International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 2, No. 2, July 2013, pp. 83~88
ISSN: 2089-4864

Low Power VLSI Design and Implementation of Area-Optimized 256-bit AEStandard for Real Time Images on Vertex 5

Shruthi AV, Electa Alice, Mohammed Bilal
ECE Department, T. John Institute of Technology

Article Info

Article history:

Received May 7, 2013
Revised Jun 16, 2013
Accepted Jun 28, 2013

Keyword:

Advanced encryption standard
Area optimization
Pipelining
Verilog
Vertex-5

ABSTRACT

A new Vertex6-chipscope based implementation scheme of the AES-256 (Advanced Encryption Standard, with 256-bit key) encryption and decryption algorithm is proposed in this paper. For maintaining the speed of encryption and decryption, the pipelining technology is applied and the mode of data transmission is modified in this design so that the chip size can be reduced. The 256-bit plaintext and the 256- bit initial key, as well as the 256-bit output of cipher-text, are all divided into four 32-bit consecutive units respectively controlled by the clock. In this novel work, substantial improvement in performance in terms of area, power and dynamic speed has been obtained.

Corresponding Author:

Shruthi AV,
ECE Department,
T. John Institute of Technology
Bangalore. Karnataka, India
Email: avshruthi02@gmail.com

1. INTRODUCTION

Advanced Encryption Standard is a symmetric key encryption and decryption technique which will replace the commonly used Data Encryption Standard (DES). The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128,192,256 and 512 bits to encrypt and decrypt data in blocks of 128,192,256 and 512 bits [1]-[2].

Throughout the remainder of this standard, the algorithm specified herein will be referred to as "the AES algorithm". The algorithm may be used with the three different key lengths indicated above, and therefore these different "flavors" may be referred to as "AES-128", "AES-192", "AES-256" and "AES-512" [2]. With the rapid development and wide application of computer and communication networks, the information security has aroused high attention. Information security is not only applied to the political, military and diplomatic fields, but also applied to the common fields of people's daily lives. With the continuous development of cryptographic techniques, the long-serving DES algorithm with 56-bit key length has been broken because of the defect of short keys [3]. AES (Advanced Encryption Standard) algorithm whose packet length is 256 bits and the key length is 256 bits. AES can resist various currently known attacks. Hardware security solution based on highly optimized programmable FPGA provides the parallel processing capabilities and can achieve the required encryption performance benchmarks. The current area-optimized algorithms of AES are mainly based on the realization of S-box mode and the minimizing of the

internal registers which could save the area of IP core significantly [4]-[8].

A new AES algorithm with 256-bit keys (AES-256) was described in this paper, which was realized in Verilog Hardware Description Language. The 256-bit plaintext and 256-bit key, as well as the 256-bit output data were all divided into four 32-bit consecutive units respectively. The pipelining technology as discussed in [5] was utilized in the intermediate nine round transformations so that the new algorithm achieved a balance between encryption speed and chip area, which met the requirements of practical application. Firstly, functional simulation and timing analysis of this algorithm had been achieved in the chipscope and the Xilinx 14.1 platform. Then we completed the synthesis simulation of this design based on IES simulator. The data of each column (32 bits) in the state matrix was used to be an operand of encryption, when the operation of ShiftRows and SubBytes were incorporated. And each round of the intermediate nine Round Transformations of encryption was processed by pipelining technology. The results show that this new algorithm with pipelining technology and special mode of data transmission can significantly decrease the quantity of chip pins and reduce the chip area.

Rest of the paper is organized as follows. Section 2 describes brief description, design and implementation of Advance encryption algorithm. Simulation results of proposed design are given in Section 3. In section 4 concludes the proposed work.

2. THE FPGA IMPLEMENTATION OF AREA-OPTIMIZED AES-256

2.1. Brief Description of Rijndael Algorithm

Rijndael algorithm consists of encryption, decryption and key schedule algorithm. The main operations of the encryption algorithm among the three parts of Rijndael algorithm include: bytes substitution (SubBytes), the row shift (ShiftRows), column mixing (MixColumns), and the round key adding (AddRoundKey) as shown in Figure 1.

