Efficient Verification of an Elaboration-Time, Key Size Configurable, Pipelined AES Encoder and Decoder using a Mentor Veloce Emulator

Alex Pearson

Daniel Collins

Outline

- Advanced Encryption Standard Overview
  - Round Computation
  - Key Schedule
- Implementation
  - Block Diagram
  - Design/Verification Methodology
- Simulation and Emulation Results

Advanced Encryption Standard Overview

AES Overview

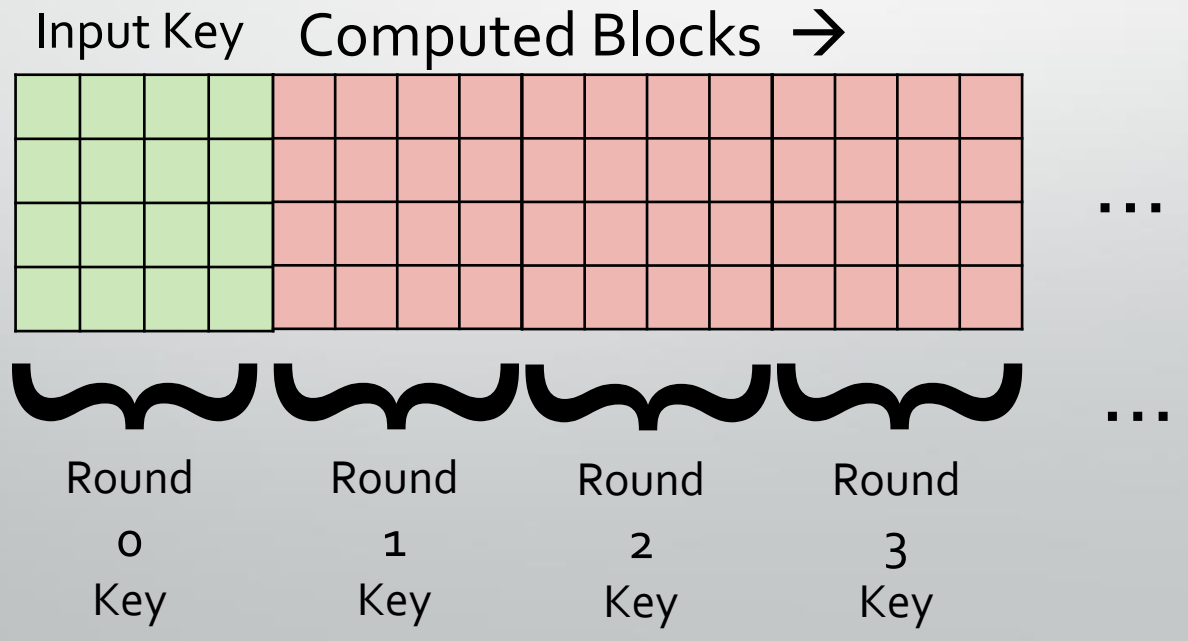
- AES is a symmetric-key block cipher used to encrypt electronic data
  - Specified in the FIPS 197 document published by NIST in 2001
  - Operates on 128-bit blocks of data
  - Supports three key sizes: 128, 192, and 256 bits
  - Widely adopted, designed with hardware implementations in mind
- Performs multiple rounds of state transformation on the input block to produce the output

