International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 07 |Oct-2015 www.irjet.net p-ISSN: 2395-0072

Realization of AES Encryption and Decryption Based On Null Convention Logic

D.V.Supriya[1], Mr.R.Niranjan[2]

[1] *PG Scholar(VLSI), Faculty of ECE, JNTU Anantapur, Andhra Pradesh, India*
[2] Member of Technical staff, Seer Akademi, Hyderabad, *Andhra Pradesh, India*

Abstract - ***In This paper we demonstrates the implementation of AES Encryption and Decryption using asynchronous NCL. The AES specifies a Federal Information Processing Standard (FIPS)-approved cryptographic algorithm that can be used to protect electronic data. In today's world there is a growing demand for real time implementation of cryptographic algorithms which are being used in secure communication systems, networks and security systems. Increased use of computer and communications systems by industry has increased the risk of theft of proprietary information. Cryptographic services required for these applications involve not only solutions for data protection but also self-implementation concerns. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher)digital information. Encryption converts data to an unintelligible form called ciphertext. decrypting the ciphertext converts the data back into its original form, called plain text. In Existing method, AES Encryption and Decryption is implemented by using 128 binary bits. The proposed design is AES Encryption and Decryption implemented based on a delay-insensitive (DI) logic paradigm known as null convention logic (NCL), which supports useful properties for resisting SCAs including duail-rail encoding, clock-free operation, self-timed logic and monotonic transitions. These advantageous properties make it difficult for an attacker to decipher secret keys. The proposed NCL based AES Encryption and Decryption approach using Mentor Graphics Modelsim and VCS tools***

Key Words: *Advanced Encryption Standard(AES), Encryption, Decryption, KeyExpansion, NCL*

1. INTRODUCTION

The National Institute of Standards and Technology, (NIST), solicited proposals for the Advanced Encryption Standard, (AES). The AES is a Federal Information Processing Standard, (FIPS), which is a cryptographic algorithm that is used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt, (encipher), and decrypt, (decipher), 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 plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. Many algorithms were originally presented by researchers from twelve different nations. Fifteen (15), algorithms were selected from the first set of submittals. After a study and selection process five(5), were chosen as finalists. The five algorithms selected were MARS, RC6, RIJNDAEL, SERPENT and TWOFISH. The conclusion was that the five Competitors showed similar characteristics. On October 2nd 2000, NIST announced that the Rijndael Algorithm was the winner of the contest. The Rijndael Algorithm was chosen since it had the best overall scores in security, performance, efficiency, implementation ability and flexibility. The Rijndael algorithm was developed by Joan Daemen of Proton World International and Vincent Fijmen of Katholieke University at Leuven.The Rijndael algorithm is a symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128, 192, and 256 bits. The Rijndael algorithm was also designed to handle additional block sizes and key lengths.

1.1 Null Convention Logic

NULL Convention Logic (NCL) is a symbolically complete logic which expresses process completely in terms of the logic itself and inherently and conveniently expresses asynchronous digital circuits. The traditional form of Boolean logic is not symbolically complete in the sense that it requires the participation of a fundamentally different form of expression, time in the form of the clock, which has to be very carefully coordinated with the logic part of the expression to completely and effectively express a process. We introduce NULL Convention Logic in relation to Boolean logic as a four value logic, and as a three value logic and finally as two value logic quite different from traditional Boolean logic. NULL Convention

Logic (NCL) provides an asynchronous design methodology employing dual-rail signals, quad-rail signals. A Dual rail signal, D consists of two wires or rails D0 and D1 . These signals take any value from the data set {DATA0, DATA1, and NULL} as illustrated in Table-1.

Table -1: DUAL RAIL LOGIC

	DATA0	DATA1	NULL	ILLEGAL
D^0	1	0	0	1
D^1	0	1	0	1

The following table-2 shows example of AND gate Implemented by using Null Convention Logic.

Table -2: EXAMPLE FOR NCL AND GATE

A	B	AND GATE	NCL LOGIC AND GATE
0	0	0	01
0	1	0	01
1	0	0	01
1	1	1	10

2. AES ENCRYPTION

The AES algorithm operates on a 128-bit block of data and executed Nr - 1 loop times. 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.

