Resource Efficient Implementation of S-Box Based on Reduced Residue of Prime Numbers using Virtex-5 FPGA

Muhammad H. Rais, *Member, IAENG* and Syed M. Qasim, *Member, IAENG*

***Abstract*—A high performance substitution box (S-Box) implementation using reduced residue of prime numbers is presented in this paper. The byte substitution implemented using S-Box is an important part of the Advanced Encryption Standard (AES). The objective of this paper is to present an efficient Field Programmable Gate Array (FPGA) realization of S-Box using very high speed integrated circuit hardware description language (VHDL). The design was implemented on Xilinx Virtex-5 XC5VLX50 FPGA and the results obtained show that the proposed method provides an improved performance with 38% improvement in maximum clock frequency as well as efficient utilization of FPGA hardware resources.**

***Index Terms*—Advanced Encryption Standard (AES), S-Box, FPGA, VHDL, Virtex-5.**

I. Introduction

Cryptographic algorithms play a critical role in the transmission of sensitive electronic financial transactions and digital signature applications. Over the fast and insecure digital communication networks cryptographic algorithms are used to offer secrecy, integrity, and non-reproduction of exchanged information. A promising solution that combines high flexibility with the speed and physical security of traditional hardware (application specific integrated circuits (ASICs)) is the implementation of cryptographic algorithms on field programmable gate array (FPGA) [1].

FPGA offers math functions, embedded memories and storage elements, so that the design of cryptography becomes easier. This provides a cheap solution for designing and implementing various cryptographic algorithms on FPGA. The implementation of security protocols on FPGA leads to several advantages including low cost, availability of sophisticated design and verification tools, ability of in-circuit reprogrammability and short time to market.

The Advanced Encryption Standard (AES) [2] is a block cipher adopted as an encryption standard by the U.S. government. AES was developed by Belgian researchers Vincent Rijmen and Joan Daemen and subsequently named as Rijndael cipher algorithm [3]. AES consists of 128 block length of bits and supports 128, 192 and 256 key length bits. This algorithm starts with initial transformation of state matrix followed by nine iteration of rounds. A round consists of four transformations: Byte Substitution (SubBytes), Row Shifting (ShiftRows), Mixing of columns (MixColumns) and followed by addition of Round Key called (AddRoundKey). From each round, a round key is generated from the original key through key scheduling process. The last round consists of SubBytes, ShiftRows and AddRoundKey transformation. SubBytes transformation is implemented using S-Box, which is computationally intensive and consumes more than 75% of FPGA resources [4].

The S-Box is based on the Galois Field GF (2^8), and it is the only non-linear component of the AES algorithm which provides confusion capability [5]. The new approach reported in [6] uses residue of prime numbers, which adds more confusion than the normally used Galois Field GF (2^8). S-Box based on Galois Field GF (2^8) is constructed by performing two transformations; first taking a multiplicative inverse in the Galois Field GF (2^8) and then applying a standard affine transformation over Galois Field GF (2^8). The S-Box is one of the most time consuming process because it is required in every round [7]. There are other S-Box based fast and memory efficient algorithms that have been reported [4, 8-12], but these reported methods are insecure and potential threat to the security of the data [6]. Other issues pertaining to this which were not given due importance is to accelerate the process, and in order to do that, a reduced residue of prime numbers can be utilized, which results in table entries similar to S-Box based on Galois Field GF (2^8) [6]. The objective of this paper is to present a comparison between reduced residue of prime number and Galois Field GF (2^8) based S-Boxes. In this paper, a new S-Box algorithm which provides more confusion based on reduced residue of prime numbers is coded using VHDL and targeted to Xilinx Virtex-5 FPGA.

The rest of the paper is structured as follows: Section II present the details of AES algorithm. S-Box design using residue of prime number and its reduced version is described in section III. Section IV presents the target FPGA technology. The FPGA implementation results and concluding remarks are provided in sections V and VI respectively.

Manuscript received March 23, 2010. This work was supported in part by the Research Center, College of Engineering, King Saud University under Grant no. 39/429.

M. H. Rais is with the Electrical Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia (phone: +966-1-467-6801; fax: +966-1-467-6757; e-mail: mhrais@ksu.edu.sa).

S. M. Qasim is with the Electrical Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia (e-mail: smanzoor@ksu.edu.sa).

ISBN: 978-988-18210-7-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Advanced Encryption Standard

The Rijndael algorithm used for the AES is a symmetric block cipher that processes fixed data of 128-bit blocks. It supports key sizes of 128, 192 and 256 bits and consists of 10, 12 or 14 iteration rounds, respectively.

