Journal of Computer Science of Newports Institute of Communications and Economics
Volume 5, Issue-2014, ISSN: 2226-3683

Implementation of Advanced Encryption Standard (AES) 192 Bit on FPGA

[1]MAHMOOD A. SIDDIQUI, [2]KANWAL ASLAM, [3]ADNAN AFROZ, [4]FAHEEM HAIDER RIZVI

[1,2,3,4]Department of Telecommunication Engineering,
Sir Syed University Of Engineering &Technology, Karachi

[1]mah.mood@live.com, [2]syedkanwal@msn.com,
[3]adnaaf@hotmail.com[4]faheem140@gmail.com

ABSTRACT

This paper presents an efficient hardware implementation of the Advanced Encryption Standard (AES) 192 bit Encryption in which we have used High Level Language (HLL) tool ISE Design Suite 14.4 Tool (Xilinx System Generator).In this work, System Generator approach on FPGA platforms has been used because it directly maps the design. The System Generator approach is best for Encryption. Our approach is better in terms of performance. Our proposed FPGA platform for the implementation of this work is Virtex-5vlx50t (ff1136) FPGA kit from Xilinx. We have used conventional blocks of Xilinx System Generator to get optimum performance in terms of speed. It operates on 100.251 MHz with throughput of 1.069 Gbps.

Keywords: *Encryption, AES, Throughput.*

1. Introduction

Security is considered as one of the main issue in today's computerized world. Cryptography is the science of converting plain or original text into cipher text so that it becomes unintelligible for unauthorized users [15]. Cryptography plays a vital role in terms of complete privacy, integrity and reliability of given information. Cryptography is largely used in online banking systems, cellular phones, computerized networks. Field Programmable Gate Arrays (FPGAs) are considered best for the implementation of cryptographic algorithms. This provides time and cost effective solutions as compared to Application Specific Integrated Circuit (ASIC).

Our proposed FPGA platform for the implementation of this work is Virtex-5vlx50t (ff1136) FPGA kit from Xilinx. The System Generator is an extension to simulink that maps the defined Xilinx block elements into architectures, entities, signals and ports.

2. Description of Advanced Encryption Standard

Due to the vulnerability of DES to Brute force attack, NIST started looking for the replacement of DES, which would be used to call Advanced Encryption Standard (AES) [1]. Different algorithms were considered but AES cryptographic algorithm designed by two Belgian Cryptographers Joan Daemen and Vincent Rijment named as 'Rijindael' algorithm was selected as Advanced Encryption Standard by NIST in October 2000 [2]. Later in December 2001, AES was published as FIPS 197 in Federal Register [3].

AES can support variable key lengths of 128,192 and 256 bits to encipher the data of 128 bits [4] [5].

In our proposed work we are designing AES-192 Encryption algorithm which comprises of twelve rounds and each round further consists of four transformations i.e. Sub Byte, Shift Row, Mix Column and Add Round Key except the last round where the Mix Column transformation is eliminated. But before the start of rounds there is initial round in which 128 bit key out of 192 bit is XORed with 128 bit data which is also called Add Round key as shown in Fig. 1. The 128 bits or 16 bytes of data is arranged in 4x4 bytes of array which is called State.

