ISSN : 2231-0711

Implementation of AES as a Reconfigurable Cryptographic Embedded system using MicroBlaze & Xilinx ISE

M. P. Jaiswal[1], Prof. G.G. Sarate[2], Prof. S. R. Hirekhan[3],

Microprocessor VLSI Laboratory, Electronics and Telecommunication Engineering Department,
Sant Gadge Baba Amravati University,
Government College of Engineering Amravati. (M. S.), India.

ABSTRACT

In this paper implementation of AES as a reconfigurable cryptographic embedded system is described. With some proposed techniques, an optimized structure of AES is discussed. The implementations of AES are described as a reconfigurable hardware approach of embedded system using MicroBlaze SCP. A MicroBlaze is a soft-core processor especially designed for Xilinx field programmable gate arrays.

Keywords: Advanced Encryption Standard (AES), Soft-core Processor (SCP), System-on-Chip (SoC), VLSI, Chiper.

I. INTRODUCTION

Cryptographic applications are becoming increasingly more important in today's world of data exchange. Big volume data needs to be transferred from one location to another through communication path but exposes to attackers. Cryptography services are essential in order to provide the authentication, privacy, non-denial and integrity of private data being transmitted. System-on-Chip (SoC) technology enters the mainstream in digital design. The advances in reconfigurable hardware create the possibility of developing a microchip with application-specific soft core processors.

In this paper we try to discuss AES as one of the cryptographic techniques and its transformation [1], various implementation techniques of AES [2] and Microblaze SCP for reconfigurable hardware which operates under system-on-chip environment [5]. For implementation of AES, a MicroBlaze SCP is proposed. A MicroBlaze soft-core processor especially designed for Xilinx a field programmable gate array which provides reconfigurable embedded system. The Xilinx ISE provides facility of pure hardware approach, while Xilinx EDK provides software plus hardware approach. For design purpose VHDL hardware description language and embedded C are used. The implementation of proposed techniques in this paper will provide a step to design a complete cryptographic system processor for security application in embedded system.

II. AES

The AES a round-based, symmetric block cipher was standardized by NIST as Advanced Encryption Standard (AES) in November 2001[1]. The AES is the preferred algorithm for implementations of cryptographic protocols that are based on a symmetric cipher. It is not only used to secure data transfers between small, mobile consumer products, but it is also used in high end servers.

AES has a block size of 128 bits and key lengths of 128, 192, and 256 bits. According to the key length, these variants of the AES are called AES-128, AES-192, andAES-256. This article mainly focuses on implementing the AES-128, which is the most commonly used AES variant shown in Fig. 1. However, the shown block diagram can also be used for the other standardized key sizes.

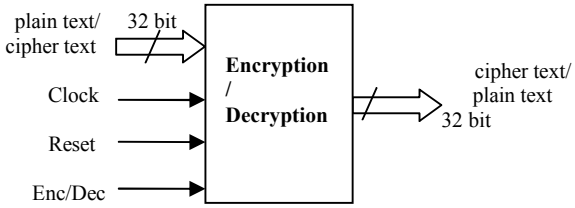

Fig 1 Block diagram of AES

The following section and subsections describes the AES transformations, which are the building blocks of AES encryptions and decryptions.

AES Transformations

The AES takes a 128-bit data block as input and performs several different transformations on this block. In case of an encryption, the input block of the AES is called plaintext and the returned block is called ciphertext. All intermediate results of this block, as well as the input and the output block, are called states. For a discussion of the different transformations, executed on the 128-bit states in an AES encryption or decryption, it is best to picture a state as a 4-by-4 matrix of bytes. A 128-bit input/output block of the AES is mapped to an AES state by putting the first byte of the block in the upper left corner of the matrix and by filling in the remaining bytes column by column as shown in Fig. 2.

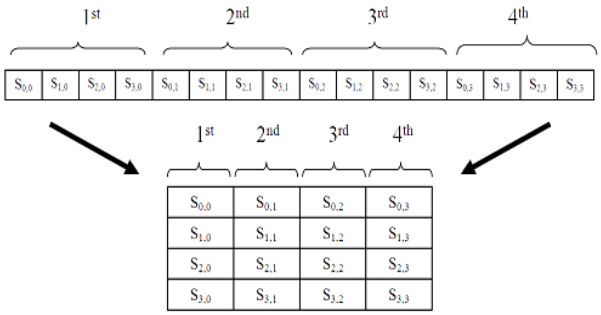

Fig 2 Array of data

AES encryptions and decryptions are based on four different transformations that are performed repeatedly in a certain sequence shown in Fig. 3. Each of these transformations, which are described in the following, maps a 128-bit input state to a 128-bit output state.

