The Design of Encryption Algorithm In Advanced Encryption Standard With The Construction of Optimum In Less Occupancy Area of S-Boxes

MOHANRAJ S.[1]
[1]Asst.Prof in Alpha College of Engineering and Technology

Abstract

Cryptography plays an important role in the security of data transmission. In digital communications the data is sent through the wires or air and thus it is not protected from eavesdropping. Therefore, confidentiality of the transferring data is of extreme importance for any communication system. Advanced Encryption Standard (AES) S-Box is constructed on the basis of Composite Field Arithmetic (CFA) with the Galois field GF $(((2^2)^2)^2)$. The previous technique accommodates more space and more circuitry elements in the implementation. To overcome this problem the new novel technique as common sub-expression elimination algorithm is employed in this implementation of S-boxes. The first case S-Box is implemented with the help of MODELSIM in Verilog language. This technique leads to the low number of gates, low occupancy area and less delay in the implementation architecture. The other case of S-Box was implemented with the corresponding functional blocks. The maximum optimized S-Box architecture was used in the AES technique for the efficient encryption algorithm in data encryptions.

Keywords

Security, Cryptography, Advanced encryption standard (AES), S-Box, VLSI, MODELSIM

1. Introduction

The Advanced Encryption Standard (AES) is an encryption standard chosen by the National Institute of Standards and Technology (NIST) in 2001, which has its origin in the Rijndael block cipher. Several studies in the area had identified the nonlinear Sub Bytes transformation as the major bottleneck in achieving both small area and high speed VLSI AES implementations. This brief presents a methodology that is based on a pure combinatorial circuitry. In which, the Galois inverse of elements in is computed prior using the composite field arithmetic (CFA). To date, there are several successful composite field constructions reported for AES S-box implementations. Summarizing from the previous works, the smallest composite field AES S-box is attributed to can right. However, the issue of critical path was not addressed in Can right's work. A short critical path is highly desirable in VLSI architectures, as it enables deep sub-pipelining for an increased performance in the clock frequency.

On the other hand, the works of Zhang and Parham and contributed an AES S-box with the shortest critical path to date However, their work requires a larger area compared to Can right's. The remainder of this paper is organized as follows. In some details on the AES algorithm are discussed. In we explain our approach to minimize the area of the S-box and compare our new solution with the S-box of Satoh. The current work improves on the compact implementation of in

the following ways. Many choices of representation (isomorphism's) were compared, and the most compact turns out to use a normal basis for each subfield (uses a polynomial basis for each subfield). And while used the "greedy algorithm" to reduce the number of gates in the bit matrices required in changing representations, here each bit matrix is fully optimized, resulting in the minimum number of gates. These various refinements result in an S-box circuit that is 20% smaller, a significant improvement.

2. AES ARCHITECUTRE FOR THE COMPOSITE S-BOXES

- **RIJNDAEL ALOGORITHM**

The Rijndael algorithm is a new generation symmetric block cipher that supports key sizes of 128, 192 and 256 bits, with data handled in 128-bit blocks - however, in excess of AES design criteria, the block sizes can mirror those of the keys. Rijndael uses a variable number of rounds, depending on key/block sizes, as follows:

9 rounds if the key/block size is 128 bits
11 rounds if the key/block size is 192 bits
13 rounds if the key/block size is 256 bits

The exact transformations occur as follows the byte subtransformation is nonlinear and operates on each of the State bytes independently - the invertible S-box (substitution table) is made up of 2 transformations. The shiftrow transformation sees the State shifted over variable offsets. The shift offset values are dependent on the block length of the State. The mixcolumn transformation sees the State columns take on polynomial characteristics over a Galois Field values (28), multiplied x4 + 1 (modulo) with a fixed polynomial. Finally, the round key transform is XORed to the State. The key schedule helps the cipher key determine the round keys through key expansion and round selection.

- ***AES ENCRYPTION AND DECRYPTION***

The figure 1 explaining the encryption of the AES algorithm, each round except the final round consists of four transformations: the SubBytes, the ShiftRows, the MixColumns, and the AddRoundKey, while the final round does not have the MixColumns transformation.

The decryption structure can be derived by inverting the encryption structure directly. However, the sequence of the transformations will be different from that in encryption. the operations involved in the decryption transformations, the InvShiftRows and the InvSubBytes can be exchanged without affecting the decryption process.

