OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

MILITARY APPLICATION FOR SECURE COMMUNICATION

Nitin Jogdand, Amit Gawali, Kalpesh Bhale, Atul Mane, Ramesh Patole

Student, B. Tech, I.T. Department, G H Raisoni College of Engineering and Management Pune, India.[1]
Asst.Prof, I.T. Department, G H Raisoni College of Engineering and Management Pune, India.[2]
nitinjogdand1995@gmail.com,Shree9195@gmail.com,bhale.kalpesh@gmail.com
atulmanecom9@gmail.com, ramesh.patole@raisoni.net

Abstract: One of the major problems in communication is the secure transportation of data over communication protocols. This paper presents a feasible resolution for Rijindael's encryption and decryption using VHDL for FPGA (cyclone III) &'C' running over Nios II processor. The Nios II is a versatile embedded processor which is high performance, of lower cost and power consumption, has low complexity combining several functions into one FPGA. This paper shows implementation of AES algorithm for 128 bit data and 128 bit key in RTL (VHDL) and its software implementation using C. To measure performance in same system that is same hardware Nios II soft core processor is used. Hence this paper shows application of Advance Encryption Standard (AES) algorithm in UART for secure transfer of data in software and hardware platforms which are (RTL)VHDL and 'C' and further to decide suitability of its implementation of specific platform(software or hardware) depending on different baud rates supported by UART. The results are compared with the help of tools modelsim (Quartus II) and Nios II 10.1 software build

Tools for eclipse

Keywords- AES,VHDL, FPGA ,RTL

I INTRODUCTION

There are millions of users who everyday generate and interchange large volumes of information in various fields. For example medical reports, bank services via Internet. All these applications require a special treatment from the security point of view. So here comes the need of implementation of cryptography techniques which are applicable. For secure data transmission cryptography always plays important role. The need for protecting data communication led to development of several cryptographic 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 Substitution Permutation Network structure. As there are increasing requirements for high speed secure communications, the application of AES algorithm in embedded system is required. In embedded system UART (Universal Asynchronous Receiver Transmitter) is widely used in serial data communication to support full-duplex serial communication. The UART is an integrated circuit which handles the conversion between serial and parallel data [2-4]. UART can be interfaced with various data exchange modules. To provide extra level of security during data exchange over UART, data can be encrypted using AES algorithm. All steps of AES-128 for all its 10 rounds of both encryption and decryption are simulated. With the speedy development and a huge applications range in computer and communication networks, the information security has aroused high attention. So the presently known attacks can be avoided by AES. In this paper Hardware and Software implementation is proposed using FPGA (Field Programmable Gate Array) and Nios II processor respectively. FPGA offers one or additional soft core processor implementations just like the Nios II processor developed by ALTERA. Nios II is a 32 bit fixed point processor that has separate buses for information and program memory that is usually referred to as Harvard design. It has Reduced Instruction Set Computer (RISC) design. It has thirty two general purpose registers, and 6 control registers used to manage the processor status for an extended time.

II. LITERATURE SURVEY

DES is a symmetric key algorithm for encryption of the data, to secure from attacker or unauthorized party. DES provides the influence in security world to protect the information. National bureau of standards firstly adopt the Data Encryption Standard (DES) in year 1997, nowadays it is known as Federal Information Processing Standards. 56 bit key is used in DES, and it transforms 64 bit input block into a 64 bit output block. The key actually looks like a 64 bit quantity, but one bit in each of the 8 octets is used for odd parity on each octet. There are many attacks and methods recorded those revealed the weaknesses of DES, which made it an insecure block cipher. The basic concept of DES algorithm is as shown below.

