

A Self-Testing Fully Pipelined Implementation for the Advanced Encryption Standard

Mahdi Nazm-Bojnordi, Naser Sedaghati-Mokhtari, and Seid Mehdi Fakhraie
{m.bojnordi, n.sedaghati} @ece.ut.ac.ir, fakhraie@ut.ac.ir
Silicon Intelligence and VLSI Signal Processing Laboratory
ECE Department, University of Tehran, Tehran, IRAN.

Abstract— In contrast to software implementation, hardware implementation of encryption protocols provides a higher level of security and cryptography speed at some flexibility cost. In this paper, different existing implementations of Advanced Encryption Standard (AES) are considered and a fully pipelined implementation for the AES is presented. Implementation considers both encryption and decryption. The design is optimized for achieving higher speed and lower area cost. The Selected algorithm for our design is Rijndael. The major part of an AES design is designing substitute boxes (S-box). S-boxes in our design are implemented at a lower cost rather than the existing implementations. Throughput of up to 6 Gbps is gained by our proposed architecture. This implementation is equipped with BIST architecture for self testing.

Index Terms—AES, self-testing, BIST, Rijndael, fully pipeline implementation.

I. INTRODUCTION

THE use of encryption/decryption is as old as the art of communication. The Advanced Encryption Standard (AES) is an encryption algorithm for securing sensitive but unclassified material by U.S. Government agencies. In March 1999, the National Institute of Standards and Technology (NIST) organized the Second Advanced Encryption Standard Candidate Conference (SAESCC) in Rome where a series of analyses of various algorithms were presented. This analysis includes evaluating not only their security capabilities, but also their performance, flexibility of implementation and other issues [1] Finally in October 2000, NIST announced the Rijndael as the winner algorithm for AES. In many implementations, Rijndael shows that it is an efficient algorithm for the AES; for example, in IPsec applications that handle more than thousands of security associations, it works very well [2].

In November 2001, the AES was accepted as a standard of the Federal Information Processing Standards (FIPS) [3]. Since then, because of high volume uses of encryption applications, many software and hardware implementations have been published for the AES, where each of them consider features of optimization differently. For example, it is used in

many places in Internet applications, such as routers.

Generally, when there is no concern on performance objectives and higher encryption or decryption speed, the software implementation may be the appropriate choice. In addition to lower speed of the software implementations, since the AES is a symmetric key encryption, the cipher key is easily vulnerable. In hardware implementation, however it is very hard to detect the cipher key by the attacker.

As a consequence, there is a growing interest in efficient implementations of the AES. For many applications, these implementations need to be resistant against side channel attacks, i.e. it should not be too easy to extract secret information from physical measurements on the device. Also there are many applications using AES as an efficient encryption method to secure their data exchange, such as the applications working on the networks. Also, the AES is an efficient and reliable method to be used in real-time applications such as multimedia broadcasting. Using the AES for real-time applications needs to consider low delay and fast architectures. In the literature, there are some AES hardware implementations for both ASIC and FPGAs. Also speed up to 609 Mbps is available for ASIC technology implementation [4]. In this paper a fast fully pipelined architecture for the AES encryption method is implemented that is suitable for securing data exchange in real-time applications such as video encryption.

The rest of this paper is organized as follows: The AES algorithm is explained in Section II, its hardware implementation and the BIST architecture are discussed in Section III and the paper conclusion appears in Section IV.

II. ALGORITHM

This section introduces a briefing on the selected algorithm by the NIST for the AES. The Rijndael algorithm is a symmetric block cipher (series of transformations that converts plaintext to ciphertext) methodology that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits. However, in excess of AES design criteria, the block sizes can mirror those of the keys [3]. Rijndael uses a variable number of rounds, depending on key/block sizes, as