Fair and Comprehensive Benchmarking of 29 Round 2 CAESAR Candidates in Hardware: Preliminary Results

Ekawat Homsirikamol,
William Diehl, Ahmed Ferozpuri,
Farnoud Farahmand,
and Kris Gaj
George Mason University
USA

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

Register-Transfer Level (RTL) and High-Level Synthesis (HLS) Designs

Ekawat Homsirikamol
a.k.a "Ice"

RTL: AES-GCM, AEZ, Ascon, Deoxys, HS1-SIV, ICEPOLE, Joltik, OCB (8 algorithms)

HLS: 15 algorithms

Working on the PhD Thesis entitled
"A New Approach to the Development of Cryptographic Standards Based on the Use of High-Level Synthesis Tools"

Register-Transfer Level (RTL) Designs provided by

Will
Diehl

OMD
Minalpher
SCREAM
POET

Ahmed
Ferozpuri

PRIMATEs-GIBBON
PRIMATEs-HANUMAN

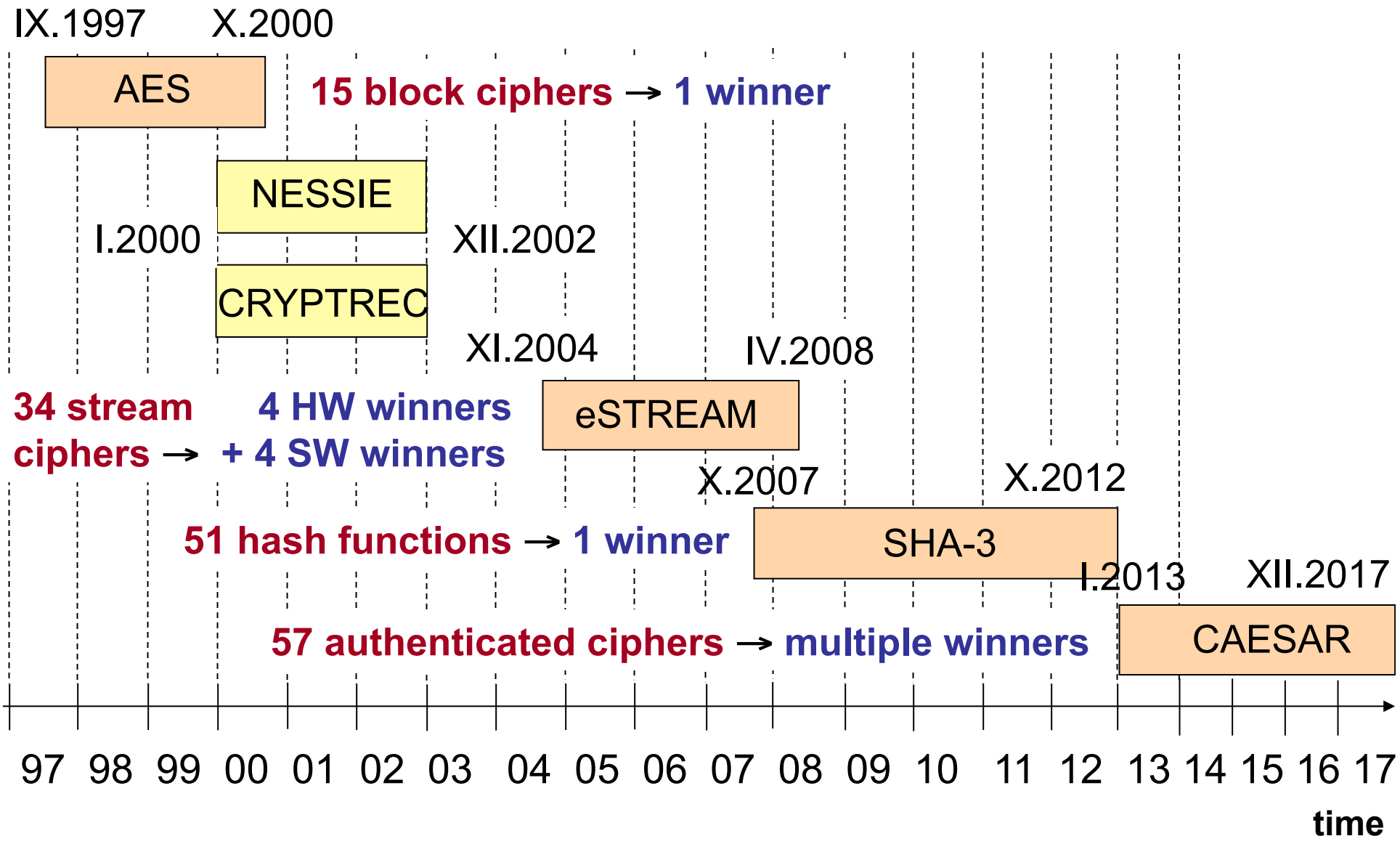
Farnoud
Farahmand

AES-COPA
CLOC

Mike
Lyons

TriviA-ck

Cryptographic Standard Contests



Evaluation Criteria

Security

Software Efficiency

µProcessors µControllers

Hardware Efficiency

FPGAs ASICs

Flexibility

Simplicity

Licensing

Hardware Benchmarking in Previous Contests

AES (1999-2000): 5 final candidates

eSTREAM (2007-2008): 8 Phase-3 candidates

SHA-3 (2010-2012): 14 Round 2 Candidates
+ 5 Final Candidates

CAESAR (2016): 29 Round 2 Candidates

2016.06.30: Deadline for Verilog/VHDL

CAESAR Hardware API

Specifies:

- Minimum compliance criteria
- Interface
- Communication protocol
- Timing characteristics

Assures:

- Compatibility
- Fairness

Timeline:

- Based on the GMU Hardware API presented at CryptArchi 2015, DIAC 2015, and ReConFig 2015
- Revised version posted on Feb. 15, 2016
- Officially approved by the CAESAR Committee on May 6, 2016

GMU Support for Designers of VHDL/Verilog Code

Implementer's Guide

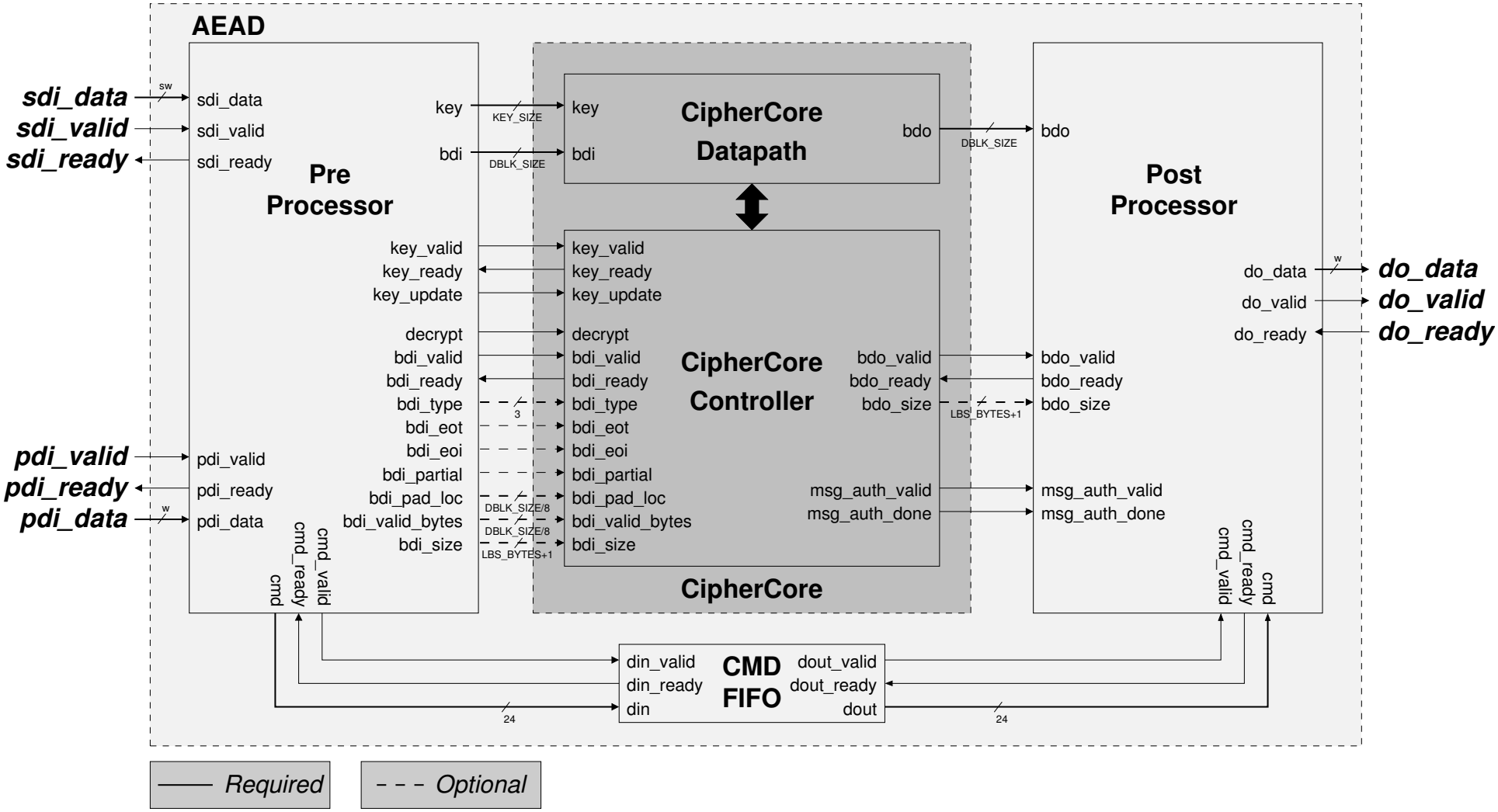
- v1.0 - May 12, 2016

Development Package

a. VHDL code of generic pre-processing and post- processing units for high-speed implementations
b. Universal testbench
c. Python app used to automatically generate test vectors
d. VHDL wrappers used to determine the maximum clock frequency and resource utilization
e. Six reference high-speed implementations of Dummy authenticated ciphers

