C vs. VHDL: Benchmarking CAESAR Candidates Using High-Level Synthesis and Register-Transfer Level Methodologies

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

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

Primary High-Level Synthesis (HLS) Support for This Project

Ekawat Homsirikamol
a.k.a "Ice"

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

Ahmed Ferozpuri

PAEQ
PRIMATEs-APE
PRIMATEs-GIBBON
PRIMATEs-HANUMAN

Will Diehl

Minalpher
POET
SCREAM

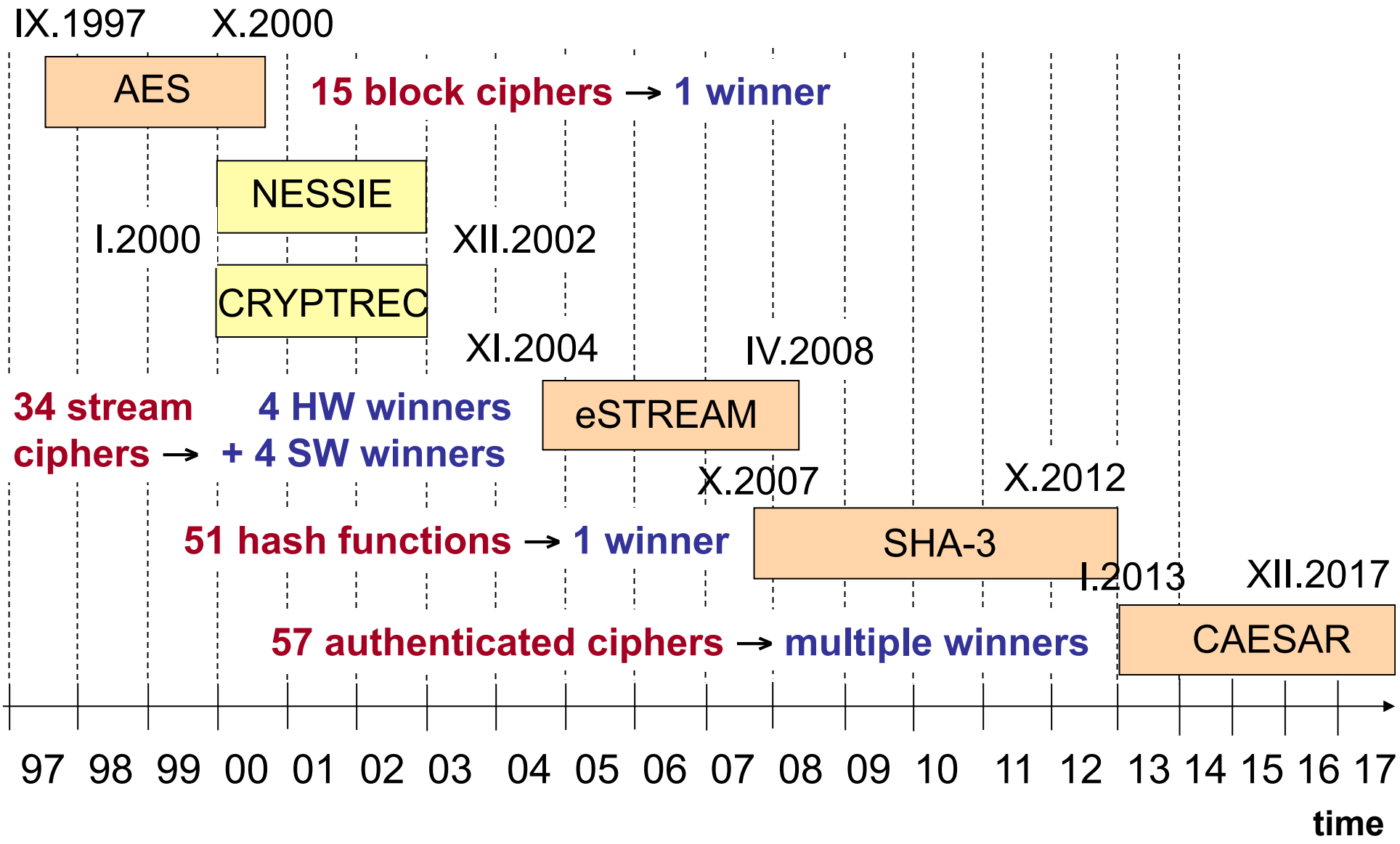
Farnoud Farahmand

AES-COPA
CLOC

"Ice" Homsirikamol

AES-GCM, ASCON, Deoxys, ICEPOLE, Joltik, Keyak, OCB

Cryptographic Standard Contests



Evaluation Criteria

Security

Software Efficiency

µProcessors µControllers

Hardware Efficiency

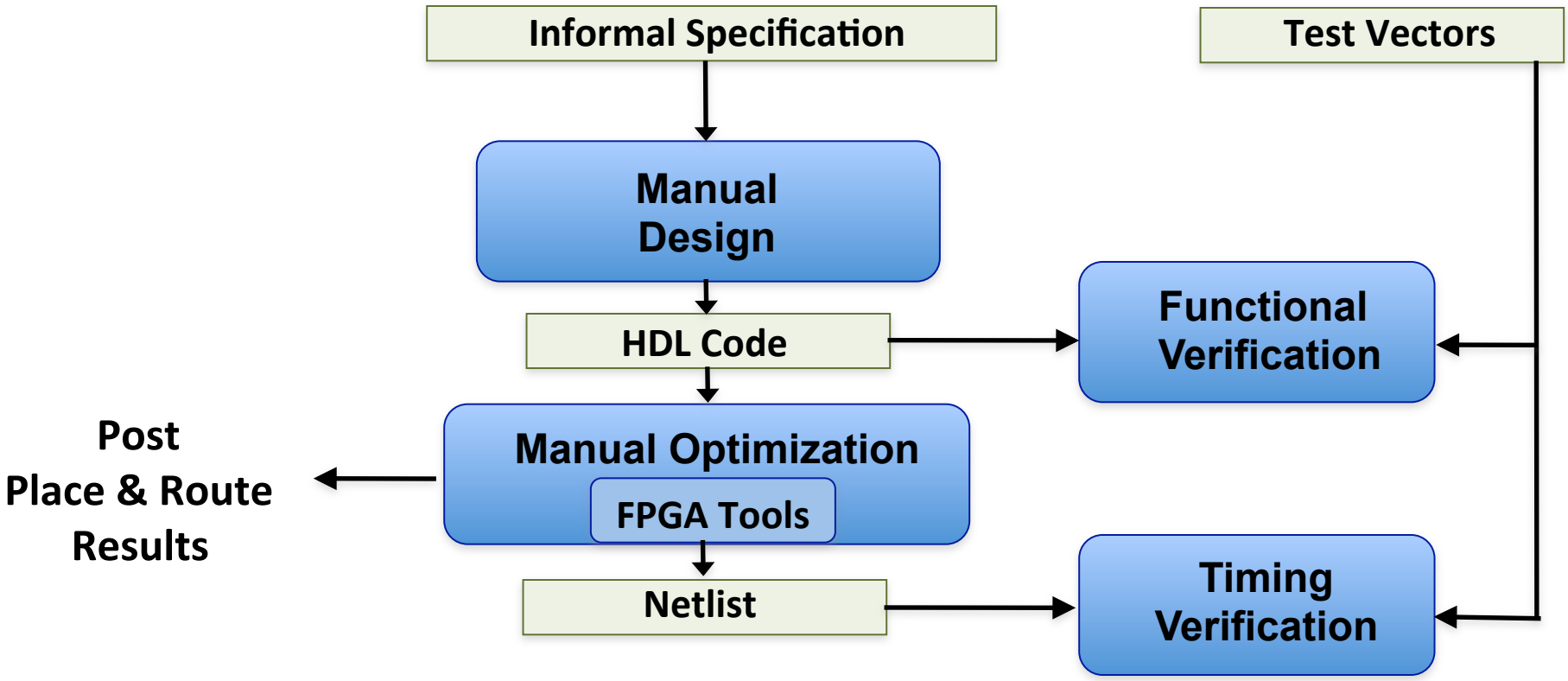
FPGAs ASICs

Flexibility

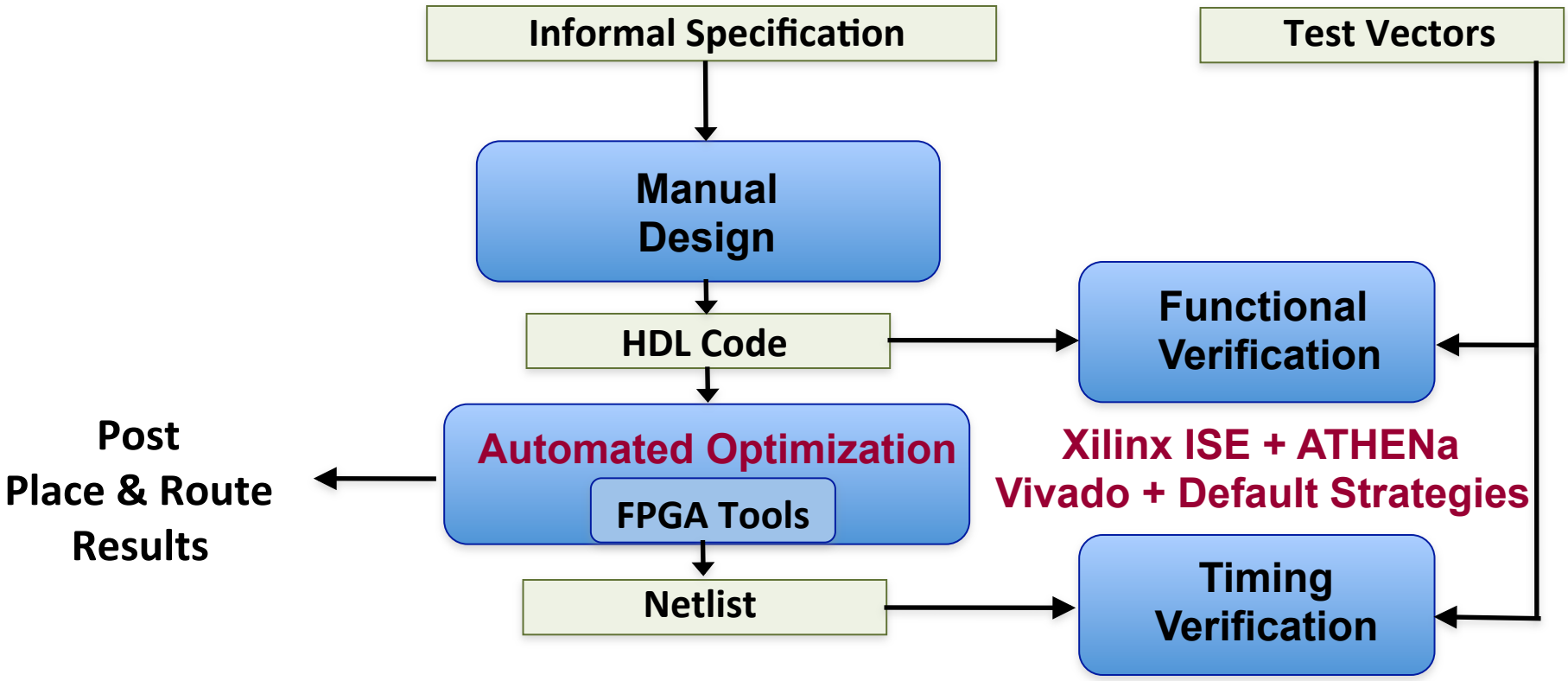
Simplicity

Licensing

Traditional Development & Benchmarking Flow