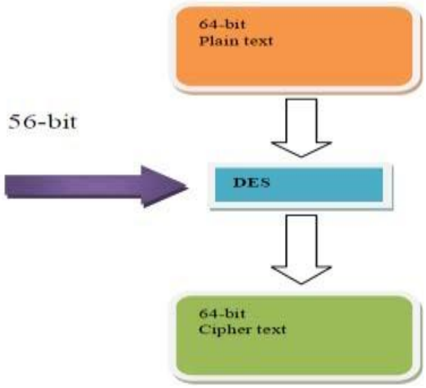


Figure 1: Basic concept of DES

In August 2000, NIST mentioned 5 algorithms along with Rijndael as the final competitors. All those algorithms were subjected to additional analysis to choose one of the simplest algorithm for the AES. Finally on October 2, 2000, office declared that the Rijndael algorithmic rule was the winner. The Rijindael algorithm becomes the new Advanced Encryption Standard (AES) suggested by the US National Institute of Standards and Technology (NIST) [1]. Rijindael's algorithm is chosen as AES because of factors like efficiency, security, performance each in hardware and software system platforms. It is a block cipher algorithm that encrypts blocks of 128, 192 or 256 bits. Therefore, it becomes harder to break the key.

Description Of AES Algorithm:

AES algorithm explanation with encryption and decryption processes is as follows:

A. AES encryption

The AES algorithm operates on a 128-bit block of data and executed for Nr - 1 loop times. A loop is called as a round and the number of iterations of a loop, Nr can be 10, 12, or 14 which depends the key length. The key length can be 128, 192 or 256 bits in length respect. The first and last rounds differ from other rounds because in that there is an additional AddRoundKey transformation at the start of the 1st round and also MixCoulmn operation is not performed in the last round. In this paper, the key length of 128 bits (AES-128) is used.

1.SubBytes Transformation:

In SubBytes transformation, substitution of bytes take place independently. The SubBytes transformation is carried out using an already calculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values.

2.ShiftRows Transformation:

In ShiftRows transformation, the rows of the state are Shifted in cylindrical manner to the left. Row 0 which is 1st row is not shifted and remains as it is. row 1 which is 2nd row is shifted one byte to the left; row 2 which is 3rd is shifted two bytes to the left and row 3 which is last is shifted three bytes to the left.

3.MixColumns Transformation: In MixColumn transformation, Each and every column is considered as polynomials and multiplied by modulo x^4 + 1 with a fixed polynomial c(x).

This operation is carried over each column.

4.AddRoundKey Transformation: In this transformation, the 128 bit data is XORed with the sub key of the current round using the key expansion operation. The add round key is used at two different points one during the start that is initial round which is noted as r=0 and then during the other rounds that is when 1 ≤ round≤ Nr, where Nr is the maximum number of rounds. The Round Key of each round is derived from the main key using the Key Expansion algorithm[1].

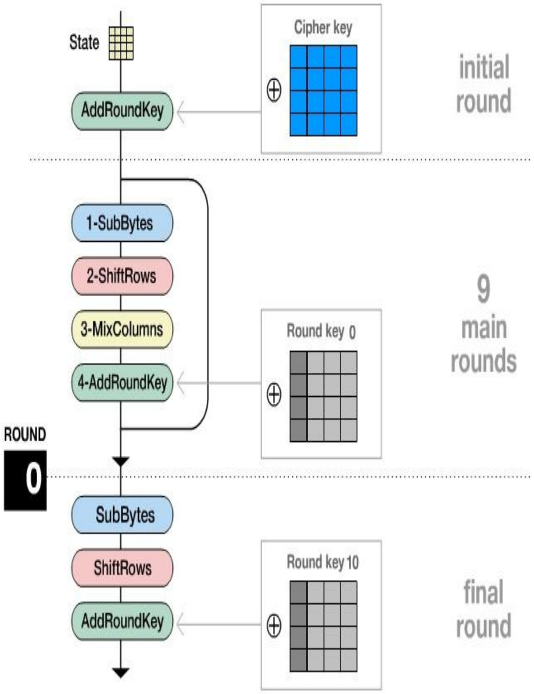


Figure 2 : AES Encryption Process

B. AES decryption

Decryption is a simple reverse process of encryption. Inverse round transformations are performed in decryption process to get original plaintext. The round transformation of decryption uses the following functions:

1. AddRoundKey

2. InvMixColumns,

3. InvShiftRows, and

4. InvSubBytes successively.

1. AddRoundKey:

This operation is its own inverse, since it only involves an application of the XOR operation.

