Single-Chip FPGA Implementation of the Advanced Encryption Standard Algorithm

Máire McLoone , John V. McCanny

DSiP™ Laboratories, School of Electrical and Electronic Engineering,
The Queen's University of Belfast, Belfast BT9 5AH, Northern Ireland
Maire.McLoone@ee.qub.ac.uk, J.McCanny@ee.qub.ac.uk

Abstract. A single-chip FPGA implementation of the new Advanced Encryption Standard (AES) algorithm, Rijndael is presented. Field Programmable Gate Arrays (FPGAs) are well suited to encryption implementations due to their flexibility and an architecture, which can be exploited to accommodate typical encryption transformations. The FPGA implementation described here is that of a fully pipelined single-chip Rijndael design which runs at a data rate of 7 Gbits/sec on a Xilinx Virtex-E XCV812E-8-BG560 FPGA device. This proves to be one of the fastest single-chip FPGA Rijndael implementations currently available. The high Block RAM content of the Virtex-E device is exploited in the design.

Keywords: FPGA Implementation, AES, Rijndael, Encryption

1 Introduction

On the 2nd October 2000 the US National Institute of Standards and Technology (NIST) selected the Rijndael algorithm [1], developed by Joan Daemen and Vincent Rijmen, as the new Advanced Encryption Standard (AES) algorithm. It proved to be a fast and efficient algorithm when implemented in both hardware and software across a range of platforms. Rijndael is to be approved by the NIST and replace the aging Data Encryption Standard (DES) algorithm as the Federal Information Processing Encryption Standard (FIPS)[2] in the summer of 2001. In the future Rijndael will be the encryption algorithm used in many applications such as:

- Internet Routers
- Remote Access Servers
- High Speed ATM/Ethernet Switching
- Satellite Communications
- High Speed Secure ISP Servers
- Virtual Private Networks (VPNs)
- SONET
- Mobile phone applications
- Electronic Financial Transactions

In this paper a single-chip FPGA implementation of the Rijndael algorithm is presented. The fully pipelined design is implemented using Xilinx Foundation Series 3.1i software on the Virtex-E XCV812E FPGA device [3]. A 10-stage pipelined

G. Brebner and R. Woods (Eds.): FPL 2001, LNCS 2147, pp. 152-161, 2001.

Rijndael design requires considerable memory; hence, its implementation is ideally suited to the Virtex-E Extended Memory range of FPGAs, which contain devices with up to 280 RAM Blocks (BRAMs).

The fastest known Rijndael FPGA implementation is by Chodowiec, Khuon and Gaj [4], which performs at 12160 Mbits/sec. Their design is implemented on 3 Virtex XCV1000 FPGA devices. Dandalis, Prasanna and Rolim [5] also carried out an implementation on the XCV1000 device, achieving an encryption rate of 353 Mbits/sec. A partially unrolled design by Elbirt, Yip, Chetwynd and Paar [6] on the same device performed at a data-rate of 1937.9 Mbits/sec. The fastest Rijndael software implementation is Brian Gladman's [7] 325 Mbit/sec design on a 933 MHz Pentium III processor. Whereas, an earlier paper [8] described a high-speed generic Rijndael design, which supported three key lengths, this paper assumes a 128-bit data block and a 128-bit key and concentrates on using an optimum number of Block RAMs to attain high throughput.

Section 2 of this paper describes the Rijndael Algorithm. The design of the fully pipelined Rijndael implementation and the exploitation of the Block RAMs on the Virtex E device are outlined in Section 3. Performance results are given in section 4 and conclusions are provided in section 5.

2 Rijndael Algorithm

The Rijndael algorithm is a substitution-linear transformation network [9]. It can operate on 128-bit, 192-bit and 256-bit data and key blocks. The NIST requested that the AES must implement a symmetric block cipher with a block size of 128 bits, hence the variations of Rijndael which can operate on larger data block sizes will not be included in the actual FIPS standard. An outline of Rijndael is shown in Fig. 2.1.

Rijndael comprises 10, 12 and 14 iterations or rounds when the key lengths are 128, 192 and 256 respectively. The transformations in Rijndael consider the data block as a 4 column rectangular array of 4-byte vectors (known as the *State* array), as shown in Fig 2.2. A 128-bit plaintext consists of 16 bytes, B_0, B_1, B_2, B_3, B_4… B_{14}, B_{15}. Hence, B_0 becomes $P_{0,0}$, B_1 becomes $P_{1,0}$, B_2 becomes $P_{2,0}$ … B_4 becomes $P_{0,1}$ and so on. The key is also considered to be a rectangular array of 4-byte vectors, the number of columns, N_k, of which is dependent on the key length. This is illustrated in Fig 2.3. This paper assumes a 128-bit key and therefore a similar rectangular array is considered for the key as for the data block. The algorithm design consists of an initial data/key addition, nine rounds and a final round, which is a variation of the typical round. The Rijndael key schedule expands the key entering the cipher so that a different sub-key or *round key* is created for each algorithm iteration. The Rijndael round comprises four transformations:

- ByteSub Transformation
- ShiftRow Transformation
- MixColumn Transformation
- Round Key Addition

The ByteSub transformation is the *s-box* of the Rijndael algorithm and operates on each of the State bytes independently. The s-box is constructed by finding the multiplicative inverse of each byte in GF(2^8). An affine transformation is then