IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 04, Issue 06 (June. 2014), ||V6|| PP 01-05

Implementation of AES Algorithm

Ankita S. Nampalliwar[1], Asst.Prof.Sheeja S. Suresh[2]
[1]Department of Electronics & telecommunication Engineering G. H. Raisoni Institute of Engineering and Technology for Women, Nagpur India
[2]Department of Electronics & telecommunication Engineering G. H. Raisoni Institute of Engineering and Technology for Women, Nagpur India

Abstract: - AES (Advanced Encryption Standard) could be a specification revealed by the American National Institute of Standards and Technology in 2001, as FIPS 197. AES describes a symmetric-key algorithmic rule that uses identical key for each encrypting and decrypting the information. The block size is of 128 bits and also the key sizes are often of 128, 192, or 256 bits. AES operates on a 4×4 matrix of bytes, referred to as the state. Some rounds of transformation converts the plaintext into the ultimate cipher-text. Six and the key size divided by thirty two rounds measure the amount of rounds needed i.e. one round reads the state into four 4-byte variables, transforms the variables, xor's them by a 16-byte round key once targeting a variable-length plaintext. The plaintext has got to be divided into separate cipher blocks and subsequently solely it are often encrypted underneath some mode of operation, usually by randomization that relies on an extra formatting vector. The cipher feedback (CFB) mode, output feedback (OFB) mode measure laid out in FIPS eighty one. The counter (CTR) mode is nominative by authority .The advantage of those modes is simply encryption algorithmic rule for each coding and decoding which can reduces the AES hardware value by five hundredth (no would like of secret writing hardware).

Keywords: - *AES, Encryption, Decryption, Pipelining, Verilog HDL.*

I. INTRODUCTION

The number of people and organizations victimization wide pc networks for private and skilled activities has recently inflated plenty. A crypto logic algorithmic rule is an important part in network security. A {widely known} crypto-graphic algorithmic rule is that the encryption normal (DES) that has been widely adopted in security product. However, serious issues arise for long security owing to the comparatively short key word length of solely fifty six bits and from the extremely no-hit cryptography attacks. In Nov 2001, the National Institute of Standards and Technology (NIST) of the US selected the AES algorithmic rule because the appropriate Advanced Encryption Standard (AES) to exchange the DES algorithmic rule. Since then, several hardware implementations are planned in literature a number of them use field programmable gate arrays (FPGA) and a few use application-specific integrated circuits (ASIC). The benefits of a package implementation embody simple use, ease of upgrade and movability. There are some limitations of package implementation because it offers solely restricted physical security, particularly with reference to key storage. Conversely, crypto logic algorithms (and their associated keys) enforced in hardware measure a lot of physically secure since they can't simply be scan or changed by an out of doors assaulter. The drawback of ancient (ASIC) hardware implementations is that the lack of flexibility with reference to algorithmic rule and parameter shift. Reconfigurable hardware devices like FPGAs measure a promising different for the implementation of block ciphers. FPGAs measure hardware devices whose perform aren't mounted and may be programmed in-system. During this paper, we tend to gift Associate in Nursing implementation of the AES block cipher with Virtex II professional FPGA using 0.13 millimeter and 90 nm method technologies. We've exploited the temporal correspondence offered within the AES algorithmic rule. The chip designed contains the 10 units. Every unit will execute one round of the algorithmic rule. 10 rounds of the algorithmic rule are measured in parallel in using external pipelined style. What is more, victimization internal pipelining and key exchange pipelining can facilitate to realize output in minimum period of time.

II. THE AES ALGORITHM AND PREVIOUS WORK

2.1. AES Algorithm.

The AES algorithmic rule could be a block cipher that processes information blocks of 128 bits employing a cipher key of 128, 192, or 256 bits length. Here every information block consists of a 4!4 array of bytes referred to as the state, on that the fundamental operations of the AES algorithmic are performed. Fig. one shows the AES coding and secret writing procedures.

