

Compilation Time Configurable Pipelined AES Encoder and Decoder


Alex Pearson

Daniel Collins

Scott Lawson

Outline

- AES Overview
 - Round Overview
 - Key Schedule
- SystemVerilog Implementation
 - Block Diagram
 - Design/Verification Methodology
 - Pertinent Features Used
- Simulation and Emulation Results



AES Overview

AES Overview

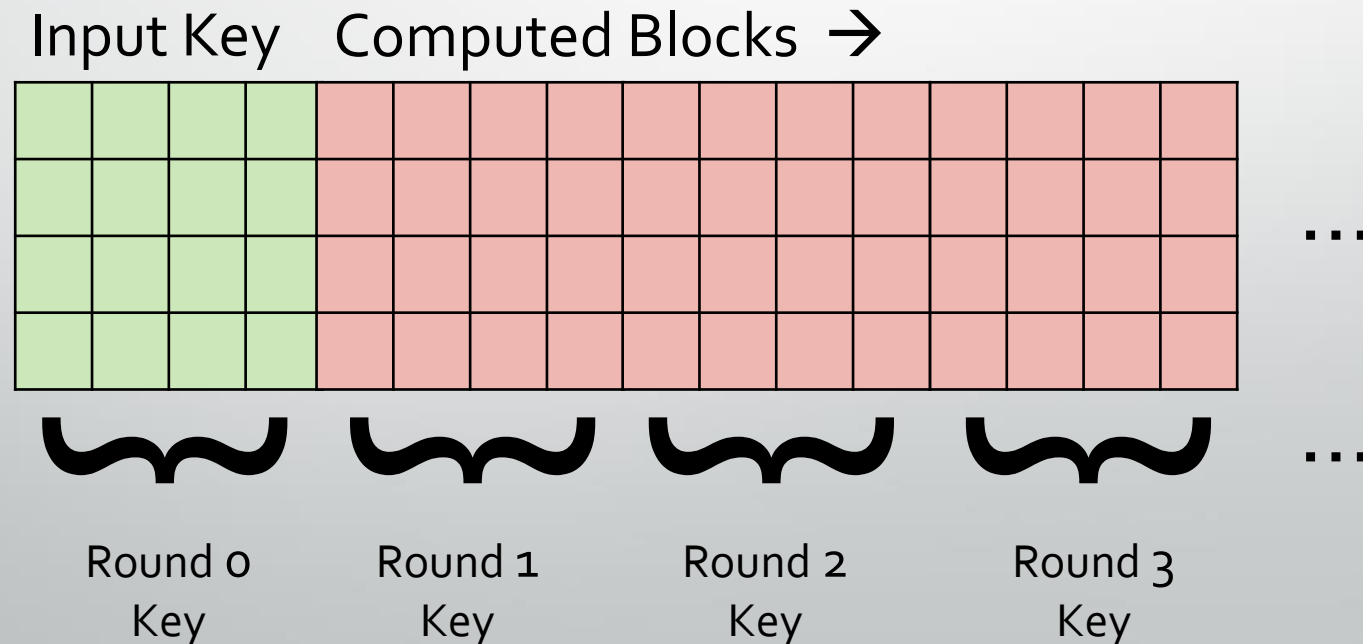
- The AES standard is based on the Rijndael Cipher but only supports a block size of 128bits and 3 key sizes of 128, 192, and 256 bits termed AES-128, AES-192, and AES-256 respectively.
- Algorithm consists of multiple rounds of input manipulation to produce output

AES Overview – Round Overview

- The encryption is performed by applying a round consisting of 4 stages (the final stage omits MixColumns) repeatedly on the state (initial state is the plain text input) and using the derived round keys.
- The number of rounds differs across the 3 AES standards with AES-128 requiring 10 rounds, AES-192 requiring 12 rounds, and AES-256 requiring 14 rounds.
- The 4 stages are: SubBytes, ShiftRows, MixColumns, AddRoundKey

AES Overview – Key Schedule

- The AES cipher requires the use of a Key Expansion module based on the Rijndael's key schedule to generate a 128-bit round key for each round (plus an additional round key added as a first step)



