ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Design and Implementation of Low Area S-Box for AES Algorithm

Ann Maria Joseph [1], Prasanthi Mortha [2]

P.G. Student, School of Electronics, Vellore Institute of Technology, Vellore, Tamil Nadu, India[1]

P.G. Student, School of Electronics, Vellore Institute of Technology, Vellore, Tamil Nadu, India[2]

ABSTRACT: Security is of great concern in all fields. Advanced Encryption Standard (AES) provides the best security of all the available cryptographic algorithms which is one of the common symmetric encryption algorithm. AES comes with different length keys of 128, 192, 256 bits for both encryption and decryption. The method which is adopted here is VHDL implementation of 128 bit AES over FPGA. As far as an AES system is concerned substitution-box(S-Box) creates complexity thereby reducing speed. It is important that we need to modify S-Box to achieve improvement in speed. Substitution box(S-Box) architecture here employs basic logic gates which reduces delay thus improving the speed. For applications that require high speed systems this will reduce the overall system delay. We have programmed in Xilinx–13.1xst software and implemented on FPGA.

KEYWORDS: *AES, FPGA, S-Box.*

I. INTRODUCTION

The AES is a private key symmetric block cipher published by NIST (National Institutes of Standards and technology) with the aim of replacing the existing DES algorithm which was vulnerable to various attacks. The primary motive behind the development of AES is of flexibility, efficient implementation in software and hardware, performance and security. With its stronger and faster execution AES finds it great application in multiple fields.

AES takes data in blocks of about 128-bit and the key length varies. The key length it can take is of 128-bit, 192-bit and 256-bit.The user can choose one of these keys based on the application and the available resources. In this paper we focus on implementing 128-bit key which requires 11 rounds of logic operations to be performed. The first stage is to perform only Add Round Key transformation followed by nine successive rounds that perform some special transformations.

These nine rounds mainly consist of SubBytes, ShiftRows, Mix Columns and AddRoundKey transformations. The final round consists of Sub Bytes, Shift Rows and Add Round Key transformations. Here the mix column transformation is omitted. The round keys are generated for each round based on the original 128 bit input key.

They can be generated on the fly or can be generated previously and stored in memory. For better implementation of AES different methods can be adopted. Implementing AES in hardware gives better results in terms of speed, security and throughput. Hardware implementation of AES provides good performance based on the availability of hardware. The complexity in S-Box algorithm is mainly dominated by the presence of AES substitution box(s-box).It is necessary to change the traditional S-Box in AES algorithm so as to achieve better performance. The basic architecture of AES algorithm is shown in Fig.1.

The AES algorithm for both encryption and decryption consist of a data processing unit and key expansion unit. The Data processing unit performs the function of sub-Byte transformation, Shift-Rows, Mix-columns and Add Round Key in various rounds. Key expansion unit generates keys for each round.

SubByte: The data processing unit first performs substitution process byte by byte. This process traditionally uses look up table of 16 X 16 to replace each of these bytes.

ShiftRows:The output from the Sub-Byte transformation is fed to Shift-Rows so as to scramble and correlate the bytes.

MixColumns:The bytes obtained from the previous stage is scrambled again by mixing up bytes in each column separately.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

AddRoundKey: This stage mainly performs XOR-ing operation of the corresponding key generated for each round with the output from the mix-columns. Decryption process is considered to be the inverse of the encryption.

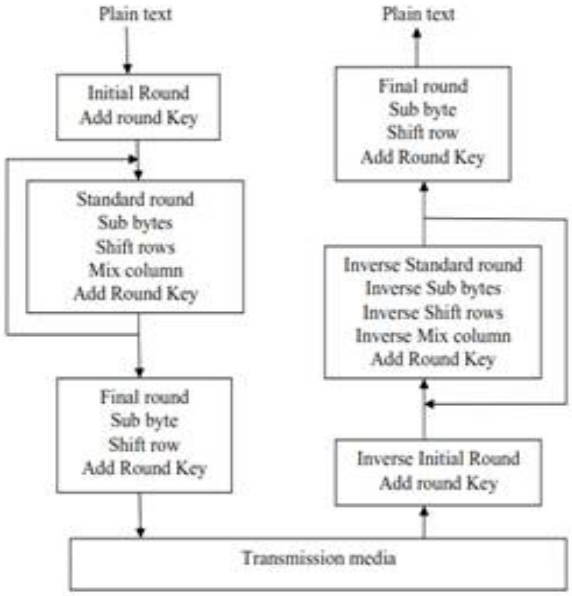

Fig.1 AES basic architecture

II. RELATED WORK

Implementing S-Box mainly is being done by several methods of which the traditional one deals with look up table (LUT) which stores all the 256 bit predefined values of the S-Box. The S-Box designs obtained from its truth table uses direct relationship between input and output values of the S-box. The advantage of using S-Box in look up table offers shorter critical path thus providing less time of execution. It can achieve a high speed design but suffer from extremely large area cost .This mainly has a drawback of unbreakable delay in high speed designs; hence it cannot be used in high speed applications.

Apart from these the total area is more as it require separate substitution box (s-box) implementation for both encryption and decryption. Another way of implementing S-Box is using combinational logic which reduces the area to a great extend. This approach has breakable delay of S-Box processing. The substitution box here employs combinational logic to decrease the area providing an optimized solution. In this paper, we focus on the Boolean simplification of the truth table to design the S-box circuit using basic gates and multiplexer.

