Verification & Result Generation
Overview
Phase I
Datapath
Phase I: Verification

Main Process

Input Data

CipherCore Datapath

Output Data

Control

Status

CipherCore_Datapath_TB
Phase I: Verification

Consists of one primary process handling input, control and verification against expected output

Testbench should focus on the verification of basic functionality of the circuit. At a minimum, it should verify a complete operation for encryption and decryption with two blocks of data.

Control signals should emulate the behavior of the future controller
Phase I: Result Generation

Generation of results typically cannot be done directly due to limit on the number of pins in the targeted chips.

As a result, one must wrap a unit using components such as SIPO (Serial-In-Parallel-Out) and PISO (Parallel-In-Serial-Out) units.

An example of the wrapper design can be found in src_rtl/wrappers/CipherCore_Wrapper.vhd
Phase I: Result Generation

CipherCore_Datapath_Wrapper

Data In
Y-bit
Control C-bit

CipherCore Datapath

Data Output
Z-bit
Status D-bit

SIPO

1 to (Y+C)

PISO

(D+Z) to 1

Output

1-bit

Input

1-bit
SIPO: Serial In Parallel Out
PISO: Parallel In Serial Out
Phase II
CipherCore
Phase II: Verification
Phase II: Design Process

1) Understand how the provided Pre- and Post-processors operate. In particular, a user must understand the following:
   A. Operation of the processors based on AEAD_Core generics
   B. Handshaking operation between CipherCore and the processors
2) Modify the generics settings for AEAD_Core.vhd
3) Develop a CipherCore Controller based on Phase I testbench and your understanding of the above step
4) Place CipherCore_Datapath and CipherCore_Controller in the provided CipherCore.vhd
5) Change default generics in CipherCore.vhd
6) Generate test vectors from AETVgen.py to use with AEAD_tb.vhd to debug your CipherCore
7) Modify generics from AEAD_TB if the default values do not match your algorithm.
Phase II: Result Generation (CipherCore) [optional]

Use the provided CipherCore_Wrapper.vhd and set generics to the values used in your AEAD_Core.
Phase II: Result Generation (AEAD_Core)

Use the provided AEAD_Core_FullWrapper.vhd and set generics to the one used in your AEAD_Core
Help session with Ice devoted to the automated generation of test vectors and the operation of a universal testbench

Saturday
1:00-4:00pm

ECE Labs ENGR 3208