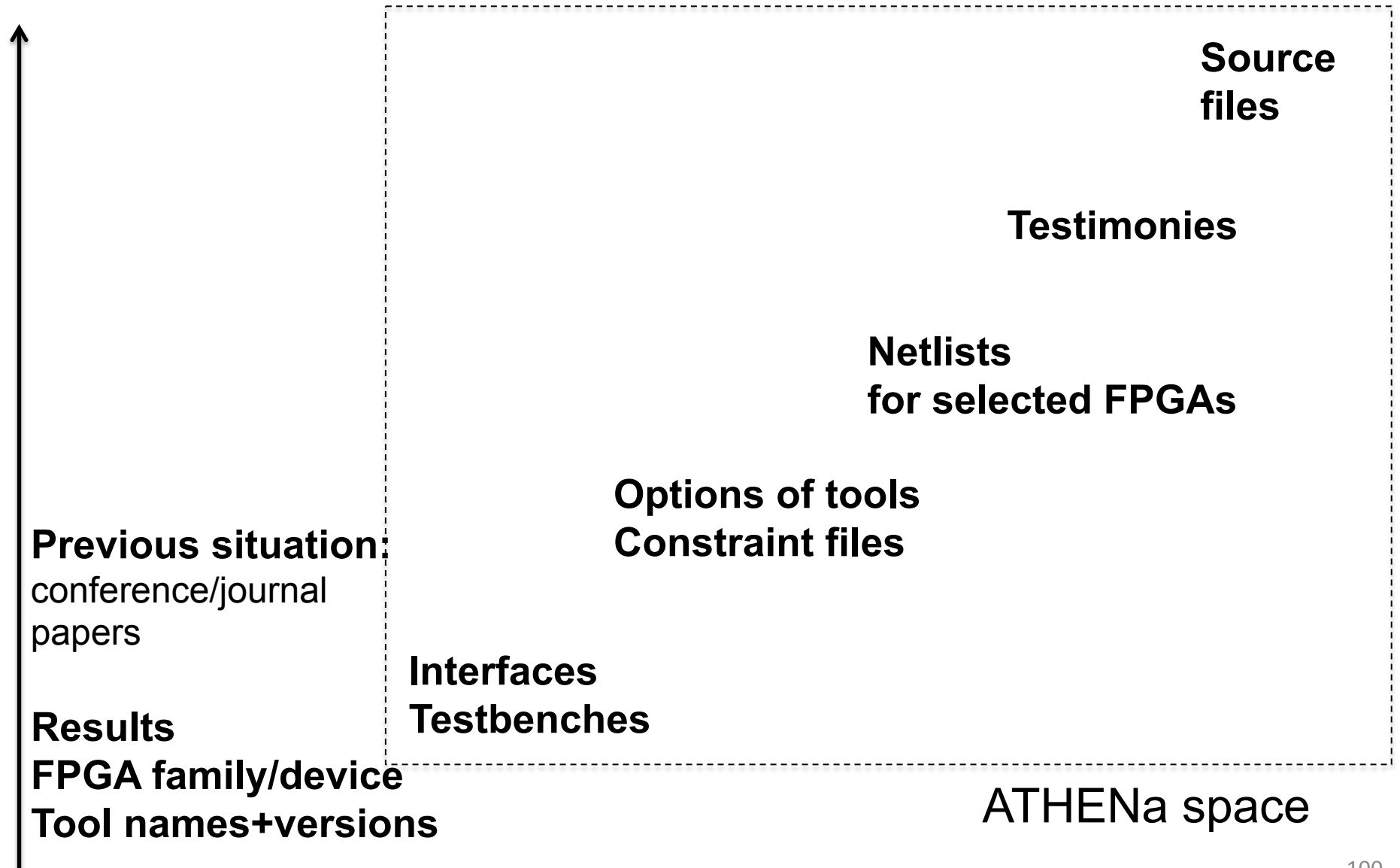
<http://www.cryptolaw.org>

Commercial interests

- **Competition with other groups** for grants and publications in the most renowned journals and conference proceedings

How to assure verifiability of results?

Level of openness



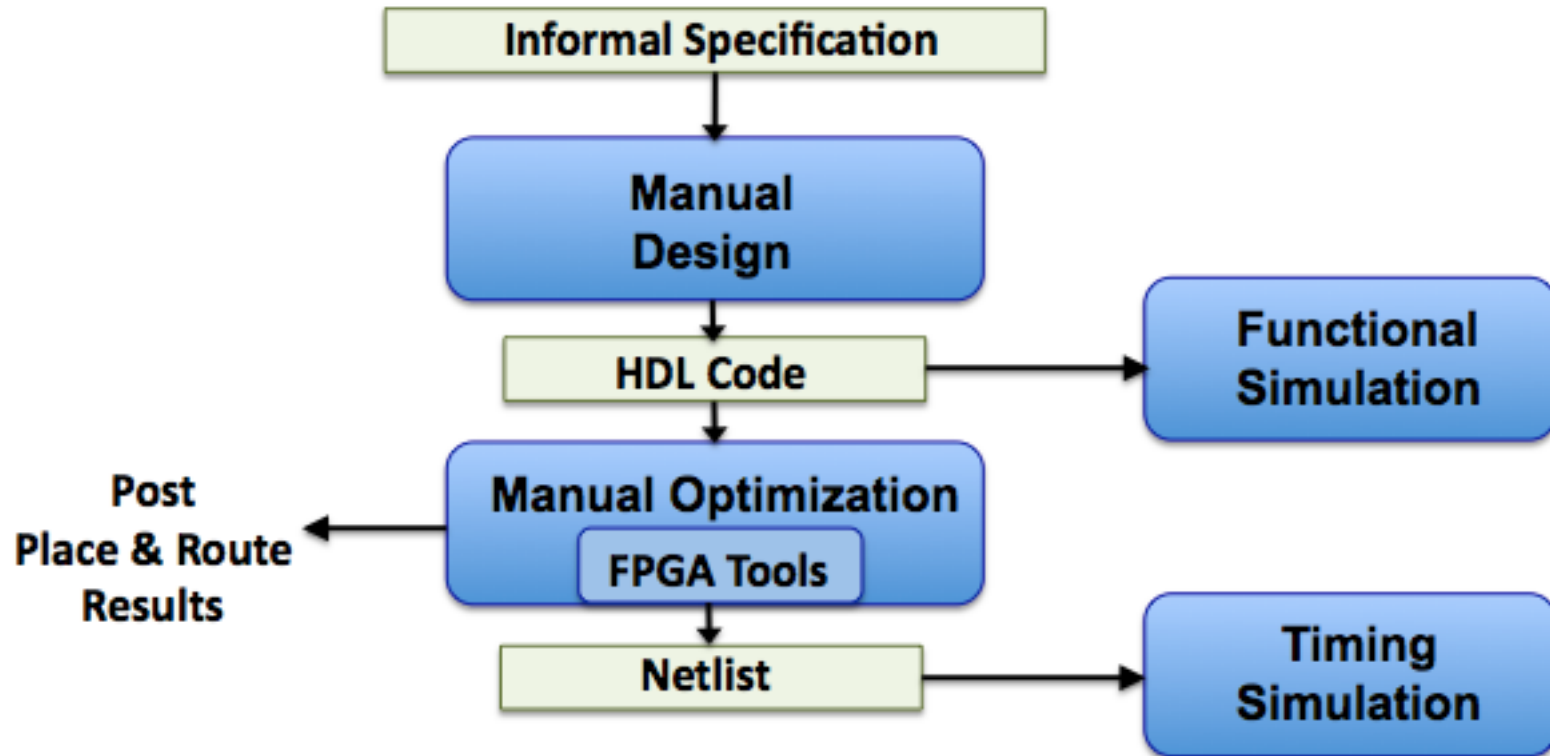
Initial Evaluation by High-Level Synthesis Tools?

Initial number of candidates

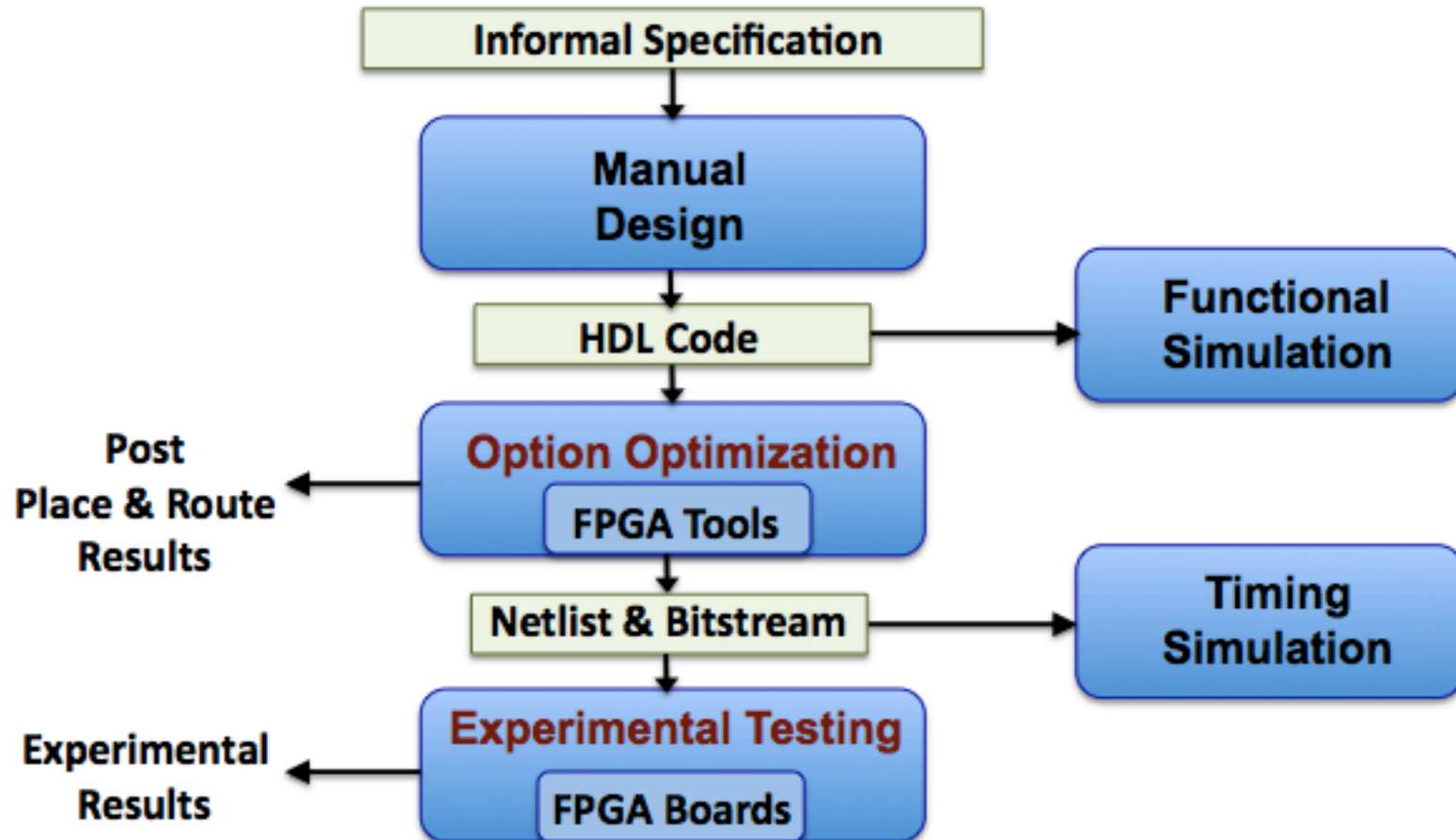
AES	15
eSTREAM	34
SHA-3	51
CAESAR	57

- All hardware implementations so far developed using RTL HDL
- Growing number of candidates in subsequent contests
- Each submission includes reference implementation in C
- Results from High-Level Synthesis could have a large impact in early stages of the competitions
- Results and RTL codes from previous contests form interesting benchmarks for High-Level synthesis tools

