IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 4, Ver. I (Jul. - Aug. 2016), PP 01-04
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

AES Implementation on Virtex-6 FPGA for Enhanced performance using pipelining and partial reconfiguration techniques

K Navatha[1] , Dr. J.Tarun Kumar [2] Pratik Ganguly[3]
[1,2] *Department of Electronics and Communication Engineering SRIT, Warangal, (Telangana), India*
[3] *Department of Electronics and Communication Engineering SREC, Warangal, (Telangana), India*

Abstract: *Implementation of Encryption Standard system (AES) by efficient code optimization and partial reconfiguration techniques has been presented in this paper. 128 bit block size and cipher key have been used for this AES implementation. Rijndael algorithm which is also referred as AES is mainly used for ensuring transmission channels security. Xilinx design tool 13.3 and Xilinx project navigator tools are used for synthesis and simulation purpose.For coding of the design, VHDL language has been used. Pipelined design has been implemented on Virtex 6 FPGA device and a throughput of 49.3Gbits/s is achieved with the frequency of 384.793 MHz.*

Keywords: *pipelined design, AES, FPGA, throughput*

I. Introduction

In data security, Cryptography has an important part. Sensitive information is stored or transmitted across insecure networks in such a way that unauthorized persons can have the access of the data. Different Encryption algorithms are used for the secured exchange of data across the nentwork. Symmetric key algorithms can be executed much faster to compared to asymmetric key algorithms. AES was originated from by the initiative of NIST in the year 1997 for selecting a new symmetric key algorithm. Rijndael algorithm was selected for the Advanced Encryption Standard (AES) because of the combination of security, efficiency, improved performance and ease of implementation in addition to its flexibility. Cipher, inverse cipher, key expansion is the main three parts of this algorithm. Cipher converts the data to an unintelligible form which is known as cipher text whereas Inverse cipher is used to convert data back to original form which is called plain text. Key expansion is used to generate a key schedule which is mainly used in cipher and inverse cipher mechanisms. The total number of rounds for Cipher and Inverse cipher which has to be performed during the execution of this algorithm depends on the key

II. Specification Of Algorithm

Length of input , output block and the State comprises of 128 bits which is represented by *Nb* = 4 and reflects the number of 32-bit words (columns) in the phase. AES algorithm uses a round function that comprises of 4 different byte-oriented functions for Cipher and Inverse Ciphe.

1) Usage of S-box or Substitution for Byte substitution
2) Array row shift mechanism by different offsets.
3) Data mixing within each column of State array.
4) Addition of a Round Key to the State.

The algorithm has been shown in Fig 1.

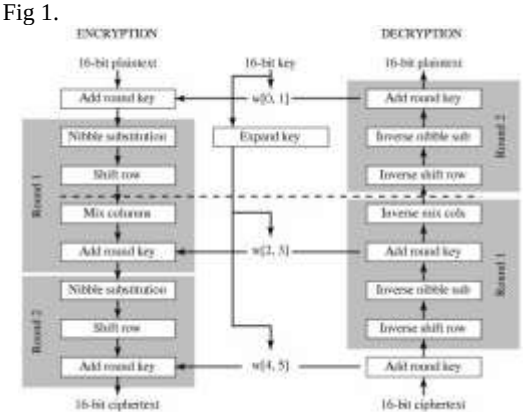

Figure1. Structure of AES Algorithm

A. Sub Bytes ( ) Transformation–

The Sub Bytes() transformation is a kind of non-linear byte substitution which operates on each byte of the State without any dependency using a S-Box or substitution table. Sub bytes transformation generally consists of two sub steps:

a) Multiplicative inverse of each byte over irreducible polynomial. ($x^8 + x^4 + x^3 + x + 1$).
b) Then application of affine transformation, that is defined by:

Fig. 2. Effect of the Sub Bytes() transformation on State.

