ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 1, January 2015

VHDL implementation of AES-128 on FPGA

Nitin R. Chavan[1], Suresh A.Annadate[2]

Assistant professor, Electronics and Telecommunication, S.T.B Collage of engineering, Tuljapur, India[1]

Associate professor, Electronics and Telecommunication, J N E Collage of engineering, Aurangabad, India[2]

Abstract*:* The importance of cryptography applied to security in electronic data transactions has acquired an essential relevance during the last few years. A proposed FPGA-based implementation of the Advanced Encryption Standard (AES) algorithm is presented in this paper. The design has been coded by Very high speed integrated circuit Hardware Descriptive Language. All the results are synthesized and simulated using Xilinx ISE and ModelSim software respectively. This implementation is compared with other works to show the efficiency. The design uses an iterative looping approach with block and key size of 128 bits, lookup table implementation of S-box. This gives low complexity architecture and easily achieves low latency as well as high throughput. Simulation results, performance results are presented and compared with previous reported designs.

Keywords*:* AES, FPGA, encryption decryption, block cipher and VHDL.

I. INTRODUCTION

Each day millions of users generate and interchange large volumes of information in various fields, such as financial and legal files, medical reports, and bank services via internet. These and other examples of applications deserve a special treatment from the security point of view, not only in the transport of such information but also in its storage. In this sense, cryptography techniques are especially applicable. For a long time, the Data Encryption Standard (DES) was considered as a standard for the symmetric key encryption. DES has a key length of 56 bits. However, this key length is currently considered small and can easily be broken. For this reason, the National Institute of Standards and Technology (NIST) opened a formal call for algorithms in September 1997. A group of fifteen AES candidate algorithms were announced in August 1998. Next, algorithms were subject to assessment process performed by various groups of cryptographic researchers all over the world. In August 2000, NIST selected five algorithms: Mars, RC6, Rijndael, Serpent and Two fish as the final competitors. These algorithms were subject to further analysis prior to the selection of the best algorithm for the AES. Finally, on October 2, 2000, NIST announced that the Rijndael algorithm was the winner. Field Programmable Gate Arrays (FPGAs) are hardware devices whose function is not fixed which can be programmed in system. The potential advantage of encryption algorithm implemented in FPGAs includes: Algorithm agility- This term refers to the switching of cryptographic algorithm during operation. Algorithm upload- It is perceivable that fielded devices upgraded with new encryption algorithm which did not exist at design time. Algorithm modification- There are applications which require modification of standardized algorithm Architecture efficiency- With FPGAs it is possible to design and optimize architecture for specific parameter set. Throughput- Although typically slower than ASIC implementation, FPGA potential of running substantially faster than software implementations. Cost efficiency- Time and cost for developing an FPGA have implementation of a given algorithm are much lower than for an ASIC implementation. In cryptography, the AES is also known as Rijndael. AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits. This paper deals with an FPGA implementation of an AES encryptor/decryptor using an iterative looping approach with block and key size of 128 bits. This method gives very low complexity architecture and is easily operated to achieve low latency as well as high throughput.

II. RELATED WORK

One of the design criteria for AES candidate algorithms is that they can be efficiently implemented in both hardware and software. In reference [2], the technical analysis used in determining which of the potential Advanced Encryption Standard candidates was selected as the Advanced Encryption Algorithm includes efficiency testing of both hardware and software implementations of candidate algorithms. Reprogrammable devices such as field-programmable gate arrays (FPGAs) are highly attractive options for hardware implementations of encryption algorithms, as they provide physical security, and potentially much higher performance than software solutions. A strong focus is placed on high-throughput implementations. Finally, the implementations of each algorithm compared in an effort to determine the most suitable candidate for hardware implementation within commercially available FPGAs. Serpent algorithm was found to yield the best results when operating in nonfeedback mode with throughput 5035.0Mbps, while Rijndael algorithm was found to yield best results when operating in feedback mode with throughput 300.1Mbps [2]. For each of the AES finalists, an implementation analysis for each architecture option when optimized for both area and speed was performed to determine the suitability for hardware implementation of each finalist. The principle of AES algorithm and the detailed description and implementation on FPGA. This system aims at reduced hardware structure [3]. And this system has high security and reliability. The advantage of this design is the fact that we do not need to store the round