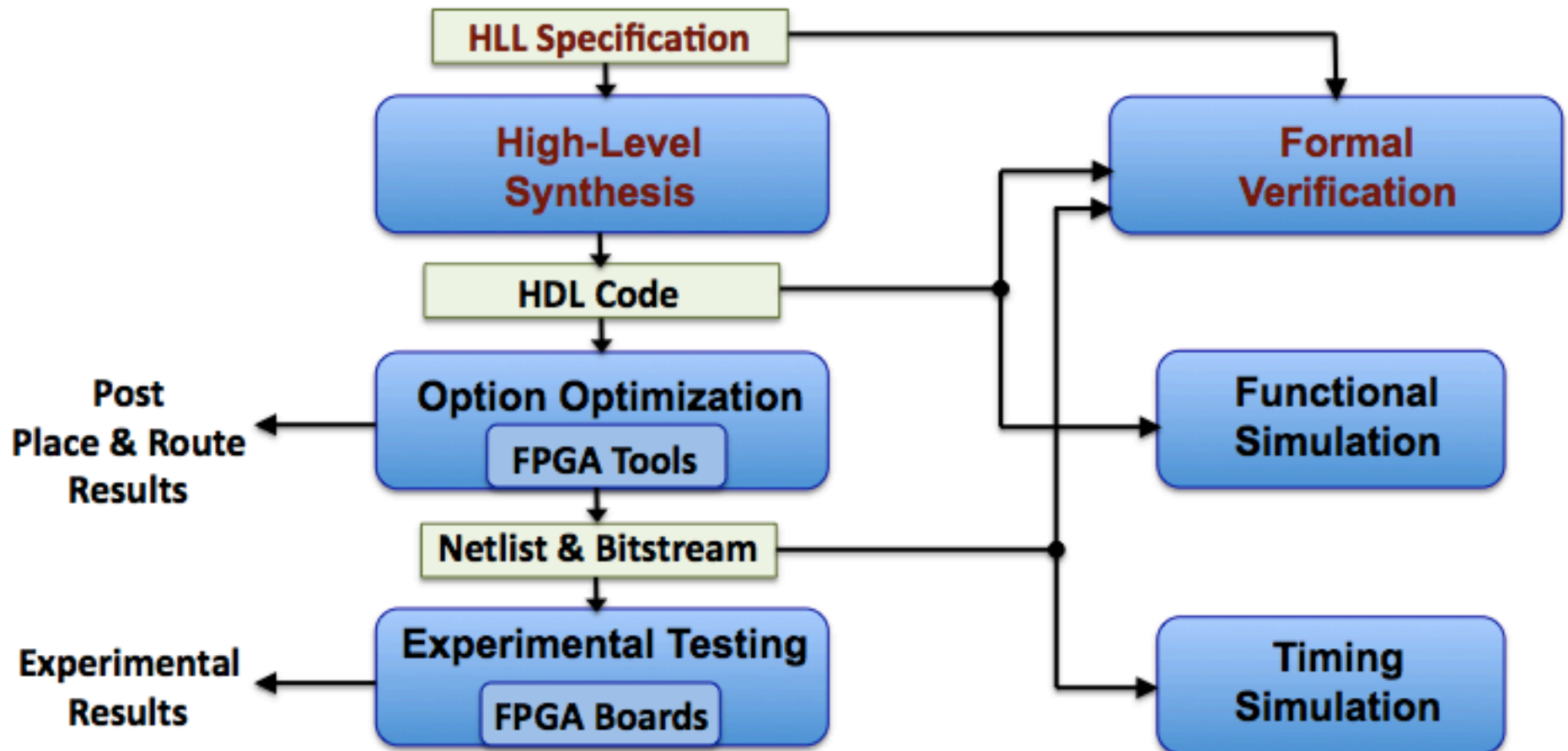
Past Methodology



Current Methodology

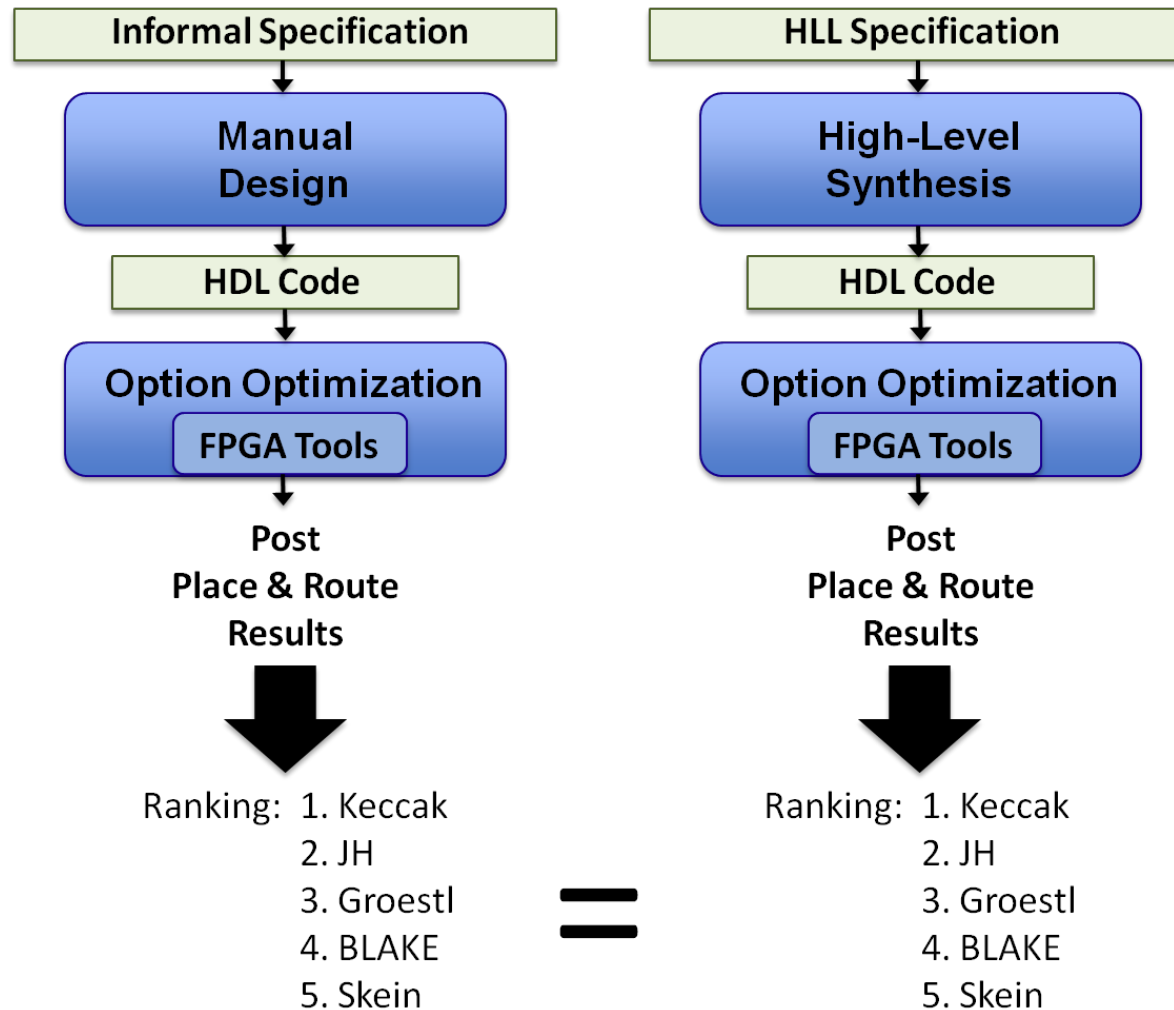


Possible Future Methodology



Currently explored at GMU

Accuracy & Development Time?



Same Ranking

Development Time = 10 weeks => Development Time = 1 week

Turning Thousands of Results into a Single Fair Ranking

- **Choosing** which **FPGA families / ASIC libraries** should be included in the comparison
 - wide range?
 - only most recent?
 - vendors with the largest market share?
 - wide spectrum of vendors?
- Methods for **combining multiple results** into single ranking

Thousands of results
on tens of platforms



- 1.
- 2.
- 3.
- 4.
- 5.

Turning Thousands of Results into Fair Ranking

- Deciding on most important **application scenarios**
 - Throughput – Cost – Power range
from RFIDs to High-speed security gateways
 - Assigning weights to different scenarios

**Help/recommendation from the system developers
highly appreciated**

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**
- **Security Experts, 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!**

Thank you!

Questions?



Questions?