Meanwhile, the InvMixColumns can be moved before the AddRoundKey, provided that the InvMixColumns are applied to the roundkeys before they are added.

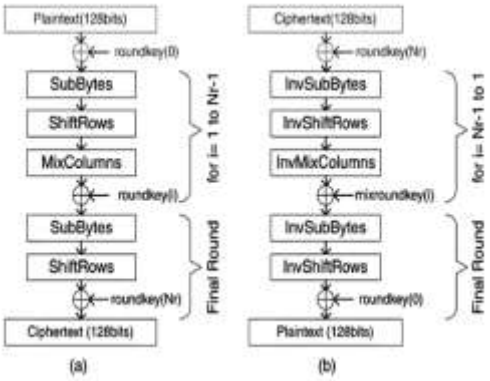

Fig.1. Encryption and Decryption block diagram

3. *CONSTRUCTION OF S- BOXES*

- **S-BOX OPERATIONS**

In cryptography, the S-Box (Substitution-box) is a basic component of Symmetric key algorithms which performs substitution. In block cipher , they are typically used to obscure the relationship between the key and the ciphertext. In general, an S-Box takes some number of input bits, m, and transforms them into some number of output bits, n: an m×n S-Box can be implemented as a lookup table with 2m words of n bits each. Fixed tables are normally used, as in the Data Encryption Standard (DES), but in some ciphers the tables are generated dynamically from the key; e.g. the blowfish and the twofish encryption algorithms.

S-box is a complex and nonlinear operation, single faults propagate through the whole architecture and may generate

more than one error. As a result of single faults, the outputs of the S-box become erroneous. Because of the complex structure of the S-box, the propagation of errors is random.

4. Proposed Methodology

- **DESIGNING OF S-BOXES**

We discuss on the actual hardware implementation of the proposed CFA AES S-box constructions. The implementations can be viewed as two-stage procedure. First, we manually coded the circuit using a hardware description language for all of the three proposed CFA AES S-boxes. Next, we employ ANF representation along with a strategic fine-grained pipeline registers insertion, in an attempt to validate the feasibility of the proposed compact CFA AES S-boxes in achieving high throughput hardware implementations. Consequently, the fastest speed achievable is bounded by the time required for the most complex sub-operations, in this case, the multiplication in GF (24). In order to overcome this constraint, we propose to divide the multiplication sub-operation into two parts, i.e., fine-grained pipelining.

To ensure the efficiency of the pipelined constructions, we need to first convert the complicated circuit into several logical expressions without violating the functionality of the circuit's properties. Converting a CFA AES S-box into logical expressions has to be done in a way that it does not induce excessive area increase. To ensure this, we first translate all the sub-operations in GF(28)inversion into logical expressions individually. These sub-operations include the isomorphism function, the 4-bit adder, the square-scaler/squarer/scaler, the GF(24) multiplier, the GF(24) inverter and the inverse isomorphism function with affine transformation. Next, we group and merge some of the sub-operations into several ANF modules before inserting pipeline stages.

The S-box function of an input byte a is defined by two sub steps:

1. Inverse: Let c = a−1, the multiplicative inverse in GF (28) (except if a = 0 then c = 0).

2. Affine Transformation: Then the output is s = M c_ b, where M is a specified 8 × 8 Matrix of bits, b is a specified byte, and the bytes c, b, s are treated as vectors of bits.

First, we use the isomorphism between GF(28) and GF(28)/GF(24) to represent a general element g of GF(28) as a polynomial (in y) over GF(24), of degree 1 or less, as g = γ1y + γ0, with multiplication modulo an irreducible polynomial r(y) = y2 + y γ + μ. Here, all the coefficients are in GF (24). Then the pair [γ 1, γ 0] represents g in terms of a polynomial basis [Y, 1] where Y is one root of r(y).

Second, using GF (24)/GF (22) we can similarly represent GF (24) as linear polynomials (In z) over GF (22), γas = Γ 1z + Γ 0, with multiplication modulo an irreducible polynomial s (z) = z2 + Tz + N, with all the coefficients in GF(22).

Third we use GF (22)/GF (2) to represent GF (22) as linear polynomials (in w) over GF(2), as Γ = g1w + g0, with multiplication modulo t (w) = w + w + 1, where g1, g0 €{0, 1}.