The coding procedure is as follows:-

- After Associate in Nursing initial round key addition, a round perform consisting of 4 totally different transformations—byte-sub, shift-row, mix-column, and add-round-key—is applied to the information block within the coding procedure.
- The round perform is performed 10, 12, or 14 times, reckoning on the key length.
- The mix-column operation isn't applied to the last round.
- The computer memory unit-sub operation could be a nonlinear computer memory unit substitution that operates severally on every byte of the state employing a substitution table (S-Box).
- The shift-row operation could be a circular shifting on the rows of the state with totally different numbers of bytes (offsets).
- The mix-column operation mixes the bytes in every column by the multiplication of the state with a set polynomial modulo x4C1.
- Add-round-key operation is Associate in Nursing XOR that adds a round key to the state in every iteration, wherever the round keys square measure generated throughout the key growth section.
- The byte-sub transformation (S-Box operation), that consists of a inverse over GF(28) and an Associate affine remodel, is the most crucial part of the AES algorithmic rule in terms of procedure quality.
- However, the S-Box operation is needed for each coding and key growth.
- Conventionally, the coefficients of the S-Box and inverse S-Box square measure keep within the search tables, or a hard-wired increasing multiplicative inverter over GF (28) are often used, in conjunction with Associate in Nursing transformation circuit.
- The secret writing procedure of the AES is essentially the inverse of every transformation.
- However, the standard decryption procedure isn't clone of the coding procedure.
- That is, the sequence of transformations for decryption is totally different from that for coding although the shape of the key schedules for coding and decoding identical.
- There is, however, a similar version of the decryption procedure that has identical structure because the coding procedure.
- The equivalent version has identical sequence of transformations as the coding procedure (with transformations replaced by their inverses).