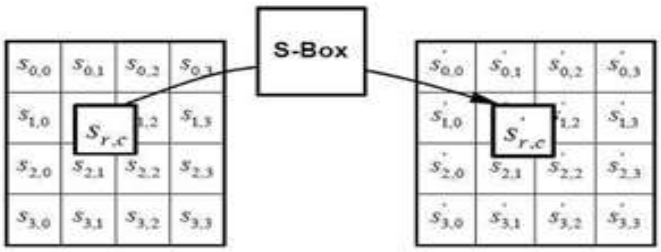

Figure 2. SubBytes() applies S-box to states each byte.

B. Shift Rows( ) Transformation–

In Shift-Rows transformation method, Last three rows bytes of the State are shifted cyclically over different numbers of bytes or offsets. The first row, $r = 0$, is not shifted, row [1] is shifted by 1, row [2] is shifted by 2 and row [3] is shifted by 3. This is shown in Fig. 3.

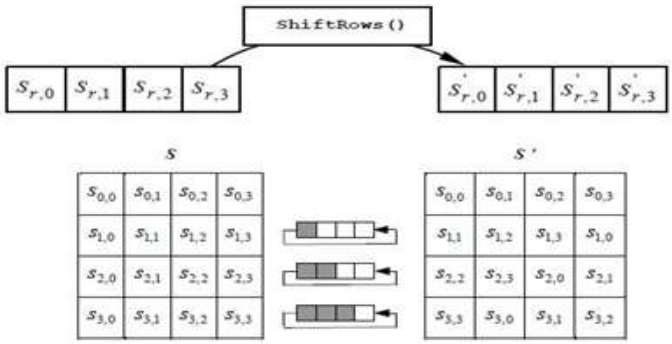

Figure 3. Shift Row Transformation

C. Mix Column Transformation

The Mix Columns transformation mechanism operates on State column wise, which treats each column as four-term polynomial. At a time, this operates on four bytes or a word which is not like other operations that operate on a byte. Each column is multiplied by a polynomial here. The expansion of this matrix is completely based on multiplication of polynomial coefficient. The columns are mainly considered as polynomials over GF (2^8) & multiplied modulo $x^4 + 1$ by a fixed polynomial $a(x)$ that is shown by the following equation below :

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}.$$

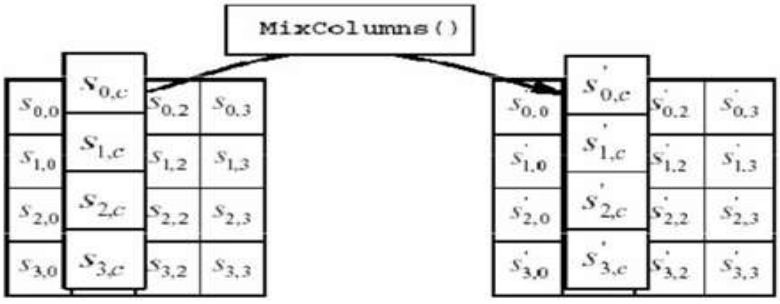

Figure 4. MixColumn operates on State column-by-colum

D. Add Round Key Transformation

In AddRoundKey() transformation mechanism, a Round Key is added to the State by bitwise XOR method. Each Round Key comprises of Nb words from key expansion. Those Nb words are added to the columns of the State. Key Addition is the same for decryption mechanism.

AES round proves in great deal of parallelism. The data bytes are ordered from MSB to LSB following big-endian presentation.

E. Key Expansion mechanism

Each round key is a 4-word (128-bit) array which is generated as a product of previous round key, a constant which changes in each round, and a series of Substitution box lookups for 32-bit word of the key. The Key schedule Expansion generates total of Nb (Nr + 1) words.

The decryption mechanism is directly inverse to the encryption mechanism. All the transformations applied in encryption mechanism are inversely applied for this process. The last round values of data and key are the first round inputs for decryption process and it follows in decreasing order.