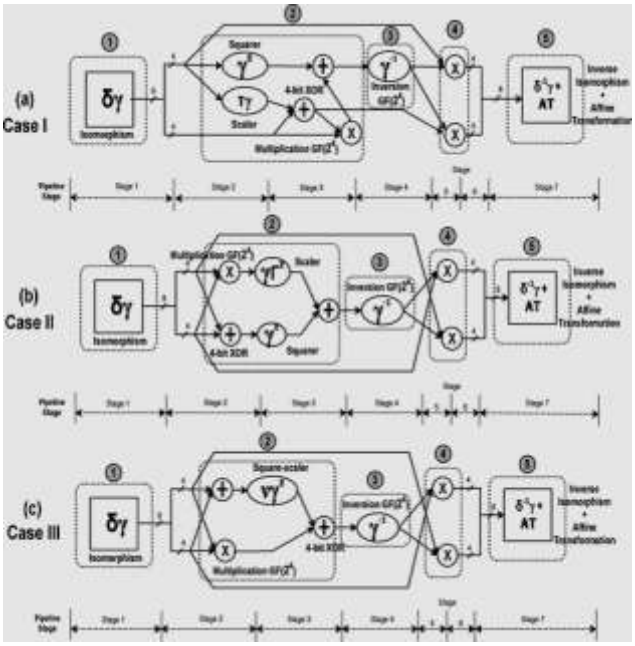

FIG.2. ARCHITEDTURE OF S-BOXES BASED ON THE THREE CASES

In this section author need to mention his simulation/experimental research model with neat block diagrams and flow charts.

5. Simulation/Experimental Results

The 1st case of s-box were designed and implemented on the basis of the algebric normal function and composite field arithmetic. The s-box were simulated with the help of the modelsim software for the VLSI implementation. The simulated waveform of the 1st case of s-box were shown in the figure 3.