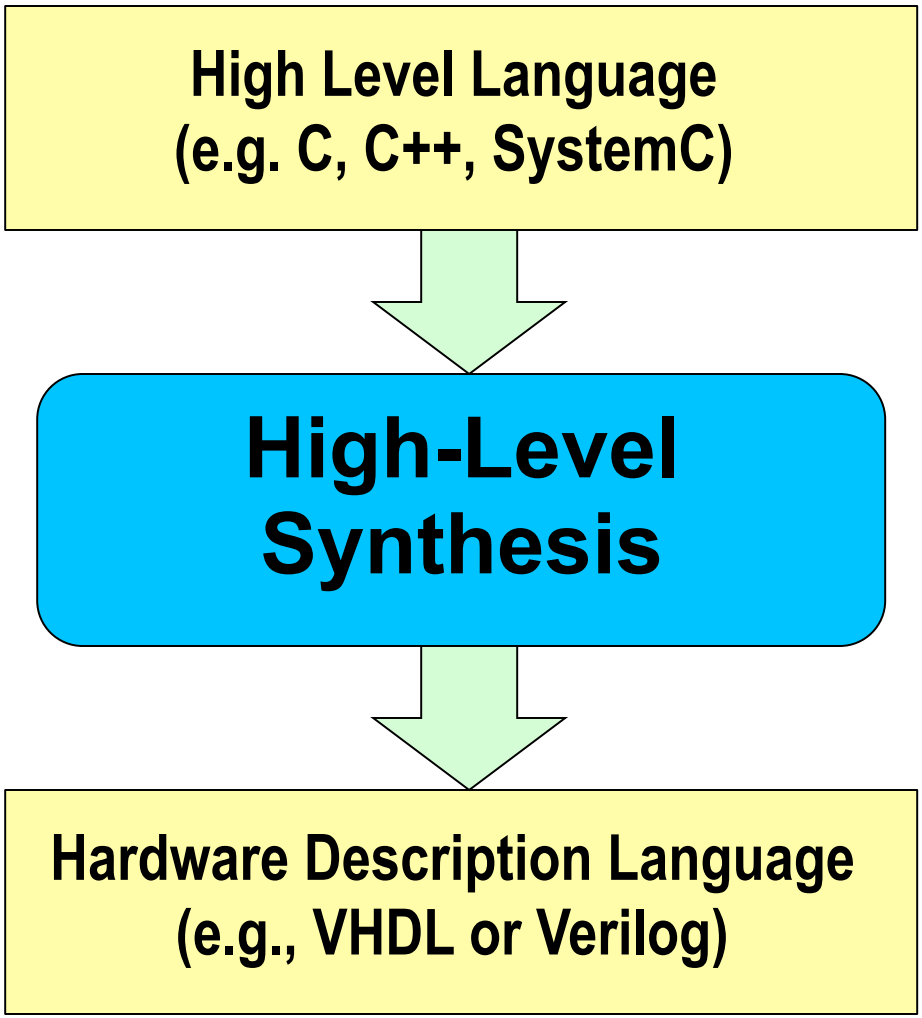
Extended Traditional Development & Benchmarking Flow



Remaining Difficulties of Hardware Benchmarking

- Large number of candidates
- 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)
- High-speed vs. lightweight algorithms
- Multiple hardware architectures
- Dependence on skills of designers

High-Level Synthesis (HLS)



Cinderella Story: Vivado HLS

AutoESL Design Technologies, Inc. (25 employees)

Flagship product:

AutoPilot, translating **C/C++/System C** to **VHDL or Verilog**

- **Acquired by the biggest FPGA company, Xilinx Inc., in 2011**
- **AutoPilot integrated into the primary Xilinx toolset, Vivado, as Vivado HLS, released in 2012**

"High-Level Synthesis for the Masses"

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**

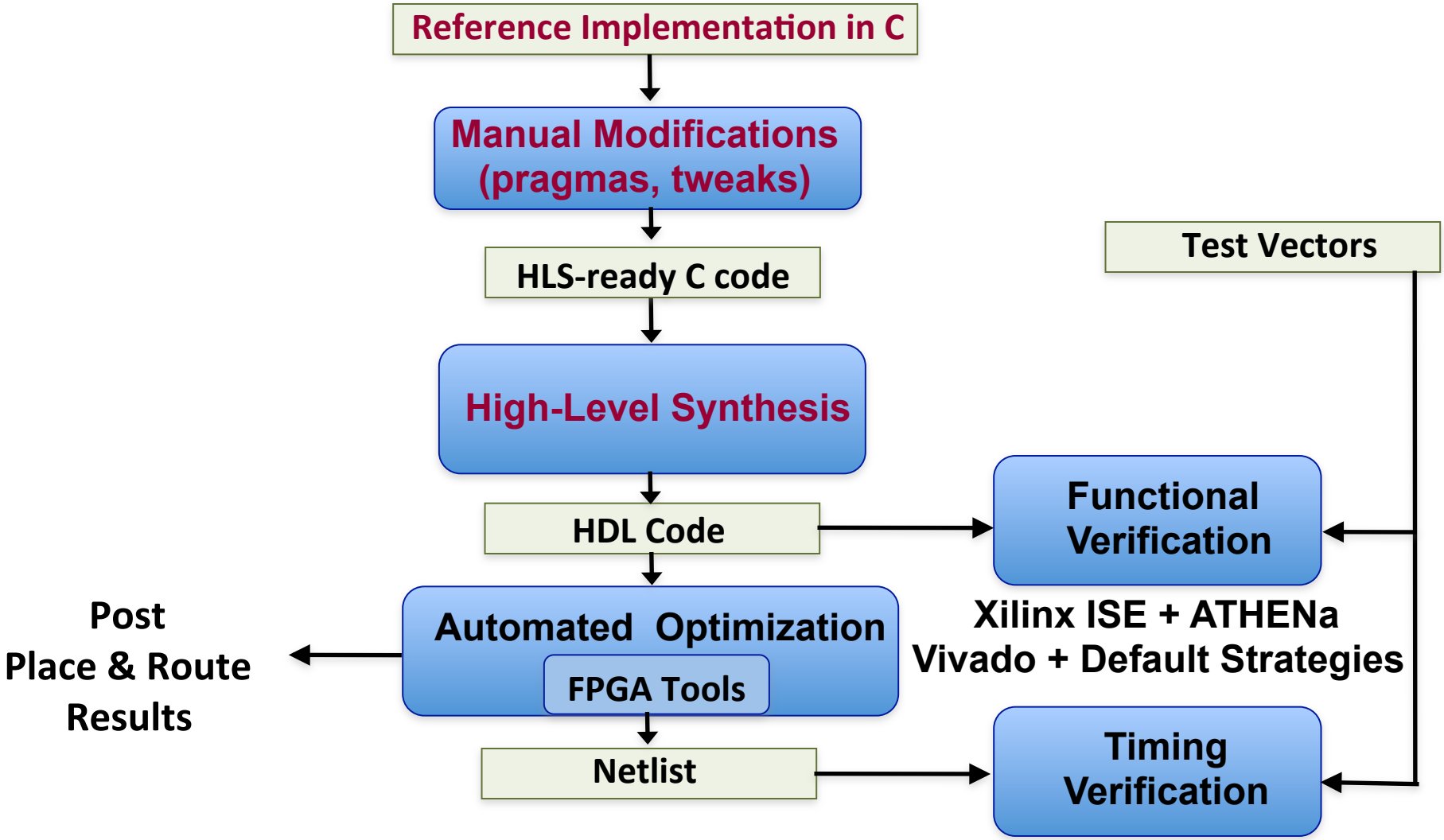
- **The development time will be reduced by at least an order of magnitude**