The AES algorithm's internal operations are performed on a two dimensional array of bytes called State. The State consists of 4 rows of bytes and each row has Nb bytes. Each bytes is denoted by $S_{i,j}$ ($0 \le i < 4$, $0 \le j < Nb$). The 128 bits are organized into state matrix which is of the size of 4×4. Therefore, each row of the State contains Nb equals to 4 bytes. The four bytes in each column of the state array form a 32-bit word, with the row number as index for the four bytes in each word. Initially, State is filled with the input data block and XOR-ed with the encryption key. At the start of encryption the array of input bytes is mapped to the State array as shown in Fig. 1. The 128-bit block can be expressed as 16 bytes: in0, in1, in2, …, in15. Encryption process is performed on the State, and then State values are mapped to the output bytes array out0, out1, out2, …, out15.

The AES algorithm is an iterative algorithm and each iteration is called a round. Each round mixes the data with a round key, which is generated from the encryption key. Fig. 2 presents AES algorithm structure with round operations. As shown in Fig. 2, each of the nine rounds consists of four transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey. The last round bypasses MixColumns transformation. SubBytes transformation uses 16 identical 256-byte substitution tables called S-box. SubBytes can be implemented either by computing the substitution or using look-up-table (LUT).

The S-Box transformation is a computationally intensive and important operation of the AES. Table I represents the S-Box based on Galois Field GF (2^8), and is the only non-linear component of the AES algorithm, which provides confusion capability [5]. The approach reported in [6] uses residue of prime numbers which adds more confusion than the normally used Galois Field GF (2^8).

III. S-Box using Reduced Residue of Prime Numbers

The S-Box based on residue of prime number is a complete S-Box with 256 entries. The entries shown in Table II are the residue of the prime number 257. The row and column headers of Table II are hexadecimal digits. The Table III shows the reduced version of Table II. As presented in Table III, the eliminated half entries are unknown, so it creates more confusion to the S-Box implementation which is not present in Galois Field GF (2^8) based S-Box.

Now if we look at Table II with couple of lookup operations, if S (F, A) = 6E and S (6, E) = FA. This implies that both numbers are inverse of each other and are stored on the table. Since every double digits hexadecimal number and its inverse are stored, it is logical to reduce the table by eliminating half of the numbers and their inverses. Therefore, the reduced table contains 128 entries or 50% of Table II.

Hence, not every lookup operation will be successful on Table III. In order to, exactly find out the value if it comes with value then one operation is necessary; otherwise, a miss requires two operations: following steps are required to lookup the table.

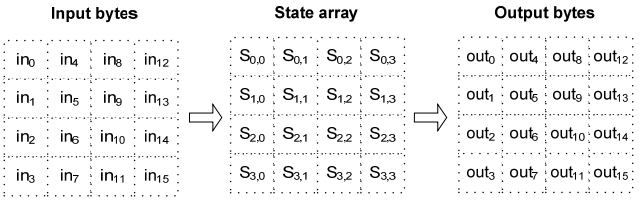

Fig. 1 Mapping of Input bytes, State array and Output bytes

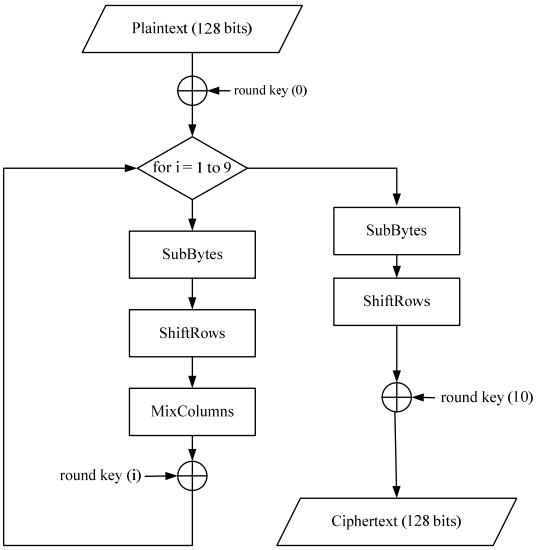

Fig. 2 AES Algorithm Structure

1) If S (F, A) returns a number then that number is the inverse of FA and the second step is not required.
2) Use the returned row and column headers as the most and least significant digits respectively of the inverse of FA.