AES Overview
Round Computation

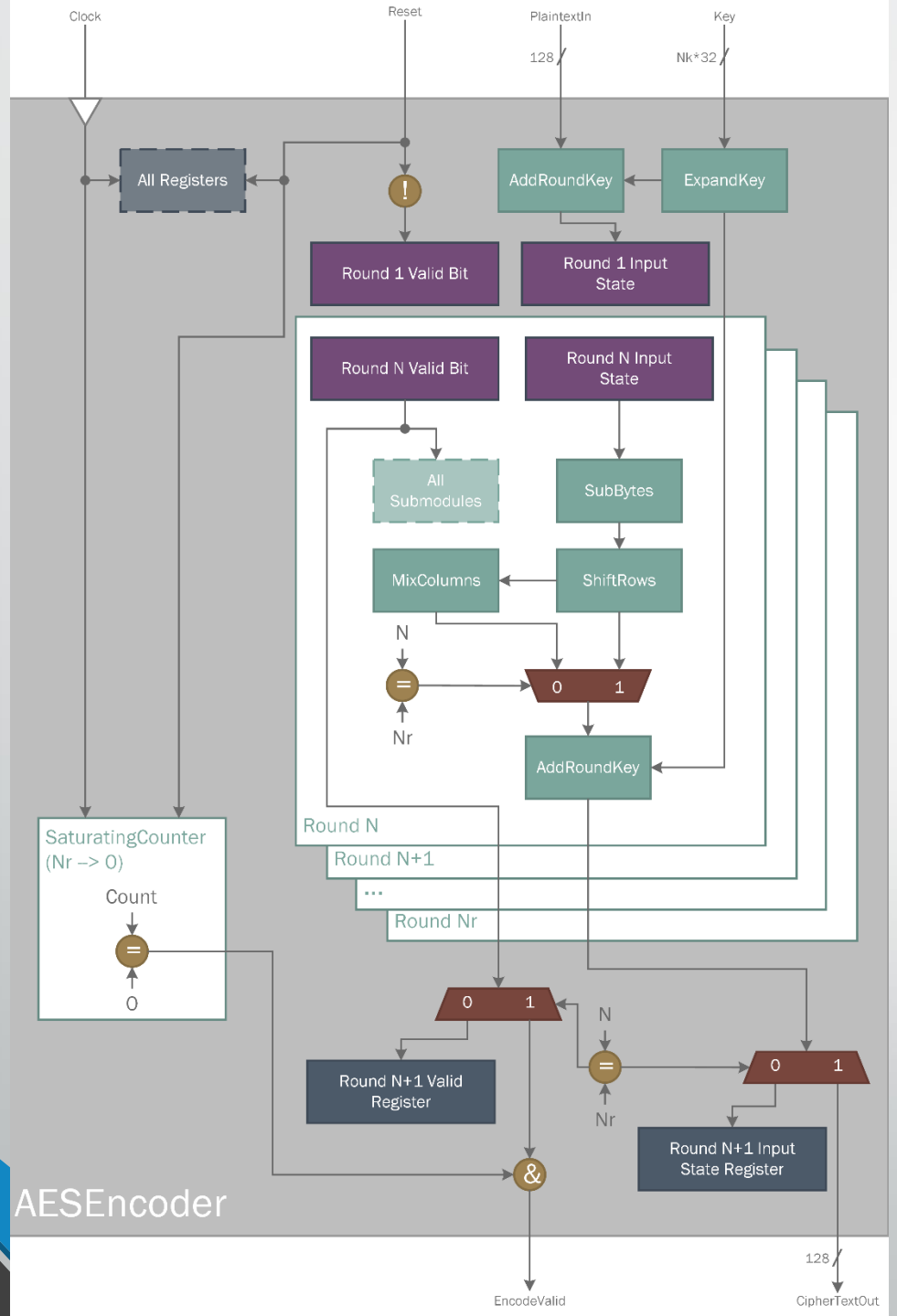
- Each AES variant performs a fixed number of *rounds*
  - AES-128: 10 rounds
  - AES-192 12 rounds
  - AES-256: 14 rounds
- Each round applies four *transformations* to the state
  - SubBytes
  - ShiftRows
  - MixColumns (omitted on the final round)
  - AddRoundKey

