International Journal of Security and Its Applications
Vol. 10, No. 7 (2016) pp.11-24
http://dx.doi.org/10.14257/ijsia.2016.10.7.02

Design of Low Power and Secure Implementation of SBox and Inverse-SBox for AES

Divya Sharma[1], Ankur Bhardwaj[2], Harshita Prasad[3], Jyoti Kandpal[4], Abhay Saxena[5], Kumar Shashi Kant[6] and Gaurav Verma[7]

[1,2,7]*Department of Electronics & Communication, JIIT-Noida (U.P.)-India.*
[3,4]*Department of Electronics & Communication, UTU-Dehradun (U.K.)-India.*
[5]*Department of Computer Science, DSVV-Haridwar (U.K.)-India.*
[6]*Department of Electronics & Communication, SIT-Pune-India.*

Abstract

In the cutting edge world, data security has turned into an essential issue furthermore the innovation is going to increment quickly. In this paper, the symmetric key standard for encryption and decoding is propelled Encryption standard (AES). The key stride in the AES is the "S-Box". S Box is an imperative segment for symmetric key calculations. An S-box takes some number of information bits "p" and interprets them in yield bits 'q', where "p" is not as a matter of course equivalent to 'q'. In AES Encryption calculation Sub Bytes change uses S-Box and Inverse S-Box uses Inverse of S-Box. The Sub Bytes substitution is a nonlinear byte substitution that uses substitution table (i.e., S-Box) takes the multiplicative reverse (GF (2^8)) and infers a relative change to do the Sub Bytes change. Though, converse Sub Bytes Substitution additionally uses gaze upward table (i.e., Reverse S-Box) takes an opposite relative change and after that suggests multiplicative backwards of Galois Field (GF (2^8)). In this printed material, we investigated substitution table/reverse substitution table, multiplicative opposite and relative change and its converse (i.e. reverse relative change) science in Galois field. A standout amongst the most basic issues in AES is the force utilization. Here, we predominantly centered around the force utilization and in addition security of S-box which is the most power devouring square in the AES. We have executed and reproduced S-Box and Inverse S-Box Lookup table and acquired another improved scrambled Lookup table for more upgraded mystery by utilizing Xilinx Spartan-3 assessment board. The Simulation and execution instruments utilized are Xilinx ISE 14.1i and ModelSim 6.0.

Keywords*: S-BOX/Inverse S-Box, AES, Power Analysis, FPG), VHDL*

1. Introduction

There is expanding need of information data in Computer Network and Communication Technology. This data is dealt with by open frameworks and it is feeble. Encryption is the change of data into a structure that is as close vast as could sensibly be relied upon to examine without the reasonable learning. Its motivation is to guarantee security by keeping data secure from unauthorized access, even the individuals who have entry to the scrambled information. Decoding is the opposite of encryption; it is the transformation of encoded information once more into a clear outline. Encryption and decryption for the most part require the utilization of some anonymous data, termed as a key. For some encryption parts, the same key is utilized for both Encryption and unscrambling; for different systems, the keys utilized for encryption and interpreting is unmistakable. There are innumerous encryption calculations that are in a matter of moments as often as possible utilized as a bit of figuring, yet the U.S. government has gotten a handle on the Advanced Encryption Standard (AES) to be utilized by Federal divisions and working environments for securing delicate data.

