International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.03,Issue.20
September-2014,
Pages:4143-4146

www.semargroup.org,
www.ijsetr.com

Data Encryption and Decryption using AES with Key Length of 256 Bits

K. GAYATHRI[1], W. YASMEEN[2]

[1]PG Scholar, Dept of ECE, Intellectual Institute of Technology, India, Email: kgayathri.vijji@gmail.com.
[2]Assistant Professor, Dept of ECE, Intellectual Institute of Technology, India.

Abstract: An implementation of high speed AES algorithm based on FPGA is presented in this paper in order to improve the safety of data in transmission. The mathematic principle, encryption process and logic structure of AES algorithm are introduced. So as to reach the purpose of improving the system computing speed, the pipelining and parallel processing methods were used. The simulation results show that the high-speed AES encryption algorithm implemented correctly. Using the method of AES encryption the data could be protected effectively.

Keywords: QSD, Carry/Borrow Free Addition/ Subtraction.

I. INTRODUCTION

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, all 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. Rijndael can be specified with key and block sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits. Therefore, the problem of breaking the key becomes more difficult. 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.

The AES algorithm can be efficiently implemented by hardware and 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. An FPGA implementation is an intermediate solution between general purpose processors (GPPs) and application specific integrated circuits (ASICs). It has advantages over both GPPs and ASICs. It provides a faster hardware solution than a GPP. Also, it has a wider applicability than ASICs since its configuring software makes use of the broad range of functionality supported by the reconfigurable device.

II. AES ENCRYPTION ALGORITHM

A. AES Principle

The AES algorithm is based on finite field $GF(2^8)$. The Algorithm includes two mathematical operations, Multiplication and exclusive or, shown as formula (1):

$$\begin{cases} S_{0,c} = (\{02\} * S_{0,c}) \oplus (\{03\} * S_{1,c}) \oplus (\{01\} * S_{2,c}) \oplus (\{01\} * S_{3,c}) \\ S_{1,c} = (\{01\} * S_{0,c}) \oplus (\{02\} * S_{1,c}) \oplus (\{03\} * S_{2,c}) \oplus (\{01\} * S_{3,c}) \\ S_{2,c} = (\{01\} * S_{0,c}) \oplus (\{01\} * S_{1,c}) \oplus (\{02\} * S_{2,c}) \oplus (\{03\} * S_{3,c}) \\ S_{3,c} = (\{03\} * S_{0,c}) \oplus (\{01\} * S_{1,c}) \oplus (\{01\} * S_{2,c}) \oplus (\{02\} * S_{3,c}) \end{cases} \quad (1)$$

AES algorithm has different keys with different length of 128 bits, 192 bits and 256 bits, shown as table 1. Nk (4, 6 or 8) stands for the key length (number of the words in the key). The round number is determined by the length of the key. The relationship is shown in the below table 1.

Table 1. Relationship between round number and key length

AES	Key Length	Group Size	Round Number
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Some FPGA devices like Xilinx Virtex-5 and Virtex-6 have on-chip AES decryption Logic which can be used for data integrity and security. Figure 1 shows the architecture of the universal AES module in FPGA devices.

B. AES encryption process

AES is an iterative group code with the Key. The encryption process includes an initial key-addition called Add Round Key, then an initial round conversion for 1 r $N-$ times, finally a final round conversion again. All the round conversions and initial key-addition make a state and a

Round Key as the input, and each cycle carries out four different operations to the Data flow, Sub Bytes, Shift Rows, Mix Columns and Add Round Key. In order to accomplish an encryption process, ten times of round must be iterative. Figure 2 shows the flow of AES encryption algorithm.

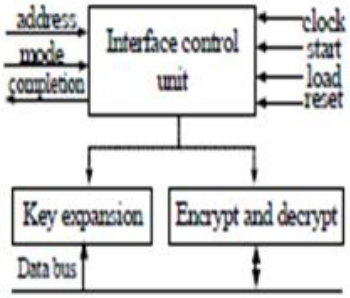

Fig1. AES Module Architecture.

