Journal of Artificial Intelligence in Electrical Engineering, Vol. 3, No. 9, June 2014

FPGA Can be Implemented Using Advanced Encryption Standard Algorithm

Shahin Shafei
Young Researchers and Elite Club, Mahabad Branch, Islamic Azad University, Mahabad, Iran
Email:Shahin_shafei@yahoo.com

ABSTRACT

This paper mainly focused on implementation of AES encryption and decryption standard AES-128. All the transformations of both Encryption and Decryption are simulated using an iterative design approach in order to minimize the hardware consumption. This method can make it a very low-complex architecture, especially in saving the hardware resource in implementing the AES InverseSub Bytes module and Inverse Mix columns module. As the S -box is implemented by look-up-table in this design, the chip area and power can still be optimized. The new Mix Column transformation improves the performance of the inverse cipher and also reduces the complexity of the system that supports the inverse cipher. As a result this transformation has relatively low relevant diffusion power .This allows for scaling of the architecture towards vulnerable portable and cost-sensitive communications devices in consumer and military applications.

KEYWORDS: AES, encryption, decryption, FPGA

1. INTRODUCTION

The need of privacy has become a high priority for both government and civilians desiring protection from signal and data interception. Widespread use of personal communication devices has only increased demand for a level security for previously insecure communication by using the DES algorithm. For a long time, the Data Encryption Standard (DES) was considered as a standard for the symmetric key encryption. Data block and key length of DES algorithm has a only about 56 bits. The 56 bits of data block and the key is to be small and can easily break. For this reason,

in September 1997, the National Institute of Standards and Technology (NIST) promoted worldwide research into a replacement for DES, or the widely accepted Data Encryption Standard.

AES algorithm candidates of 15 members were announced in August 1998. Next all algorithms were subject to assessment process performed by various group of cryptographic researchers all over the world. NIST selected the 5 algorithm in August 2000 they are, RC6, Mars, Rijndeal, Serpent, and Two fish as the final competitors. On October 2, 2000, NIST announced that the Rijndeal algorithm was the winner. As per the Reijndeal data block and key size of multiple of 32 bits, with a minimum of 128 bits and maximum of 256 bits. As a result of breaking of the data block and key can be difficult [1]. Advanced Encryption Standard (AES) algorithm also known as Rijndeal [2].

AES has fixed data block size of 128 bits and key size of 128, 192 or 256 bits. AES minimize cost, focusing on efficiency reduced overall hardware complexity. By incorporating most of the algorithm complexity into the controller, components are reused and efficiency increased. A Verilog hardware implementation is also presented, utilizing a field programmable gate array (FPGA) as a prototyping platform. Thus, the design can be easily migrated to an ASIC implementation in an SoC [3]. This paper is organized as follows. Section deals with the introduction about the Cryptography algorithm generation. Section II deals with the AES algorithm. Section III describes about the Implementation of AES algorithm.

Section IV shows the simulation results and the last Section V concludes the paper and followed by references.

2. AES ALGORITHM

AES algorithm is a symmetric block cipher that can encrypt and decrypt the information. Encryption can convert the original data in to unintelligible data is called cipher text. Decryption can convert the cipher text form in to original data, which is called plain text.

A. AES Encryption

The AES algorithm (AES-128) operates on a 128 bit block of data size and 128 bit key size and executed N-1 loop times. A loop is called a round and the number of iterations of a loop N can be 10. The first and last rounds differ from other rounds in that there is an additional AddRound key transformation at the beginning of the first round and no Mix columns transformation is performed in the last round. AES encryption is given in Fig.1.

1) SubBytes Transformation: The SubBytes transformation is a nonlinear byte substitution, operates on each of the state bytes independently. The SubBytes transformation is done by using S-box. S-box is a pre calculated substitution table contains 256 numbers (from 0 t0 255) and their corresponding resulting value. More details of the calculating the S-box table refers to [4]. S-box table as shown inTable1.

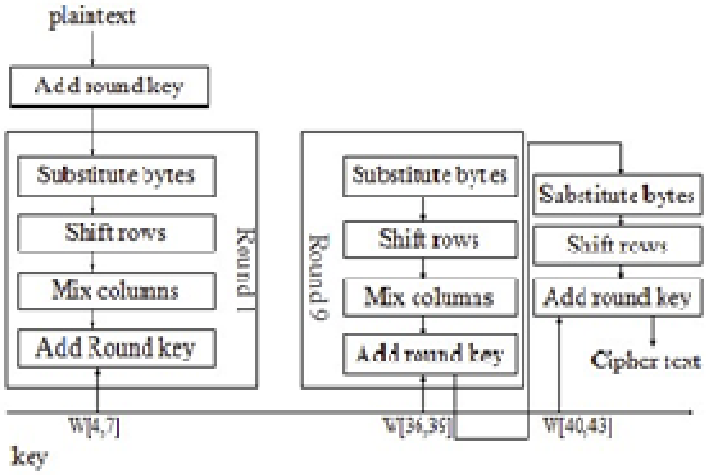

Fig.1. AES encryption structure

Table.1 S-Box Table

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

2) Shift Rows Transformation: In Shift Rows transformation, the rows of the state are left shifted in a cyclic manner. 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.

