Advance Encryption Standard (AES) core implemented in VHDL

INTRODUCTION

This project consists of a synchronous AES encryption core that operates on a 128-bit keys and a 4 x 4 column-major order matrix of bytes termed the *state*. The core has been tested on a Pynq-Z1 development board from Digilent at 250 MHz from the Zynq PS PLL.

SPECIFICATION

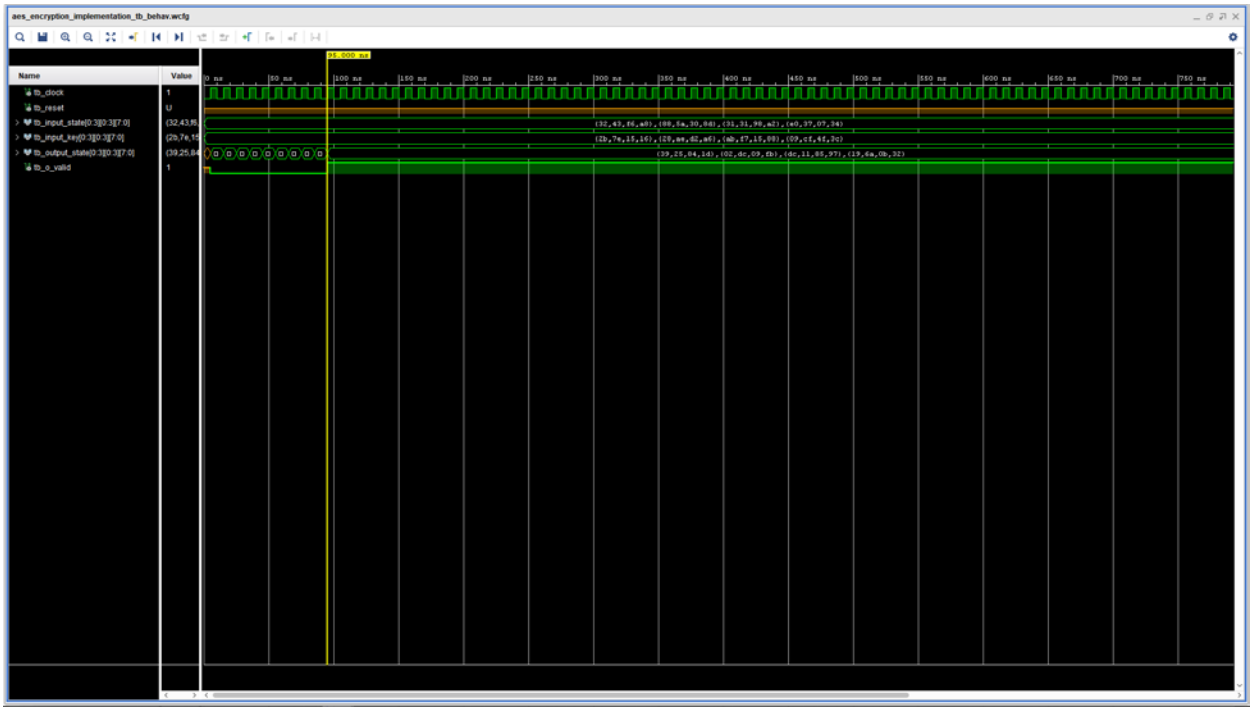
- Input:
  - 4 x 4 column-major order matrix *i_state*
  - 4 x 4 column-major order key *i_key*
  - Clock *clock*
  - Reset *reset*
- Output:
  - 4 x 4 encrypted column-major matrix *o_state*
  - Valid encrypted output state *o_valid*
- Toolset: Vivado 2018.1

