International Journal of Engineering Research
Volume No.3, Issue No.12, pp : 780-785
ISSN:2319-6890)(online),2347-5013(print)
01 Dec. 2014

Implementation of Multi Mode AES Algorithm Using Verilog

P Penchala Reddy, Dr.V.Thrimurthulu, K. Jagadeesh Kumar
Dept.of ECE, CREC, Tirupathi, A.P, India
[1]penchalreddy1020@gmail.com, [2]vtmurthy.v@gmail.com, [3]jagadish.kasula@gmail.com

Abstract— Increasing need of high security in communication led to the development of several cryptographic algorithms hence sending data securely over a transmission link is critically important in many applications. NIST in the beginning selected Rijndael within October 2000 and formal adoption as being the AES standard started in December 2001. FIPS PUB 197 explains a 128-bit block cipher making Ause of a 128, 192, or 256-bit key. In cryptography, modes of operation enable the repeated and secure use of a block cipher under a single key. This paper presents implementation of multi mode AES algorithm with three modes ECB, CBC and CTR modes. All these three modes are implemented with 128-bit plain text and 128 bit, 192 bit and 256 bit key lengths. Each program results are verified with ModelSim PE and are synthesized in Xilinx ISE 9.2i. These results are also useful for implementing hardware.

Index Terms—Cryptography; Rijndael, Mode selection logic, Key expansion block; Cipher block; Decipher block.

1. INTRODUCTION

Cryptography is the science of information and communication security. Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. It uses a cryptographic system to transform a plaintext into a cipher text, using most of the time a key.

The Advanced Encryption Standard, in the following referenced[1] as AES, is the winner of the contest, held in 1997 by the US Government, after the Data Encryption Standard was found too weak because of its small key size and the technological advancements in processor power. Fifteen candidates were accepted in 1998 and based on public comments the pool was reduced to five finalists in 1999. In October 2000, one of these five algorithms was selected as the forthcoming standard: a slightly modified version of the Rijndael.

There are many architecture proposals for AES Rijndael algorithm [1], but many of them are poor in terms of area and speed. This paper proposes a different approach to increase speed by utilizing lesser resources available in FPGA. This paper is structured as follows: Section 2 describes the existing AES algorithm and Section 3 describes the proposed method. The result and conclusion are described in Section 4 and 5 respectively.

2. EXISTING AES ALGORITHM

AES use Rijndael algorithm [2] by Joan Daeman and Vicent Rijimen for both encryption and decryption. The AES cryptography algorithm is capable of encrypting and decrypting 128 bit data using cipher keys of 128, 196 or 256 bits (AES128, AES196 and AES256) [3].

The AES is a computer security standard from NIST intended for protecting electronic data. Federal Information Processing Standards (FIPS) Publication 197 gives the specification of AES[1].

Rijndael encryption consist of four operations

1. Key addition
2. Substitution
3. Shift Row
4. Mix Column

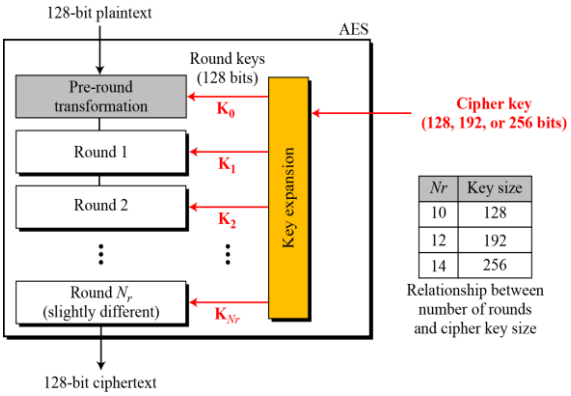

Fig. 1: Algorithm for AES Encryption

2.1 Key addition

2.1.1 Add round key

State is represented as follows (16 bytes):

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

Add round key (state, key):