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



Backup

Conferences & Journals



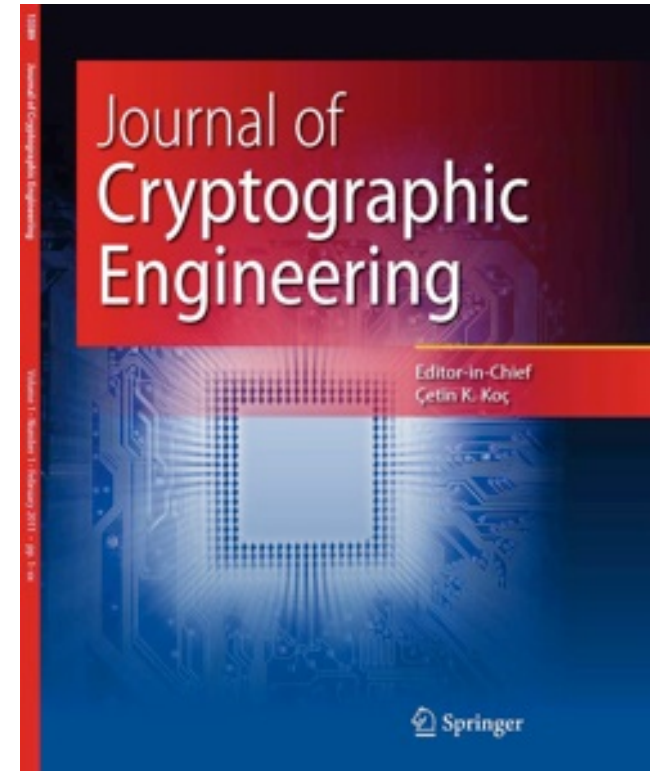
ECRYPT II
↓↑↔↻⊕↻^ ↓

**Conferences
& workshops
devoted
to specific contests**

**CAESAR:
DIAC – Directions
In Authenticated
Ciphers**



**Since 1999
USA-Europe-Asia
CHES 2014, Busan, Korea
Sep. 23-26, 2014**



Since Jan. 2011

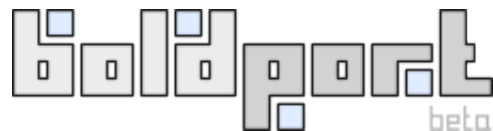
Other (Somewhat) Similar Tools



ExploreAhead (part of PlanAhead)



Design Space Explorer (DSE)



Boldport Flow



EDAx10 Cloud Platform

Distinguishing Features of ATHENa

- Support for **multiple tools** from **multiple vendors**
- Optimization strategies aimed at the **best possible performance** rather than design closure
- Extraction and **presentation of results**
- Seamless **integration with** the ATHENa **database of results**

How to measure hardware cost in FPGAs?

1. Stand-alone cryptographic core on an FPGA



Cost of the smallest FPGA that can fit the core?

Unit: USD [FPGA vendors would need to publish MSRP (manufacturer's suggested retail price) of their chips]

– not very likely, very volatile metric

or **size of the chip in mm²** - easy to obtain

2. Part of an FPGA System On-Chip

Resource utilization described by a vector:

(#CLB slices, #MULs/DSP units, #BRAMs) for Xilinx

(#LEs/ALUTs, #MULs/DSP units, #membits) for Altera

Difficulty of turning vector into a single number
representing cost

Limitations of the AES Evaluation

- Optimization for **maximum throughput**
- **Single** high-speed **architecture** per candidate
- **No use of embedded resources** of FPGAs (Block RAMs, dedicated multipliers)
- **Single FPGA family** from a single vendor:
Xilinx Virtex

Features of the SHA-3 Round 2 Evaluation

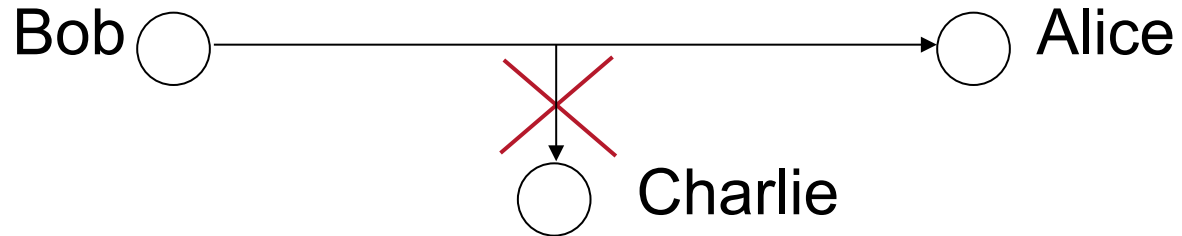
- Optimization for **maximum throughput to area ratio**
- **10 FPGA families** from two major vendors :
Xilinx and Altera

But still...

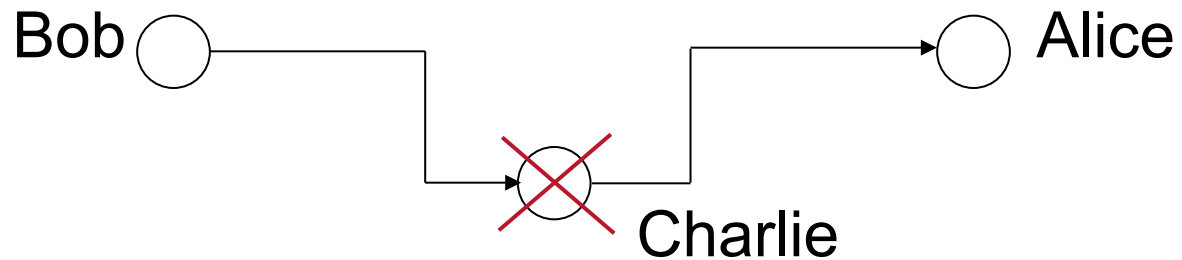
- **Single high-speed architecture** per candidate
- **No use of embedded resources** of FPGAs (Block RAMs, dedicated multipliers, DSP units)

Authenticated Ciphers: Security Services

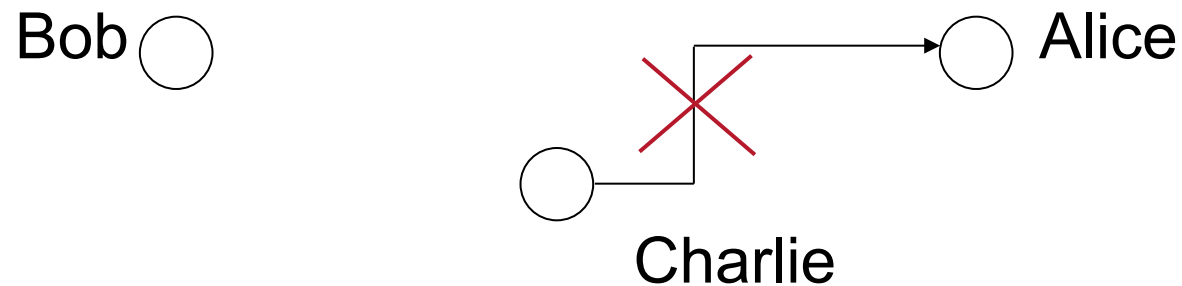
1. Confidentiality



2. Message integrity



3. Message authentication



Similarities in comparing software and FPGA designs

- **relatively few major vendors**
 - Intel and AMD for general-purpose microprocessors
 - Xilinx and Altera for FPGAs
- **good quality tools available for free**
 - GNU compilers for software
 - full or slightly reduced versions of tools for FPGAs
- **multiple options of tools**
- software programs can be written **targeting** a specific processor;
HDL codes can be written targeting a specific FPGA family
- **low level optimizations** possible but typically not portable:
in software - assembly language; in FPGAs - low level macros

Differences in comparing software and FPGA designs

- in software **speed** is a major parameter; in hardware **speed and area** need to be taken into account and can be often traded one for the other
- in software clock frequency is fixed for a given processor; tools try to optimize the **sequence of instructions**; in FPGAs **clock frequency** is determined by the implemented circuit; tools try to optimize the most critical paths, and thus minimize the clock period
- in software execution time is **measured directly** with some non-negligible measurement error; in FPGAs minimum clock period is reported by software tools; minimum execution time is **calculated**;
- **open source** software cryptographic libraries widely available; very few open source cryptographic hardware designs

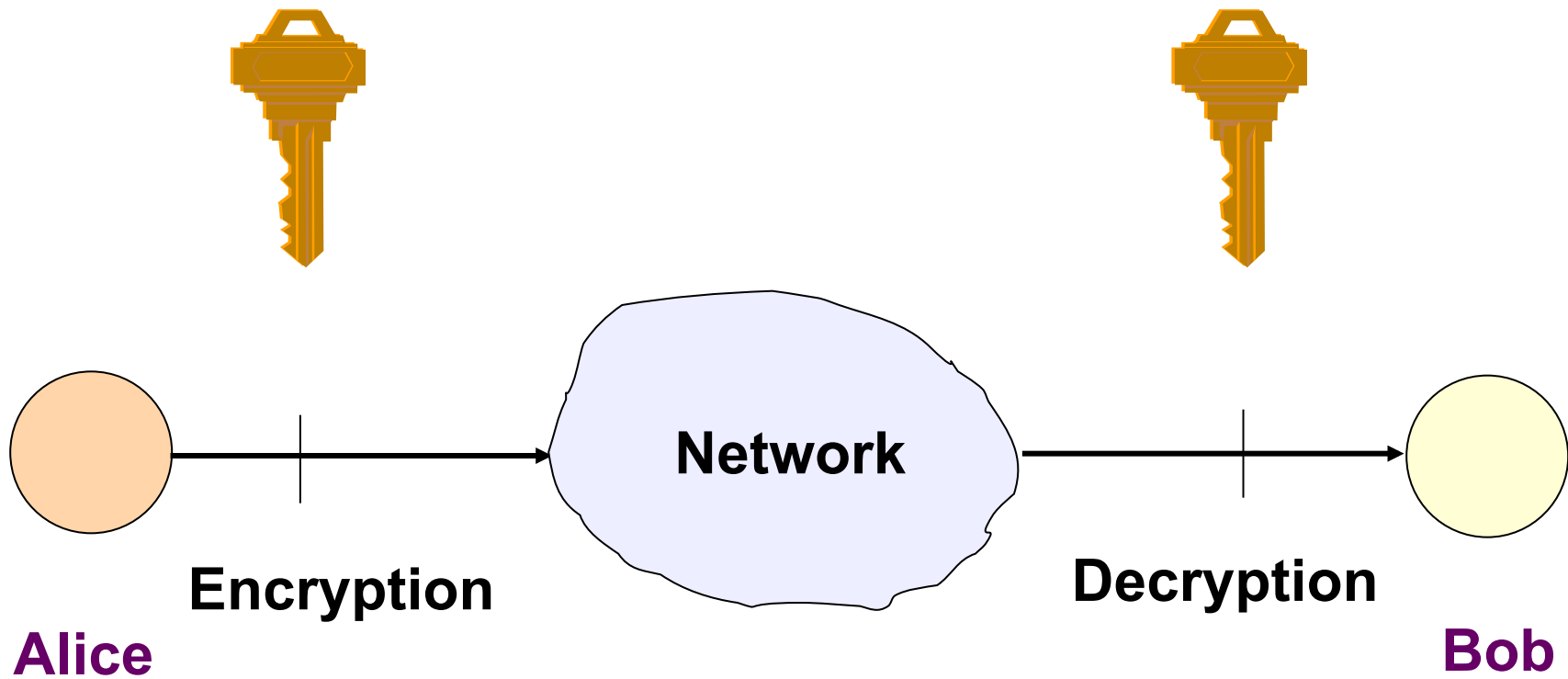
ATHENa Has a Potential to Serve

- **Researchers** – fair, automated, and comprehensive comparison of new algorithms, architectures, and implementations with **previous work**
- **Designers** – informed **choice of technology** (FPGA, ASIC, microprocessor) and a specific **device/library** within a given technology
- **Developers of Tools** – comprehensive **comparison across various tools; optimization methodologies** developed and comprehensively tested as a part of this project
- **Standardization Organizations** (such as NIST) – evaluation of **existing and emerging standards; support of contests** for new standards.

