Design and Verification of Encryption of AES Algorithm

Ajinkya Sunil Naik, Dr.G.V.Maha Lakshmi
M.Tech., Professor,
VLSI & Embedded System ,ECE Department,
SNIST,Ghatkesar,RR (Dist), Telangana, SNIST,Ghatkesar,RR(Dist), Telangana

Abstract: The paper presents an efficient reconfigurable hardware implementation of Advance Encryption Standard (AES) algorithm on Field Programmable Gate Array (FPGA);using High Level Language (HLL) approach with less hardware resources. The FPGA platform used for AES implementation is Xilinx. Time-to-market is one of the key factors for any design in FPGA and digital system designing industry. This time can be reduced considerably with HLL approach. The presented algorithm is designed on a HLL tool, namely Xilinx system generator. It is very user friendly despite giving detailed control in designing the required system design. For actual testing and hardware implementation of the algorithm, the HLL-tool generates a bit file that can be directly burnt on the FPGA. To get the implementation of design on hardware, the presented work uses a similar approach to directly map the System Generator described design on FPGA. The presented work emphasizes on optimization for less hardware utilization.

Key Words: AES, System Generator, FPGA, System Generator

I. INTRODUCTION

An important aspect to be considered with the evolution of internet in the current information age is secrecy and privacy. Cryptography provides confidentiality and reliability to data during communication. It is used in different application which includes e-commerce, wireless communications, cellular networks, online banking, computerized networks etc. Cryptography is related to the study of secret writing i.e. conversion of plaintext into cipher-text.Information retrieved by the desired entity over a non secured channel. Text cannot be transform into intelligible form (plaintext) unless receiver has a cipher key.

Since a few decades, digital hardware design technology has become more similar to software design and has evolved tremendously with the introduction of reconfigurable platforms like FPGA. There configurable platform provides perfect customization of the hardware with time and cost. ASIC belongs to configurable platform but it configures permanently and provides high performance for a specific application. Whereas, software provides reprogrammable flexibility for different applications but lacks in performance and efficiency as compared to ASIC's.

FPGA.fills the gap to achieve a balance between hardware and software in terms of performance and flexibility. FPGA provides improved performance than software implementation; and it can also be reconfigured. It executes the hardware design efficiently over software by minimizing the time required to process the algorithm. HLL tool generate a bit file that can be directly burnt on the FPGA. The work emphasizes on optimization for less hardware utilization. Due to the merits described, FPGA scan be considered to implement the cryptographic algorithms. The presented work shows efficient implementation of AES algorithm using High Level Language (HLL) approach i.e. Xilinx System Generator on FPGA. The proposed FPGA platform for the implementation of this work is Xilinx AtlysVertix-6.There configurable platform using system generator provides the better way in designing of hardware. System generator has environment similar to linking which Xilinx blocks are used in the architecture of hardware. It generates the file for synthesis and simulation; and also provides access to FPGA blocks used in the design.

II. DESCRIPTION OF AES ALGORITHM

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plain-text.

A. AES Encryption

The AES algorithm operates on a different bit block data like 128,192,256. But here in this paper AES algorithm operates on a 128-bit block of data and executed Nr-1looptimes.

A loop is called around and the number of Iterations of a loop, Nr, can be 10**,** 12, or 14 depending on the key length. The key length is 128,192 or 256 bits in length respectively. The first and last rounds differ from other rounds in that there is an additional AddRoundKey transformation at the beginning of the first round and no MixCoulmns transformation is performed in the last round. In this paper, we use the key length of 128 bits (AES-128) as a model for general explanation. About line of AES encryption is given in Fig.1.

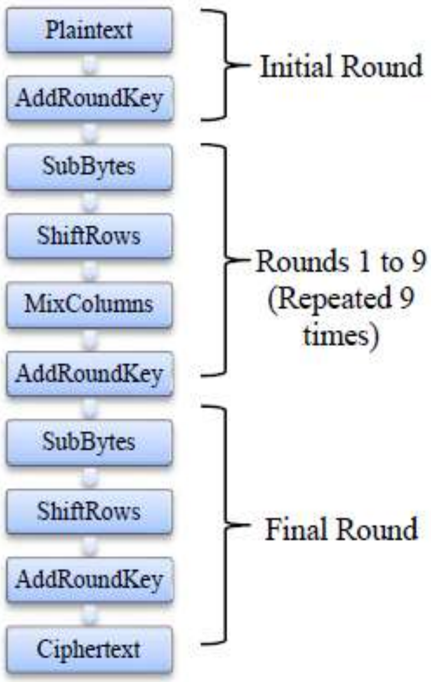
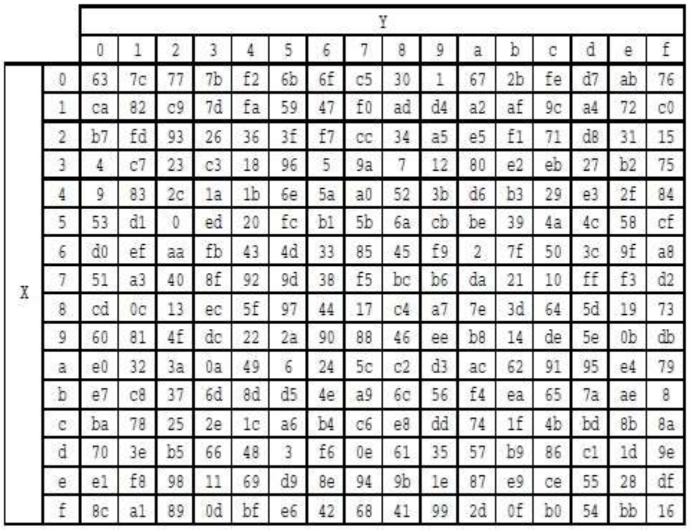