MODULE HIERARCHY

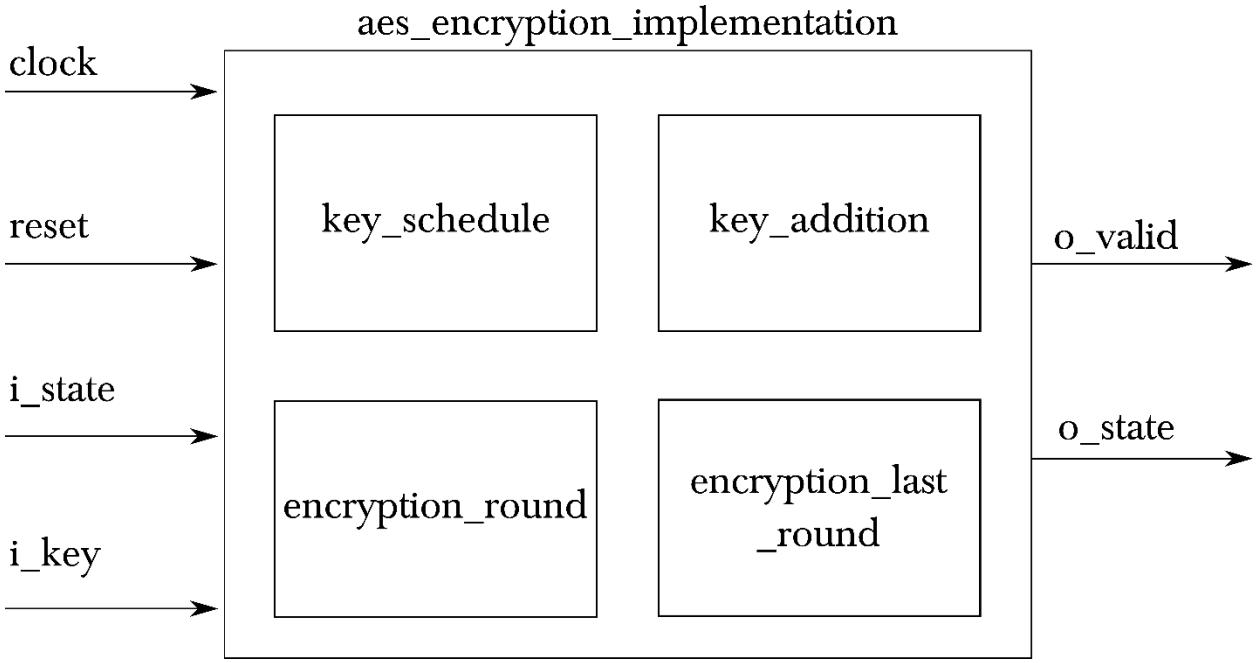
- **Aes_encryption_implementation** – top level with state machine
  - **Aes_encryption_key_schedule** – generates the key schedule
    - **g_function** – required for generating key schedule
      - **s_box** – s-box substitution
  - **Aes_encryption_key_addition** – first round addRoundkey implementation
  - **Aes_encryption_round**
    - **s_box** – s-box substitution
    - **Aes_encryption_ShiftRows** – implements ShiftRows
    - **Aes_encryption_MixColumns** – implements mixColumns
    - **Aes_encryption_key_addition** – implements addRoundKey
  - **Aes_encryption_last_round**
    - **s_box** – s-box substitution
    - **Aes_encryption_ShiftRows** – implements ShiftRows
    - **Aes_encryption_key_addition** – implements addRoundKey
- **Aes_encryption_wrapper** – used to test on the Pynq-Z1 development board. A state and key can is configurable by the user in the VHDL code.

TEST BENCH

- The project contains various testbenches for different modules that makes up the AES encryption core.
- This includes the top-level testbench for the overall implementation. Below illustrates the waveform of the top-level waveform.
- Running a testbench can be done by going into simulation sources in project manager, selecting the testbench as top then “run simulation” from the flow navigator.



