

Fully pipelined implementations of AES with a speed exceeding 20 Gbits/s with S-boxes implemented using logic only

Project Specification

Chethan Ananth and Karthick Ramu

Introduction and Motivation

Objective:

The main objective of the project is to develop AES encryption/decryption systems that can encrypt/decrypt at the rate of 20gbs/s. It is decided to implement the counter mode AES with the key size of 128bit to achieve the proposed rate of 20gbs/s

Introduction:

Advanced Encryption Standard (AES) is the block cipher adopted symmetric key encryption algorithm used widely to encrypt the data transmitted over Internet. The basic idea behind this project is to develop and implement an AES encryption/decryption system that can operate at the rate of 20gbs/s. This high-speed data rate can be achieved by replacing the traditional method (table look-up) of S-box implementation in hardware with logic functions in counter mode AES algorithm. AES algorithm supports three key sizes: 128, 192 and 256 bits key to encrypt the data block that is 128 bits long.

Motivation:

There is always a necessity of transferring data over the Internet and security of the data from attacks is considered one of the major issues. To overcome these attacks, data transmitted over the network are encrypted and then decrypted by authorized personal. It is also necessary to consider that performing encryption/decryption on the data does not reduce the data transfer speed because of the additional process. Thus AES serves as the best-suited algorithm for strong encryption and moderate speed. The latest counter mode AES algorithm built by GMU provides the data rate of 10gbs/s. However, as the technology advances the usage of 20gbps network line is not far away and it will be necessary to build an algorithm that matches this data rate. Thus we feel that it will be practical to develop an AES algorithm that can provide the data rate of 20gbs/s or more. Also, implementing AES in FPGAs with traditional table lookup method for S-boxes utilizes a lot of BRAMs in the FPGAs. By, replacing table lookup with logic functions we also intend to reduce the utilization of BRAMs in FPGAs.

Design Entry Method: VHDL
Target Implementation: Xilinx Virtex-5
CAD Tools Used: Aldec Active HDL, Xilinx ISE

Functional Description:

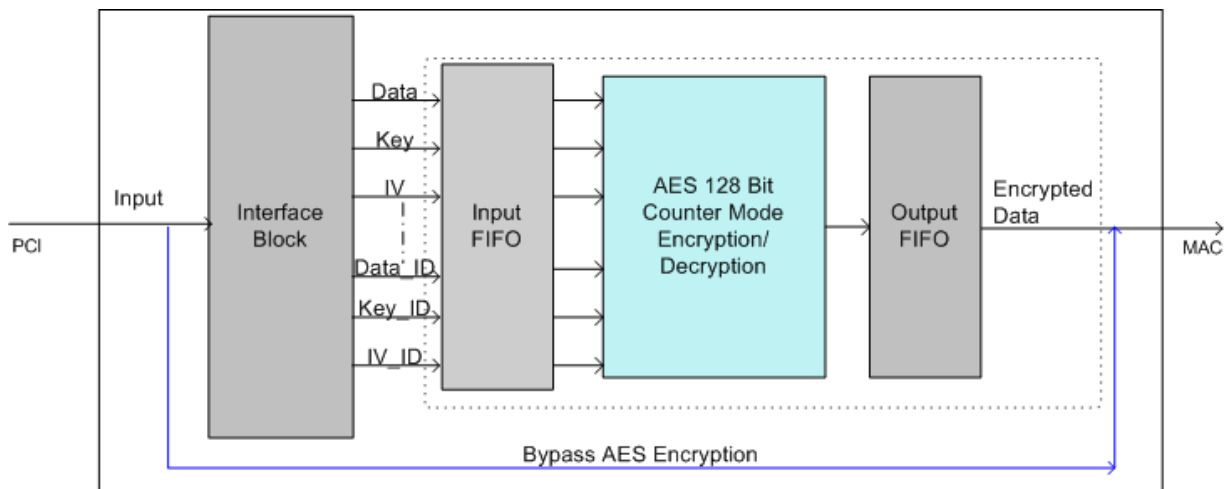
The following tasks are proposed to be completed as part of the project.