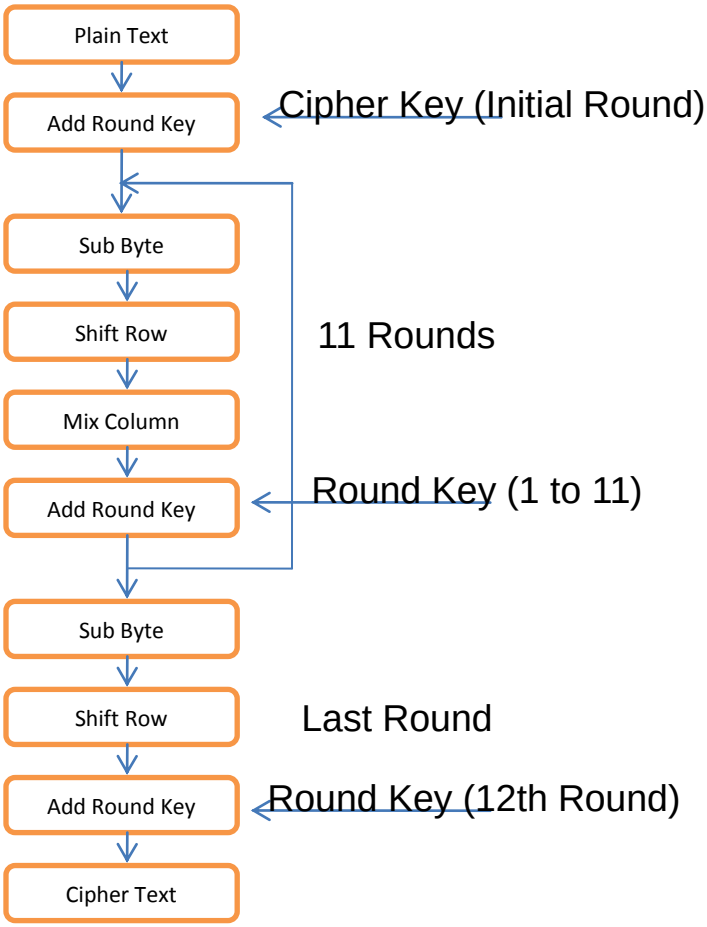

Figure 1: Advanced Encryption Standard 192 bit

2.1 Byte Substitution:

It is the first transformation of each round which is called Byte Substitution or Sub Bytes in which each byte of state is replaced with another byte as defined in S-box. It is basically nonlinear substitution of each byte of state [4]. S-box can be constructed by taking the multiplicative inverse of GF (2^8) and then by applying affine transformation. S-box consists of 256 values which are used to substitute each byte of state.

2.2 Shift Row:

Shift Row is the second transformation of each round of AES in which all the rows of state are cyclically shifted towards left except first row which is not shifted, second row is shifted one byte, third row is shifted two bytes and fourth row is shifted three bytes respectively [6].

2.3 Mix Columns:

Mix column is the third transformation performed in each round except last round because mix column is eliminated from last round of AES algorithm. Mix Column is basically column by column process in which each column is multiplied by a constant square matrix. Process of multiplication of bytes is performed within GF (2^8) field [7].

2.4 Add Round Key:

Add Round Key is the last round transformation or process in which keys that will be generated for each round will be Xored with state or data byte by byte.

2.5 Implementation

We have designed AES 192 bit Encryption algorithm on ISE Design Tool 14.4 (Xilinx System Generator).In our proposed work we have used pipelining architecture to implement AES 192 bit encryption functioning order to get better throughput as shown in fig3.It has been designed by using ISE Design tool 14.4(Xilinx System Generator) as shown in Fig. 2.

In the Pipelined architecture of AES 192 bit Encryption all the twelve rounds are designed separately and each is enclosed in a subsystem. Each subsystem contains four transformations which are SubByte, Shift row, Mixcolumn and Add Round Key excluding last round in which Mix Column is eliminated [8].

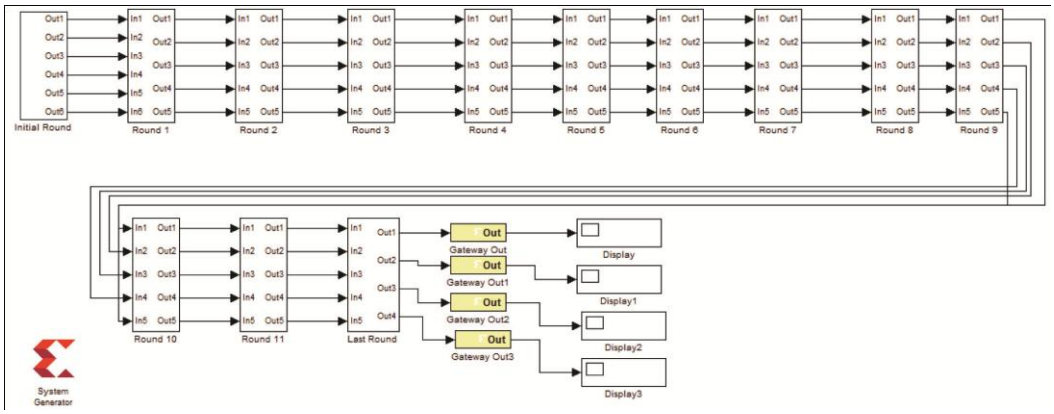

Figure: 2 Pipelined Architecture of AES 192 in System Generator

Here, four transformation and key expansion implementation has been given:
Sub Byte In our proposed work, we have used BRAM method to design Byte Substitution. First, the data of 128 bits is divided into four words of 32 bits or 4 bytes then each 32 bits value is further divided into four 8 bits value and provided

to the Dual port RAM block as shown in Figure. 3.
First we have extracted 8 bits data from 32 bits value by using bit basher block. The detail of 32 bit to 8 bits conversion is shown in fig. 4