https://cryptography.gmu.edu/athena/index.php?id=download

Top-level block diagram of a high-speed architecture



GMU Support for Designers of VHDL/Verilog Code

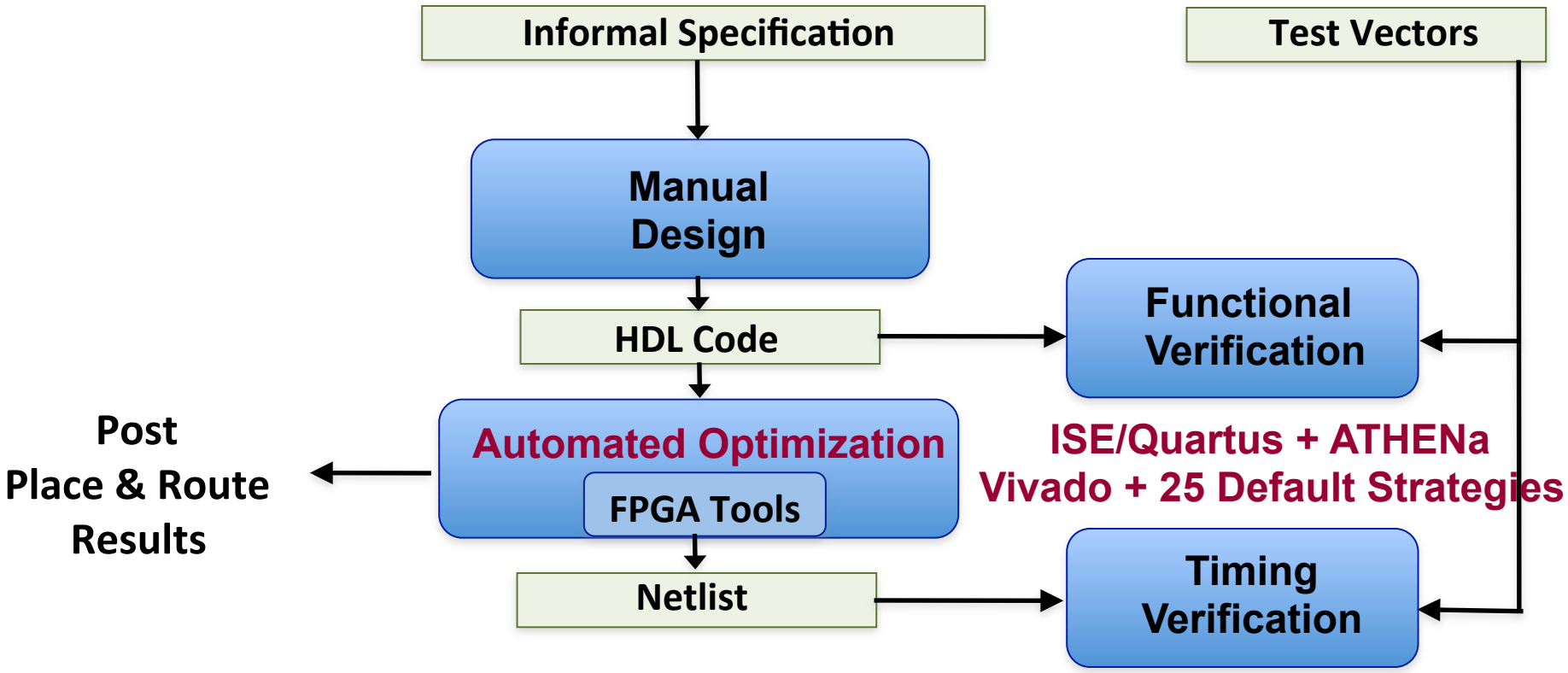
RTL VHDL Code

- AES (Enc/EncDec, 10/11 cycles per block, SubBytes in ROM/logic)
- Keccak Permutation F
- Ascon – example CAESAR candidate

Suggested List of Deliverables

a. VHDL/Verilog code (folder structure)
b. Implemented variants (corresponding generics & constants)
d. Non-standard assumptions
e. Verification method (test vectors)
f. Block diagrams (optional)
g. License (optional)
h. Preliminary results (optional)

RTL Development & Benchmarking Flow



FPGA Families & Devices Used for Benchmarking

High-Speed

- Xilinx Virtex-6: xc6vlx240tff1156-3
- Xilinx Virtex-7: xc7vx485tffg1761-3
- Altera Stratix IV: ep4se530h35c2
- Altera Stratix V: 5sgxea7k2f40c1

Lightweight:

- Xilinx Spartan-6: xc6slx16csg324-3
- Xilinx Artix-7: xc7a100tcsg324-3
- Altera Cyclone IV E: EP4CE22F17C6
- Altera Cyclone V E: 5CEBA4F23C7

RTL Implementations Developed by GMU

CAESAR Candidates:

1. AES-COPA
2. AEZ
3. Ascon
4. CLOC
5. Deoxys
6. HS1-SIV
7. ICEPOLE
8. Joltik
9. Minalpher
10. OCB
11. OMD
12. POET
13. PRIMATEs-HANUMAN
14. PRIMATES-GIBBON
15. SCREAM
16. TriviA-ck

Current Standard:

17. AES-GCM

Parameters of Implemented Authenticated Ciphers

Algorithm	Key size	Nonce size	Tag size	Basic Primitive
		Block Cipher Based		
AES-COPA	128	128	128	AES
AES-GCM	128	96	128	AES
AEZ	384	96	128	AES
CLOC	128	96	128	AES
Deoxys≠	128	64	128	Deoxys-BC (AES)
Joltik	128	32	64	Joltik-BC
Minalpher	128	104	128	TEM
OCB	128	96	128	AES
POET	128	128	128	AES
SCREAM	128	88	128	TLS

Parameters of Implemented Authenticated Ciphers

Algorithm	Key size	Nonce size	Tag size	Basic Primitive
Permutation Based				
ASCON	128	128	128	SPN
ICEPOLE	128	128	128	Keccak-like
PRIMATEs-GIBBON	120	120	120	PRIMATE
PRIMATEs-HANUMAN	120	120	120	PRIMATE
Stream Cipher and/or Hash Function Based				
HS1-SIV	128	96	128	Salsa 20 (Cha-Cha 20)
OMD	128	96	128	SHA-2
TriviA-ck	128	128	128	TriviA-SC VPV-Hash

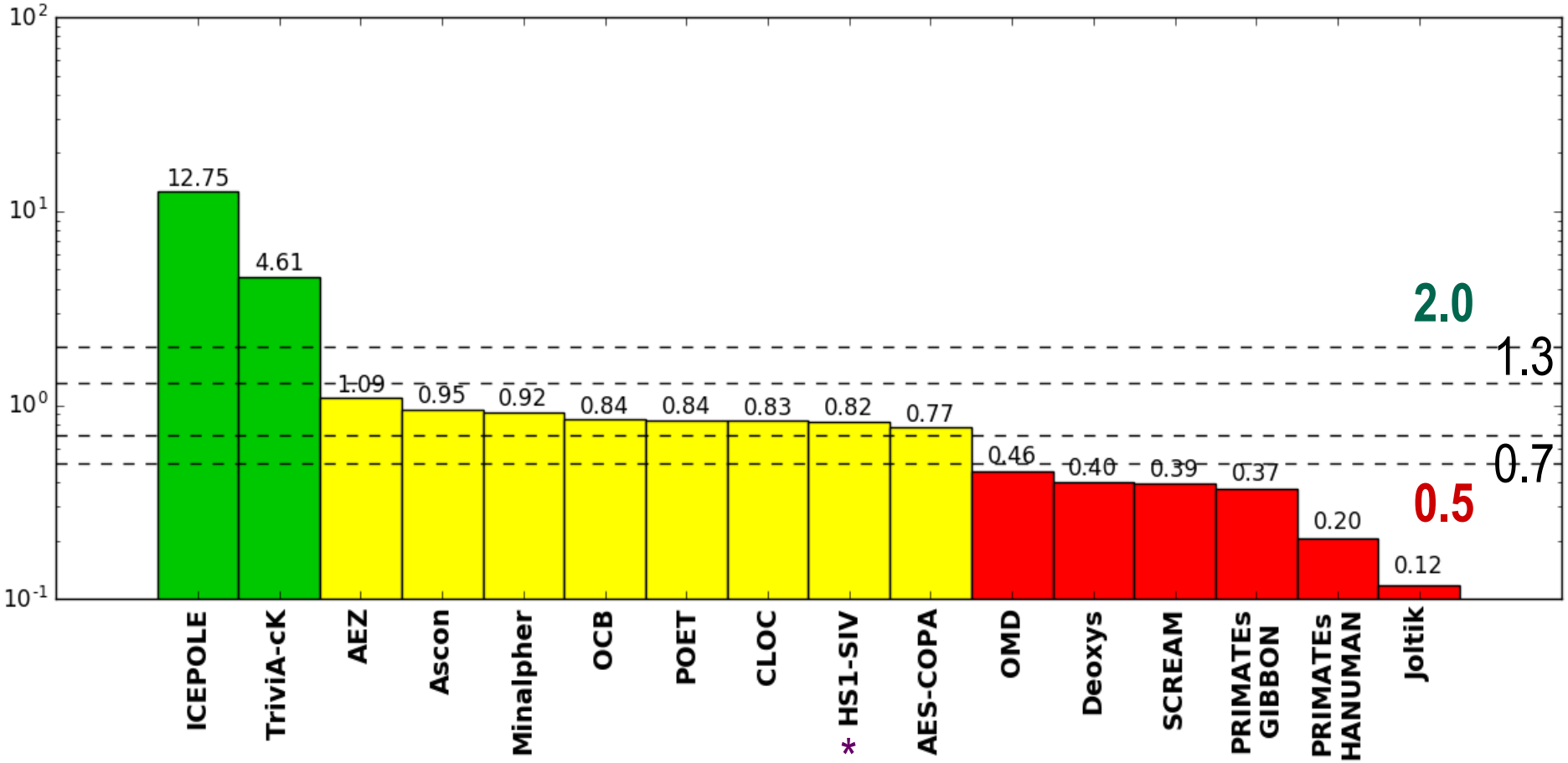
Parameters of Ciphers & GMU Implementations

Algorithm	Word Size, w	Block Size, b	#Rounds	Cycles/Block
Block-cipher Based				
AES-COPA	32	128	10	11
AES-GCM	32	128	10	11
AEZ	64	256	20	25
CLOC	32	128	10	11
Deoxys	32	128	14	29
Joltik	32	128	32	65
Minalpher	32	256	18	19
OCB	32	128	10	12
POET	32	128	10/4	10
SCREAM	32	128	10	11

Parameters of Ciphers & GMU Implementations

Algorithm	Word Size, w	Block Size, b	#Rounds	Cycles/Block
Permutation Based				
ASCON	32	64	6	7
ICEPOLE	256	1024	6	7
PRIMATEs-GIBBON	40	40	6	7
PRIMATEs-HANUMAN	40	40	12	13
Stream Cipher and/or Hash Function Based				
HS1-SIV	128	512	12	41 Enc/25 Dec
OMD	32	256	64	66
TriviA-ck	64	64	1	1

