# Tea Algorithm for FPGA Hardware Acceleration and Reconfigurable Encryption

Hmayak Apetyan, Hyung Jin Kim, Jacob London, Yongyuan Zhang, Ziyan Lin, Dr. Aly
California State Polytechnic University, Pomona Department of Electrical and Computer Engineering

## Introduction

Implementing the Tea, tiny encryption algorithm, on an FPGA, field-programmable gate array, allows for a simple reconfigurable encryption system implemented into a hardware design. This algorithm is simple enough to create a base design which can be easily modified to allow more complex designs to be implemented. Through the use of hardware design in VHDL, a wide variety of FPGAs can be targeted and designed for, while utilizing the simple base design implemented by this system.

## Method

The resulting hardware implementation of Tea utilized a state machine which processed different parts of the calculation of each round in order. Because of this state machine, the process for performing rounds of encryption or decryption became simpler than it would have otherwise been in a combination design for example.

## Evaluation

The resulting hardware implementation of Tea utilized a state machine which processed different parts of the calculation of each round in order. Because of this state machine, the process for performing rounds of encryption or decryption became simpler than it would have otherwise been in a combination design for example. However, due to the sequential nature of the design, the implementation is not as fast as it could be. The design was built for the Nexys 4 FPGA, with a clock speed of 100 MHz, and utilized a state machine with twenty-five states total, with twenty-three of those states dedicated to performing each round out of sixty-four rounds. In addition to this, along with the nature of state machines, the algorithm spends two clock cycles in each state.

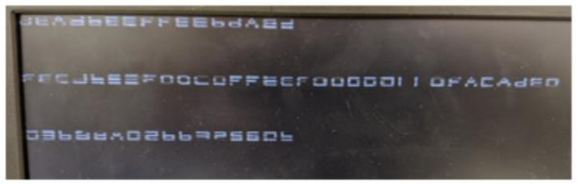

Fig. 1. Data (top), key (middle), encrypted data (bottom) with the number of rounds equal to 0x0C.

## Conclusion

For the encipher and decipher modules, we had a difficult time implementing the tea algorithm at the start. We assume that VHDL can do the complicated math directly like we wanted and first coded in a similar way we would for C, but in VHDL. However, after running testbenches we found out that that was not the case and tried many different implementations, rechecking our code, and rerunning testbenches until we decided to implement a finite state machine for each math operation in the algorithm. After doing that we were able to get the results that the C code was giving us in VHDL.

## Acknowledgements

1. Dr. Mohamed El-Hadedy for the assistance in both consultancy and reference for the development of this project.