ISSN: 1738-9976 IJSIA

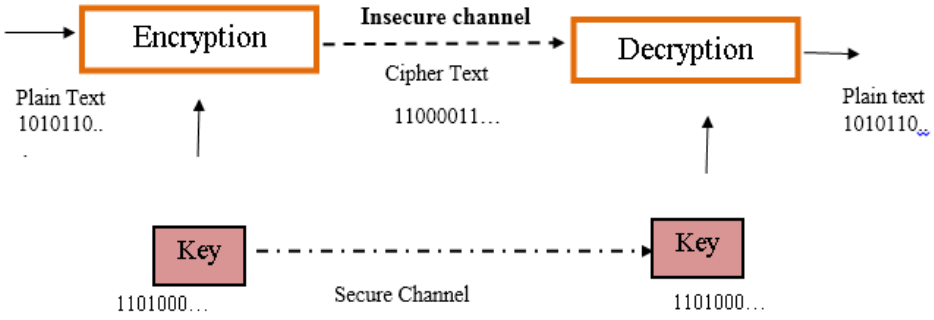
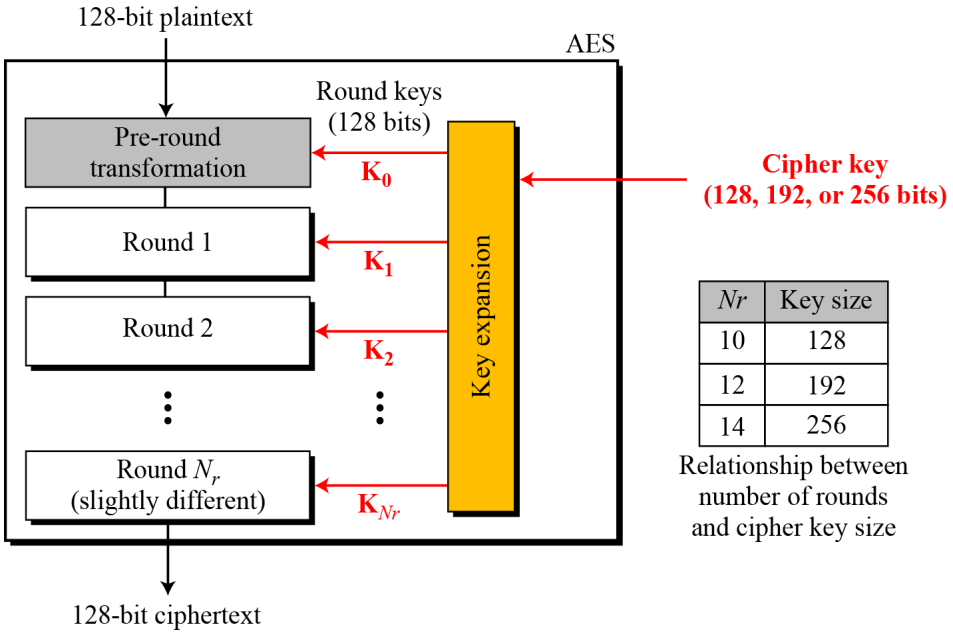


Figure 1. Symmetric Key Cryptography

By using a key, a plaintext can be transformed into a ciphertext and vice-versa. There are certain cipher exists that needn't bother with a key by any means. Delineation is direct Caesar-assume that mists content by supplanting every memo with the memo thirteen spots down in the memo set. Taking after our letter set has 26 characters; it is adequate to encode the figure message again to recoup the principal message. The converse procedure of encryption is second process which is the decoding; here Cipher content will be changed over into Plain content utilizing all the opposite strides connected for encryption. The National Institute of Standards and Technology (*i.e.,* NIST) have distributed the particulars of this encryption standard in Federal Information Processing Standards (*i.e.,* FIPS) Publication [4]. Any ordinary symmetric figure, for example, AES, requires a singular key for both encoding and decoding, which is autonomous of the plaintext and the figure itself. It ought to be illogical to recover the plaintext exclusively in view of the Cipher content and the encryption calculation, without knowing the encryption key. In this way, the mystery of the encryption key is of high significance in symmetric figures, for example, AES. Programming execution of encryption calculations does not give extreme mystery of the key following the working framework, on which the encryption programming runs, is constantly powerless against assaults.

2. The AES Algorithm

The Advanced Encryption Standard, in the running with referenced as AES, is the victor of the test, held by the US Government in 1997, after the Data Encryption Standard was discovered irrationally feeble in context of its minimal key size and the mechanical developments in processor power. Fifteen candidates were perceived in 1998 and considering open remarks the pool was diminished to five finalists in 1999. In October 2000, one of these five computations was picked as the unavoidable standard: a barely changed sort of the Rijndael. The Rijndael, whose name in light of the names of its two Belgian pioneers, Joan Daemen and Vincent Rijmen, is a Block figure, which derives that it takes a shot at changed length social affair of bits. It takes an information square of a specific size, frequently 128, and produces a differentiating yield bit of the same size. The change requires a second data that is the question key. We understand that the mystery key can be of any size (ward upon the piece utilized) and that AES utilizes three specific key sizes: 128, 192 and 256 bits. While AES strengthens basically key sizes of 128, 192 and 256bits and square sizes of 128 bits, the primary Rijndael underpins key and piece sizes in any different of 32, with no under 128 and a most extraordinary of 256 bits as showed up in Figure 2.