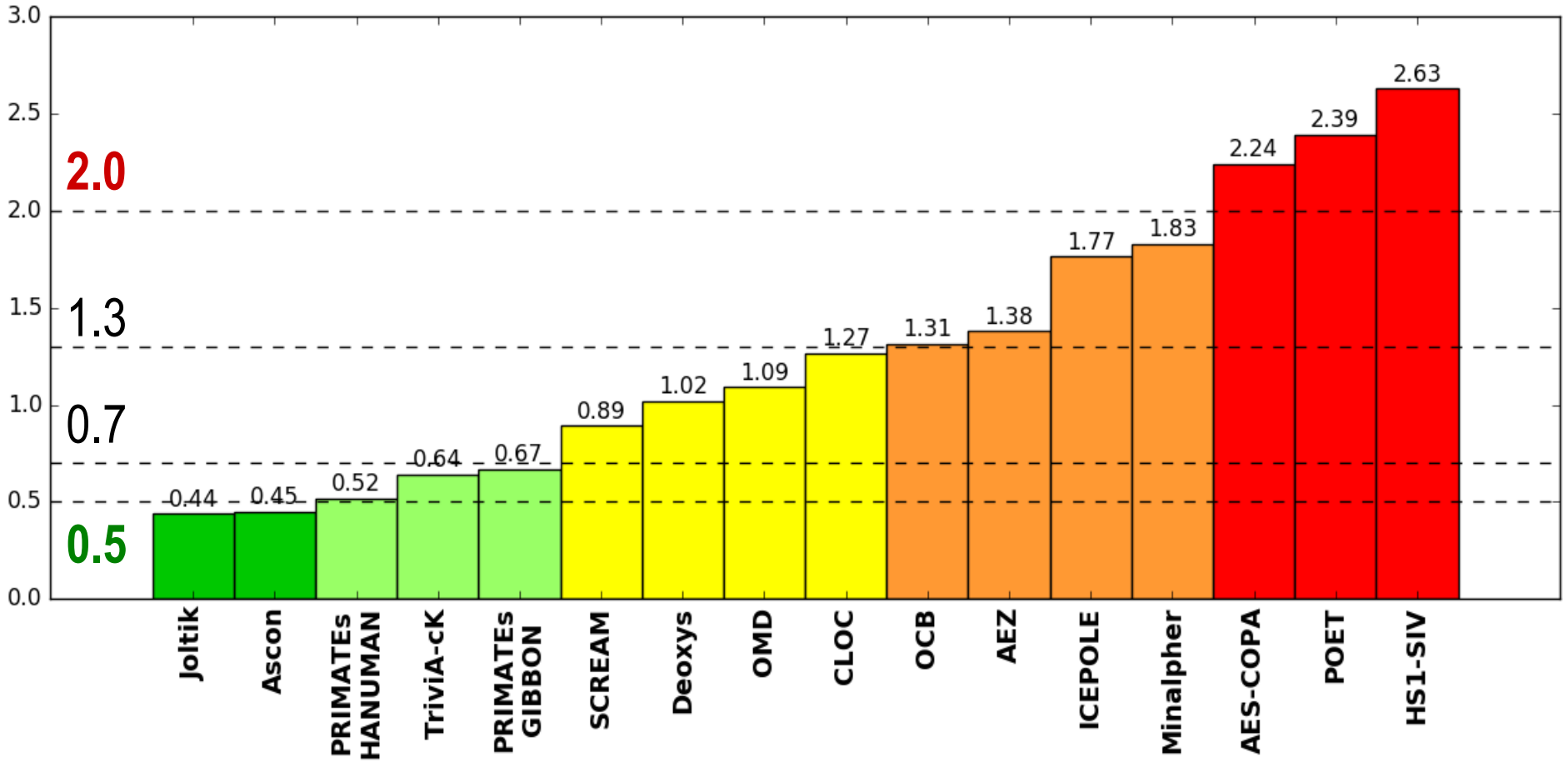
Relative Enc/Dec Throughput in Virtex 7
Ratio of a given Cipher Throughput/Throughput of AES-GCM



Throughput of AES-GCM = 3398 Mbit/s

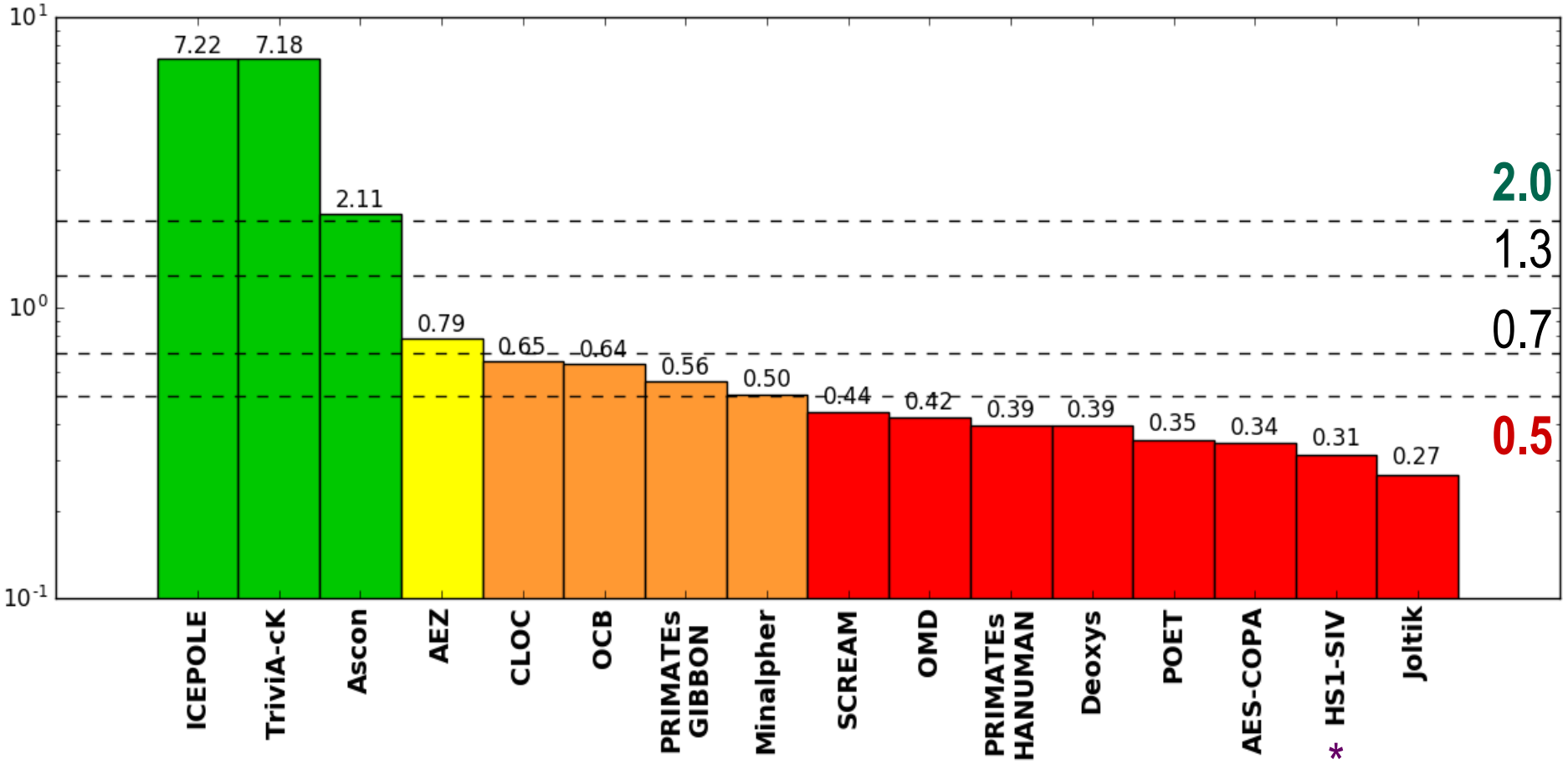
*The HS1-SIV result represents encryption only

Relative Area (#LUTs) in Virtex 7
Ratio of a given Cipher Area/Area of AES-GCM



Area of AES-GCM = 3257 LUTs

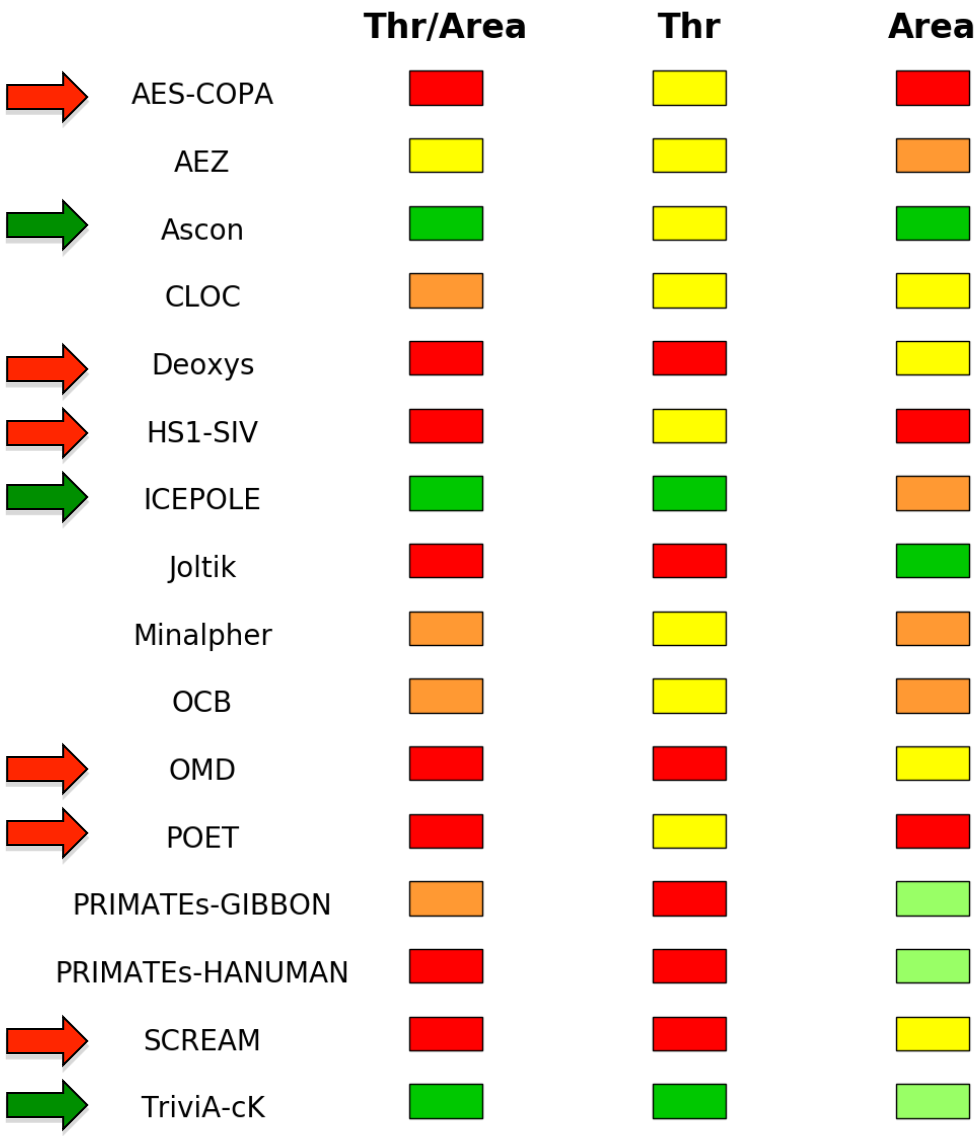
Relative Enc/Dec Throughput/Area in Virtex 7