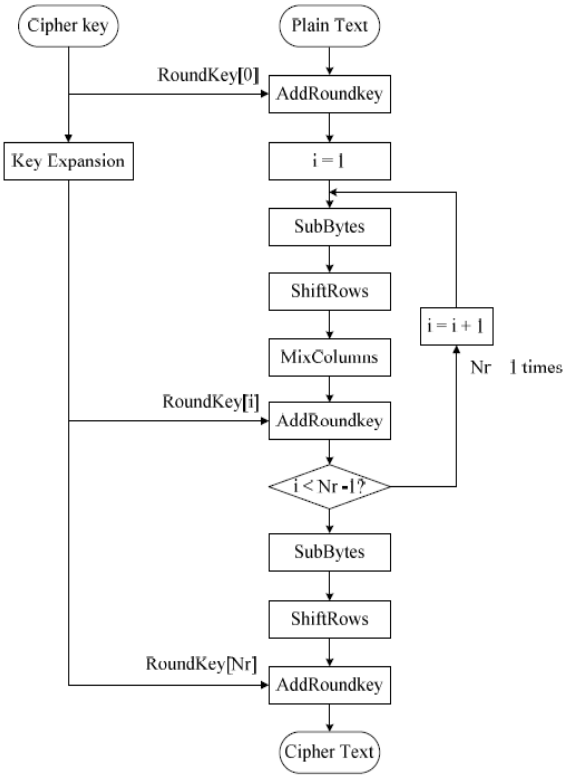

Fig2. AES algorithm flow

SubBytes: The Sub Bytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The Sub Bytes transformation is done using a once-pre calculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. More details of the method of calculating the S-box table refers to [4]. In this design, we use a look-up table as shown in Table II. This is a more efficient method than directly implementing the multiplicative inverse operation followed by affine transformation. To make the four high-order bits in input byte as the value of the row, and the four low-order bits as the value of the column, then put out the corresponding value of row and column in the S-box matrix, expressed as:

$$S[x] = f(g(x)) \quad (2)$$

$g(x)$ signifies the transformation:

$$x \rightarrow x^{-1} GF(2^8) \quad (3)$$

This process includes two steps: first, executing the inverse of multiplication on the finite field GF(28), then making the transforming. Figure 3 shows the transformation process of Sub Bytes.

TABLE II: S-BOX

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	1	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	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	2	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	6	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	8
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	3	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

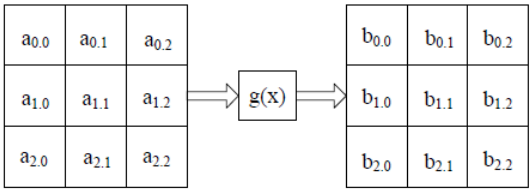

Fig3. Sub Bytes Transformation.

Shift Row Transformation: Shift Rows is the operation of cyclic shift. The specific processes include the following steps: 1.Row 0 remains intact. 2. The first row moves 1 byte to the left. 3. The second row moves 2 bytes. 4. The third row moves 3 bytes [2]. Thus the ith row and the jth column move to (j- ai) mod Xn 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}.

Add Round Key: The process of Add Round Key is that making a key I K with 128 bits to exclusive or the data in state one by one, the prime key always expands and produces a group of round keys in each of the encryption cycle, the size of the round keys as same as the foregoing matrix.

C. 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. The round transformation of decryption uses the functions Add RoundKey, Inv Mix Columns, 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. The description of the other transformations will be given as follows.

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. InvS-box is presented in Table III.

TABLE II. INVS-BOX TABLE

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

InvMixColumns Transformation: In the InvMixColumns 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.

