Low-Latency, Small-Area FPGA Implementation of the Advanced Encryption Standard Algorithm

Hoang Trang, University of Technology, Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam

Nguyen Van Loi, IC Design Research & Education Center, Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam

ABSTRACT

This paper presents a Field-Programmable Gate Array (FPGA) implementation of an Advanced Encryption Standard (AES) algorithm using approach of combination iterative looping and Look-Up Table (LUT)-based S-box with block and key size of 128 bits. Modifications in the way of loading data out in AES encryption/decryption, loading key_expansion in Key_Expansion blocks are also proposed. The design is tested with the sample vectors provided by Federal Information Processing Standard (FIPS) 197. The design is implemented on APEX20KC Altera’s FPGA and on Virtex XCV600 Xilinx’s FPGA. For all the authors’ proposals, they are found to be very simple in FPGA-based architecture implementation, better in low latency, and small area, but large in memory, moderate throughput.

Keywords: Advanced Encryption Standard, Data Loading Modification, Field-Programmable Gate Array (FPGA), Iterative Loop, Look-Up-Table, Low Latency, Small Area

INTRODUCTION

For a long time, the Data Encryption Standard (DES) with a key length of 56 bits was considered as a standard for the symmetric key encryption. However, this key length was considered small and could easily be broken. For this reason, the National Institute of Standards and Technology (NIST) opened a formal call for algorithms in September 1997. After careful selection and evaluation processes, on October 2, 2000, NIST announced that the Rijndael algorithm of two inventors Joan Daemen and Vicent Rijmen was the winner. Rijndael algorithm can be specified with key and block sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits. Therefore, the problem of breaking the key becomes more difficult (Daemen & Rijmen, 2002; Tessier & Burleson, 2001; Bertoni, Guajardo & Paar, 2004; Boklan, 2009; Al-Hamdani, 2009). In cryptography, the Advanced Encryption Standard (AES) is the name of the standard, but in practice, Rijndael algorithm is also referred
to as AES (FIPS PUB 197, 2001). AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits.

The AES algorithm can be efficiently implemented by hardware and software. Software implementations cost the smallest resources, but they offer a limited physical security and the slowest process. Besides, growing requirements for high speed, high volume secure communications combined with physical security, hardware implementation of cryptography takes place. In contrast to software implementation, the pure hardware implementation could provide higher security and the higher data rate (Gbits/second) by pipelining and parallelization (Verbauwhede, Schaumont & Kuo, 2003), and also provides a cost-effective solution for many application specific systems.

An FPGA implementation is chosen in this study since it is an intermediate solution between general purpose processors (GPPs) and application specific integrated circuits (ASICs). It has advantages over both GPPs and ASICs in which FPGA provides a faster hardware solution than a GPP; wider applicability than ASICs since FPGA is more flexible, and small development cost (Liu, Lu, Kuehn & Jantsch, 2012; Shih & Hsiung, 2009; Imran et al., 2012).

This paper deals with an FPGA implementation of an AES encryptor/decryptor using an iterative looping approach with block and key size of 128 bits. Besides, our design uses the lookup- table (LUT) implementation of S-box in AES algorithm. The proposed approach of combination iterative looping and LUT-based S-box is chosen since it gives very low complexity architecture and is easily operated to achieve low latency as well as moderate throughput (Hoang & Nguyen, 2012). In the our previous work (Hoang & Nguyen, 2012), it is shown that the combination of iterative looping and LUT-based approach provides low latency of 13 clock cycles in encryption, 23 clock cycles in decryption which is lower than other works (Panato, Barcelos & Reis, 2002; Atul, Borkar, Kshirsagar & Vyawahare, 2011; Elbirt, Yip, Chetwynd & Paar, 2000; McLoone & McCanny, 2001; Standaert, Rouvroy, Quisquater & Legat, 2003; Saggese, Mazzeo, Mazocca & Strollo, 2003). The iterative looping and LUT-based S-box approach also gives smaller area, moderate throughput but large in memory (Hoang & Nguyen, 2012). This paper is an extended version of our previous work (Hoang & Nguyen, 2012), but in this study, we are aiming to improve the latency in our AES design on FPGA since speed is an important issue beside of security mean while these two issues play in the favour of the hardware implementation of encryption/decryption applications.

Organization of the rest of this paper is as follows. Section “AES Algorithm” provides a brief overview of AES algorithm and remarks for FPGA design. Designs and architectures of AES encryption and decryption modules based on FPGA are presented in section “FPGA Implementation of AES encryption/decryption”. In this section, two designs are presented, in which design 2 is an optimized version of design 1. Based on this section, the next section provides the structure that combines encryption and decryption to build AES core based on FPGA. Section “Results” gives simulation results and comparisons with other works. Finally, last section shows the conclusion of this work.

AES ALGORITHM

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plain-text.

AES Encryption

The AES algorithm operates on a 128-bit block of data and executed Nr - 1 loop times. A loop is called a round and the number of iterations of a loop, Nr, can be 10, 12, or 14 depending on the key length of 128, 192 or 256 bits respectively. The first and last rounds differ from other rounds in that an additional AddRoundKey
Related Content

Grid Service Level Agreements Using Financial Risk Analysis Techniques
[www.igi-global.com/chapter/grid-service-level-agreements-using/40823?camid=4v1a](www.igi-global.com/chapter/grid-service-level-agreements-using/40823?camid=4v1a)

Balanced Job Scheduling Based on Ant Algorithm for Grid Network
[www.igi-global.com/chapter/balanced-job-scheduling-based-ant/61980?camid=4v1a](www.igi-global.com/chapter/balanced-job-scheduling-based-ant/61980?camid=4v1a)

A Novel Energy Efficient and SLA-Aware Approach for Cloud Resource Management
[www.igi-global.com/article/a-novel-energy-efficient-and-sla-aware-approach-for-cloud-resource-management/224031?camid=4v1a](www.igi-global.com/article/a-novel-energy-efficient-and-sla-aware-approach-for-cloud-resource-management/224031?camid=4v1a)
Persistence and Communication State Transfer in an Asynchronous Pipe Mechanism
www.igi-global.com/chapter/persistence-communication-state-transfer-asynchronous/54932?camid=4v1a