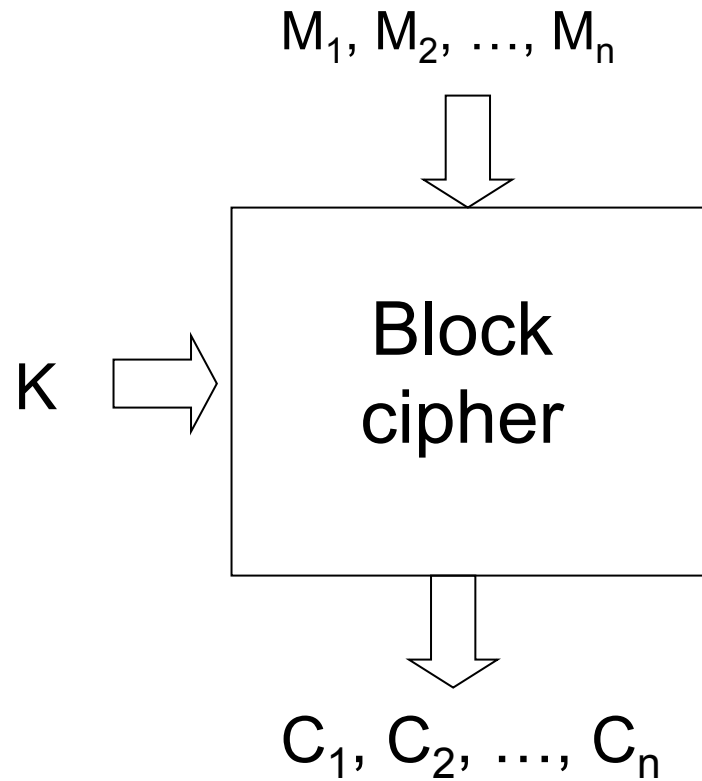
Secret-Key Ciphers

key of Alice and Bob - K_{AB}

key of Alice and Bob - K_{AB}

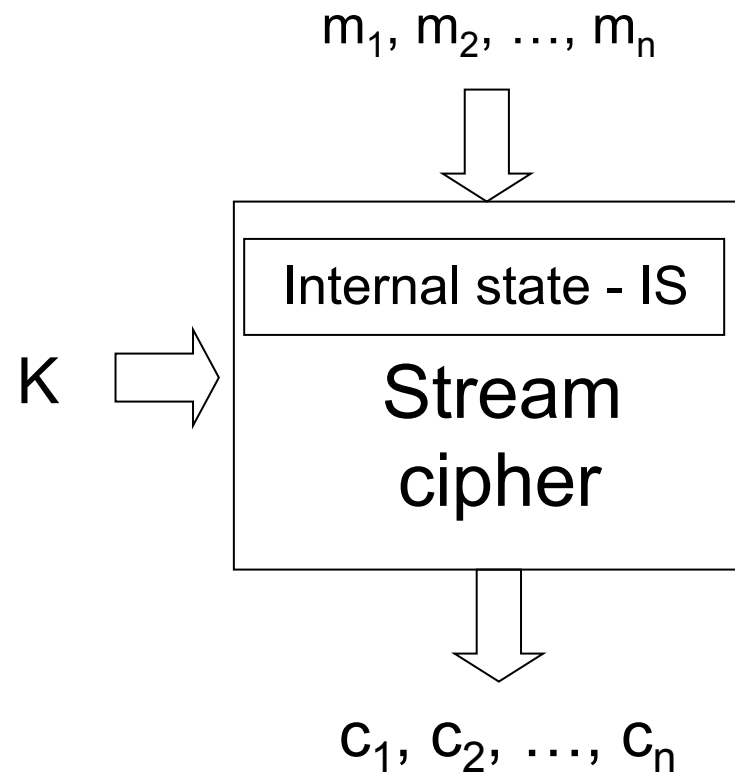


Block vs. Stream Ciphers



$$C_i = f_K(M_i)$$

Every block of ciphertext is a function of only **one** corresponding **block** of plaintext



$$c_i = f_K(m_i, IS_i) \quad IS_{i+1} = g_K(m_i, IS_i)$$

Every block of ciphertext is a function of the **current block** of plaintext and the current **internal state** of the cipher

Hash Function as a Swiss Knife of Cryptography

- storing passwords
- antivirus software
- key update and derivation
- message authentication
- user authentication
- RFID tag security protocols
- etc.



Software or Hardware?

Software

Hardware

security of data
during transmission

low cost

flexibility

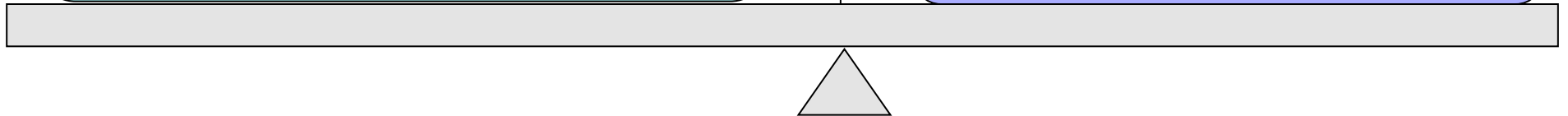
*(new cryptoalgorithms,
protection against new attacks)*

speed

**random key
generation**

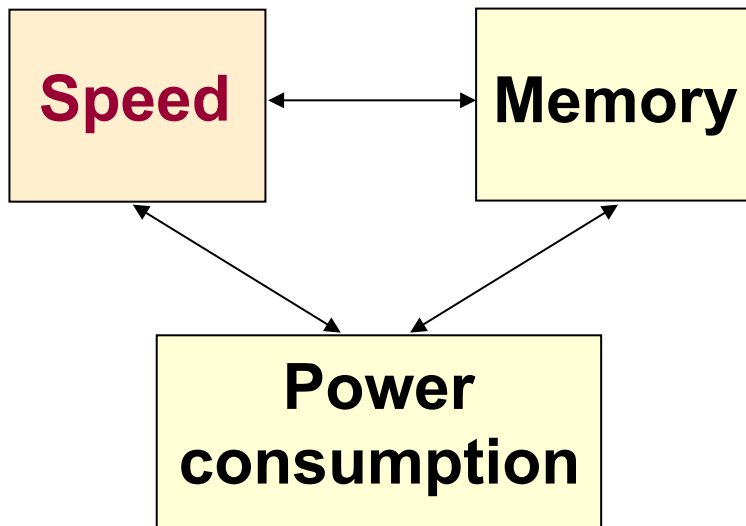
**access control
to keys**

tamper resistance
(viruses, internal attacks)

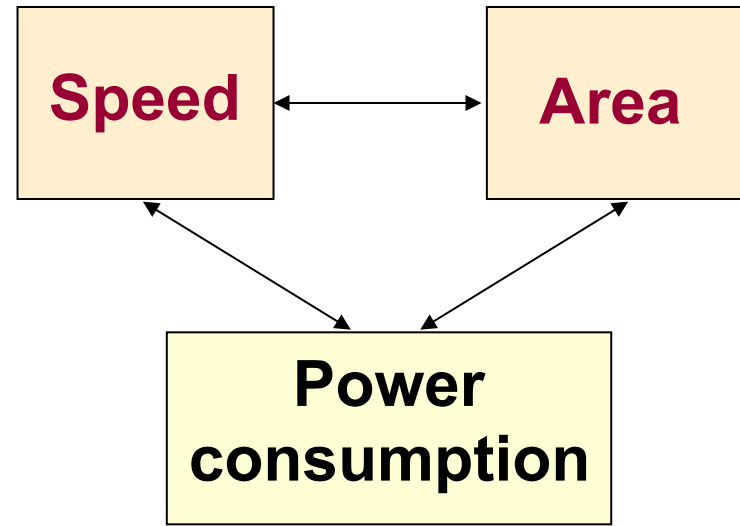


Primary Efficiency Indicators

Software



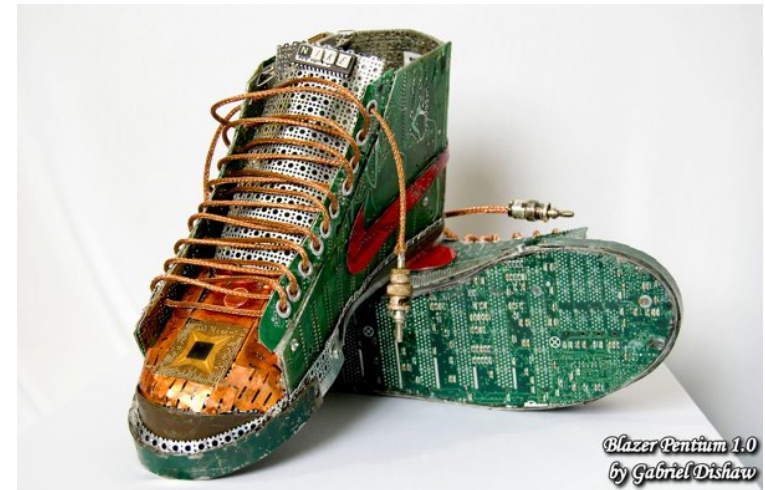
Hardware



Efficiency in Software

Strong dependence on:

1. Instruction set architecture
(e.g., variable rotations)
2. Programming language
(assembler, C, Java)
3. Compiler
4. Compiler options
5. Programming style







Hardware Efficiency in FPGAs

Xilinx Spartan 3, GMU SASC 2008

Candidate	Area (slices)	Candidate	Throughput/Area (Mbps/slices)
Grain v1	44	Trivium (x64)	39.26
Grain 128	50	Grain 128 (x32)	7.97
Trivium	50	Grain v1 (x16)	5.98
DECIM v2	80	Trivium	4.80
DECIM 128	89	F-FCSR-16	4.53
MICKEY 2.0	115	Grain v1	4.45
MICKEY 128 2.0	176	Grain 128	3.92
Moustique	278	F-FCSR-H v2	3.23
F-FCSR-H v2	342	MICKEY 2.0	2.03
Trivium (x64)	344	MICKEY 128 2.0	1.27
Grain v1 (x16)	348	Moustique	0.81
F-FCSR-16	473	DECIM v2	0.58
Grain 128 (x32)	534	DECIM 128	0.49
Pomaranch	648	Edon80	0.10
Edon80	1284	Pomaranch	0.08