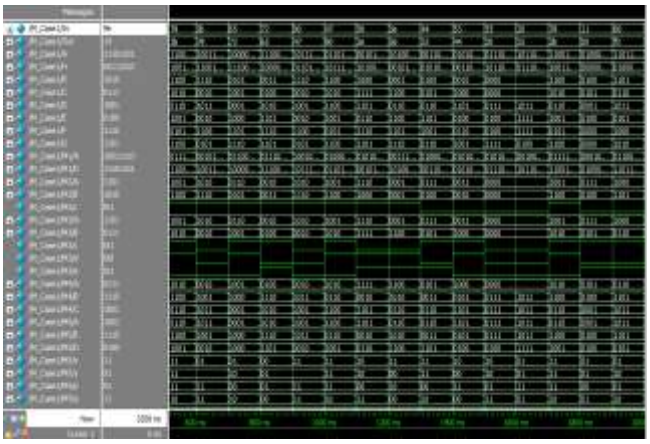

FIGURE .3. THE OUTPUT WAVEFORM OF THE 1ST CASE OF S-BOX

The simulated outputs of the 2nd and 3rd cases of s-boxes were shown below

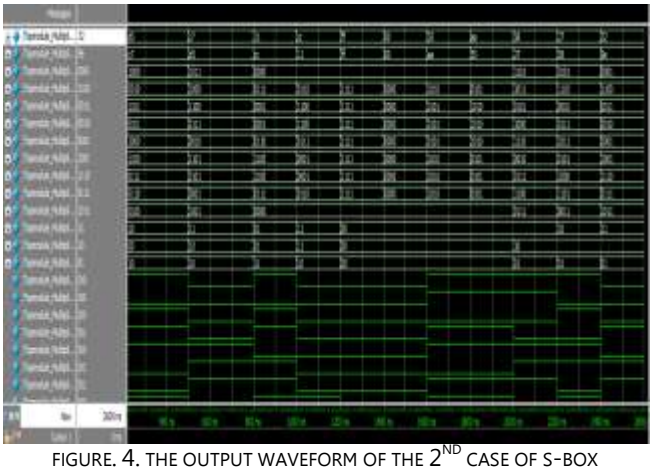

FIGURE. 4. THE OUTPUT WAVEFORM OF THE 2ND CASE OF S-BOX

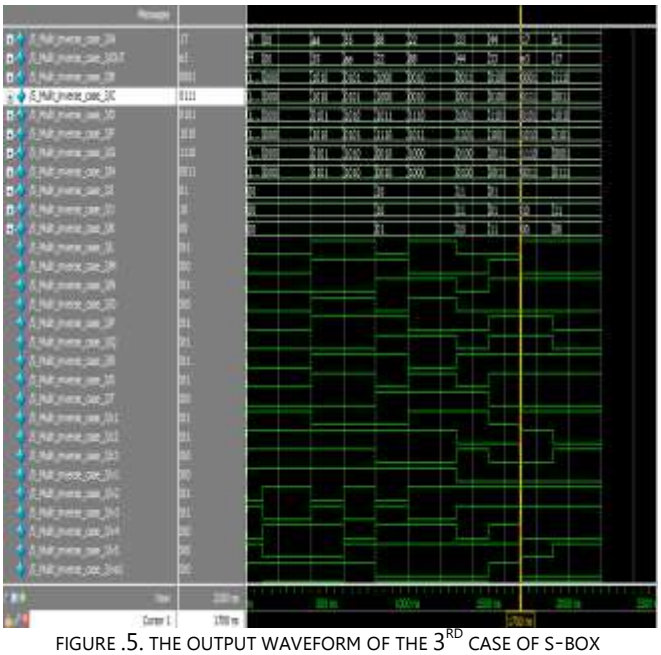

FIGURE .5. THE OUTPUT WAVEFORM OF THE 3RD CASE OF S-BOX

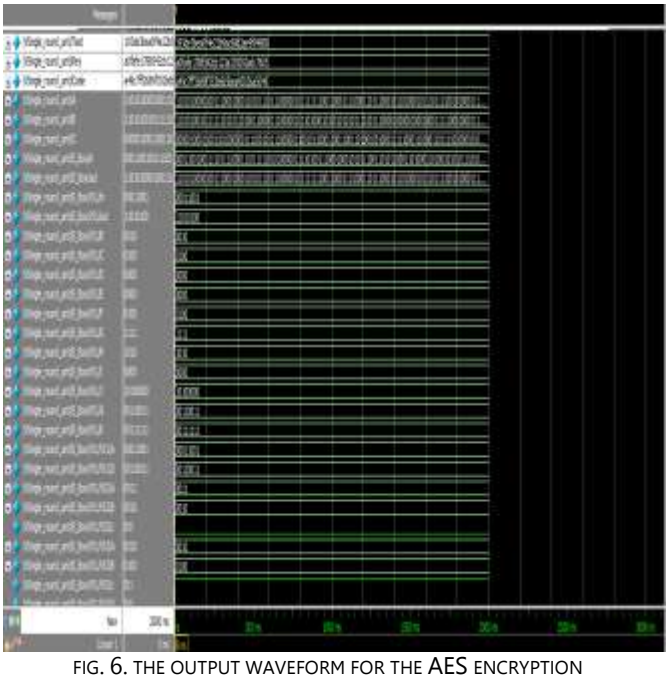

FIG. 6. THE OUTPUT WAVEFORM FOR THE AES ENCRYPTION

- **DEVICE UTILIZATION FOR THE THREE CASES OF S-BOXES**

Topmodule_Multiplicative_Inverse Project Status (02/20/2013 - 11:36:42)			
Project File:	CASEII.xise	Parser Errors:	No Errors
Module Name:	M_MultiplicativeInverse	Implementation State:	Placed and Routed
Target Device:	xc3s400-5pq208	• Errors:	No Errors
Product Version:	ISE 13.2	• Warnings:	2 Warnings (2 new)
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	60	7,168	1%	
Number of occupied Slices	31	3,584	1%	
Number of Slices containing only related logic	31	31	100%	
Number of Slices containing unrelated logic	0	31	0%	
Total Number of 4 input LUTs	60	7,168	1%	
Number of bonded IOBs	16	141	11%	
Average Fanout of Non-Clock Nets	3.60			

Table.1.Denotes the utilization of the number of LUTs and the number of occupied slices in the simulation of the case 1 S-box in the xillinx software.