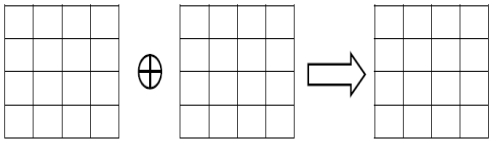

Fig. 2: Add round key

2.2. Substitution

2.2.1 Sub bytes transformation:

Bytes are transformed using a non linear s-box

ISSN:2319-6890)(online),2347-5013(print)
01 Dec. 2014

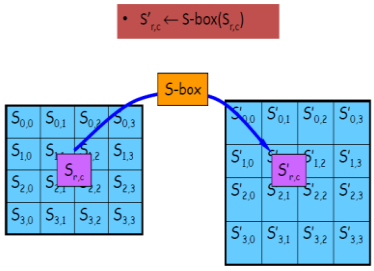

Fig. 3: Sub bytes transformation

Byte substitution using a non-linear (but invertible) S-Box (independently on each byte). S-box is represented as a 16x16 array, rows and columns indexed by hexadecimal bits. 8 bytes replaced as follows: 8 bytes define a hexadecimal number rc, then Sr,c = binary(S-box(r, c))

2.3. Shift rows:

Circular Left Shift of a number of bytes equal to the row number

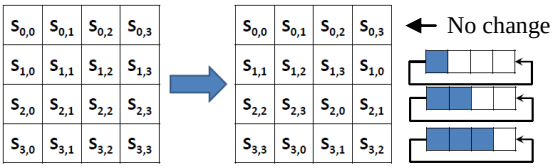

Fig4: Shift rows

2.4. Mix column transformation:

Bytes in columns are combined linearly. Interpret each column as a vector of length 4. Each column of State is replaced by another column obtained by multiplying that column with a matrix in a particular field (Galois Field).

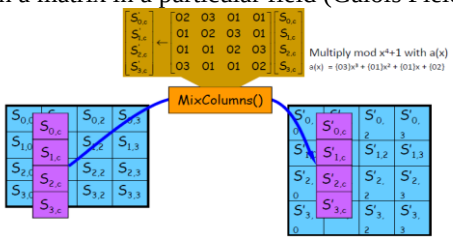

Fig. 5: mix column transformation

The Rijandael decryption consists of four inverse operations of encryption which are compliment functions of encryption. They are

1. Inverse Substitution
2. Inverse Shift Row
3. Inverse Mix Column
4. Key addition

3. PROPOSED METHOD

The proposed method, multi mode AES consists of four parts

1. Mode selection logic
2. Key expansion block
3. Cipher block
4. Decipher block.

Fig. 6 shows Multi mode AES block diagram which has Key expansion, Mode selection logic, and Encryption and Decryption blocks.

3.1. Key Expand Function:

Key expansion block functionality is to expand the key for the given key data length depending on the key select length. Start_key_exp is used to start Key expansion for the given key length once its expansion done the key data is used for encryption and decryption block for to encrypt data or decrypt the data respectively.

Encryption block is to encrypt the data of block size 128 bit depending on the mode selection logic. Mode selection logic is to select one of the modes out of three they are ECB (Electronic codebook), CBC (Cipher Block Chain) and CTR (Counter mode).

3.2 Modes Operation:

In cryptography, modes of operation enable the repeated and secure use of a block cipher under a single key. A block cipher by itself allows encryption only of a single data block of the cipher's block length. When targeting a variable-length message, the data must first be partitioned into separate cipher blocks. Typically, the last block must also be extended to match the cipher's block length using a suitable padding scheme. A mode of operation describes the process of encrypting each of these blocks, and generally uses randomization based on an additional input value, often called an initialization vector, to allow doing so safely.