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Figure 2. General Design of AES Encryption Cipher [8]

Throughout each encircling, the following alteration are applied on the status:

1. Sub Bytes: each byte in the status is substituted by another, by means of the Rijndael S-Box.

2. Shift Row: each line in the 4x4 cluster is moved an unmistakable amount to one side**.**

3. Blend Column: an immediate change on the sections of the status.

4. AddRoundKey: every byte of the status is joined with a encircling key, which is a various key for each encircling and got from the Rijndael key calendar.

3. Encryption and Decryption

The Encoding and Decoding process of AES algorithm is presented below, in Figure 3.

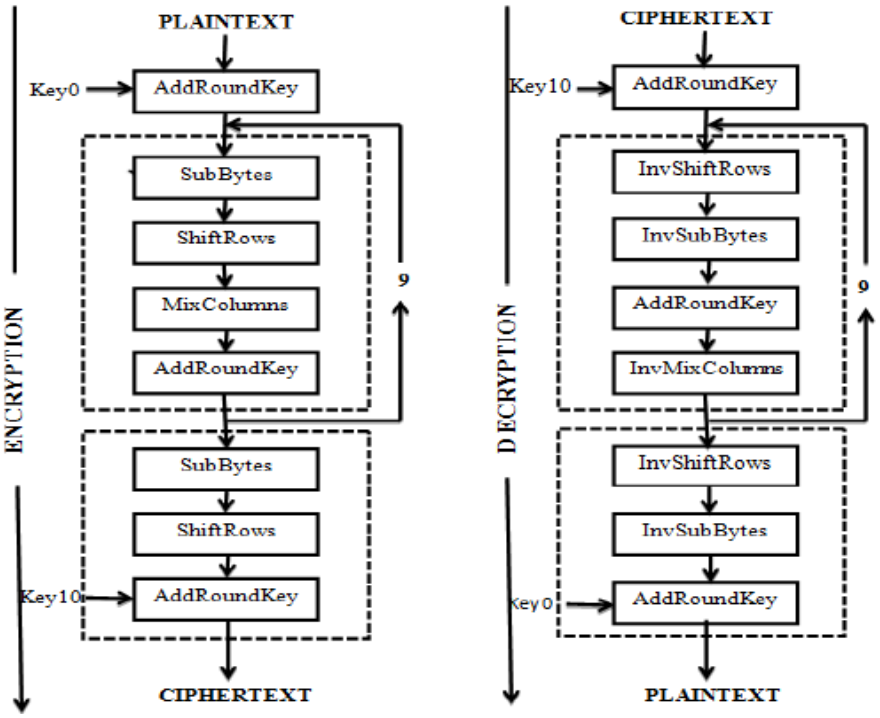


Figure 3. AES Process [4]

This square outline is standard for AES details. It contains various different changes connected successively over the information piece bits, in a settled number of emphases, called rounds. The quantity of rounds relies on upon the length of the key utilized for the encryption and decoding process.