III. Implementation Of Aes

For full parallel processing of the state, it is mandatory to implement every transformation over 128 bits. The most expensive transformation is the Byte substitution as it is table lookup operation that is implemented as ROM. Each 8 bits needs a 2048 bit ROM. To process 128 bits, 32768 numbers of bits are needed. The Key Expansion uses a Byte substitution operation over 32 bits, so 8192 bits has to be allocated further.

The design & implementation of AES algorithm is developed by optimization of each component. We have implemented the AES algorithm on Virtex 6 xc6vlx240t, package ff1156 device. VHDL has been used as the hardware description language because of the ease and flexibility of execution. The code for the design is fully VHDL that can easily be implemented on other devices as well, without any design modification. In addition, Xilinx Project Navigator has been used . This is used for debugging and for reducing optimization efforts. Simulation and checking of the performance results is performed by Questa software.

IV. Partial Reconfiguration

Primary building block for partial reconfiguration is revision. Initial design is the base revision where boundaries for the static and reconfigurable regions on the FPGA are defined. The static region is the area of the FPGA that is not reconfigured without entire FPGA programming. PR region is the area of the chip that is planned to be partially reconfigured. By a special region LogicLock PR regions are being defined, that is related with a logical design partition which supports multiple implementations.

From the base revision, additional revisions are created, that contain the static region but describe the differences in the reconfigurable areas.

Revisions contain personas, subsidiary archives that describes the characteristics of both static and reconfigurable regions, combining unique logic which implements a specific functionality to reconfigure a PR region of FPGA. Generally, two types of revisions are very specific for partial reconfiguration : aggregate and reconfigurable, both of these import the persona from the base revision for the static region. A reconfigurable revision is used to generate personas for PR regions. Those define the main differences from the base revision. An aggregate revision is mainly used for combining personas from reconfigurable revisions for creation of a complete design suitable for timing analysis.

V. Result

The results are based on simulations from the Xilinx ISE tools, using Test Bench Waveform Generator. The top module results are shown on Virtex 6 FPGA and xc6vlx240t, package ff1156 device.

From the result we can infer that the proposed implementation scheme is pretty compact and efficient in all aspects. The synthesis is carried out with the Virtex 6 series device that results in minimal consumption of hardware components with very high throughput.

A. *Synthesis Report*

Table-1

gic Utilization	Used	Utilization
No. of Slice Registers	1049 out of 301440	1%
No. of Slice LUTs	2507 out of 150720	1%
No. of fully used LUT-FF pairs	3840 out of 5686	67%
No. of bonded IOBs	388 out of 600	64%
No. of BUFG/BUFGCTRLs	1 out of 32	3%

Synthesis result for AES on Vertex 6 (device xc6vlx240t, package ff1156)

Timing Summary for AES: Minimum period: 2.136ns (Maximum Frequency: 468.246MHz)

