Benchmarking of Round 3 CAESAR Candidates in Hardware: Methodology, Designs & Results

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

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

Outline

- CAESAR Hardware API & the Compliant Code Development
- Overview of Submitted Designs
- Use Cases
- Benchmarking Methodology
- Results
- ATHENa Database of Results
- Conclusions

CAESAR
Hardware API

CAESAR Hardware API: ePrint 2016/626

Specifies:

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

Enhances:

- Compatibility
- Fairness

Timeline:

- Officially approved by the CAESAR Committee on May 6, 2016
- Last revised on May 12, 2016
- Posted on ePrint on June 17, 2016

URL: https://eprint.iacr.org/2016/626

Addendum to the CAESAR Hardware API

Specifies:

- Minor change to supported maximum size of AD/plaintext/ciphertext
- Clarification regarding the Length segment
- Recommended interface of two-pass algorithms
- Recommended support for two maximum lengths of AD/plaintext/ciphertext in case of single-pass algorithms

Enhances:

- Compatibility between implementations of the same algorithm
- Fairness in comparing single-pass vs. two-pass algorithms

Timeline:

- Last revised on June 10, 2016
- Officially approved by the CAESAR Committee on Nov 24, 2016

URL: https://cryptography.gmu.edu/athena/CAESAR_HW_APICAESAR_HW_API_v1.0_Addendum.pdf

GMU Development Package

Development Package:

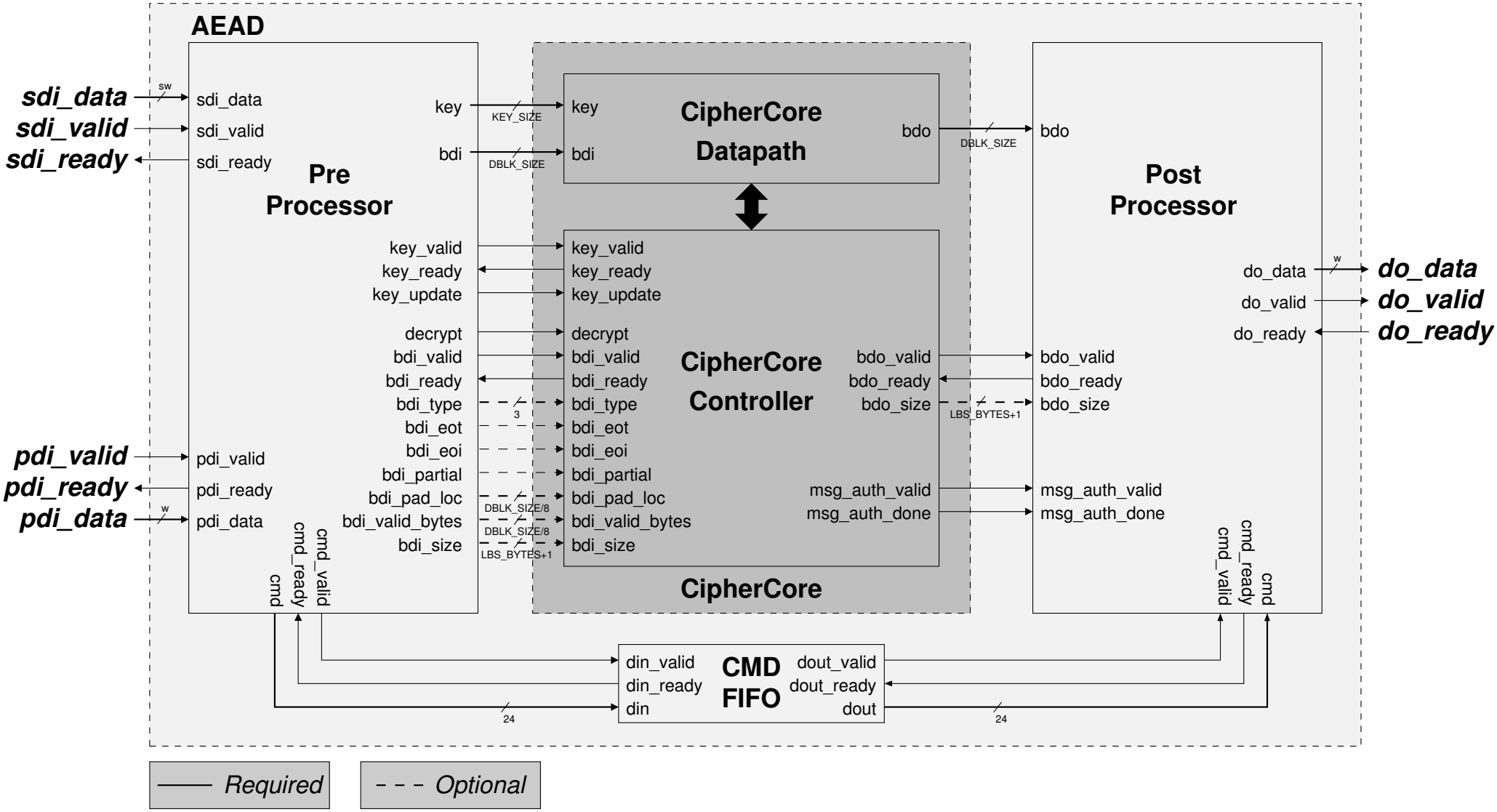
a. VHDL code of a generic PreProcessor, PostProcessor, and CMD FIFO, common for all Round 2 and Round 3 CAESAR Candidates (except Keyak) as well as AES-GCM (src_rtl)
b. Universal testbench common for all the API compliant designs (AEAD_TB)
c. Python app used to automatically generate test vectors (aeadtvgen)
d. Reference implementations of Dummy authenticated ciphers (dummyN)

Last Update: June 10, 2016

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

New, enhanced version under development

Top-level block diagram of a High-Speed architecture



GMU Implementer's Guide

a. Proposed Top-Level Block Diagram
b. Development of High-Speed vs. Lightweight Implementations
c. Configuration of the top-level entity, AEAD
d. CipherCore Development for High-Speed Implementations
e. Test Vector Generation
f. Simulation
g. Generation of Results

Last Update: June 10, 2016

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

New, enhanced version under development

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. Formulas for the execution time
f. Verification method (test vectors)
g. Block diagrams (optional)
h. License (optional)
i. Preliminary results (optional)

CAESAR Hardware API vs. GMU Development Package

CAESAR Hardware API:

1) Approved by the CAESAR Committee, **stable**

2) Necessary for **fairness** and **compatibility**

3) Obligatory

GMU Development Package:

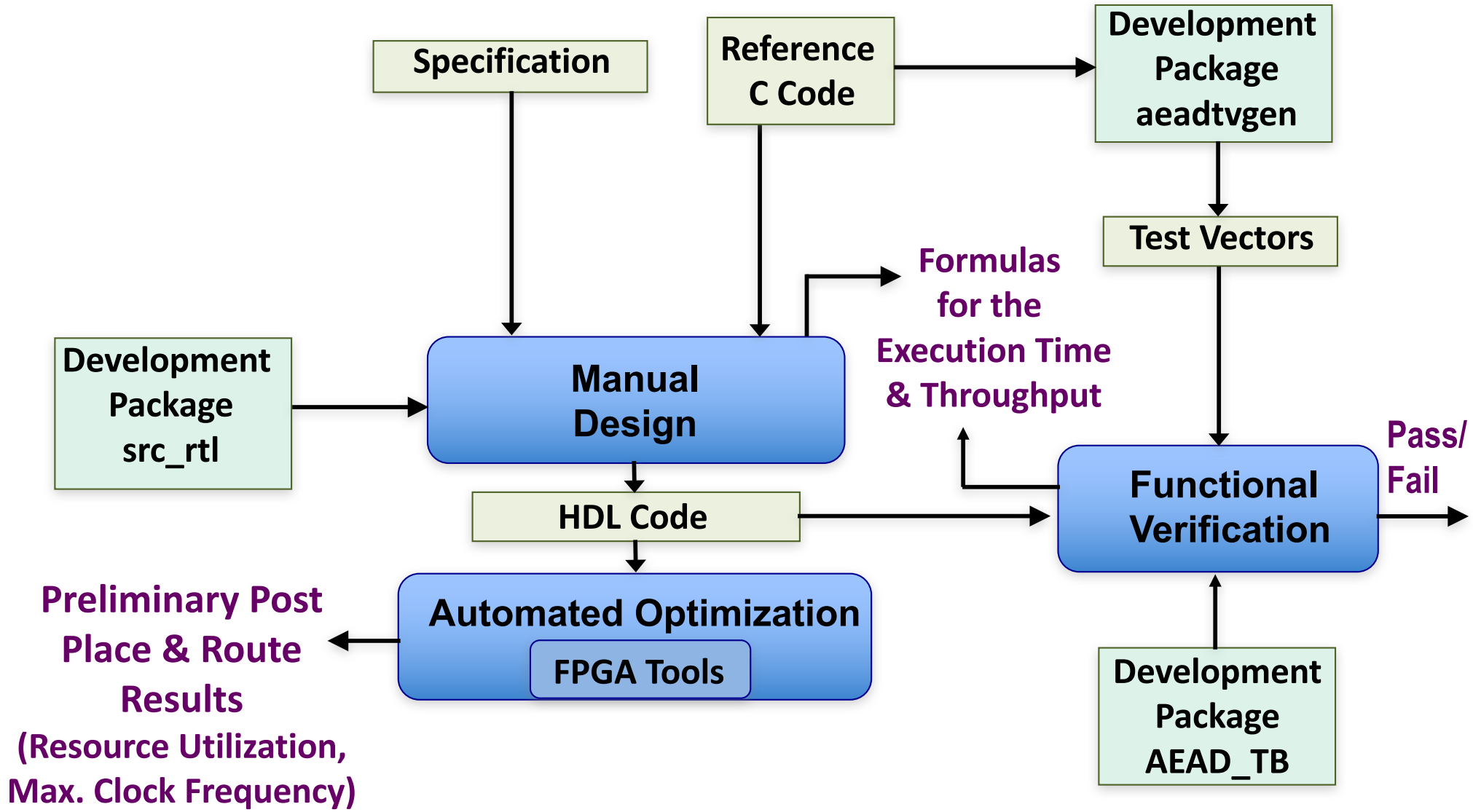
1) First version published in May 2016, gradually **evolving**

2) Recommended in order to reduce the **development time**

3) Totally optional

The API Compliant Code Development

The API Compliant Code Development



Overview of Submitted Designs

Round 3 VHDL/Verilog Submitters

1. CERG GMU - AEGIS, AEZ, Ascon, CLOC-AES, COLM, Deoxys-I, JAMBU-AES, NORX, OCB, SILC-AES, Tiaoxin (11)
2. CCRG NTU Singapore – ACORN, AEGIS, JAMBU-SIMON, MORUS (4)
3. CLOC-SILC Team, Japan – CLOC-AES, CLOC-TWINE, SILC-AES, SILC-LED/PRESENT (4)
5. Ketje-Keyak Team – Ketje x 2 & Keyak (3)
6. NEC Japan – AES-OTR
7. IAIK TU Graz, Austria – Ascon
8. CINVESTAV-IPN, Mexico – COLM
9. Axel Y. Poschmann and Marc Stöttinger – Deoxys-I & Deoxys-II
10. NTU Singapore – Deoxys-I

Total: 27 submissions

Summary of VHDL/Verilog Submissions

- 2 Compliant Submissions + 1 Non-Compliant Submission
  - 1: Deoxys-I
- 2 Compliant submissions
  - 4: AEGIS, CLOC-AES, COLM, SILC-AES
- 1 Compliant Submission + 1 Non-Compliant Submission
  - 2: Ascon, Ketje
- 1 Compliant Submission
  - 11: ACORN, AES-OTR, AEZ, CLOC-TWINE, JAMBU-AES, JAMBU-SIMON, MORUS, NORX, OCB, SILC-LED/PRESENT, Tiaoxin
- 1 Partially Compliant Submission
  - 1: Keyak
- 1 Non-Compliant Submission
  - 1: Deoxys-II

Non-Compliant Implementations (1)

Ascon (by IAIK TU Graz)

- Included countermeasures against side-channel attacks
- Custom interface (including random masks, narrow data in/data out/ key/tag buses, custom command inputs)
- No support for the CAESAR HW API Protocol

[not benchmarked]

Ketje (by the Ketje-Keyak Team)

- Custom interface aimed at more compact hardware (no SDI port, custom control inputs, such as go, auth_data, data, tag, tag_p_one, last, hash, squeeze, din_size, etc.)
- No support for the CAESAR HW API Protocol

[not benchmarked]

Non-Compliant Implementations (2)

Deoxys-I and Deoxys-II (by Axel York Poschmann & Marc Stöttinger)

- Missing non-optional ports of CipherCore
- Use of gated clock, not recommended in the FPGA technology
- Implementations targeting ASIC tools, incompatible with FPGA tools
  - Xilinx ISE trims about 90% of the circuit resources (including one of the clock signals), reports more than 1000 warnings
  - Xilinx Vivado reports hundreds of timing loops

[not benchmarked]

Partially Compliant Implementation

Keyak (by the Ketje-Keyak Team)

- Compliance criteria:
  - supported maximum size for AD should be 2^{32}-1 bytes
- Implementation:
  - supported maximum size for AD is 24 bytes

[treated as compliant in the database of results]

Variant vs. Architecture

- Two different variants of the same algorithm produce different outputs for the same input
  (e.g., they differ in terms of the key/nonce/tag size)

- Two different architectures of a specific variant produce the same output, but differ in terms of performance and/or resource utilization
  (e.g., basic iterative and unrolled x2 architectures)

Architectures

- Majority of algorithms have designs based on

  Basic Iterative Architecture (One Round per Clock Cycle)

Exceptions:

ACORN (NTU):	8bit & 32bit lightweight
AEGIS (NTU):	Folded /8v
AES-OTR (NEC):	Unrolled x2
COLM (CINVESTAV-IPN):	Quasi-pipelined
Deoxys-I (NTU):	4-stream pipelined
Deoxys-I (GMU):	Basic iterative with speculative pre-computation
JAMBU-SIMON:	Unrolled x4

Ciphers vs. Variants

For the purpose of benchmarking:

- CLOC and SILC are treated as separate ciphers, rather than variants
- JAMBU-AES and JAMBU-SIMON are treated as separate ciphers, rather than variants
- Each cipher may have multiple variants, e.g.
  - KetjeJr, KetjeSr, KetjeMinor, and KetjeMajor
  - CLOC-AES and CLOC-TWINE
  - NORX64-4-1, NORX32-4-1, NORX64-6-1, NORX32-6-1
- In the ranking graphs, **each cipher is represented by only one variant** with the best value of a particular performance metric used for ranking

Other Factors Affecting Comparison

- Key sizes
- Security properties
  (lightweight vs. non lightweight,
  single-pass vs. two-pass,
  nonce misuse resistance, etc.)
- Nonce sizes
- Tag and/or authenticator sizes
- PDI & DO port width, w

Key sizes

- Majority of the implemented ciphers support 128-bit keys only

  Exceptions:
  - CLOC-TWINE, SILC-LED, SILC-PRESENT: 80
  - JAMBU-SIMON, KetjeJr: 96
  - Deoxys-I, Deoxys-II, NORX: 128 & 256
  - AEZ: 384

Possible allowed key ranges:

$|K| \geq 80$

- covers all families

$|K| \geq 128$

- excludes lightweight variants with 80 and 96-bit keys

PDI & DO Ports Width, w

- The CAESAR API Minimum Compliance Criteria allow
  - High-speed: $32 \leq w \leq 256$
  - Lightweight: $w = 8, 16, 32$

- Majority of the API compliant implementations support w=32 or w=64 only

  Exceptions:
  - ACORN: 8 & 32
  - JAMBU-SIMON: 48
  - KetjeMinor: 128
  - NORX: 128 & 256
  - AEGIS, KetjeMajor, MORUS, Tiaoxin: 256

Use Cases

Use Cases

Use Case 1: Lightweight applications (resource constrained environments)

- Critical: fits into **small hardware area** and/or small code for 8-bit CPUs

Use Case 2: High-performance applications

- Critical: efficiency on 64-bit CPUs (servers) and/or **dedicated hardware**

Use Case 3: Defense in depth

- Critical: authenticity despite nonce misuse

Use Case 1 Variants

ACORN: acorn128v3

Ascon: ascon128av12, ascon128v12

CLOC: aes128n12t8clocv3 = aes128n12t8clocv2

aes128n8t8clocv3 = aes128n8t8clocv2

twine80n6t4clocv3 = twine80n6t4clocv2 [no Xilinx FPGA results yet]

JAMBU: jambusimon96v2

Ketje: ketjejrv2, ketjesrv2, ketjeminorv2

NORX: norx3241v3, norx3261v3

SILC: aes128n12t8silcv3 = aes128n12t8silcv2

led80n6t4silcv3 = led80n6t4silcv2 [no Xilinx FPGA results yet]

present80n6t4silcv3 = present80n6t4silcv2

[no Xilinx FPGA results yet]

Lightweight Features of Implementations of the Use Case 1 Variants

Candidate	Variant	w	sw	Architecture
ACORN	acorn128v3	8 & 32	8 & 32	8-bit & 32-bit
Ascon	ascon128av12	32	32	Basic Iterative
	ascon128v12	32	32	Basic Iterative
CLOC	aes128n12t8clocv3	32	32	Basic Iterative
	aes128n8t8clocv3	32	32	Basic Iterative
	twine80n6t4clocv3	64	40	Basic Iterative
JAMBU	jambusimon96v2	48	48	Basic Iterative
Ketje	ketjejrv2	32	32	Basic Iterative
	ketjesrv2	32	32	Basic Iterative
	ketjeminorv2	128	128	Basic Iterative
NORX	norx3241v3	128	32	Basic Iterative
	norx3261v3	128	32	Basic Iterative
SILC	aes128n12t8silcv3	32	32	Basic Iterative
	led80n6t4silcv3	64	40	Basic Iterative
	present80n6t4silcv3	64	40	Basic Iterative

Implementations of the Use Case 1 Variants Compliant with the CAESAR HW API

Candidate	Variant	w	sw	Architecture
ACORN	acorn128v3	8 & 32	8 & 32	8-bit & 32-bit
Ascon	ascon128av12	32	32	Basic Iterative
	ascon128v12	32	32	Basic Iterative
CLOC	aes128n12t8clocv3	32	32	Basic Iterative
	aes128n8t8clocv3	32	32	Basic Iterative
Ketje	ketjejrv2	32	32	Basic Iterative
	ketjesrv2	32	32	Basic Iterative
SILC	aes128n12t8silcv3	32	32	Basic Iterative

CAESAR Hardware API requires that the lightweight implementations have

w = 8, 16, or 32 (pdi and do bus width)

sw = 8, 16, or 32 (sdi bus width)

No specific architecture is required by the API, however, architectures with extended resource sharing (compared to the Basic Iterative) are likely to achieve significantly lower area

Additional Developments Required for Use Case 1

- New version of the **GMU Development Package** with the lightweight versions of the PreProcessor & PostProcessor **[at the final stages of development]**
- New version of the **GMU Implementer's Guide [to be released soon]**
- **Lightweight implementations** of all Use Case 1 variants with

  w = 8, 16, or 32

  sw = 8, 16, or 32

  Extended resource sharing compared to the Basic Iterative architecture.
- **Power and energy per bit** estimated by the tools and measured experimentally
- Natural **resistance to side-channel attacks** evaluated
- **Countermeasures against side channel attacks** (such as threshold implementations) developed and their effectiveness evaluated
- **Penalty in terms of area, throughput, power, and energy per bit** determined using FPGA tools and experimental setup

Use Case 2 Variants

AEGIS:	aegis128, aegis128l
AES-OTR:	aes128otrcv3
	aes128otrpv3 = aes128otrpv2
	aes128otrsv3 = aes128otrsv2
Ascon:	ascon128av12, ascon128v12
Deoxys-I:	deoxysi128v141, deoxysi256v141
Ketje:	ketjemajorv2
MORUS:	morus1280128v2
NORX:	norx6441v3, norx6461v3
OCB:	aeadaes128ocbtaglen128v1
Tiaoxin:	tiaoxinv2

Use Case 3 Variants

AEZ:	aezv5
COLM:	colm0v1
Deoxys-II:	deoxysii128v141, deoxysii256v141
	[no compliant implementation available]
JAMBU:	aesjambuv2=jambuaes128v2
Keyak:	lakekeyakv2, riverkeyakv2

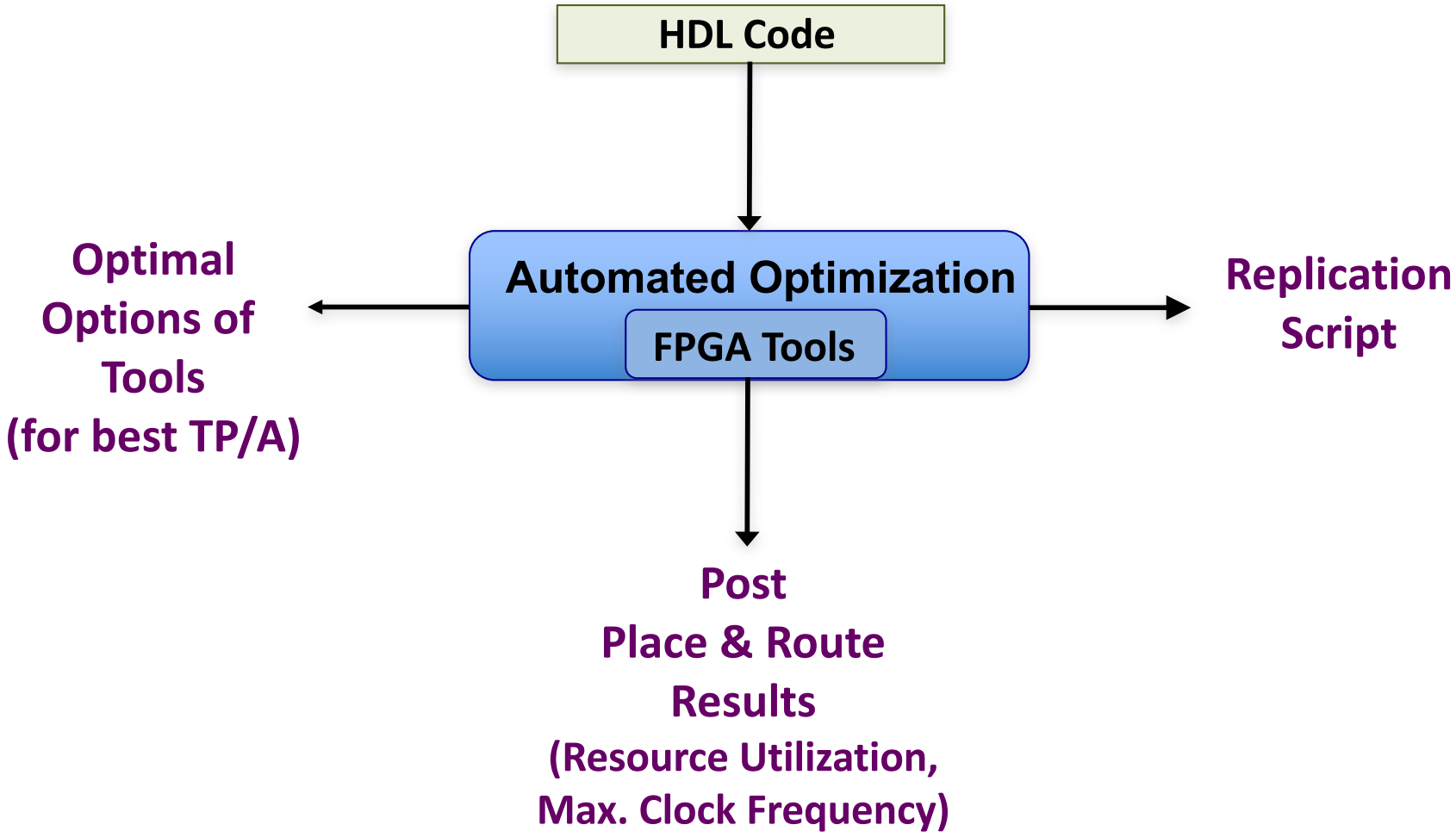
Warning: Candidates in this Use Case differ substantially in terms of their enhanced security features

Benchmarking Methodology

FPGA Families & Devices Used for Benchmarking

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

RTL Benchmarking



FPGA Tools (1)

For Benchmarking Targeting Xilinx FPGAs (other than Virtex-7):

Target FPGAs:	Virtex-6
Synthesis Tool:	Xilinx XST 14.7
Implementation Tool:	Xilinx ISE 14.7
Automated Optimization:	ATHENa

For Benchmarking Targeting Altera FPGAs:

Target FPGAs:	Stratix IV, Stratix V
Synthesis Tool:	Quartus Prime 16.0.0
Implementation Tool:	Quartus Prime 16.0.0
Automated Optimization:	ATHENa

FPGA Tools (2)

For Benchmarking Targeting Xilinx Virtex-7 FPGAs:

Target FPGAs:	**Virtex-7**
Synthesis Tool:	**Xilinx Vivado 2015.1**
Implementation Tool:	**Xilinx Vivado 2015.1**
Automated Optimization:	**Minerva**

ATHENa – Automated Tool for Hardware EvaluatioN

- Open-source
- Written in Perl
- Developed 2009-2012
- FPL Community Award 2010
- Automated search for optimal
  - Options of tools
  - Target frequency
  - Starting placement point
- Supporting Xilinx ISE, Altera Quartus

No support for Xilinx Vivado

Extension of ATHENa to Vivado: Minerva

- **Programming language:**
  Python
- **Target synthesis and implementation tool:**
  Xilinx Vivado Design Suite
- **Supported FPGA families:**
  All Xilinx 7 series and beyond
- **Optimization criteria:**
  1. Maximum frequency
  2. Frequency/#LUTs
  3. Frequency/#Slices

Expected release for use by other groups – September 2017

Embedded Memories & DSP Units

- No embedded memories and no embedded DSP units allowed inside of
  - AEAD: for single-pass algorithms, and
  - AEAD-TP: for two-pass algorithms

- Their use eliminated using options of the respective tools
  (including, if necessary, the synthesis tool directives added to HDL code)

- **Without** this approach
  - Area = Resource Utilization Vector
    e.g. Area = (1056 Slices, 4 BRAMs, 67 DSP units)
  - No known way of comparing FPGA Resource Utilization Vectors
  - No way of calculating Throughput/Area

- **Additional Benefit**
  - Good correlation of the obtained results with the corresponding ASIC results, as demonstrated during the SHA-3 Competition.
    See http://eprint.iacr.org/2012/368, Section 9

Dealing with I/O Ports

- No wrappers used
- Ports of
    - AEAD: for single-pass algorithms, and
    - AEAD-TP: for two-pass algorithms,

  connected directly to the I/O pins of a target FPGA

Results

Performance Metrics

Use Cases 2 & 3

Primary:

- Throughput/Area
- Throughput

Secondary:

- Area

Use Case 1

Primary:

- Area
- Throughput/Area

Secondary:

- Throughput

Throughput Types

- **Authenticated Encryption Throughput**
  - **primary throughput reported in all graphs**
- **Authenticated Decryption Throughput**
  - Different only for
    - **Deoxys-I & Deoxys-II (by Axel & Marc)**
      [not reported due to non-compliance]
- **Authentication-Only Throughput**
  - Different only for
    - **AEZ** [2.5x greater]
    - **CLOC-AES & SILC-AES (by CLOC-SILC Team)** [1.9x greater]
    - **Deoxys-I & Deoxys-II (by Axel & Marc)**
      [not reported due to non-compliance]

Area Units

For Xilinx FPGAs:

Target FPGAs: Virtex-6, Virtex-7

Units of Area: LUTs (Look-up Tables)

Slices (1 Slice contains 4 LUTs,
8 registers & additional logic)

For Altera FPGAs:

Target FPGAs: Stratix IV, Stratix V

Units of Area: ALUTs (Adaptive Look-up Tables)

ALM (Adaptive Logic Modules)
(Stratix IV ALM contains 2 adaptive ALUTs,
2 registers & additional logic
Stratix V ALM contains 2 adaptive ALUTs,
4 registers & additional logic)

Included in High-Speed Rankings

Ciphers & Their Variants:

- AES-GCM
- CLOC, SILC
- JAMBU-AES, JAMBU-SIMON
- 13 other Round 3 Candidates

= **18 Ciphers**

- **Key size ≥ 80 bits**

Designs:

- **Only Compliant with the CAESAR Hardware API**
  (including the Partially Compliant design for Keyak with |AD| ≤ 24 bytes)

Relative Results vs. [Absolute] Results

- **Relative Results**
  - Results divided by the corresponding results for AES-GCM, e.g.,

    Relative Throughput of Candidate X = Throughput of Candidate X / Throughput of AES-GCM
  - Represent speed-up, area savings, efficiency improvement compared to AES-GCM
  - No units
  - 17 results reported for All Use Cases (all results for AES-GCM by definition 1)

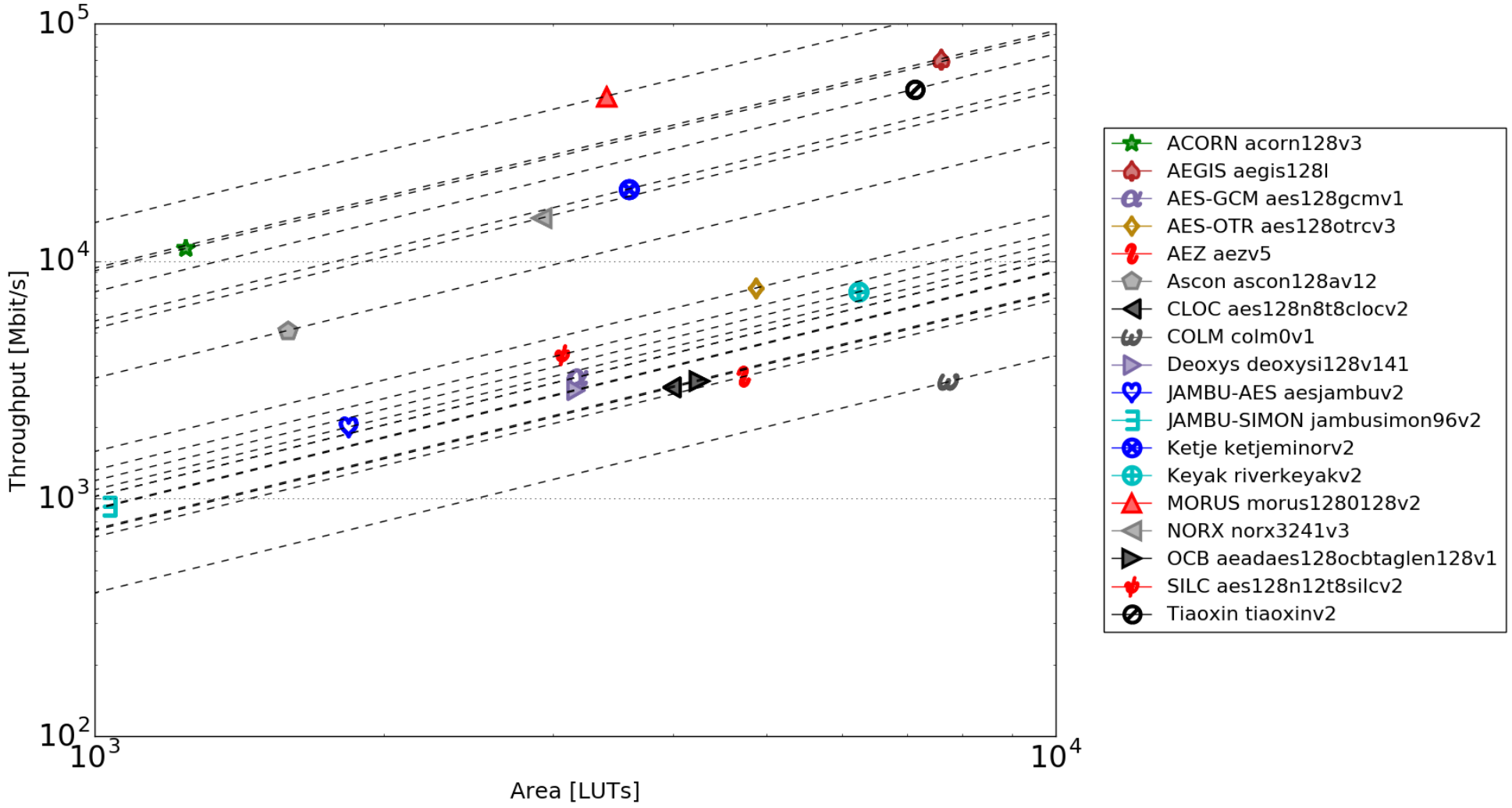
- **[Absolute] Results** ("Absolute" portion in the metric name optional)
  - "Regular" results for each candidate
  - Reported in the ATHENa Database of Results
  - Units appropriate for the given performance metric,

