Advanced Computing: An International Journal ( ACIJ ), Vol.3, No.4, July 2012

# IMPLEMENTATION OF AES AS A CUSTOM HARDWARE USING NIOS II PROCESSOR

Meghana Hasamnis[1] and Priyanka Jambhulkar[2] and S. S. Limaye[3]
[1]Associate Professor, Department of Electronics Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India
meghanahasamnis@rediffmail.com

[2]Department of Electronics Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India
priyaj157@gmail.com

[3]Professor, Department of Electronics Engineering, Jhulelal Institute of Technology, Nagpur, India
shyam_limaye@hotmail.com

***ABSTRACT***

*In this paper Advanced Encryption Standard (AES) algorithm has been designed and implemented as custom hardware. The algorithm is controlled through C-code written in NIOS II IDE. AES as a custom hardware is interfaced with the system designed around NIOS II Processor using SOPC builder tool. AES is written in hardware in VHDL language and the interface is through GPIO (General Purpose Input / Output Port). AES implemented using data size of 128 bits, while the length of the key used is of 128 bits. The key size of AES used is of 128 bits, as it is secure from the different attacks in existence. The FPGA used is CYCLONE II from Altera. AES as a custom hardware increases the speed of encryption and serves as an accelerator and hence improves the performance of the system.*

***.Keywords***

*Advanced Encryption Standard (AES), NIOS II Processor, SOPC Builder, NIOS II IDE.*

## 1. INTRODUCTION

Cryptography plays an important role in the security of data. It enables to store sensitive information or transmit it across insecure networks so that unauthorized persons cannot read it. Advanced Encryption Standard (AES) is the most common encryption algorithm widely used in applications such as wireless communication [1]. The Advanced Encryption Standard (AES) is well known block-cipher algorithm which is easily portable and reasonable security. For secure exchange of digital data, resulted in large quantities of different encryption algorithms this can be classified into two groups: Symmetric encryption algorithm (with private key algorithms) and Asymmetric encryption algorithm (with public key algorithms). Symmetric key algorithms are in general much faster to execute electronically than asymmetric key algorithms.

The most commonly used symmetric encryption algorithm is AES. The input plain text and the cipher key are in state array fashion and hence known as a block cipher. The plaintext input are of fixed size, blocks of 128 bits and produces a block of ciphertext of equal size for each plaintext block. The most commonly used symmetric encryption algorithms are the data encryption standard (DES), triple data encryption algorithm (TDEA) and advanced encryption standard (AES). TDEA has two features which ensure its widespread use over years. First, with its 168-bit key length, it overcomes the vulnerability to brute-force attack of DEA which has 56-

bit key length. Second, the underlying encryption algorithm in TDEA is the same as in DEA. The main drawback of TDEA is that the algorithm is relatively slow in software. The original DEA was designed around mid-1970's suitable for hardware implementation but does not produce efficient software code. TDEA, which has three times as many as round as DEA, is correspondingly slower. A secondary drawback is that both DES and TDEA use a 64-bit block size and hence are prone to attacks [2].

For reasons of both efficiency and security a larger block size is desirable. As a replacement, NIST in 1997 issued a call for proposals for a new advanced encryption standard (AES),which should have a security strength equal to or better than TDEA and significantly improved efficiency. In addition to these general requirements, NIST specified that AES must be a symmetric block cipher with a block length of 128-bits and support for key lengths of 128,192 and 256 bits.

## 2. AES ALGORITHM

The AES algorithm is a symmetric-key cipher, in which both the sender and the receiver uses a single key for encryption and decryption. The length of the plain text is fixed to be 128 bits, while the key length can be either 128,192, or 256 bits. The key length selected is of 128 bits. AES algorithm is an iterative algorithm. Every iteration can be called a round, and the total number of rounds is 10, 12, or 14, when key length is 128, 192, or 256 respectively. The 128 bit algorithm is divided into 16 bytes. These bytes are represented into 4x4 array called the state array, and all the different operations of the AES algorithm such as addroundkey, subbytes, shiftrows, mixcolumns and key expansion are performed on the state [3].