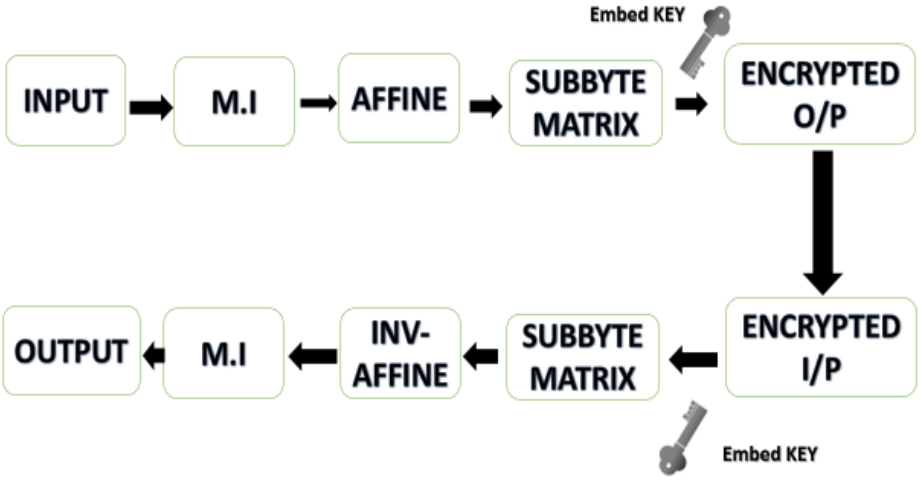


Figure 4. System Process

4. S-Box and Inverse S-Box Architecture

Byte exchange and additionally Inverse Byte exchange is the mainly composite strides in the encoding and decoding forms. In these operations every byte of the state exhibit will be substituted with its proportionate byte in the S-box or the Inverse S-box. As AES calculation using components inside the GF (28), every component in the state exhibit connotes a byte with a worth that varies between 00H-FFH. The S-box has a settled size

of 256 bytes spoke to as (16*16) bytes lattice [7]. SBOX is executed with various styles utilizing two regular methods:

1. look up table (LUT)
2. Composite field arithmetic.

A gaze upward table (LUT) ROM based and computational utilizing combinational rationale doors computational technique is adaptable for rate advancement, and requires littler range.

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	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 1. Look Up Table of S-Box [8]

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Table 2. Look Up Table of Inverse S-Box [8]

SubByte is non-straight change that utilizations 16 byte S-BOX. Each byte is a multiplicative backwards over Galois Field GF (28) took after by a relative change, a complete procedure of SubByte is appeared in Figure 5; Whereas, Inverse Sub Bytes Substitution additionally uses turn upward table (*i.e.*, Opposite S-Box) takes a reverse relative change and after that suggests multiplicative converse of Galois Field (GF (28)) is appeared in Figure 6. m (x) = x8+x4+x3+x+1 is used as an irreducible polynomial for GF (28). ShiftRow is a cyclic moving where first line is predictable with no development and whatever is left of lines are moving reliably with first demand, second demand, additionally, third demand progressively. MixColumn change works at each segment freely and considers it as a polynomial limit over GF (28). Each area is copied by a coefficient framework modulo (x4+1). AddRoundKey is a bitwise XOR between figures subkeys which are removed from essential key through key improvement methodology and 128 piece information block. In AES calculation, SubByte disseminates the biggest part of force. Multiplicative converse procedure is the primary wellspring of force dispersal and it contains the longest basic way as appeared in Figure 5.2. S-BOX procedure is accomplished more than two phases; named multiplicative converse and relative change. Because of many-sided quality of Galios Field (GF (28) to discover multiplicative reverse for every S-BOX component, Composite Galois Field is utilized.