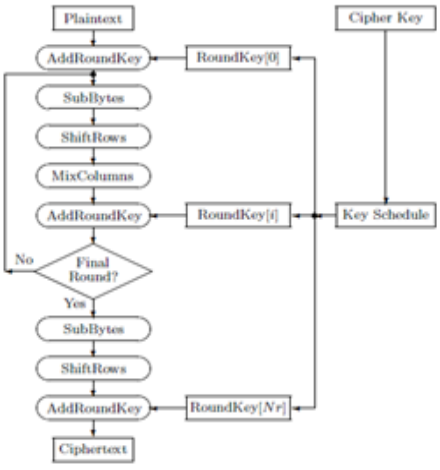

Fig -1: The AES ENCRYPTION

The AES encryption algorithm is broken up into four transforms known as SubBytes, ShiftRows, MixColumns and AddRoundKey. There is also a Key Expansion that must be performed on the cipher key. The individual transforms are performed in a number of rounds that are dependent on the cipher key size. For key sizes of 128, 192, or 256 bits there are a total of 10, 12, or 14 rounds respectively

2.1 SubBytes Transformation:

The SubBytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The SubBytes transformation is done using a once-precalculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. This approach has the significant advantage of performing the S-box computation in a single clock cycle, thus reducing the latency and avoids complexity of hardware implementation.

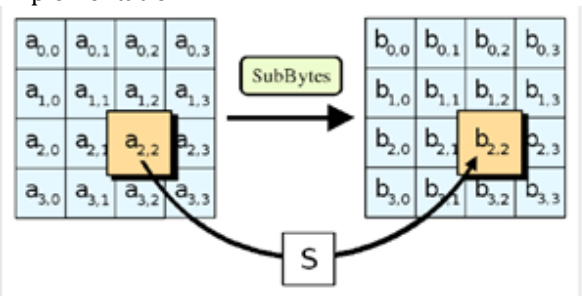

Fig -2: The Subbytes Transformation

2.2 ShiftRows Transformation:

In ShiftRows transformation, the rows of the state are cyclically left shifted over different offsets. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.

Fig -3: The ShiftRows Transformation

2.3 MixColumns Transformation:

In MixColumns transformation, the columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial c(x), given by:

c(x)={03}x3 + {01}x2 + {01}x + {02}.

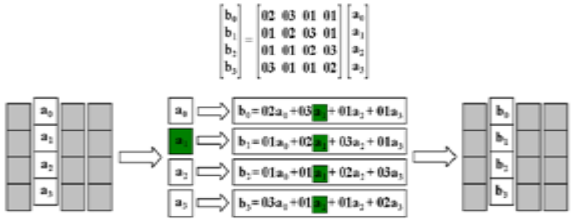

Fig -4: The MixColumns Transformation

2.4 Add RoundKey Transformation:

In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the MixColumns transformation - by a simple bitwise XOR operation. The RoundKey of each round is derived from the main key using the KeyExpansion algorithm. The AES encryption algorithm needs eleven 128-bit RoundKey, which are denoted RoundKey[0]

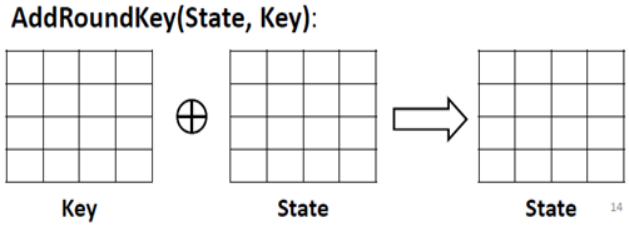

Fig -5: The AddRound Key Transformation

2.5 Key Expansion :

The algorithm for generating the 10 rounds of the round key is as follows: The 4th column of the i-1 key is rotated such that each element is moved up one row.