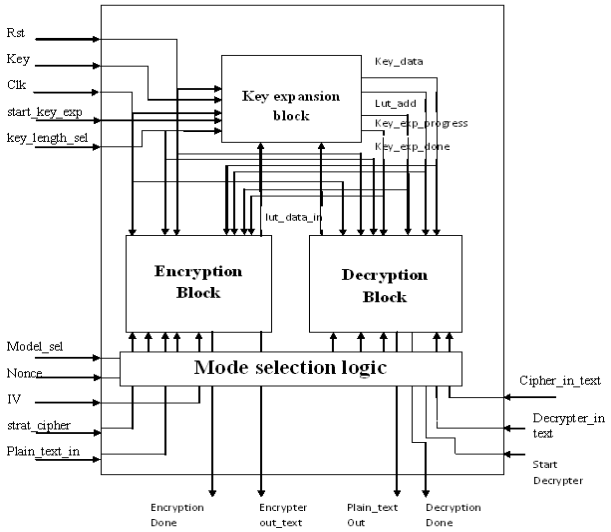

Fig. 6: Block Diagram of Multimode AES

Modes of operation have primarily been defined for encryption and authentication. Historically, encryption modes have been studied extensively in regard to their error propagation properties under various scenarios of data modification. Later development regarded integrity protection as an entirely separate cryptographic goal from encryption. Some modern modes of operation combine encryption and authentication in an efficient way, and are known as authenticated encryption modes.

Mode selection logic is used select which mode is of operation is to perform. Here we are used ECB, CBC and CTR mode encryption. Each encryption is has its own applications. Mode 00, a mode 01 and mode 1x selection activates ECB, CBC and CTR respectively for encryption and decryption.

3.3. Mode selection logic

ISSN:2319-6890)(online),2347-5013(print)
01 Dec. 2014

3.3.1 Encryption mode selection description

As shown in figure 7 encryption mode selection logic which has three modes of operation they are (ECB, CBC and CTR) in that for mode 00(Electronic code book mode) during encryption process plaintext is directly applied to encryption block and the output is directly connected to the output. The width of the plain text is 128 bit

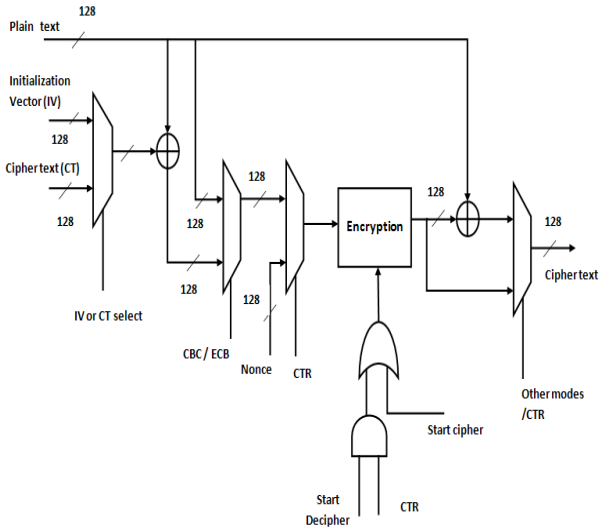

Fig. 7: Encryption mode selections

Mode 01 is (Cipher block chaining mode). In this mode initialization vector (IV) is XORed with plain text as input to the encryption block and the output cipher text is again given back as IV to the next iteration of encryption.

Mode 1x is (Counter mode) which as nonce of 120bit is concatenated with the counter of 8 bit as input to the encryption block and the output cipher text is XORed with the plain text, from the next iteration count value is incremented and concatenated with the same nonce value to input of encryption block. Encryption output is XORed with plain text.

In CTR mode both encryption and decryption is done with the same encryption block. During decryption process start decipher and CTR mode is selected then the same encryption block used by giving data to the same encryption block as shown in figure 7.

3.3.2. Decryption mode selection description

In decryption mode selection logic for mode 00(ECB) cipher text of 128 bit data is applied to the decryption block and the output of that decryption block is directly connected to the output pin in ECB mode as shown in figure 8

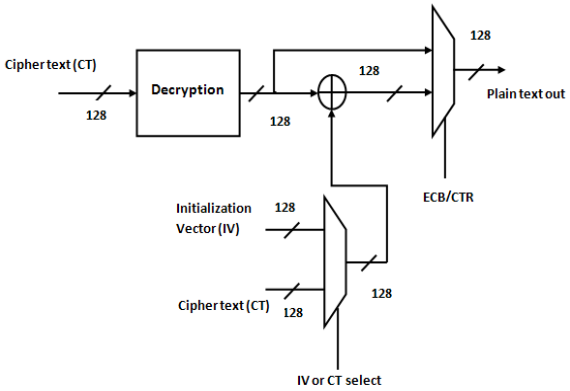

Fig. 8: Decipher mode selection logic

For mode 01 (CBC) cipher text is given to decryption block the output from the block is XORed with IV to get the plain text out. From the next iteration previous cipher in text is given as IV for getting the plain text out.

