2012 International Conference on Advances in Computing and Communications

IMPLEMENTATION OF AES ALGORITHM IN UART MODULE FOR SECURED DATA TRANSFER

Debjani Basu , Dipak K Kole, Hafizur Rahaman

School of VLSI Technology
Bengal Engineering and Science University, Shibpur, India
Email: deb_basu_07@yahoo.co.in,dipak.kole@gmail.com,hafizur@vlsi.becs.ac.in

***Abstract*—This work proposes the application of Advanced Encryption Standard (AES) algorithm in Universal Asynchronous Receiver Transmitter (UART) module for secure transfer of data. The proposed architecture implements AES-128 algorithm that encrypts the data before transmission through UART transmitter and decrypts after receiving the data at UART receiver module. In this work, we present the AES-128 encryption and decryption circuit using iterative architecture. The design has a clock generator circuit which provides the different clock frequencies to different sub modules. The complete design is described in Verilog Hardware Description Language (HDL) and is functionally verified using Xilinx ISE 9.1i software. It takes *47.2µsec* to transmit 128 bit encrypted data and 36.7µsec to receive decrypted data on a Xilinx xc2vp70-7ff517 device. All the blocks of the proposed architecture are designed using FPGA technology.**

Keywords- UART, Transmitter, Receiver, AES, encryption, decryption, Shift Register, Field Programmable Gate Array (FPGA).

I. INTRODUCTION

The cryptography plays an important role in the security of data transmission. The increasing need for protecting data communication has led to development of several cryptography algorithms. The National Institute for Standard and Technology (NIST) has recommended the Rijndael block cipher algorithm as the new Advanced Encryption Standard (AES) in 2000 [1]. The AES algorithm has an SPN (Substitution Permutation Network) structure. Because of the growing requirements for high speed secure communications, the application of AES algorithm in UART (Universal Asynchronous Receiver Transmitter) module which is a widely used in serial data communication to support full-duplex serial communication is proposed here. The UART is an integrated circuit which handles the conversion between serial and parallel data [7-9]. In this proposed design each bytes of data per clock cycle is stored in a 128 bit shift register and then encrypted using AES encryption algorithm after that each encrypted bytes are shifted serially to the input of the UART transmitter. The UART transmitter takes bytes of data and transmits the individual bits in a sequential way. The reverse operation is performed in the UART receiver portion. The UART receiver re-assembles the bits into complete byte. These received bytes are serially stored in a 128 bit shift register for the AES decryption operation. The proposed UART module is operated without parity bit, eight data bit and one stop bit. Here we choose AES-128 algorithm for encryption and decryption operation. It takes 47.2µsec to store and then transmit the 128 bit encrypted data. The round transformations of AES-128 for both encryption and decryption are simulated using an iterative design approach in order to minimize the area.

The rest of the paper is organized as follows. Section II discusses the basic functionality of AES algorithm and UART. Section III describes the proposed architecture and its sub modules in details. Section IV explains the experimental results of the design. Finally section V concludes the paper.

II. PRELIMINARIES

AES Algorithm: The AES algorithm is an iterative algorithm composed of 10 rounds. After the initial secret key addition (roundkey (0)), the first 9 rounds are identical, with different the final round [10]. Each of the first 9 rounds consists of 4 transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey. The final round excludes the MixColumns transformation. The above encryption scheme can be inverted to get a decryption structure. 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* is constructed by composing two transformations: multiplicative inverse in the finite field $GF(2^8)$ and affine transformation.

UART: A Universal Asynchronous Receiver Transmitter (UART) is a circuit that sends parallel data through a serial line. A UART includes a transmitter and a receiver. So, the main function of a UART is the conversion of parallel-to-serial when transmitting and serial- to- parallel when receiving. The transmitter is essentially a special shift register that loads data in parallel and then shifts it out bit by bit at a specific rate. The receiver, on the other hand, shifts in data bit by bit and then reassembles the data. The serial line is 1 when it is idle. The transmission starts with a start bit, which is 0, followed by data bits and an optional parity bit, and ends with stop bits which are 1.