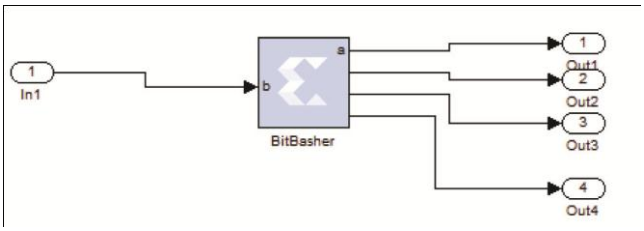

Figure 3: Byte Substitution

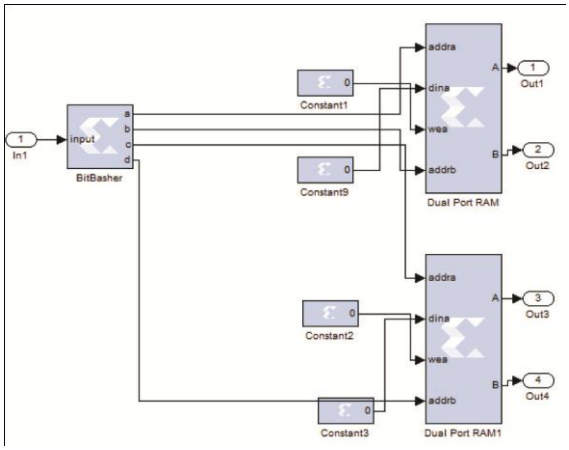

Figure 4: 32 bits into 8 bits Conversion

Two 8 bits value is given to each Dual port RAM input addresses. Basically, Single port RAM or Dual Port RAM Xilinx blocks can be used to store 256 look up values of S-box. But in our proposed work we have used Dual port RAM to make our resources utilization more efficient. The Dual-Port RAM is set as Block RAM and Distributed memory to access 8-bit lookup values corresponding to the 8-bit input addresses.

2.6 Shift Row

To implement Shift Row transformation, we have just rearranged the positions of output of Sub byte transformation according to the procedure followed in AES.

2.7 Mix Column

Matrices multiplication in Mix Column shifts and adds method has been used. To implement this method we used shift block to save the utilization cost. Data is passed directly through a connection wire when multiplied by number "1". We designed two multipliers to perform the multiplication between input data and the numbers "2" and "3" as shown in Fig. 5.

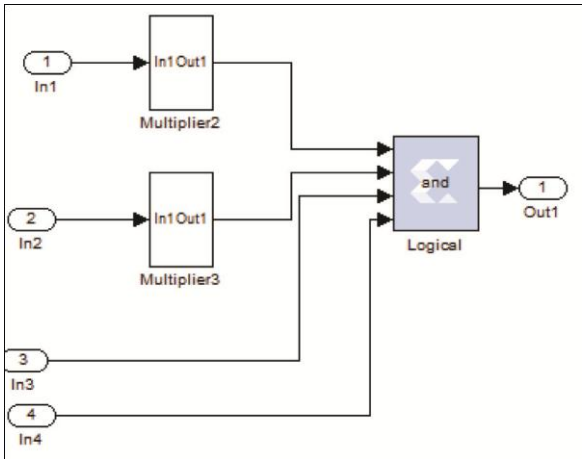

Figure 5: Mix Column

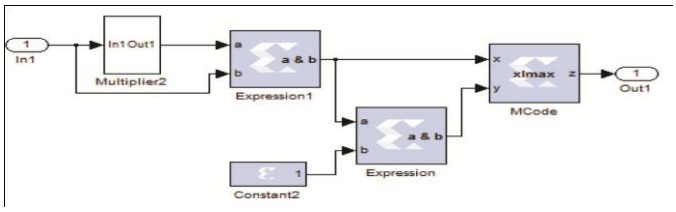

Figure 6: Multiplier 2

Multiplication by 2 is made by shift block where left shift is applied to the input data which results in the multiplication by number "2" as shown in Fig. 6.

Multiplier of 3 is designed by breaking the number '3" into (2+1) where multiplication of state matrix by number "2" is performed by single left shift as shown in fig 5 and the resultant number is the added with the state matrix by using XOR operation as shown in Fig.7.