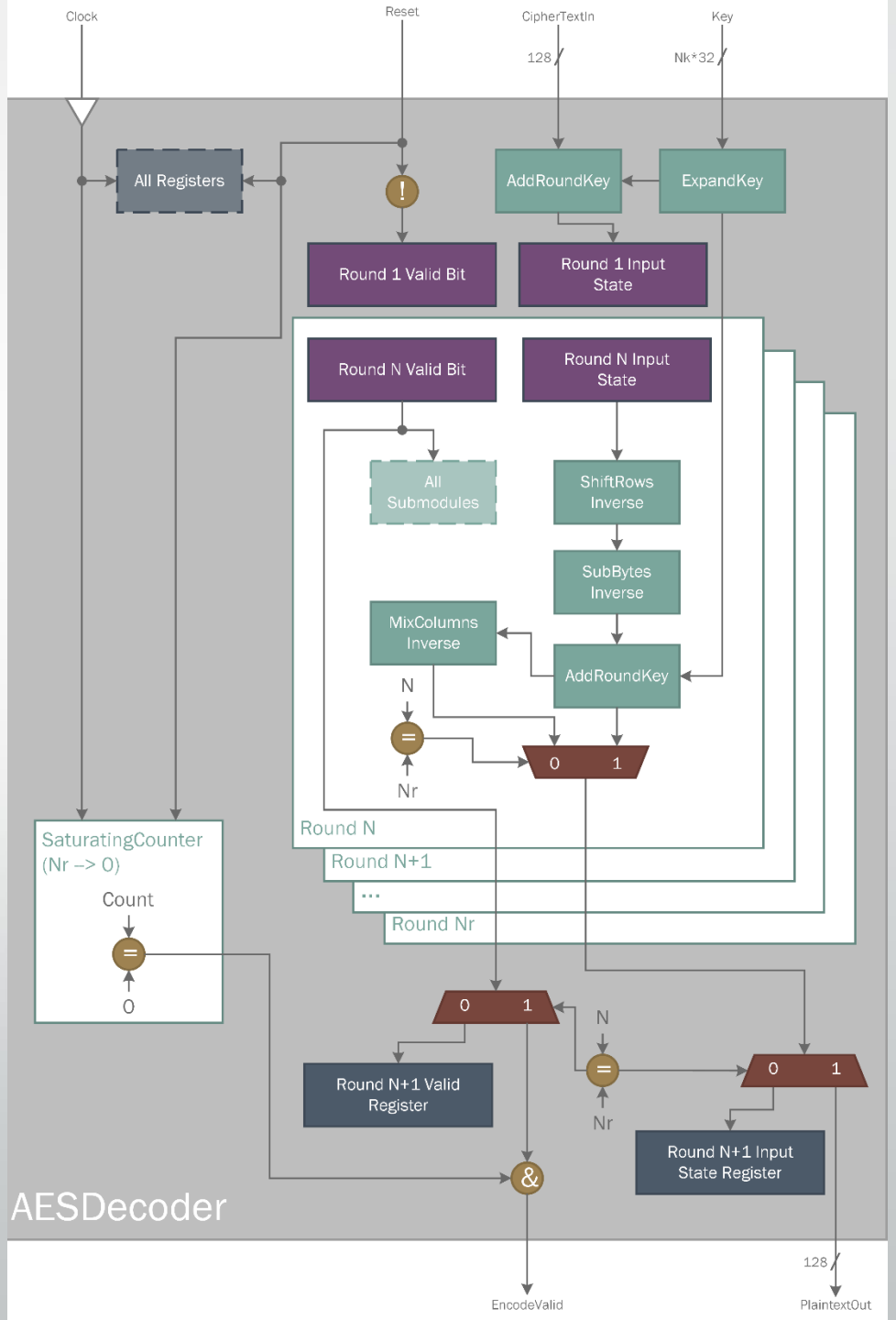
AES Overview
Key Schedule

- The initial input key is expanded into separate 128-bit *round keys*
- Copy last 4B of previous key block, perform *schedule core* on it
  - Rotate word, Sbox substitution, Rcon operation on leftmost byte, XOR with corresponding column of previous block