3) Mix Columns Transformation: In Mix Columns 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}.

4) AddRound Key Transformation: In the AddRound Key transformation, a Round Key is added to the State - resulted from the operation of the Mix Columns transformation - by a simple bitwise XOR operation. The Round Key of each round is derived from the main key using the Key Expansion algorithm [1]. The encryption/decryption algorithm needs eleven 128-bit Round Key, which are denoted Round Key [0].

B. AES Decryption

Decryption is a reverse of encryption, which inverse round transformations to compute out the original plaintext of an encrypted cipher-text in reverse order. The round transformation of decryption uses the functions AddRound Key, InvMix Columns, InvShift Rows, and InvSubBytes successively.

1) AddRound Key: AddRound Key is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order. The description of the other transformations will be given as follows.

2) InvShift Rows Transformation: InvShift Rows exactly functions the same as Shift Rows, 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.

3) 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.

4) Inv Mix Columns Transformation: In the Inv Mix Columns transformation, the 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 {0B}, {0D}; {09}, {0E} denote hexadecimal values.

In the next section, a description of the proposed design based on FPGA implementation of AES encryption / decryption function is detailed.

3. FPGA IMPLEMENTATION OF AES ALGORITHM

Fig.2 shows the detailed design of AES core based on FPGA implementation, where the control signals are described in Table 2.

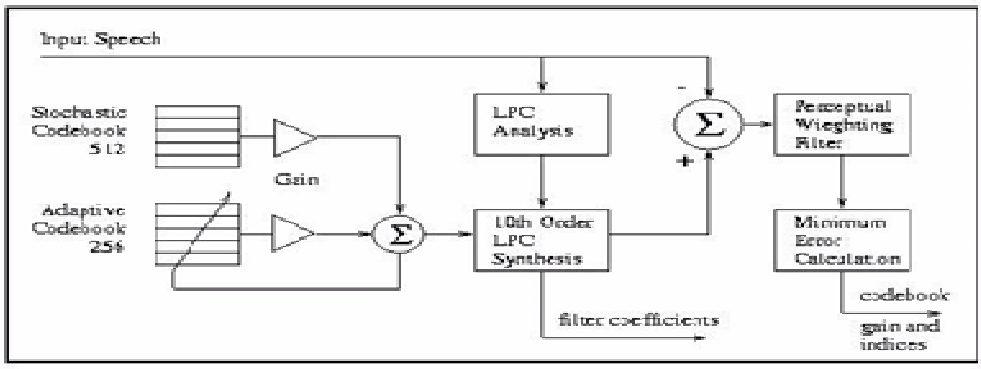

Fig.2. Block diagram of CELP & AES encoder

The total design has 390 pins. It requires the text in, text out and key which have a 128 bits length. And the Control signals used to control the proper operations of the core are clock (clk), reset in, write, direction, done and enable pins. The Key block loads keys and combines with Key Round block to perform Key Expansion transformation, and generates proper Round keys under the control signals from the Controller block.

Table2. Control Signals of Aes Core

Pin Name	I / O Port	Pin Number (bit)	Pin Description
clk	I	1	Chip clock
Reset - in	I	1	Clear all signals
Write	I	1	1:write key and text-in
Direction	I	1	1:Encryption 0:Decryption
Enable	I	1	1:Enable AES core 0:Disable AES core
Key	I	128	Key data
Text-in	I	128	Plain text / Cipher text data
Done	O	1	1:Encryption/Decryption is completed
Text-out	O	128	Plain text / cipher text data

Controller block takes write signal, direction signal, and enable signal from outside and generates all the control signals for the whole system. AES algorithm candidates of 15 members were announced in August 1998. Next all algorithms were subject to assessment process performed by various group of cryptographic researchers all over the world. NIST selected the 5 algorithm in August 2000 they are, RC6, Mars, Rijndeal, Serpent, and Twofish as the final competitors. On October 2, 2000, NIST

announced that the Rijndeal algorithm was the winner. As per the Reijndeal data block and key size of a multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits. The plain text (text_in) and key is loaded only when the write signal makes a low-high-low transition (basically a pulse).

The process is going to complete when the done signal is pulsed after some clock cycles from the write signal goes low.

The "done" signal actives only in one clock cycle. Each round key as well as round is completed in one clock cycle. In this paper Advanced Encryption Standard (AES) algorithm is implemented, that can process with the data block of 128 bits and cipher key length of 128 bits. The usage of the 128 bit cipher keys to achieve higher security, because 128 bits cipher key is difficult to break. As a result of this we obtain a secure transmission of data in both encryption and decryption. While computing the existing AES, it takes more area, so we are going to implement the new algorithm for mix Column in AES flow. When we use a new mix column instead of existing one, we can obtain a less area compare to existing AES algorithm.

