Available online at www.sciencedirect.com
SCIENCE DIRECT
Microprocessors and Microsystems 29 (2005) 317–326
MICROPROCESSORS AND MICROSYSTEMS
www.elsevier.com/locate/micpro

An AES crypto chip using a high-speed parallel pipelined architecture

S.-M. Yoo[a,*], D. Kotturi[b], D.W. Pan[a], J. Blizzard[b]

[a]*Electrical and Computer Engineering Department, The University of Alabama in Huntsville, 301 Sparkman Dr, Huntsville, AL, 35899 USA*
[b]*Cadence Design Systems, Inc., Plano, TX, USA*

Received 14 September 2004; revised 10 November 2004; accepted 16 December 2004
Available online 7 January 2005

Abstract

The number of Internet and wireless communications users has rapidly grown and that increases demand for security measures to protect user data transmitted over open channels. In December 2001, the National Institute of Standards and Technology (NIST) of the United States chose the Rijndael algorithm as the suitable Advanced Encryption Standard (AES) to replace the Data Encryption Standard (DES) algorithm. Since then, many hardware implementations have been proposed in literature. We present a hardware-efficient design increasing throughput for the AES algorithm using a high-speed parallel pipelined architecture. By using an efficient inter-round and intra-round pipeline design, our implementation achieves a high throughput of 29.77 Gbps in encryption whereas the highest throughput reported in literature is 21.54 Gbps.

Keywords: Encryption algorithm; Hardware implementation; Parallel pipelined design; Throughput

1. Introduction

The number of individuals and organizations using wide computer networks for personal and professional activities has recently increased a lot. A cryptographic algorithm is an essential part in network security. A well-known cryptographic algorithm is the Data Encryption Standard (DES) [13], which has been widely adopted in security products. However, serious considerations arise for long-term security because of the relatively short key word length of only 56 bits and from the highly successful cryptanalysis attacks.

In November 2001, the National Institute of Standards and Technology (NIST) of the United States chose the Rijndael algorithm as the suitable Advanced Encryption Standard (AES) [1] to replace the DES algorithm. Since then, many hardware implementations have been proposed in literature [2–12,15–22]. Some of them use field programmable gate arrays (FPGA) and some use application-specific integrated circuits (ASIC). The advantages of a software implementation include ease of use, ease of upgrade, portability, and flexibility. However, a software implementation offers only limited physical security, especially with respect to key storage [13]. Conversely, cryptographic algorithms (and their associated keys) implemented in hardware are, by nature, more physically secure, as they cannot easily be read or modified by an outside attacker. The downside of traditional (ASIC) hardware implementations is the lack of flexibility with respect to algorithm and parameter switching. Reconfigurable hardware devices such as FPGAs are a promising alternative for the implementation of block ciphers. FPGAs are hardware devices whose function is not fixed and can be programmed in-system.

In this paper, we present an implementation of the AES block cipher with Virtex II Pro FPGA using 0.13 μm and 90 nm process technology [14]. We have exploited the temporal parallelism available in the AES algorithm. Our chip contains the same ten units, and each unit can execute one round of the algorithm. Using external pipelined design, ten rounds of the algorithm are executed in parallel in a chip. Furthermore, using internal pipelining and key exchange pipelining, our implementation operating at 233 MHz achieves a throughput of 29.77 Gbps in encryption which

* Corresponding author. Tel.: +1 256 824 6858; fax: +1 256 824 6803.
E-mail addresses: yoos@ece.uah.edu (S.-M. Yoo), dkotturi@cadence.com (D. Kotturi), dwpan@ece.uah.edu (D.W. Pan), blizzard@cadence.com (J. Blizzard).

0141-9331/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2004.12.001

is much higher than the best (in terms of throughput) implementation reported in literature.

The rest of the paper is organized as follows. Section 2 describes briefly the AES cryptographic algorithm. Section 3 explains the details of our design on the AES cryptographic chip. Section 4 compares the performance of our implementation to earlier ones. Finally, Section 5 concludes the paper.

2. The AES algorithm and previous work

2.1. The AES algorithm

The AES algorithm is a symmetric block cipher that processes data blocks of 128 bits using a cipher key of length 128, 192, or 256 bits. Each data block consists of a 4×4 array of bytes called the *state*, on which the basic operations of the AES algorithm are performed. Fig. 1 shows the AES encryption and decryption procedures.