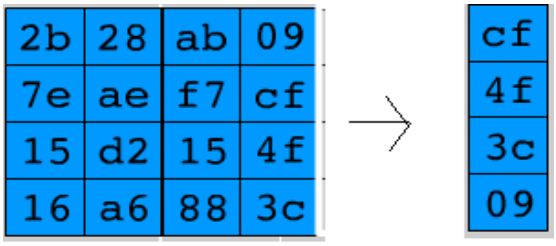

Fig -6: First Step of Key Expansion

It then puts this result through a forwards S-Box algorithm which replaces each 8 bits of the matrix with a corresponding 8-bit value from S-Box.

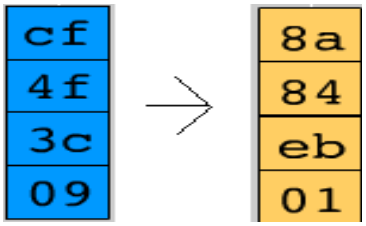

Fig -7: Second Step of Key Expansion

To generate the first column of the ith key, this result is XOR-ed with the first column of the i-1th key as well as a constant (Row constant or Rcon) which is dependent on i.

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Fig -8: Rcon values

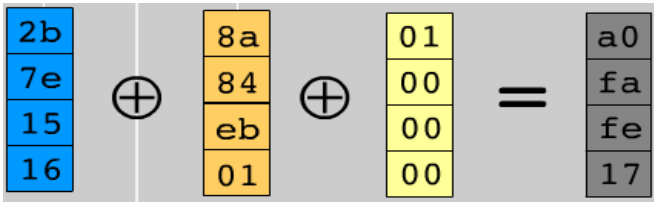

Fig -9: Third Step of Key Expansion

The second column is generated by XOR-ing the 1st column of the ith key with the second column of the i-1th key.

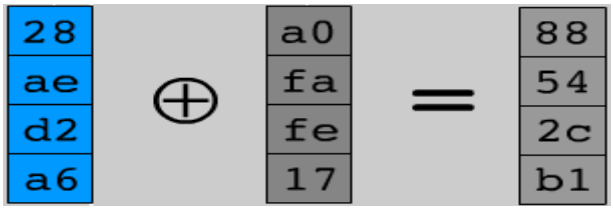

Fig -10: Fourth Step of Key Expansion

This continues iteratively for the other two columns in order to generate the entire ith key.

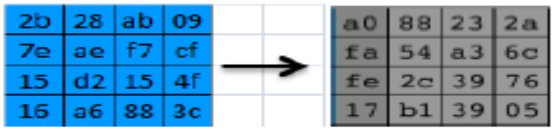

Fig -11: Round-1 Key Generation

Additionally this entire process continues iteratively for generating all 10 keys. As a final note, all of these keys are stored statically once they have been computed initially as the ith key generated is required for the (10-i)th round of decryption.

3. AES DECRYPTION

AES Decryption is a reverse process of AES Encryption which inverse round transformations to computes out the original plaintext of an encrypted cipher-text in reverse order shown in fig. The round transformation of decryption uses the functions AddRoundKey, InvMix-Columns, InvShift-Rows, and InvSubBytes successively.

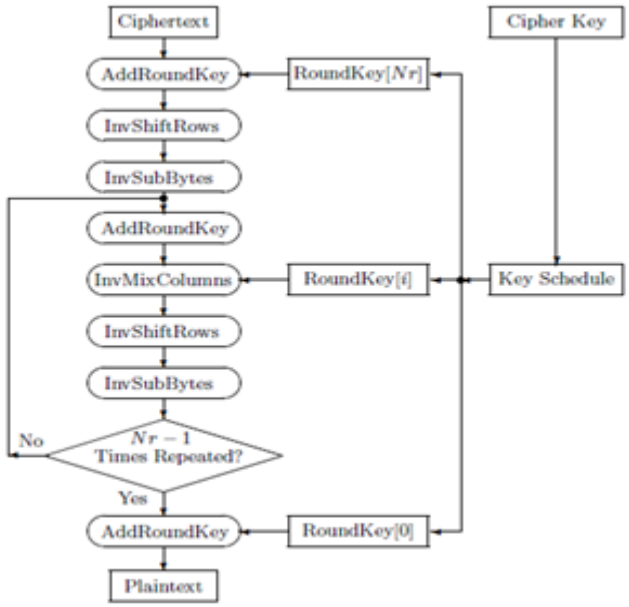

Fig -12: AES DECRYPTION

3.1 Add RoundKey:

AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order.