III. S-BOX ARCHITECTURE USING COMBINATIONAL LOGIC

The S-Box thus employed here uses combinational logic as a solution to the unbreakable delay offered by the traditional look up table .The proposed architecture reduces the critical path delay by using composite field arithmetic. The S-box has here has 8 bit input and 8 bit output. The (MSB) most significant bits are fed as input to each of these 16 modules logic function (M1, M2, M3….. M16) and the least significant bits are given as the input to the 16:1 multiplexer. The output of the multiplexer will derive output for s-box. This architecture can be used for Sub-Byte Transformation. The proposed S-Box architecture [3] is shown in Fig.2.There are sixteen logic function modules. Each logic function module consists of gates.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

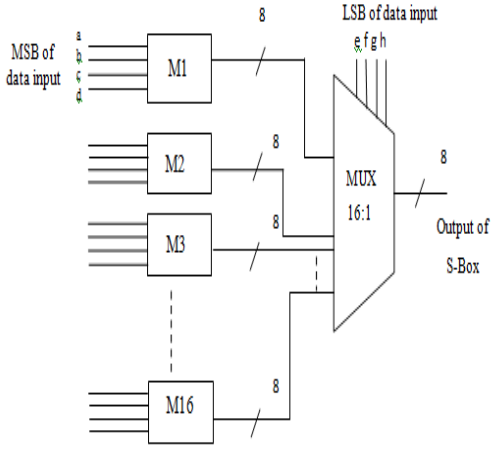

Fig.2 S-Box architecture using basic gates

For example, the Boolean Module 1(M1) as follows:

$$y_7 = b\bar{c}(\bar{a}+\bar{d}) + b(a\bar{d} + \bar{a}cd)$$

$$y_6 = \bar{a} + \bar{b}cd + b(\bar{c}+d)$$

$$y_5 = a\,\square\, c + \bar{d} + \bar{b}c$$

$$y_4 = \bar{a}.\bar{b}(c+d) + \bar{c}\bar{d}(a+b) + abd$$

$$y_3 = c(b \oplus d) + d(\bar{a}\bar{c} + ab)$$

$$y_2 = ab\bar{c} + c(b\,\square\, d) + \bar{a}c\bar{d} + \bar{a}\bar{b}\bar{c}d$$

$$y_1 = (b \oplus c) + (\bar{a}\bar{d}) + ab$$

$$y_0 = c(\bar{b}+\bar{d}) + \bar{a}(b\,\square\, d) + a\bar{c}d$$

IV. EXPERIMENTAL RESULTS

The Substitution box(S-Box) architecture in fig.2 is implemented in Altera IDE 1 using VHDL programming. Xilinx ISE 13.1 software is used to synthesize the design. The results are obtained in terms of cell area, cell, total power, and timing. From table.1, it is observed that inverse s-box consumes more power and area than s-box.

Look-up Table	Cell area	cell	Total power(μW)	Timing(ps)
S-Box	7371	506	290.544	35308
Inverse S-Box	7405	499	277.278	35470

TABLE.1 S-Box

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

The substitution box when implemented using logic gates can be verified. The timing information obtained in both cases varies and are compared. It has been observed that logic gate implementation could give a better delay profile. The total power consumed for both s-box and inverse s-box is as given in table.2.

Combinational	Cell area	cell	Total Power(μW)	Timing(ps)
S-Box	6141	442	293.890	36230
Inverse S-Box	6001	426	274.931	36185

TABLE.2 Combinational logic S-Box

AES simulation results are obtained as shown in Fig.3.The substitution box when implemented using logic gates can be verified and one such output is shown in fig.4.The information obtained in both cases varies and are compared. It has been observed that logic gate implementation could give a better area profile.

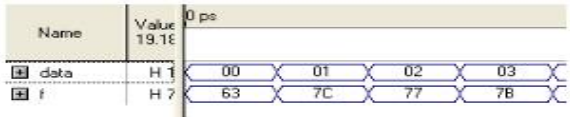

Fig.4 Waveform of sub-byte transformation

V. CONCLUSION

The ultimate goal is to obtain a system can address hardware complexity issues existing in AES algorithm. It is observed that an optimal implementation of substitution box(s-box) can reduce the hardware complexities giving better performance. Further the algorithm can be modified by incorporating concepts of neural network. Each of the other modules like mix-columns, Add Round Key etc can be modified further to optimize the architecture can also be carried out so as to get still more optimization.

REFERENCES

[1] Z. Xinmiao and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, pp. 957-967, 2004.

[2] Nabihah Ahmad, Rezaul Hasan, Warsuzarina Mat Jubadi "Design of AES S-Box using combinational logic optimization" 2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2010), October 3-5, 2010, Penang, Malaysia

[3] Pallavi Atha" Design & Implementation of AES algorithm over FPGA using VHDL" International Association of Scientific Innovation and Research (IASIR), IJEBEA 13-215, 2013

[4] Mg Suresh,Dr.Nataraj.K.R "Area Optimized and Pipelined FPGA Implementation of AES Encryption and Decryption" *International Journal Of Computational Engineering Research Vol. 2 Issue. 7*

[5] Uday Modha, Preeti Dave, " Image Inpainting-Automatic Detection and Removal of Text From Images", International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622 Vol. 2, Issue 2, 2012

[6] Muthukumar S, Dr.Krishnan .N, Pasupathi.P, Deepa. S, "Analysis of Image Inpainting Techniques with Exemplar, Poisson, Successive Elimination and 8 Pixel Neighborhood Methods", International Journal of Computer Applications (0975 – 8887), Volume 9, No.11, 2010