    e.g., Mbit/s for Absolute Throughput

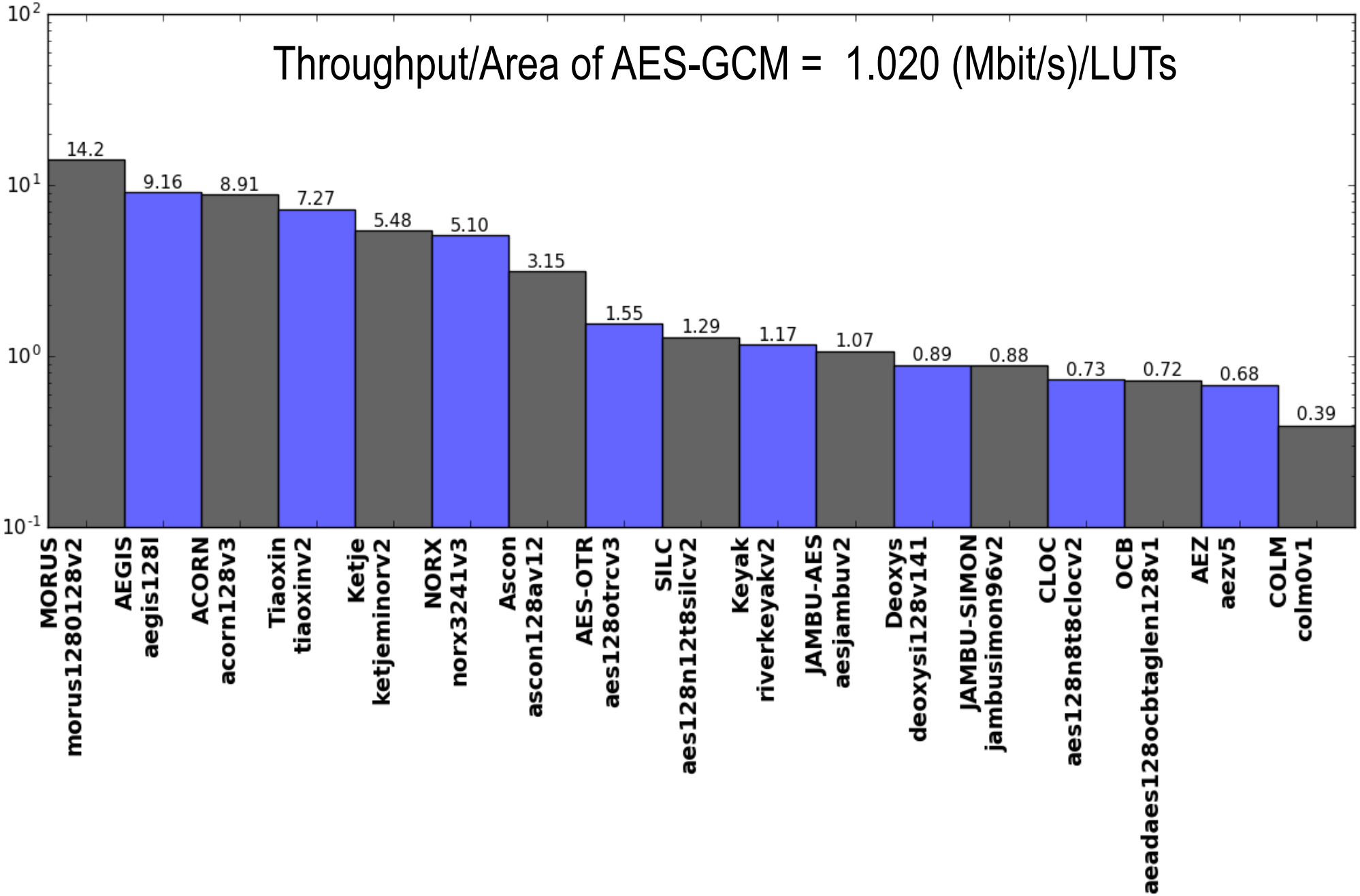
All Use Cases

Virtex-6

Results for Virtex-6 – Throughput vs. Area Logarithmic Scale

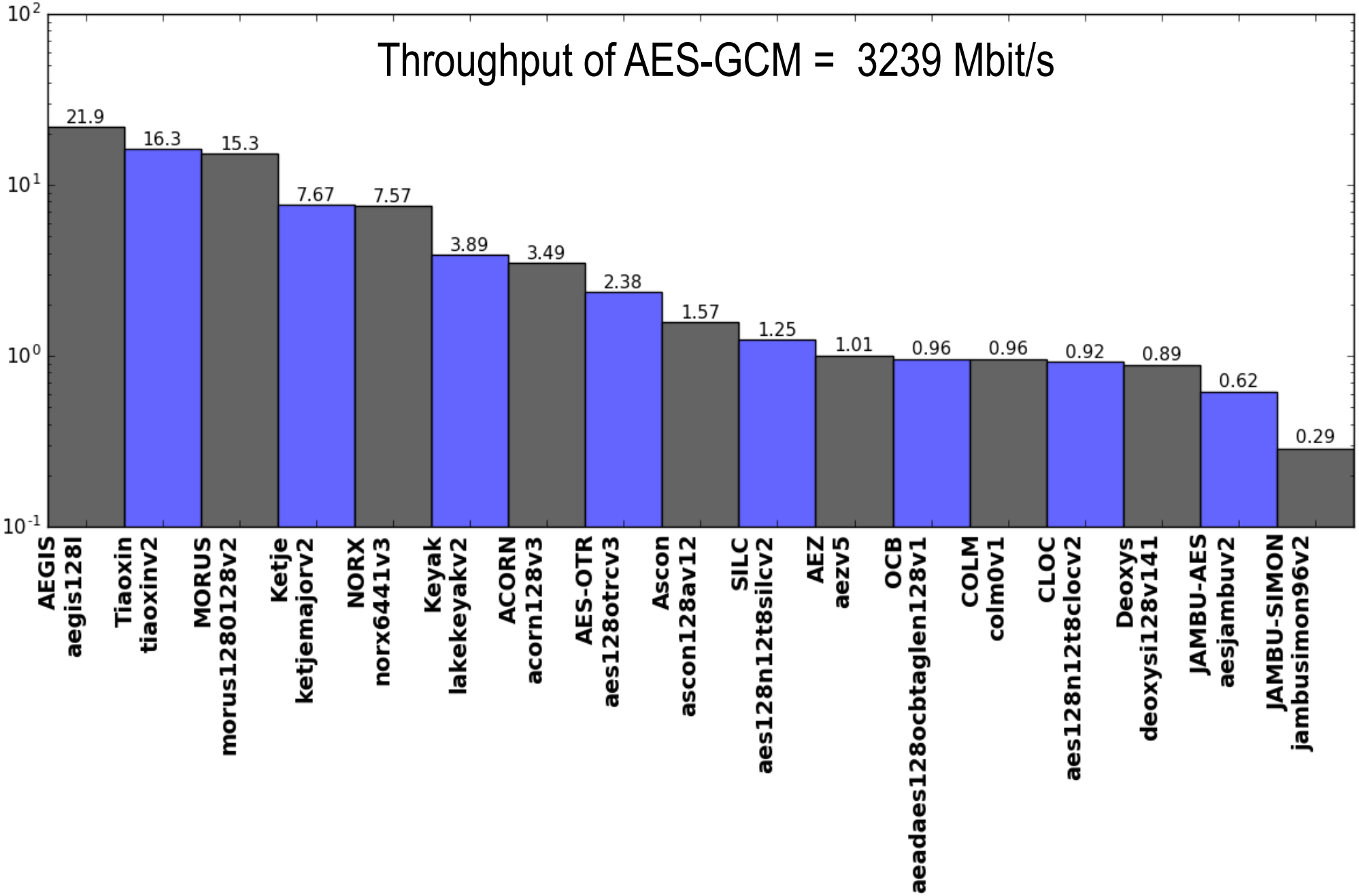


Relative Throughput/Area in Virtex-6 vs. AES-GCM



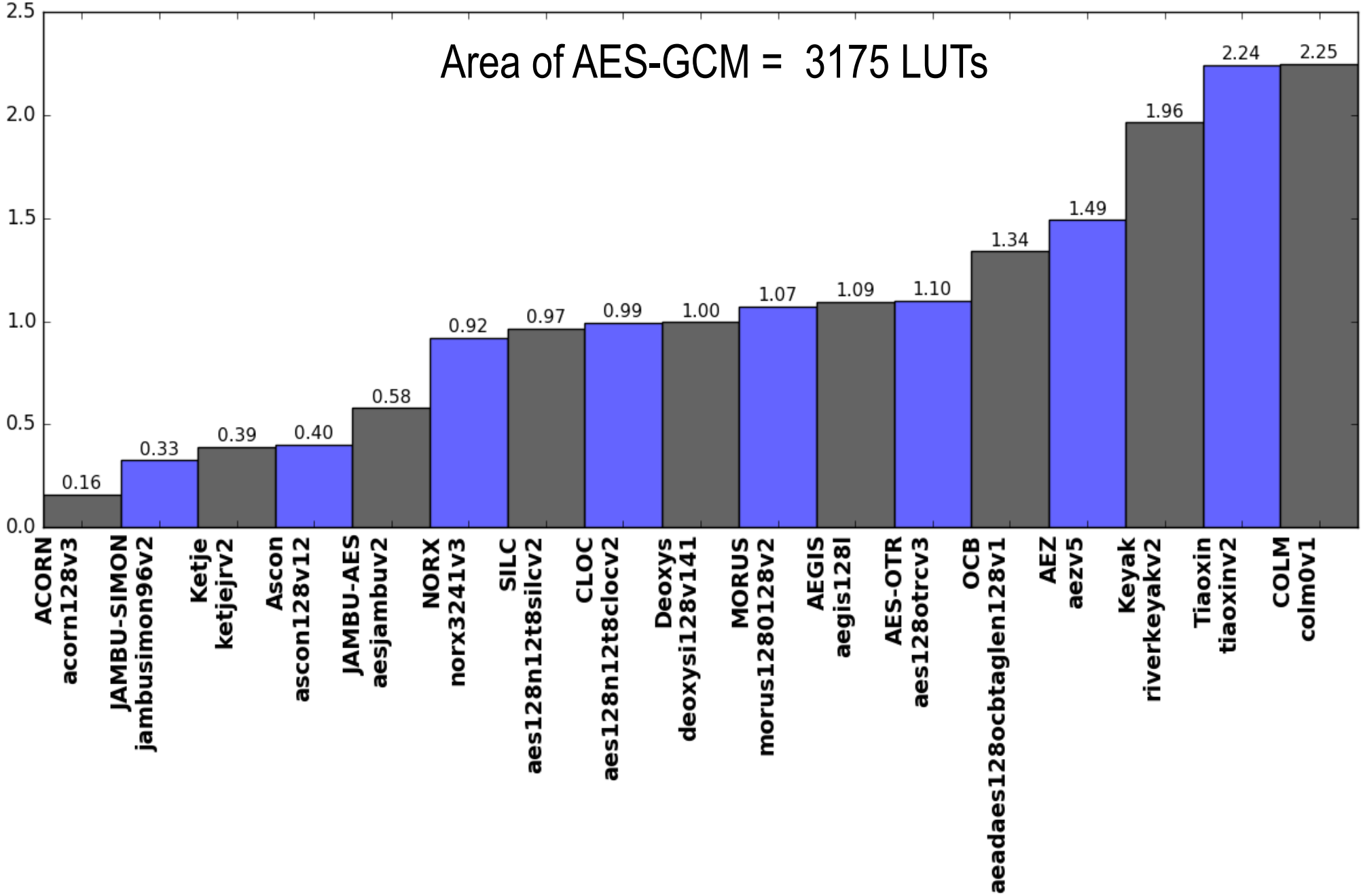
Relative Throughput in Virtex-6

Ratio of a given Cipher Throughput/Throughput of AES-GCM



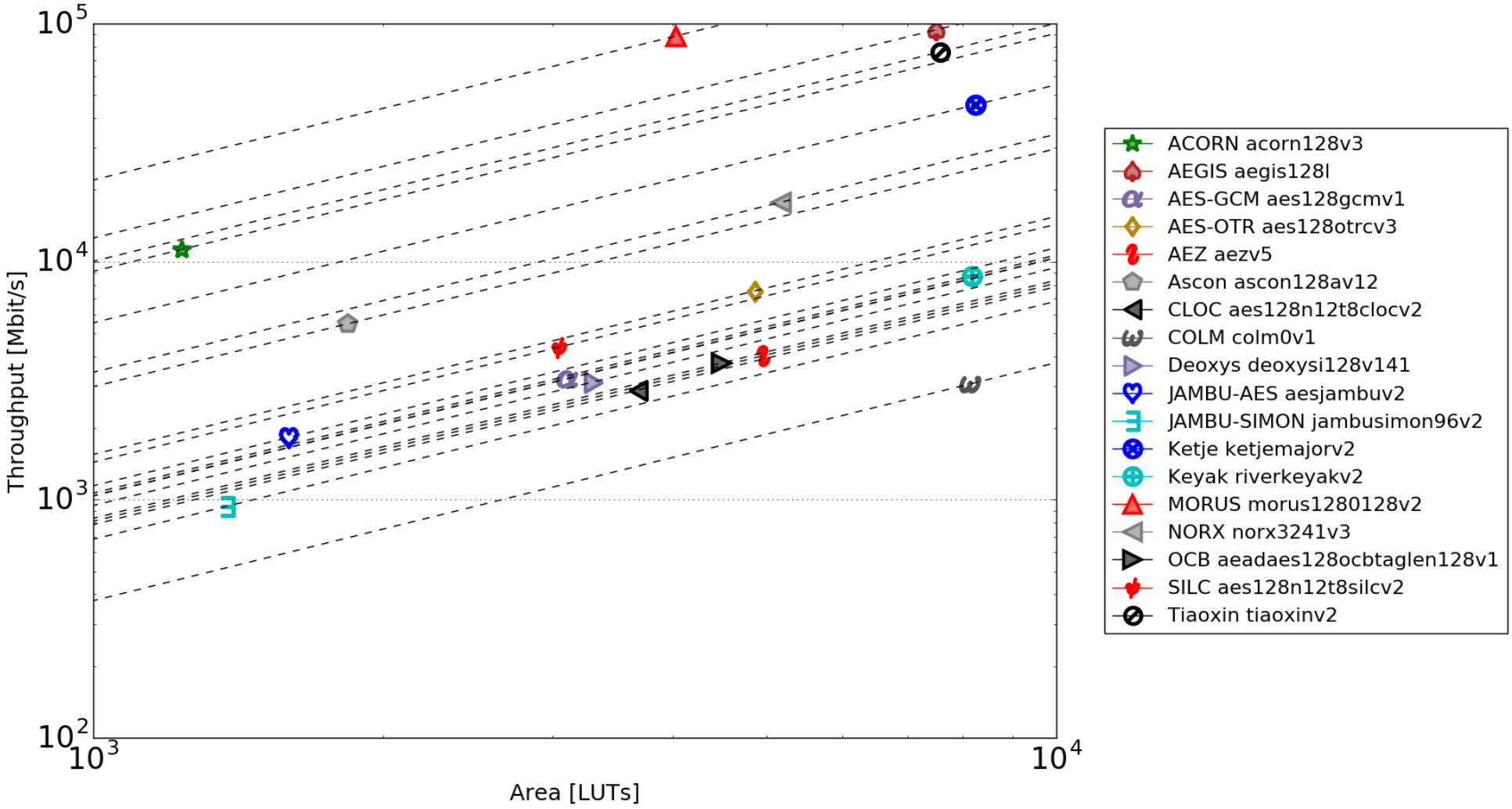
Relative Area (#LUTs) in Virtex-6

Ratio of a given Cipher Area/Area of AES-GCM

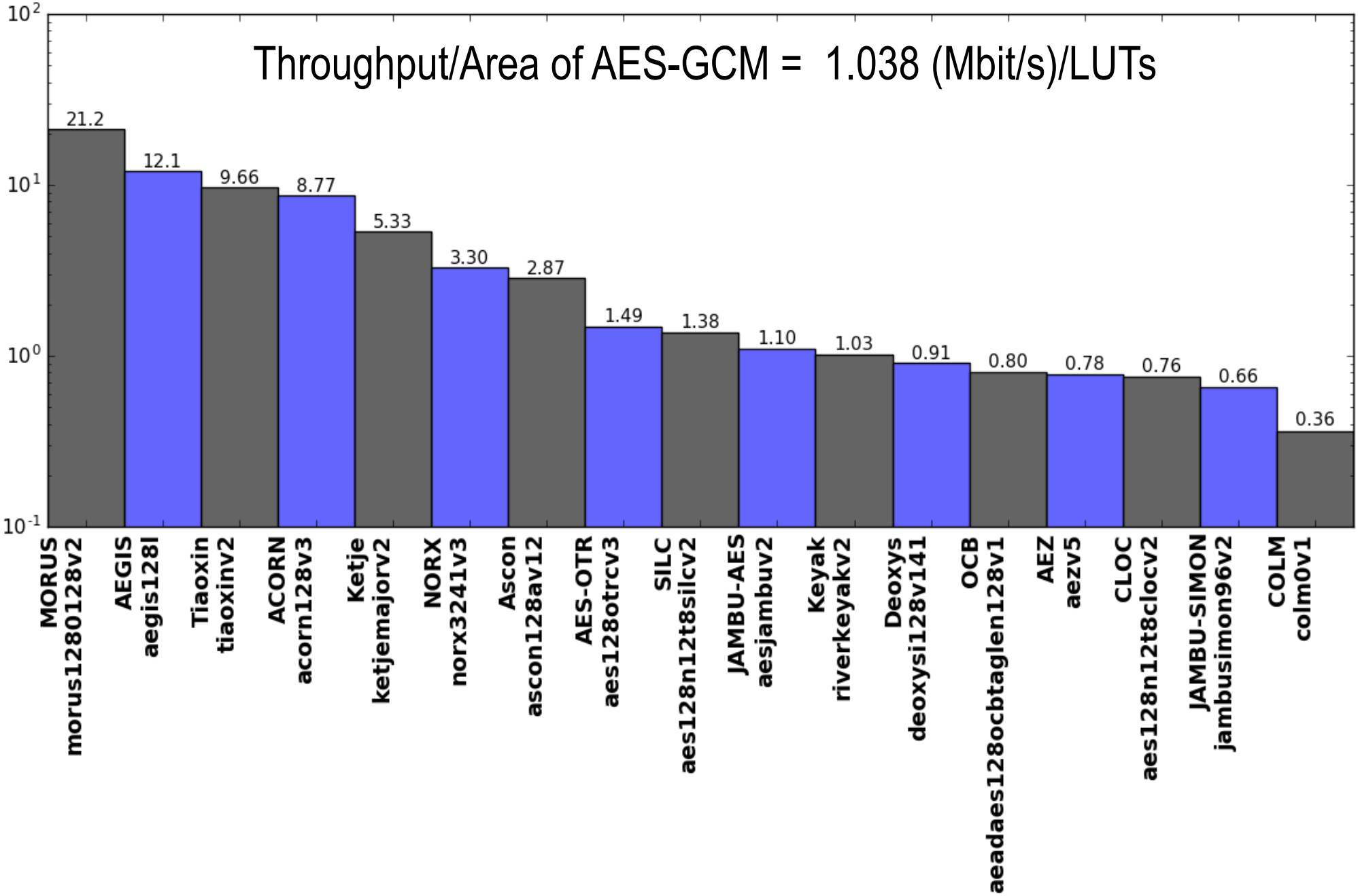


Virtex-7

Results for Virtex-7 – Throughput vs. Area
Logarithmic Scale

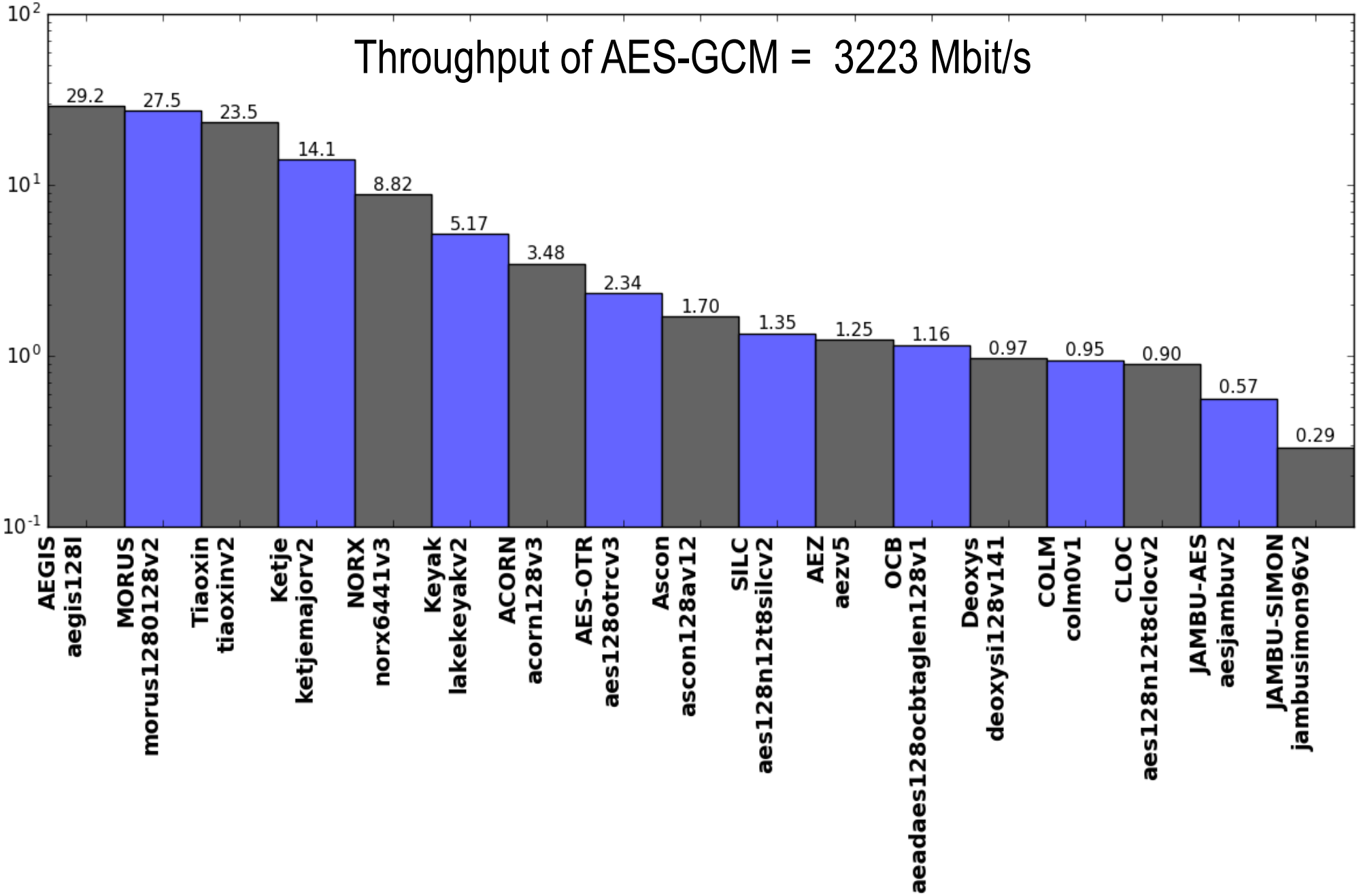


Relative Throughput/Area in Virtex-7 vs. AES-GCM



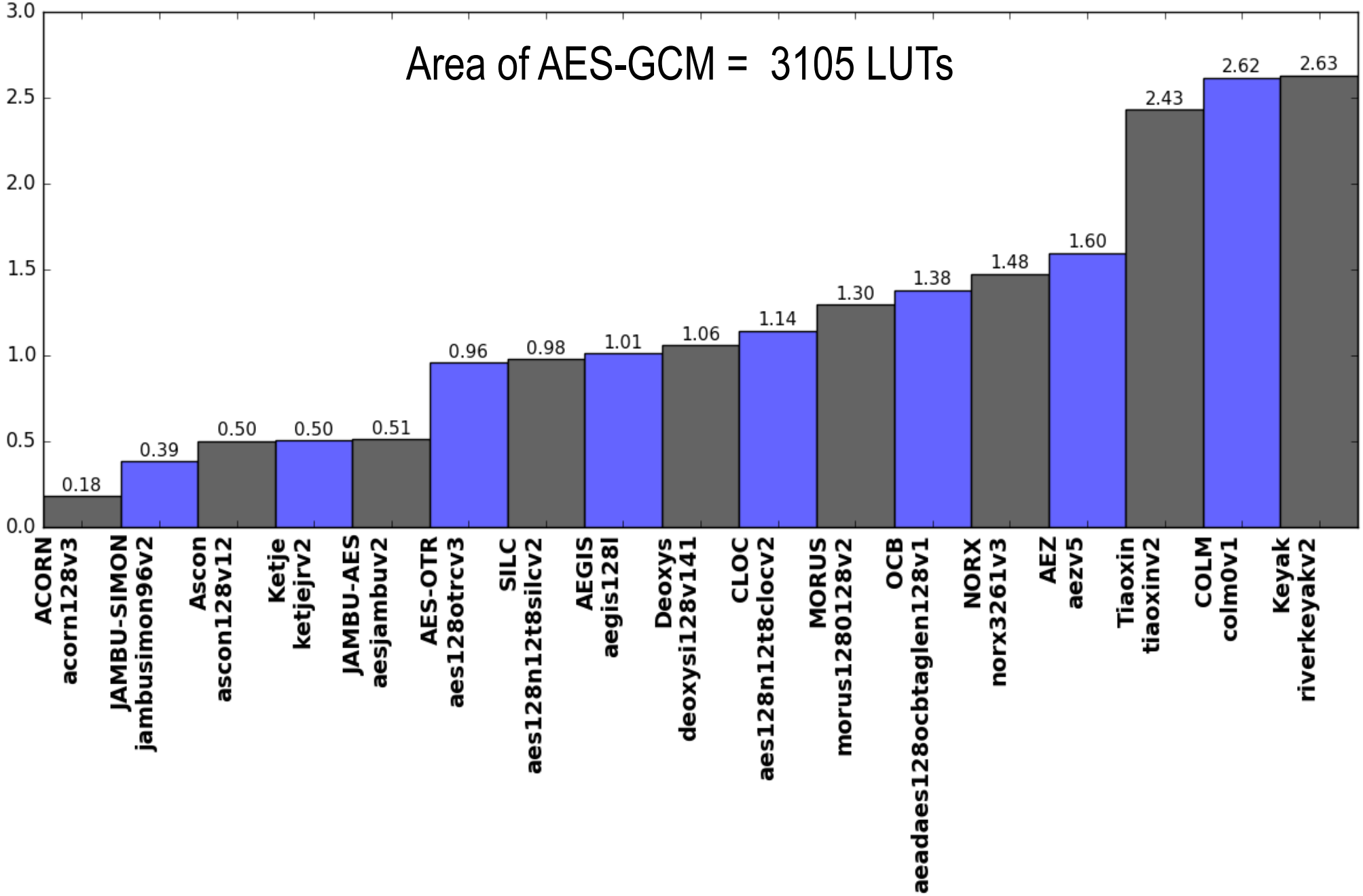
Relative Throughput in Virtex-7

Ratio of a given Cipher Throughput/Throughput of AES-GCM



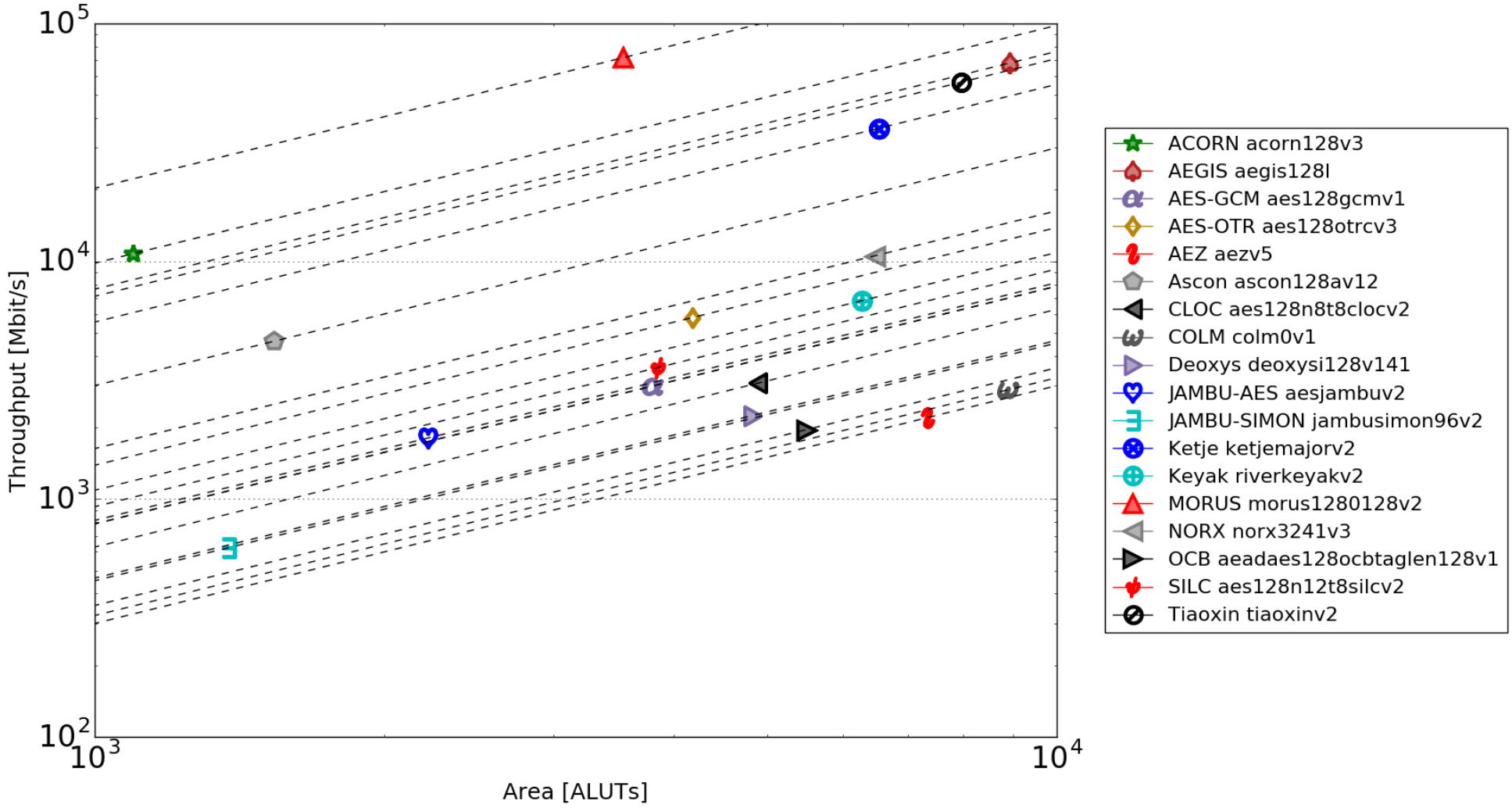
Relative Area (#LUTs) in Virtex-7

Ratio of a given Cipher Area/Area of AES-GCM

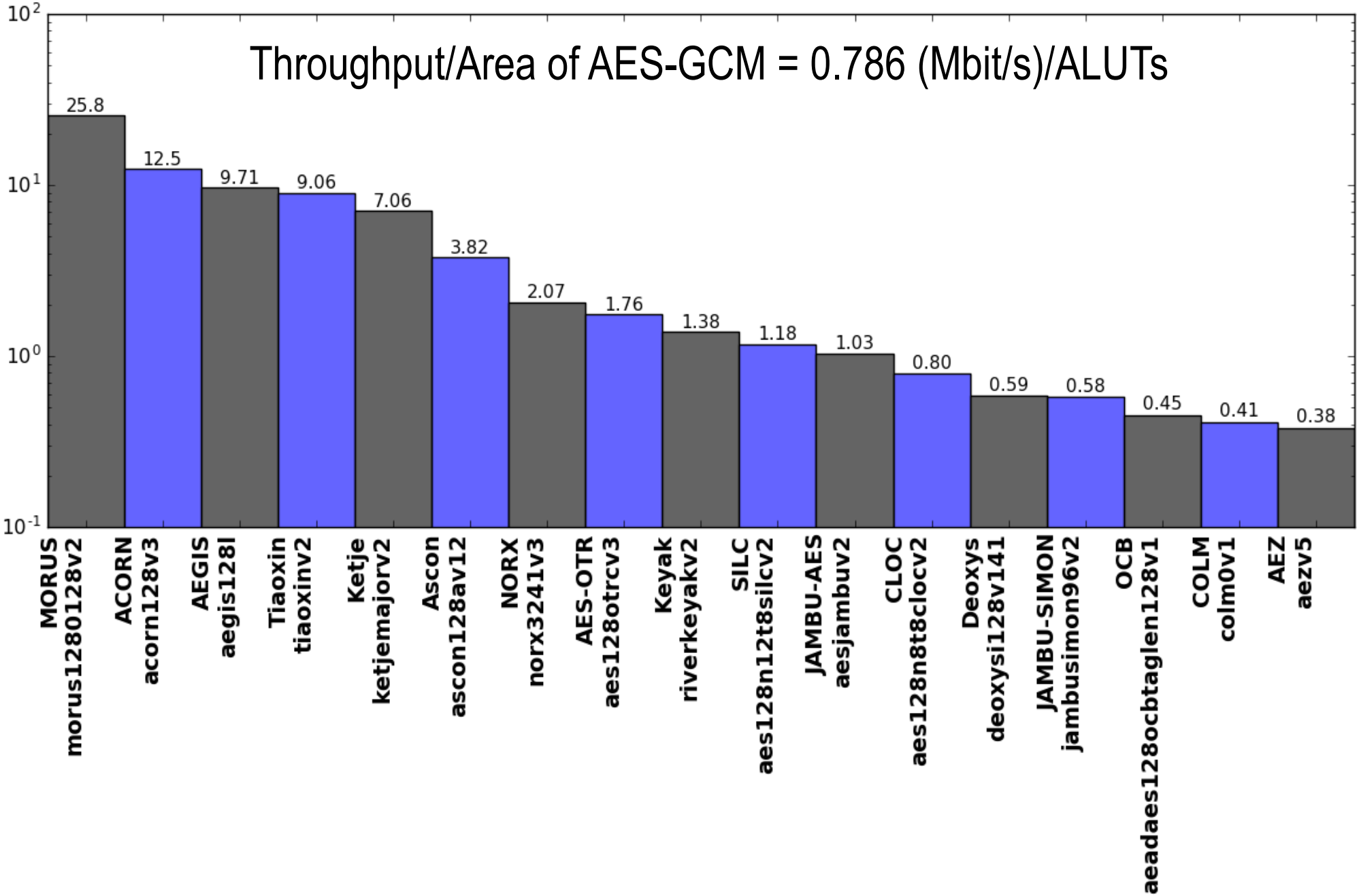