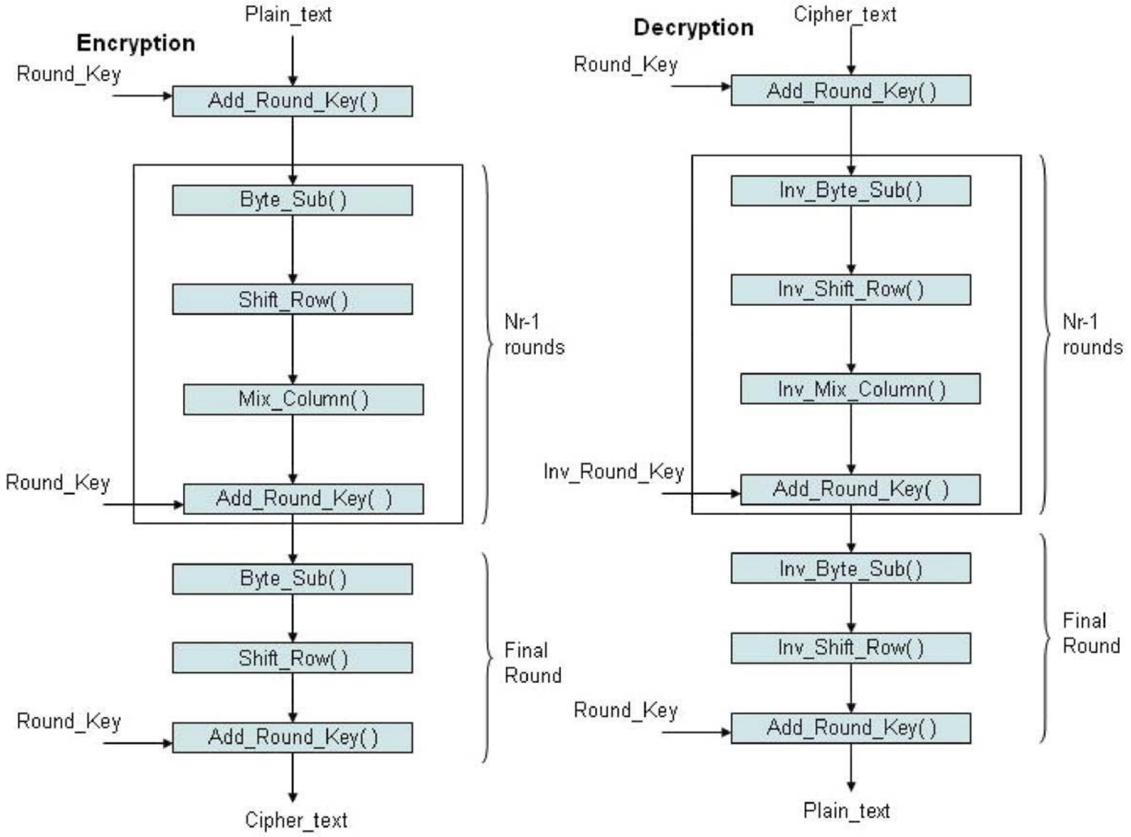


Fig 1. The AES algorithm (equivalent version) (Nr: 10, 12, or 14 depending on key length).

To attain this equivalence, a modification of key schedule is required. Additionally, two separate changes are required to bring decryption structure.

- The standard decryption round has the structure inv-shift-row, inv-byte-sub, add-round-key, and inv-mix-column.
- Thus, the primary 2 stages of the standard decryption round got to be interchanged, and also the second 2 stages of the decryption round got to be interchanged.
- The equivalent version of the decryption procedure is additionally shown in Fig. 1.

2.2. Previous Work.

There exist several displays of hardware implementations of AES algorithms in literature. A number of them are in short introduced here considering output. In 2001, Elbirt et al compared 5 candidate algorithms (including AES algorithm) for AES block cipher victimization FPGA implementations. Here, the throughputs of AES algorithmic rule were in 187.8 Mbps w1.94 Gbps. In 2003, several implementations are shown in literature. Verbauwhede et al conferred Associate in Nursing ASIC implementation underneath the through-put of two.29 Gbps. Su et al reduced hardware overhead of the S-Box by sixty fourth and also the output of their pipelined implementation victimization ASIC was a pair of.38 Gbps. McLoone and McCanny used look-up tables to implement the complete AES round perform underneath the output of twelve Gbps victimization FPGAs. In 2004, Hodjat and Verbauwhede’s FPGA implementation showed a high output of twenty one.54 Gbps employing a absolutely pipelined approach with inner-round pipelining and outer-round pipelining.

III. THE AES IMPLEMENTATION USING A FULLY PIPELINED DESIGN

3.1. Encryption Data path - Pipeline Design.

The goal of this implementation is to realize the very best potential output. We've used the bottom-up style approach, implementing the elementary operations 1st before coming up with the ultimate information path. A block primarily based commanding implementation of the look for coding is shown in Fig. 2. Round_1 through Round_10 represent the individual rounds within the AES-128 coding. The pipelining between every of the rounds can attain a high performance coding implementation. though implementing Associate in Nursing repetitive pipelining primarily based approach is one possibility, for clarity and ease, we've used a totally enlarged implementation for all 10 rounds. the information generated in every individual round is employed because the input for successive round. Exploiting the loop, level parallelism that the AES offers, we tend to found that, an easy 10 stage pipelining of the highest level with pipelined at the lower level blocks of the hierarchy is all that's required to realize high output. {this is this is often this are often} in an exceedingly}ll|one amongst|one in every of} the best strategies wherever high performance can be achieved in a very lowest quantity of your time therefore reducing the general style implementation cycle. In fig 2 , there's a pipeline stage between every round, i.e. the look is absolutely pipelined. However, our internal (inside every round) pipeline style is totally different .In every round, our style has 3 pipeline stages, one in real time once byte-sub operation, one simply once shift-row operation, and also the last simply before information output. In every round, the look have four or seven pipeline stages, one once a byte-sub operation and 3 or six in a very byte-sub operation. additionally, our style has one pipeline stage (before XOR operation between round Key Block) in key generation blocks Internally, the key growth block renders itself as a pipelined implementation between every of the key creations from Key1 through Key10. this can be mechanically realised by the parallel nature of the look. These extra pipelines build it potential for our implementation to get the next output.