III. AES ALGORITHM IMPLEMENTATION IN FPGA

A. System Design Overall

It is incompatible to implement the AES algorithm on hardware between the throughput and hardware resource. Different architecture should be selected according to the fields it is applied to. To make AES algorithm suitable to high-speed rate data application, we need to optimize the architecture. Meanwhile by sharing resource and eliminating common sub expression we can reduce the hardware resource utilization. There are three basic architectures of AES to improve the throughput: Loop unrolled, pipelined, sub-pipelined that could be chosen [4]. The top design of AES encryption system is shown in Figure4. The top design includes three modules: Round module, Key_scheduel module and control module. Round module, which contains four sub modules: Sub Bytes, Shift Rows, Mix Columns and Add Round Key, performs the prime transformation process of AES. Key_scheduel module includes an S box macro module to realize the nonlinear transformation. Control module in charge of the signals for other modules and the data in Input/ Output.

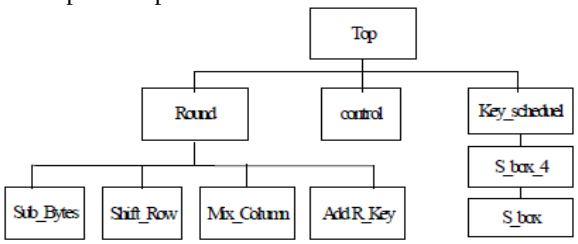

Fig4. Top Design of AES Encryption System

B. System Implementation

The design uses a synchronous clock in order to make the circuit works with a unified clock and uses pipeline architecture to improve the working speed. Figure 5 shows the system implementation structure. Round module includes Sub Bytes, Shift Rows,Mix Columns, Add Round Key and an S-box matrix. Sub Bytes is a substituted operation to execute the operation and the affine transformation on finite field. Shift Rows is a cycle shift with bytes for unit. The most important process in Mix Columns is the multiplication on finite field. Add Round Key is a process that makes a 128 bits key to exclusive or the data in state one by one. S-box is a matrix that be defined to make a nonlinear replacement for Sub Bytes. The structure of round operation is shown in Fig 5

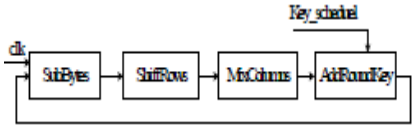

Fig5. Structure of round operation.

The pipeline scheme is utilized for implementations. In the pipeline scheme, the register arrays are assigned among the operational circuits of Sub Bytes, Shift Rows, Mix Columns and Add Round Key. Key_schedue module includes Byte-substitution, Byte shifting, round constant, exclusive or operation and an S-box. A key is composed of four words in the design, key_word(0),key_word(1), key_word(2) and key_word(3). The initial input key is 128 bits. The ranges of each word are defined as follows:

```
key_word(0) <= key_reg_out(127 downto 96);
key_word(1) <= key_reg_out(95 downto 64);
key_word(2) <= key_reg_out(63 downto 32);
key_word(3) <= key_reg_out(31 downto 0);
```

C. Control module

The task of Control module is Co ordinating working signaling for other modules. Control module manages the crypto key memory and the output of the Round constant, and controls the storage of Key_schedue and saves the encryption control signal, to ensure it can be output correctly. In the design, a register is defined to counter the number of the rounds. The sequence is from 1 to 10. Control module controls the 10 round constants in order and makes the transformation in a correct running order.

IV.IMPLEMENTATION AND RESULTS:

The implementation of the proposed system using VHDL Hardware Description Languages and the simulation Results are as Follows

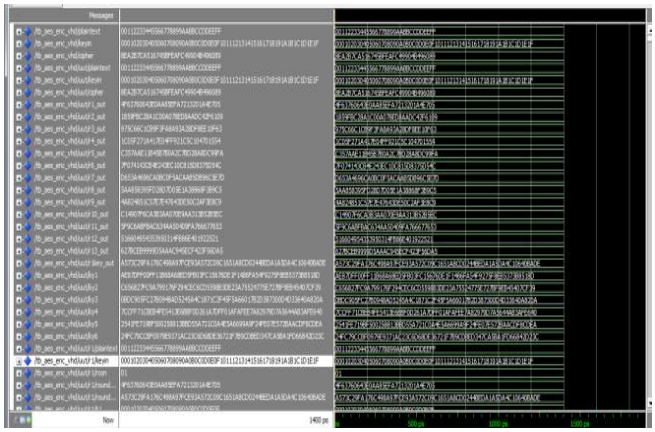

Fig6. simulation result of encryption.