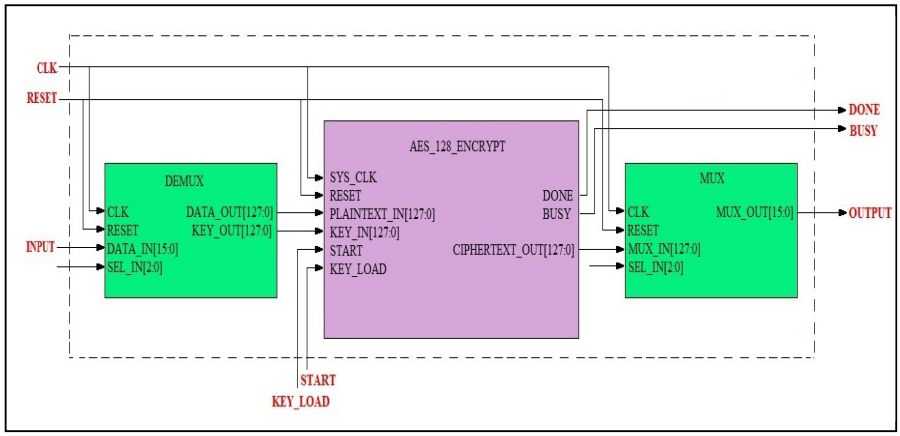


Fig.1 Block Diagram of AES

In AES algorithm encryption of data consists of ten rounds. Each round consists of four operations or transformations. Only the last round i.e. the tenth round has only three operations to be performed. The four steps of the algorithm are as below [4].

### 2.1 SubBytes

Each entry in the state array is of bytes. S-box is a standard substitution table. Every byte in the state array is substituted by the corresponding byte from the S-box. Each byte of the state array is changed.

### 2.2 ShiftRows

Each row of the state array is rotated to the left by a specific indent. Second row is shifted by one position to left, third row by two position and fourth row by three positions to left.

### 2.3 Mixcolumns

In mixcolumn operation the columns of the State are considered as polynomials over GF ($2^8$) and are multiplied with a fixed polynomial. The mixcolumn is not used in the last round of the algorithm.

### 2.4 Addroundkey

It is a simple bitwise XOR of the current state array block with a portion of expanded key, expanded by key expansion block.

The flow of AES algorithm is very simple. For encryption, the cipher begins with an Addroundkey stage, followed by nine rounds. Each round includes all four stages, followed by tenth round of three stages. Only the Addroundkey stage makes use of key. For this reasons, the cipher begins and ends with an Addroundkey stage. Any other stage, used at the beginning or at the end, is reversible without knowing the key and hence would provide no security. The Figure 2 below shows the flow of the encryption algorithm [5].

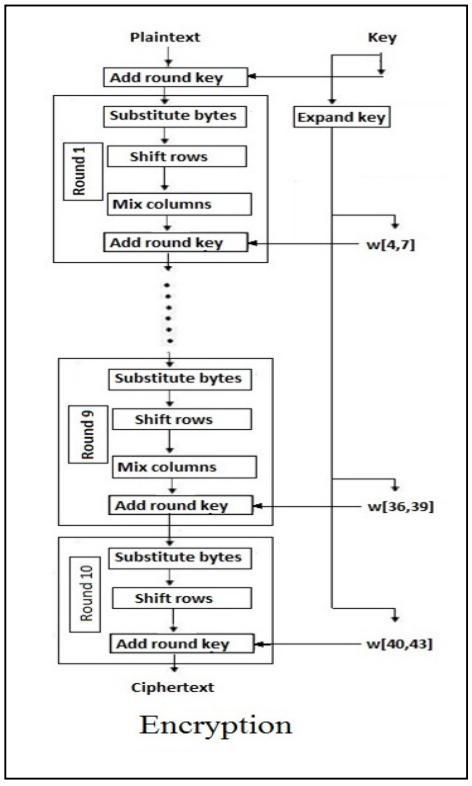


Fig.2 An AES Encryption Flow Chart

## 3. SYSTEM DESIGNED USING NIOS II PROCESSOR

A NIOS II is 32 bit soft core processor. A NIOS II processor system consists of a NIOS II processor, a set of on-chip peripherals, on-chip memory, GPIO's, all connected with Avalon bus to generate a system [6, 7]. The Nios II processor is a configurable soft-core processor, as

opposed to a fixed, off-the-shelf microcontroller. As Nios II processor is configurable adding and removing components or features on a system is easy to meets performance goals in terms of area. Soft-core processor can be targeted to any Altera FPGA family and is not fixed in silicon [8, 9, 10].