- 1) Complete implementation of the AES algorithm that includes the Encryption/Decryption Unit (ByteSub, ShiftRow, MixColumn, Key Schedule).
- 2) Develop the S-box (used for ByteSub) using Logic only and using a combination of Logic and Look-Up Tables.
- 3) Develop a non-pipelined implementation initially and extend it to a pipelined version.
- 4) Analyze the different architectures suitable to implement the S-Box using Logic.

S-Box Implementation using Logic:

The amount of memory required to implement SubBytes and InvSubBytes can be reduced to zero by utilizing the internal logic structure of inversion in $GF(2^8)$. In FPGAs, memory blocks are always present independently whether they are used or not, but their replacement by logic may be still justified. For example, memory might be already used to implement some other functions, such as input/output buffers. Additionally, in case of deeply pipelined architectures, memory-based implementation can impose an artificial restriction on the minimum clock period, while the logic-based implementation can be further pipelined. The basic idea of the logic-only implementation is to notice that inversion in $GF(2^8)$ can be decomposed into a sequence of operations in $GF(2^4)$ (including addition, multiplication, and inversion). Similarly, operations in $GF(2^4)$ can be expressed in terms of operations in $GF(2^2)$, and operations in $GF(2^2)$ in terms of operations in $GF(2)$. The operations in $GF(2)$ can be implemented using simple XOR gate (addition) and AND gate (multiplication). An inverse of 1 in $GF(2)$ is 1, and the inverse of 0 does not exist. Thus, the entire inversion in $GF(28)$ can be decomposed into a logic circuit composed of XOR and AND gates only.

Inputs and Outputs of the circuit.



Procedures for testing the functionality and performance of the circuit:

- a. Simulator in use: Active HDL, ModelSim
- b. Source of Test Vectors: CrypTool, Magma.
- c. Format of input stimuli:
VHDL Advanced Test Bench, Test Vectors using input and output text files.
- d. Performance parameters to be determined by simulation;
Throughput, Latency, Throughput to Area Ratio, Time to encrypt one block.
- e. Parameters to be determined using the implementation tools.
Number of CLB Slices (Area), Minimum size Virtex-5 FPGA Device required.

Time Schedule:

First two weeks of March – References read and analyzed.

Last two weeks of March – Begin VHDL coding.

First two weeks of April – Complete Coding, begin testing.

Last two weeks of April – Complete testing, analyze the results obtained, and prepare a draft report of the project.

First week of May – Prepare final version of the project report and presentation.

Possible Areas of specification Change:

At this preliminary stage we are not totally aware of the changes that could happen to the project on the basis of progress. We anticipate that we will be able to complete the project goal. However, if we are not able to achieve good progress

- We may plan on using the existing GMU AES core and replace the subbyte implementation with logic functions.
- Since a number of different architectures have been proposed to implement the S-box, we may have to restrict this project to only one type of architecture depending on the feasibility of the different architecture implementations in the given time frame.
- Depending on the progress of the project, the Implementation of S-box using a combination of both S-box and Look-Up tables may be done-away with.

Tentative table of contents.

- 1) Introduction
- 2) Brief Description of AES
- 3) S-Box using Logic v/s S-Box using Look-up Tables
- 4) Implementation S-Box using Logic
- 5) Results Obtained
- 6) Comparisons of performance parameters(Logic v/s Look-Up Table)
- 7) Conclusion
- 8) References

List of Literature:

- 1) **FPGA and ASIC Implementations of AES, by Kris Gaj and Pawel Chodowiec**
http://ece.gmu.edu/courses/ECE746/project/S08_Project_resources/AES.pdf
- 2) **FPGA and ASIC Implementations of AES, Section 6, Implementation of basic operations of AES in hardware.**
http://ece.gmu.edu/courses/ECE746/project/S08_Project_resources/AES.pdf
- 3) **D. Canright, "A very compact Rijndael S-box," Technical Report NPS-MA-05- 001, 2005.**
http://ece.gmu.edu/courses/ECE746/project/S08_Project_resources/Canright_NPS_report.pdf
- 4) **D. Canright, "A very compact S-box for AES," CHES 2005**
http://ece.gmu.edu/courses/ECE746/project/S08_Project_resources/Canright_CHES.pdf
- 5) **Using Advanced Encryption Standard (AES) Counter Mode**
<http://www.ietf.org/rfc/rfc3686.txt>