3.4. Key Expansion block description

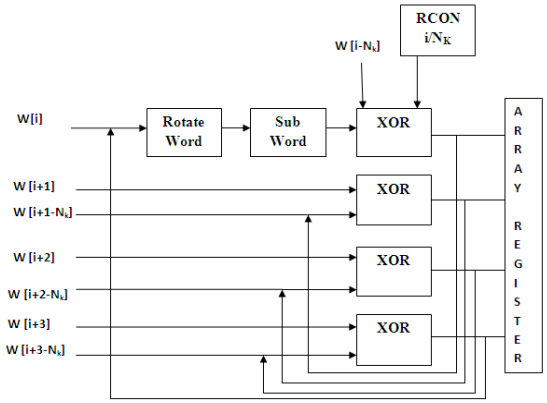

Fig. 9: Key expansion block diagram

The AES algorithm takes the Cipher Key, ***K***, and performs a Key Expansion routine to generate a key schedule. The Key Expansion block generates a total of ***Nb***(***Nr*** + 1) words: the algorithm requires an initial set of ***Nb*** words, and each of the ***Nr*** rounds requires ***Nb*** words of key data. The resulting key schedule consists of a linear array of 4-byte words, denoted [*wi*], with *i* in the range 0< = *i* < ***Nb***(***Nr*** + 1).

In each round of key expansion it has to pass through rotate, sub word, or with RCON and XOR with W[i-Nk].Each round final value is stored in array register to start the next iteration of the key expansion.

3.5. Cipher Block description

Cipher block contain add round, shift rows, sub bytes and mix columns rounds. In encryption it has total Nr number of rounds. During the first round input data is XORed with key data of size 128, then the result data is applied through an iterative process of shift rows, SubBytes, mix columns and XOR with key data till number of rounds equal to Nr-1. During the last round mix column step is skipped.

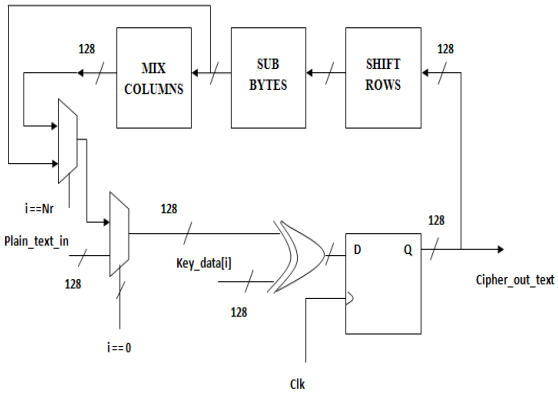

Fig. 10: Cipher block

3.5. Decipher block description

ISSN:2319-6890)(online),2347-5013(print)
01 Dec. 2014

The individual transformations used in the Decipher are Inv Shift Rows (), Inv Sub Bytes (), Inv Mix Columns (), and Add Round Key ().

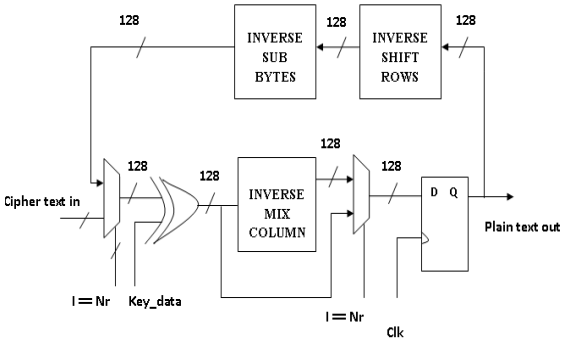

Fig. 11: Decipher block

During decryption process first round cipher in text XORed with key data when number of rounds equal to Nr, second round onwards till Nr –1 rounds Inv Shift Rows (), Inv Sub Bytes (), Add Round Key () and Inv Mix Columns () as shown in figure below. Flip flops are used to store 128 bit data in each round; stored data is used in each start of round. During the final round mix column step is skipped.