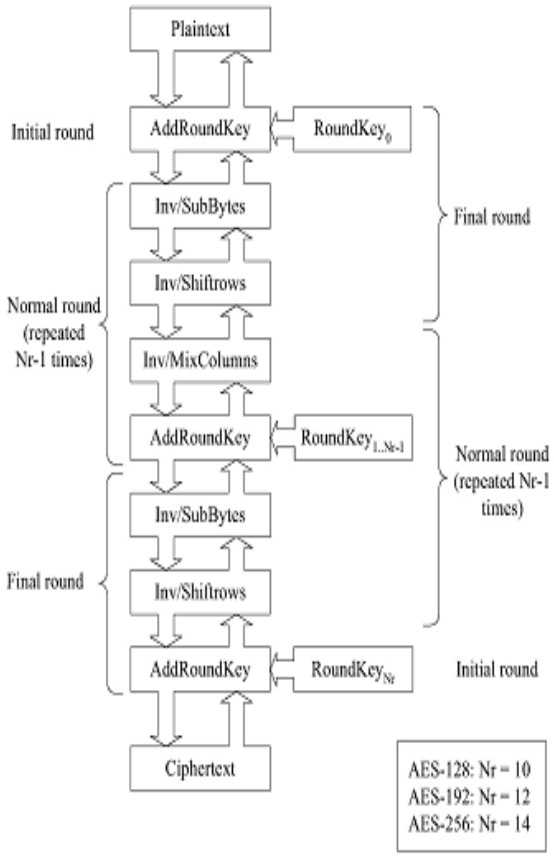

Fig 3 Flowchart of AES transformation

- **AddRoundKey transformation**

In the AddRound Key transformation, a Round Key is added to the State by a simple bitwise XOR operation. The Add Round Key transformation is self-inverting.

- **SubBytes transformation**

The SubBytes transformation is a non-linear byte substitution that operates independently on each byte of the state using a substitution table (S-box). This S-box, which is invertible, is constructed by composing two transformations:

i) Take the multiplicative inverse in the Galois Field GF(2^8) with the irreducible polynomial m(x)= $x^8+x^4+x^3+x+1$. The element {00} is mapped to itself.

ii) Apply the affine transformation (over GF(2)): The inverse of SubBytes transformation, which is needed for decryption, is the inverse of the affine transformation followed by the same inversion as the SubBytes transformation.

- **ShiftRows transformation**

The ShiftRows transformation rotates each row of the input state to the left, the offset of the rotation corresponds to the row number. The inverse of this transformation is computed by performing the corresponding rotations to the right.

- **MixColumns transformation**

The MixColumns transformation operates on the State column-by-column, treating each column as a four term polynomial. The columns are considered as polynomials over GF (2^8) and multiplied modulo $x^4 + 1$ with a fixed polynomial a(x), given by

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

The coefficients of a(x) are also elements of GF(2^8) and are represented by the hexadecimal values in this equation. The inverse MixColumn Transformation is the multiplication of each column with $a^{-1}(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}$ modulo $x^4 + 1$

III. Implementation of AES

AES can be implemented by Pure Software, Pure Hardware or both. A pure software approach suffers from low speed and throughput; it is also exposed to viruses and hackers attack which hampers the performance of Crypto-System. A pure hardware gives high speed. This technique can be implemented by using Xilinx ISE s/w and Spartan3E xc3s500e-4fg320 h/w. But it does not gives flexibility. Hence designer has to look towards new option i.e. hybrid of above two system which gives speed as well as flexibility i.e. by using FPGAs based or ASIC based design [5]. Although ASIC gives the highest performance and the lowest unit cost but it has no flexibility at all. While FPGAs are idle for the run-time hardware configuration. But this fact is unacceptable for some of the systems, due to high cost.

Recently the research work is going on for implementing the algorithm with minimum area, linearity, high speed and design flexibility option in VLSI hardware [6]. Hence new architecture has come which is based on a cooperation between a general purpose core processor and reconfigurable hardware.

This technique can be implemented by using Xilinx EDK s/w in MicroBlaze SCP and Spartan3E xc3s500e-4fg320 h/w.

This approach is known as system-on-chip which comes under Embedded System. This type of embedded system speeds up the performance and provides facility of reconfiguration without interrupting present execution.

IV. Xilinx MicroBlaze SCP

MicroBlaze is a 32-bit RISC Harvard-style SCP [5].It is offered with the Embedded Development Kit (EDK), the tool provided by Xilinx Inc. to design FPGA-based systems-on-a-chip. The processor architecture includes 32-bit general-purpose registers and an orthogonal instruction set. It features a three-stage instruction pipeline, with delayed branch capability for improved instruction throughput. As it is a SCP, the functional units incorporated into the processor architecture can be customised in order to fit as much as possible the target application. Thus, the barrel shifter unit, hardware divider unit, data cache and instruction cache are optionally instantiated along with the processor. Also, in FPGAs with embedded multipliers, a multiplication unit is available. A typical system based on MicroBlaze is shown in Fig. 4.