Stratix IV

Results for Stratix IV – Throughput vs. Area Logarithmic Scale

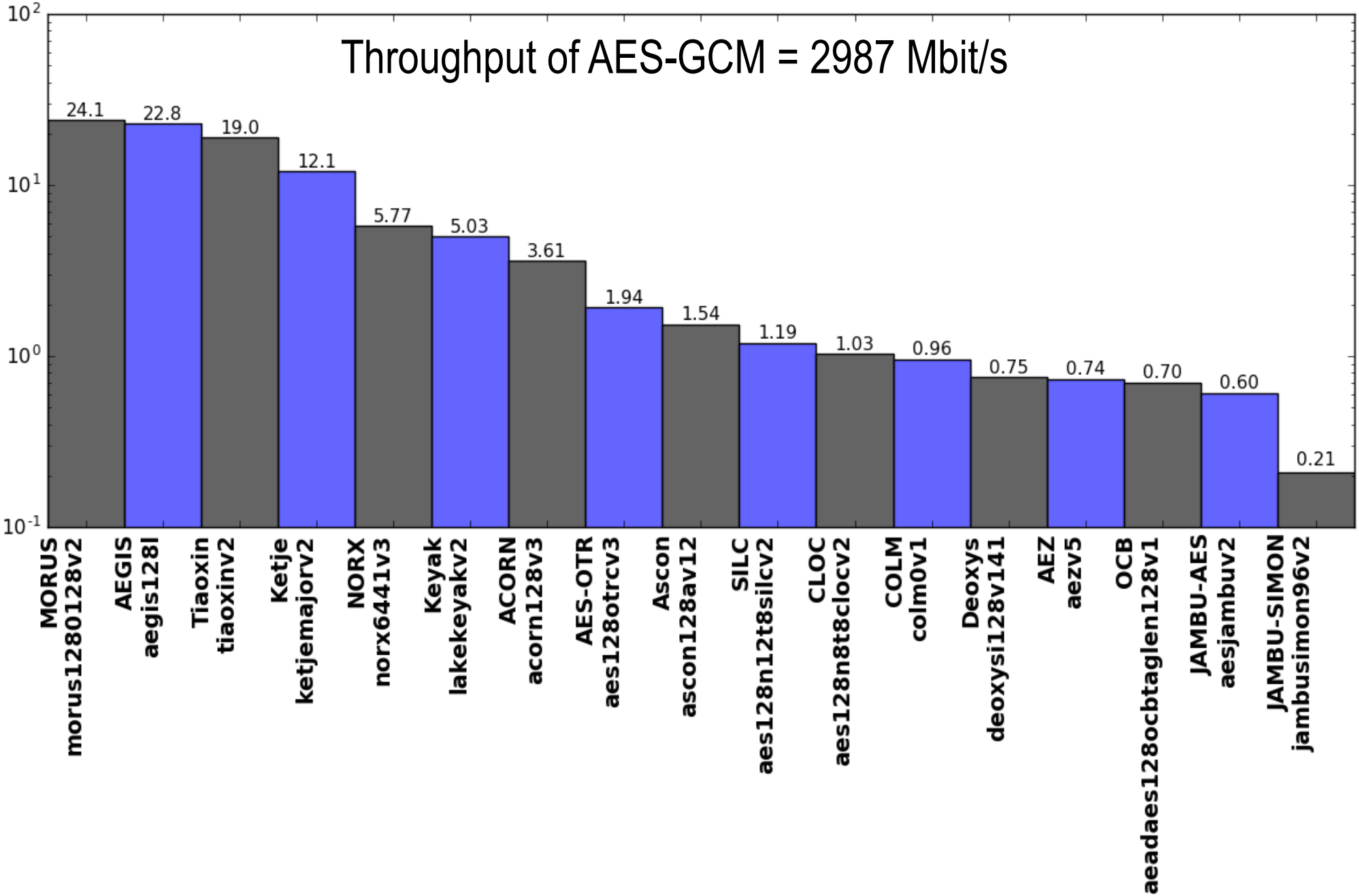


Relative Throughput/Area in Stratix IV vs. AES-GCM



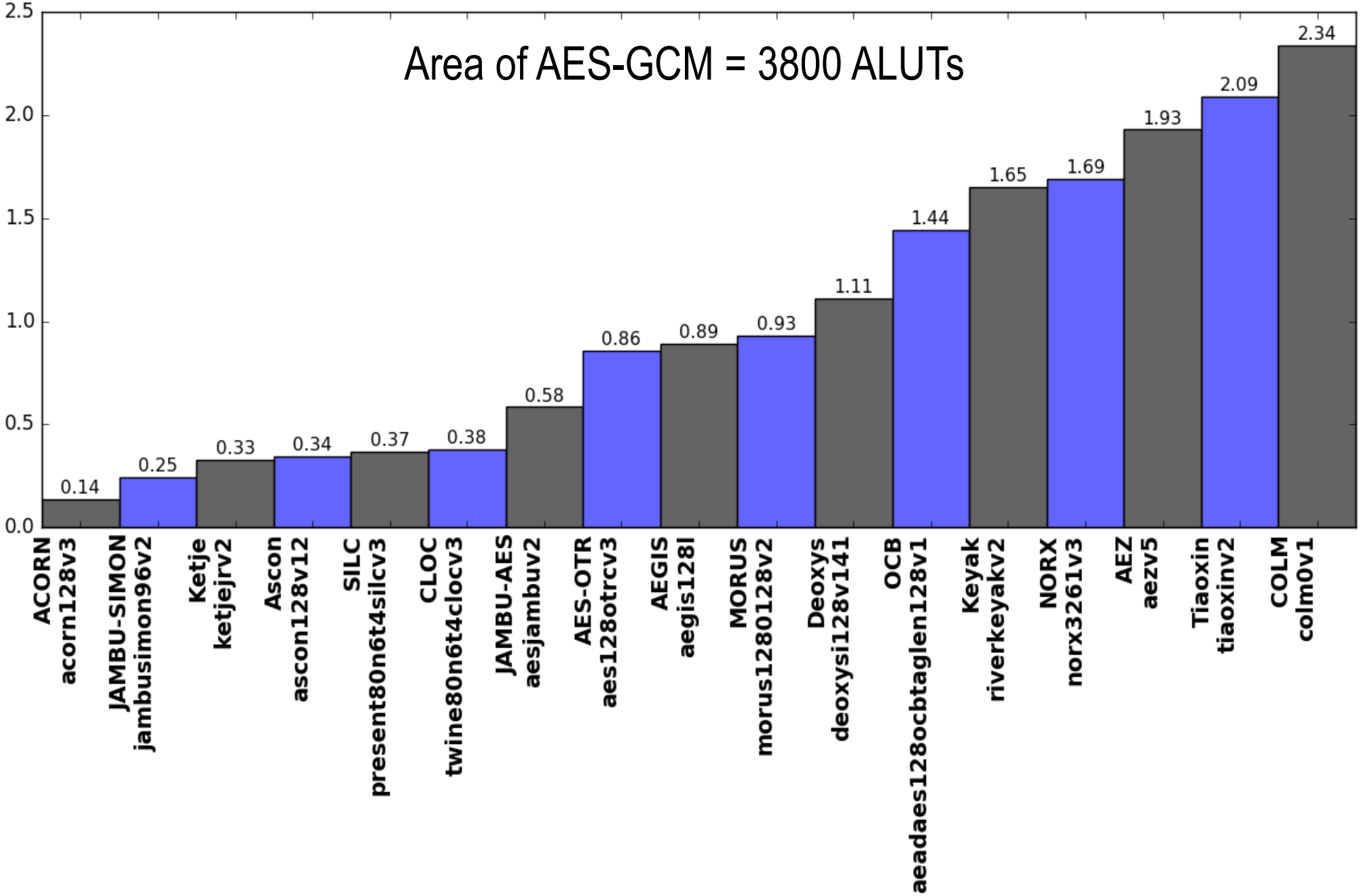
Relative Throughput in Stratix IV

Ratio of a given Cipher Throughput/Throughput of AES-GCM



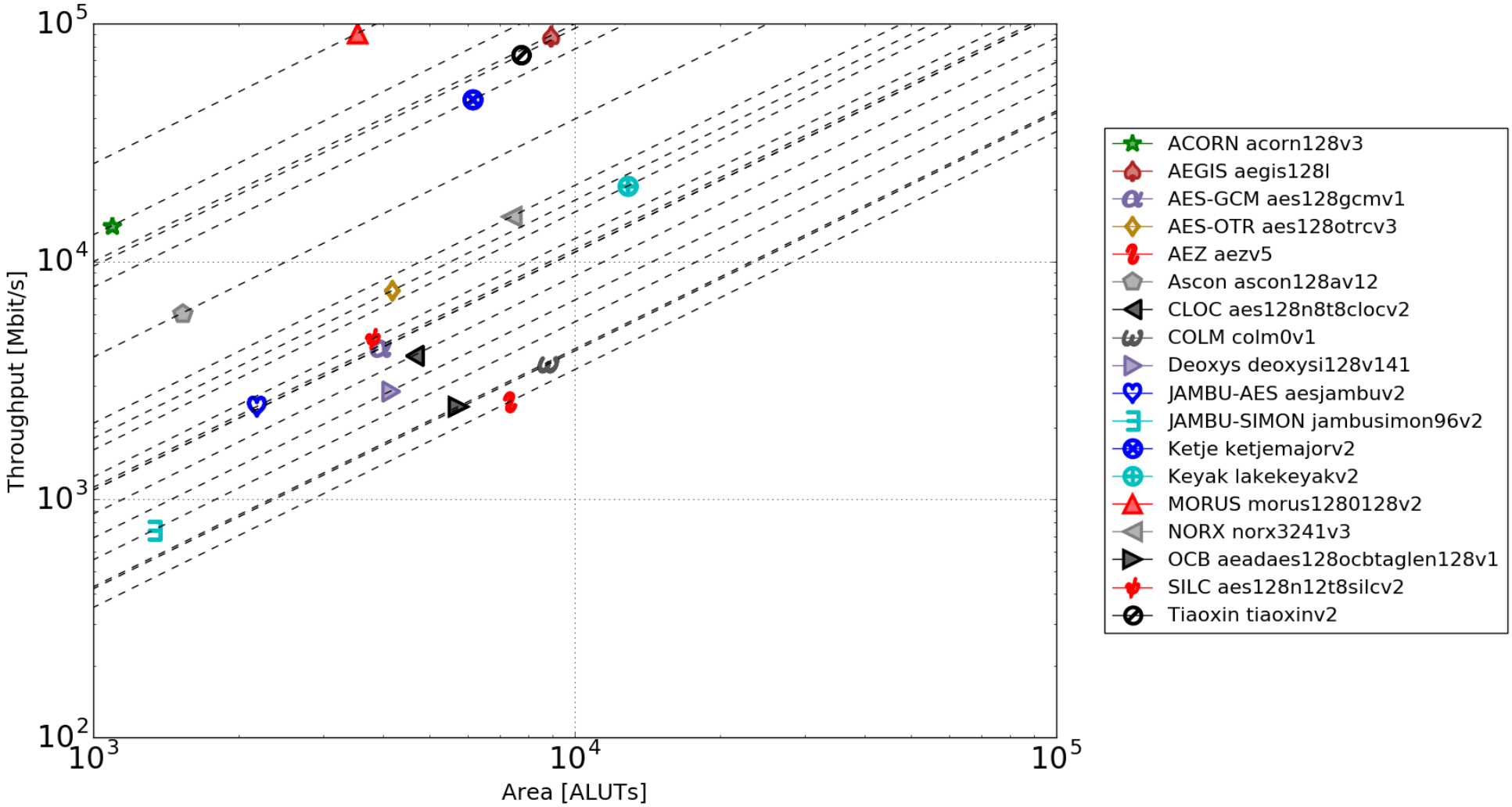
Relative Area (#ALUTs) in Stratix IV

Ratio of a given Cipher Area/Area of AES-GCM

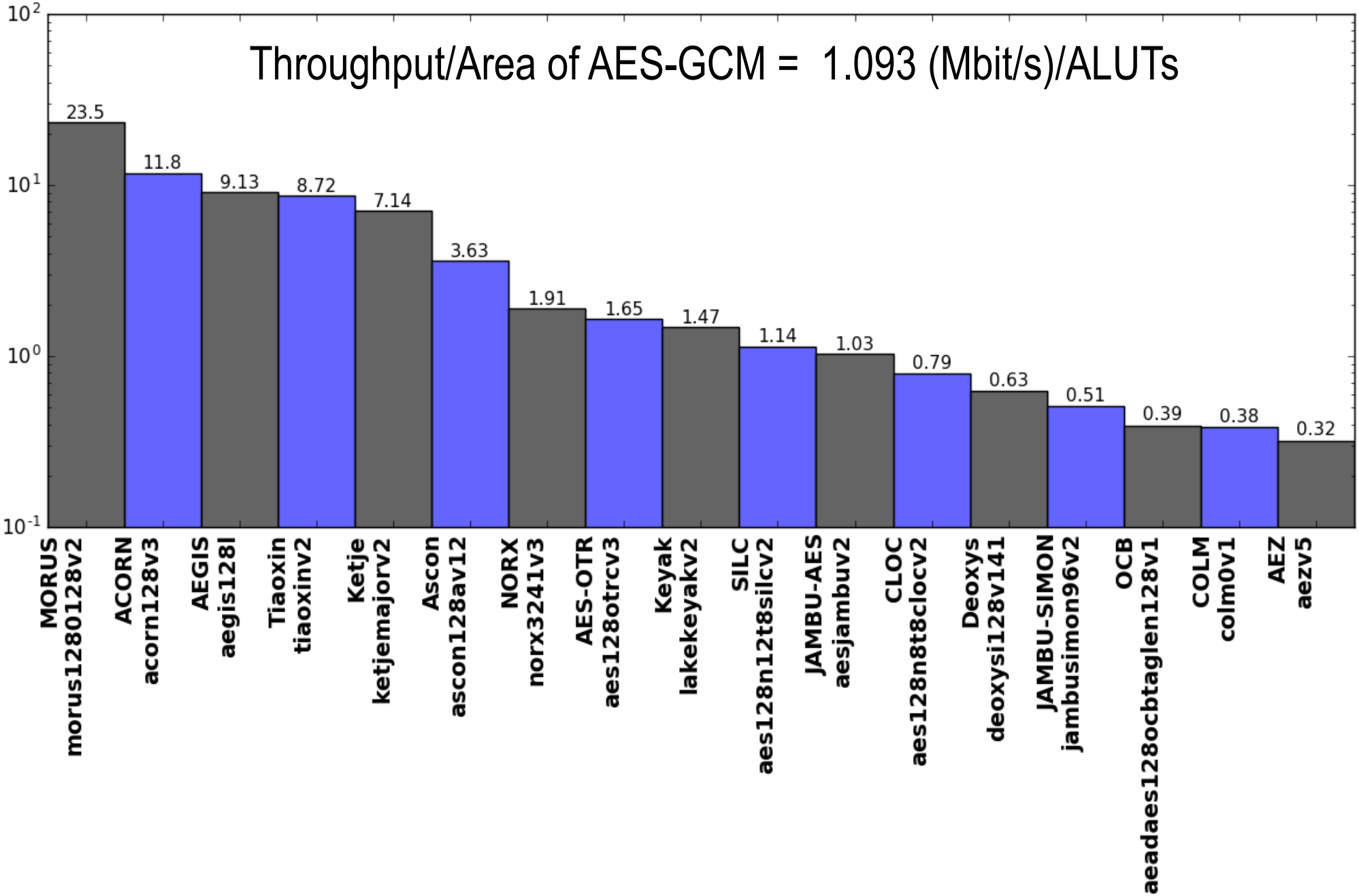


Stratix V

Results for Stratix V – Throughput vs. Area Logarithmic Scale

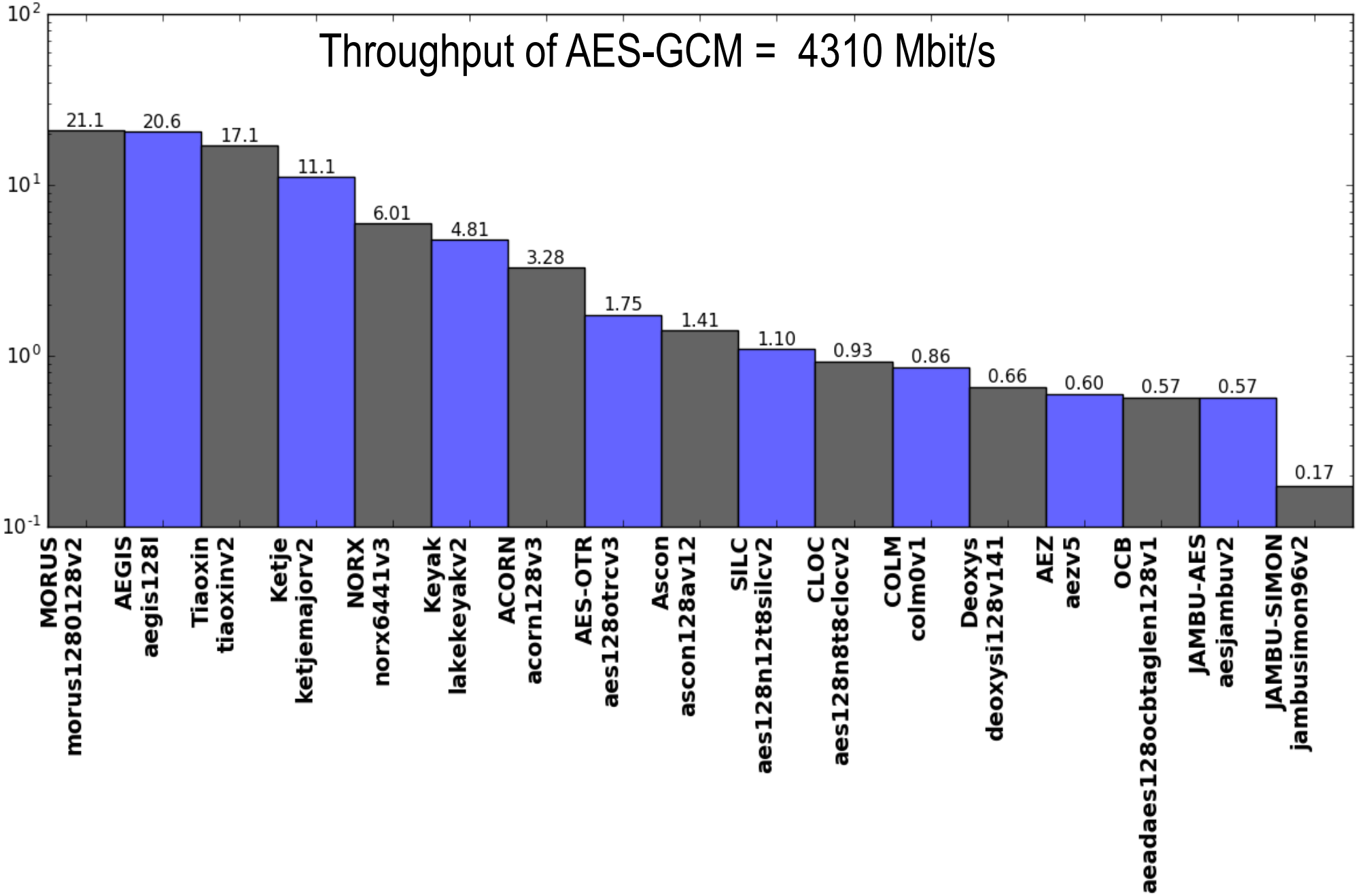


Relative Throughput/Area in Stratix V vs. AES-GCM

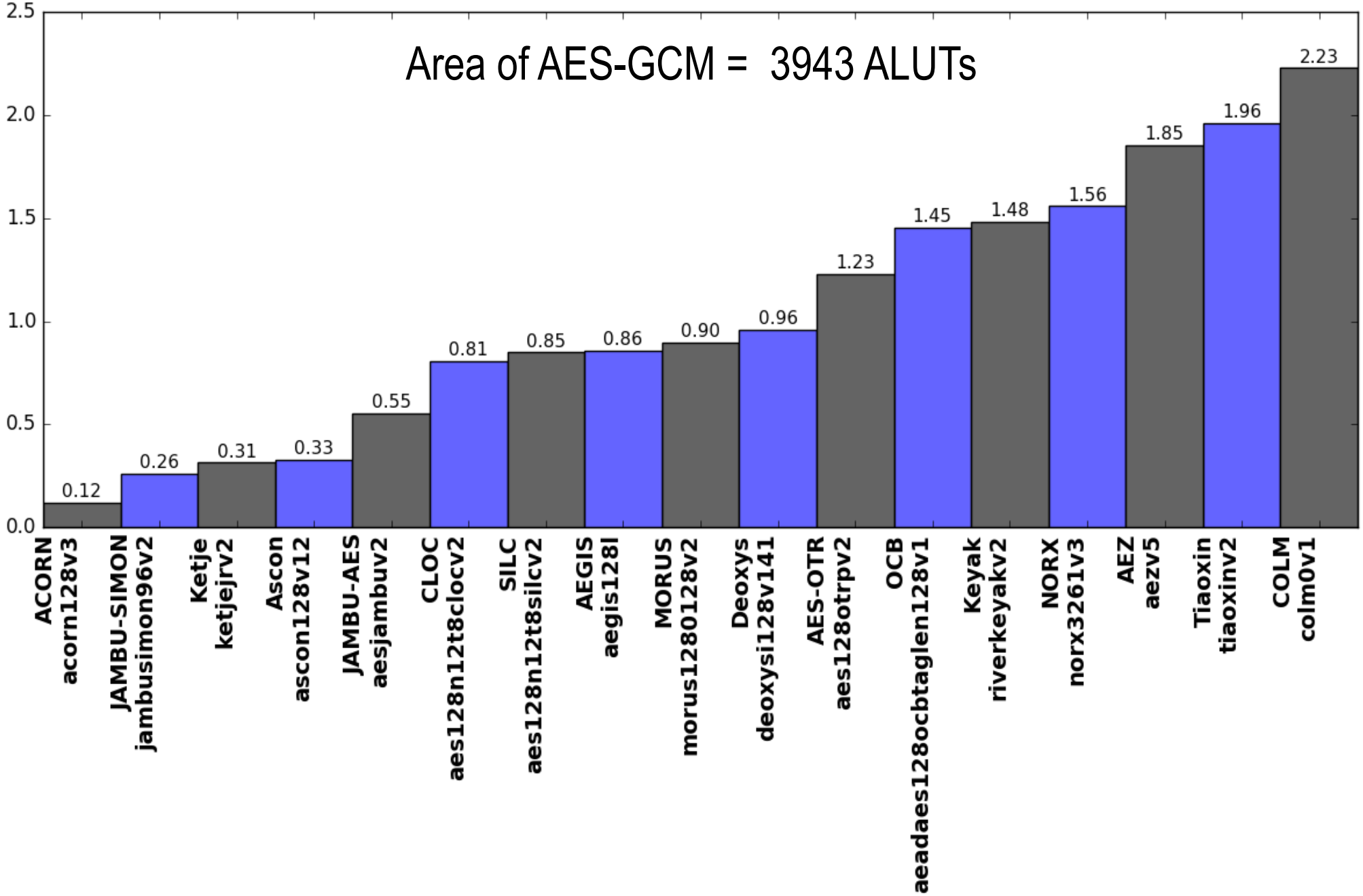


Relative Throughput in Stratix V

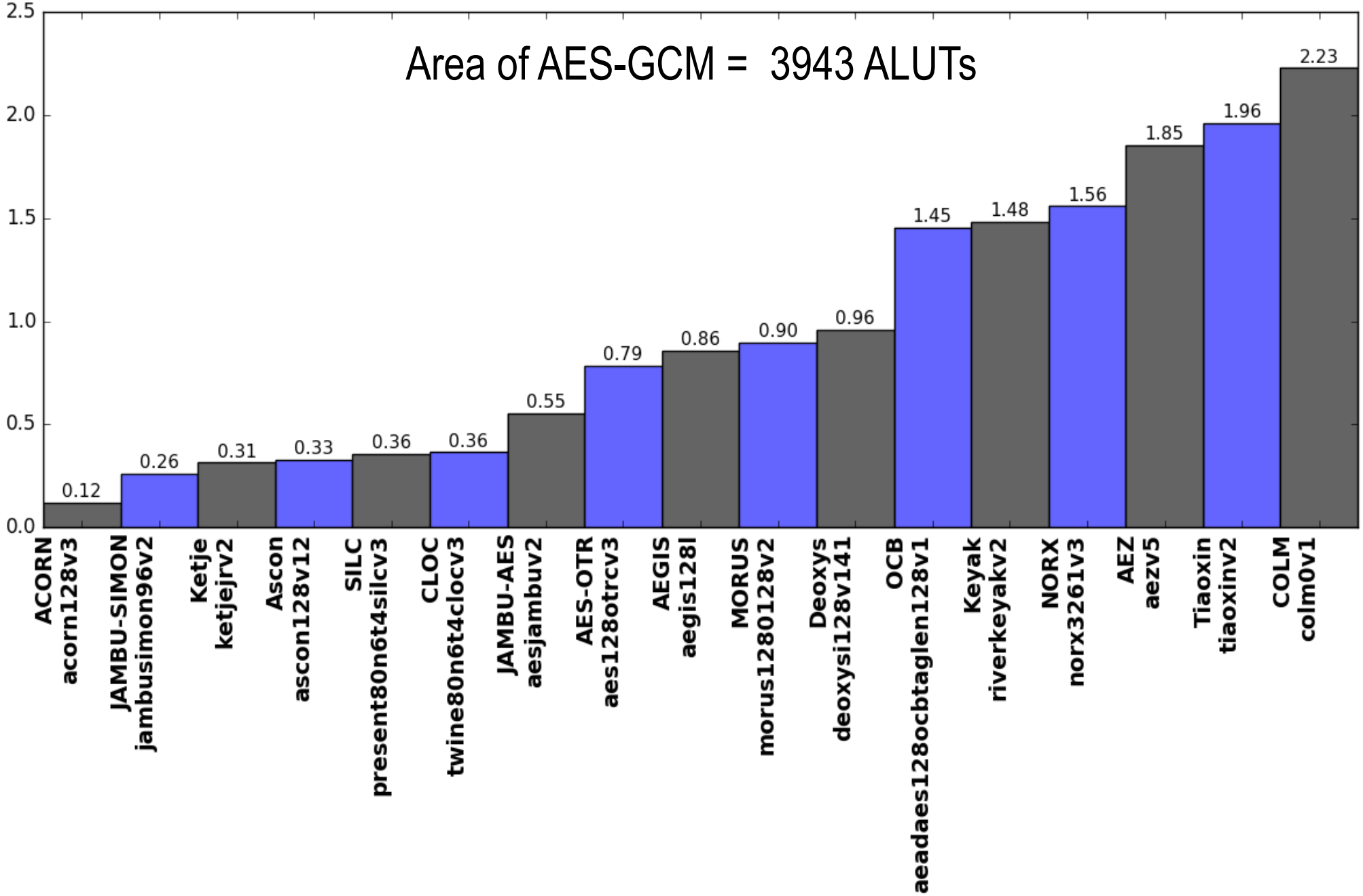
Ratio of a given Cipher Throughput/Throughput of AES-GCM



Relative Area (#ALUTs) in Stratix V
Ratio of a given Cipher Area/Area of AES-GCM



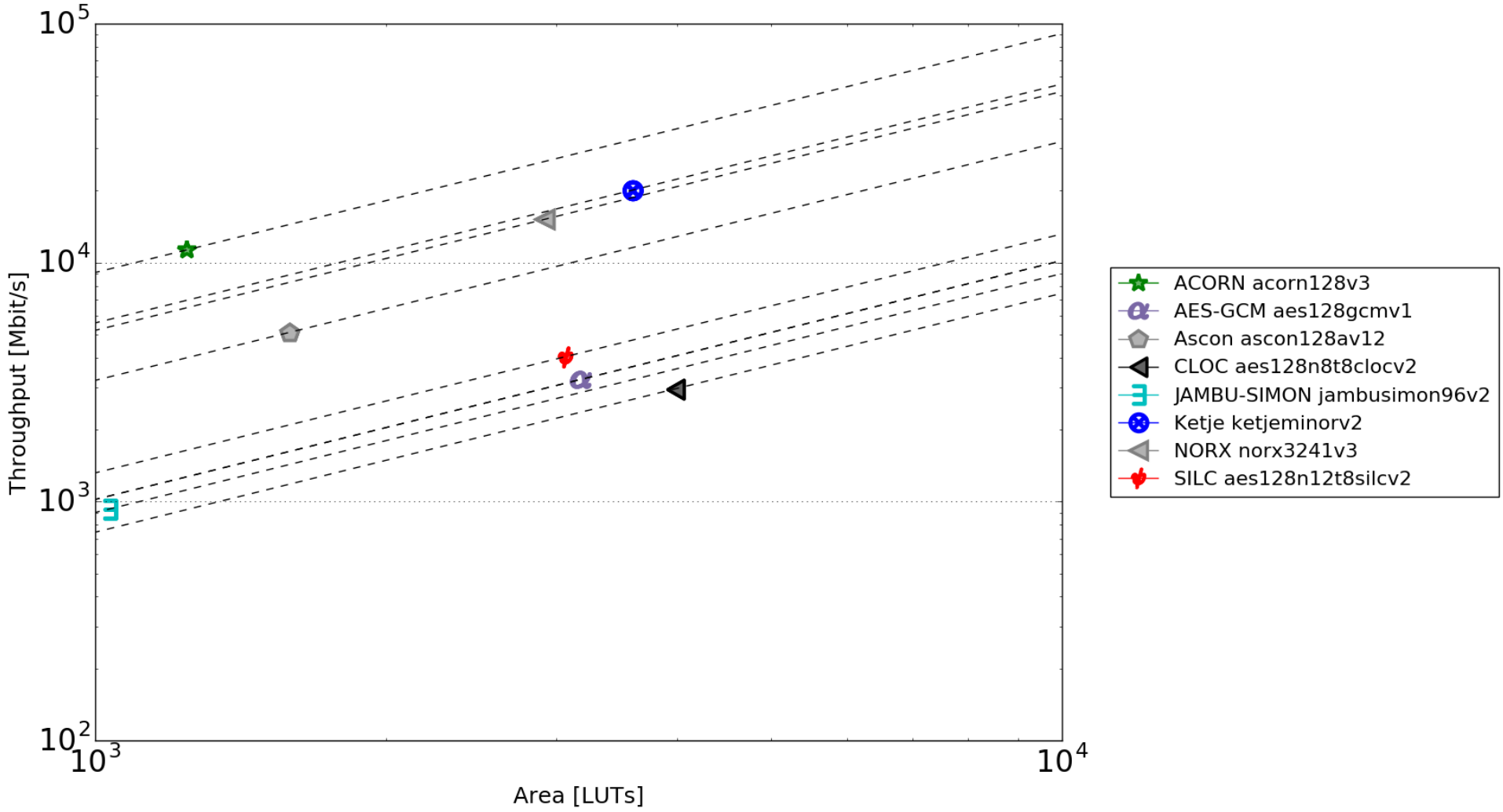
Relative Area (#ALUTs) in Stratix V
Ratio of a given Cipher Area/Area of AES-GCM