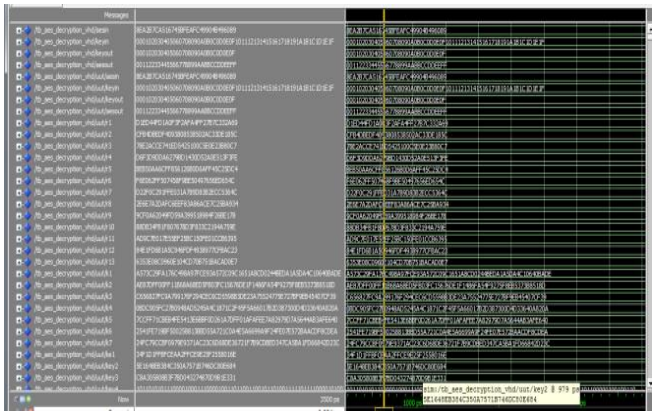

Fig7. simulation result of decryption.

V. CONCLUSION & FUTURE SCOPE

This paper presents the design and implementation based on Distributed Arithmetic, which is used to realize a 31-order FIR low-pass filter. Distributed Arithmetic structure is used to increase the resources usage while pipeline structure is used to increase the system speed. The test results indicate that the designed filter using Distributed Arithmetic can work stable with high speed and can save almost 50 percent hardware resources. Meanwhile, it is very easy to transplant the filter to other applications through modifying the order parameter or bit width and other parameters, and therefore have great practical applications in digit signal processing.

The research performed in this thesis also exposed several new research areas that could be explored. The first of these is the potential to make further improvements to the energy efficient full AES design. Though the use of a composite field GF(((22)2)2 based multiplicative inversion in the S-Boxes was examined, it has been suggested that GF((24)2) based implementations could dissipate less power [44] [42]. A Pipelined version of this form of S-Box with Opportunistic Combinational Operand Gating applied could lead to efficient results. Also, it would be valuable to experiment with other AES structures, such as using lower data path widths or fully unrolling the loop architecture in order to evaluate the impact of these design decisions on energy efficiency. Experimenting with enabling data to be input over multiple clock cycles to reduce the number of input pins required would also be useful.

VI. 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.

[5] Alex Panato, Marcelo Barcelos, Ricardo Reis, “An IP of an Advanced Encryption Standard for Altera Devices”, SBCCI 2002, pp. 197-202, Porto Alegre, Brazil, 9 and 14 September 2002.

[6] Mr. Atul M. Borkar, Dr. R. V. Kshirsagar and Mrs. M. V. Vyawahare, “FPGA Implementation of AES Algorithm”, International Conference on Electronics Computer Technology (ICECT), pp. 401-405, 2011 3rd.

Author’s Profile:

K.Gayathri received the B.Tech degree in Electronics and Communication Engineering in the year 2010 and pursuing M.Tech degree in VLSI System design from Intellectual institute of technology. Her area of interests includes Communication and VLSI Design.

W.Yasmeen received the B.Tech degree in Electronics and Communication Engineering and M.Tech degree inVLSI system design. Currently, her working as Assistant Professor in the Department of Electronics and Communication Engineering, Intellectual institute of technology, Ananthapuramu. Having one years of teaching experience. Area of interest includes Microprocessor & micro controller, Embedded systems and Optical networks