Research efforts in this area are underway where researchers are trying to implement other fast and efficient algorithms to generate S-Box [4-5, 8].

IV. Target Technology

The Virtex-5 device built on a 65 nm state-of-the-art copper process technology is used as the target technology for implementing S-Box. The Virtex-5 comprises: hard-IP system-level blocks, including Block RAM/first in first out (FIFO), second generation 25×18 DSP slices, SelectIO technology with built-in digitally-controlled impedance, ChipSync source-synchronous interface blocks, enhanced clock management tiles with integrated DCM and phase locked loop (PLL) clock generators, and advanced configuration options.

In addition to the regular programmable functional elements, Virtex-5 family provides power-optimized high speed serial transceiver blocks for enhanced serial connectivity, tri-mode Ethernet MACs and high-performance PPC 440 microprocessor embedded blocks. Virtex-5 devices also use triple-oxide technology for reducing the static power

ISBN: 978-988-18210-7-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table I S-Box based on Galois Field GF (2^8)

	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	01	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	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	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	06	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	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	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

Table II S-Box based on Residue of Prime Number 257

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
0	00	01	81	56	c1	67	2b	93	e1	c8	b4	bb	96	b2	ca	78
1	f1	79	64	e6	5a	31	de	be	4b	48	59	ee	65	c3	3c	c7
2	f9	94	bd	eb	32	84	73	91	2d	a3	99	06	6f	28	5f	af
3	a6	15	24	7e	ad	61	77	f3	b3	f8	e2	3d	1e	3b	e4	66
4	fd	57	4a	ea	df	95	f6	b5	19	a9	42	18	ba	f7	c9	f4
5	97	a5	d2	60	cd	7f	03	41	b8	1a	14	d1	b0	98	d8	2e
6	53	35	8b	87	12	1c	3f	05	d7	a4	b1	f5	bc	e0	fa	2c
7	da	74	7c	26	71	86	9f	36	0f	11	9e	8c	72	dc	33	55
8	ff	02	ac	ce	25	8f	75	63	f0	f2	cb	62	7b	90	db	85
9	8d	27	d5	07	21	45	0c	50	5d	2a	fc	c2	e5	ef	7a	76
a	cc	ae	d3	29	69	51	30	ed	e7	49	c0	fe	82	34	a1	2f
b	5c	6a	0d	38	0a	47	e9	bf	58	e8	4c	0b	6c	22	17	b7
c	aa	04	9b	1d	c6	e3	c4	1f	09	4e	0e	8a	a0	54	83	dd
d	ec	5b	52	a2	d9	92	fb	68	5e	d4	70	8e	7d	cf	16	44
e	6d	08	3a	c5	3e	9c	13	a8	b9	b6	43	23	d0	a7	1b	9d
f	88	10	89	37	4f	6b	46	4d	39	20	6e	d6	9a	40	ab	80

Table III S-Box based on Reduced version of Residue of Prime Number 257

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
0	00	01	81	56	c1	67	2b	93	e1	c8	b4	bb	96	b2	ca	78
1	f1	79	64	e6	5a	31	de	be	4b	48	59	ee	65	c3	3c	c7
2	f9	94	bd	eb	32	84	73	91	2d	a3	99		6f		5f	af
3	a6			7e	ad	61	77	f3	b3	f8	e2	3d			e4	66
4	fd	57	4a	ea	df	95	f6	b5		a9			ba	f7	c9	f4
5	97	a5	d2	60	cd	7f			b8			d1	b0	98	d8	
6				87					d7	a4	b1	f5	bc	e0	fa	
7	da	74	7c			86	9f				9e	8c		dc		
8	ff		ac	ce		8f			f0	f2	cb	62		90	db	
9			d5								fc	c2	e5	ef		
a	cc	ae	d3					ed	e7		c0	fe				
b							e9	bf		e8						
c					c6	e3										dd
d	ec				d9		fb									
e																
f																

ISBN: 978-988-18210-7-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

consumption. Their 1.0 V core voltage and 65 nm implementation process leads also to dynamic power consumption reduction as compared to Virtex-4 devices.Virtex-5 family uses a new diagonally symmetric interconnects to minimize the number of interconnects [13].