Use Case 1

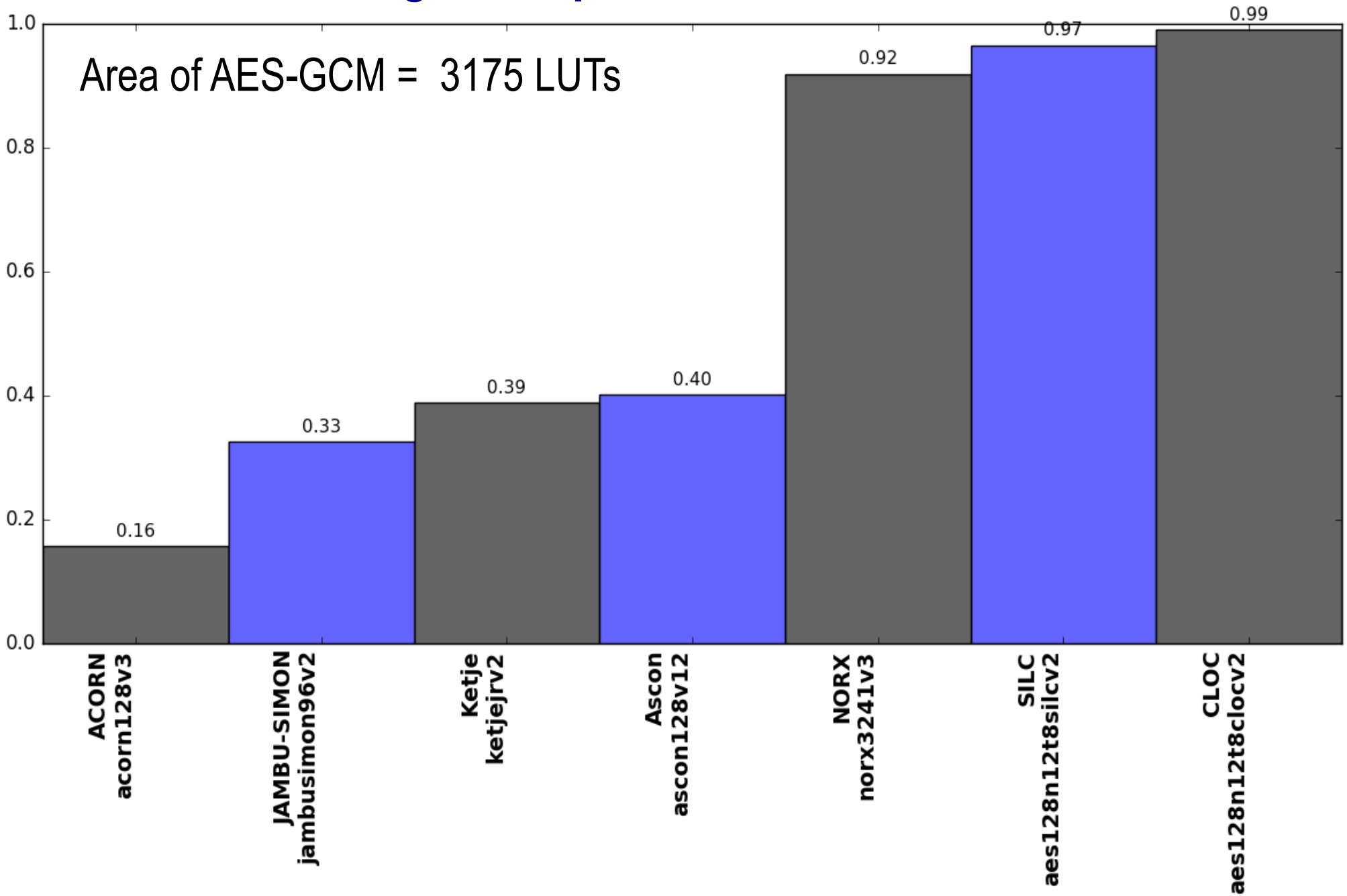
Virtex-6

Results for Virtex-6 – Throughput vs. Area
Logarithmic Scale

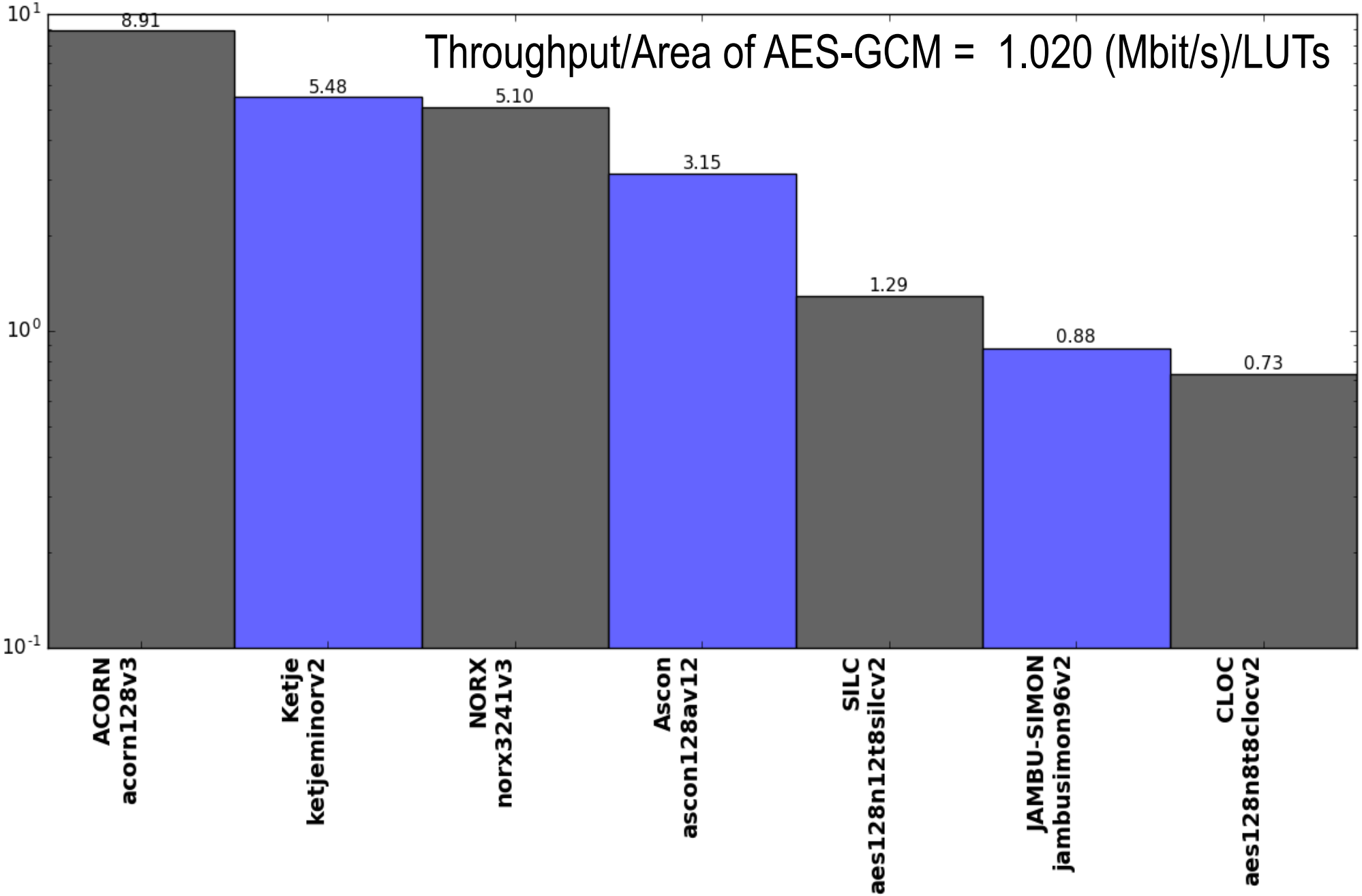


Relative Area (#LUTs) in Virtex-6

Ratio of a given Cipher Area/Area of AES-GCM

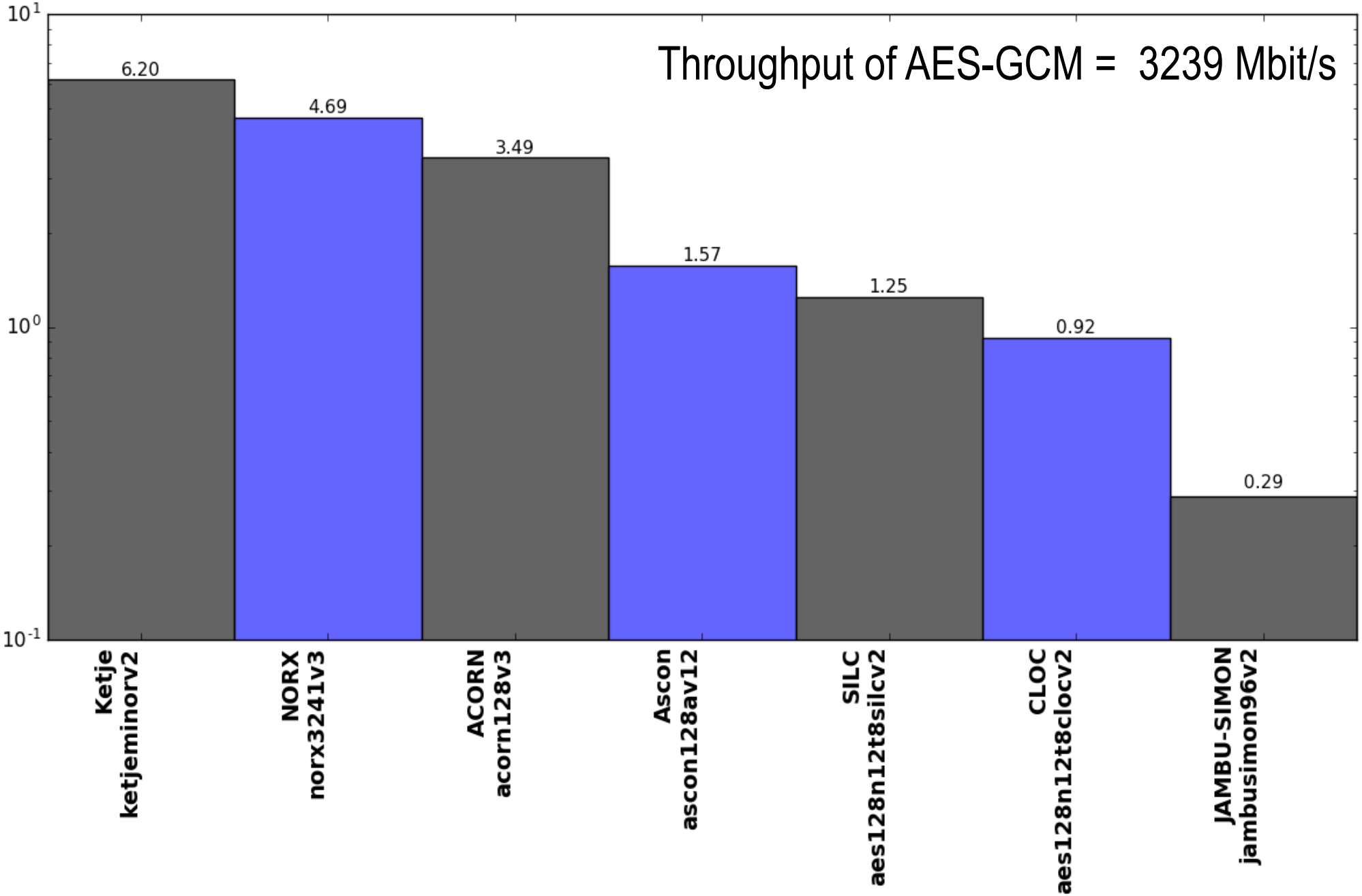


Relative Throughput/Area in Virtex-6 vs. AES-GCM



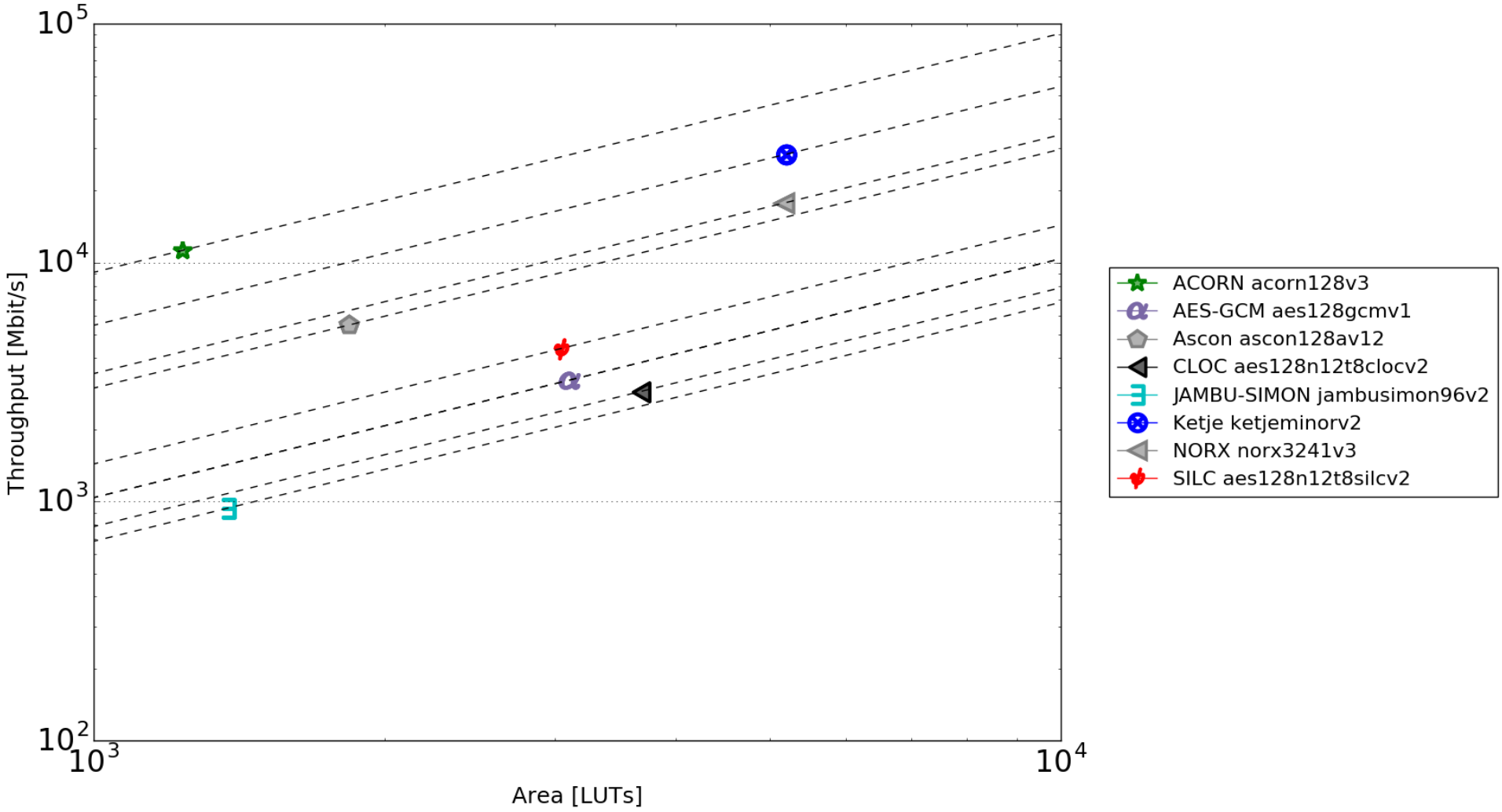
Relative Throughput in Virtex-6

Ratio of a given Cipher Throughput/Throughput of AES-GCM



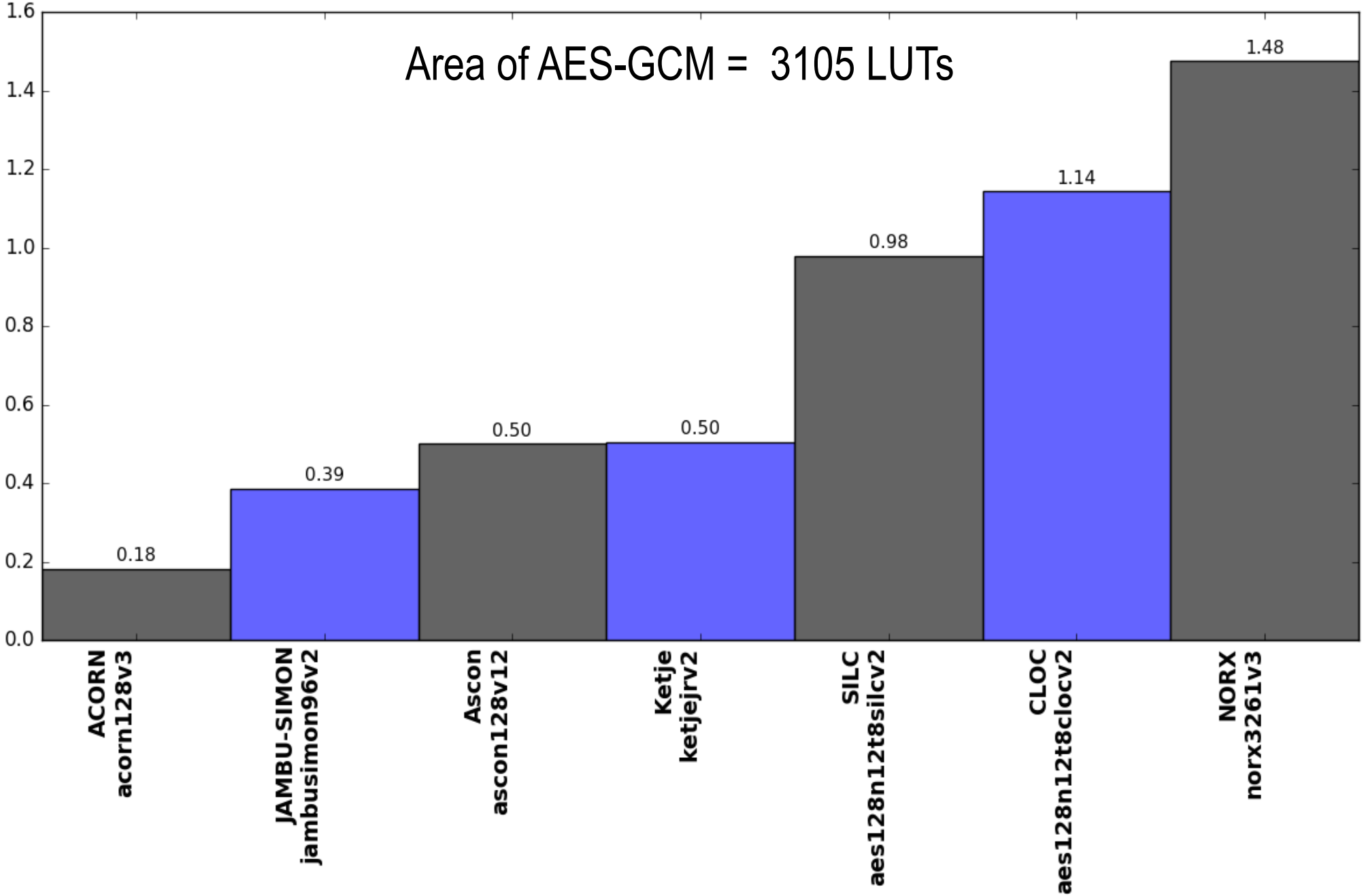
Virtex-7

Results for Virtex-7 – Throughput vs. Area
Logarithmic Scale

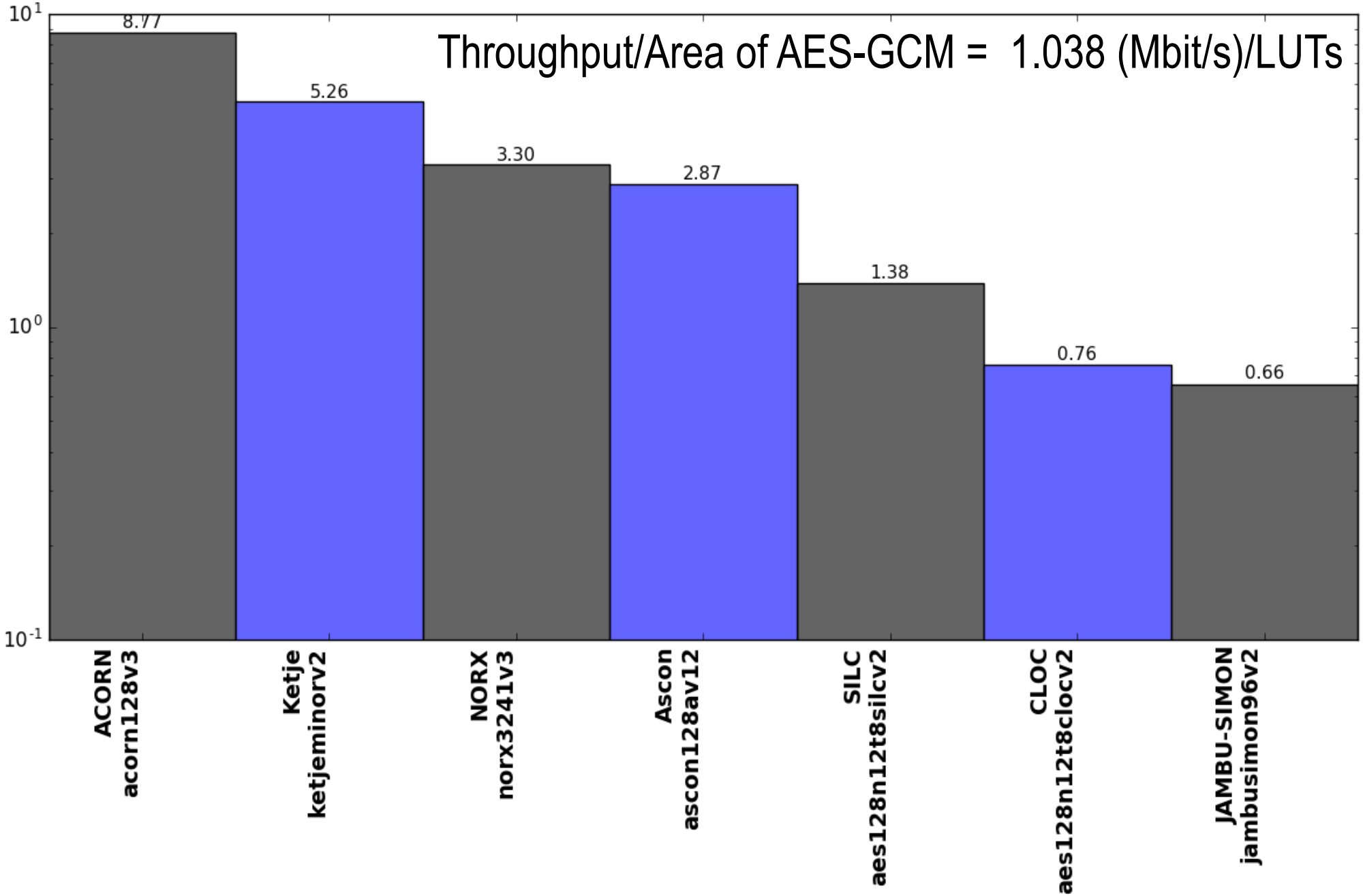


Relative Area (#LUTs) in Virtex-7

Ratio of a given Cipher Area/Area of AES-GCM

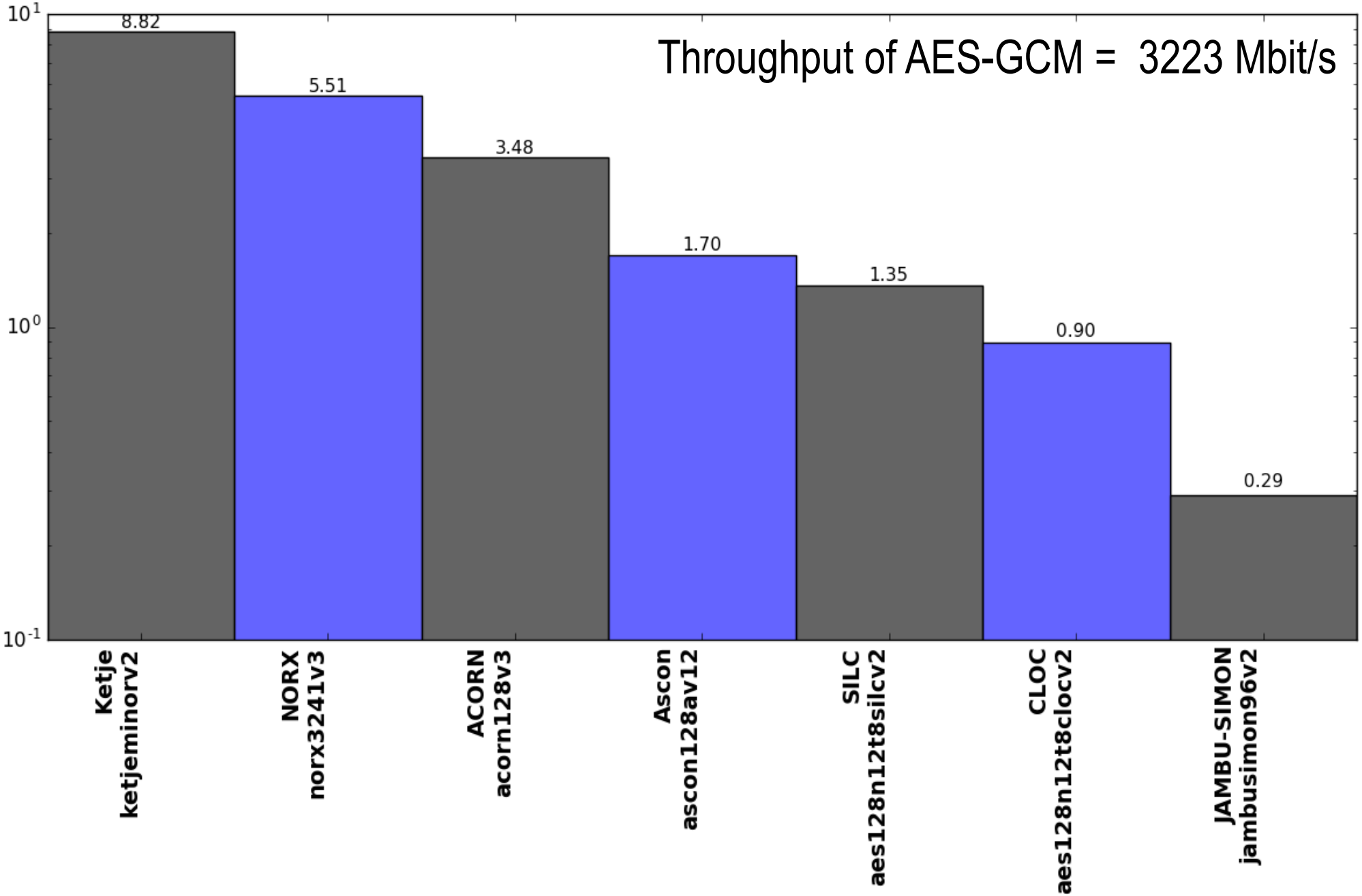


Relative Throughput/Area in Virtex-7 vs. AES-GCM



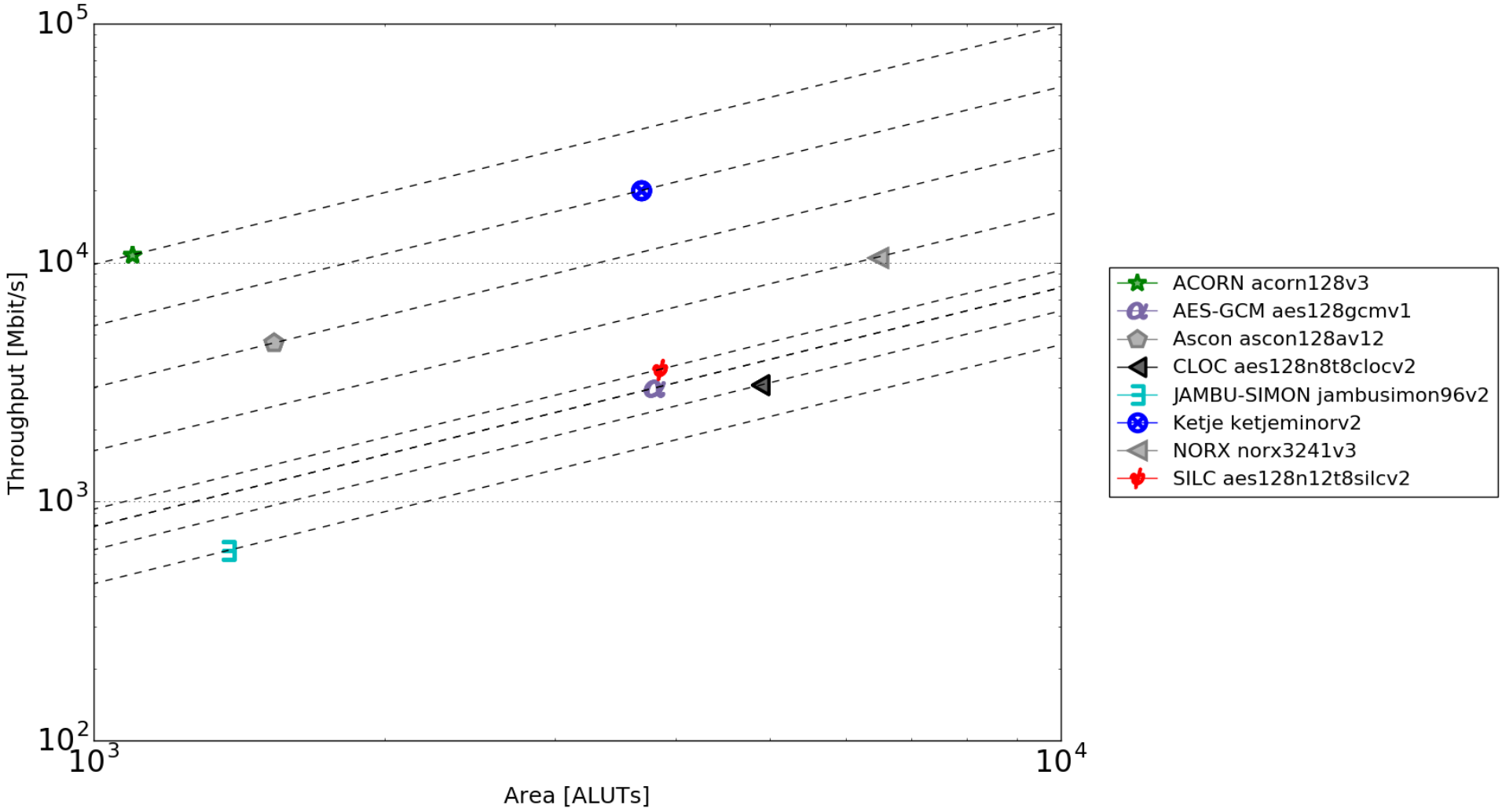
Relative Throughput in Virtex-7

Ratio of a given Cipher Throughput/Throughput of AES-GCM

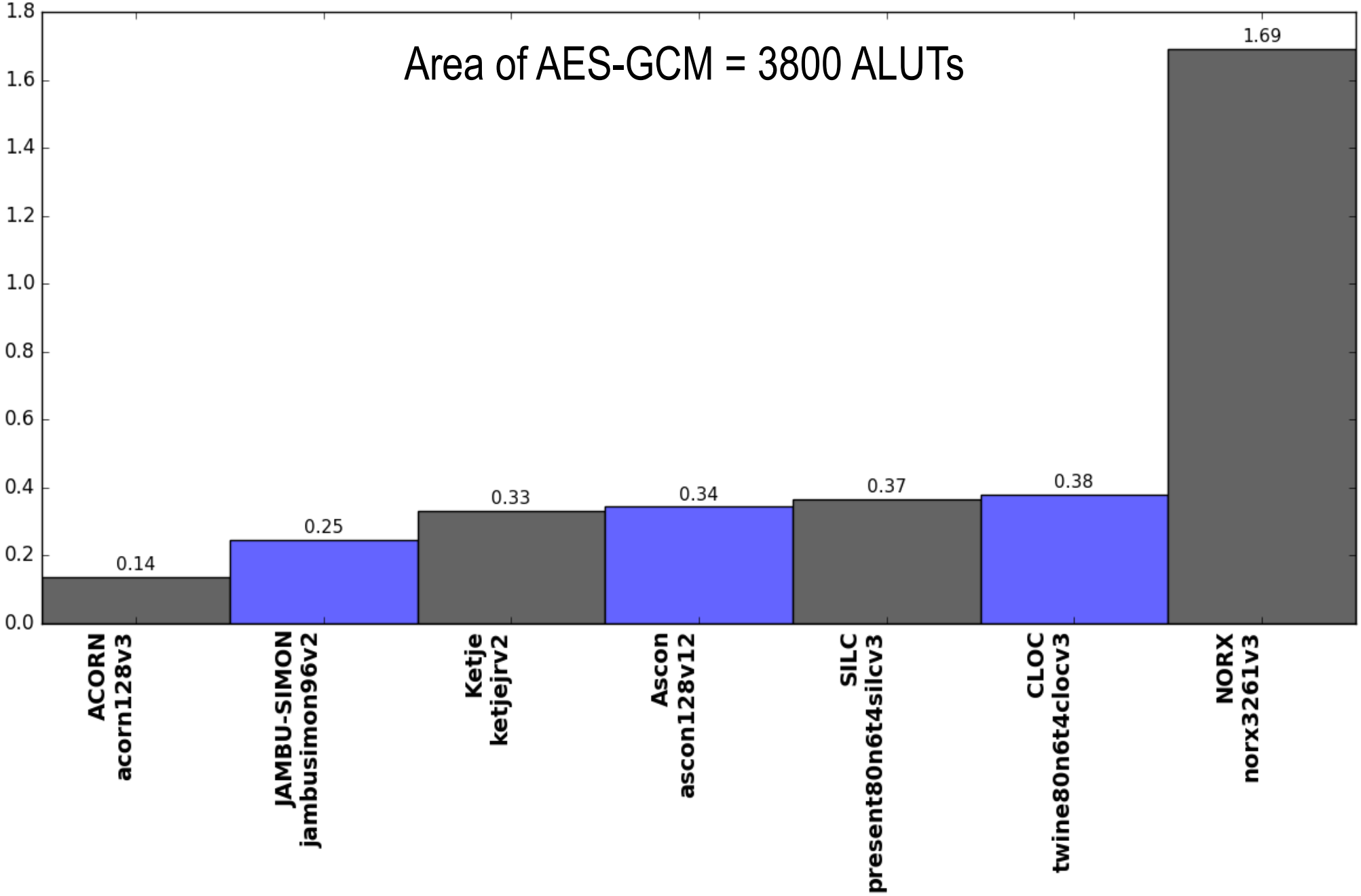


Stratix IV

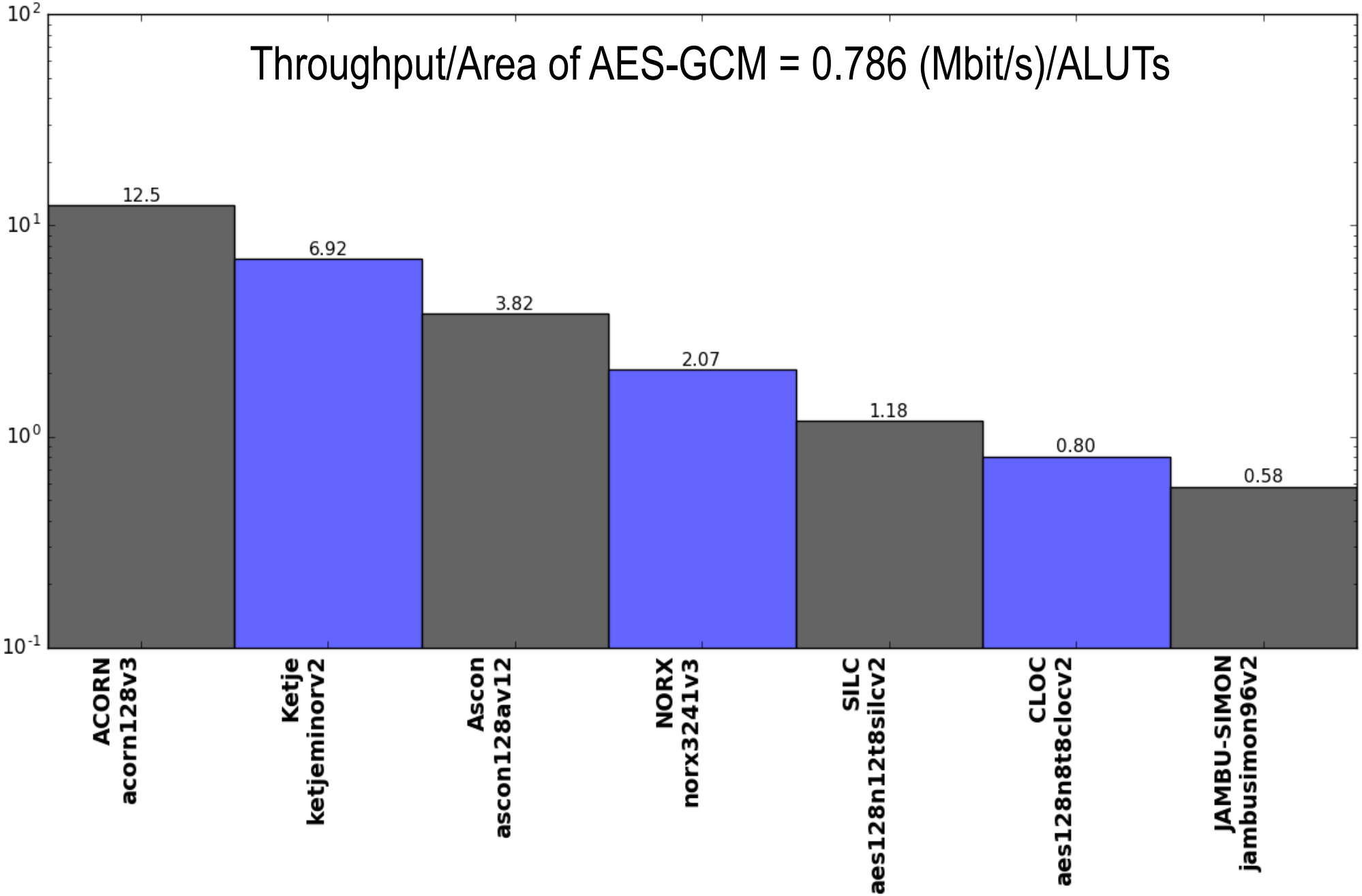
Results for Stratix IV – Throughput vs. Area
Logarithmic Scale



Relative Area (#ALUTs) in Stratix IV
Ratio of a given Cipher Area/Area of AES-GCM