During inverse cipher the number of rounds is same as cipher block rounds but it is not exactly reverse of it. In inverse cipher after sub bytes key data is XORed with sub byte output then it is given to inverse mix column.

4. SIMULATION RESULTS

Advanced Encryption Standard (AES) is a symmetric key cipher technique used to secure and encrypt operating systems, hard drives, networking systems, files, e-mails, and other similar data. In cryptography, AES consist of three block ciphers taken from a larger collection published originally as Rijndael. Each cipher has a 128-bit block size with three different key sizes of 128, 192, and 256 bits. After expansion of Key length 128 or 192 or 256 bits stored in the memory.

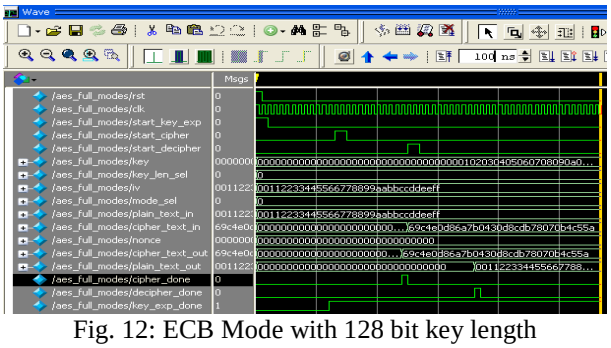

Fig. 12: ECB Mode with 128 bit key length

ECB(Electronic code book) Mode is a normal mode encryption it has plain text as a input with 128 bit key length which makes the normal encryption which is not much stronger compare to 192 and 256 bit key expanded encryption because every time for the same input of data we get same encrypted data as the output.

Initially design is reset; by making reset low key expansion is started. After key expansion done cipher (encryption) process is start by asserting the start_cipher as shown in figure 12 after 11 rounds of iterations in encryption block cipher_done become high, decipher is asserted by giving cipher_text_in to the decryptor to get back the plain text out (original data).

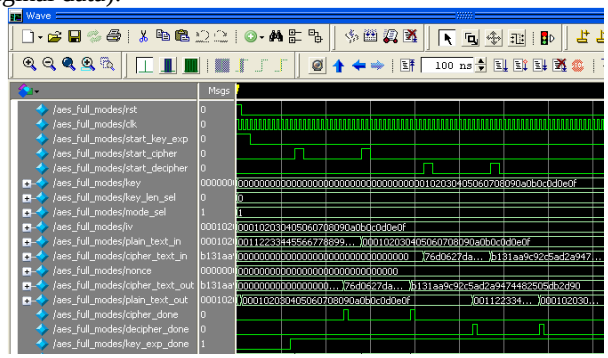

Fig. 13: CBC Mode with 128 bit key length

The Figure 13 shows simulation results of CBC encryption and decryption process with 128 bit key length. In CBC mode encryption for the same given plain text input output cipher text differs, with. In CBC mode input for encrypter (cipher) is XORed output of Initialization vector (IV) and plaintext. During simulation initially reset is equal to one which makes the device to come to initial state. By making reset low start key expansion. After two or three clock cycles make the start expansion pin to low, run the simulation till the key expansion is done to one, start cipher after key expansion is done and keep it high for two to three clock cycles and make it low, run the simulation till cipher is done. In the next encryption cipher takes the plain text and previous cipher output to encrypt the plain text instead of IV. Repeat the above process for series of encryption.

In decryption respective cipher outputs are given as input to the decrypted. In the same as encryption decryption process takes first XORed IV and cipher as input next iteration the previous cipher is taken as input instead of IV. It repeats the same for the next iterations.