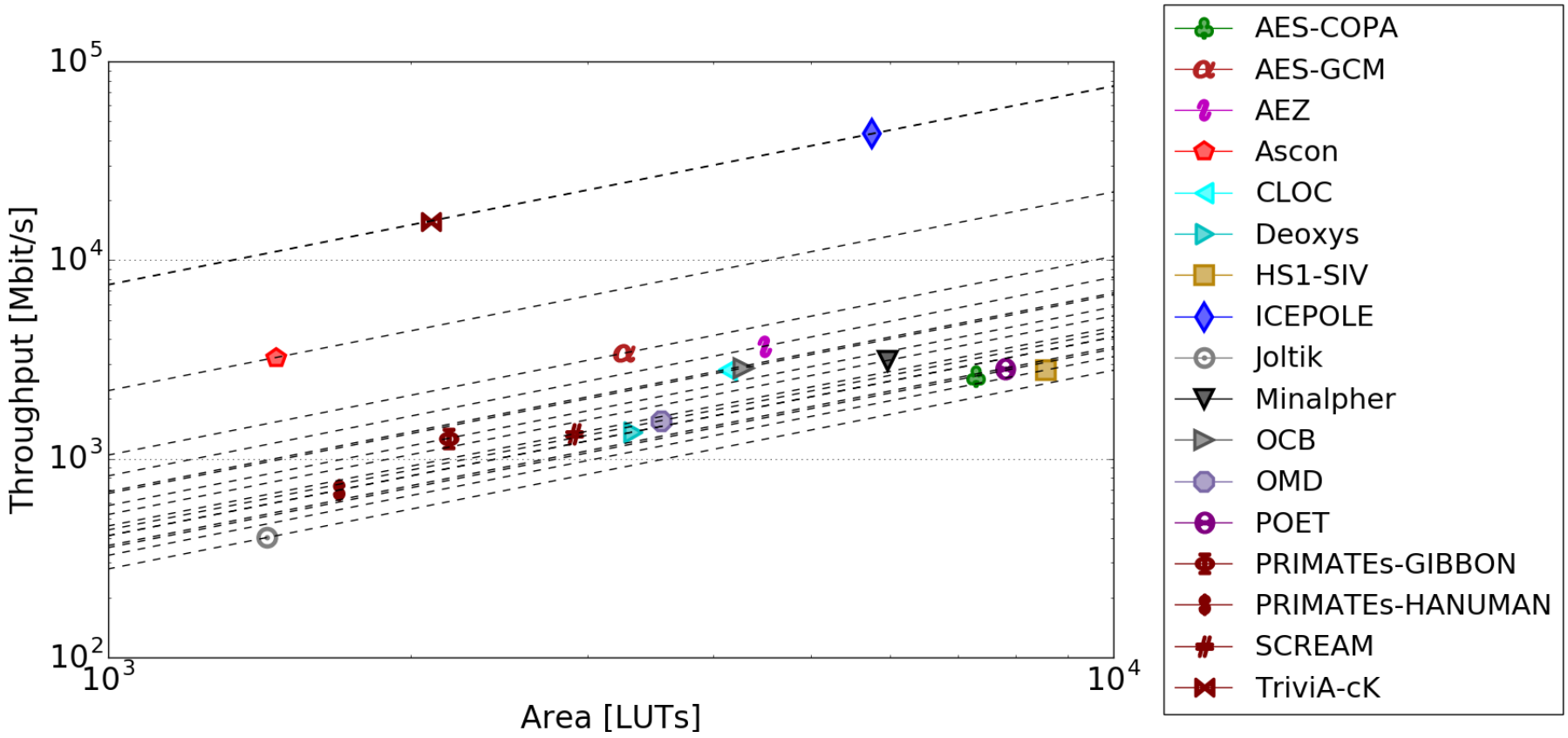
Throughput/Area of AES-GCM = 1.04 (Mbit/s)/LUTs