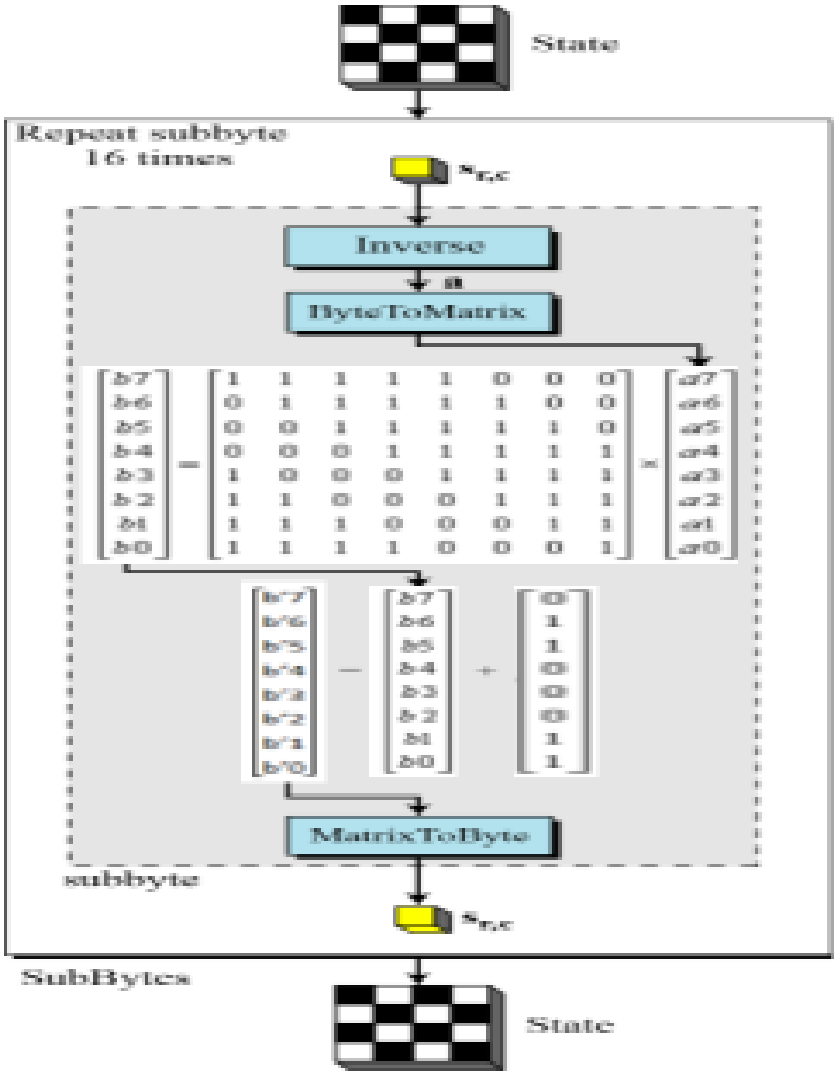


Figure 5. SUBBYTE Process [8]

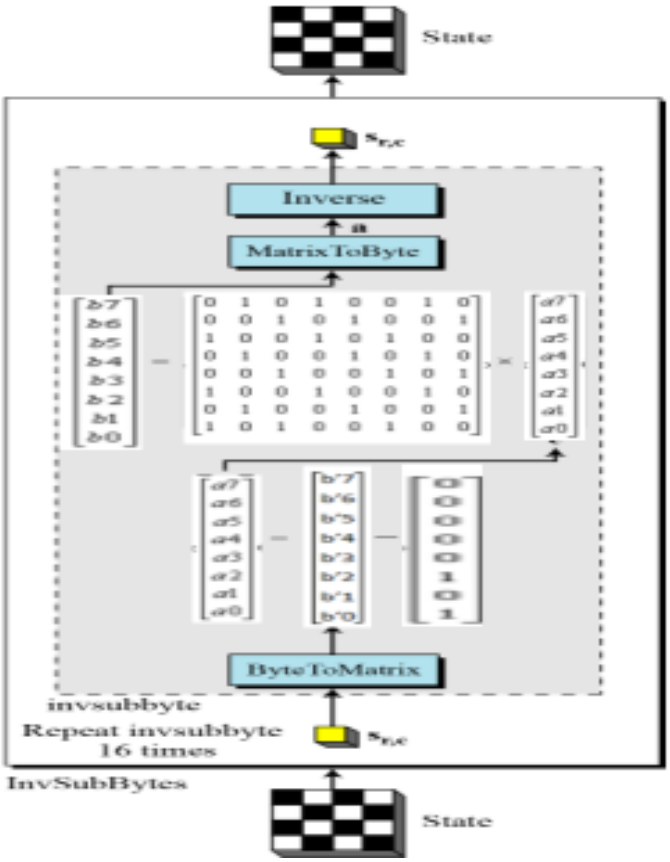


Figure 6. Inverse SUBBYTE Process [8]

5. RTL View and Simulation Results of S-Box and Inverse S-Box

5.1. Results and RTL View of S-BOX and Inverse S-BOX

RTL is nothing but the register transfer logic. The basic diagram of RTL consist the routing and the design flow. This is shown in Figure 7and Figure 8.