Potential Additional Benefits

Early feedback for designers of cryptographic algorithms

- Typical design process based only on security analysis and software benchmarking
- Lack of immediate feedback on hardware performance
- Common unpleasant surprises, e.g.,
  - Mars in the AES Contest
  - BMW, ECHO, and SIMD in the SHA-3 Contest

Proposed HLS-Based Development and Benchmarking Flow



Examples of Source Code Modifications

Unrolling of loops:

```
for (i = 0; i < 4; i ++)
#pragma HLS UNROLL
     for (j = 0; j < 4; j ++)
#pragma HLS UNROLL
         b[i][j] = s[i][j];
```

Flattening function's hierarchy:

```
void KeyUpdate (word8 k[4][4],
                word8 round)
{
 #pragma HLS INLINE
        ...
}
```

Function Reuse:

```
// (a) Before modification
  for(round=0; round<NB_ROUNDS; ++
       round)
  {
    if (round == NB_ROUNDS-1)
      single_round(state, 1);
    else
      single_round(state, 0);
  }
```

```
// (b) After modification
  for(round=0; round<NB_ROUNDS; ++
       round)
  {
    if (round == NB_ROUNDS-1)
      x = 1;
    else
      x = 0;
    single_round(state, x);
  }
```

Our Test Case

- 13 Round 1 CAESAR candidates + current standard AES-GCM (2 more in progress)
- Basic iterative architecture
- GMU AEAD Hardware API
- Key scheduling and padding done in hardware
- Implementations developed in parallel using RTL and HLS methodology
- Starting point: Informal specifications and reference software implementations in C provided by the algorithm authors
- Post P&R results generated for
  - Xilinx Virtex 6 using Xilinx ISE + ATHENa, and
  - Virtex 7 and Zynq 7000 using Xilinx Vivado with 25 default option optimization strategies
- No use of BRAMs or DSP Units in AEAD Core

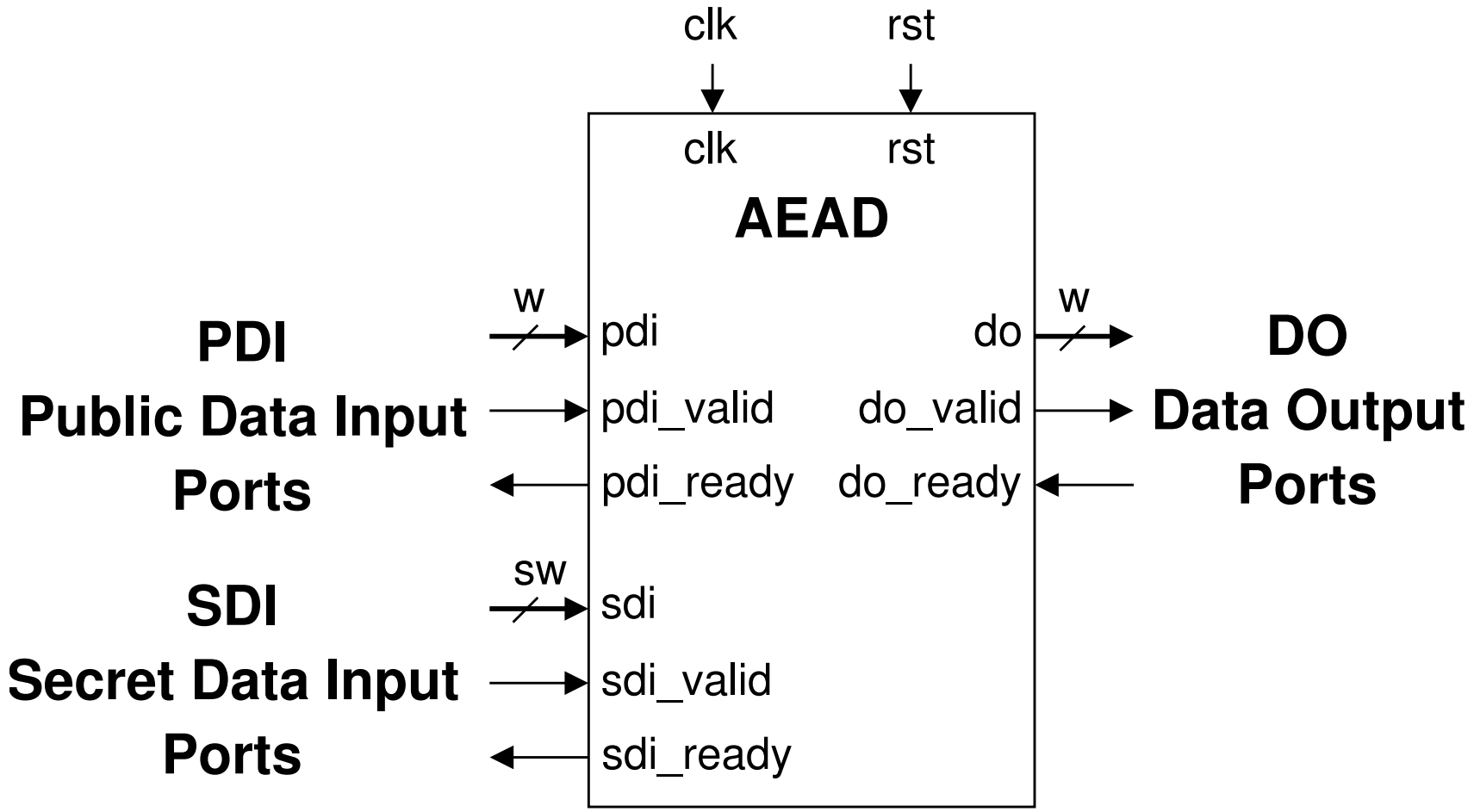
Parameters of Authenticated Ciphers

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

Parameters of Authenticated Ciphers

Algorithm	Key size	Nonce size	Tag size	Basic Primitive
		Permutation Based		
ASCON	128	128	128	SPN
ICEPOLE	128	128	128	Keccak-like
Keyak	128	128	128	Keccak-f
PAEQ	128	96	128	AESQ
PRIMATEs-GIBBON	120	120	120	PRIMATE
PRIMATEs-HANUMAN	120	120	120	PRIMATE

AEAD Interface



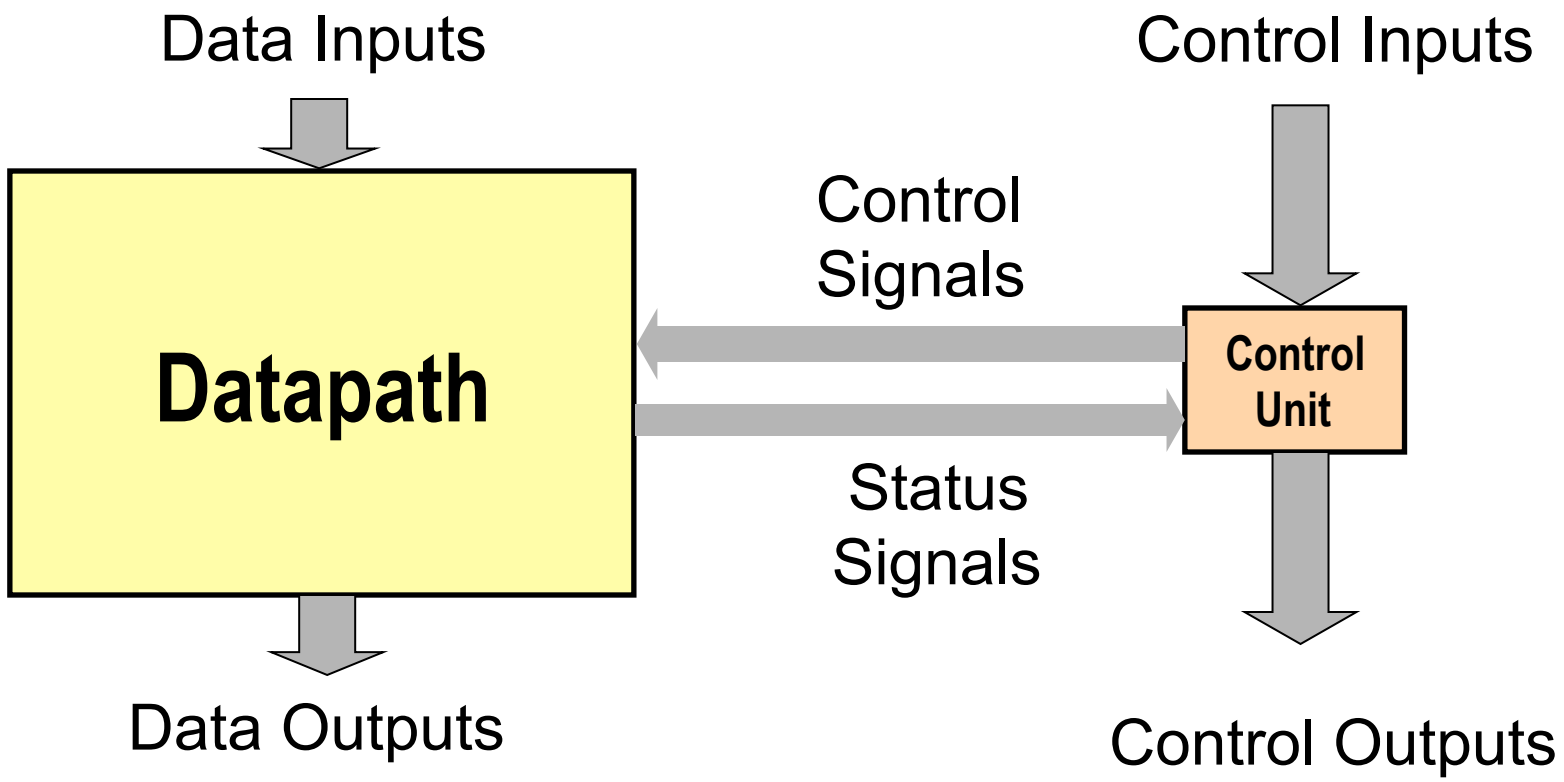
Parameters of Ciphers & GMU Implementations