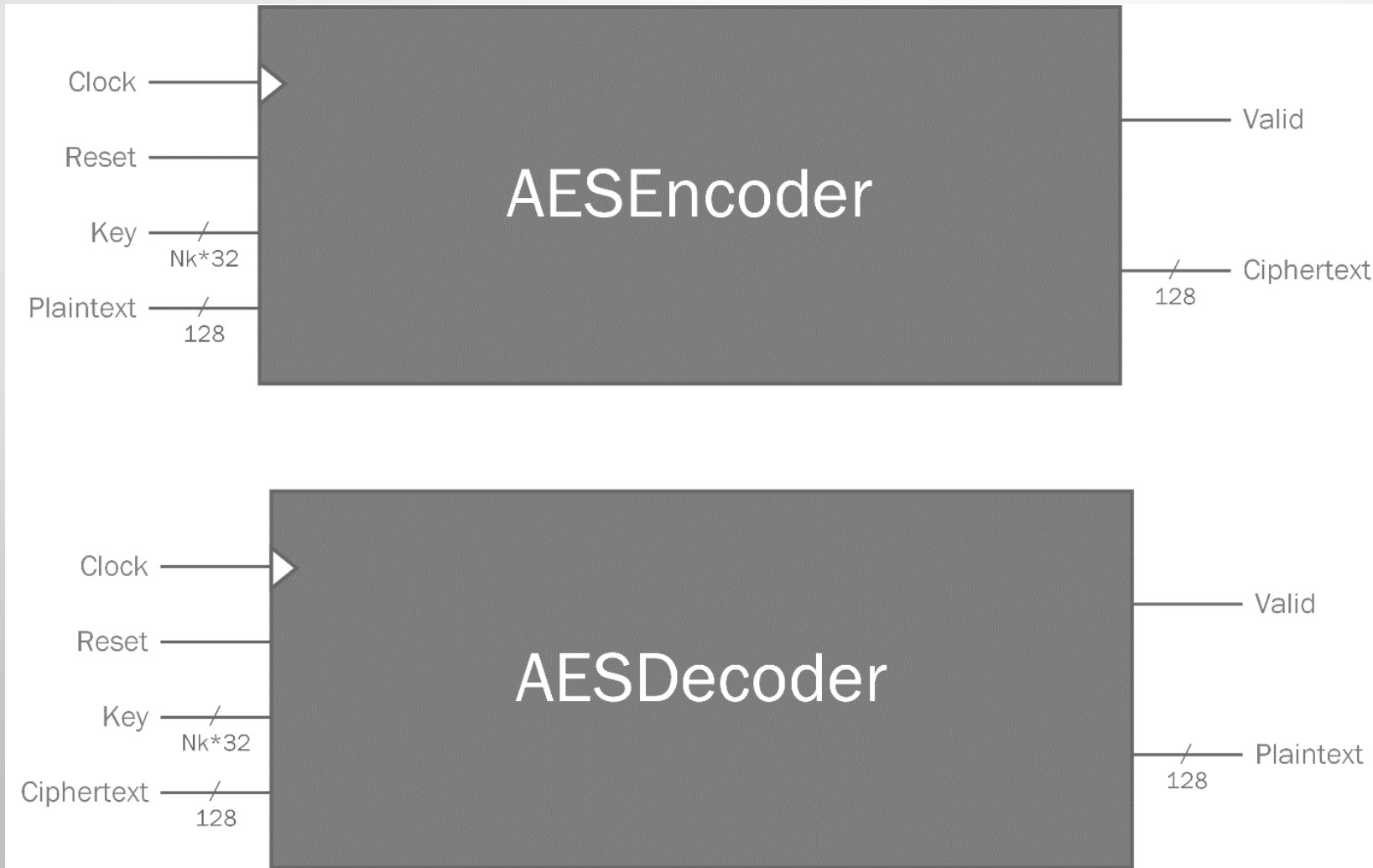


