A new approach for highly throughput AES encryption method on FPGA

Ashish kumar Agrawal, M. tech scholar, MIT, Bhopal, ashish.hcet@gmail.com
Prof. Santosh Yadav, EC Department, MIT, Bhopal/RGPV Bhopal MP

Abstract: The significance of the security problems is greater in current data networks than in earlier systems because users are provided with the way to accomplish very critical operations like banking transfer and sharing of confidential business data, which require very high levels of protection. Weak security architectures allow successful eavesdropping (unauthorised attack), message tampering and modification attacks to occur, with huge consequences for end companies, users and other departments. The Advance Encryption standard (AES) block encryption present at the core of the f8 data hiding algorithm and also the f9 data reconstructing algorithm for Universal data Telecommunications System networks. The design aim is to enhance the data conversion rate means the throughput to an appropriate value hence the design can be used as a cryptographic sub-processor in very high speedy network uses. The work is to design an optimised solution for secure data communication AES is the standard encryption technique but proposed work is more optimised solution for the same when the chip area and encryption time considers as design parameters. Thesis work describe a new method for the Sbox-8 AES substitution and Key gen approach.

I. Introduction

AES is a symmetric block cipher[5]. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in a number of ways. The algorithm Rijndael allows for a variety of block and key sizes and not just the 64 and 56 bits of DES' block and key size[5]. The block and key can in fact be chosen independently from 128, 160, 192, 224, 256 bits and need not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES- 256 respectively. As well as these differences AES differs from DES in that it is not a feistel structure[5]. Recall that in a feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. In this case the entire data block is processed in parallel during each round using substitutions and permutations[14]. A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key[114]. This description of the AES algorithm therefore describes this particular implementation[14]. Rijndael[5] was designed to have the following characteristics: Resistance against all known attacks[5]. Speed and code compactness on a wide range of platforms[5]. Design Simplicity[5].

II. Literature Survey

Hassen Mestiri, Noura Benhadjyoussef, Mohsen Machhout and Rached Tourki published paper An FPGA Implementation of the AES with Fault Detection Countermeasure in IEEE year 2013, In this paper, they present the implementation details for the AES 128-bit cipher and decipher. They conduct a fault injection attack test against the unprotected AES. Also they proposed a new fault detection scheme for the AES. Their simulation result shows that the fault coverage can be achieves up to 99.998%. Their protected AES has been implemented on Xilinx Virtex-5 FPGA. Their fault coverage, frequency degradation, area overhead and throughput have been compared old work in same area and they shown that their proposed fault detection scheme allows a dealing between the ability to detect faults and the implementation cost for the AES. To improve the security of the AES, they implemented the AES algorithm in encryption and decryption both. They conduct a fault attack against the unsecure AES by using MATLAB. And they proposed a new fault detection scheme for the Advanced Encryption Standard. They also discuss the strengths and the weak points of this scheme against the fault attacks.

Dr.R.V. Kshirsagar, M.V.Vyawahare, published a paper FPGA Implementation of High speed VLSI Architectures for AES Algorithm in IEEE year 2012, In this paper, they have proposed high data throughput in AES hardware architecture by partition of ten rounds into sub-blocks of repeated AES blocks. The blocks then separated by intermediate storing buffers which provides full ten stages of AES pipeline module. Also, the AES is internally equally divided into ten pipeline stages, with the some additional feature that the shift rows block (*Shift Row)* is structured to performed before the byte substitute (SBox - Substitute Byte) block. This proposed new swapping operation makes no effect on the AES encryption algorithm but, it smoothly process four blocks of data in parallel rather than 16 blocks, which can be considered as the key advantage for area saving. They have evaluated the performance of their implementation in terms of overall throughput rate and hardware area for Xilinx SPARTAN-3E FPGA. Their simulation results shows that their proposed AES has higher throughput rate of about 4.25% than the normal AES pipeline structure with a saving hardware area upto 56%.

III. Proposed Methodology

As discuss above AES requires lots of computation in four modules (Substitution block, shift rows, mix columns &Add round Key) of nine Full and one Sub-round. But it can be easily observe that Optimization in Area and speed possible only with block Key-Generator and S-Box only. Thesis works on new optimized S-box though Key-Generator technique remains unchanged.

Table 5.2 below shows the input and output of AES S8-Box, and the clear and complete observation gives the clue for the optimization in AES.