2. InvShiftRows Transformation: InvShiftRows function is the same as ShiftRows, only in the opposite direction. The 1st row is not shifted, while the 2nd , 3rd and 4th rows are shifted right by one, two and three bytes respectively.

3.InvSubBytes transformation:

The InvSubBytes transformation is done using an already calculated substitution table called InvS-box.

That InvS-box table contains 256 numbers (from 0 to 255) and their corresponding values.

III. PROPOSED METHOD

This paper deals with an FPGA implementation of AES algorithm interfacing with the Nios II processor.

A. Nios II processor-

This processor is provided by ALTERA and implemented in FPGA. The FPGA is capable to do hardware modification, it also offers the possibility of microprocessor implementations, which can be programmed in Assembly or C. Altera's Nios II processor has that kind of flexibility to achieve the required performance for your embedded design. Also, Nios II processors help you to avoid last-minute hand-tuning of assembly language code, giving you configurable performance features that can be designed in as needed. Altera FPGAs and Nios II processors give you the best and feasible performance features, as well as many options for reducing risk in embedded. The NIOS II Processor appeases flexibility such as selecting the exact set of CPUs, peripherals, and interfaces, accelerating only relevant functions and eliminates the risk of processor being out of date. The focus of this paper is the use of this processor to control and test the peripherals on the board.

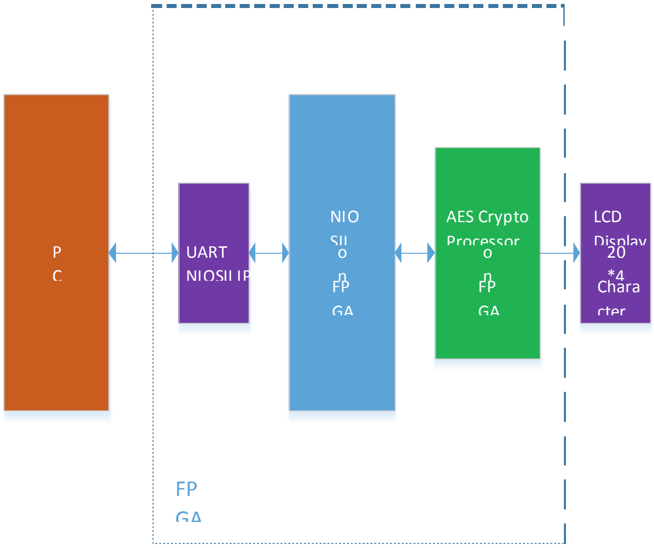


Figure 3: Proposed Method

In this paper I have implemented AES in RTL (VHDL) and software implementation using C. To measure performance in same system i.e. same hardware I have used NIOSII soft core processor. This processor IP is synthesizable and configurable for Altera FPGA. ALTERA FPGA is used for implementation of this project. NIOSII is synthesizable soft core processor available from ALTERA. NIOSII is available as configurable and customizable IP. As per design requirements NIOSII can be modified. AES Crypto Processor is implemented using VHDL. It can encrypt or decrypt the 128bit data. This AES implementation is interfaced with NIOSII processor. NIOSII processor feeds data to AES processor. As per requirement NIOSII sends commands for encryption or decryption. Encryption and decryption results are displayed on LCD display. This display is controlled by AES crypto processor. From PC over UART 16 byte (128bit) data (plain text or cipher text) is transferred. Also encryption and decryption commands NIOSII instructs AES crypto processor to perform requested operation. With this hardware design (NIOSII processor + AES RTL Engine) two things are demonstrated.