V. Implementation Results

The design of Galois Field GF (2^8) and reduced residue of prime number based S-Box is done using VHDL and implemented in a Xilinx Virtex-5 XC5VLX50 (package: ffg676, speed grade: -1) FPGA using the ISE 9.2i design tool and the performance of the proposed design is evaluated based on the FPGA implementation results. Fig. 3 shows the block diagram of S-Box. Fig. 4 presents the part of RTL schematic of S-Box using reduced residue of prime numbers. FPGA layout of S-Box using reduced residue of prime number is shown in Fig. 5. Fig. 6 and 7 show the RTL schematic and FPGA layout using Galois Field GF (2^8) based S-Box. Table IV presents the FPGA implementation results of both the designs. Compared with the design using Galois Field GF (2^8), reduced residue of prime number based S-Box operates at a maximum clock frequency of 512.821 MHz and shows an improvement of 38%. The proposed design utilizes only 31 slices of Virtex-5 FPGA as compared to 2 slices and 1 block RAM (BRAM) used in Galois Field GF (2^8) based S-Box design.

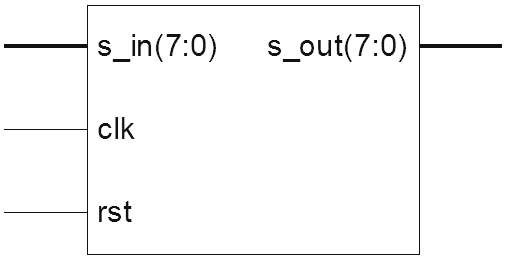

Fig. 3 Block Diagram of S-Box

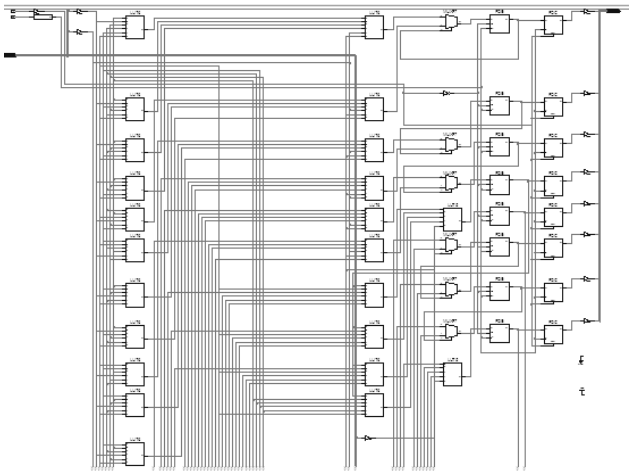

Fig. 4 Part of RTL Schematic of S-Box using Reduced Residue of Prime Numbers

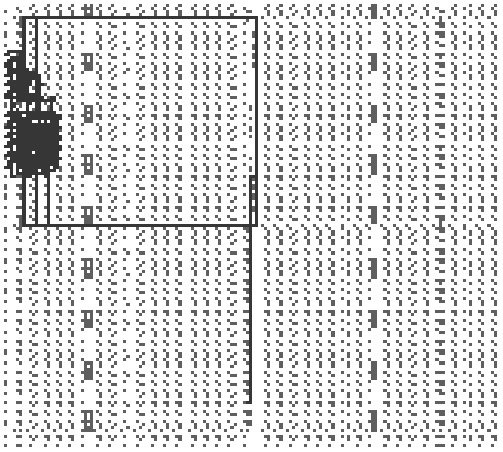

Fig. 5 FPGA layout of S-Box using Reduced Residue of Prime Numbers

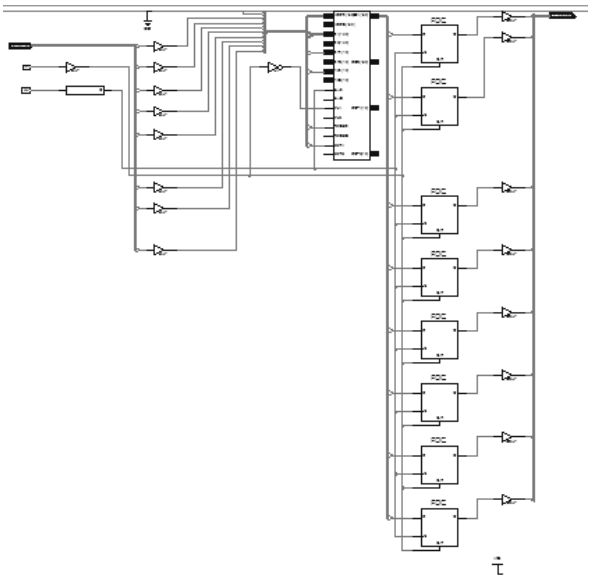

Fig. 6 Part of RTL Schematic of Galois Field GF (2^8)based S-Box

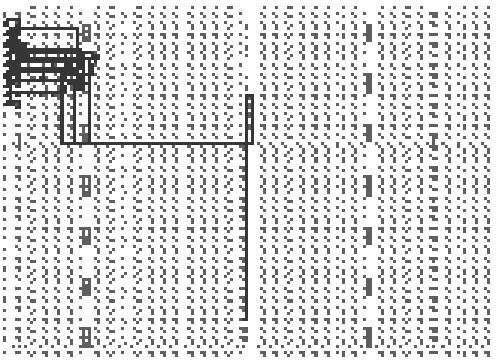

Fig. 7 FPGA layout of Galois Field GF (2^8) based S-Box

ISBN: 978-988-18210-7-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

VI. Conclusion

An efficient S-Box design based on the reduced residue of prime numbers has been presented here. The proposed design is implemented in Xilinx Virtex-5 XC5VLX50 FPGA and the results are compared with that of Galois Field GF (2^8). The new reduced version shows promising results which could be used in AES to increase its complexity and add more confusion in order to provide further resistance against algebraic attacks.

Table IV Performance evaluation of a Galois Field GF (2^8) and Reduced Residue of Prime Number based S-Box design

	Galois Field GF (2^8) / Residue of Prime Numbers	Reduced Version of Residue of Prime Numbers
Frequency (MHz)	371.609	512.821
Period (ns)	2.691	1.950
BRAMs	1	Zero
Equivalent Gate Count	65600	786