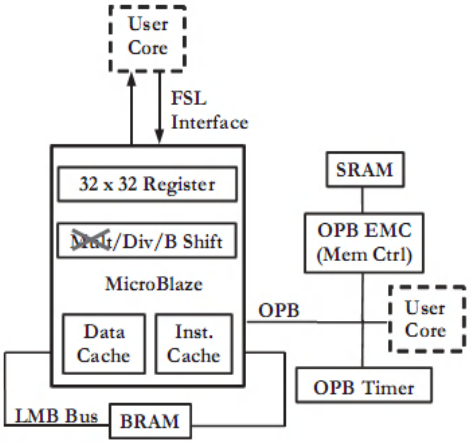

Fig. 4. MicroBlaze system

The EDK toolkit allows the designer to easily create platforms based on either MicroBlaze or PowerPC-405. EDK provides many peripherals (UARTs (universal asynchronous receiver transmitters), timers, Ethernet, memory controllers, general purpose I/O (input/output) and so on) and an interconnection solution based on IBM's Core Connect bus fabrics [5]. The GNU compiler tools for MicroBlaze and PowerPC-405 is used in the software flow. The source code for the application can be written in high-level languages, such as C and CPP, as well as in assembly language.

Three types of systems can be implemented by an embedded cipher system based on MicroBlaze. The first type of embedded system consists of the SCP without any customised core, that is, a complete software solution. The other two types of embedded systems integrate a customised user core, which implements the complete ciphering algorithm in order to increase the application performance. The former uses the OPB bus to connect the cipher module as external peripheral, whereas the latter uses the same hardware implementation but using the FSL interface to connect it as coprocessor.

Fig. 5 describes the architecture core of AES. This core can implement fully unrolled and pipelined approach which gives very high clock frequency. A key generator unit is designed to generate the different sub-keys employed in each round. In the proposed MicroBlaze-based cryptographic systems, the design efforts have been minimized and the performance obtained is suitable to execute secure applications with high throughput.

Designers can use FPGAs to create efficient hardware designs, as many systems require a combination of both software and hardware. SCPs give designers the flexibility to configure the processors and facilitate those designers to quickly build FPGA systems incorporating one or more processors and coprocessors.

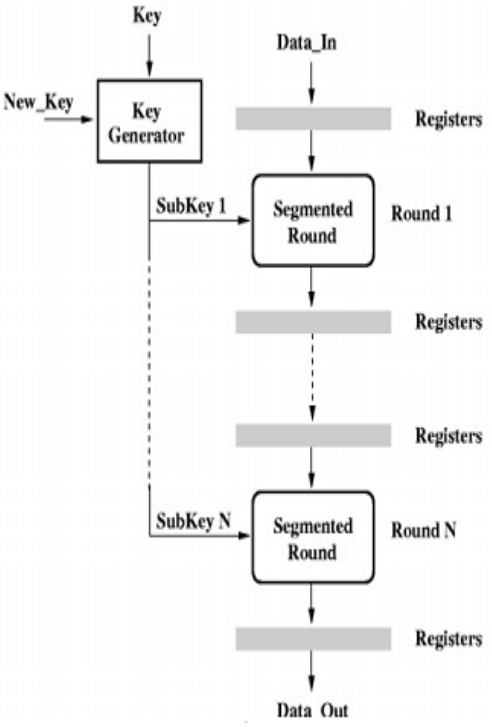

Fig. 5 Architecture of AES core.

V. Conclusion

FPGA-based systems are able to combine general-purpose microprocessors and dedicated hardware cores as a system-on-a-chip solution. Some factors limit the improvement of the performance in FPGA-based embedded system. This technology provides high data transfer rate of the interface between the embedded processor and the configurable logic, it also provides the speed of the embedded processor and the memory bandwidth. One of the most important bottlenecks of this technology is the bandwidth and the latency of the interface connecting the embedded processor to the reconfigurable logic. Thus, in a MicroBlaze-based system, the flexibility offered by FPGAs can be used to notably increase the throughput of secure applications.

REFERENCES

[1] National Institute of Standards and Technology (NIST), Federal Information Processing Standard 197, "The Advanced Encryption Standard (AES)", Nov. 2001.
http://csrc.nist.gov/publications/fips/fips197/fips-197

[2] Liang Deng and Hongyi Chen, "A New VLSI Imple-mentation of the AES Algorithm",vol 2 pp 7803-7547, IEEE.

[3] S. Mangard; M.Aigner; S. Dominikue, "A Highly Regular and Scalable AES Hardware Architecture," IEEE Transactions on Computers, volume 52 pp.483- 491, April 2003.

[4] X. Zhang and K. K. Parhi, "Implementation approaches for the advanced encryption standard algorithm," IEEE Circuits Syst. Mag., vol. 2, no. 4, pp. 24–46, 2002.

[5] I. Gonzalez and F.J. Gomez-Arribas, "Ciphering algorithms in Micro-Blaze-based embedded systems," IEE Proc.-Comput. Digit. Tech., Vol. 153, No. 2, March 2006.

[6] Raoel Ashruf, Georgi Gaydadjiev, Stamatis Vassiliadis, "Reconfigurable Implementation for the AES Algorithm", IEEE 2004.

[7] Ivan Gonzalez, Estanislao Aguayo Sergio Lopez-Buedo, "Self-Reconfigurable Embedded Systems On Low-Cost FPGAS," IEEE 2007

[8] J. Viejo, M. J. Bellido, A. Millan, E. Ostua J. Juan, P. Ruiz-de-Clavijo and D. Guerrero, "Efficient Design and Implementation on FPGA of a MicroBlaze Peripheral for Processing Direct Electrical Networks Measurements," IEEE 2006.