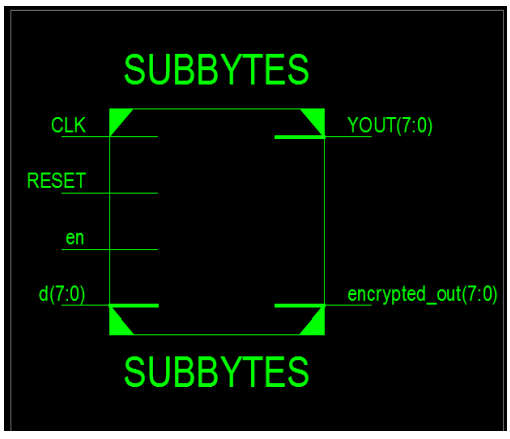


Figure 7. RTL View of S-Box

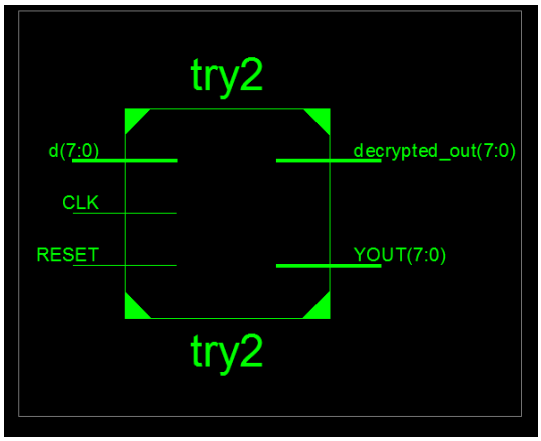


Figure 8. RTL View of Inverse S-Box

5.2. Simulation Results

The S-box engineering has been actualized utilizing HDL code. The gadget in use for the usage is XC6SLX4 on Spartan6 board. The projected engineering has actualized in Spartan6 .The configuration union has been done in various Xilinx gadgets. The S-confine outline Spartan6 FPGA takes 1 check cycle in SubBytes change and likewise Inverse S-Box plan in Spartan6 FPGA takes 1 check cycle in InvSubBytes change. The recreated yield can be found in Figure 9 and Figure 10.