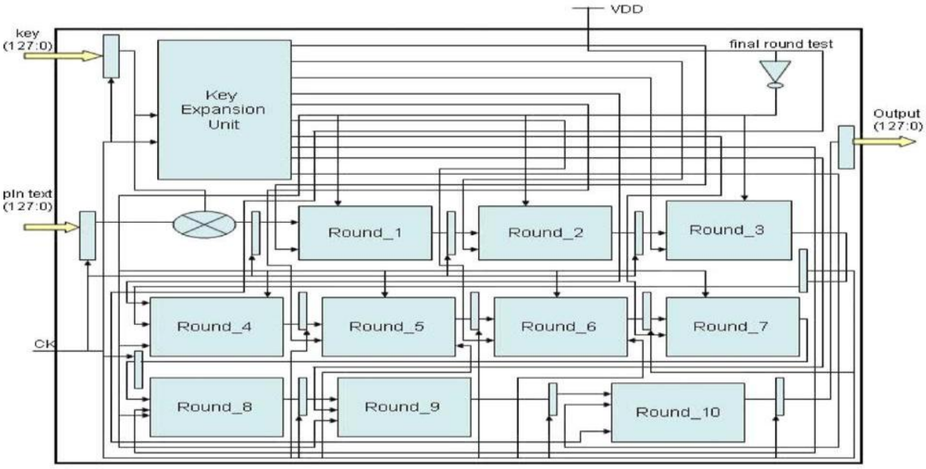


Fig 2. A pipelined AES and AES-128 encryption implementation.

3.2. Round Key Generation.

For our implementation, we've chosen to use a class-conscious synchronic key generation methodology. This can be similar in approach to the fly key generation technique. However, there's internal sub-pipelining for every of the sub-stages of the key creation. This is able to produce the key for one round. The XST FPGA synthesis surroundings are ready to mechanically acknowledge constant logic and assign constant logic '0' or '1' to the acceptable 'Tie' cells from the library. The output is that the key for successive round. we've enforced the round key generation as an easy state table substitution of the thirty two bits of the input key and thereby implementing XOR operations for manufacturing the enlarged 128 bits round key. Victimization internal pipelining would greatly scale back the minimum clock amount required to assure the right practicality of round key generation. Hence, implementing a very quick block during this preliminary stage of the look itself is extremely a lot of potential. most post synthesis clock frequency of 478.9 megahertz has been for key generation, with this implementation. putting the pipeline registers and loading the information victimization input/output registers (IR/OR) is that the key to achieving high performance. IR is Associate in nursing input register accustomed load the input file. PR is that the pipeline register used with intermediate processing and OR is that the output register. It are often inferred that it takes 3 clock cycles for the output to seem from key in to identify. What this is able to mean is that, for every of the 10 rounds, to supply the key in every serial round, would need a minimum of 3 clock cycles. The round Keys Block merely uses the round Key Block delineate antecedent to form the round keys for all the individual rounds. Hence, for 128 bits data/key coding, it uses 10 instances of the round Key Block mentioned higher than to form all the 10 round keys. within this block, really every of the keys from Key1 to Key10 is made victimization the key growth algorithmic rule. during this implementation, every round key's created by instantiating the round Key Block. Any good parallel hardware design offers naturally high performance. Exploiting this idea, we've used internal pipelining among every of the round key creation stages. The employment of balanced internal pipelining in between stages of a parallel design helps in reducing the flip-flop to flip-flop clock delay. As a result, it maximizes the performance of a style whereas guaranteeing minimum clock speed.