*The HS1-SIV result represents encryption only

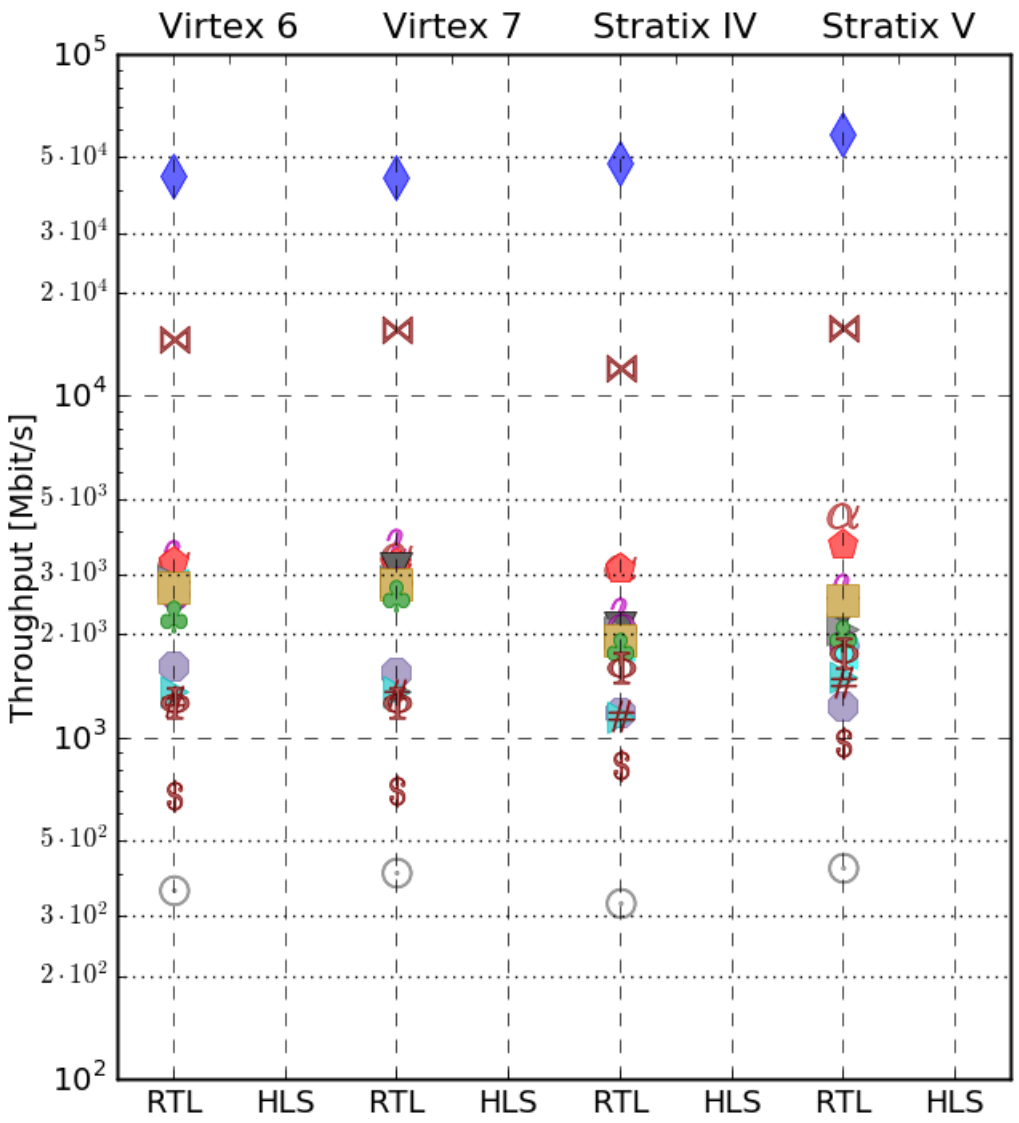
Summary of RTL Results for Virtex 7



RTL Results for Virtex 7 – Throughput vs. Area

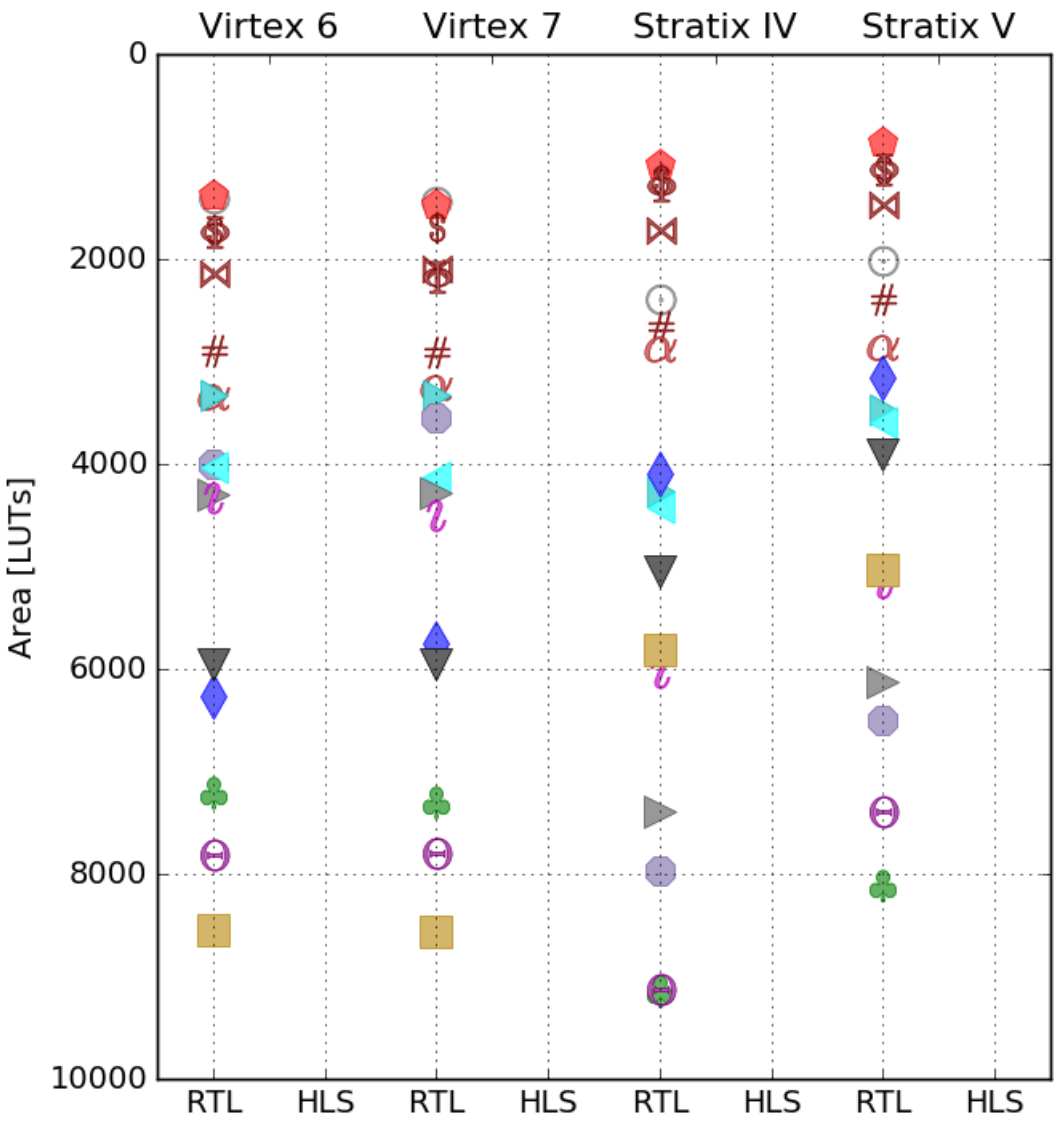


RTL Results – Throughput



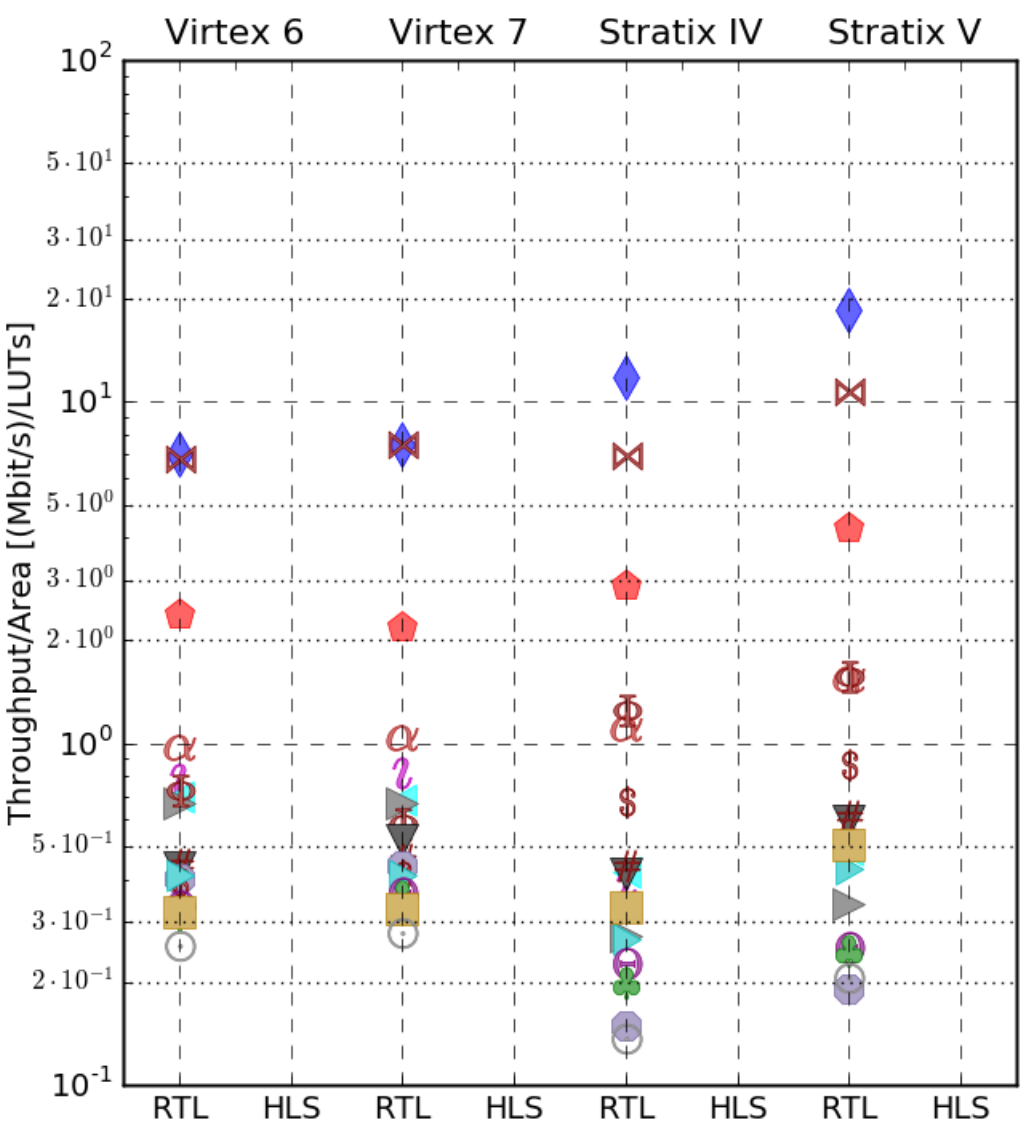
Algorithm	X-V6	X-V7	A-SIV	A-SV
ICEPOLE	(1)	(1)	(1)	(1)
TriviA-cK	(2)	(2)	(2)	(2)
AEZ	(3)	(3)	(5)	(5)
AES-GCM	(5)	(4)	(3)	(3)
Ascon	(4)	(5)	(4)	(4)
Minalpher	(10)	(6)	(6)	(7)
OCB	(6)	(7)	(8)	(8)
POET	(9)	(8)	(7)	(10)
CLOC	(7)	(9)	(10)	(12)
HS1-SIV	(8)	(10)	(9)	(6)
AES-COPA	(11)	(11)	(11)	(9)
OMD	(12)	(12)	(13)	(15)
Deoxys	(13)	(13)	(15)	(13)
SCREAM	(14)	(14)	(14)	(14)
PRIMATEs GIBBON	(15)	(15)	(12)	(11)
PRIMATEs HANUMAN	(16)	(16)	(16)	(16)
Joltik	(17)	(17)	(17)	(17)

RTL Results – Area



Algorithm	X-V6	X-V7	A-SIV	A-SV
Joltik	(2)	(1)	(5)	(5)
Ascon	(1)	(2)	(1)	(1)
PRIMATEs HANUMAN	(3)	(3)	(2)	(2)
TriviA-cK	(5)	(4)	(4)	(4)
PRIMATEs GIBBON	(4)	(5)	(3)	(3)
SCREAM	(6)	(6)	(6)	(6)
AES-GCM	(8)	(7)	(7)	(7)
Deoxys	(7)	(8)	(9)	(9)
OMD	(9)	(9)	(15)	(15)
CLOC	(10)	(10)	(10)	(10)
OCB	(11)	(11)	(14)	(14)
AEZ	(12)	(12)	(13)	(13)
ICEPOLE	(14)	(13)	(8)	(8)
Minalpher	(13)	(14)	(11)	(11)
AES-COPA	(15)	(15)	(17)	(17)
POET	(16)	(16)	(16)	(16)
HS1-SIV	(17)	(17)	(12)	(12)

RTL Results – Throughput/Area

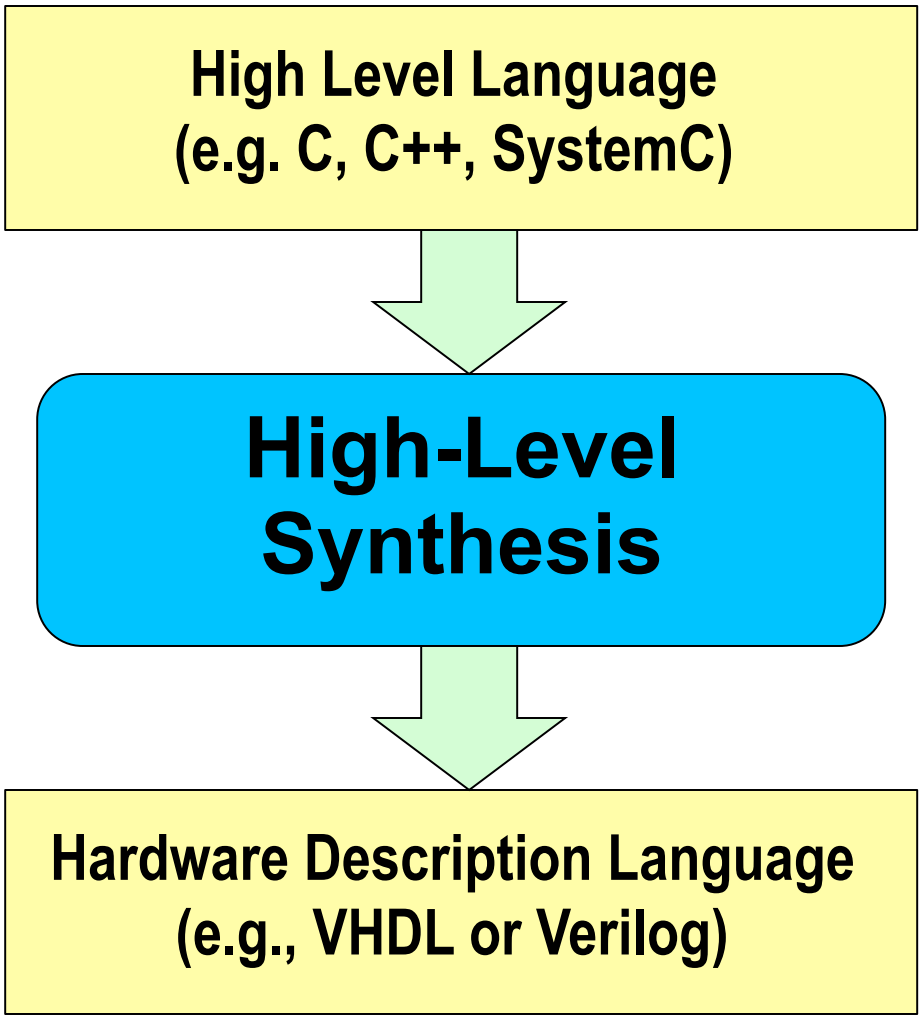


Algorithm	X-V6	X-V7	A-SIV	A-SV
ICEPOLE	(1)	(1)	(1)	(1)
TriviA-cK	(2)	(2)	(2)	(2)
Ascon	(3)	(3)	(3)	(3)
AES-GCM	(4)	(4)	(5)	(5)
AEZ	(5)	(5)	(10)	(9)
CLOC	(7)	(6)	(8)	(11)
OCB	(8)	(7)	(12)	(13)
PRIMATEs GIBBON	(6)	(8)	(4)	(4)
Minalpher	(10)	(9)	(9)	(8)
SCREAM	(9)	(10)	(7)	(7)
OMD	(12)	(11)	(16)	(17)
PRIMATEs HANUMAN	(13)	(12)	(6)	(6)
Deoxys	(11)	(13)	(13)	(12)
POET	(14)	(14)	(14)	(14)
AES-COPA	(16)	(15)	(15)	(15)
HS1-SIV	(15)	(16)	(11)	(10)
Joltik	(17)	(17)	(17)	(16)