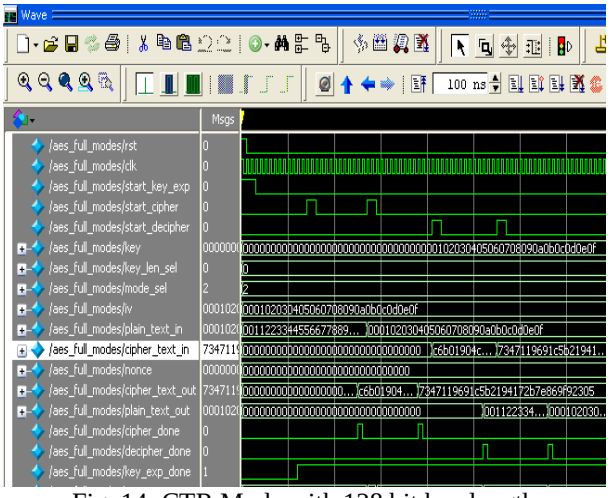

Fig. 14: CTR Mode with 128 bit key length

ISSN:2319-6890)(online),2347-5013(print)
01 Dec. 2014

Counter Mode Encryption has nonce and count value as input it dose encryption for nonce and count, the block cipher output is then XORed with plain text to get the cipher out text. Next succeeding encryptions incremented count value is append with user defined nonce as an input for the encryption block. In Decipher same block cipher is used for decryption. As stated above in an encryption process the same process is repeated except at the output stage XOR with cipher text is done with the block cipher output to get the plain text as the output.

CTR mode encryption is stronger because of every iteration takes the next cipher out data as IV. In the figure 14 shows the both encryption and decryption for the CTR mode encryption and decryption with 128 bit key length data.

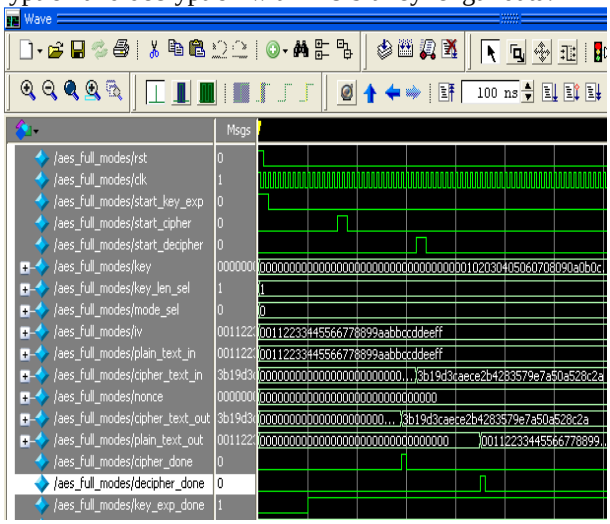

Fig. 15: ECB Mode with 192 bit key length

Figure 15 show ECB mode with 192 bit key length encryption process is same as ECB with 128 bit key shown in 11 in this 192 bit key length no of rounds increased by 2 from 11 to provide sufficient key data for 13 rounds from 11 rounds of encryption.

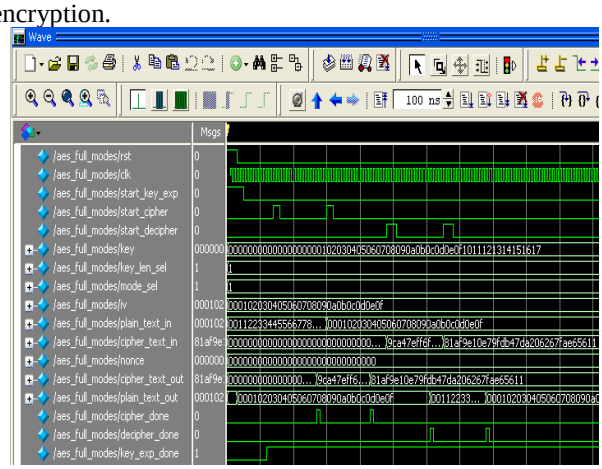

Fig. 16: CBC Mode with 192 bit key length

Figure 16 show CBC mode with 192 bit key length encryption process is same as CBC with 128 bit key shown in 12 in this 192 bit key length no of rounds increased by 2 from 11 to provide sufficient key data for 13 rounds from 11 rounds of encryption.