Algorithm	Word Size, w	Block Size, b	#Rounds	Cycles/Block RTL	Cycles/Block HLS
		Block-cipher Based			
AES-COPA	32	128	10	11	12
AES-GCM	32	128	10	11	12
CLOC	32	128	10	11	12
Deoxys	32	128	14	29	32
Joltik	32	128	32	65	70
OCB	32	128	10	11	12
POET	32	128	10	11	12
SCREAM	32	128	10	11	12

Parameters of Ciphers & GMU Implementations

Algorithm	Word Size, w	Block Size, b	#Rounds	Cycles/Block RTL	Cycles/Block HLS
		Permutation Based			
ASCON	32	64	6	7	8
ICEPOLE	256	1024	6	6	8
Keyak	128	1344	12	12	14
PAEQ	32	368 (M)/ 240 (AD)	20	21	22
PRIMATEs-GIBBON	40	40	6	7	8
PRIMATEs-HANUMAN	40	40	12	13	14

Datapath vs. Control Unit



Determines

- **Area**
- **Clock Frequency**

Determines

- **Number of clock cycles**

Encountered Problems

Control Unit suboptimal

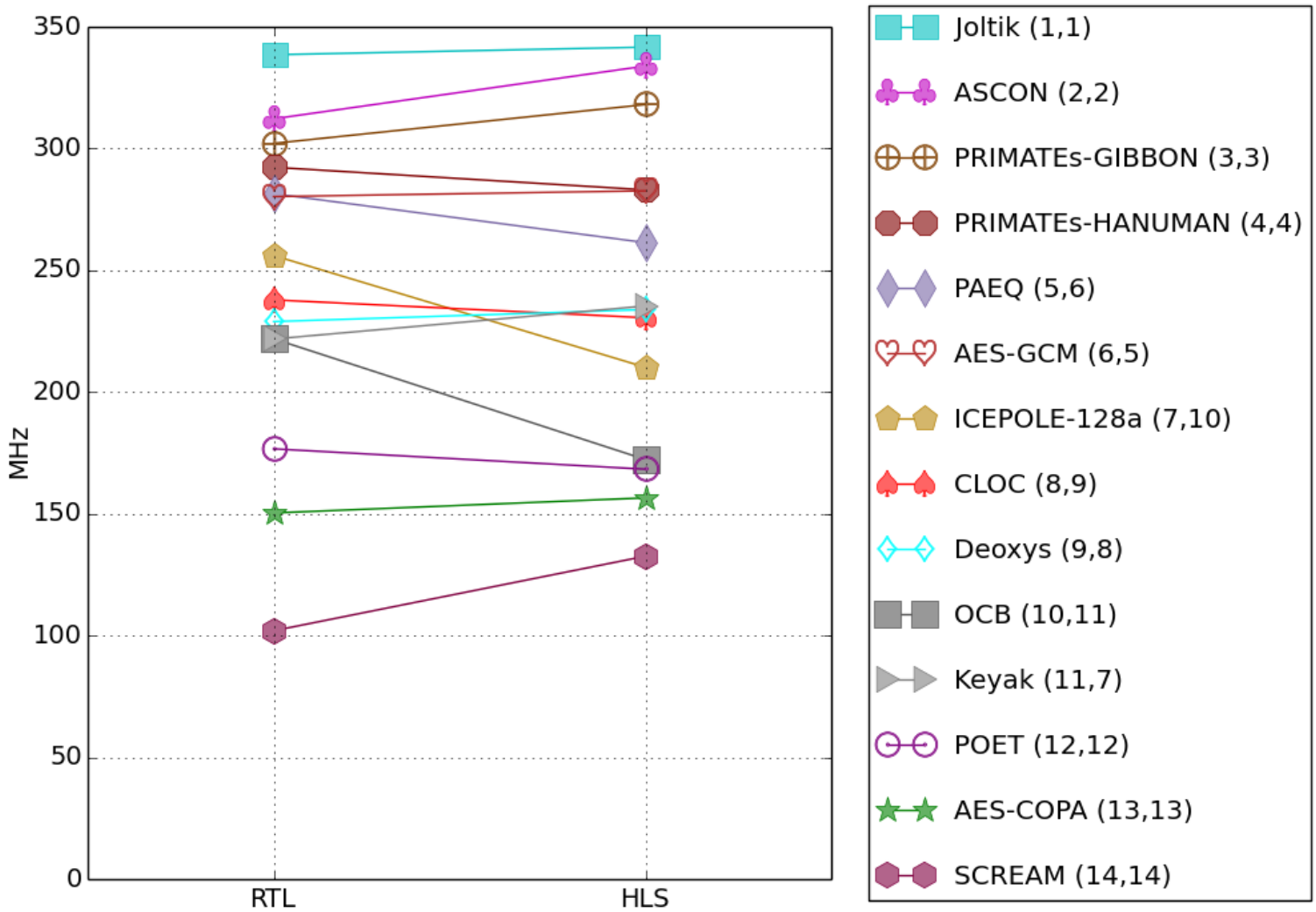
- Difficulty in inferring an overlap between completing the last round and reading the next input block
- One additional clock cycle used for initialization of the state at the beginning of each round
- The formulas for throughput:

HLS: Throughput = Block_size / ((#Rounds+2) * T_{CLK})

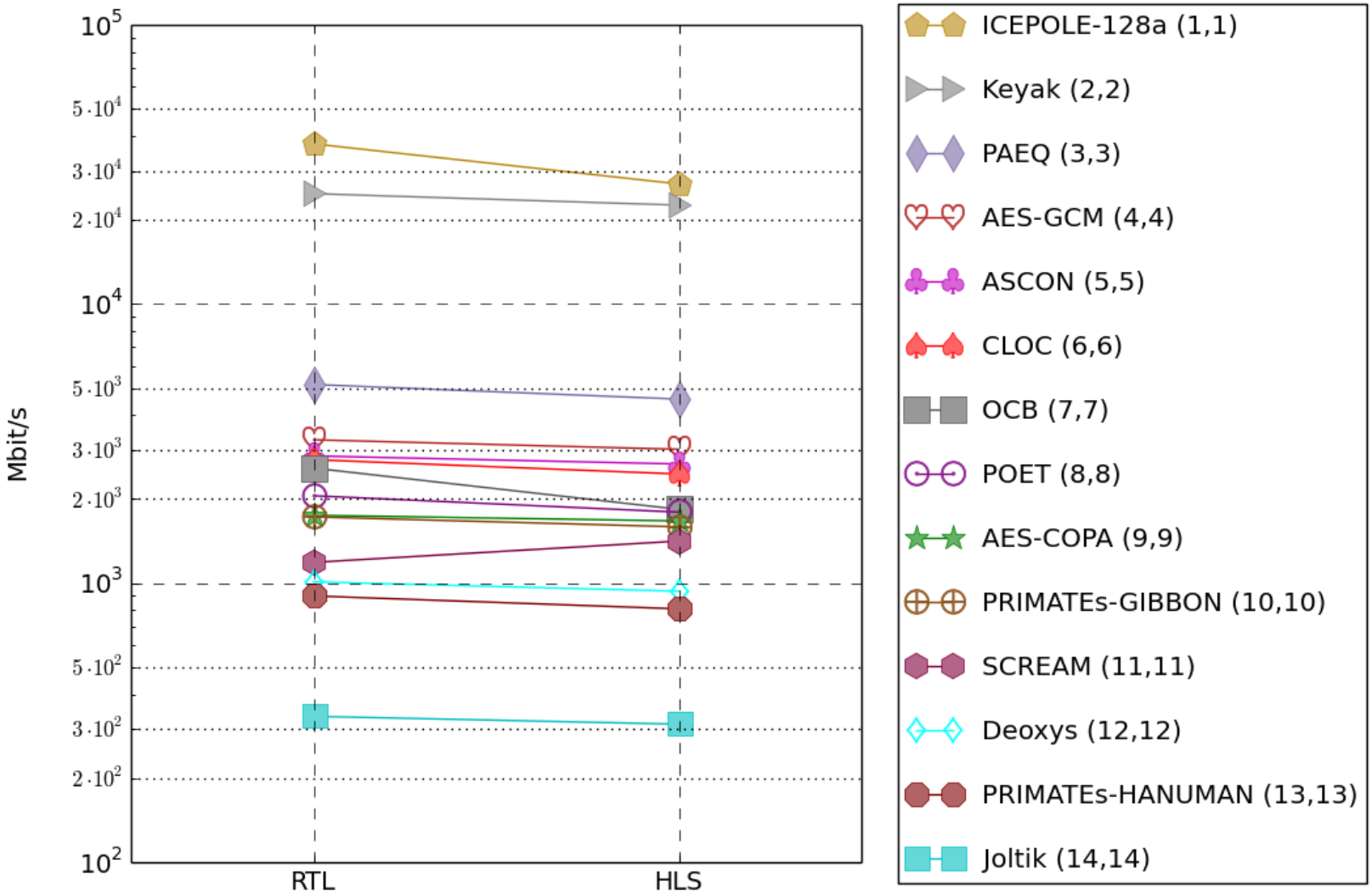
RTL: Throughput = Block_size / (#Rounds+C * T_{CLK})