Following are the features of NIOS II processor

- The Nios II processor is a general-purpose RISC processor core, providing:
- Full 32-bit instruction set, data path, and address space
- 32 general-purpose registers
- 32 interrupt sources
- External interrupt controller interface for more interrupt sources
- Access to a variety of on-chip peripherals, and interfaces to off-chip memories and peripherals
- Single-instruction 32 × 32 multiply and divide producing a 32-bit result
- Optional memory protection unit (MPU)
- Instruction set architecture (ISA) compatible across all Nios II processor systems
- Software development environment based on the GNU C/C++ tool chain and NIOS IDE
- Performance up to 250 DMIPS

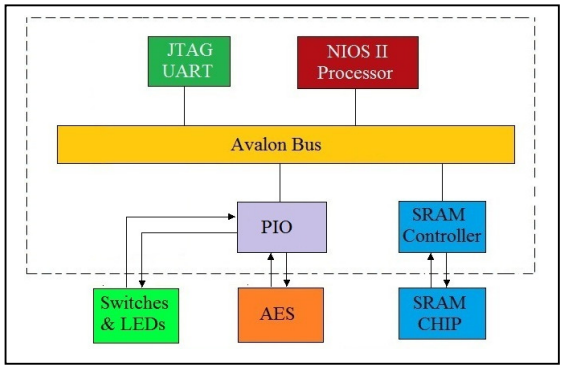


Fig.3 AES connected with NIOS II Processor through GPIO

## 4. DESIGN STEPS OF AES AS AN ACCELERATOR

The design steps are as follows:

### 4.1 Integrating the SOPC Builder System into the Quartus II Project

The AES is written in VHDL which is driven as an accelerator /custom hardware and the inputs and outputs are applied through GPIO in SOPC builder which is given through c-code in IDE. The contents are as follows

- Standard processor, component cores as a controller part.
- AES is given as hardware written in VHDL.
- Inputs and Outputs are given through GPIO.

Following are the components required in SOPC builder as a controller part of AES

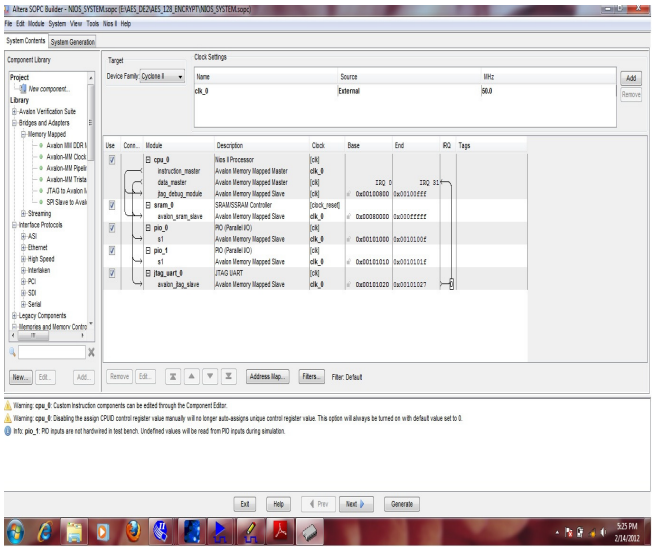

Fig.4 System generation using SOPC Builder

## 4.2 AES program written in hardware is connected with system designed through General Purpose Input / Output (GPIO)

VHDL program for AES algorithm is written and is connected with the system designed using the NIOS II processor through GPIO.

## 4.3 AES Block Diagram File view

Using the Quartus II software, all tasks are performed required to create the final FPGA hardware design. Using the Quartus II software, pin assignment, locations for I/O signals, timing requirements are specified, and also other design constraints are applied. Compile the Quartus II project to produce a **.sof file** to configure the FPGA.