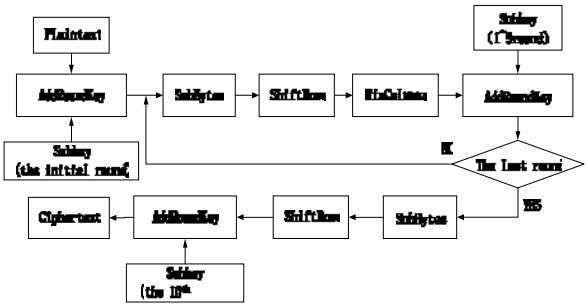

Figure 1. The structure of Rijndael encryption algorithm

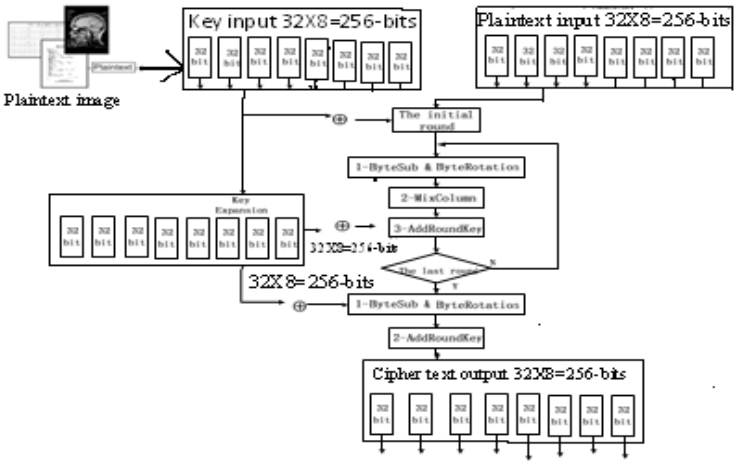

Figure 2. Proposed architecture of selective encryption algorithm.

Figure 2 shows the proposed architecture of selective encryption algorithm. The basic unit for processing in the AES algorithm is a byte**,** a sequence of eight bits treated as a single entity. The input, output and Cipher Key bit sequences described are processed as arrays of bytes that are formed by dividing these sequences into groups of eight contiguous bits to form arrays of bytes. For an input, output or Cipher Key denoted by a, the bytes in the resulting array will be referenced using one of the two forms, an or a[n], where n will be in one of the following ranges:

Key length = 128 bits, 0 £ n < 16;
Block length = 128 bits, 0 £ n < 16;
Key length = 192 bits, 0 £ n < 24;
Key length = 256 bits, 0 £ n < 32.

Transformation of the plaintext for the ciphertext, the value of Nr in AES algorithm whose packet length is 256 bits should be 10, 12, or 14 respectively, corresponding to the key length of 256 bits. In this paper, only the (AES-256) encryption scheme with 256-bit keys is considered. All byte values in the AES algorithm will be presented as the concatenation of its individual bit values (0 or 1) between braces in the order {b7, b6, b5, b4, b3, b2, b1, b0}. These bytes are interpreted as finite field elements using a polynomial representation:

$$b_7x_0^7 + b_6x_0^6 + b_5x_0^5 + b_4x_0^4 + b_3x_0^3 + b_2x_0^2 + b_1x_0^1 + b_0x_0^0$$

For example, {01100011} identifies the specific finite field element $x^6 + x^5 + x + 1$.

2.2. The Design of Improved AES-256 Encryption Algorithm

Two main processes of AES encryption algorithm:

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation. Key schedule consists of two modules: key expansion and round key election. Key expansion means mapping Nk bits initial key to the so-called expanded key, while the round key selection selects Nb bits of round key from the expanded key module. Round Transformation involves four modules by ByteSubstitution, ByteRotation, MixColumn and AddRoundKey. Figure 3 shows the state matrix format.

a00	a01	a02	a03
a10	a11	a12	a13
a20	a21	a21	a23
a30	a31	a32	a33

Figure 3. The state matrix representation

In the four transformation modules of round transformation, the ByteRotation, MixColumn and AddRoundKey are all linear transformations except the ByteSub. Take analysis of the AES algorithm principle and we can find:

a) ByteSubstitution operation simply replaces the element of 256-bit input plaintext with the inverse element corresponding to the Galois field GF (2^8), whose smallest unit of operation is 8 bits/ group.
b) ByteRotation operation takes cyclic shift of the 256-bit state matrix, in which one row (32 bits) is taken as the smallest operand.
c) MixClumns operation takes multiplication and addition operations of the results of ByteRotation with the corresponding irreducible polynomial x8 + x4 + x3 + x + 1 in GF(2^8), whose minimum operating unit is 32 bits.
d) Addroundkey operation takes a simple XOR operation with 8-bit units.

The inputs of plaintext and initial key, intermediate inputs and outputs of round transformation, as well as the output of ciphertext in the AES algorithm are all stored in the state matrixes, which are processed in one byte or one word. For the AES algorithm, the length of the input block, the output block and the State is 128 bits. This is represented by Nb = 4, which reflects the number of 32-bit words (number of columns) in the State. For the AES algorithm, the length of the Cipher Key, K, is 128, 192, or 256 bits. The key length is

represented by Nk = 4, 6, or 8, which reflects the number of 32-bit words (number of columns) in the Cipher Key. For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by Nr, where Nr = 10 when Nk = 4, Nr = 12 when Nk = 6, and Nr = 14 when Nk = 8.