III. PROPOSED FPGA BASED ARCHITECTURE

A detailed description of the proposed combined architecture for UART module with AES crypto circuit is explained in this section. The design consists of five main units; the first unit is Encryption Function unit which loads

978-0-7695-4723-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ICACC.2012.32

the eight bit words per clock cycle in 128 bit serial in parallel out shift register and then encrypt the data using AES-128 encryption algorithm, after that serially transmit the eight bit encrypted word per clock cycle to the input of the UART_Tx module through 128 bit parallel in 8 bit serial out shift register, the second unit is UART_Tx module that frames the eight bit word coming from AES-128 encryption unit with a START bit (logic '0') at the beginning, and a STOP bit (logic 'l') at the end of the word and sends the framing information in a serial manner from the Least Significant Bit (LSB) to the Most Significant Bit (MSB).The third main module is UART_Rx module which performs the reverse operation of UART_Tx module. It removes the START bit and STOP bit and collects the data in its output port as eight bit word format. Decryption Function unit, the forth unit performs the decryption operation and sends the decrypted data to the output of the UART module. Fifth unit is clock_generator module which takes two main signals (system clock and reset) as its input and generates the new clock signals. All the five basic units of the proposed architecture are shown in Fig. 1. All modules are designed using Verilog Hardware Description Language (HDL). Details of every module of the proposed UART architecture are described below.

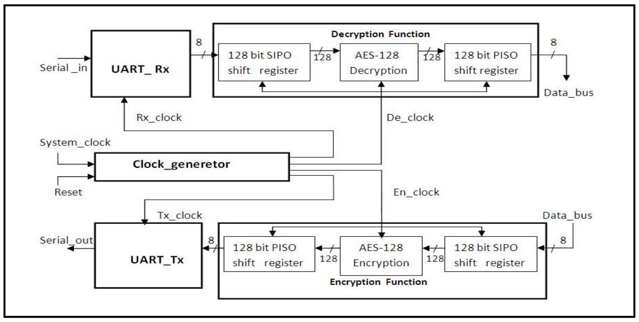

Figure 1. Proposed UART architecture

A. *Encryption Function Module*

Encryption Function Module consists of three sub units, two 128 bit shift register and AES encryption module. The first serial in parallel out shift register stores the data bytes in each clock cycle and sends them to the AES-128 Encryption module for encryption. The 128 bit encrypted data is transmitted to the UART transmitter through parallel in serial out shift register. This AES Encryption module perform the encryption operation on 128bit of data using AES algorithm which is a symmetric block cipher that processes data blocks of 128 bits using three different cipher key lengths 128, 192 and 256 bits. The proposed design supports the AES-128 Encryption.

In encryption procedure after an initial round key addition, a round function consisting of four different transformations – Sub Bytes, Shift Rows, Mix Columns and Add Round Keys are applied to the 128 bit data block as shown in the Fig. 2. This single round function is performed iteratively 10 times. In the last round of encryption the round function does not contain the Mix Column transformation.

SubBytes transformation replaces each byte of a state by its substitute in an s-box. The s-box is an invertible substitution table which is constructed by a composition of two transformations [2, 3]:

1. First, each byte is replaced with its reciprocal in *GF* (2^8) (except that0, which has no reciprocal, is replaced by itself).
2. Then, an affine transformation is applied.

ShiftRows transformation cyclically shifted left the bytes in the last three rows over different offsets (offset value is respective of the row it is applied, i.e. the offset of row 1 is 1, the offset of row 2 is 2, etc).