- **DEVICE UTILIZATION SUMMARY CASE2**

Topmodule_Multiplicative_Inverse Project Status (02/20/2013 - 11:31:48)			
Project File:	CASEII.xise	Parser Errors:	No Errors
Module Name:	Topmodule_Multiplicative_Inverse	Implementation State:	Placed and Routed
Target Device:	xc3s400-5pq208	• Errors:	No Errors
Product Version:	ISE 13.2	• Warnings:	2 Warnings (2 new)
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	41	7,168	1%	
Number of occupied Slices	21	3,584	1%	
Number of Slices containing only related logic	21	21	100%	
Number of Slices containing unrelated logic	0	21	0%	
Total Number of 4 input LUTs	41	7,168	1%	
Number of bonded IOBs	16	141	11%	
Average Fanout of Non-Clock Nets	3.51			

TABLE .2. Device Utilization Summary Of Case2

Table .2. Denote the utilization of the number of LUTs and the number of occupied slices in the simulation of the case 1 S-box in the xillinx software.

- **DEVICE UTILIZATION SUMMARY CASE3**

Topmodule_Multiplicative_Inverse Project Status (02/20/2013 - 11:34:09)			
Project File:	CASEII.xise	Parser Errors:	No Errors
Module Name:	S_Multi_inverse_case_3	Implementation State:	Placed and Routed
Target Device:	xc3s400-5pq208	• Errors:	No Errors
Product Version:	ISE 13.2	• Warnings:	1 Warning (1 new)
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	40	7,168	1%	
Number of occupied Slices	20	3,584	1%	
Number of Slices containing only related logic	20	20	100%	
Number of Slices containing unrelated logic	0	20	0%	
Total Number of 4 input LUTs	40	7,168	1%	
Number of bonded IOBs	16	141	11%	
Average Fanout of Non-Clock Nets	3.25			

TABLE 4.3 - Device Utilization Summary Of Case3

Table.3. Denote the utilization of the number of LUTs and the number of occupied slices in the simulation of the case 1 S-box in the xillinx software.

6. Conclusion

The detailed study on composite field construction for the S-box function in AES was presented. The major contribution of our work was the derivation of a new composite field AES S-box that achieves an optimally balanced construction in terms of area of implementation and critical path, compared to the previous studies. Furthermore, we had explored all of the possible isomorphic mapping for each of the composite field construction and employed a new CSE algorithm to derive the most optimum isomorphic and inverse isomorphic mapping with affine transformation. The three cases of s-box were constructed on the based on the CFA and their output waveforms were verified with the help of MODELSIM software by the Verilog language.

The most optimized third case of s-box was used in the Advanced Encryption Standard (AES) algorithm implementation in the VLSI. AES is used for Encrypt the image or data in communication. Here we encrypt the image or data

by AES with the help of our S-box. The input of AES is image or data pixels, which consist of 4x4 image or 128 bit applied to the both test and reference circuit. For encryption a specific key is required, in this encrypt key also we use 4x4 key or 128 bit data which makes the better performance of the AES.

7. Future Scopes

Our future work is to implement the S-box in AES (Advanced Encryption Standard). AES is used for decrypt the image or data in communication. Here we encrypt the image or data by AES with the help of our S-box. The input of AES is image or data pixels, which consist of 4x4 image or 128 bit applied to the both test and reference circuit. For encryption a specific key is required, in this encrypt key also we use 4x4 key or 128 bit data which makes the better performance of the AES.in future we can implement the decryption algorithm in AES for decrypting the data from cipher text.

References

[1] SMathew, F. Sheikh, A. Agarwal, M. Kounavis, S. Hsu, H. Kaul, M. Anders, and R. Krishnamurthy, "53 Gbps native GF (24)2 composite- field AES-encrypt/decrypt accelerator for content protection in 45 nm high-performance microprocessors," in Proc. IEEE Symp. VLSI Circuits (VLSIC), 2010, pp. 169–170.Name of Auhtors, "Title of the research", Citation Details, year.

[2] Rijmen, "Efficient implementation of the Rijndael S-box," 2000. [Online].Available: at the following website http://ftp.comms.scitech.susx.ac.uk/fft/crypto/rijndael- sbox.pdf.

[3] A. Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi, "Efficient rijndael encryption implementation with composite field arithmetic," in Proc. CHES, 2001, pp. 171–184.

Author's Profile

MOHANRAJ .S has received his Bachelor of Technology degree in Electronics & Communication Engineering in CCET, Puducherry in the year 2011 and Master Of Engineering in Embedded System Technologies in KCET, Cuddalore in the year 2013. At present I am working as a assistant professor in the ECE department of alpha college of engineering and technology, Puducherry. His area of interests were VLSI Design, Embedded System and Digital Electronics.