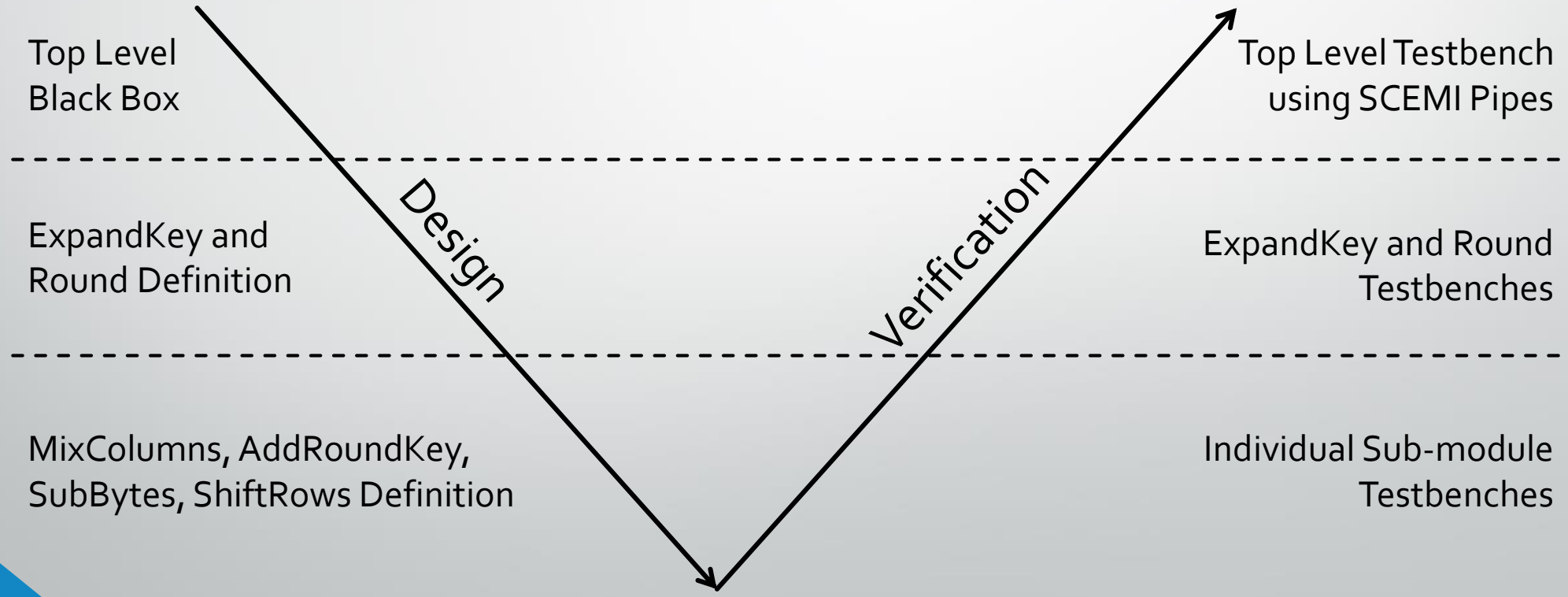


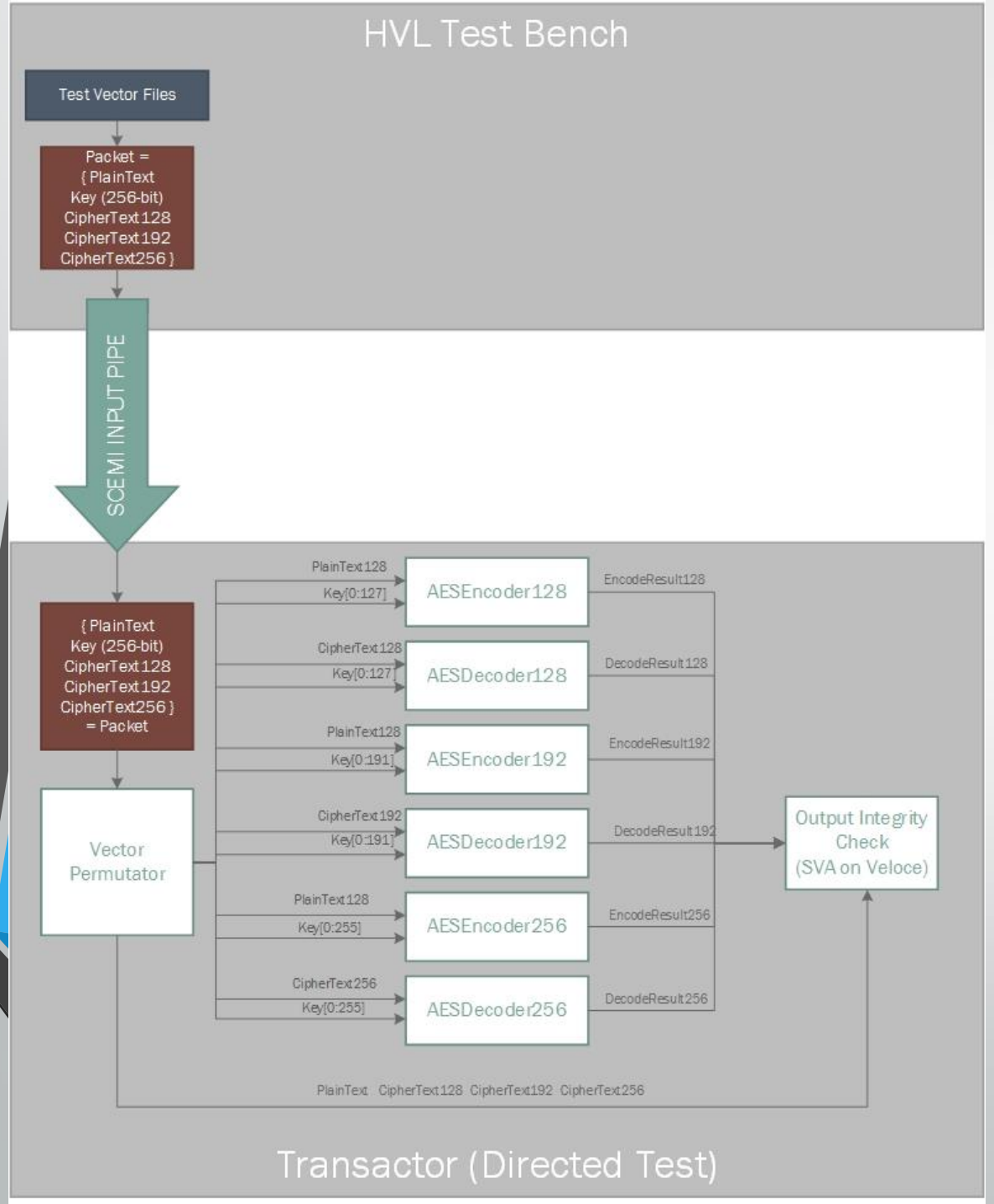
Implementation

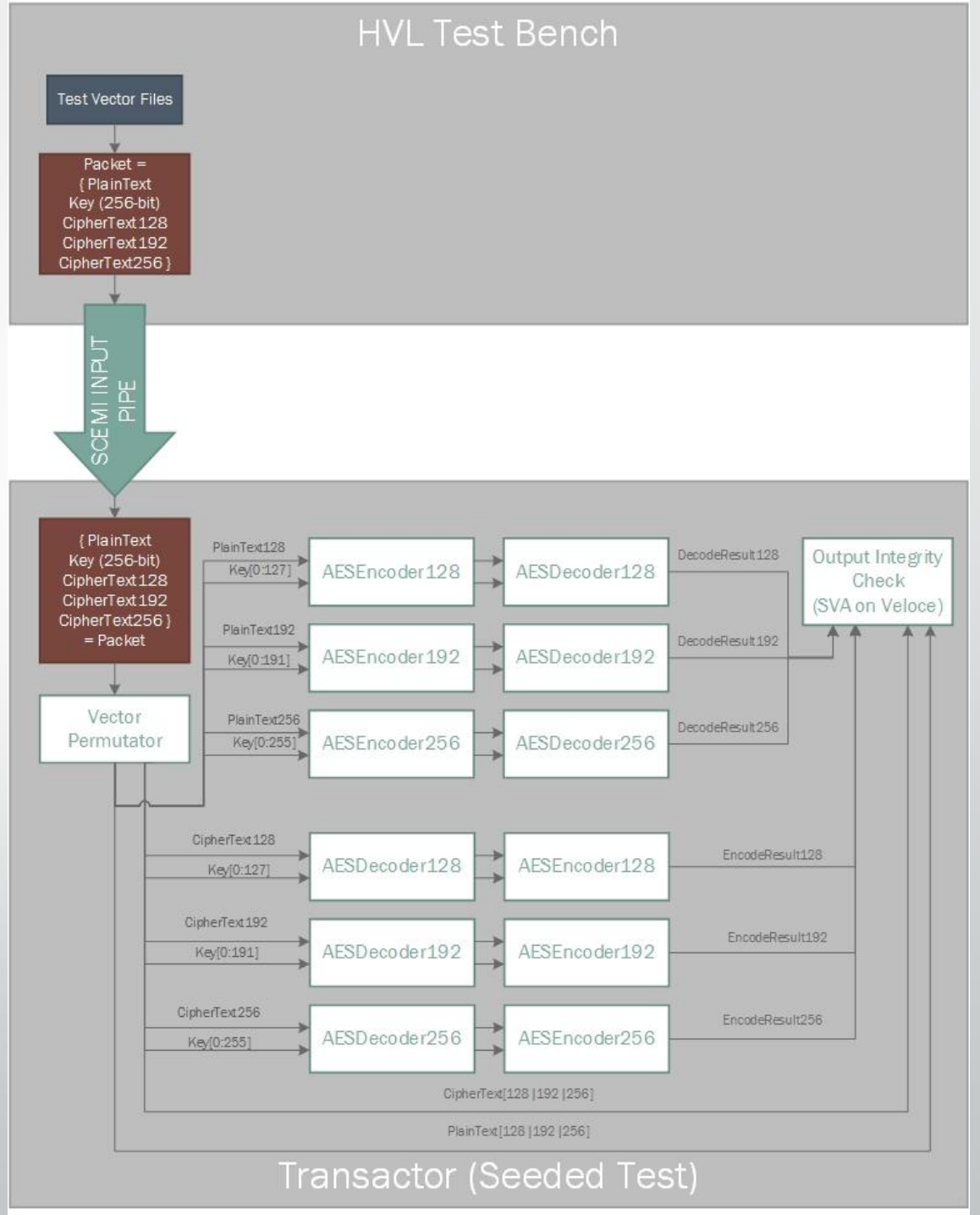

Clock
Reset
PlaintextIn
128
Key
Nk*32
All Registers
!
AddRoundKey
ExpandKey
Round 1 Valid Bit
Round 1 Input State
Round N Valid Bit
Round N Input State
All Submodules
SubBytes
MixColumns
ShiftRows
N
=
Nr
0
1
AddRoundKey
Round N
Round N+1
...
Round Nr
SaturatingCounter (Nr –> 0)
Count
=
0
0
1
N
=
Nr
Round N+1 Valid Register
0
1
Round N+1 Input State Register
&
EncodeValid
128
CipherTextOut
AESEncoder


Clock
Reset
CipherTextIn
128
Key
Nk*32
All Registers
!
AddRoundKey
ExpandKey
Round 1 Valid Bit
Round 1 Input State
Round N Valid Bit
Round N Input State
All Submodules
ShiftRows Inverse
SubBytes Inverse
MixColumns Inverse
AddRoundKey
N
=
Nr
0
1
Round N
Round N+1
...
Round Nr
SaturatingCounter (Nr –> 0)
Count
=
0
0
1
N
=
Nr
Round N+1 Valid Register
0
1
Round N+1 Input State Register
&
EncodeValid
128
PlaintextOut
AESDecoder