Hardware Efficiency in ASICs

	Power-Area-Time Max. clock	Power-Area-Time WLAN	Power-Area-Time RFID/WSN	Flexibility (design space)	Simplicity (code lines)
	Trivium (x64)	Grain80 (x8)	Grain80 (x8)	Trivium	Mickey128
	Grain80 (x16) Grain128 (x32) F-FCSR-H F-FCSR-16	Trivium (x8–x32) F-FCSR-H	Grain128 (x16) Trivium (x8–x32)	Grain128 Grain80	Grain128 Mickey80v2 Grain80 Trivium F-FCSR-H F-FCSR-16
	Mickey80v2 Mickey128 Moustique *	F-FCSR-16 Mickey80v2	F-FCSR-H Mickey80v2 Decim80	Edon80 Decim80 Decim128 Moustique *	Decim128 Decim80 Moustique *
	Decim80 Edon80 Pomaranch80 Decim128 Pomaranch128	Mickey128 Decim80 Pomaranch80 Decim128 Pomaranch128 Moustique * Edon80	Mickey128 Pomaranch80 F-FCSR-16 Moustique * Decim128 Edon80 Pomaranch128	F-FCSR-H F-FCSR-16 Mickey80v2 Mickey128 Pomaranch80 Pomaranch128	Pomaranch80 Pomaranch128 Edon80

* Moustique is the only self synchronising stream cipher so should be considered of significant merit irrespective of other performance metrics.

ASIC Evaluations

- **Two major projects**

- T. Good, M. Benaissa, University of Sheffield, UK
(Phases 1-3) – 0.13 μ m CMOS

eSCARGO 

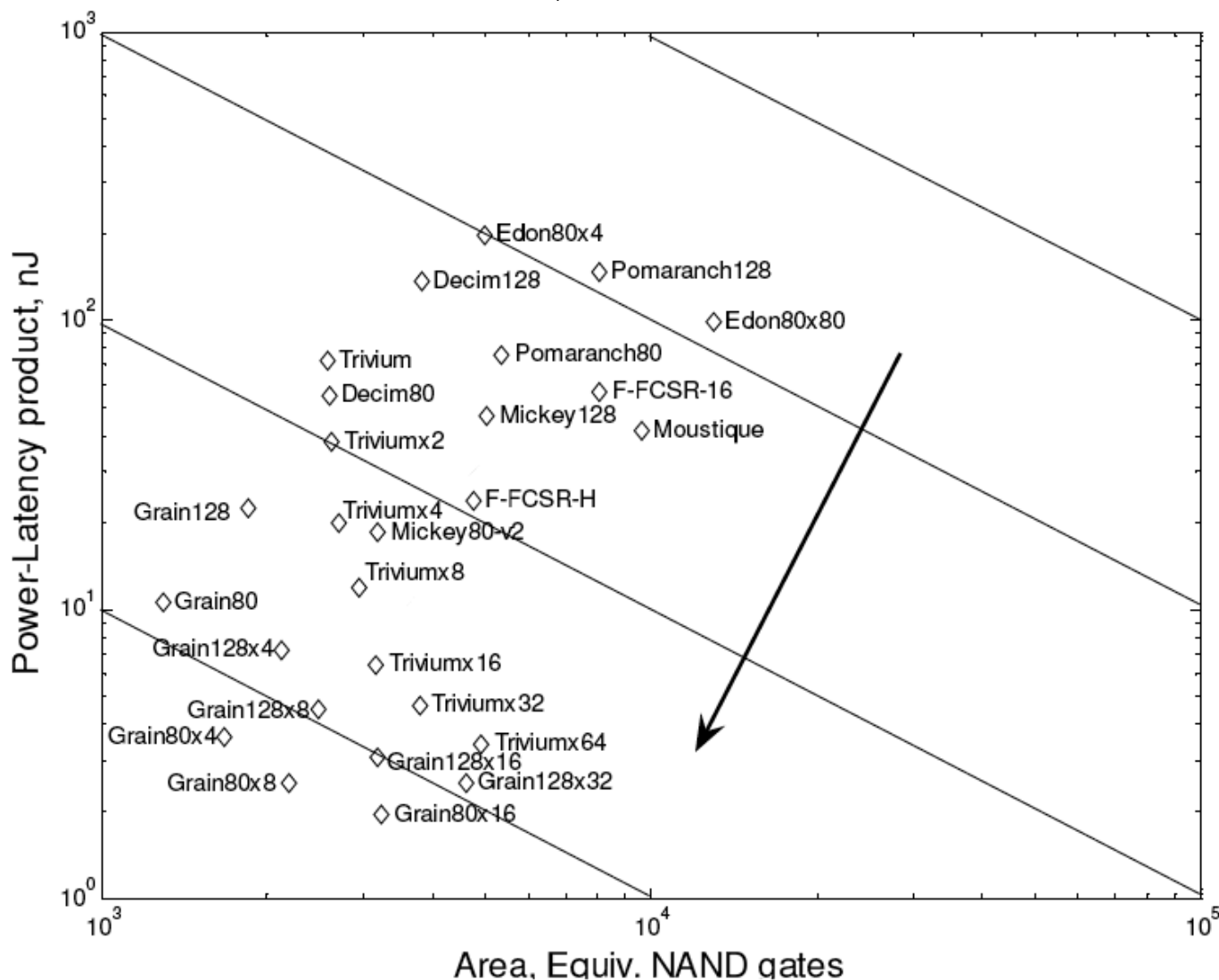
- F.K. Gürkaynak, et al., ETH Zurich, Switzerland
(Phase 1) - 0.25 μ m CMOS

- **Two representative applications**

- **WLAN @ 10 Mbits/s**
- **RFID / WSN @ 100 kHz clock**

Hardware Efficiency in ASICs

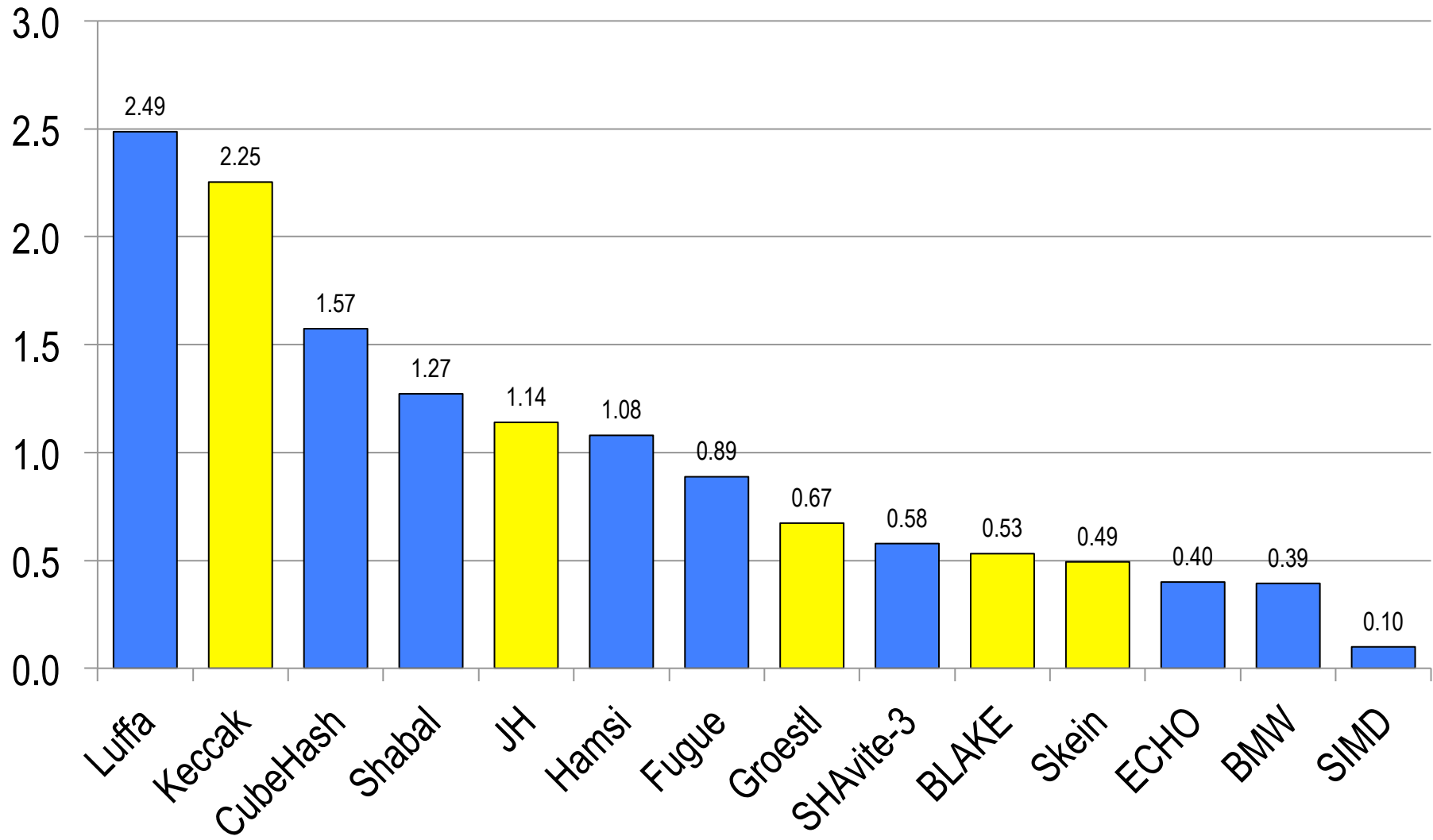
RFID/WSN, 100kHz clock



T. Good and M. Benaissa, Hardware Performance of eSTREAM
Phase III Stream Cipher Candidates, SASC 2008, Lausanne, Feb. 2008