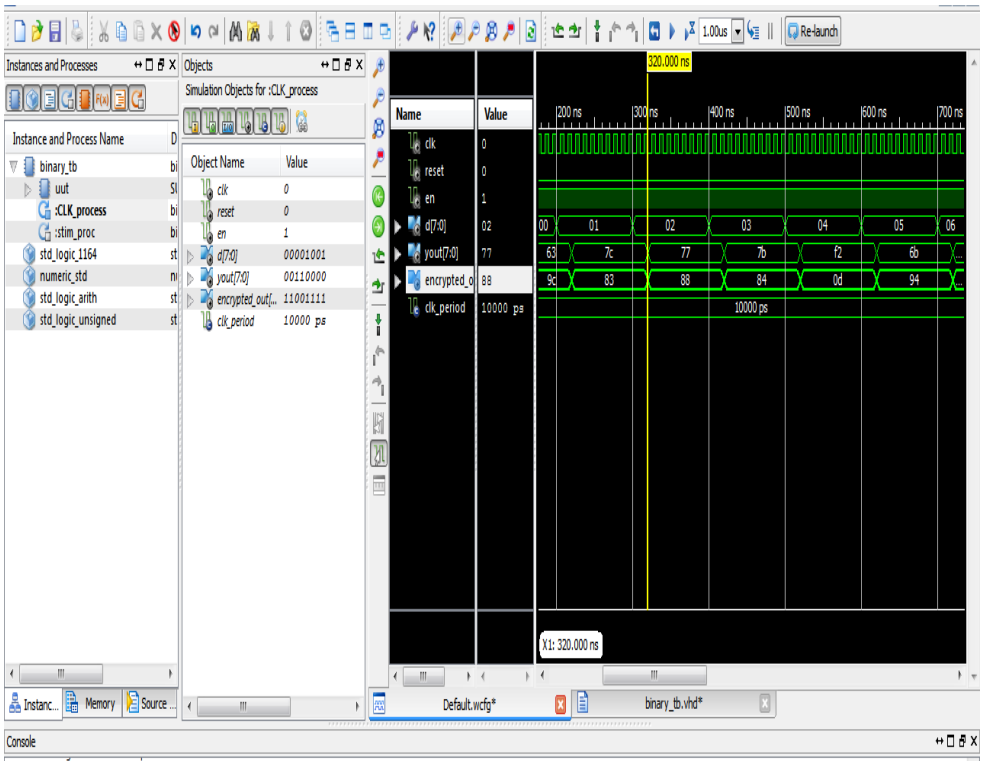


Figure 9. Waveforms Representing the Formation of Substitution Table Elements

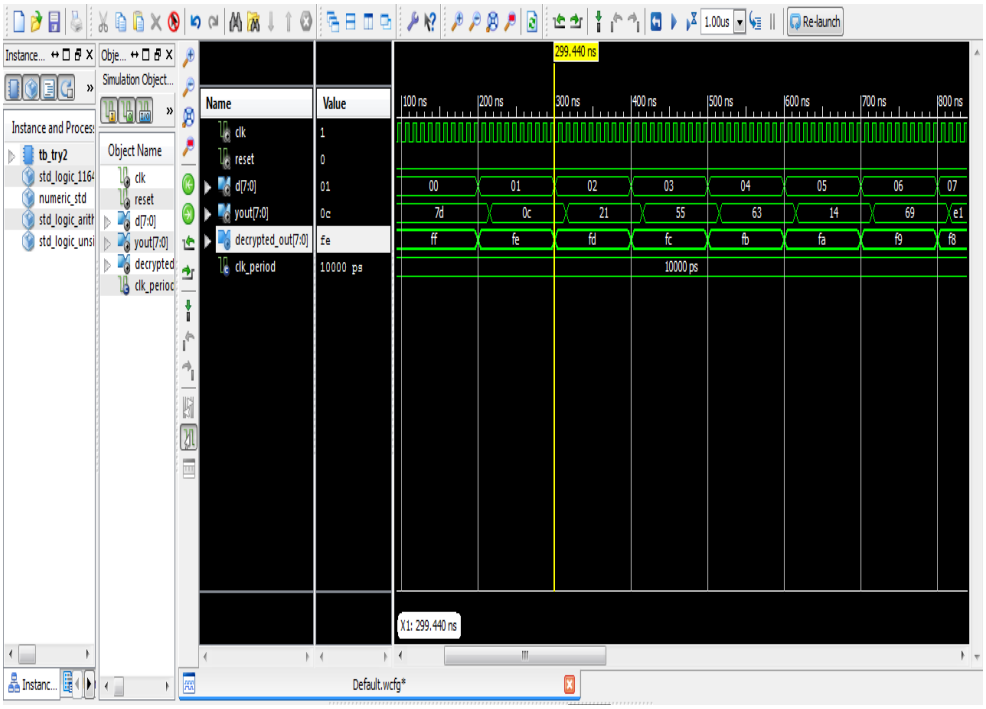


Figure 10. Waveforms Representing the Formation of Inverse of Substitution Table Elements

5.3.2. Synthesis Table

Power is analysed by Xpower Analyzer tool available in Xilinx ISE 14.1. As shown in Table 3 and Table 4, it is obvious that there is a change regarding defer and control utilization in the proposed structure. It can be seen that there is significant territory change as far as FPGA cuts, and speed change (the basic way defer) in our proposed strategy design of S-box and Inverse S-Box. The substantial change can be seen in the wake of actualizing in Spartan6 (xc6slx4-3tqg144).

Logic Utilization	Used	Available	Utilization
No. of Slices	16	2400	1%
No. of fully used LUT FF	8	16	50%
No. used as Logic	16	2400	1%
No. of Bonded IOBs	27	102	26%

Table 3. Shows the S-Box FPGA Resource Utilization

Logic Utilization	Used	Available	Utilization
No. of Slices	7	960	1%
No. of fully used LUT FF	8	16	50%
No. used as Logic	21	1920	1%
No. of Bonded IOBs	28	66	42%

Table 4. Shows the Inverse S-Box FPGA Resource Utilization

The result clearly shows that the number of slices is used 1% only. The number of input LUTs are utilized only 50%. Number of bonded IOBs is 26% used for S-Box and 42% used for Inverse S-box.