B. *Simulation Waveform*

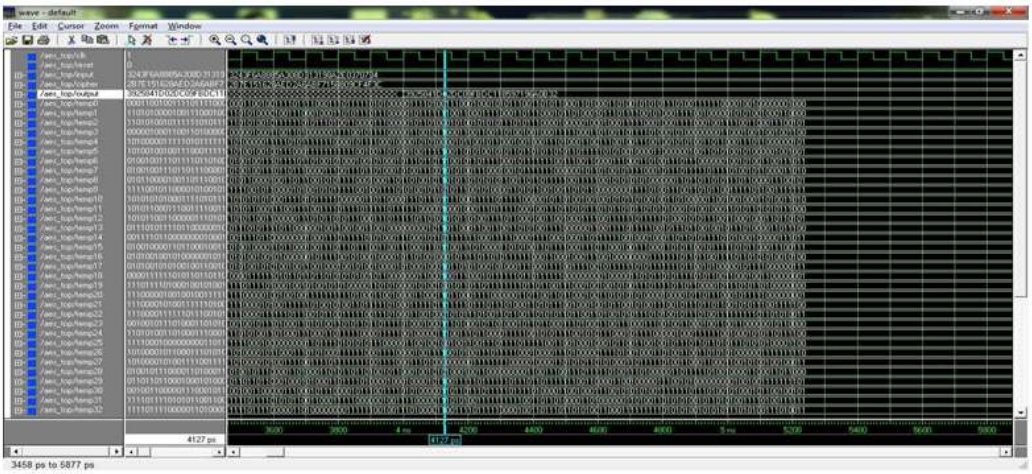

Fig 5. Encryted output waveform

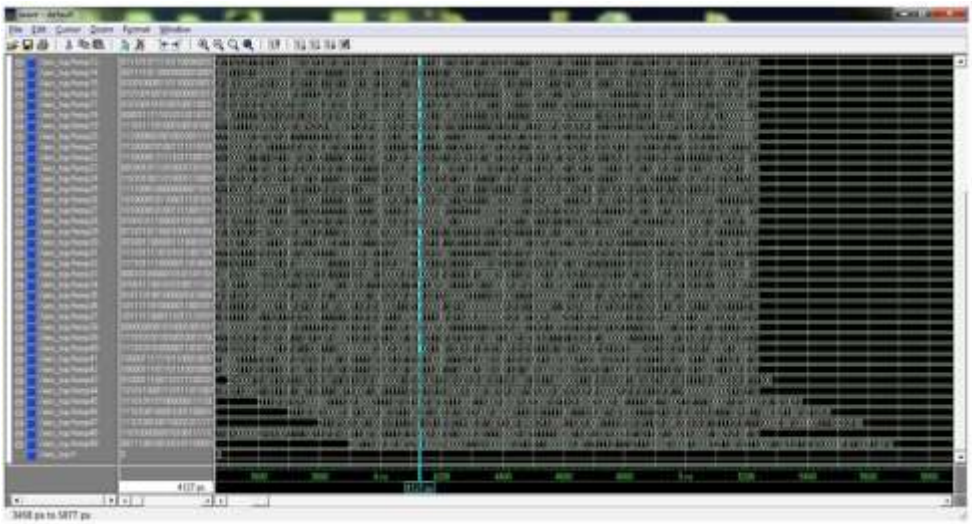

Fig 6. Encryted output waveform

VI. Conclusion

Our Proposed work consists of fully pipelined architecture for AES. In fully pipelined architecture, we achieved a throughput of 49.3Gbits/s at an operational frequency of 468.246MHz MHz. The improved results have been achieved by using Virtex 6 xc6vlx240t, package ff1156 FPGA family. This research work consists of fully pipelined architecture for cryptographic algorithms for Enhanced performance and to achieve maximum throughput by using efficient optimization and pipelining techniques on Virtex-6 series xc6vlx240t, package ff1156 FPGA device.

Very high speed performance and efficient hardware implementation have been proposed here for cryptographic systems. This is the most efficient and flexible solution for providing advanced security in crypto systems and wireless protocols secured layers. The proposed hardware implementations shows quite enough performance and efficiency improvement for the proposed system

Acknowledgement

One of the authors K.Navatha is thankful to DST, Ministry of Science and Technology, New Delhi, India for the financial assistance (DST Grant no. SR/WOS - A/ET-22/2012) and to the Management and Principal of SRIT, Warangal for the encouragement.

References

[1]. Peter J. Ashenden, "The Designer's guide to VHDL",2 nd Edition ,San Francisco, CA, Morgan Kaufmann, 2002.
[2]. FIPS 197, "Advanced Encryption Standard (AES)", No vember 26, 2001.
[3]. A.J. Elbirt et al.,"An FPGA implementation and Performance Evaluatio n of the AES block cipher Candidate Algorithm Finalists", The Third Advanced Encryption Standard (AES3) Candidate Conference New York,USA,2000.
[4]. Nation Institute of Standards and Technology (NIST), Data Encryption Standard (DES), National Technical Information Service, Sprinfield, VA 22161, Oct. 1999
[5]. A J. Daemen and V. Rijmen , "AES Proposal: Rijndae l", AES Algorithm Submission, September 3, 1999