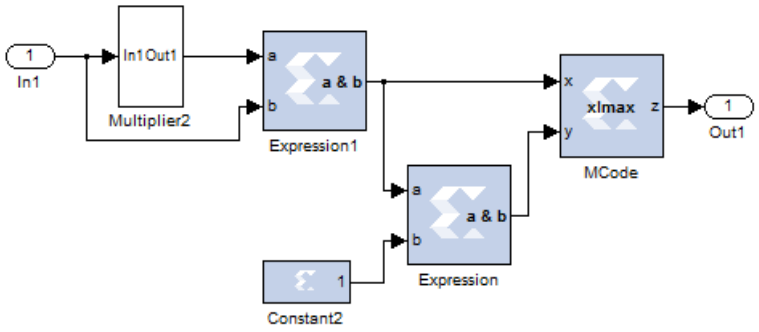

Figure 7: Multiplier 3

During multiplication of state matrix with number 2 and 3, the result exceeds from number 255, so to satisfy the condition of irreducible polynomial that the number < 255. The mod 27(dec) or mod 1b (hex) must be applied on the results of multiplication in order to get the number within the range of 255. Mcode block has programmed to define the condition of irreducible polynomial [9].

3. Key Expansion

In our work, we have generated the key for each round through key expansion process. In key expansion process of AES 192 bit Key is of 192 bit and N_k=6. First the key of 192 bit is divided into 6 words of 32 bits i.e. W_0 to W_5.First,The last 2 word has been taken as it is for the key of First round and last word W_5 has also gone under transformations to produce remaining key for the first round [10].The transformation applied on last word includes Rotate word, Substitute word, Add round constant and Xor between output of Add Round constant and $W_{[i-k]}$ as shown in fig 8.By Xor operation between recently generated key word and $W_{[i-k]}$,we can generate the required number of keywords for each round as shown in fig 8. Same process has used to generate the key for all rounds. Then, the generated key is Xored with the output of mix column byte by byte to generate cipher text. Therefore, each generated word of key is divided into 8 bits as shown in fig 8.

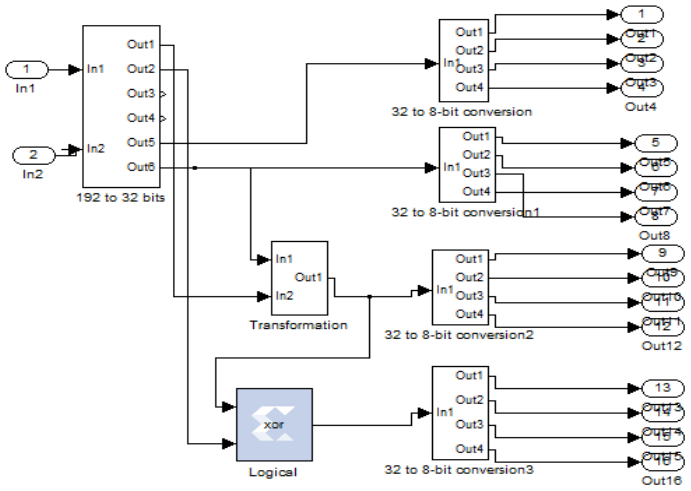

Figure 8:Key Generation Process

4. Implementation Results

AES 128,192 and 256 has been implemented on many different platforms including hardware, software and high level language tool. But as far as our implementation is concerned we are mainly focused on high level language tool [12].

Few implementations of AES by using high level language platform are as follows:

NCVHDL of Cadence [10], Handel C [11] , System C [12]. But in these implementations the algorithm or design has to be manually converted into HDL code so that it can be implemented on any Hardware platform. But in our proposed work, we have used High level language tool i.e ISE Design tool14.4 (Xilinx System Generator) that directly maps the design in HDL code and it has more flexibility as compared to other implementations [13].

AES 192 FPGA implementation of Anna Labbe [6] in which algorithm has been designed in NCVHDL of Cadence and then implemented on V1000BG560 Xilinx Virtex FPGA that operates at a frequency of 31.01MHz and offers a through put of 0.496Gbps. The designing of algorithm by using Handel C [14] which is implemented on Celoxica RC 1000 and it operates at a frequency of 74.4MHz and offers a through put of 7.76Mbits/sec.

Table 1 shows the comparison of implementation results of our work and other implementations of AES.