Implementation
Design and Verification Methodology



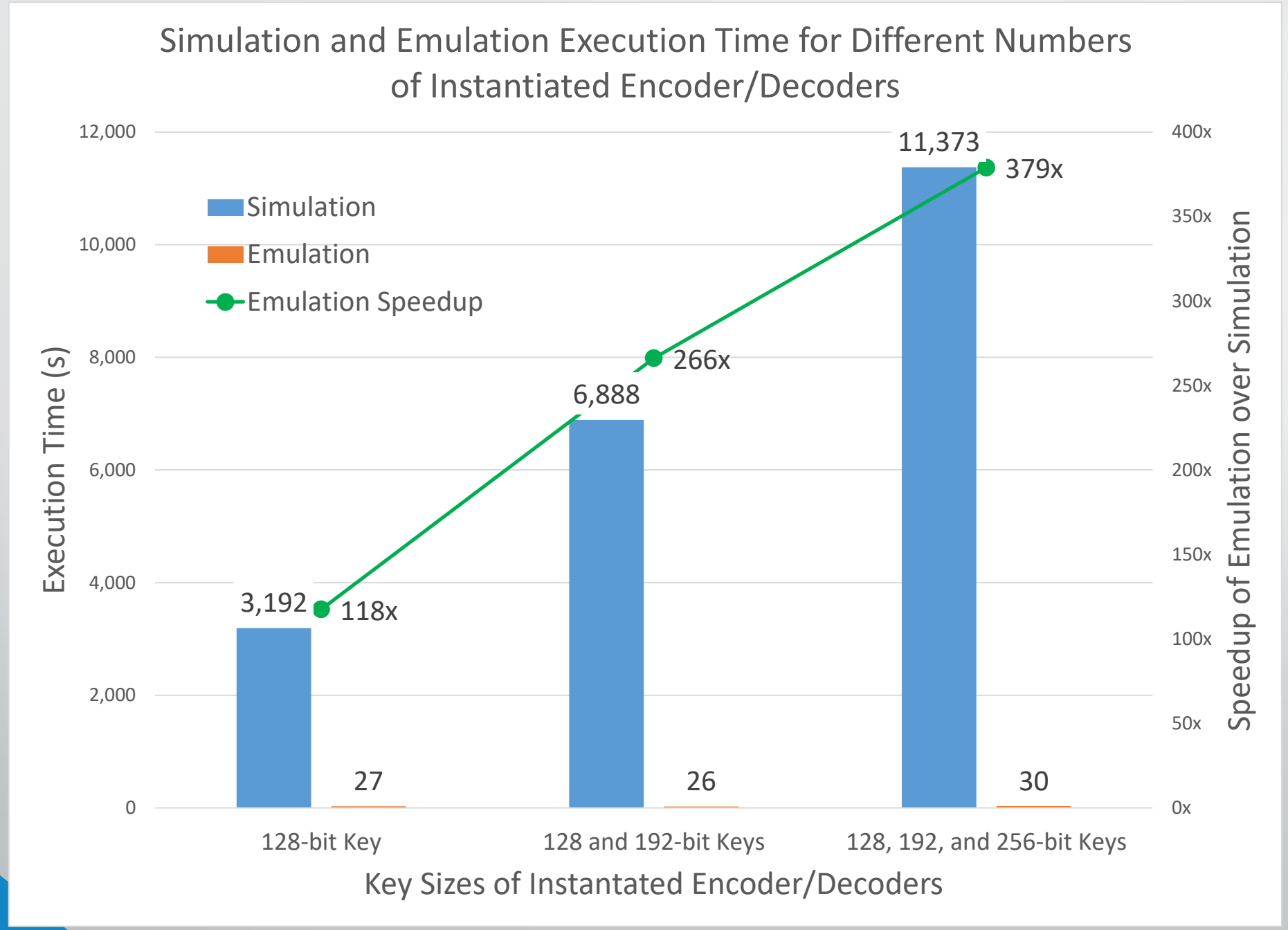

HVL Test Bench
Test Vector Files
Packet =
{ PlainText
Key (256-bit)
CipherText128
CipherText192
CipherText256 }
SCEMI INPUT PIPE
{ PlainText
Key (256-bit)
CipherText128
CipherText192
CipherText256 }
= Packet
Vector Permutator
PlainText128
Key[0:127]
AESEncoder128
EncodeResult128
CipherText128
Key[0:127]
AESDecoder128
DecodeResult128
PlainText128
Key[0:191]
AESEncoder192
EncodeResult192
CipherText192
Key[0:191]
AESDecoder192
DecodeResult192
PlainText128
Key[0:255]
AESEncoder256
EncodeResult256
CipherText256
Key[0:255]
AESDecoder256
DecodeResult256
Output Integrity Check
(SVA on Veloce)
PlainText CipherText128 CipherText192 CipherText256
Transactor (Directed Test)