The encryption procedure is as follows. After an initial round key addition, a round function consisting of four different transformations—byte-sub, shift-row, mix-column, and add-round-key—is applied to the data block in the encryption procedure. The round function is performed iteratively 10, 12, or 14 times, depending on the key length. The mix-column operation is not applied to the last round. The byte-sub operation is a nonlinear byte substitution that operates independently on each byte of the state using a substitution table (S-Box). The shift-row operation is a circular shifting on the rows of the state with different numbers of bytes (offsets). The mix-column operation mixes the bytes in each column by the multiplication of the state with a fixed polynomial modulo x^4+1. Add-round-key operation is an XOR that adds a round key to the state in each iteration, where the round keys are generated during the key expansion phase.

The byte-sub transformation (S-Box operation), which consists of a multiplicative inverse over $GF(2^8)$ and an affine transform, is the most critical part of the AES algorithm in terms of computational complexity. However, the S-Box operation is required for both encryption and key expansion. Conventionally, the coefficients of the S-Box and inverse S-Box are stored in the lookup tables, or a hard-wired multiplicative inverter over $GF(2^8)$ can be used, together with an affine transformation circuit.

The decryption procedure of the AES is basically the inverse of each transformation. However, the standard decryption procedure is not identical to the encryption procedure. That is, the sequence of transformations for decryption differs from that for encryption, although the form of the key schedules for encryption and decryption is

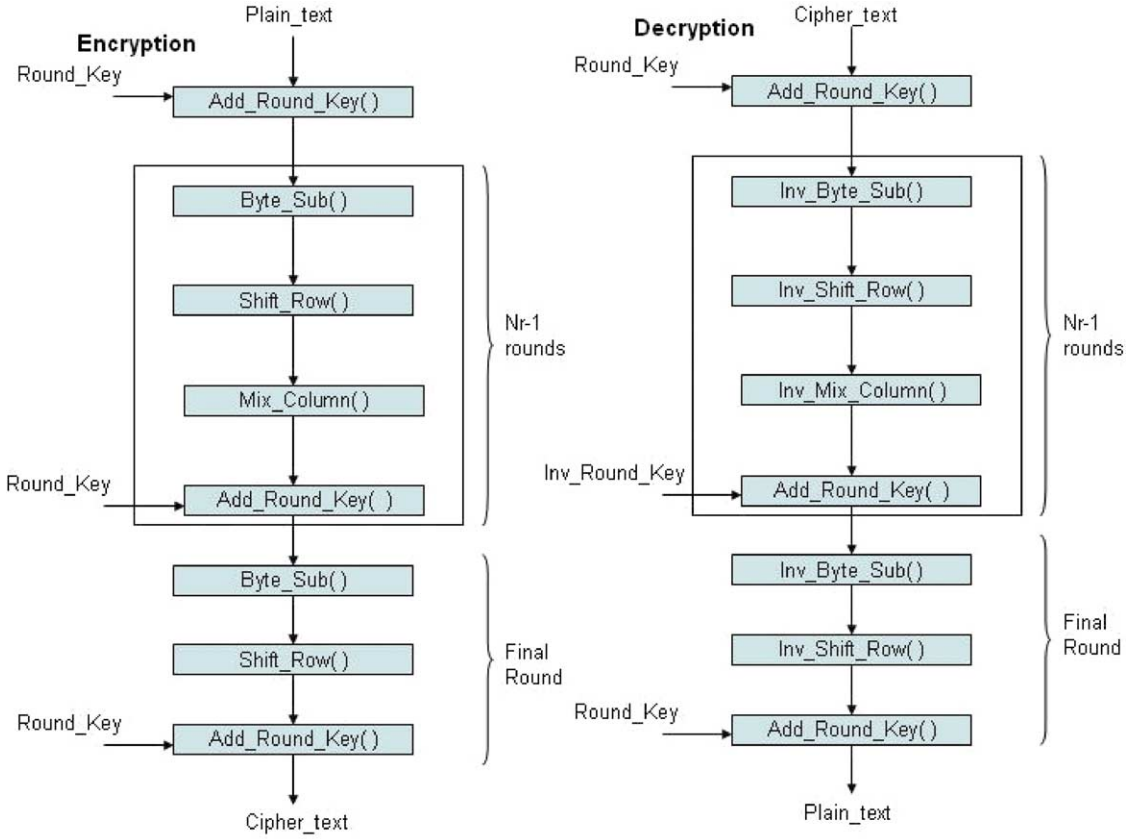

Fig. 1. The AES algorithm (equivalent version). (Nr: 10, 12, or 14 depending on key length).