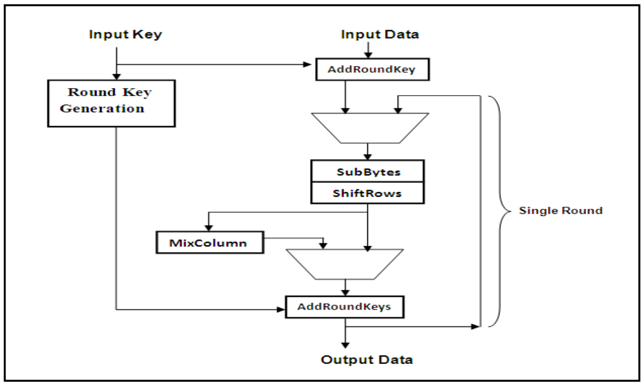

Figure 2. Iterative Architecture of AES Encryption

MixColumns transformation operates on the State column-by-column, treating each column as a four-term polynomial [11]. The columns are considered as polynomials over GF (2^8) and multiplied modulo $x^4 + 1$ with a fixed polynomial $a(x)$, given by

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

AddRoundKey transformation is a simple bitwise XOR operation of 128-bit state with a 128-bit round key.

Round Key Generation unit extended the initial 128 bit input key to 10 round keys of same length. Round keys are produced when a process is applied to the previously generated round key. Suppose the process is key expansion then it may be defined as

$$\text{Round Key}_i = \text{Key Expansion (Round Key}_{i+1}) \quad \text{for } 1 \leq i \leq 10;$$

Round Key_0 is the input key.

B. *UART Transmitter Module*

UART transmitter module ie UART_Tx that frames the eight bit word coming from AES-128 encryption unit with a START bit (logic '0') at the beginning, and a STOP bit (logic 'l') at the end of the word and sends the framing information in a serial manner from the Least Significant Bit (LSB) to the Most Significant Bit (MSB). The architecture of the transmitter will consist of a controller, a data register (XMT_datareg), a data shift register (XMT_shftreg) and a status register (bit_count) to count the bits that are transmitted. Load_XMT_datareg signal is asserted to indicate that XMT_datareg now contains the data_bus value and that is now transferred to the internal shift register that is XMT_shftreg shown in the above Fig. 3.

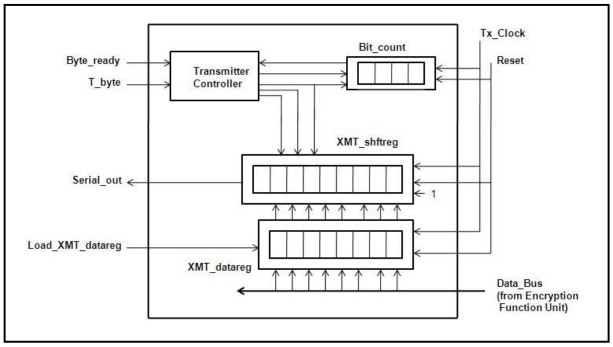

Figure 3. Block diagram of UART Transmitter

C. *Decryption Function Module*

Decryption Function Module consists of three sub units, two 128 bit shift register and one AES decryption module. It performs the operation in the same manner like encryption function module, instead the AES-128 encryption operation it performed the AES-128 decryption operation on the received data from UART_Rx module. The individual transformations performed in the decryption process are Inv Sub Bytes, Inv Shift Rows, Inv Mix Columns and Add Round keys shown in the Fig. 4. These transformations are the inverse of the corresponding transformations in the encryption process. More details concerning the AES-128 decryption process may be found in [1].