Table 1: Comparison of Implementation Results of AES

Implementation	Platform	Data Path	Frequency	Throughput
Anna Labbe [6]	NCVHDL	128	31.01 MHz	496.10 Mbits/sec
M.Mali [7]	Handel C	128	74.4 MHz	7.76 Mbits/sec
M.Askar [8]	System C	32	153Machinecycles	90 Mbits/sec
Our Implementation AES 192 bit	Xilinx System Generator	128	100.251 MHz	1.069 Gbps

In our implementation only 16 BRAM out of 60 has been utilized and 1280 slices out of 7200 has occupied. Operating frequency is 100.251 MHz and our implementation offers a throughput of 1.069 Gbps which is best among other implementations as shown in Table 1.As there is a tradeoff between Area and through put so our proposed work shows better results in terms of Area and throughput [13].

5. Conclusion

In this paper we have presented implementation of AES 192 Bit Encryption algorithm using Xilinx System Generator. Our implementation is better in terms of throughput. We have discussed implementation of AES on different High Level Language Tool but our implementation on Xilinx System Generator proved to be far better as System Generator approach directly maps the design in HDL code. In our work, the operating frequency is 100.251MHz and offering a throughput of 1.069 Gbps.

References

[1] Ashwini R. TondeȦ and Akshay P. DhandeȦ, (2014), Implementation of Advanced Encryption Standard (AES) Algorithm Based on FPGA, International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 - 5161 ©2014 INPRESSCO® , All Rights Reserved Available at http://inpressco.com/category/ijcet.

[2] Pravin B. Ghewari Mrs. Jaymala K. Patil Amit B. Chougule, (2010), Efficient Hardware Design and Implementation of AES Cryptosystem, International Journal of Engineering Science and Technology Vol. 2(3), pp. 213-219.

[3] J. Zambreno, D. Honbo, A. Choudhary, R.Simha, and B. Narahari, (2006) "High Performance Software Protection Using Reconfigurable Architectures", Proceedings of the IEEE, volume 94, No. 2, February.

[4] FIPS-197, (1999)"Federal Information Processing Standards Publication FIPS- 197,Advanced Encryption Standard (AES)", http://csrc.nist.gov/ publications/fips/fips197/fips-197.pdf, October.

[5] Behrouz A. Forouzan, (2007), "Cryptography and Network Security" Edition.

[6] Anna LabbÈ, Annie PÈrez. "AES Implementation on FPGA: Time Flexibility Tradeoff" Laboratoire L2MP-ICF (UMR CNRS 6137), IMT Technop Ùle de Ch,teau Gombert.

[7] M. Mali, F. Novak and A. Biasizzo, (2005) "Hardware Implementation of AES Algorithm", Journal of Electrical Engineering, vol. 56, pp 265–269.

[8] M.Askar and T. Egemen, (2007), "Design and System C Implementation of a Crypto Processor for AES and DES Algorithms", Information Security and Cryptology Conference with International Participation, Dec.

[9] Alia Arshad, Kanwal Aslam, Dure Shahwar, (2014) "FPGA Implementation Of AES using Xilinx System Generator" Asian journal of applied sciences, vol.2.

[10] DAEMEN, J.—RIJMEN, V.(1999) : AES Proposal: Rijndael, The Rijndael Block Cipher, AES Proposal, pp. 1–45, (http://csrc.nist.gov/CryptoToolkit/aes/).

[11] P. Aatheeswaran , Dr. R. Suresh Babu , FPGA Can be Implemented By Using Advanced Encryption Standard Algorithm, (2013), International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 1, January 2013 Copyright to IJAREEIE www.ijareeie.com.

[12] Samir El Adib and Naoufal Raissouni, (2012), AES Encryption Algorithm Hardware Implementation: Throughput and Area Comparison of 128, 192 and 256-bits Key, International Journal of Reconfigurable and Embedded Systems (IJRES), Vol. 1, No. 2, July 2012, pp. 67-74, ISSN: 2089-4864.

[13] Shylashree. N, Nagarjun Bhat and V. Shridhar, (2012), FPGA IMPLEMENTATIONS OF ADVANCED ENCRYPTION STANDARD: A SURVEY, International Journal of Advances in Engineering & Technology, May 2012. ©IJAET ISSN: 2231-1963.

[14] Ashwini R. Tonde , Akshay P. Dhande, **(2014),** REVIEW PAPER ON FPGA BASED IMPLEMENTATION OF ADVANCED ENCRYPTION STANDARD (AES) ALGORITHM, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, January.

[15] Rehan Shams, Fozia Hanif Khan, Umair Jillani and M. Umair, (2012), Introducing Primality Testing Algorithm with an Implementation on 64 bits RSA Encryption Using Verilog, SSU Res .J. of Engg. & Tech. Vol. 2. Issue 1, pp. 12-17.