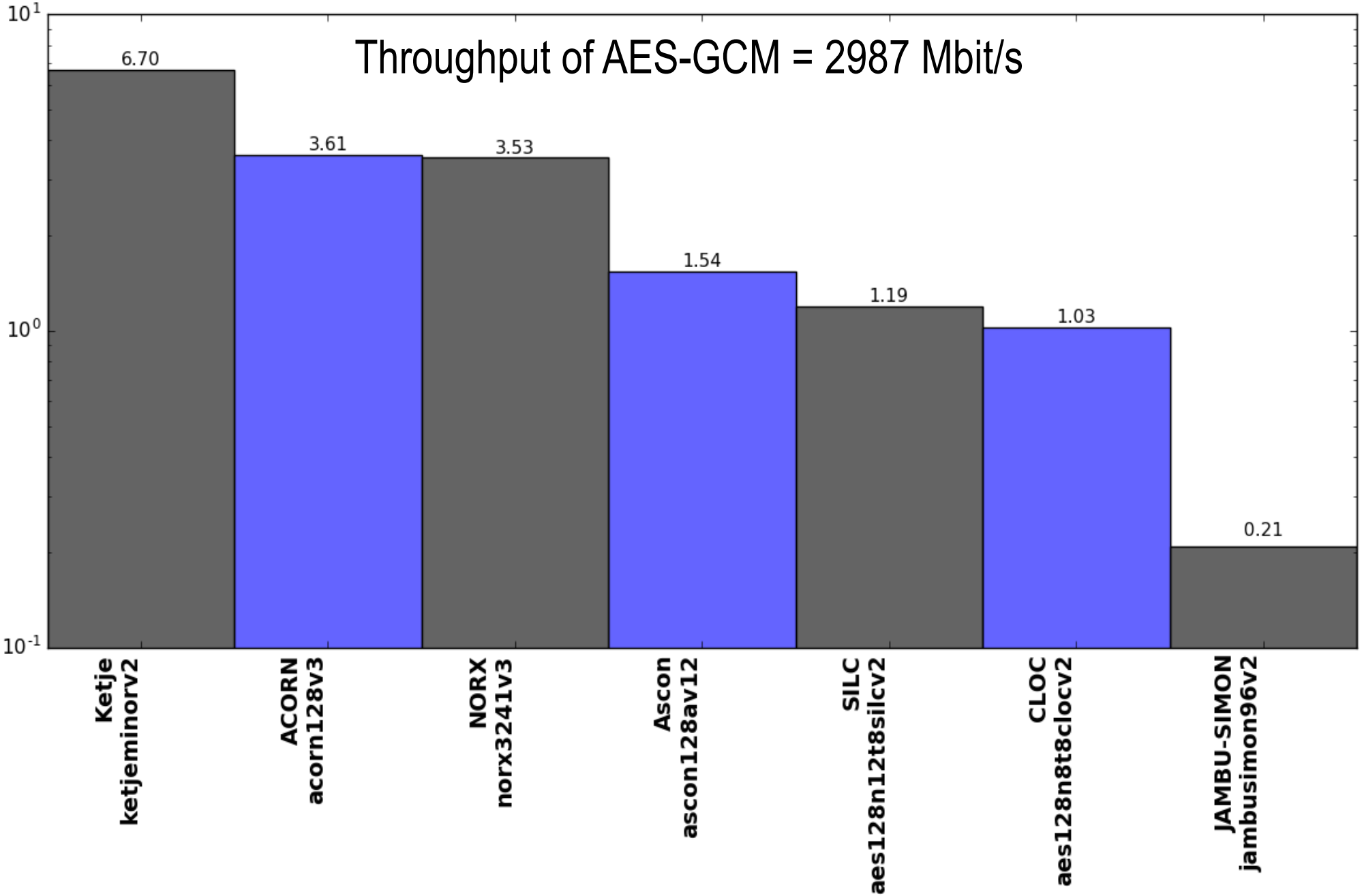


Relative Throughput/Area in Stratix IV vs. AES-GCM



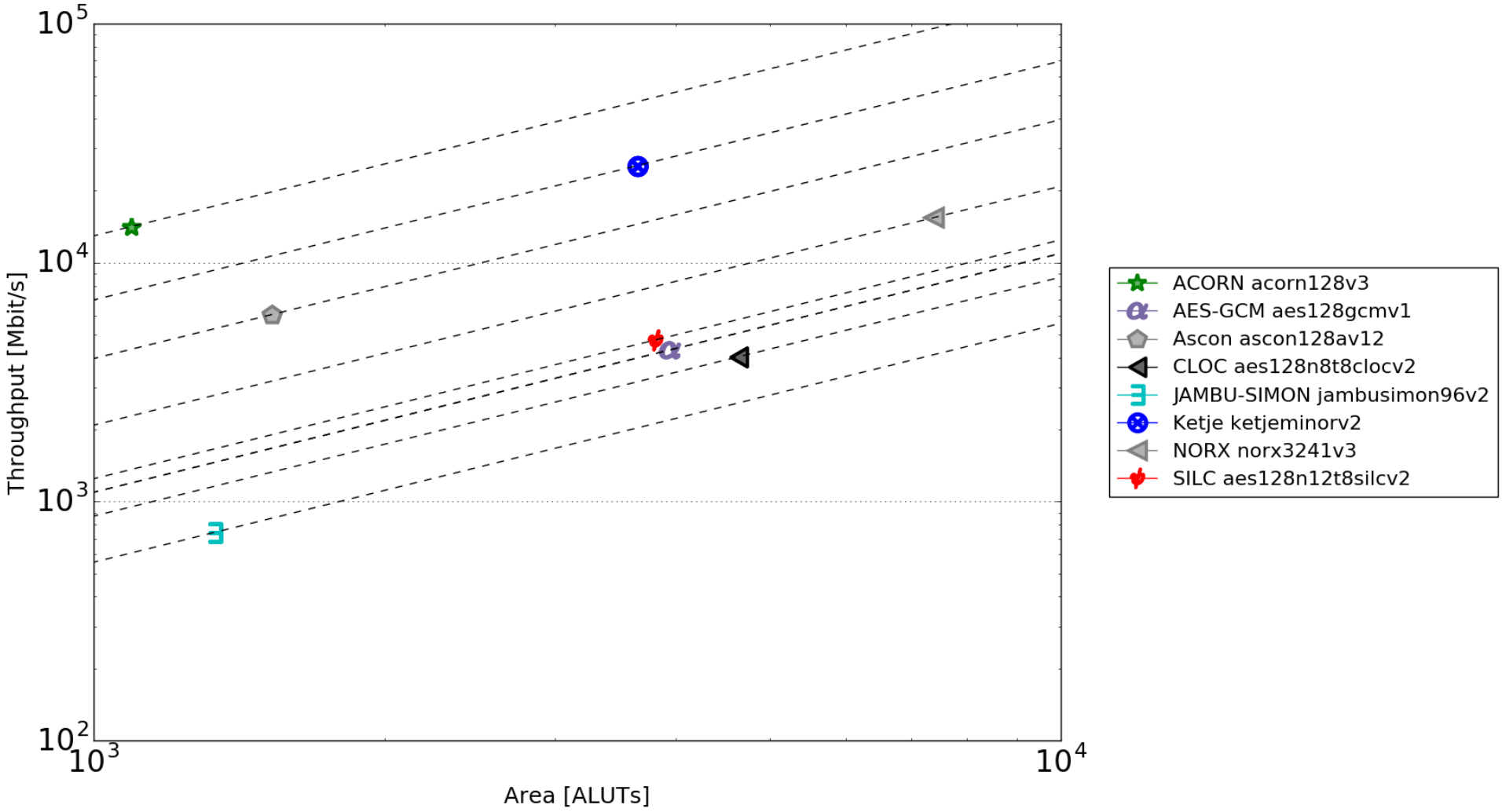
Relative Throughput in Stratix IV

Ratio of a given Cipher Throughput/Throughput of AES-GCM

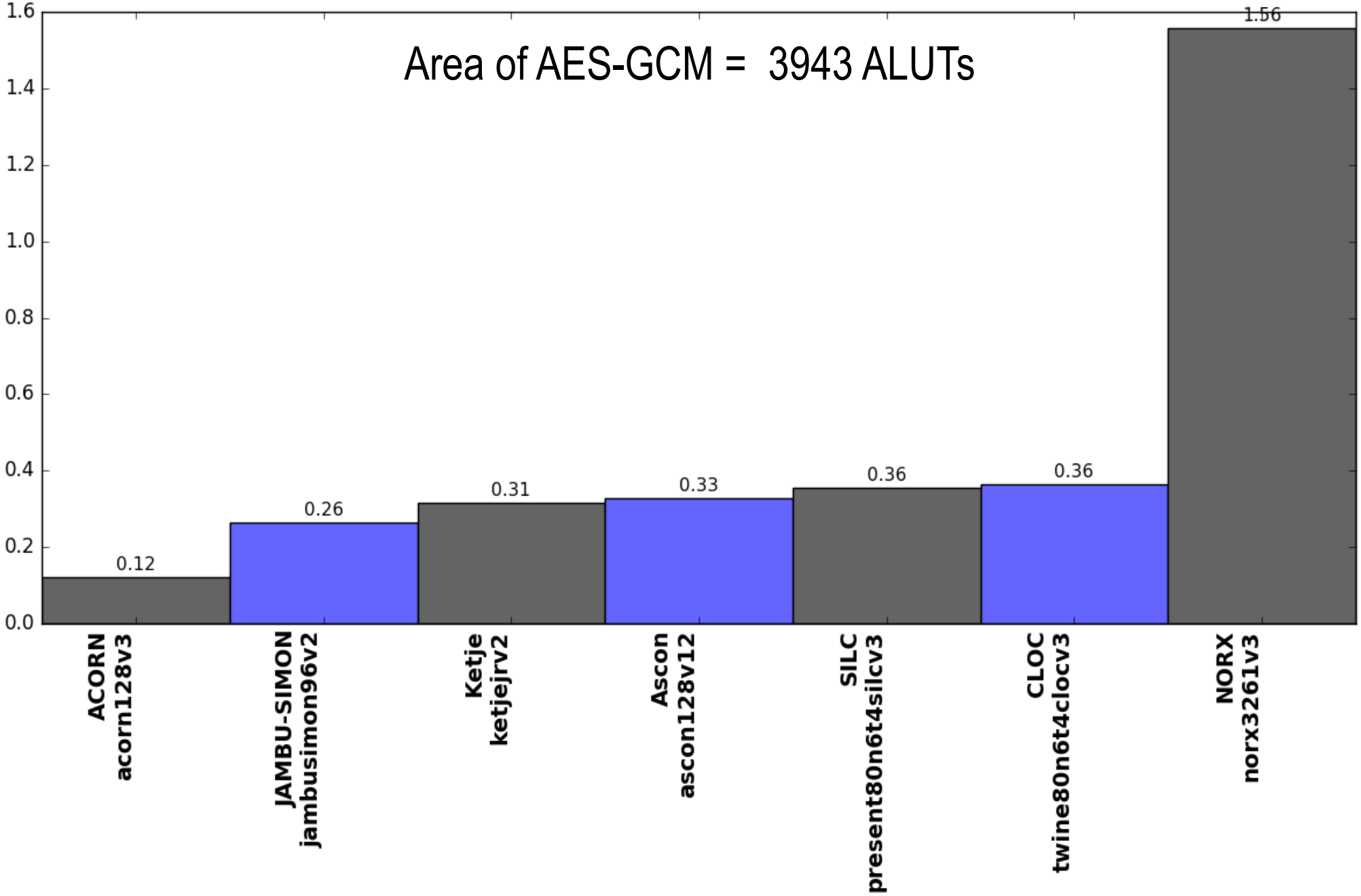


Stratix V

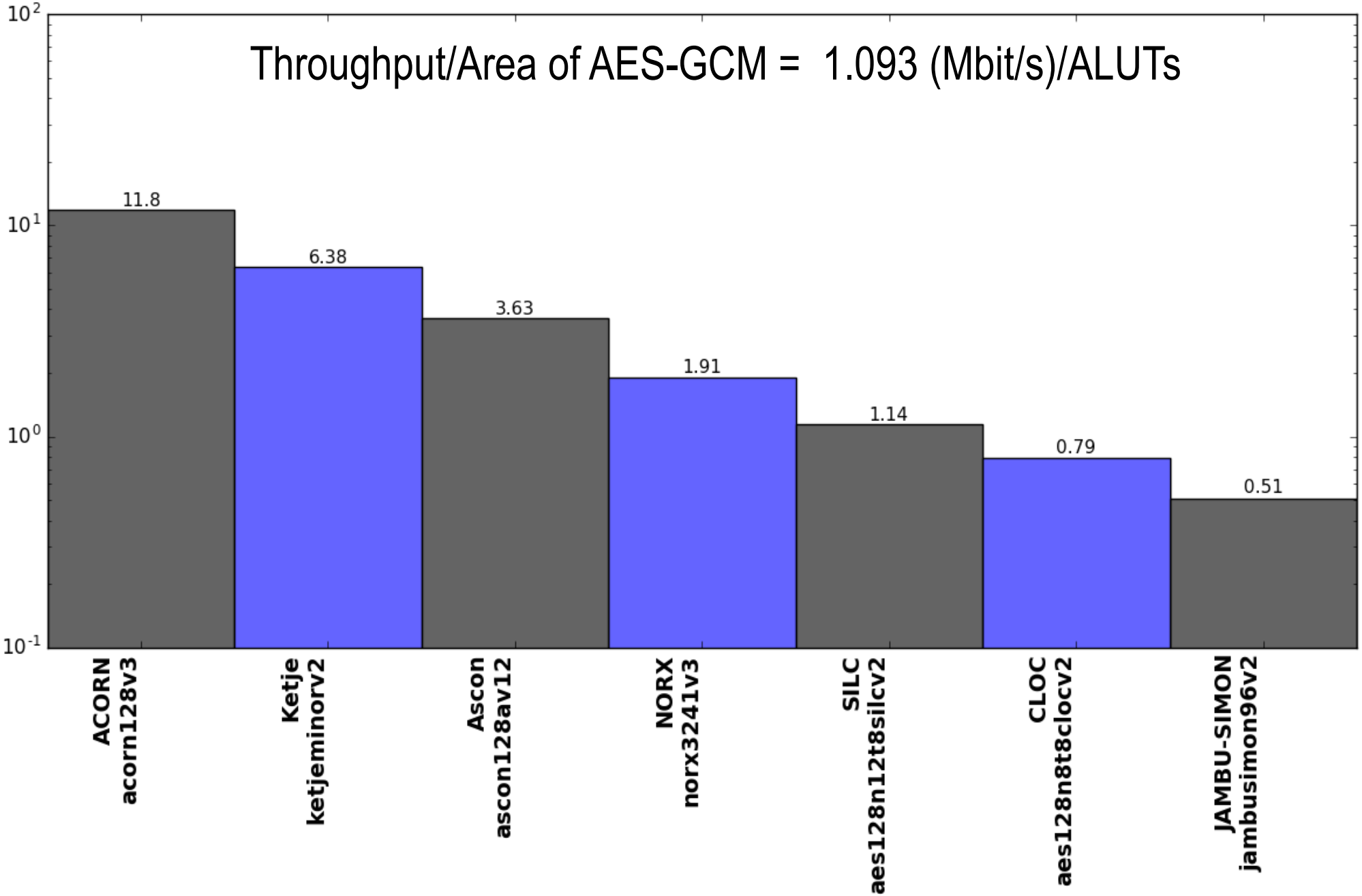
Results for Stratix V – Throughput vs. Area
Logarithmic Scale



Relative Area (#ALUTs) in Stratix V
Ratio of a given Cipher Area/Area of AES-GCM

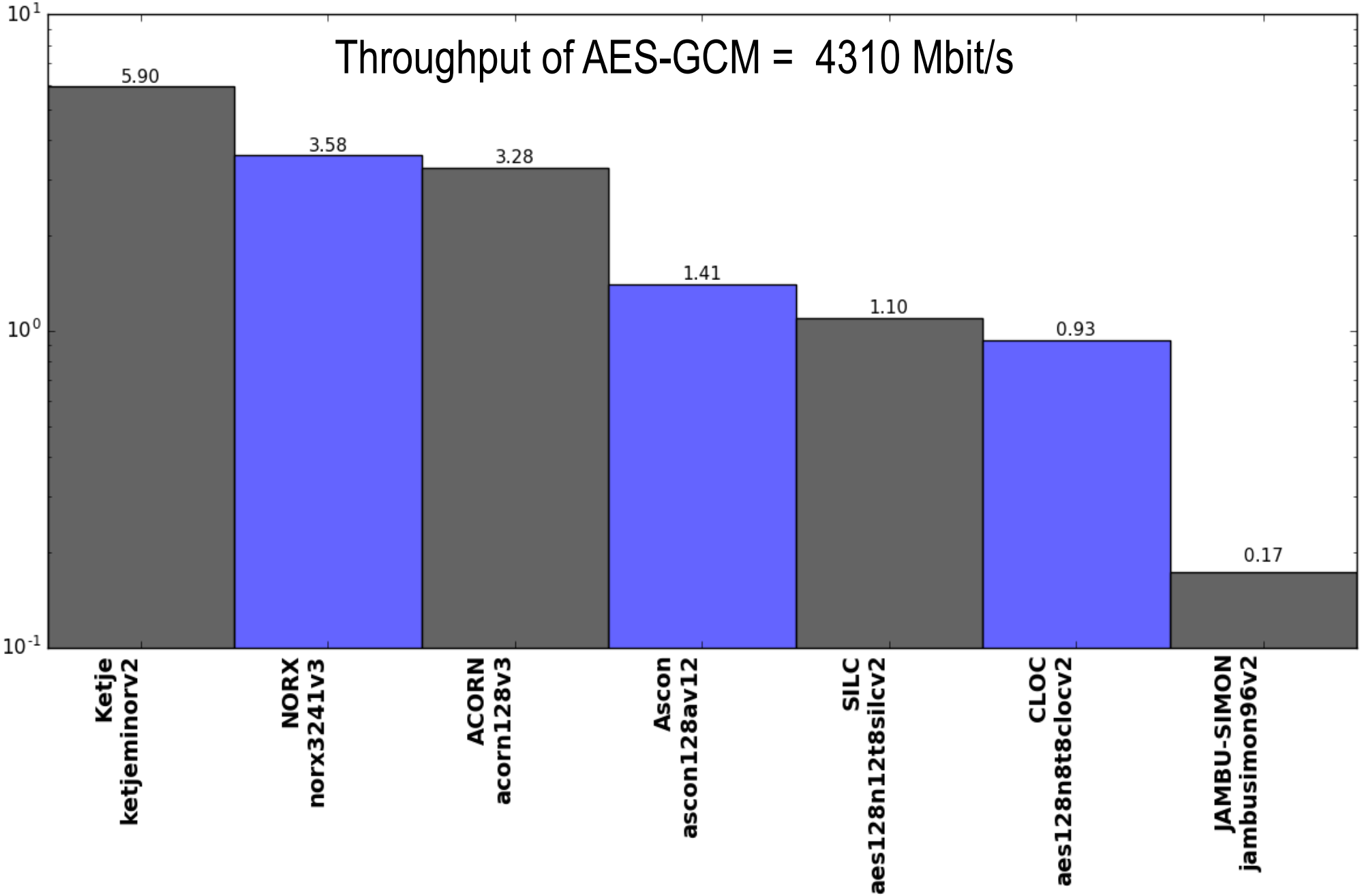


Relative Throughput/Area in Stratix V vs. AES-GCM



Relative Throughput in Stratix V

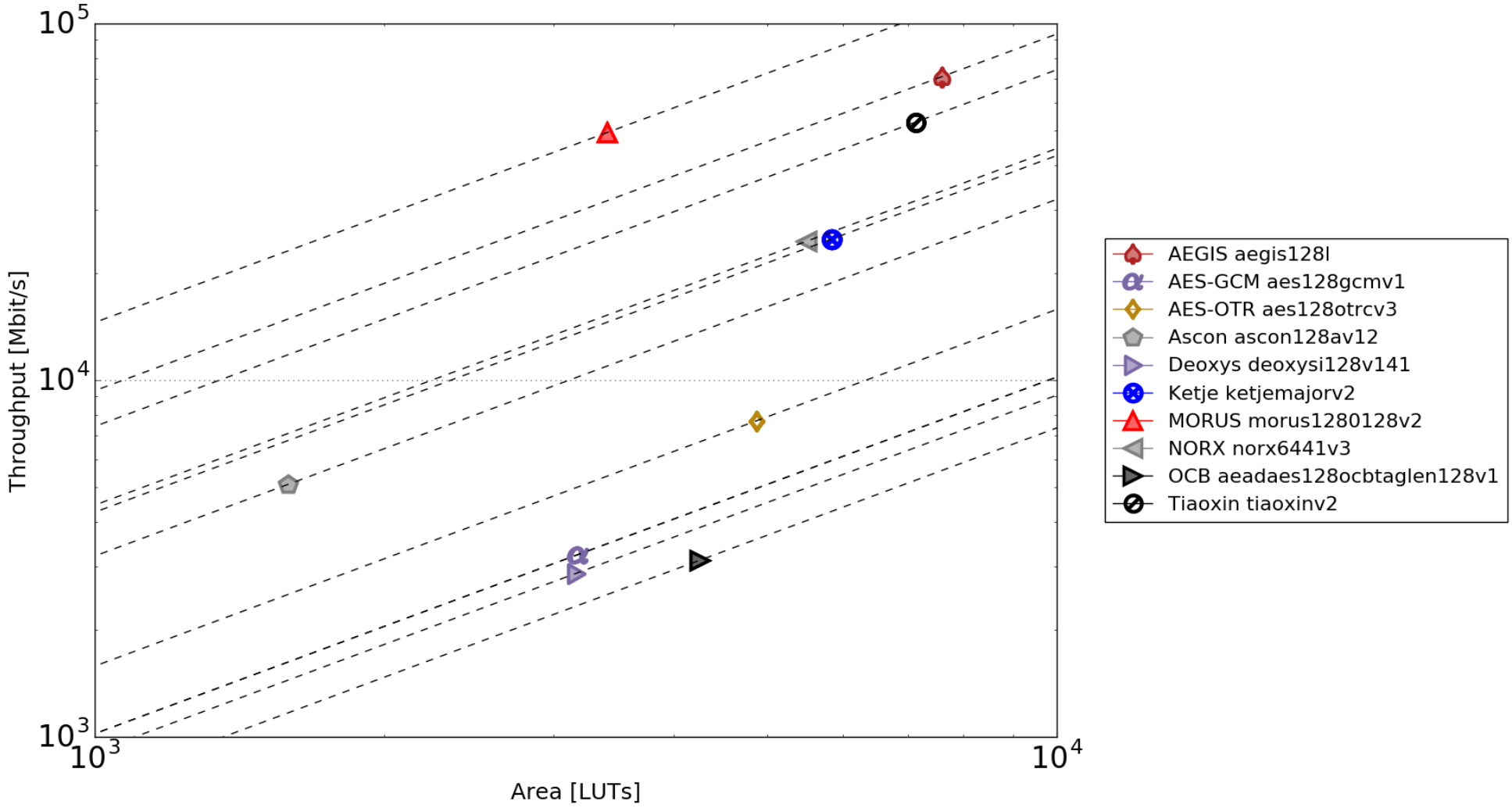
Ratio of a given Cipher Throughput/Throughput of AES-GCM