Remaining Difficulties of Hardware Benchmarking

- Long time necessary to develop and verify
  RTL (Register-Transfer Level)
  Hardware Description Language (HDL) codes
- Multiple variants of algorithms
  (e.g., multiple key, nonce, and tag sizes)
- Multiple hardware architectures
- Dependence on skills of designers

High-Level Synthesis (HLS)



Selected Tool: Xilinx Vivado HLS

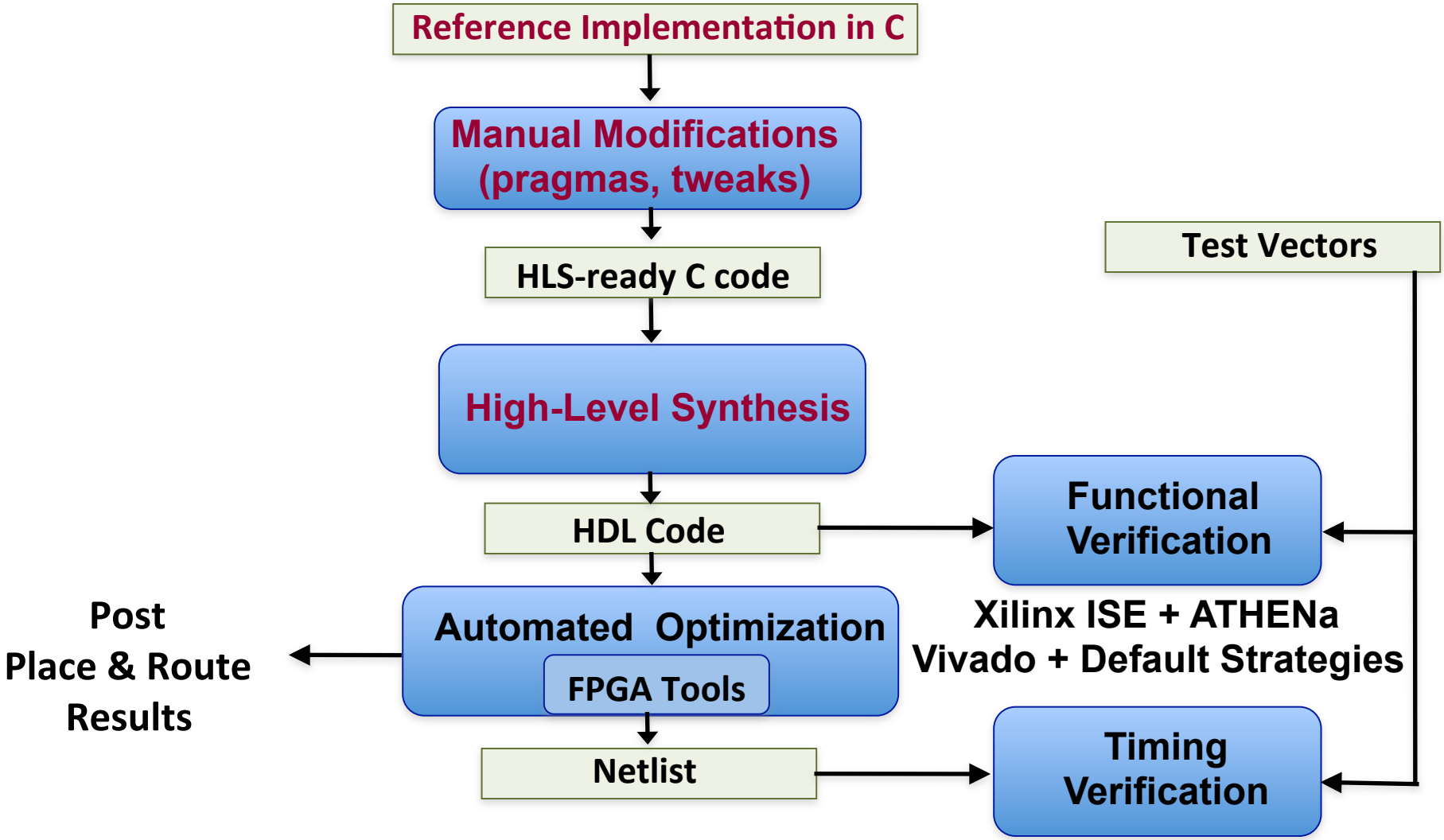
- **Design and verification orders of magnitude faster** than at the RTL level (HLL testbench)
- Support for **C/C++/SystemC**
- Educational licenses and trial versions = **low cost**
- Regular releases and **constant improvement**

Our Hypotheses

- Ranking of candidate algorithms in cryptographic contests in terms of their performance in modern FPGAs & All-Programmable SoCs will remain the same independently whether the HDL implementations are *developed manually* or *generated automatically* using High-Level Synthesis tools

- The development time will be reduced by at least an order of magnitude

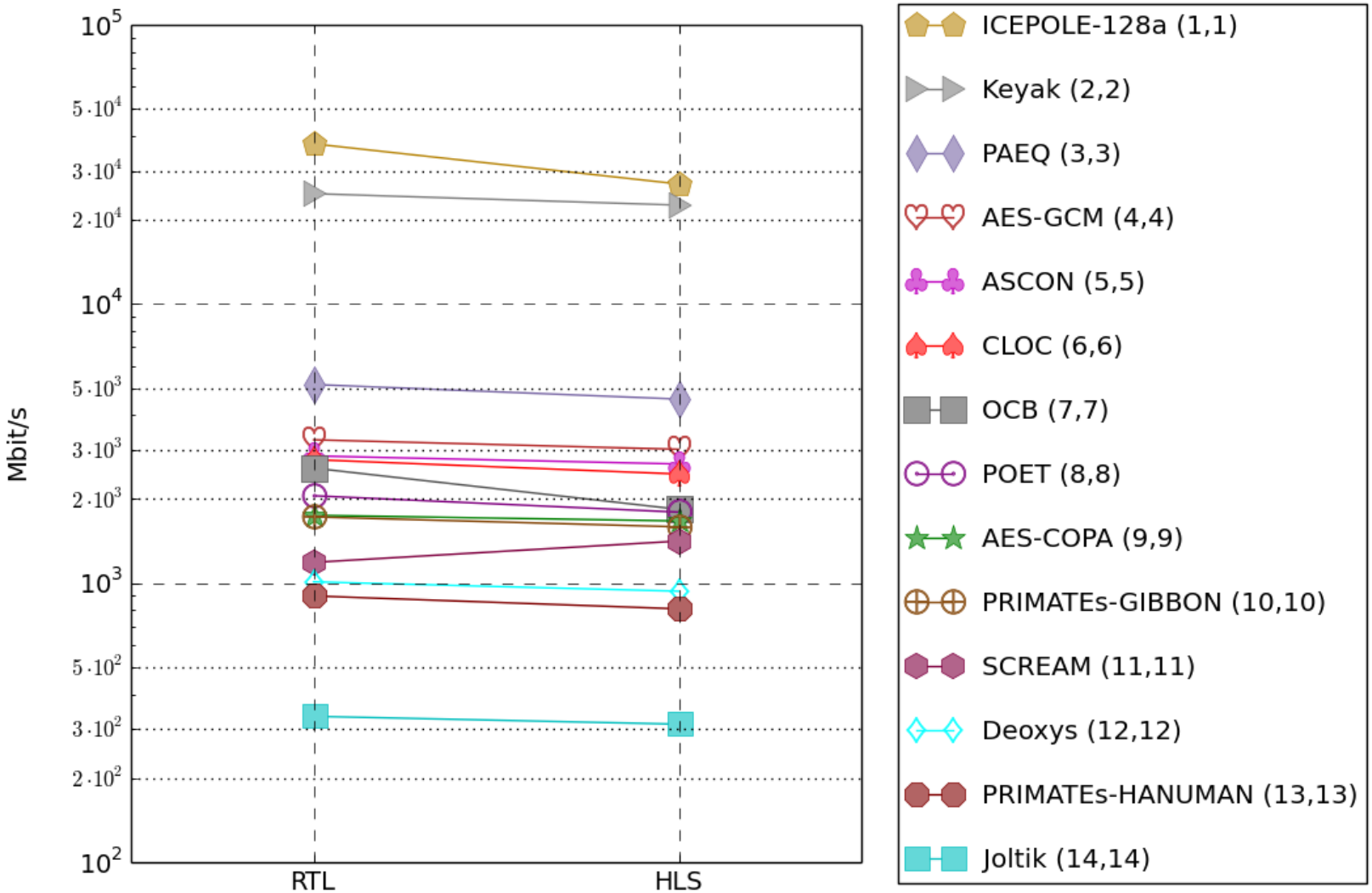
Proposed HLS-Based Development and Benchmarking Flow