C=0, 1 depending on the algorithm

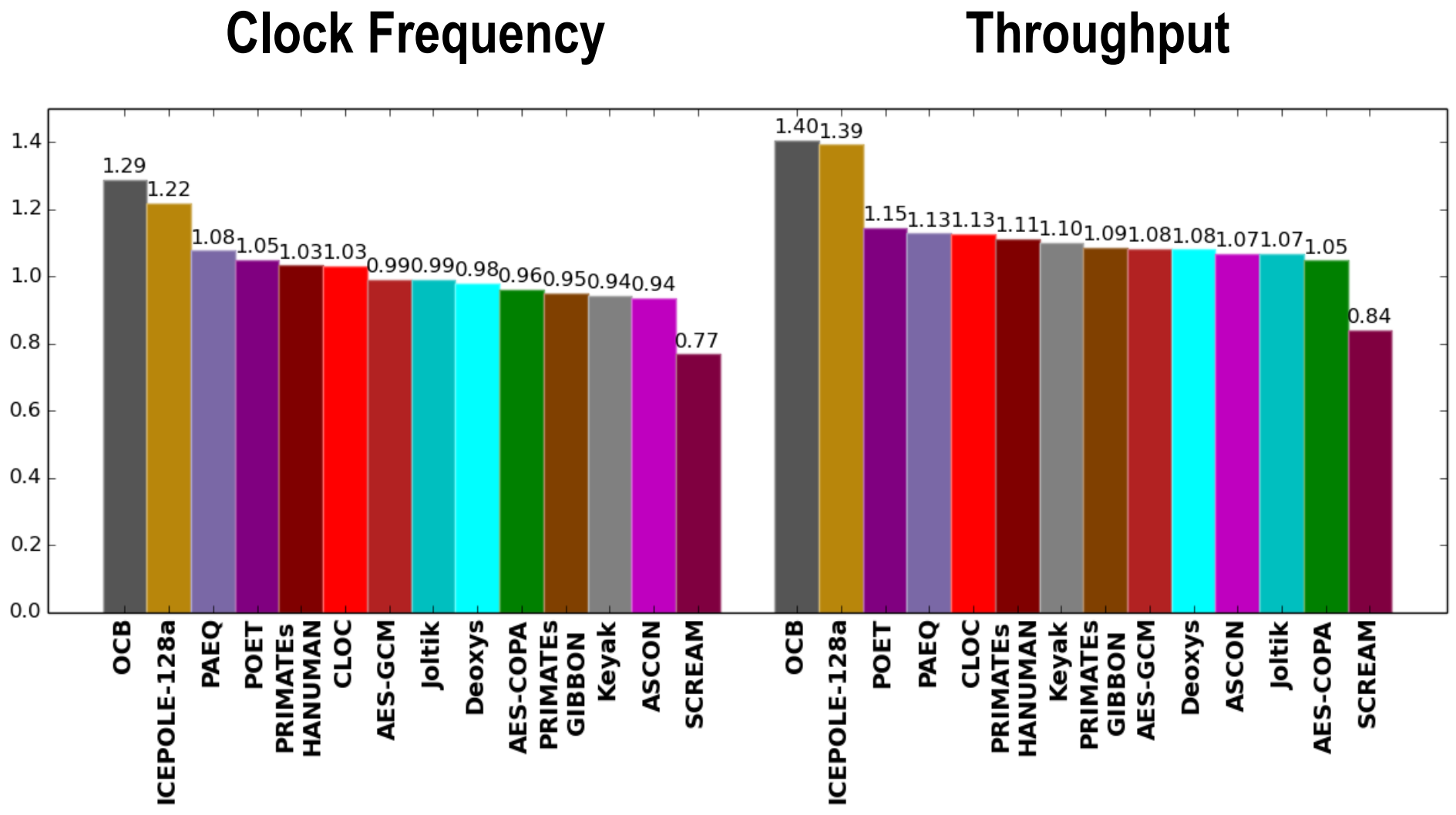
RTL vs. HLS Clock Frequency in Virtex 7



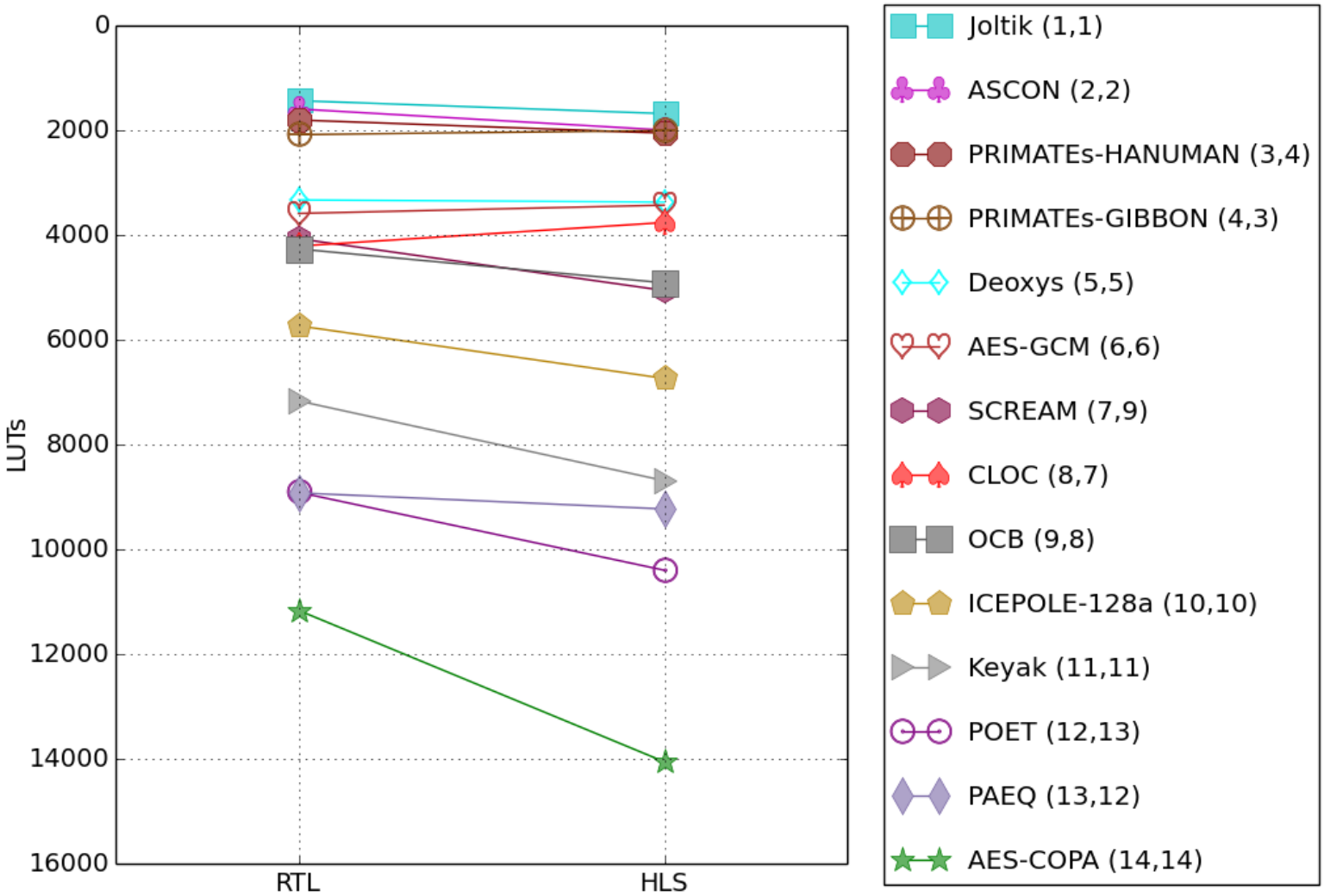
RTL vs. HLS Throughput in Virtex 7



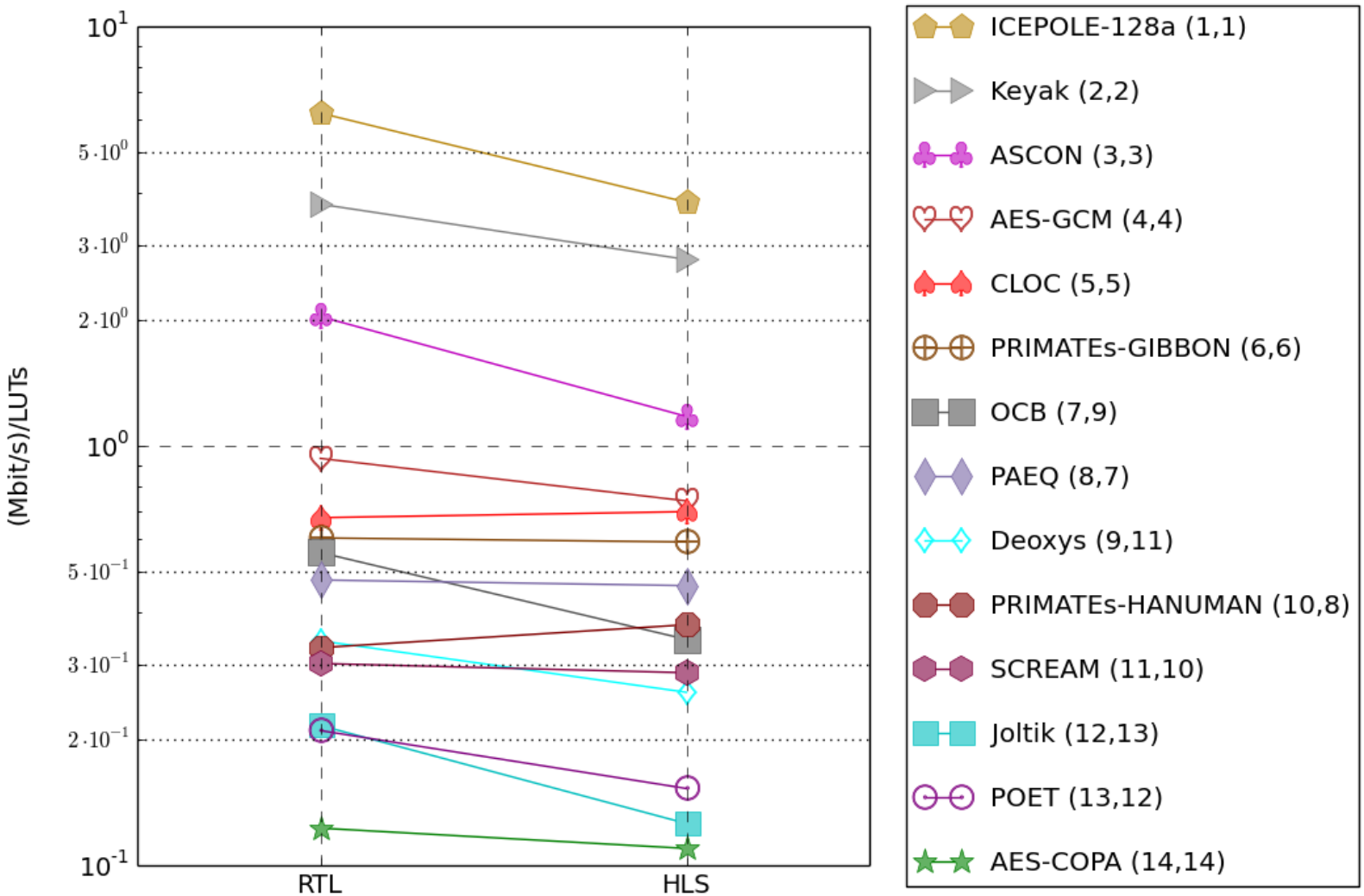
RTL vs. HLS Ratios in Virtex 7



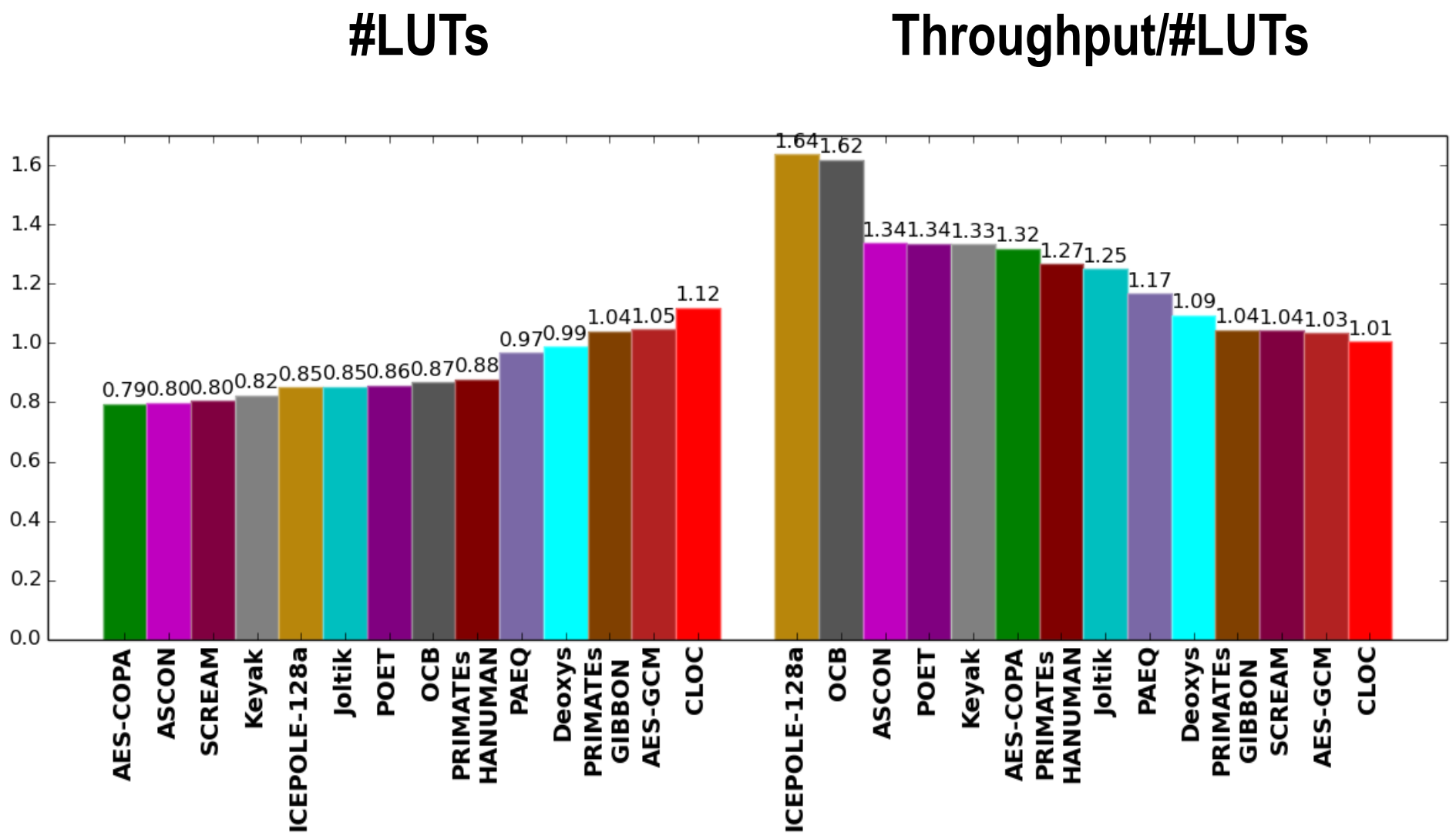
RTL vs. HLS #LUTs in Virtex 7



RTL vs. HLS Throughput/#LUTs in Virtex 7

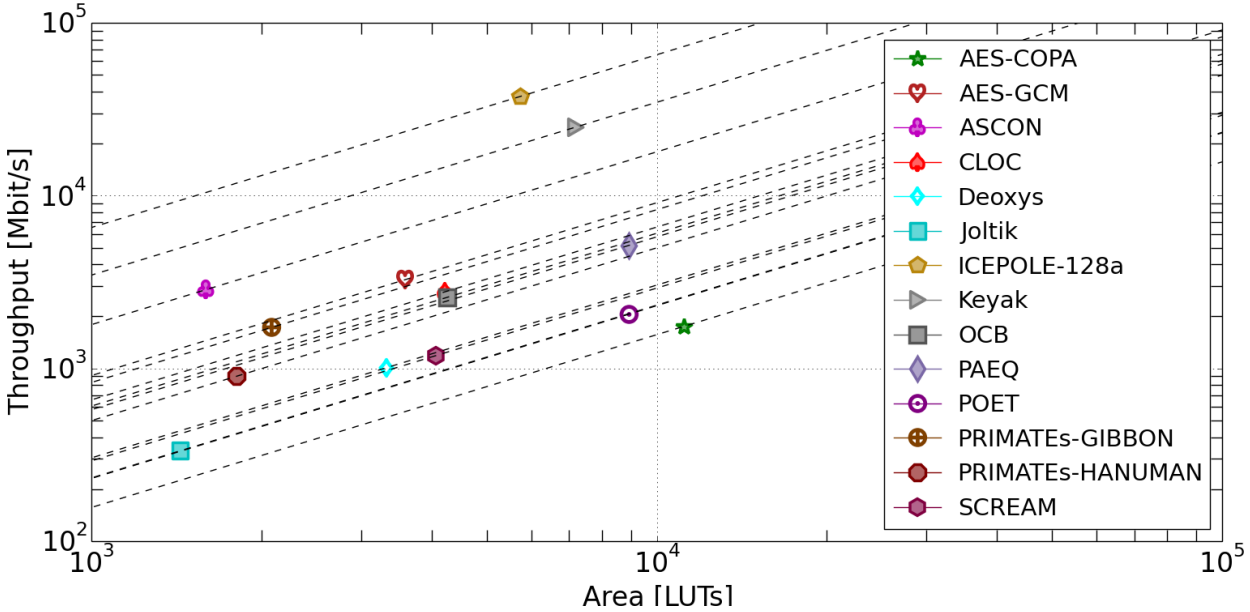


RTL vs. HLS Ratios in Virtex 7

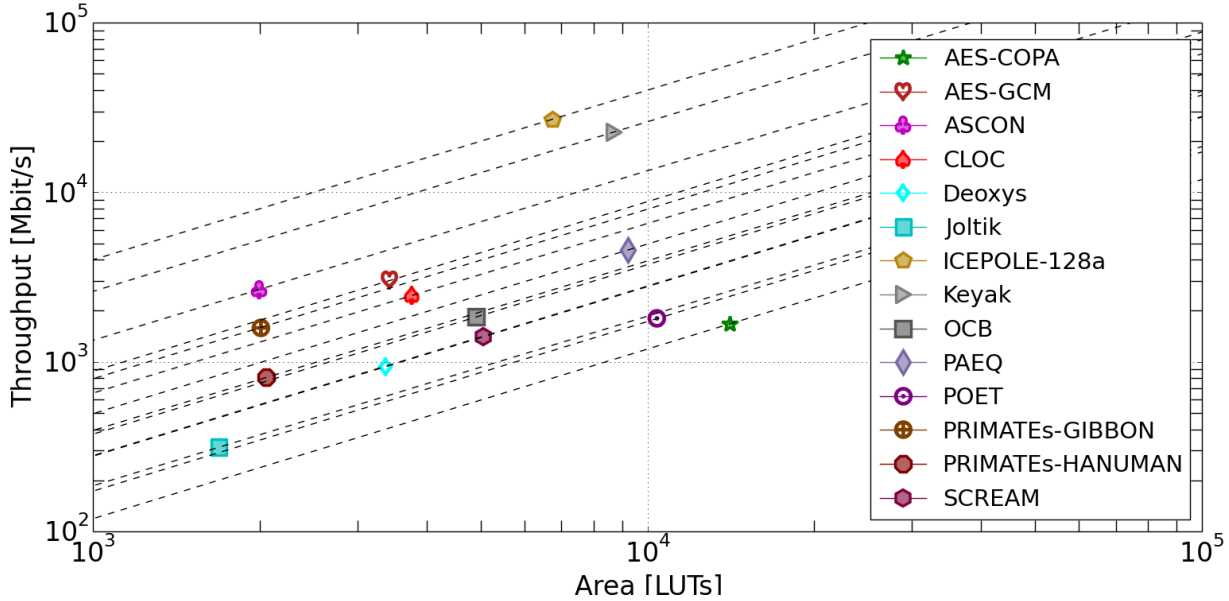


Throughput vs. LUTs in Virtex 7

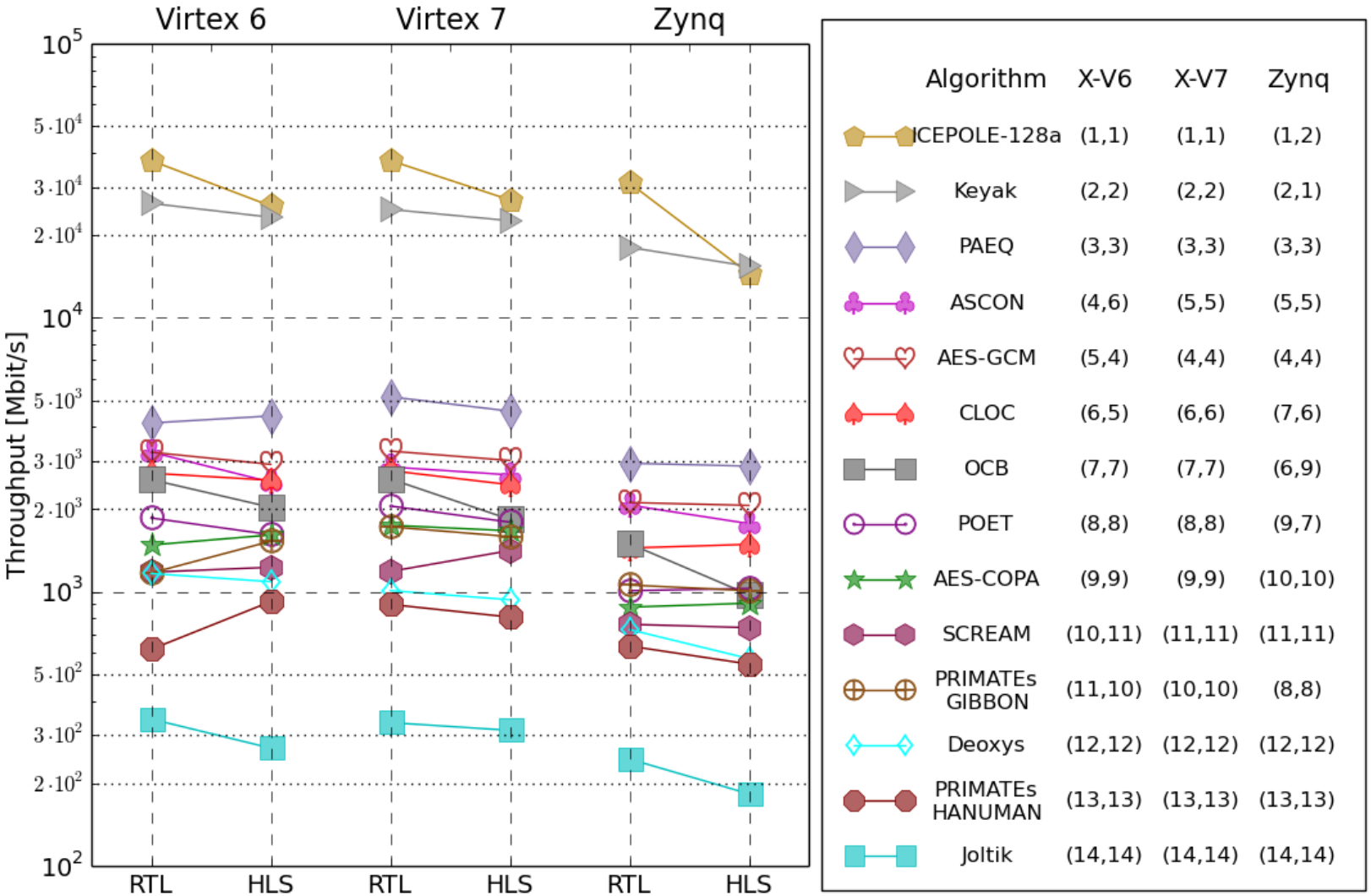
RTL



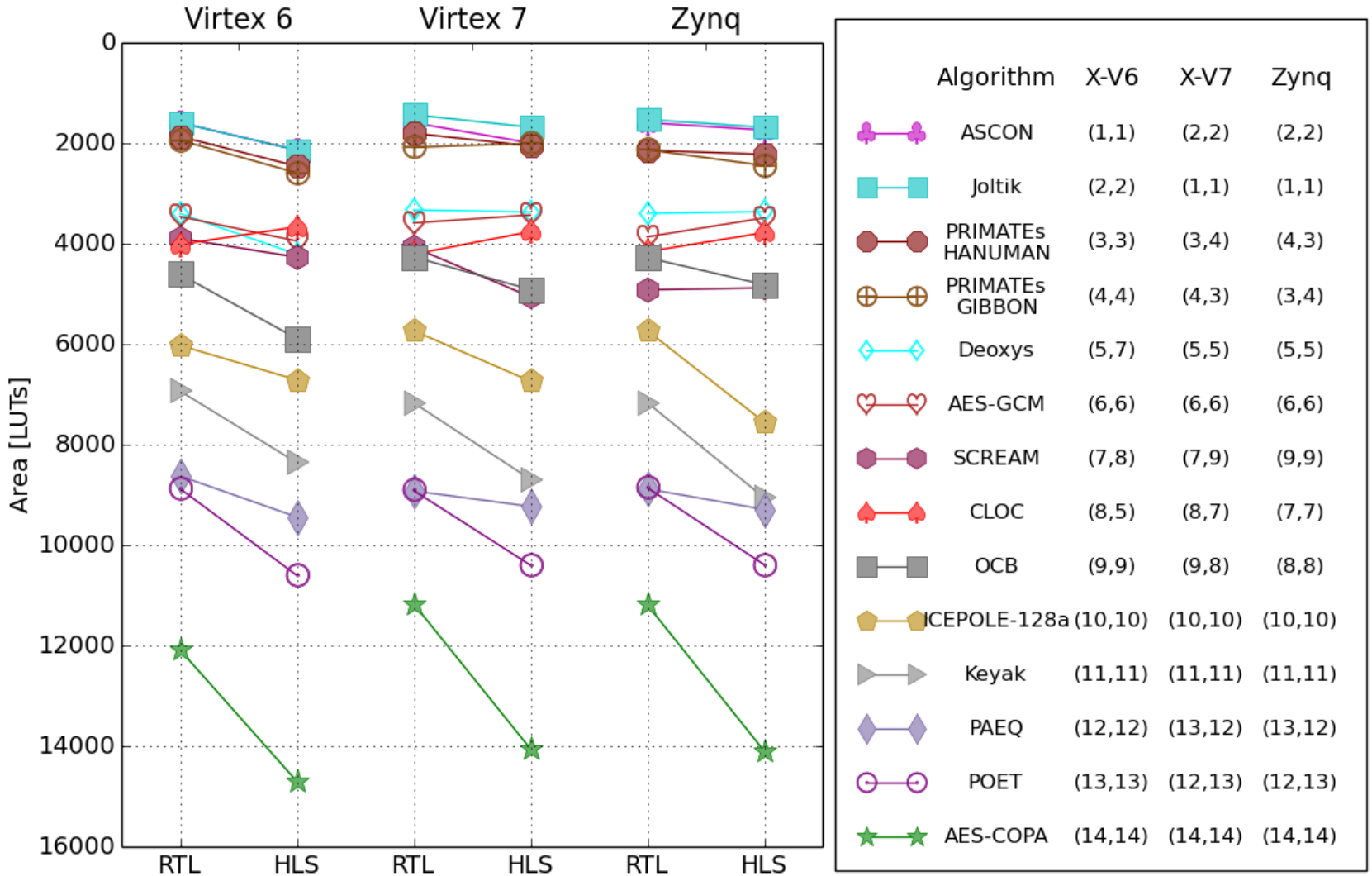
HLS



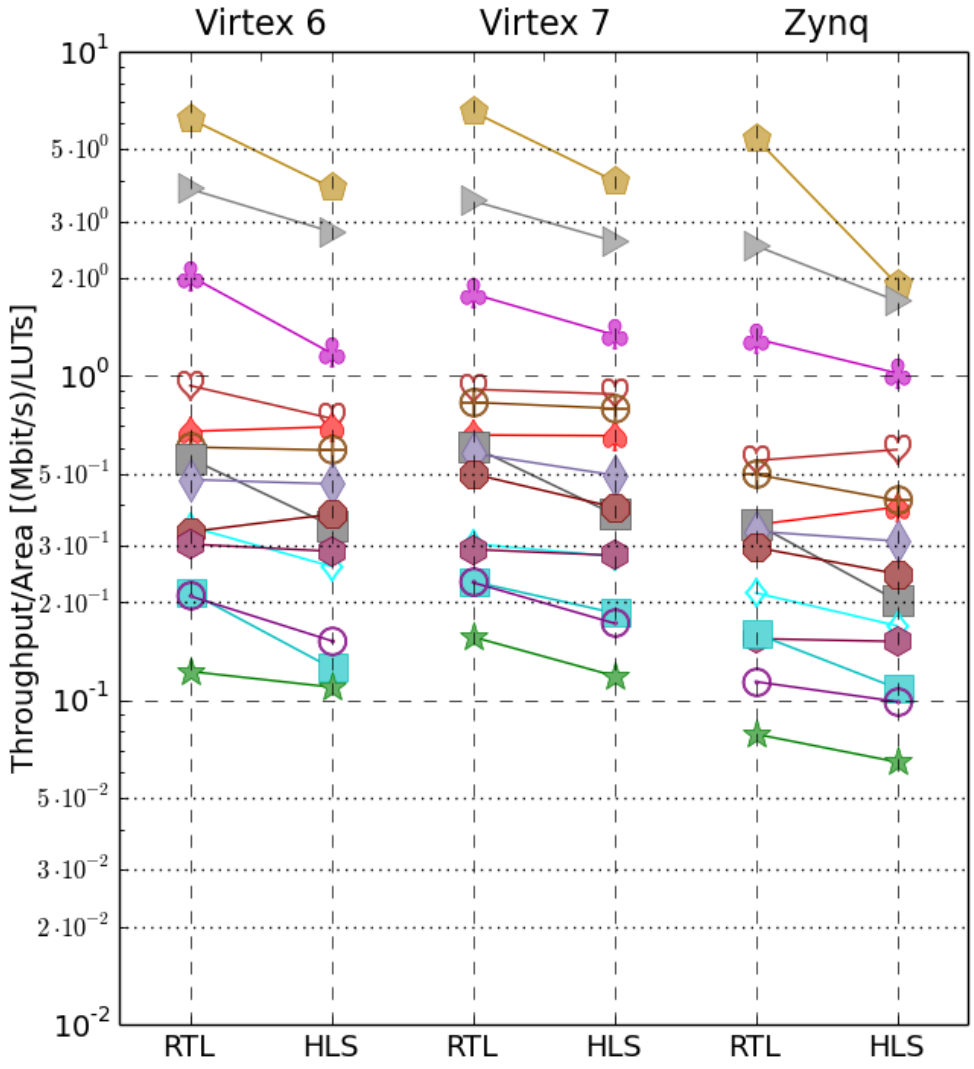
RTL vs. HLS Throughput



RTL vs. HLS #LUTs



RTL vs. HLS Throughput/#LUTs



Algorithm	X-V6	X-V7	Zynq
ICEPOLE-128a	(1,1)	(1,1)	(1,1)