1A	A4	95	3E		D0	C9	E1	B6		CA	6D	74	88
A9	B9	4C	3A	+	14	EE	3F	63	=	BD	57	73	59
13	CD	78	27		F9	25	0C	0C		EA	E8	74	2B
86	F1	33	A8		A8	89	C8	A6		2E	78	FB	0E

Fig -13: The AddRound Key Transformation in AES Decryption

3.2 InvSubBytes transformation:

The InvSubBytes transformation is done using a once precalculated substitution table called InvS-box. That InvS-box table contains 256 numbers (from 0 to 255) and their corresponding values.

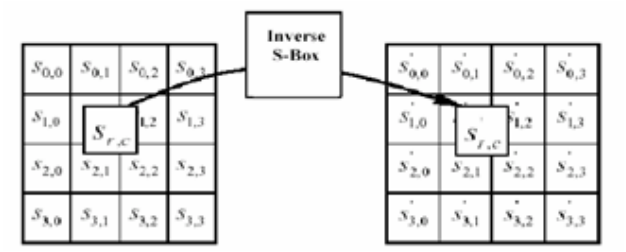

Fig -14: The InvSubbytes Transformation

3.3 InvShiftRows Transformation:

InvShiftRows exactly functions the same as ShiftRows, only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.

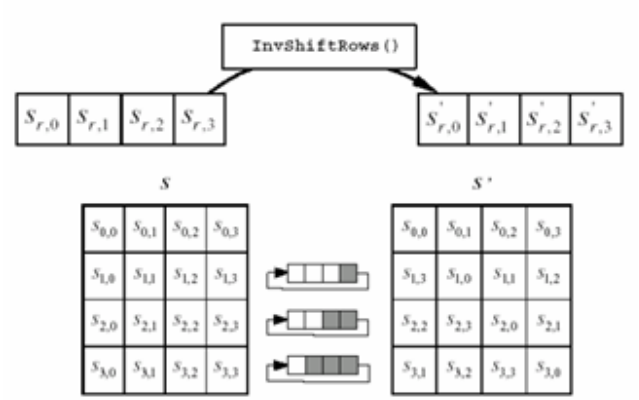

Fig -15: The InvShiftRows Transformation

3.4 InvMixColumns Transformation:

The InvMixColumns transformation is done using polynomials of degree less than 4 over GF(28), which coefficients are the elements in the columns of the state, are multiplied modulo ($x4 + 1$) by a fixed polynomial $d(x)$ = {0*B*}$x3$ + {0*D*}$x2$ + {09}x + {0*E*}, where {0*B*}, {0*D*};{09}, {0*E*} denote hexadecimal values.

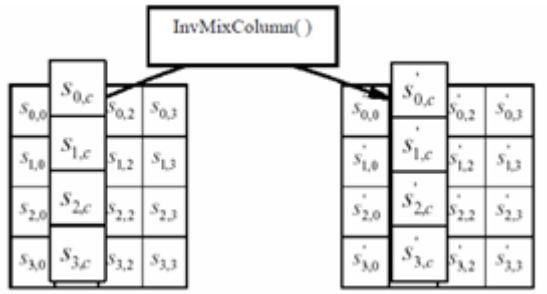

Fig -16: The InvMixColumns Transformation

3.5 Key Expansion

Key Expansion of AES Decryption is similar to the Key Expansion of AES Encryption.