Occupied Slices	2	31

References

[1] S. Mangard, M. Aigner and S. Dominikus, "A highly regular and scalable AES hardware architecture", *IEEE Trans. Computers*, vol. 52, no. 4, 2003, pp. 483-491.
[2] Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, 2001.
[3] J, Daemen and V.Rijmen, "The design of Rijndael AES-The advanced encryption standard", Springer, 2002.
[4] A. Aziz and N. Ikram, "Memory efficient implementation of AES S-boxes on FPGA", *J. Circuits, Systems, and Computers*, vol. 16, no. 4, 2007, pp. 603-611.
[5] M. T. Tran, D. K. Bui and A. D. Duong, "Gray S-Box for Advanced Encryption Standard", *in Proc. of International Conference on Computational Intelligence and Security*, vol. 1, pp. 253-258, 2008.
[6] E. S. Abuelyman and A. A. S. Alsehibani, "An optimized implementation of the S-Box using residue of prime numbers", *Inter. J. Computer Science and Network Security*, vol. 8, no. 4, 2008, pp. 304-309.
[7] I. Harvey, "The effects of multiple algorithms in the Advanced Encryption Standard", nCipher Corporation Ltd., 2000.
[8] F. R-. Henriquez, N. A. Saqib and A. D-. Perez, "4.2 Gbits/s single chip FPGA implementation of AES algorithm", *Elect. Lett*, vol. 39, no. 15, 2003, pp. 1115-1116.
[9] I. A-. Badillo, C. F-. Uribe and R. C-. Para, "Design and implementation of an FPGA-based 1.452 Gbps non pipelined AES architecture", *in Proc. of the International Conference on Computational Science and its applications*, Lecture Notes in Computer Science, Springer-Verlag, Vol. 3982, pp. 446-455, 2006.
[10] J. Zambreno, D. Nguyen and A. Choudhary, "Exploring area/delay tradeoffs in an AES FPGA implementation", *in Proc. of International Conference on Field Programmable Logic and its Applications*, Lecture Notes in Computer Science, Springer-Verlag, Vol. 3203, pp. 575-585, 2004.
[11] D. S. Kundi, S. Zaka, Q. Ain and A. Aziz, "A compact AES encryption core on Xilinx FPGA", *in Proc. of 2nd International Conference on Computer, Control and Communication*, pp.1-4, 2009.
[12] M. Li, G. Dai, H. Liu, and W. Hu, "Design of an instruction for fast and efficient S-Box implementation", *in Proc. of International Conference on. Computational Intelligence and Security*, pp. 623-626, 2007.
[13] Xilinx, Virtex-5 FPGA family overview, 2009. http://www.xilinx.com/support/documentation/virtex-5.htm

ISBN: 978-988-18210-7-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)