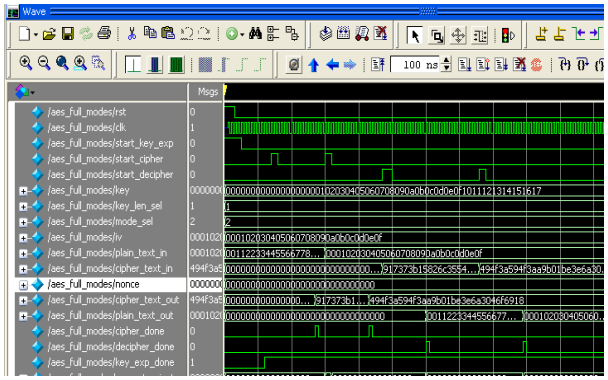

Fig. 17 CTR Mode with 192 bit key length

Figure 17 show CBC mode with 192 bit key length encryption process is same as CBC with 128 bit key shown in 12 in this 192 bit key length no of rounds increased by 2 from 11 to provide sufficient key data for 13 rounds from 11 rounds of encryption.

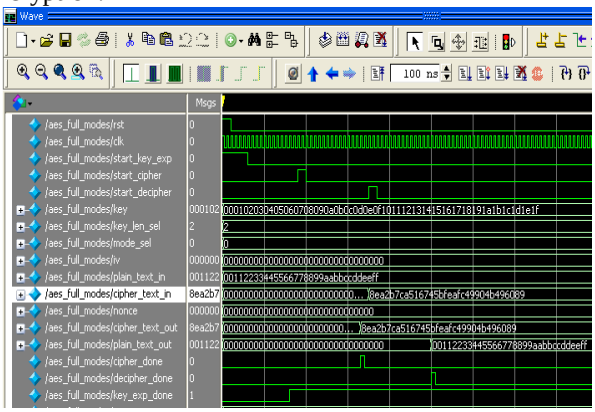

Fig. 18: ECB Mode with 256 bit key length

ECB mode with 256 bit key length as shown in figure 18 makes even strong encryption than the 192, but the process is same as shown in figure 14 except the number rounds in key expansion, encryption as well as decryption increased to 2.

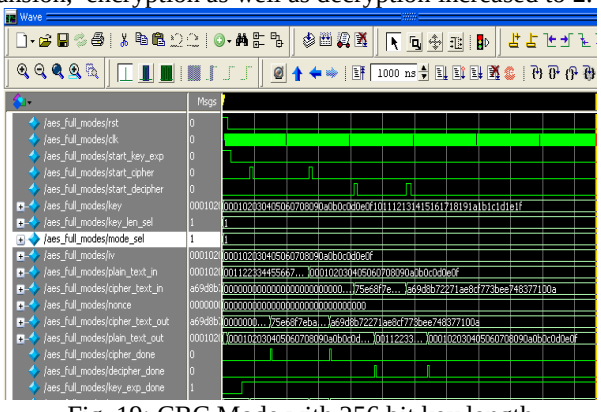

Fig. 19: CBC Mode with 256 bit key length

CBC mode with 256 bit key length as shown in figure 19 makes even strong encryption than the 192, but the process is same as shown in figure 15 except the number rounds in key expansion, encryption as well as decryption process increased by 2. Each round has 128 bit data.

ISSN:2319-6890)(online),2347-5013(print)
01 Dec. 2014

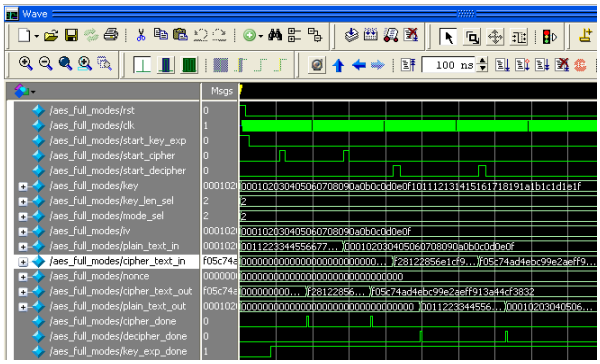

Fig. 20: CTR Mode with 256 bit key length

CTR mode with 256 bit key length as shown in figure 20 makes even strong encryption than the 192, but the process is same as shown in figure 16 except the number rounds in key expansion, encryption as well as decryption process increased by 2. Each round has 128 bit data.