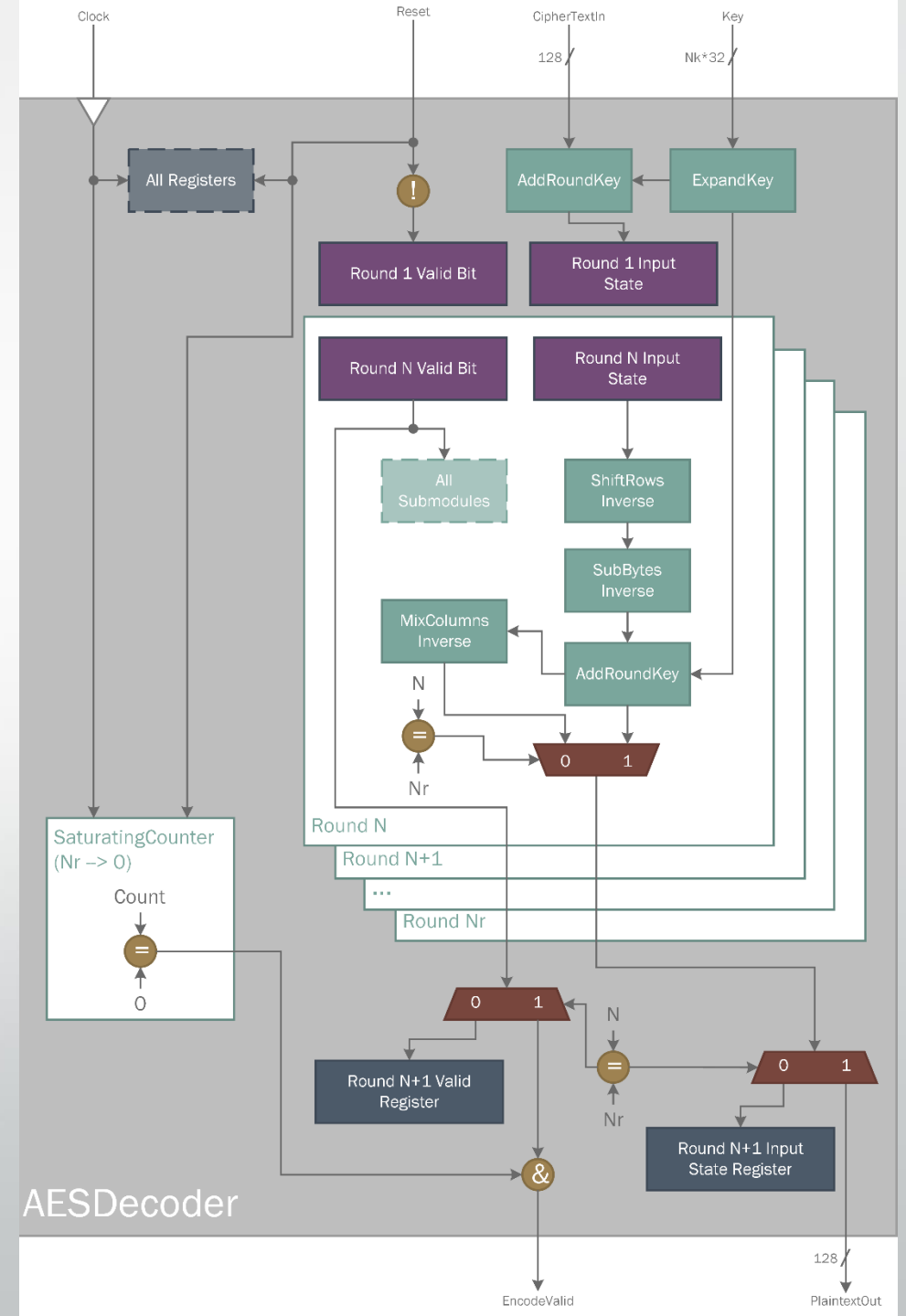
