**Implementation of 128 bit AES (Advanced Encryption Standard) algorithm**

Pankaj Vishwakarma(19116046) Shivam Kumar(19116070) Vaibhav Anuragi(19116085) Yash Raj(19116089)

Abstract

Advanced Encryption Standard(AES), is an approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a block cipher that can encrypt and decrypt digital information. The AES algorithm is capable of using keys of 128 bit, 192 bit and 256 bit. This project implements the 128 bit standard using the Verilog HDL.

Keywords: AES algorithm, Verilog HDL, Block cipher

1.Introduction

Encryption is the process of transforming information make it unredable to anyone. This is controlled by an algorithm and a key. AES is an approved cryptographic algorithm that can encrypt and decrypt using cryptographic keys of 128, 192, and 256 bits. The algorithm used in the AES encryption system remains the same, so it is necessary to change keys frequently in order that identical encryption is not applied to messages in longer period.

2.Architecture of the Design

The AES algorithm(encryption part) consists of two main process- Cypher and Key Expansion. Cipher converts data to an unintelligible form known as cipher text. Key expansion generates a key schedule that is used in Cipher.

The input of AES is a 128 bit block and is known as the matrix. This block is copied into a state array which is modified at each stage of the algorithm and then copied to the output matrix. Both the plaintext and key are depicted as a 128 bit square matrix of bytes. This key is then expanded into an array of key schedule words. The algorithm begins with an Add Round Key stage followed by 9 rounds of four stages and a tenth round of three stages. The four stages are as follows:

1. Substitute bytes
2. Shift rows
3. Mix columns
4. Add round key

The tenth round simply leaves the Mix column stage.

A close up of text on a black background

Description automatically generatedFig: Structure of AES

2.1 Encryption Process:

It consist of a number of different transformations applied over the data block bits, in a fixed number of iterations, called rounds. For key length of 128 bits, the number of rounds required is 10 each of the first 9 rounds consists of four transformations: SubBytes(), ShiftRows(), MixColumns(), AddRoundKey().

The four different transformations are as follows:

2.1.1 Sub Bytes Transformation: It is a non-linear substitution of bytes that operates independently on each byte of the state using a substitution table. The S-box which is invertible is constructed by first taking the multiplicative inverse in the finite field GF(2^8) with irreducible polynomial m(x)=(x^8)+(x^4)+(x^3)+(x)+1. The transformation is applied over(GF(2^8)).

A square white text on a keyboard

Description automatically generated

Fig: Sub Bytes stage of algorithm

A close up of a keyboard

Description automatically generated

Fig: S-Box

2.1.2 Shift Rows Transformation: Cyclically shifts the rows of the state over different offsets. The first row of the matrix is left unchanged. Each byte in the second row is shifted one position to the left. Bytes in the third and fourth row are shifted by offsets of two and three respectively.

A picture containing clock

Description automatically generated

Fig: Shift Rows Transformation

2.1.3 Mix Columns Transformation: In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.  
During this operation, each column is multiplied by the known matrix that for the 128 bit key is

[
\begin{bmatrix}
2 & 3 & 1 & 1 \\
1 & 2 & 3 & 1 \\
1 & 1 & 2 & 3 \\
3 & 1 & 1 & 2
\end{bmatrix}.
](http://upload.wikimedia.org/wikipedia/en/math/e/5/f/e5fcda9d6f88b2625706c251cc8462ed.png)

The multiplication operation is defined as: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial unshifted value. After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF.

A close up of a device

Description automatically generated

Fig: Mix Columns Transformation

2.1.4 Add Round Key Transformation: In the Add Round Key step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

A close up of a clock

Description automatically generated

Fig: Add Round Key Transformation

3. Challenges faced during Implementation

Due to the ongoing global pandemic COVID-19 we had to face a lot of issues for making this project.

Because of colleges shutdown in the global pandemic a proper knowledge of Verilog could not be imparted to us due to which it was quite difficult for us to work on this project.

Despite this being a group project it was difficult to work in a group and contribute equally to the project for all group members because of the ongoing global pandemic.

4. Simulation Results

|  |  |  |
| --- | --- | --- |
| S.No | Input Text | Cipher Text |
| 1 | 128’hdda97ca4864cdfe06eaf70a0ec0d7191 | hef0bc156ed8ff21223f247b3e0318a99 |
| 2 | 128’h3243f6a8885a308d313198a2e0370731 | hf91914cd01924b124c2ec316b4b35a79 |
| 3 | 128’h00112233445566778899aabbccddeeff | h8df4e9aac5c7573a27d8d055d6e4d64b |
| 4 | 128’h8ea2b7ca516745bfeafc49904b496089 | hec8ce641087165a463d4118dc35f9001 |

Key used in above simulations: 128’h2b7e151628aed2a7abf7158809cf4f3c

AES block length=128 bits

Key length=128 bits

No of rounds=10

The correctness of the Verilog model was tested using simulation in ModelSim.

5. Conclusion

Optimized and Synthesizable Verilog HDL code is developed for the implementation of the encryption part. Each program is tested with some of the input data and key and output results are perfect with minimal delay.

REFRENCES:

[1] Marko Mali, Franc Novak and Anton Biasizzo “Hardware Implementation of AES Algorithm” -Journal of Electrical Engineering, Vol 56, No. 9-10, 2005, 265-269.

[2] FIPS 197, “Advanced Encryption Standard(AES)”, November 26,2001.

[3] J.Daeman and V.Rijmen, “AES Proposal: Rijndael”, AES Algorithm Submission, September 3,1999.

[4] J.Nechvatal et.al., Report on the development of Advanced Encryption Standard, NIST publication, Oct 2,2000.

[5] Behrouz A.Forouzan and Debdeep Mukhopadhyay “Cryptography and Network Security”.