5. CONCLUSION

The combination of a simple, portable and efficient AES cryptographic algorithm implemented in Verilog source code provides an excellent platform for high security applications. A synthesizable Verilog code is developed for the implementation of both encryption and decryption process with different modes. Each program results are verified with ModelSim PE and are synthesized in Xilinx ISE 9.2i. These results are also useful for implementing hardware.

References

i. FIPS-197, NIST - National Institute of Standards and Technology," Announcing the ADVANCED ENCRYPTION STANDARD (AES)," http://csrc.nist.gov/publications/fips/fips197/fips-97.pdf, 2001.

ii. H. Trang and N.V. Loi, "An efficient FPGA implementation of the Advanced Encryption Standard algorithm," IEEE Int. Conf. on Computing and Communication Technologies, Research, Innovation and Vision for the Future (RIVF), 2012, pp. 1-4.

iii. W. Stallings, "Cryptography and network security principles and practice," Pearson edition 2009, pp. 135-160.

iv. An FPGA Implementation of 30Gbps Security Module for GPON Systems BY Truong Quang Vinh1, Ju-Hyun Park1, Young-Chul Kim1, Kwang-Ok Kim2 2008 IEEE.

v. FPGA Implementation of AES Encryption and Decryption by Ashwini M. Deshpande, Mangesh S. Deshpande and Devendra N. Kayatanavar, International Conference On "Control, Automation, Communication And Energy Conservation -2009, 4th-6th June 2009. October 2, 2000.

vi. Rijndael: Beyond the AES by Joan Daemen ERG Group – Proton World Belgium Vincent Rijmen Cryptomathic NV, Belgium, and IAIK, Graz University of Technology, Austria.

vii. Report on the Development of the Advanced Encryption Standard (AES) by James Nechvatal, Elaine Barker, Lawrence Bassham, William Burr, Morris Dworkin, James Foti, Edward Roback , October 2, 2000.

viii. AES Proposal by Rijndael Joan Daemen, Vincent Rijmen Joan Daemen Proton World Int.l Zweefvliegtuigstraat 10 B-1130 Brussel, Belgium Vincent Rijmen Katholieke Universiteit Leuven, ESAT-COSIC K. Mercierlaan 94 B-3001 Heverlee, Belgium.

ix. A Versatile Pipelined Hardware Implementation for Encryption and Decryption using Advanced Encryption Standard Nadia Nedjah1 and Luiza de Macedo Mourelle2 by i. Department of Electronics Engineering and Telecommunications, Faculty of Engineering, State University of Rio de Janeiro, Brazil ii. Department of Systems Engineering and Computation, Faculty of Engineering, State University of Rio de Janeiro, Brazil.

x. Reconfigurable Implementation for the AES Algorithm by Raoel Ashruf, Georgi Gaydadjiev, Stamatis Vassiliadis Computer Engineering Laboratory, Electrical Engineering Department, Delft University of Technology, Delft, The Netherlands.

xi. Cipher and its Verification with FPGA-based Simulation Accelerator by Jae-Gon Lee, Woong Hwangbo, Seonpil Kim and Chong-Min Kyung Department of EECS Korea Advanced Institute of Science and Technology Guseong-dong, Yuseong-gu, Daejeon, 305-701, Korea.

***xii.** Compact and Efficient Encryption/Decryption Module for FPGA Implementation of the AES Rijndael Very Well Suited for Small Embedded Applications) by Ga¨el Rouvroy, Franc¸ois-Xavier Standaert, Jean-Jacques Quisquater and Jean-Didier Legat UCL Crypto Group Laboratoire de Micro´electronique Universit´e catholique de Louvain Place du Levant.*