DOI 10.17148/IJIREEICE.2015.3108

key since they are currently calculated, in accordance that AES algorithm is used in the low requirements of the terminal throughput at present, the high safety and cost effective reduced AES system is designed and validated on the Altera Cyclone EP2C35F672C6 chip, aiming at reduced hardware structure. From the test and synthesis results, this system has the significant features such as less hardware resources, high speed, high reliability, high cost effective. Furthermore, this system can be widely used in the terminal equipments which less demand on the throughput. Throughput found for encryption decryption is 593.45Mbps and 267.63Mbps respectively [3]. The design uses an iterative looping approach with block and key size of 128 bits, lookup table implementation of S-box [1]. This gives low complexity architecture and easily achieves low latency as well as high throughput. The algorithm achieves a low latency and throughput reaches value of 1054 Mbps for encryption and 615Mbps for decryption. Latency of encryption is only 13 clock cycles and latency of decryption is 25 clock cycles [1].

III. PROPOSED WORK

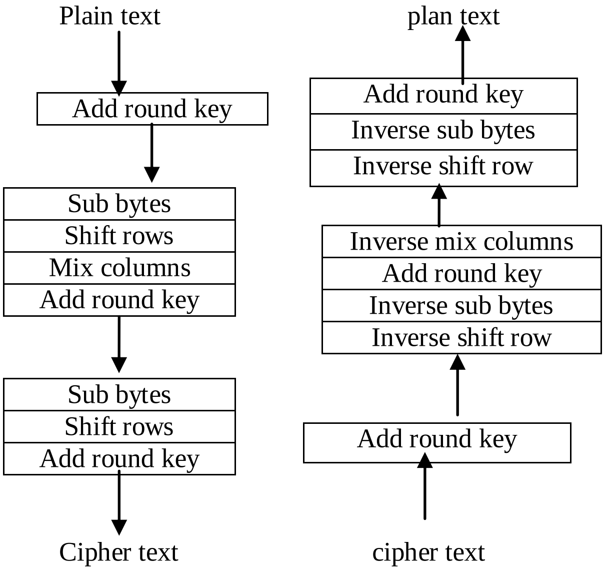

Fig. 1(a) Encryption process (b) decryption process

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to an unintelligible form called cipher-text. Encryption of the cipher-text converts the data back into its original form, which is called plain-text.

A. *AES encryption*

The AES algorithm operates on a 128-bit block of data and executed Nr - 1 loop times. The number of rounds depends on the length of the key used for the encryption process. The Advanced Encryption Standard can be programmed in software or built with pure hardware [8]. The key length is 128, 192 or 256 bits in length respectively.

TABLE I
KEY BLOCK ROUND COMBINATION

Block size (Nb words)= 4		
Bit Mode	**Key Length (Nk words)**	**Number of Rounds (Nr)**
128	4	10
192	6	12
256	8	14

The first and last rounds differ from other rounds in that there is an additional AddRoundKey transformation at the beginning of the first round and no MixCoulmns transformation is performed in the last round. In this paper, we use the key length of 128 bits (AES-128) as a model for general explanation. An outline of AES encryption is given in Fig. 1.a)

SubBytes Transformation:
The SubBytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The SubBytes transformation is done using a once-precalculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. This approach has the significant advantage of performing the S-box computation in a single clock cycle, thus reducing the latency and avoids complexity of hardware implementation.

ShiftRows Transformation:
In ShiftRows transformation, the rows of the state are cyclically left shifted over different offsets. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.

MixColumns Transformation:
In MixColumns transformation, the columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial c(x), given by: c(x)={03}x3 + {01}x2 + {01}x + {02}.

AddRoundKey Transformation:
In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the MixColumns transformation - by a simple bitwise XOR operation. The Round Key of each round is derived from the main key using the Key Expansion algorithm. The encryption/ decryption algorithm needs eleven 128-bit Round Key, which are denoted Round Key[0] Round Key[10].

B. *AES decryption*

Decryption is a reverse of encryption which inverse round transformations to computes out the original plaintext of an encrypted cipher-text in reverse order shown in fig.1.b). The round transformation of decryption uses the functions AddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes successively.

AddRoundKey:

AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order.

InvShiftRows Transformation:

InvShiftRows exactly functions the same as ShiftRows, only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.

InvSubBytes transformation:

The InvSubBytes transformation is done using a once precalculated substitution table called InvS-box. That InvS-box table contains 256 numbers (from 0 to 255) and their corresponding values.