However, the round key is finished before the round is calculated by one clock cycle. Hence, combining with one clock cycle for registering the input, a total clock cycle need for processing 128-bit data is 13 clocks in encryption mode. In decryption, eleven round keys must be completed before the first round is calculated. Because the last round key is used in the first round process, it takes 25 clock cycles to complete. By using the above iterative looping approach, a minimal number of clock cycles required performing Encryption/decryption for each data block of 128-bit.

4. SIMULATION RESULTS

The design has been coded by Verilog HDL. All the results are synthesized and simulated based on the Quatus 9.0, the Model Sim - Altera 6.4a and EP20K400CB652C7 device. The results of simulating the encryption / decryption Algorithms from the ModelSim simulator are shown in Fig.3.

1. CONCLUSION

In this paper Advanced Encryption Standard (AES) algorithm is implemented, that can process with the data block of 128 bits and cipher key length of 128 bits. The usage of the 128 bit cipher keys to achieve higher security, because 128 bits cipher key is difficult to break. As a result of this we obtain a secure transmission of data in both encryption and decryption. While computing the existing AES, it takes more area, so we are going to implement the new algorithm for mix Column in AES flow. When we use a new mix column instead of existing one, we can obtain a less area compare to existing AES algorithm.

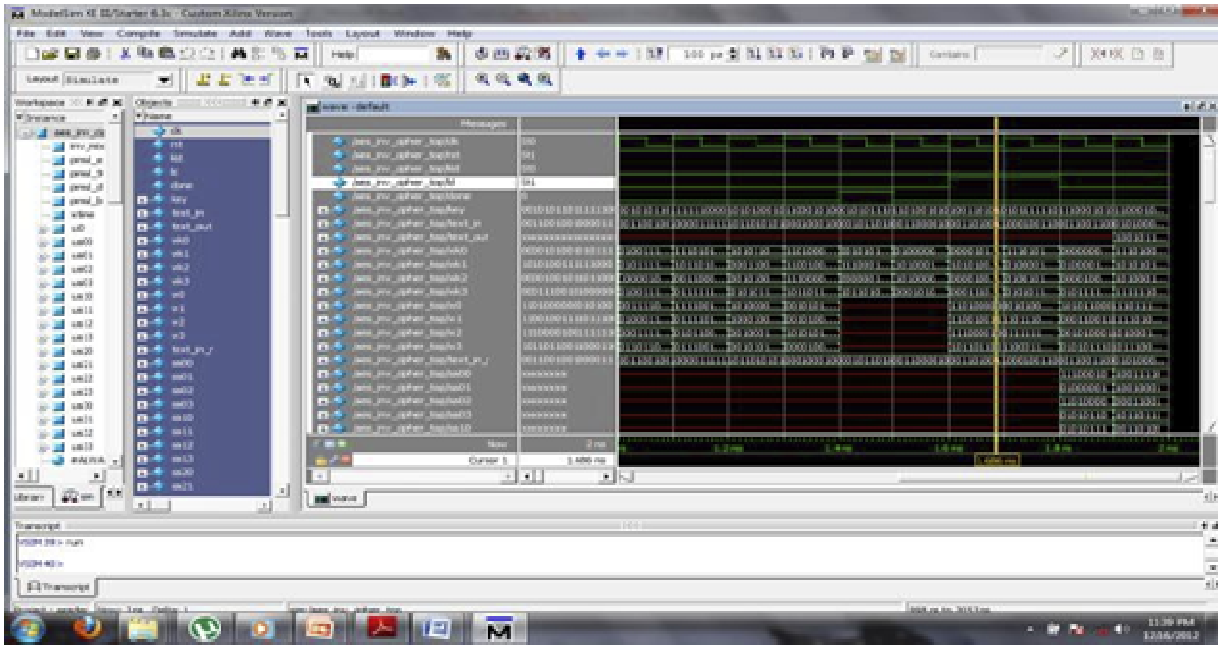

Fig.3. Timing simulation of AES decryption algorithms

REFERENCES

[1]Daemen J., and Rijmen V, "The Design of Rijndael: AES-the Advanced Encryption Standard", Springer-Verlag , 2002

[2]FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[3]Tessier, R., and Burleson, W., "Reconfigurable computing for digital signal processing: a survey", J.VLSI Signal Process, 2001, 28, (1-2), pp.7-27.

[4]Ahmad, N.; Hasan, R.; Jubadi, W.M; "Design of AES S-Box using combinational logic optimization", IEEE Symposium on Industrial Electronics & Applications (ISIEA), pp. 696-699, 2010.