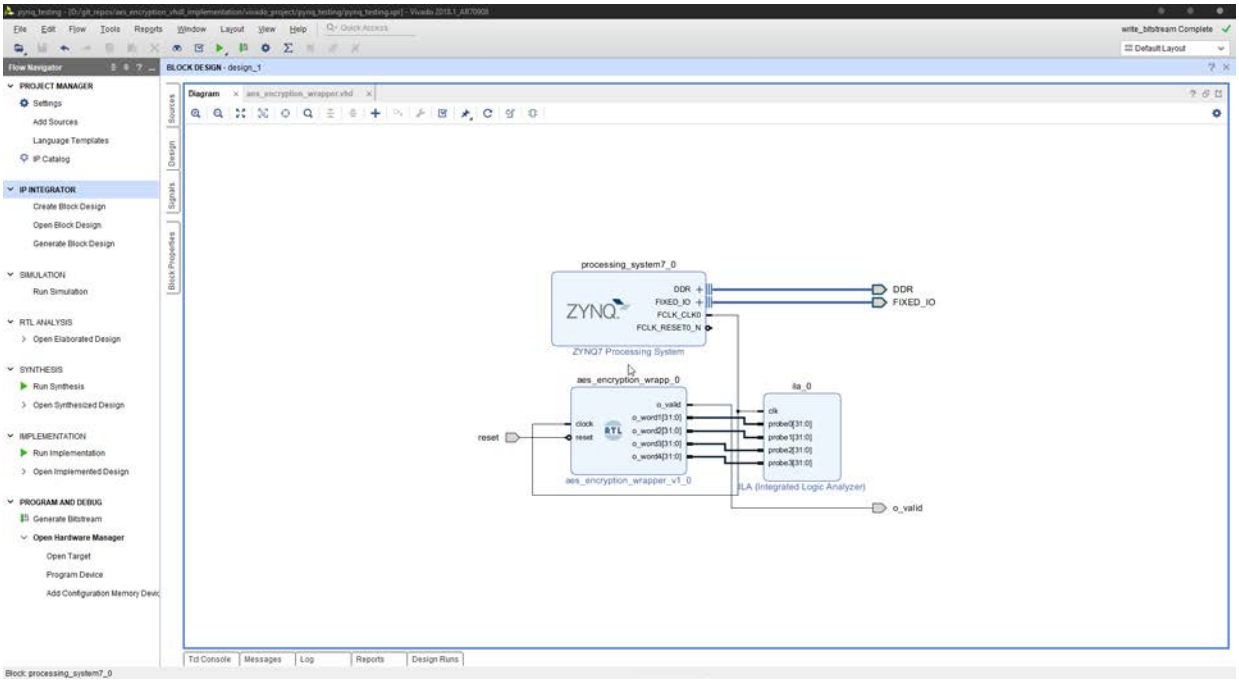
HIGH LEVEL ARCHITECTURE



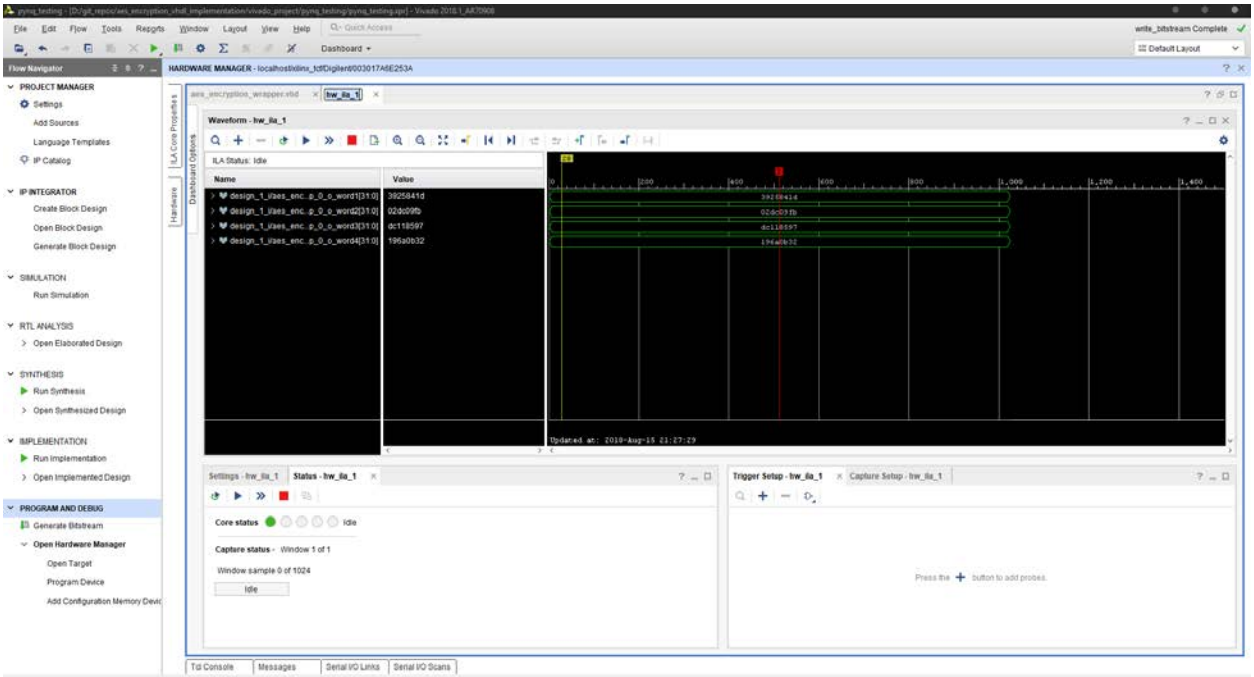
- The design of the AES encryption core is pipelined to minimize logic-depth and hence to increase the clock frequency capable by the core
- The initial design of the core calculates the key schedule before going through the encryption rounds. Although the design is simple, it is not elegant and has large logic depth.
    - To combat this, a round's key transform is calculated in the previous round while the intermediate state is still being calculated. Once the state is calculated for a round, the key is already calculated and is used to calculate the next intermediate state.

TESTING

- Testing of the core is carried on a Pynq-Z1 development board. The Vivado project *vivado_project/pynq_testing* consists of a block design that utilizes the AES encryption core.
- The block design utilizes a Zynq PS, and an ILA, as shown below:



- The Pynq provides a 125 MHz on pin H16. However, testing on higher frequencies utilizes the Zynq PS PLL which can generate up to 250 MHz.
- Additionally, the ILA allows user verification of the ciphered output in hardware using the hardware manager in Vivado. Shown below is the screen shot of the output ciphered state.



REFERENCES

Pub, N. F. (2001). 197: Advanced encryption standard (AES). *Federal information processing standards publication*, *197*(441), 0311.

Paar, C., & Pelzl, J. (2009). *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media.