6. Encryption with Secrecy Enhancement

The information encryption keeps up information privacy, uprightness and verification [3] [5]. AES is broadly utilized for securing of audio/film information substance [3], information on brilliant cards, computerized teller machines (ATMs), Network activity, WWW servers, mobile phones, and so on. S-box has numerous Computation cycles.

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

Table 5. Encryption Table

7. Power Analysis of S-BOX and Inverse S-BOX with and without Clock Gating

Without clock gating: In this table, we have shown the power consumption at different frequencies when clock is disabled

With Clock Gating: In this table, we have shown the power consumption different frequencies when clock is enabled.

FREQUENCY	WITHOUT CLOCK GATING	WITH CLOCK GATING
1GHZ	0.842	0.839
1.5 GHZ	1.040	1.034
2 GHZ	1.239	1.234
2.5 GHZ	1.438	1.431

Table 8. Power Consumption of S-Box at Different Frequencies:

FREQUENCY	WITHOUT CLOCK GATING	WITH CLOCK GATING
1GHZ	0.384	0.360
1.5 GHZ	0.291	0.274
2 GHZ	0.478	0.456
2.5 GHZ	0.198	0.153

Table 8. Power Consumption of Inverse S-Box at Different Frequencies

8. Conclusion

The designed core defines encryption and decryption process and power handling terminology. Its functionality has been verified using simulation (on Xilinx ISE Simulator) by taking various inputs; and is synthesized by using Xilinx ISE 14.1.This design is verified on FPGA (Spartan3E). In the world of digital era information is transmitted in the form of binary bits so it may be vulnerable for external attacks therefore in present paper I have introduced further encryption in S-Box output to make it more secure (However it introduces complexity in design). An encryption table is proposed to achieve this where each table element of Substitution table is XORed with FF. The coding has been done in VHDL for S-BOX and I-SBOX and program results are verified with ModelSim PE and synthesized in Xilinx ISE 14.1.

References

[1] NIST Advanced Encryption Standard (AES), FIPS PUBS 197, *National Institute of Standards and Technology*, **(2001)** Nov.
[2] J. Daemen and V. Rijmen, "The design of Rijndael; AES–TheAdvanced Encryption Standard", Springer-Verlag, **(2002)**.
[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", Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), **(2001)** May, pp. 175–188.
[4] Federal Information Processing Standards Publication 197 (FIPS 197), available online, http : // csrc . nist. gov / publications / fips / fips197/fips-197.pdf.
[5] E. NC Mui, "Practical Implementation of Rijndael S-Box Using Combinational Logic", Custom R&D Engineer Taxco Enterprise Pvt. Ltd.
[6] M. Jridi and A. AlFalou, "A VLSI implementation of a new simultaneous images compression and encryption method", 2010 IEEE International Conference on Imaging Systems and Techniques (IST), **(2010)** July, pp. 75-79.

[7] N. Ahmad, R. Hasan and W. M. Jubadi, "Design of AES S-Box using combinational logic optimization", IEEE Symposium on Industrial Electronics & Applications, **(2010)** Oct., pp. 696-699.
[8] B. A. Forouzan and D. Mukhopadhyay, "Cryptography and Network Security", 2nd Ed., Tata McGraw Hill, New Delhi, **(2012)**.
[9] G. Verma, S. Shekhar, K. Shashi Kant, V. Verma, H. Verma and B. Pandey, "SSTL IO Standard Based Low Power Arithmetic Design Using Calana Kalanabhyam On FPGA", International Journal of Control and Automation, vol. 9, no. 4, **(2016)** April, pp. 271-278.
[10] G. Verma, V. Verma, D. Sharma, A. Kumar, H. Verma and K. Kalia", Design Goal Based Implementation of Energy Efficient Greek Unicode Reader for Natural Language Processing", International Journal of Smart Home, vol. 10, no. 3, **(2016)** March, pp. 181-190.
[11] G. Verma, M. Kumar and V. Khare, "Low Power Techniques for Digital System Design", Indian Journal of Science and Technology", vol 8, issue 17, IPL063, **(2015)** August.
[12] S. Mishra and G. Verma, "Low Power and Area Efficient Implementation of BCD Adder on FPGA", in Proceedings of IEEE International Conference on Signal Processing and Communication (ICSC-2013), **(2013)** December 12-14, pp. 461-465.