1. AES software implementation and its performance measurement. Here performance is measured in terms of number of clock cycles required for CPU to complete encryption/decryption. 2. AES hardware implementation and its operation. AES VHDL performance can be measured from simulation or by doing static timing analysis. This can be manually computed as design is synchronous. Operations are triggered by clock. In simulation number of clock cycles can be counted for single round of AES and then scaled by number of rounds performed in AES.

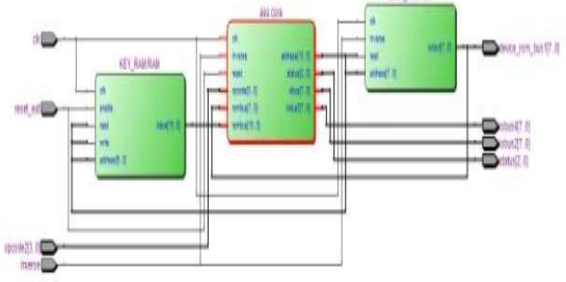


Figure 4: RTL Diagram

IV. IMPLEMENTATION RESULTS

The AES algorithm's one of the implementation is made using VHDL & simulated using ModelSim- Altera 6.6c Edition. Another implementation is in 'C' language and compiled using Nios II 10.1 software build tools for Eclipse . Then the results are analyzed with the help of both tools. The algorithm is simulated for functional implementation, analysis in terms of no. of clock cycles and hence time required for its execution.

A. VHDL Simulation results-

AES-128 bit algorithm encryption simulation result which includes all basic operations.

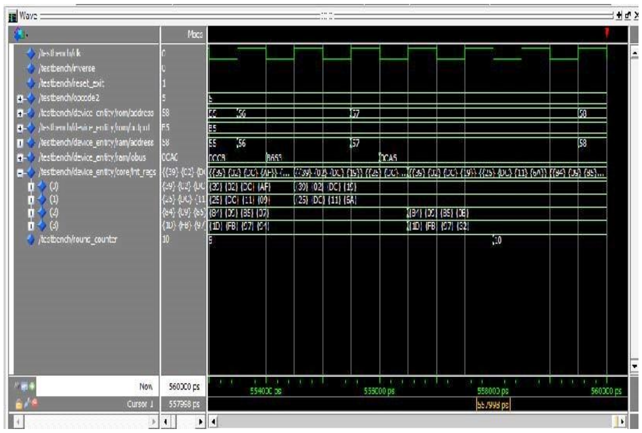
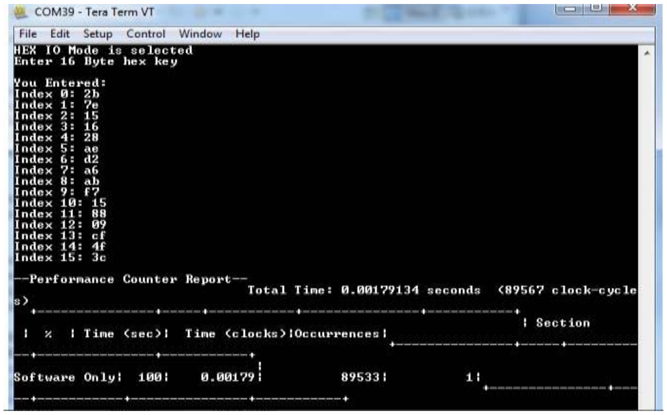