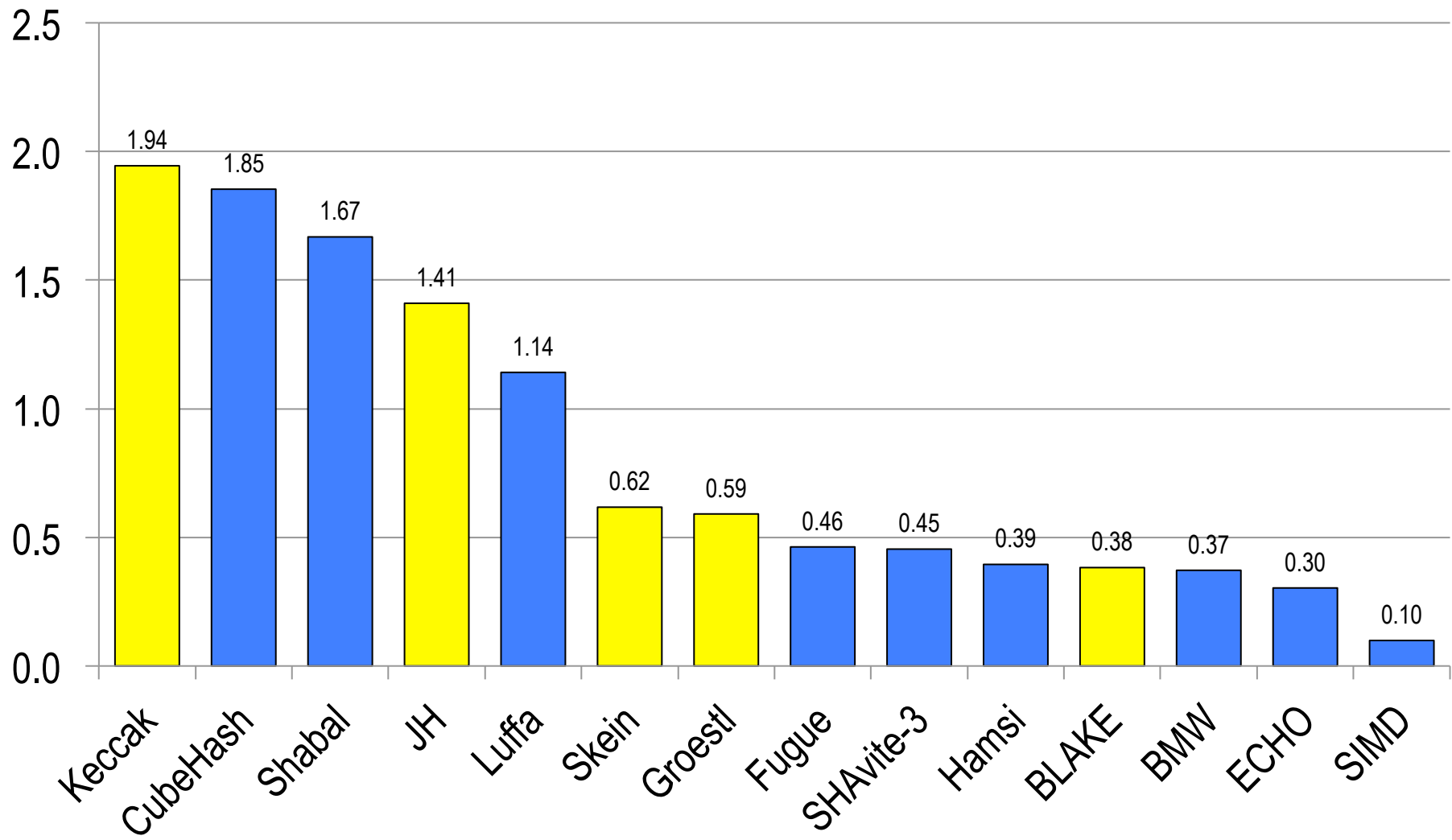
Overall Normalized Throughput/Area: 256-bit variants

Normalized to SHA-256, Averaged over 10 FPGA families



Overall Normalized Throughput/Area: 512-bit variants

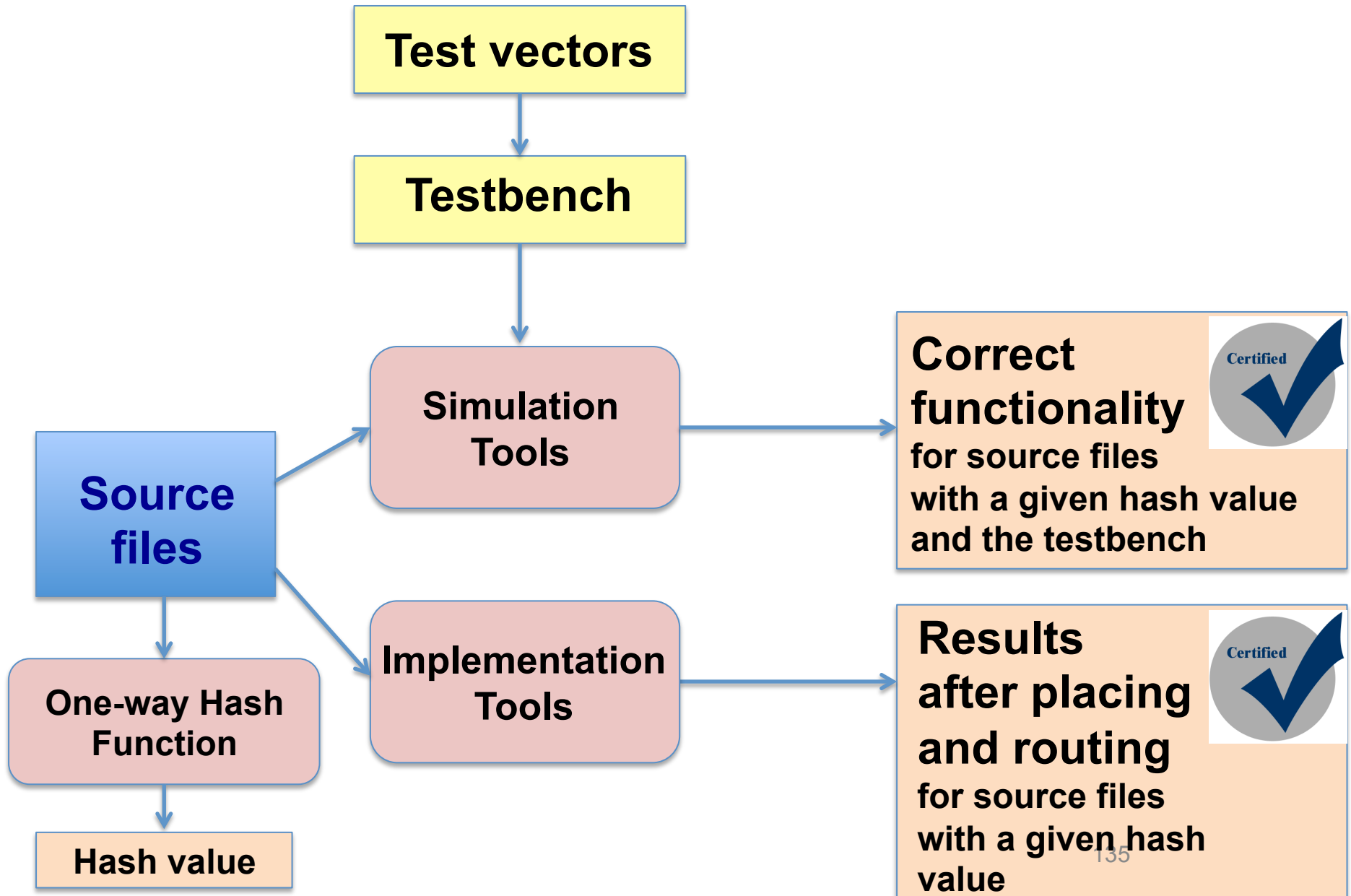
Normalized to SHA-512, Averaged over 10 FPGA families



Round 3 Evaluations



Tool Certificates



Normalization & Compression of Results

Absolute result

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



Normalized result

$$\textit{normalized_result} = \frac{\textit{result_for_SHA-3_candidate}}{\textit{result_for_SHA-2}}$$



Overall normalized result

Geometric mean of normalized results for
all investigated FPGA families

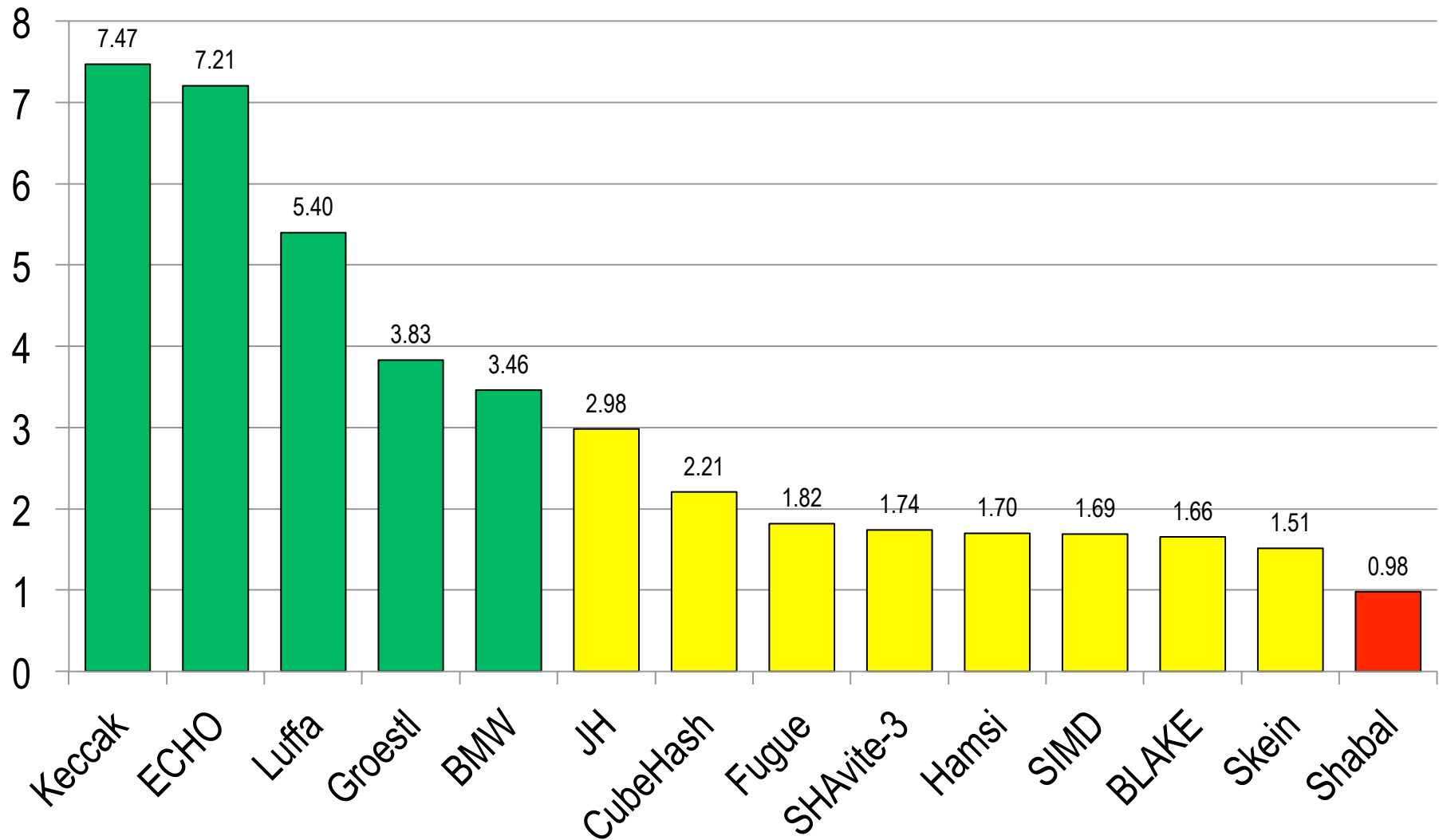
SHA-3 Round 2: Normalized Throughput & Overall Normalized Throughput


