Microprocessors and Microsystems 41 (2016) 37–46

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

A high performance ST-Box based unified AES encryption/decryption architecture on FPGA

D.-S. Kundi*, Arshad Aziz, Nassar Ikram

Department of Electrical Engineering (PNEC), National University of Sciences and Technology (NUST), Islamabad, H-12, Pakistan

ARTICLE INFO

Keywords:
AES
BRAM
Look-Up-Table
FPGA
Unified encryptor and decryptor

ABSTRACT

In this paper, a unified Field Programmable Gate Array (FPGA) based Advanced Encryption Standard (AES) encryptor/decryptor design is presented by proposing a symmetric ST-Box structure. This structure fully utilizes high capacity (32 Kb) Block RAM (BRAM) by accommodating all encryption and decryption lookup operations within a single BRAM in the form of single integrated Look-Up-Table. This design also caters the inherent asymmetric nature of encryption and decryption coefficients for a unified hardware. Further the symmetry at BRAM output is maintained to use a single XOR network during both encryption and decryption. The performance of design is enhanced by proposing a duty-cycle based accessing technique. It explores the switching capabilities of BRAM and effectively minimizes the ON time of BRAM by changing duty-cycle of input clock. This enables us to access single BRAM 4 times per clock. Effectiveness of design is further measured by implementing it, in both iterative and pipelined architectures. Our proposed iterative design on Virtex-7 proved to be the smallest 128-bit unified AES core with 48.70% reduced resources and the best Throughput Per Slice (TPS) of 11.56. Similarly our pipelined design saved 59.01% area and has the highest throughput of 45.69 Gbps.

1. Introduction

With technological evolution of computerized systems, security and confidentiality have become one of the main concerns for these systems to avoid frauds. Cryptography plays a vital role in this regard to provide privacy, authentication and integrity protection. Nowadays, cryptography is largely used in Cellular phones, Cable/Sat TV broadcasts, radio modems, Smart cards, ATM networks, Garage door openers, online banking etc. Advanced Encryption Standard (AES) Algorithm [1] is one of the most popular and widely used algorithms for symmetric key cryptography to protect sensitive data. A lot of efficient hardware implementations of AES are available in open literature on both Application Specific Integrated Circuit (ASIC) [2–5] and FPGA [6–9] platforms.

FPGA platforms are ideal for the implementation of cryptographic algorithms. They are reconfigurable that give both time and cost effective solutions as compared to ASICs, that require largest development time and are expensive [10]. In addition to this, FPGAs also provide far better speed performance than software implementations and at the same time can be re-programmed on the fly to store updated encryption standard. Modern generations of FPGA apart from LUTs are now equipped with special embedded features such as Multi-mode Clock Manager (MMCM) and BRAM for the implementation of high-density and high-performance designs. An active area of research in optimization of crypto-system on FPGAs focuses not only to use these new embedded features of FPGA but how efficiently and effectively these features are to be used in order to enhance the performance of these crypto-system in terms of both area and speed [11,12].

For efficient implementation of AES on FPGA, two Look-Up-Table based approaches exist; S-box and T-box. In these approaches, all the computational complexities of most critical transforms of AES are replaced by a simple Look-Up-Table. But encryptor and decryptor based on these Look-Up-Tables are not only memory intensive but also asymmetric in nature due to AES operations and sequence of transformations in encryption and decryption. Therefore in most of these Look-Up-Table based AES implementations [7–9,13], encryptor and decryptor cores are implemented separately and occupied considerable amount of BRAM resources on FPGA. So there is a need to design a unified AES encryption and decryption module to minimize BRAM resources and also to efficiently utilize full memory space of 32 Kb BRAM available in new generations of FPGA devices.

In this research we present an efficient Look-Up-Table based unified AES encryptor and decryptor core on 7 Series (Virtex-7, Atrix-7) FPGA that hides all the complexities of separate encryptor and decryptor cores in a form of single-unit hardware. The architecture efficiently utilizes the 32 Kb BRAM by proposing a symmetric

* Corresponding author. Tel.: +92 3212297022.
E-mail address: shahwar@pnec.nust.edu.pk, durshi1@gmail.com (D.-S. Kundi).

http://dx.doi.org/10.1016/j.micpro.2015.11.015
0141-9331/

and integrated ST-Box Look-Up-Table for both encryptor and decryptor followed by a single XOR network. In addition to this, design also includes enhanced duty-cycle based accessing technique for BRAM that provides multiple outputs in one clock cycle without degrading overall system performance. The effectiveness of our unified AES encryptor and decryptor core is then evaluated in both iterative and pipelined architectures. Our both unified AES iterative and pipelined cores saved considerable amount of area resources thereby utilizing minimal number of BRAMs and at the same time result in best design efficiencies (that is, throughput/area) and throughput than all the known and up-to-date FPGA based AES combined implementations.

This paper is organized as follows: In Section 2, we give a brief introduction of AES and its four transformations. Section 3 describes general architecture of FPGA and its special features while Section 4 summarizes the related work. In Section 5 we present the details and features of our unified AES hardware including the enhanced duty-cycle based accessing technique. Implementation results and comparison with up-to-date AES implementations are given in Section 6. While in Section 7, we conclude our work.