InvMixColumns Transformation:

The InvMixColumns transformation is done using polynomials of degree less than 4 over GF(28), which coefficients are the elements in the columns of the state, are multiplied modulo (x4 + 1) by a fixed polynomial $d(x) = \{0B\}x3 + \{0D\}x2 + \{09\}x + \{0E\}$, where {0*B*}, {0*D*}; {09}, {0*E*} denote hexadecimal values.

C. Design Flow Chart

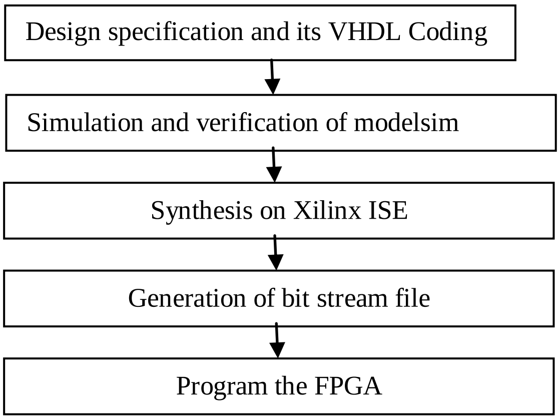
Design specification and its VHDL Coding
Simulation and verification of modelsim
Synthesis on Xilinx ISE
Generation of bit stream file
Program the FPGA

Fig.2. Shows design of project flow. From design specification, design will be coded using Very High Speed Integrated Circuit Hardware Descriptive Language. Simulation and verification will be done on ModelSim software. Results are then synthesized on Xilinx ISE. Generated Bit stream file will need to program the FPGA.

IV.SIMULATION AND RESULTS

VHDL is used as the hardware description language because of the flexibility to exchange among environments. The software used for this work is Xilinx ISE and the waveforms are simulated with the help of model sim simulator. This is used for writing, debugging, simulating and checking the performance results using the simulation tools available on Xilinx ISE. The delay is calculated with three different Device families [7]. The delay have been generated as result is shown in TABLE II. As different Delay calculations by virtex 2.

TABLE II.
Delay and Frequency of Different Modules

Modules	**Delays(ns)**	**Frequency(MHz)**
Sub Byte	10.69	93.54
Shift Rows	7.158	139.70
Mix Column	8.168	122.428
Key Expansion	13.475	74.211

IV. CONCLUSION

The AES algorithm can be efficiently implemented by software. Software implementations cost the smallest resources, but they offer a limited physical security and the slowest process. Besides, growing requirements for high speed, high volume secure communications combined with physical security, hardware implementation of cryptography takes place. We are going to design FPGA based hardware implementation of AES algorithm. The design is coded using VHDL language. The design is simulated on ModelSim software and synthesize on Xilinx ISE software. Performance parameters of design will be calculated in terms of throughput and latency. Results will be compared with previous work

REFERENCES

[1] Hoang Trang and Nguyen Van Loi HoChiMinh City, VietNam- "*An Efficient FPGA implementation of the Advanced Encryption Standard Algorithm*" (IEEE 2012)

[2] AdamJ.Elbirt, W. Yip, B. Chetwynd, and C. Paar- "*An FPGA Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists*" (IEEE 2001).

[3] WANG Wei, CHEN Jie & XU Fei, China-"*An Implementation of AES Algorithm Based on FPGA*" (IEEE2012).

[4] Yang Jun Ding Jun Li Na Guo Yixiong School of Information Science And Engineering, Yunnan University Kunming, China - "*FPGA based design and implementation of reduced AES algorithm*"(IEEE 2010).

[5] Nalini C, Nagaraj, Dr. Anandmohan P.V, & Poornaiah D.V, V.D.kulkarni -"*An FPGA Based Performance Analysis Pipelining and Unrolling of AES Algorithm*" (IEEE2006).

[6] Tessier. R. and Burleson W-"Reconfigurable computing for digital signal Processing: a survey", *J.VLSI Signal Process*, 2001, 28.

[7] Chih-Peng Fanand and Jun-Kui Hwang "*FPGA Implementations Of High Throughput Sequential And Fully Pipelined AES Algorithm*" International journal of Electrical Engineering, vol.15, no.6, pp447-455, 2008

[8] Saambhavi Baskaran and Pachamuthu Rajalakshmi "*Hardware Software Co-Design of AES on FPGA*" *ICACCI '12, ACM* August 2012.

DOI 10.17148/IJIREEICE.2015.3108