Substitution Box S-Box-8	
INPUT	OUTPUT
0000_0000	1111_1111
0000_0001	0111_1100
0000_0010	0111_0111
. . . .	. . . .
1111_1110	0111_0001
1111_1111	1111_0000

Table 1: Substitution Box 8 contains

Table shows the relation between input and output for s8 box (f8).Observation from table was that as for small size S-box (2-5 bit), memory based S-box is better area optimized and for bigger S box(more than 5 bit) Combinational architecture is better area optimized. Proposed work is a combination of memory and combinational architecture. The table show is relation between input and output for 8 bit S-box, here thesis proposed architecture divided the total range 0-255 into 16 sub-ranges (0-15,16-31,32-47,48-63,64-79,80-97,96-111,112-127,128-143,144 159, 160-175,176-191, 192-207,208-223,224-239, 240-255) isolation shown by orange lines. For each sub-range, upper four MSB of output (separated by Red line) are generated using 4 input K-map and lower four LSB of output are generated using Memory architecture. Figure 5.5 above shows the architecture of proposed work which reflects the idea behind the new logic for architecture as explain above.

As the proposed work uses the S8-box explained above, and as proposed s8box is area and time efficient and as very much known S8-box use in AES one round about six time (i.e. one time in substitution, four times in Mix coulombs, one time in round key generation), and total nine full rounds requires 9x6 = 36 time use of s8-box and 2 times in sub-round which makes total 38 times use of s8-box in single plaintext to cipher-text generation. So If proposed S8-box is optimized in terms of area and speed the Full

AES is also optimized as S8-box gets in use 38 times.

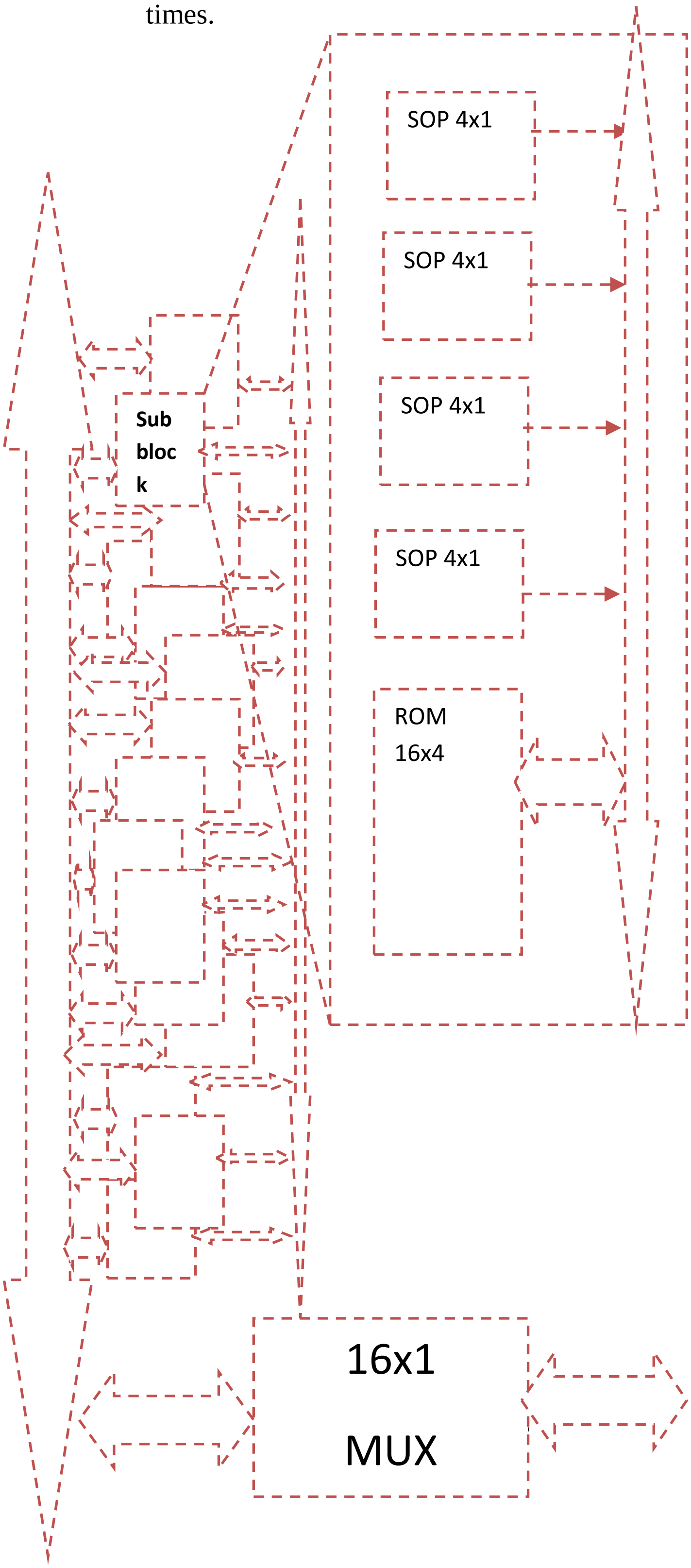
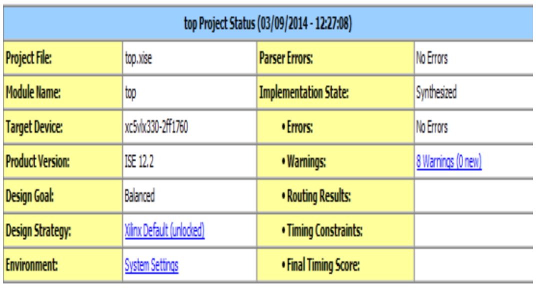