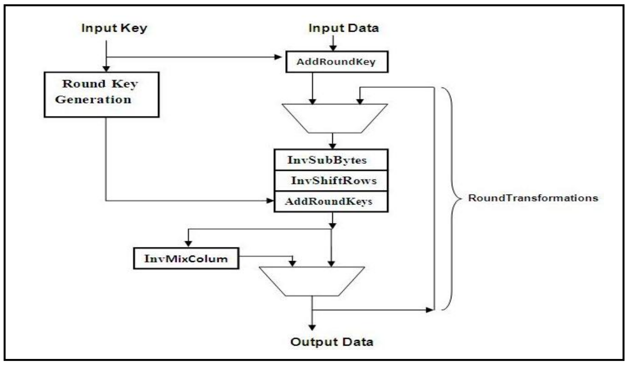

Figure 4. Iterative Architecture of AES Decryption

D. *UART Receiver Module*

The UART receiver module that is UART_Rx performed the task of receiving the serial bit stream of data, removing the start bit and transferring the data in a parallel format to a storage register connected to the host data bus. The data arrives at a standard bit rate, but it is not necessarily synchronized with the internal clock at the host of the receiver, and the transmitter's clock is not available to the receiver. This issue of synchronization is resolved by generating a local clock (Rx_clock) at a higher frequency and using it to sample the received data in a manner that preserves the integrity of the data. The UART receiver circuit includes RCV_shftreg (the shift register receiving Serial_in), RCV_datareg (the 8- bit register holding the received word) and Receiver controller to control the UART receiver circuit as shown in the Fig. 5.

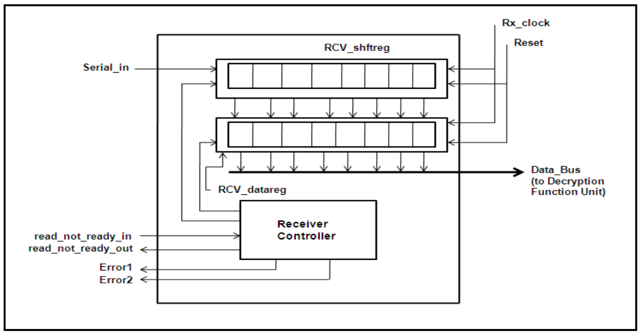

Figure 5. Block diagram of UART Receiver

E. *Clock Generator Module*

The rate at which the data is transmitted is known as baud rate. UART receiver module (UART_Rx) operates on the frequency which is 8 times higher than transmitter. Clock generator module provides the four clock signals for UART transmitter, UART receiver, and encryption and decryption unit to maintain data integrity between the four different modules. In our design clock generator module uses 100MHz clock frequency. We chose 12500000 baud rate for our design

IV. EXPERIMENTAL RESULTS

The proposed UART design and its sub modules are implemented using Verilog Hardware Description Language [5-6] in Xilinx ISE 9.1i. To minimize the hardware implementation iterative looping technique is used for AES encryption and decryption algorithm. The device utilization summary of the complete design and the AES-128 encryption and decryption algorithm for the targeted FPGA device xc2vp70-7ff1517 are shown in Table 1.

TABLE I. Device Utilization Summary

Architecture	FPGA Device	No. of Slices	No. of LUT's	No. of IOBs
AES-128 Encryption	xc2vp70-7ff1517	3988	7551	387
AES-128 Decryption	xc2vp70-7ff1517	3992	7551	387
Propose UART module	xc2vp70-7ff1517	2748	5221	38

The simulation result of full AES-128 encryption module and decryption module are shown in Fig. 6 and Fig. 7 where input vectors and input keys are given from NIST standard publication [1] and outputs are verified.

The encrypted data is obtained after 12 clock cycle and in decryption process decrypted plain data is obtained after 21 clock cycle.

Simulation waveform of the proposed UART module is described in Fig. 8. The UART transmitter starts the transmission only after it received the full 128 bit encrypted data. The designed architecture transmits the first bit after 30.9µsec and receives the first decrypted byte after 23.9µsec.

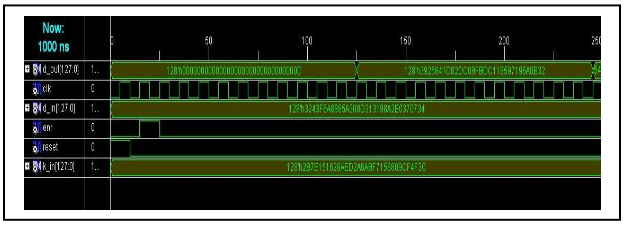
Now:
1000 ns
0
50
100
150
200
250
d_out[127:0]
1...
128'h00000000000000000000000000000000
128'h3925841D02DC09FBDC118597196A0B32
54
clk
0
d_in[127:0]
1...
128'h3243F6A8885A308D313198A2E0370734
enr
0
reset
0
k_in[127:0]
1...
128'h2B7E151628AED2A6ABF7158809CF4F3C