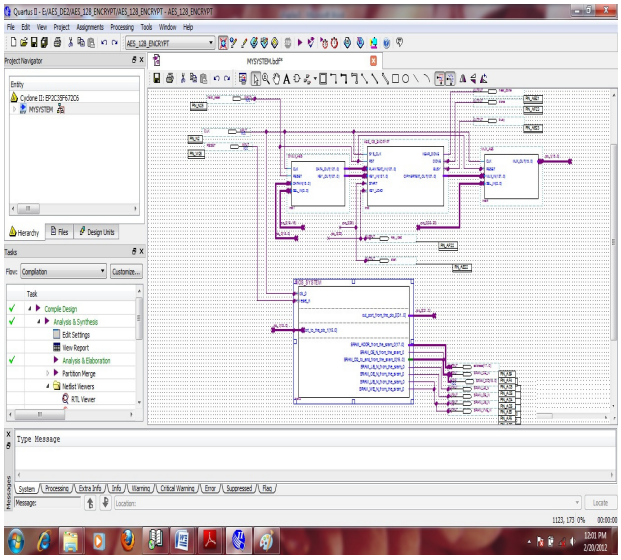

Fig.5 AES Block Diagram File view

### 4.4 Compiled and downloaded in FPGA

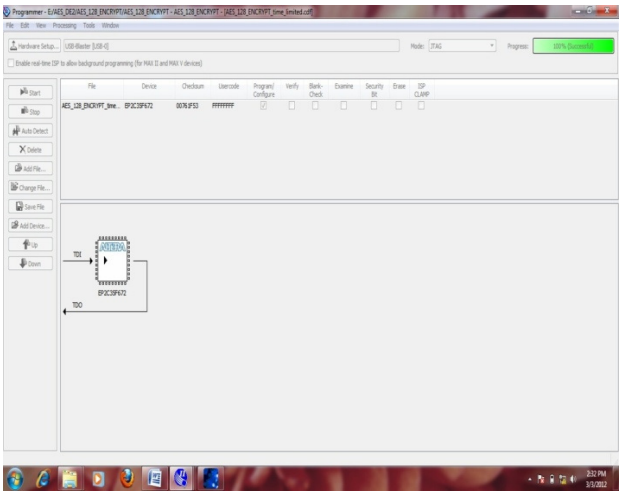

Fig. 6 Programmer Window

### 4.5 Control program written in Nios II IDE

The control part of the AES algorithm is written in C language in NIOS II IDE. The input and the output is given through C program written in IDE.

### 4.6 Encrypted result in console window

The inputs and outputs are applied through GPIO in NIOS II IDE which is written in C-code.

- To compile a Nios II project, right click the project in the Nios II C/C++ Projects view, and click Build Project.
- To run the program on a target board, right click the project in the Nios II C/C++ Projects view, click on Run As, and then click Nios II Hardware.