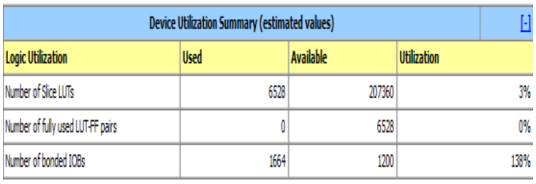
Figure 1: Proposed S-box

IV. Results

Synthesis Summary of Full AES



top Project Status (03/09/2014 - 12:27:08)			
Project File:	top.xise	Parser Errors:	No Errors
Module Name:	top	Implementation State:	Synthesized
Target Device:	xc5vlx330-2ff1760	• Errors:	No Errors
Product Version:	ISE 12.2	• Warnings:	8 Warnings (0 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	6528	207360	3%
Number of fully used LUT-FF pairs	0	6528	0%
Number of bonded IOBs	1664	1200	138%

Timing Results

S8box
Total 7.741ns (5.970ns logic,
Shifter
Total 4.781ns (4.468ns logic,
Mix column
Total 6.436ns (4.800ns logic,
Sub Round
Total 8.359ns (6.136ns logic,
Round
Total 9.465ns (6.627ns logic,
Key scheduler
Total 45.090ns (21.966ns logic,
Full AES Design
Total 27.410ns (8.394ns logic,

Comparative results
Full AES design

		[1] Dr. R.V. K. shirsagar, IEEE, 2012		
	OUR	General AES without pipelining	Fully pipelining AES	Fully pipelining with 10 sub pipelining AES
No. Of LUT	6528	--	--	--
Logical Delay	8.394 ns	1,150.970 ns	116.867 ns	111.890 ns
Max frequency	220 MHz	--	--	--

V. Conclusion

An optimised and compact hardware design of the AES algorithm has been described in this thesis work, as well as with the results of its implementation in FPGA technology. These proposed S8-box method might be use to design high performance compact implementations of Feistel-like block ciphers (AES, IDEA etc.). Not only does this proposal achieve a high performance, but is one of the most cost efficient designs in terms of area.

It can be concluded as discuss that S8-box is an important requirement in AES cipher generation and it get use 38 times for generating 64 bit cipher-text from plaintext. proposed S8-box 7.741 ns time delay and only 64 slices, which is less as compare to all existing works.

References

[1] Hassen Mestiri, Noura Benhadjyoussef, Mohsen Machhout and Rached Tourki, An FPGA Implementation of the AES with Fault Detection Countermeasure,IEEE,2013

[2] Dr.R.V. Kshirsagar, M.V.Vyawahare, FPGA Implementation of High speed VLSI Architectures for AES Algorithm, IEEE, 2012

[3] Shylashree.N, Nagarjun Bhat and V. Shridhar3FPGA IMPLEMENTATIONS OF ADVANCED ENCRYPTION STANDARD: A SURVEY, IJAET, 2012

[4] Mr. Atul M. Borkar, Dr. R. V. Kshirsagar, Mrs. M. V. Vyawahare FPGA Implementation of AES Algorithm, IEEE, 2011

[5]Rijndael (by Joan Daemen & Vincent Rijmen)A Specification for Rijndael, the AES Algorithm, Dr. Brian Gladman, v3.16, 1st August 2007

[6] http://www.xilinx.com/support.html