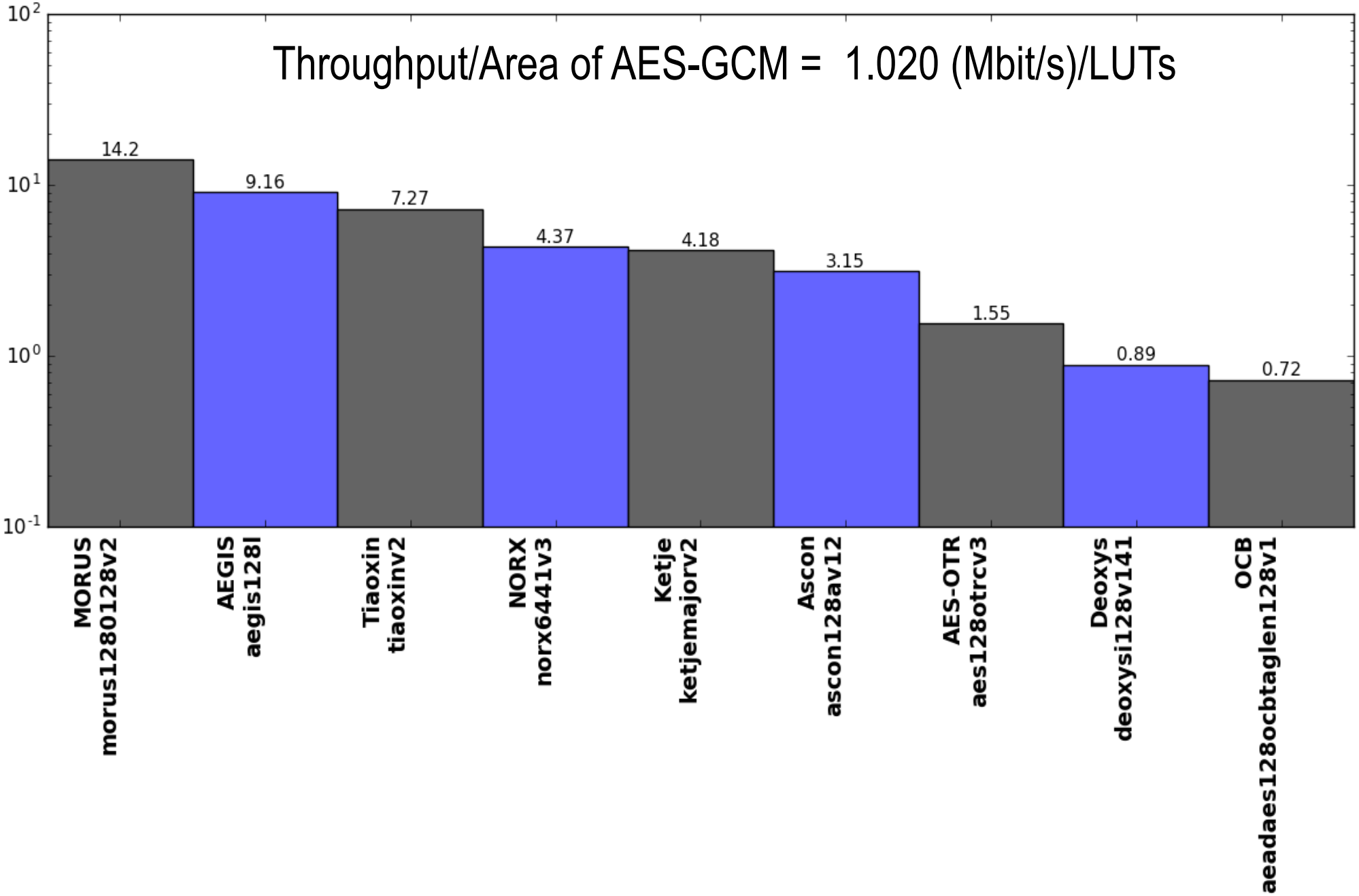
Use Case 2

Virtex-6

Results for Virtex-6 – Throughput vs. Area
Logarithmic Scale

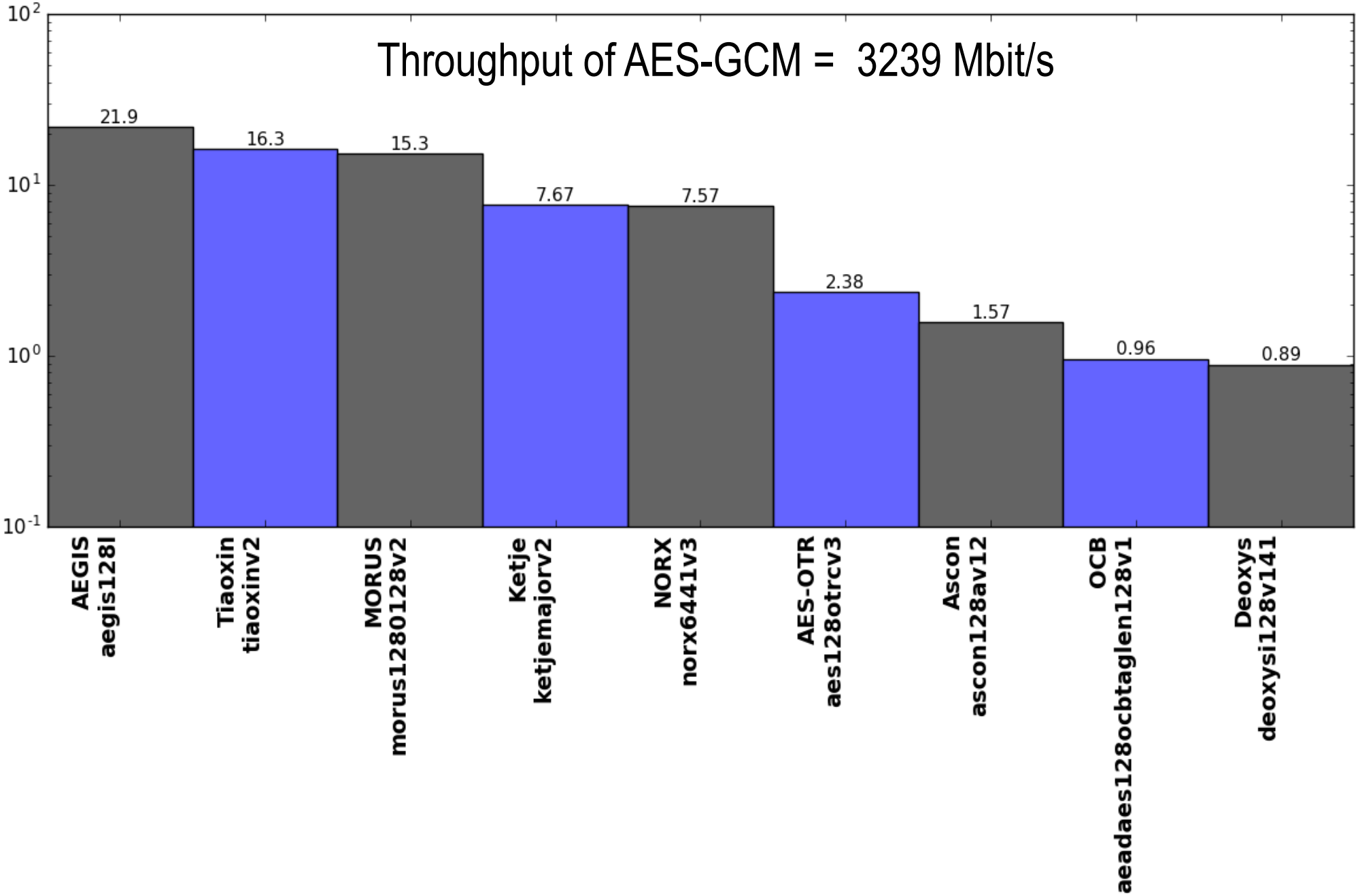


Relative Throughput/Area in Virtex-6 vs. AES-GCM



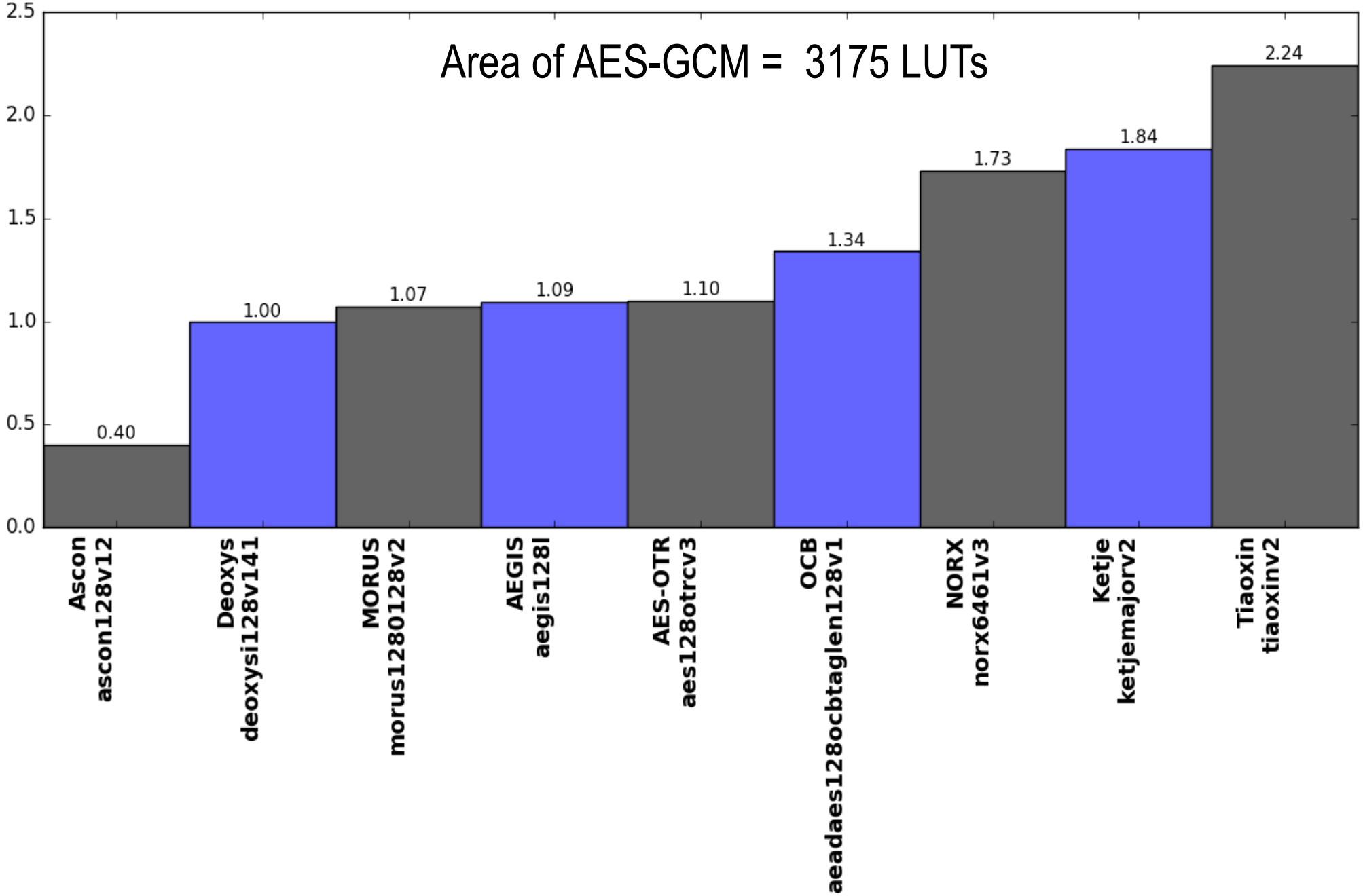
Relative Throughput in Virtex-6

Ratio of a given Cipher Throughput/Throughput of AES-GCM



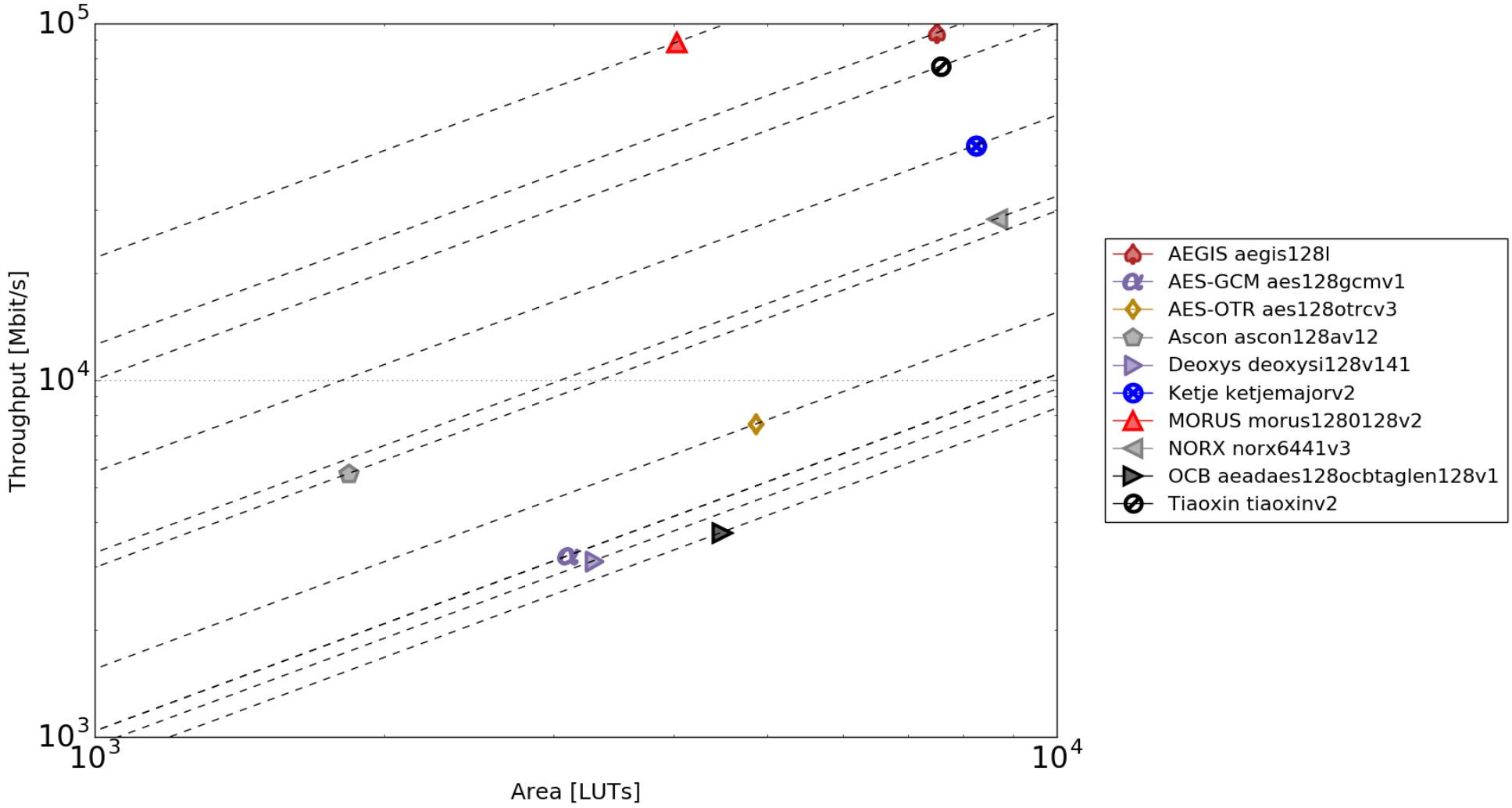
Relative Area (#LUTs) in Virtex-6

Ratio of a given Cipher Area/Area of AES-GCM

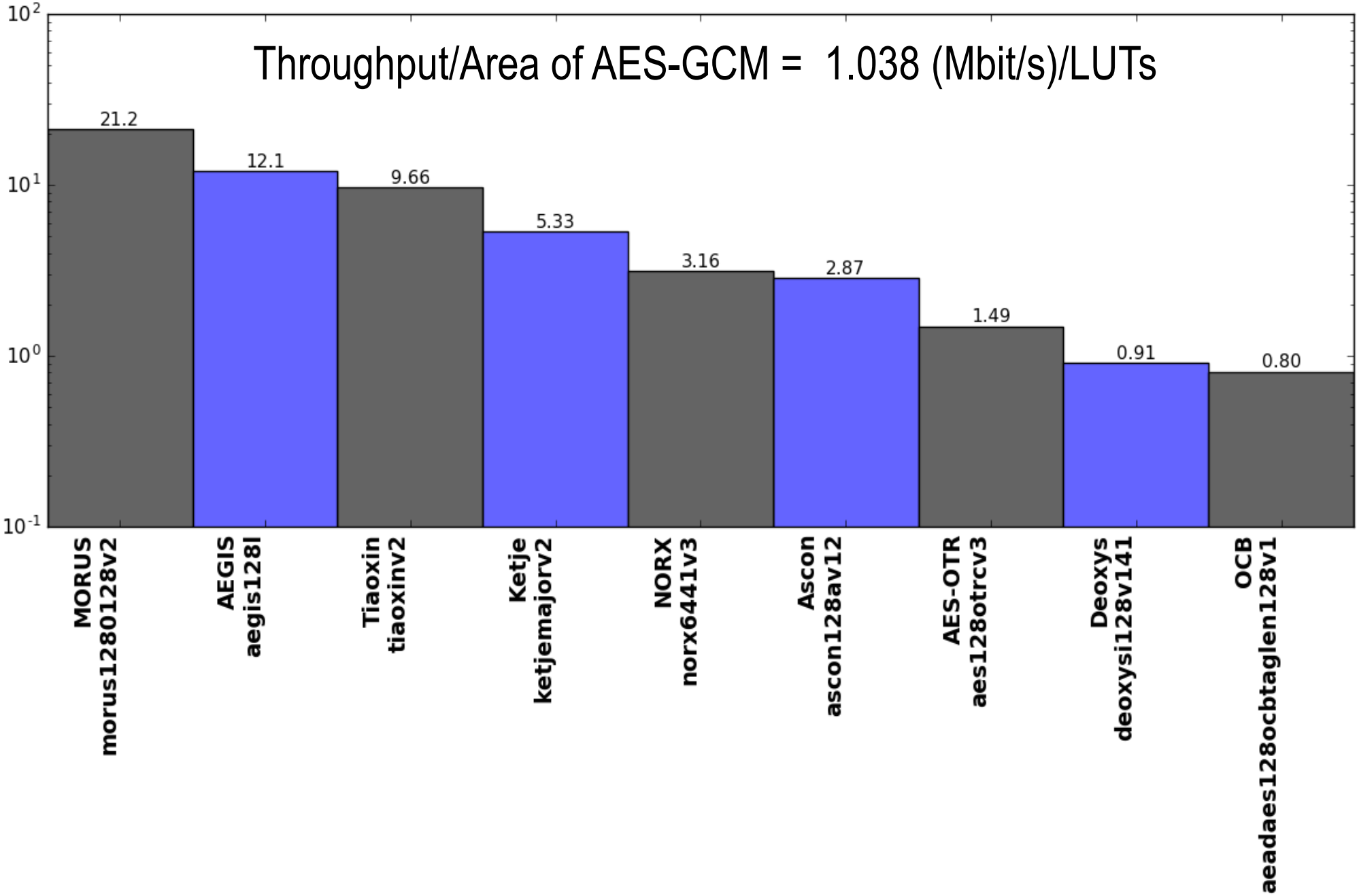


Virtex-7

Results for Virtex-7 – Throughput vs. Area
Logarithmic Scale

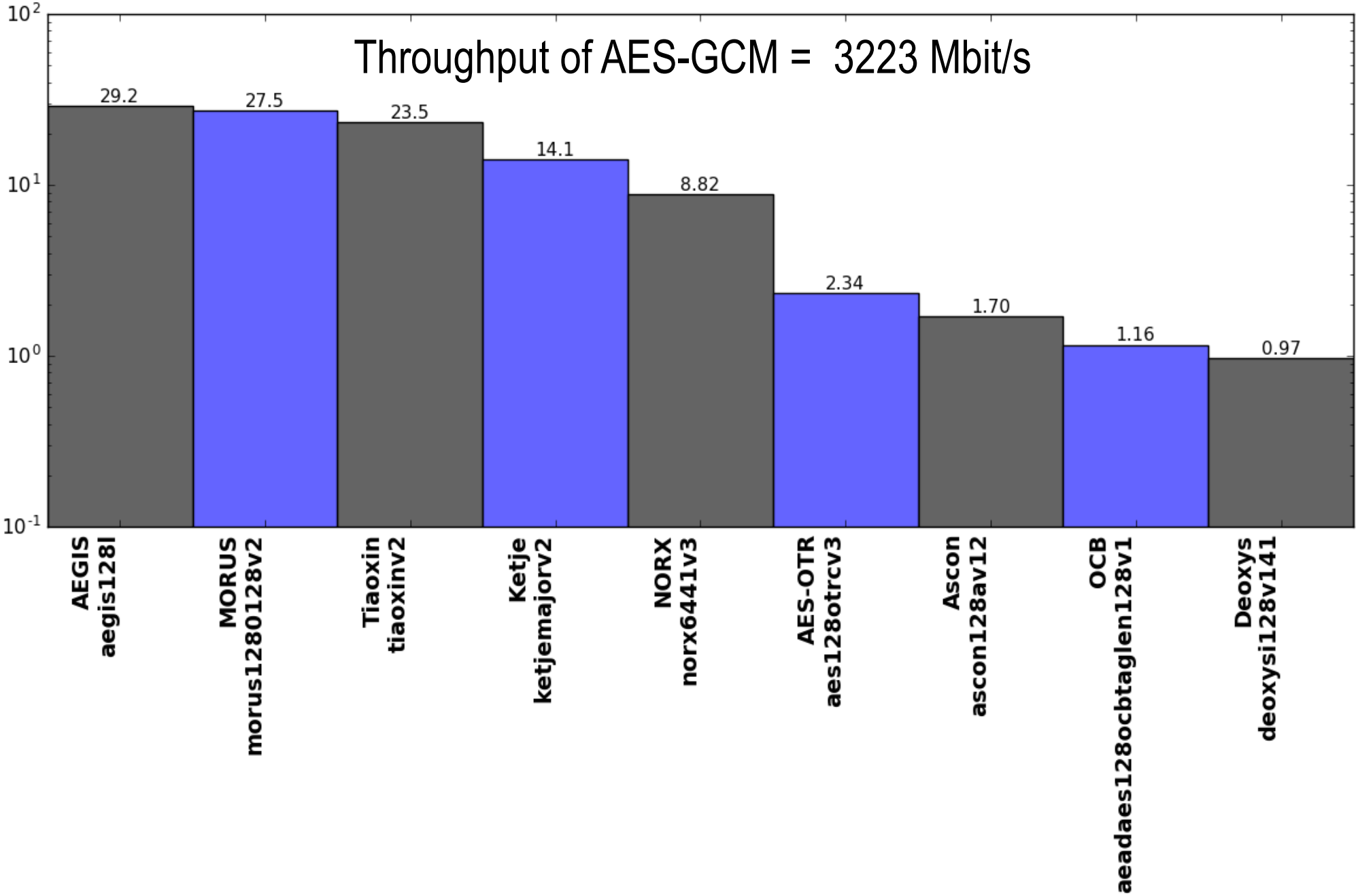


Relative Throughput/Area in Virtex-7 vs. AES-GCM



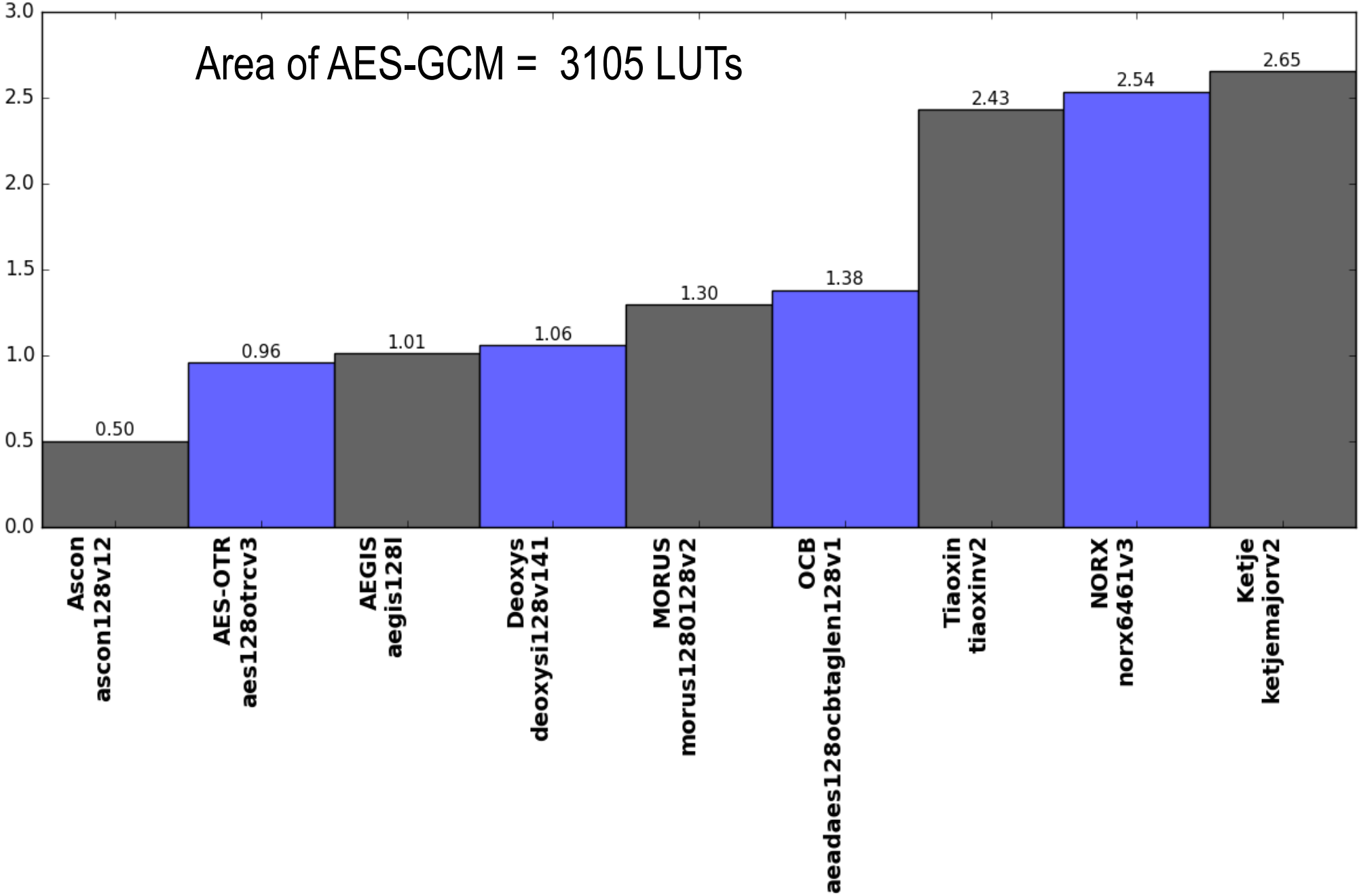
Relative Throughput in Virtex-7

Ratio of a given Cipher Throughput/Throughput of AES-GCM



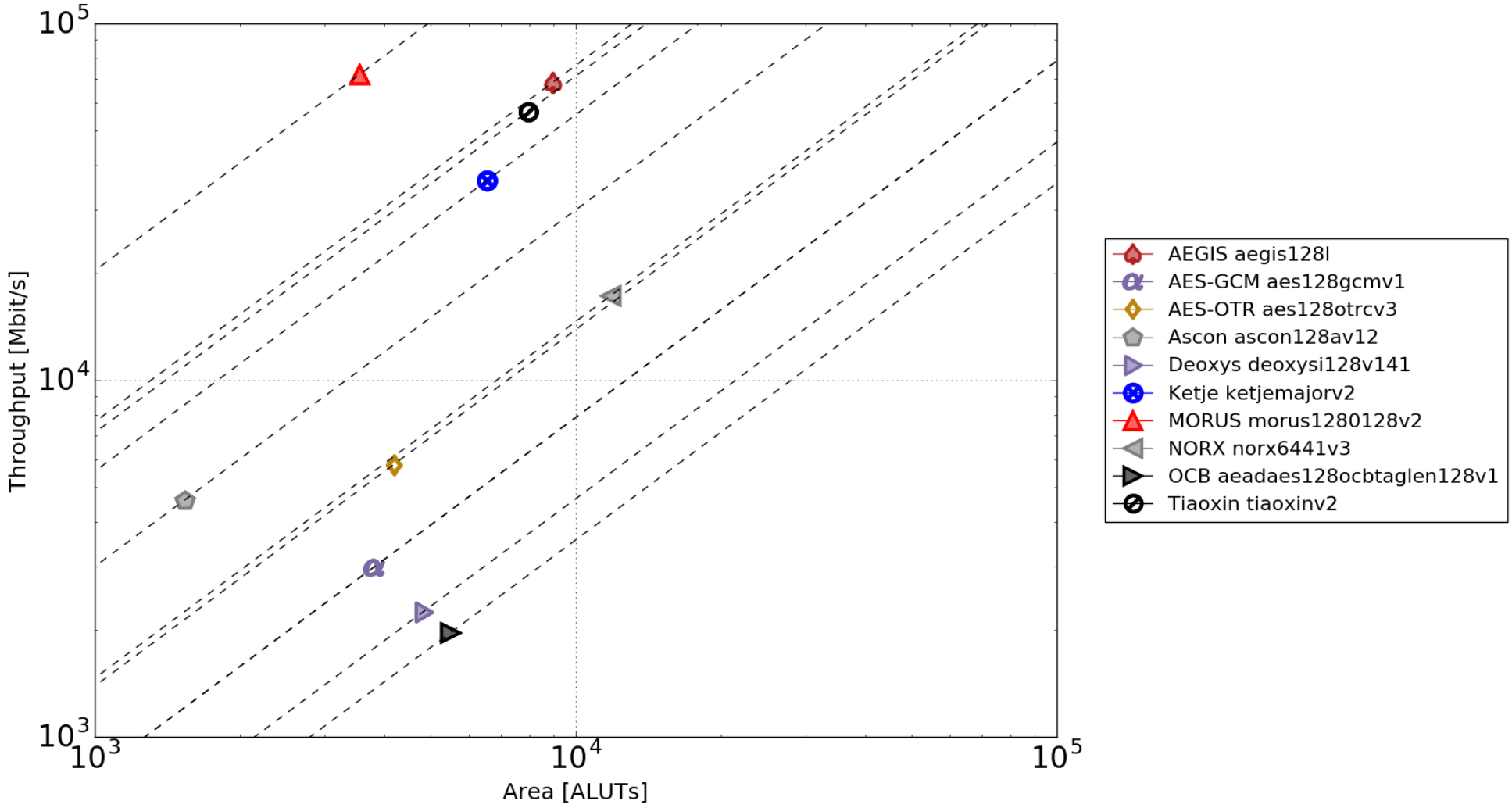
Relative Area (#LUTs) in Virtex-7

Ratio of a given Cipher Area/Area of AES-GCM

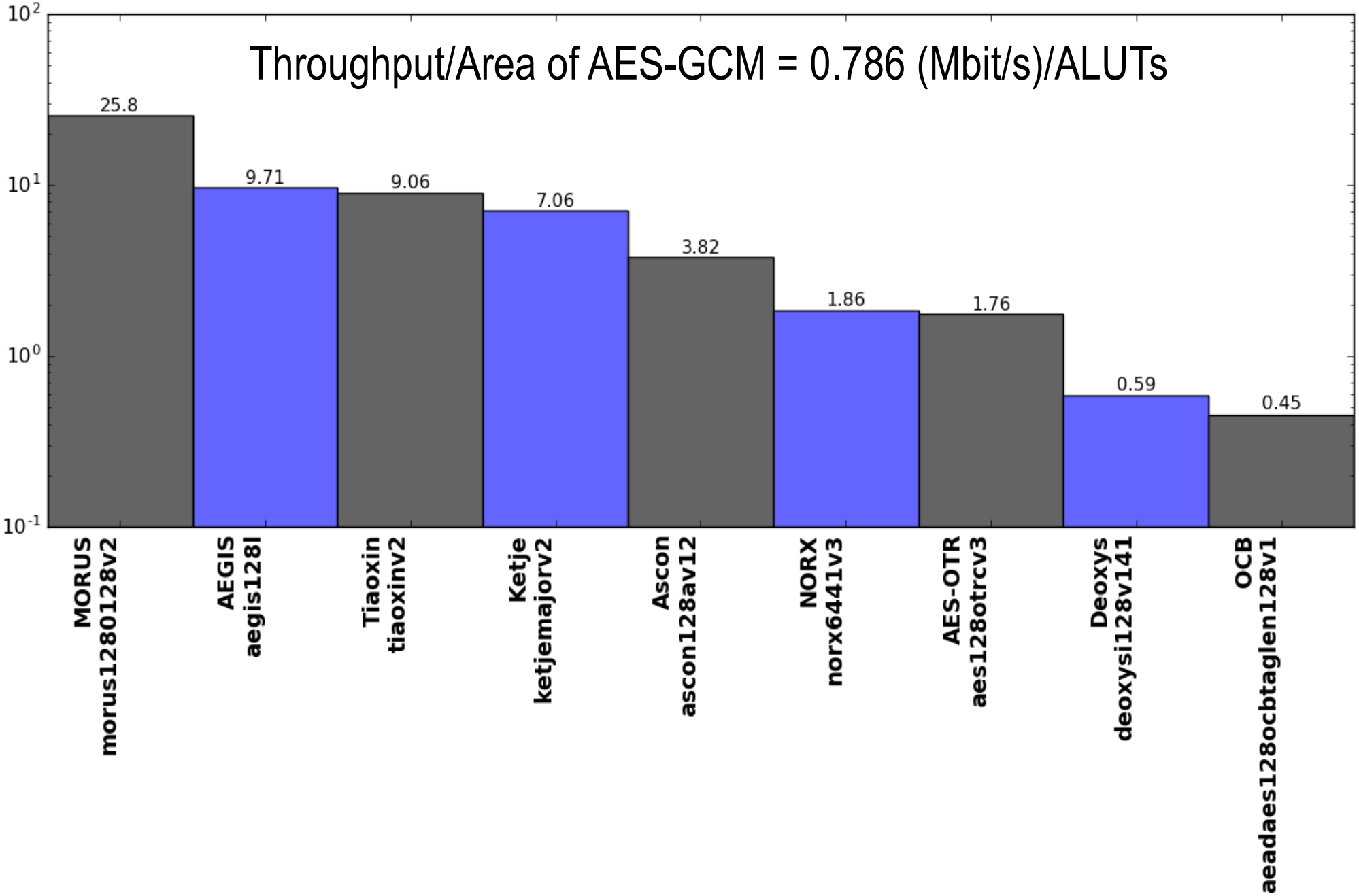


Stratix IV

Results for Stratix IV – Throughput vs. Area Logarithmic Scale

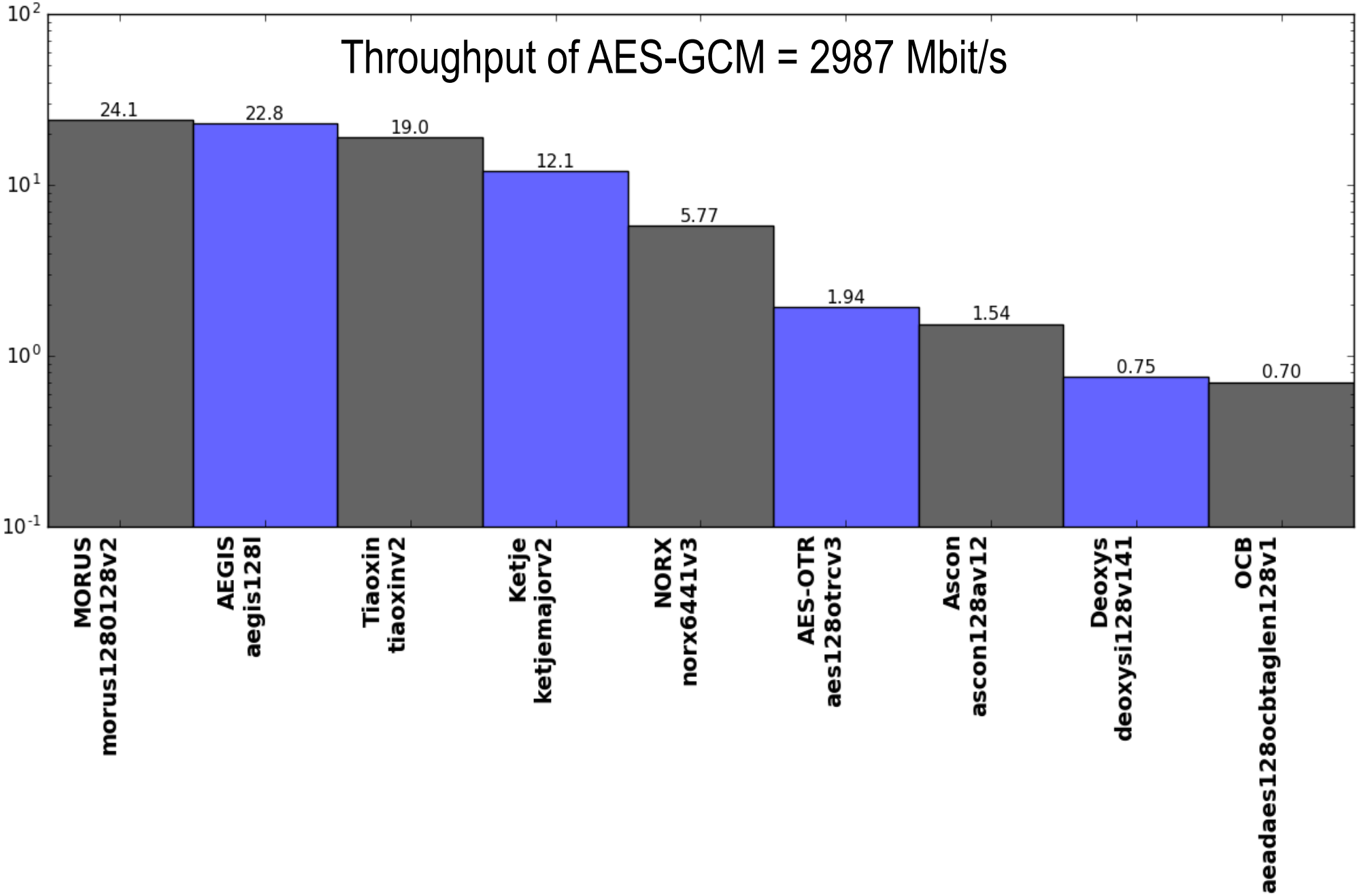


Relative Throughput/Area in Stratix IV vs. AES-GCM

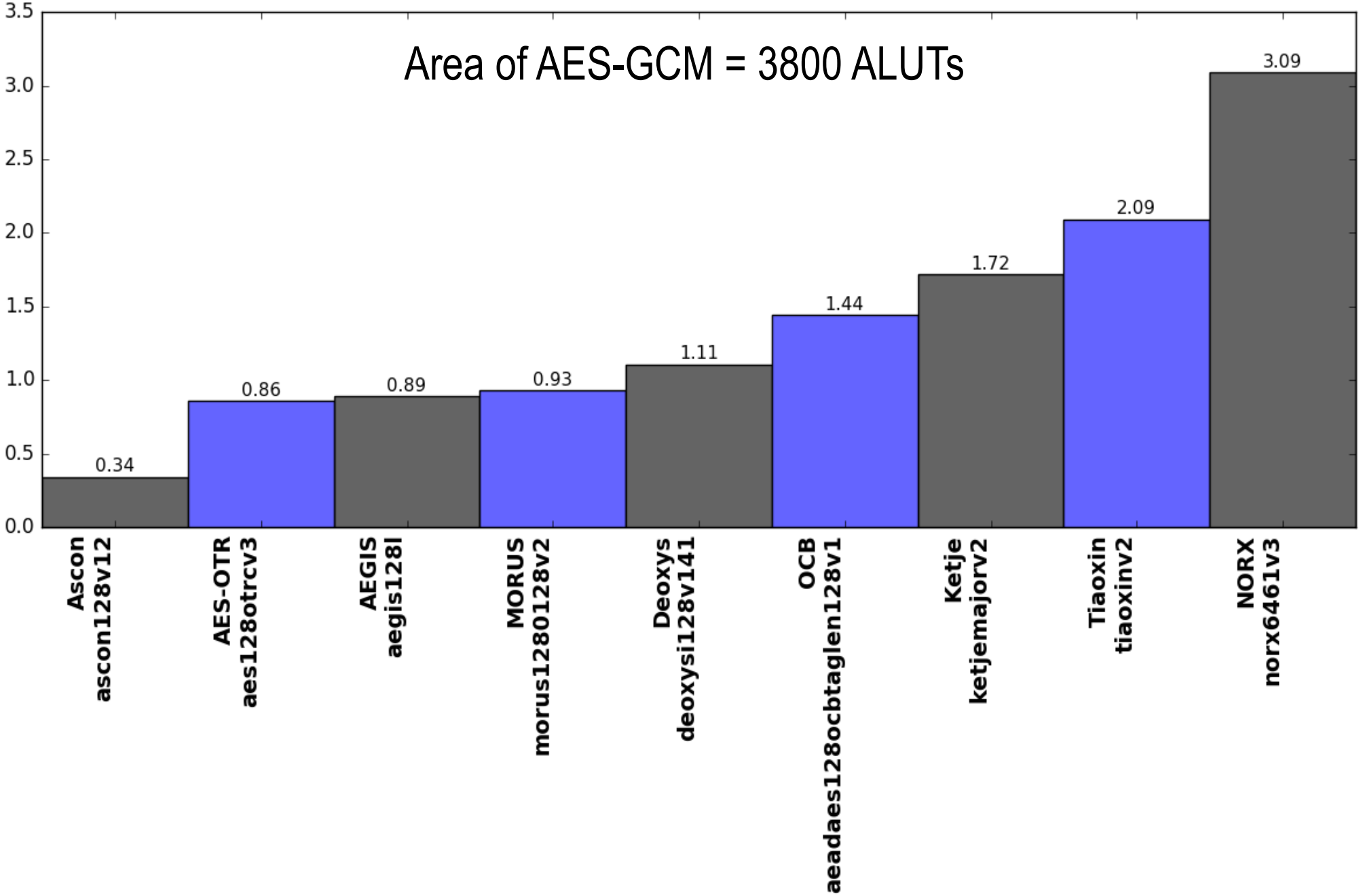


Relative Throughput in Stratix IV

Ratio of a given Cipher Throughput/Throughput of AES-GCM

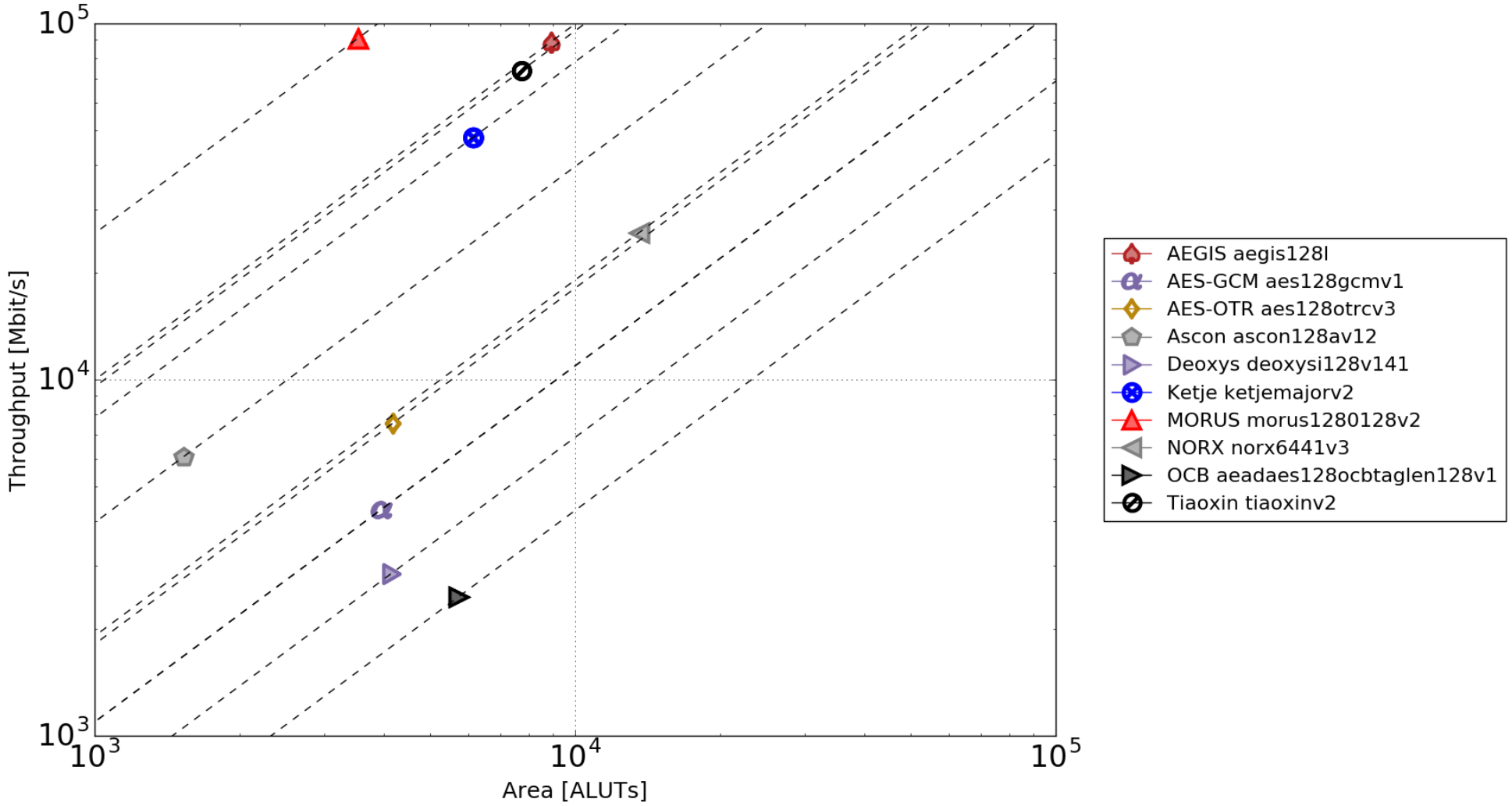


Relative Area (#ALUTs) in Stratix IV
Ratio of a given Cipher Area/Area of AES-GCM

