Computers and Electrical Engineering 37 (2011) 1160–1170

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Design of an ultra high speed AES processor for next generation IT security ☆

Liakot Ali [a,*], Ishak Aris [b], Fakir Sharif Hossain [a], Niranjan Roy [a]

[a] Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
[b] Institute of Advanced Technology (ITMA), Universiti Putra Malaysia, Malaysia

ARTICLE INFO

Article history:
Received 12 June 2010
Received in revised form 8 June 2011
Accepted 8 June 2011
Available online 22 July 2011

ABSTRACT

The Advanced Encryption Standard (AES) has added new dimension to cryptography with its potentials of safeguarding the IT systems. This paper presents the design of an ultra high speed AES processor to generate cryptographically secured information at a rate of multi-ten Gbps. The proposed design addresses the next generation IT security requirements: the resistance against all crypto-analytical attacks and high speed with low latency. This work optimizes AES algorithm to eliminate algebraic operations from the datapath, which contributes to achieve ultra high speed and to reduce the latency. The AES processor is designed using Verilog HDL and then simulated using FPGA platform. The performance of the processor is compared with that of other researchers in terms of speed and latency, which shows its superiority over them. The soft core can be reused to convert it to ASIC to achieve much better performance.

1. Introduction

Since demand for privacy and security of information are gradually emerging due to the rapid growth of information and communication technology, the research in defending information for coming generation is getting enormous importance [1,2]. Cryptographic algorithms form the fundamental aspect within this research field. Since the National Institute of Standards and Technology (NIST) accepted the AES to be the next generation IT security algorithm [3,4], a lot of research is going on to harness the power of AES in different security applications. There are two ways to put into practice any algorithm, i.e. hardware or software. A software implementation offers only limited physical security. But hardware implementation, by nature, is more physically secure, as they cannot easily be read or modified by an outside attacker. When any design is implemented into FPGA, it is possible to use 32-bit security code to protect the design from the attackers and it is very difficult to crack an FPGA based design while a software based design can be hacked any time. The most significant disadvantage of software based solutions is that the speed performance is significantly lower than that based on hardware. This research addresses the hardware realization of AES algorithm for any application requiring high speed security infrastructure.

There have been many different hardware realization and implementation of AES algorithm on FPGA and ASIC platform. Refs. [5–21] present the FPGA implementations of the AES algorithm. All of the architectures used in those works can achieve the throughput rate of several Gbps. The maximum throughput achieved on FPGA is 29.77 Gbps [14]. Refs. [22–31] depict the ASIC implementations of the AES algorithm. The works [22–24] present the possibilities of achieving throughput of not more than 3 Gbps but later on the work [25] able to present the throughput about 10 Gbps encryption. The effort [26] presents the possibilities of achieving a throughput of over 30 Gbps encryption using a 0.18 μm CMOS technology. Refs. [27,28] present an AES processor that runs from 30 to 70 Gbits/s with minimum area cost. Refs. [32–35] present the implementation of other aspects such as low power, Differential Power Attack (DPA), and Side channel analysis of the AES algo-

☆ Reviews processed and approved for publication to the Editor-in-Chief Dr. Manu Malek.

* Corresponding author. Tel.: +880 2 9665602.
E-mail addresses: liakot@iict.buet.ac.bd (L. Ali), ishak@eng.upm.edu.my (I. Aris).

0045-7906/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2011.06.003

rithm. Our research focuses on hardware realization of AES on Altera provided FPGA platform. The subsequent sections of this paper highlights the AES algorithm briefly, some novel design considerations for achieving high speed from the proposed AES processor, architecture of the chip, performance of the processor, results and its comparison with those of other researchers.

2. The AES algorithm

Detail description of the AES algorithm is presented in the literature [2,3]. It is seen from the literature that the input data is 128 bits in the AES. The input key can be 128, 192, or 256 bits long and the algorithm repeats for 11 rounds when 128-bit key is used. Different steps of the AES encryption and decryption algorithm for each round are presented in Fig. 1.