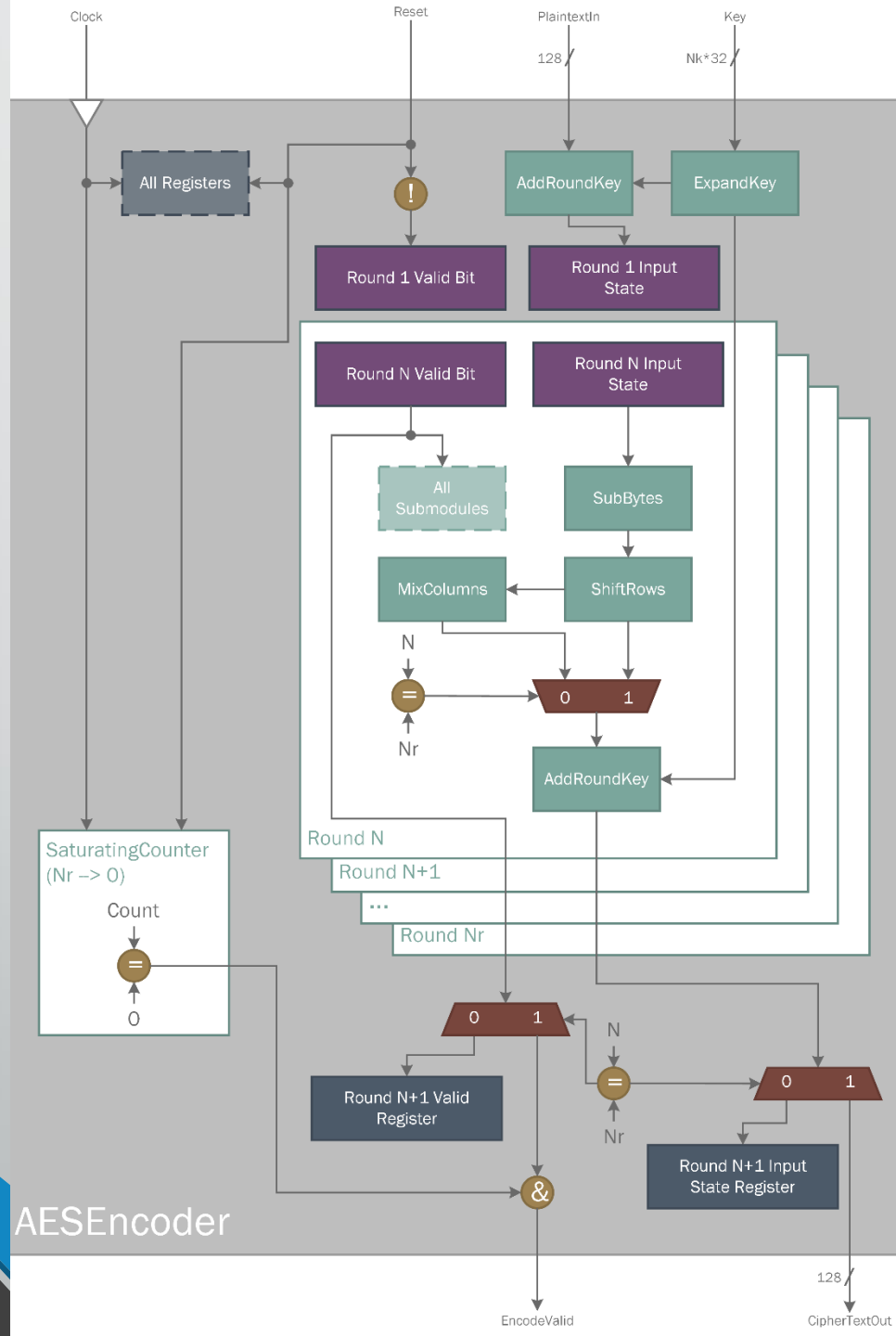
SystemVerilog Implementation