Keyak	(2,2)	(2,2)	(2,2)
ASCON	(3,3)	(3,3)	(3,3)
AES-GCM	(4,4)	(4,4)	(4,4)
CLOC	(5,5)	(6,6)	(7,6)
PRIMATEs GIBBON	(6,6)	(5,5)	(5,5)
OCB	(7,9)	(7,9)	(6,9)
PAEQ	(8,7)	(8,7)	(8,7)
Deoxys	(9,11)	(10,11)	(10,10)
PRIMATEs HANUMAN	(10,8)	(9,8)	(9,8)
SCREAM	(11,10)	(11,10)	(12,11)
Joltik	(12,13)	(12,12)	(11,12)
POET	(13,12)	(13,13)	(13,13)
AES-COPA	(14,14)	(14,14)	(14,14)

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.
- The ATHENa Option Optimization Tool supports automatic generation of results suitable for uploading to the database

Ranking View (1)

ATHENA

AUTOMATED TOOL FOR HARDWARE EVALUATION

Authenticated Encryption FPGA Ranking

Show Help

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

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

Comparison of Result #s 95 and 97		
Platform		
Device Vendor:	Xilinx	Xilinx
Family:	Virtex 7	Virtex 7
Device:	xc7vx485tffg1761-2	xc7vx485tffg1761-2
Timing		
Encryption/Authentication Throughput [Mbits/s]:	3261	3015
Decryption/Authentication Throughput [Mbits/s]:	3261	3015
Authentication-Only Throughput [Mbits/s]:	3261	3015
Synthesis Clock Frequency [MHz]:	-	-
Key Scheduling Time [ns]:	-	-
Requested Synthesis Clock Frequency [MHz]:	-	-
Requested Implementation Clock Frequency [MHz]:	-	-
Implementation Clock Frequency [MHz]:	280.27	282.65
(Encryption/Authentication Throughput)/LUT [(Mbits/s)/LUT]:	0.909	0.879
(Encryption/Authentication Throughput)/Slice [(Mbits/s)/Slice]:	2.797	2.728
(Decryption/Authentication Throughput)/LUT [(Mbits/s)/LUT]:	0.909	0.879
(Decryption/Authentication Throughput)/Slice [(Mbits/s)/Slice]:	2.797	2.728
(Auth-Only Throughput)/LUT [(Mbits/s)/LUT]:	0.909	0.879
(Auth-Only Throughput)/Slice [(Mbits/s)/Slice]:	2.797	2.728
Resource Utilization		
CLB Slices:	1166	1105
LUTs:	3588	3430
Flip Flops:	-	-
DSPs:	0	0
BRAMs:	0	0

Conclusions

- High-level synthesis offers a potential to facilitate hardware benchmarking during the design of cryptographic algorithms and at the early stages of cryptographic contests

- Case study based on 13 Round 1 CAESAR candidates & AES-GCM demonstrated correct ranking for majority of candidates using all major performance metrics

- More research & development needed to overcome remaining difficulties
    - Suboptimal control unit of HLS implementations
    - 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

Thank you!

Comments?

Questions?

Suggestions?

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