Thus in order to take operations at least bits, the original 256-bit data should be segmented. We design some external controllers in the new algorithm, so that the data transmission and processing can be implemented on each column of the state matrix (32bit). That means the data should be packed and put into further operations. Take the independent and reversible bytes substitution operation of S-box as example. Firstly, the state matrix is divided into four columns. And then byte replacement is achieved by the operation of look-up table shown as Figure 4. Therefore, the original 256-bit input of plaintext and key will be replaced with four consecutive 32-bit input sequences respectively.

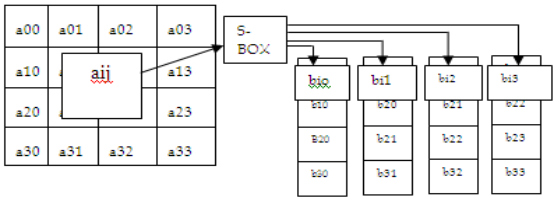

Figure 4. Bytes segmentation and replacement processing

In order to decrease the output ports, four continuous 32-bit ciphertext sequences have taken place of the original 256-bit output by adding a clock controller. The 256-bit data in the round transformation is also split into four groups of 32-bit data before the operation of pipelining.

2.3. The Process of New Algorithms

From the analysis, we can resolve the process of AES encryption mainly divided into two parts: key schedule and round transformation. The improved structure is also divided into these two major processes. The initial key will be sent to the two modules: Key expansion and Key selection, while the plaintext is to be sent to the round transformation after the roundkey is selected. But the operand of data transmission is turned into a 32-bit unit. The functions of various parts of the structure shown above are described as follow:

The initial round of encryption:

a) The four packets of consecutive 32-bit plaintext (256 bits) have been put into the corresponding registers. Meanwhile, another four packets of consecutive 32-bit initial key (256 bits) have been put into other registers by the control of the enable clock signal. Furthermore, this module should combine the plaintext and initial key by using the XOR operators.

b) Round Transformation in the intermediate steps:

A round transformation mainly realizes the function of SubBytes and MixColumns with 32-bit columns. Four packets results of MixColumns and the 32-bit keys sourced from Key expansion are combined by using XOR operators. Here, the round transformation is a module with 64 input ports (32- bit plaintext+32-bit key) and 32 output ports. The function of SubBytes is realized by Look-Up Table (LUT). It means that the operation is completed by the Find and Replace after all replacement units are stored in a memory (256×8bit = 1024 bit).

The implementation of MixColumn is mainly based on the mathematical analysis in the Galois field GF(2^8). Only the multiplication module and the 32-bit XOR module of each processing unit (one column) are needed to design, because the elements of the multiplication and addition in Galois field are commutative and associative. Then the function of MixColumn can be achieved.

In the process of pipelining, the 256-bit data is divided into four consecutive 32-bit packets that take round transformation independently. Store the unprocessed data in the 256-bit register, and control the clock for re-starting the 256-bit register to read the new data when the four groups' operations have been overcome. Thus the 256-bit round-operating unit has been transformed into four 32-bit round-operating elements. The internal pipelining processing should be implemented during the whole nine intermediate Round Transformations of the four packets before achieving the 256-bit ciphertext.

a) The process of the last round the final round is a 128-bit processor. After nine rounds of operations included Shiftrows, SubByte and Mixclumns, the 128-bit intermediate encrypted data will be used in XOR operation with the final expanded key (8*32bit), which is provided by the key expansion module. The output of final round in the processor is the desired 256-bit ciphertext.

3. RESULT ANALYSIS

3.1. Simulation in the ISE Chip Scope Simulator

In order to study the performance 8-bit grayscale 256X256 standard lena image is used. Firstly, all project files of the design were compiled in ISE chipscope simulator platform. If the files were all compiled successfully, the simulated waveforms could be obtained when loading the test file test_bench_top. Figure 6 shows the simulation results for the newly adopted algorithm for the various possible combinations.

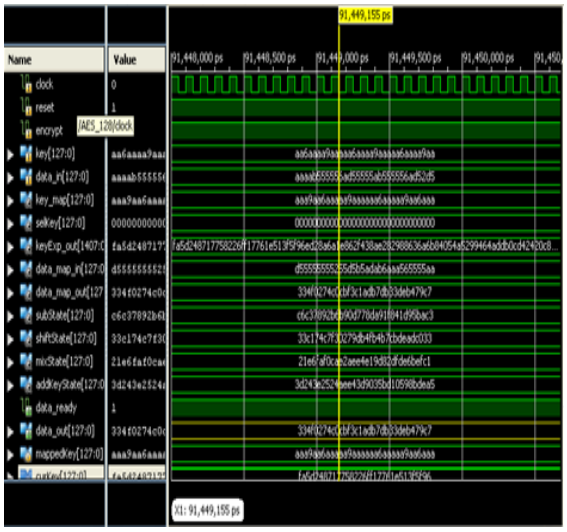

Figure 6. Encryption the 256-bit plaintext, 256-bit initial key and 256-bit cyphertext

Figure 6A. Decryption The 256-bit plaintext, 256-bit initial key and 256-bit cyphertext.