Figure 5: AES-128 bit algorithm Encryption simulation result

B. AES-128 bit algorithm encryption and decryption simulation result (Software Performance)

NIOSII supports hardware performance measurement counters. These counters are clocked with CPU clock. These counters can be started or stopped from software. In the demonstration one of such counter is started just before start of AES software implementation and stopped immediately after software computation. The value of counters is displayed in terminal. With these values hardware performance can be compared to conclude results.



```
HEX IO Mode is selected
Enter 16 Byte hex key

You Entered:
Index 0: 2b
Index 1: 7e
Index 2: 15
Index 3: 16
Index 4: 28
Index 5: ae
Index 6: d2
Index 7: a6
Index 8: ab
Index 9: f7
Index 10: 15
Index 11: 88
Index 12: 09
Index 13: cf
Index 14: 4f
Index 15: 3c

--Performance Counter Report--
                              Total Time: 0.00179134 seconds  (89567 clock-cycle
s)
  +---------------+-----+-----------+---------------+-----------+
                                                                | Section
 |  %  | Time (sec)|  Time (clocks)|Occurrences|
                                                 +---------------+-----+---------
--+---------------+-----------+
                                 |
Software Only|  100|    0.00179|          89533|        1|
                                                              +---------------+---
--+-----------+---------------+-----------+
```

Figure 6: AES-128 bit algorithm software Encryption/ decryption simulation result

The results are tabulated below to show the timing analysis of AES-128 for both the platforms. Clock of 20nsec is used for software performance and the clock of 1nsec is used for VHDL(Hardware performance)

TABLE **1:** TIME FOR AES-128 FOR 2 PLATFORMS.

Platforms	Time for encryption/ decryption	No. of clock cycles required
'C'	1792120 ns	89606
'VHDL'	558 ns	558

As UART works on different baud rates which are as follows:

110, 300, 600, 1200, 2400, 4800, 9600, 14400,19200, 38400, 57600, 115200, 230400, 460800 and 921600 , the threshold value of baud rate is defined for either software implementation and hardware implementation.

C. Explanation for defining the Threshold value of baud rate to decide specific implementation of AES algorithm for UART as follows:

Threshold value of Baud rate =115200bps.

1. Software encryption and decryption time required for above baud rate-

89567 clock cycles required 16 bytes encryption and decryption @ 50MHz freq.(20nsec.) For 16 bytes- 89567 clk cycles

For 160 bytes- 89567 X 10 clk cycles

Time required for 160 bytes = 89567 X10 X 20nsec.

= 17.91msec.

2. Time required transferring data through UART- 11520 bytes transferred in 1 sec.

Hence 160 bytes transferred in 13.88msec. (160 bytes are considered as AES is block cipher encryption so we should consider No. in multiples of 16 so its 16X10= 160).

3. If baud rate is 115200bps. and above then time required for encryption and decryption is more as compared to time required to transfer data through UART in that case hardware implementation of AES is required.
4. If baud rate is below 115200bps. then software implementation of AES can be preferred as time required for encryption and decryption for software is less as compared to time required to transfer data through UART.

TABLE 2**:** BAUD RATE VERSUS TIME REQUIRED TO TRANSFER DATA OVER UART

Baud rate	Time required to transfer 160 bytes over UART
9600	166msec.
19200	83.33msec.
38400	41.66msec.
57600	27.77msec.
115200	**13.78msec.**

TABLE 3**:** APPLICATIONS FOR DIFFERENT BAUD RATES OF UART

High speed (baud rate= 115200 and bove)	Low speed(baud rate is below 115200)
Synchronization of telecommunication router in telecommunication	Sensor data communication
Mobile computing in WAN	Magnetic card reader
Simultaneous Control of system which provides factory automation	2G GSM Modem

V. CONCLUSION

Hardware and software implementations of AES algorithm in UART is carried out for data security. Further analysis is done with respect to no. of clock cycles required and timing required for its execution. According to different baud rates supported by UART, either implementation of software or hardware is suggested for the same by defining the threshold baud rate. These numbers can vary based on CPU speed. If CPU speed is higher than specified in paper then software implementation can be preferred over hardware implementation and vice versa.

REFERENCES

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

[2] Asynchronous Receiver Transmitter Design", ICECC, 2011, pp. 691- 694.

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

[4] 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.