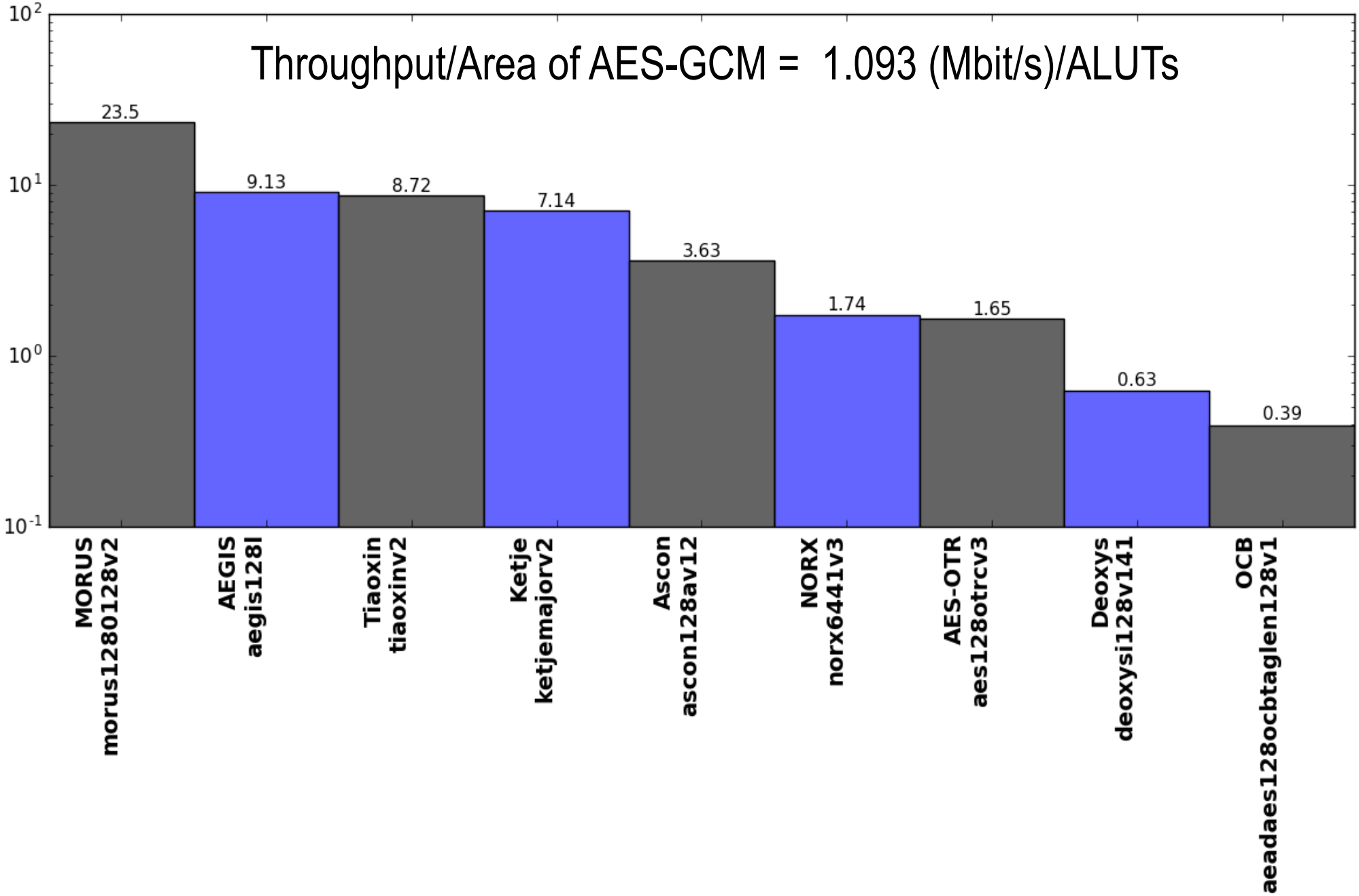


Stratix V

Results for Stratix V – Throughput vs. Area
Logarithmic Scale

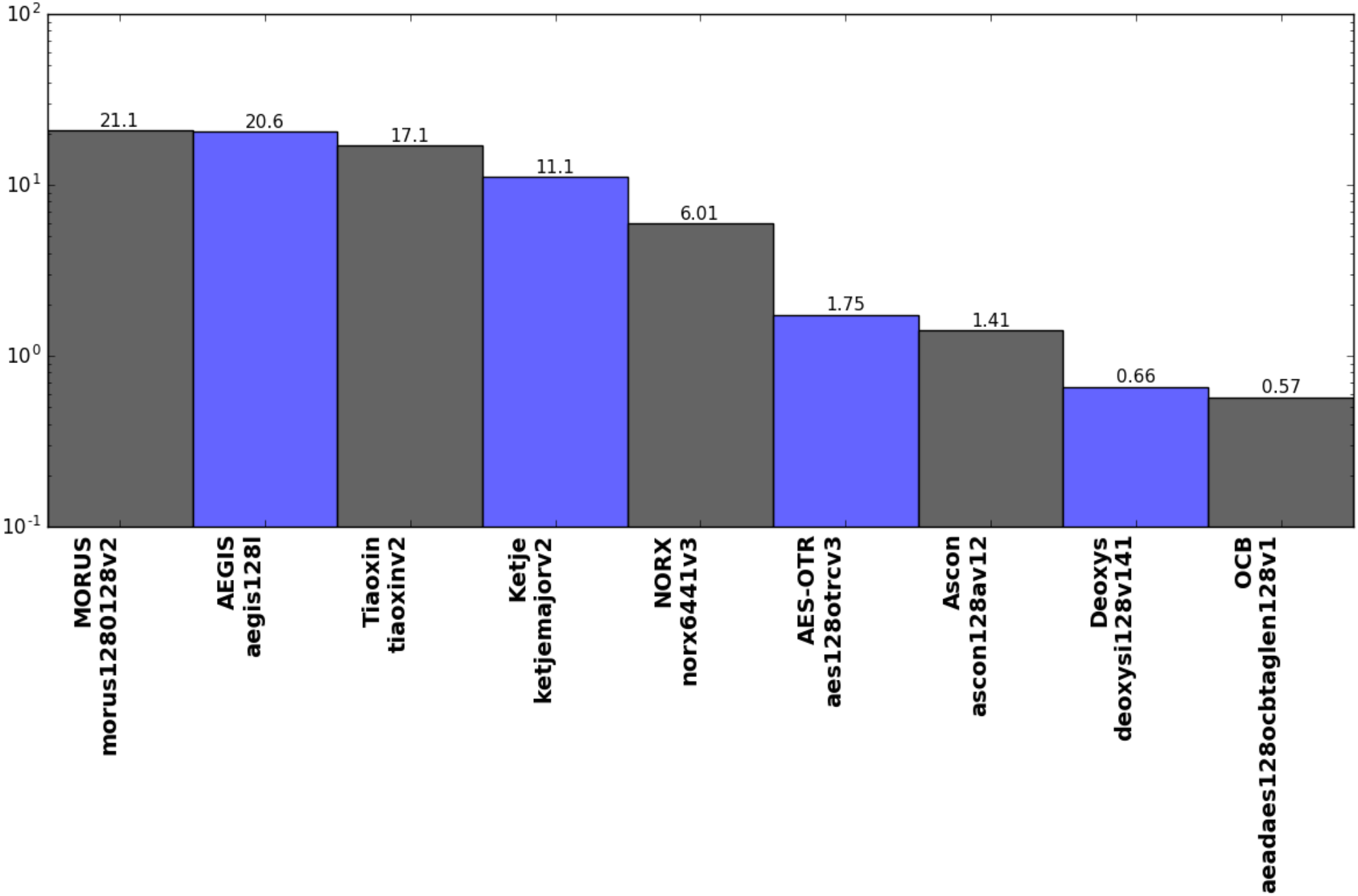


Relative Throughput/Area in Stratix V vs. AES-GCM



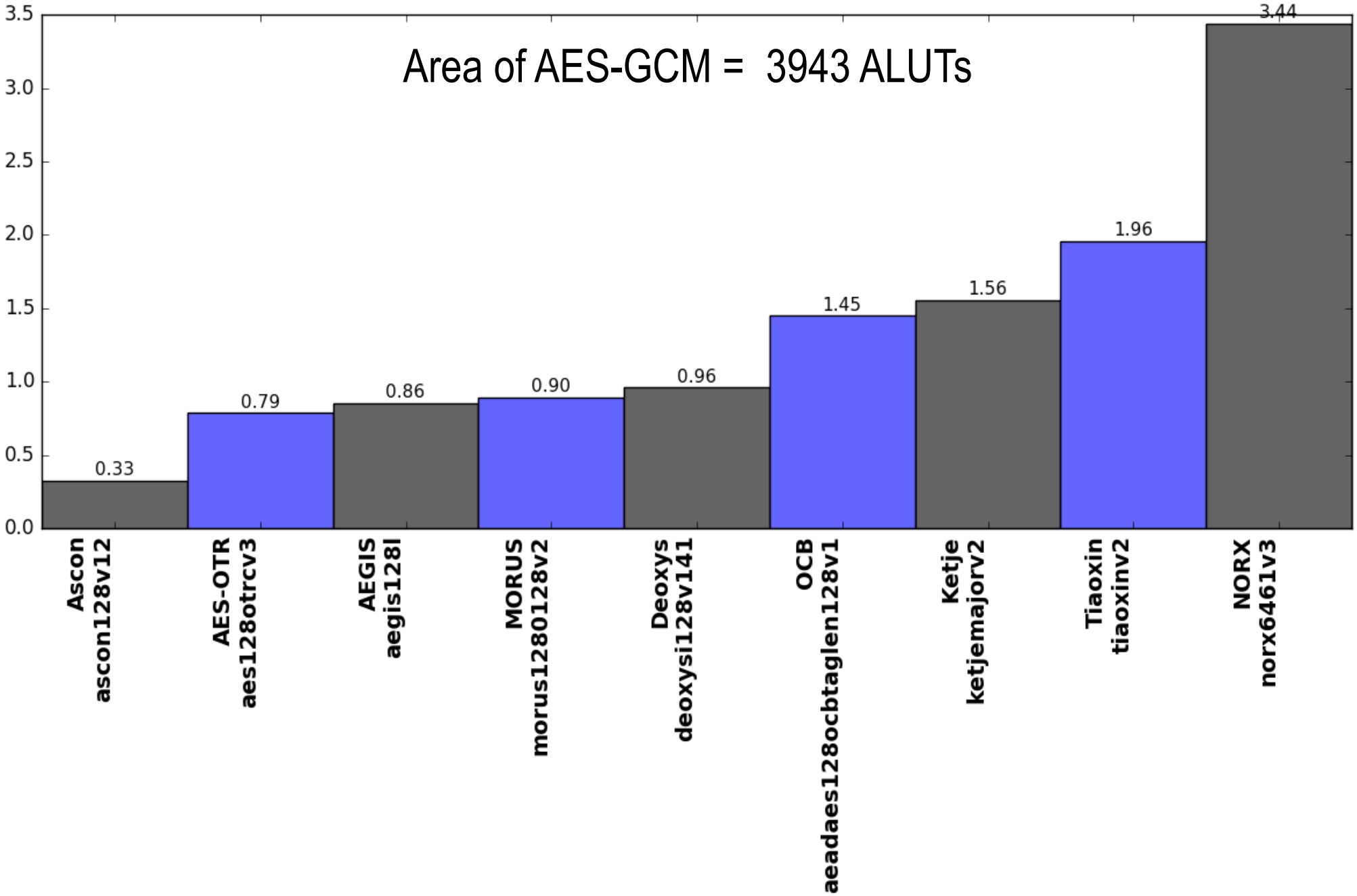
Relative Throughput in Stratix V

Ratio of a given Cipher Throughput/Throughput of AES-GCM



Relative Area (#ALUTs) in Stratix V

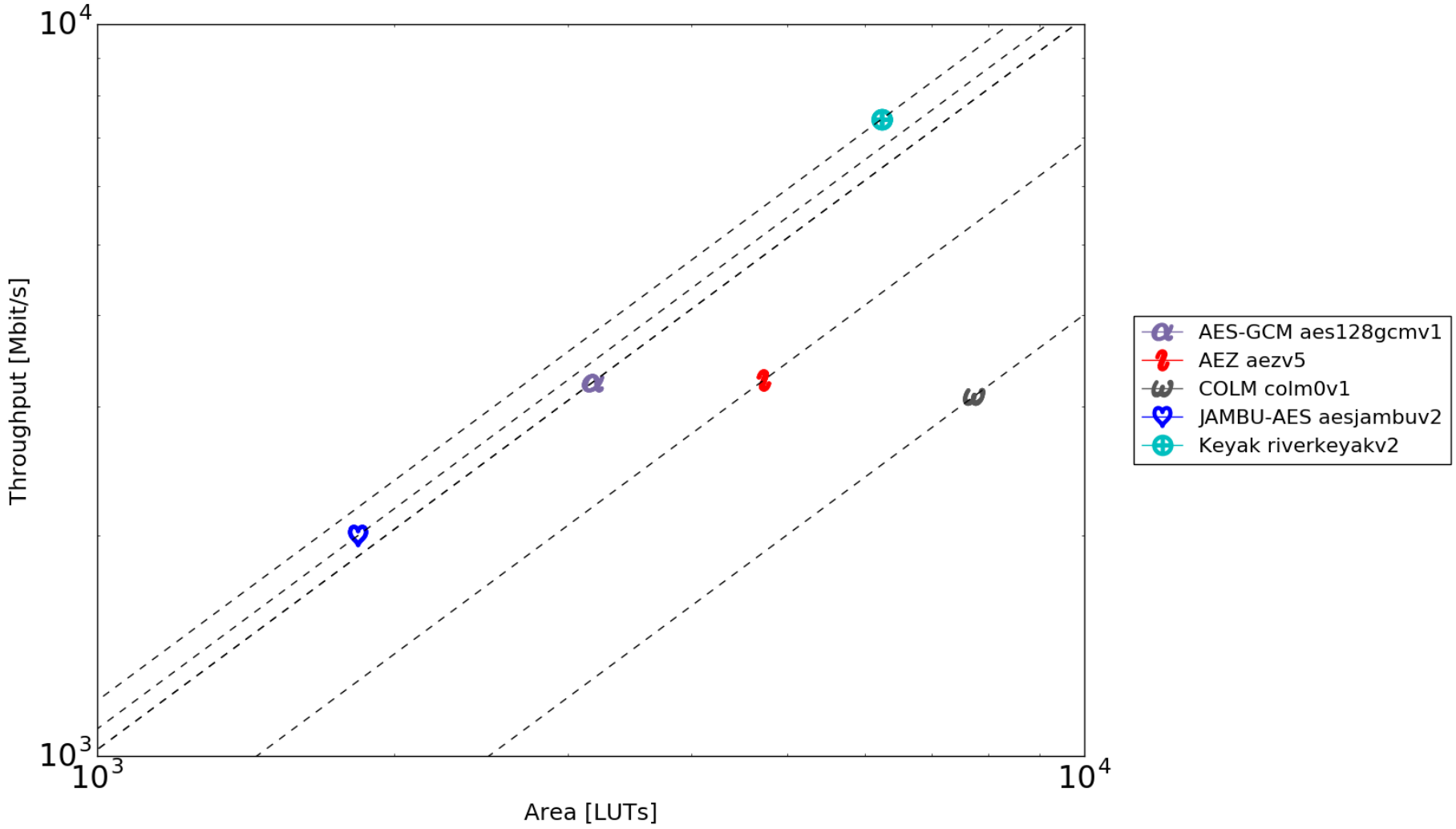
Ratio of a given Cipher Area/Area of AES-GCM