The gray scale image pixels are converted into bits and blocks of 256 bits were grouped. Initial 256-bit input tmp0 sequences are extracted to eight 32-bit words as the plaintext (256bit) shown as Figure 6 meanwhile, the 256-bit input sequences tmp1 are extracted to four 32-bit words as initial key (256bit) the sequences of tmp2(256bit) are the correct cipher text data, which is used for validating the correctness of the new encryption scheme. four consecutive state-words of input in1 are consistent with 256 bits key. After a complete process of AES encryption, the output stream data_out_32 exports four continuous 32-bit sequences, which are consistent with the 256bits cipher text tmp2.

3.2. The Simulation in the Xilinx 14.1 Platform

The logic function of new AES has been verified in the ISE chipscope simulator platform. In order to take the pre-analysis about the physical parameters of the chip before synthesis verification, a successful simulation has been done in the platform of Xilinx 14.1. We obtain some basic different information between the unimproved algorithm and improved algorithm when contrasting two reports in the platform. The results are shown in Table 1.

Table 1. The Comparison of Parameters

	Total logic elements	Total registers	Total Pins
Previous work for 32 bits	1511	674	384
Current work for 256 bits	388	5756	388

Table 2. Comparison in Encryption Chip Parameters

	Cell area	Total Dynamic Power	Total Dynamic Time
Previous work for 32 bits	200504.48 10-12m	22.021mw	6.00ns
Current work for 256-bits	52131.16 10-12m	32.56mw	4.5ns

4. CONCLUSION

A FPGA implementation of area-optimized AES algorithm which meets the actual application is proposed in this paper. After being coded with Verilog Hardware Description Language, the waveform simulation of the new algorithm was taken in the ISE chipscope simulator and Xilinx 14.1 platform. Ultimately, a synthesis simulation of the new algorithm has been done. The result shows that the design with the pipelining technology and special data transmission mode can optimize the chip area effectively. Meanwhile, this design reduces power consumption to some extent, for the power consumption is directly related to the chip area. As the S-box is implemented by look-up-table in this design, the chip area and power can still be optimized. So the future work should focus on the implementation mode of S-box. Mathematics in Galois field (2^8) can accomplish the bytes substitution of the AES algorithm, which could be another idea of further research.

ACKNOWLEDGEMENTS

I would like to thank the Management and Principal of R.L.Jalappa Institute of Technology, Doddabalapur, Bangalore, for their kind support and encouragement for this research work to complete.

REFERENCES

[1] Hamdan O Alanazi, BB Zaidan, AA Zaidan, Hamid A Jalab, M Shabbir, Y Al-Nabhani. New Comparative Study between DES, 3DES and AES within Nine Factors. *Journal of Computing*. 2010; 2(3), Issn 2151-9617

[2] J Yang, J Ding, N Li, YX Guo. FPGA-based design and implementation of reduced AES algorithm. *IEEE Inter.Conf. Chal Envir Sci Com Engin (CESCE).* 2010; 02(5-6): 67-70.

[3] Luke St Clair, Lisa Johansen, William Enck, Matthew Pirretti, Patrick Traynor, Patrick McDaniel, Trent Jaeger. Password Exhaustion: Predicting the End of Password Usefulness. *ICISS 2006, LNCS 4332*. 2006; 37–55.

[4] Hiremath S, Suma MS. Advanced Encryption Standard Implemented on FPGA. *IEEE Inter. Conf. Comp Elec Engin. (IECEE)*. 2009; 02(28): 656-660.

[5] Abdel-hafeez S, Sawalmeh A, Bataineh S. High Performance AES Design using Pipelining Structure over GF (28). *IEEE Inter Conf.Signal Proc and Com*., 2007; 24-27: 716-719.

[6] Rizk MRM, Morsy M. Optimized Area and Optimized Speed Hardware Implementations of AES on FPGA. *IEEE Inter Conf. Desig Tes Wor*., 2007; 1(16): 207-217.

[7] Liberatori M, Otero F, Bonadero JC, Castineira J. *AES-128 Cipher. High Speed, Low Cost FPGA Implementation.* IEEE Conf. Southern Programmable Logic (SPL). 2007; 04(07): 195-198.