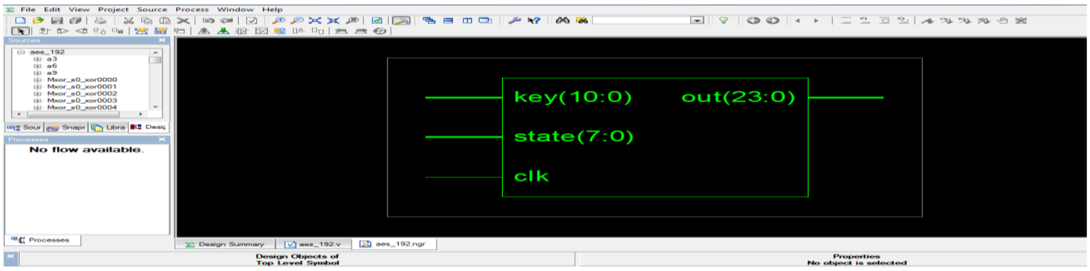
IV. CONCLUSION

In this paper we tend to confer a hardware implementation increasing output for AES coding algorithmic rule. By using an efficient inter-round and intra-round pipeline design, we get the 20 clock cycles in 1 sec and hence we achieved a throughput much higher than any other implementations reported in the literature.

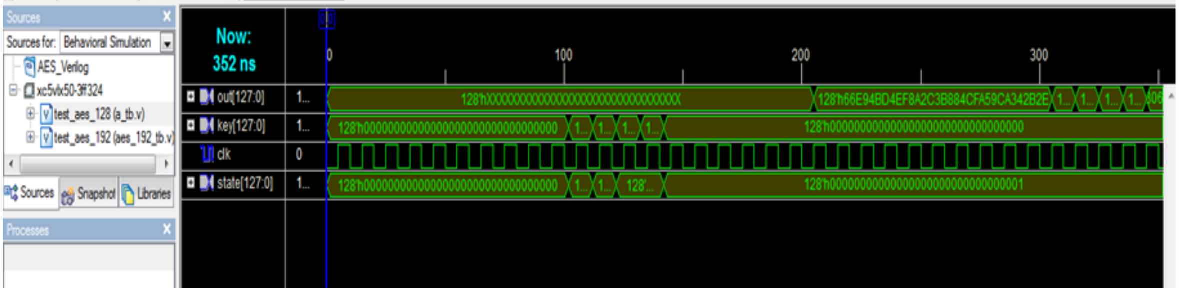
V. RESULT

The 192bits and eight bit keys are designed and enforced in Xilinx. The output of each the codes and RTL read square measure specifically same. Therefore the coming up with and implementation of AES algorithmic rule is completed.

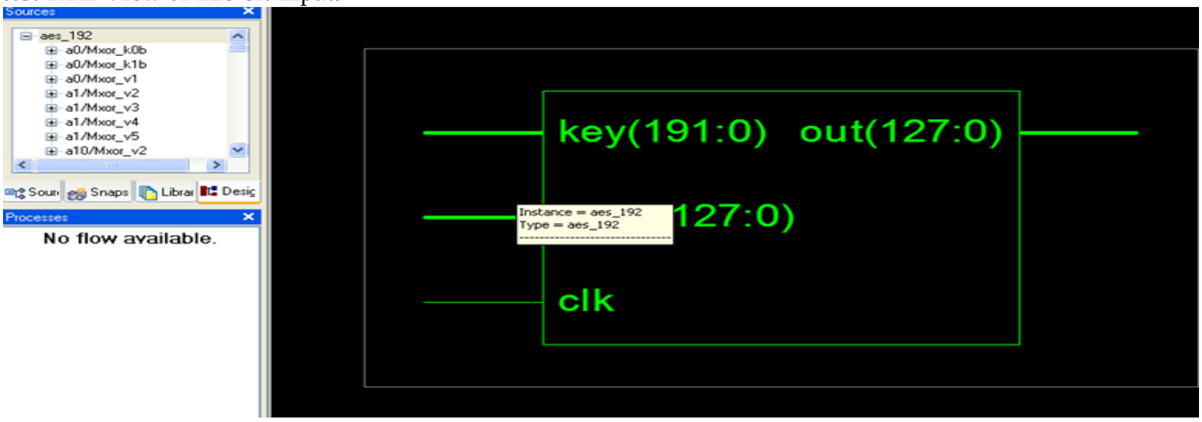
6.1 RTL View for 8 bit Input.