Candidate	Spartan 3	Virtex 4	Virtex 5	Virtex 6	Cyclone II	Cyclone III	Cyclone IV	Stratix II	Stratix III	Stratix IV	Overall
Keccak	7.78	8.31	8.03	5.71	7.02	8.09	7.86	7.34	7.33	7.59	7.47
ECHO	5.79	7.88	7.51	6.47	N/A	7.45	6.63	7.59	7.71	8.18	7.21
Luffa	6.09	6.6	5.68	4.05	5.65	6.06	5.95	4.70	4.58	5.19	5.40
Groestl	3.85	4.07	4.68	3.92	3.62	3.37	3.74	3.62	3.73	3.85	3.83
BMW	N/A	3.59	2.68	3.23	N/A	3.71	3.6	4.03	3.42	3.62	3.46
JH	3.01	3.27	3.43	2.47	2.85	3.42	3.28	3.21	2.90	2.24	2.98
CubeHash	2.21	2.51	2.37	2.19	2.08	2.2	2.13	2.20	1.99	2.24	2.21
Fugue	1.95	1.72	2.04	1.91	1.72	1.77	1.79	1.75	1.73	1.82	1.82
SHAvite-3	1.71	1.62	2.24	1.69	1.58	1.66	1.64	1.76	1.76	1.83	1.74
Hamsi	1.78	1.71	1.79	1.28	1.90	1.88	1.83	1.61	1.63	1.68	1.70
SIMD	N/A	1.83	1.86	1.51	1.54	1.57	1.55	1.84	1.68	1.89	1.69
BLAKE	1.69	1.59	1.90	1.45	1.53	1.62	1.57	1.88	1.58	1.82	1.66
Skein	1.52	1.49	1.71	1.66	1.43	1.58	1.53	1.50	1.29	1.47	1.51
Shabal	1.18	1.19	1.03	1.06	0.89	0.92	0.92	0.86	0.86	0.96	0.98

**Overall = Geometric mean of
normalized results
for 10 investigated FPGA families**

Overall Normalized Throughput: 256-bit variants of algorithms

Normalized to SHA-256, Averaged over 10 FPGA families





**SHA-3
Lightweight
Implementations**

Study of Lightweight Implementations in FPGAs

- **Two major projects**
 - J.-P. Kaps, et al., George Mason University, USA
 - F.-X. Standaert, UCL Crypto Group, Belgium
- **Target:**
 - Low-cost FPGAs (Spartan 3, Spartan 6, etc.)
for stand-alone implementations
 - High-performance FPGAs (e.g., Virtex 6)
for system-on-chip implementations

Typical Assumptions – GMU Group

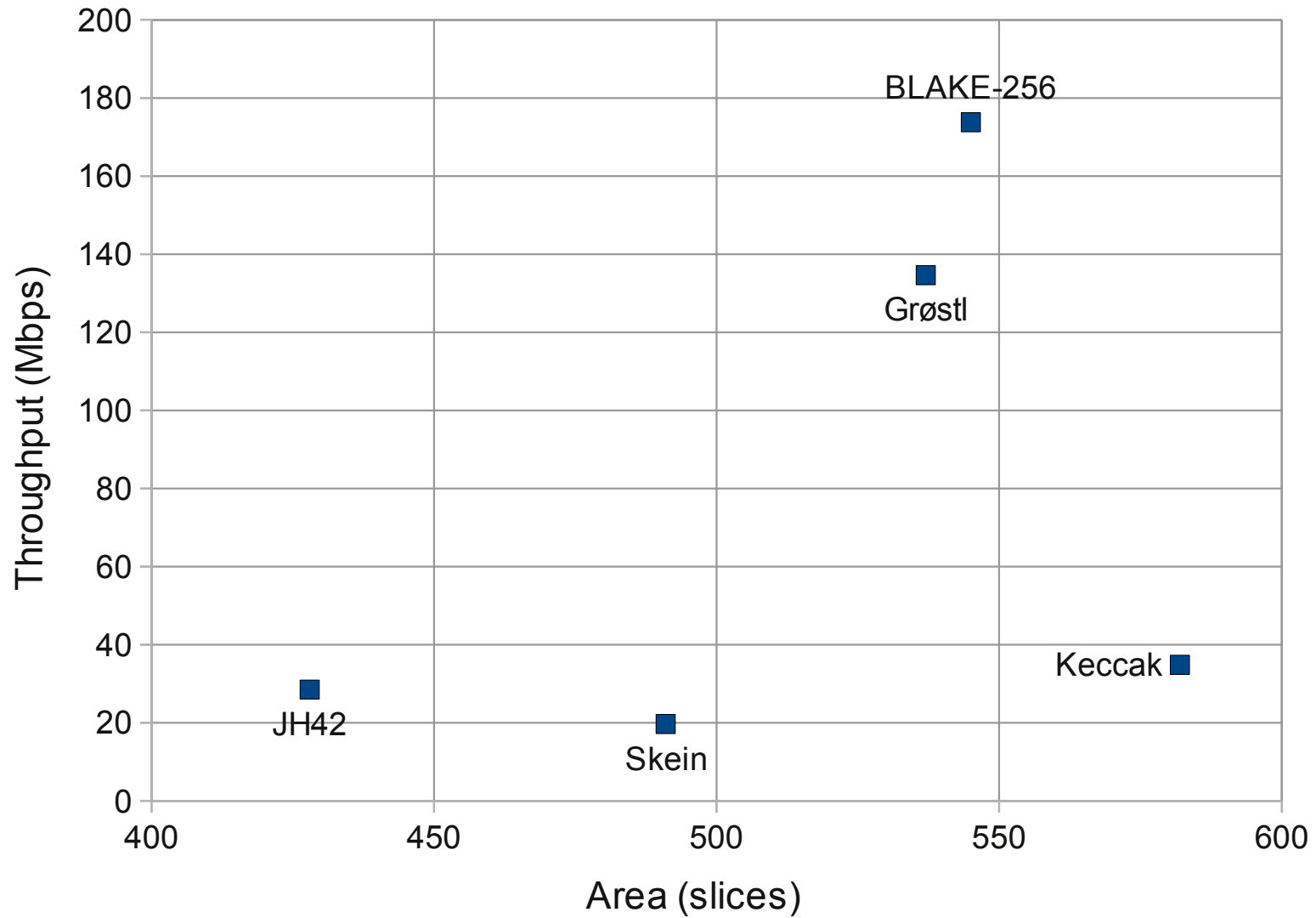
Assumptions

- Implementing for minimum area alone can lead to unrealistic run-times.
- ⇒ Goal: Achieve the maximum Throughput/Area ratio for a given area budget.
- Realistic scenario:
 - System on Chip: Certain area only available.
 - Standalone: Smaller Chip, lower cost, but limit to smallest chip available, e.g. 768 slices on smallest Spartan 3 FPGA.

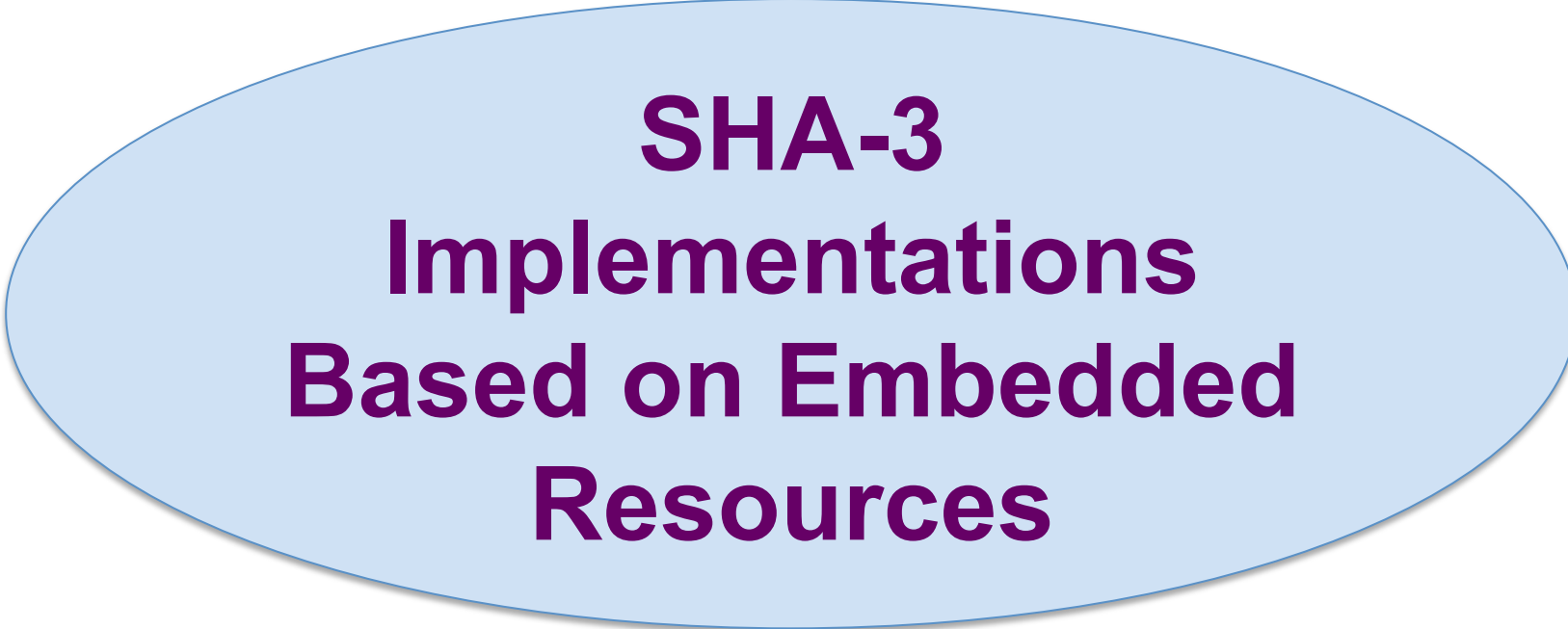
Target

- Xilinx Spartan 3 low cost FPGA family
- Budget: 500 slices, 1 Block RAM (BRAM)