Fig.1 AES Rounds

a) SubBytesTransformation

The SubBytestransformation is a non-linear byte substitution, operating on each of the state bytes independently. The SubBytestransformation is done using a once-pre-calculated substitution table called S-box. That S-box table contains 256 numbers (from0to255) and their corresponding resulting values. More details of the method of calculating the S-box table. In this design, we use a look-up table as shown in Table. This is a more efficient method than directly implementing the multiplicative inverse operation followed by affine transformation.

TABLE I S-BOX TABLE



		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	1	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	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
	4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	2	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	6	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	8
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	3	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

Fig.2 S-Box

b) ShiftRowTransformation

In ShiftrowTransformation, the rows of state matrix cyclically do left shift. Row 0 is kept as it is; Row1 is one byte shifted to left; Row2 is two byte shifted to left; Row3isthree byte shifted to left.

c) MixCoumnsTransformation

In MixColumnsTransformation each column in state matrix is represented as one word. Multiplication operation is performed with each column with fixed polynomial.

d) AddRoundKeyTransformation

In the AddRoundKeyTransformation, sub key is combined with each byte of state matrix using bitwise XOR operation.

III. IMPLEMENTATION

The proposed AES encryption function is designed and implemented using the Xilinx System Generator for MATLAB. The figure shows the outline of the structure. The implementation uses a pipelined architecture, as shown in figure; which is most commonly used reconfigurable architectures for implementation of encryption functions. Xilinx System Generator for MATLAB provides flexibility in design and scalability in FPGA chip selection.

It is a pipeline architecture of AES-128 encryption function which consist of 10 rounds. Each round is implemented separately enclosed in a subsystem, comprising of five transformations i.e. SubBytes, ShiftRows, MixColumns, AddRoundKey and Key Generation shown in figure5; where

MixColumns is eliminated in the final round. The initial round is just AddRoundKey transformation in which input state is XOR-ed with the initial round key. The Plaintext and Key are defined in separate subsystems each as shown in figure; in which each column of the state can be defined separately as shown in figure.

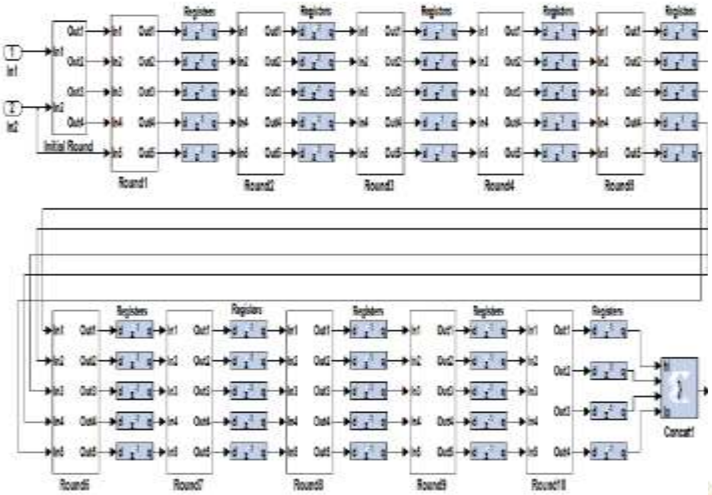


Fig.3 Pipeline Architecture of AES

Firstly 128to 8 bit conversion is carried out with Bit Basher block. The first word (1st Column) of 32-bit of Round key is generated by substitution of last word (4th column) of key state matrix using SubBytes block. The result of SubBytes is then rotated by simply rearranging the connecting wires. The rotated word is XOR-ed with 1st column of key and round constant for the generation of 1st word of RoundKey, as shown in figure.