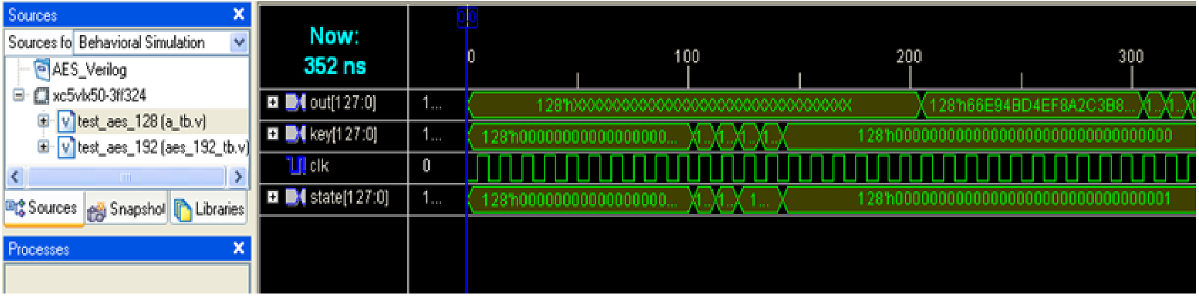
6.2 Simulation of 8 bit Input.


Sources
Sources for: Behavioral Simulation
AES_Verilog
xc5vlx50-3ff324
test_aes_128 (a_tb.v)
test_aes_192 (aes_192_tb.v)
Sources
Snapshot
Libraries
Processes
Now:
352 ns
0
100
200
300
out[127:0]
1...
128'hXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
128'h66E94BD4EF8A2C3B884CFA59CA342B2E
key[127:0]
1...
128'h00000000000000000000000000000000
128'h00000000000000000000000000000000
clk
0
state[127:0]
1...
128'h00000000000000000000000000000000
128'...
128'h00000000000000000000000000000001

6.3. RTL View of 128 bit Input.



6.4. Simulation of 128 bit Input.


Sources
Sources fo Behavioral Simulation
AES_Verilog
xc5vlx50-3ff324
test_aes_128 (a_tb.v)
test_aes_192 (aes_192_tb.v)
Sources
Snapshot
Libraries
Processes
Now:
352 ns
0
100
200
300
out[127:0]
key[127:0]
clk
state[127:0]
128'hXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
128'h66E94BD4EF8A2C3B8...
128'h000000000000000000...
128'h00000000000000000000000000000000
128'h000000000000000000...
128'h00000000000000000000000000000001

REFERENCES

[1] National Institute of Standards and Technology (US), Advanced Encryption Standard, http://csrc.nist.gov/publication/drafts/dfips- AES.pdf.

[2] C.P. Su, T.F. Lin, C.T. Huang, C.W. Wu, A high-throughput low-cost AES processor, IEEE Commun. Mag. 42 (12) (2003) 86–91.

[3] M. McLoone, J.V. McCanny, AES FPGA implementations utilizing look-up tables, J. VLSI Signal Process. Syst. 34 (3) (2003) 261–275.

[4] F.X. Standaert, G. Rouvroy, J.J. Quisquater, J.D. Legat, Efficient implementation of AES encryption in reconfigurable hardware: improvements and design tradeoffs, in: CHES 2003, LNCS 2779, 334–350.

[5] G.P. Saggese, A. Mazzeo, N. Mazzocca, A.G.M. Strollo, An FPGA-based performance analysis of the unrolling, tiling, and pipelining of the AES algorithm, in: FPL 2003, LNCS 2778, pp. 292–302.

[6] K. Jarvinen, M. Tommiska, J. Skytta, A fully pipelined memoryless 17.8 Gbps AES-128 encryptor, in: International Symposium on Field Programmable Gate Arrays, 2003, pp. 207–215.

[7] A. Hodjat, I. Verbauwhede, A 21.54 Gbits/s fully pipelined AES processor on FPGA, in: IEEE Symposium on Field-Programmable Custom Computing Machines, 2004.

[8] A.J. Elbirt, W. Yip, B. Chetwynd, C. Paar, An FPGA-based performance evaluation of the AES block cipher candidate algorithm