Implementation Results

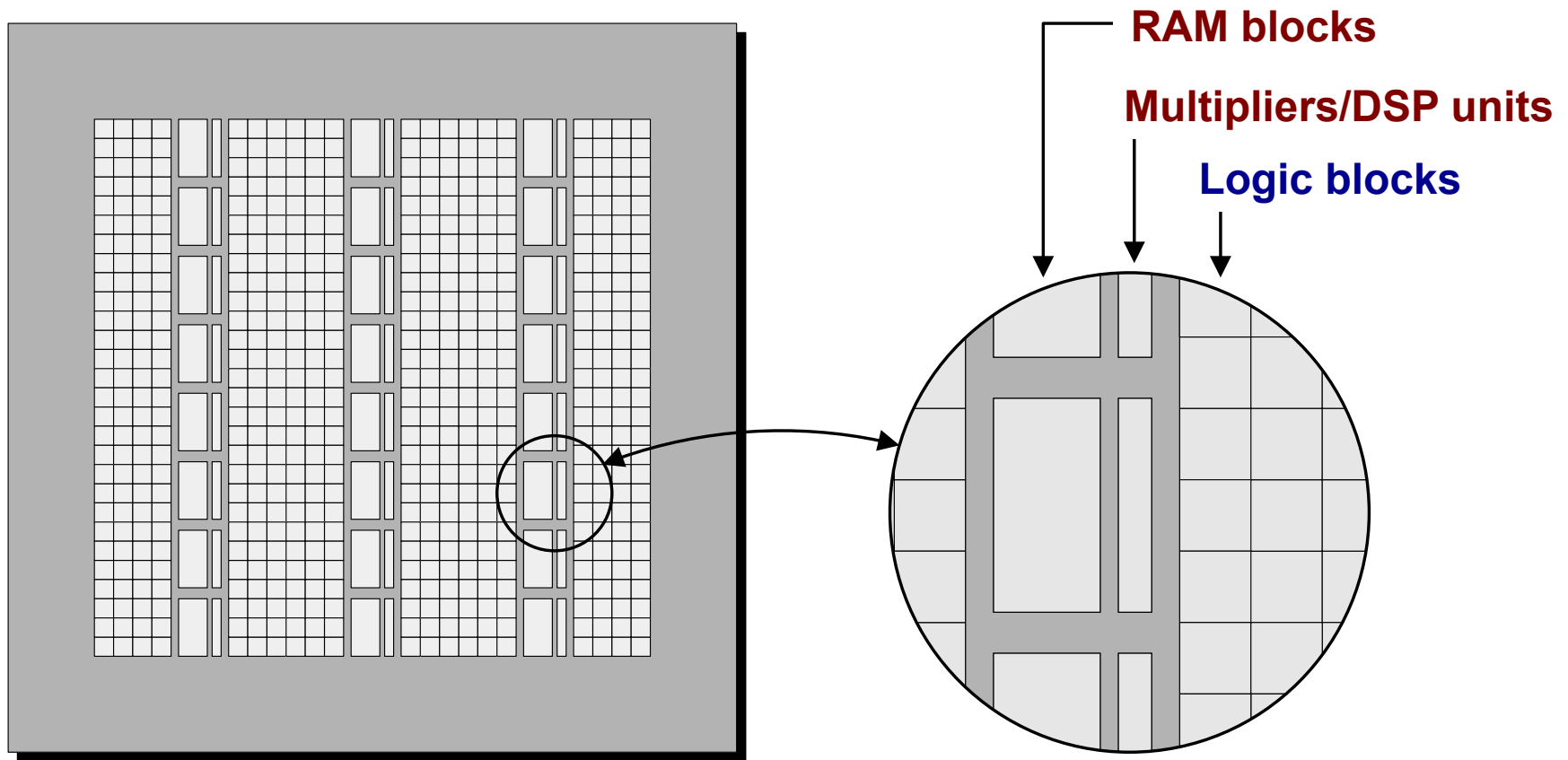


- Xilinx Spartan 3, ISE 12.3, after P&R, Optimized using ATHENa



**SHA-3
Implementations
Based on Embedded
Resources**

Implementations Based on the Use of Embedded Resources in FPGAs



(#Logic blocks, #Multipliers/DSP units, #RAM_blocks)

Resource Utilization Vector

(#Logic blocks, #Multipliers/DSP units, #RAM blocks)

Xilinx

Spartan 3: (#CLB_slices, #multipliers, #Block_RAMs)

Virtex 5: (#CLB_slices, #DSP units, #Block_RAMs)

Altera

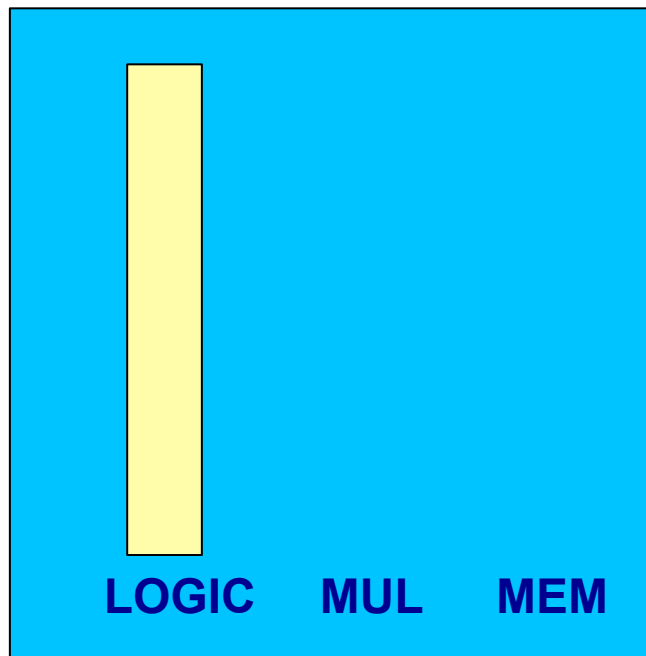
Cyclone III: (#LEs, #multipliers, #RAM_bits)

Stratix III: (#ALUTs, #DSP units, #RAM_bits)

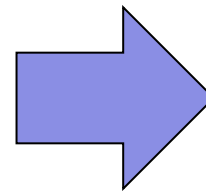
Fitting a Single Core in a Smaller FPGA Device

BLAKE in Altera Cyclone II

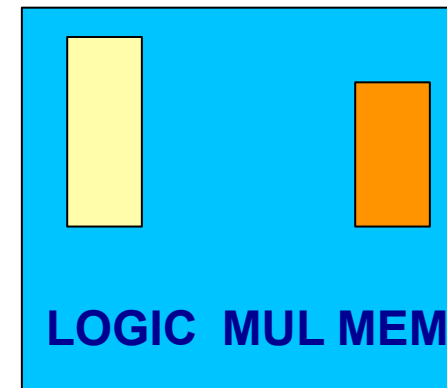
EP2C20



(6862, 0, 0)
LEs, MULs, bits



EP2C5

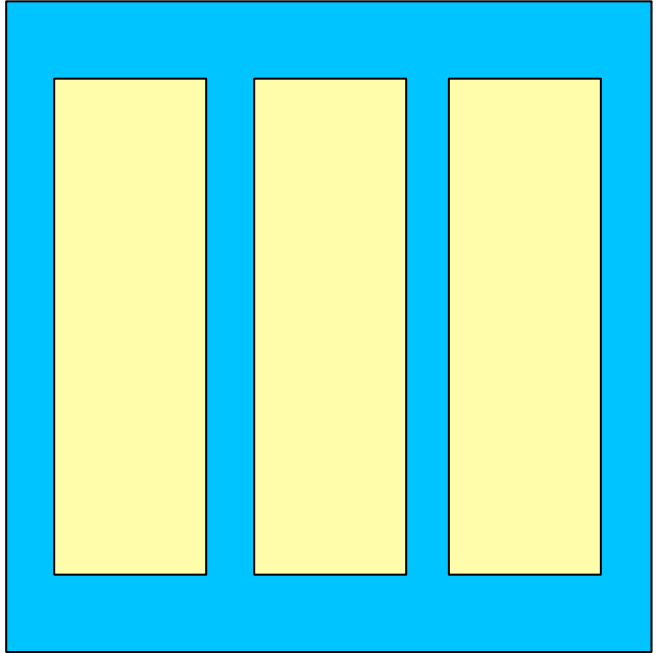


(3129, 0, 12k)
LEs, MULs, bits

Fitting a Larger Number of Identical Cores in the same FPGA Device

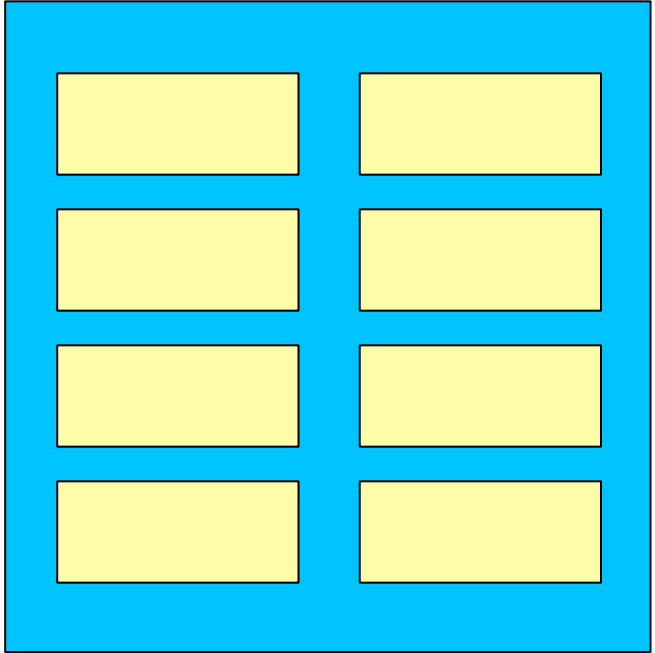
BLAKE in Virtex 5

XC5VSX50

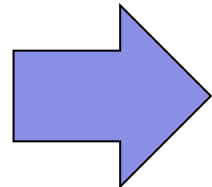


3 BLAKE cores

XC5VSX50



8 BLAKE cores



Cumulative
Throughput

6.8 Gbit/s



20.6 Gbit/s

Cumulative Throughput for the Largest Device of a Given Family