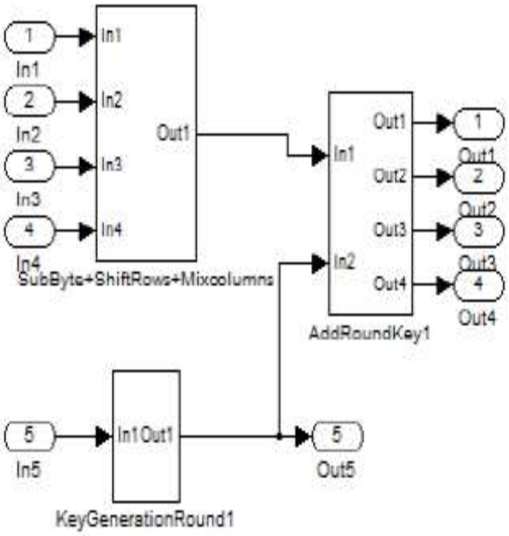


Fig.4 Structure of Each Rounds

As per AES Algorithm, the 1st word of Roundkey is then used to generate other words by using XOR operation. In AddRoundKey, Bitwise XOR-ing between result from MixColumns and RoundKey is done. Here the expression blocks are used for XOR operation. Also Bit Basher for 128 to 32 bit and 32 to 8 bit conversion is used as shown in figure.

IV. RESULTS

The simulation results of AES Algorithm consists of 4 internal operational architecture (Sub Byte, Shift Row, Mix Column and Add Round key) and one externally generated key using Key generator Hardware.

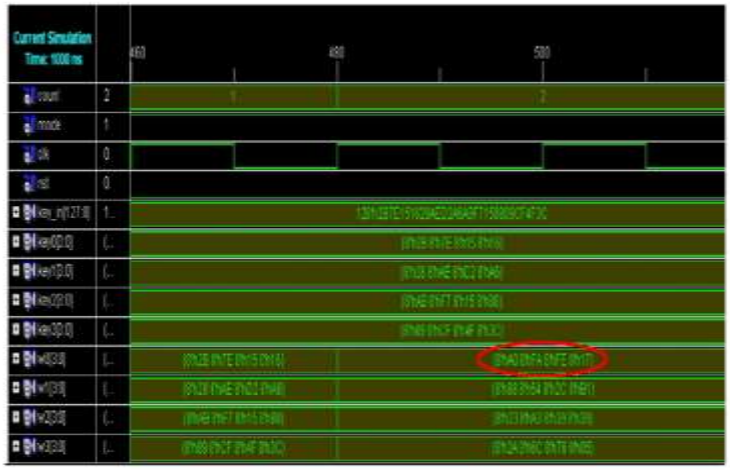


Fig.5 Simulated Result

V. CONCLUSION

Optimized and Synthesizable VHDL code is developed for the implementation of both encryption and decryption process. Each program is tested with some of the sample vectors provided by NIST and output results are perfect with minimal delay. Therefore, AES can indeed be implemented with reasonable efficiency on an FPGA, with the encryption and decryption taking an average of 320 and 340 ns respectively (for every 128 bits). The time varies from chip to chip and the calculated delay time can only be regarded as approximate. Adding data pipelines and some parallel combinational logic in the key scheduler and round calculator can further optimize this design.

REFERENCES

[1] A.M. Deshpande, M.S. Kayatanavar, D.N. "FPGA implementation of AES encryption and decryption", IEEE Transactions, Print ISBN: 978-1-4244-4789-3 ,Jun 2009.

[2] William Stalling, “Cryptography and Network Security Principles and Practices” PrenticeHall, sixth edition, 2013.

[3] L. Floyd, “Digital Fundamental with VHDL” Pearson Education, pp.362-368, 2003.

[4] J. Zambreno, D. Honbo, A. Choudhary, R. Simha and B. Narahari, “High performance Software Protection Using Reconfigurable Architectures”, ProceedingsoftheIEEE,volume94,No.2, February2006.

[5] Rajender Manteena, “A VHDL Implementation of the Advanced Encryption Standard-Rijndael Algorithm” College of Engineering University of South Florida,2004.

[6] S. Morioka and A. Satoh “A 10-Gbps Full AES-Crypto Design with a Twisted BDD S-Box Architecture” IEEE Transaction on VLSI Systems, Vol.12,No.7, July2004,pp.686-691.

[7] C. P. Su, T. F. Lin, C. T. Huang, and C. W. Wu, "A high-throughput low-cost AES processor," IEEE Commun. Mag., vol. 41, pp. 86-91, Dec. 2003.

[8] An introduction to Xilinx System Generator http://homes.esat.kuleuven.be/~mknezevi/documents/sysgen_intro.pdf.

[9] Xilinx, Virtex-5 FPGA User Guide, available at http://www.xilinx.com/support/documentation/user_guides/ug380.pdf

[10] Data Encryption Standard (DES), FIPSPUB46-3, October 1999 [8] NIST. November 26, 2001. Retrieved October 2, 2012. Advanced Encryption Standard (AES) (FIPSPUB197).

[11] S.Qu, G.Shou, Y.Hu, Z.Guo and Z.Qian, “High Throughput Pipelined Implementation of AES on FPGA”, International Symposium on Information Engineering and Electronic Commerce, pp.542-545, 2009.