AES-128 CIPHER, MINIMUM AREA, LOW COST FPGA IMPLEMENTATION

M. C. LIBERATORI† and J. C. BONADERO‡

† Laboratorio de Comunicaciones, Facultad de Ingeniería, UNMDP, Mar del Plata, Argentina
milibera@fi.mdp.edu.ar.
‡ Laboratorio de Comunicaciones, Facultad de Ingeniería, UNMDP, Mar del Plata, Argentina
jbona@fi.mdp.edu.ar

Abstract — The Rijndael cipher, designed by Joan Daemen and Vincent Rijmen and recently selected as the official Advanced Encryption Standard (AES) is well suited for hardware use. This implementation can be carried out through several trade-offs between area and speed. This paper presents an 8-bit FPGA implementation of the 128-bit block and 128 bit-key AES cipher. Selected FPGA Family is Altera FPGA, a minimum area core cipher that exploits the symmetry between encryption and decryption operations. The resulting design requires 957 LE’s, 6528 memory bits, operates at 11 Mbps and is compared with other known implementations.

Keywords — FPGA, cryptography, AES, cipher, VHDL.

I. INTRODUCTION

The Rijndael Algorithm, developed by Joan Daemen and Vincent Rijmen, has been approved by the U. S. National Institute of Standards and Technology (NIST) as the new Advanced Encryption Standard (AES). It became official in October 2000, replacing DES (FIPS 197, 2001). As this block cipher is expected to be widely used in an extensive variety of products, its efficient implementation becomes a significant priority. Hardware implementation is, by nature, more physically secure than software implementation.

The three major design targets with respect to hardware realization are: optimization for area or cost, low latency that minimizes time to encrypt a single block and high throughput to encrypt multiple blocks in parallel. All these design criteria involve a trade off between area and speed. There are a wide range of equipment where encryption is needed for authentication and security but throughput is not the principal concern. A low cost, small area design could be used in smart card applications as well as in other storage devices and low speed communication channels.

This paper presents an architecture for the 10 rounds AES Algorithm implemented on an Altera FPGA device. The goal of this design is to produce, in a low cost FPGA, a minimum area core cipher that exploits the symmetry between encryption and decryption operations.

The final architecture is based on previous work on the cipher design. In this work a decryption core is added, the number of clock cycles required to encrypt a single block has been reduced and the amount of hardware resources has been optimized with respect to the original design (Liberatori and Bonadero, 2005). A hierarchical design was adopted so that a collection of components were identified. Specified functions related with AES internal transformations were developed independently and composed to create the core cipher. The resulting design requires 957 LE’s, 6528 memory bits, operates at 11 Mbps and is compared with other known implementations.

II. THE AES ALGORITHM

AES is an iterative private-key symmetric block cipher (Stalling, 1999), operating on a block size of 128 bits. It comprises 10, 12 or 14 rounds when the key size is 128, 192 or 256 bits respectively. In an iterative block cipher, the block of information (plaintext) is transformed into another block of the same length (ciphertext) by repeated application of a round function. The intermediate cipher result is called the state in the AES proposal (Daemen and Rijmen, 1999). Each round involves an addition or bitwise EXOR of the plaintext and the key, so the original key must be expanded into a number of Round Keys and this transformation is known as the Key Schedule. A Round Key consists of a \(N_c\) word sub-array from the Key Schedule (Bonadero et al, 2005). In general the length of the cipher input, the cipher output and the cipher state is also \(N_c\), and is measured in multiples of 32 bits. Rijndael Algorithm allows \(N_c\) to take values 4, 6 or 8 but the AES standard only allows a length of 4. The length of the cipher key, \(N_k\) again measured in multiples of 32 bits, is also 4, 6 or 8, all of which are allowed by both Rijndael and the AES standard.

Each encryption round is composed of four operations: \SubBytes(), ShiftRows(), MixColumns() and AddRoundKey(). The last round is slightly different because MixColumns() is not present.

\SubBytes() transformation is a non-linear byte substitution as depicted in Fig. 1. Each byte of the state is inverted over GF(2^8) followed by an affine transformation (Murphy and Robshaw, 2001). The overall operation is known as the S-Box (Rijmen, 2003) and can be performed by using a look up table.

\ShiftRows() transformation operates on a whole state (128 bits). Rows of the state are cyclically shifted to the left as shown in Fig. 2.