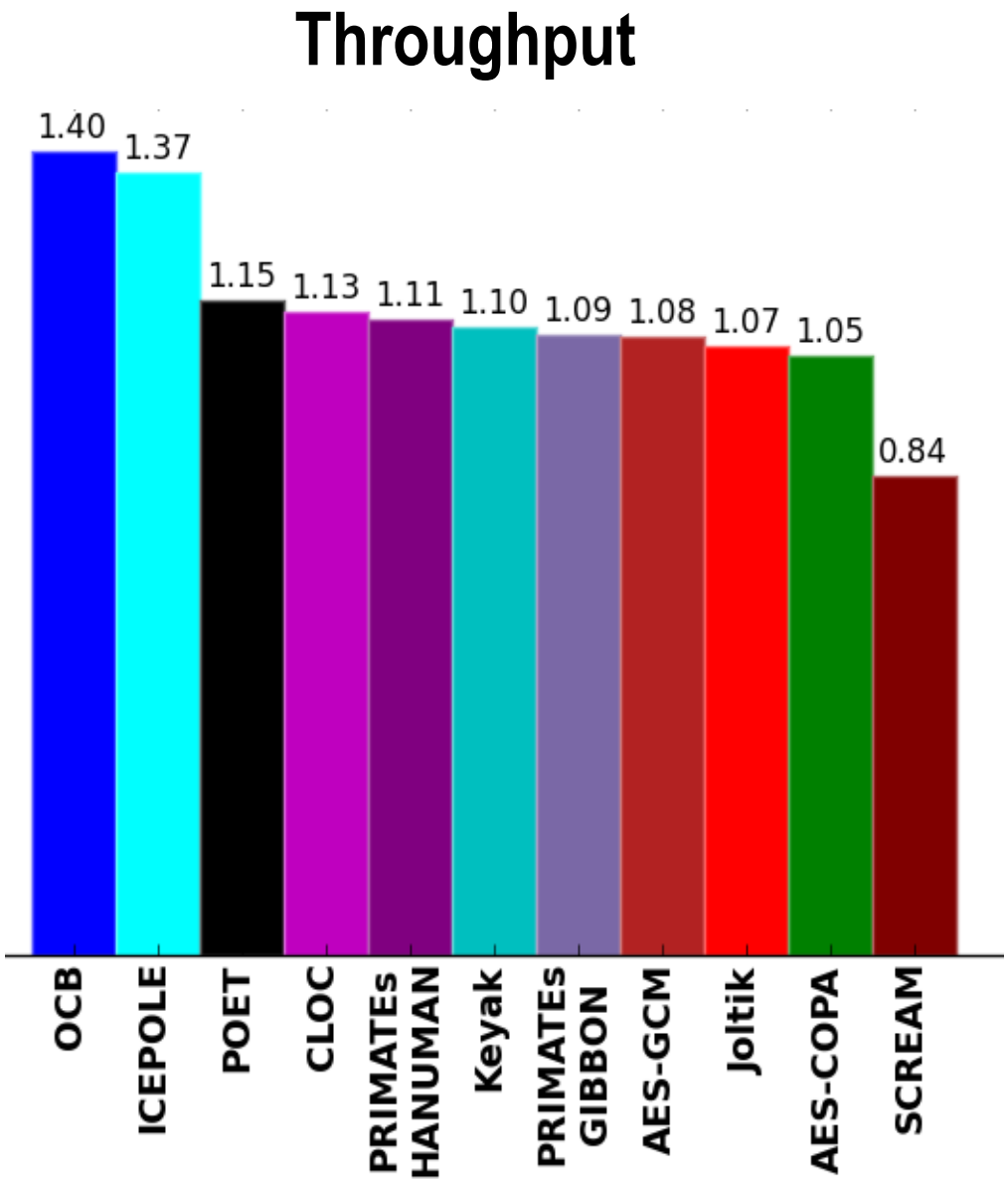
Our Test Case

- 14 Round 2 CAESAR candidates + current standard AES-GCM
- High-speed architecture
- Implementations developed in parallel using RTL and HLS methodology
- Starting point: Informal specifications and reference software implementations in C provided by the algorithm authors
- All RTL & HLS results obtained using a previous version of the GMU hardware API from DIAC 2015 (transition to the new API in progress)