HVL Test Bench
Test Vector Files
Packet =
{ PlainText
Key (256-bit)
CipherText128
CipherText192
CipherText256 }
SCEMI INPUT PIPE
{ PlainText
Key (256-bit)
CipherText128
CipherText192
CipherText256 }
= Packet
Vector Permutator
PlainText128
Key[0:127]
AESEncoder128
AESDecoder128
DecodeResult128
PlainText192
Key[0:191]
AESEncoder192
AESDecoder192
DecodeResult192
PlainText256
Key[0:255]
AESEncoder256
AESDecoder256
DecodeResult256
Output Integrity Check
(SVA on Veloce)
CipherText128
Key[0:127]
AESDecoder128
AESEncoder128
EncodeResult128
CipherText192
Key[0:191]
AESDecoder192
AESEncoder192
EncodeResult192
CipherText256
Key[0:255]
AESDecoder256
AESEncoder256
EncodeResult256
CipherText[128 |192 |256]
PlainText[128 |192 |256]
Transactor (Seeded Test)

Simulation and Emulation Results

Simulation and Emulation Execution Times with and without inferred RAMs

Simulation and Emulation Execution Time with and without inferred RAMs		
	128-bit Key w/ Inferred RAMs	128-bit Key w/out Inferred RAMs
Simulation Runtime	3,174 s	3,192 s
Emulation Runtime	64 s	27 s
Emulation Speedup over Simulation	50x	118x
Emulator HDL Time Advance (Throughput)	87.59%	75.70%
Emulator Clock Speed	207 kHz	740 kHz
Veloce Compilation Time	5 minutes	10 minutes


Simulation and Emulation Execution Time for Different Numbers of Instantiated Encoder/Decoders
Simulation
Emulation
Emulation Speedup
Execution Time (s)
Speedup of Emulation over Simulation
12,000
10,000
8,000
6,000
4,000
2,000
0
400x
350x
300x
250x
200x
150x
100x
50x
0x
3,192
27
118x
6,888
26
266x
11,373
30
379x
128-bit Key
128 and 192-bit Keys
128, 192, and 256-bit Keys
Key Sizes of Instantated Encoder/Decoders

Future Work

- More carefully tune the performance tradeoff of logic vs RAM for look up tables.
- Investigate advanced concurrent strategies for inbound and outbound streaming of data.
- Increase compiled frequency of the emulator by reducing the design critical path or finding additional compilation options.

References

[1] Advanced Encryption Standard. (2016, October 12). In *Wikipedia, The Free Encyclopedia.* Retrieved 13:29, October 12, 2016. (http://en.wikipedia.org/w/index.php?title=Advanced_Encryption_Standard&oldid=743995771)

[2] Pub, NIST FIPS. "197: Advanced encryption standard (AES)." Federal Information Processing Standards Publication 197 (2001): 441-0311. (http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf)

[3] S.-M. Yoo, D. Kotturi, D.W. Pan, J. Blizzard, An AES crypto chip using a high-speed parallel pipelined architecture, Microprocessors and Microsystems, Volume 29, Issue 7, 1 September 2005, Pages 317-326, ISSN 0141-9331, http://dx.doi.org/10.1016/j.micpro.2004.12.001. (http://www.sciencedirect.com/science/article/pii/S0141933104001632)

[4] A. Pearson, D. Collins, S. Lawson, Compilation Time Configurable Pipelined AES Encoder and Decoder

PSU ECE571, Spring 2016

Questions?

https://github.com/kinap/AES-Processor/