2. Advance encryption standard

AES is a symmetric block cipher that supports data block of 128-bit and variable key sizes of 128, 192 and 256 bits. In AES, input data is arranged in 4×4 array of bytes called a State, with four rows and four columns consisting of 16 bytes in total. It uses a round function that is composed of four different byte-oriented transformations, i.e. ShiftRow, SubByte, MixColumn and AddRoundKey, except for final round in which MixColumn transformation is not performed and initial round at start that comprises of only AddRoundKey transformation [1]. The number of 10/12/14 rounds to be executed depending on key size of 128-bit/192-bit/256-bit respectively.

2.1. SubByte

SubByte is a non-linear byte substitution in which each byte is replaced with another byte according to S-box table as given by (1), where $b_{i,j}$ is byte with ith row and jth column of a State. S-box is invertible and is derived by the application of two transformations: First taking multiplicative inverse (MI) in GF(2^8) with element 00 being mapped to itself and then applying an affine transform (AF) over GF(2). Alternately it can be implemented as a Look-Up-Table with pre-computed values called S-box.

$$b_{i,j} = S[b_{i,j}] \tag{1}$$

2.2. ShiftRow

ShiftRow is a transposition step that cyclically shifts last three rows of State with a certain number of steps to the left. The second row is shifted by one byte, third row is shifted by two bytes and fourth row is shifted by three bytes.

2.3. MixColumn

MixColumn operates on every column of the State. Each column is considered as a four-term polynomial over GF(2^8) and multiplied modulo x^4+1 with fixed polynomial $c(x)$:

$$c(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \tag{2}$$

The above can be written in terms of a simple matrix multiplication:

$$\begin{bmatrix} b'_{0,c} \\ b'_{1,c} \\ b'_{2,c} \\ b'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 01 & 01 & 03 \\ 01 & 01 & 03 & 02 \\ 01 & 03 & 02 & 01 \\ 03 & 02 & 01 & 01 \end{bmatrix} \begin{bmatrix} b_{0,c} \\ b_{1,c} \\ b_{2,c} \\ b_{3,c} \end{bmatrix} \tag{3}$$

As a result of this multiplication, four bytes in a column are replaced by the following equations:

$$\begin{aligned} b'_{0,c} &= (\{02\} \bullet b_{0,c}) \oplus (\{03\} \bullet b_{1,c}) \oplus b_{2,c} \oplus b_{3,c} \\ b'_{1,c} &= b_{0,c} \oplus (\{02\} \bullet b_{1,c}) \oplus (\{03\} \bullet b_{2,c}) \oplus b_{3,c} \\ b'_{2,c} &= b_{0,c} \oplus b_{1,c} \oplus (\{02\} \bullet b_{2,c}) \oplus (\{03\} \bullet b_{3,c}) \\ b'_{3,c} &= (\{03\} \bullet b_{0,c}) \oplus b_{1,c} \oplus b_{2,c} \oplus (\{02\} \bullet b_{3,c}) \end{aligned} \tag{4}$$

2.4. AddRoundKey

AddRoundKey performs an addition (bitwise XOR) of a State with RoundKey and each RoundKey is derived from cipher key using a key scheduler.

3. Field programmable gate array

FPGAs are best leading representative of reconfigurable devices of modern era and are suitable for hardware implementation of cryptographic primitives. Basic architecture of modern Xilinx FPGA consists of Configurable Logic Blocks (CLBs), I/O pads and embedded resources such as enhanced BRAM, MMCM etc.

CLBs are main building blocks to implement different logic designs, whether its combinational or sequential logic. They are available in the form of grid array and each CLB consists of two type of Slices, SliceL with only logic functionality and SliceM with additional capability to be configured as distributed memory (64×1 bit) or Shift Register (32-bit) in addition to logic implementation. Each Slice consists of four 6-input LUT technology that have improved performance and great abilities as compared to previous generations of Xilinx FPGA [14].

Although SliceM LUTs can be configured as distributed RAM or ROM to store Look-Up-Table, but this method is not resourceful for storage of large Look-Up-Table as most of Slice LUTs will be utilized. So for this purpose embedded BRAM modules [15] are available on Xilinx FPGA chip in the form of 36 Kb blocks (32 Kb for data and 4 Kb for parity). Each 36 Kb BRAM can be configured as a two independent 18 Kb blocks or a single 36 Kb block or even can be combined to produce large memory blocks of 64 Kb without any extra logic resources. These BRAMs, if efficiently utilized are ideal for the implementation of Look-Up-Table based designs but they are synchronous in nature in which we can access data from its ports only once per cycle.

The MMCMs [16] are available at the top and bottom of the FPGA chip and provide an extensive range of powerful and build in clock management features such as frequency synthesizer, phase shifter and delay locked loop (DLL). These MMCM resources can be used to produce optimized clocking circuitry for digital designs.

4. Previous work

Asymmetric nature of AES encryption and decryption processes not only doubles number of area resources required for complete core but also limits resource sharing among the encryptor and decryptor. Due to this, limited number of implementations of combined AES encryption and decryption cores exist. Among these combined AES encryption and decryption cores, the most commonly used two techniques are composite field arithmetic GF(2^8) [17–32] and Look-Up-Table [33–40].

In composite field technique, it is possible to share the hardware of Multiplicative Inverse (MI) unit between SubByte (SB) and inverse SubByte (ISB) transform while in case of MixColumn (MC) and inverse MixColumn (IMC), the constant matrix of IMC is usually decomposed into product of two matrices; the same constant matrix as in MC and another new constant matrix with less Hamming weight in order to reuse hardware of MC in decryption. So for efficient integrated AES core, different optimized architectures based on