## 5. RESULT AND CONCLUSION

### 5.1 Simulation result of AES used as an accelerator

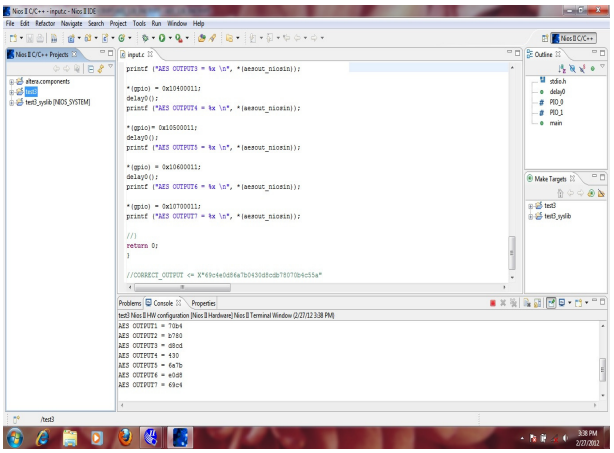

Fig.7 Encryption results on console window of NIOS II IDE

## 5.2 Synthesis Report of AES as an custom hardware

AES is synthesized in Quartus II software and the report is given below

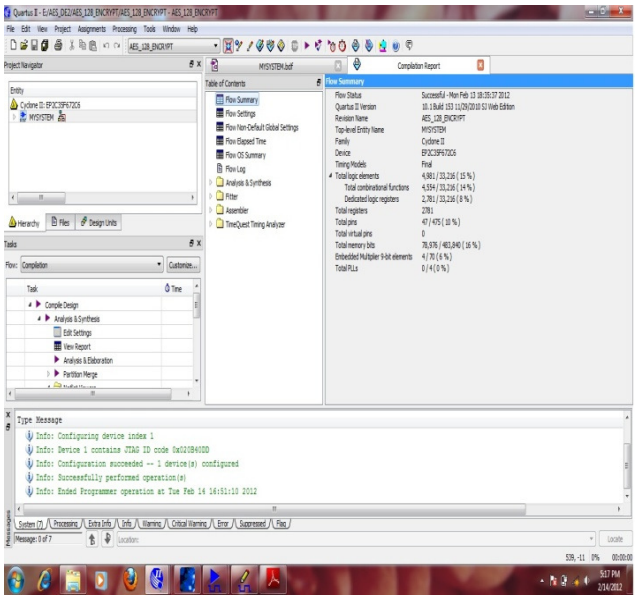

Fig.8 Synthesis Report of AES

## 5.3 RTL view of AES

AES is synthesized in Quartus II software and the RTL view is given below

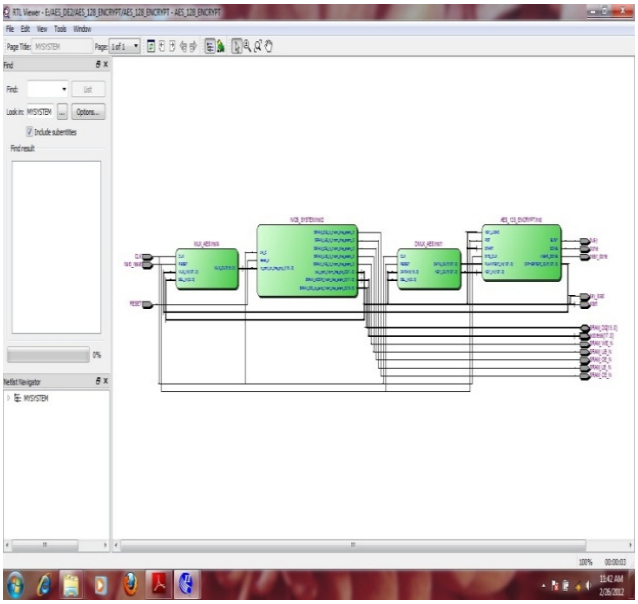

Fig.9 RTL view of AES

Table 1 below shows the CPU clock cycles taken for encryption when AES algorithm is completely written in software and when AES algorithm written in VHDL and connected with the system generated using NIOS II processor as a custom hardware.

Table 1. CPU clock cycles for encryption of data

| AES algorithm | | Total time for encryption (CPU cycles) |
|---|---|---|
| Hardware | Software | |
| | AES | 21731020 |
| AES as custom hardware | control, inputs and outputs | 26425 |

From the above table 1 number of clock cycles required for AES as custom hardware is very less as compared to AES in software. Hence AES as custom hardware accelerates the speed of operation.

## REFERENCES

[1] FIPS PUB 197, Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S. Department of Commerce, November 2001 (http://csrc.nist.gov/ publications/fips/fips197/fips-197.pdf).

[2] William Stallings "Cryptography and Network Security," 3rd Edition published by Pearson Education Inc and Dorling Kindersley Publishing Inc. Advanced Encryption Standard (AES), Nov. 26, 2001.

[3] Stallings W. "Cryptography and Network Security: Principles and Practices, 4th ed., Pearson Education, Inc. pp. 63-173., 2006.

[4] A. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar., An FPGA implementation and performance evaluation of theAES block cipher candidate algorithm finalist., presented at Proc. 3rd AES Conf. (AES3). [Online]. Available: http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html

[5] DAEMEN, J.—RIJMEN, V.: AES Proposal: Rijndael, The Rijndael Block Cipher, AES Proposal, pp.1–45,1999 (http://csrc.nist.gov/CryptoToolkit/aes/)

[6] Altera Corporation, "Quartus II Development Software Handbook v4.0," [Online Document],2004 February, Available HTTP: http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf

[7] Altera corporation, "Nios Embedded Processor, 32-Bit Programmer's Reference Manual" [Online Document] January, 2003
Available:http://www.altera.com/literature/manual/mnl_nios_programmers32.pdf

[8] Rahman T., Pan S. and Zhang Q., "Design of a High Throughput 128-bit (Rijndael BlockCipher)",Proceeding of International Multiconferrence of Engineers and computer scientists 2010 Vol II IMECS 2010, March 17- 19,2010, Hongkong.

[9] Nios II Hardware Development Tutorial, altera, December 2009 Altera Corporation Website, www.altera.com, June 2006

[10] Altera Corporation, "Nios Software Development Tutorial," [Online Document], 2003 July, [Cited 2004 March 1], Available HTTP: http://www.altera.com/literature/tt/tt_nios_sw.pdf

**Authors**

**Meghana A. Hasamnis** Pursuing Ph.D. from RTM Nagpur University, Nagpur, India in the field of embedded system. She is M.Tech. from VNIT, Nagpur, India. She is working as an Associate Professor at Shri. Ramdeobaba College of Engineering and Management Nagpur, India in the Department of Electronics Engineering for last 10 years.

**Priyanka Jambhulkar** Post graduate student in the Department of Electronics Engineering at Shri. Ramdeobaba College of Engineering and Management, Nagpur, India, pursuing her M. Tech in VLSI. Done her B.E. in Electronics and Communication from Nagpur University, India.

**Dr. S. S. Limaye** Ph.D. from Nagpur University in the faculty of Electronics Engineering. Currently he is working with JIT as a Principal. He has 38 years of experience in teaching as well as in industry. He also carried out a number of consultancy projects at DCM Data products, Delhi, PSI Data Systems Bangalore, Zen and Art New York. His area of specialization includes digital signal processing, VLSI design, LDPC codes, Embedded System, CORDIC algorithm. He is a recognized Supervisor RTM Nagpur University, Nagpur.