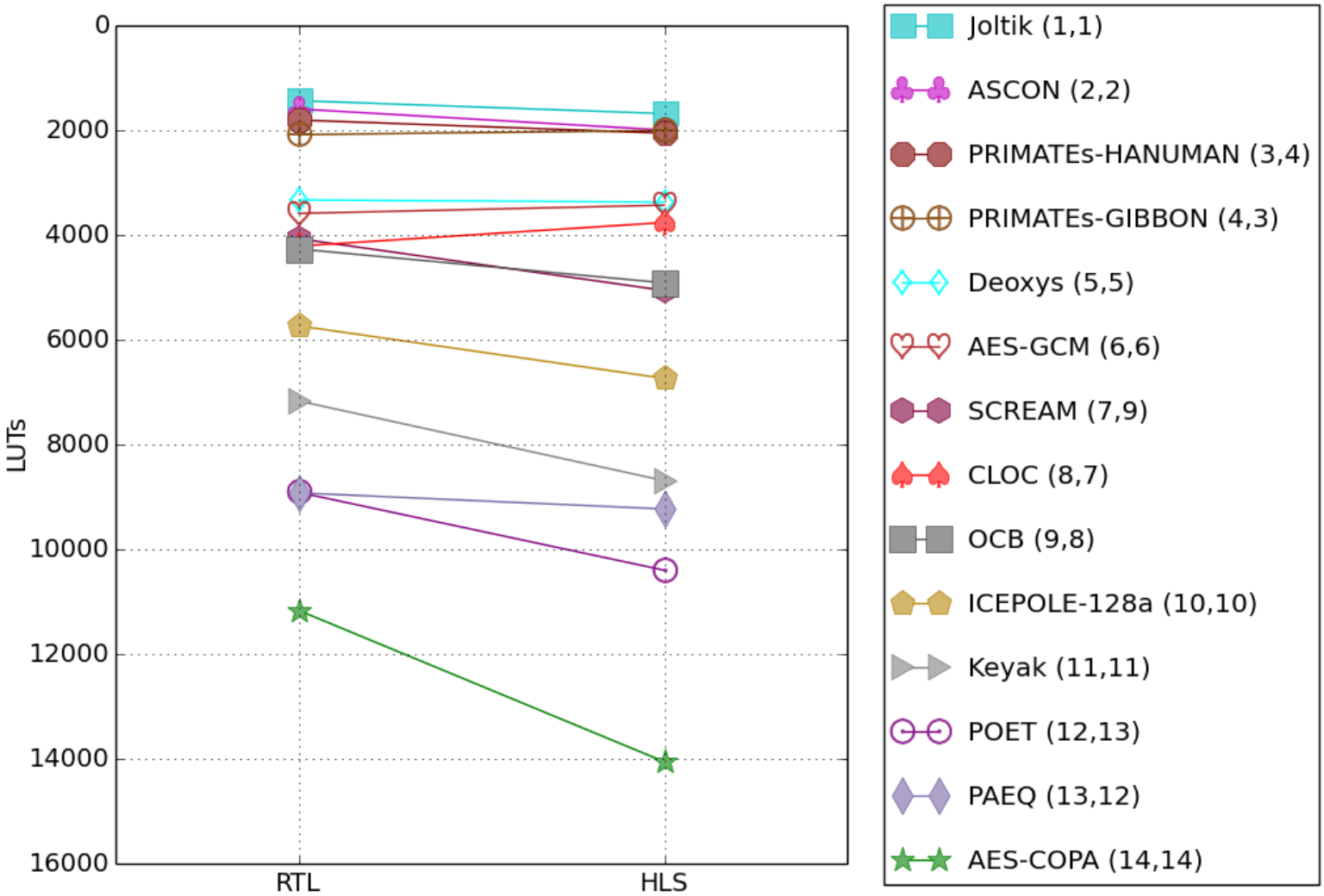
RTL vs. HLS Throughput in Virtex 7



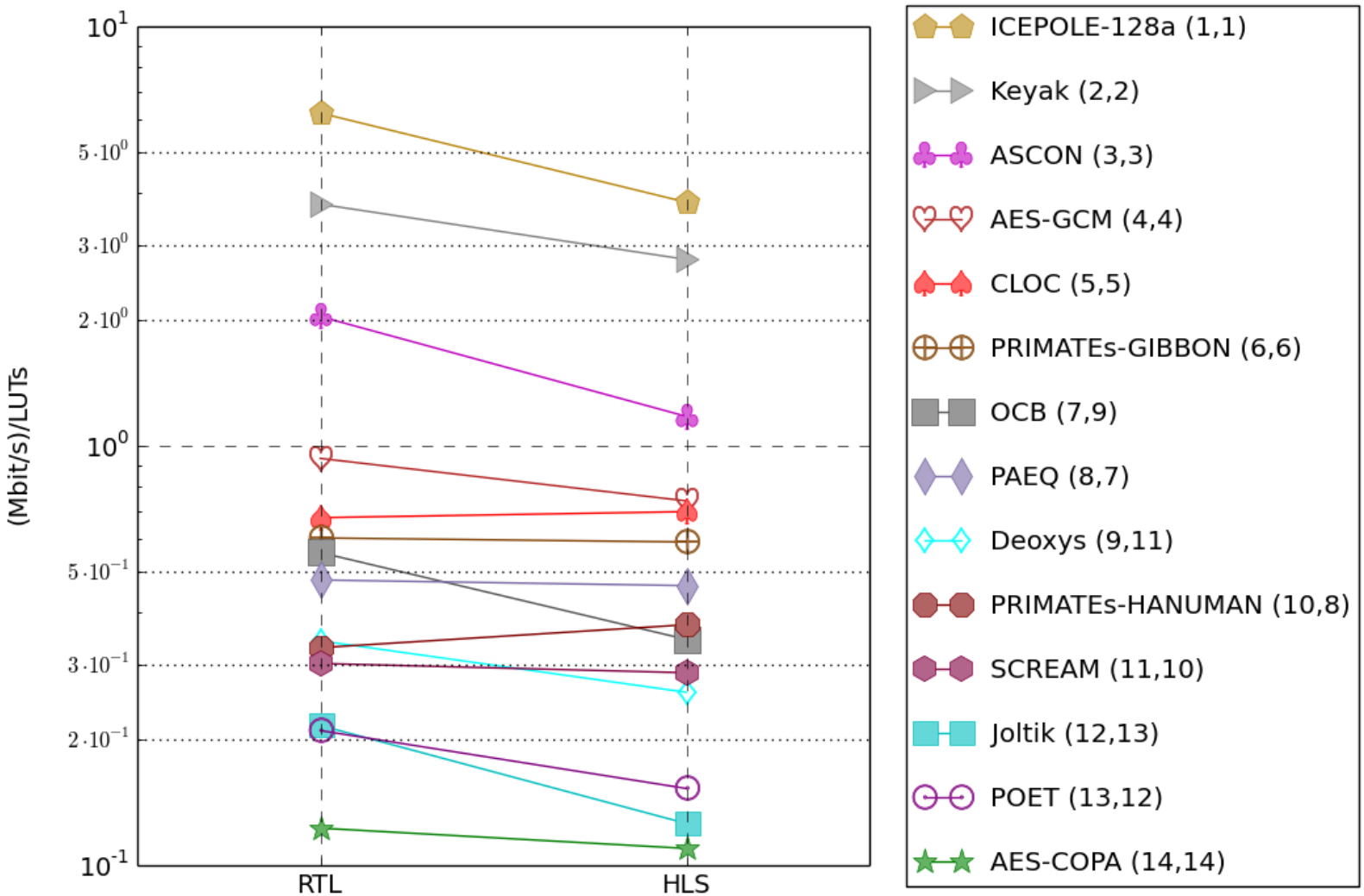
RTL vs. HLS Ratios in Virtex 7



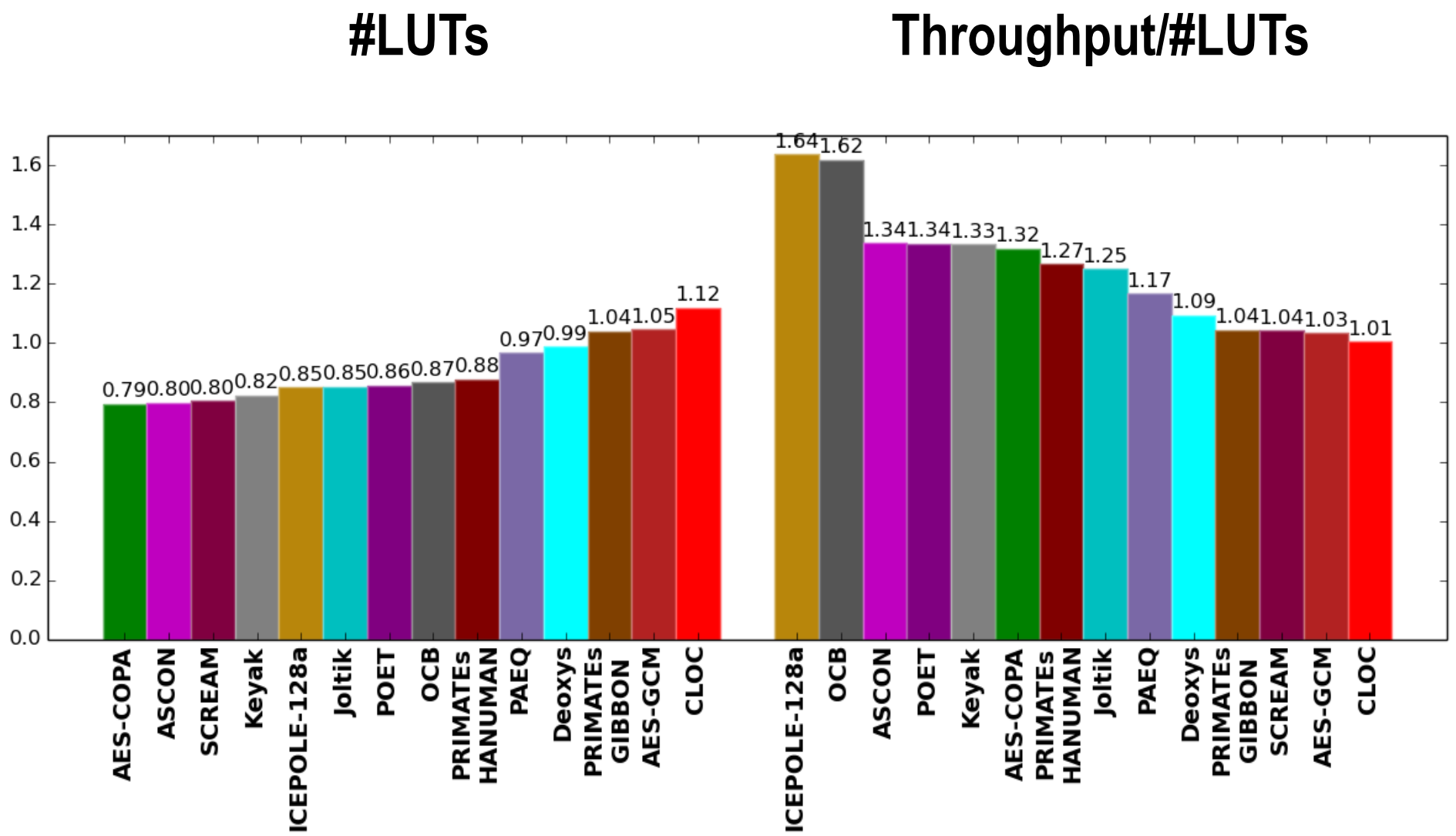
RTL vs. HLS #LUTs in Virtex 7



RTL vs. HLS Throughput/#LUTs in Virtex 7



RTL vs. HLS Ratios in Virtex 7



Tentative Results & Conclusions

- Case study based on 14 Round 1 CAESAR candidates & AES-GCM demonstrated correct ranking for majority of candidates using all major performance metrics
- High-level synthesis offers a potential to facilitate hardware benchmarking during the design of cryptographic algorithms and at the early stages of cryptographic contests
- More research & development needed to overcome remaining difficulties
    - Wide range of RTL to HLS performance metric ratios
    - A few potentially suboptimal HLS or RTL implementations
    - Efficient and reliable generation of HLS-ready C codes

ATHENa Database of Results for Authenticated Ciphers

- Available at
  http://cryptography.gmu.edu/athena
- Developed by John Pham, a Master's-level student of Jens-Peter Kaps
- Results can be entered by designers themselves.
  If you would like to do that, please contact us regarding an account.

Ranking View (1)

ATHENA

AUTOMATED TOOL FOR HARDWARE EVALUATION

About

All FPGA Results

FPGA Rankings

Login

Authenticated Encryption FPGA Ranking

Show Help

Result Filtering

Algorithm Group

- ○ Round 2 CAESAR Candidates and current standards
- ◉ Round 1 CAESAR Candidates and current standards

Implementation Type:

- ◉ High Speed Implementations, Single Message Architectures
- ○ High Speed Implementations, All Architectures
- ○ Low Area Implementations

Implementation Approach:

- ○ Register Transfer Level
- ○ High Level Synthesis
- ○ HW/SW Codesign
- ◉ Any

Hardware API:

- ☑ GMU_AEAD_Core_API_v1
- ☐ GMU_AEAD_API_v1
- ☐ GMU_CipherCore_API_v1
- ☐ Full-Block width(custom)
- ☐ GMU_AEAD_Core_API_v0

Key Size:

- ○ 128
- ○ From 80 To 256
- ◉ Any

Ranking View (2)

Throughput for:

- (•) Authenticated Encryption
- ( ) Authenticated Decryption
- ( ) Authentication Only

Min Area: 0

Max Area: 1000000

Min Throughput: 0

Max Throughput: 1000000

Source:

- [ ] Source Available

Ranking:

- (•) Throughput/Area
- ( ) Throughput
- ( ) Area

Please note that codes with primitives, megafunctions, or embedded resources are not fully portable.

Update

Compare Selected

Show 25 entries

Result ID	Algorithm Disable Unique	Key Size [bits]	Implementation Approach	Hardware API	Arch Type
154	ICEPOLE	128	RTL	GMU_AEAD_Core_API_v1.1	Basic Iterative
73	Keyak	128	RTL	GMU_AEAD_Core_API_v1	Basic Iterative
62	AES-GCM	128	RTL	GMU_AEAD_Core_API_v1	Basic Iterative
65	CLOC	128	HLS	GMU_AEAD_Core_API_v1	Basic Iterative
80	PRIMATEs-GIBBON	120	RTL	GMU_AEAD_Core_API_v1	Basic Iterative
144	OCB	128	RTL	GMU_AEAD_Core_API_v1	Basic Iterative
124	PRIMATEs-HANUMAN	120	HLS	GMU_AEAD_Core_API_v1	Basic Iterative
86	SCREAM	128	RTL	GMU_AEAD_Core_API_v1	Basic Iterative
142	Joltik	128	RTL	GMU_AEAD_Core_API_v1	Basic Iterative
75	POET	128	RTL	GMU_AEAD_Core_API_v1	Basic Iterative
60	AES-COPA	128	RTL	GMU_AEAD_Core_API_v1	Basic Iterative

Details of Result ID 97

Algorithm

IV or Nonce Size [bits]:	96
Transformation Category:	Cryptographic
Transformation:	Authenticated Cipher
Group:	Standards
Algorithm:	AES-GCM
Tag Size [bits]:	128
Associated Data Support:	-
Key Size [bits]:	128
Secret Message Number:	-
Secret Message Number Size [bits]:	-
Message Block Size [bits]:	128
Other Parameters:	-
Specification:	**SP-800-38D.pdf**
Formula for Message Size After Padding:	-

Design

Design ID:	**21**
Impl Approach:	HLS
Hardware API:	GMU_AEAD_Core_API_v1
Primary Optimization Target:	Throughput/Area
Secondary Optimization Target:	-
Architecture Type:	Basic Iterative
Description Language:	VHDL
Use of Megafunctions or Primitives:	No
List of Megafunctions or Primitives:	-
Maximum Number of Streams Processed in Parallel:	1
Number of Clock Cycles per Message Block in a Long Message:	12
Datapath Width [bits]:	128
Padding:	Yes
Minimum Message Unit:	-
Input Bus Width [bits]:	32
Output Bus Width [bits]:	32

Comparison of Result #s 95 and 97

Algorithm		
IV or Nonce Size [bits]:	96	96
Transformation Category:	Cryptographic	Cryptographic
Transformation:	Authenticated Cipher	Authenticated Cipher
Group:	Standards	Standards
Algorithm:	AES-GCM	AES-GCM
Tag Size [bits]:	128	128
Associated Data Support:		
Key Size [bits]:	128	128
Secret Message Number:		
Secret Message Number Size [bits]:	-	-
Message Block Size [bits]:	128	128
Other Parameters:		
Specification:	**SP-800-38D.pdf**	**SP-800-38D.pdf**
Formula for Message Size After Padding:		
Design		
Design ID:	20	21
Impl Approach:	RTL	HLS
Hardware API:	GMU_AEAD_Core_API_v1	GMU_AEAD_Core_API_v1
Primary Optimization Target:	Throughput/Area	Throughput/Area
Secondary Optimization Target:		
Architecture Type:	Basic Iterative	Basic Iterative
Description Language:	VHDL	VHDL
Use of Megafunctions or Primitives:	No	No
List of Megafunctions or Primitives:		
Maximum Number of Streams Processed in Parallel:	1	1
Number of Clock Cycles per Message Block in a Long Message:	11	12
Datapath Width [bits]:	128	128
Padding:	Yes	Yes
Minimum Message Unit:		
Input Bus Width [bits]:	32	32

Final Benchmarking for Round 2

- Implementations developed by multiple groups worldwide
- High-speed & lightweight designs; RTL & HLS
- Deadline for the submission: June 30, 2016
- Benchmarking by the GMU Team using ATHENa and optimization tools of FPGA vendors: July 1-July 15, 2016
- All results available in ATHENa database on July 18, 2016
- Independent benchmarking efforts, aimed at better optimization of tool options and assuring reproducibility of results, very welcome!

Thank you!

Comments?

Questions?

Suggestions?

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