4. SIMULATION RESULT

4.1 AES ENCRYPTION

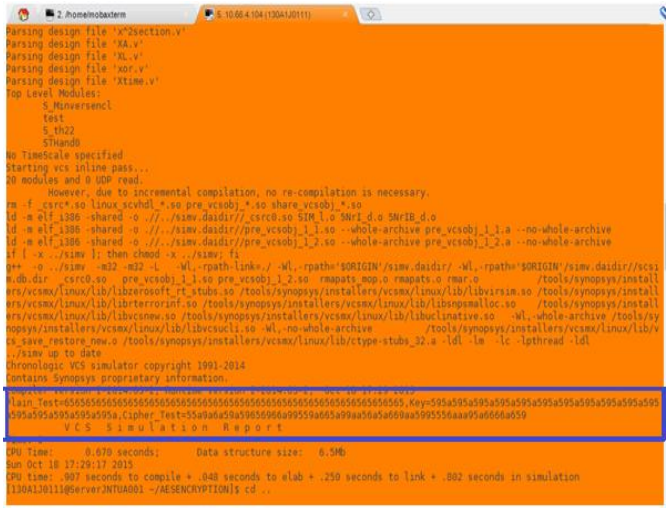

Fig -17: Simulated report of AES ENCRYPTION

4.2 AES DECRYPTION

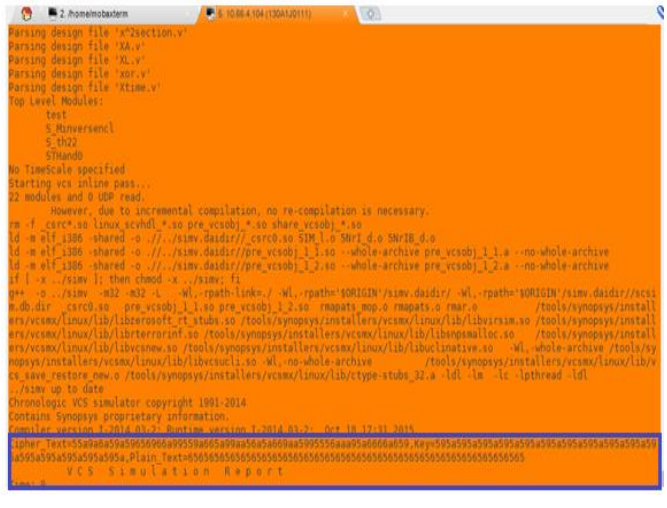

Fig -18: Simulated report of AES DECRYPTION

7. CONCLUSION

In this research we show how we can impletemented NCL based AES Encryption and Decryption Techniques successfully by using 'verilog' language. The proposed design is AES Encryption and Decryption implemented based on a delay-insensitive (DI) logic paradigm known as null convention logic (NCL), which supports useful properties for resisting SCAs including duail-rail encoding, clock-free operation, self-timed logic and monotonic transitions. These advantageous properties make it difficult for an attacker to decipher secret keys. The modifications brought about in the code was tested and proved to be accurately encrypting and decrypting the data messages with even higher security and immunity against the unauthorized users. From the simulation result, we evaluated that NCL based AES algorithm performed better secure and safety communication both in Encryption and Decryption process. The proposed NCL based AES Encryption and Decryption approach using Mentor Graphics Modelsim and VCS tools. Our future work will focus on compared and analysed existing cryptographic algorithm RSA

REFERENCES

[1] A. Lee, NIST Special Publication 800-21, Guideline for Implementing Cryptography in the Federal Government, National Institute of Standards and Technology, November 1999.
[2] A. Menezes, P. Van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, New York, 1997, p. 81-83.
[3] J. Nechvatal, ET. al., Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000,available at [1].
[4] K. Fant and S. Brandt, "Null convention logic TM: a complete and consistent logic for asynchronous digital circuit synthesis," International Conference on Application Specific Systems, Architectures and Processors, pp. 261–273, 1996.
[5] K. M. Fant and S. A. Brandt, "Null convention logic: A complete and consistent logic for asynchronous digital circuit synthesis," in Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors, p. 261, 1996.
[16] K. M. Fant and S. A. Brandt, "Null convention logic system," US Patent (US5305463), 04 1994.
[7] S. C. Smith, "Speedup of null convention digital circuits using null cycle reduc-tion," Journal of System Architecture, vol. 52, no. 7, pp. 411–422, 2006.