Figure 6. Functional simulation of AES-128 encryption module

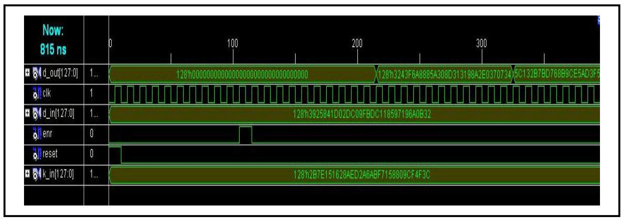
Now:
815 ns
0
100
200
300
d_out[127:0]
1...
128'h00000000000000000000000000000000
128'h3243F6A8885A308D313198A2E0370734
5C132B7BD768B9CE5AD3F5
clk
1
d_in[127:0]
1...
128'h3925841D02DC09FBDC118597196A0B32
enr
0
reset
0
k_in[127:0]
1...
128'h2B7E151628AED2A6ABF7158809CF4F3C

Figure 7. Functional simulation of AES-128 decryption module

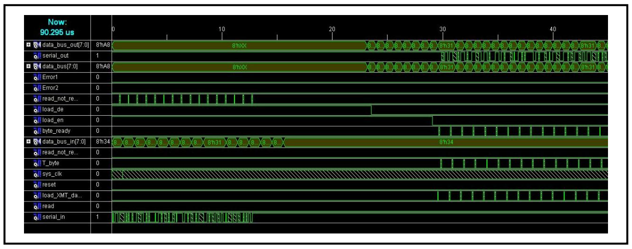

Figure 8. Functional simulation waveform of proposed UART architecture

V. CONCLUTIONS

This work has presented an efficient of AES algorithm in UART architecture. This architecture provides secure data transfer through UART. This new approach may provide a satisfactory level of security for serial communication through UART. It uses only 2748 slices using a Virtex – II Pro xc2vp70-7target device [4].

REFERENCES

[1] NIST, *Advanced Encryption Standard (AES)*, FIPS PUBS 197, National Institute of Standards and Technology, November 2001.

[2] P.V.S.Shastly, Anuja Agnihotri, Divya Kachhwaha, Jayasmita Singh and Dr.M.S.Sutaone, "A Combinational Logic Implementation of S-box of AES", *IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS),* 2011, pp. 1-4

[3] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer – Verlag, 2002.

[4] Xilinx's Virtex- II Pro Platform FPGA's: Complete Datasheet.

[5] Z. Navabi, *Verilog Digital System Design*, McGraw- Hill.

[6] Pong P. Chu," FPGA Prototyping by Verilog Examples", WILEY.

[7] He Chun-zhi, Xia Yin-shui, Wang Lun-yao, *"A Universal Asynchronous Receiver Transmitter Design", ICECC*, 2011, pp. 691-694.

[8] J. Norhuzaimin and H.H Maimun*,"*The Design of High Speed UART"*, Asia-Pasific Conference on Applied Electromagnetic Proceedings,* 2005, pp. 306-310.

[9] Mohd Yamani Idna Idris, Mashkuri Yaacob, Zaidi Razak, "A VHDL IMPLEMENTATION OF UART DESIGN WITH BIST CAPABILITY*", Malaysian Journal of Computer Science, Vol. 19 (1), 2006, pp. 73 – 86*

[10] H. Rahaman, J. Mathew, and D. K. Pradhan, "Secure Testable S-box Architecture for Cryptographic Hardware Implementation", *The Computer Journal, OXFORD University Press,* 2010, 53(5), pp. 581-591.

[11] Hua Li Zac Friggstad, "An Efficient Architecture for the AES Mix Columns Operation", *IEEE International Symposium on Circuits and Systems*, 2005, pp. 4637- 4640