SystemVerilog Implementation

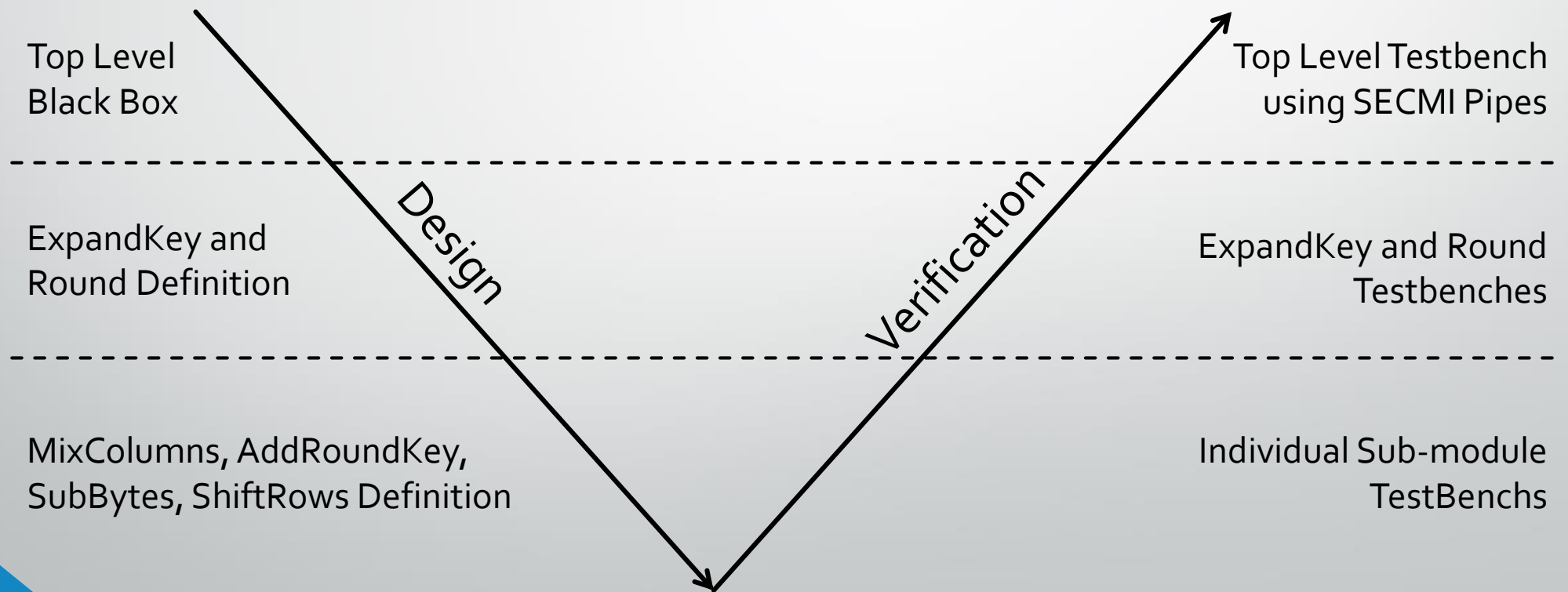
- Our project was based on the Federal Information Processing Standard Publication 197 that describes the Advanced Encryption Standard (AES)
- We didn't start with any existing hardware level description of an AES encoder or decoder
- We also utilized a paper (An AES crypto chip using a high-speed parallel pipelined architecture by Yoo, et. al) that shows one method for pipelining the AES cipher that we based our pipelined methodology on