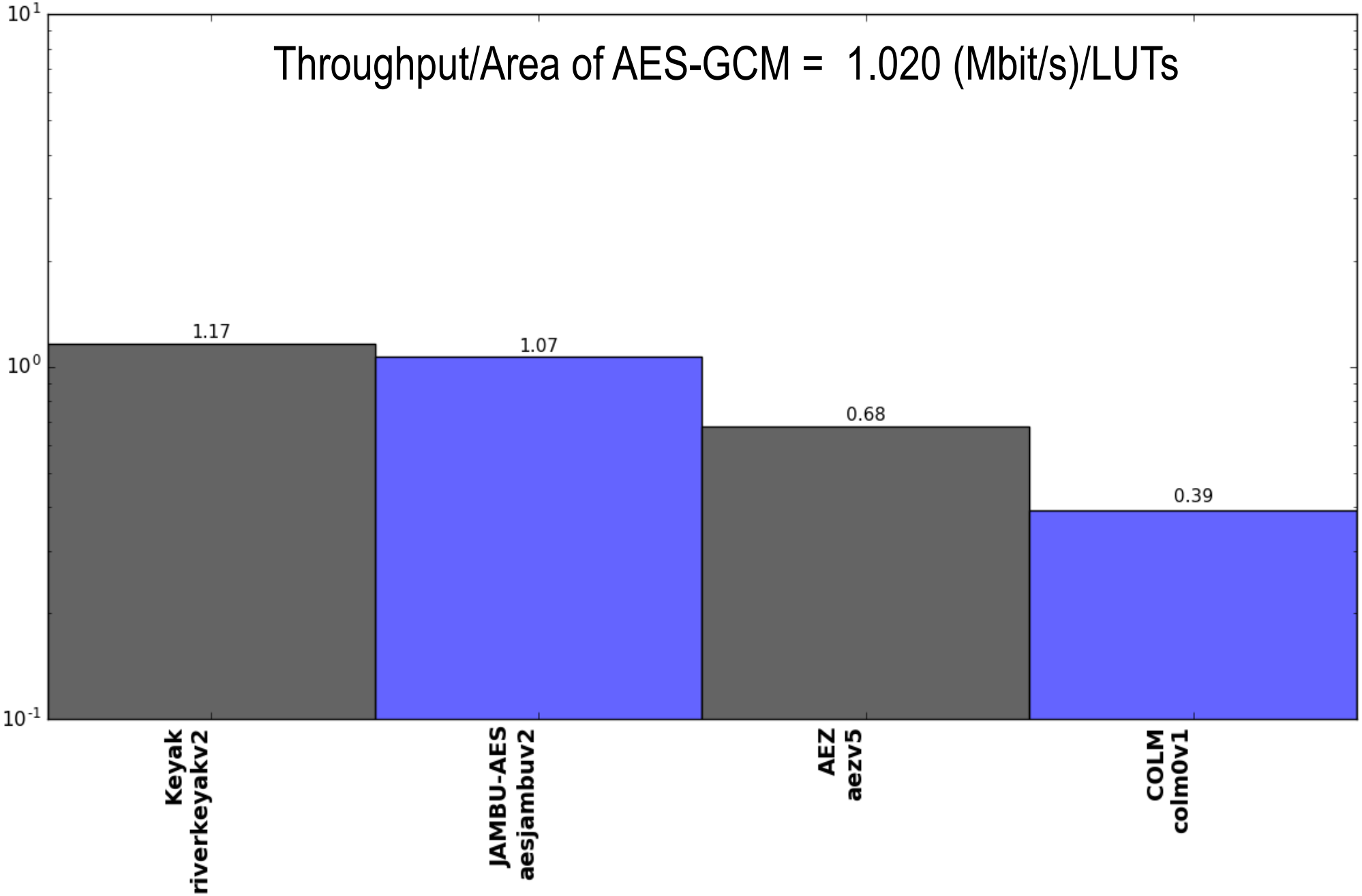
Use Case 3

Virtex-6

Results for Virtex-6 – Throughput vs. Area
Logarithmic Scale

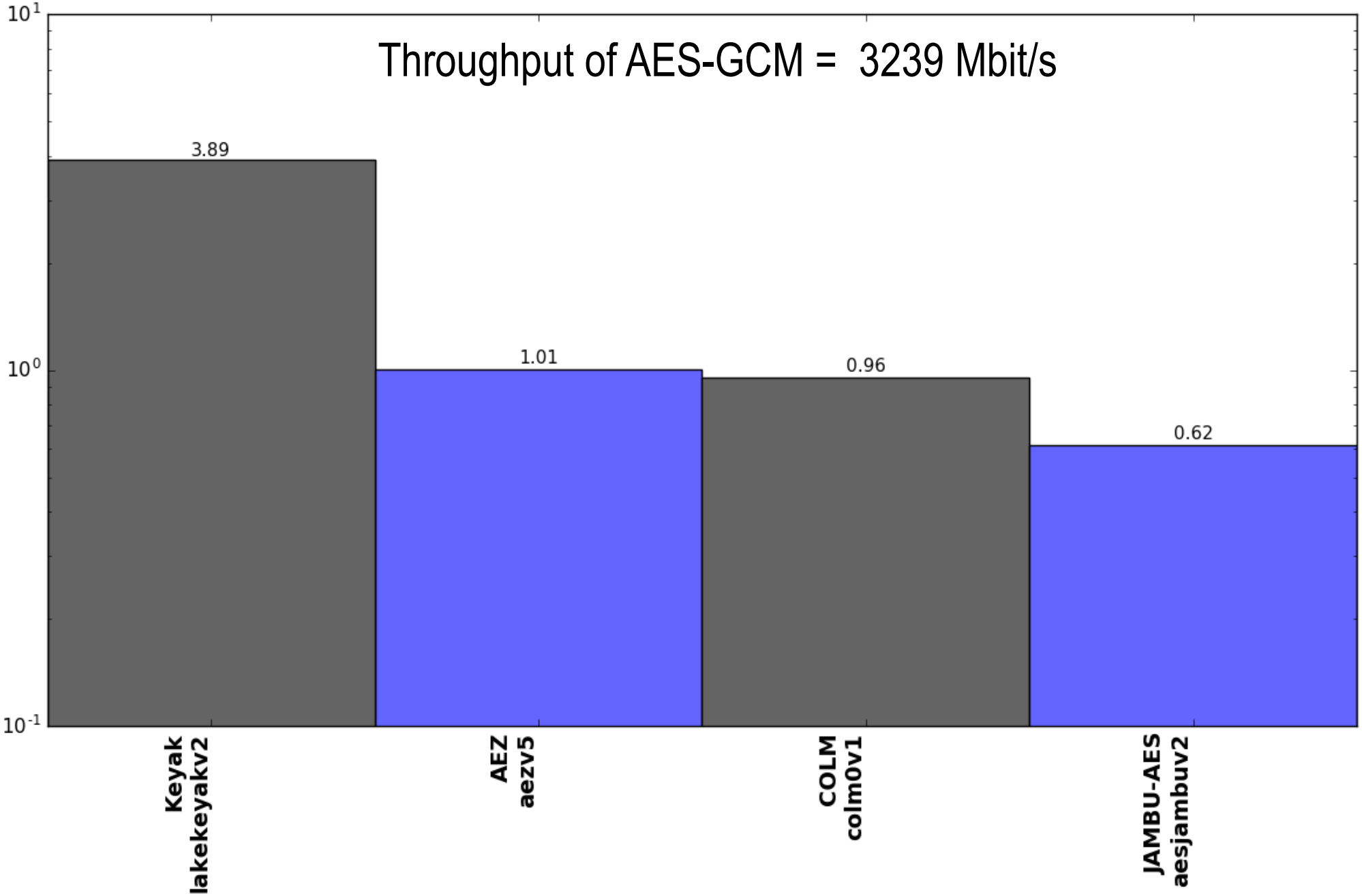


Relative Throughput/Area in Virtex-6 vs. AES-GCM



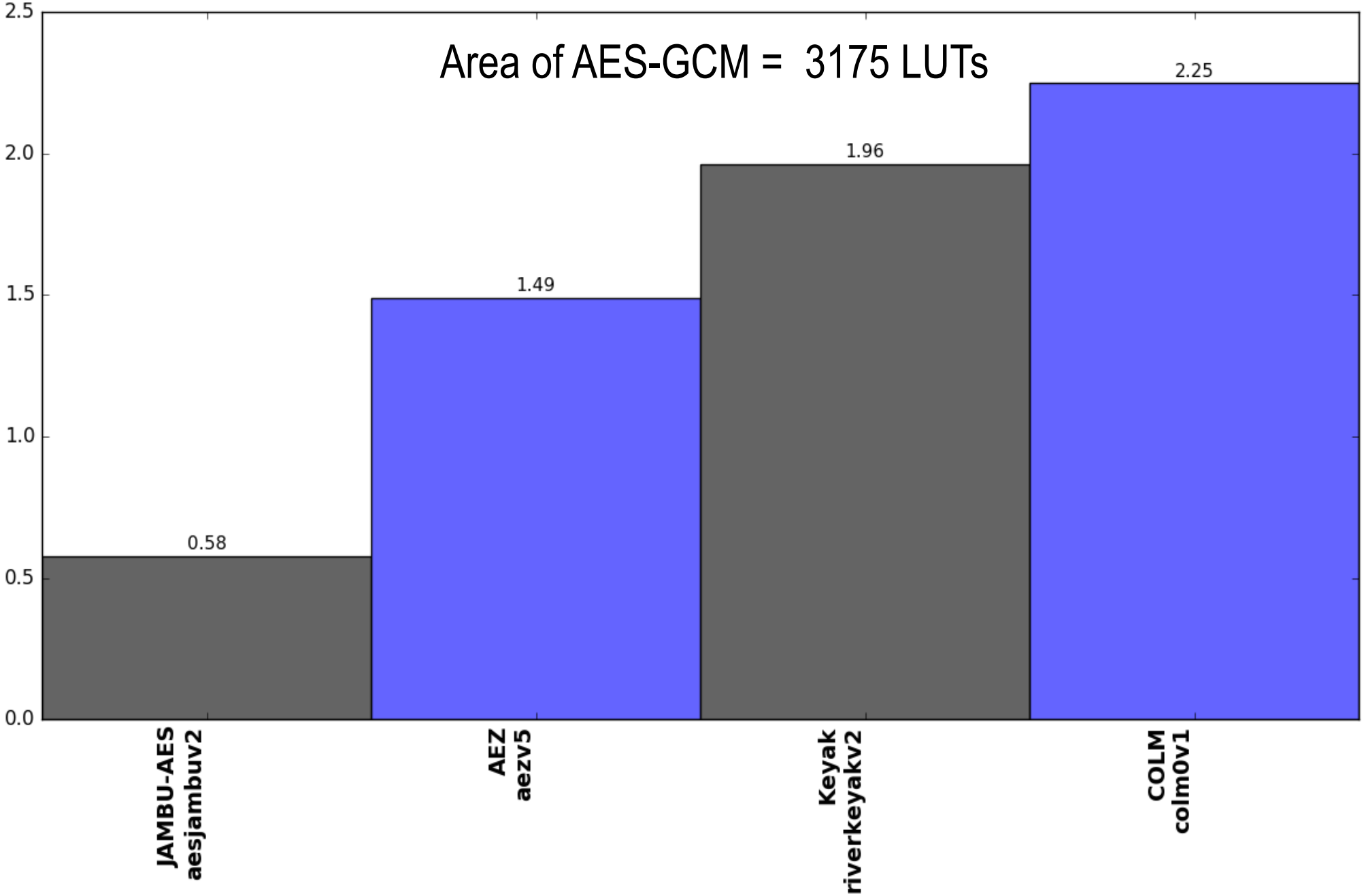
Relative Throughput in Virtex-6

Ratio of a given Cipher Throughput/Throughput of AES-GCM



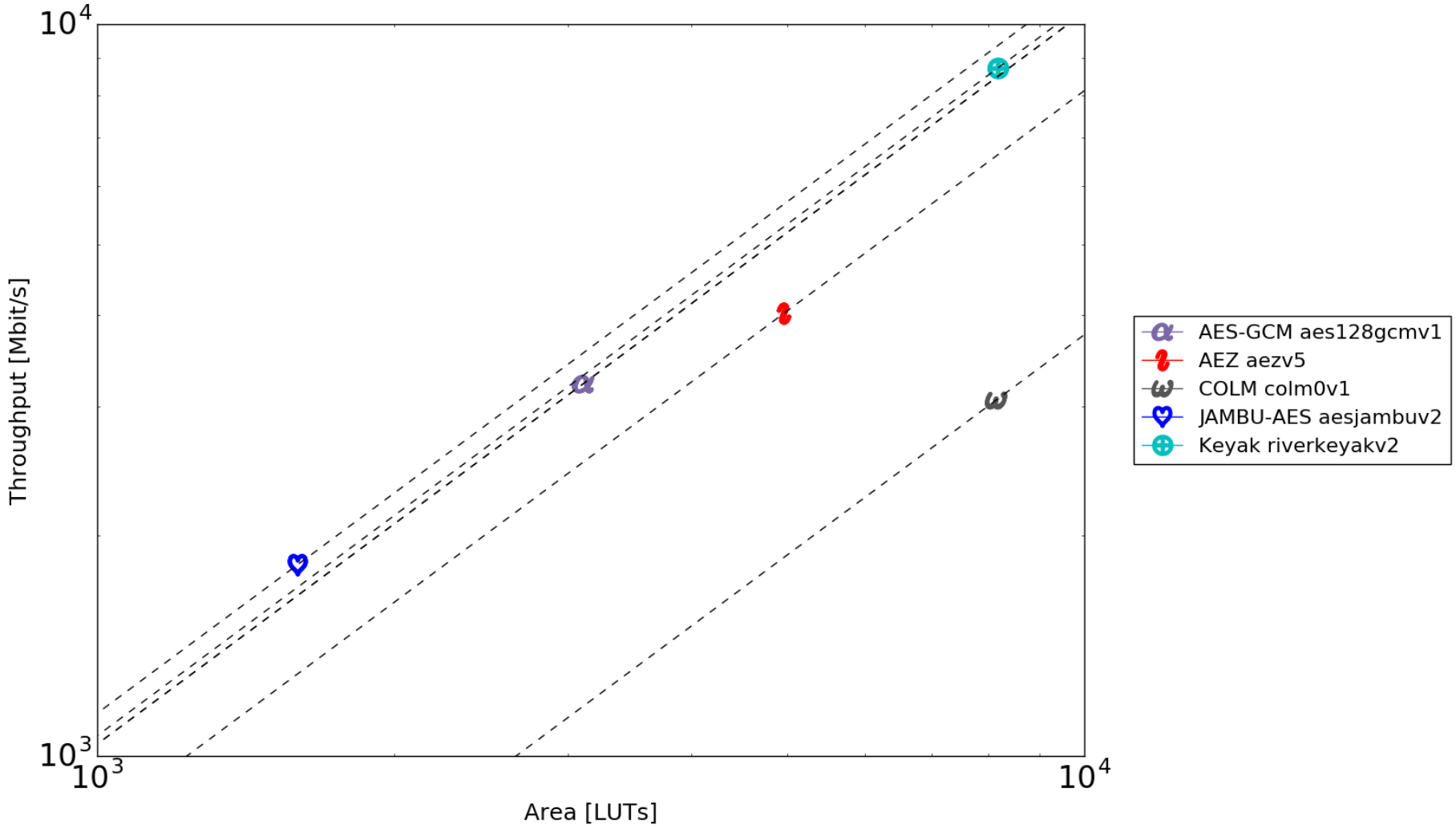
Relative Area (#LUTs) in Virtex-6

Ratio of a given Cipher Area/Area of AES-GCM

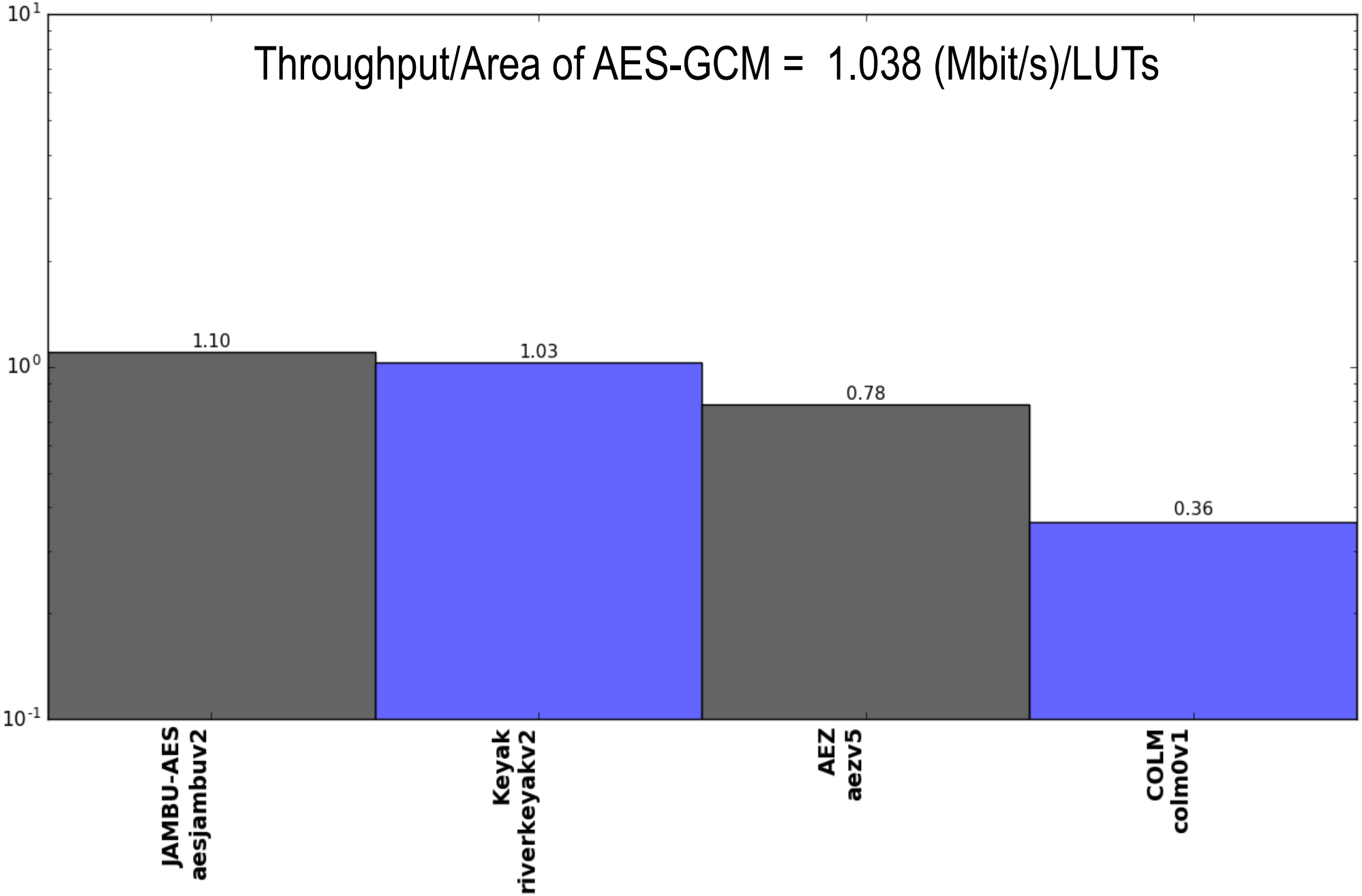


Virtex-7

Results for Virtex-7 – Throughput vs. Area
Logarithmic Scale

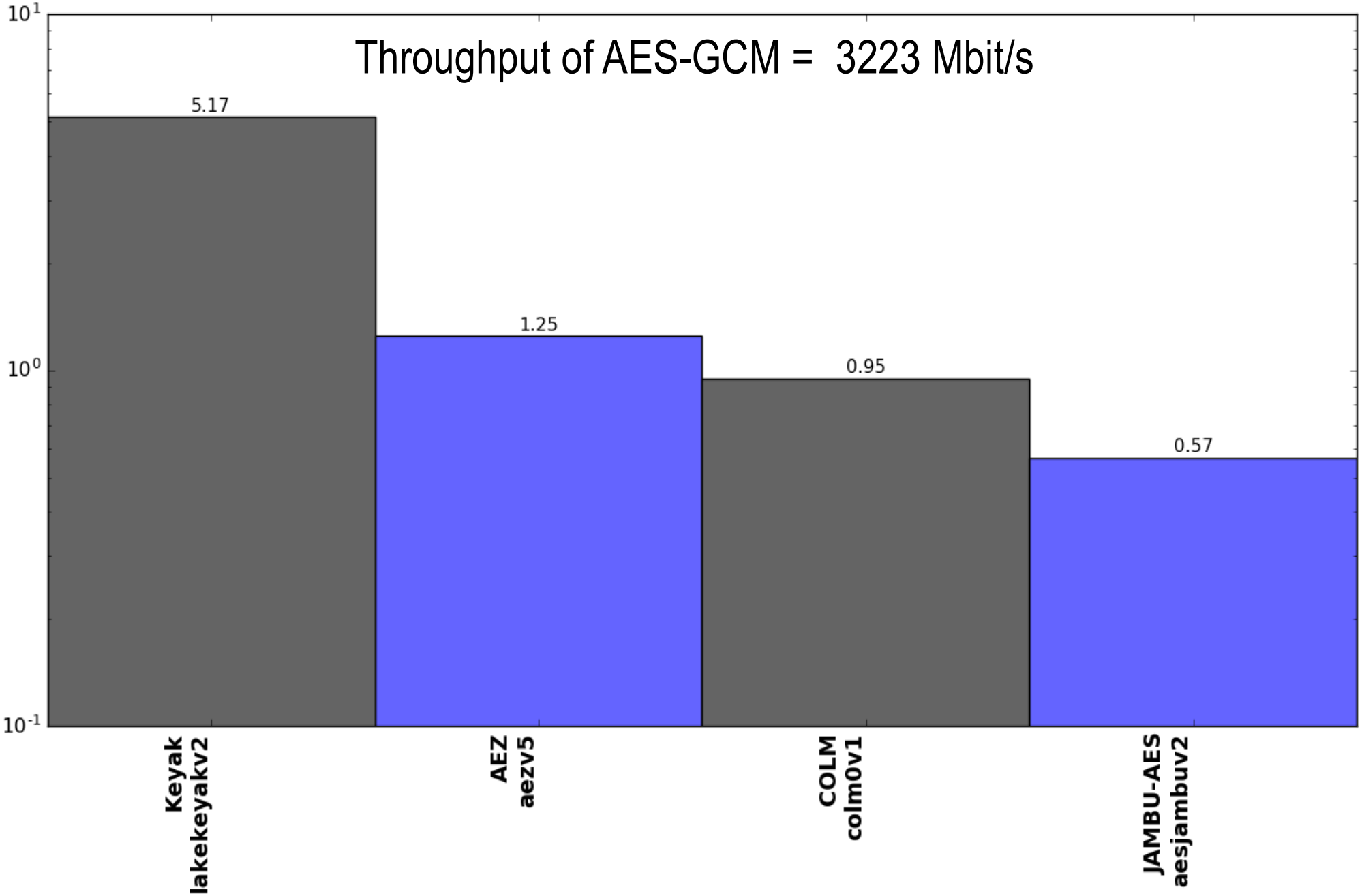


Relative Throughput/Area in Virtex-7 vs. AES-GCM



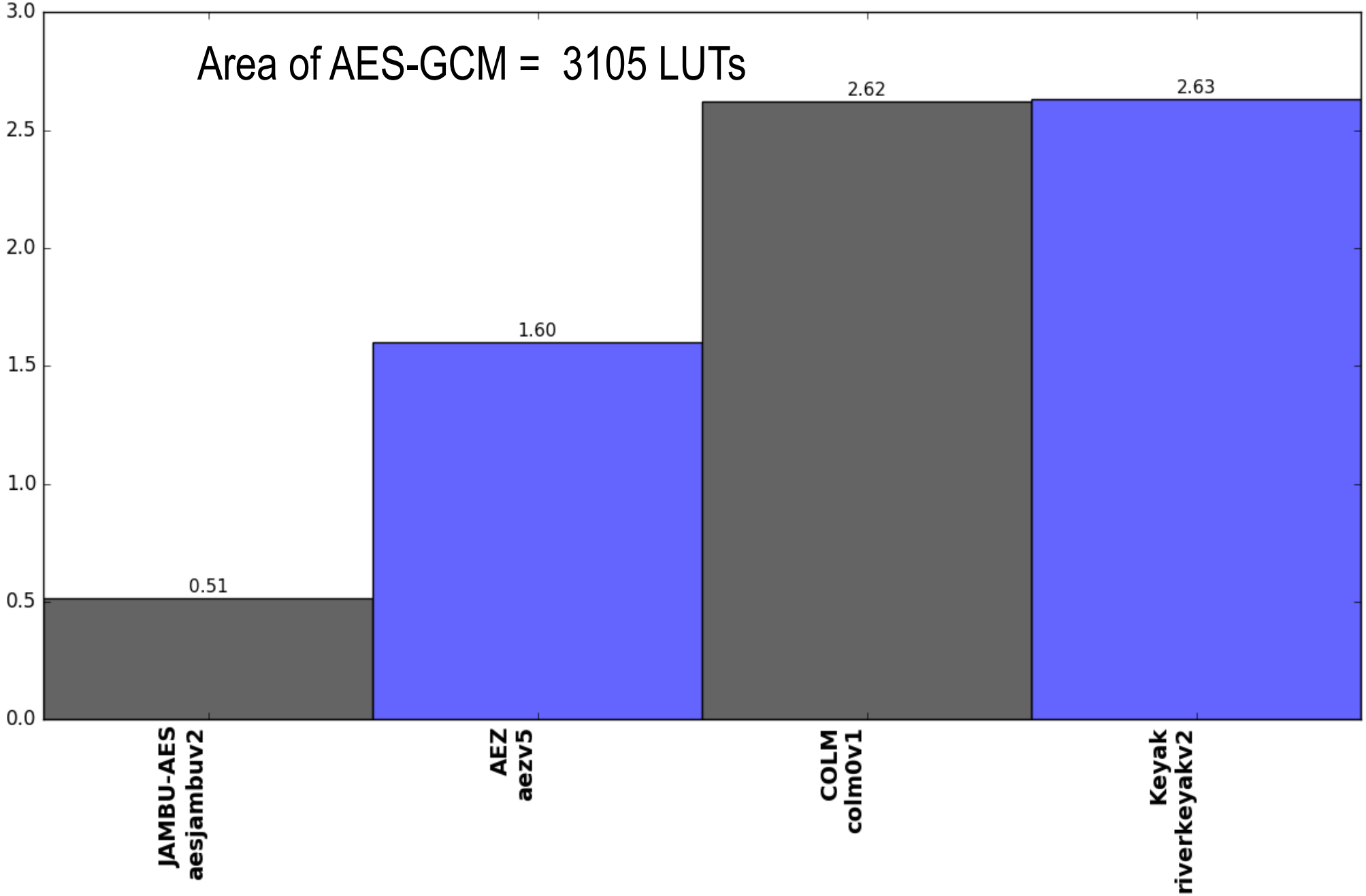
Relative Throughput in Virtex-7

Ratio of a given Cipher Throughput/Throughput of AES-GCM



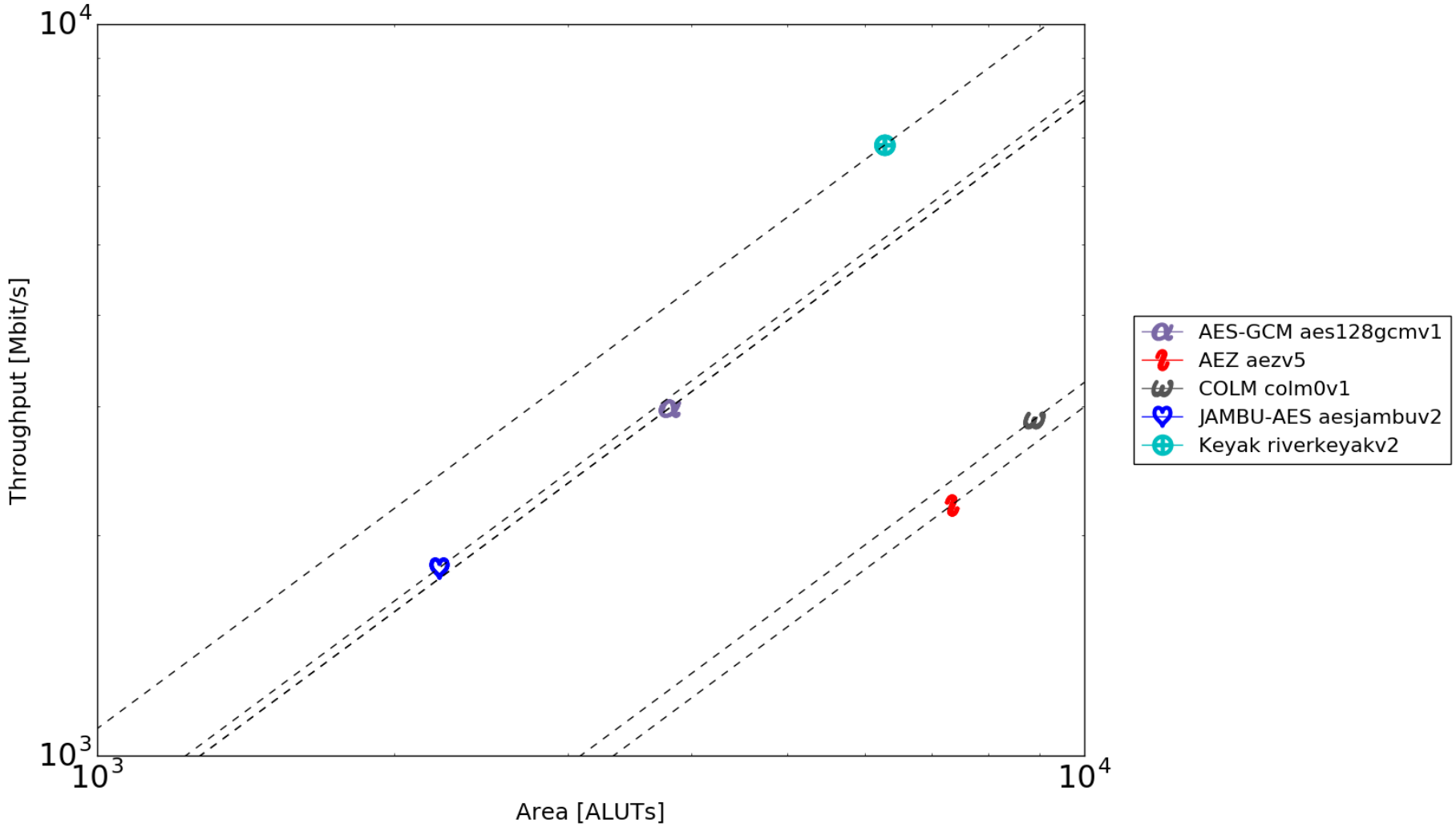
Relative Area (#LUTs) in Virtex-7

Ratio of a given Cipher Area/Area of AES-GCM

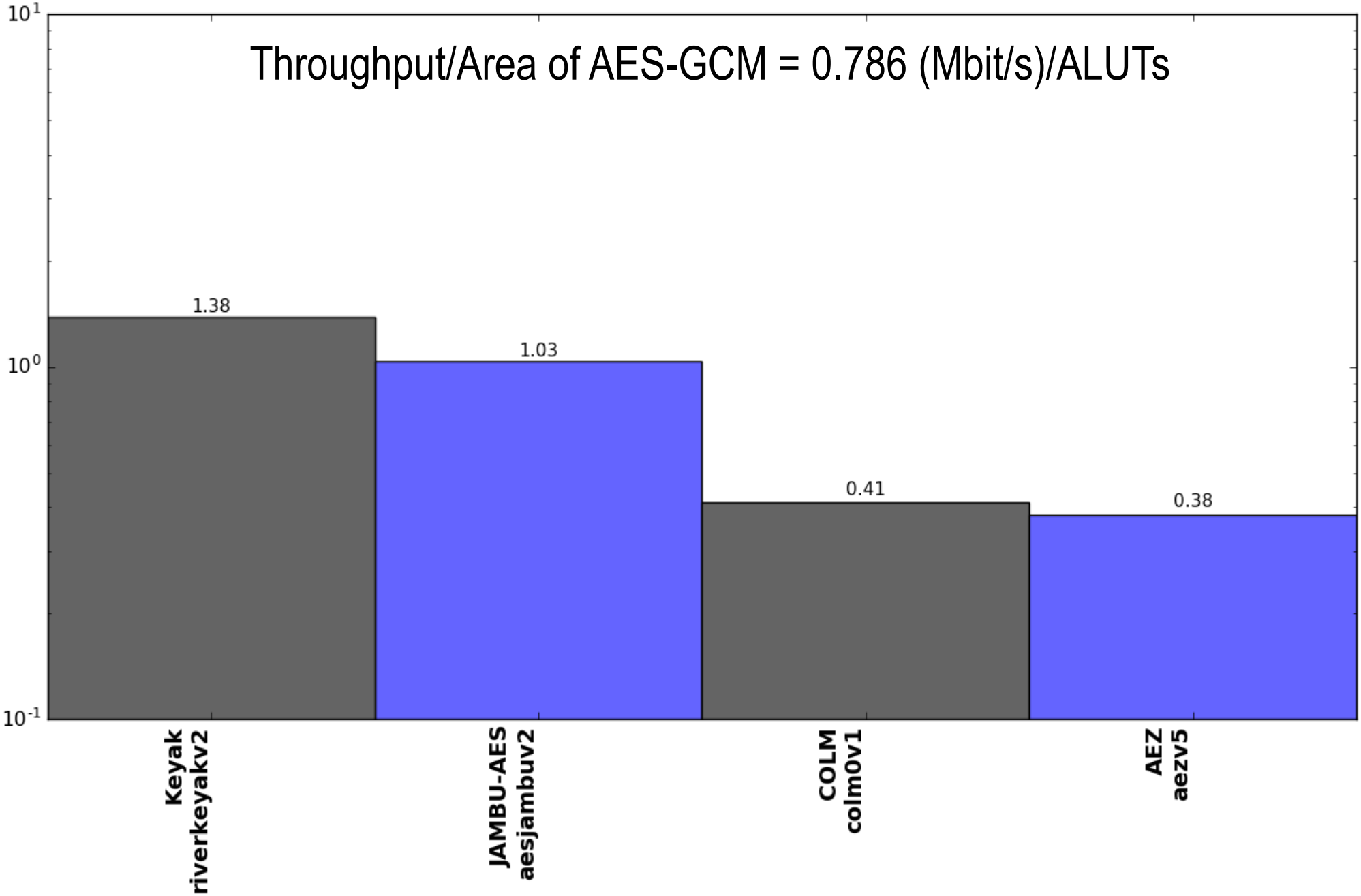


Stratix IV

Results for Stratix IV – Throughput vs. Area
Logarithmic Scale

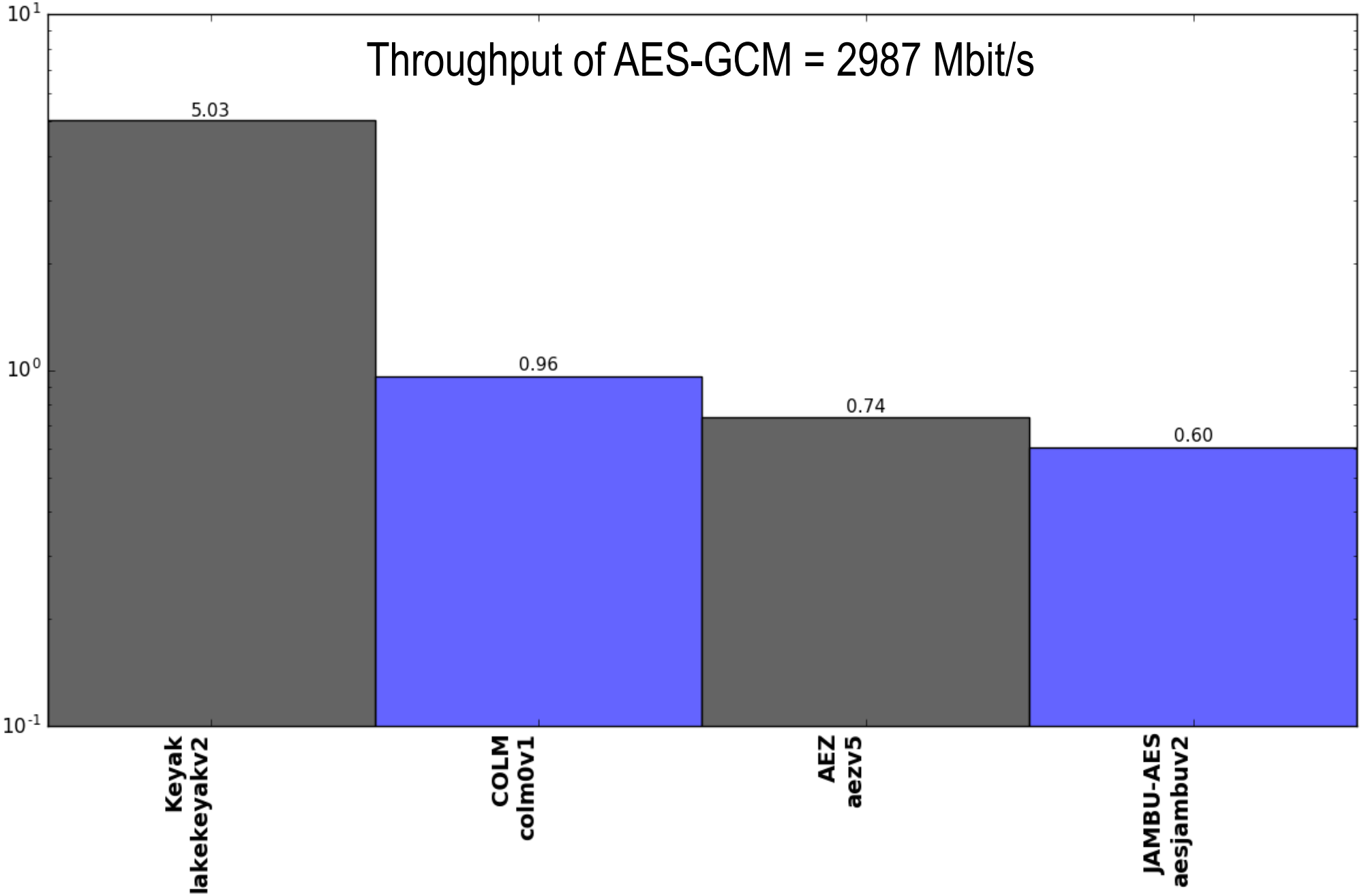


Relative Throughput/Area in Stratix IV vs. AES-GCM



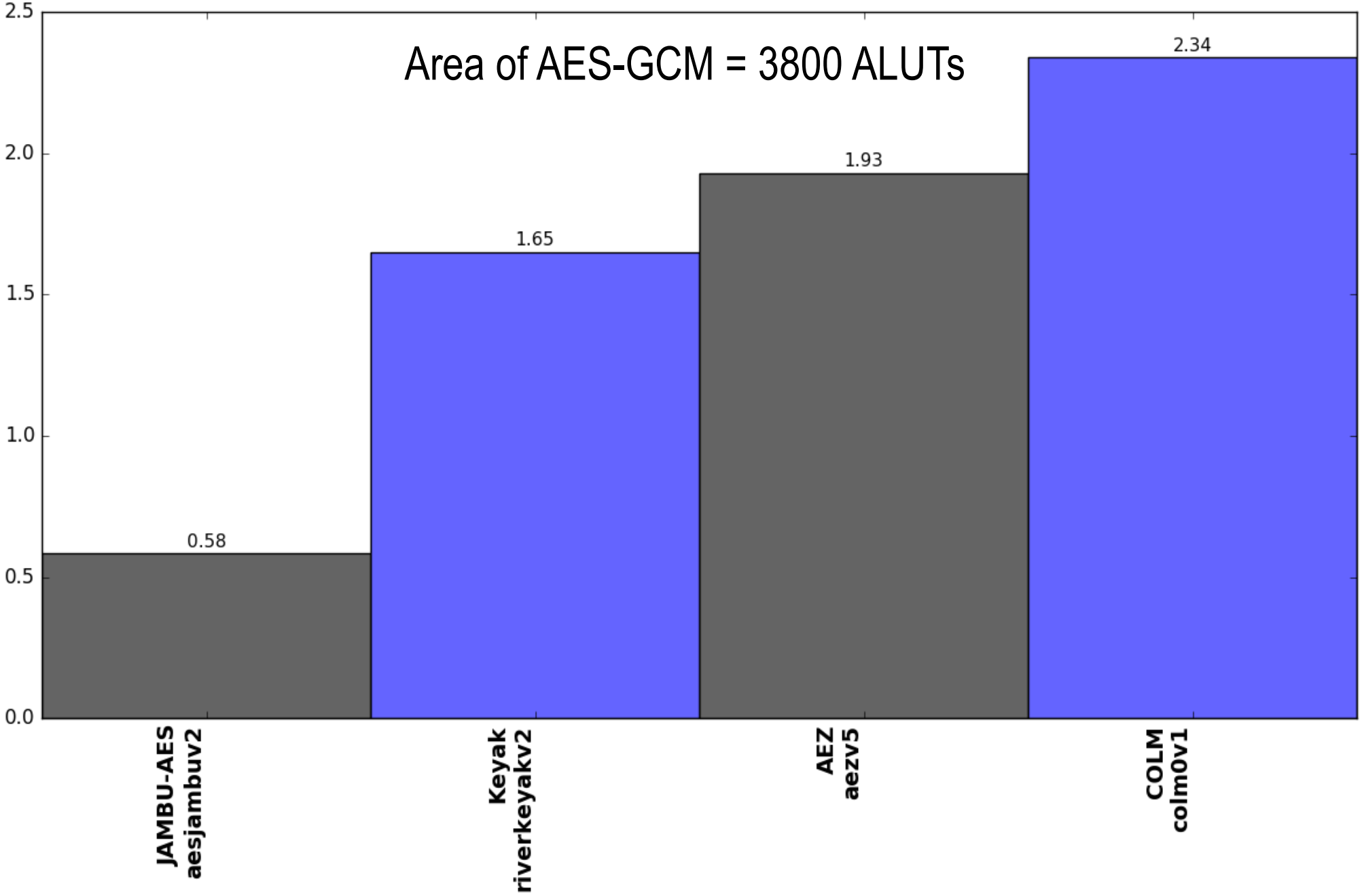
Relative Throughput in Stratix IV

Ratio of a given Cipher Throughput/Throughput of AES-GCM



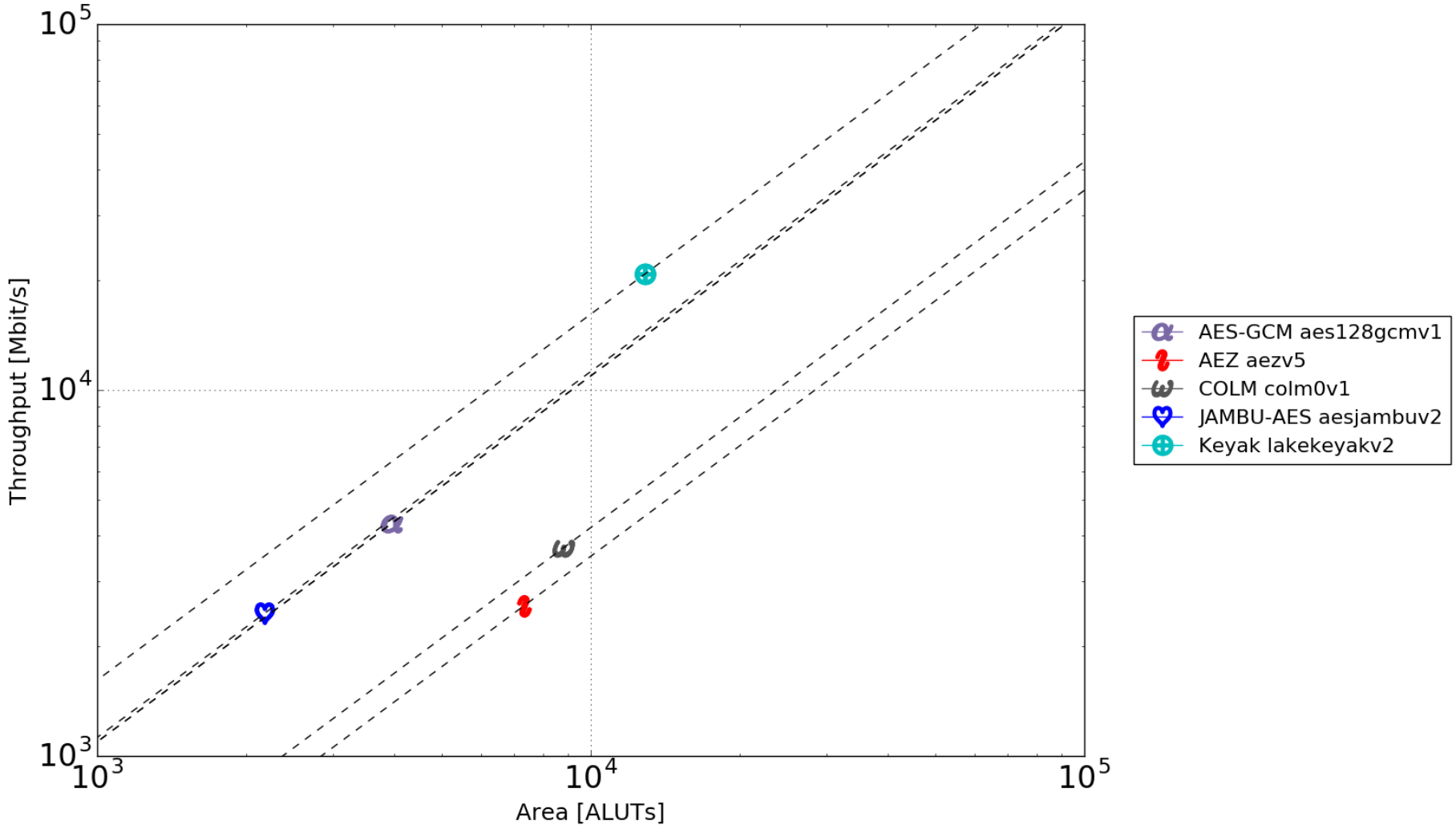
Relative Area (#ALUTs) in Stratix IV

Ratio of a given Cipher Area/Area of AES-GCM

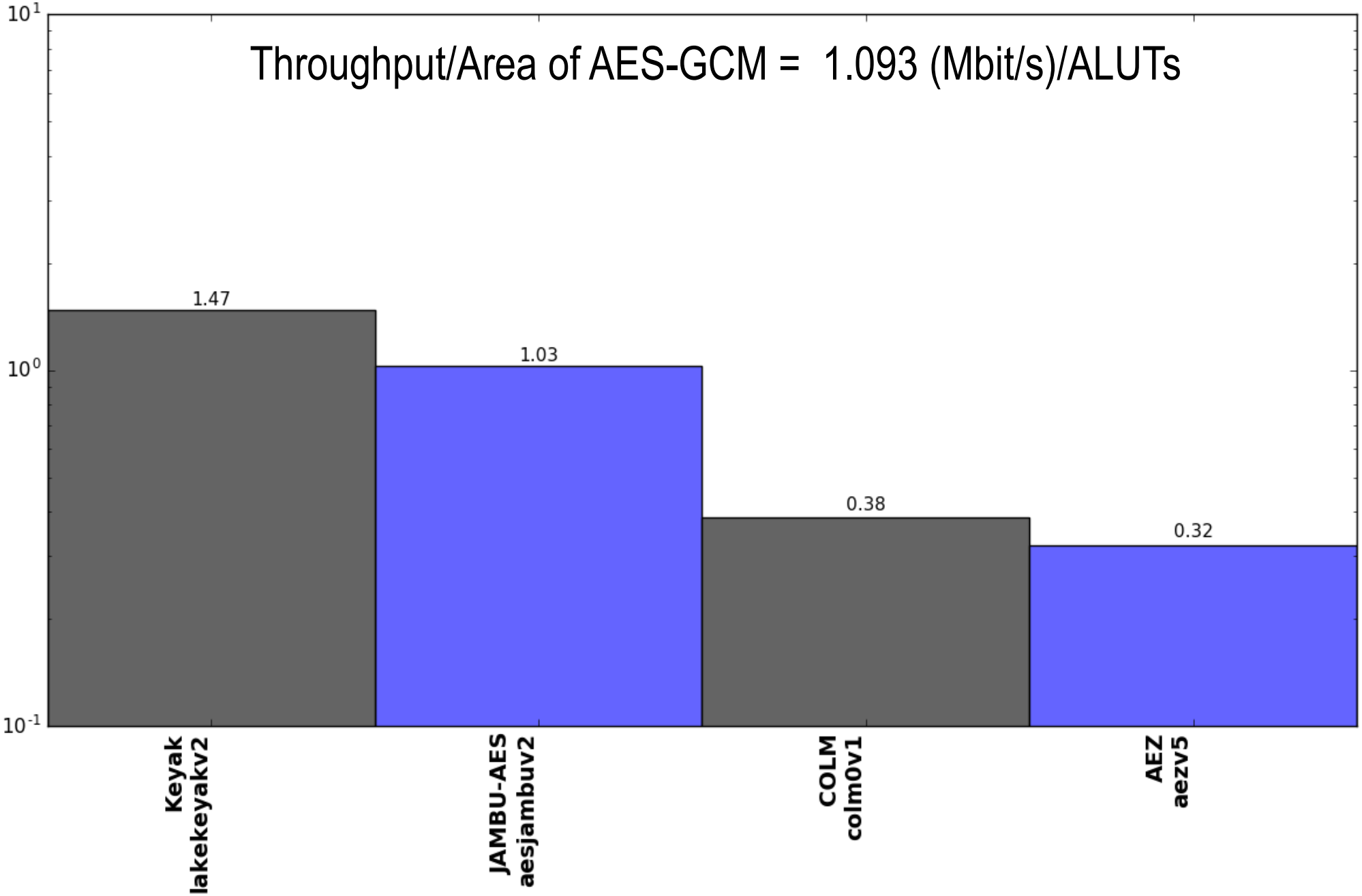


Stratix V

Results for Stratix V – Throughput vs. Area
Logarithmic Scale

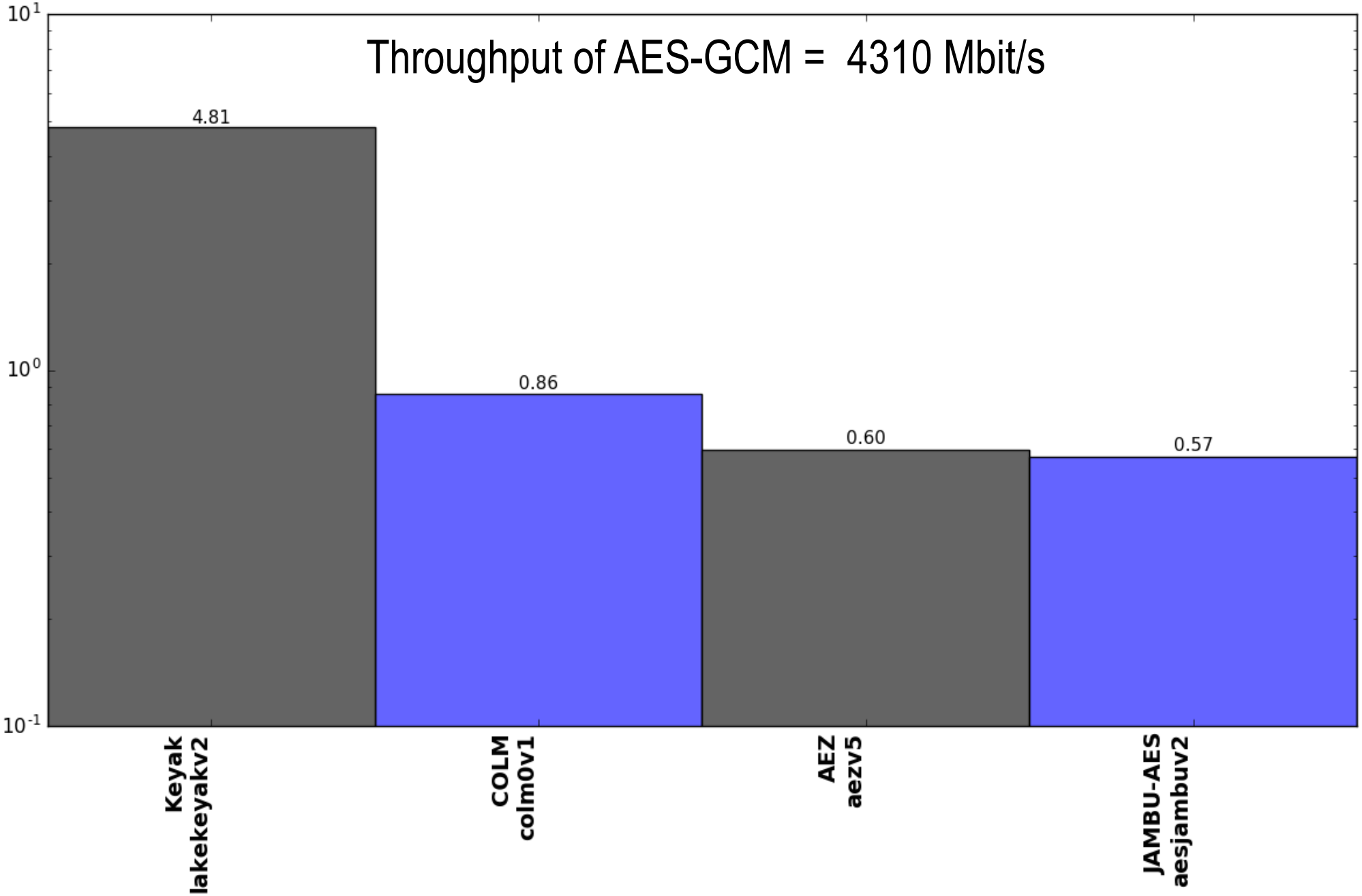


Relative Throughput/Area in Stratix V vs. AES-GCM

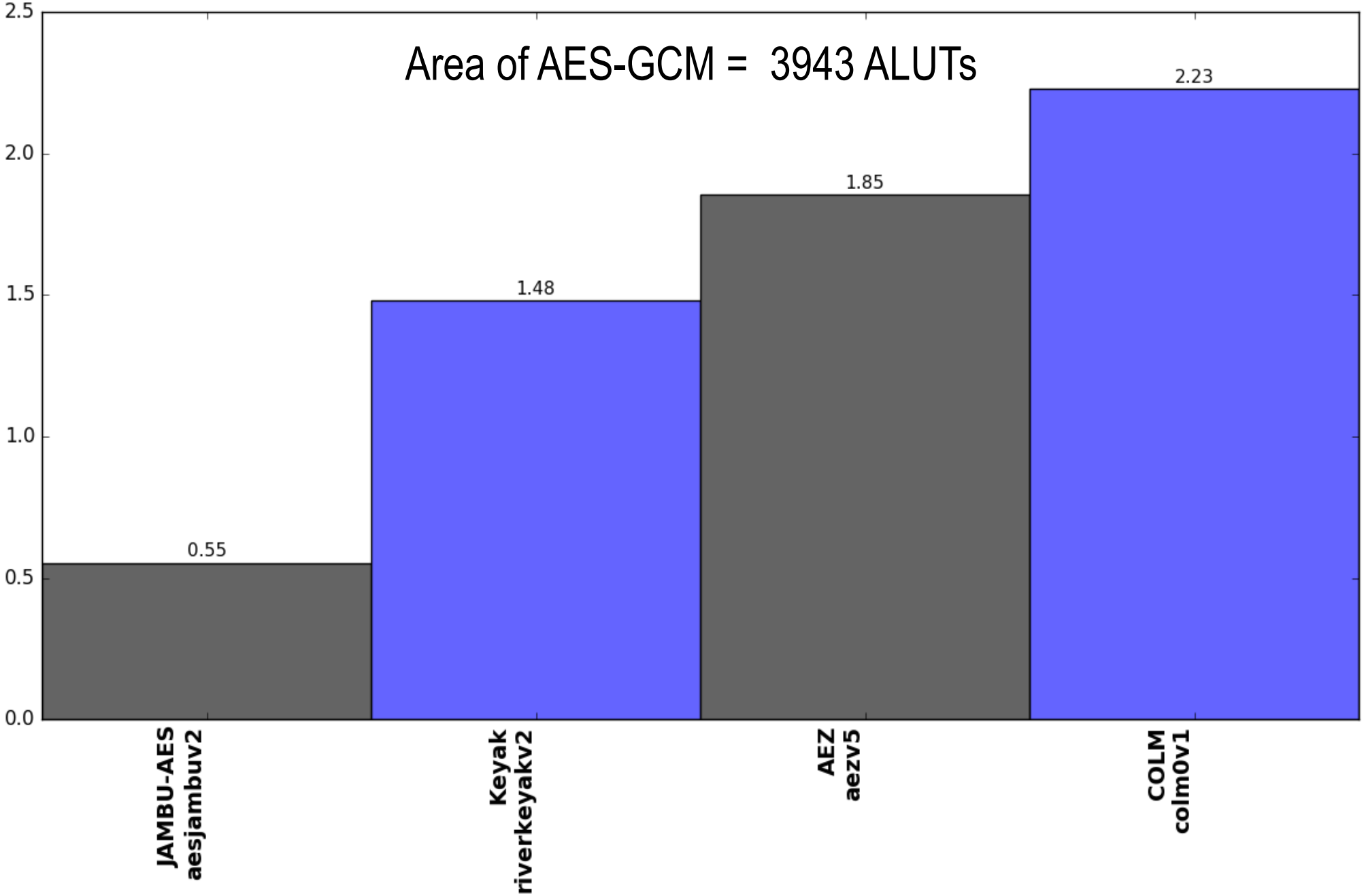


Relative Throughput in Stratix V

Ratio of a given Cipher Throughput/Throughput of AES-GCM



Relative Area (#ALUTs) in Stratix V
Ratio of a given Cipher Area/Area of AES-GCM



ATHENa Database of Results

ATHENa Database of Results

- Available at
  http://cryptography.gmu.edu/athena
- Developed by John Pham, a Master's-level student of Jens-Peter Kaps as a part of the SHA-3 Hardware Benchmarking project, 2010-2012, (sponsored by NIST)
- In June 2015 extended to support Authenticated Ciphers
- In July 2017 extended to support the CAESAR Use Cases and ranking of candidate variants

Two Views

- **Rankings View**
  https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/rankings_view
  - Easier to use
  - Provides Rankings

- **Table View**
  https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/table_view
  - More comprehensive
  - Allows close investigation of all designs & comparative analysis
  - Geared toward more advanced users
  - On-line help

Hints on Using the Rankings View

- After each change of options, click on **Update**
- If you want to return to the default settings, please click on

  FPGA Rankings,

  in the menu located on the left side of the page
- If you want to limit the key size to a particular range, please choose the option

  Key size:

  From <min> To: <max>
- You can further narrow down your search by using

  Min Area:

  Max Area:

  Min Throughput:

  Max Throughput:

Hints on Using the Rankings View

- For the results of High-Speed Benchmarking, choose

  Family:

  - **Virtex-6 (default)**
  - **Virtex-7**
  - **Stratix IV**
  - **Stratix V**

Hints on Using the Rankings View

- You can switch between ranking criteria by using the option:

  Ranking:

  [X] Throughput/Area

  [ ] Throughput

  [ ] Area

- **Unit of Area:**

  allows you to choose between two alternative units of area for each type of FPGA:

  - for Xilinx Virtex-6, Virtex-7: **LUTs and Slices**
  - for Altera Stratix IV, Stratix V: **ALUTs and ALMs.**

Please note that after each change a different variant may be used to represent a given family of authenticated ciphers.
The displayed variant is the best in terms of the current ranking criteria.

One Stop Website

https://cryptography.gmu.edu/athena/index.php?id=CAESAR
OR
https://cryptography.gmu.edu/athena
and click on CAESAR

- VHDL/Verilog Code of CAESAR Candidates: Summary I
- VHDL/Verilog Code of CAESAR Candidates: Summary II
- ATHENa Database of Results: Rankings View
- ATHENa Database of Results: Table View
- Benchmarking of Round 3 CAESAR Candidates in Hardware: Methodology, Designs & Results [this presentation]
- GMU Implementations of Authenticated Ciphers and Their Building Blocks
- CAESAR Hardware API v1.0

Conclusions

- **Results for Use Case 2, High-performance applications**, should have **strong influence on the selection** of the final portfolio in this category
  - High-speed hardware architectures matching the intended applications
  - No major changes in rankings since Round 2
- **Results for Use Case 3, Defense in depth**, may be **used to resolve ties** between candidates with very similar security properties. However,
  - Candidates differ substantially in terms of their enhanced security features
  - No results for Deoxys-II
  - Difficulty in comparing single-pass and two-pass algorithms
- **Results for Use Case 1, Lightweight applications, very preliminary**. Much more development effort required.

Thank you!

Comments?

Questions?

Suggestions?

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

CERG: http://cryptography.gmu.edu