Fig. 1 shows that for both encryption and decryption, the algorithm starts with an add round key stage, followed by nine rounds, each of which contains all four stages; and then followed by a tenth round containing three stages, excluding Mix columns stage. Each stage is reversible. For SubBytes ( ), ShiftRows ( ) and MixColumns ( ) stages, there are corresponding inverse function – InvSubBytes ( ), InvShiftRows ( ) and InvMixColumns ( ). For the AddRoundKey ( ) stage, the inverse is achieved by XORing the same round key to the block, using the result: $A \oplus A \oplus B = B$. The decryption process makes use of the expanded key in reverse order. The description of the stages as shown in Fig. 1 is as follows:

2.1. Substitute bytes

This function uses an S-box to perform a byte-by-byte substitution of the block. For encryption and decryption, this function is indicated by SubBytes ( ) and InvSubBytes ( ), respectively. The SubBytes ( ) is a simple lookup table. This transformation is a non-linear byte substitution that operates independently on each byte of the state using a 16×16 matrix of byte values, called 'S-box'. This S-box is constructed by first computing the multiplicative inverse of each element in GF(2^8) with

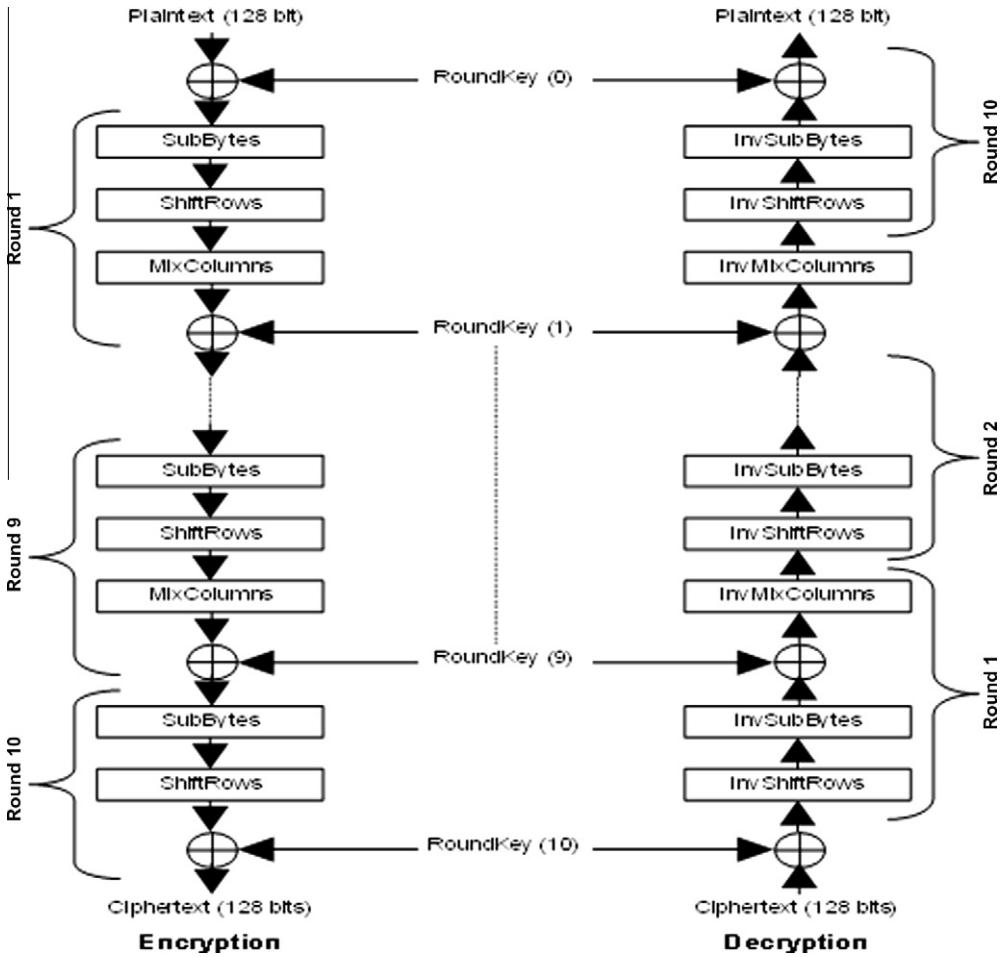

Fig. 1. AES encryption and decryption.