SystemVerilog Implementation Block Diagram

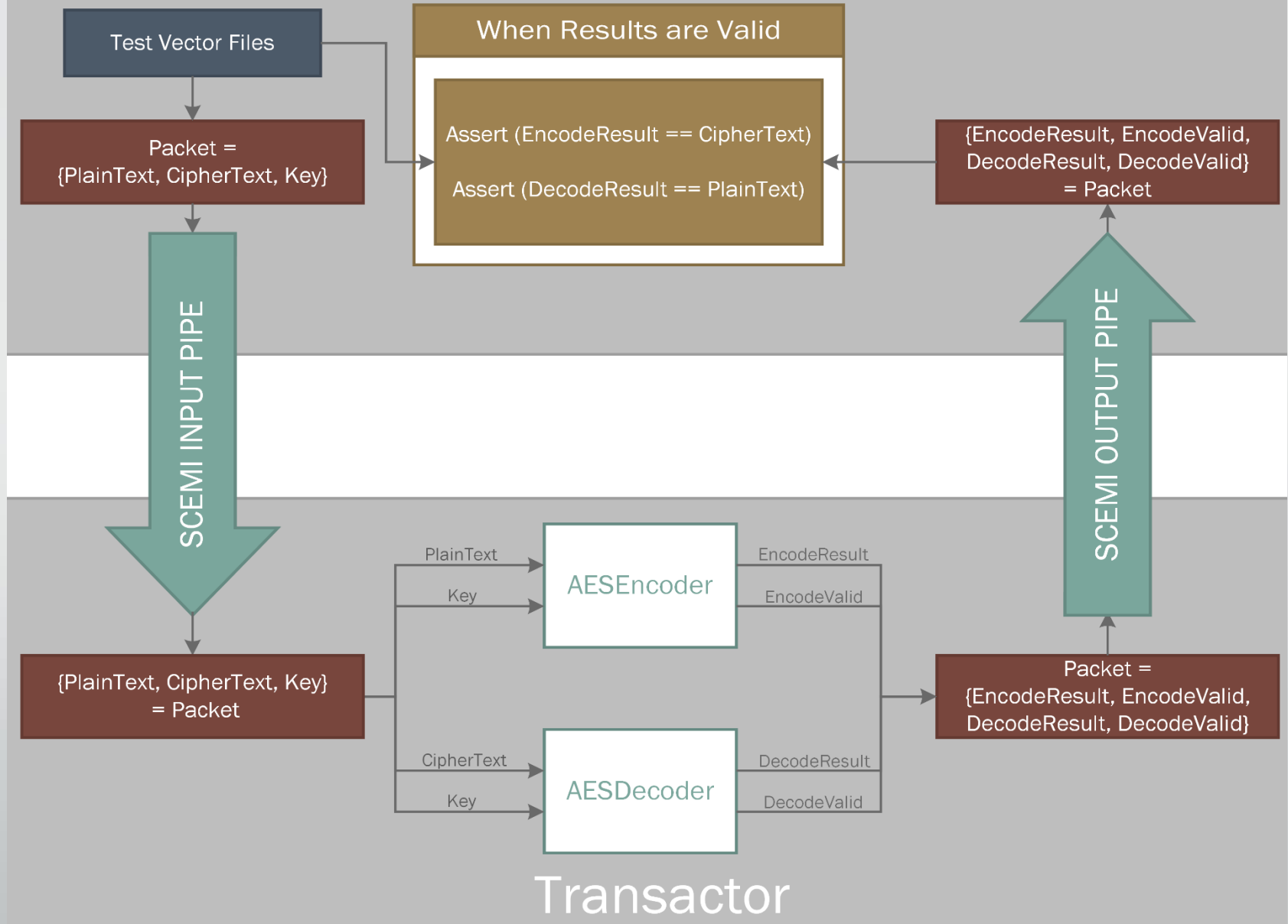




SystemVerilog Implementation Design and Verification Methodology



HVL Test Bench



SystemVerilog Implementation

Pertinent Features Used

- Used [classes](#) to implement our test methodology and allow code reuse between testbenches with the same or similar requirements
- Used [queues](#) to create variable length storage that we could use as FIFO input from our testbenches
- Used [always_comb](#) and [always_ff](#) blocks for our logic
- Used [typedefs](#) for all of our signals, which greatly aided in implementing compile-time configurability of our project
- Used [packed structs](#) to organize groups of signals
- Used [packages](#) to organize our code and share definition and functionality among modules
- Used [parameterized types](#) to create generic modules that could be instantiated to operate on different types of data
- Used [byte streaming operators](#) to send and receive data through the SCEMI pipes, transforming it from an unpacked to packed array



Simulation and Emulation Results

Simulation and Emulation Results

- For top level testing we had a single testbench split across a HVL file written in SystemVerilog (but containing non-synthesizable constructs) and a SystemVerilog “Transactor” only containing synthesizable constructs that can run on the emulator in TBX mode using SCEMI pipes
- Our testbench is able to run in “puresim” mode solely on the simulator or in “veloce” mode, splitting the tesbench with the HVL code running in Questa and the Transactor code running on Veloce

Simulation and Emulation Results

- 10,000,000 randomly generated vectors for each key size
- All key sizes performed similarly
 - Simulation
 - Compile / load time: < 1 minute
 - Execution time: ~42 minutes
 - Emulation
 - Compile / load time: ~4 minutes
 - Execution time: ~38.45 minutes

```
# Started at:
# Mon May 30 20:13:02 PDT 2016
# ** Note: $finish      : test/hvl/EncoderDecoderTestBench.sv(84)
#   Time: 200000650 ns  Iteration: 1  Region: /EncoderDecoderTestBench/run
#
# Ended at:
# Mon May 30 20:51:27 PDT 2016
# 20000000 tests passed successfully
#
# =====
#                               SIMULATION STATISTICS
# =====
# Simulation finished at time 200000650
#
# Total number of TBX clocks:                      526142972
# Total number of TBX clocks spent in HDL time advancement: 20000065
# Total number of TBX clocks spent in HDL due to callee execution: 0
# Percentage TBX clocks spent in HDL time advance: 3.80 %
# -----
# Total CPU time (user mode):                      625.74 seconds.
# Total time spent:                                2307.31 seconds.
# -----
# Info!      [TCLC-5501]: : Disconnected from emulator.
# Info!      [TCLC-5501]: : project database unlocked.
# Info!      [TCLC-5663]: : Shutting down the user runtime session.
```


Future Work

- Optimize AES design
 - Investigate methods for combining multiple stages in a round
 - Add intermediate pipelining for key expansion module
 - Reduce the amount of expanded key data being buffered between each round
- Encrypted/Decrypted output check on emulator via assertions



Questions?